POL314 Handbook

Introduction

This document describes how to use a statistical analysis package to conduct basic analysis on a database. Please skip material you already know.

Three main statistical packages are used in quantitative social science research. These are R, SPSS, and STATA. Since SPSS used to be very similar to STATA, but has recently been dramatically changed, this tutorial will only cover R and STATA.

The first section discusses the pros and cons of each program. The second section gives some basic reminders and definitions. The third section explains how to install R and use it for analysis, while the fourth section discusses analysis in STATA.

Questions about this document can be sent to samuel.baltz@utoronto.ca, but it's always a good idea to first consult Google, Stack Overflow, and Wikipedia.

1 Which package should I use?

This question is less important than it seems. R, SPSS, and STATA will all do what you want them to do.

SPSS and STATA are decades-long mainstays of quantitative analysis in political science. Any discussion of the merits of SPSS versus STATA is typically years or decades out of date, especially since SPSS has recently been dramatically overhauled. The relevant comparison is between STATA and R. Conventional wisdom is that STATA is made to be accessible, and is easier for basic analysis than R. If you will use unusually intricate statistical techniques in your project, you may have trouble executing them in R.

R has some advantages over STATA. Most importantly, it is open source and free. You can download R and do analysis at home, while you can only legally use STATA on campus computers. R also has a very nice user interface (a program which sits on top of R to make it better-looking and easier to use). Since R is community-developed, it also has many free packages, which provide useful functions, like very detailed correlations or the ability to load STATA files into R. Finally, it has an active community that provides strong online support, so it is very easy to Google questions and find good answers for them.

Hopefully this section has given you a clear idea of which software package you might prefer to use. Whether you choose STATA or R, this document will show you how to use it for basic analysis.

2 Some quick reminders

Each section will demonstrate the same analysis using one software package. There are three administrative tasks that come before statistical analysis. First I will show you how to download the software package (if that's possible), how to download the dataset, and how to choose the right directory. Then you will need to open your dataset, and you may wish to view your data. Once your dataset is set up, there are certain basic data maintenance techniques that you will need to understand. I will illustrate ten of these: in approximate order of complexity, these are **renaming variables**, **recoding variables**, **generating variables**, **displaying frequencies**, **crosstabulation**, **summarizing variables**, **if commands**, **identifying missing data**, **basic images**, and **very simple linear regressions**.

I will define each of these terms in this section. Please don't assume that these tools are necessary or sufficient for your project. This is just a general review. Finally, I strongly encourage you to skip definitions that you already know; otherwise, you risk getting bogged down right at the start.

Renaming a variable: Your dataset is organized into rows, each of which represents an individual person, and columns, each of which represents a certain variable. For example, if I am a respondent to your survey, I will have a row which records each of my answers to each of the questions, in order. Column 1 would be my answer to question 1, column 2 is my answer to question 2, etc. Each of these questions has a variable name associated with it, which records everyone's answer to that particular question. For example, the question "How old are you?" might reasonably be called age. However, datasets usually come with very confusing variable names. If you open a dataset and the age variable is called something like "AG0THRU99INCCANSURV", you will want to **rename the variable** to something like "Age". For each program, I will show you how to do that.

Recoding a variable: Responses are typically recorded in a database as natural numbers (numbers like 1, 2, 3, 999, etc). For example, an age variable might simply record the age of each respondent. A 20 year old will have the Age value "20", a 41 year old will have the Age value "41", and so forth. However, for certain analyses you might not care at all about the difference between a 41-year old and a 42-year old. Let's say instead you want 1 to mean "18-24", 2 to mean "25-30", and so forth. In this case, you may want to alter the values associated with the data in the database: this is what it means to **recode the variable**.

Generating a variable: You will frequently want to mess around with data in ways that would be very difficult to reverse. In this case, it is always best to **generate a new variable** that duplicates the variable you are interested in. That way, if something goes horribly wrong, you have only messed up a clone and still have the original version to fall back on.

Displaying a variable's frequency: To **display the frequency** of a variable is to view how many times each possible response was given. For example, the frequency of an age variable would be how many people said they were 18, how many said they were 19, how many said they were 20, and so on. Frequency is typically displayed in a histogram, which plots each possible response against the number of times that response was given. This shows you a visual frequency distribution.

Crosstabulation: **Crosstabulation** is a means of checking the frequency of two variables simultaneously. Let's say that you want to understand the relationship between age and social liberalism. Suppose that we have a question which we think measures social liberalism very well, where 1 is a very socially liberal answer, 2 is a fairly socially liberal answer, 3 is in the middle, 4 is fairly a socially conservative answer, and 5 is a very socially conservative answer. If we crosstabulated age with this variable, we would get a table showing how many times each respondent within the same age category answered the social liberalism question in a certain way. In other words, we would see how many 18 year olds are very socially liberal, how many 18 year olds are fairly socially liberal, how many 18 year olds are in the middle, how many 18 year olds are fairly socially conservative, how many 18 year olds are very socially conservative, how many 19 year olds are very socially liberal, and so on. This gives you a very preliminary sense for how social liberalism tracks with age.

Summarizing a variable: The function to **summarize a variable** gives basic descriptive information about that variable, like the type of variable (such as interval, ordinal, or text), the number of responses, and maybe the standard deviation and the mean.

If statements: Logical operators such as **if statements** are simply unavoidable in any sort of programmming (or, for that matter, in rational thought). Very often, you will want whatever you are doing to only apply to some of the data. In the crosstabulation example, say that you don't want to see every single age category; with an if command you can easily filter the results to just show people who are in a certain age range, or who answered a certain way on a different question.

Missing data: **Missing data** is a huge component of social science analysis. Often people cannot answer certain questions on a survey, and this is recorded as missing information. Identifying missing data and deciding how to treat it is a crucial part of any quantitative research project.

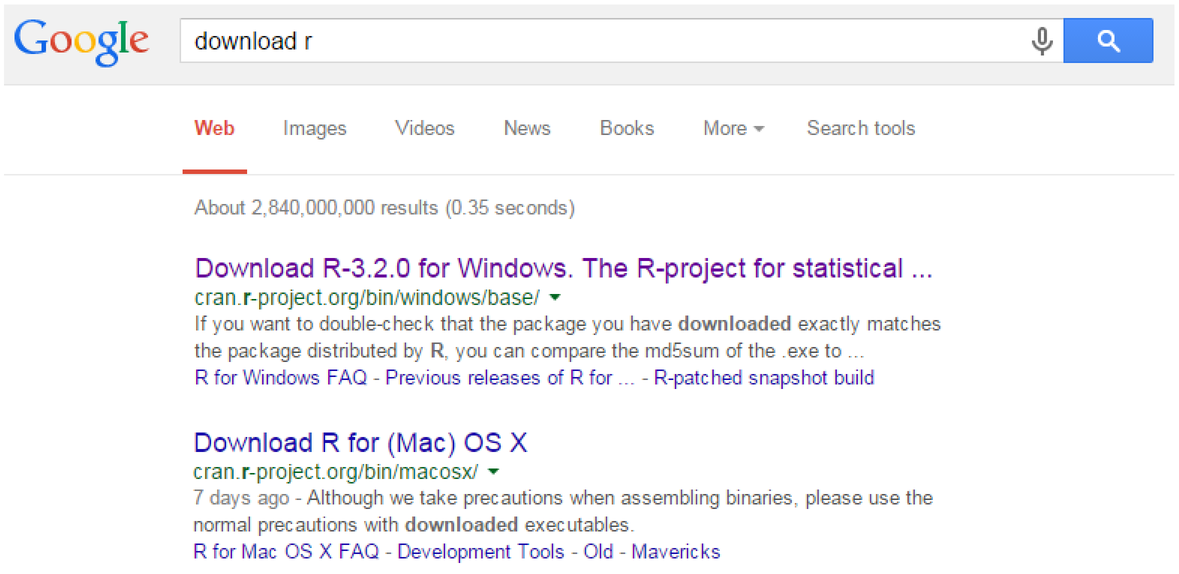
Basic images: Although the functionality is usually very limited, I will show you how to make some very **basic images** in each program.

