

## **COLLECTING, SUMMARIZING AND ANALYZING HARVEST DATA**

*A step by step guide to estimating annual harvest of migratory birds*

Unless otherwise specified, you may not reproduce materials in this publication, in whole or in part, for the purposes of commercial redistribution without prior written permission from Environment and Climate Change Canada's copyright administrator. To obtain permission to reproduce Government of Canada materials for commercial purposes, apply for Crown Copyright Clearance by contacting:

Environment and Climate Change Canada  
Public Inquiries Centre  
7th Floor, Fontaine Building  
200 Sacré-Coeur Boulevard  
Gatineau QC K1A 0H3  
Telephone: 819-997-2800  
Toll Free: 1-800-668-6767 (in Canada only)  
Email: [ec.enviroinfo.ec@canada.ca](mailto:ec.enviroinfo.ec@canada.ca)  
Cover photo: © Environment Canada  
Photos: © Environment and Climate Change Canada

© Her Majesty the Queen in Right of Canada, represented by the Minister of Environment and Climate Change, 2017

Aussi disponible en français

# **COLLECTING, SUMMARIZING, AND ANALYZING HARVEST DATA**

*A step by step guide to estimating annual harvest  
of migratory birds*

**January 2017**

# TABLE OF CONTENTS

<b>Introduction</b>	4
<b>Sunlight and Pocket Gophers</b>	7
<i>The Science and Art of Answering Questions with Data</i>	
Harvest Data	9
Sources of Bias in Harvest Data	12
<b>The Importance of Data Collection Protocols</b>	15
Addressing Reporting Bias	16
<b>Data Entry and Management in Excel</b>	19
Tools and Tricks	20
Checking for Errors	23
Creating a Working Dataset	25
<b>Summarizing and Visualizing Patterns in Your Data</b>	32
Summarizing Predictor Variables	32
Summarizing Response Variables	36
visualizing Summary Data Using Excel	46
<b>Getting Started with R</b>	58
What is R and Why Should You Use It?	58
Downloading R	58
Understanding How R Works	59
Importing, Exporting, and Managing Data	68
Graphing in R	76
Using Scripts	79
<b>Statistical Modelling of Your Data</b>	83
How Do We Measure Certainty?	83
Probability	96
Hypothesis Testing and P Values	115
<b>Linear Regression</b>	132
<b>Generalized Linear Models</b>	145
Poisson Versus Negative Binomial Sample	145
Distributions	145
Maximum Likelihood Estimation	148
Model Interpretation	154



Off-Setting	156
Quadratic Effects	156
AIC	159
Measuring Model Fit	161
Dealing with Over-Dispersion	163
Interactions	164
Categorical Predictors	166
<b>Putting It All Together</b>	<b>170</b>
<i>Modelling Harvest Data Using Poisson and Negative Binomial GLM</i>	
Visualizing an Interaction Effect	174
The Negative Binomial Model	175
Predicting Total Harvest	178
<b>Appendix 1</b>	<b>186</b>
<i>Example Harvest Survey and Instructions</i>	
<b>Appendix 2</b>	<b>189</b>
<i>Mapping Your Data</i>	
<b>Appendix 3</b>	<b>196</b>
<i>Creating a Map Using ArcGIS</i>	
Getting Started	196
Setting Up Your Map	197
Navigating Around Maps	199
Drawing on Maps	202
Creating and Saving a Map	204
Adding Points	206
Labels and Symbols	208
<b>Appendix 4</b>	<b>209</b>
<i>Glossary</i>	
<b>Bibliography</b>	<b>212</b>
<b>Index</b>	<b>213</b>

# I N T R O D U C T I O N

This guidance document was written to help you collect, summarize, visualize, and analyze data on harvesting of migratory game birds. To use this guide, you are not required to have had any prior experience working with data. All that is required of you is motivation to learn, and maybe a little patience.

Collecting harvest data to estimate the annual take of migratory birds is, in a word, research. This document is organized according to the steps you need to take as a researcher conducting a study. We start with getting acquainted with some fundamental terms. We brush up against the philosophical problem inherent to research – that is, how do we know that our research is telling us the truth? We then move onto the importance of carefully considering what we're actually trying to measure, and how best to take measurements. We overview the importance of creating a data collection protocol, and of regular communication with our data collectors – the hunters - to avoid biases and errors. Next we overview the basics of data summary and visualization using Microsoft Excel. The rest of the document works through the process from data summary to analysis and interpretation using example data sets of harvest data, provided on the CD accompanying this document.

Four computer programs are used in this guide. Microsoft Excel is used for data summary and



# INTRODUCTION

visualization, R is used for statistical analysis, and GoogleEarth and ArcGIS are used for mapping. This guide was written assuming you have no prior experience working with these programs. You will need to work through Appendix 3 Getting Started with R, before you work through the sections on statistical analysis.

A considerable amount of statistical theory has been distilled into this guidance document. If you find some of the concepts difficult to grasp at first, know that you are not alone. Learning statistics is learning a new language – it takes time and patience and a lot of practice. Almost all of the background mathematical theory has been left out – if you are interested, you will find good translations of the math theory, in addition to much more thorough statistical theory and tools, in the books listed in the bibliography. You don't need to be a mathematician to be a good researcher. However, it will help you to be able to visualize statistical tests, to really understand how the tests are helping you as a researcher find the truth.





# SUNLIGHT AND POCKET GOPHERS

## *the Science and Art of Answering Questions with Data*

**Data** is information. We use data to answer questions. When we ask questions, we want truthful answers.

We almost never have data on everything. Statistical analysis is the practice of deriving truth from a part of something we define as everything. In statistical terms, everything is referred to as the **population**, and the part that we observe to say something truthful about the whole, is referred to as the **sample**.

Imagine you're standing in front of a field of flowers, and your job as a scientist is to find out what proportion of the flowers in the field are blue. Everything in this case - the population - is the whole collection of every single flower in the field. Intuitively, you can imagine that you do not need to count and record the colour of every single flower in the field to answer your question. Instead of counting the whole population, you take a sample.

To ensure that flower counting will be manageable, you decide that your **sample unit** will be a 2 x 2 foot square area. A sample unit is the unit of observation used to answer the question.

In your sample unit of an area of 2 x 2 feet, you count 12 flowers, and 3 of these are blue. The proportion of blue flowers in your sample is  $3/12 = 25\%$ . You conclude that the proportion of blue flowers in the entire field is about 25%. **We practise the science of statistics when we use data collected from samples to estimate the truth about an entire population.**

But we must always remember that our efforts do not result in the real truth; the best we can ever do is to **estimate** truth. Your estimate of the proportion of blue flowers in the field would be more accurate if you used more sample units to derive the estimate. **Sample size** refers to the number of sample units used to estimate truth. Sample size is a key issue in statistical analysis. The higher the sample size used to represent the population, the more accurate the estimate of truth.

Because we can only estimate truth, as good scientists we are obligated to imagine all the ways we could be wrong. This where science becomes something of an art, because deriving truth from data requires us to think imaginatively about the world. For example, what if there were no blue flowers in the back half of the field, because a blue-flower-eating pocket gopher lived there? What if the blue-flowered plants in

## SUNLIGHT AND POCKET GOPHERS

### *the Science and Art of Answering Questions with Data*

this field love sunlight and thus grow better along the edges of the field?

If we imagine that these conditions were actually true, then your estimate of 25% blue flowers was wrong; the true proportion was much lower. You happened to count the blue flowers where there were many of them, and thus you over-estimated the true proportion. Gasp.

Because the distribution of blue flowers varied across the field due to sunlight and a pocket gopher, the data that you collected at the perimeter of the front half of the field was **biased**. Bias refers to systematic inaccuracy – that is, biased data is sample data that does not represent the population because of certain attributes of the sample. In this case, bias resulted from the positioning of the sample within the field.

To avoid collecting biased data, you need to count flowers at random locations across the whole field. In other words, you need to take a **random sample**. For example, you might mark out a grid of 2 x 2 foot squares across the field, resulting in 552 squares. You decide you want a sample size of 100, which is about 18% of the population (i.e. the field). You assign a number to each square, write numbers from 1 to 100 on individual pieces of paper, put them in a hat, shake it, and withdraw 100 numbers. You would



## SUNLIGHT AND POCKET GOPHERS

*the Science and Art of Answering Questions with Data*

then go count flowers in each numbered square, and the result would be a random sample, and the sample size would be 100.

You need to consider one more important characteristic of your **sampling design** to ensure the data are not biased. Sample units must be **independent** of one another. Imagine that two of the 2 x 2 randomly chosen squares were adjacent to one another. You count flowers in each square in the morning when there was no wind, and then you count flowers again in the afternoon when the wind is blowing from the east. You don't count the same number of blue flowers in each square in the morning as in the afternoon.

What could have happened? The wind blew the blue flowers that grew on the border between the two squares into the western square in the afternoon. Because what happened in one square was related to what happened in the other square, in statistical terms, these two sampling units were not independent. Non-independence of sampling units results in inaccurate estimation.



## HARVEST DATA

Now let's apply these terms to the questions you're interested in and the types of data you've collected. The main question you want to know from the data your organization collected from hunters is:

“What was the total number of birds of each group (e.g. ducks, geese, ptarmigan, etc.) harvested in each year?”

You may find it useful to ask other questions, such as, how do harvest levels differ across areas, across seasons, or between age groups of hunters?

The **population** for these questions is the total number of hunters within your organization that

# SUNLIGHT AND POCKET GOPHERS

## *the Science and Art of Answering Questions with Data*

harvested migratory birds in a given year. The **sample** is the collection of hunters that provided information on their harvest, and the sample size is the number of hunters that provided information. The **sample unit** is a hunter.

The simplest way to answer the question is by calculating the average number of birds harvested per hunter and then multiplying by the total number of hunters.

$$\text{Total estimated number of birds harvested} = \text{Average number of birds harvested} \times \text{Total number of hunters}$$

The table below shows the harvest success of 15 duck hunters, 9 of which responded to a harvest survey and provided information on how many ducks they harvested. In this scenario, there are only 15 hunters in the whole region for which harvest levels are being estimated. In reality, you likely must estimate the total number of hunters in your region. **As you'll see below, either the actual or estimated total number of hunters is essential to estimate the total annual harvest.**

*Table 1. Harvest success of 15 duck hunters, 9 of which responded to the harvest survey*

Hunter ID	Responded to Survey?	Reported Number of Ducks Killed
X02	Yes	12
Q30	Yes	12
N30	Yes	3
M90	Yes	22
B77	Yes	16
N10	Yes	3
R45	Yes	0
E33	Yes	7
A22	Yes	20
Total		95
P12	No	Unknown
U38	No	Unknown
G23	No	Unknown
T67	No	Unknown
Y09	No	Unknown
D88	No	Unknown

## SUNLIGHT AND POCKET GOPHERS

*the Science and Art of Answering Questions with Data*

Applying statistical terms, the population of hunters in this scenario is 15, and the sample size of hunters is 9, because 9 hunters answered the survey and provided data.

Now let's apply the simple formula to calculate the total number of birds harvested.

The average number of ducks killed per hunter of the sample of 9 hunters is  $95/9 = 10.6$ . The total number of hunters is 15. And the total estimated number of ducks harvested =  $10.6 \times 15 = 158.3$ .

Now let's imagine that all hunters had responded.

*Table 2. Harvest success of 15 duck hunters, all of which responded to the harvest survey*

Hunter ID	Responded to Survey?	Reported Number of Ducks Killed
X02	Yes	12
Q30	Yes	12
N30	Yes	3
M90	Yes	22
B77	Yes	16
N10	Yes	3
R45	Yes	0
E33	Yes	7
A22	Yes	20
P12	Yes	15
U38	Yes	10
G23	Yes	7
T67	Yes	5
Y09	Yes	12
D88	Yes	8
Total		152

Note that the total number of ducks harvested is very close to the estimate of 158 that we calculated using the simple formula.

In fact, if we took any **random sample** of 9 hunters from the total population of 15 hunters, we would arrive at similar estimates.

## SUNLIGHT AND POCKET GOPHERS

### *the Science and Art of Answering Questions with Data*

Let's try. Throw these Hunter Identification numbers into a hat, pull out 9, calculate the average, and multiply the average by 15.

Example of random sample of 9 hunters:

$$\text{Average} = 12 + 3 + 16 + 3 + 20 + 10 + 7 + 12 / 8 = 10.4$$

$$\text{Sample average} \times \text{Total number of hunters} = 10.4 \times 15 = \mathbf{156}$$

## SOURCES OF BIAS IN HARVEST DATA

Now, just as with the blue flower example, we need to consider how we could be wrong if we answer our questions from sample data. One possibility is that hunters who felt unsuccessful in their harvesting do not respond to harvest surveys. Unsuccessful hunters are those that went out hunting but did not harvest any birds, or, hunters that harvested birds but not as many as they'd hoped to harvest. Perhaps they didn't feel that one or two birds was worth reporting. This is referred to as **non-response bias**. Let's observe the effect of such non-response bias on our estimate of the total number of ducks harvested.

Imagine that only successful hunters responded to our harvest survey. Refer to [Table 2](#) on previous page.

$$\text{Example: Sample average} = 12 + 12 + 22 + 16 + 20 + 15 + 10 + 12 / 8 = 14.9$$

$$\text{Sample average} \times \text{Total number of hunters} = 14.9 \times 15 = \mathbf{223}$$

Non-response bias resulted in an over-estimate of the true number of ducks harvested by the total population of 15 hunters. This was because hunters that did not harvest many ducks did not respond to the survey. The numbers of ducks killed by successful hunters was used to represent the whole population of hunters, which included successful and not so successful hunters.

**It is clear that the sample of hunters must be a random sample; otherwise, the estimated total harvest of birds will be wrong.**

Now let's investigate the effect of **non-independence of sample units** on the estimated total harvest of

## SUNLIGHT AND POCKET GOPHERS

### *the Science and Art of Answering Questions with Data*

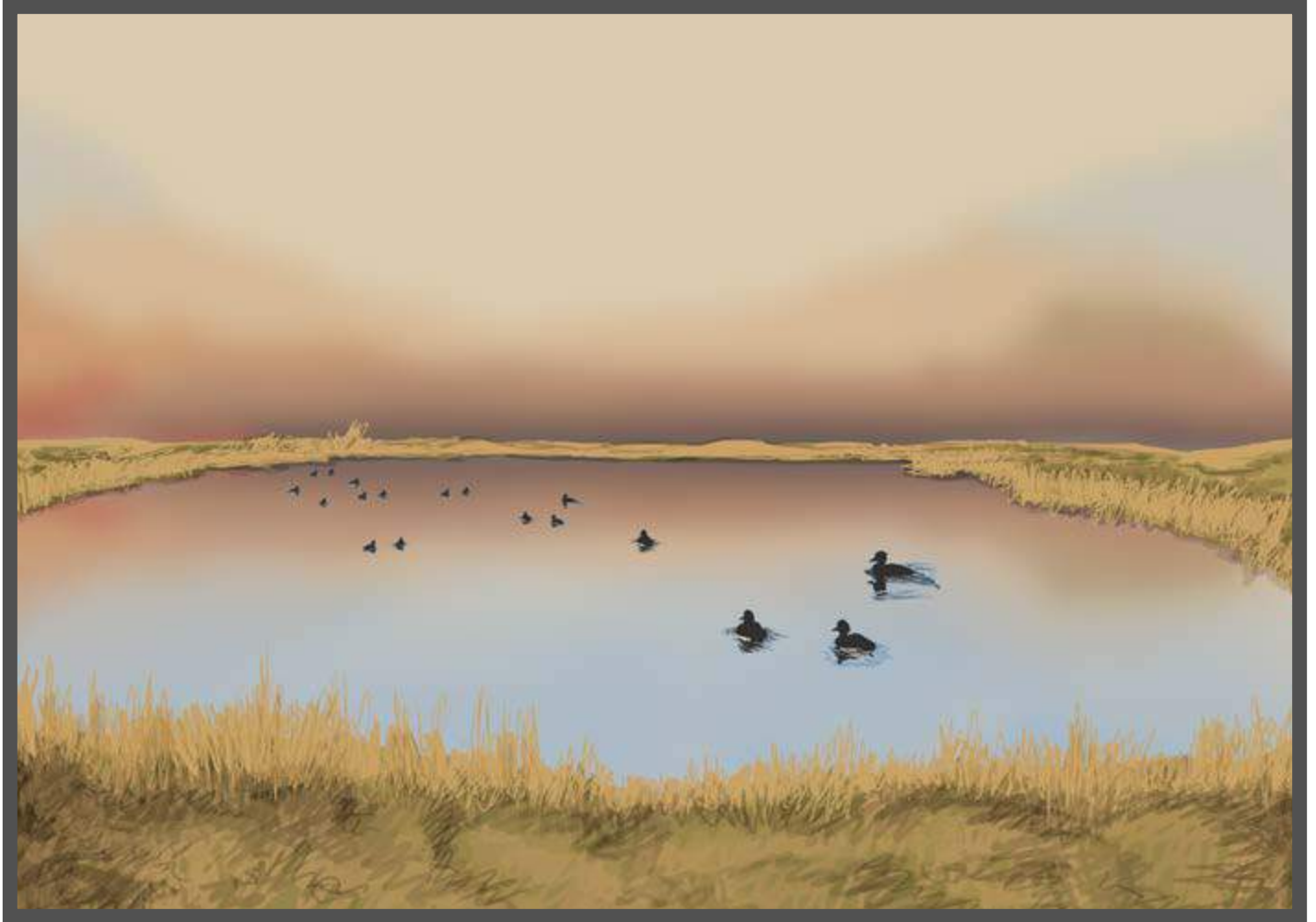
birds. Recall that the sample unit is a hunter. Suppose that one hunter decided to respond to the survey three times with his duck kills for the same area and time frame. The duck kills from the same hunter were entered as though the kills were made by three different hunters - Q30, B77, and P12. Refer to [Table 2](#) on previous page. These counts of duck kills are not independent of one another, because they were kills made by the same hunter.

If we were unaware that Q30, B77, and P12 were indeed the same hunter, the estimated total harvest of ducks from a sample size of 10 hunters who responded to the survey would be  $110 / 10 * 15 = 165$ .

Now suppose the hunter called to inform us that he had responded to the survey three times, and that he had actually killed 16 ducks.

The independent data set, now with a new ID number assigned to that hunter results in an estimated total harvest of ducks of  $83 / 8 * 13 = 135$ . Note that the number of hunters is now 13 and not 15, because the non-independent entries are deleted. Also note that the sample size is 8 hunters.

Ensuring that our data set only included independent data resulted in a different estimate of the total number of ducks harvested (135 versus 165). Since 125 is the true number of ducks harvested, the independent data set resulted in a much more accurate estimate.





# THE IMPORTANCE OF DATA COLLECTION PROTOCOLS

One of the most important tools in your tool box as a researcher trying to gather the truth from data is a **data collection protocol**. A data collection protocol is essentially a recipe book that details your research goals and questions, exactly what data you collected, exactly how you collected it, issues you encountered while collecting the data, and the decisions you made to deal with these issues.

In the absence of a data collection protocol, many research efforts have gone seriously awry. Suppose for example that you conducted a harvest survey and asked hunters to report on their effort. But you didn't define exactly what you meant by effort. One hunter correctly reported just the days she spent in the field hunting. Another hunter reported the days he spent hunting, plus all the days he spent preparing to hunt, including travel time to get to his hunting location. Clearly in this case you would be collecting inaccurate data.

Without first defining for yourself exactly what effort means, you will be more likely to neglect to inform your study participants what you expect them to report for their hunting effort. A data collection protocol forces you as a researcher to think carefully and thoroughly before you conduct research. This avoids expensive mistakes and helps in highlighting potential biases and ways to deal with them. A key component of a data collection protocol is the data sheet you will be using to collect data, along with a very detailed description of each entry on the data sheet. An example data sheet is provided in Appendix 1, and a detailed description of each entry is provided in the table below. In our case, hunters are our data collectors, and thus it is very important that hunters be sent very clear instructions along with the data sheet ([Appendix 1](#)).

A data collection protocol also helps to identify potential pitfalls and ways to avoid them. Aside from reporting bias, there are at least three major pitfalls that must be avoided when conducting a harvest survey:

1. Hunters believing that they don't need to report unsuccessful hunting trips. That would mean that your data would contain no zeros, and your data would be for successful hunters only. **It may be that a large portion of hunters are unsuccessful in some years due to environmental conditions. If these hunters failed to report on their effort, you would overestimate the harvest. But equally important, you would not be able to rigorously test for environmental effects of, for example, climate change**

## THE IMPORTANCE OF DATA COLLECTION PROTOCOLS

**and development pressure on hunting success.** This pitfall is easily overcome simply by stating clearly in the instructions to hunters that they need to report on their hunting effort even if they were not successful.

2. Inability of hunters to identify the common names of the species they harvested. This is easily overcome simply by sending along a photo identification guide with common names of species.

3. Hunters reporting a confusing array of hunting locations that you are not able to pinpoint on a map. You can avoid this by asking hunters to report coordinates, or, simply by providing them with a sub-regional map, and asking them to mark their hunting locations on the map as accurately as possible.

A general rule of thumb is to collect data at the ‘highest resolution’ possible, while still collecting data efficiently. A data collection protocol helps to separate core data needs from unnecessary data. This is especially important when conducting research based on a voluntary survey. The ideal survey is a careful balance of being quick and easy to fill out and return, while also collecting all the necessary information. An example of this careful balance on the example data sheet is the entry for month. Asking hunters to report the month of harvest is better than asking them to report season, because the definition of seasons may vary across regions. For example, a hunter on the southwest coast of BC where there is very little snow might define fall as September to December, while hunters in the rest of the province would likely define fall as September to November. ‘Month’ presents a higher resolution of data than ‘season’.

## ADDRESSING REPORTING BIAS

Collecting harvest data depends on hunters voluntarily, and accurately, reporting their harvests. One of the key issues with a harvest survey is non-response, as we have investigated. Addressing non-response bias is thus one of the most important aspects of collecting reliable harvest data. There are two ways to address it. The first is to encourage higher reporting rates. For example, the US Fish and Wildlife Service conducts an annual harvest survey for migratory birds <http://www.fws.gov/birds/surveys-and-data/harvest-surveys/diary-surveys.php>. They have achieved a 50% response rate from approximately 3.5 million hunters in the US every year! Their method depends on sending out repeat reminders to hunters

## THE IMPORTANCE OF DATA COLLECTION PROTOCOLS

---

to fill out survey forms. They send out the initial survey package (data sheet, instructions, species ID sheets, etc) at the start of the hunting season so that hunters can record their take as they hunt. This avoids memory bias, resulting from hunters forgetting their harvest and thus reporting inaccurately. A reminder is then sent to all hunters at the end of the hunting season to fill out the harvest survey, then a month later, another reminder to hunters that haven't yet responded. A third reminder is sent a month later to the remaining non-responders.

Another way to address non-response bias is to measure it, which requires conducting a follow-up phone survey to ask non-responding hunters the same information on the data sheets. This is labour intensive, and it may be more fruitful to put the added effort into increasing the response rate.



# DATA ENTRY AND MANAGEMENT IN EXCEL

So your organization has conducted a harvest survey, and you now have a stack of data sheets in front of you. Now what?

The first step is to store the data in a way that the information is very accessible. In short, you need to create a database. The easiest program for this is Microsoft Excel.

In essence, a database is a digital copy of the data sheets.

Open the excel file named <example\_harvest\_data.xls> provided on the CD in the back page of this document.

Excel can be used to store, organize, and summarize data, which all together are called **data management**. Excel stores data in 'spreadsheets'. Different spreadsheets within the same excel document usually contain different subsets of the same kinds of data. The basic layout of a spreadsheet is a very large number of columns labelled with letters, and a very large number of rows, labelled with numbers.

You'll see two tabs at the bottom left hand corner – one tab is named '2015'. If you click on '2015 NB', you'll see that you've now entered a different spreadsheet. You can add spreadsheets to your file simply by pressing the + button next to the spreadsheet tabs. And you can rename each spreadsheet by right clicking on a tab, and then scrolling to 'rename'.

Excel is used is to create a table of information, with each column containing a different set of information. Usually, the first row is used to label the columns, and often it is helpful to bold the typeface to make the labels stand out more.

Compare the example data sheet in Appendix 1 with the '2015' spreadsheet. Note that the column titles match the entries on the data sheet, and follow one another in roughly the same order as the data sheet boxes are read from left to right, and top to bottom. Setting up a data base in this way makes data entry easier, because you need only move your cursor across the excel spreadsheet as your eyes move across the data sheet.

## DATA ENTRY AND MANAGEMENT IN EXCEL

Another tip for creating a good database is to keep column titles simple, one or two words rather than phrases. This makes it easier especially with large databases with many columns to find columns with a quick glance at the title. Note how the entry for “Number of days you spent hunting” on the data sheet is simply “Effort” in the database.

Note also that instead of creating a column for “Year”, the spreadsheet is simply named ‘2015’. The spreadsheet contains all the data for 2015, so a column for year is unnecessary. The other spreadsheets could be used for future years of data.

**It is essential that every cell within a row of a database contain data (See Tools and Tricks).** As you enter data, you may be tempted to leave out information that repeats. For example, in the 2015 NB tab there are 3 lines of data for the first hunter (Hunter ID BB200), so the information for the first eleven columns is exactly the same. You might be tempted to not enter the data on the second and third lines for those columns. But, if you did not enter the second and third rows of repeated information, and the information in the database were sorted (which we learn how to do in a moment), then you would lose track of which hunter harvested the mallard and the green winged teal and the northern shoveler. Avoid the frustration of having to re-enter your data!

## TOOLS AND TRICKS

### View Menu

One handy menu is the **View** menu, accessed by pressing the View tab from the main menu on the top of the screen. Here you can zoom into and out of the document (which is just like in Word). The **Freeze Panes** button is also very useful. This allows you to keep your column titles showing as you scroll down the page. This is handy when you have a lot of data and can't remember the title for each column. If you freeze the top row and then scroll down the page, you'll see that the column titles don't move and are thus always visible. Note you can also freeze the first column so that it always shows as you scroll from left to right.

### Moving Data

Moving data in excel is relatively easy. You can select columns of data to move, delete, or copy simply by left clicking the mouse, holding the clicker down, and then dragging over the column letters. You can select rows in the same way, except this time by holding and dragging over the row numbers. When data is selected, it shows up as shaded in grey.

Once you have columns or rows selected, right clicking brings up the **cut, copy, and paste** menus. Cut will delete your selection, which you can then paste elsewhere. Put the cursor in the top left hand corner of the block where you want the data to be.

You can also insert columns or rows by first selecting where you would like a new column or row to go. Then right click, and press **insert**. New columns are inserted to the left of the selected column, new rows on top of the selected row.

### Shortcuts

Once you've made a selection, simultaneously pressing the **control and c** keys will copy the information, which you can place where ever you want by moving the cursor. Simultaneously pressing the **control and v** keys will paste the information you just copied, starting in the cell where you've placed the cursor. If you want to move information and place it somewhere else, highlight what you want to move, press the **control and x** keys simultaneously, move the cursor where you want the information to go, and then press **control and v** to paste.

If you ever mess up and need to undo what you just did, simply press the **control and z** keys at the same time. If you want to re-do something you just did, press **control and y** keys at the same time.

And the **control and s** keys pressed simultaneously will save the file.

**It's a good idea to get in the habit of pressing control and s (save) often as you work.**

### Sorting Data

It is inevitable that you will have to sort your data at some point. Data sorting is used to group information together in the database. This can make it easier to check over data, to get a sense of general patterns in the data, or simply to create summary columns, which we will do in a moment.

**It is essential when sorting data that you select the entire data sheet.** Otherwise, if you select just one column, the data will be sorted *only* in that column. This will result in a mismatch in the data, as the information in the sorted column will end up in the wrong rows. Because this essentially ruins a database, and has been a common issue in the past, new versions of Excel will automatically select the entire spreadsheet for you when press the sort button.

Under the **Data** tab on the main menu, press **Sort** (note that the entire spreadsheet is selected).

Under **Column**, you'll see **Sort by** – this is where you can choose the column by which you want to sort your data. **Sort on** gives a few options, but **values** is pretty much all you need. Then you can choose whether to sort small to large numbers (A to Z for words), or large to small numbers (Z to A for words).

If you check the **My data has headers** button, then your column labels won't be sorted and end up somewhere in the data – they'll stay on the top as columns labels.

### Creating Summary Columns

Suppose we wanted to sort by Region. Under **Sort by** and **Column**, scroll to and click on Region, and then okay. You'll see that the spreadsheet is now organized according to region, in alphabetical order from A to Z. Note that you can sort by multiple columns by clicking on the **Add Level** button. If you need to remove a column from the sorting set, select it, and then click on **Delete Level**.

Often it is useful to create summary columns. For example, from the data sheet you will enter the number of birds harvested by species, but you will likely want to be able to summarize the data by species groups, such as by ducks, geese, or other species groups of interest.



## DATA ENTRY AND MANAGEMENT IN EXCEL

The column titled 'SpeciesGroup' is just such a summary column. This column was created by clicking on the Species column (click on the letter above the first row), right clicking, and scrolling to insert. Now there is a blank column in which to enter names for species groups. And now you can see the usefulness of the sort function. Instead of having to assign a species group to a mixed up jumble of species, you can sort by species, and then more efficiently enter names for species groups. This is even faster if you write a name once, copy that cell, and then paste down the column until a different name is needed.

## CHECKING FOR ERRORS

Data entry errors can result in very inaccurate answers to research questions. The good news is that these errors are the easiest to fix. **It is good and highly recommended practice once you have entered all your data to check for data entry errors.** When you find an error, fix it, and then check back again through all of your data entry once more, even the data you already checked. Keep checking your data in this way until you find no more errors.

Now you are sure that the information reported by hunters is exactly the information that your database contains.

The next step for error checking is simply to ensure that you have spelled things correctly and have written the same words in exactly the same way. For example, excel will distinguish between "Surf Scoter" and Surf scoter". To make sure that all the information associated with a grouping is included when we summarize and visualize data, we need to make sure that the names for groups appear in exactly the same way.

A **pivot table** is a quick and easy way to check for spelling and format errors. You can create a pivot table for any or all columns of data by highlighting the columns of interest (clicking on the letter above the column), or to highlight the whole spreadsheet, click on the small arrow in the upper left hand corner of the spreadsheet. As an example, highlight the Region column. On the menu at the top, click on **Insert, Tables, Pivot Table**. Then click ok.

## DATA ENTRY AND MANAGEMENT IN EXCEL

Under **PivotTable Fields**, check the box next to Region. Note that Region appears under the **Rows** box on the bottom right hand side, and a list of the regions in the database appears in the pivot table on the left.

There are 5 unique regions. “Vancouver Island & Powell River” and “Vancouver Island and Powell River” are redundant – we must choose to use one or the other so that all names in our database are spelled exactly the same way. To generate a count of the number of rows in the database for which each region appears, click on Region again under the **PivotTable Fields**, and drag and drop it into the **Values** box on the bottom right hand side. A count of the number of times that each region appears in the database is shown in the pivot table.

“Vancouver Island & Powell River” appears only once – thus that’s the name to edit. You can now delete the pivot table spreadsheet – right click on the spreadsheet tab, and scroll to delete. Be careful not to accidentally delete the spreadsheet with your data!!

**Spreadsheet deletion is not reversible.**

## Search and Replace

Now all you need to do is replace ‘&’ with an ‘and’ in one row for the Vancouver Island & Powell River region. But where is that row?

One option is to sort the data by region and then simply look for the ‘&’. But when you have hundreds of lines of data, this can be quite onerous. Instead you can use excel to quickly find what you’re looking for.

A commonly used function in excel is **Find and Replace**. Under the **Home** tab on the top menu, press the **Find and Select** button, then scroll to **Find**. In **Find what**, write <Vancouver Island & Powell River>. The spreadsheet will instantly be moved to the cell containing that exact phrase. You can either manually edit the cell, or, you can use the **Replace** function. In the **Find and Replace** box, click on the **Replace** tab, and in **Replace with**, write <Vancouver Island and Powell River>, click **Replace**, and notice how Vancouver Island & Powell River is instantly replaced with Vancouver Island and Powell River in the cell.

### CREATING A WORKING DATASET

The data set you created by entering data from the harvest survey data sheets is referred to as **raw data**. Raw data have not been processed in any way; these data are simply a digital representation of what was collected in written format on data sheets. In contrast, processed data, sometimes referred to as **working data**, have been summarized to the point that the data are ready for summary and analysis. Data management includes the types of actions you've already taken, such as sorting, error checking, and creating summary columns. **Data processing involves manipulating the data so that it can be used efficiently to answer research questions.** Usually, data processing involves collapsing a raw data set into a smaller data set.

Recall that statistical analysis requires that data be independent. For these example data, statistical independence was integrated into the study design, such that a random sample of hunters in each season were asked to provide information. Thus, note that Hunter IDs are different among the seasons with a region. In reality, your organization likely collected data from hunters that harvested in multiple seasons. Prior to statistical analysis, it is highly recommended that you draw random samples from your data when analyzing seasonal differences in harvest levels so that each season is comprised of data from different hunters.

In order to begin the process of data summary and analysis, we need to ensure that each row in our excel spreadsheet contains all the relevant data for one sample unit. In our example data set, the sample unit is one hunter. In our raw data, some hunters have several rows of harvest information, because they hunted more than one species, or the same species but in different locations. To create an independent data set ready for statistical analysis, we need to collapse the raw data set into a working data. In our working data, each row will contain the total harvest counts for each hunter.

Creating a working data set requires us to think about our research questions a bit first before we create the data set. We need to ponder, for example, whether we're interested in analyzing harvest rates per species, species group, or whether we just want to know the total harvest of all birds. Perhaps we're not interested in analyzing harvest rates per season, and thus it's fine to sum harvests across all seasons. **Usually different working data sets are needed to address different research questions.**

## DATA ENTRY AND MANAGEMENT IN EXCEL

For now, suppose we're interested in summarizing harvest rates of all birds, per region, and per season. Thus, we can sum across different locations per hunter, and we can sum across the different species that each hunter harvested, to derive one overall count of birds harvested per hunter.

Since you're about to create working data from your raw data, start by renaming the 2015 spreadsheet to 'Raw Data 2015'.

### Introduction to Pivot Tables

( Pivot Tables are fun! )

The easiest way to create working data is to use the **Pivot Table** tool in excel. Begin by selecting the entire spreadsheet by clicking on the small arrow in the top left hand corner. Click on **Insert** in the top menu, then **tables**, then **PivotTable**, then **OK**. Note that the default to insert the pivot table into a new worksheet is almost always the best option – that way, you won't overwrite any of your data with the pivot table.

Note that on the right hand side, all the column titles are listed in the **PivotTable Fields window**. Below that window, note the **Columns**, **Rows**, and **Values** windows.

Recall that we're interested in summarizing harvest levels of all birds per region for each hunter. Check Hunter ID and note that it automatically enters into the **Rows** window. That results in each individual hunter being listed in the pivot table on the left. Already we have started to collapse our raw data set, because each hunter ID is now listed just once. We can add Age to the **Rows** window, which simply tags each hunter's age to the hunter ID.

For each hunter, we want to know the total number of birds they harvested per region and season. So check region and season, and note again that they automatically enter into the **Rows** window. Note what happens in the pivot table. For each hunter, the region and season in which they hunted appear below hunter ID.

Now we're ready to sum the total number of birds they harvested. Check Quantity in the **PivotTable Fields** box – this is the count of birds. Quantity appears in the **Rows** window, but we want it in the

## DATA ENTRY AND MANAGEMENT IN EXCEL

**Values** window, because we're interested in summing the numbers in this column. Click on Quantity in the **Rows** window and drag it into the **Values** window.

By default, the operator applied to data placed in the **Values** window is 'count'. The count is simply the count of the number of rows of data for each of the columns placed in the **Rows** window. For example, for Hunter AA45554 who hunted in the Lower Mainland in fall, there are three rows of data in the quantity column, which means that he (or she) harvested three different species, or in three different locations in the fall.

We want the sum instead of the count. Click on Count of Quantity in the **Values** window, then click on **Value Field Settings**, and then click on **sum**. Now the pivot table is showing the total number of birds harvested, per hunter, per season, per Region. The table also shows hunter age. Note the other options in the **Value Field Settings**, such as **average**, **minimum**, and **maximum**. These operators are used very frequently when summarizing data with pivot tables.

### Cleaning the Pivot Table Up

The pivot table by default is in an inconvenient format – the information for each hunter is listed in the rows below the hunter ID. Fortunately, this is easy to change. In the **Rows** window, click on Hunter ID, then **Field Settings**. On the **Subtotals and Filters** tab, under **Subtotals** click on **None**. On the **Layout and Print** tab, under **Layout**, check **Show item labels in tabular form**. Click **OK**. Notice how Hunter ID is now on the same row as Region. Now we have to do the same to Region in the **Rows** window. And once we do that, you now see that all of the unique data for each hunter is contained within one row. That's what we want.

There are three final steps before we can call this a working data set. The data is stored as a pivot table; we need to copy the table and paste it so that it becomes just data. Click on the upper left hand corner of the spreadsheet, right click **copy**, scroll to **Paste Special**, and then check **Values**. This removes all formatting for the cells and pastes just the data.

The second step is to clean up the data table a bit. Remove the top two blank rows. Re-name column A back to its original label "Hunter ID". Scroll to the bottom and delete the last two rows. Re-name the

## DATA ENTRY AND MANAGEMENT IN EXCEL

'Sum of Quantity' column to something more informative, like 'TotalBirds'.

Okay. Now we have the total number of birds harvested by each hunter, and we have columns telling us their age, and the region and seasons in which they hunted.

### Generating Descriptive Statistics

In addition to using a pivot table, you can quickly generate summary statistics of column data in excel using the **function** window. First make sure the TotalBirds column is column D. Click on an empty cell, press the '=' key on the keyboard. Notice an equal sign appears in the function window below the top menu. Write **min**( and then select the values in the TotalBirds column. Then write in the other bracket ), and press **Enter**. In the function window it should read =MIN(D2:D186). When you press enter, the minimum value in the TotalBirds column appears in the previously blank cell. Move to a blank cell, and do the same except this time write max instead of min. It should read =MAX(D2:D186) in the cell, and the result should be 26. Now we know that the minimum harvest of birds by one hunter in 2015 was 0 and the most was 26.

There was considerable variation in the numbers of birds harvested. Hmm. But did hunters with high takes simply spend more time hunting? One of the main reasons to collect information on hunting effort is so that harvest levels can be compared between **strata**.

Strata are homogenous sub-groups of a population. An example of a stratum in our example data is region. Due to geographic, biological, cultural or other reasons, hunters that live in the same area are more likely to hunt the same species, hunt for similar amounts of time, and harvest similar quantities of birds.

Without information on effort, we have no way of knowing whether higher harvest levels in one region compared to another are due to regional effects, or simply because hunters spent more time hunting there. The same is true of the other strata we are considering here – season and age group.

### Harvest Effort

Effort allows us to standardize harvest levels, so that we are comparing the same thing across strata. Comparing unstandardized harvest levels across regions is like comparing apples and oranges. By standardizing harvest levels to a harvest rate (kills per day), we eliminate the possibility that any differences in harvest levels are due to effort.

So now that we've summed total take of birds per hunter, we need to also sum their effort. Which is easy right? We just have to set up our table as before but this time put Effort in the Values window.

Nooooooo! This brings us to a key point in data summary. It can save you precious time to always think carefully about the question you want to answer, and how your data is organized with respect to your question. Recall that the data sheet asks hunters to fill in information separately for different months and locations. Species harvested in the same month and location have the same information entered into the rows in the excel spreadsheet, including effort. Thus, if we were to simply sum effort, we would be summing duplicate information.

So first we need to create a pivot table of unduplicated information. The highest resolution of our data is hunter- region-location-season-month, so first we need to set up a pivot table with these columns in the **Rows** window. Now we can also place Effort in the **Rows** window. Note how some hunters harvested in different locations, and thus could have different effort in those locations. Now we have a table which includes the effort of each hunter in different locations within a region, and different months within a season. We first need to 'clean up' the table following the steps above, to get all non-duplicate information for each hunter on one row.

Now we need to create a second pivot table from this table. **Copy, Paste Special, Values**, clean up.

### Fun Trick: Filling in Blank Cells in Excel

#### QUICKLY FILLING IN BLANKS

But wait! Note that there are blank cells in this pivot table. We need to copy down the uppermost cell values to fill in the blank cells. This will ensure that if the data were sorted, row information would remain correct. We could do this manually, but there is a much faster way. Highlight the columns with blank cells down to the bottom most row of data. Do not highlight the whole columns.

Press **F5** on the keyboard. On the **Go To window**, click on **Special**. On the **Go To Special window**, click on **blanks**, then **OK**. Press **=** on the keyboard, then click on the cell above the first blank cell. Then press **Control** and **Enter** at the same time. All blank cells should now be filled with the information above.

The following step is very important. The previously blank cells contain formatting to copy the contents of the cell above. If you were to sort the data, the cell values would change because they are formatted to copy the contents above. Thus, as with copying the pivot table, you need to copy all the formatted columns, and then paste the contents as values only. **Copy, Paste Special, Values**.

Now we're ready to create the second pivot table. HunterID into **Rows**, and Sum of Effort into **Values**. Change Count to Sum. **Copy, Paste Special, Values**, clean up, fill in blanks.

The result is the total number of days within a season spent hunting per hunter (recall that the data are independent, such that each hunter's data are from one season only). This column, re-labelled 'Effort' can be added to the table of Totalbirds. We just need to sort each table of information by HunterID to match them up.

Now you have some working data! Rename this spreadsheet 'working data total birds'. Now you're ready to summarize and visualize these data.





# SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

Data summary is essential for understanding your data, and is the first step toward statistical analysis.

A **variable** is a characteristic of the world of which measurements have been taken. Our working data is comprised of five variables – Age, Region, Season, Effort, and Quantity. Hunter ID is the sample unit.

