Explorations: Conducting Empirical Research in Canadian Political Science (4th Edition)

Stata Handbook[1]

Jason Roy and Loleen Berdahl

Welcome to the *Explorations: Conducting Empirical Research in Canadian Political Science Stata Handbook!* In this handbook, we provide you with a basic introduction to Stata. The procedures outlined follow from the statistical methods described in *Explorations: Conducting Empirical Research in Canadian Political Science (4th Edition)*. We encourage you to work closely with the textbook as you move through this handbook; here we cover the technical "how to" of basic statistics, but we do not cover the critical issues of which statistics to use when. We use 2019 Canadian Election Study (CES) data. As we explain throughout the textbook, the CES data are a great publicly available resource for studying politics in Canada. We encourage you to practice the techniques outlined in this handbook with the CES datasets.

Please note that the screen shots included are those captured working with Stata 16.1 on a Mac. The appearance of the Stata work environment may differ across operating systems and versions of Stata. Open your copy of Stata and let's begin!

Part I: Getting Comfortable with Stata

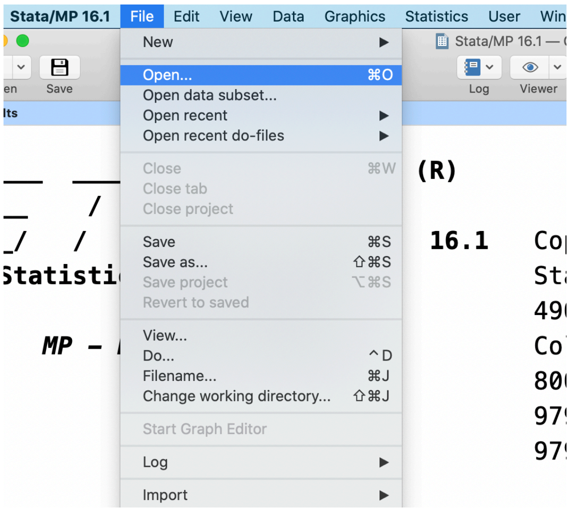
Destination

By the end of this section, you will be able to:

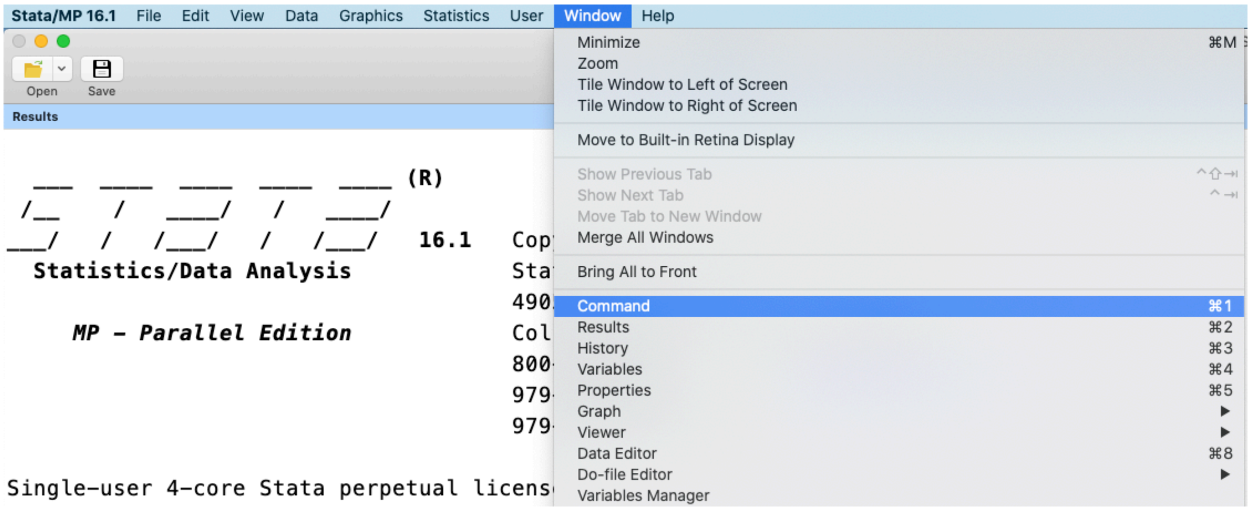
- *open a dataset in Stata;*
- *navigate between Stata windows;*
- *explain what a do-file is, and why it is a valuable tool for researchers;*
- *record your work in Stata; and*
- *use search functions*

Like many other software programs, users can open Stata datasets via the **File – Open** option on the top menu bar. To follow the procedures below, download and open the 2019 CES dataset (we use the telephone survey), available for free at https://doi.org/10.7910/DVN/8RHLG1. You should also download the 2019 CES technical documentation and codebooks for reference.

[1] We wish to acknowledge Stuart Soroka's Poli 618 Stata training manual (McGill University, 2010), which inspired this work.