Linear regressions: **Linear regressions** are a fundamental method in statistics, but there is not sufficient space to define "linear regression" here. Very simply, a linear regression estimates a line of best fit for a given distribution of data. If you are truly unfamiliar with the concept of a linear regression, search online or ask for assistance. In this document I will show you how to conduct a linear regression in each program.

3 R

This section will explain how to download R, load your database into it, and perform the statistical commands mentioned in the previous section.

3.1 Downloading R



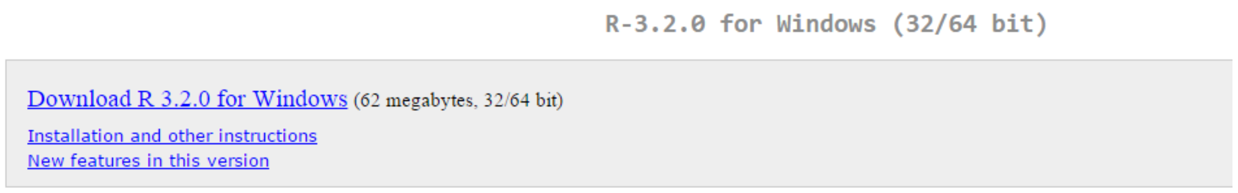
About 2,840,000,000 results (0.35 seconds)

Download R-3.2.0 for Windows. The R-project for statistical ...
cran.r-project.org/bin/windows/base/
If you want to double-check that the package you have **downloaded** exactly matches the package distributed by R, you can compare the md5sum of the .exe to ...
R for Windows FAQ - Previous releases of R for ... - R-patched snapshot build

Download R for (Mac) OS X
cran.r-project.org/bin/macosx/
7 days ago - Although we take precautions when assembling binaries, please use the normal precautions with **downloaded** executables.
R for Mac OS X FAQ - Development Tools - Old - Mavericks

Figure 1: Finding R online.

R is offered by the CRAN project, and you should simply download the latest distribution from that website. If you're using Windows, click on the Windows result from the CRAN project. For Mac, click on the Mac result for the CRAN project.

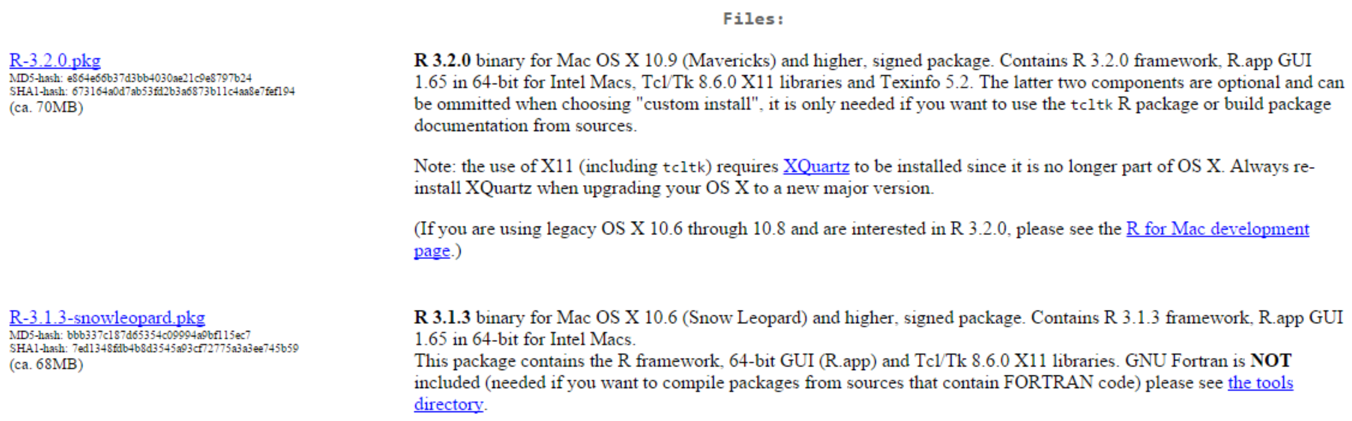


R-3.2.0 for Windows (32/64 bit)

Download R 3.2.0 for Windows (62 megabytes, 32/64 bit)
Installation and other instructions
New features in this version

Figure 2: The Windows download page for R.

For windows, simply click on the Download R button on the top of the page and install it the way you normally install a Windows program.



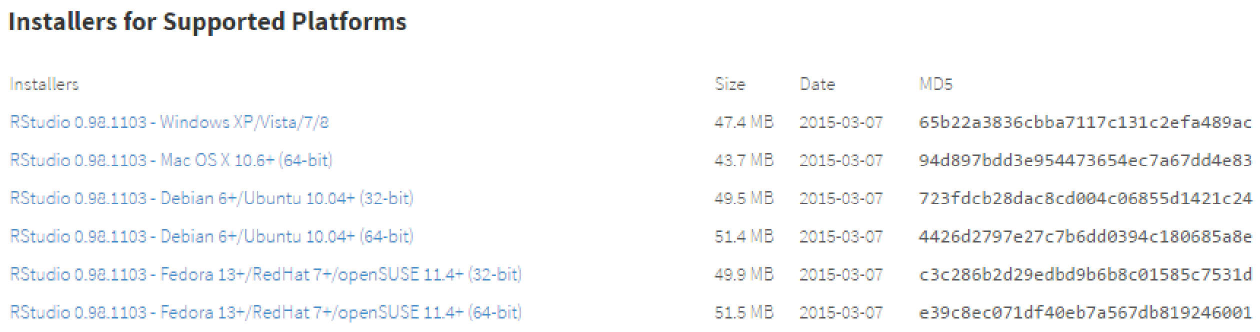
Files:

R-3.2.0.pkg MD5-hash: e864e66b37d3bb4030ae21c9e8797b24 SHA1-hash: 673164a0d7ab53fd2b3a6873b11c4aa8e7fef194 (ca. 70MB)	**R 3.2.0** binary for Mac OS X 10.9 (Mavericks) and higher, signed package. Contains R 3.2.0 framework, R.app GUI 1.65 in 64-bit for Intel Macs, Tcl/Tk 8.6.0 X11 libraries and Texinfo 5.2. The latter two components are optional and can be ommitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package or build package documentation from sources. Note: the use of X11 (including `tcltk`) requires XQuartz to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your OS X to a new major version. (If you are using legacy OS X 10.6 through 10.8 and are interested in R 3.2.0, please see the R for Mac development page.)
R-3.1.3-snowleopard.pkg MD5-hash: bbb337c187d65354c09994a9bf115ec7 SHA1-hash: 7ed1348fdb4b8d3545a93cf72775a3a3ee745b59 (ca. 68MB)	**R 3.1.3** binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.1.3 framework, R.app GUI 1.65 in 64-bit for Intel Macs. This package contains the R framework, 64-bit GUI (R.app) and Tcl/Tk 8.6.0 X11 libraries. GNU Fortran is **NOT** included (needed if you want to compile packages from sources that contain FORTRAN code) please see the tools directory.

Figure 3: The Mac download page for R.

Similarly, to install R on a Mac, simply select the most recent distribution that is appropriate for your Operating System and install it the same way that you normally install programs to your computer.

After R is installed, I highly recommend also installing RStudio. RStudio is a graphical user interface which sits on top of R and serves as a very nice-looking and intuitive intermediary between you and R. I will conduct the rest of this R tutorial using RStudio. To download RStudio, go to http://www.rstudio.com/products/rstudio/download/ and get the installer that is correct for your computer from the list in Figure 4.