When we collect data to answer research questions, we are usually asking a question something like, ‘how do these things (variables) affect these things (variables)? As we ask that question, we observe sample units as the basis of our questioning and answering. In statistical terms, the variables that are affected are referred to as **response variables**. The response variables in our case are harvesting effort and the quantity of birds harvested. Variables that we think might affect response variables are referred to as **predictor variables** in statistical terminology. These are hunter age, region, and season. In essence, we want to test with our data how hunter age, region, and season affect harvest levels of birds and harvesting effort.

When we summarize data, we are usually interested in summarizing both predictor and response variables. When we summarize predictor variables, it can give us a sense of the layout of our study, which is referred to as **study design**. When we summarize response variables, it gives us a sense of general patterns in the data and begins us on the path toward answering our research questions.

## SUMMARIZING PREDICTOR VARIABLES

Open the excel document titled ‘harvest\_worked\_data.xlsx’. This file contains the raw data, and a copy of the working data – click on the spreadsheet titled ‘working data total birds’.

Before we begin analyzing data, it’s important to get a good sense of the study design. One of the main questions regarding study design is, what is the sample size? It’s easy to see the total sample size from the working data – we created a set of independent working data, so each row corresponds to one hunter. The number of rows is our sample size, which is 185 hunters. But when we’re comparing response variables among strata, we need to know the sample size per strata. The key question is whether sampling was generally balanced with more or less similar sample sizes per strata, or whether it was highly uneven. Estimates of responses in strata with relatively low samples sizes may be less accurate

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

than those with higher sample sizes. We want to know, how many hunters comprise our random sample per region, season, and age group?

First let's use pivot tables to do some quick summaries. Create a pivot table from the whole spreadsheet. Move Age to the **Values** window, and change the field value to **minimum**, then to **maximum**, and then to **average**. This is a quick way to get some useful summary information on hunter age. We've just found out that the range of hunter ages is quite broad - 17 to 75, and the average age is 47 years old. Move Age out of the **Values** window.

Now let's find out sample size per strata. Drag Region into the **Rows** window, and, since each row corresponds to one hunter, any of the columns can be counted per region to generate the number of hunters per region. Try that out - place Hunter ID in the **Values** window, and Season. Both result in the same count, which is the sample size of hunters per Region. Sample sizes per Region appear adequate at first glance, with at least 24 hunters being sampled per region. The sample sizes for the Lower Mainland and Thompson and Okanagan are twice that of the other regions, but this is probably appropriate given the higher population size of hunters in those two regions. Removing Region and placing Season in the **ROWS** window results in the sample size per Season.

Ah. The sample size for winter is just 9 hunters. This is likely too low of a sample size to be able to generate accurate estimates of harvest levels of all hunters of all ages in all regions of BC during winter. It would be okay summarize this information as a cautious estimate of harvest levels in winter, but it should not be used in statistical models that test for effects of other strata.

Now we can summarize by hunter age. First create a summary column in the working data sheet titled AgeGroup. Sort the working data by age, then fill in the AgeGroup column using the following groupings: 17-30, 31-40, 41-50, 51-60, 61-75. Using the pivot table to count sample size per age group shows relatively even sampling across age groups. Sampling is relatively lower in age group 31-40, but at 26 hunters it should be sufficient to generate reliable estimates.

So now what about sample sizes per region per season per age group? This is our full study design. We need Region, Season, and AgeGroup in the **Rows** window, and then any column placed in the **Values** window. By default this will give us the count of rows, which is the sample size. To make the table more

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

compact and readable, shift Season to the **Columns** window.

The resulting table of sample sizes per strata is a key aspect of your study design, and is a recommended table to include your report.

*Table 3. Sample size (number of hunters) per strata of 2015 harvest data collected from a questionnaire sent to BC resident hunters*

Region	Age Group	Fall	Spring	Summer	Winter
Kootenays	17 - 30	2	2	2	
	31 - 40	1	1	1	
	41 - 50	1	1	1	
	51 - 60	2	3	2	
	61 - 75	2	1	3	
Lower Mainland	17 - 30	6	3	4	1
	31 - 40	1	1	1	
	41 - 50	5	7	4	1
	51 - 60	4	2	2	1
	61 - 75	3	3	2	1
North	17 - 30	2	1	3	
	31 - 40	4	4	1	
	41 - 50	1	1		
	51 - 60	2	2	1	
	61 - 75	2	2		
Thompson and Okanagan	17 - 30	3	3	2	
	31 - 40	3	3	3	1
	41 - 50	4	4	3	1
	51 - 60	3	3	2	1
	61 - 75	6	6	5	2
Vancouver Island and Powell	17 - 30	1	2	2	
	31 - 40	1			
	41 - 50	2	1	1	
	51 - 60	2	2	2	
	61 - 75	2	3	3	

The table shows that the sample size for winter is insufficient for statistical modelling across other strata. It also shows that sampling across ages was relatively good – all age groups were sampled in all regions, except for two age groups in summer in the north, and 31-40 year olds in Vancouver Island & Powell River in spring and summer.

The distribution of sample size across strata is important to keep in mind when conducting statistical

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

modelling. If the effect of age on harvest levels is the same across regions, then harvesting by 31-40 year olds in other regions can be used to estimate their harvest levels in Vancouver Island and Powell River. If it is not, then we cannot make any conclusions about the harvest levels of 31-40 year olds in this region.

This study design based on a sample size of 185 hunters is pretty good – the sample size is adequate and relatively evenly-distributed across the strata, except for winter. There are no ‘big holes’, such as no sampling of a particular age group in any region. In addition, the data is relatively well ‘balanced’. For example, sampling was not really high in one region, and really low in other regions.

However, the study design can be improved – note that the sample sizes of hunters per region and per season is relatively low. For example, the data for the north in summer is based on a sample size of only four hunters. As you begin to work with your data, become aware of any sample size issues. These could be addressed increasing your level of engagement with hunters in a particular region, through mail outs, phone conversations, community workshops, etc.

### **Does Size Really Matter?**

As we have seen, the higher the sample size, the higher the proportion of the population we sample, and thus the more accurate our estimates and the more truthful our answers. You may be asking, how will I know if my sample size is big enough? The answer to that question is not simple, and touches on some deep philosophical arguments in the scientific community. There is no simple answer, except the reminder that the more variable the data, the higher the sample size needed to ‘capture’ that variability. For example, if 31-40 year old hunters from the same region harvest relatively the same numbers of birds, then we don’t need data from all that many 31-40 year old hunters to be able to get an accurate estimate of their harvest levels. But if hunters of this age group do things very differently, then we will need data from a lot of them within each region to be able to capture all of that variability in our sampling net.

### SUMMARIZING RESPONSE VARIABLES

Now that we have a sense of our predictor variables, let's summarize our response variables. The average and variability of response variables among sample units are the most common summary statistics.

Let's first make sure we understand these statistics. But let's first make sure we understand the meaning of a **statistic**.

Recall our earlier discussion about a **sample** versus a **population**. We imaginatively took a 2 x 2 foot sample of flowers from a population of flowers, an entire field. We used the proportion of blue flowers in the sample to estimate the proportion of blue flowers in the population. In other words, we used the sample as representative of the population. The sample proportion is a statistic.

A statistic is a quantified characteristic of a sample taken from a population. When we perform statistical analysis, we assume that the sample statistic is an accurate estimate of the same characteristic in the population.

We need to introduce one other helpful statistical term – **observations**. Observations are the measurements and descriptions of the world that we collect to form our data to answer our research questions. Measurements in the form of numbers are observations, but so are qualitative descriptions, such as low and high, small and large, and categorical descriptions like 'sandy', 'blue', and 'cloudy'.

### Estimating Averages

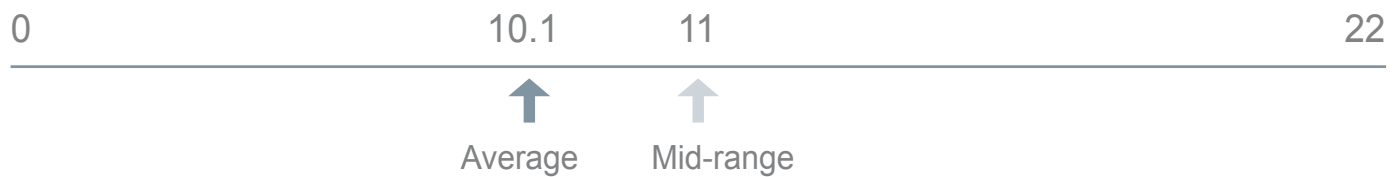
The average, otherwise referred to as the mean, of a group of observations is a measure of the 'central tendency' of the observations ordered from small to large. Let's look at our imaginary harvest data from 15 hunters [above](#). Ordering the observations from smallest number of ducks killed to the largest number of ducks killed looks like this:

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA



We calculate the sample average simply by summing all the values of the observations and dividing by the number of observations.

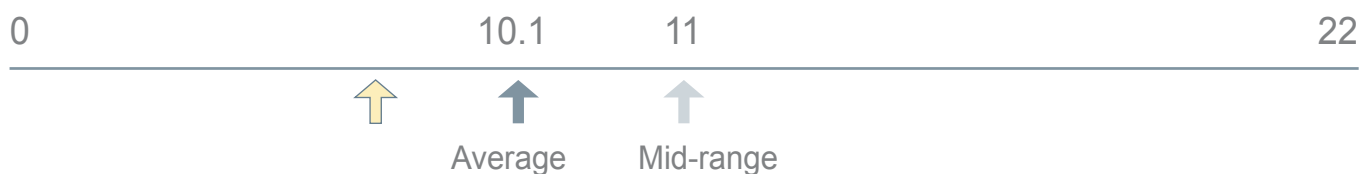
The sum of the values of the observations is 152. There are 15 observations. Thus the average is  $152/15 = 10.1$ . Note that the average is NOT the centre of the range of values, which is 11.



You can get an intuitive sense that the average is a measure of central tendency by stacking observations on one end of the range versus the other. Suppose a higher proportion of the 15 observations had values below 10.



With 9 observations with values below 10, the average shifts lower, to 9.5.

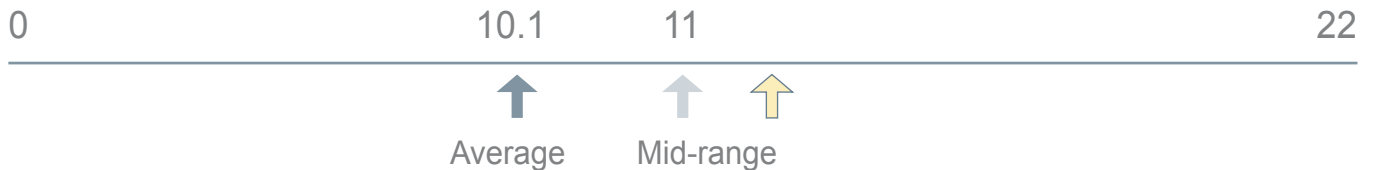


## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

Now let's see how the average changes if a higher proportion of the 15 observations were above 10.

0	3	3	5	7	7	11	12	13	14	15	19	19	21	22
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

With 9 numbers with values above 10, the average shifts higher, to 11.4.



**The main reason to calculate the average of a sample, is to use that average as a statistic. That is, we use the sample average as an estimate of the population average.**

Let's imagine that our 15 observations comprise an entire population, and that we take three random samples of 9 observations from the population.

Observation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Duck Kills	0	3	3	5	7	7	11	12	13	14	15	19	19	21	22

To ensure our sample is random, we number the observations from 1 to 15, write the numbers 1 to 15 on pieces of paper, throw the pieces of paper into a hat, and then pull out 9 numbers. Or, we can use a random number generator available from an online google search (e.g. <http://graphpad.com/quickcalcs/randomN1.cfm>). We could also use the R script `<round(runif(9,1,15))>`.

The first random draw is for observations 2, 3, 4, 5, 7, 10, 12, 13, and 14.



## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

Observation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Duck Kills	0	3	3	5	7	7	11	12	13	14	15	19	19	21	22

The average of this sample of 9 observations is **11.3**.

The second random draw is for observations 1, 2, 3, 8, 10, 12, 13, and 15.

Observation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Duck Kills	0	3	3	5	7	7	11	12	13	14	15	19	19	21	22

The average of this sample of 9 observations is **11.5**.

The third random draw is for observations 1, 3, 5, 6, 7, 9, 11, 12, and 15.

Observation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Duck Kills	0	3	3	5	7	7	11	12	13	14	15	19	19	21	22

The average of these 9 observations is **10.8**.

Each of these sample averages are quite close to the true population average of 10.1 that we originally calculated. By now, you may be starting to get an intuitive sense of the importance of sample size. If our sample size were 12 instead of 9, our estimate of the population average would have been even closer to the true value. The degree of closeness of sample estimates to the true population value is referred to as the **accuracy** of estimates. The degree of closeness of sample estimates to one another is referred to as the **precision** of estimates. The first and second samples above are more precise than the second and third samples. The precision of estimates is dependent on sample size, and also on the amount of variability in the population and sample data, which is the topic of the next section.

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

Other measures that can be used to describe a set of observations include the **range**, **median**, and **mode**. The range is simply the difference between the minimum number and maximum number, for example, the range of the duck kills data above is 22. The median is the number that divides the data set in half, with the observations ordered from smallest to largest. The median duck kill is the 8th number, which is 12. The mode is the number that appears most often in the data set, which is 3,7, and 19 in the duck kills data.

### Estimating Variability

We calculate the sample average because we want to estimate the central tendency of a population. Estimates of central tendency alone are sometimes useful, but in terms of inferring the truth about a population from a sample, we also need estimates of the amount of difference – the variability - within the population.

**Variability** refers to the spread of differences within a population.

Let's consider a data set comprised of weights in pounds of 24 Canada geese.

*Table 4. Weight in pounds of 24 Canada Geese*

Weight (pounds) of 24 Canada Geese	
16	14
10	10
4	8
9	9
17	17
12	9
19	15
11	16
10	13
10	12
16	8
15	7

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

We can now easily calculate the average goose weight as the sum of weights divided by 24, which is 12 pounds.

But how can we measure the variability? One way is to simply look at the range of values. The minimum weight is 4 pounds (a runt for sure) and the maximum goose weight is a beastly 17 pounds. Goose weights in this sample range from 4 to 17 pounds. That might give us a sense of the variability of all Canada geese everywhere, but we really have no way of knowing. The more geese we weigh, the larger the range in weights we will measure.

We need a measure of variability that measures the amount of difference in the sample that is also a reliable estimate of the variability in the whole population. By reliable we mean repeatable. Since the average of a sample is a reliable estimate of the population average (provided the data are not biased), a measure of variability that is based on the average is also reliable.

Imagine in your mind variability as the difference between the value of each observation and the population mean. Each observation **deviates** from the mean by some amount, some observations are larger than the mean, and some are smaller. Some geese are heavier than the average of 12 pounds, and some are lighter. One reliable measure of variability is the **average deviation from the mean**, which is referred to as the **variance**. The variance is a measure of the amount of difference in a population.'

Some observations are smaller than the mean, and thus the difference between the mean and the observation is a negative number. It is the actual amount of the difference that we're concerned with, not whether or not the observation is below or above the mean. We could just take the absolute values (just the value, without the positive to negative sign), but various combinations of absolute values of deviations can result in the same sum. What we want is a measure that changes as the total amount of variability changes.

Thus, to calculate the variance, we first need to multiply each deviate by itself. Then we calculate the average of these now scaled deviates. But with one minor difference. Instead of calculating the average by dividing by the number of observations, we divide by one less than the number of observations.

This is referred to as the **degrees of freedom**, which is an infamously difficult concept to grasp at first.

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

It can be understood perhaps best by considering what population parameters we first need to estimate in order to derive another estimated population parameter. To estimate the variance, we first need to estimate the mean. In general, the degrees of freedom for a parameter estimate are the number of observations ( $n$ ) in the sample minus the number of parameters ( $p$ ) that need to be estimated first in order to estimate the parameter (degrees of freedom =  $n - p$ ).

The degrees of freedom can be viewed as the information that you can freely 'spend' to estimate the parameter. If you knew the mean, and you knew the values for all  $n$  observations except one, you would then know the value of that last observation. You have to estimate the mean to estimate the variance, and thus one observation is tied to the mean and can't be spent to estimate the variance.

Next, we will calculate the variance of our 24 goose weights step by step. We already calculated the average, and found it is 12. The differences from each observation from 12 is given in the second column. These are the deviates. Each deviate is then squared (multiplied by itself), and then the squared deviates are summed, which totals 335. The variance is then calculated as  $335 / 24 - 1$ . Remember that instead of a usual average, we divide by the degrees of freedom, which in this case is the number of observations minus 1. The variance of our sample of 24 goose weights is 14.6.

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

Table 5. Weight, deviates and deviates squared for 24 Canada Geese

Weight (pounds) of Canada Geese	Deviates - Difference from Average of 12	Differences Squared (Deviate x Deviate)
16	4	16
10	-2	4
4	-8	64
9	-3	9
17	5	25
12	0	0
19	7	49
11	-1	1
10	-2	4
10	-2	4
16	4	16
15	3	9
14	2	4
10	-2	4
8	-4	16
9	-3	9
17	5	25
9	-3	9
15	3	9
16	4	16
13	1	1
12	0	0
8	-4	16
7	-5	25
		SUM = 335

Now, remember that we just squared the deviates, so to get back to our original scale we need to take the square root of the variance, which is referred to as the **standard deviation**. The standard deviation is akin to the average difference from the mean in a sample, and is used as an estimate of the average difference from the mean in the population from which the sample was taken. The standard deviation of our sample of goose weights is  $\sqrt{14.6} = 3.8$ . That means that on average goose weights in our sample vary from the mean weight by 3.8 pounds.

Using these two sample statistics, we can now make the statement that we estimate that ALL Canada geese weigh on average 12 pounds, and that Canada goose weights vary from the mean on average by 3.8 pounds.

Now that we know how to calculate two important statistics, and more importantly, we know what these

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

represent, we can apply them to create summaries of our response variables.

Step by step calculations have been created for you to ‘manually’ calculate the standard deviation in the excel file excel\_SD\_calculations.xls.

### Using Pivot Tables to Calculate Averages and Standard Deviations

Back to our working data <harvest\_worked\_data.xls>. We have two response variables – the total number of birds harvested per hunter, and their effort, the number of days they hunted. And we have three predictor variables – hunter age, region, and season. A good way to begin summarizing these data is to begin by asking how harvest levels vary across seasons. We can calculate average harvest levels and the variability of harvest levels across seasons.

We can do this easily using pivot tables. These steps are a repeat of what we’ve already done, but it’s good to practise. Remember that we created the working data as the sum of birds for each hunter per season and region. Thus, we know that the averages calculated across seasons are statistically independent – there is one row of data per hunter per season. If we created a pivot table from our raw data, with multiple rows of data per hunter, our average would be calculated incorrectly, because the raw data are not independent.

Select the entire sheet, click **Insert, Tables, PivotTable, OK**. Our question is how harvest levels of birds varies across seasons, so then we need to place the predictor variable Season in the **Rows** window. Click Season. We want to know the average number of birds harvested by hunters per season. Thus, click on Totalbirds and drag it into the **Values** window, click on it, and then scroll to **Value Field Settings** and change **Count** to **Average**. Click **OK**. Now the pivot table is showing the average number of birds harvested by hunters in fall, spring, summer, and winter. We can see that these averages are what we would expect, harvest levels are highest in spring and fall, and low in summer and winter.

Now we just need the standard deviation and the sample size. Click and drag Sum of Totalbirds once more into the **Values** window, click on it, scroll to **Value Field Settings**, and this time change **Count** to **StdDev** (NOT StdDevp). Click **OK**. For the sample size, drag any of the columns into the **Values**

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

window.

Once again, we have a pivot table with all its fancy formatting that must be converted to just a table of data. Select and copy the spreadsheet, **Paste Special, Values**, clean it up, re-label the columns (e.g. Season, Average birds harvested, Standard deviation), and fill in any blanks (in this case there are none). If you need to, refer back to the instructions [above](#).

Note the very long string of numbers after the decimal point. These are not necessary. It's standard practice to just have one or two digits after the decimal point. Select the table, right click, scroll to **Format Cells**, under **Category**, select **Number**, then insert a 2 in the window for **Decimal places**. Click **OK**.

### Creating Informative Summary Tables

Voila! Your first summary table. Now we just need to give it a title and format it. It's important to write informative table titles so that readers know exactly what the table is showing. Here's an example:

*Table 6. Average numbers ( $\pm$  standard deviation) of migratory birds harvested across seasons in 2015 throughout BC by members of the Metis Nation BC*

Season	Average Birds Harvested	Standard Deviation	Sample Size (Number of Hunters)
Fall	10.72	6.19	65
Spring	7.61	5.82	61
Summer	3.20	2.56	50
Winter	1.89	1.76	9

It's good practice to follow these general guidelines when writing table titles.

**The table title states:**

1. *what* the numbers represent. Standard deviation is usually represented in brackets after the average using the  $\pm$  symbol (recall that deviates can be below or above the mean),

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

2. *more* specifically than the table column that these are counts of migratory birds,
3. *where* the birds were harvested,
4. *by* whom the birds were harvested,
5. *when* the birds were harvested.

Note that the table is very simply formatted with just three lines – two to separate the column titles, and one at the bottom of the table. Simple formatting is easy to read and avoids the fuss of wasting your valuable time on unnecessary formatting.

Before we move onto visualizing summary data, let's rename the sheet 'Summary Data'. It's good practice to store summary tables with their titles written to the side or above the table in the same spreadsheet. You should now have one excel file with one spreadsheet of raw data, one spreadsheet of working data, and one spreadsheet for summary tables.

## VISUALIZING SUMMARY DATA USING EXCEL

It's often very helpful to visualize data summaries using plots. We have calculated average and standard deviations of bird harvests across the four seasons. Averages are best visualized using **column or bar charts**. These are very easily created in excel.

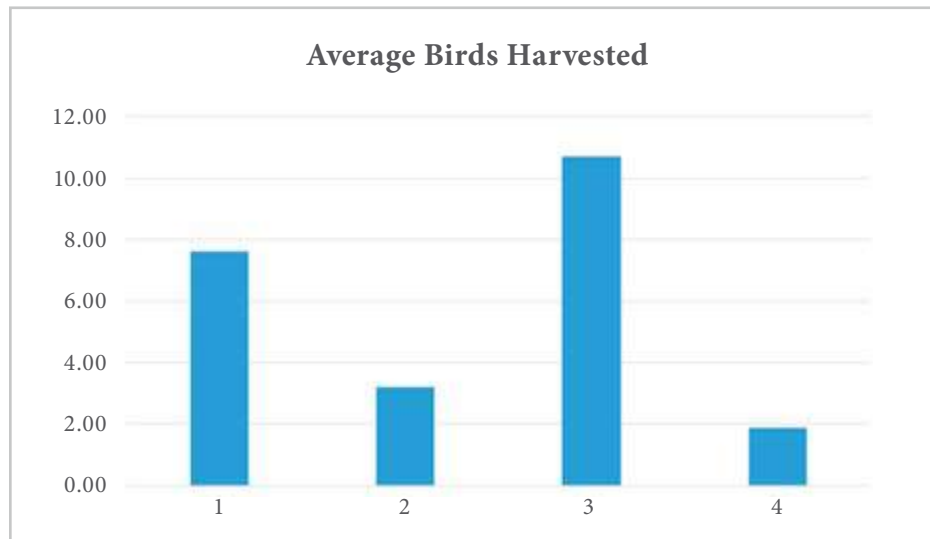
First let's organize the information a little more logically. Note how the seasons are listed alphabetically. It would be easier to understand the chart if the seasons were displayed chronologically. Let's start the list with spring – move the Fall row until the seasons are in the correct order.

### Creating Bar Charts

Select the Average Birds Harvested column, click on **Insert** in the top menu, and then on the column chart icon. You should get a chart something like this:



## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA



This needs some work. First, we need to change the x (horizontal) axis numbers 1 to 4 to the seasons, and we don't really need two digits after the decimal place on the y (vertical) axis. Most importantly, we need to visualize the variability, the standard deviation.

So let's clean this up. First we'll add the seasons to the x axis. Right click on the chart and scroll to **Select Data**. Under **Horizontal Axis Labels**, click **Edit**. When the **Axis Labels** window pops up, with the cursor clicked into the **Axis label range** box, select the seasons in the Season column, from column 2 to 5. Click **OK** and **OK** again. The seasons should now appear on the chart.

Right click on the y axis and scroll to **Format Axis**. This opens to many options for formatting the axis. Under **Number**, put 0 in the **Decimal places** window.

Now we need to show the standard deviation.

Depending on the version of excel you're using, the following instructions may or may not work. If not, you can google the version of excel you're using and 'adding error bars to charts' and it should bring you to step by step instructions.

Click on the chart, under the **Chart Tools** menu, click on **Design**, **Add Chart Elements**, **Error Bars**, **More Error Bar Options**. In the **Vertical Error Bar** window, select **Plus**, and then under **Error Amount** select **Custom** and click on **Specify Value**. The **Custom Error Bars** window should pop up. Click in the

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

window under **Positive Error Value**, and then select the four values in the standard deviation column in the spreadsheet. Click **OK**.

We are now two steps away from charting perfection:

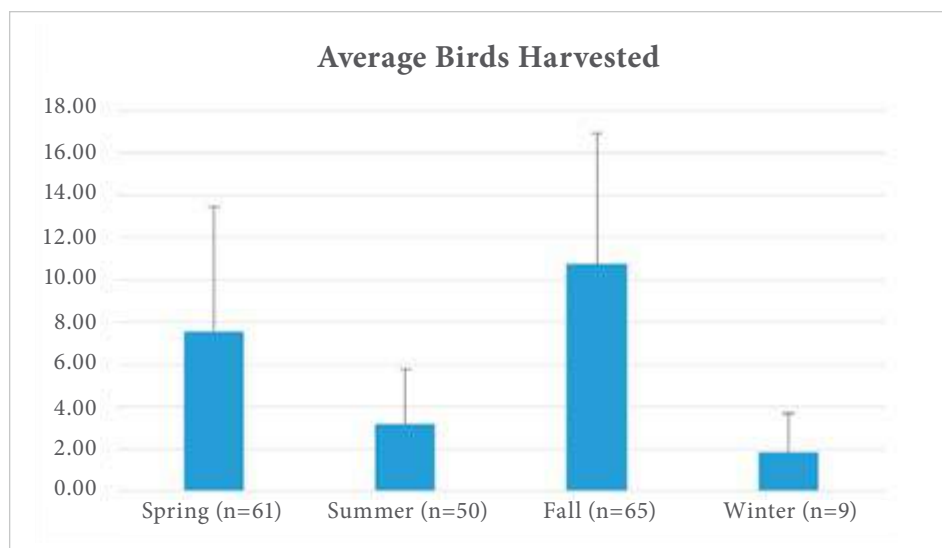
1. Our chart needs a title,
2. We need to indicate the sample size used to calculate these averages.

Just as for the summary table we created, the title for our chart needs to be informative, and in fact, can be exactly the same. In reports, it's customary to present either a summary table or the visual representation of the table, because they show the same information.

And now for the sample sizes. The easiest way to show the sample sizes on the chart is to simply write them into the season column beside each season name.

Below is the finished product:

*Figure 1. Average numbers ( $\pm$  standard deviation) of migratory birds harvested across seasons throughout Ontario by members of the Metis Nation of Ontario in 2015.*



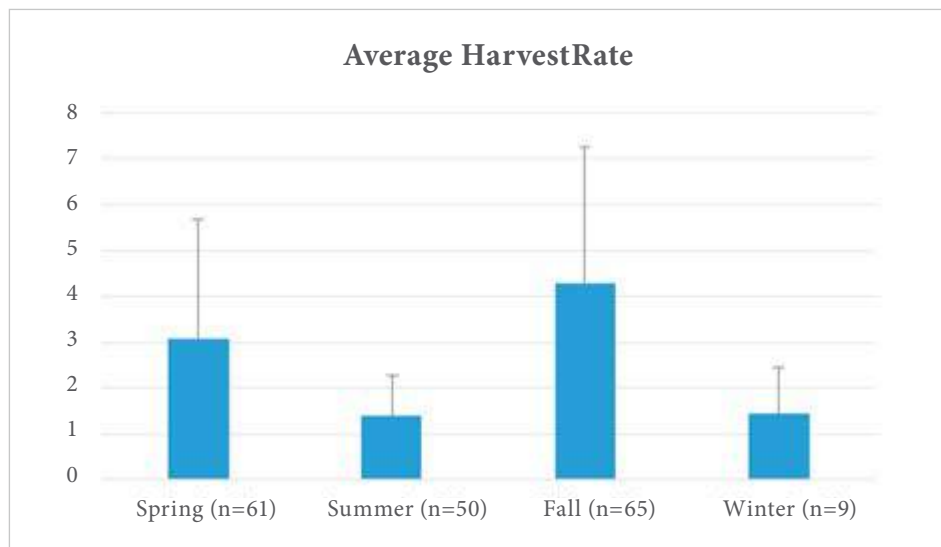
Note that these are unstandardized averages – we did not divide harvests by effort, and thus we

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

are left with the question, were harvest levels highest in the fall because hunters spent more time hunting then?

We can answer this question by first dividing the Totalbirds column by the effort column, to derive the harvest rate per hunter per season. To create this new variable, place the cursor in the second row of column. Press the = key on the keyboard, click in the cell in the second row of the Totalbirds column, then press the / key on the keyboard, and then click in the cell in the second row of the Effort column. Then press **Enter** on the keyboard. You've just used excel as a calculator. The value showing in the Harvest Rate column should be equal to the value of the Totalbirds in the second row divided by the value of the Effort in the second row.

Then we repeat all of the above steps, but this time on our newly created HarvestRate column. We change the chart title accordingly.



Here we can see that high harvest levels in fall were not the result of higher effort. High harvest levels in the fall is in fact not surprising, and is the seasonal effect on harvest levels that we would expect, given that that's when migrating birds are most abundant and visible.

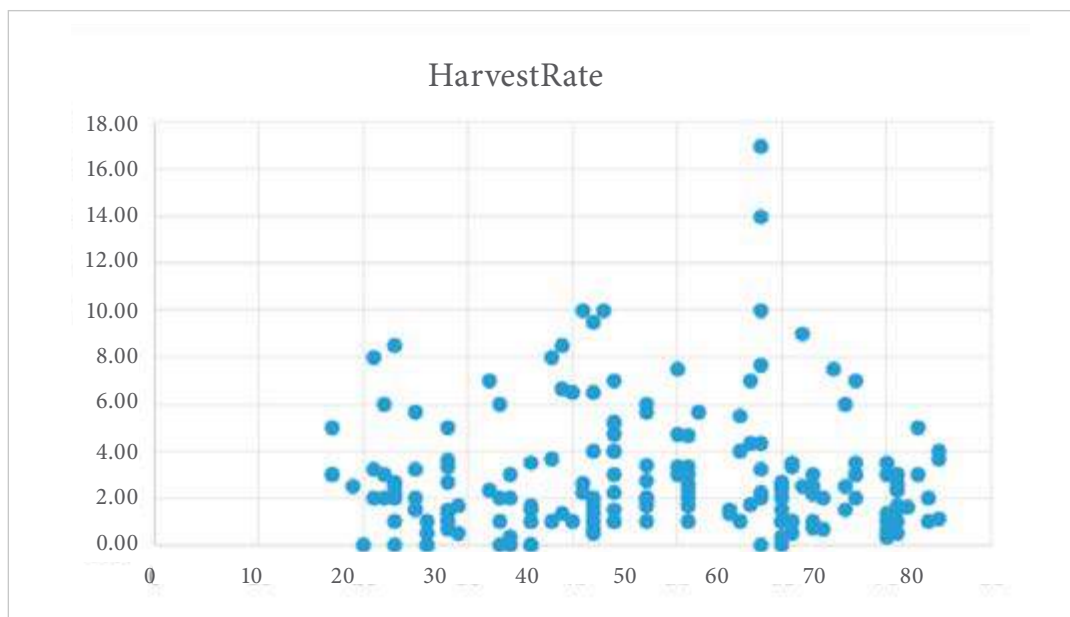
Now let's summarize harvest levels with respect to hunter age. Age is a **continuous variable**, which means that it is a numerical variable that varies from the lowest possible number (0) to the highest possible number. In contrast, season is a **categorical variable**, since it contains categories of

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

information. We could transform age into a categorical variable by grouping ages into categories, such as young, middle age, senior, elderly, or simply into numerical categories, such as 20-30, 31-40, and so on.

### Creating Scatter Plots

For now, let's leave age as a continuous variable. This allows us to create a **scatter plot** of harvest rates on hunter age. To create a scatter plot, move the Harvest Rate column adjacent to the Age column. When creating scatter plots in excel, the horizontal axis must come before the vertical axis, from left to right. Select both columns, then click on **Insert** on the main menu at the top, and then on the scatterplot icon.



The plot displays harvest rates by hunter age.

There is one easily noticeable pattern in these data? Can you see it?

Hunters aged 36 to about 58 were always successful – note that there are no 0's for this age range.

Patterns in data result from one of two processes – either from real processes, or, from biases in the data. A good data analyst always considers both. Unless we measure bias, we never really know whether the patterns we observe in data are real living patterns out there in the world, or whether they result from

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

biases in our data collection. Let's consider both.

A real age effect in harvest rate makes sense – we would expect that with age comes experience, and more experienced hunters would be more likely to always kill at least one bird every time they go hunting. But at some point, the effects of old age kick in. Older hunters perhaps aren't the sharp shooters they used to be, which might explain the decline in harvest rate noticeable on the scatter plot after about age 60.

Alternatively, it could be that middle aged hunters are reluctant to admit when they got skunked, so they didn't report on the harvest survey questionnaire the hunting trips when they didn't get any birds, which is referred to as non-response bias. The relatively high takes of middle aged hunters might be real, or, these might be exaggerations, which is referred to as **prestige bias** – the tendency of some hunters to report a higher take than they actually harvested. Without conducting a follow-up study to ask hunters of this age group about their reporting, we have no way of knowing whether this pattern is real, or the result of bias.

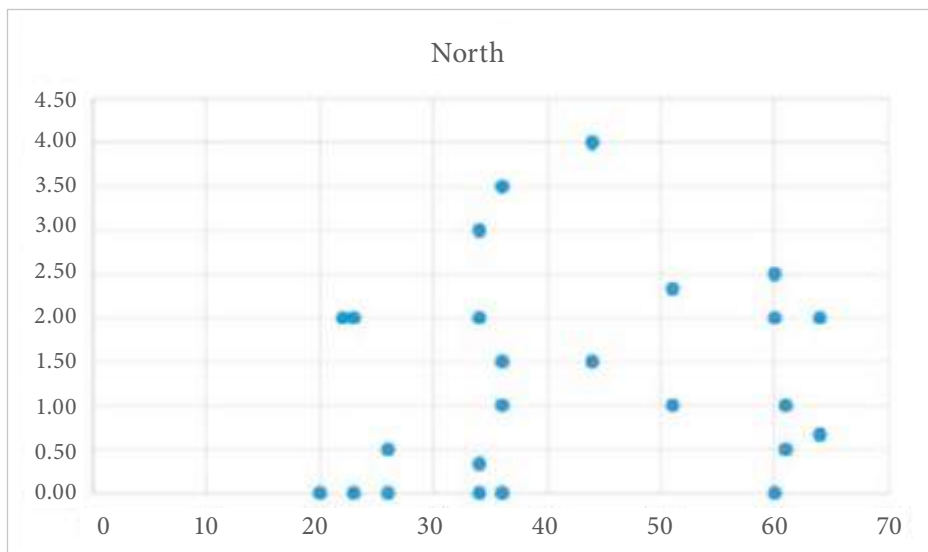
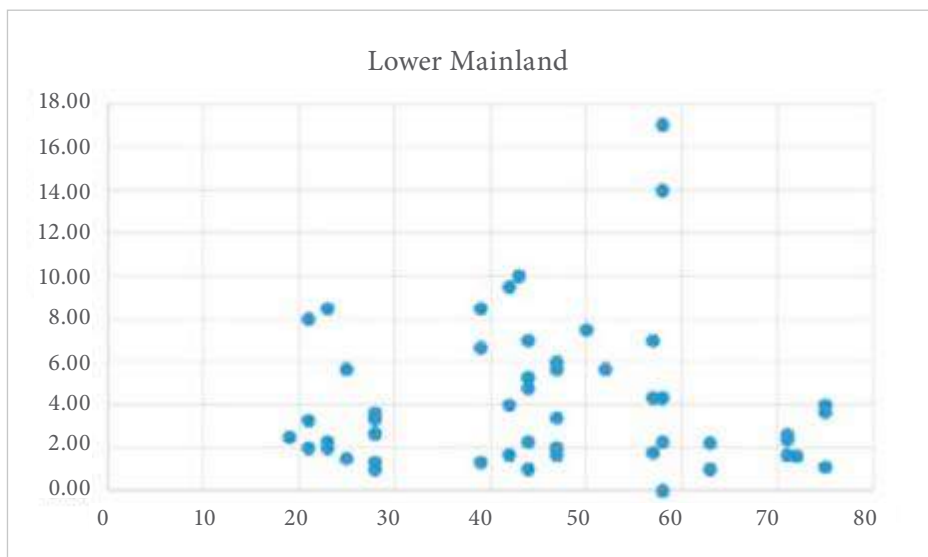
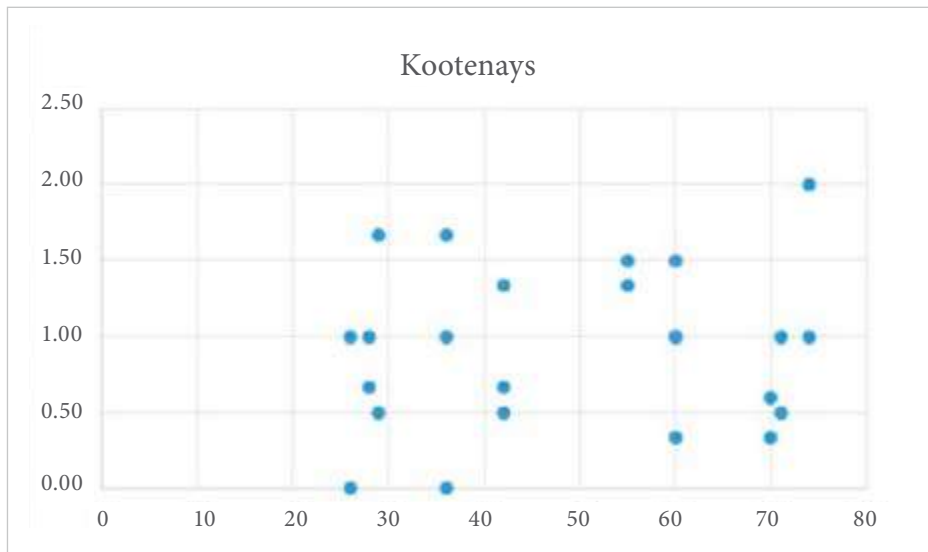
However, data summaries can help us get a sense of whether we're seeing real patterns or the effects of bias. For example, we might expect that prestige bias in this age group exists in one or two regions, but it would be more unlikely to exist across all regions. In fact, we might expect that prestige bias would exist more prominently in the youngest age group, in at least one region.

So let's investigate the relationship of harvest rate on age per region. The easiest way to do this is to sort the data by season, and then just create separate scatter plots of harvest rate on age, with the data selection limited to data for one region only.

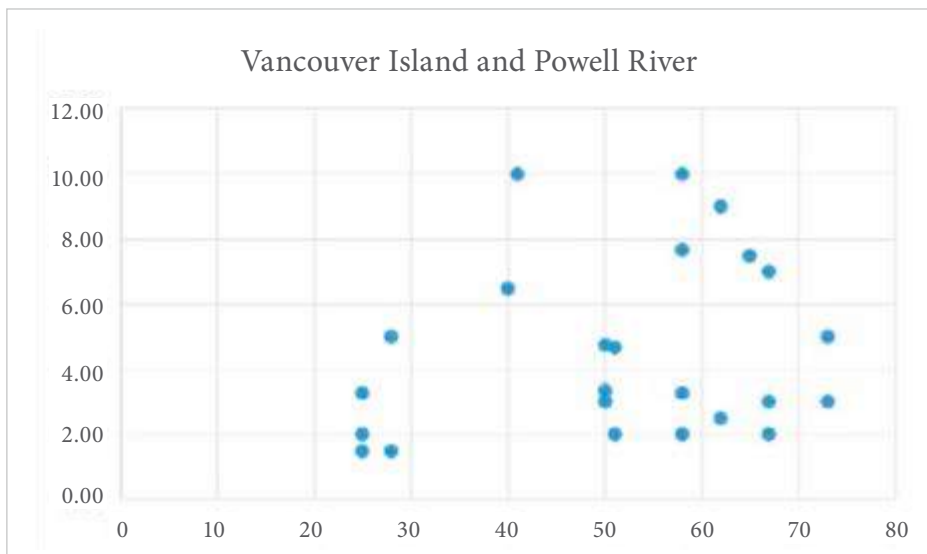
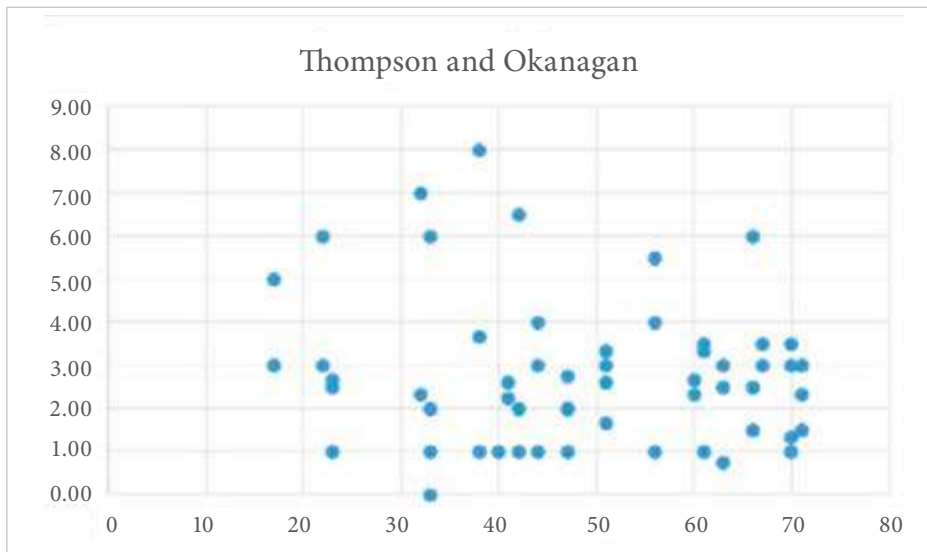
You'll notice that scatterplots created from selections that don't include the column title are labelled "Chart Title". You can edit this by clicking on the label, and then entering a label of your own choosing. Title each chart you create by the corresponding region.