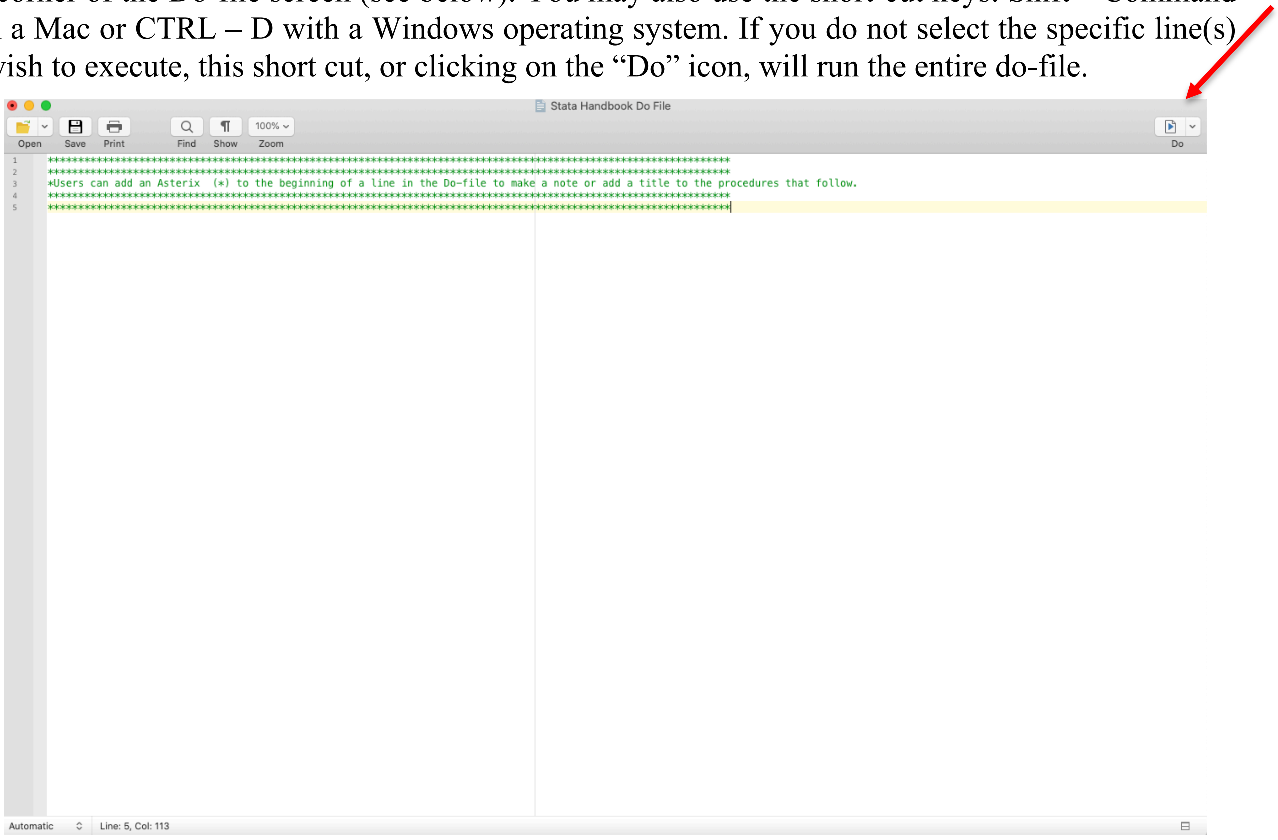
The Stata work environment consists of a number of windows/screens, each with different information. (Note: The windows that are displayed when you first open Stata will vary according to the version of the program that you are using and any user settings that have been applied.) To open any of the windows/screens listed below, select the window you wish to access from the top menu bar under "Window". For example, to open the **Command** window select *Window* > *Command* from the drop-down menu (note that this window is already visible when you open Stata 16.1) .



Using the dropdown menu, let's explore the Stata windows in the order they are presented within Stata.

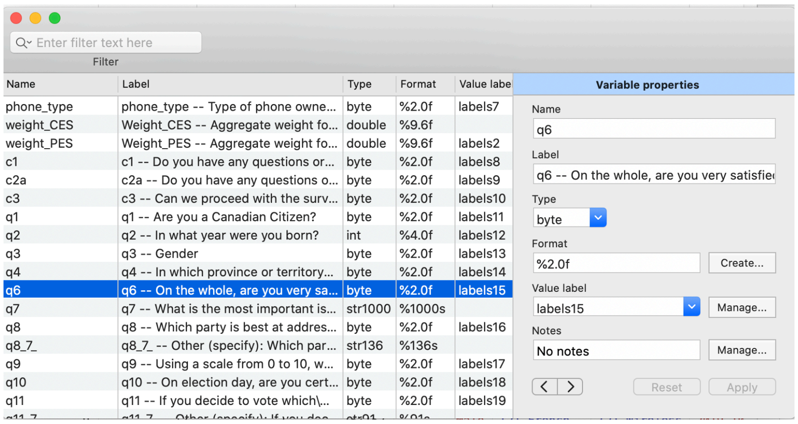
- The **Command window** is where all your commands are typed (unless you are using a do-file, as we will discuss shortly).
- The **Results window** displays results. It can display only a limited number of lines at a time. If your results are going to be very long, use a **log file** (see below).
- The **History (Review) window** records all commands (from the command window or do file) as they are entered. You can click on an old command, and it will appear again in the command window.
- The **Variables window** lists all variables in the working file. Click on a variable, and it will appear in the command window.
- The **Properties window** provides information for variables selected in the variables window as well as information on the dataset (e.g. number of variables, number of cases, etc.).

