

Haan & Godley - R Lab Manual

Dave Armstrong
University of Western Ontario
Department of Political Science
e: dave.armstrong@uwo.ca

Much of what below comes from *An Introduction to Statistics for Canadian Social Scientists, 3rd ed.* by Michael Haan and Jenny Godley (Oxford University Press, 2017). I have added the R-based content.

Lab #1: Introduction to R

The focus of this lab is to introduce you to R and the R Commander (a graphical user interface to R). To use R to analyze data, you will need to become familiar with the technical components of this software package. This lab will help familiarize you with the R software, including how to access data files, the various base components and how to define new variables and how to enter data.

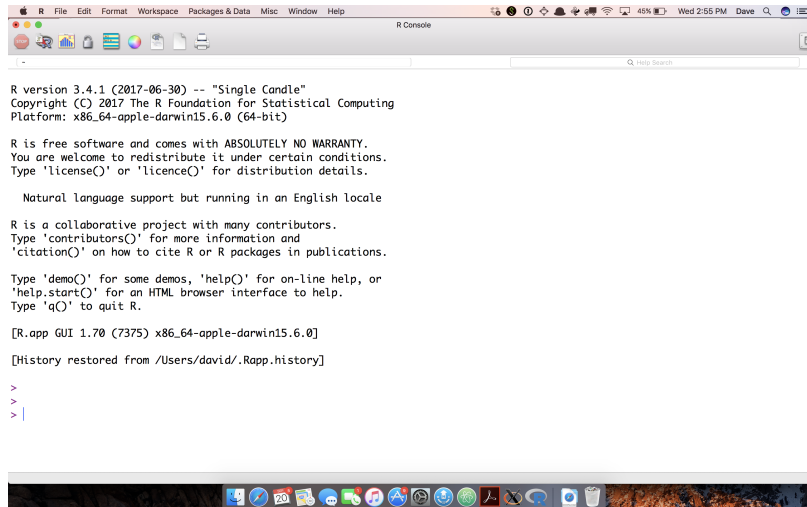
Since R is open-source, you can freely download it and all of its components on your computer. It is also available in the SSTS labs.

Defining and Downloading R

R is an open source statistical computing environment that is becoming increasingly popular among social scientists. It works by taking a series of commands and applying those commands to the data you specify. We will use the R commander initially to reduce the steepness of the learning curve for R which is notoriously steep.

To use R, you will have to download it. Below are some instructions for downloading and installing R and the components you'll need for now.

- Download and install R v 3.4.1 from here: <https://cran.r-project.org>.
- Open R, you should see something that looks like this:



- In R, type the following (in fact, you should be able to copy the line from the handout and paste it into R):

```
install.packages("Rcmdr", dependencies=TRUE)
```

This will pop up a dialog box of CRAN Mirror sites from which you can choose. Pick site number 0 (the first one on the list) and click “OK”. You will get a set of messages that look something like the one below. You might have more packages to install, so if your message is longer, that’s fine.

```
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'RcmdrMisc'

trying URL 'https://cloud.r-project.org/bin/macosx/el-capitan/contrib/3.4/RcmdrMisc_1.0-6.tgz'
Content type 'application/x-gzip' length 169471 bytes (165 KB)
=====
downloaded 165 KB

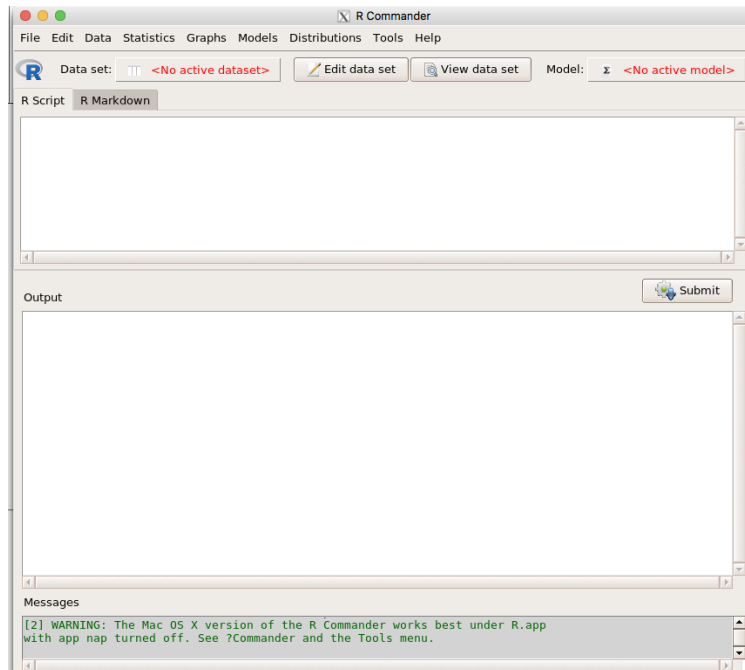
trying URL 'https://cloud.r-project.org/bin/macosx/el-capitan/contrib/3.4/Rcmdr_2.4-0.tgz'
Content type 'application/x-gzip' length 4976885 bytes (4.7 MB)
=====
downloaded 4.7 MB

The downloaded binary packages are in
  /var/folders/fp/vk4vpbv959146ckgyb191t68000gp/T//RtmpUuf4xA/downloaded_packages
>
|
```

- Now, in R, type the following and hit enter:

```
library("Rcmdr")
```

You should get a window that looks like this:



You'll notice that R Commander has both an input and output window. The input window is where you (or more likely right now, R) will type the commands. The output window shows numerical output produced by the commands you issue either by typing in the input window, or by choosing menu options.

Saving Your Files

In the R Commander, you can save three different kinds of files through the `file` menu. You can choose "Save script" which will save the contents of the R Script (input) window. You can choose "Save output" which will save the contents of the output window to a text file. You can also choose "Save R workspace" which will save all of the data, models and other objects that you produce in your R session. It is a good idea to keep track of all of your inputs in the script window and save them for every R session.

Lab #2: Identifying Types of Variables: Levels of Measurement

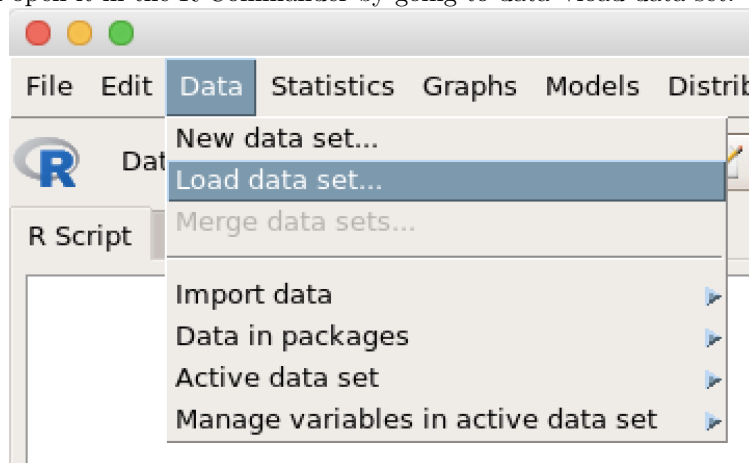
The focus of this lab is to introduce you to the four different levels of variable measurement, how to identify different types of variables within R and how different levels of measurement are coded and organized within R. This material corresponds with the material presented in Chapter 2.

Understanding Levels of Measurement

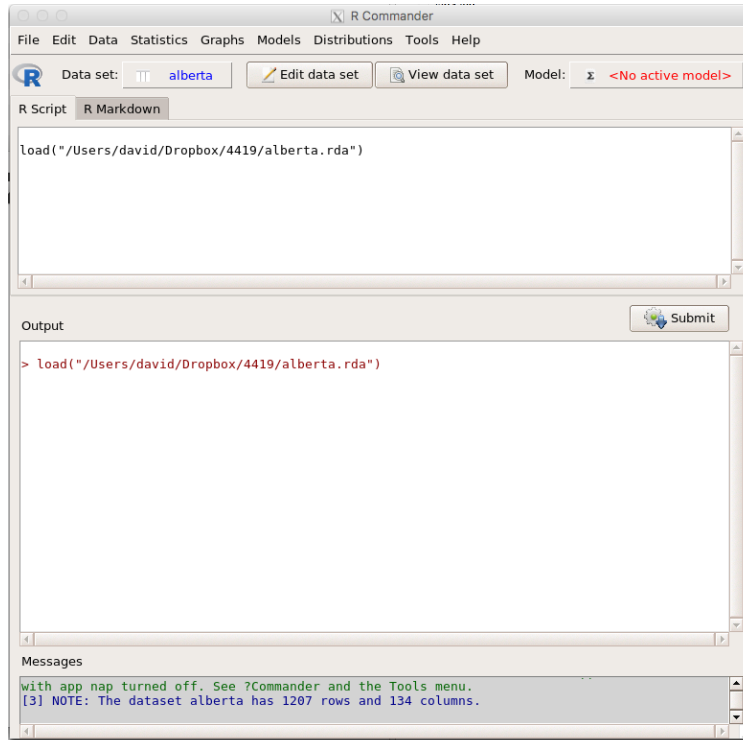
Variables are measured at four different levels: nominal, ordinal, interval and ratio. Each of these levels has unique characteristics that define them. For nominal data, numeric values are typically used for identification purposes. It is not possible to rank the response categories and there is not a quantifiable difference between categories. For ordinal data, the numeric values can signify an inherent ordering because you can rank the response categories, but you cannot measure the distance between those categories. For interval data, the data can be organized into an order that can be added or subtracted, but (theoretically) not multiplied or divided, because there is no true zero. Ratio data are similar to interval data except they have a true zero value.

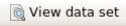
There are many ways to retrieve information about a variable's level of measurement in R. The primary distinction in R is between qualitative variables (nominal and ordinal) and quantitative variables (interval and ratio). The former are generally called *factors* in R. They can be ordered, but generally are not as there is no inherent difference in the way those variables are treated in models relative to unordered factors.

To open the Alberta survey, download the file `alberta.rda` from the course website. The codebook is also there. The codebook tells you what each variable means and how it's coded. Once the dataset has been downloaded, you can open it in the R Commander by going to data→load data set.



This will bring up a dialog box that will allow you to browse to the data. Click on the data file and click “OK”, then you should see something like this (though presumably with a different path to the file).

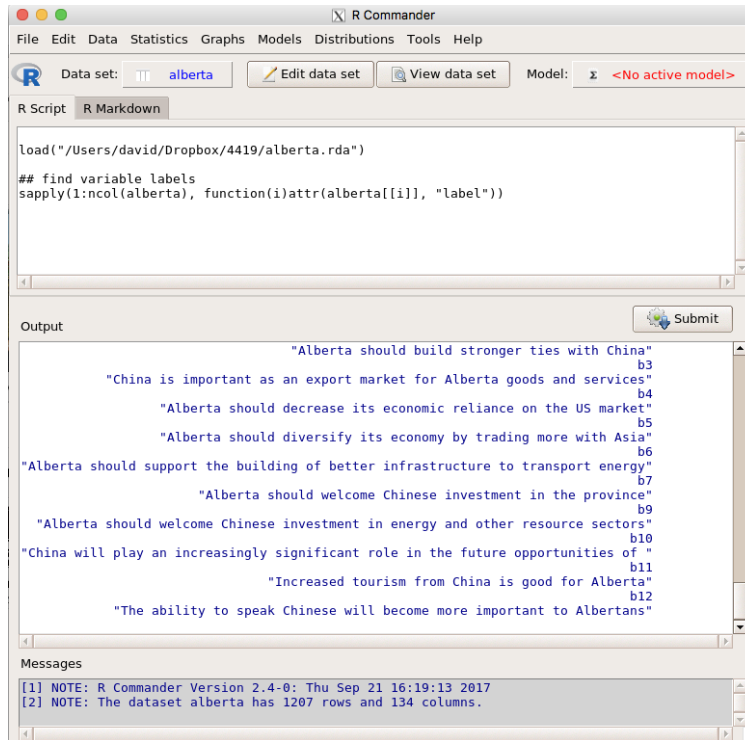


Clicking on the  will show a spreadsheet-style look at the dataset.

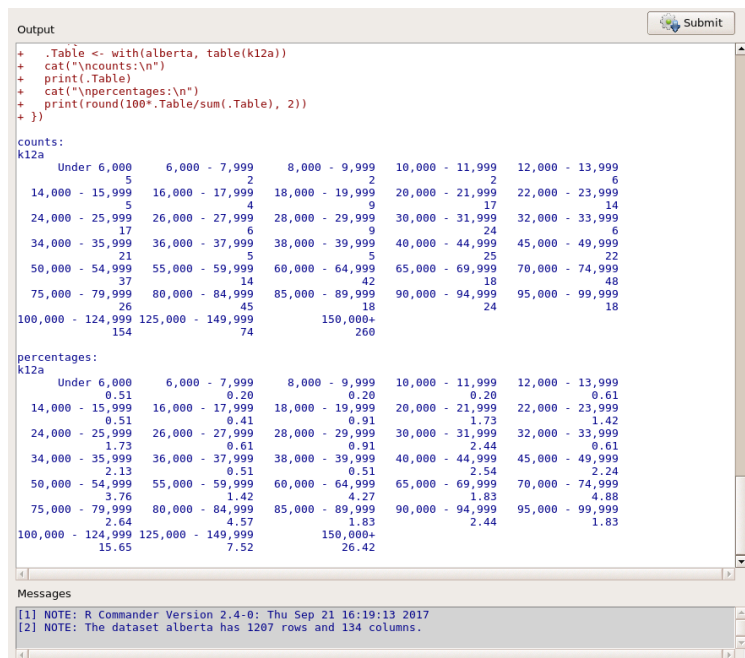
When you have opened the file, take a moment to scroll through the list of variables and observations in the Data Browser. You will notice that there are many variables (134 to be exact) contained within the dataset. If you scroll down the observations, you will see that there are 1207 respondents. Return to the main screen by closing the data browser. You may have noticed that the variable names were not all that intuitive (this happens frequently in survey data you download from the internet). The variable labels do provide some insight, but in R, these are less easily accessible than they might be in other programs. If you type what is below, all of the variable labels will get listed along with all of the variable names:

```
sapply(1:ncol(alberta), function(i)attr(alberta[[i]], "label"))
```

After you're done typing in the command, hit the  and you should see something like the following output:

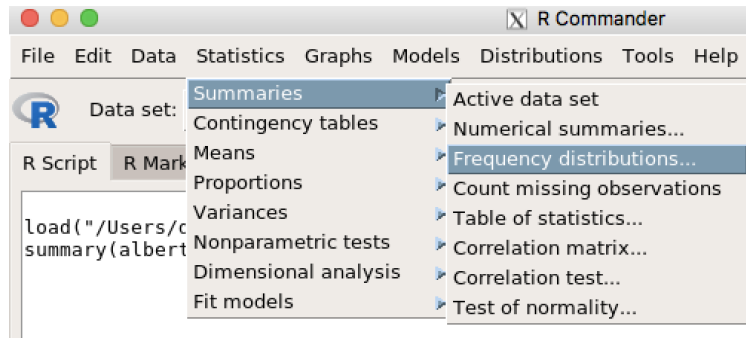



For example, variable `k12a` has the label “Total household income for the past year before taxes and deductions.” Given that this is an income variable, a person can feasibly have no income, you might be inclined to think it’s a ratio-level variable that records the amount of money a person made. However, looking at the frequency distribution for the variable, a different story emerges.