Let's change the range of the horizontal axis. Right click on each horizontal axis, then Format Axis. There are no hunters less than 17 years old or older than 75 so let's change the range to 15 to 75. In Bounds, change the minimum to 15 and the maximum to 75.

# SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA



## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA



These plots of harvest rate on age don't really provide any conclusive evidence to help us decide whether we're seeing real patterns or biases. For example in Vancouver Island and Powell River and Thompson and Okanagan, all hunters except one reported harvesting at least one bird. Middle aged hunters in the North and Kootenays are clearly the drivers of the observed pattern, but we have no way of knowing whether this is real or not. This pattern isn't as clear in the Lower Mainland.

However, a decline in harvest rate after about age 60 is fairly consistent across regions, suggesting this is a real effect.

Now let's summarize regional differences in harvest rates per season. We will ignore winter since

## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

sample sizes were low. Region is a categorical variable, so we need to create a bar chart of the average and standard deviation of harvest rates per region.

If you have difficulty calculating the averages and standard deviations, refer back to the instructions [above](#). Creating pivot tables will become automatic to you before long, since it is the same steps repeated over and over. A hint here is to put the Region column in the **Rows** window, and the Season column in the **Columns** window. Remember to change the field values for Harvest Rate to Average. Once you've calculated the average, copy and paste the average values to the Summary Data spreadsheet. Now go back to the pivot table, and change the field values from averages to standard deviation. Add sample sizes to the table. Then copy and paste the standard deviation values next to the average values in the Summary Data spreadsheet.

You should end up with a summary table that looks something like this:

*Table 7. Average harvest rate, standard deviation and number of samples (n) for five areas of BC in the fall, spring and summer*

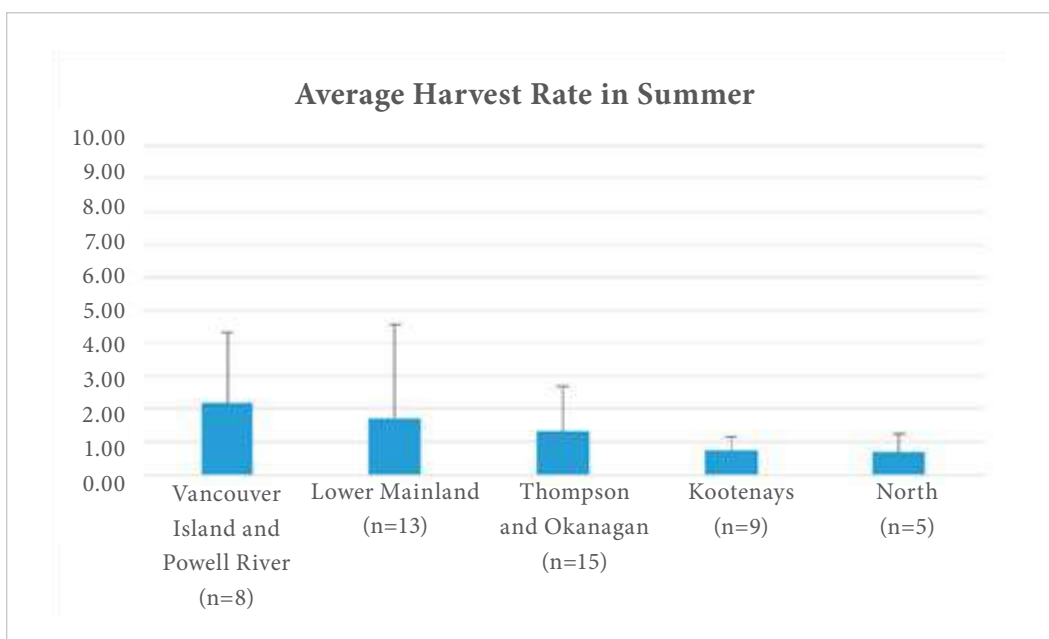
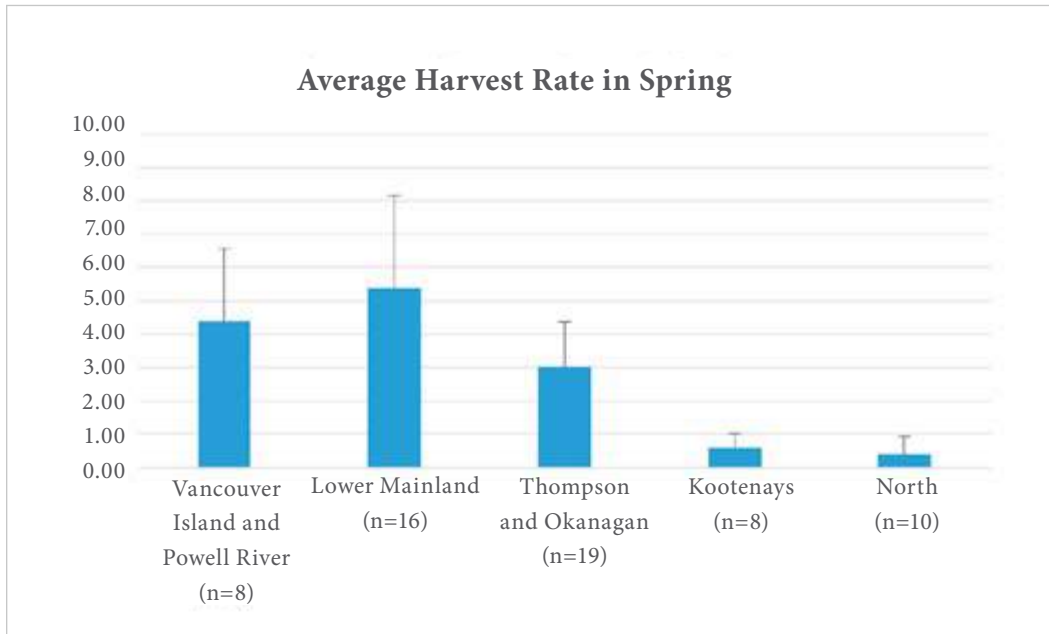
Region	Fall			Spring			Summer		
	Average Harvest Rate	Standard Deviation	n	Average Harvest Rate	Standard Deviation	n	Average Harvest Rate	Standard Deviation	n
Kootenays	1.46	0.34	8	0.60	0.42	8	0.73	0.36	9
Lower Mainland	5.70	3.86	19	5.37	2.82	16	1.73	0.36	13
North	2.35	0.87	11	0.40	0.52	10	0.70	0.84	5
Thompson and Okanagan	4.03	1.54	19	3.02	1.36	19	1.32	1.09	15
Vancouver Island and Powell River	7.08	2.33	8	4.41	2.14	8	2.19	0.59	8

We can now chart these averages with their corresponding standard deviations and sample sizes. Note that on each chart the regions are arranged geographically, the charts are presented chronologically, and the range of the vertical axis is the same across all charts. It is good practice to lay out charts in ways that help in thinking about patterns, whether just for your own use or as a figure in a report. Note

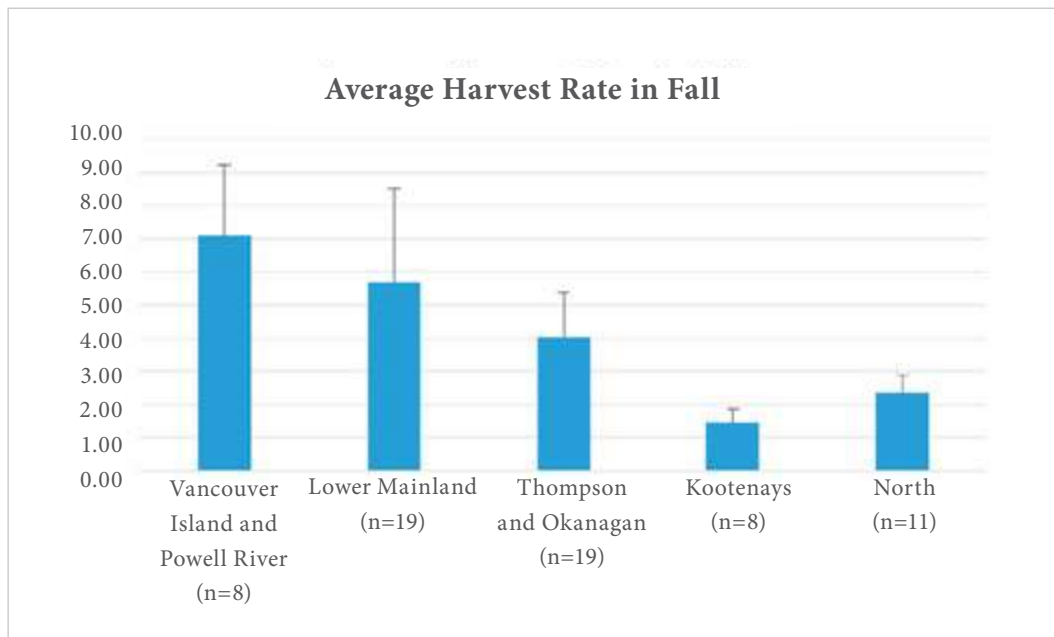


## SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA

also that the two decimal places on the vertical axes are unnecessary, and should be removed. You will become familiar with creating charts with practice. The key is simplicity – less is better – so that the data are on display, not the chart formatting.



# SUMMARIZING AND VISUALIZING PATTERNS IN YOUR DATA





# GETTING STARTED WITH R

## WHAT IS R AND WHY SHOULD YOU USE IT?

R is a programming language, and is equivalent to having many different computer programs all in one place. People worldwide have contributed to R more than 4,000 different software programs, which are referred to as R 'packages'. Thus, R is much more powerful than any one 'canned' software program. R users have access to a vastly wider array of statistical tests, and much more control over the inputs and outputs of statistical analyses.

Not only is R powerful, it's also free. Yahoo!

And not only is R powerful and free, it's also really user-friendly, provided you understand the language and get over the initial, rather steep and somewhat frustrating learning curve. One of the reasons R is user-friendly, is because you can access help readily. R users around the world communicate regularly through email, and the communications are posted on the R website. So if you have a particular problem, chances are someone else has had that problem too. If you 'google' your problem, you'll encounter relevant discussions on the web.

R also has an online help function, and, it's actually helpful!

## DOWNLOADING R


Using an internet browser such as 'Explorer' or 'Firefox' or 'Google Chrome', go to this website:

<http://cran.stat.sfu.ca>

Click on **Download R for Windows** (or Download R for (Mac) OS X if you have a Macintosh computer). On the **R for Windows** page, click on **install R for the first time**, then on the next page click on **Download R 3.3.x for Windows**. (the x refers to the version number) The executable file titled **R-3.3.x-win.exe** (.x refers to the version number) should now automatically download to your computer. Double click on the file, and then click on Run to install the R onto your computer.

## GETTING STARTED WITH R

In the Setup window (**Welcome to the R for Windows 3.2.x Setup Wizard**), press **Next >** for the following six windows (press **Next >** seven times), until the program begins installing. Once R has been installed, click **Finish**.

You should now see two icons for R on your desktop. One icon **R i386** is for a 32 bit operating system, and the other **R x64** is for a 64 bit operating system. It is likely that your computer is a 64 bit operating system, so you can just delete the R i386 icon and use the R x64 icon to interface with R from your desktop. To make sure your operating system is 64 bit, open **System** by clicking the **Start** button , right-clicking **Computer**, and then clicking **Properties**. The **View basic information about your computer** will open, and next to **System type**: the operating system will be shown.

## UNDERSTANDING HOW R WORKS

When you open R for the first time, you'll see a couple of menu buttons on the top of the screen, and then the rectangular R console box with this...

```
> |
```

... blinking at you expectantly. Now what, says you?

## Operators, Objects, and Workspace

Let's get started simply by mucking about.

```
27 + 30 - 23
```

You can see that R can function like a simple calculator. The + and - are called **operators**.

```
1:10
```

## GETTING STARTED WITH R

The `:` operator means 'output this range of numbers'.

```
10:1
```

```
100^2
```

The `^` operator means 'to the power of'

```
(100^2)/(400 + 23.4)
```

Note that just as with excel, you need to put brackets around calculations that need to be performed first. Otherwise, `100^2` would be divided by 400 and then 23.4 would be added to the result.

Here are some other simple operators:

Operator	MATHEMATICAL
<code>+</code>	addition
<code>-</code>	subtraction
<code>*</code>	multiplication
<code>/</code>	division
<code>^</code>	exponentiation, i.e. to the power of x
	LOGICAL
<code>&lt;</code>	less than
<code>&lt;=</code>	less than or equal to
<code>&gt;</code>	greater than
<code>&gt;=</code>	greater than or equal to
<code>==</code>	exactly equal to
<code>!=</code>	not equal to
	OBJECT ASSIGNMENT
<code>&lt;-</code>	'object name' is right-hand input (e.g. data, stats analysis, graph plot, etc)
<code>\$</code>	A referencing operator used to refer to a column within a dataframe, e.g. dataframe\$Column1

An important thing to understand is that R is an object-based programming language. The primary way you tell R to do something is by assigning data and analyses to **objects**. And the main way you do that is by the **assignment operator** `<-`

## GETTING STARTED WITH R

To show you how to assign something as an object, put an object name and the object assignment operator `<-` before the calculation. Let's assign this calculation to the object 'goats'. You can use whatever object name you want.

```
goats<-27 + 30 - 23
```

What happened when you pressed the enter key? R saved your calculation as the object 'goats' in your **workspace**.

The workspace is a virtual memory space for your current analysis session. Anytime you assign something to an object, it is then stored in your workspace. You can then retrieve the something (data, stats analysis, graph plot, etc) simply by typing the object name.

```
goats  
[1] 34
```

Or you can do further analysis on the something by inserting its object name into further calculations. Note the number 1 in the square brackets above is just R telling you that there's only one result to return to you (34).

If you want R to remember the output, you have to assign the calculation to a new object. If you use the same name, R will overwrite and thus you won't be able to retrieve the previous object.

```
goats333<-goats-2  
goats333  
[1] 32
```

Here's the original object retrieved ...

```
goats  
[1] 34
```

## GETTING STARTED WITH R

Just to prove that R will overwrite your first object, let's do all that again ...

```
goats<-27 + 30 - 23
```

```
goats<-goats-2
```

#Do this on your calculator to satisfy how R can be used as a calculator.

```
goats
```

```
[1] 32
```

So, an important thing to keep in mind is that R only 'remembers' the last object assignment you gave to the workspace.

### Exercises

Assign the calculation 9 squared times 2 to the object x. What is x?

Assign the calculation x to the power of 4 to the object aa. What is aa?

Assign the value 8 to x.

Did changing the value of object x change the value of object aa? How can you quickly find out the new value of aa if you change x to 8?

Suppose you've been working all day and you've lost track of how many objects you've assigned, or maybe you just want to clear the whole thing and start again.

**ls()** Lists all objects in your workspace

**rm(x)** Remove object x from your workspace

**rm(list=ls())** Removes all objects from your workspace

Now make sure your workspace only contains the object aa.



## GETTING STARTED WITH R

### Packages, Functions, Arguments, and Scripts

Almost all of R's ability to run statistical analyses comes from programs called **packages**, which must be loaded into R in order for them to function. A bunch of packages are already loaded into R; these are the default packages.

To see a list of default packages loaded onto R type

```
search()
```

There are over 4000 R packages, but in order to use one of them, you first need to download it onto your computer. Press the **Packages** menu button on the top of the window, and then **Install Packages**. You'll see that you need to choose a **Cran mirror**, which is a server from which to download the packages. Choose **Canada (BC)**. If **HTTPS CRAN mirror** opens first, scroll to the bottom of the list and select (HTTP mirrors), then select **Canada (BC)**. Next a window will open called **Packages**, which lists all of the available packages to download. Click on the package **gam** (which stands for Generalized Additive Modelling), and then let it download.

That's the first step. Now you have to load the package in order to use it. Under the **Packages** menu, simply click on **Load Package**. The packages you have installed on your computer will be listed. Now click on the package you want to load, and it will now be ready for you to use.

Alternatively, to load a package you can type

```
library(package name)
```

In this case we type

```
library(gam)
```

### Troubleshooting

If you have difficulty getting packages to load once they have been installed on your computer, it may be because the anti-virus software on your computer is blocking R files. Most anti-virus software allows you to manually select programs that you want don't want to block. A google search can usually help you find out to do this for the anti-virus software installed on your computer.

So how do packages work? Packages are created around specific types of statistical analyses, and are usually created by statistician programmers who are considered to be experts in that particular type of analysis. Often the names of the packages are abbreviations of the type of analysis, e.g. gam is short for Generalized Additive Modelling.

Packages work because they contain **functions**. Functions are the engines of the stats analysis. Most packages contain several functions.

If you want to see the functions contained in a package, type

`ls('package:package name')` e.g.

```
ls('package:gam')
```

So what is a function? Let's work with something familiar. First let's figure out what simple packages are loaded as default programs.

```
search()
```

The package 'base' contains all of the basic functions needed to run basic analysis. Let's practice loading a package by loading that one (it's already loaded as a default package so there's actually no need to load it, but we're practising).

```
library(base)
```

## GETTING STARTED WITH R

Now let's see what functions are available in the package base.

```
ls('package:base')
```

Scroll to line [691]. The function “mean” is a function built into R to calculate the average of a group of numbers.

To check what a function does, you can access the online help for any function simply by typing

```
?function name
```

for example

```
?mean
```

R webpages for functions can appear a bit daunting at first. But once you get used to the language, they're actually very helpful.

Usage

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

After the description of the function, the full extent of all possible attributes for the function are given, with all possible inputs. Here you can see why R is a powerful program. This is a function simply for calculating the average of a group of numbers, and R gives you at least two additional options for calculating the average exactly the way you want it calculated.

At this point, you simply need to understand that a function, in this case, mean, is comprised of **arguments**. Arguments are the inputs into the function. Entering values into the arguments of functions is where YOU, the R User Statistician Nerd, enter into the equation. ha ha.

The trim argument of the mean function allows you to cut off data from either end of your group of numbers to make a smaller data set, and then the average is calculated from the smaller data set.

The na.rm argument is a very common argument. It's shorthand for 'NA remove'. NA is shorthand for

## GETTING STARTED WITH R

‘not applicable’ or ‘not available’, which is simply what you would enter in your database if, for example, you were missing data for a particular measurement. You can input TRUE into the na.rm argument, in which case R will ignore the NAs in your dataset and calculate the average as if those values did not exist at all. If you do not specify anything for the na.rm argument, R will use the default value, which in this case is FALSE. The default is to not remove NA values from the dataset – if NA values are present, then the average cannot be calculated.

Let’s see how all this works.

First, let’s create some data. Imagine we had just caught 25 geese, and we weighed each goose in pounds and rounded up to the nearest pound.

Our data set then is the weights in pounds of 25 geese. To create this data set in R, we need to do three things – we need to write out the weights for each goose, we need to group the 25 weights together into one data set, and we need to assign an object name to our data set.

```
geeseweights<-c(16, 10, 4, 9, 17, 12, 19, 11, 10, 10, NA, 16, 15, 14, 10, 8, 9,  
17, 9, 15, 16, 13, 12, 8, 7)
```

geeseweights is the object name

<- is the assignment operator. This assigns everything to the right of the operator to the name on the left of the operator.

The ‘c’ is a function, and stands for concatenate, which simply means ‘stick together into one thing’.

Brackets are used after function names and basically mean, ‘apply this function to the numbers inside the brackets’.

And the brackets go around our data set of goose weights. Note the comma in between each weight.

Also note that one goose flew away before we could weigh it. Hence the NA value.

## GETTING STARTED WITH R

So, now we've just said to R – “here's a bunch of numbers, stick them all together, and call this bunch of numbers `geeseweights`”. Please. Thank you.

Now type `geeseweights` and R will recall your data set.

To check and make sure that all 25 weights were entered, you can use the `length` function, which computes the number of elements in an object.

```
length(geeseweights)
```

```
[1] 25
```

Now we're ready to try out the `mean` function! (mean as in average, not mean as in not very nice)

```
mean(geeseweights)
```

```
[1] NA
```

Oops. What happened? Hmm, let's find out. Back to our help page for the `mean` function

```
?mean
```

Aha!

```
## Default S3 method:
```

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

On the help page it states that the default for the `na.rm` argument is `FALSE`. That means that if you don't specify a value for the `na.rm` value, which we did not in our line of code above, then R uses the default which is `FALSE` in this case. That means that the `NA` values were not removed from the dataset before R tried to calculate the average. Since there was an `NA` value in our dataset, the average could not be calculated, because a value was missing.

So, we need to write in the `na.rm` argument into the coding for the `mean` function, and set it to `TRUE`.

## GETTING STARTED WITH R

We do indeed want R to remove the NA values, and then calculate the average of the goose weights.

```
mean(gooseweights,na.rm=TRUE)
[1] 11.95833
```

You can check that this is correct, by using R as a calculator (or by using a hand calculator).

```
sumgooseweights<-sum(gooseweights,na.rm=TRUE)
sumgooseweights
[1] 287
average<-sumgooseweights/24
average
[1] 11.95833
```

Remember that we divide by 24, not by 25, because we had one missing (NA) value.

Probably by now you have a sense that operators and functions are two different things. Operators are the very basic language of R, and are like the buttons on a calculator. Functions are like task managers, designed to perform a suite of tasks, which are defined by the values you give to the function's arguments.

## IMPORTING, EXPORTING, AND MANAGING DATA

R requires a folder on your computer from which to retrieve data. It will also output data for some functions into this folder. Call the folder whatever you want, the important thing is that you tell R where to find the folder by using the “set working directory” function `setwd`. It's best to create a folder just for R on your desktop and store everything you need for R in it. You might call the folder ‘R Working Folder’.

The working directory for your R working folder might look something like this.

## GETTING STARTED WITH R

```
setwd('C:/Users/yourname/Desktop/R Working Folder')
```

To find out exactly what it is, right click on the folder, then click **Properties**. Under the **General** tab, you'll see **Location**: and that's what you need to write in between the brackets of the `setwd` function. But note that you need to replace backward slashes `\` with forward slashes `/`. You also need to add the name of your R working folder.

Now that we've set the working directory, we can import data into R.

You import data into R using a command. In order for R to recognize the data, it must be saved as a 'comma separated values' file, or `.csv` file.

Let's work with the excel file provided in the CD.

Use excel to open the file `example_harvest_data.xls`

Under the file menu in excel, click on 'save as' and then select your R working folder. Rename the file simply 'harvest' in the **File name**: window. In the **Save as type**: window, scroll to **CSV (Comma delimited)**. Note that the file extension changes to `.csv`. Also note that there are three `.csv` file types, you don't want the macintosh or msdos types. Then click on save, and check that 'harvest.csv' appears in your R working folder.

In R, you can retrieve the data from the working folder using the **read.csv** function. You need to assign it to an object. Let's call the data `h` for harvest

```
h<-read.csv("harvest.csv")
```

To see the data, simply type `h` and R will display your data set.

### Troubleshooting

Did you just get this error message? If so, then this is your first lesson in how finicky R can be.

**Error: unexpected input in "h <- read.csv(""**

R is telling you that it doesn't like your quotation marks. R needs straight up and down quotation marks, and not slanting quotation marks.

Not this: “10”

This: "10"

To get straight quotation marks, you'll need to set the preference for that in a word processing program like Word.

In Word, on the **Tools** menu, click **Options**, then click **Proofing**, and then click **AutoCorrect Options**. In the **AutoCorrect** dialog box, click the **AutoFormat** tab, and under **Replace**, select the “**Straight quotes**” with “**smart quotes**” check box.

There are a number of different ways to export a dataframe from R. In this script, you would just change ‘mydata’ to whatever name you gave your dataframe.

To tab delimited text file:

```
write.table(mydata, "c:/mydata.text", sep="\t")
```

To an excel spreadsheet:

```
library(xlsx)
```



## GETTING STARTED WITH R

To an html file in your R working folder. You then open the file called 'mydata' in your R working folder in a web browser, press control and 'a', then control and c for copy, then you can paste into excel.

```
library(xtable)
mydata<-xtable(mydata)
print(table, type="html", file="mydata.html")
```

## Classes of Data

One of the most important first steps once you've imported your data is to make sure that R knows what kind of data you have. There's some terminology you need to be familiar with first. These terms are used in the help webpages for all the R functions, so it's important you understand what these terms mean.

A **vector** is a string of values. Usually your data columns are the vectors you'll be concerned with; each column is a vector. Sometimes we might want to create vectors. We can create vectors using the concatenate, sequence, and replicate functions.

```
c(1,40,37,3)
seq(1,100,1)
rep(1:100,3)
```

A **dataframe** in R is simply a dataset stored in your workspace as a table. You need to make sure your data is set up the way R reads it. R reads each column of data in the dataframe as a variable (vector), and each row contains the observations collected for each sample unit. In our case, each row should be the harvest data for one hunter.

You can create a dataframe in R, using ... ta da! the dataframe function

```
dataframe1<-data.frame(hunters=1:20, regions=1:5, seasons="Fall")
dataframe1
```

## GETTING STARTED WITH R

Now that you understand vectors and dataframes, it's important that you understand what kind of data you have, and to make sure R has classified your data the way you want. So, you need to understand the concept of classes. In R, objects fall into a number of different classes – a function is a class 'function', a dataframe is a class 'dataframe, and vectors can be one of a number of different kinds of classes. And you can tell R to change the class of a vector from one to a different class. If you want to check to make sure that R has coded your data columns correctly, you can use the class function, and R will output the class type for your data column.

Here are some common examples:

```
class(c(1.2,1.5,1.9))  
[1] "numeric"
```

```
class(c(1:39))  
[1] "integer"
```

```
class(c(TRUE,FALSE))  
[1] "logical"
```

```
class(c("one", "1.2", "thirty nine"))  
[1] "character"
```

Now let's focus on learning what happens when we convert vectors from one class to another. We can use the 'as.\_\_\_\_\_' function to convert vectors from one class to another

```
ducks<-c("one", "1.2", "thirty nine")  
class(ducks)  
[1] "character"
```

```
ducks2<-as.numeric(ducks)  
class(ducks2)  
[1] "numeric"
```

## GETTING STARTED WITH R

```
ducks2
```

```
[1] NA 1.2 NA
```

Notice what happens when we try to convert a character vector to a numeric vector – all character elements are returned as NA, and all numeric elements are returned as numbers.

And now to find out the classes of each vector in our dataset ‘harvest.csv’ we can use the structure function `str`

```
str(h)
```

Age is classed as an integer (int), latitude and longitude as numbers. If we had a vector of true and false in our dataset, R would class it as ‘logical’.

R tells us that it recognizes HunterID as a factor with 250 levels, meaning there are 250 unique identification numbers of hunters. A factor is a special case of a character vector, for which each unique level is assigned a number. Categorical variables have levels designated by text and can be treated in R as character vectors or as factor vectors. An example is the categorical variable Season in our harvest data, which has four categories of season.

R by default imports categorical variables as factor vectors. It does this because character vectors can’t be used for any kind of numerical analysis. Also by default, R assigns numeric levels to the factor categories alphabetically. Depending on the statistical modelling you plan to do, you may need to pay attention to this because you may need to re-order the levels for your categorical variable. For example, R would assign level 1 to ‘Fall’ because it’s first in the alphabet. But you may want to re-assign ‘Winter’ to level 1 if you suspect, for example, that hunting is low in winter, such that winter becomes a reference against which you compare hunting in all other seasons.

We’ll come back to how to do this later, for now let’s just play around with converting a factor vector to a numeric vector.

```
data<-c("good", "seven", "5", "10")
```

## GETTING STARTED WITH R

```
data2<-as.factor(data)
data3<-as.numeric(data2)
data3
[1] 3 4 2 1
```

Whoa nelly! What happened?

The `as.numeric` function is simply changing the numeric values assigned to the factor levels to numbers, rather than the actual value of the numbers. To correctly convert the numbers contained within factor vectors to numbers in a numeric vector, you must first convert to a character vector, then convert that to a numeric vector.

Here's a list of common conversions for changing vector classes:

- `as.character`
- `as.numeric`
- `as.integer`
- `as.logical`
- `as.factor`
- `as.Date`

## Missing Values

The easiest way to deal with missing values is by creating a dataset in excel with just the variables you're interested in analyzing, and then delete the rows with missing values, before you import your data into R.

You can create a data set without missing values in R easily. The `na.omit` function is used to delete missing values

```
catz<-c(1:120, NA, 1:4)
catz
```

## GETTING STARTED WITH R

```
catz2<-na.omit(catz)
catz2
```

Now let's try with our data set

```
h2<-na.omit(h)
h2
```

Oops, what's the problem here? Using `na.omit` on a data frame results in all rows containing an NA value being deleted from the data frame. Removing rows with NA for some variables means you've just cut out valuable data that you do have for other variables.

Because missing values are so commonplace in datasets, most R functions contain an argument ('na.action') for dealing with missing values, which are coded by R as NA (not available). Usually the default is `na.action=na.omit`, so therefore you don't have to worry about it. The function automatically ignores missing values.

It's important to keep in mind that some functions don't have an NA argument. If you find yourself trying endlessly to get a function to work, it may be because the function needs a dataset free of NAs in order to work, or the function doesn't have a default `na.action` set to `na.omit`. Some of the most basic functions have a `na.rm` (na remove) argument instead of an `na.omit`, and it needs to be set to `TRUE` to work. We discovered this fact when we used the `mean` function.

## Referencing and Sub-setting

At some point, you'll find it necessary to be able to extract parts of your data frame to analyze separately from the whole data set. Again, sometimes it's easiest just to create a subset of your data in excel and then import it into R. But, real R users know how to subset, so we'd better learn.

There are two main ways to extract portions of a data frame. The `[ ]` operator and the `subset` function. The `[ ]` operator has two positions separated by a comma, the first is to specify the row (s) you want

## GETTING STARTED WITH R

extracted, the second is for columns. [row, columns]

```
h[ , "Age"]
```

or you can specify the 'Age' column by its order in the dataframe as the second column

```
h[ , 2]
```

Rows are specified by their number

```
h[1:11]
```

hmm... that didn't work. Why? We left out the comma! Try again

```
h[1:11,]
```

We can also use the **subset** function to do the same thing, but it's more flexible. The arguments are as follows: subset(dataframe from which you want to subset, column title == "level of the factor by which you want to subset")

```
subsetharvest<-subset(h, Region=="Kootenays")  
subsetharvest
```

Note the == operator – this means exactly equal to and is needed when specifying categorical values of vectors. Here we have just subset our harvest dataframe to pull out just the Kootenay data.

## GRAPHING IN R

FUN!

R has a huge graphics capacity and it's worthwhile to gain some proficiency in graphics. There are a lot

## GETTING STARTED WITH R

of options that you don't normally find in 'canned' packages like Sigmaplot. For example, it's relatively easy to create a three-dimensional plot in R. There are a lot of graphics packages, but the default graphics packages in R are pretty good and will likely fill your graphics boots. To see a bunch of options type

```
demo(graphics)
```

Each time you create a new plot in R it will over-write your existing plot. To avoid that, use the `windows()` function to create a new window.

```
windows()
```

Then you need to set up the amount of white space between the plot and the window border. You can change these numbers depending on your preferences, but this is a standard layout for plots that tends to work well.

```
par(mai=c(1, 1, 0.1, 0.1))
```

Then you're ready to create a plot.

```
plot(Quantity/Effort ~ Age, data=h)
```

Congrats! You've made your first data plot in R. This is a simple plot of the total birds harvested by each hunter, standardized by their effort (number of days that they hunted), plotted against the age of the hunter. This is across all seasons, and so hunters that hunted in more than one season have more than one data point on the plot. The plot clearly shows that middle-aged hunters, those around 30 to 55 years old, are more successful than younger or older hunters.

Note that the vertical y axis is given before the `~` operator, and the x axis is given after. The `~` operator means "plot this against that". Note also that we had to use the data argument. This is a very common argument in functions; it tells R in which dataframe to find the variables you want to plot.

## GETTING STARTED WITH R

The plot function has many arguments. In the code above, we only used the two essential arguments. Without those two arguments the function would not have worked. From the basic function, we can build a plot using additional arguments to suit our preferences.

```
plot(Quantity/Effort~Age, xlab="Hunter Age", ylab="Bird Harvest", cex.axis=2,  
cex=2, data=h)
```

The xlab and ylab arguments add x and y labels to the plot, while the cex.axis argument changes the size of the numbers on each axis. The cex argument changes the size of the data points.

Here are some additional arguments for the plot function

**xlim=c(lower limit,upper limit)**

specifies lower and upper limits for the x axis

**ylim=c(lower limit,upper limit)**

specifies lower and upper limits for the y axis

**add=TRUE**

superimposes a plot on the previous one

**axes=FALSE**

adds just the plotted data to a plot (no axes or box)

**type="p"**

"n" nothing plotted, "p" points, "l" lines, "b" points connected by lines, "h" vertical lines, "s" top of vertical lines, "S" bottom of vertical lines

And here are some functions that will add bits and pieces to your plot at the x, y coordinate that you specify. The x, y coordinate is dependent on the data you plotted, e.g. if your x axis is a plot of data from 0 to 210, and y axis from 0 to 100, then you can plot a point in the middle of your graph by specifying 105, 50.



## GETTING STARTED WITH R

### **legend(x,y,legend)**

The location may also be specified by setting x to a single keyword from the list "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center"

e.g.

```
legend("topleft", legend="Density", pch=16,  
col="blue")
```

### **title()**

Adds a title and sub-title

### **points(x,y)**

Adds points, and 'type=' can be specified

### **lines(x,y)**

Adds lines

### **text(x,y,label)**

Adds text at x,y given by label.

## USING SCRIPTS

As you can probably guess by now, using R requires you to constantly write out code. Rather than having to write fresh code every time you use R, most R users write scripts for analyses in a word processor such as Word.

It's a good idea to start all of your scripts with the following commands – the first sets the working directory, which you must do every time you open R, the second creates a clean slate (which is clean if you've just opened R, but not if you've been working all day), and the third imports your data into R as an object. I'm assigning my data to the object 'bird'.

## GETTING STARTED WITH R

```
setwd('C:/Users/yourname/Desktop/R Working Folder')  
rm(list=ls())
```

You can organize code and write notes to yourself within your code by using the # symbol followed by the title for each analysis. When you copy and paste the code into R, any text following the # sign is ignored by R as just text.

Here's the full script for the plot coding that we just finished, which we could save in a Word file and label using an informative name like 'harvest analysis R code'.

```
setwd('C:/Users/yourname/Desktop/R Working Folder')  
rm(list=ls())  
h<-read.csv("harvest.csv")  
windows()  
par(mai=c(1, 1, 0.1, 0.1))  
plot(Quantity/Effort~Age, xlab="Hunter Age", ylab="Bird Harvest", cex.axis =  
1.5, cex.lab = 1.5, cex = 1, col = "green", pch = 3, data=h)
```

And once again, a reminder that you have to be as finicky with your coding as R demands.

This code below resulted in an error. Can you tell why?

```
plot(Quantity/Effort~Age, xlab="Hunter Age", ylab="Bird Harvest", cex.axis =  
1.5, cex.lab = 1.5, cex = 1, col = "green", pch = 3, data=h)
```

Error: unexpected symbol in "plot(Quantity/Effort~Age, xlab="Hunter Age", ylab="Bird Harvest", cex.axis = 1.5, cex.lab = 1.5, cex = 1, col = "green" pch"

You can use the error statement to try to figure out what went wrong. Here R returns the code up to the point that it encountered an error. So from this you can tell that there was some kind of error around the pch argument. Right. R expected a comma in between the col argument and the pch argument. The correct code is:

## GETTING STARTED WITH R

---

```
plot(Quantity/Effort~Age, xlab="Hunter Age", ylab="Bird Harvest", cex.axis =  
1.5, cex.lab = 1.5, cex = 1, col = "green", pch = 3, data=h)
```



# STATISTICAL MODELLING OF YOUR DATA

As we have seen so far, statistics is the science of measuring a small part of the world to say something truthful about a much bigger part of the world. We're using the harvest levels of a sample of hunters to estimate the total harvests of *all* hunters.

So then, let's say that we conducted a harvest survey, analyzed the data, and estimate that the total harvest of all birds in 2015 was 12,713 birds. Does it feel like something is missing? If someone said to you, 'I think this is true', isn't there a tiny voice inside you asking, 'how certain are you about that?'

Statistics is not just about estimating characteristics of populations, such as the average or standard deviation. Such characteristics of populations are referred to as **population parameters**.

Inherent in any estimate is **uncertainty**. How certain we are about our estimates of population parameters is probably the most important aspect of statistical analysis. Perhaps it isn't a life or death situation when estimating bird harvesting (at least not for the humans), but there are many instances on a daily basis where lives depend on the certainty of statistical estimates. Think about the science behind sending a space shuttle to the moon, and the administering of the correct dosage of medicine in emergency rooms.

## HOW DO WE MEASURE CERTAINTY?

Suppose for a moment that we had conducted a harvest study, and we measured the average number of birds harvested per hunter. We found the average to be 2.2 birds per hunter. We state that we estimate the population average is 2.2 birds per hunter. How can we be certain that our sample average of 2.2 is a good estimate of the population average?

**We could repeat the study again.** If we get a similar result, it suggests that our estimate is good. If we get a much different result, we know that either our first or second or both estimates were off. It may be intuitively obvious that the more we repeat studies, the more certain we will become of the true population average, because eventually most of the sample averages will begin to cluster around the true population average.

## STATISTICAL MODELLING OF YOUR DATA

It might also be intuitively obvious that the more variable the population, then the more variable our samples will be, thus the more variable our sample averages, and the more times we need to repeat the study until the sample averages begin to cluster around the true population average.

Let's see this in action. Our imaginary situation is that we have just weighed in pounds every single individual of a duck species. There are 30,000 individuals of this duck species in the whole world. The population is thus 30,000 individual ducks.

Open R. Review [chapter 5](#) if you need help getting started.

Create a column (a 'vector') of data in R using the following code.

```
set.seed(1234)
x<-rnorm(30000, mean=12.3, sd=2.1)
populationweights<-round(x,0)
#the round function with '0' in the decimal argument tells R to round up to whole numbers

#Calculate the population average and the population standard deviation.
mean(populationweights)
12.3

sd(populationweights)
#though, actually, R sd function is for the sample sd not the population sd but the difference is
negligible for a population of 30000
2.12
```

Now we will take random samples of 12 duck weights per sample from the population of 30000 duck weights, and calculate the average of each random sample. Actually, rather than repeating that code 'manually', here is a simple function that will run that code for you as many times as you want. It's been set to repeat the code 100 times, which results in 100 sample averages, of a sample of 12 duck weights per sample, from a population of 30,000 duck weights. You just need to change the 100 in bold if you want more or less sample averages.

## STATISTICAL MODELLING OF YOUR DATA

### Sample 1

```
set.seed(1234)
sample_duckweights<-
sample(x=populationweights, size=12,
replace=TRUE)
sample_duckweights
[1] 14 9 13 12 11 11 11 13 13 12 16
13
mean(sample_duckweights)
[1] 12.3
```

### Sample 2

```
set.seed(1235)
sample_duckweights<-
sample(x=populationweights, size=12,
replace=TRUE)
sample_duckweights
[1] 12 12 14 11 13 11 13 11 16 12 11
11
mean(sample_duckweights)
[1] 12.3
```

```
sampleaverages<-rep(NA,100)
n<-100
for(i in 1:n) {
set.seed(i)
randomsample<-sample(x=populationweights, size=12, replace=TRUE)
sampleaverages[i]<-mean(randomsample)
}
```

What we've just done in essence is conduct 100 individual studies to measure duck weights. In each study we randomly sampled 12 ducks, weighed them, calculated the average of the 12 weights, and entered the average into our 'sample averages' data set. **By the way, we put the ducks back into the population after we weighed them.**

Here is our sample of 100 sample averages:

```
round(sampleaverages,2)
[1] 12.75 13.17 11.42 12.83 12.42 13.08 12.67 12.33 12.83 12.92 11.92
[12] 12.17 11.08 12.42 12.33 12.50 12.08 12.50 13.25 12.67 11.17 11.17
[23] 11.67 12.25 11.92 12.00 12.92 11.17 13.75 12.92 12.58 13.75 12.00
[34] 12.17 11.83 12.83 11.67 11.92 12.17 11.92 11.75 12.33 12.25 12.08
[45] 12.67 12.00 12.00 13.75 11.58 12.75 11.33 12.50 13.58 12.25 11.17
[56] 13.08 11.42 11.17 12.42 12.00 11.58 12.17 11.17 12.50 14.08 12.33
```

## STATISTICAL MODELLING OF YOUR DATA

```
[67] 12.75 11.58 12.08 11.50 12.17 11.75 10.92 12.17 11.50 12.00 11.83
[78] 12.75 12.58 12.67 13.00 12.58 11.83 12.83 11.67 12.33 12.08 11.58
[89] 12.75 12.50 11.75 12.50 12.83 12.08 11.50 12.00 12.83 10.83 12.08
[100] 11.42
```

Remember that our population average, that is the true average, is 12.3 pounds. You can see from our sample of sample averages that some are pretty close to the true average, some are up to 1.8 pounds heavier, and some up to 1.5 pounds lighter (`sort(sampleaverages)`).