- The **Graph window** displays any open graphs.
- The **Viewer window** displays information such as the results from a search of the help files (see below).
- The **Data Editor window** allows you to enter, view, or edit your data file. It looks like a spreadsheet. Variables are listed across the top (columns), and cases are listed down the side (rows).
- The **Do-file Editor window** is a workspace where you can write, edit, and save Stata commands. Rather than entering these commands in the command window, you can run them from the do-file editor. The advantage is that you can easily edit, save and re-run all your commands. We strongly recommend working with a Do-file in Stata. Doing so allows you to record all of the procedures that you run and easily re-produce all results as needed. It also allows you to easily collaborate with colleagues working on the same project with you, and to share a history of your analysis with others, thus increasing research transparency. (See Chapter 3 for a discussion of ethics and data analysis.) Users can add an Asterix (*) to the beginning of a line in the Do-file to make a note or add a title to the procedures that follow. To run a command from the Do-file, select the line and then click on the "Do" icon in the top right corner of the Do-file screen (see below). You may also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system. If you do not select the specific line(s) you wish to execute, this short cut, or clicking on the "Do" icon, will run the entire do-file.



```
****************************************************************************************************************
****************************************************************************************************************
*Users can add an Asterix  (*) to the beginning of a line in the Do-file to make a note or add a title to the procedures that follow.
****************************************************************************************************************
****************************************************************************************************************
```

- The **Variable Manager window** displays all of the variables in the dataset along with the variable properties. To view the properties of a variable, select the variable from the list.



Name	Label	Type	Format	Value labe
phone_type	phone_type -- Type of phone owne...	byte	%2.0f	labels7
weight_CES	Weight_CES -- Aggregate weight fo...	double	%9.6f	
weight_PES	Weight_PES -- Aggregate weight fo...	double	%9.6f	labels2
c1	c1 -- Do you have any questions or...	byte	%2.0f	labels8
c2a	c2a -- Do you have any questions o...	byte	%2.0f	labels9
c3	c3 -- Can we proceed with the surv...	byte	%2.0f	labels10
q1	q1 -- Are you a Canadian Citizen?	byte	%2.0f	labels11
q2	q2 -- In what year were you born?	int	%4.0f	labels12
q3	q3 -- Gender	byte	%2.0f	labels13
q4	q4 -- In which province or territory...	byte	%2.0f	labels14
q6	q6 -- On the whole, are you very sa...	byte	%2.0f	labels15
q7	q7 -- What is the most important is...	str1000	%1000s	
q8	q8 -- Which party is best at addres...	byte	%2.0f	labels16
q8_7_	q8_7_ -- Other (specify): Which par...	str136	%136s	
q9	q9 -- Using a scale from 0 to 10, w...	byte	%2.0f	labels17
q10	q10 -- On election day, are you cert...	byte	%2.0f	labels18
q11	q11 -- If you decide to vote which\...	byte	%2.0f	labels19