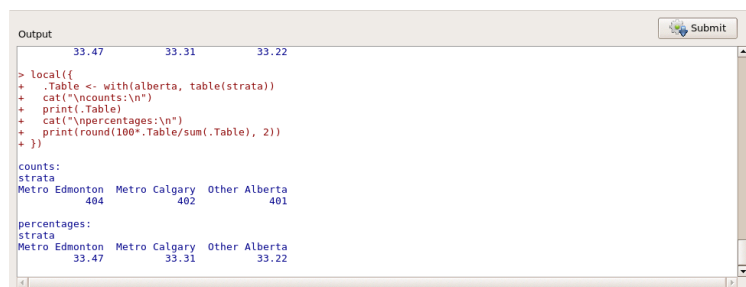


As we can see, the response categories for this variable are grouped into discrete categories. Therefore, this variable is actually an ordinal level variable because you can rank the categories, but not calculate the difference between them.

Let's look at one more example. Find the variable named **strata** in the dataset. Without going to the codebook, we do not know how this question was answered by respondents. The easiest way to see the values is to make a frequency distribution. To do this, choose statistics→summaries→frequency distributions.



This will pop up another dialog box, from that, scroll down to **strata**, the last variable in the list. Click the  button and you should get the following:



Lab #3: Univariate Statistics

The focus of this lab is to begin to introduce you to analysis with one variable. Generating frequencies is a basic procedure used to obtain a summary of a variable by looking at the number of cases associated with each value of the variable. This material corresponds with the material presented in Chapter 3.

Learning Objectives:

The following lab is directed at helping you understand ways of studying the characteristics of data. Specifically, this lab assignment challenges you to clarify your understanding of:

1. How to generate and interpret frequency distributions
2. Data presentation
3. The connection between data presentation and levels of measurement.

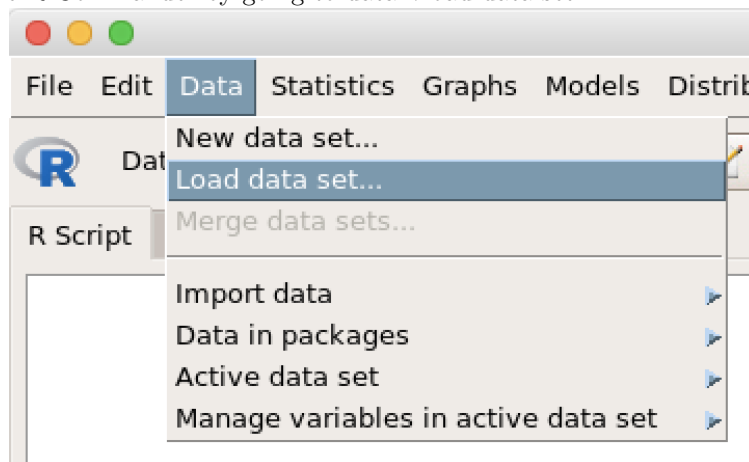
Part 1: Producing Frequency Distributions

We will first learn to produce a frequency table.

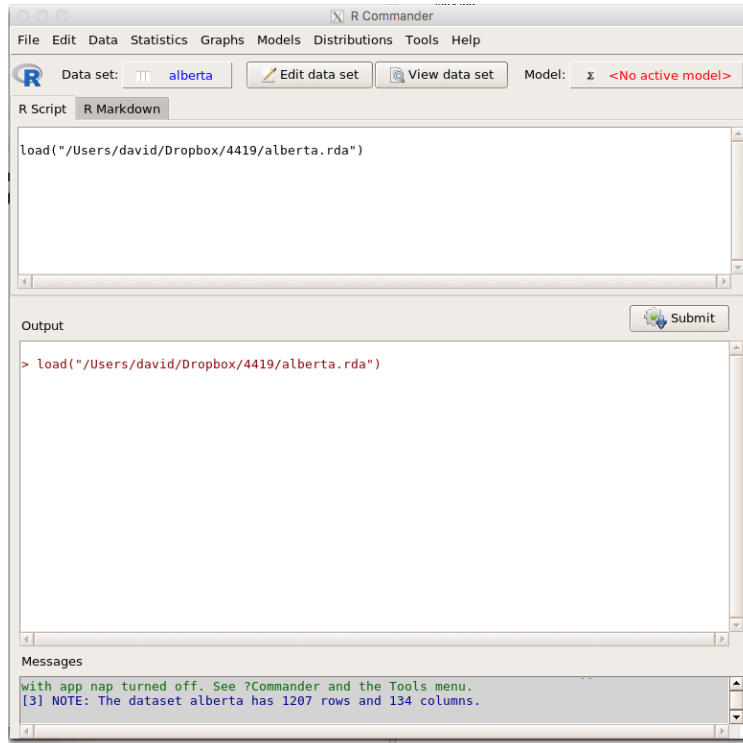
Producing a Frequency Distribution

Step 1: Open the Alberta Survey Data

Download the data from the Data folder in the Resources tab of the OWL site. The codebook is also there. The codebook tells you what each variable means and how it's coded. Once the dataset has been downloaded, you can open it in the R Commander by going to data→load data set.

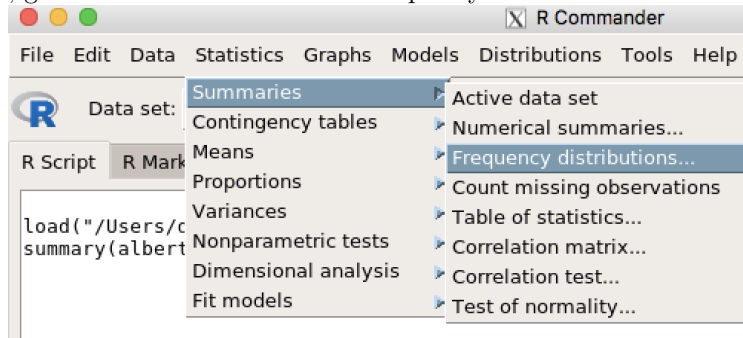


This will bring up a dialog box that will allow you to browse to the data. Click on the data file and click “OK”, then you should see something like this (though presumably with a different path to the file).

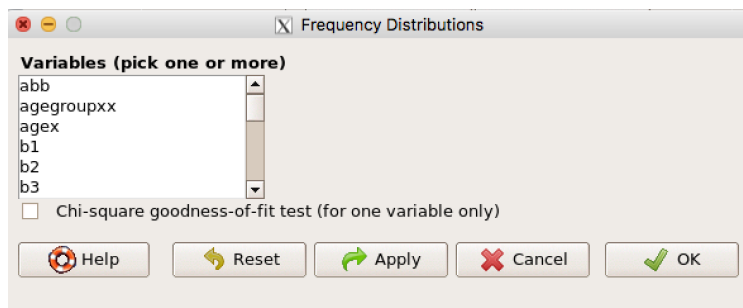



Step 2: Use menus to create a frequency distribution

In the R Commander, go to statistics→summaries→frequency distributions:



This will bring up a dialog box:



The dialog box has names in alphabetical order, you can scroll down to the variable `sex` and click on the  to produce the result. This leaves the window open for creating more frequencies. Clicking



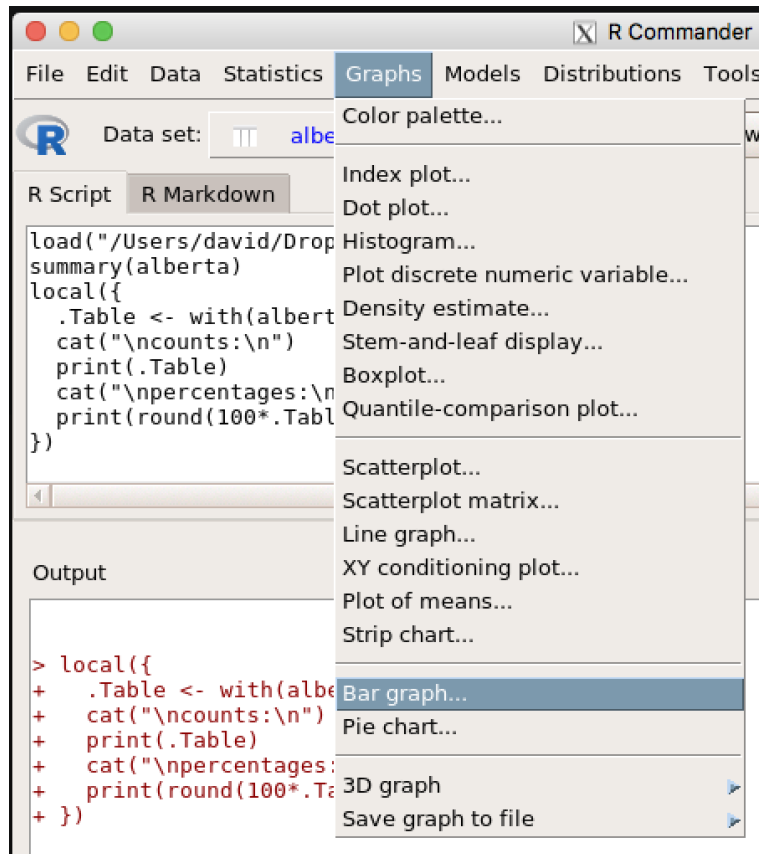
will produce the frequency and close the dialog box. This will give you the following output:

```
Output Submit
> local({
+   .Table <- with(alberta, table(sex))
+   cat("\ncounts:\n")
+   print(.Table)
+   cat("\npercentages:\n")
+   print(round(100*.Table/sum(.Table), 2))
+ })
counts:
sex
  Male Female
  595    612
percentages:
sex
  Male Female
  49.3    50.7
```

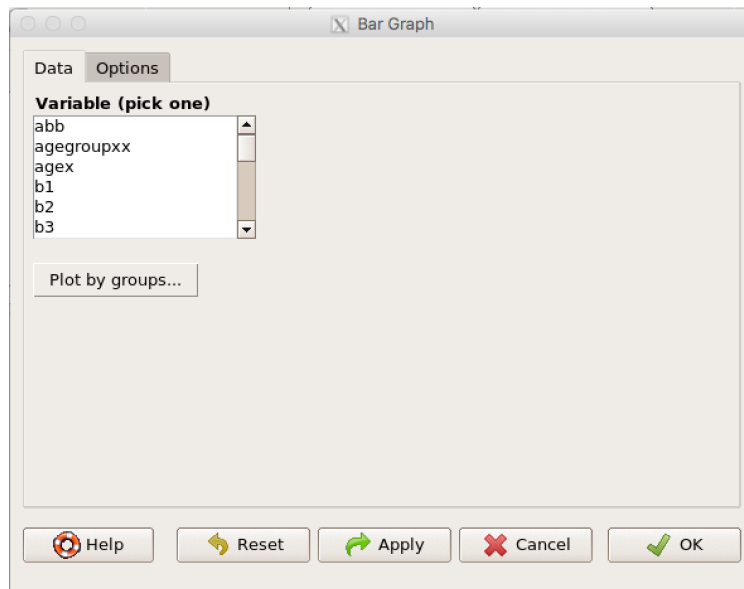
This suggests that there are 595 males in the data (49.3% of the sample) and 612 females (50.7% of the sample). In total, there are $595 + 612 = 1207$ observations in the dataset.

Part 2: Charts

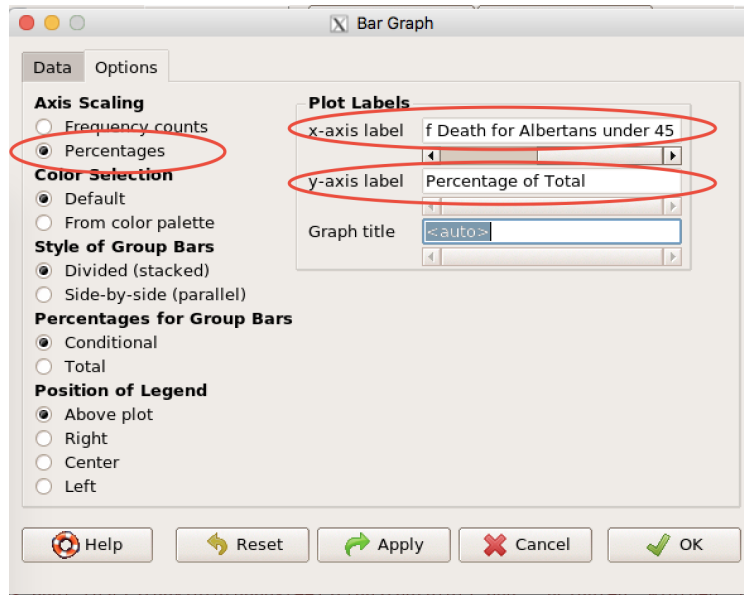
A bar chart is a way of summarizing a set of categorical data. It displays data using a number of rectangles of the same width, each of which represents a particular category. The length of each rectangle is proportional to the number of cases in the category it represents. Below, we will make a figure of the question “To the best of your knowledge, what is the leading cause of death for Albertans under the age of 45?”, question **g1** in the survey. You can make a bar chart in the R Commander by choosing Graphs→Bar Graph from the R Commander menus.



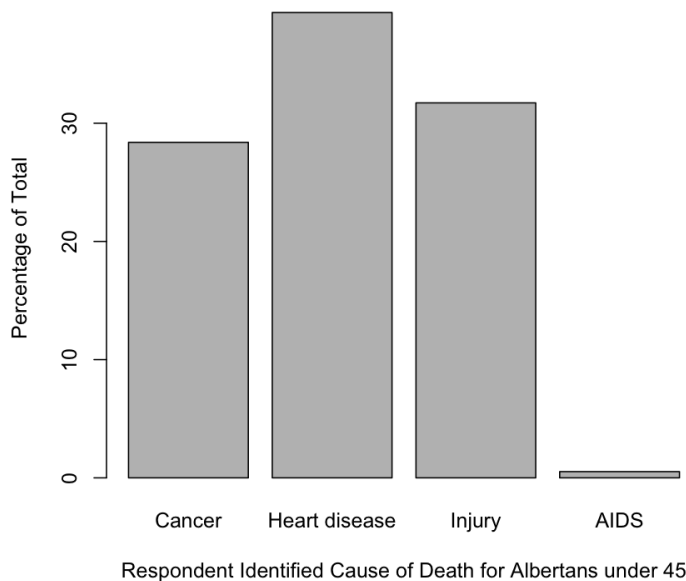
That will open up a dialog box that looks like the following:



You can also choose different options for how the figure is constructed (e.g., frequencies or percentages) by clicking on the percentages tab, which switches the dialog to the one below:



That should produce a bar plot that looks like the following:



Which Mode of Presentation is Best?