We expect there to be clustering around the true population average, that is, we can measure the central tendency of our sample of average duck weights by measuring the *average* of our averages.

```
mean(sampleaverages)
12.2
```

Pretty good. That's pretty close to the true population average of 12.3.

**But in terms of actually measuring certainty, it would be helpful to know how many sample averages are close to the true average versus how many are further from the true average.**

To begin to measure certainty around our estimate of the true population average, we count the number of sample averages within say half pound intervals across the range of sample averages, from 10 up to 14.5. That would give us an actual measure of the amount of sample averages that were close to the true mean. We could do this manually, but heck, let's get R to do all the work for us.

```
h<-hist(sampleaverages, plot=FALSE)
h$breaks
[1] 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5
h$count
[1] 2 14 24 29 22 4 4 1
```

This means that of our 100 sample averages, two averages are within 10.5 to 11 pounds, 14 averages are

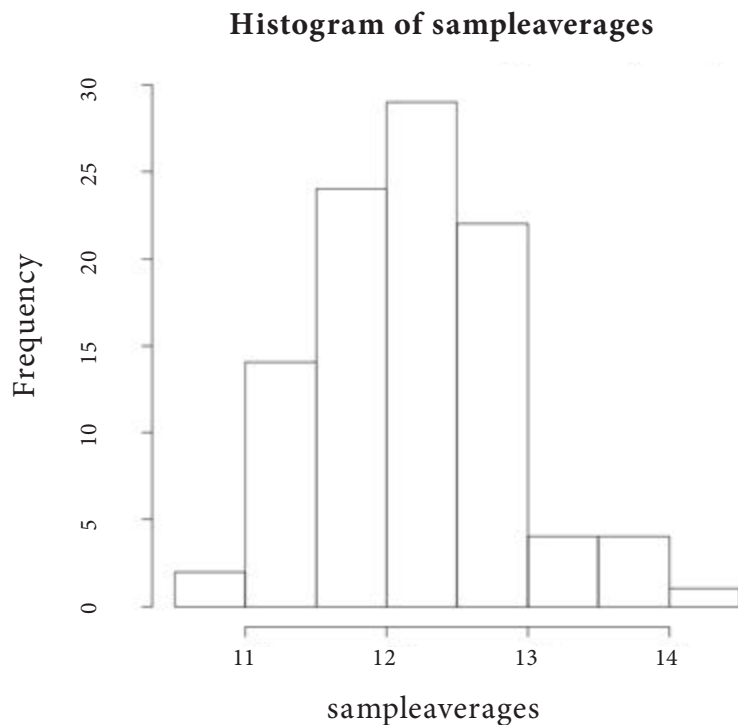


## STATISTICAL MODELLING OF YOUR DATA

between 11 and 11.5 pounds, and so on. We can view the distribution around the average of the sample averages as a plot, referred to as a **histogram**.

```
h<-hist(sampleaverages, breaks=9)
```

```
h
```



On the vertical axis is the count of the number of sample averages per interval of sample averages. We can see visually that 14 sample averages were between 11 and 11.5 pounds, etc.

This is referred to as a **sampling distribution**; it shows in a histogram the distribution (i.e. counts per interval) of sample estimates across the range of sample estimates.

This sampling distribution is based on 100 samples of averages. What if we had taken 1000 samples instead? We just need to change our code a wee bit:

```
sampleaverages<-rep(NA,1000)
```

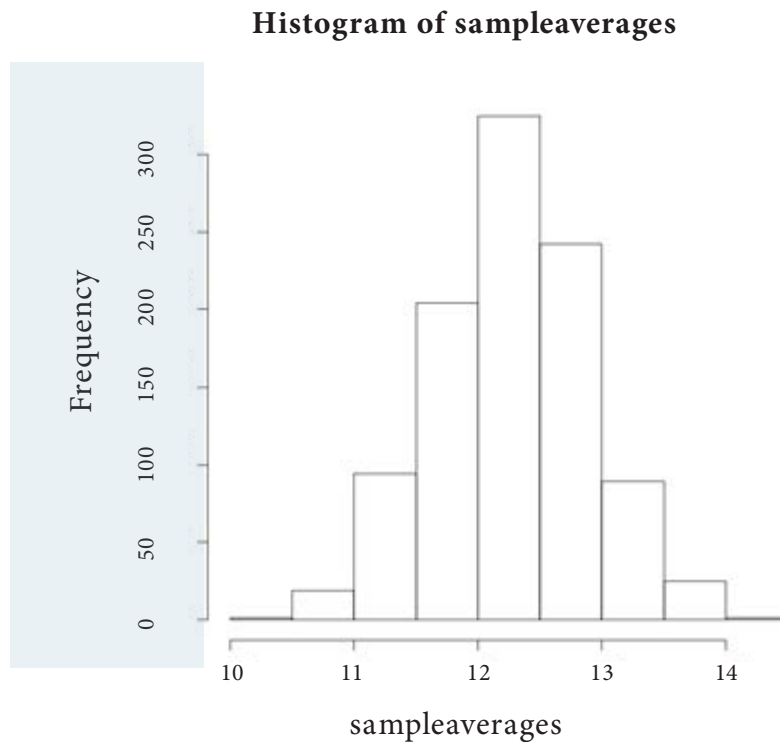
```
n<-1000
```

```
for(i in 1:1000) {
```

## STATISTICAL MODELLING OF YOUR DATA

```
set.seed(i)
randomsample<-sample(x=populationweights, size=12, replace=TRUE)
sampleaverages[i]<-mean(randomsample)
}
hist(sampleaverages)
```

The **shape of the sampling distribution** didn't really change, but it did become smoother, that is, more symmetrical. For example, there were the same number of averages in the intervals 4 and 10 (corresponding to averages from 9.0 to 9.5, and 12 to 12.5). Also, the counts of averages on the vertical axis changed, because the sample size increased from 100 to 1000.



Let's try 10,000 samples

```
sampleaverages<-rep(NA,10000)
n<-10000
```

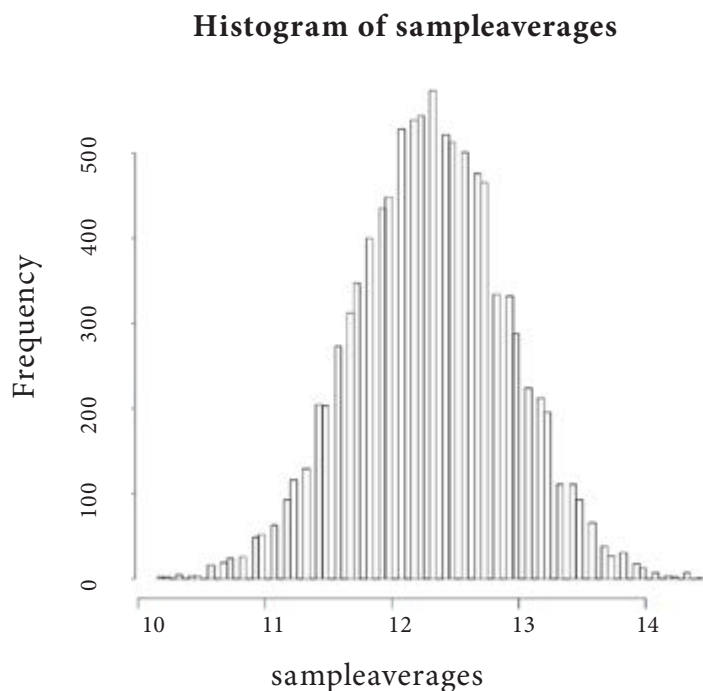
## STATISTICAL MODELLING OF YOUR DATA

```
for(i in 1:10000) {  
  set.seed(i)  
  randomsample<-sample(x=populationweights, size=12, replace=TRUE)  
  sampleaverages[i]<-mean(randomsample)  
}  
h<-hist(sampleaverages,breaks=12)
```

That's even smoother. If we had a very large sample of estimates of the population average derived from random sampling from a population of a continuous variable, each column on the histogram would become tiny. As our sampling increases to infinity, the separate columns would merge into one smooth curve. The distribution of parameter estimates then becomes the **normal distribution**, otherwise known as the normal curve.

Here's what the same histogram looks like with more intervals ('breaks')

```
h<-hist(sampleaverages, breaks=120)
```



The following is true for any sample of averages from any population of anything on earth – the

## STATISTICAL MODELLING OF YOUR DATA

sampling distribution of the sample averages will approach the normal distribution as sample size increases.

**One way to derive a measure of certainty around our estimate of the true population average, is to measure the variability in the sampling distribution of sample averages.** The standard deviation of the sampling distribution is a measure of the variability we could expect across sample averages if we were to repeat the study many times. The standard deviation of sample averages is referred to as the standard error of the mean. **The standard error of the mean is the standard deviation of averages calculated from random samples drawn from a population. More generally, the standard error of a sample statistic, is the standard deviation of the sampling distribution of the statistic. For example, we can also estimate the standard error of a regression coefficient.**

If the standard deviation is high relative to the average, we expect that our repeated draws of samples would have very different averages. That would mean we would be relatively uncertain that any one particular sample average is close to the true population mean. If the standard deviation of sample averages is low, then we expect the opposite – that with each repeat study, we would get similar sample averages, with most of them being close to the true mean.

In this imaginary situation, we know the true population weight and we've been pretending to conduct a very large number of repeat studies. In reality, we don't know the true population parameter – in fact the reason we conduct studies is to estimate it. And we usually only have the time and money to conduct one study.

Now what? Well, we're in luck. Turns out that we can estimate the standard deviation of the sampling distribution of the sample averages using the standard deviation of **one** sample.

So again, the standard deviation of one sample of 12 duck weights drawn randomly from our population of 30,000 duck weights can be used to estimate the standard deviation of sample averages that would be measured if we had repeated the study many times. The standard deviation of individual samples tends to over-estimate the standard deviation of sample averages, so the sample standard deviation needs to be divided by the square root of the sample size, in this case by the square root of 12.

The standard deviation of our sample of 10,000 average duck weights (10,000 random samples of 12

## STATISTICAL MODELLING OF YOUR DATA

ducks per sample) ...

```
sampleaverages<-rep(NA,10000)
n<-10000
for(i in 1:n) {
  set.seed(i)
  randomsample<-sample(x=populationweights, size=12, replace=FALSE)
  sampleaverages[i]<-mean(randomsample)
}
standarderror<-sd(sampleaverages)
... is 0.61
```

This is what we could consider to be the ‘true’ standard deviation of sample averages (for a particular sample size), or the ‘true’ standard error of the mean. The actual way to calculate this is:

$$\text{‘True’ standard error} = \text{standard deviation of the population} / \text{squareroot of sample size} = \\ 2.12 / \text{sqrt}(12) = 0.61$$

Now we’ll draw one random sample of 12 ducks weights from our population of 30 000 duck weights

```
set.seed(1)
sample _ duckweights<-sample(x=populationweights, size=12, replace=TRUE)
sample _ duckweights
[1] 16 15 9 12 12 9 14 12 11 16 12 15
mean(sample _ duckweights)
[1] 12.75
```

The sample standard deviation is

```
sd(sample _ duckweights)
[1] 2.454125
```

Thus, an estimate of the standard deviation of sample averages, the standard error of the mean, is equal

## STATISTICAL MODELLING OF YOUR DATA

to the sample standard deviation divided by the square root of 12.

```
sd(sample _ duckweights)/sqrt(12)
[1] 0.7084447
```

That's pretty close, 0.71 versus 0.61.

Recall that the standard deviation of a group of numbers is akin to the average amount by which the numbers deviate from the mean. The same is true of the standard error of the mean – the group of numbers is a collection of sample averages – and the standard error of the mean is akin to the average amount by which each sample average deviates from the average of the averages.

So now from one study, we can state that we estimate that ducks of this particular species weigh on average 12.8 pounds, and an estimate of the average deviation of sample means (if we had repeated the study many times) is 0.71 pounds higher or lower than 12.8 pounds. **Make sure you understand the difference between the sample standard deviation, which is 2.45, and the standard error of the mean, which we estimated from the sample standard deviation, as 0.71.**

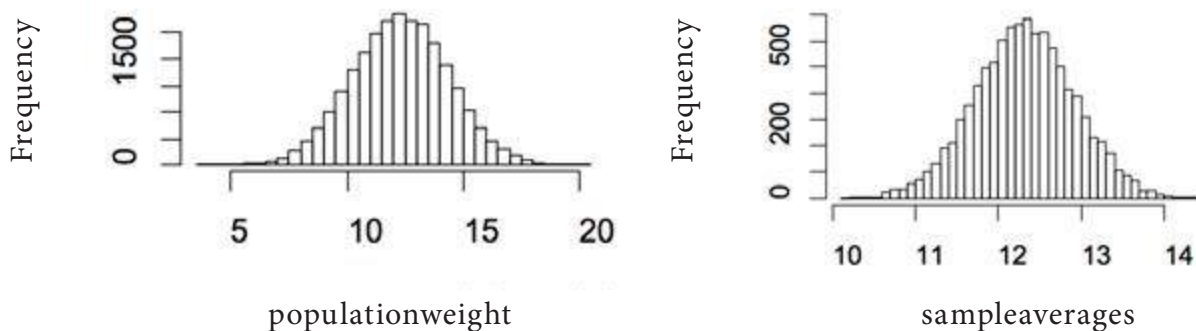
The usual way to write the estimated average and certainty in the estimate is  $12.8 \pm 0.71$  (SE) pounds. Because 0.71 could refer to the sample standard deviation, you need to state whether you're referring to either the sample standard deviation (SD) or the standard error of the mean (SE).

**All of that may have been a bit confusing on first read through, so you are invited to run through it all again. Keep in mind that in measuring statistical certainty, there are two sets of numbers you're dealing with. One set of numbers is your actual data, from which you calculate the average and which you use to estimate the population average. The other set of data is imaginary – it's a collection of sample averages as though you had repeated your study a zillion times. You want to know the amount of variability in that imaginary set of sample averages, so you need to know the standard deviation of these imaginary sample averages. But you didn't repeat the study a zillion times. Luckily statisticians, bless their hearts, have figured out that you can estimate the standard deviation of sample averages quite simply as the standard deviation of your real sample of real numbers divided by the square root of the number of numbers in your real data set (i.e. your sample**

## STATISTICAL MODELLING OF YOUR DATA

size), aka, the standard error of the mean.

*Sample Distribution Vs Sampling Distribution*



On the left is the distribution of the population of 30,000 duck weights, and on the right is the sampling distribution of the average weight of 12 ducks per randomly selected sample, and 10,000 of these samples of size  $n=12$  were drawn from the population of 30,000. It makes sense that the variability in the sample averages is less than in the population. The average is a measure of central tendency. So it's like sticking your hands into a barrel of apples (the histogram on the left), pulling handfuls of apples again and again, and each time choosing the most average looking apple (the histogram on the right).

**Before we move on let's confirm our original intuition that the more variable the population that we're sampling from, the more uncertain our estimate will be - that is, the larger the standard error of the mean.** This time we measured weights of every single duck (30000 in total) of a different duck species, with the same average weight as the first species, but weights in this second species are more variable. Note the larger standard deviation of this second duck species is 4.2 (versus 2.1).

```
set.seed(1234)
x<-rnorm(30000, mean=12.3, sd=4.2)
populationweights2<-round(x,0)
#Calculate the population average and standard deviation.
mean(populationweights2)
```

## STATISTICAL MODELLING OF YOUR DATA

12.3

```
sd(populationweights2)
```

4.20

As before we generate a sampling distribution of averages, as though we had conducted the study 10,000 times. Note the standard deviation of the sampling distribution for this species is larger (1.21 versus 0.61).

```
sampleaverages2<-rep(NA,10000)
n<-10000
for(i in 1:10000) {
  set.seed(i)
  randomsample<-sample(x=populationweights2, size=12, replace=TRUE)
  sampleaverages2[i]<-mean(randomsample)
}
sd(sampleaverages2)
```

1.21

And once again we randomly choose ONE sample of 12 ducks, weigh them, calculate the average weight and then estimate the standard error of the mean.

```
set.seed(2)
sample _ duckweights2<-sample(x=populationweights2, size=12, replace=TRUE)
sample _ duckweights2
[1] 13 12 22 15 14 11 13 11 8 15 16 16
mean(sample _ duckweights2)
```

13.8

```
sd(sample _ duckweights2)
```

3.49

**Standard error of the mean**

```
sd(sample _ duckweights2) / sqrt(12)
```

1.01



## STATISTICAL MODELLING OF YOUR DATA

The standard error of the mean for the more variable duck population is about 40% larger than that of the less variable duck population (1.01 versus 0.71). So we've just confirmed our intuitive sense that we will be less certain of our estimates when we're sampling from a variable population.

Let's also now confirm our intuitive sense that larger sample sizes will result in more certain estimates.

Instead of 12 ducks per sample, let's randomly select 144 ducks per sample. We'll sample from the original duck species.

```
set.seed(2)
sample_duckweights<-sample(x=populationweights, size=144, replace=TRUE)
sample_duckweights
[1] 13 12 17 14 13 12 13 12 10 14 14 14 13 14 14 12 15 13 10 16 10 12 11 11 13
[26] 11 13 14 14 10 11 9 14 11 17 16 11 12 11 13 12 14 14 14 8 13 17 16 16 14
[51] 9 13 7 14 16 11 12 13 12 13 11 10 14 14 12 13 13 14 13 13 15 13 13 16 12
[76] 15 9 13 15 14 14 11 13 8 12 12 10 12 11 16 12 9 12 11 11 10 16 12 13 14
[101] 15 13 17 11 12 11 11 8 12 11 13 13 11 14 7 13 11 9 17 14 14 16 13 13 11
[126] 12 12 13 10 10 14 14 10 8 12 15 5 13 14 17 11 12 15 10
sd(sample_duckweights)
2.25
Standard error of the mean
sd(sample_duckweights)/sqrt(144)
0.19
```

Increasing the sample size from 12 to 144 resulted in the standard error of the mean decreasing by a factor of 3.4 (0.71 to 0.21).

### PROBABILITY

Okay, now back to our histogram.

```
sampleaverages<-rep(NA,10000)
n<-10000
for(i in 1:10000) {
  set.seed(i)
  randomsample<-sample(x=populationweights, size=12, replace=TRUE)
  sampleaverages[i]<-mean(randomsample)
}
h<-hist(sampleaverages, breaks=12)
```

Calculating the standard error of the mean is one way to measure certainty in our estimation of the population mean. There is another way, referred to as **confidence intervals**. Let's see how these are calculated, and more importantly, what they actually mean.

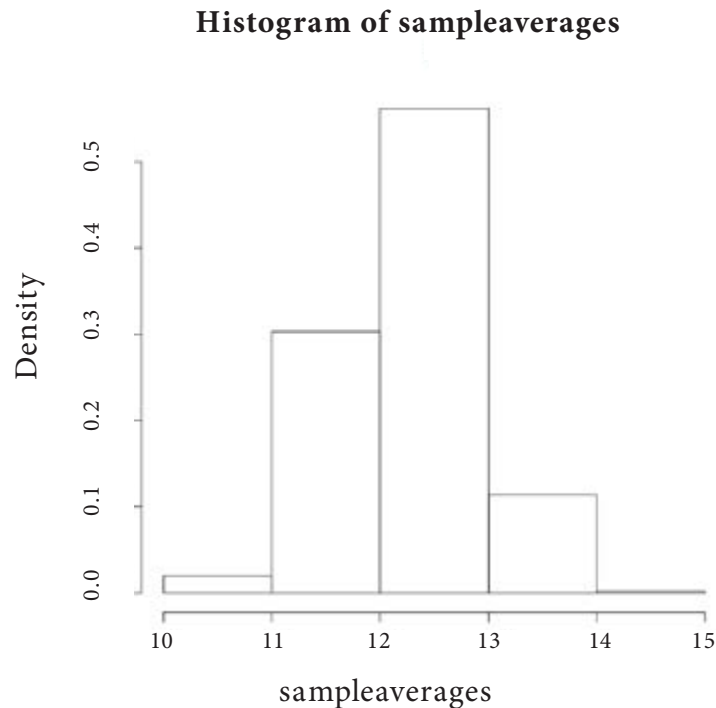
Instead of displaying counts of sample averages on the vertical axis of a histogram, we can convert the counts (which are called frequencies on the R histogram) to proportions (called densities on the R histogram), as the counts of observations within a certain interval out of the total number of observations. For example, in our sample of 10,000 averages of duck weights

```
h$breaks
[1] 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5
h$counts
[1] 16 186 808 2215 3218 2396 947 193 21
```

just 16 of the 10,000 averages are between 10 and 10.5 pounds. Thus the proportion (or density) of averages between 10 and 10.5 pounds is  $16/10,000 = 0.0016 = 0.16\%$ . Less than 1 percent of the 10,000 sample averages were between 10 and 10.5 pounds. Calculating the proportions for the other intervals and plotting the results looks like this ...

## STATISTICAL MODELLING OF YOUR DATA

```
h<-hist(sampleaverages, breaks=5, prob=TRUE)
```



... which shows us that about 30% of the averages were between 11 and 12 pounds. Note that the `prob` argument has been added to the `hist` function, which tells R to convert counts to proportions (densities). Note also that the intervals have been changed from half to full intervals (e.g. from 10 to 10.5 to 10 to 11), simply by lowering the number of intervals (`breaks`).

We can check the exact proportions by taking the output from the histogram function for the intervals (`breaks`), counts, and proportions (density) and putting them all in one table.

```
intervals<-h$breaks[1:5]  
counts<-h$count  
proportions<-h$density  
data.frame(intervals, counts, proportions)
```

## STATISTICAL MODELLING OF YOUR DATA

Table 8. Intervals, frequencies and proportions for a sample of 10,000 average duck weights.

Intervals	Frequencies	Proportions
10	202	0.0202
11	3023	0.3023
12	5614	0.5614
13	1140	0.1140
14	21	0.0021
Sum	10,000	1

Now we can see that exactly 30.23% (3023 of 10,000) of the averages of ducks weights were between 11 and 12 pounds in our sample of 10,000 averages.

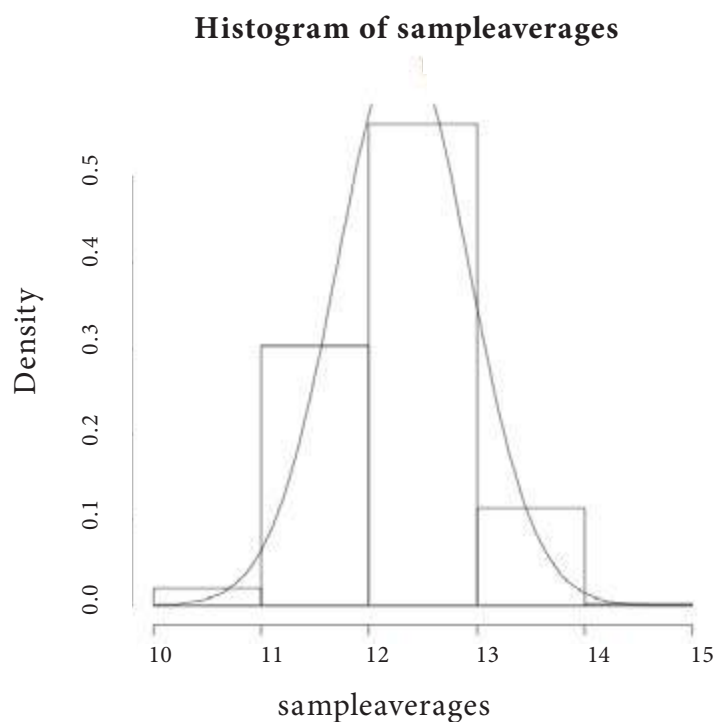
Let's add the curve to the histogram.

```
x<-seq(9, 16 , 0.01)
```

```
curve(dnorm(x,mean=mean(sampleaverages),sd=sd(sampleaverages)),add=TRUE)
```

and note that the proportions sum to 1

```
sum(h$density)
```



## STATISTICAL MODELLING OF YOUR DATA

Consider that the proportion of 10,000 samples of average duck weights that are between 11 and 12 pounds is 30%. Given that no matter how many samples of averages we take from this population, we will always end up with a sampling distribution with this normal shape. The shape itself depends only on the variability in the data – with high variability (and thus high standard deviation), the shape will look fat, and with low variability it will look skinny. But as long as sampling is from the same population, the shape of the sampling distribution will be the same. With low sample sizes, the distribution will look more ‘bumpy’, and with high samples sizes it will begin to resemble a smooth curve. Given that the shape of the sampling distribution from the same population does not change, then if we were to take an additional say 100 samples of averages, chances are that about 30 of the sample averages would be between 11 and 12 pounds.

Let’s try:

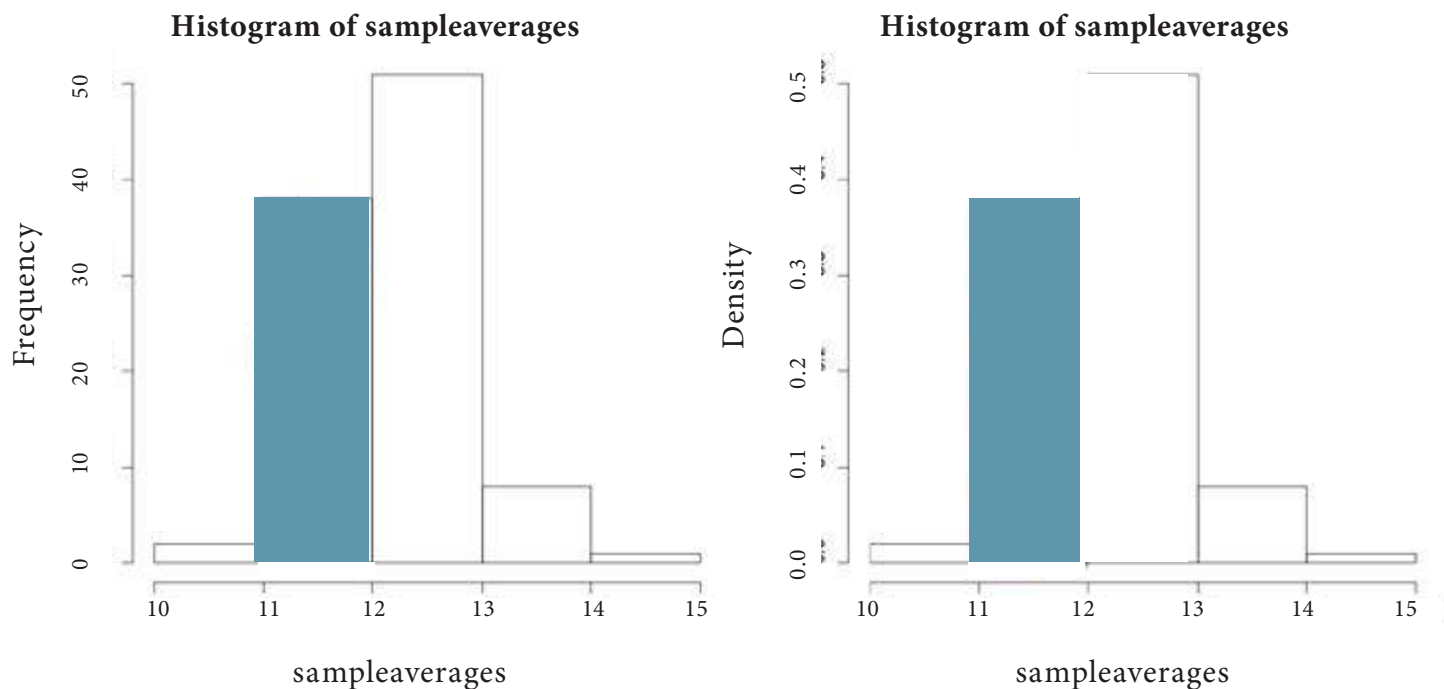
```
sampleaverages<-rep(NA,100)
n<-100
for(i in 1:100) {
  set.seed(i)
  randomsample<-sample(x=populationweights, size=12, replace=TRUE)
  sampleaverages[i]<-mean(randomsample)
}
h<-hist(sampleaverages,breaks=4)
h
intervals<-h$breaks[1:5]
counts<-h$counts
proportions<-h$density
data.frame(intervals, counts, proportions)
```

Sure enough – 38 sample averages are between 11 and 12 pounds. Note that we used the words chances are. By chances we are referring to probability. Probability is the most fundamental concept to statistics. **Probability refers to the likelihood of getting a certain study result, which we can estimate based on a sampling distribution.**

## STATISTICAL MODELLING OF YOUR DATA

About 30% of our samples of average duck weights are between 11 and 12 pounds. We got nearly the same result from a sample size of 100 averages, and from 10,000 averages.

Let's have another look at our histogram for our sample of 100 averages. The coloured in bar on the left is the count of averages between 11 and 12 pounds. We've seen how we can convert these counts to proportions (aka density) – that's the histogram on the right.



Based on the sampling distribution of 100 sample averages, we can now say that if we were to repeat our duck weighing study yet another 100 times, average duck weights would be between 11 and 12 pounds in about 38 of the 100 studies. In other words, **the chances of getting average duck weights of between 11 and 12 pounds is about 38%.**

But we're basing this statement of certainty on a pretty small sample size of 100. It would be much better to base it on an infinite sample size – that is, from the normal curve. Using the normal curve, we can make statements of **probability** from the expected distribution of data across the observed range.

**Given any normally distributed data, we can figure out the probability associated with any values**

## STATISTICAL MODELLING OF YOUR DATA

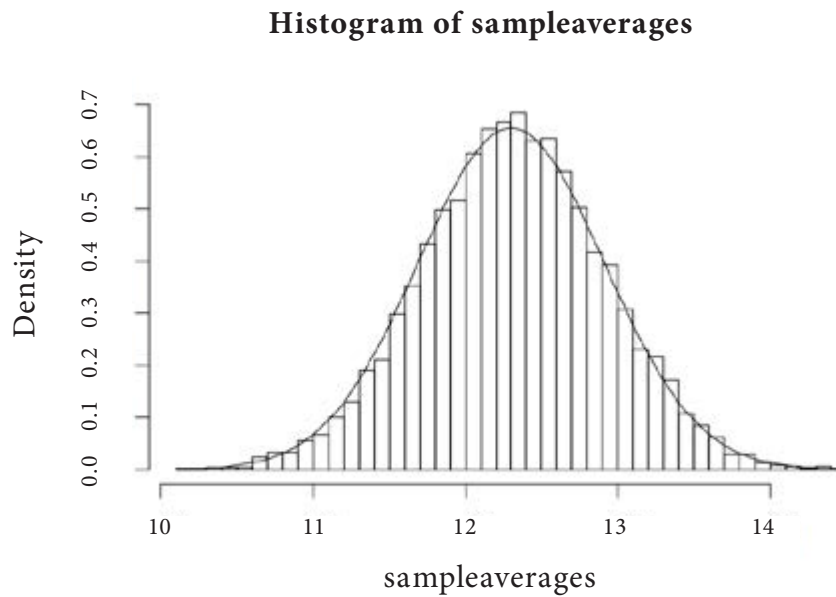
within a certain interval within the range of observed data.

We can use the `rnorm` function in R to create a data set from random samples drawn from a normal distribution. You simply need to input sample size, and the mean and standard deviation. In fact, that's how we created our original data set of 30,000 duck weights. Note the `set.seed` function – this ensures that you get the same random draw each time. If you change any of the numbers in the `set.seed` function you will get a different random draw.

It's a very nifty fact that many continuous variables in nature are normally distributed, especially the sizes of living things – measurements like height, weight, and length of animals and plants are usually normally distributed (or the logarithm of the measures are normally distributed).

Let's consider our sample of 10,000 averages again. The curve on the histogram is **THE normal distribution** curve for a population with mean 12.3 and standard deviation 1.6. The histogram shows the sampling distribution of our 10,000 average weights, with the proportions shown instead of the frequencies. Our sample data is pretty close to normal, but it is not exactly normal. So we can't use the histogram of our sample data to say for example that we should expect to measure about 3000 ducks between 11 and 12 pounds in our sample of 10,000 ducks. Instead we assume that if we had sampled enough, our data would become THE normal distribution, and thus to derive an exact probability of weighing ducks between 11 and 12 pounds, we instead use the proportions from the normal curve.

By **THE normal curve**, we mean the **standard normal curve**, which is centred on 0 and has a **standard deviation of 1**. Statisticians have worked out exact probabilities for all possible values along the x axis. The only thing you need to do is take your normally distributed data, and standardize it to the standard normal curve. Standardizing your data to the standard normal curve is like lifting up your sampling distribution curve and sticking it on top of the standard normal curve, such that every value in your data has an equivalent value on the standard normal curve. Basically, you're just re-scaling your data to a standard. That way, you can figure out probabilities for your data using the standard, for which probabilities have been calculated already.



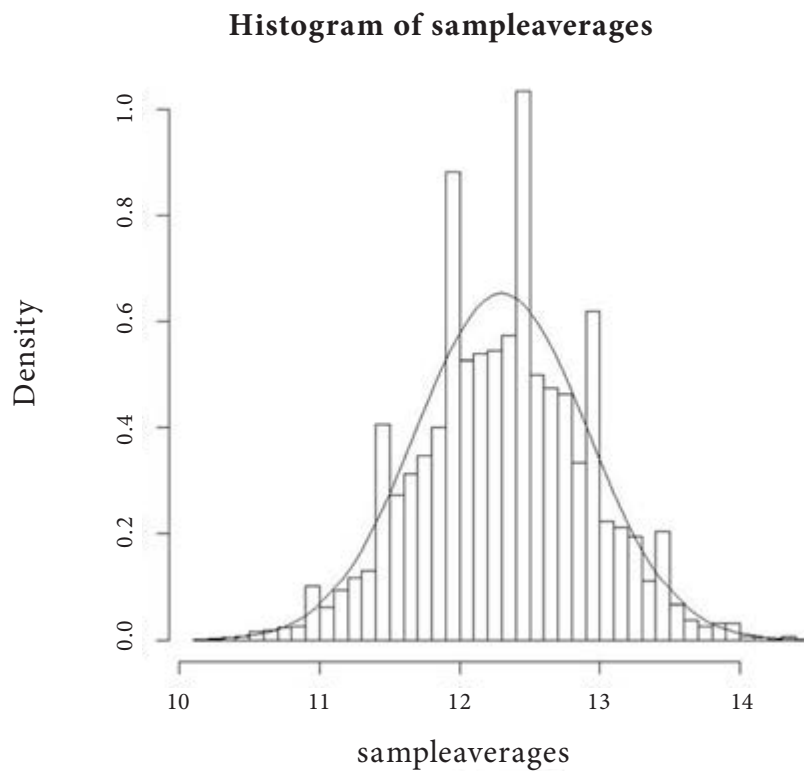
# for the right sample number

```
set.seed(1234)
x<-rnorm(30000, mean=12.3, sd=2.1)
populationweights<-round(x,0)
sampleaverages<-rep(NA,10000)
n<-10000
for(i in 1:n) {
  set.seed(i)
  randomsample<-sample(x=populationweights, size=12, replace=TRUE)
  sampleaverages[i]<-mean(randomsample)
}
```

# to create the histogram

```
h<-hist(sampleaverages, breaks=40, main="", prob=TRUE)
x<-seq(9,16,0.01)
curve(dnorm(x,mean=12.3,sd=0.61),add=TRUE)
```





```
intervals<-h$breaks[1:44]
counts<-h$counts
proportions<-h$density
table2<-data.frame(intervals, counts, proportions)
#output table to excel
#intall package xtable
library(xtable)
table2<-xtable(table2)
print(table2, type="html", file="table2.html")
```

#open the file in your R working folder, press control and a to select everything, then press control and c to copy, then open excel and paste in the table.

## STATISTICAL MODELLING OF YOUR DATA

Table 9. Intervals, frequencies and proportions for a sample of 10,000 average duck weights from data that was standardized to a normal distribution curve.

	Intervals	Counts	Proportions
1	10.10	2	0.00
2	10.20	2	0.00
3	10.30	5	0.00
4	10.40	7	0.01
5	10.50	16	0.02
6	10.60	19	0.02
7	10.70	24	0.02
8	10.80	26	0.03
9	10.90	101	0.10
10	11.00	63	0.06
11	11.10	93	0.09
12	11.20	116	0.12
13	11.30	129	0.13
14	11.40	407	0.41
15	11.50	273	0.27
16	11.60	312	0.31
17	11.70	347	0.35
18	11.80	400	0.40
19	11.90	883	0.88
20	12.00	528	0.53
21	12.10	539	0.54
22	12.20	544	0.54
23	12.30	573	0.57
24	12.40	1034	1.03
25	12.50	501	0.50
26	12.60	476	0.48
27	12.70	465	0.47
28	12.80	334	0.33
29	12.90	620	0.62
30	13.00	224	0.22
31	13.10	212	0.21
32	13.20	196	0.20
33	13.30	111	0.11
34	13.40	204	0.20
35	13.50	66	0.07
36	13.60	38	0.04
37	13.70	27	0.03
38	13.80	31	0.03
39	13.90	31	0.03
40	14.00	7	0.01
41	14.10	4	0.00
42	14.20	2	0.00
43	14.30	7	0.01
44	14.40	1	0.00

### Standardizing to the Standard Normal Curve

Standardizing our 10,000 sample averages to the standard normal curve (mean=0, sd=1) is pretty simple.

Each deviate is simply scaled in units of standard deviation, which is akin to the average deviation. What does that mean? Remember that a deviate is the amount of difference between an observation and the average. To standardize to the standard normal distribution, we calculate the deviation of each sample average from the true population average, and scale these by the true standard deviation. Our average of sample averages happens to be exactly the same as the population average, but we'll be technically correct here.

Recall the true average of our population of 30,000 duck weights is:

```
mean(populationweights)
```

```
12.3
```

And the true standard deviation of the averages of all possible samples of size 12 drawn from this population (aka the true standard error of the mean) is:

```
sd(populationweights)/sqrt(12)
```

```
0.61
```

Therefore to standardize our 10,000 averages to the standard normal distribution, for each average we need to calculate

$Z = \text{sampleaverage} - \text{population mean} / \text{true standard error of the mean}$

```
Z<-(sampleaverages - 12.3)/0.61
```

These are referred to as Z scores (for some unknown reason).

Let's see what the distribution of our now standardized data looks like.

## STATISTICAL MODELLING OF YOUR DATA

Don't forget the brackets!! Otherwise R will do the division first, and then the subtraction. If you got an error, it may have been because you input a long dash – instead of a short dash - for subtraction. Yes, R is THAT picky.

```
mean(Z)
```

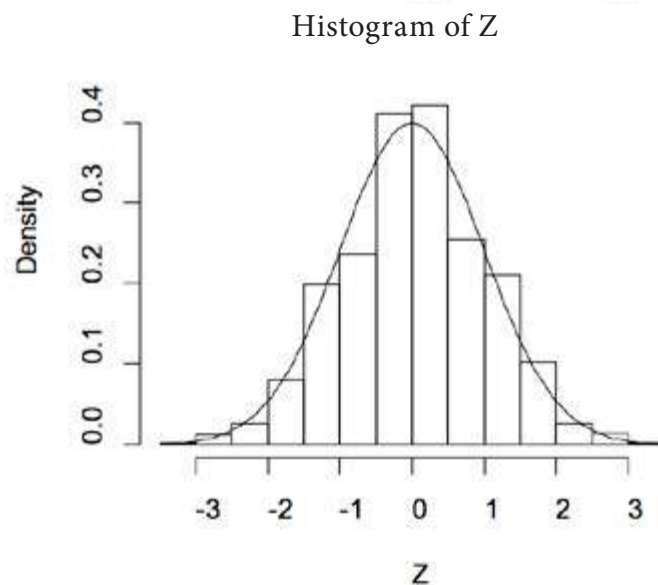
```
sd(Z)
```

Because our sample of averages was a bit off normal, so too is our standardized data. But, it's pretty good (mean = 0.0240 and sd = 1.0004)

Let's add the standard normal curve.

```
x<-seq(-4,4,0.01)
```

```
curve(dnorm(x,mean=0,sd=1),add=TRUE)
```



In order to calculate probabilities now, we just need to match up our standardized data with the probabilities that have been calculated for the standard normal curve.

Well, actually, we don't. That's the way things used to be. Now we can actually just use R to calculate probabilities for any normally distributed data. This time it's like doing the opposite – we're shifting the

## STATISTICAL MODELLING OF YOUR DATA

standard normal curve on top of our data. And then as before, matching up our data values within those on the standard normal curve. But keep somewhere in the back of your mind that we can standardize any data set to a standard distribution to calculate probabilities, because that will be very important in the next section.

We're going to see the functions `dnorm` and `pnorm` in action. The `dnorm` function gives us the same information as the proportions output from the `histogram` function, but is instead the proportions from the standard normal curve.

Let's compare the proportions in our sample data to the proportions (densities) expected from the normal curve. Note that the function needs to know where the normal curve will be centred (the mean) and its shape (how fat or skinny it is), which is defined by the standard deviation, and then it needs to know the value for which you want to find the proportion. Let's arbitrarily choose 13.4

```
dnorm(13.4, mean=12.3, sd=0.61)
```

```
0.13
```

Compare the proportion under the normal curve with the proportion of our sample data in the table. Pretty close. Try another value

```
dnorm(12.9, mean=12.3, sd=0.61)
```

```
0.41
```

Again, quite close. Read off the figure to see that 12.9 is indeed associated with a density of 0.41 on the normal curve.

We know that all of the proportions (densities) under the normal curve sum to 1, since the curve is simply showing us the distribution of counts of values across the set of values. For the normal curve, the set of values is infinite.

Now we are ready to make the jump from the frequency of observations to probability.