Name: q6

Label: q6 -- On the whole, are you very satisfie

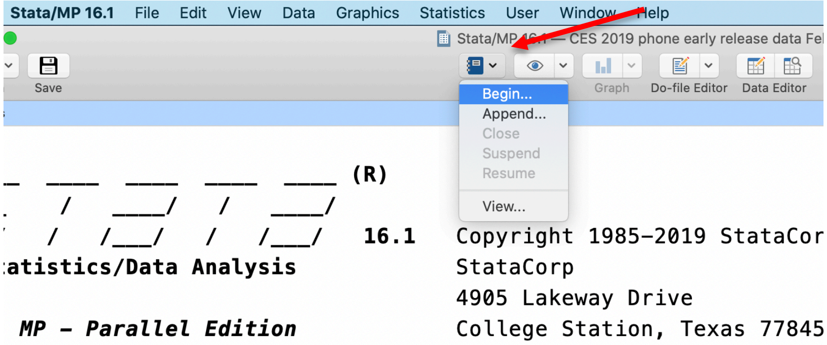
Type: byte

Format: %2.0f

Value label: labels15

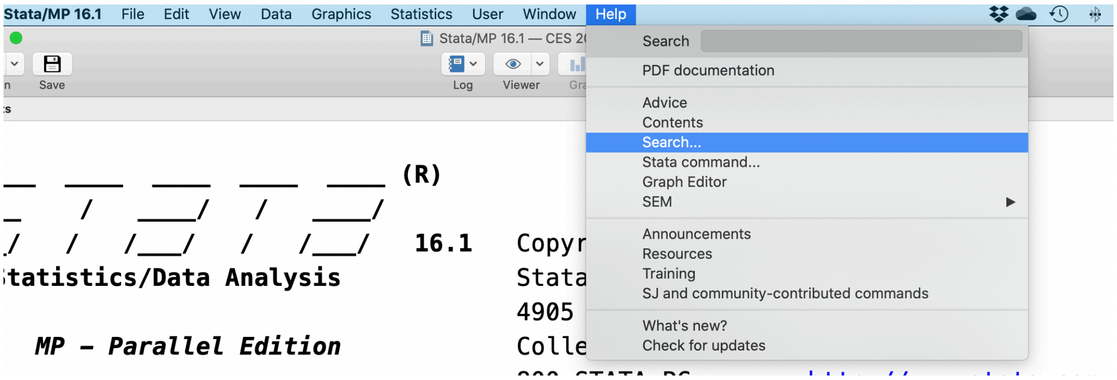
Notes: No notes

Recording your work with log files: Stata allows you to record everything from the **Results** window – commands and results – in a log file. To start a new log file, select *File* > *Log* > *Begin* from the drop-down menu, or click on the Log icon (see below). Name the file, select a location to save it, and click *Save*. The log file is now recording everything you do in Stata. Note that you can view log files by following the same steps, only select *View* instead of *Begin*. Finally, you can add to an existing log file by following the same set of procedures, only in this instance, select *Append*.

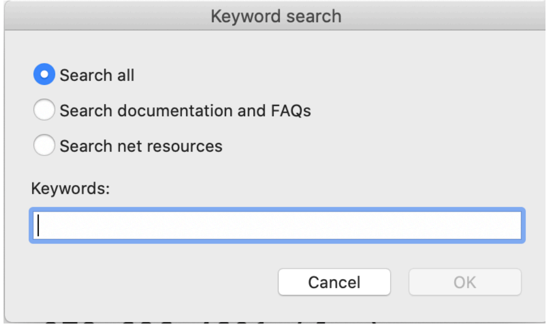


```
_  ____  ____  ____  ____ (R)
.   /   ____/   /   ____/
'  /   /___/   /   /___/    16.1   Copyright 1985-2019 StataCor
:atistics/Data Analysis            StataCorp
                                   4905 Lakeway Drive
 MP - Parallel Edition             College Station, Texas 77845
```

Using Search functions: One of the strengths of the Stata program is that it uses relatively simple and intuitive syntax. We provide the syntax necessary to run the procedures outlined in this handbook. However, because the syntax presented here is only a small sample of what can be done in Stata, you should familiarize yourself with the **help** and **search** features in Stata. These are invaluable resources for Stata commands, including sample syntax, with detailed explanation of the various procedures and options available. You can access the Stata help features by using the drop-down *Help* menu at the top of the screen. Selecting *Search* from the drop-down menu will open a new window that allows you to search all Stata resources for keywords.



```
__  ____  ____  ____  ____ (R)
_    /   ____/   /   ____/
/   /   /___/   /   /___/   16.1   Copyr
Statistics/Data Analysis           Stata
                                   4905
  MP - Parallel Edition            Colle
```



● Search all
○ Search documentation and FAQs
○ Search net resources

Keywords:

Check-In Point

If you are working alongside us, at this point you have opened the 2019 CES dataset in Stata. You have explored a variety of Stata windows to increase your familiarity with the various screens available to you. You now know what a do-file is and why it is a valuable tool for researchers. You also plan to use log files to record your work in Stata, and know how to do so. Finally, being aware that this handbook teaches you only a small amount of Stata's capacities, you are familiar with how to search the Stata help files.

With this foundation in place, you are ready to continue your explorations.

Part II: Familiarizing Yourself with Variables in the Dataset

Destination

By the end of this section, you will be able to:

- *use codebook and other commands to familiarize yourself with the variables in the dataset.*

Once you have opened your dataset, you will want to take a preliminary look at the variables.[2] There are several commands that are particularly useful:

list – lists the values of variables.

codebook– produces a codebook describing the dataset.

inspect – displays a summary of a variable, including a small histogram.

describe – describes contents of data in memory.

summarize – provides summary statistics, such as means and standard deviations.

You can use these commands by typing them in the Command window or in your Do-file.

> **Tip**: Use the Stata **Keyword Search** feature (see above) to learn more about the ways in which these commands can be modified and the various options available for each. To quickly search for help with specific commands, you can type ***help*** followed by the keyword you wish to search for in the Command window or your Do-file. For example:
>
> ***help*** list

[2] We also recommend reviewing the technical documentation and codebooks available for download with the dataset. The latter provides a listing of the survey questions asked, their corresponding variable name, and the response categories.