Installers for Supported Platforms

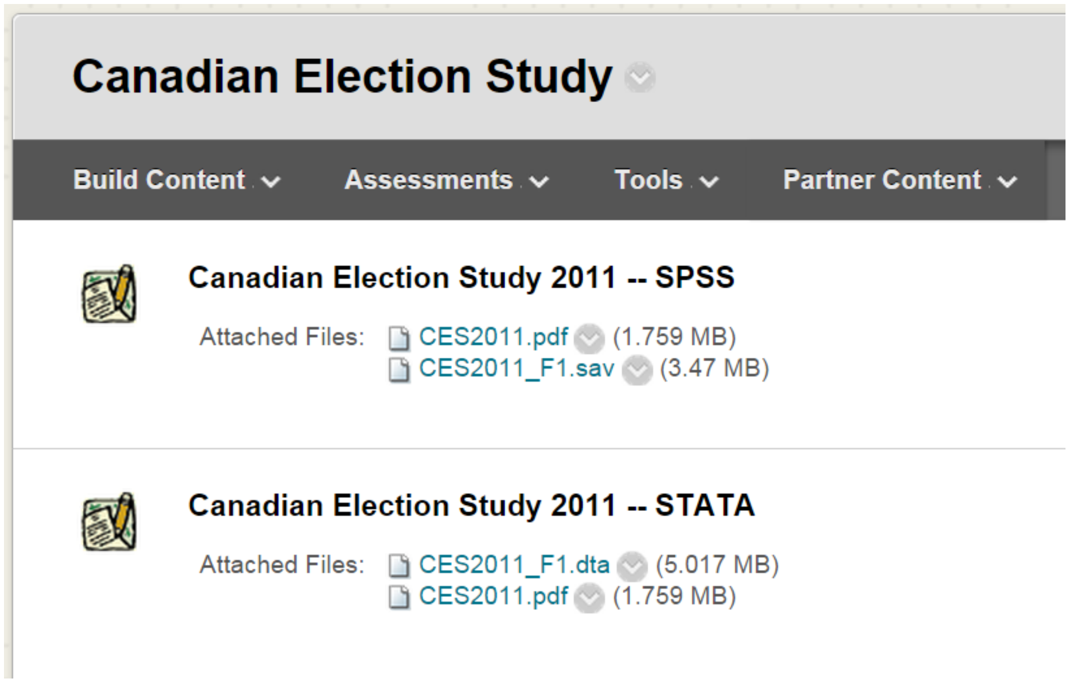
Installers	Size	Date	MD5
RStudio 0.98.1103 - Windows XP/Vista/7/8	47.4 MB	2015-03-07	65b22a3836cbba7117c131c2efa489ac
RStudio 0.98.1103 - Mac OS X 10.6+ (64-bit)	43.7 MB	2015-03-07	94d897bdd3e954473654ec7a67dd4e83
RStudio 0.98.1103 - Debian 6+/Ubuntu 10.04+ (32-bit)	49.5 MB	2015-03-07	723fdcb28dac8cd004c06855d1421c24
RStudio 0.98.1103 - Debian 6+/Ubuntu 10.04+ (64-bit)	51.4 MB	2015-03-07	4426d2797e27c7b6dd0394c180685a8e
RStudio 0.98.1103 - Fedora 13+/RedHat 7+/openSUSE 11.4+ (32-bit)	49.9 MB	2015-03-07	c3c286b2d29edbd9b6b8c01585c7531d
RStudio 0.98.1103 - Fedora 13+/RedHat 7+/openSUSE 11.4+ (64-bit)	51.5 MB	2015-03-07	e39c8ec071df40eb7a567db819246001

Figure 4: The download page for RStudio.

Note that I will also be using a specific color scheme in RStudio, where the background is black and the fonts are different colors. Don't worry if my screenshots look a little different from your version of RStudio.

3.2 Opening the dataset

Once you have RStudio, you are ready to open your database. Let's say that you have decided to use the 2011 Canadian Election Study. On Portal, go to Course Materials, then Datasets, then Canadian Election Study, and you'll see the view in Figure 5:



Canadian Election Study 2011 -- SPSS

Attached Files: CES2011.pdf (1.759 MB)
CES2011_F1.sav (3.47 MB)

Canadian Election Study 2011 -- STATA

Attached Files: CES2011_F1.dta (5.017 MB)
CES2011.pdf (1.759 MB)

Figure 5: The Datasets download page on Portal.

Download the .dta. Make sure to keep track of which folder you downloaded it to on your computer. For the purposes of this tutorial, I'm going to pretend I put it in the folder `` ``C:\example" ``, so whenever you see that file destination, replace it with whatever folder you actually downloaded the dataset to. Now, open RStudio.

In order to open a STATA file using R, we're going to have to download a new library so that R can figure out how to read another program's file type. Use the following commands:

```
> library(foreign)
> ces <- read.dta(``c:/example/CES2011_F1.dta'')
```

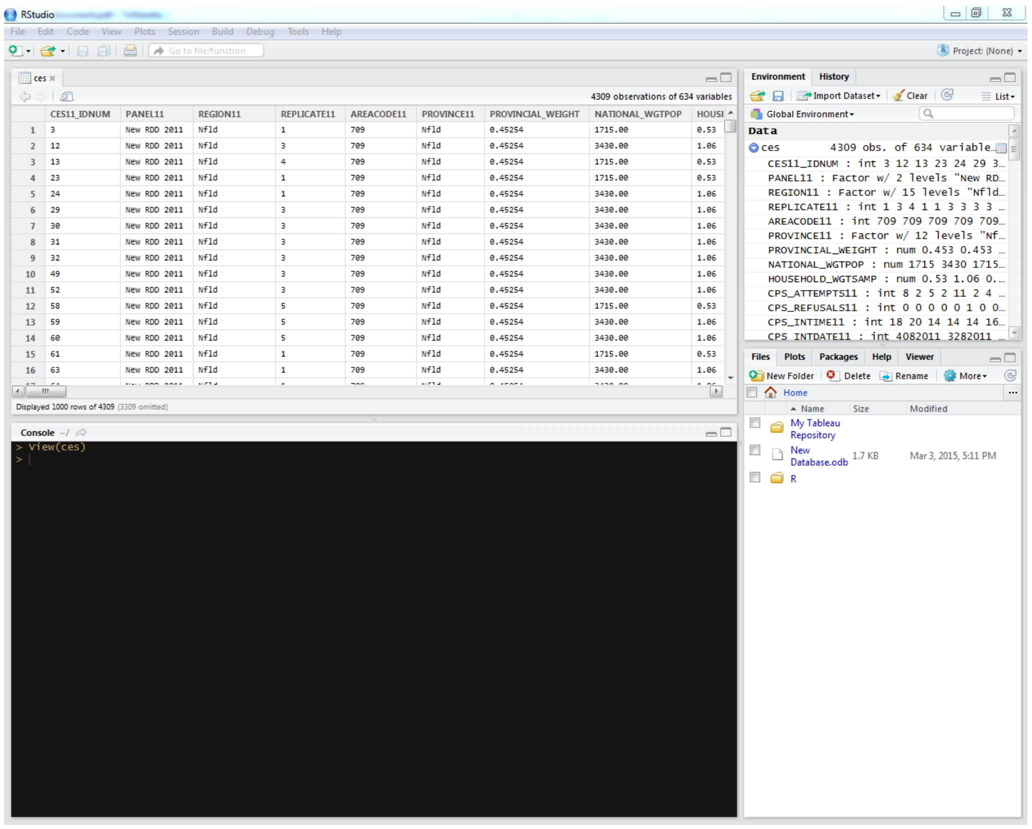
"ces" is just the name we're giving our dataset, so you can replace it with whatever string you wish to name your dataset. I will use the name "ces" to refer to the dataset for the remainder of this section. Once you issue these commands, the dataset will enter R's memory and you can begin to manipulate it.

3.3 Viewing your data

To look at your data as a grid, write the following:

```
> View(ces)
```

In RStudio, this will result in the view displayed in Figure 6. Now you can scroll through the columns and rows if you want to get a visual sense of what the data looks like. This can be very useful for quickly understanding what different variables mean and how all the objects in the data relate to one another. Note that, depending on the size of the dataset, it will likely not display every single variable.