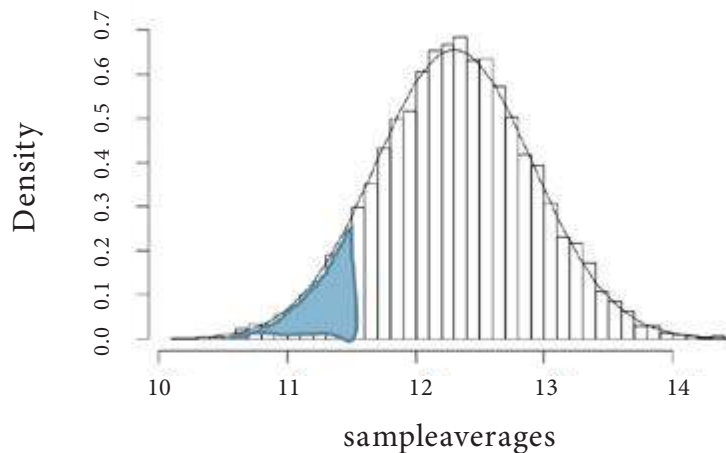
Let's ponder human psychology a bit first. We tend to think that things that have happened often will

## STATISTICAL MODELLING OF YOUR DATA

happen again. Thus, it feels intuitively right to say, hey, I just grabbed a handful of nuts off this tree here and squirrels had chewed 80% of them. On this next tree, I expect that about 80% of those nuts will be chewed too.

That is pretty much statistics in a nutshell.

Given that the area under the normal curve is equal to 1, we can figure out the probability of sampling certain values based on the proportion of the total area associated with that value. For example, from eye-balling the curve, the proportion of the area under the normal curve associated with 11.5 pounds is about ... say 12% ish of the total area?



No need to guess. `pnorm(11.5,12.3,0.61)` Ah. It's 9.48%.

How about 12.8?

```
pnorm(12.8,12.3,0.61)
```

79.4%

Now we can make a probability statement. Given a sample of average duck weights of mean 12.3 and standard deviation 0.61, the probability of weighing a duck of 11.5 pounds or less is 9.5% and the probability of weighing a duck of 12.8 pounds or less is 79%.

What about a duck of weight greater than 13.2?

## STATISTICAL MODELLING OF YOUR DATA

```
1-pnorm(13.2,12.3,0.61)
```

7.0%

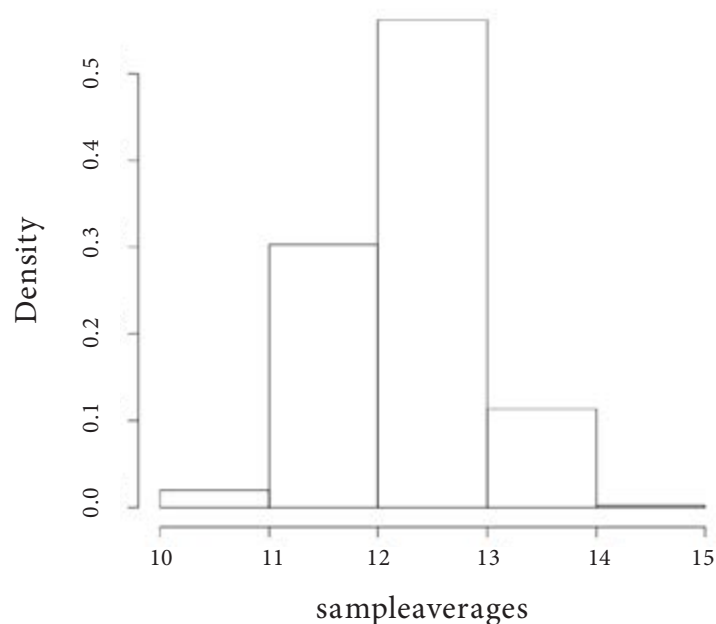
What about a duck between 11 and 12 pounds?

```
pnorm(12,12.3,0.61)-pnorm(11,12.3,0.61)
```

29.4%

Aha. That's familiar. Referring back to our histogram of 10,000 duck weights, and note the density associated with weights between 11 and 12 pounds – 30%.

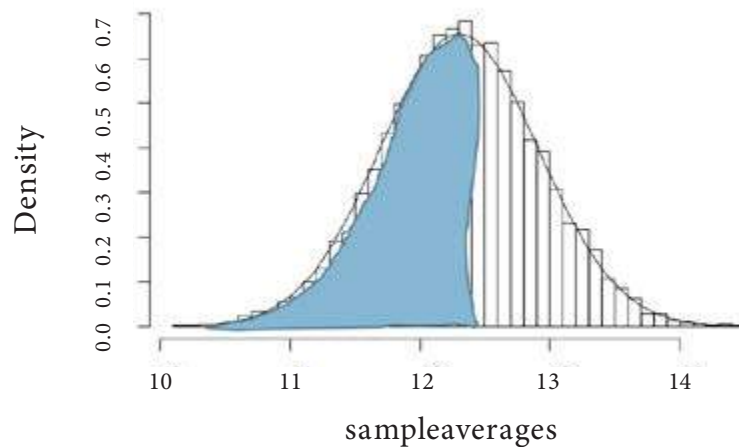
**Histogram of sampleaverages**



We've just seen how we can calculate the probability associated with a particular value or range of values given normally distributed data. What this actually means is, given a normally distributed sample from a population, we can calculate the number of times we would expect to get that value, or a less value, if we had repeated our study many times. Remember that the normal distribution is continuous, which means there is no exact probability of getting a particular value. We can only state the probability of getting a particular value or a value that is less than or greater than that value.

## STATISTICAL MODELLING OF YOUR DATA

Back to our sampling distribution of sample averages. Aside from the standard error of the mean, what other way could we measure certainty around our estimate of the population average of duck weights?



We can use the `qnorm` function to figure out the value on the x axis associated with a given probability. Values associated with probabilities are referred to as **quantiles** (hence the q in `qnorm`).

```
qnorm(0.60,12.3,0.61)
```

```
12.4
```

Thus, 60% of the sample averages are less than 12.4. In other words, the probability of weighing a 12.4 pound duck is 60% (if you repeated the study again and again, 60% of sample averages would be 12.4 pounds or less).

And once more ...

```
qnorm(0.40,12.3,0.61)
```

```
12.1
```

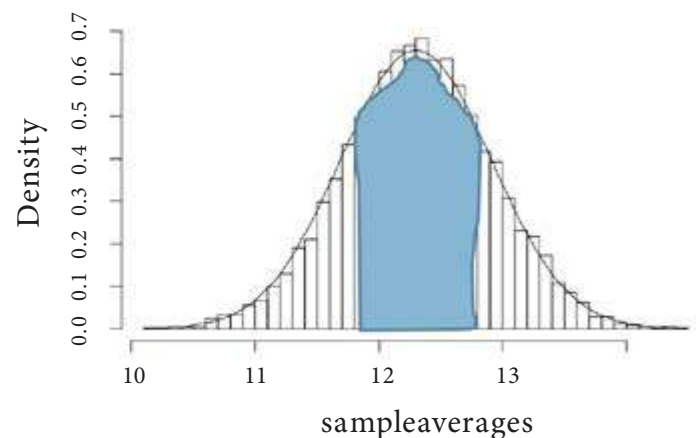
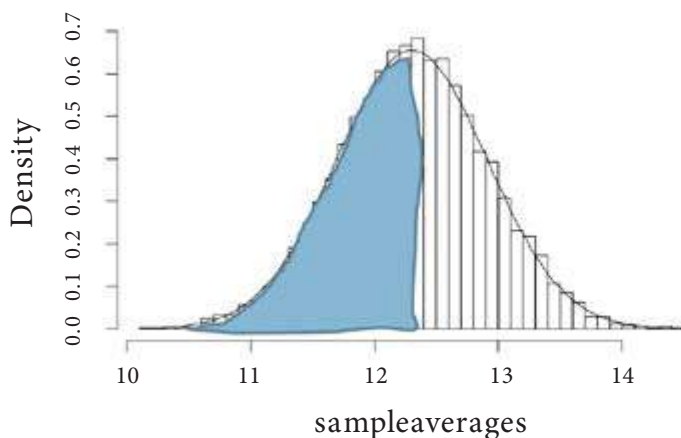
40% of the sample averages are less than 12.1

You can see that the `qnorm` function calculates the value associated with the probability *to the left* of the value.



## STATISTICAL MODELLING OF YOUR DATA

What we're really interested here is the amount of sample averages clustered around the average of the sample averages. Let's say we want to know the range of average duck weights associated with 50% of the sample averages, 25% above the average and 25% below the average? If we were to input 0.50 into the probability argument of the `qnorm` function, it would just return 12.3, the average. We want to take that 50% and shift it onto the average. That means that 25% remains below and 25% is above. For the upper value, that's the same thing as inputting into the probability argument  $0.50+0.25=0.75$



```
qnorm(0.75,12.3,0.61)
```

```
12.711
```

And the lower is 25% so 0.25

```
qnorm(0.25,12.3,0.61)
```

```
11.889
```

Therefore, 50% of sample averages are between 11.889 and 12.711.

Recall that the shape of the normal curve is defined by the standard deviation – the average deviation of values from the mean. Thus for any normal curve, the range of values that represents different proportions of values can be defined as a factor of the standard deviation.

## STATISTICAL MODELLING OF YOUR DATA

For example, the upper range value defining 50% of the sample averages is 12.711, which is 0.411 pounds above the sample average of 12.3. Likewise, the lower range value of 11.899 is also 0.411 pounds below the sample average of 12.3 (the normal distribution is symmetrical after all).

The standard deviation of our sample averages is 0.61. Thus, 50% of the sample averages are  $0.411/0.61 = 0.674$  standard deviations either above or below the sample average. In fact, for any normal curve, 50% probability is found within 0.674 standard deviations of the average.

What about 80% probability? That would mean 20% probability on either end, so the lower range value is half of that

```
qnorm(0.10,12.3,0.61)  
11.518
```

and the upper range is 80% + 10%

```
13.082
```

Thus, if we repeated the study many times, in 80% of studies average duck weight would be between 11.518 and 13.082.

Defined in terms of standard deviations

```
13.082-12.3 = 0.782  
0.782 / 0.61 = 1.28 standard deviations.
```

Thus, 80% of average duck weights would be 1.28 standard deviations above the mean and 1.28 standard deviations below the mean.

And now for 95% probability, which you will learn is a standard used to define statistical certainty.

95% probability means that 5% probability is 'leftover', 2.5% at the lower end and 2.5% at the upper.

## STATISTICAL MODELLING OF YOUR DATA

So the lower range value is

```
qnorm(0.025, 12.3, 0.61)
11.104
```

The upper range values is 95% + 2.5%

```
qnorm(0.975, 12.3, 0.61)
13.496
(13.496-12.3)/0.61 #or (11.104-12.3)/0.61
1.96
```

**For any normal curve, 95% of the observations are found within 1.96 standard deviations of the mean.**

What does this mean in terms of certainty around our estimate of average duck weights?

For our sample of 10,000 duck weight averages, it means that 95% of duck weight averages are found within 1.96 standard deviations of the average of the averages.

$$12.3 + 1.96 \times 0.61 = 13.496$$

$$12.3 - 1.96 \times 0.61 = 11.104$$

If we conducted 10,000 duck weight studies, in 9500 studies, the average of the 12 ducks per sample would be between 13.5 and 11.1.

#And now we are ready to calculate the **95% confidence interval of the average duck weight.**

```
set.seed(27)
sample_duckweights<-sample(x=populationweights, size=12, replace=TRUE)
sample_duckweights
mean(sample_duckweights)
```

## STATISTICAL MODELLING OF YOUR DATA

12.9

```
upperinterval<-mean(sample _ duckweights) + 1.96 * sd(sampleaverages)
lowerinterval<-mean(sample _ duckweights) - 1.96 * sd(sampleaverages)
upperinterval
```

14.1

```
lowerinterval
```

11.7

**This allows to make the statement that 11.7 to 14.1 is one interval of many intervals that contains the true population average 95% of the time.**

A standard practice for reporting confidence intervals is to write them as follows:

sample mean  $\pm$  1.96\*SE

e.g. 12.9  $\pm$  1.20 (95% CI)

Note that the probability contained within the interval is stated in brackets.

**So now you have learned two ways to measure certainty around estimates – the standard error of the mean, and confidence intervals around the mean.**

By the way, the branch of statistics that we have been applying is referred to as **frequentist statistics**, because it is based on probability being measured with reference to the relative frequency of observations, in other words, from sampling distributions.

# HYPOTHESIS TESTING AND P VALUES

We've figured out how to estimate a population parameter, the average, and how to state how certain we are of our estimate.

Next up: using data to answer research questions. In other words, hypothesis testing.

## The T Distribution

In order to more fully explore hypothesis testing, we first need to understand an important sampling distribution called the t distribution.

In the previous section, we calculated the Z score, which standardized our 10,000 sample averages to the standard normal curve. But note that in order to calculate the Z score, we needed to know the true standard error of the mean, which is the same as the standard deviation of the 10,000 sample averages.

$$Z = \text{sample average} - \text{population mean} / \text{true standard error of the mean}$$

But we almost never know the true standard error of the mean. Now what?

Remember that we can estimate the 'true' standard error of the mean from the sample, simply as sample standard deviation / square root of sample size. So now instead of the 'actual' standard deviation of the sample averages, which is 0.61, we estimate it from one sample. We've done that a few times now but let's do it again. We'll use a different number in the `set.seed` function to ensure we get a completely new random sample.

```
set.seed(390)
sample_duckweights<-sample(x=populationweights, size=12, replace=TRUE)
sample_duckweights
mean(sample_duckweights)
13.1
```

## STATISTICAL MODELLING OF YOUR DATA

```
sd(sample _ duckweights)
```

```
1.56
```

```
estimatedstandarderror<-sd(sample _ duckweights)/sqrt(12)
```

```
0.45
```

When we replace the ‘true’ standard error of the mean with the estimated standard error of the mean, the z statistic becomes the t statistic. Just like the z statistic, the t statistic is a deviate expressed in units of the standard deviation, only this time we don’t know the standard deviation, we have to estimate it from our data.

$t = \text{deviate} / \text{average of deviates}$

$= (\text{sample average} - \text{population average}) / \text{estimated standard error of the mean}$

The t statistic has a sampling distribution, just like any statistic does – as we have seen the sampling distribution of the average is the normal distribution. The sampling distribution for the t statistic is called (drum roll) .... the t distribution.

The t statistic does exactly the same thing as the z statistic – that is, it standardizes any normal curve of data to a standard curve, from which probabilities can be calculated. Only, it standardizes the curve to the t distribution.

To see what this looks like, we have to also estimate the standard error of the mean from each of our 10,000 random samples.

```
estimatedSEmean<-rep(NA,10000)
```

```
n<-10000
```

```
for(i in 1:10000) {
```

```
  set.seed(i)
```

```
  randomsample<-sample(x=populationweights, size=12, replace=TRUE)
```

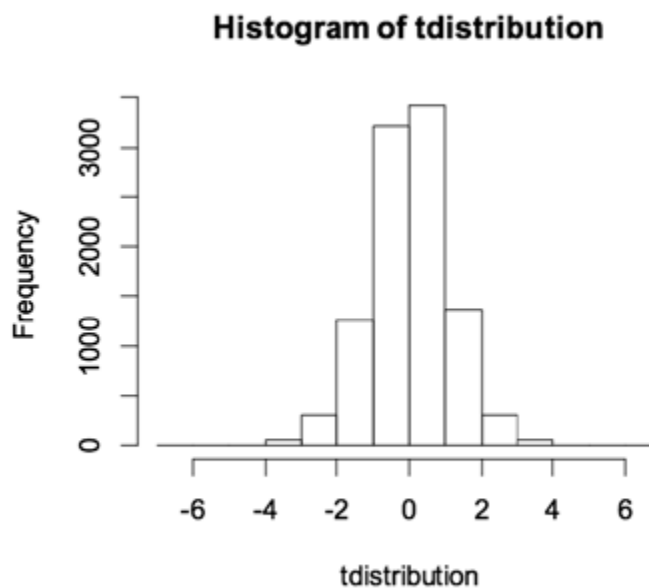
```
  estimatedSEmean[i]<-sd(randomsample)/sqrt(12)
```

```
}
```

## STATISTICAL MODELLING OF YOUR DATA

This is the t distribution for our 10,000 sample averages

```
tdistribution<-(sampleaverages - 12.3)/estimatedSEmean  
h<-hist(tdistribution)
```



It *looks* like the normal distribution but the t distribution is different.

The t statistic is derived from the standard error of the mean, which itself is derived from the sample standard deviation, which is dependent on the sample size. The t statistic will be different for different sample sizes, and thus the shape of the t distribution will be different for different sample sizes. The t distribution gets skinnier with increasing sample size.

Because of the dependency of the shape on sample size, there are an infinite number of t distributions. In order to find the probability associated with values on the x axis of the t distribution, you have to find the right t distribution for your sample size.

The only kink is that the right t distribution is the one associated with your sample size *minus the number of population parameters you're trying to estimate, in other words, with the correct **degrees of***

## STATISTICAL MODELLING OF YOUR DATA

*freedom.* In our case of average duck weights, it's  $12-1 = 11$ . The degrees of freedom can be understood most simply as the sample size minus the number of parameters being estimated. When we conduct a t test for the population mean, we need to estimate the population mean from the sample, and thus one observation is tied to the sample mean, and is not 'free'. So the degrees of freedom for a t test about the population mean is  $n - 1$ .

The t distribution is useful for two things in particular – hypothesis testing and deriving confidence intervals for estimates of the mean of a population.

What the heck says you? We already derived confidence intervals for the mean using the normal distribution. Welllll, remember, we never actually know the true standard error of the mean, we almost always have to estimate it. That means we need to standardize our data to the t distribution in order to get probabilities. BUT, the t distribution approaches the normal distribution when sample sizes are above about 40. So really, it's only small sample sizes for which you need to use the t distribution to calculate confidence intervals on the sample mean. Otherwise, the normal distribution is fine, because the t distribution is just like the normal anyway when sample size is greater than about 40.

For sample sizes below about 30, the 95% confidence interval is no longer  $\text{mean} \pm 1.96 \times \text{standard error}$  of the mean. Just like `qnorm`, R has a function for the quantile associated with 95% probability. But remember that the t distribution depends on sample size, so there is a second argument for the sample size. The mean of the t distribution, since it is a standard distribution, is always 0.

#with large samples sizes the t distribution is the normal distribution, for example

```
qt(0.975,500)
```

```
1.96
```

#for small sample sizes, the quantile is different from the normal curve

```
qt(0.975,30)
```

```
2.04
```

Therefore, for a sample size of 30, the confidence interval would be calculated as  $\text{mean} \pm 2.04 \times$



estimated standard error of the mean.

For a sample size of 15, the confidence interval would be

```
qt(0.975,15)
```

```
2.13
```

mean  $\pm$  2.13 x estimated standard error of the mean

Now back to hypothesis testing, the other useful application of the t distribution.

## One and Two Sample Hypothesis Testing

Now we can talk about hypothesis testing. Statistical hypothesis testing arises from a research question. For example, our research question might be something like, are surf scoters larger than white-winged scoters?

To answer this research question, we conduct statistical hypothesis testing, which in brief proceeds like this:

1. State a null hypothesis. Null refers to 0 or none. In this case, our null hypothesis is that the average wing length of surf scoters and white-winged scoters are equal, there is no difference.
2. State the alternative hypothesis. Given our research question, the alternative hypothesis is that surf scoters are larger than white-winged scoters. For other questions, the alternative might simply be that wing length does differ, or that surf scoters are smaller than white-winged scoters.
3. Choose a test statistic such that the sampling distribution of the test statistic defines the null expectation of no difference.
4. Calculate the test statistic for your actual data, and determine the probability associated with the test

## STATISTICAL MODELLING OF YOUR DATA

statistic of your data on the null sampling distribution.

5. If your test statistic is associated with a low probability on a sampling distribution that assumes no difference, then conclude that the null sampling distribution is wrong. Therefore, reject your null hypothesis of no difference, and accept your alternative hypothesis. Conclude there is a difference between the populations. For example, we would conclude that white-winged scoters are indeed larger than surf scoters.

6. If your test statistics is associated with a high probability on a sampling distribution that assumes no difference, then conclude that you have no evidence to state that the null sampling distribution is wrong. Therefore, conclude that you have no evidence suggesting that there is any difference in size between surf scoters and white-winged scoters.

Think that's convoluted thinking? You're not alone - there have been many arguments against it. Two issues are particularly important to keep in mind:

1. What should a low probability on the null distribution be in order to decide that the null hypothesis of no difference is incorrect?
2. It is often the magnitude of differences between things that we care about. We could conduct a very elaborate and expensive study and conclude that yes, this is different from that. But what a waste of time and money if we had not also stated by how much this is different from that, and how certain we are of our estimation of difference.

A more general way to calculate the t statistic is:

$$t = \text{sample statistic} - \text{true population value} / \text{estimated standard error of the statistic}$$

In other words, the t distribution is relevant for other types of statistics, as we'll see in the next section.

Instead of the true population value, we could enter into the calculation a hypothetical population value. For example, suppose we think (we hypothesize) that surf scoters and white winged scoters have the

## STATISTICAL MODELLING OF YOUR DATA

same wing length. We happen to have a fairly good estimate of white winged scoter wing length of 26.0  $\pm$  2.27 (SE) cm.

So now we have a null hypothesis regarding surf scoters, which is symbolized by  $H_0$ .

$$H_0 = 26$$

The alternative hypothesis that we'll test is that surf scoters are smaller than white winged scoters and thus have smaller average wing lengths.

$$H_a < 26$$

Now we ask hunters to send us wings of surf scoters that they've harvested. We end up with a sample of wings from 67 surf scoters. We measure each one, calculate the average of 20.0 cm and the standard error of the mean, of 0.29 cm.

First let's generate these data

```
set.seed(407)
surfwings<-rnorm(67,20.3,2.11)
meansurfwings<-mean(surfwings)
meansurfwings
SEmeansurfwings<-sd(surfwings)/sqrt(67)
SEmeansurfwings
```

In order for us to figure out probabilities associated with these data, they have to be standardized to the t distribution with degrees of freedom = n - 1, which is 66. Recall that the shape of this t distribution will be quite similar to the normal distribution.

To test our null hypothesis, we need only match our sample data to the t distribution with 66 degrees of freedom and figure out the probability associated with our sample average. Remember that this t distribution describes our null expectation, which in this case is that average surf scoter wing length is



## STATISTICAL MODELLING OF YOUR DATA

Therefore, we reject the null hypothesis, and accept the alternative hypothesis. We conclude that average surf scoter wing length is less than 26 cm, and therefore surf scoters are smaller than white winged scoters.

What we've just done is called a one-sample t test. And more specifically, a one-tailed one-sample t test. One sample because we are testing hypotheses about just one population, surf scoters. Even though our research question is actually about two species, our **statistical hypothesis** is only about one species.

This was a one-tailed t test because our alternative hypothesis was concerned only with one tail of the t distribution, the left side. That's because our alternative hypothesis was that surf scoters were smaller than white-winged scoters.

We can use the R `t.test` function to do exactly the same thing.

```
t.test(surfwings, alternative = "less", mu = 26, conf.level = 0.95)
```

R output:

```
One Sample t-test
```

```
data: surfwings
```

```
t = -20.865, df = 66, p-value < 2.2e-16
```

```
alternative hypothesis: true mean is less than 26
```

```
95 percent confidence interval:
```

```
 -Inf 20.48395
```

```
sample estimates:
```

```
mean of x
```

```
20.00457
```

Let's try another example and this time we'll visualize it as we work through it.

$H_0 = 20.4$

## STATISTICAL MODELLING OF YOUR DATA

$H_a < 20.4$

$t = 20.0 - 20.4 / 0.29 = -1.3761$

```
pt(-1.3761, 66)
```

```
=0.0867
```

OR

```
t.test(surfwings, alternative = "less", mu = 20.4)
```

Output from R:

```
One Sample t-test
```

```
data: surfwings
```

```
t = -1.3761, df = 66, p-value = 0.08672
```

```
alternative hypothesis: true mean is less than 20.4
```

```
95 percent confidence interval:
```

```
-Inf 20.48395
```

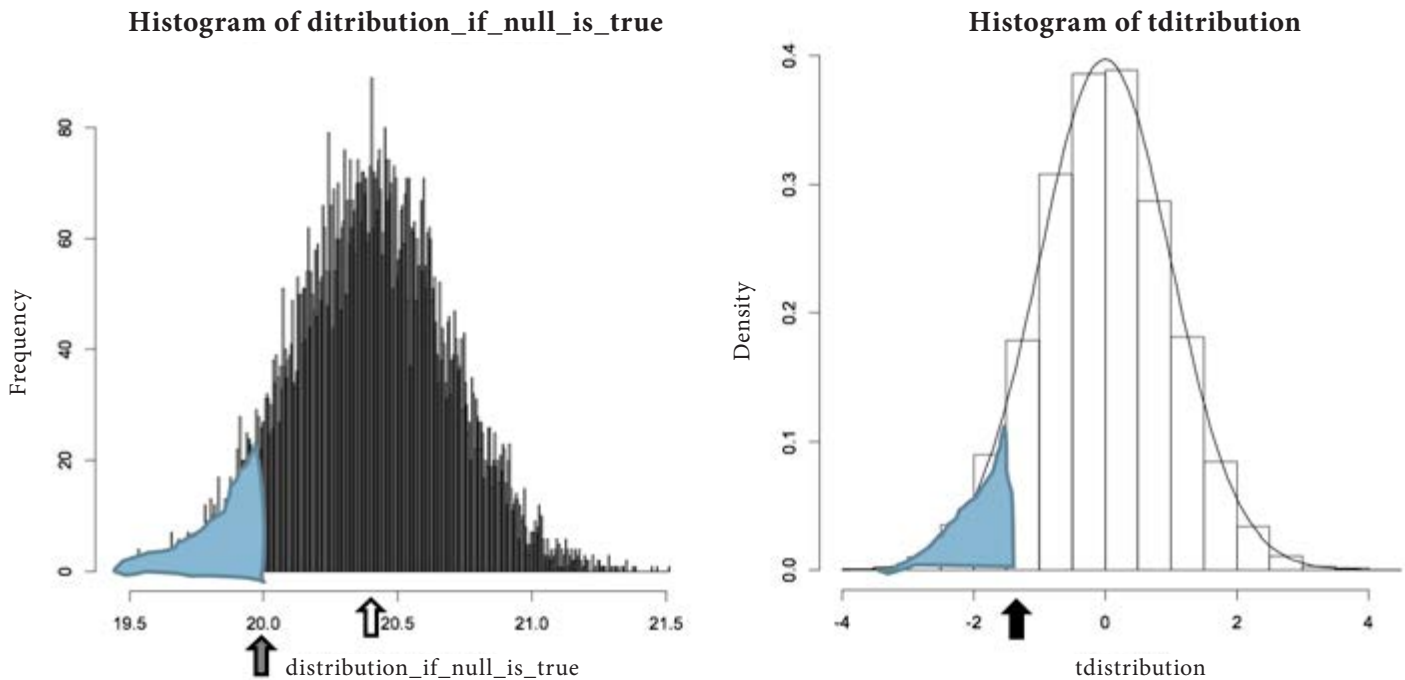
```
sample estimates:
```

```
mean of x
```

```
20.00457
```

Let's look at this:

## STATISTICAL MODELLING OF YOUR DATA



On the left is the expected distribution of samples if the null hypothesis is true, that is, that average surf scoter wing length is 20.4 cm (white arrow). The result that we got from our study was average wing length of 20.00 cm (grey arrow). The probability of getting the result we got, given that the null hypothesis is true, is  $P = 0.08672$ , or 8.67%. In other words, if the null hypothesis is true and we had repeated this study 100 times, we would get the result we got in only 9 studies out of the 100 studies. That seems pretty low, therefore, we conclude that the null hypothesis is unlikely to be true, and therefore we accept our alternative hypothesis that the wing length is less than 20.4. We were able to calculate that exact probability, because we standardized the plot on the left to t-distribution, the plot on the right, with degrees of freedom = 66. On the t distribution, 20.5 corresponds to 0, and 20.0 corresponds to  $t = -1.3761$  (black arrow), which corresponds to the probability of 0.08673.

Another way of saying this plain speak is:

**“If the null hypothesis is true that average surf scoter wing length is 20.4 cm, what is the probability of drawing a sample with average wing length at least as far from 20.4 as 20.0 cm, or even shorter wings?”**

That last part refers to the direction of the alternative hypothesis.

## STATISTICAL MODELLING OF YOUR DATA

The probability of getting the average length that we got in this study (i.e. average of 20.0 cm) or even *shorter* wing lengths, is referred to as the **P value**.

**Stated more accurately, if you had repeated your study many times, a P value is the probability of getting the study result you got, or a more extreme result (in this case smaller) given that the null hypothesis is true.**

It is a convoluted misunderstood sorry beast that P value, so best that you really understand what a P value means. Feel free to review the last few pages again and again if you need to.

Now what about an alternative hypothesis in the opposite direction?

$$H_o = 20.4$$

$$H_a > 20.4$$

$$t = 20.0 - 20.4 / 0.29 = -1.3761$$

$$1-pt(-1.3761, 66)$$

$$=0.913$$

OR

```
t.test(surfwings, alternative = "greater", mu = 20.4)
```

```
One Sample t-test
```

```
data: surfwings
```

```
t = -1.3761, df = 66, p-value = 0.9133
```

```
alternative hypothesis: true mean is greater than 20.4
```

```
95 percent confidence interval:
```

```
19.5252      Inf
```

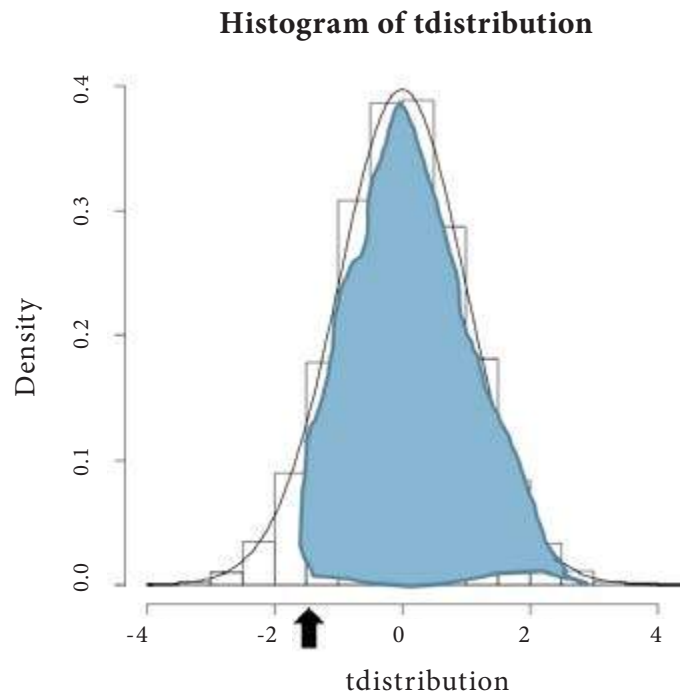
```
sample estimates:
```

```
mean of x
```

```
20.00457
```



## STATISTICAL MODELLING OF YOUR DATA



The t statistic is exactly the same. But now our alternative hypothesis is that surf scoter wing length is greater than 20.4.

In layman terms it would be:

**“If the null hypothesis is true that average surf scoter wing length is 20.4 cm, what is the probability of drawing a sample with average wing length at least as far from 20.4 as 20.0 cm, or even longer wings?”**

At least as far from 20.4 as 20.0 but larger than 20.4 is all of the area under to curve to the left of  $t = 1.3761$ , which is equal to 0.9133, which of course must be  $1 - 0.0867$ .

Thus, given that the null hypothesis is true, that average surf scoter wing length is 20.4 cm, the probability of getting surf scoter wings of 20.0 cm average length or larger wings is 91.33%. That’s a pretty high probability, so we’re going to conclude that our data are consistent with the null hypothesis and we reject the alternative hypothesis that average surf scoter wing length is larger than 20.4 cm.

And now how about an alternative hypothesis is both directions, that is, that average surf scoter wing

## STATISTICAL MODELLING OF YOUR DATA

length is not equal to 20.4? **This is referred to as a two sided test. This is the recommended test in most situations.**

$$H_0 = 20.4$$

$$H_a \neq 20.4$$

$$t = 20.0 - 20.4 / 0.29 = -1.3761$$

$$pt(-1.3761, 66) + (1-pt(1.3761, 66))$$

$$0.173$$

OR

```
t.test(surfwings, alternative = "two.sided", mu = 20.4)
```

One Sample t-test

```
data: surfwings
```

```
t = -1.3761, df = 66, p-value = 0.1734
```

```
alternative hypothesis: true mean is not equal to 20.4
```

```
95 percent confidence interval:
```

```
19.43087 20.57828
```

```
sample estimates:
```

```
mean of x
```

```
20.00457
```

Again, it helps to visualize what we're doing and write it out in plain speak.

**“If the null hypothesis is true that average surf scoter wing length is 20.4 cm, what is the probability of drawing a sample with average wing length at least as far from 20.4 as 20.0 cm, either shorter or longer?”**

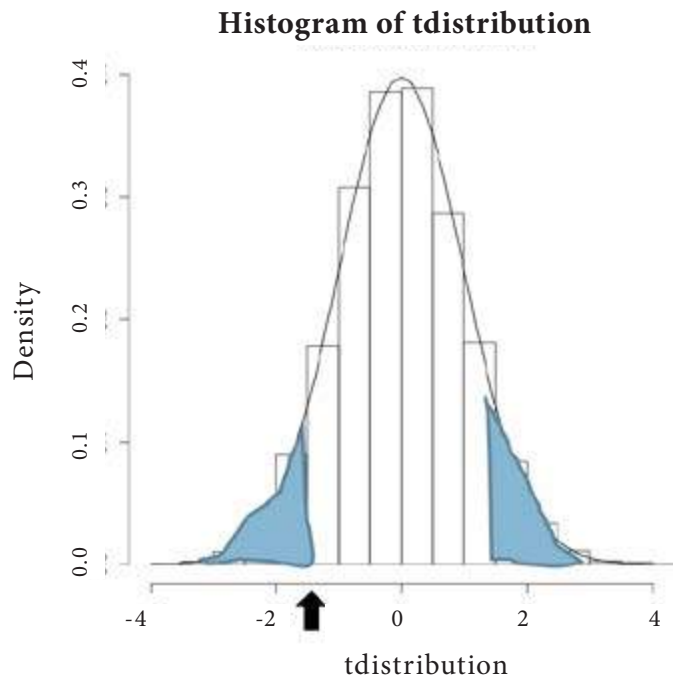
Remind yourself that the arrow on the histogram corresponds to 20.0, and the 0 corresponds to 20.4. We've simply lifted our data distribution up and stuck it onto the t distribution, so we can figure the probabilities associated with getting a sample average of 20.0 cm, given the null distribution centred on

## STATISTICAL MODELLING OF YOUR DATA

20.4 cm.

Notice that the P value for a two sided test is twice the P value for the left hand one sided test.

Given that the null hypothesis is true, the probability of getting average surf scoter wing lengths at least 0.40 cm from 20.4 cm in either direction is only 17.34%. Hmm. That's not a very high probability, implying that the null hypothesis is unlikely, but it's also not a low probability either. So do we reject the null hypothesis or not? That conundrum brings us to the next section.



## P Values and Statistical Significance

Notice in the last section that when we arrived at a P value, we used phrases like, 'that seems like a pretty high probability', or, 'that seems like a pretty low probability'. Well, that doesn't cut the mustard. When conducting statistical hypothesis testing, you must decide before you conduct the test the probability that you're willing to accept to reject the null hypothesis. This is referred to as the **alpha level**, at which we state whether the test is statistically significant ( $P < \alpha$ ) or not ( $P > \alpha$ ).

## STATISTICAL MODELLING OF YOUR DATA

The typical alpha level is 5%. In our example above, if we had specified our alpha level to be 5%, then we would not have rejected the null hypothesis. We would have concluded that we have no evidence to support that average surf scoter wing length was different from 20.4 cm.

If our sample average had been 19.9 instead of 20.0, then our left hand one sided test ( $H_a < 20.4$ ) would have resulted in a P value of 0.045. Since this is less than the 5% cut off, we would have rejected the null hypothesis and concluded that we feel justified in rejecting the null hypothesis that average surf scoter wing length is equal to 20.4.

**There is no good reason to use a 5% alpha level.**

The value of statistical tests depends on sample size and the variability in the data, because the estimated standard error of the mean depends on sample size and variability. High sample sizes result in large t statistics, which are associated with small P values, in the tails of the null distribution. High variability results in small t statistics, which are associated with large P values, and with not rejecting null hypotheses. Ecological data tend to have small sample sizes and high variability, and thus P values greater than the religious standard of 0.05.

These arguments have been raised repeatedly in the last decade or so. Lately, there has been a general shift toward treating P values up to about 20% as fair game to reject the null hypothesis.



# LINEAR REGRESSION

Now that we know how to conduct statistical hypothesis testing, we're going to construct our first statistical model. A statistical model is a measure of the strength of relationships between a response variable and predictor variables.

One of the simplest statistical models is the simple linear regression model, which relates a response variable on the y axis to a predictor variable on the x axis. As an example, here is 10 years of data measuring plant growth in a study plot from the length of leaves. Each data point represents the average leaf length of 15 plants randomly chosen from the plot per year since the beginning of the study (year 0).

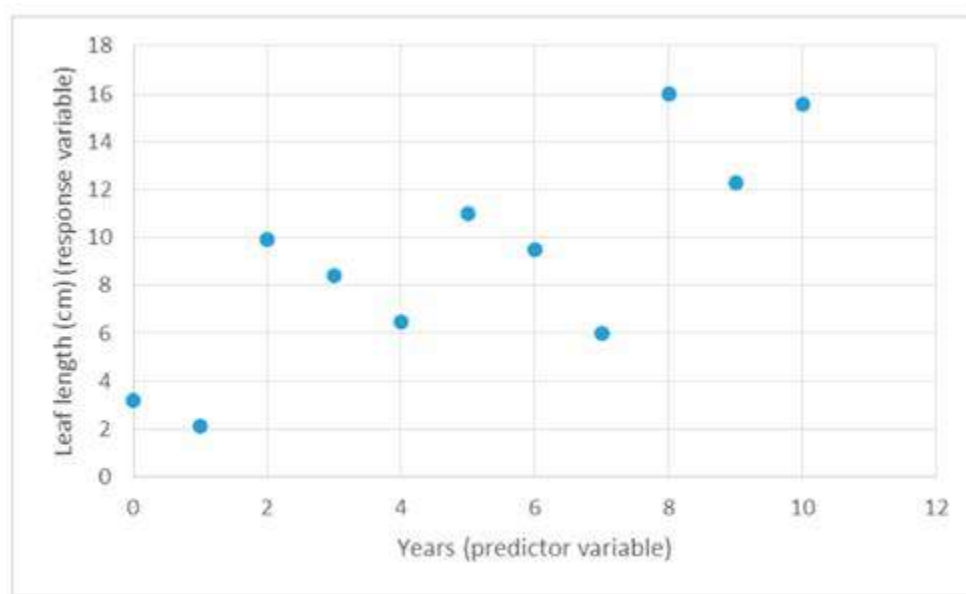


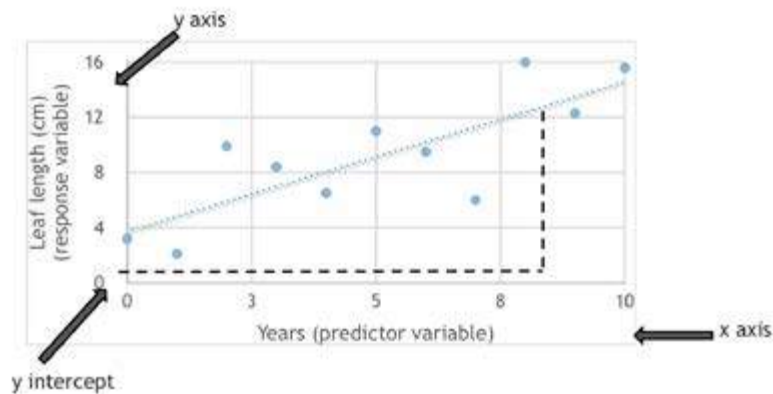
Table 10. Leaf length, in cm, each year of growth.

Years (Predictor)	Leaf Length (cm) (Response Variable)
0	3.2
1	2.1
2	9.9
3	8.4
4	6.5
5	11
6	9.5
7	6
8	16
9	12.3
10	15.6

# LINEAR REGRESSION

So how can we measure the relationship between plant growth and time?

A regression line is comprised of two parameters that describe it – where the line hits the x axis, called the y intercept, and the slope of the line – the steepness or flatness of the line. The slope of the line, the **slope coefficient**, is the quantitative relationship between the response, in this case leaf length, and predictor, in this case, time in years.



The y intercept is the value of the y axis at which the value of the predictor is 0. The slope is the amount by which the response variable changes per unit of predictor variable. An easy way to estimate the slope from eye-balling a scatter plot like this is simply the amount by which the response variable changes from one end of the line to the other, here about  $14.2 - 3.9$ , divided by the amount by which the predictor variable changes from one end to the other, here it is  $10 - 0 = 10$ . So then an eyeball estimate of the slope is 1.03, which means that leaf length increases by about 1.03 cm per year.

The problem with drawing in a regression line from eye balling your data is that different people will draw different lines. Instead, we need to use the mathematics of a linear regression model to find the 'best fit line' through the data.

A regression line can be described mathematically as the following:

$$\text{predicted response} = \text{y intercept} + \text{slope} * \text{predictor}$$

Let's have a look at this. Instead of eyeballing the scatter plot above, regression analysis was run and resulted in an estimate of the y intercept of 3.71 (which you can see looks about right) and the estimate for the slope of 1.08 (our eye balling was pretty close). Thus, for every value of the predictor, the value

## LINEAR REGRESSION

on the regression line, which is referred to as the predicted value, is equal to  $3.71 + (1.08 * \text{years})$ . For every year, the predicted value has been calculated and added to the table below. Note for example that the predicted leaf length at year 4 is 8.06.

Table 11. Leaf length (cm), measured, and predicted leaf length from the regression line ( $y=3.7+1.08*\text{Years}$ ) for each year of growth.