Deciding which format to use to present your data depends on the level of measurement of your selected variable and the clarity of the presentation. For example, if you have a categorical variable, but a large number of categories, a pie chart may make the presentation crowded and confusing. Selecting a bar graph may be more appropriate. Here are some general suggestions to consider.

- Use tables to display data details that would be lost in graphs or charts (e.g., if you want people to

know the precise numerical values).

- Opt for a bar graph to compare data
- Focus on the main point and consider your audience.
- Use charts for non-technical audiences when possible.

Summary

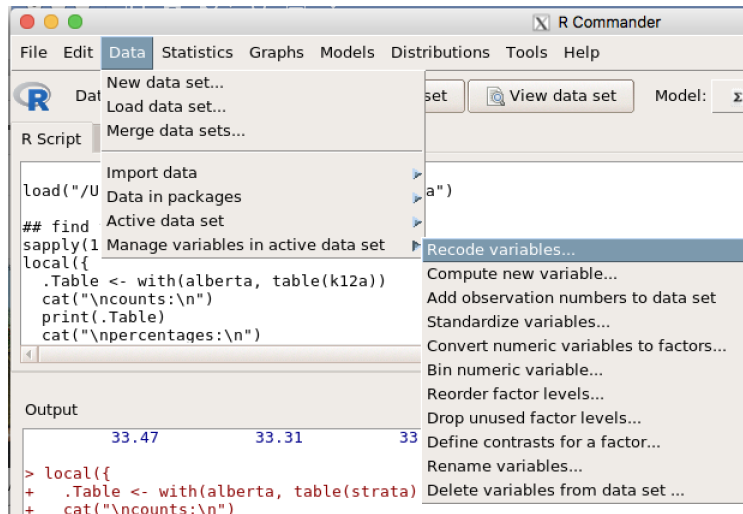
In this section, you were introduced to the basics of data presentation. You explored two methods of data presentation: frequency tables and bar graphs. Specifically, you learned how to generate and interpret frequency distributions and how to create two modes of presentation, and the connection between data presentation and levels of measurement.

Lab #4: Introduction to Probability

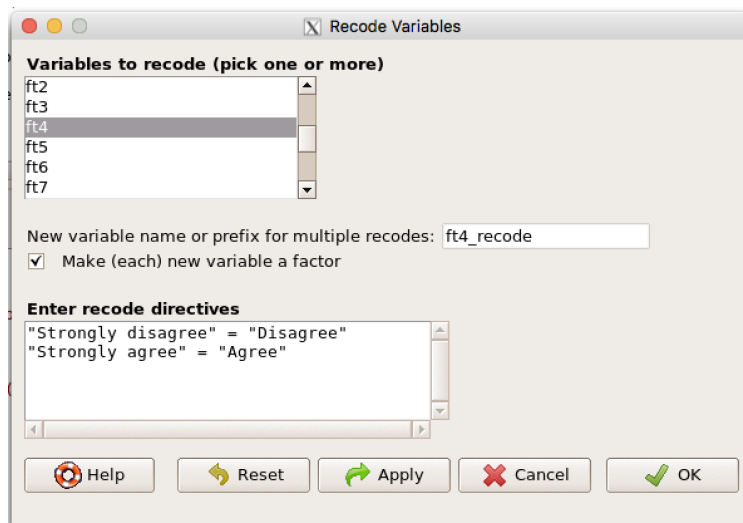
The focus of this lab is to review what you have already learned and to introduce you to the concept of recoding variables. Recoding variables is an important component in conducting analysis because the categories of a variable as they were asked in the questionnaire might not work for your specific needs. For example, if you are interested in comparing individuals who have a high school education or less with those who have a post-secondary education, you may not require a level of detail that looks at all the specific types of post-secondary education available. This lab will show you how to manipulate variables. We will also look at how to calculate probabilities from R output. This material corresponds with the material presented in Chapter 4.

Recoding Variables

Let's assume you are interested in the opinions Albertans have regarding whether temporary foreign workers are needed to fill jobs in the Alberta labour market. Within the current data set, we have a variable that asks, "Indicate how much you agree or disagree with the following statement: Temporary foreign workers are needed to fill jobs in the ALberta lobour market" (`ft4`). After obtaining a frequency distribution of the variable, you realize that you do not require this level of detail. You are interested in whether people disagree, neither agree nor disagree, or agree. Therefore, you realize you will need to collapse the first two categories together (Strongly Disagree and Disagree) and the last two categories together (Agree and Strongly Agree). You can do this in the R commander by choosing Data→Manage variables in active data set→Recode variables...



This will bring up a dialog box that will allow you to 1) choose the variable, 2) choose the name of the new variable that will be created and 3) provide the recode directives. Fill in the dialog box as below:



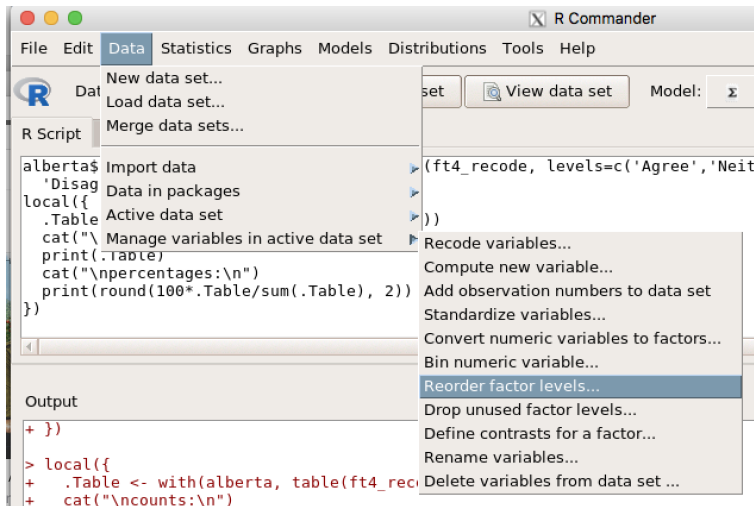
While the first two fields are mostly self explanatory, the third is not. Here, you want to tell R what to do in terms of <original value> = <new value> pairs. If the original value is a number, then you don't need quotation marks around it. If the original value is a factor label (e.g., "Strongly disagree"), then you need double quote marks around it, as in the figure above. All other values not recoded will remain their original values. You can create a frequency distribution of the variable (as we learned above) to make sure that you did it right.


```

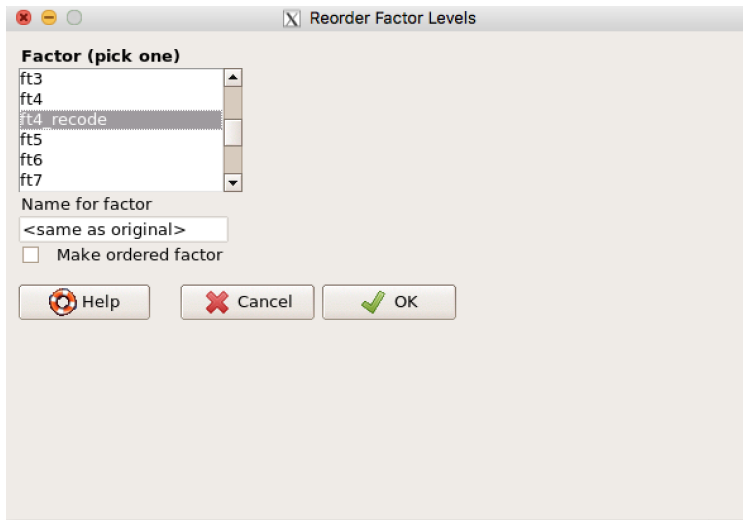
Output
+ })
> local({
+   .Table <- with(alberta, table(ft4_recode))
+   cat("\ncounts:\n")
+   print(.Table)
+   cat("\npercentages:\n")
+   print(round(100*.Table/sum(.Table), 2))
+ })
counts:
ft4_recode
           Agree           Disagree Neither disagree nor agree
           621             267             288
percentages:
ft4_recode
           Agree           Disagree Neither disagree nor agree
           52.81           22.70           24.49

```

What you might notice now is that the levels are out of order. R will automatically put them in alphabetical order. So the levels are Agree, Disagree and then Neither disagree nor agree. If you want to change that, you can choose Data→Manage variables in active data set→Reorder factor levels.



This will pop up a dialog box that asks you to pick the variable the levels of which you would like to reorder. Pick `ft4_recode` and click .



Clicking OK will pop up a warning, to which you should say “Yes”. The new dialog box that pops up will have all of the factor levels and then boxes where you can place the integer values between 1 and the total number of levels (in this case, 3). Choose the ordering you want for the new variable.

Reorder Levels

Old Levels	New order
Agree	3
Disagree	1
Neither disagree nor agree	2

Click and then create another frequency distribution of the recoded variable to make sure that everything worked.

```

Output
+ 'Agree'))
> local({
+   .Table <- with(alberta, table(ft4_recoded))
+   cat("\ncounts:\n")
+   print(.Table)
+   cat("\npercentages:\n")
+   print(round(100*.Table/sum(.Table), 2))
+ })

counts:
ft4_recoded      Disagree Neither disagree nor agree      Agree
                267          288                621

percentages:
ft4_recoded      Disagree Neither disagree nor agree      Agree
                22.78          24.49                52.81

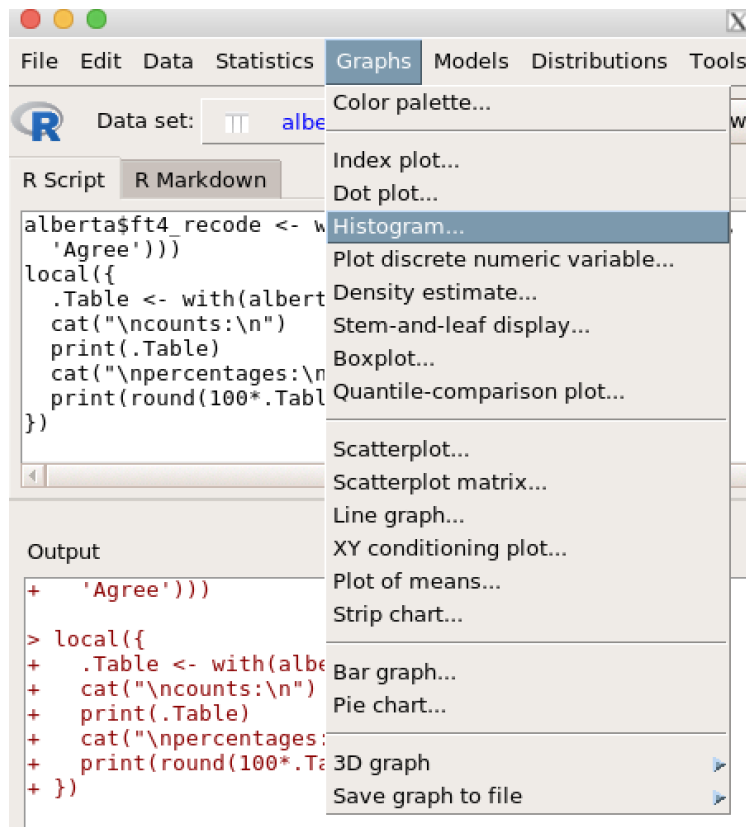
```


Lab #5: The Normal Curve

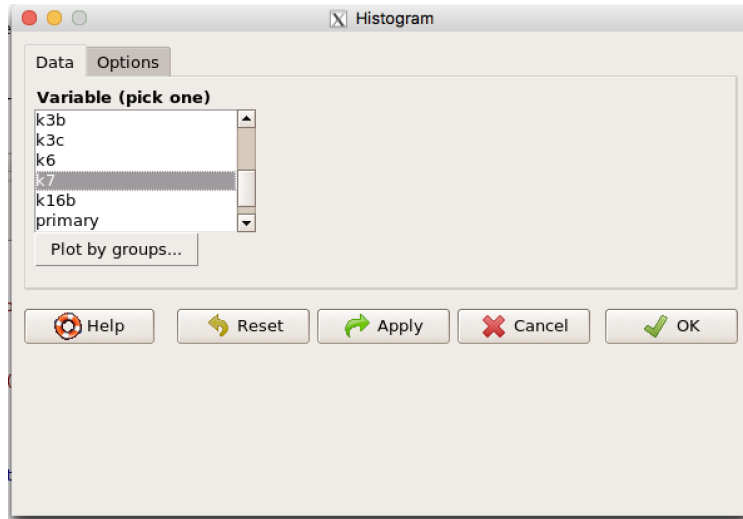
The focus of this lab is to introduce you to the concept of the normal curve. The distribution of data can take on different forms. Understanding how data are distributed is important for more complex analysis, which you will learn as this course progresses. This lab corresponds with the material taught in Chapter 5.

Creating a Histogram in R

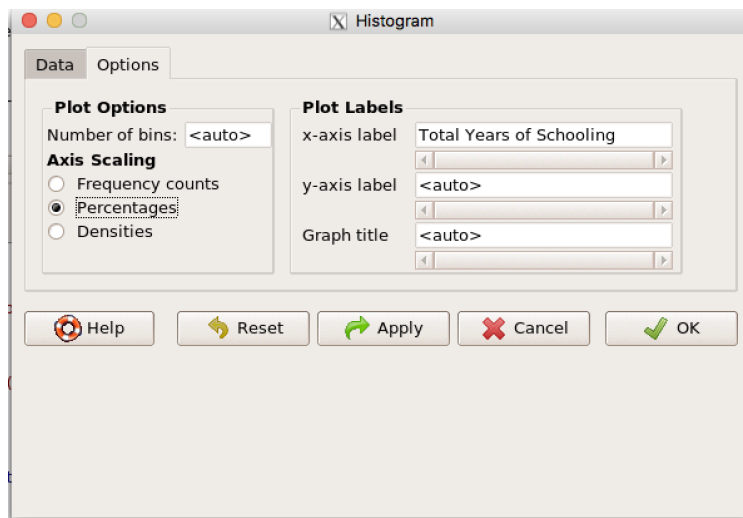
You can use the Graphs→Histogram dialog from the R Commander to make a histogram.




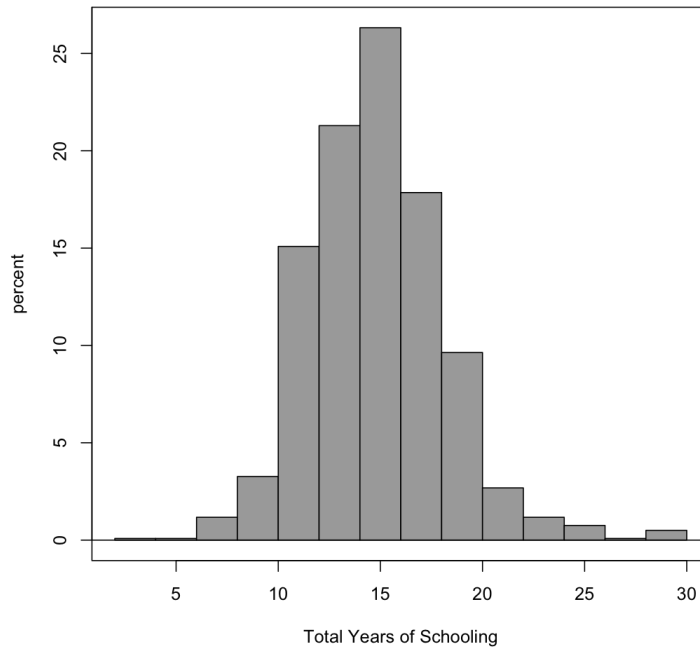
That will pop up a dialog box that allows you to pick the variable you want to use in the histogram. In this case, pick k7.