4309 observations of 634 variables

	CES11_IDNUM	PANEL11	REGION11	REPLICATE11	AREACODE11	PROVINCE11	PROVINCIAL_WEIGHT	NATIONAL_WGTPOP	HOUSI
1	3	New RDD 2011	Nfld	1	709	Nfld	0.45254	1715.00	0.53
2	12	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
3	13	New RDD 2011	Nfld	4	709	Nfld	0.45254	1715.00	0.53
4	23	New RDD 2011	Nfld	1	709	Nfld	0.45254	1715.00	0.53
5	24	New RDD 2011	Nfld	1	709	Nfld	0.45254	3430.00	1.06
6	29	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
7	30	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
8	31	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
9	32	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
10	49	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
11	52	New RDD 2011	Nfld	3	709	Nfld	0.45254	3430.00	1.06
12	58	New RDD 2011	Nfld	5	709	Nfld	0.45254	1715.00	0.53
13	59	New RDD 2011	Nfld	5	709	Nfld	0.45254	3430.00	1.06
14	60	New RDD 2011	Nfld	5	709	Nfld	0.45254	3430.00	1.06
15	61	New RDD 2011	Nfld	1	709	Nfld	0.45254	1715.00	0.53
16	63	New RDD 2011	Nfld	1	709	Nfld	0.45254	3430.00	1.06

Displayed 1000 rows of 4309 (3309 omitted)

```
> View(ces)
>
```

```
Data
ces          4309 obs. of 634 variable…
  CES11_IDNUM : int 3 12 13 23 24 29 3…
  PANEL11 : Factor w/ 2 levels "New RD…
  REGION11 : Factor w/ 15 levels "Nfld…
  REPLICATE11 : int 1 3 4 1 1 3 3 3 3 …
  AREACODE11 : int 709 709 709 709 709…
  PROVINCE11 : Factor w/ 12 levels "Nf…
  PROVINCIAL_WEIGHT : num 0.453 0.453 …
  NATIONAL_WGTPOP : num 1715 3430 1715…
  HOUSEHOLD_WGTSAMP : num 0.53 1.06 0.…
  CPS_ATTEMPTS11 : int 8 2 5 2 11 2 4 …
  CPS_REFUSALS11 : int 0 0 0 0 0 1 0 0…
  CPS_INTIME11 : int 18 20 14 14 14 16…
  CPS_INTDATE11 : int 4082011 3282011 …
```

Name	Size	Modified
My Tableau Repository		
New Database.odb	1.7 KB	Mar 3, 2015, 5:11 PM
R		

Figure 6: The CES2011 data shown as a table in RStudio using the command View(ces).

3.4 Renaming variables

As with many things in R, there are countless ways to rename variables, and they all involve different tradeoffs. For this tutorial, I have chosen the simplest command that is actually changing the name of the variable. Let's take the example of CPS11_18, which asks respondents how they feel about the federal Conservative Party on a scale from 0 to 100. Quite sensibly, you may not want to use the confusing and arbitrary name "CPS11_18". To rename the variable to something that makes more sense, like "constherm" (shot for "conservative thermometer"), you can input

```
> names(ces)[names(ces)=="CPS11_18"] <- "constherm"
```

The dataset has a "names" attribute which stores all of the variable names. The above command has found every instance of "CPS11_18" in the names attribute and replaced it with "constherm", so that your database now has "constherm" everywhere that it used to have "CPS11_18". This is effectively renaming the variable.

Right now, this line of code probably looks very confusing and intimidating. But it's actually composed of little pieces that we'll learn individually in this tutorial. So for now, it's perfectly okay not to understand this line of code at all, but when you finish **§4.10**, come back and see if you can understand it.

3.5 Recoding variables

Perhaps you don't like that the constherm is on a scale from 0 to 100. That level of granularity can be very unwieldy, and you might decide that a scale from 0 to 10 captures all of the same essential information. Again, there are plenty of very clever ways to do this in R, but I have tried to offer the simplest and most readable syntax I can think of. In order to change constherm as described, you might use the following code:

```
> ces$constherm[0<=ces$constherm&ces$constherm<10] <- 0
> ces$constherm[10<=ces$constherm&ces$constherm<20] <- 1
> ces$constherm[20<=ces$constherm&ces$constherm<30] <- 2
> ces$constherm[30<=ces$constherm&ces$constherm<40] <- 3
> ces$constherm[40<=ces$constherm&ces$constherm<50] <- 4
> ces$constherm[50<=ces$constherm&ces$constherm<60] <- 5
> ces$constherm[60<=ces$constherm&ces$constherm<70] <- 6
> ces$constherm[70<=ces$constherm&ces$constherm<80] <- 7
> ces$constherm[80<=ces$constherm&ces$constherm<90] <- 8
> ces$constherm[90<=ces$constherm&ces$constherm<=100] <- 9
```

This syntax takes every thermometer score between 0 and 10 and assigns it to 0, then it takes every thermometer score between 10 and 20 and assigns it to 1, and so on. In this way, it collapses ranges of 10 into 1 number, which is exactly what we aimed to do. As a bonus exercise, you can try to find the two very awkward things about this code before reading the footnote[1].

3.6 Generating variables

Generating a new variable in R is mercifully easy. Let's say that I wanted a new variable that is exactly the same as constherm, so that I can fiddle with the conservative thermometer scores without messing up the original dataset. This is always a good idea. I can easily create such a variable by saying:

```
> consthermduplicate <- ces$constherm
```

Now consthermduplicate is exactly the same as constherm.

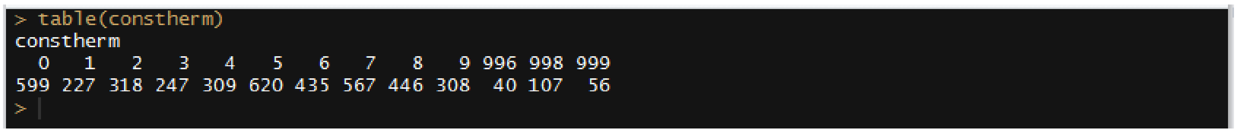
3.7 Displaying frequencies

Displaying a frequency is similarly trivial in R. Simply use the table function, as:

```
> table(ces$constherm)
```

This should result in the table seen in Figure 7.

[1]The first awkwardness is that the 9 category has more different responses assigned to it than every other category, since it has 90 to 100 inclusive while every other bin excludes the upper bound. The second awkwardness is that this syntax has absolutely nothing to say about missing data, which is a problem. As a bonus awkwardness, these 9 lines of code should really be 1 line of code, but that would make for a very confusing example.



```
> table(constherm)
constherm
  0   1   2   3   4   5   6   7   8   9 996 998 999 
599 227 318 247 309 620 435 567 446 308  40 107  56 
> 
```

Figure 7: A table showing the frequency distribution of the recoded conservative thermometer.

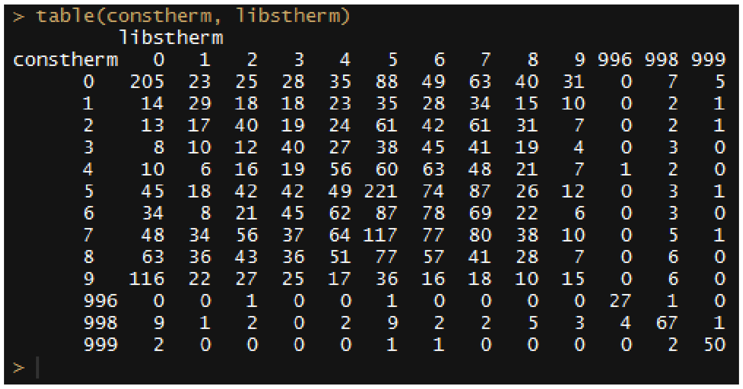
The table in Figure 7 means that 599 respondents are in the 0 bin (meaning they ranked the federal Conservative Party something between 0 and 9), 227 respondents are in the 1 bin, and so forth. This is a frequency distribution.