Years (Predictor)	Leaf Length (cm) (Response Variable)	Predicted Leaf Length (cm) ( $y = 3.7 + 1.08*\text{Years}$ )
0	3.2	3.70
1	2.1	4.79
2	9.9	5.88
3	8.4	6.97
4	6.5	8.06
5	11	9.15
6	9.5	10.2
7	6	11.3
8	16	12.4
9	12.3	13.5
10	15.6	14.6

Now let's find out how these parameters (y intercept, slope coefficient) were estimated.

Let's generate and use some normally distributed data. Suppose we harvested 30 mallard ducks, weighed them (grams) and measured the length of one wing per bird (cm) as a measure of the size of each mallard. We're interested in estimating the relationship between mallard size and weight.

First let's simulate these data (create the data set using random sampling from a sampling distribution), which we will call 'mallards' with the following fancy ish R code:

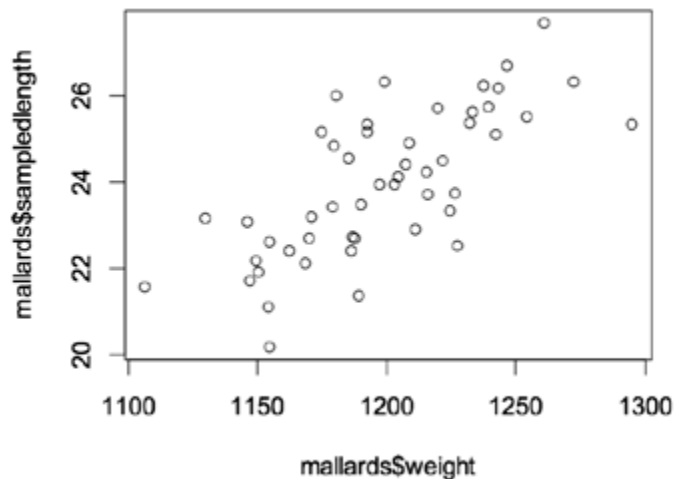
```
set.seed(1114)
data.fn<-function(n = 50, alpha = -12 , beta = 0.03)
{
weight<-rnorm(n, 1200, 40)
predictedlength<-alpha + beta * weight
sampledlength<-rnorm(n = n, mean = predictedlength, sd=1.2)
```



## LINEAR REGRESSION

```
return(list(n = n, alpha = alpha, beta=beta, predictedlength=predictedlength,  
samplelength=sampledlength, weight=weight) )  
}  
mallards<-data.fn()  
  
plot(mallards$sampledlength~mallards$weight)
```

Here is the scatterplot of the data. As you can see, there appears to be quite a strong positive relationship between length and weight. But what is the relationship? In other words, what is the one and only regression line that best fits these data?



As we have seen, a regression line is described by:  
predicted  $y = y$  intercept + slope  $\times$  predictor

A simple linear regression model is described by:  
observed  $y = y$  intercept + slope  $\times$  predictor + **randomness**

In our example, it's:  
observed mallard length =  $y$  intercept + slope  $\times$  mallard weight + **randomness**

What is the randomness about?

You are now familiar with the concept of a population average versus a sample average. There is a

## LINEAR REGRESSION

population out there, for example, of all mallard ducks in the world, and if we calculated the average weight of every last mallard we would get THE population average. But we don't need to do that. We just need to take one sample, and use the average of the mallards in our sample to estimate the population average.

The same is true of the y intercept and slope. If we related the wing length to the weight of every last mallard in the world, we would calculate THE y intercept and THE slope of the relationship. But of course, we can't and we don't need to – we just need to take one sample of ducks, measure the wing length and weight for each, and use the y intercept and slope of our sample to estimate the population y intercept and slope.

Now, you need to imagine that for every predictor value, there is an entire population of y values. For example, for each mallard weighing 1300 g in the world, there is a range of wing lengths (they don't all have the same wing length, because that's the way nature works – everything is variable), and this range of wing lengths is normally distributed.

So there is a normal distribution around every y value.

The mean of each of these normal distributions is the value on THE population regression line – in other words, the regression line is the line through the mean of the y values at each x.

The standard deviation is the same for all the normal distributions at every y value.

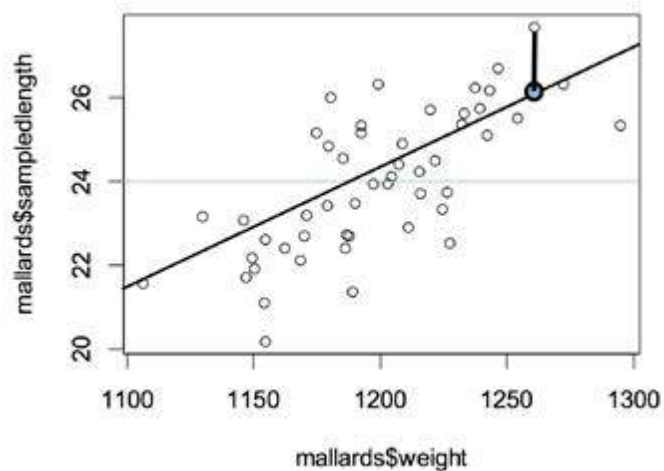
Each observed y value, each wing length we measured, is one random draw from the population of all mallard wings for mallards at that corresponding weight.

Each observed y value deviates from the true population mean. In regression this is referred to as the **residual error** – that's the randomness part of the regression equation. If we repeated the study many times, we would have a sampling distribution of residuals. These are normally distributed with mean of 0 and their standard deviation is the same across all y values. Because it is the same, then we can estimate it as the average of the residuals across all y values. These need to be squared before averaged (because some are negative and sum positive), so this is referred to as the **mean squared error**. The sum

## LINEAR REGRESSION

of the squared residuals is the total variation not explained by the regression line, and is referred to as the **residual sum of squares**.

The best fit line is simply the line for which the difference between the observed lengths and predicted lengths is minimized. To find the y intercept and slope coefficient we could do the math by hand, but let's let R do the work for us.



But first, two things. The first is that we need to pause and remember that what we're about to do is test a statistical hypothesis. Since we want to know whether there is a relationship between size and weight, what we really want to know is whether the slope of the regression is different from 0. If it was a flat line, the slope would equal 0, and there would be no relationship. The null hypothesis is that the slope is equal to zero; typically we test alternatively that the slope is not equal to 0.

Second thing: the estimated slope from a regression has a sampling distribution – it is one sample regression slope of the true population slope. Like the average, its sampling distribution is also the t distribution. So even though this might seem like a more complicated statistical analysis, we're really just doing the same thing we did before. That is, we're testing the probability associated with the regression slope estimated from our data on a t sampling distribution describing the null expectation of slope = 0.

```
mallardsmodel<-lm(samplelength~weight, data = mallards)
summary(mallardsmodel)
```

## LINEAR REGRESSION

#the summary function in R is used often to summarize the output from statistical models

Call:

```
lm(formula = sampledlength ~ weight, data = mallards)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.42897	-0.77762	-0.02165	0.78235	2.59560

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-13.577443	5.271572	-2.576	0.0131 *
<b>weight</b>	<b>0.031334</b>	<b>0.004396</b>	<b>7.128</b>	<b>4.68e-09 ***</b>

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.189 on 48 degrees of freedom

Multiple R-squared: 0.5142, Adjusted R-squared: 0.5041

F-statistic: 50.81 on 1 and 48 DF, p-value: 4.683e-09

The only output that we need to know right now is the part in bold. The estimate for weight is the slope of the regression line, 0.03. Next to it is the standard error of the slope estimate – again picture repeating the study a zillion times, estimating a zillion slope estimates and calculating their standard deviation. The t value is our trusty t statistic from the t distribution describing the null expectation that the slope is 0. Remember, we calculate the t statistic as:

t

= estimated statistic – hypothetical value of population parameter / standard error of statistic

= estimated slope – hypothetical slope / standard error of slope

= 0.03 – 0 / 0.0044

= 7.128 which you can see is the t value in the output table.

And the two sided P value associated with that t statistic is very small at 0.00000000468.

Since the probability of getting the slope that we got from our data is very small under the null hypothesis, we reject the null hypothesis that the slope is 0 and that there is no relationship between size and weight in mallards. Instead we accept the alternative hypothesis that the slope is greater than or less

## LINEAR REGRESSION

than 0, and therefore there is a 'statistically significant relationship between mallard wing length and weight'.

Of course, what we're really interested in is the magnitude of the slope and our certainty around our estimate. That we can now state easily.

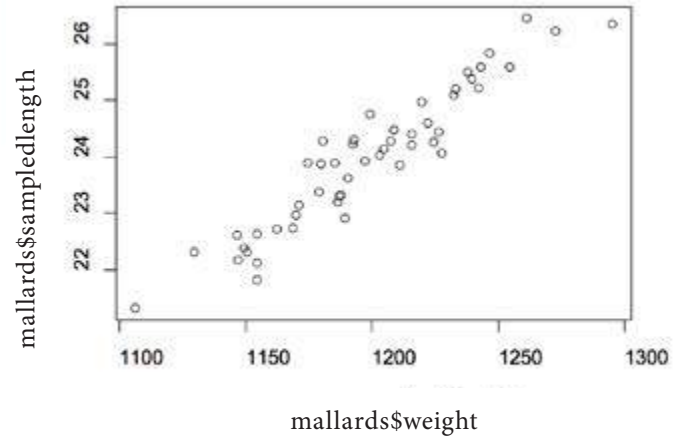
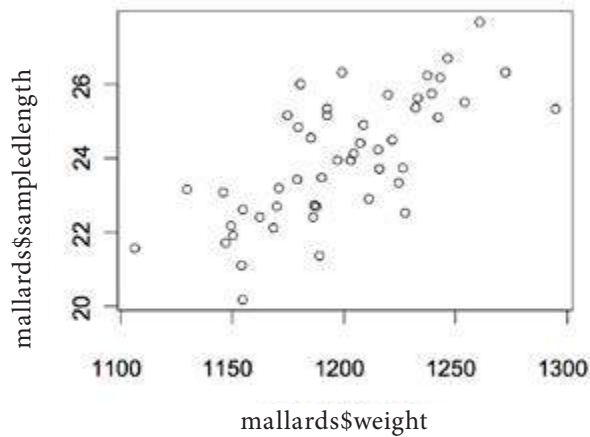
**We estimate that for every one gram increase in mallard weight, mallard size measured as wing length increases by  $0.03 \pm 0.004$  (SE) cm. In other words, across the range of 200 gram difference in mallard weights, wing length is estimated to vary by 6 cm. Just check the plot above and satisfy for yourself that rise over run is  $\sim 6 / 200 = 0.03$ .**

A note about the y intercept, which technically is the wing length at weight = 0. The y intercept is estimated to be negative 13.6, which is biologically impossible, but then, it's biologically impossible for a mallard to weigh nothing. Y intercepts are often meaningless biologically, but are obviously still needed to estimate the best fit line.

Before we go on to slightly more complicated models, we need to pause a moment and reflect. Okay, so we found a statistically significant relationship between mallard size and weight. But, how much of the variability in our data is explained by this statistically significant relationship?

Consider for a moment the data we've been working with on the left, and the data on the right (on the next page), which was generated using the same code, but this time the standard deviation of the normal distributions around each y value was decreased by a third from 1.2 to 0.4. A regression model fit to these data results in very similar parameter estimates and the same highly statistically significant result (try it yourself). But because the amount by which each observation deviates from the regression is less, then the model will fit the data better. In other words, if the regression line were used to predict wing length for a given weight, the regression line for the data on the right (with lower variability) will be more accurate.

## LINEAR REGRESSION



Hence, the process of data analysis does not end simply with finding statistically or biologically significant results. It is also necessary to describe how well the model fit the data – in other words, to measure the residual error. Recall the residual error (or just ‘residuals’) is another term for the difference between an observation and its corresponding predicted value, which is itself an estimate of the true population mean of  $y$  values at that  $x$ .

The residual error is used to derive a measure of how much variability in the data is explained by the model. This is quite simply calculated as  $1 - (\text{sum of the squared deviations of observations from their corresponding predicted values, divided by the total variation})$ . This is referred to as the **R Squared** value; it is interpreted as the proportion of the total variation explained by the regression model.

You could calculate R squared by hand but R does the work for you. It’s given as a part of the model output from the summary function.

Call:

```
lm(formula = sampledlength ~ weight, data = mallards)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.42897	-0.77762	-0.02165	0.78235	2.59560

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-13.577443	5.271572	-2.576	0.0131 *

## LINEAR REGRESSION

```
weight          0.031334          0.004396          7.128          4.68e-09 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.189 on 48 degrees of freedom
```

```
Multiple R-squared:  0.5142
```

```
F-statistic: 50.81 on 1 and 48 DF,  p-value: 4.683e-09
```

Thus, 51% of the variability in the mallard wing lengths is explained by their weight (either measure of R squared is fine to use). In the second data set (with lower variability), the R squared is much higher at 90%.

Residuals are also used to check the assumptions of the model. Model assumptions matter. For simple linear regression, it is assumed that:

1. The population of y values are normally distributed, and the deviations of the observed y values from the mean of the population of y values are also normally distributed. Again, the normal distribution arises from imaginary repeat studies over and over again. This assumption matters because the predicted values are chosen by the model fitting process such that they are the mean of a normal distribution, and the observations are a random draw from that normal distribution.
2. It is assumed that the standard deviation of each population of Y values is the same.
3. It is assumed that each y value is independent of all other y values. In our case, independence means that each x,y pair represents measurements on different ducks, that no two x,y pairs are of the same duck.

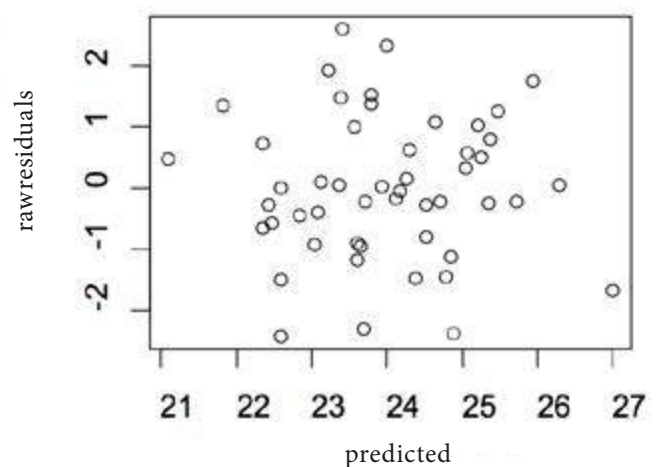
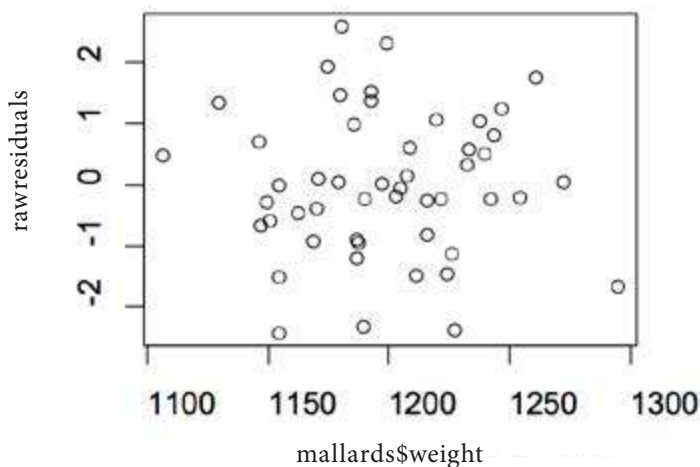
Residuals can be used to visually check model assumptions by plotting them against predictors (weight) or against predicted y values, which in this case is length. Again, you could calculate 'raw' residuals 'by hand':

```
rawresiduals<-mallards$samplength - predict(mallardsmodel)
#the predict function generates a predicted value for each predictor
```

## LINER REGRESSION

Or you can just use the R function residuals:

```
rawresiduals<-residuals(mallardsmodel)
plot(rawresiduals~mallards$weight)
windows()
predicted<-predict(mallardsmodel)
plot(rawresiduals~predicted)
```



These plots are showing us exactly what we want to see – nothing. There is no pattern in the residuals. That is, the size of residuals does not increase or decrease with increasing predictors or increasing predicted values, both of which would suggest non-normal data and different variances across the data set.

The residuals are assumed to be random draws from a normally distributed population of deviates, with mean centred on 0. The standard deviation of the residuals is simply calculated as their average deviation from 0 (which is just their average), after squaring them of course since some are positive and some are negative. The catch is that to calculate the average, instead of dividing by the number of residuals, we divide by the number of residuals minus 2. The average deviation of the residuals is referred to as the **Mean Squared Error**.

```
meansquarederror<-sum(rawresiduals*rawresiduals)/48
```



## LINEAR REGRESSION

1.41

The square root of the mean squared error is the **'residual standard error'**, which is given as output from the linear regression summary function in R:

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.189 on 48 degrees of freedom
Multiple R-squared: 0.5142, Adjusted R-squared: 0.5041
F-statistic: 50.81 on 1 and 48 DF, p-value: 4.683e-09
```

When you become more experienced with statistical analysis you will see firsthand how it is important when applying a statistical model to make sure that the assumptions are met by your data. That is, it is important to use the right model for your data. For example, simple linear regression is not the right model for count data, which is the type of data you will analyze from harvest surveys (counts of bird kills). In the next section, we'll find out why.



# GENERALIZED LINEAR MODELS

In the last section, we used simple linear regression to fit a regression line to normally distributed data. Another name for a simple linear regression model is a general linear model.

What if our data are not normally distributed? For data that are not normally distributed, we can use a type of model called a generalized linear model. Generalized linear models are suitable for ‘discrete’ variables, which are not normally distributed. Unlike continuous variables like weights and lengths, discrete variables can only be certain values. An example is a count – you can only count 2 or 3 or 4 ducks, not 2.5 or 3.5 ducks.

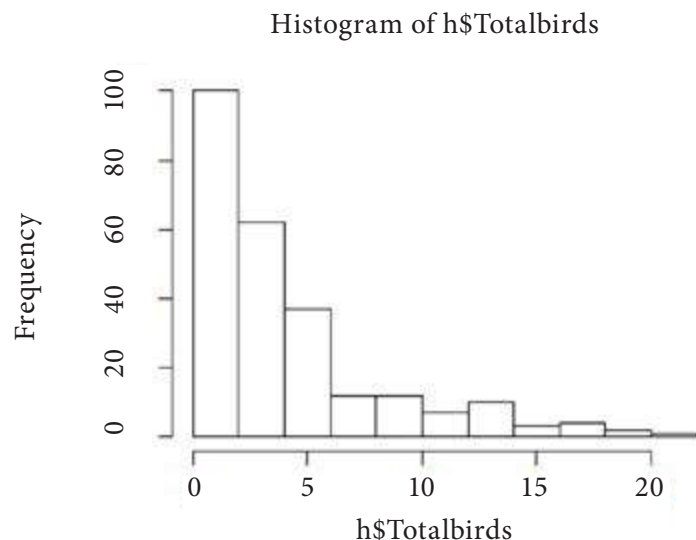
The sampling distribution of counts is the **Poisson distribution**, with which we are about to become familiar.

## POISSON VERSUS NEGATIVE BINOMIAL SAMPLE DISTRIBUTIONS

Load the example data set `harvest.csv` in R, and give it the object name `h` (don’t forget to set your working directory if this is the first time you’ve opened R today). This dataset is similar to but larger than the working data we created in excel above.

Let’s produce a histogram of the total birds harvested per hunter.

```
hist(h$Totalbirds)
```



## GENERALIZED LINEAR MODELS

Clearly these data are not normally distributed. So then, what sample distribution do these data have?

Install the package `dplyr`. Instead of using excel to create a pivot table, we're going to learn how to create one in R.

```
library(dplyr)
pivot<-group_by(h, Region, Season, AgeGroup)
pivot<-summarize(pivot,average=mean(Totalbirds),variance=var(Totalbirds))
#This is to export the pivot table from R
library(xtable)
pivot<-data.frame(pivot)
pivot<-xtable(pivot)
print(pivot, type="html", file="pivot.html")
```

Here are the first few lines of the output ... you can see that, for the most part, the average and variance within a region x season x agegroup stratum are roughly equal.

*Table 12. A few lines of the output from R showing the region, season, age group, average and variance from the harvest.csv document.*

Region	Season	Age Group	Average	Variance
Kootenays	Fall	18 - 30	2.4	3.3
Kootenays	Fall	31 - 40	2.67	0.33
Kootenays	Fall	41 - 50	2.5	0.5
Kootenays	Fall	51 - 60	2.33	4.33
Kootenays	Fall	61 - 75	0.25	0.25
Kootenays	Spring	18 - 30	3.5	12.5
Kootenays	Spring	31 - 40	0.5	0.5
Kootenays	Spring	41 - 50	1.5	4.5
Kootenays	Spring	51 - 60	1.5	0.33
Kootenays	Spring	61 - 75	0.71	0.9

## GENERALIZED LINEAR MODELS

Harvest data are comprised of counts, and the appropriate sample distribution for counts is the Poisson distribution. An important characteristic of the Poisson distribution is that the mean is equal to the variance. Thus, we have good reason to suspect that a model including region, season, and age fit to these data to predict harvest levels will be a Poisson model.

Note that if you calculate the mean and variance of the whole data set, the variance is much bigger than the mean. That's because the variance in the whole data set is comprised of the variability in harvest levels across seasons and regions (which we estimated from our data summaries). As an initial eye-ball to determine whether our data are Poisson distributed, we're concerned with the sample distribution of our data *at the smallest level of resolution*, which in our case, is a Region x Season x Age combination.

It is very common for the variance of ecological count data to be larger than the mean. One of the most common reasons for the added variability is a 'missing' variable that wasn't measured. For example, if we didn't have data on when and where hunters harvested birds, then we would have highly over-dispersed data. Because we have data on these factors, then the variability in harvest levels across regions and seasons is accounted for, and thus the data are Poisson distributed within each strata. If the variance were still higher for each region x season combination, then it could be possible that we'd still be missing a variable. For example, we have no data on hunter preferences. In some regions in the fall preferences by just a few hunters for hunting large numbers of migrating geese could cause the variability in the harvest levels to be higher than the mean.

A sample distribution for which the variance is larger than the mean is referred to as the **negative binomial distribution**. It is described by three parameters, the mean, the variance, and the **dispersion parameter**, which is the amount by which the variance is greater than the mean.

We're going to see how important it is to correct for **over-dispersion** in count data by using a model based on the negative binomial versus the Poisson sample distribution.

Be aware that data can also be under-dispersed, for which the variance is less than the mean. This can arise from highly aggregated data, that is, data that are clustered around a few values. Under-dispersed data are probably more common in ecological data than generalist statisticians think. Most statistical tools have been developed for over-dispersed data; there has been much less focus on under-dispersed

## GENERALIZED LINEAR MODELS

data, and thus it's more difficult to find R statistical programs to help deal with it. That is starting to change, and so if you're so inclined to explore on your own, the COM Poisson model has recently been promoted as one of the best ways to deal with under-dispersed count data.

## MAXIMUM LIKELIHOOD ESTIMATION

Hopefully you now have a fairly good sense of general linear regression, because you'll need to understand the concepts we learned in that section to be able to understand this one. We learned that, for each predictor value on the x axis, there is a normal distribution of possible Y values. That normal distribution of possible Y values is centred on the regression line – that is, the mean of the normal distribution is the predicted value on the regression line. The actual observed y value is assumed to be one random draw from this normal distribution (one normal curve for each y observation). The regression line was fit such that the difference between the mean of the distribution and the observed value was minimized across all y observations. Because some of these differences, referred to as residuals are negative and some are positive, it is their squared value that is minimized. This is called **least squares fitting**. Finding this minimum distance for general linear regression is fairly simple math.

There is another way that regression parameters are estimated from data; the math is complex but the process can be relatively easily visualized. Minimizing the distance between the mean of a population, which is what the predicted value estimates, and the actual observation would increase the probability of pulling that observation as a random sample from the population. For example, a duck weighing close to the average weight will be more likely to be randomly selected from a normal distribution population of ducks simply because there are more ducks of that weight than ducks either lighter or heavier.

For every possible population mean, there is an associated likelihood of the observed value. The combined likelihoods of the observed value for many possible population means is referred to as the likelihood function, and this is combined across all the y values. **Maximum likelihood estimation** is kind of like a trial and error machine that tries out many different possible values of the population mean of the sample distribution at each y, and finds the mean that maximizes the probability of randomly pulling that y value. The best fit regression line is the one for which the observed data are the most likely to have been randomly selected given each population mean. The best estimate of each

## GENERALIZED LINEAR MODELS

population mean are the predicted values at each  $x$ .

Maximum likelihood estimation begins with the set of observed data, then means of the population distribution at each  $y$  value are proposed. It's then determined the probability of pulling each  $y$  value given that particular population mean, and this probability is combined across all  $y$  values into an overall likelihood. Then a different population mean is proposed for each  $y$  value, the probability again calculated for each observation, summed across all  $y$  values, and the overall likelihood calculated across all  $y$  values. That's the second trial. This process is repeated over and over across all possible values for the population mean at each  $y$  value so that we end up with many individual likelihoods that together form a likelihood function. The likelihood function has a maximum value, and the population means that result in that maximum likelihood are the predicted values, which line up to form a regression line, from which the  $y$  intercept and slope are derived. The best fit line then, is the line that makes the observed data the most likely.

**In addition to the way they are fit to data, GLMs differ from general linear models in two important ways.**

**First, the sampling distribution around the  $y$  values can be normal, Poisson, negative binomial, or one of a few other sampling distributions.** These sampling distributions are usually not symmetrical, and for that reason, least squares regression will not work to find the predicted values and thus the best fit line.

Let's try to confuse you a little. The normal distribution is symmetrical, and thus least squares regression will work to find a best fit line. Maximum likelihood estimation will also work to find the best fit line given normally distributed data. In fact, it will result in exactly the same best fit line minimizing the distance between observed and predicted values. Minimizing the distance from the observed to predicted values represents maximizing the probability of randomly drawing the observed value from a population with mean = predicted value. Because the normal distribution is symmetrical, then the probability will be the same if the predicted value is less than or greater than the observed. For non-symmetrical distributions, this no longer works – an observed value 2 units higher than a predicted value will have a probability different than an observed value 2 units lower than the predicted value. Maximum likelihood estimation instead finds the likelihood of different predicted values, given the

## GENERALIZED LINEAR MODELS

observed value, and thus the fitting process is no longer dependent on the symmetry of the distribution.

So that means that general linear models (normally distributed data) can be fit with least squares or maximum likelihood estimation using a generalized linear model. Non-normally distributed data (i.e. count) can only be fit with maximum likelihood estimation using a generalized linear model.

**The second way GLMs differ from general linear models is that the model fit is on the logarithm scale.** Huh, says you.

That part is a bit tricky to understand at first. So, let's start with understanding what we mean by the logarithm (or log for short) scale.

The logarithm (with base 10) of a number is the power to which 10 must be raised to equal that number. For example, the log of 100 is equal to 2 because 10 must be raised to the power of 2 to get 100.

$$\log_{10} 1000 = 3 = 10^1 \times 10^1 \times 10^1$$

$$\log_{10}(430) = 2.63 = 10 \times 10 \times 100.63$$

The natural logarithm of a number is the power to which  $e$  must be raised to equal that number.  $e$  is one of those slightly strange numbers like  $\pi$  (3.14...). It has the value 2.71828... etc etc and to distinguish it from log with base 10,  $\ln$  is used instead of log.

$$\ln 100 = 4.61 = 2.718 \times 2.718 \times 2.718 \times 2.718 \times 2.718 \times 0.61$$

$$\ln 2.718 = 1$$

$$\ln 7.3875 = 2$$

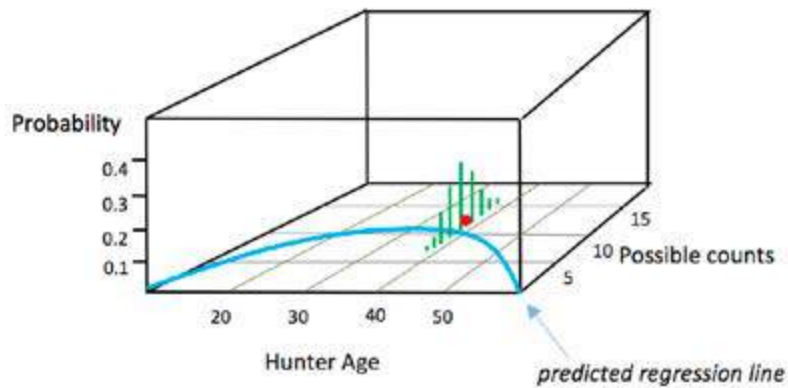
So the logarithm is really just a way of expressing numbers on a different scale. We can transform back to the original scale simply by *exponentiating*. When we exponentiate we call it back-transformation.

e.g.  $\log$  (base 10)  $100 = 2$  then back-transformed is  $10^2 = 100$ .  $\ln(6) = 1.79$  then back-transformed is  $e^{1.79} = 6$



## GENERALIZED LINEAR MODELS

When you analyze data using GLMs with Poisson error, imagine a Poisson distribution (green distribution in the figure below) around each observed  $y$  value (the red dot in the figure below). The blue line in the figure is the regression line fit using maximum likelihood estimation for the relationship of counts of bird kills to hunter age.



The log of the mean of each distribution is equal to the predicted value. So you need to imagine a background log scale where predicted values are found. The model is fit on the log scale. The log scale is always positive. We need the log scale to find predicted values for counts to ensure that the sampling distributions never include negative numbers, i.e. you can't count negative ducks! **This is not the same as taking your observations, log transforming them, and then fitting a general linear model.**

Fitting a Poisson GLM on the log scale to count data means that you need to back-transform to the count scale if you want to interpret your model in terms of counts. Otherwise, you need to interpret your model in terms of log counts.

Now we're ready to fit our first Poisson GLM.

Load the harvest data into R and name it to object

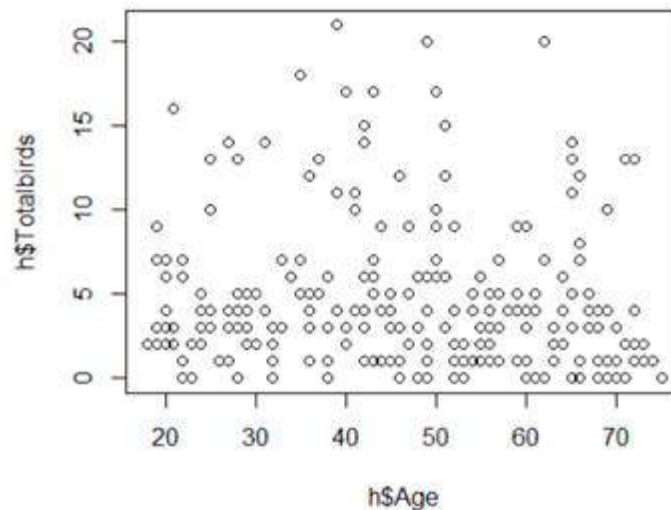
```
h (h<-read.csv("harvest.csv"))
```

We're going to test the effect of hunter age on the total take of birds per hunter.

## GENERALIZED LINEAR MODELS

It's good practice to plot data prior to running analyses to get a visual sense of the relationship you're attempting to estimate

```
plot(h$Totalbirds~h$Age)
```



Our model is:

Totalbirds = y intercept + slope \* hunter age + random error

This is the corresponding R code for that model given a Poisson error distribution.

```
glm<-glm(Totalbirds~Age, data = h, family=(poisson(link=log)))
```

Note the similarity of the summary output to that of the general linear regression we tested in the previous section. That makes sense because we've just tested the same statistical hypothesis. We just

**What does 'Poisson error distribution' mean? It means that the sample distribution around each Totalbird count per hunter is a Poisson distribution, and the predicted value is the best estimate of the mean of the population of bird kills of all hunters of that age. The actual total number of birds killed by a hunter of a particular age is a random draw from this population distribution of the kills of hunters of that age. The predicted value, the estimate of the mean of the population, is the mean that makes the observed kill count the most likely to have been observed.**

And that's it. The function is glm (i.e. generalized linear model), the relationship we want to test is

## GENERALIZED LINEAR MODELS

Totalbirds~Age, the data set is specified by the data argument, the type of sampling distribution is given in the family argument and that we need a log transformation is given by the link argument. Since the default for the poisson distribution is the log this argument isn't actually necessary and can be left out.

```
glm<-glm(Totalbirds~Age, data = h, family=poisson)
```

To see the output, we use the summary function

```
summary(glm)
```

Call:

```
glm(formula = Totalbirds ~ Age, family = (poisson(link = log)), data = h)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-3.3018	-1.7485	-0.5571	0.5668	5.7861

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.884831	0.091856	20.520	< 2e-16 ***
Age	-0.008594	0.001897	-4.529	5.91e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 965.81 on 249 degrees of freedom

Residual deviance: 945.35 on 248 degrees of freedom

AIC: 1648

Number of Fisher Scoring iterations: 5

degrees of freedom

Residual deviance: 945.35 on 248

degrees of freedom

AIC: 1648

Number of Fisher Scoring iterations:

5

### MODEL INTERPRETATION

tested the null hypothesis that the slope of the relationship is 0, versus the alternative hypothesis that the slope is not equal to 0.

Notice, however, that instead of a  $t$  value the output from `glm` gives a  $z$  value. Because the model was fit to the data using maximum likelihood estimation, the standard errors for the slope coefficients are approximations. Thus, the null hypothesis test that the slope is equal to 0 is based on a slightly different test than the  $t$  test, which is referred to as the Wald test. The Wald test statistic is the same as the  $t$  test, slope coefficient/standard error of the slope coefficient, except that the standard error of the slope is an approximation. The Wald test is compared to a standard normal curve (the  $z$  curve, hence the  $z$  value is given).

The best fit line is estimated to have a  $y$  intercept of 1.89 and slope of -0.009. Before interpreting estimates of regression coefficients from GLMs, we have to remember that this estimation is on the log scale. It's okay to interpret on the log scale, but you need to carefully state that the relationships you are reporting are on that scale. In this case, you would report that the predicted log-number of birds killed decreased by 0.009 with each year of hunter age.

In order to report on numbers of birds, we have to back-transform to our original counting scale, 1 duck 2 duck etc

So, the best estimate of the  $y$  intercept is  $e^{1.89}$  and the best estimate of the slope is  $e^{-0.009}$ , which is 6.61 and 0.99 respectively. Recall that often  $y$  intercepts are meaningless when fit to ecological data, as they are from these data. The  $y$  intercept is the estimated birds killed by hunters of age 0, which is certainly meaningless.

The slope of 0.99 means that for every increase in age of hunters by one year, their total take per year of birds decreases by **1% (  $(1-0.99)*100$  )**. It's very important to interpret the back-transformed slope coefficient from a GLM model as a rate ratio – the ratio of two rates.

## GENERALIZED LINEAR MODELS

The P value for the slope is less than 0.001, which is highly statistically significant. What does this mean? Remember that the null hypothesis is that the slope is 0. Given a sampling distribution with mean = 0, the probability of sampling a slope of  $\exp(1)$  or of  $-1$ , or more extreme in either direction, is very very small – in fact, it is 0.00000591. Thus, given a mean slope of 0, we would get the slope that we got essentially never. Thus, it seems likely that the null hypothesis is not true. We do reject the null hypothesis. We conclude that there is a relationship between hunter age and the total number of birds they harvested.

Note in the output that statement: “Dispersion parameter for poisson family taken to be 1”. That’s R telling you that this model was fit to the data under the assumption that the data are Poisson distributed, that is, that there is no over-dispersion. You can calculate the amount of over-dispersion in a model quite simply, since it is just the amount of variability in the sample data above the amount that is expected from a Poisson model.

#Dispersion statistic

```
dispD<-glm$deviance/glm$df.residual
```

```
dispD
```

```
[1] 3.811911
```

Oops! That’s quite a lot of over-dispersion. For a Poisson model, the dispersion statistic should be 1 or fairly close to 1 (less than about 1.5).

## GENERALIZED LINEAR MODELS

### OFF-SETTING

Before we move on with how to deal with over-dispersed data, we have a few topics to cover and another oops to deal with first. We found a statistically significant effect of decreasing harvests with hunter age. But what if that effect is due to decreased effort on the part of older hunters? Maybe older hunters are not able to go hunting for as long as younger hunters, and therefore they harvest less?

Right, we forgot. **We have to standardize the total birds harvested by dividing each count of total birds kills by effort, the number of days spent hunting.** Recall that a Poisson model is used for discrete data – non-integers are not allowed. Therefore, we can't use HarvestRate as a response variable in the model, because those numbers are proportions i.e. 0.33 birds per day, etc.

Of course the smart R programmers thought of this. Hence the **offset** argument in the glm function. The offset allows conversion from counts to rates, while still fitting the model to the counts as integers.

```
plot(h$HarvestRate~h$Age)
```

```
glm<-glm(Totalbirds~Age + offset(log(Effort)), data = h, family = poisson)
```

### QUADRATIC EFFECTS

Before we describe a quadratic effect, we need to first understand that the linear part of a general or generalized linear model is not actually referring to a straight linear line. Instead it's referring to the model components being linear combinations - that is, the sum of the parts rather than the multiplication of parts. Linear means sum of separate parts in stats street talk. For example, the y intercept is added to the slope\*x variable to form the linear regression equation.  $y = \text{intercept} + \text{slope} * x$  variable

So it's quite possible to have a generalized linear model that is actually a curved line. **Sometimes a curved line fits data better than a straight line, and it's up to you as a reliable analyst to**

## GENERALIZED LINEAR MODELS

**test for that.** A curved line is referred to as a **quadratic model** or quadratic effect. In our case, a quadratic model that was humped shaped would mean that harvest levels increase with age to a certain point, and then begin to decline. It would be prudent to test for a quadratic effect in our data, because it's easy to imagine that harvest levels increase to an intermediate age with hunting experience, but then decline with decreasing physical abilities by older hunters (with vision problems, etc).

If you suspect from a data plot that the response variable has a quadratic relationship with a predictor, you can test for this simply by adding the square of the predictor to the model. In our case, Age x Age, or Age<sup>2</sup>. This is treated as another variable with its own slope coefficient. The way to view the quadratic coefficient is simply as a factor that works to take the straight line relationship of the response to the predictor and bend it to whichever curve is the best fit. In the glm code, a capital I needs to be added outside brackets around the quadratic part. Note we have added the offset to the model to ensure we're modelling the rate of harvesting (totalbirds/effort), not the total number of birds harvested.

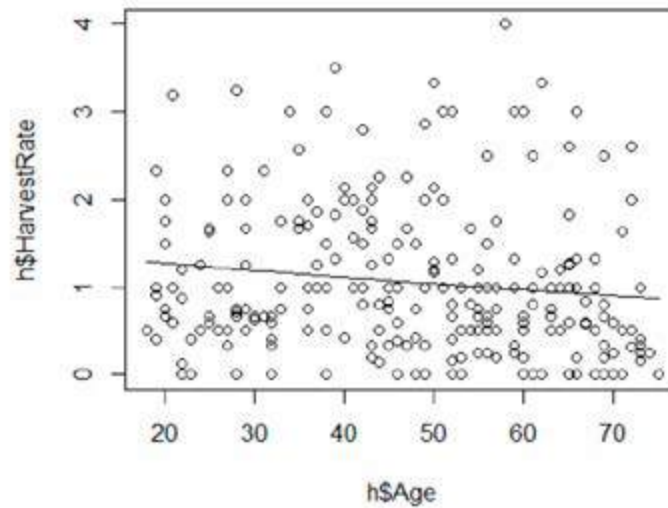
#Straight line linear model

```
glm<-glm(Totalbirds~Age + offset(log(Effort)), data = h, family = poisson)
summary(glm)
```

#let's see how the predicted regression line looks

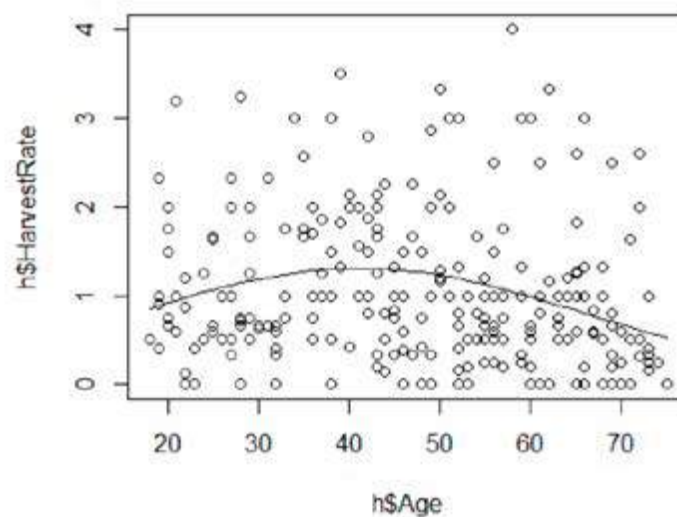
```
Age<-seq(18,75,1)
Effort<-rep(1,58)
Age<-data.frame(Age,Effort)
predicted<-predict(glm, newdata = Age, type = "response")
predicted<-data.frame(predicted=predicted, Age)
plot(h$HarvestRate~h$Age)
lines(predicted~Age, data=predicted, col=1, pch=16)
```

## GENERALIZED LINEAR MODELS



#Curved line linear ('curvilinear') model

```
glmquadratic<-glm(Totalbirds~Age + I(Age^2) + offset(log(Effort)), data = h,  
family = poisson)  
summary(glmquadratic)  
predicted<-predict(glmquadratic, newdata = Age, type = "response")  
predicted<-data.frame(predicted=predicted, Age)  
plot(h$HarvestRate~h$Age)  
lines(predicted~Age, data=predicted, col=1, pch=16)
```



Now we have a little bit of a problem – how do we know which model is the better fitting model?

**AIC** to the rescue! One extremely important piece of information that we've been ignoring so far is the AIC value. It's given on the second last line of the output from the summary of the glm function. The



## GENERALIZED LINEAR MODELS

AIC of the linear model is 1371 and that of the quadratic model is 1336. We'll find out what that means and how to use the AIC value in the next section.