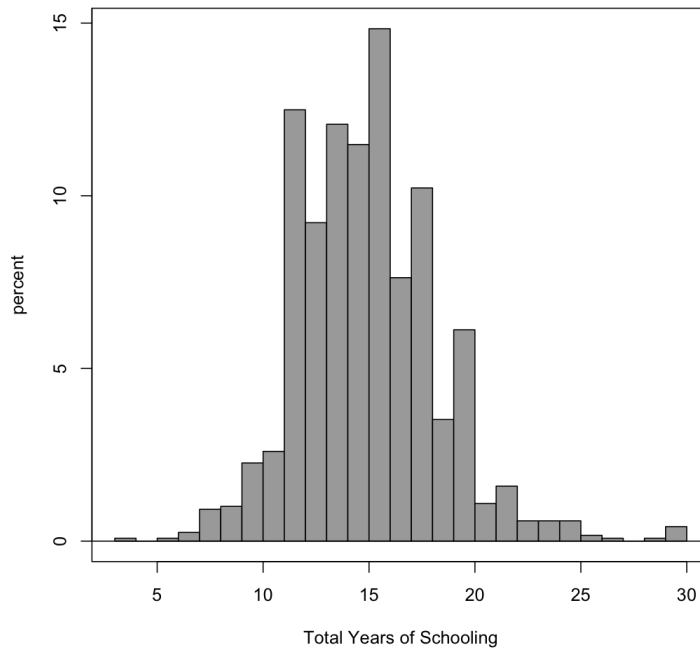
Next, click on the “Options” tab at the top of the dialog. There, you can specify the number of bins (i.e., groupings), label the axes and choose whether you want the heights of the bars to represent frequency, percentages or density (proportion divided by width of the bar).



Once you’ve filled in all of the relevant pieces, click the  button and behold! You should get something that looks like the following:



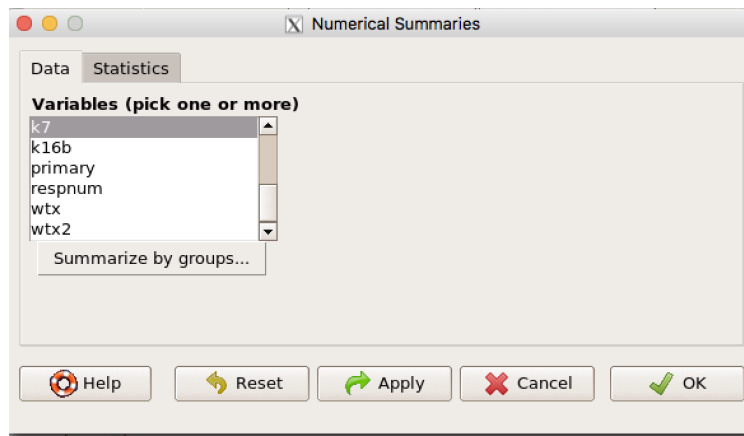
If you wanted more resolution, you could go back to the “Options” tab and change the number of bins argument from <auto> to a big number, like 30. This would produce the histogram below.



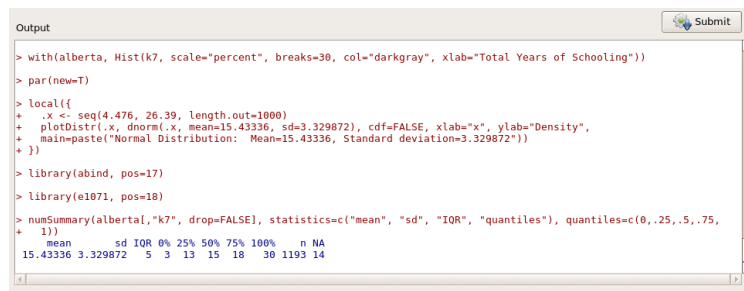
Notice that the first graph we made looks unimodal with a slight right (positive) skew, the second graph we made looks multimodal. We notice spikes at 12, 16, 18 and 20 - all times when degrees tend to get

awarded. The right skew is a bit more prominent in the second graph relative to the first. As we discussed in class, neither of these is necessarily “right”, either might be useful depending on the point you’re trying to make. For example, if you were trying to highlight the fact that very few people were especially poorly educated, the first figure would be enough. If you were trying to highlight the fact that many people persevere until the end of their degree programs, the second figure would be much more useful.

If you wanted a numerical summary (rather than a graphical one), you could get that from the Statistics→Summaries→Numerical summaries dialog.



You could pick different types of statistics to present in the “Statistics” tab, but for our purposes, the defaults are fine. The results can be seen below:



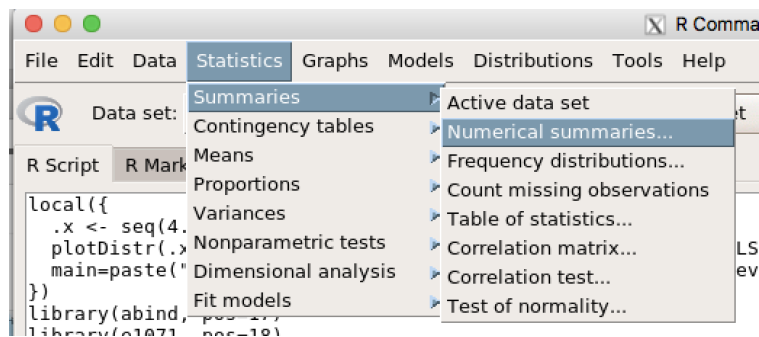
For things like the mean and standard deviation, we’ll be tackling those in the coming weeks. Other quantities, like the quartiles and the IQR, we’ve talked at least briefly about those already.

Lab #6: Measures of Central Tendency and Dispersion

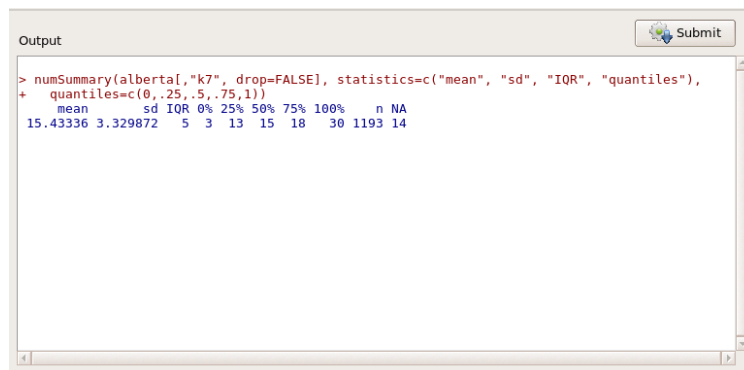
The focus of this lab is to introduce you to various measures of central tendency. Measures of central tendency allow you to further understand the distribution of a variable. In this lab, you will learn how to generate various measures of central tendency in R. This lab corresponds with the material presented in Chapter 6.

Generating Measures of Central Tendency

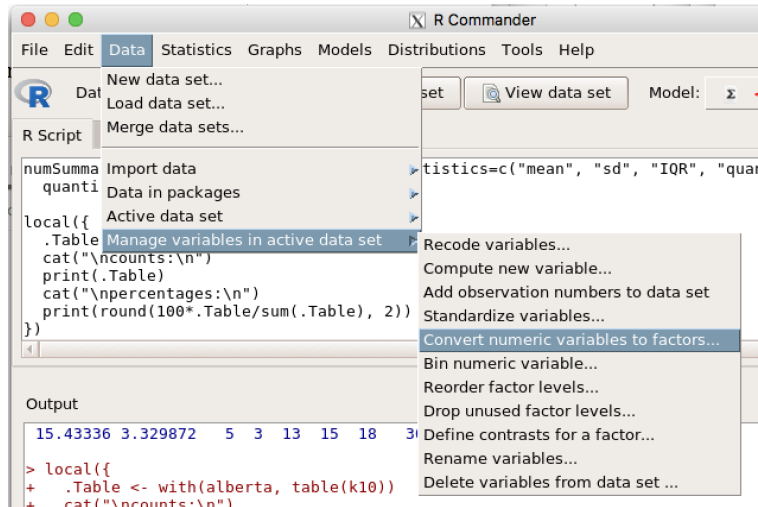
With the Alberta survey data open, the easiest way to generate a mean is to use the “numerical summaries” function. You can get there by going to Statistics→Summaries→Numerical summaries in R Commander.



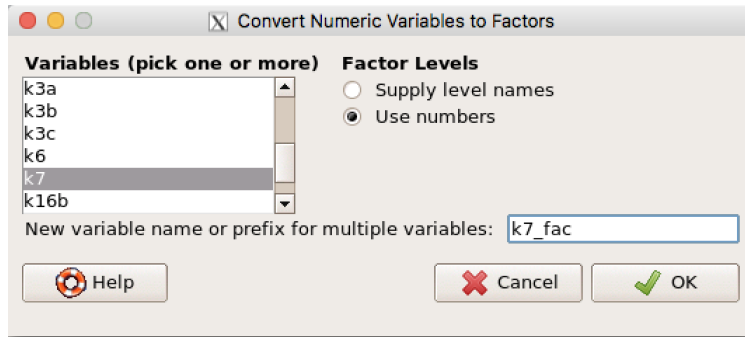
Choose the variable you're interested in, k7 in this case. Click the and you will get the following result.



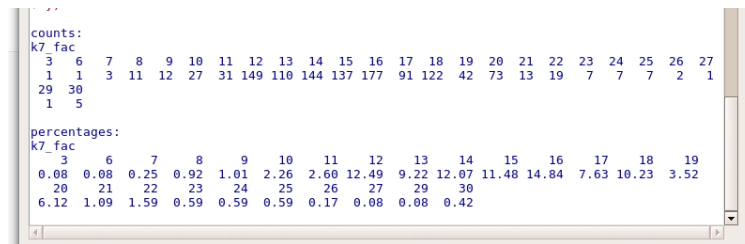
We can see that the mean here is 15.433336. The median is also a measure of central tendency and that value (the 50th percentile) is 15. Most statistical software doesn't actually produce the mode (the most frequent value), which is also a measure of a distribution's centre. This can be obtained with a frequency distribution. Now, R Commander is a bit too smart for us, in that it only gives us options for frequency distributions for factors. You'll notice that `k7` is not in the list of variables for frequency distributions. We can calculate a frequency distribution for `k7` if we change it into a factor. We can do this by going to Data→Manage variables in active dataset→Convert numeric variables to factors.



This will open a dialog box. You can pick the variable k7 from the list, click the “use numbers” option in the factor levels and make sure to put a new name, like k7_fac.



Now, making a frequency distribution of k7_fac will give you the following:



We can see that the number that has the most observations is 16 with 177 observations. This is the modal value. So, we have a mean of 15.4, a median of 15 and a mode of 16. So, all three measures tell us roughly the same thing about the centre of the distribution.

Returning to the summary statistics, we see that the standard deviation is 3.33 and the variance is 11.09. As you progress through this course, the meaning of these statistics will become clearer. However, for now, knowing how to generate these statistics in R is sufficient. We also see that we have a range of 27, by subtracting the smallest value (3) from the highest value (30). This means that the distance between the highest level of education and the lowest level of education is 27 years. Since the number is plausible, we should have faith that the data have no errors, or that we didn't do anything wrong (not trivial occurrences in statistics!).

Let's look at one more example to illustrate the difference between the measures of central tendency.

Consider the variable `k16a` - “If an election were held today, how would you vote federally?”. In particular, let’s look at its frequency distribution.

```

Output
Submit

> local({
+   .Table <- with(alberta, table(k16a))
+   cat("\ncounts:\n")
+   print(.Table)
+   cat("\npercentages:\n")
+   print(round(100*.Table/sum(.Table), 2))
+ })

counts:
k16a
      PC/Tory   Green Party   Liberals   NDP Other specified
Would not vote 502           28         191           91           15
              58           21         248
percentages:
k16a
      PC/Tory   Green Party   Liberals   NDP Other specified
Would not vote 43.50         2.43         16.55         7.89         1.30
              5.03           1.82         21.49


```

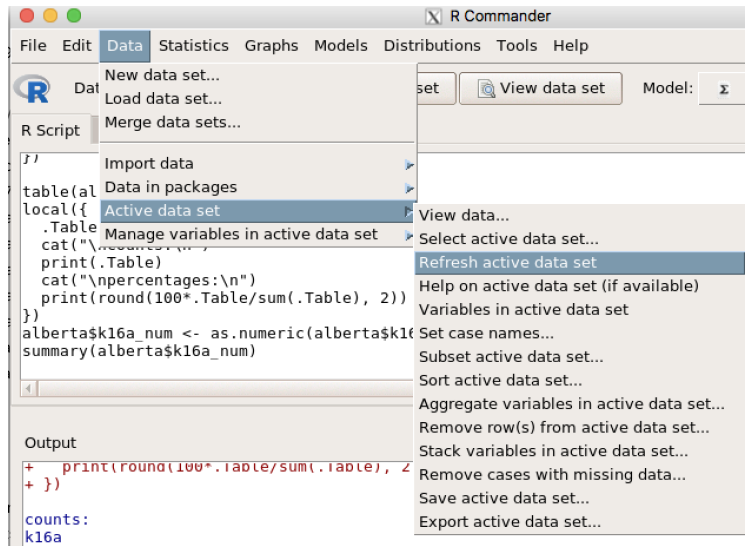
1

This variable is a nominal level variable because the categories cannot be ranked. Referring back to the material presented in Chapter 5, we know that the mode is most commonly used for nominal or ordinal level data. This is a good rule to memorize, and here is the reason. If we look at the mean for this variable (removing the don’t know and refused values, of course) we have a value of 2.28. If we were to translate that value into words, it would mean that, on average, Albertans would vote for the Green Party with a slight tendency toward the Liberals. If you have ever voted in a federal election, you know that you have to pick one, and only one, party or your ballot will be deemed invalid. Furthermore, the frequency above reveals a strong preference for the PC/Tory party. Therefore, the mathematical mean does not make sense for nominal level data. Instead, we should pick the mode, which tells us that the most frequently selected category is the PC/Tory party. Can you see why this is the case?