3.8 Crosstabulation

Displaying 2-way frequencies is barely more complicated than displaying the frequency distribution of a single variable in R. Let's say that I am interested in the relationship between support for the federal Conservative Party and support for the Liberal Party (I might boldly hypothesize that the more you like the Conservatives, the less likely you are to like the Liberals), and I would like to see the frequency distribution of the two variables together. Let's also say that I have set up a Liberal Party variable that is exactly equivalent to what we've done to constherm so far, called libstherm: it is a ranking from 0 to 10 of the Liberal Party[2]. Then I might write:

```
> table(ces$constherm, ces$libstherm)
```

This command should result in the table shown in Figure 8.



```
> table(constherm, libstherm)
         libstherm
constherm   0   1   2   3   4   5   6   7   8   9 996 998 999
      0   205  23  25  28  35  88  49  63  40  31   0   7   5
      1    14  29  18  18  23  35  28  34  15  10   0   2   1
      2    13  17  40  19  24  61  42  61  31   7   0   2   1
      3     8  10  12  40  27  38  45  41  19   4   0   3   0
      4    10   6  16  19  56  60  63  48  21   7   1   2   0
      5    45  18  42  42  49 221  74  87  26  12   0   3   1
      6    34   8  21  45  62  87  78  69  22   6   0   3   0
      7    48  34  56  37  64 117  77  80  38  10   0   5   1
      8    63  36  43  36  51  77  57  41  28   7   0   6   0
      9   116  22  27  25  17  36  16  18  10  15   0   6   0
      996   0   0   1   0   0   1   0   0   0   0  27   1   0
      998   9   1   2   0   2   9   2   2   5   3   4  67   1
      999   2   0   0   0   0   1   1   0   0   0   0   2  50
> 
```

Figure 8: A table showing the frequency distribution of the recoded conservative thermometer.

[2]if you feel like testing your knowledge so far, see if you can set it up and follow along.

This table is saying that 205 people are in the 0 bin for both parties, 14 people are in the 1 bin for the Conservatives and the 0 bin for the Liberals, 23 people are in the 0 bin for the Conservatives and the 1 bin for the Liberals, and so on. In other words, this table shows the two way frequency distribution between constherm and libstherm.

3.9 Summarizing variables

We can get various summary statistics about the Conservative party thermometer by typing

```
> summary(ces$constherm[ces$constherm<=9])
```

The bit in square brackets is meant to exclude non-answers (mostly those who don't know or refuse to answer), because otherwise the summary statistics would be tremendously wrong.

3.10 If commands

You have already seen a lot of if commands throughout this tutorial. This is the use of square brackets to choose portions of the data. For an example, see the previous subsection; a human might read that command as "summarize constherm if constherm is less than or equal to 9". This was useful because non-answers, and only non-answers, have a value greater than 9 in constherm.

3.11 Identifying missing data

This is a particularly easy task in R. To find all the missing data in constherm, we can simply execute

```
> is.na(constherm)
```

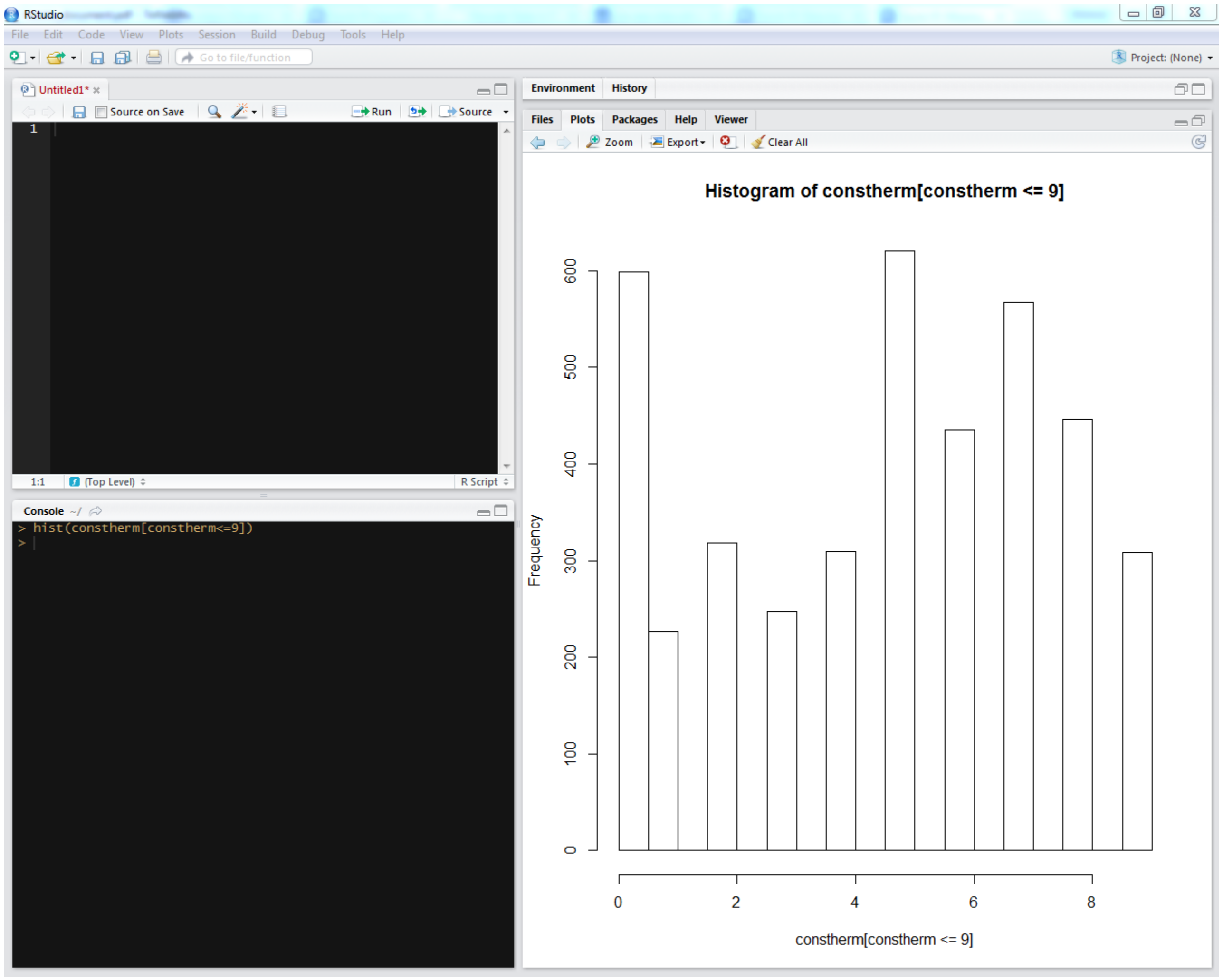
This is not a very useful way of looking for missing data, since it will print an array equal to the length of constherm with a "FALSE" everywhere that constherm has a value and a "TRUE" everywhere that constherm does not have a value, but by combining this basic syntax with if commands you can achieve very powerful results.

3.12 Basic images

Say that we don't like text tables very much because they can be very hard to read, so we want to actually see a frequency distribution of constherm. We can simply execute

```
> hist(ces$constherm[ces$constherm<=9])
```

RStudio has a built in image environment where this histogram will suddenly appear, as shown in Figure 9.



```
> hist(constherm[constherm<=9])
>
```

Figure 9: A histogram showing the frequency distribution of the recoded conservative thermometer.

This graph is not very good, but it at least gives us a visual sense for the variance of constherm. With a little searching, you can try to experiment with changing the size, adding colors, and adding labels.

3.13 Simple linear regression

It is well outside the bounds of this tutorial to truly explain what a linear regression is, but I can show you as an example how to do a linear regression on three hypothetical variables, assuming that you already know what a linear regression is and why you might want to conduct one. So, consider three variables x, y, and z. We can examine how z responds to independent variables x and y using the syntax

```
> lm(z ~ x + y)
```

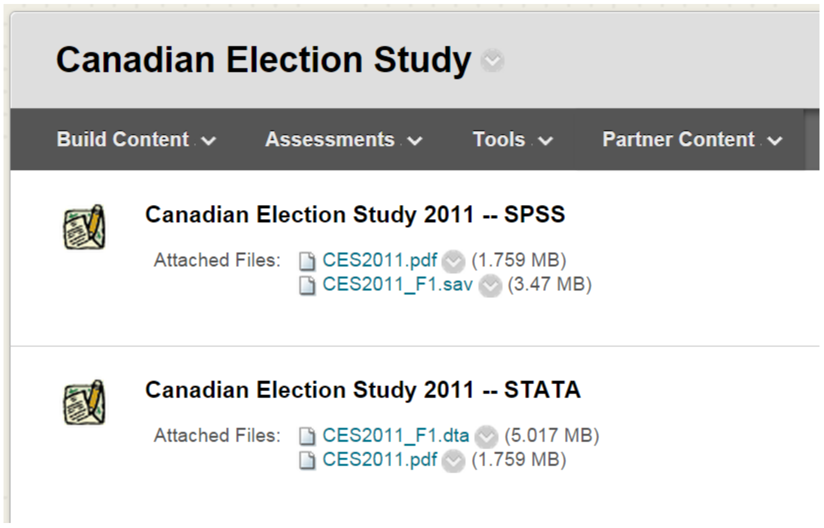
4 STATA

4.1 Finding STATA

You can't download STATA, but it is available to you on campus. The political science computer lab on the third floor of Sidney Smith has several copies of STATA, and you can look for it on the computers there.