As discussed in Chapter 8, when using secondary datasets such as the CES, it is important to familiarize yourself with the dataset before conducting your analyses. Your first step to do so is to generate a codebook for the dataset. To do this, you can use the ***codebook*** command:

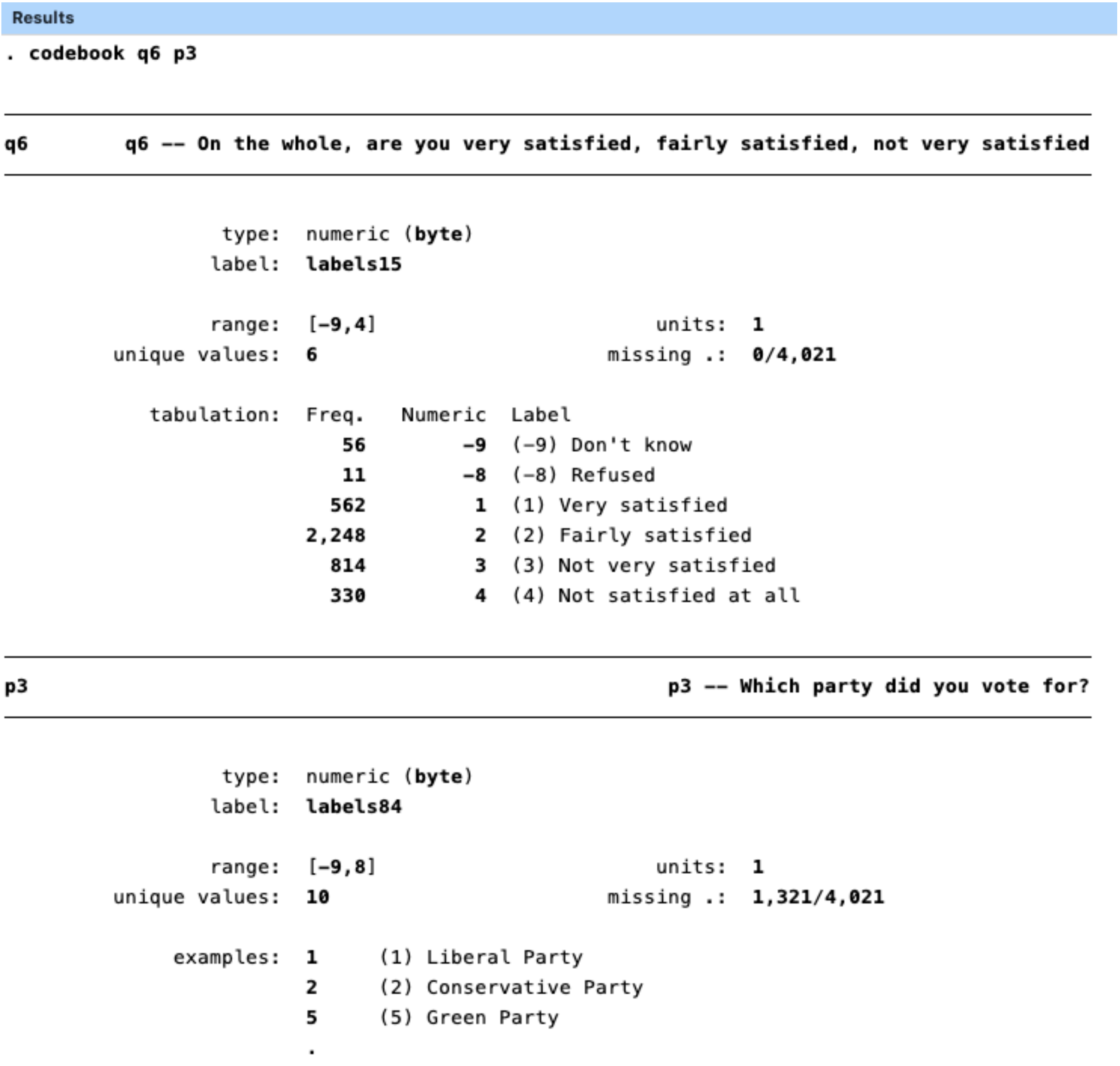
codebook

Using this command on its own provides information on every variable in the dataset. For a very large dataset such as the CES, the information provided can be overwhelming, given the number of variables in the 2019 CES dataset. One solution is to limit the results by listing only the variables you wish to explore after the command. To do this, you need to find the variable names. You then simply list these after the codebook command.