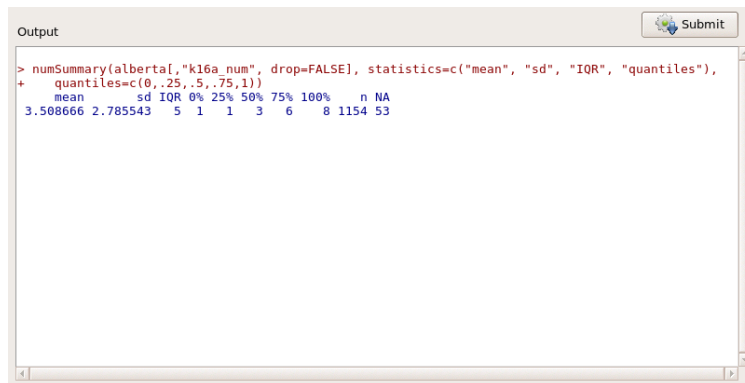
As an aside, R won’t even let you calculate the median or mean of a factor. If you wanted to do that, you would have to first convert it to a numeric variable and then find the mean. In this case, R tries to prevent us from doing something that is not meaningful statistically. It’s not always this proactive, but there are certainly times when that is the case. If, for some reason, you wanted to do this, you could always turn the factor into a numeric variable and then calculate a summary. First, we could turn it into a numeric variable with the following command:

```
alberta$k16a_num <- as.numeric(alberta$k16a)
```

Type the command into the “R Script” window and click . To get R Commander to recognize the new variable, you have to go to Data→Active data set→Refresh active data set.



Then, producing the numerical summary would give you this:

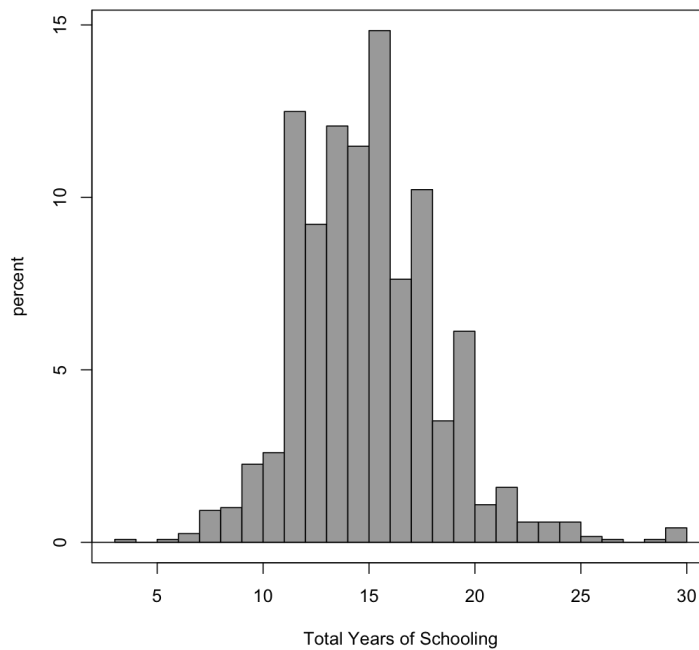


Lab #7: Standard Deviations, Standard Scores and the Normal Distribution

The focus of this lab is to introduce you to z-scores and help you further understand how the standard deviation relates to the normal curve. The most commonly used standard score, the z-score, is a measure of the relative location in a distribution. Specifically, z-scores, in standard deviation units, give the distance that a particular score is from the mean. In this lab, you will learn how to generate various z-scores with R. This lab corresponds with the material presented in Chapter 7.

Part 1: Reviewing the SHape and CHARACTERISTICS of Distributions

Before learning to calculate z-scores, let's first refresh our memories on the shape and characteristics of distributions. In the figure below is a histogram of the variable k7 - "In total, how many years of schooling do you have?" (We learned how to make this graph in Lab #5)



Now, what can we say about this distribution? It is multimodal because there are several spikes (12, 16, 18 and 20). Since the two spikes at 12 and 16 are bigger, perhaps you could argue that the distribution is bimodal. The curve appears to have a slight positive skew because there are more people at higher levels of education than at lower levels of education. When looking at the numerical summary (from Lab#5), we see that on average, respondents have 15.43 years of school, with a standard deviation of 3.33.

```

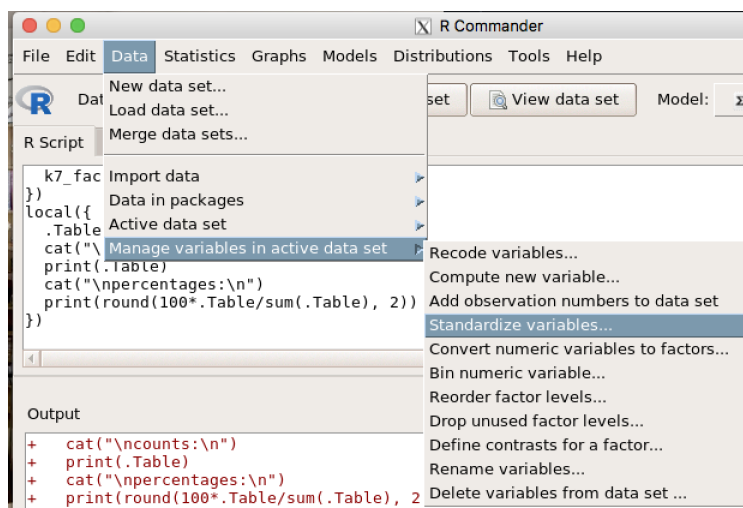
Output
Submit
> numSummary(alberta[, "k7", drop=FALSE], statistics=c("mean", "sd", "IQR", "quantiles"),
+ quantiles=c(0, .25, .5, .75, 1))
  mean      sd IQR 0% 25% 50% 75% 100%  n NA
15.43336 3.329872  5  3 13 15 18  30 1193 14

```

All right, so now we have refreshed our memory regarding the shape and characteristics of distributions. Keeping these elements in mind will help us to further understand z-scores and what it means to standardize a distribution.

Part 2: Calculating z-scores

Calculating a z-score is very similar to other things you've learned in R. You can find the command to standardize variables with Data→Manage variables in active dataset→Standardize Variables.



This makes a variable with a Z. suffix. In our case, if we pick k7 from the dialog box and click , then there will be a new variable in our dataset called Z.k7 which has the z-scores for the observations in k7. If you want to see the values that get produced, the easiest way is to type the following in the “R Script” menu and click the .

```
table(alberta$Z.k7)
```

You will get a result that looks like the one below:

```

table(albertasZ.k7)
+ })
+ })
> table(albertasZ.k7)
-3.73388571153294 -2.83295016500989 -2.53263831616887 -2.23232646732786
 1 1 3 11
-1.93201461848684 -1.63170276964583 -1.33139092080481 -1.03107907196379
 12 27 31 149
-0.730767223122775 -0.430455374281759 -0.130143525440742 0.170168323400274
 110 144 137 177
0.470480172241291 0.770792021082307 1.07110386992332 1.37141571876434
 91 122 42 73
1.67172756760536 1.97203941644637 2.27235126528739 2.57266311412841
 13 19 7 7
2.87297496296942 3.17328681181044 3.47359866065146 4.07422235833349
 7 2 1 1
4.37453420717451
 5

```

In a standardized variable, the mean should be 0 and the variance and standard deviation should both be 1. We can see whether this happened by summarizing the new variable. Recall that you can make the numerical summary with Summaries→Numerical summaries.

```

Output
> numSummary(alberta[,c("k7", "Z.k7"), drop=FALSE], statistics=c("mean", "sd", "IQR",
+ "quantiles"), quantiles=c(0,.25,.5,.75,1))
      mean      sd      IQR      0%      25%      50%      75%      100%  n NA
k7  1.543336e+01 3.329872 5.000000 3.000000 13.000000 15.000000 18.000000 30.000000 1193 14
Z.k7 1.254003e-17 1.000000 1.501559 -3.733886 -0.7307672 -0.1301435 0.770792 4.374534 1193 14

```

Here the mean is close to 0, technically, 0.00000000000000001254003, and the standard deviation (and as a result the variance, which is the standard deviation squared) is equal to 1.

You could convince yourself that the transformation worked by calculating a z-score yourself. We know from the top line of the summary above that $\mu = 15.43336$ and $\sigma = 3.329872$. So, we could calculate a z-statistic for an observation with value of 13 as:

$$z = \frac{(13 - 15.43336)}{3.329872} = -0.73077 \quad (1)$$

```

Output
> numSummary(alberta[,c("k7", "Z.k7"), drop=FALSE], statistics=c("mean", "sd", "IQR",
+ "quantiles"), quantiles=c(0,.25,.5,.75,1))
      mean      sd      IQR      0%      25%      50%      75%      100%  n NA
k7  1.543336e+01 3.329872 5.000000 3.000000 13.000000 15.000000 18.000000 30.000000 1193 14
Z.k7 1.254003e-17 1.000000 1.501559 -3.733886 -0.7307672 -0.1301435 0.770792 4.374534 1193 14

> (13-15.43336)/3.329872
[1] -0.7307668

```

Lab #8: Sampling

The focus of this lab is to introduce you to case selection in R and to help you further understand how larger sample sizes improve the accuracy of estimates. A sample that is accurately and carefully selected allows for a more precise analysis, without including the full population. Because we are often unable to survey every individual, we make decisions about how much of the population to include based on our knowledge of the population parameter. In this lab, we are going to pretend that the total number of respondents who participated in the survey represents the entire population of Alberta (as though the survey were a census). This lab corresponds with the material presented in chapters 8 and 9.

Installing the ISCSS Package

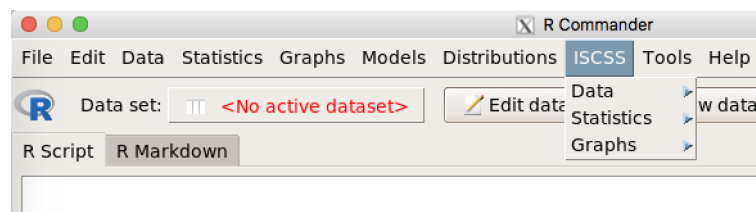
As I mentioned previously, I wrote a little package that produces a new menu structure for the R Commander. The ISCSS (Introductory Statistics for Canadian Social Scientists) package can be downloaded from github. To accomplish this, do the following things in R:

```
install.packages("devtools")
library(devtools)
install_github("davidarmstrong/RcmdrPlugin.ISCSS")
```

Now, the package is installed. To open the new menu, you can invoke the R Commander interface and load the new package:

```
library(Rcmdr)
library(RcmdrPlugin.ISCSS)
```

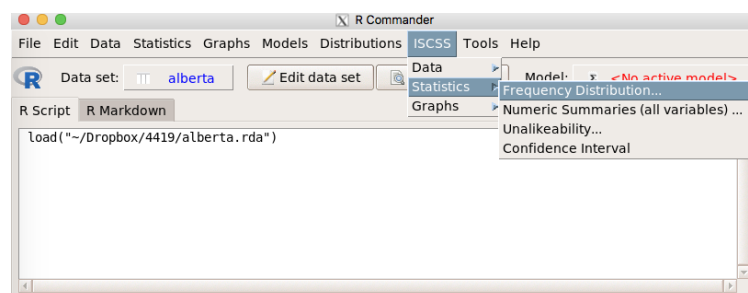
This should produce the R Commander window with a new ISCSS menu at the top:



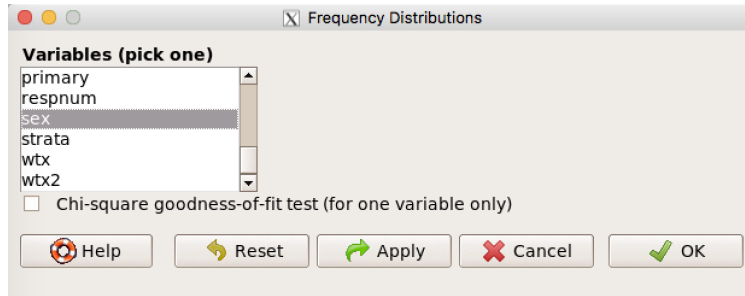
This will help us perform some of the tasks needed for this lab.

How to Select Cases

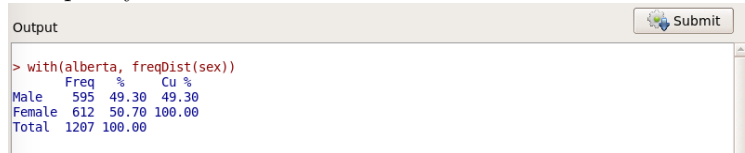
There is a different frequency distribution function in the ISCSS menu. We can use that to make a frequency distribution of sex from the Alberta data. To do this, go to ISCSS→Statistics→Frequency Distribution.



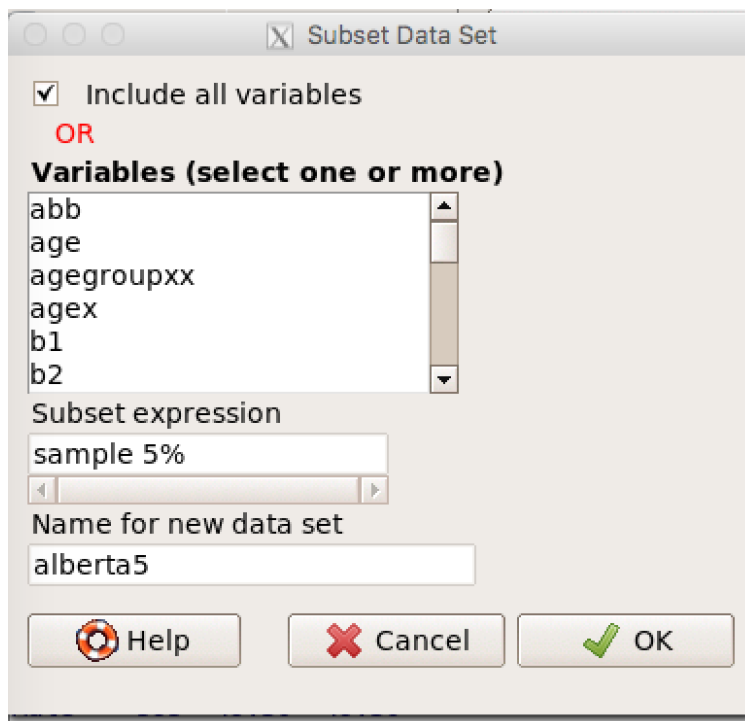
Pick the **sex** variable, click .



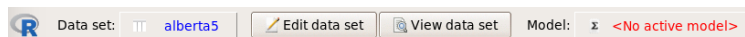
You should see the frequency distribution as below:



Note that we have 595 males and 612 females. This will be the basis of our population parameters. However, remember that these numbers, in reality, do not represent the true population of Alberta. We are only using them as a population for illustrative purposes. We can imagine random sampling from this “population”. This can be done with ISCSS→Data→Subset data (including random sampling). In the subset expression field, enter the following: sample 5% and use `alberta5` as the new dataset.