4.2 Opening the dataset

Opening the dataset is much easier in STATA than in R. Just download the .dta



Canadian Election Study 2011 -- SPSS

Attached Files: CES2011.pdf (1.759 MB)
CES2011_F1.sav (3.47 MB)

Canadian Election Study 2011 -- STATA

Attached Files: CES2011_F1.dta (5.017 MB)
CES2011.pdf (1.759 MB)

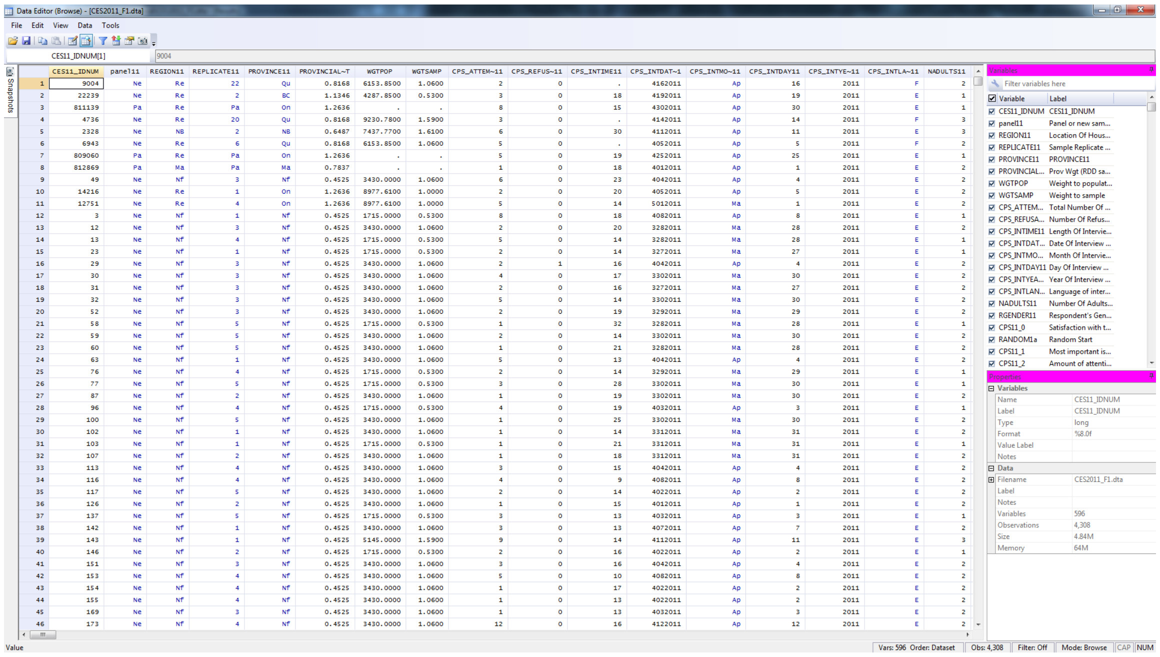
Figure 10

and double click on it. Alternatively, you can open STATA, go to the correct directory (say, "C:\example") by typing

```
cd "c:\example"
use CES2011_F1.dta
```

4.3 Viewing your data

To view the data, click on Data in the toolbar, then mouseover Data Editor, then click Data Editor (Browse)


Data Editor (Browse) - [CES2011_F1.dta]
File Edit View Data Tools
CES11_IDNUM[1]
9004
Snapshots

	CES11_IDNUM	panel11	REGION11	REPLICATE11	PROVINCE11	PROVINCIAL~T	WGTPOP	WGTSAMP	CPS_ATTEM~11	CPS_REFUS~11	CPS_INTIME11	CPS_INTDAT~1	CPS_INTMO~11	CPS_INTDAY11	CPS_INTYE~11	CPS_INTLA~11	NADULTS11
1	9004	Ne	Re	22	Qu	0.8168	6153.8500	1.0600	2	0	.	4162011	Ap	16	2011	F	2
2	22239	Ne	Re	2	BC	1.1346	4287.8500	0.5300	3	0	18	4192011	Ap	19	2011	E	1
3	811139	Pa	Re	Pa	On	1.2636	.	.	8	0	15	4302011	Ap	30	2011	E	1
4	4736	Ne	Re	20	Qu	0.8168	9230.7800	1.5900	3	0	.	4142011	Ap	14	2011	F	3
5	2328	Ne	NB	2	NB	0.6487	7437.7700	1.6100	6	0	30	4112011	Ap	11	2011	E	3
6	6943	Ne	Re	6	Qu	0.8168	6153.8500	1.0600	5	0	.	4052011	Ap	5	2011	F	2
7	809060	Pa	Re	Pa	On	1.2636	.	.	5	0	19	4252011	Ap	25	2011	E	1
8	812869	Pa	Ma	Pa	Ma	0.7837	.	.	1	0	18	4012011	Ap	1	2011	E	2
9	49	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	6	0	23	4042011	Ap	4	2011	E	2
10	14216	Ne	Re	1	On	1.2636	8977.6100	1.0000	2	0	20	4052011	Ap	5	2011	E	2
11	12751	Ne	Re	4	On	1.2636	8977.6100	1.0000	5	0	14	5012011	Ma	1	2011	E	2
12	3	Ne	Nf	1	Nf	0.4525	1715.0000	0.5300	8	0	18	4082011	Ap	8	2011	E	1
13	12	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	2	0	20	3282011	Ma	28	2011	E	2
14	13	Ne	Nf	4	Nf	0.4525	1715.0000	0.5300	5	0	14	3282011	Ma	28	2011	E	1
15	23	Ne	Nf	1	Nf	0.4525	1715.0000	0.5300	2	0	14	3272011	Ma	27	2011	E	1
16	29	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	2	1	16	4042011	Ap	4	2011	E	2
17	30	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	4	0	17	3302011	Ma	30	2011	E	2
18	31	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	2	0	16	3272011	Ma	27	2011	E	2
19	32	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	5	0	14	3302011	Ma	30	2011	E	2
20	52	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	2	0	19	3292011	Ma	29	2011	E	2
21	58	Ne	Nf	5	Nf	0.4525	1715.0000	0.5300	1	0	32	3282011	Ma	28	2011	E	1
22	59	Ne	Nf	5	Nf	0.4525	3430.0000	1.0600	2	0	14	3302011	Ma	30	2011	E	2
23	60	Ne	Nf	5	Nf	0.4525	3430.0000	1.0600	1	0	21	3282011	Ma	28	2011	E	2
24	63	Ne	Nf	1	Nf	0.4525	3430.0000	1.0600	5	0	13	4042011	Ap	4	2011	E	2
25	76	Ne	Nf	4	Nf	0.4525	1715.0000	0.5300	2	0	14	3292011	Ma	29	2011	E	1
26	77	Ne	Nf	5	Nf	0.4525	1715.0000	0.5300	3	0	28	3302011	Ma	30	2011	E	1
27	87	Ne	Nf	2	Nf	0.4525	3430.0000	1.0600	1	0	19	3302011	Ma	30	2011	E	2
28	96	Ne	Nf	4	Nf	0.4525	1715.0000	0.5300	4	0	19	4032011	Ap	3	2011	E	1
29	100	Ne	Nf	5	Nf	0.4525	3430.0000	1.0600	1	0	25	3302011	Ma	30	2011	E	2
30	102	Ne	Nf	1	Nf	0.4525	3430.0000	1.0600	1	0	14	3312011	Ma	31	2011	E	2
31	103	Ne	Nf	1	Nf	0.4525	1715.0000	0.5300	1	0	21	3312011	Ma	31	2011	E	1
32	107	Ne	Nf	2	Nf	0.4525	3430.0000	1.0600	1	0	18	3312011	Ma	31	2011	E	2
33	113	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	3	0	15	4042011	Ap	4	2011	E	2
34	116	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	4	0	9	4082011	Ap	8	2011	E	2
35	117	Ne	Nf	5	Nf	0.4525	3430.0000	1.0600	2	0	14	4022011	Ap	2	2011	E	2
36	126	Ne	Nf	2	Nf	0.4525	3430.0000	1.0600	1	0	15	4012011	Ap	1	2011	E	2
37	137	Ne	Nf	5	Nf	0.4525	1715.0000	0.5300	3	0	13	4032011	Ap	3	2011	E	1
38	142	Ne	Nf	1	Nf	0.4525	3430.0000	1.0600	3	0	13	4072011	Ap	7	2011	E	2
39	143	Ne	Nf	1	Nf	0.4525	5145.0000	1.5900	9	0	14	4112011	Ap	11	2011	E	3
40	146	Ne	Nf	2	Nf	0.4525	1715.0000	0.5300	2	0	16	4022011	Ap	2	2011	E	1
41	151	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	3	0	16	4042011	Ap	4	2011	E	2
42	153	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	5	0	10	4082011	Ap	8	2011	E	2
43	154	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	1	0	17	4022011	Ap	2	2011	E	2
44	155	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	1	0	13	4022011	Ap	2	2011	E	2
45	169	Ne	Nf	3	Nf	0.4525	3430.0000	1.0600	1	0	13	4032011	Ap	3	2011	E	2
46	173	Ne	Nf	4	Nf	0.4525	3430.0000	1.0600	12	0	16	4122011	Ap	12	2011	E	2