> **Tip**: You can search for variables containing key words in the dataset by using the Stata command ***lookfor***. Running the following in the command window or your Do-file will provide a list of all variables that include the word "vote" in the variable name or label:
>
> ***lookfor*** vote

To practice, let's look at two variables in the dataset. The first, *q6*, reports the level of satisfaction with the way democracy works in Canada. The second, *p3*, reports the respondents' vote choice in the 2019 Canadian federal election. Follow these steps:

1. Open a Do-file.
2. Type the following: ***codebook*** *q6 p3*
3. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
4. Compare your results with our results, reported below.



```
. codebook q6 p3

-------------------------------------------------------------------------------
q6        q6 -- On the whole, are you very satisfied, fairly satisfied, not very satisfied
-------------------------------------------------------------------------------

                  type:  numeric (byte)
                 label:  labels15

                 range:  [-9,4]                       units:  1
         unique values:  6                        missing .:  0/4,021

            tabulation:  Freq.   Numeric  Label
                            56        -9  (-9) Don't know
                            11        -8  (-8) Refused
                           562         1  (1) Very satisfied
                         2,248         2  (2) Fairly satisfied
                           814         3  (3) Not very satisfied
                           330         4  (4) Not satisfied at all

-------------------------------------------------------------------------------
p3                                                  p3 -- Which party did you vote for?
-------------------------------------------------------------------------------

                  type:  numeric (byte)
                 label:  labels84

                 range:  [-9,8]                       units:  1
         unique values:  10                       missing .:  1,321/4,021

              examples:  1     (1) Liberal Party
                         2     (2) Conservative Party
                         5     (5) Green Party
                         .
```

The ***codebook*** results provide a wealth of information. For example, looking at the results for *q6*, we find the variable name (*q6*), the variable label (q6 -- On the whole, are you very satisfied, fairly satisfied, not very satisfied), the variable label name (labels15), the range of values (-9 to 4), the number of unique values (6), the number of missing cases and total number of cases (0/4021) as well as the raw frequency distribution with the corresponding values and value labels.

The other commands listed above can provide subsets of this information, for example, ***describe*** (or just ***des***) will report the variable name, value label name, and the variable label. Try this for yourself:

1. In your Do-file, type the following: ***des*** *q6 p3*
2. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
3. Compare your ***describe*** results with your ***codebook*** results.
4. Repeat steps 1-3 with the ***list***, ***inspect***, and ***summarize*** commands.

Check- In Point

If you are working alongside us, at this point you have used a number of basic commands to examine two variables in the dataset. Before you move forward, be sure to practice these skills: Use the ***lookfor*** command to identify variables in an area of interest to you. Once you have the variable names, experiment with different commands, using your Do-file.

Part III: Applying Survey Weights

Destination

By the end of this section, you will be able to:

- *explain why survey weights are often used in analysis.*

As we discuss in Chapter 5, when researchers sample from populations, they often over-sample certain population segments and then create design weights to adjust for over or under representation of certain segments of the population. When you use secondary survey datasets, be sure to consult the metadata (technical documentation, as discussed in Chapter 8) to review information on the sampling procedures and weight variables.

The 2019 CES employs a disproportionate random sampling technique that oversamples in some areas of the country, such as Quebec, while under sampling in others. The CES dataset includes two weight variables, weight_CES and weight_PES, to account for provincial over/under sampling as well as phone ownership (landline and/or cell phone. See CES technical documentation for more details). The former weights according to the full sample and the latter is a weight based on only those respondents who completed both waves of the survey (campaign period and post-election). We use the full sample weight (weight_CES) in our analyses.

Stata offers four different types of weights, the discussion of which is beyond the scope of this handbook (interested users can learn more via the Stata help files). The most appropriate weight for the 2019 CES is one that accounts for sampling design factors, namely a Stata pweight. However, because many of the basic commands that we introduce within this handbook do not allow for such weights, we instead use an "analytic" weight (aweight) with the CES data. [3]

To apply survey weights in Stata, you add additional syntax to your commands text. We include the syntax to weight the CES data in the examples in the following sections where possible. As you work through this handbook, be sure to reflect upon how the use of survey weights affects the results.

Check-In Point

Survey weights can be a challenging idea for many new researchers. Before you move forward, ensure that you are comfortable with your understandings. Why do researchers use survey weights? How can you as a user of secondary survey datasets determine how the original researchers constructed their survey weights? We encourage you to review both Chapter 5 and Chapter 8 of the *Explorations* textbook before moving forward to the next section.

[3] It is possible to designate the survey design for the dataset along with the appropriate weight using Stata's ***svyset*** command (see ***help*** svyset). We do not cover this procedure.

Part IV: Examining Frequency Distributions and Univariate Statistics

Destination

By the end of this section, you will be able to:

- *generate frequency distributions;*
- *apply survey weights; and*
- *generate univariate statistics.*

In Chapter 12, we discuss how researchers start their analyses by examining the frequency distributions and summary statistics for each individual variable in their analysis. These can be generated in several ways in Stata. The ***tabulate*** command is especially useful:

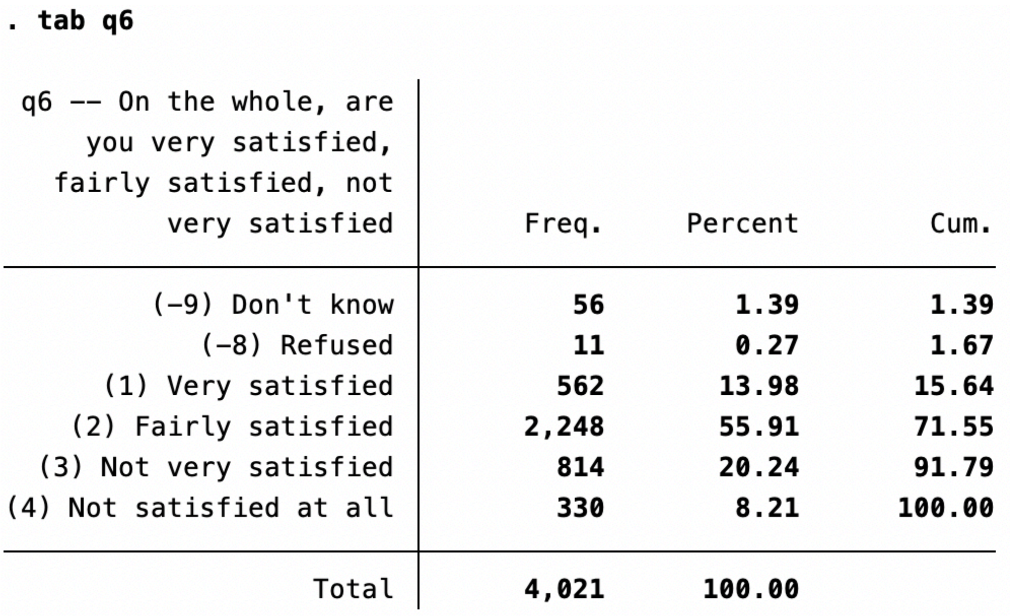
tabulate *variable*

Like many of the commands in Stata, you can shorten this command to:

tab

Try this for yourself to generate a frequency table for the satisfaction with the way democracy works in Canada variable:

1. In your Do-file, type the following: ***tab*** *q6*
2. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
3. Compare your results with the results displayed below.



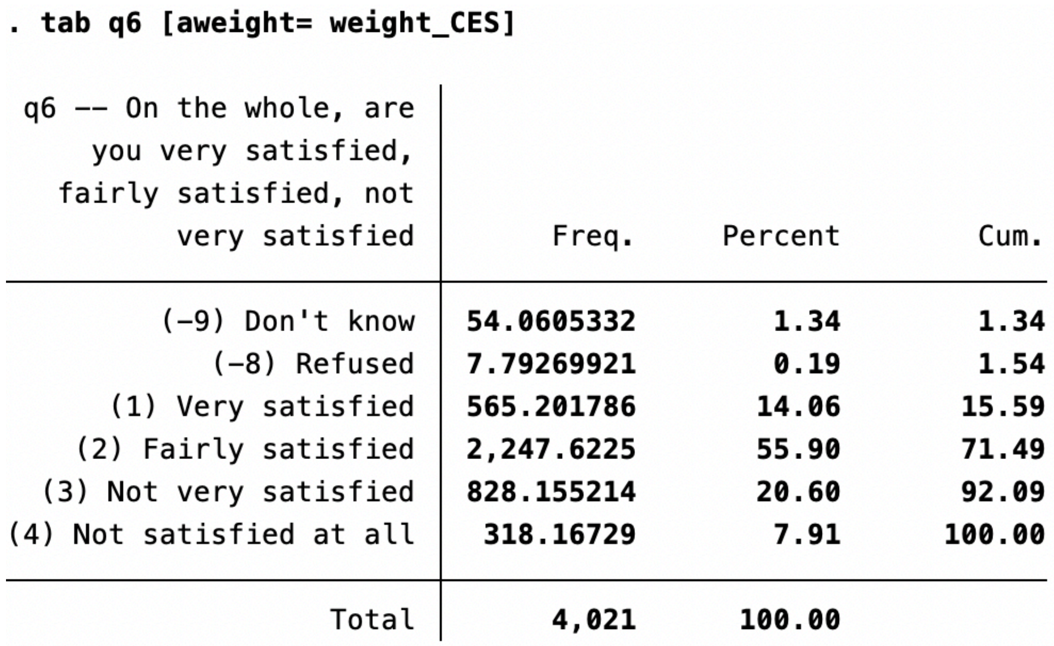
```
. tab q6

q6 -- On the whole, are |
    you very satisfied, |
  fairly satisfied, not |
         very satisfied |      Freq.     Percent        Cum.
------------------------+-----------------------------------
        (-9) Don't know |         56        1.39        1.39
           (-8) Refused |         11        0.27        1.67
     (1) Very satisfied |        562       13.98       15.64
   (2) Fairly satisfied |      2,248       55.91       71.55
 (3) Not very satisfied |        814       20.24       91.79
(4) Not satisfied at all |       330        8.21      100.00
------------------------+-----------------------------------
                  Total |      4,021      100.00
```

Note that the results include the raw frequency, relative frequency (Percent) and the cumulative frequency.