Note that this has changed the active dataset to `alberta5`:



Now, we can create the frequency distribution of sex for this new, smaller dataset

```

Output
Submit

> with(alberta, freqDist(sex))
  Freq %   Cu %
Male  595 49.30 49.30
Female 612 50.70 100.00
Total 1207 100.00

> alberta5 <- subset(alberta, subset=1:nrow(alberta) %in% c(615, 92, 406, 778, 59, 908, 782,
+ 579, 307, 824, 213, 503, 1061, 305, 200, 214, 366, 1014, 982, 32, 360, 677, 807, 601, 688,
+ 750, 153, 283, 519, 890, 977, 44, 148, 969, 407, 179, 421, 796, 825, 668, 587, 282, 331,
+ 916, 577, 311, 40, 1052, 38, 474, 329, 489, 397, 57, 1045, 632, 432, 216, 80, 94))

> with(alberta5, freqDist(sex))
  Freq %   Cu %
Male   34 56.67 56.67
Female 26 43.33 100.00
Total  60 100.00

```

Note here that we have 34 of 60 observations who are male (56.67%) whereas in our “population” we have roughly 49.3% males. Since we’re drawing a random sample, it is possible that your values will be different from mine.

Standard Error of a Sample Mean

The next thing we’re going to do is use R to help us calculate the standard error of a sample mean. Recall from chapter 9 that the equation is:

$$s_{\bar{x}} = \frac{s_x}{\sqrt{n}}$$

Using this, it is possible to estimate the distance that your sample is likely to be from a population mean. You can do this even though you don’t know what the population mean actually is, using statistical theory and what we know about the normal distribution.

Suppose that you wanted to know the average age of respondents. Remember that you would do this by using the `Numerical summaries` command in R Commander. The resulting output would give you the mean, standard deviation and number of observations (among other things). Note, we want to switch our active data set back to `alberta` and recode all values of 99 on age to NA.

```

R Script  R Markdown

load("/Users/david/Dropbox/4419/alberta.rda")
alberta <- within(alberta, {
  age <- Recode(age, '99=NA', as.factor.result=FALSE)
})
numSummary(alberta[, "age", drop=FALSE], statistics=c("mean", "sd", "IQR", "quantiles"),
  quantiles=c(0, .25, .5, .75, 1))

Output
Submit

> load("/Users/david/Dropbox/4419/alberta.rda")
> alberta <- within(alberta, {
+ age <- Recode(age, '99=NA', as.factor.result=FALSE)
+ })
> numSummary(alberta[, "age", drop=FALSE], statistics=c("mean", "sd", "IQR", "quantiles"),
+ quantiles=c(0, .25, .5, .75, 1))
  mean      sd IQR 0% 25% 50% 75% 100%  n NA
52.44133 16.34969 24 18 40 53.5 64 94 1176 31

```

We could calculate the standard error of the mean here as:

$$s_{\bar{x}} = \frac{s_x}{\sqrt{n}} = \frac{16.35}{\sqrt{1176}} = 0.477$$

If we activate the `alberta5` dataset again, we could figure out the same sampling variable for a 5% sample from our population.

```

Output
> Load("/Users/david/Dropbox/4419/alberta.rda")
> alberta <- within(alberta, {
+   age <- Recode(age, '99=NA', as.factor.result=FALSE)
+ })
> numSummary(alberta[, "age", drop=FALSE], statistics=c("mean", "sd", "IQR", "quantiles"),
+   quantiles=c(0, .25, .5, .75, 1))
   mean      sd IQR 0% 25% 50% 75% 100%  n NA
52.44133 16.34969 24 18 40 53.5 64 94 1176 31
> alberta5 <- within(alberta5, {
+   age <- Recode(age, '99=NA', as.factor.result=FALSE)
+ })
> numSummary(alberta5[, "age", drop=FALSE], statistics=c("mean", "sd", "IQR", "quantiles"),
+   quantiles=c(0, .25, .5, .75, 1))
   mean      sd IQR 0% 25% 50% 75% 100%  n NA
55.44828 14.51699 18.75 19 46.25 56.5 65 84 58 2
  
```

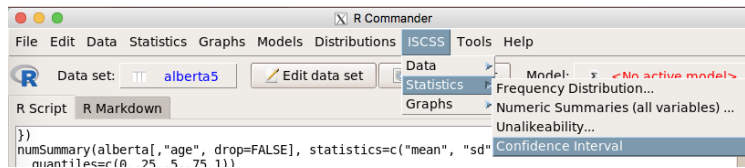
Here, the standard error is:

$$s_{\bar{x}} = \frac{14.517}{\sqrt{58}} = 1.906$$

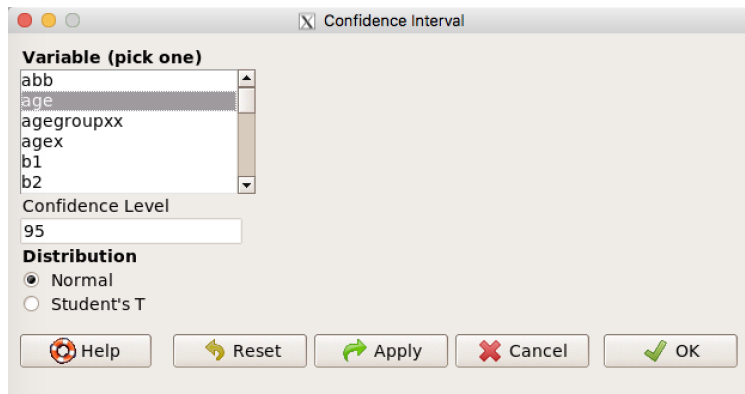
Notice that this value is a lot bigger.

Confidence Intervals

It is easy to make confidence intervals for the mean with the ISCSS menu in R Commander. Simply go to ISCSS→Statistics→Confidence Interval.



Select the variable `age`, and you can optionally choose a confidence level other than 95% and you can choose whether to use the normal or Student's T distribution. Click and you should get the following output. (make sure you switch your active dataset back to `alberta`)



You should get the following result:

```
R Script R Markdown
with(alberta, confidenceInterval(age, confidence= 0.95, distr = 'normal'))

Output Submit
> with(alberta, confidenceInterval(age, confidence= 0.95, distr = 'normal'))
  Estimate CI lower CI upper Std. Error
52.4413265 51.5068812 53.3757719  0.4767666
```

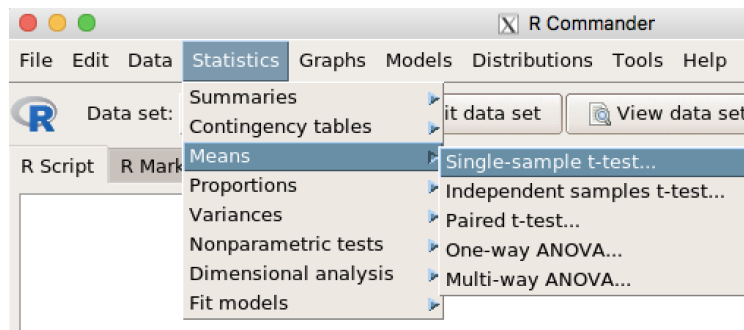
This suggests that if this is one of the “lucky” 95% of means that is within ± 2 standard errors from the true population mean, that the true population mean is in the interval [51.51, 53.38].

Lab #9: Hypothesis Testing: Testing the Significance of the Difference Between Two Means

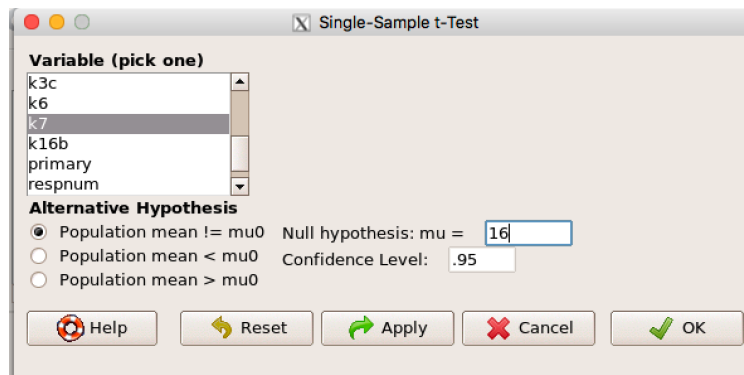
The focus of this lab is to introduce you to the “t-test” functions in R and help you further understand how we use confidence intervals to determine generalizing our samples to populations.

Calculating a One-sample t-test in R

Let’s pretend we are interested in knowing whether the average years of schooling of our sample differs significantly from the population average (which is known to be 16 years, the equivalent of high school plus an undergraduate degree) at the 95% confidence level. Suppose that we got this number from the Canadian census and that it accurately represents the entire population. After loading the Alberta survey data, you could go to Statistics→Means→Single-sample t-test.



Then, you could choose the variable `k7`, pick the population null hypothesized value as 16 and then choose whichever alternative hypothesis you’d like to evaluate. In this case, leaving the radio button for “Population mean $\neq \mu_0$ ” is the right answer because we only want to evaluate whether there is a difference (not in any particular direction).



This gives you the following result:

```
Output Submit  
  
> with(alberta, (t.test(k7, alternative='two.sided', mu=16, conf.level=.95)))  
  
      One Sample t-test  
data:  k7  
t = -5.8776, df = 1192, p-value = 0.00000005397  
alternative hypothesis: true mean is not equal to 16  
95 percent confidence interval:  
 15.24422 15.62251  
sample estimates:  
mean of x  
15.43336
```

This tells us that the sample mean is 15.43336. It has a 95% confidence interval of (15.24422, 15.62251). The t -statistic is -5.8776, meaning that the observed sample mean of 15.43336 is more than 5 standard deviations smaller than the null hypothesized value. This leads to a p -value that is approximately zero and certainly less than 0.05. If you wanted to know how many cases there are, you could look at the degrees of freedom (df) number and add 1, so 1193 in this case.

Lab #10: Hypothesis testing: One- and Two-tailed Tests

The focus of this lab is to introduce you to two-sample t-tests in R. For this type of analysis, you need a dichotomous independent variable (i.e., one that only has two values) and an interval- or ratio-level dependent variable. This lab corresponds with the material in Chapter 11.

Calculating a t-test with Two Samples in R

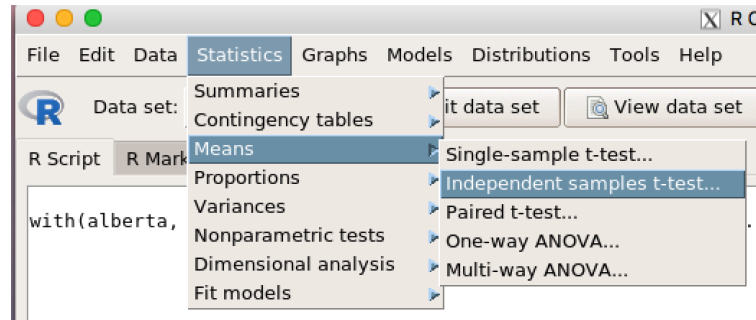
Let's pretend we are interested in knowing whether the average years of schooling of our sample differs significantly between males and females. Since both are samples and since scores on the outcome of interest are independent of each other (presumably the total years of schooling among women has nothing to do with the total years of schooling among men), it is most appropriate to conduct a *t*-test on independent samples.

In this case, we are going to test the alternative hypothesis that men's average years of schooling will differ from the average years of schooling for women. That is,

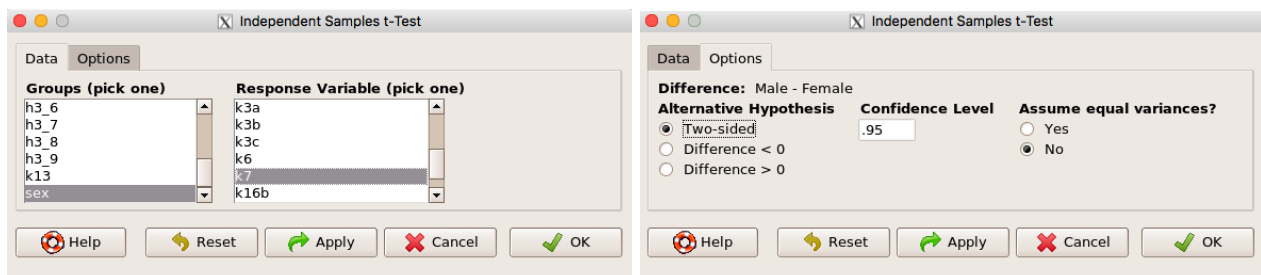
H_0 : In the population, the mean years of schooling for men and women does not differ ($\mu_{\text{men}} = \mu_{\text{women}}$)

H_A : In the population, the mean years of schooling for men differs from that of women ($\mu_{\text{men}} \neq \mu_{\text{women}}$)

To do this in R, we would go to Statistics→Means→Independent samples t-test.



You will need to pick a grouping variable in the first variable box and a quantitative response in the second variable box (left pane below). Then, you can specify the options, though in this case we would probably leave them how they are (right pane below).



This will give you the following result:

```
Output
Submit
> t.test(k7~sex, alternative='two.sided', conf.level=.95, var.equal=FALSE, data=alberta)

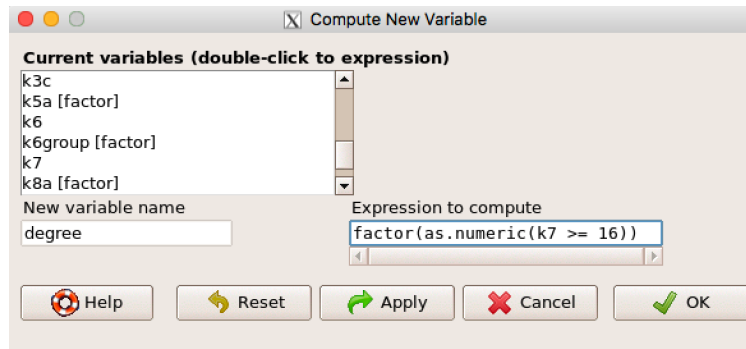
Welch Two Sample t-test

data:  k7 by sex
t = 2.2047, df = 1190.9, p-value = 0.02766
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04671283 0.80177458
sample estimates:
mean in group Male mean in group Female
15.64957             15.22533
```

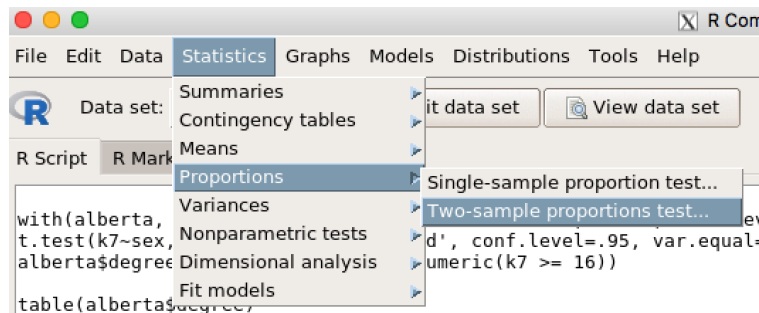
The result tells us that the two means are 15.64957 for males and 15.22533 for females. Note, not surprisingly, that these two means are on either side of the overall mean we calculated in the last lab of 15.43336. If

there were equal numbers of men and women, this number would be exactly between the two group means. The t -statistic is 2.204 on roughly 1191 degrees of freedom. The p -value is 0.02766 which is less than 0.05, meaning that there is a statistically significant difference between the average education of men and women.