Variables

Filter variables here

Variable	Label
CES11_IDNUM	CES11_IDNUM
panel11	Panel or new sam...
REGION11	Location Of Hous...
REPLICATE11	Sample Replicate ...
PROVINCE11	PROVINCE11
PROVINCIAL...	Prov Wgt (RDD sa...
WGTPOP	Weight to populat...
WGTSAMP	Weight to sample
CPS_ATTEM...	Total Number Of ...
CPS_REFUSA...	Number Of Refus...
CPS_INTIME11	Length Of Intervie...
CPS_INTDAT...	Date Of Interview ...
CPS_INTMO...	Month Of Intervie...
CPS_INTDAY11	Day Of Interview ...
CPS_INTYEA...	Year Of Interview ...
CPS_INTLAN...	Language of inter...
NADULTS11	Number Of Adults...
RGENDER11	Respondent's Gen...
CPS11_0	Satisfaction with t...
RANDOM1a	Random Start
CPS11_1	Most important is...
CPS11_2	Amount of attenti...

Properties

Variables	
Name	CES11_IDNUM
Label	CES11_IDNUM
Type	long
Format	%8.0f
Value Label	
Notes	
Data	
Filename	CES2011_F1.dta
Label	
Notes	
Variables	596
Observations	4,308
Size	4.84M
Memory	64M

Value

Vars: 596 Order: Dataset | Obs: 4,308 | Filter: Off | Mode: Browse | CAP | NUM

Figure 11

4.4 Renaming variables

As in **§3.4**, suppose that we want to rename the variable "CPS11_18", which asks respondents how they feel about the federal Conservative Party on a scale from 0 to 100, to something which actually describes what the variable contains. To rename it to "constherm" (short for "conservative thermometer"), input:

```
> rename CPS11_18 constherm
```

4.5 Recoding variables

Just as in **§4.5**, you may want to turn constherm from a 0 to 100 scale into a scale from 0 to 10.

```
replace constherm = 0 if constherm >= 0 & constherm < 10
replace constherm = 1 if constherm >= 10 & constherm < 20
replace constherm = 2 if constherm >= 20 & constherm < 30
replace constherm = 3 if constherm >= 30 & constherm < 40
replace constherm = 4 if constherm >= 40 & constherm < 50
replace constherm = 5 if constherm >= 50 & constherm < 60
replace constherm = 6 if constherm >= 60 & constherm < 70
replace constherm = 7 if constherm >= 70 & constherm < 80
replace constherm = 8 if constherm >= 80 & constherm < 90
replace constherm = 9 if constherm >= 90 & constherm <= 100
```

This syntax takes every thermometer score between 0 and 10 and assigns it to 0, then it takes every thermometer score between 10 and 20 and assigns it to 1, and so on. In this way, it collapses each range of 10 into just 1 number, which is exactly what we aimed to do. As a bonus exercise, you can try to find two awkward things about this code. The answer is in the footnote in **§3.5**.

4.6 Generating variables

Let's say that I want a new variable that is exactly the same as constherm, so that I can fiddle with the conservative thermometer scores without messing up the original dataset. This is always a good idea. I can easily make this duplicate variable by typing:

```
> gen consthermduplicate = constherm
```

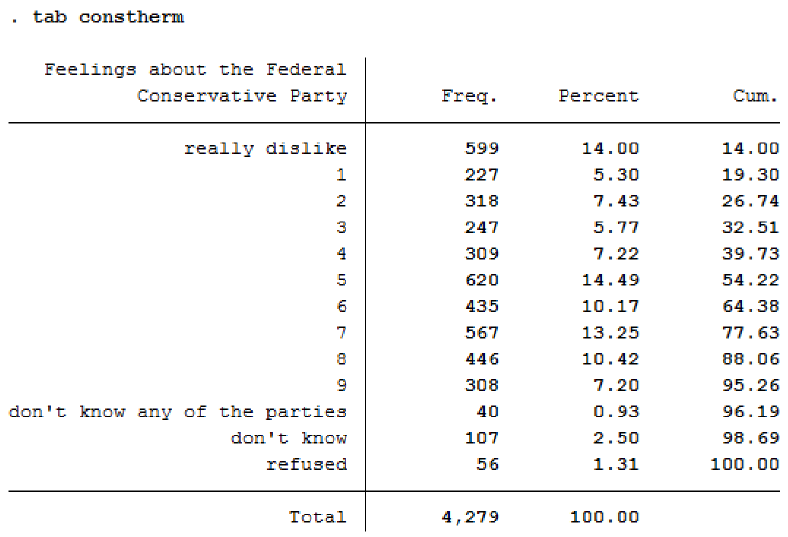
Then "consthermduplicate" contains exactly the same information as constherm.

4.7 Displaying frequencies

To see the frequency of our constherm variable, just type:

```
tab constherm
```

This should result in the table seen in Figure 12.



```
. tab constherm

   Feelings about the Federal |
            Conservative Party |      Freq.     Percent        Cum.
------------------------------+-----------------------------------
                really dislike |        599       14.00       14.00
                             1 |        227        5.30       19.30
                             2 |        318        7.43       26.74
                             3 |        247        5.77       32.51
                             4 |        309        7.22       39.73
                             5 |        620       14.49       54.22
                             6 |        435       10.17       64.38
                             7 |        567       13.25       77.63
                             8 |        446       10.42       88.06
                             9 |        308        7.20       95.26
don't know any of the parties |         40        0.93       96.19
                    don't know |        107        2.50       98.69
                       refused |         56        1.31      100.00
------------------------------+-----------------------------------
                         Total |      4,279      100.00
```

Figure 12: A table showing the frequency distribution of the recoded conservative thermometer.

The table in Figure 6 means that 599 respondents are in the 0 bin (meaning they ranked the federal Conservative Party something between 0 and 9), 227 respondents are in the 1 bin, and so forth. This is a frequency distribution.