### AIC

AIC stands for Akaike's Information Criterion, which is named after the Japanese statistician who invented it. The rule is: the lower the AIC value, the better fit the model. Thus, the quadratic model fits the harvest data better than the straight line model, which means that we have good evidence that, indeed, harvesting levels increase to an intermediate age, and then decline with age thereafter rather than increasing across all ages.

But what exactly does the AIC measure? Recall that maximum likelihood estimation finds the regression parameters that maximize the likelihood of the observed data. It's the largest value of the likelihood function, created by iteratively (over and over again) proposing predicted values for the y intercept and slope, then combining the individual probabilities of the observed values for each set of proposed values. The likelihood value is a measure of how well the model fits the data, the larger the likelihood the better the fit. The natural logarithm of this maximum likelihood value is referred to as the .... **log likelihood** (it should be ln likelihood but statisticians like to be difficult). Log likelihoods are usually negative numbers, because the likelihoods are usually less than 1 (the ln of any proportion is a negative number).

The AIC value is twice the difference between the number of parameters estimated by the model, symbolized by k, and the log likelihood. Because larger negative log likelihood convey poorer fit, then larger AIC values convey poorer fit of the model to the data.

$$\text{AIC} = 2(k - \log \text{likelihood})$$

It's important to remember that the number of population parameters we're measuring in our model is 3 – one y intercept, one slope, and one quadratic coefficient.

Thus for our totalbirds model

## GENERALIZED LINEAR MODELS

$$\text{AIC} = 2(k - \log \text{likelihood})$$

R has a function to derive the log likelihood

```
logLik(glmquadratic)
'log Lik.' -665.15 (df=3)
```

The df symbolizes degrees of freedom and in this case is referring to the number of parameters being estimated by the model, the y intercept, the slope, and quadratic coefficient.

So  $\text{AIC} = 2(3 - (-665.15)) = 1336$ , which is exactly the same as the AIC value given in the summary output. That still doesn't really tell us what the AIC does.

First, let's think deep thoughts about what a statistical model represents. It represents an idea about how the world works. Given our harvest data, we're testing whether harvest levels are related to hunter age. Imagine we had data on every single factor that affected harvest levels, in addition to region and time of year. Examples could include hunter preferences, environmental conditions that affect bird abundance, whether hunter A's great auntie died the day before he went hunting which affected his ability to hunt well. Imagine a model then with as many predictors for harvest levels as we have hunters in our data base. We would end up with a model that would connect every single observation.

To get a regression line through all the data points, we would have to estimate a lot of complicated parameters from the data. The more parameters we estimate from data, the wider our confidence intervals around each parameter. Which means we would end up with a perfectly fit model and no confidence in our ability to say anything about any other hunters out there in the world. A model like this would tell the truth only about our 185 hunters. This is called over-fitting the data. The 'best' model then does the job – that is, offers us the ability to describe truth beyond our sample with a comfortable degree of certainty - with the fewest predictor variables and thus the fewest parameters. This is referred to as the **principle of parsimony** – do the best with the fewest.

So then, Akaike came up with a pretty simple way to penalize models with a lot of predictor variables, i.e. parameters.  $\text{AIC} = 2(k - \log \text{likelihood})$ . You add the number of parameters to the log likelihood. Models with the same log likelihood but fewer parameters will have lower AIC values. The lower the

## GENERALIZED LINEAR MODELS

AIC value, the higher the likelihood that the underlying population distributions estimated by the model produced the data you observed. A general rule of thumb is that the best (most parsimonious) model is at least 2 AIC units lower than any other model.

## MEASURING MODEL FIT

In the section on general linear models with normal error distributions, we learned that the R squared is a measure of the amount of variability in the data that is explained by the model. The R squared is part of the summary output from the general linear model function of R (`lm`). The `glm` function does not output the R squared, and that's because there isn't an R squared for generalized linear models.

But there are what are referred to as **pseudo R squared** estimators for GLMs. There are several estimators, and they result in slightly different values, but they all are meant to represent the same thing – the proportion of the variability in the data explained by the model.

But first we need to understand the term **deviance** as it is applied to GLMs. In short, the deviance is the GLM analogue to the sum of the squared residuals from a general linear model. Recall the residuals are simply the difference between observations and predicted values, these are squared, and then summed to arise at a measure of the unexplained variance.

A saturated model is a model with one parameter per observation, that is, a model that explains all the variability in the data. The deviance is twice the difference in log likelihood of a specific model and that of the saturated model. Thus, the deviance statistic is a measure of the likelihood that the data were produced by the specific model compared to the likelihood of the data being produced by a perfect model. The smaller the difference, the smaller the deviance and thus the better the model fits the data.

The deviance statistic =  $-2 \times (\log \text{likelihood of a model} - \log \text{likelihood of the saturated model})$

The `glm` output calculates the deviance for the null model – that is, the model with no predictors – and for the fitted model. The deviance for the fitted model is referred to as the residual deviance.

## GENERALIZED LINEAR MODELS

By the way, you can fit the null model simply by using  $\sim 1$  instead of any predictors in your model.

```
glmnull<-glm(Totalbirds~1 + offset(log(Effort)), data = h, family = poisson)
summary(glmnull)
```

```
Null deviance: 681.28 on 249 degrees of freedom
Residual deviance: 681.28 on 249 degrees of freedom
AIC: 1381.9
```

Note that the residual deviance of the null model is the same as the null deviance of the quadratic model.

We can use the deviance of the null compared to the specified model to approximate the amount of variability in the data that is explained by the model.

Here is R code to calculate the Nagelkerke's pseudo R squared for a glm model.

```
Rcsnagel<-function(glm) {
  n<-length(glm$fitted.values)
  Rcs<-1 - exp( (glm$deviance - glm$null.deviance)/n)
  Rnagel<-Rcs/1 - exp(-glm$null.deviance/n)
  out<-list('Rcs'=Rcs, 'Rnagel'=Rnagel)
  class(out)<-c("list", "table")
  return(out)
}

Rcsnagel(glmquadratic)
```

## GENERALIZED LINEAR MODELS

### DEALING WITH OVER-DISPERSION

Back to our data set (`h<-read.csv("harvest.csv")`). Recall that we found considerable over-dispersion in our model of the total take of birds per hunter in relation to hunter age. Now what?

We have a couple of choices but typically, we can choose one of two roads. We can fit more predictors to the data to ‘take care’ of all that additional variability, or, we can fit a negative binomial model, which has a parameter in its distribution that accounts for added variability.

Let’s start with fitting additional predictors. When you add anymore than one predictor to a model, you are entering the sometimes scary but also interesting realm of **multiple regression**. Suddenly things are more complicated. Instead of one x axis, you now have two, or three, or four.

With a little mental effort, we can picture a multiple regression with two predictors. Let’s imagine that harvest levels increase with hunter age as before, but decreases with hunter height (for some strange reason that probably isn’t realistic but is good for illustrating the point). Tall hunters harvested less birds. So, now we have harvest levels increasing with hunter age on one horizontal axis, while simultaneously decreasing on another horizontal axis with hunter height.

A multiple linear regression estimates the relationship between the response and predictors *independently*. **The slope coefficient for each predictor is the estimate of the effect of the predictor on the response, with values for the other predictors held constant.** Thus, for 23 year old hunters, the tallest 23 year olds would have the lowest harvests and the shortest 23 year olds would have the highest harvests, and the slope of the line for all 23 years olds of varying heights would equal the slope coefficient for height. Same for all other ages. Conversely, older 5 foot tall hunters would have higher harvests than young 5 foot tall hunters, and the slope of the predicted line would equal the slope coefficient for age. The effects of the predictors are added up to give one overall predicted value at each possible combination of predictors.

Confused? Let’s add to that. Once you add a predictor to a model, you now have two models. The original model with only one predictor (Model A), and a second model which contains an added

## GENERALIZED LINEAR MODELS

predictor (Model B). Your task as an analyst is to find the most parsimonious model. How do you choose which model is best? The selection process to choose the most parsimonious model is referred to as **model selection**, which we'll review in the next section. The punch line is that we use AIC to compare models, and we choose the model with the lowest AIC. But we must only compare nested models using AIC. That is, we compare models fit to the same data, the only difference being the numbers of predictors in each model. For example, Model A is nested within Model B. If we were to add another predictor to Model B, and call that Model C, then Model B is nested in Model C.

## INTERACTIONS

Another potential layer of complication arises if there is an **interaction effect** between predictors. Interactions are sometimes hard to visualize, because our brains don't work that way. But like quadratic effects, it is upon you good analyst to test for interactions if you suspect they may be present in your data. And just like a quadratic effect, you would test for an interaction by comparing the AIC value of a model without an interaction term to the AIC of a model with an interaction term.

The way to visualize an interaction is actually quite simple. Let's imagine a regression model with two predictors:

$$\text{harvest level} = y \text{ intercept} + \text{hunter age} + \text{hunter height} + \text{error}$$

As we have discussed, the effect of hunter age is independent of hunter height, and vice versa. The slope of the line for the effect of hunter height on harvest levels is the same across all hunter ages. Likewise, the slope of the line for the effect of hunter age on harvest levels is the same across all hunter heights.

Now here is the same model with an interaction between hunter age and height

$$\text{harvest level} = y \text{ intercept} + \text{hunter age} + \text{hunter height} + \text{hunter age} * \text{hunter height} + \text{error}$$

**The interaction term has its own slope coefficient, which like the quadratic term, works to change the relation of hunter age to harvest levels *as the values of hunter height change*.** Thus, the slope

## GENERALIZED LINEAR MODELS

coefficients, that is, the effect of both hunter age and hunter height on harvest levels are no longer independent of one another. And thus their individual slope coefficients must be carefully interpreted. For example, in the interaction model above, the slope coefficient for hunter age is the relationship between harvest level and hunter age only at hunter height 0. The relationships between harvest level and hunter age are different at all other hunter ages.

Knowing the relationship between harvest level and hunter age for 0 year old hunters is of course very unhelpful. There is a simple trick to make models with interactions more interpretable - predictors first need to be **centred** before the model is applied. Centring is done simply by subtracting the mean from each predictor value.

For example

```
h$CentredAge<-h$Age-mean(h$Age)
```

adds a column to the data set that is just a re-scaling of the ages, with mean now equal to 0.

Now with CentredAge added to the interaction model, the slope coefficient for hunter age is the relationship between harvest level and hunter age for the average age of hunters. That's more helpful.

If evidence (i.e. lower AIC value) suggests that our model needs to include an interaction, then the best way to deal with this is simply to calculate the slopes of one predictor at different values of the other predictor. For example, we would determine the estimated slope of harvest level on hunter age, for short, medium, and tall hunters. In the 'putting it all together' example, we will investigate an interaction and how to interpret it in more detail.

### CATEGORICAL PREDICTORS

Last but not least, **categorical predictors**. Until now, all of our examples have been of continuous predictors, like age, height, length, and weight. But two of our three predictors in our harvest data are categorical. There are four categories of Season, and five categories of Region. As you will see, slope coefficients between response and categorical predictors can also be estimated using general or generalized linear modelling.

To fit a model with a categorical predictor, **dummy variables** are created. This is done by glm (or any model fitting function) automatically for you, but you can do the same thing yourself by creating one ‘dummy’ variable per category of a categorical predictor, with the exception of a reference category. For example, four dummy variables for region are created with one category being set as the reference. If we set North as the reference, then in the dummy variable for Kootenays, 0’s are input for Lower Mainland, Thompson and Okanagan, and Vancouver Island and Powell River, and 1’s are input for hunters from the Kootenays. Dummy variables have been created in the harvest.csv file for you so you can visualize how this works.

R glm will automatically choose the reference category by alphabetical order, so you have to specify in the relevel function which category you want as the reference. This is up to you and generally you would just choose whatever makes the most sense. If from data summaries you knew that harvest levels were on average lowest in the north, you might decide to set the north as the reference. Or perhaps it really doesn’t matter and alphabetical choice is as good as any.

Note how these models result in exactly the same output.

```
#set north to be the reference level
h$Region<-relevel(h$Region, ref="North")
glm_categorical<-glm(Totalbirds~Region, family=poisson, data=h)
summary(glm_categorical)
glm_categorical2<-glm(Totalbirds~dummykootenays+
dummylowermainland+dummysouth+dummythompson+dummyvancouverisland, family=poisson,
```



## GENERALIZED LINEAR MODELS

```
data=h)
```

```
summary(glmcategorical2)
```

```
Call:
```

```
glm(formula = Totalbirds ~ Region, family = poisson, data = h)
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	0.79159	0.08839	8.956	<2e-16	***
RegionKootenays	-0.11864	0.13423	-0.884	0.377	
RegionLower Mainland	0.96774	0.10712	9.034	<2e-16	
***					
RegionThompson and Okanagan	1.01607	0.10472	9.702	<2e-16	***
RegionVancouver Island and Powell River	1.09970	0.10640	10.335	<2e-16	***
---					

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 965.81  on 249  degrees of freedom
```

```
Residual deviance: 711.08  on 245  degrees of freedom
```

```
AIC: 1419.8
```

The p values are indicating that the slope coefficient for the Kootenays is not different from that of the North. The slope coefficient for the Lower Mainland, Thompson, and Vancouver Island regions are statistically different from that of the north. If this were the end of our analysis, we would report the magnitude of the slopes and use the standard errors of the slopes to calculate confidence intervals around them.

How do we interpret these coefficients? Note that there is no coefficient for the North region, and that's because we set it to be the reference against which all the other regions are compared. As the reference, the slope coefficient for the north is the y intercept. **The slope coefficients for the other regions is the best estimate of the difference in harvest levels in that region compared to the north region, with all other variables in the model held constant.**

## GENERALIZED LINEAR MODELS

The slope of one region is an estimate of the average amount by which harvest levels differ between that region and the north region (with all other variables held constant). For example, an estimate of the harvest levels in the lower mainland is  $0.7916 + 0.9677 = 1.759$ . Remember this all on the natural logarithm scale to get the estimate on the counting scale, we back transform.  $e^{1.759} = 5.808$ .

This is simply the sample average of harvest levels for the lowermainland!

```
lm<-subset(h,Region=="Lower Mainland")
mean(lm$Totalbirds)
[1] 5.808511
```

Fitting a model with only one categorical predictor is not interesting. It only becomes interesting when we add another predictor. If we added a continuous predictor like hunter age, then the slope coefficient for a particular region would be an estimate of the difference in harvest rate between a particular region and the north region, with hunter age held constant.

```
glm<-glm(Totalbirds ~ Region + Age+I(Age^2) + offset(log(effort)),family=poisson,
data=h)
summary(glm)
```



# PUTTING IT ALL TOGETHER

## *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

Now that we have all of that background knowledge simmering away in our heads, we're ready to take on GLM modelling.

So to review, we have one response variable, the total count of birds harvested in one year by a sample size of 250 hunters. These hunters come from five regions, and they hunted across three seasons (we exclude winter data because there isn't enough of it). We also know their age. So we have three predictor variables. We need to standardize harvesting by effort, which we can do simply by adding 'effort' to the offset argument.

We think the data might be Poisson distributed, but we're not totally sure. We know that if we fit the fullest possible model we have (that is, with all three predictors), and the variance in the data is greater than that expected by the Poisson model, then we have an over-dispersed Poisson model. That could mean we're missing a variable to explain that extra variation. But since all we have are three predictors, then we'll have to fit a negative binomial glm instead. But we're not there yet, since we have yet to fit a full model.

We're also aware that we have to keep a look out for quadratic effects in our continuous variable age, and that there could be an interaction effect between any of our predictors.

We know we will use AIC to compare nested models to the full model to select the most parsimonious model.

So let's begin!

**We're working with 'harvest.csv'.**

Let's start with the fullest possible model, excluding quadratic and interaction effects ...

```
#make sure you set the references to north and summer
```

```
h$Region<-relevel(h$Region, ref="North")
```

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

```
h$Season<-relevel(h$Season, ref="Summer")
summary(glm(Totalbirds~Age + Region + Season + offset(log(Effort)), data = h,
family = poisson))
```

The AIC is 1045.

We've already seen good evidence that the effect of age on harvest levels is quadratic. In reality, you might have suspected something from the scatter plots in excel during the data summary process. Intermediate aged hunters were often successful and there was a decrease in harvest levels by older hunters, suggesting a humped relationship.

```
glm<-glm(Totalbirds~Age +I(Age^2)+ Region + Season + offset(log(Effort)), data =
h, family = poisson)
summary(glm)
```

The AIC is 1034.

Thus, the data are much more likely to have been sampled from a population in which the relationship of harvest levels to hunter age is quadratic rather than a straight line.

The P values for all categories of Region except Kootenays are highly statistically significant, meaning that we reject the null hypothesis that the slope coefficients in each region are the same as in the north. We do not reject this null hypothesis for the Kootenays, and conclude that the slope coefficient for the Kootenays is the same as in the north. Given the very high probability that there is a region effect, we would definitely keep Region in the model. But let's just see what happens to the AIC value if we drop it.

```
summary(glm(Totalbirds~Age +I(Age^2)+ Season + offset(log(Effort)), data = h,
family = poisson))
```

The AIC is 1246.

Since this AIC is much higher, it indicates that this model without the Region effect represents a model of the world that is much less likely to have produced the observed data than the model with the Region effect.

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

Same goes for Season and Age. Test out how dropping Season and Age result in much higher AIC values than the full model.

Okay, so model selection by AIC has resulted in all three of our predictors staying in the model. All of them contribute meaningfully to explaining variation in harvest levels. The effect of Age is quadratic.

```
glm<-glm(Totalbirds~Age +I(Age^2)+ Season + Region + offset(log(Effort)), data =  
h, family = poisson)
```

Let's pause here and see if we've got the right model. Are the data Poisson distributed?

```
dispD<-glm$deviance/glm$df.residual  
dispD  
[1] 1.3
```

We'll look at that. Recall that the dispersion statistic was 3.8 in the model that included only a straight line effect of age (it is 2.6 for a quadratic effect). Region and Season thus accounted for the added variation, above that expected by a Poisson model. Thus, the Poisson error distribution is the correct distribution to use for these data. A rule of thumb for when to correct for over-dispersion by adding predictors or using the negative binomial is an over-dispersion statistic greater than about 1.5.

Now that we've confirmed we're working with the correct error distribution, we need to determine whether there are interactions between the predictors.

Before testing for interaction effects, it is prudent to first think about what an interaction would actually mean. It doesn't make any sense to test for an interaction that is highly unlikely to occur. In our case, interactions between each set of two predictors could conceivably occur, so we need to test for them. Note that three way interactions are possible, but are difficult to interpret so we'll leave that discussion out.

An interaction between hunter age and region would mean that the effect of hunter age on harvest levels differs between regions. This could arise, for example from cultural differences between regions.

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

Perhaps in some regions, younger hunters are the successful hunters, while in another region it is the older hunters who harvest more.

An interaction between hunter age and season would mean that the effect of hunter age on harvest levels differs across seasons. This could arise, for example if hunter experience is correlated with hunter age, and birds that require considerable experience to kill are harvested more in one season than another. In that season, the older hunters would have the highest harvest levels. In the other seasons, there could be no relationship between hunter age and harvest levels if birds are simply killed in proportion to their encounter rate.

An interaction between region and season would mean that the difference between regions in comparison to the reference region would not be the same across all seasons. Again, that could be due to cultural reasons, with hunter preferences for a particular season in one region causing the interaction effect.

So let's test for these interactions in turn. Recall that the AIC of our most parsimonious model so far is 1034.

```
summary(glm(Totalbirds~Age +I(Age^2)+ Region + Season + Region*Season +  
offset(log(Effort)), data = h, family = poisson))
```

```
summary(glm(Totalbirds~Age +I(Age^2)+ Region + Season + Region*Age +  
offset(log(Effort)), data = h, family = poisson))
```

```
summary(glm(Totalbirds~Age +I(Age^2)+ Region + Season + Season*Age +  
offset(log(Effort)), data = h, family = poisson))
```

Of these three models, the model with the lowest AIC units by far stands out as the most parsimonious model:

```
glm<-glm(Totalbirds~Age +I(Age^2)+ Region + Season + Season*Age +  
offset(log(Effort)), data = h, family = poisson)
```

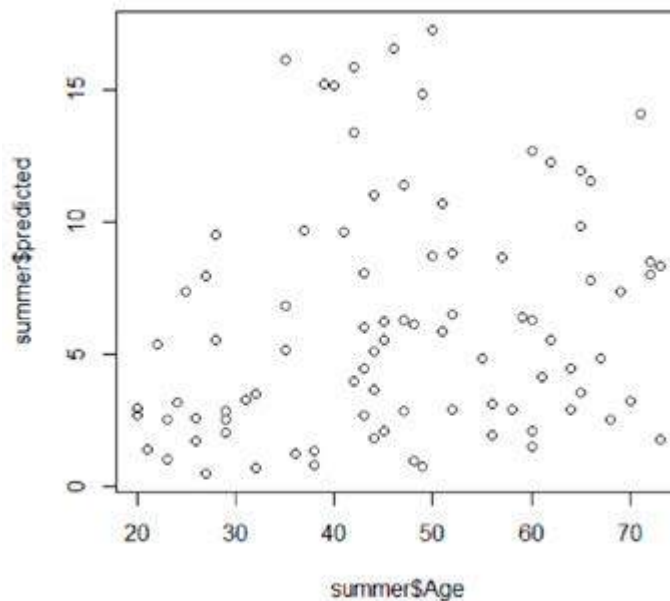
## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

## VISUALIZING AN INTERACTION EFFECT

The predict function can be used to generate predicted values from a model. If you input 'response' into the type argument, the predictions are given on the count scale; 'link' gives predictions on the log scale. You can either supply a new predictor dataset, or by default, simply use your observed data set, as we do below.

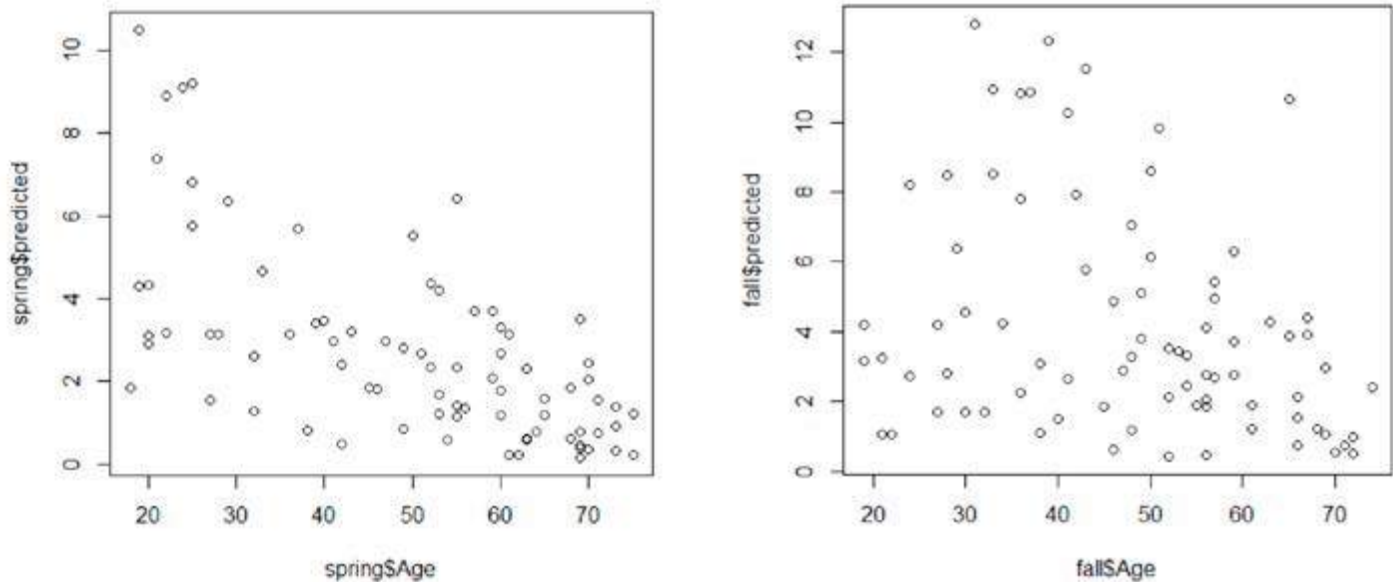
```
predicted<-as.numeric(predict(glm, type="response"))
h$predicted<-predicted
spring<-subset(h,Season=="Spring")
plot(spring$predicted~spring$Age)
windows()
fall<-subset(h,Season=="Fall")
plot(fall$predicted~fall$Age)
windows()
summer<-subset(h, Season=="Summer")
plot(summer$predicted~summer$Age)
```





## PUTTING IT ALL TOGETHER

### Modelling Harvest Data Using Poisson and Negative Binomial GLM



Now we can see more clearly visualize an interaction at work – put simply, the relationship between harvest levels and age is not the same across the three seasons. Summer stands out as the odd ball. We can confirm this by setting the reference level to be either spring or fall, and then running the model again to see if the P value suggests whether the slopes between spring and fall are different. You'll find that the P value is 0.95, and the slope coefficient is very small, indicating no difference in the relationship between harvest levels and age between spring and fall, which is what we can see above.

Rather than interpret this model any further, we're now going to switch over to what might be a more realistic data set for a harvest study. The data we've been working with are nicely, almost perfectly Poisson distributed. Too perfect! Let's become familiar with working with the negative binomial model.

## THE NEGATIVE BINOMIAL MODEL

To fit a negative binomial model, you need to install the package MASS and use the `glm.nb` function

```
library(MASS)
```

Load the negative binomial data set ('harvest\_nb.csv') on the CD and name the data set `h`. Don't forget

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

to set the reference levels.

```
h$Region<-relevel(h$Region, ref="North")
h$Season<-relevel(h$Season, ref="Summer")
```

Let's have a look at these data first.

```
library(dplyr)

pivot<-group_by(h, Region, Season, AgeGroup)

pivot<-summarize(pivot, average=mean(Totalbirds), variance=var(Totalbirds))
pivot
```

Unlike the Poisson data, you can see in the pivot table that many of the variances per Region\*Season\*Age Group are much larger than the mean. That's an indication of over-dispersed data. To account for that extra variability, we could try fitting an additional predictor to the model to see if it can 'take care' of all that added variability. But we don't have one. Let's compare the Poisson and negative binomial models fit to these data.

```
glm<-glm(Totalbirds~Age +I(Age^2)+ Region + Season + Season*Age +
offset(log(Effort)), family=poisson, data = h)
```

```
glmnb<-glm.nb(Totalbirds~ Age +I(Age^2) + Region + Season + Season*Age
+offset(log(Effort)), data = h)
```

So first of all, note that the negative binomial model is very similar to the Poisson model – each predictor contributes significantly toward explaining variability in harvest levels. Feel free to proceed with model selection using AIC to confirm that this is the most parsimonious model by far.

We can also see from the output of this negative binomial model that the data are indeed over-dispersed. The amount of over-dispersion (the amount by which the variance is larger than the mean) is measured

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

by the dispersion parameter, which is

$$\text{overdispersion} = 1 + \text{mean}/\theta$$

The model output gives 13.01 as an estimate for  $\theta$ . The expected mean count is

```
mean(exp(predict(glm)))  
28.8
```

Thus, the over-dispersion, the multiplicative amount by which the variance is greater than the mean is  $1 + 28.8 / 13 = 3.2$

Or we can estimate this way ...

```
dispD<-glm$deviance/glm$df.residual  
dispD  
2.9
```

Now, let's see if the negative binomial model did a good job of accounting for that added variability ....

```
dispD<-glmnb$deviance/glmnb$df.residual  
dispD  
1.3
```

In calculating the dispersion statistic (note that the dispersion parameter  $\theta$ , and the dispersion statistic are not the same thing!), we see that it is less than 1.5, which is great – that means the negative binomial has done its job. The negative binomial function has fit a model to these over-dispersed data so that the over-dispersion was accounted for by the dispersion parameter of the negative binomial sampling distribution. The model was then fit using a sampling distribution with the mean equal to the variance, and the actual variance was very close to the variance expected by this distribution.

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

#### The Consequences of Fitting the Wrong Model

Fitting a Poisson model to over-dispersed data results in smaller P values than the observed data actually support. That's because the parameters are fit assuming that the true variance is not different from the population mean. Predicted values are the best estimates of population means at each  $x$ , but these are predicted using a sampling distribution with a smaller variability than is actually present in the data. The actual observations are much further from the predicted values than they should be. This results in biased estimates of the true population means, and lower P values leading analysts to conclude that there is a relationship when in fact there may be no evidence for one.

## PREDICTING TOTAL HARVEST

Once you have arrived at the most parsimonious model given your data, you're ready to estimate the total number of birds harvested per year by hunters in your organization. Keep in mind that you can easily construct models per species group, i.e. ducks, geese, upland birds, rather than models for all species combined, as we have done. The only difference is at the data summary stage – you need to sum within species groups per hunter, rather than the total number of birds per hunter.

Estimating the total harvest is really just a matter of arithmetic once you have a good model. The hardest part practically may be arriving at a good estimate of the actual number of hunters per region per season per age. Once you have created a data set of all hunters, you can use the predict function applied to the most parsimonious model to generate the estimated harvest per hunter, which you simply add up. To generate confidence intervals around this estimate, you can use a method referred to as **bootstrapping**.

Bootstrapping is a process of re-sampling your data, usually with the original sample size. Say your sample size is 185. You would create one bootstrap sample by random selection of one observation, which you would put 'back' in the dataset, then random selection of a second observation, then you

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

would put that one back, and so on until you reached 185 randomly selected observations. That means that any particular observation might get selected more than once for the same bootstrap sample. For a bootstrap sample of 100, you would create 100 bootstrap samples of 185 observations each. Then you would apply the model to each of the 100 datasets. At the end, you would end up with 100 predictions of the total harvest using the model, from which you can calculate the standard error and thus confidence intervals.

## Creating a Data Set of All Hunters

In an ideal situation, you have a very good estimate of the total number of hunters that went hunting each year per hunter age, region, and season. For permitted hunts, these data are collected as part of the permit issue. For example, migratory bird hunters in the US must provide their age and region of harvesting when they purchase a hunting licence. Since not everyone who buys a licence uses it, the total number of active hunters must be estimated. The proportion of active hunters is calculated from those who responded to a harvest survey, and then simply multiplied by the number of licences issued. For example, if 1,000 licences were issued, and 300 hunters responded to the survey, of which 250 used the licence and went hunting, and 50 did not, then the total number of active hunters is estimated to be  $1000 \times 250/300 = 833$ .

Since the harvesting you are trying to estimate is not permitted, it will be more difficult for you to estimate the total number of hunters per strata, and thus to derive accurate estimates of the total bird harvest. You could perhaps overcome this challenge through community engagement, by encouraging hunters who intend to hunt to register every year.

Another less rigorous way to overcome this challenge is to use local knowledge to estimate the total number of hunters in each strata. Local community members may have a very good idea of the number of hunters, their age, and when they generally hunt. Regular communication with several key local knowledge holders could be used to build a data base containing estimated total numbers of hunters per sub-region, season, and age group, which you could then merge into a regional database.

It may be easier to estimate the actual numbers of hunters per age group, and then just assume a uniform distribution of numbers of hunters per age within that age group. For example, if you estimate

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

there were in total 40 hunters aged 20-29, then you'd just assume there were four hunters per age.

Okay, suppose you got there. You have an estimate of the numbers of active hunters per region, season, and age group. Open the csv file titled 'totalhunters.csv'. Let's pretend that this is your final estimate of the total number of active hunters per stratum. The 'Hunters' column is the count of hunters per stratum.

We need to transform this table into a data set, then apply our parsimonious model to this data set, to derive estimates of total harvest.

We first start by creating a matrix of our predictor variables. Note that this has already been created in the csv file, but it's handy to know how to do this in R.

```
#create matrix of predictors
```

```
total<-read.csv("totalhunters.csv")
```

```
Season<-c('Fall', 'Spring', 'Summer')
```

```
#define a vector called Season
```

```
Region<-c('Kootenay', 'Lower Mainland', 'North', 'Thompson and Okanagan',  
'Vancouver Island and Powell River')
```

```
AgeGroup<-c('17-30', '31-40', '41-50', '51-60', '61-75')
```

```
allcombos<-data.frame(expand.grid(Region, Season, AgeGroup))
```

```
#the expand.grid function creates a matrix of all possible combinations of the categories of predictors,  
e.g. Fall + Kootenays + 17-30, then Fall + Kootenays + 31-40, etc
```

```
allcombos$Hunters<-total$Hunters
```

```
#add to this data frame the count of hunters. Again, this is just a copy of the csv file.
```

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

#now we start using R to create the data set we want

```
library(splitstackshape)
```

```
fulldataset<-expandRows(allcombos, "Hunters")
```

#the expandRows function replicates rows according to the number of hunters per strata combination. For example, 24 rows are created of Kootenays + Fall + 17-30.

#now change the categorical variable AgeGroup to a continuous variable "Age" assuming uniform distribution of ages within each age group.

```
age1<-c()
```

```
for(i in c(24,45,27,51,24,19,35,21,39,18,14,27,16,30,14)){
```

```
x<-round(seq(17,30, length.out=i),0)
```

```
age1<-c(age1,x)
```

```
}
```

```
age2<-c()
```

```
for(i in c(32,60,36,68,32,25,47,28,53,25,14,27,16,44,14)){
```

```
x<-round(seq(31,40, length.out=i),0)
```

```
age2<-c(age2,x)
```

```
}
```

```
age3<-c()
```

```
for(i in c(49,90,54,101,49,38,70,42,79,38,15,15,15,40,18)){
```

```
x<-round(seq(41,50, length.out=i),0)
```

```
age3<-c(age3,x)
```

```
}
```

```
age4<-c()
```

```
for(i in c(32,60,36,68,27,25,47,28,53,20,14,27,16,35,14)){
```

```
x<-round(seq(51,60, length.out=i),0)
```

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

```
age4<-c(age4,x)
```

```
}
```

```
age5<-c()
```

```
for(i in c(25,60,25,30,25,19,35,21,39,19,15,25,15,20,15)){
```

```
x<-round(seq(61,75, length.out=i),0)
```

```
age5<-c(age5,x)
```

```
}
```

```
fulldataset$Age<-c(age1,age2,age3,age4,age5)
```

#add the continuous age predictor to the data set

```
library(plyr)
```

```
fulldataset<-rename(fulldataset,c("Var1"="Region","Var2"="Season","Var3"="AgeGroup"))
```

#now we need to add an Effort column, where effort is set to 1

```
fulldataset$Effort<-rep(1, 2528)
```

#that completes the creation of the full data set containing the estimated total number of active hunters per age, season, and region.

#Now apply the most parsimonious model to this data set to generate predicted harvest of birds per hunter per day.

Predict on the log scale. Here we use the negative binomial data set

#most parsimonious model, and just in case, the sample data and reference categories are provided again

```
library(MASS)
```



## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

```
h<-read.csv('harvest_nb.csv')
```

```
h$Region<-relevel(h$Region, ref="North")
```

```
h$Season<- relevel(h$Season, ref="Summer")
```

```
glm<-glm.nb(Totalbirds~Age +I(Age^2)+ Region + Season + Season*Age +  
offset(log(Effort)), data = h)
```

```
HunterHarvest<-predict(glm, newdata=fulldataset, type="response")
```

#the predict function applies the best model to our full data set of all hunters to generate a predicted total harvest per hunter. 'Response' in the type argument means we're telling R to give us the predictions on the counting scale, not on the log scale.

```
TotalHarvest<-sum(HunterHarvest)
```

#then we simply sum these to derive the total harvest across all hunters.

```
fulldataset$Totalbirds<-round(HunterHarvest,0)
```

Note that if we preferred to predict the rate of harvest per hunter, we would need to include an Effort column in the newdata dataframe, with each row =1. That way, the predict function will output the harvest per day rather than the total harvest per hunter.

## PUTTING IT ALL TOGETHER

### *Modelling Harvest Data Using Poisson and Negative Binomial GLM*

## Bootstrapping to Derive Confidence Intervals

#Bootstrapping the total harvest. First we need to create a function to extract the total predicted harvest

```
library(boot)
```

#function to derive total harvest from the data

```
totalharvest<-function(formula, data, indices) {  
  d<-data[indices,]  
  fit<-glm.nb(formula, data=d)  
  return(sum(predict(glm, newdata=d, type="response")))  
}
```

#bootstrapping with 1000 replications. The full data set of all hunters is re-sampled 1000 times, each time the model is applied to the data set and the predicted total harvest calculated, resulting in 1000 predicted total harvests.

```
set.seed(123)  
results<-boot(data=fulldataset, statistic=totalharvest, R=1000, formula=  
Totalbirds~Age +I(Age^2)+ Region + Season + Season*Age+offset(log(Effort)))
```

```
results
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = fulldataset, statistic = totalharvest, R = 1000,  
      formula = Totalbirds ~ Age + I(Age^2) + Region + Season +  
              Season * Age)
```

## PUTTING IT ALL TOGETHER

### Modelling Harvest Data Using Poisson and Negative Binomial GLM

Bootstrap Statistics :

	original	bias	std. error
t1*	24112.42	-3.521493	368.3359

```
plot(results)
```

#95% confidence intervals note that this assumes normality (which the plot suggests is an ok assumption)

```
boot.ci(results, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

```
boot.ci(boot.out = results, type = "norm")
```

Intervals :

Level	Normal
95%	(23394, 24838 )

Calculations and Intervals on Original Scale

The median of the bootstrap samples is our best estimate of the total annual harvest.