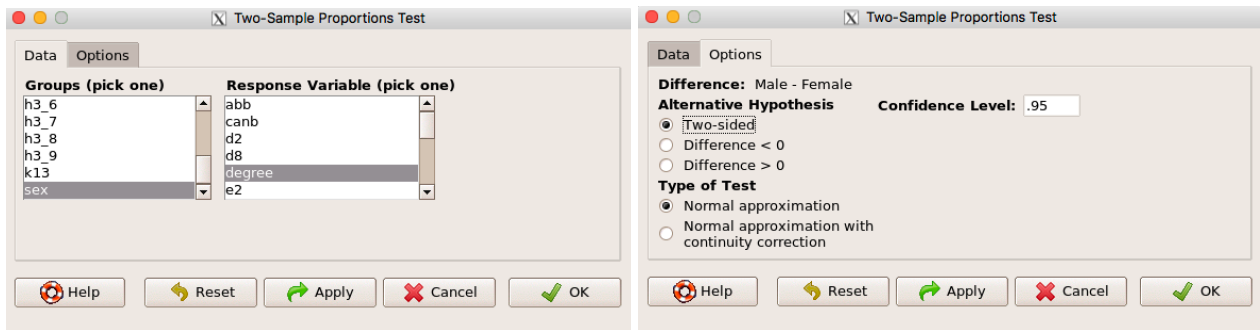
The interesting question now, is - is there a *substantively* significant difference between men and women's average education. This question is a bit more difficult to answer. The difference, on average, is about one half of a year. Certainly a half of a year could make a difference if that extra half of a year resulted in an extra credential (e.g., a bachelor's degree). However, it is difficult to infer that kind of impact from this result. To figure something like this out, we could see if men get degrees more often than women. First, we could create a dummy variable coded 1 if the respondent had 16 or more years of education and 0 otherwise. To do this, go into Data→Manage variables in active data set→Compute new variable. Then, you can fill in the dialog box as follows:



Next, you could use the a two-sample difference of proportions test. To do this, go to Statistics→Proportions→Two-sample Difference of Proportions.



Then, you should be able to choose the grouping variable (**sex**) in the left-hand variables box and the response variable (**degree**) from the right-hand variables box (as in the left panel below). You can switch over to the options tab and change the options if you like, but there's no real need here.



Clicking OK, you will get the following result:

```

Output
Submit
> local({ .Table <- xtabs(~sex+degree, data=alberta)
+ cat("\nPercentage table:\n")
+ print(rowPercents(.Table))
+ prop.test(.Table, alternative='two.sided', conf.level=.95, correct=FALSE)
+ })

Percentage table:
      degree
sex    0    1 Total Count
Male  47.0 53.0   100    585
Female 57.7 42.3   100    608

      2-sample test for equality of proportions without continuity correction

data: .Table
X-squared = 13.743, df = 1, p-value = 0.0002096
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.16358705 -0.05084727
sample estimates:
 prop 1    prop 2
0.4700855 0.5773026

```

This suggests that men get degrees at a rate of 53% and women at a rate of 42.3% (if having 16 years of education is equivalent to a degree). This difference in proportions of 10.7% is significant with a χ^2 statistic (which we haven't talked much about yet) of 13.74 with 1 degree of freedom (more on this in a couple of weeks). The p -value here is 0.0002, certainly less than 0.05, so we would reject the null hypothesis that men and women get bachelors degrees at the same rate. Here, the difference seems quite substantial. In this case, getting another perspective on this question provides us more and perhaps more interesting information.

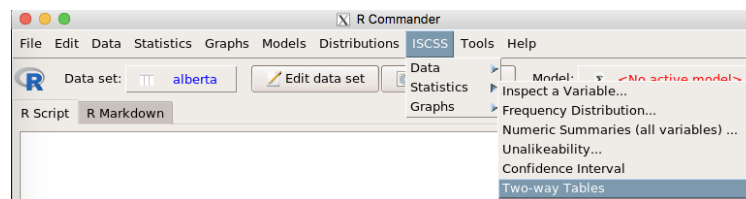
Lab #11: Bivariate Statistics for Nominal Data

The focus of this lab is to introduce you to the association or relationship between nominal variables. Specifically, this lab will help clarify your understanding of independent and dependent variables and how to interpret the χ^2 test of independence. This lab corresponds with the material in Chapter 12.

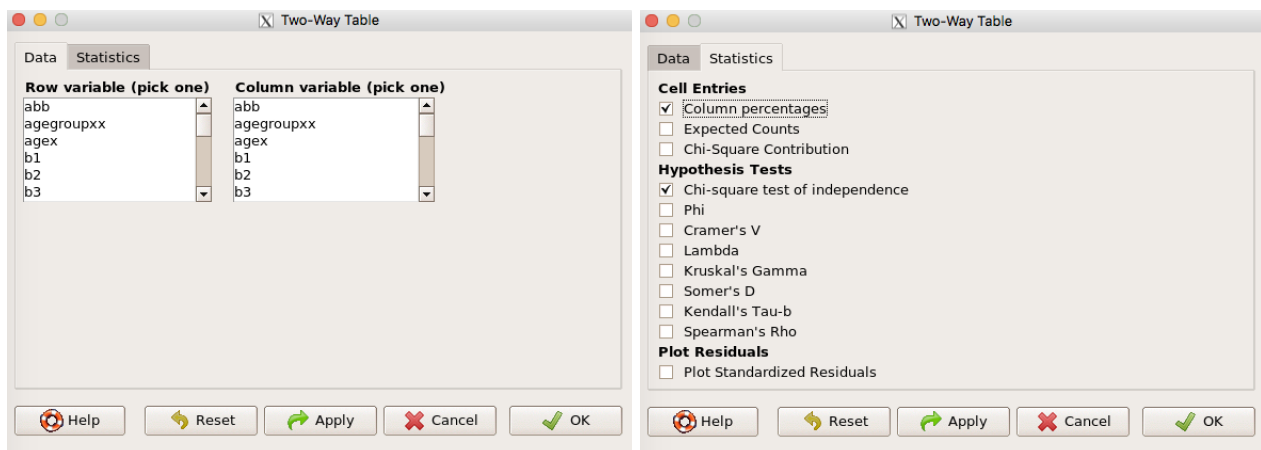
Creating Contingency Tables in R

Now, let's learn how to create a contingency table in R. For this example, we will ask this question: "Who is more likely to have been diagnosed with high blood pressure - males or females?" An independent variable can be thought of as modifying the outcome. The dependent variable can be thought of as the outcome of interest. In this situation, we are interested in seeing if gender (**sex**) will modify patterns of high blood pressure (**e14_1**); therefore, it is our independent variable. The outcome we are interested in is high blood pressure; therefore, it is our dependent variable.

To make the cross-tabular in R, you can use the "Two-way Tables" function in the ISCSS menu in the R Commander.



This will bring up a dialog box that has two tabs - one where you can pick variables (on the left below) and one where you can specify options (on the right below)



If you choose the two variables mentioned above, you can also pick the **Chi-square test of independence**, **Cramer's V**, **Phi**, and **Lambda** - all of which are amenable to nominal 2×2 tables. You will get the following result:

Output Submit

Cell Contents

		N	
		N / Col Total	
Total Observations in Table: 1207			
e14_1	sex	Male	Female
Not selected		426	460
		0.716	0.752
Selected		169	152
		0.284	0.248
Column Total		595	612
		0.493	0.507

	statistic	p-value
Chi-squared	1.9660	0.1480
Phi	-0.0404	0.9172
Cramers V	-0.0404	0.9172
Lambda	0.0000	1.0000

Note that there's not much of a relationship here. 28% of men have high blood pressure and 25% of women have high blood pressure. This is certainly not a huge difference. Measures of statistical significance would suggest, similarly, that the result is not statistically reliable. That is to say, we cannot be sufficiently sure that in the population these two variables are not independent. This leads us to the inference that gender and high blood pressure are not related.

Lab #12: Bivariate Statistics for Ordinal Data

The focus of this lab is to introduce you to the association or relationship between ordinal variables. Specifically, this lab will help clarify your understanding of independent and dependent variables, how to interpret the χ^2 test as well as other measures of association. This lab corresponds with the material presented in Chapter 13.

Establishing Your Research Question and Identifying Your Variables

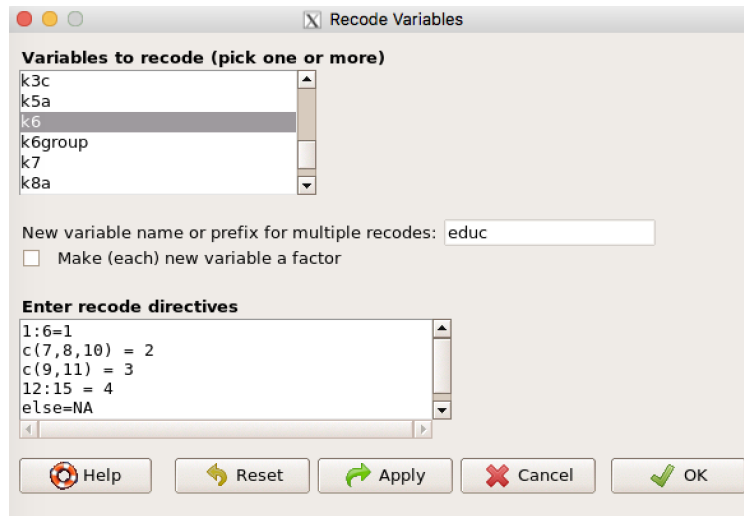
In this example, we will ask this question: Does level of education affect an individual's opinion about whether Alberta should build stronger ties with China?

An independent variable can be thought of as modifying the outcome, and the dependent variable can be thought of as the outcome of interest. In this situation, we are interested in seeing if the highest level of education will modify opinions about Alberta's ties to China.

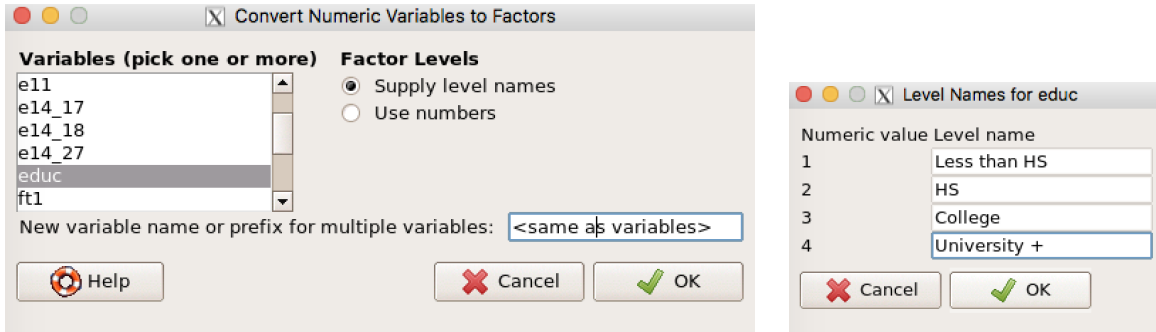
In this example, we are going to recode our dependent variable in the following way:

Old Value	New Value
1,2,3,4,5,6	1 (Less Than High School)
7,8,10	2 (High School)
9, 11	3 (College, certificate, diploma)
12, 13,14, 15	4 (University Degree or Higher)
99	Missing

You can do this as follows:



Next, you'll need to change the result into a factor. You can do this with the Manage Variables in Active Data Set→Convert Numeric Variables to Factors command. You can choose `educ` (or whatever you called it) from the variables box and keep the "supply levels" radio button clicked (left panel below). If you choose not to change the name of the variable, when you click OK, you'll have to choose "yes" from the warning box that pops up. Then you can fill in the second value labels dialog box as in the right panel below.



Next, we can make the cross-tabulation between `educ` and `b2` and ask for a number of measures of fit for the table. The result is below:

Output Submit

```

-----+-----+-----+-----+-----+
|                N |
| N / Col Total |
+-----+-----+-----+-----+-----+

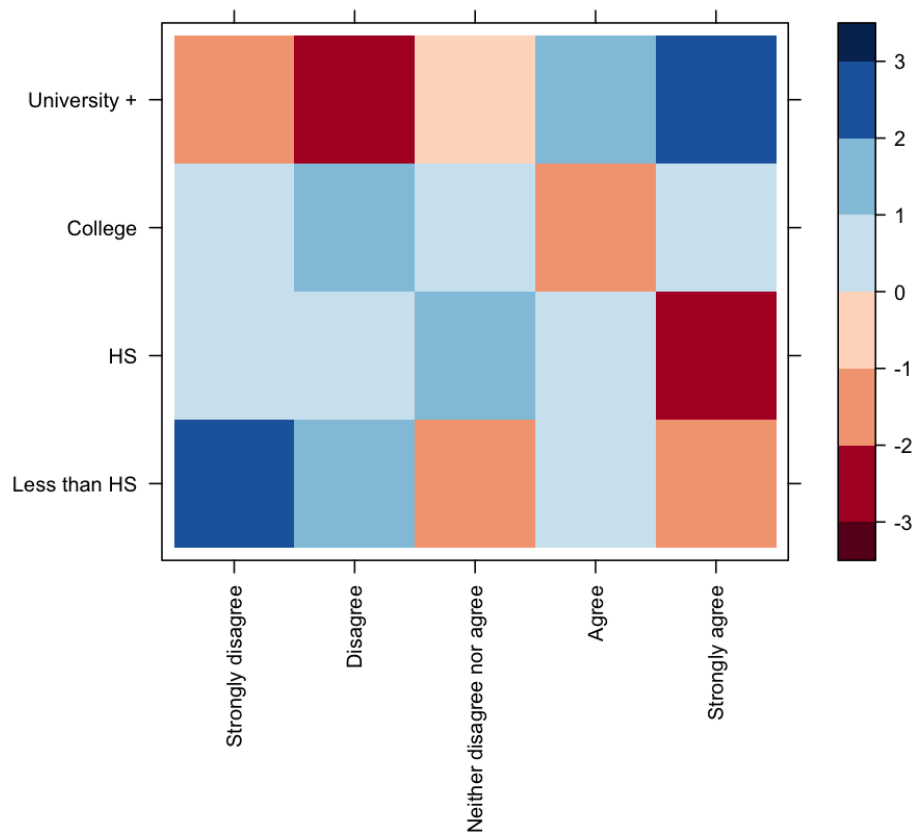
Total Observations in Table: 1151

      b2 | educ
      | Less than HS |      HS |      College |      University + |      Row Total |
-----+-----+-----+-----+-----+-----+
Strongly disagree |          8 |        15 |          15 |          15 |          53 |
|          0.113 |        0.047 |        0.048 |        0.033 |          |
-----+-----+-----+-----+-----+-----+
Disagree |         17 |        59 |         66 |         67 |         209 |
|          0.239 |        0.185 |        0.211 |        0.150 |          |
-----+-----+-----+-----+-----+-----+
Neither disagree nor agree |          9 |        71 |         64 |         88 |         232 |
|          0.127 |        0.223 |        0.204 |        0.196 |          |
-----+-----+-----+-----+-----+-----+
Agree |         36 |       161 |        142 |        235 |         574 |
|          0.507 |        0.505 |        0.454 |        0.525 |          |
-----+-----+-----+-----+-----+-----+
Strongly agree |          1 |        13 |          26 |          43 |          83 |
|          0.014 |        0.041 |        0.083 |        0.096 |          |
-----+-----+-----+-----+-----+-----+
Column Total |         71 |       319 |        313 |        448 |        1151 |
|          0.062 |        0.277 |        0.272 |        0.389 |          |
-----+-----+-----+-----+-----+-----+

      statistic p-value
Chi-squared      30.1227 0.0052
Cramers V         0.0934 0.0052
Lambda           0.0000 0.0008
Kruskal-Goodman Gamma 0.1304 0.0000
Somers D          0.0892 0.0000
Tau-b            0.0892 0.0000

```

The results suggest a weak ordinal relationship, though there is certainly an association between the two variables. If there is a positive relationship, looking at the table, higher column percentages should be seen moving from the upper-left to the lower-right of the table. If there is a negative relationship, it should be moving from the lower-left to the upper-right of the table. In this case, the relationship looks vaguely positive, but not strong. The numeric measures (γ , d , τ_b). We could also look at a plot of studentized residuals.