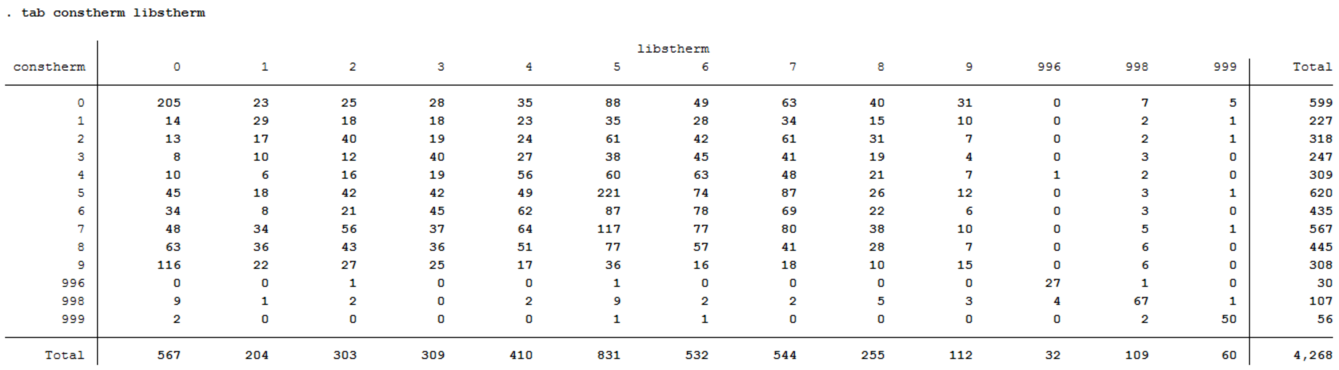
4.8 Crosstabulation

Displaying 2-way frequencies is barely more complicated than displaying the frequency distribution of a single variable in STATA. Let's say that I am interested in the relationship between support for the federal Conservative Party and support for the Liberal Party (I might boldly hypothesize that the more you like the Conservatives, the less likely you are to like the Liberals), and I would like to see the frequency distribution of the two variables together. Let's also say that I have set up a Liberal Party variable that is exactly equivalent to what we've done to constherm so far, called libstherm: it is a ranking from 0 to 10 of the Liberal Party[3]. Then I might write the following:

```
> table(ces$constherm, ces$libstherm)
```

This command should yield the table shown in Figure 13.

[3]if you feel like testing your knowledge so far, see if you can set it up and follow along.



```
. tab constherm libstherm

           |                                                libstherm
 constherm |         0          1          2          3          4          5          6          7          8          9        996        998        999 |     Total
-----------+-----------------------------------------------------------------------------------------------------------------------------------------------------+----------
         0 |       205         23         25         28         35         88         49         63         40         31          0          7          5 |       599
         1 |        14         29         18         18         23         35         28         34         15         10          0          2          1 |       227
         2 |        13         17         40         19         24         61         42         61         31          7          0          2          1 |       318
         3 |         8         10         12         40         27         38         45         41         19          4          0          3          0 |       247
         4 |        10          6         16         19         56         60         63         48         21          7          1          2          0 |       309
         5 |        45         18         42         42         49        221         74         87         26         12          0          3          1 |       620
         6 |        34          8         21         45         62         87         78         69         22          6          0          3          0 |       435
         7 |        48         34         56         37         64        117         77         80         38         10          0          5          1 |       567
         8 |        63         36         43         36         51         77         57         41         28          7          0          6          0 |       445
         9 |       116         22         27         25         17         36         16         18         10         15          0          6          0 |       308
       996 |         0          0          1          0          0          1          0          0          0          0         27          1          0 |        30
       998 |         9          1          2          0          2          9          2          2          5          3          4         67          1 |       107
       999 |         2          0          0          0          0          1          1          0          0          0          0          2         50 |        56
-----------+-----------------------------------------------------------------------------------------------------------------------------------------------------+----------
     Total |       567        204        303        309        410        831        532        544        255        112         32        109         60 |     4,268
```

Figure 13: A table showing the frequency distribution of the recoded conservative thermometer.

This table is saying that 205 people are in the 0 bin for both parties, 14 people are in the 1 bin for the Conservatives and the 0 bin for the Liberals, 23 people are in the 0 bin for the Conservatives and the 1 bin for the Liberals, and so on. In other words, this table shows the two way frequency distribution between constherm and libstherm.

4.9 Summarizing variables

We can get various summary statistics about the Conservative party thermometer by typing

```
summarize constherm
```

However, be careful! We have included the non-response values, 996, 998, and 999, which mean either "Refused" or "Don't know". So if we include these, both the mean and standard deviation will be much higher than they actually are. We will need to recode those values, or generate a new variable without those values. We'll cover how to do that in the next section.

4.10 If commands

To recode those values so that they won't throw off our summary statistics, we might want to recode them to missing values. To do this, we need a command that says: if the variable is bigger than 9 (which covers 996, 998, and 999), then replace it with a missing value.

```
replace constherm = .  if constherm > 9
```

4.11 Identifying missing data

Missing data is identified by a '.' in STATA. So, for example, we could see the rating given to the Liberal Party by those who did not rate the Conservative Party by typing:

```
tab libstherm if constherm == .
```

4.12 Basic images

STATA does very basic images, which can help to understand the distribution of a variable. Unfortunately, STATA may not be able to produce a presentable image to show to other people. We could see a distribution of Conservative Party ratings by typing:

```
hist constherm
```

And that histogram looks like this:

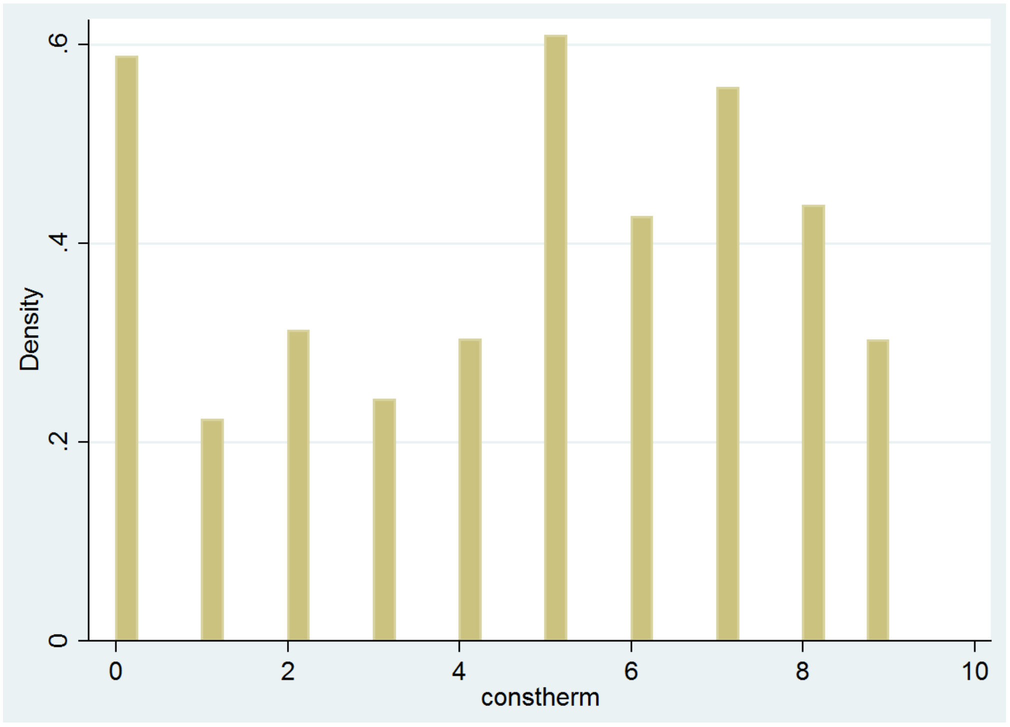


Figure 14: A histogram of the recoded conservative thermometer.

4.13 Simple linear regression

It is well outside the bounds of this tutorial to truly explain what a linear regression is, but I can show you as an example how to do a linear regression on three hypothetical variables, assuming that you already know what a linear regression is and why you might want to conduct one. So, consider three variables x, y, and z. We can examine how z responds to independent variables x and y using the syntax

```
> regress z x y
```