```
median(results$t)
```

```
[1] 24099.71
```

**Thus, we conclude that we estimate the total annual harvest of migratory birds to be 24,100 (23394-24838 95% CI), or stated differently ...  $24,100 \pm 3\%$  (95% CI) migratory birds per year.**

# APPENDIX 1

## Example Harvest Survey and Instructions

THIS SURVEY IS CONFIDENTIAL. YOUR INFORMATION WILL BE KEPT CONFIDENTIAL AND WILL ONLY BE USED FOR MONITORING PURPOSES

Your age \_\_\_\_\_ Do you have a Migratory Game Bird Hunting Permit? (circle) Yes No

Region (see map for names) \_\_\_\_\_

Date you filled out this sheet (month/day/year) \_\_\_\_\_

PLEASE FILL OUT THE FOLLOWING WHETHER OR NOT YOUR HUNTING TRIPS WERE SUCCESSFUL

HUNTING LOCATION			
_____			
Year _____	Month _____	Number of days you spent hunting _____	
Geographic coordinates _____ e.g. 49° 15'49.90 N 122° 41' 13.29 W			
Species	How many?	Species	How many?
HUNTING LOCATION			
_____			
Year _____	Month _____	Number of days you spent hunting _____	
Geographic coordinates _____ e.g. 49° 15'49.90 N 122° 41' 13.29 W			
Species	How many?	Species	How many?
HUNTING LOCATION			
_____			
Year _____	Month _____	Number of days you spent hunting _____	
Geographic coordinates _____ e.g. 49° 15'49.90 N 122° 41' 13.29 W			
Species	How many?	Species	How many?

## APPENDIX 1

### *Example Harvest Survey and Instructions*

## Instructions

Please give your age, and whether or not you have a Migratory Game Bird Hunting permit.

Refer to the map for the name of the region where you hunt.

Fill in the month, day, and year that you filled out the form.

**The form is organized so that you can record the birds you harvested separately for each place and month that you hunted.** If you hunted in the same place in different months, fill out a separate block for each month. For each location and month, record the number of days you spent hunting.

Please describe where you hunted, with reference to approximate distance and direction from nearby towns or other landmarks, and mark the approximate locations on the map. Do not give descriptions of place names that are only locally known.

Please also give the geographic coordinates if you know them, either in UTM (e.g. 522770 E 5456836 N), decimal degrees (e.g. 49.263860 -122.687025), or degrees, minutes, and seconds (e.g. 49° 15'49.90 N 122° 41' 13.29 W). Give either the actual location where you harvested birds, or the approximate centre of the area where you hunted within a hunting location.

For each location and month, give the names (species) of the birds you harvested, and the number of each species of bird that you harvested. Please be specific with names, for example, write 'surf scoter' instead of 'scoter', as there are three different species of scoters in Canada.

**The information you provide is completely confidential and will only be used for environmental monitoring purposes.**

## APPENDIX 1

### Example Harvest Survey and Instructions

*Definitions of variables being asked in survey. It is important to include to minimize ambiguity and ensure respondents all understand the questions being asked in the same manner.*

Variable	Description	Example
Hunter ID	A unique identification number assigned to each individual hunter	GH29784
Hunter Age	The age of the hunter in years	
Permit	Whether the hunter holds a Migratory Game Bird Hunting Permit	Yes or no
Region	The name of the geographic area where birds were harvested. A map provided to hunters with regions clearly marked will ensure consistency in the data. The map can also be used to mark approximate hunting locations.	Kootenay BC, Interlake Manitoba
Date	The month and day that harvesting effort was recorded on the data sheet	
Location	The area where harvesting occurred, in reference to nearby towns, major rivers or other landmarks. Locations are distinct if, for example, they occur in different types of landscapes, or are separated by more than ~25 km.	Siwash Mountain, approximately 20 km southwest of Nelson, BC
Month	The month in which harvesting occurred at each hunting location	
Effort	The number of days spent in the field actively searching for birds for each hunting location and for each month	
Geographic coordinates	Geographic coordinates where birds were harvested - either the actual location, or, the approximate centre of multiple places where harvesting occurred within the same general area. Coordinates can be given in UTM, decimal degrees, or degrees, minutes or seconds.	UTM 522770 E 5456836 N Decimal degrees 49.263860 -122.687025 Degrees, minutes, and seconds 49° 15'49.90 N 122° 41' 13.29 W
Species	The common names of birds harvested at each hunting location and for each month	Surf scoter, Brant, Snow Goose, etc
Quantity	The number of birds of a particular species harvested at each hunting location and for each month	

# APPENDIX 2

## *Mapping Your Data*

You may have a data set from a harvest survey for which hunters provided geographic coordinates for where they hunted. This is the ideal. Alternatively, hunters may have written descriptions of where they hunted. In either case, you can generate maps of hunting locations very easily using either GoogleEarth, which is freely downloaded from the internet, or using ArcGIS, which is not free but can be downloaded from the internet for a 60 day free trial. Site licences can be purchased for about \$150 per year.

Let's begin with the less-than-ideal scenario. Hunters wrote down descriptions of where they hunted on the harvest survey questionnaire forms. Hopefully, hunters followed the instructions mailed along with the questionnaires and referenced places that you can find online using Google maps, or that you know from your own knowledge. As a back-up in case you're not able to pin point their hunting locations, hopefully they also marked their hunting locations on the map you provided.

Now you just need to generate geographic coordinates from these descriptions. This is easiest in GoogleEarth.

First let's overview **geographic coordinate systems**.

There are two geographic coordinate systems commonly used. You may be familiar **latitude** and **longitude**, measured in degrees, minutes, and seconds – one for the north location (latitude) and one for the west location (longitude). All latitude and longitude coordinates are in reference to the Prime Meridian over in England.

Example:

49°14'39.38"N

122°32'40.54"W

## APPENDIX 2

### Mapping Your Data



**Decimal degrees** is a different way to write latitude and longitude. For example, Latitude 49.356074 is Latitude 49°21'21.87"N. The 21 degrees and 21 seconds part of the latitude unit can be re-written as a proportion of 1, hence decimal 356074 (21 degrees of 60 degrees of a full circle).



## APPENDIX 2

### Mapping Your Data

The **Universal Transverse Mercator (UTM)** system is a completely different system than latitude and longitude. The system is based on a grid of the entire planet, each square or 'zone' is assigned a number and letter. Vancouver BC is in zone 10 U. Each UTM unit of 'easting' is one metre east to west, each unit of 'northing' is one metre north to south.

Example for Vancouver BC:

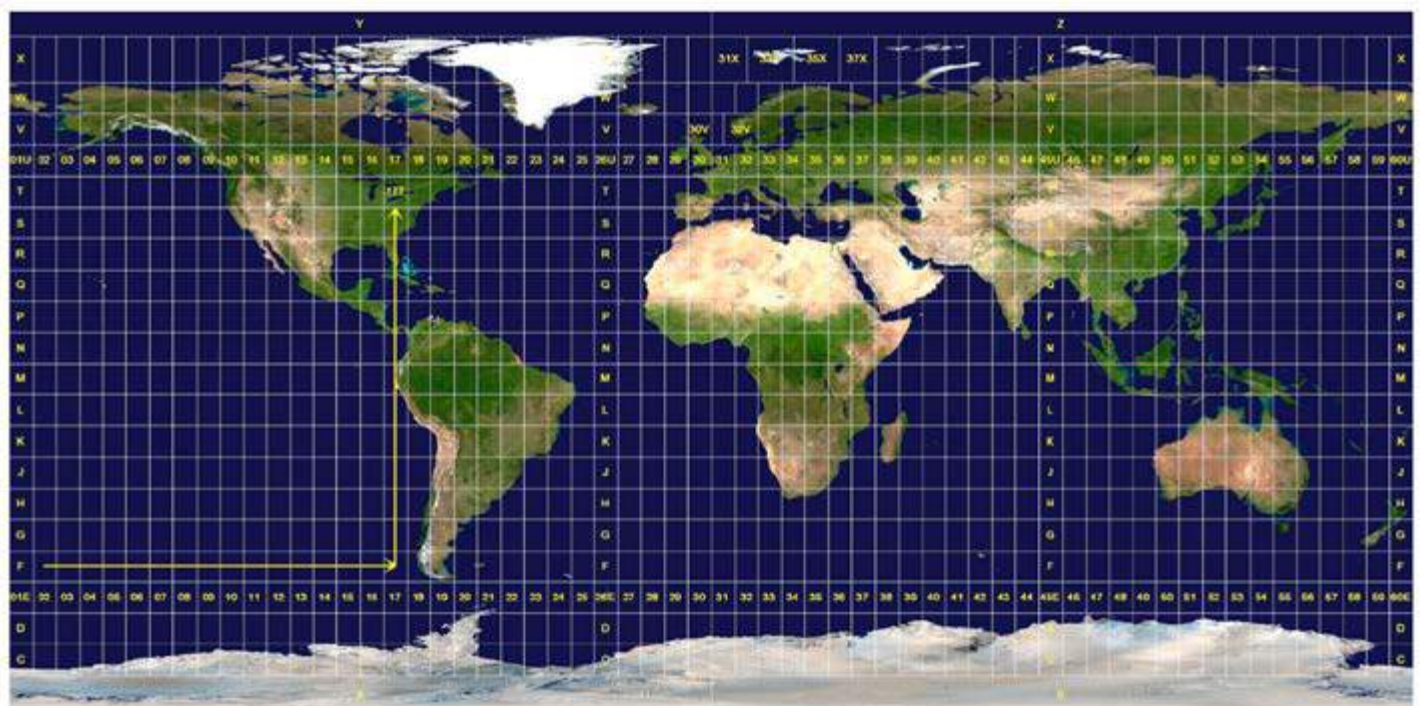
10 U

533340.53 m E

5454658.34 m N

The easting coordinates are the distance in metres from the centre of a zone, with the centre being arbitrarily assigned the value of 500,000. The northing coordinate is measured as distance from the equator. Thus, Vancouver is about 5400 km from the equator.

UTM Zones of the world



## APPENDIX 2

### Mapping Your Data

If there is not already a copy of GoogleEarth downloaded on your computer, download and install it from <http://www.google.com/earth>. Click on the icon to open it.

Note on the left hand side of the screen beside the map are two windows – **Places** and **Layers**. Under **Places**, right click on **My Places**, **Add**, **Folder**, and then title the folder **Harvest**. Click **OK**. You’ve just created a folder to store the hunting locations you are about to create. Make sure the Harvest folder is always highlighted as you work to ensure your work is saved in the right folder.



Click on the yellow push pin button (in the top right corner of the photo). A push pin appears on the centre of the map, with a blinking square around it. When the square is blinking, the pin can be moved wherever you want to move it, by clicking and dragging.

Based on hunter descriptions of their harvest locations, you can add pins to the map, one pin for each location. The ruler button will be handy for this, to measure out distances from hunting locations to points of reference, such as towns or waterbodies.

In the **New Placemark** window, after **Name:** you can add whatever labels you like. Perhaps you will create a map of hunter locations by Hunter ID. Thus, you would pin point a hunting location on the map based on a description, and then label the pin by the hunter’s ID. When you press OK, you can no longer move the pin. To make further edits to the pin, you need to right click the pin’s label in the harvest folder, in the Places window, then click **Properties**. The **Edit Placemark** window will open, and the blinking square will once again appear around the pin, signalling that it can be moved. You can change

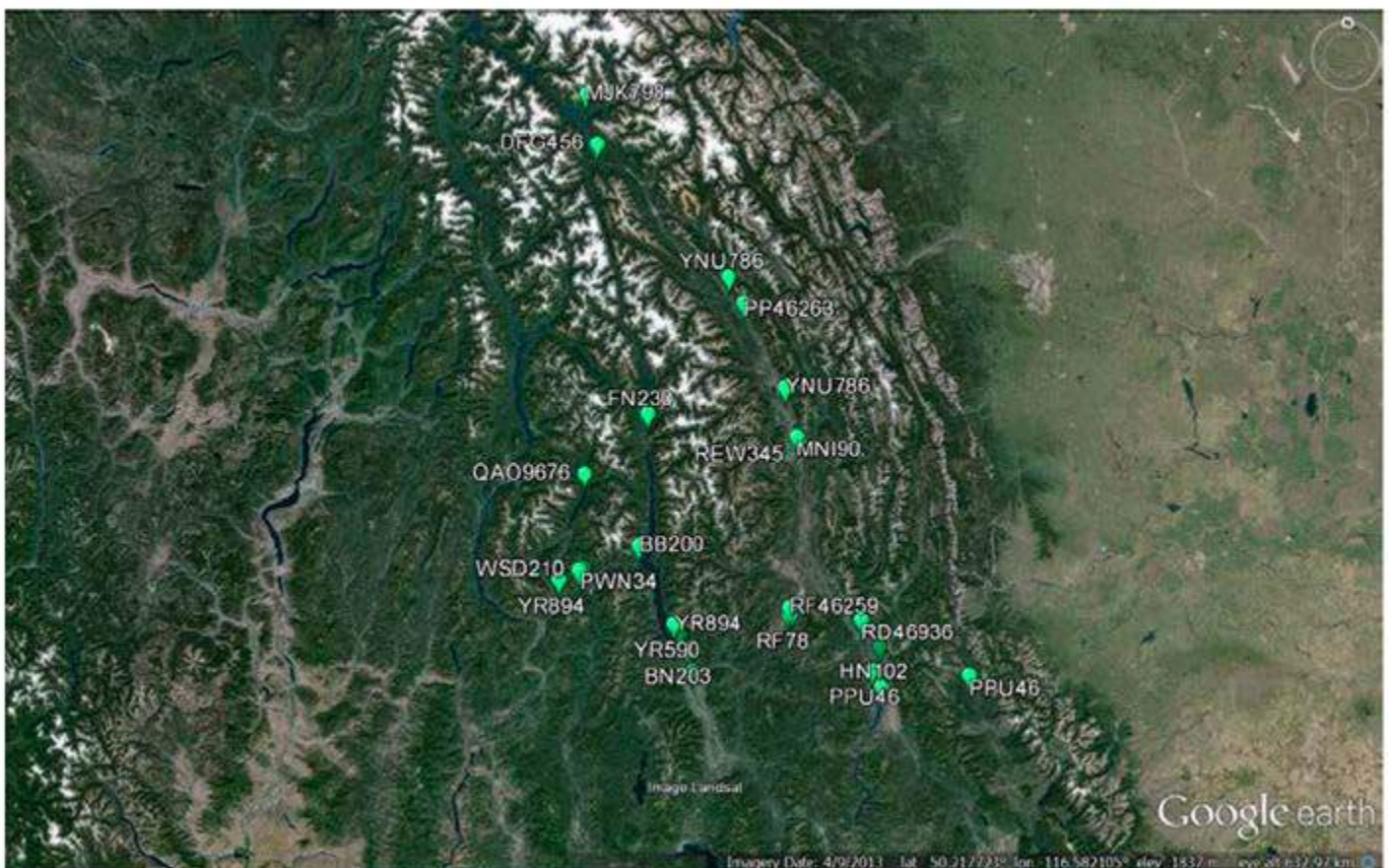
## APPENDIX 2

### Mapping Your Data

the pin colour on the Style, Colour tab, and you can change the pin symbol by pressing the pin icon next to the **Name:** window.

Now suppose you have entered all hunting locations, and labelled these according to Hunter ID. You can now easily produce a map in the format of a digital picture, which can be inserted into a report, emailed, or printed. Under **File, Save, Save Image**, give your map a title and save it on your computer. You can now open the file and view your map as you would a photo on your computer.

An example map produced using GoogleEarth is shown below – these map hunter locations for the Kootenay region in the example data set. From a quick glance, it's very easy to see that hunting locations are primarily in wetland valley bottoms, unsurprisingly.



## APPENDIX 2

### Mapping Your Data

Another key thing to know when using Google Earth is how to set the units. Under **Tools** on the main menu at the top of the screen, click **Options**. You'll see options for showing the location in UTM versus degrees, minutes, seconds, and also the units of measurement. You can set it to metres versus feet.

Not only do you now have a map, you also have a file containing the approximate geographic coordinates of hunting locations. You can export these from GoogleEarth into your working data set in excel. Making sure the harvest folder is highlighted, click on **File, Save, Save Place As...**, change **Save as type:** .kmz to .kml, then click **Save**.

Open the .kml file in excel (ignore any warnings). The file isn't exactly friendly-looking, but you'll see that it does contain three columns of interest (ignore the rest). Copy the hunter ID and latitude and longitude columns, and then **Paste Special, Values**, into the working data. Match up the hunter IDs and now you have geographic coordinates for hunting locations as part of your working data.

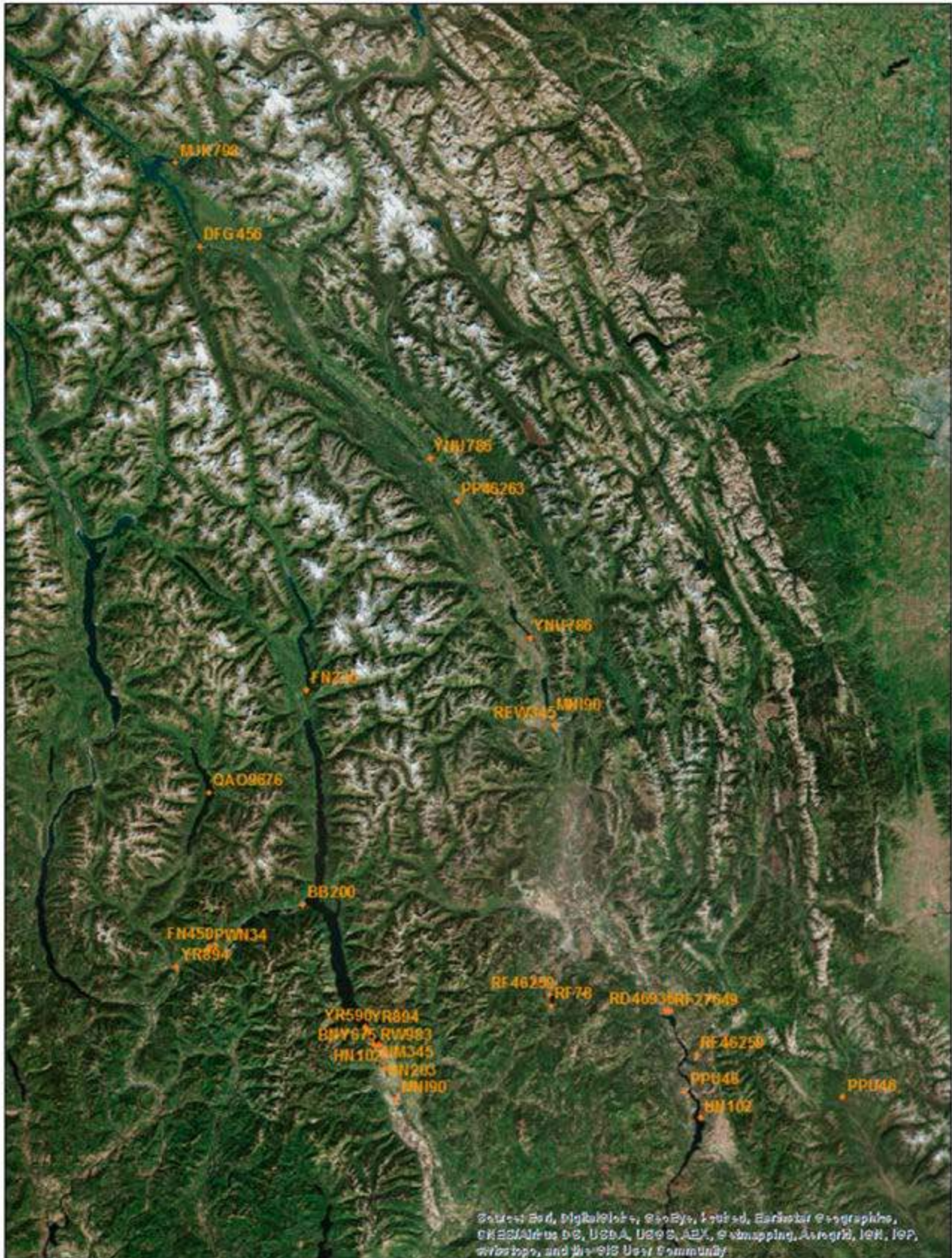
Now let's consider our options given that hunters provided geographic coordinates for hunting locations, and you have entered these into the data spreadsheet. Unfortunately it isn't possible to upload an excel file of coordinates into GoogleEarth. You can manually enter each coordinate using the push pins. This time instead of placing the pin on the map, you already know the coordinates and you just need to enter these into the coordinate windows on the **New Placement** window that pops up when you press the pin icon. When you press OK after entering the coordinates manually, the pin will be placed at that location.

A faster alternative is to use ArcGIS to convert an excel file to a kml file, and then open the kml file with GoogleEarth. However, if you are able to do this in ArcGIS, then you can make a map in ArcGIS, which will be of higher quality because the mapping options are much broader. Compare the map above in which all the hunter IDs are not shown, to the same map below made using ArcGIS.

Instructions for importing and mapping data in ArcGIS are found in Appendix 3 [below](#).

## APPENDIX 2

### Mapping Your Data



# APPENDIX 3

## Creating a Map Using ArcGIS

### GETTING STARTED

When you open ArcMap, the first thing you'll notice is that it's asking you whether you want to open an existing map or create a new one. Let's create a new one.

#### New Maps

##### Blank Map

Now you're staring at a blank screen.

The first thing to notice is the **Table of Contents dock**. You should see

#### Layers

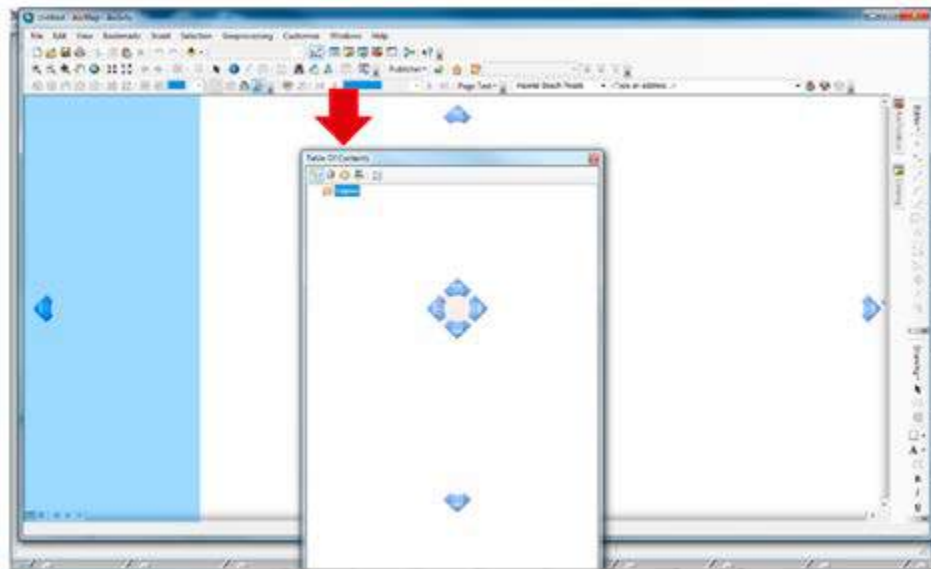
This is where all the action is. The Layers icon is called a 'data frame' and it functions like a folder in Windows Explorer, holding all the layers in it, like a folder holds different files.

GIS analysis and mapping relies on *layers* of data. A layer of data is simply a data set of information, called *attributes*, with spatial reference. In other words, a layer is a table of information with varying numbers of columns for attributes, and one column for x coordinates and one column of y coordinates. Because this information has a spatial reference, it can be shown visually on a map.

Just above layers, you'll see five buttons. If you hover the mouse over each button, dialogue windows pop up explaining what each button does. All these buttons do is change the way the layers in your map are listed in the table of contents window. Usually, you want the first button – list by drawing order. Under this button, the layers are drawn on top of each other as they appear in the list.

## APPENDIX 3

### Creating a Map Using ArcGIS



## SETTING UP YOUR MAP

In ArcGIS, one of the first things you do when creating a new map is to set the coordinate system.

Right click on

**Layers**

Then scroll to

**Properties**

**Coordinate System**

Click on

**Geographic Coordinate Systems**

**North America**

**NAD 1983 (2011)**

They updated this datum in 2011, I think to account for shifting of the tectonic plates!

The scroll down to get into the Projected options

## APPENDIX 3

### *Creating a Map Using ArcGIS*

---

#### Projected Coordinate Systems

##### UTM

Then you can choose either the NAD 1983 or WGS 1984 datums.

Before you make a further selection, you need to know the UTM Zone for the map you want to make.

Choose **WGS 1984**

then your UTM zone

Now we need to set the units that our map will display in. Still under **Data Frame Properties**

#### General

#### Display

##### UTM

Now we're ready to start creating a map.

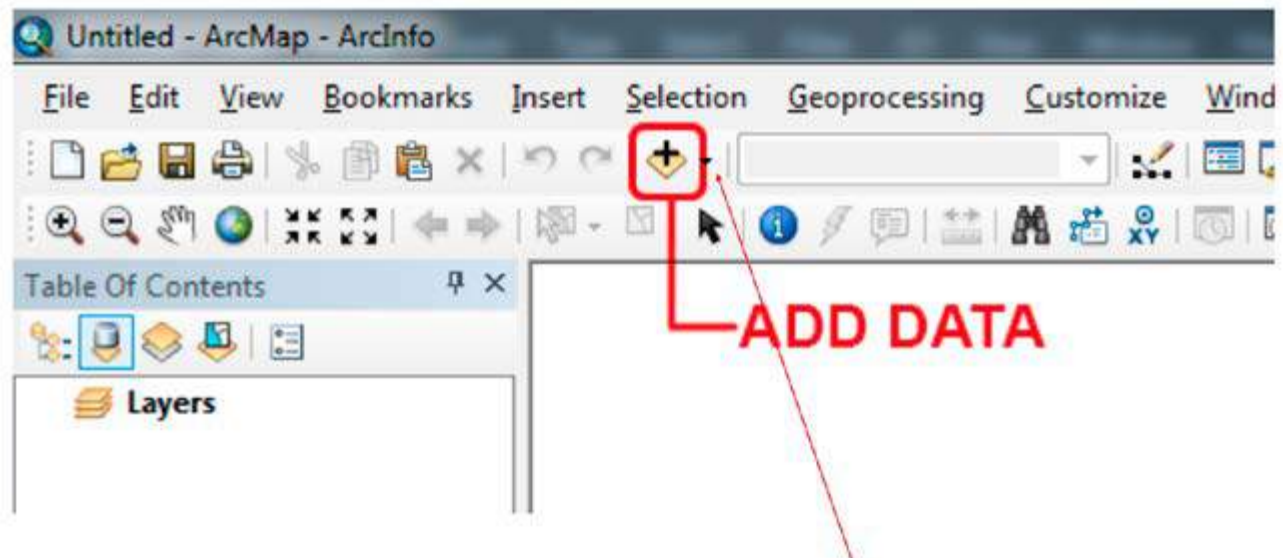
First let's start with a basemap. In the future, you will have your own basemap already created. But for now, let's just add some satellite imagery.

One of the most important buttons, is the Add Data button in the top menu.



## APPENDIX 3

### Creating a Map Using ArcGIS



Notice the little tiny down arrow next to it.

#### Add Basemap

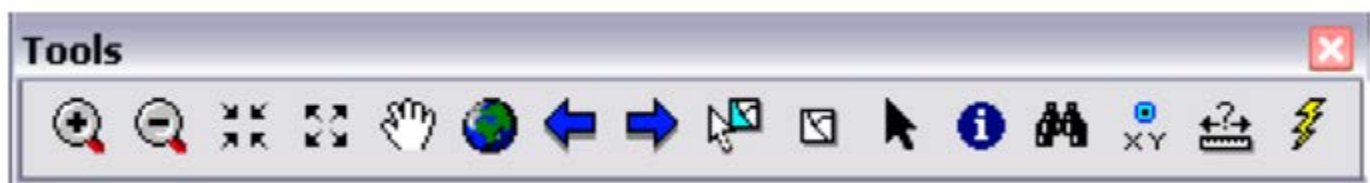
#### Imagery

Now we actually have something to work with.

Before we start learning how to add things to the map, let's learn how to move around a map.

## NAVIGATING AROUND MAPS

Find the Tools toolbar




## APPENDIX 3

### Creating a Map Using ArcGIS

## Panning

Some of these will be familiar to you from GoogleEarth, but they work differently in ArcGIS. Click on the pan tool (the white hand) to activate it. This works by clicking on an area where you want to centre the map. Once you left click, the hand 'grabs' that area. And then with the mouse clicker held down (left click and hold), you can move the map around as you want.

## Zooming

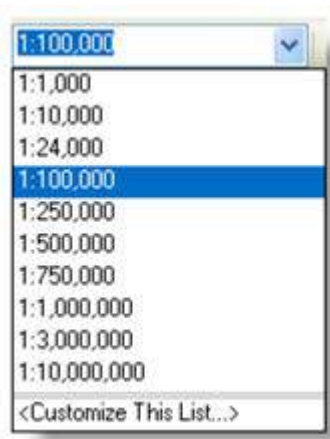
Now that we have the map centred on an area, we can zoom in by clicking on the zoom in tool to activate it. 

Once it's activate, the cursor becomes the zoom in tool. You can keep clicking on the area you want to zoom to,

OR

An easy way to navigate is to left click and hold on the top left corner of the map extent you want – you'll notice a rectangle is outlined depending on where you move the mouse. Then when you let go of the mouse, the map will zoom to the extent of that rectangle. Saves a lot of time!


You can also move in and out using the set scale menu, next to the add data button. The bigger the number, the more zoomed out the map.




## APPENDIX 3

### Creating a Map Using ArcGIS

## Refresh or Reload

A few of the buttons can be helpful for refreshing the map. Maybe you zoomed in too far somewhere and got lost. You can press the Full Extent button  and it will move to the fullest extent of whatever layers are in the data frame.


Press it and you'll see you're back to the whole world again. You can reverse that by using the  Back button. It'll take you back to the extent you were before the last button pressed.

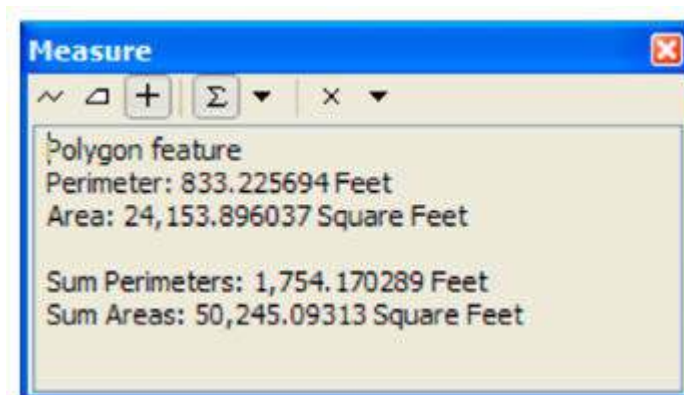
One other button that's important for refreshing your map is helpful for when Arc stops thinking and you get stuck with a half drawn map.

Right at the bottom of the map, you'll notice very tiny buttons.



If you ever get stuck with a half drawn map, if you look under those tiny buttons and it says 'cancelled', it means ArcGIS got overwhelmed and cancelled the drawing of the map. Pressing the Refresh button will force the program to draw the map again, this time without cancelling.

The other tools we will learn later on. For now, just note the measurement tool.  Same as in GoogleEarth, you can measure distances on your map, only in ArcGIS we have more options.



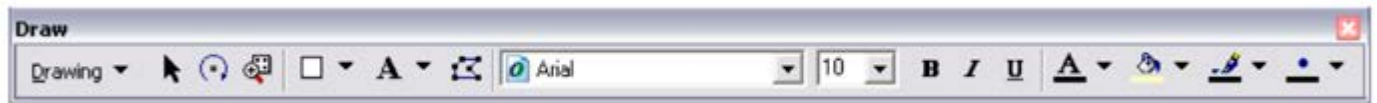
## APPENDIX 3

### Creating a Map Using ArcGIS

The squiggle lets us measure one or several line segments, the polygon with give us total area as we create a shape with by left clicking. Use the down arrow to select the units of measurement.

## DRAWING ON MAPS

The Drawing toolbar is another toolbar that we commonly use.



f it isn't showing in your program, then under


### Customize Toolbars

check

### Draw

and it'll show up on your top menu options. Take note of the zillion other toolbars you could add. Note the ones that are checked, and then compare that with what you see at the top of your screen. Note also that you can actually move these around however you want them.

The drawing toolbar is usually used most often when you're just about finished with your map, and you want to add some text to it.

The key to not being frustrated with the drawing options is to remember to activate the **Select Elements**  button (the thick black arrow). Once you create text, you then might want to move the text around. You'll need to activate the Select Elements button to do that. Instead of cutting and pasting into an appendix what each drawing button means (as I did for the Tools toolbar), you can simply access the functions on the ESRI website here [http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?topicname=draw\\_toolbar](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?topicname=draw_toolbar)

## APPENDIX 3


### Creating a Map Using ArcGIS

(they have nice explanations of all the toolbars on their website)

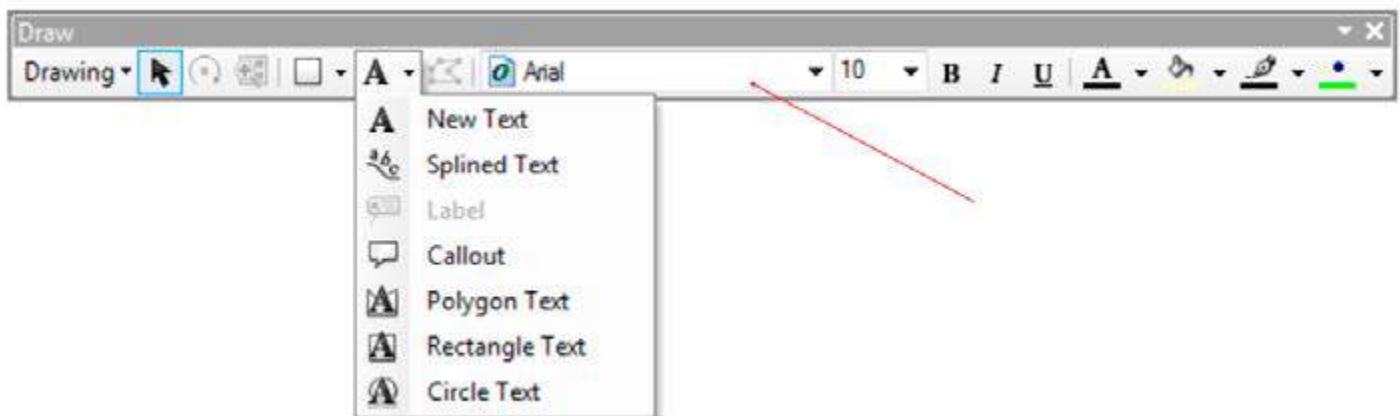
So let's add text





then left click on the map where you want to add the text

Let's say it's too small and the wrong colour of text. To change the size, make sure the text box is highlighted – it should be a bright colour. If it isn't, then use the Select Element tool  and click on the text.

Just like in Word, you can change the font type, the size, and the colour simply by pressing the appropriate buttons. The red arrow is pointing to where you change the font, and next to it is the size of the text. Etc...




You can also left click and hold on the text box and move it around where ever you want.

Now let's say you want to draw something on your map. You have lots of options under the polygon button. But first note the little downward arrow next to the polygon button . Clicking on that lets you see all the options. Choose **freehand**. Now choose **circle**. Now let's say you want to change the colour of the freehand. You have to move the cursor back to the freehand drawing you made and select it. Otherwise you can't edit it. With a light coloured box showing around the drawing (meaning it's been selected), now you can change the colour of the lines using the  button.

## APPENDIX 3

### Creating a Map Using ArcGIS



Now you want to change the colour of the circle. You need to use the  fill button. But first you have to select the circle.

You can access the menus to change the properties of any drawing object simply by right clicking when the cursor is hovering some part of the object. Scroll down to **Properties**, and it'll open a box with lots of options for changing things like the thickness of drawn lines.

## CREATING AND SAVING A MAP

Now we're ready to create and then save a map.

First, you need to introduce yourself to the two map views. One is called the Data View, and the other is called the Layout View. You can switch between the two in two ways.

View at the top main menu

or at the bottom of the map next to the Refresh button ...



... you'll see those teensy tiny buttons again. If you press on the second one to the right, it'll switch the view to Layout View.

You can see that you were working in Data View, which is normally the view we use in ArcGIS. Then, when we're ready to actually create a map, we switch to Layout View, so we can actually get a sense of

## APPENDIX 3

### Creating a Map Using ArcGIS

what our map will look like once it's printed out.

Once this view is selected, you should see a new toolbar at the top menu. If not, under

#### Customize

##### Toolbars

make sure

**Layout** is ticked

Now you've got different zoom and pan buttons for moving around your map. These are different than the buttons you used in the Data View. Check out how.

The main things you'll need when finalizing your map are on the main top menu under

#### Insert

It's almost always a good idea to insert a scale bar and a north arrow. Under each of those are lots of options. The main one under scale is to change the units to something that you like, usually metric. Once you select **Scale Bar**, under **Properties, Scale and Units** tab, you can change the units.

You can change the colour of the text on the scale and of the scale bar line under the **Format** tab. To change the colour of the ticks (vertical lines) on the scale bar, under **Numbers and Marks**, in the **Marks** section, you'll see **Division Height** and **Subdivision Height**. To the right of both, is **Symbol**, and the Symbol Selector menu opens up with options for colour and width.

In addition to our north arrow and scale bar, let's add some text to our map, then save it.

The document we've created in ArcGIS is a GIS document, from which we can make endless maps. This is very different from this one map that we're going to save from this GIS document.

When you save a GIS document, the file extension is .mxd, which is an ArcGIS document. When you export a map, you save it as a picture file – and there are a number to choose from. The best is JPEG .jpg

## APPENDIX 3

### *Creating a Map Using ArcGIS*

– these can easily be inserted into Word documents as pictures.

Under

**File**

**Export Map**

choose where you want to save your map, name it e.g. map.jpg, then

**Save**

Now open up a word document

Under

**Insert**

**Pictures**

browse for your map.jpg, click on it, then it'll appear in your word document. This is an easy way to insert nice maps into reports.

## ADDING POINTS

One the best things about ArcGIS is a very easy ability to make a map showing the locations of things, for example, hunting locations. Of course, that means that you have to have some coordinate data. You can create coordinate data in Google Earth and then export it to ArcGIS, if you want a higher quality map than can be produced in Google Earth.

### From a .KML File

From the .kml file that you saved in Google Earth (see [above](#)), you first need to convert the .kml file to a layer file.



## APPENDIX 3

### *Creating a Map Using ArcGIS*

#### Geoprocessing

#### Arc Toolbox

#### Conversion Tools

#### From KML

#### KML to Layer

Browse for your kml file on your computer in **Input KML File**, then choose where you want to save it on your computer, then click ok.

Now you have your GPS waypoints saved as a layer. You'll see that ArcGIS saved the layer as a header layer, which doesn't contain the spatial coordinates, and a points layer which does.

Now we want to save our layer file as a shapefile. Right click on the Points layer, then **Data, Export Data, use the same coordinate system as the data frame**, and under **Output feature class**, you can change the name of your shapefile to whatever you want, and save it to wherever you want. Once you click on ok, ArcGIS will then ask if you want to add the shapefile to the map. Click ok. And now you can delete your layer file.

You've just created your first shapefile, which is the basis of everything in ArcGIS.

One thing you might want to do right away is add the x,y coordinates of your data to the shapefile's Attribute Table, which is where all the information about the layer is stored.

Right click on the shapefile's name, then **Open Attribute Table**. You'll see a bunch of pretty useless columns, the only one you're interested in is the Name column. The rest are basically empty columns, which you can populate with information, like the x,y coordinates. Right click on a column, then **Calculate Geometry**, under **Property** you can change between x and y, then Use coordinate system of the data frame, then choose your units, like degrees, minutes, seconds.

## APPENDIX 3

### *Creating a Map Using ArcGIS*

## From an Excel File

You may also have an excel file of coordinates, if for example, hunters provided coordinates of hunting locations on the harvest survey data sheet. This is very straightforward. But you must first save the excel file containing the coordinates as a .csv file (see [above](#)). Click on **File, Add Data, Add xy Data**, browse to your .csv file, and then in **X Field**, choose the column in your file for the east to west coordinate, and in the **Y Field**, choose the column for the north to south coordinate. Click ok, and each hunting location will now display on the map.

## LABELS AND SYMBOLS

To change the symbology (e.g. do you want blue or green dots, how big, etc), simply left click on the symbol for the layer, and it brings up the symbol selector menu.

To change the labelling (and a bunch of other stuff), right click on the title of the layer, and scroll down click on **properties**. You'll see a bunch of tabs, click on **labels**. To label each record in your layer (which is each row in your attribute table, for example, each waypoint), select **Label features in this layer**, and then there a bunch of options for changing the way labels show.

# APPENDIX 4

## Glossary

---

**Accuracy** The degree of closeness of sample estimates to the true value.

**Biased** Refers to systematic inaccuracy – that is, biased data is sample data that does not represent the population because of certain attributes of the sample.

**Categorical variable** Contains categories of information (ex. season).

**Clustering** Measure of the central tendency of our sample of average.

**Confidence interval** Measure of certainty in our estimation of the population mean.

**Continuous variable** A numerical variable that varies from the lowest possible number (0) to the highest possible number.

**Data management** The combined actions of storing, organizing, and summarizing data.

**Degrees of freedom** Your sample size minus the number of population parameters you're trying to estimate.

**Frequentist statistics** The branch of statistics that is based on probability being measured with reference to the relative frequency of observations, in other words, from sampling distributions.

**Median** The number that divides the data set in half, with the observations ordered from smallest to largest.

**Mode** The number that appears most often in the data set.

**Non-response bias** The tendency of not reporting when nothing is harvested.

**Observations** Measurements and descriptions of the world that we collect to form our data to answer our research questions.

## APPENDIX 4

### Glossary

---

**Precision** The degree of closeness of sample estimates to one another.

**Prestige bias** The tendency of reporting a higher number than is true.

**Probability** The likelihood of getting a certain study result, which we can estimate based on a sampling distribution.

**Quantiles** Values associated with probabilities.

**Range** The difference between the minimum number and maximum number.

**Raw data** Data that has not been processed in any way; this data are simply a digital representation of what was collected in written format on data sheets.

**Sample size** Refers to the number of sample units used to estimate truth.

**Sampling distribution** Counts per interval.

**Standard deviation** The square root of the variance.

**Standard error** Of a sample statistic, is the standard deviation of the sampling distribution of the statistic.

**Statistic** A quantified characteristic of a sample taken from a population.

**Strata** Strata are homogenous sub-groups of a population.

**Variability** The spread of differences within a population.

**Variable** A characteristic of the world of which measurements have been taken.

**Variance** The average deviation from the mean.

## APPENDIX 4

### Glossary

---

**Working data** Data that has been summarized to the point that the data is ready for summary and analysis.

# B I B L I O G R A P H Y

---

**Crawley, M. J. 2012. *The R book*. Wiley, New York.**

This is a good book for getting started with R. It starts at the very beginning – installing R - and works through to some moderately complicated statistical analyses. It is best as a recipe book for learning the basics of R – especially how to manage data once you’ve imported it into R.

**Hilbe, J.M. 2014. *Modeling Count Data*. Cambridge University Press, New York.**

This book is a relatively easy read, and provides a thorough overview of Poisson versus negative binomial modelling. There are, however, mistakes, typos, and gaps throughout the book that make it hard to follow in places. R code is provided to work through all the examples, which are particularly useful for learning how to test for over-dispersion, how to deal with it, and the consequences of not dealing with it. A go-to handbook for count data despite it needing a thorough edit.

**Quinn, G. P., and M. J. Keough. 2002. *Experimental Design and Data Analysis for Biologists*. Cambridge University Press, New York.**

This is probably the best overall introductory statistics book for ecologists. The authors are careful to explain statistics in plain speak, and use many examples to describe in detail what they’re trying to convey. Unfortunately generalized linear modelling is only briefly introduced, so the book is best used a reference on basic statistical terms and tests.

**Zuur, A. F., E. N. Ieno, N. J. Walker, A. A. Saveliev, and G. M. Smith. 2009. *Mixed Effects Models and Extensions in Ecology with R*. Springer, New York.**

This book picks up where Quinn and Keough leave off. The authors use a similar informal style, explaining concepts in plain speak and avoiding dense mathematical explanations. Generalized linear modelling is explained in detail, using examples of typical messy and hard-to-analyze ecological data. And excellent, easy to read book. It may be worthwhile to review their introductory book first (“Analyzing Ecological Data”).

# I N D E X

- A**  
accuracy 35  
AIC 159  
alpha level 129
- B**  
bar charts 46  
bias 8  
boot-strapping 184
- C**  
categorical predictors 166  
categorical variable 49  
confidence intervals 96  
continuous variable 49
- D**  
data collection protocol 15  
degrees of freedom 41  
deviance 161  
deviates 41  
dispersion parameter 147  
dummy variables 166
- F**  
frequentist statistics 114
- G**  
geographic coordinate systems 189
- H**  
histogram 87
- I**  
independence 9  
interaction effect 164
- L**  
latitude and longitude 189  
least squares 148  
log likelihood 159
- M**  
maximum likelihood estimation 148  
model selection 164
- N**  
negative binomial distribution 147  
non-response bias 12  
normal distribution 89
- O**  
observations 36  
offset 156  
over-dispersion 147
- P**  
Poisson distribution 145  
population 7  
population parameters 83  
precision 39  
predictor variables 32  
prestige bias 51  
principle of parsimony 160  
probability 96  
pseudo R squared 161  
P value 126
- Q**  
quadratic model 157  
quantiles 110
- R**  
random sample 11  
residual error 136  
response variables 32  
R squared 140
- S**  
sample 7  
sample size 7  
sample unit 7  
sampling distribution 87  
scatter plot 50

# INDEX

---

slope coefficient 133  
standard deviation 43  
standard error of the mean 90  
standardization 29  
statistic 36  
study design 32

## T

t distribution 115

## U

uncertainty 83  
Universal Transverse Mercator (UTM) 191

## V

variance 41



Additional information can be obtained at:

Environment and Climate Change Canada

Public Inquiries Centre

7th Floor, Fontaine Building

200 Sacré-Coeur Boulevard

Gatineau QC K1A 0H3

Telephone: 1-800-668-6767 (in Canada only) or 819-997-2800

Email: [ec.enviroinfo.ec@canada.ca](mailto:ec.enviroinfo.ec@canada.ca)