In the figure above (which you can get by checking the “plot standardized residuals” box on the options page in the two-way tables dialog), we see basically a pattern that makes sense. Moving in this case from the lower-left (low values on both variables) to the upper-right (high values on both variables), the values seem to be mostly blue (higher than expected counts) with lower than expected counts out toward the other corners (upper-left and lower-right).

Lab #13: Bivariate Statistics for Interval/Ratio Data

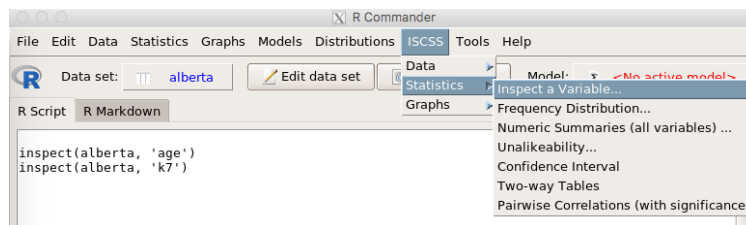
The focus of this lab is to introduce you to the association or relationship between interval/ratio level variables. Specifically, this lab will help clarify your understanding of Pearson's r and explained variance. This lab corresponds with the material presented in Chapter 14.

Calculating Pearson's r in R

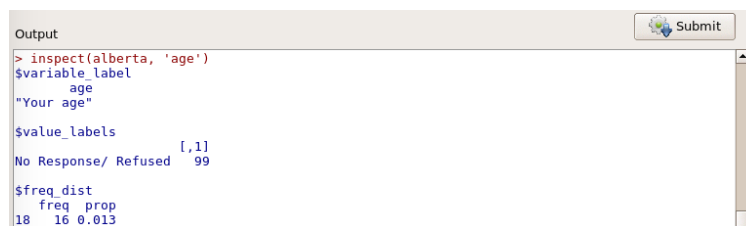
When calculating Pearson's r , we make the assumption that the two variables we are correlating (evaluating the extent to which the variables are related) have a linear relationship. That is, the relationship between the two variables is the same, regardless of what the value of either of those variables is. So, the first step in calculating Pearson's r is to evaluate whether the relationship between the two variables is, in fact, roughly linear.

In this example, we are interested in whether an individual's age is correlated with the number of years of schooling they have completed. In this case, our dependent variable is the number of years of schooling ($k7$) and our independent variable is an individual's age (age).

The first thing we need to do is to figure out whether the variables have any values we need to recode to missing. We could do this by using the ISCSS→Statistics→Inspect Variable option.

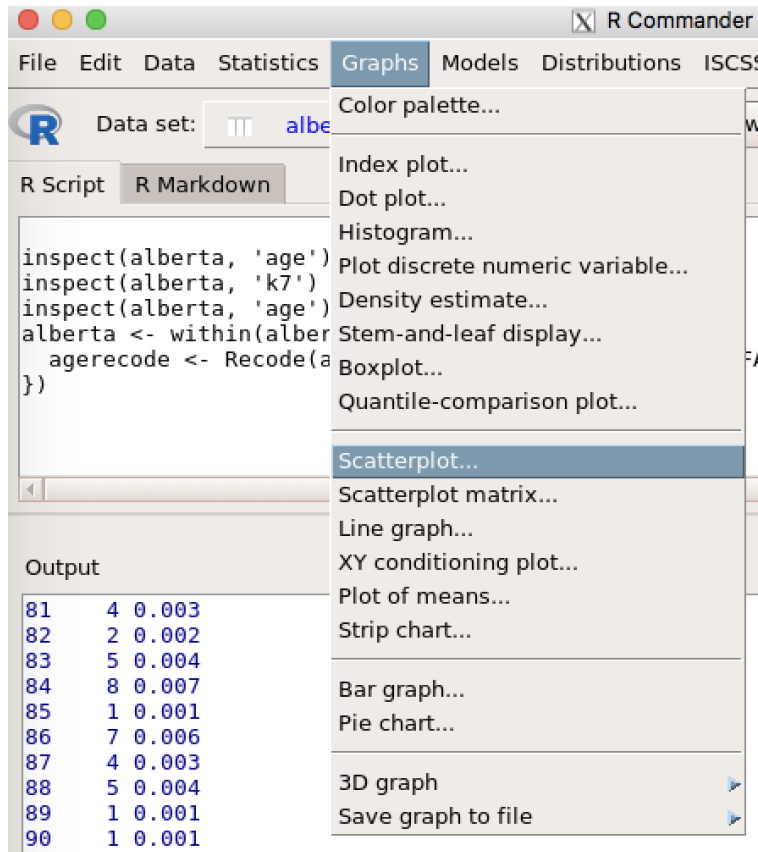


If you click the age variable and click , you should get the following result:

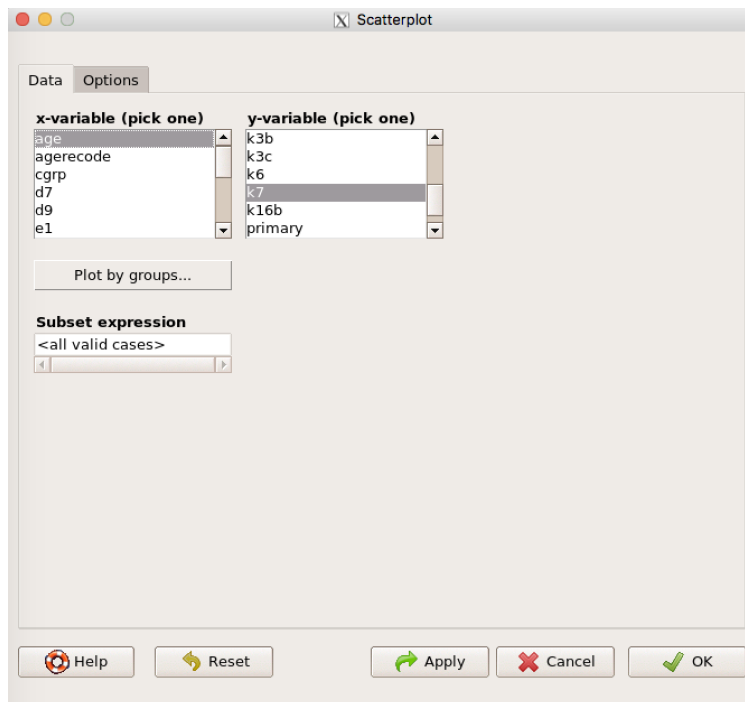


This shows that 99 is a variable that you need to remove. You will also find through the same procedure that the values 98 and 99 are missing for education, but those values don't actually show up in the dataset, so we don't have to recode $k7$. Recoding age such that the 99s are NA should be old hat now, so I will assume that you can either do it, or go back to Lab 4 if you need some help. Put the recoded value of age in $agerecode$.

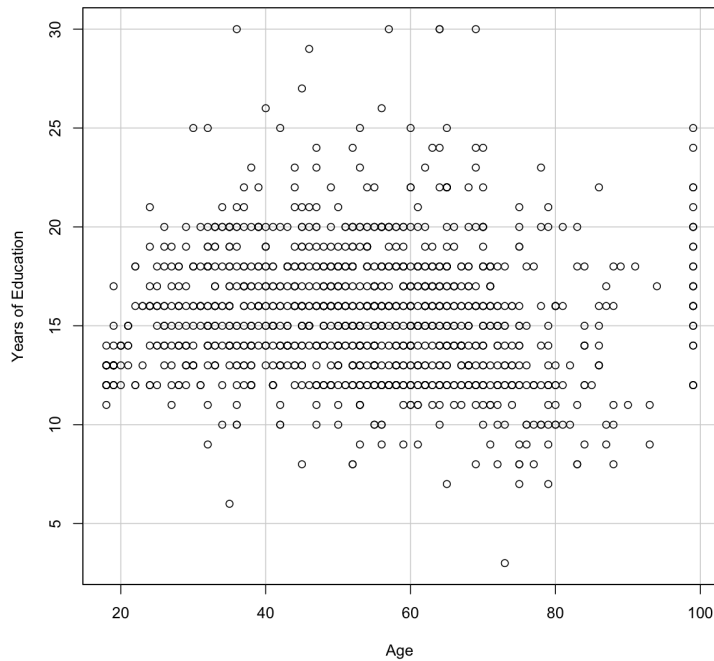
Next, we can make a scatterplot to see whether the variables are sufficiently linearly related. You can go to the scatterplot menu:



Then, you can fill in the scatterplot dialog box as follows:



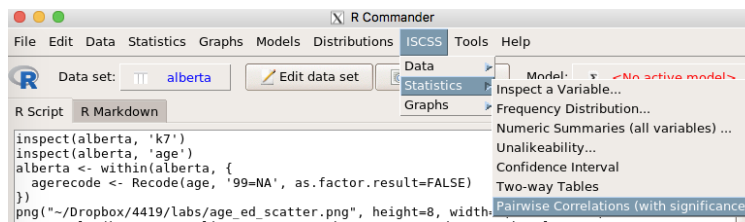
That will produce the following:



Here are a few observations we could make about this scatterplot:

1. Though not as linear as we may have liked, the dots do not appear to be arranged in a non-linear pattern. There may be a weak relationship there, but whatever we find will not be a because of non-linearity.
2. There are quite a few outlying points on both sides of the point cloud.

Now that we've done the necessary diagnostics, we are ready to calculate Pearson's r . The computation of Pearson's r is done "behind the scenes" and is quite complex despite its easy implementation. In a nutshell, it considers the amount of covariation between your X variable and your Y variable. As you know, Pearson's r ranges from -1 to 1. To calculate the correlation you could use the ISCSS→Statistics→Pairwise Correlation option to find the pairwise correlation with significance.



You can choose `agerecode` and holding the `ctrl` key down, click on `k7`, too. Keep the t -test option radio button and click . That will produce the following result:

```
Output Submit  
> pwCorrMat(alberta[,c("agerecode", "k7")], method='t')  
All Correlations  
      agerecode  k7  
agerecode    1.0 -0.1  
k7           -0.1  1.0  
  
Only Significant Correlations  
      agerecode k7  
agerecode    -0.100  
k7
```

This shows that the correlation of -0.1 is statistically distinguishable from zero even if it is not substantively all that strong.

Lab #15: OLS Regression - Modelling Continuous Outcomes

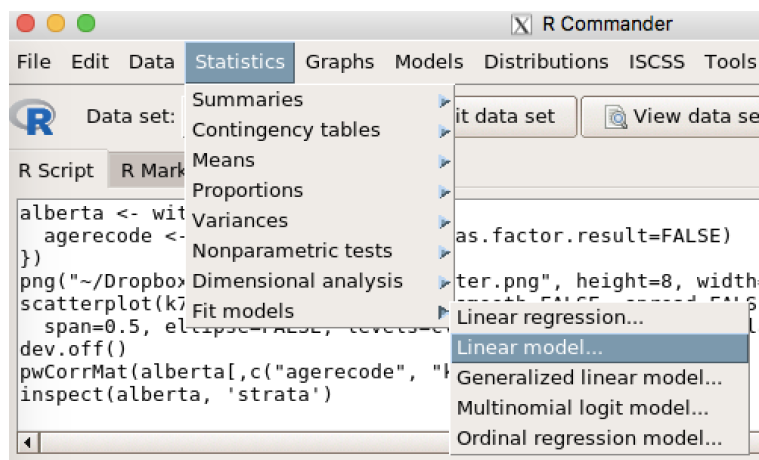
The focus of this lab is to introduce you to multiple regression analysis. Specifically, this lab will help clarify your understanding about when this procedure should be used, how to calculate and interpret OLS regression and how to compute dummy variables. This lab corresponds with the material presented in Chapter 16.

Calculating OLS Regression

In this example, we are interested in which variables might affect the number of years of schooling an individual has completed. For various reasons, we think that the number of years of schooling an individual has completed may vary by gender (`sex`) and by what part of Alberta they live in (`strata`).

Because of the conditions of OLS regression, all variables must be numeric, meaning we have to convert factors with m categories into $m - 1$ dummy variables. The nice thing is that R will do this for us automatically. The only time we have to worry about this is when the variable is not a factor in our dataset. By using the inspect variables feature mentioned above, we can see that both `sex` and `strata` are categorical variables.

We can use the Statistics→Fit Models→Linear Model option.



This will bring up the following dialog box. You can fill in the dependent variable by double-clicking the variable from the variable box (or by typing it into the dependent variable box). Then, click into the independent variables box and then double-click variables into the box. Make sure to separate them (at least for now) by the plus sign +.

Linear Model

Enter name for model:

Variables (double-click to formula)

k6group [factor]
k7
k8a [factor]
k10 [factor]
k11 [factor]
k12a [factor]

Model Formula

Operators (click to formula):

Splines/Polynomials:
(select variable and click)

df for splines:
deg. for polynomials:

~

Subset expression
Weights

Clicking will produce the following output.

Output

```

Labels:
value          label
  98             No Response
  99 NA-No response/Refused to K6

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      15.8064    0.1907  82.906 < 2e-16 ***
strata[T.Metro Calgary]  0.2902    0.2334   1.243  0.214049
strata[T.Other Alberta] -0.7994    0.2336  -3.421  0.000644 ***
sex[T.Female]     -0.4008    0.1909  -2.099  0.036001 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.295 on 1189 degrees of freedom
(14 observations deleted due to missingness)
Multiple R-squared:  0.02315, Adjusted R-squared:  0.02069
F-statistic: 9.394 on 3 and 1189 DF, p-value: 0.000003877

```

Note that we are explaining about 2% of the variance (Multiple R-squared: 0.02315). There are two coefficients for the `strata` variable. The reference category (the one left out) is the Metro Edmonton area. This would suggest that people in other parts of Alberta have get significantly less education than those in Metro Edmonton holding constant gender. Those in Metro Calgary get, on average, more education than those in Metro Edmonton holding constant gender, but that difference is not statistically different from zero. That is, we are not sufficiently confident that if we could survey everyone, that the average level of education in Edmonton and Calgary would be any different. The coefficient for `sex` suggests that women get, on average, less education than men, holding region constant.