> **Tip**: To produce a frequency distribution table for multiple variables in a single command, simply add '1" to the ***tabulate*** command: ***tab1*** *variable1 variable2 variable3…*

In the previous section, we discussed survey weights, and noted that to apply weights we simply add syntax to our command. Let's do this now, re-running our tab command to account for the disproportionate random sample by weighting the data.

1. In your Do-file, type the following: ***tab*** *q6* [***aweight***= *weight_CES*]
2. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
3. 3. Compare your results with the results displayed below, and with your original results. Note how the addition of the weight variable affects the results.



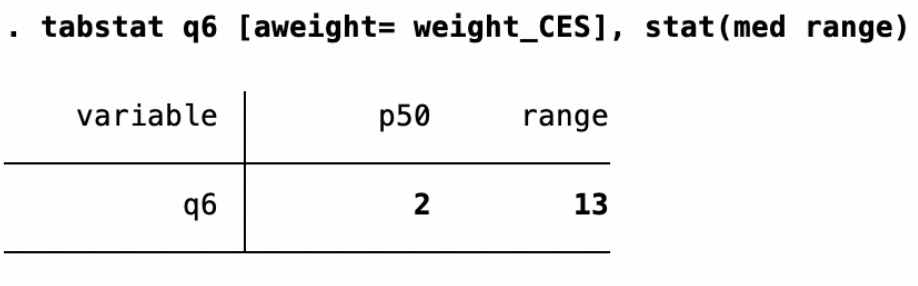
```
. tab q6 [aweight= weight_CES]

q6 -- On the whole, are │
    you very satisfied, │
   fairly satisfied, not │
          very satisfied │      Freq.     Percent        Cum.
─────────────────────────┼─────────────────────────────────────
         (-9) Don't know │ 54.0605332        1.34        1.34
            (-8) Refused │ 7.79269921        0.19        1.54
      (1) Very satisfied │ 565.201786       14.06       15.59
    (2) Fairly satisfied │ 2,247.6225       55.90       71.49
  (3) Not very satisfied │ 828.155214       20.60       92.09
(4) Not satisfied at all │  318.16729        7.91      100.00
─────────────────────────┼─────────────────────────────────────
                   Total │      4,021      100.00
```

In examining a variable, you will also want to consider the appropriate measures of central tendency and dispersion. (Review Chapter 12 if you need refreshing on the appropriate measures of central tendency and dispersion by variable level.) As noted earlier, the frequency distribution results include the raw frequency, relative frequency and the cumulative frequency. This is an ordinal variable, and from these results you can visually identify the appropriate measure of central tendency (median) and dispersion (range). To move beyond a visual assessment, you can use a related Stata command, ***tabstat***, that allows you to specify the summary statistics that you wish to view. Note that Stata does not include the mode or the variation ratio as statistics for ***tabstat***. Fortunately, both are easily identified with the information reported in the frequency distribution table (again, see Chapter 12).

For example, to obtain the median and range for this variable we would run the following command to produce the results reported below:

1. In your Do-file, type the following: ***tabstat*** *q6* [***aweight***= *weight_CES*], stat(med range)

2. Select the line and then click on the “Do” icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
3. Compare your results with the results displayed below.



```
. tabstat q6 [aweight= weight_CES], stat(med range)

    variable |       p50     range
-------------+--------------------
          q6 |         2        13
----------------------------------
```

Let’s interpret these univariate statistics, starting with the median. Stata reports the median as p50 (the 50th percentile), which is reported as “2”. Note that these are the actual values for the variable, not the value labels as reported in the frequency distribution. To determine the value labels associated with these values, you need to first identify the label name for the variable and then list the value labels for the variable. You can identify the label name for the variable from the codebook or by using the ***describe*** command. You can then use the ***label list*** command to view the value labels (we discuss value labels in more detail below). To do so, use the following syntax:

des q6
label list labels15

Based on the results, we see that the value label for 2 is “Fairly satisfied”. But how do we interpret the range value of 13? Recall from Chapter 12 that the range is estimated by subtracting the lowest value from the highest value. When you look at the range result, the number should strike you as a bit curious – how does a variable with four possible response categories have a range of 13? When you see results like this, you should always ask questions and seek out the answer. In this case, the answer lies with the coding. In the CES dataset “don’t know” is coded as -9 (see above). Thus, Stata calculated the range as (-9)– (4) for a range of 13. In this example, some recoding is necessary to estimate the range. We will return to this topic shortly; for now, simply know that it is always important to critically assess your results, as statistical software will not catch such issues for you!

Check-In Point

At this point, you should understand how to use the Stata ***tabulate*** **(*tab*)** and ***tabstat*** commands to generate frequency distributions and univariate statistics (specifically the median and the range). You should also be able to add syntax to your command to apply survey weights. Before moving forward, be sure to practice these skills with other variables, using the codebook to help interpret categories and to identify curious results that may reflect coding. Be sure as well to continue to compare how results change with the addition of the survey weight. We also encourage you to view the **help** files associated with ***tabulate*** **(*tab*)** and ***tabstat*** to learn more about the various options and statistics available for these commands. For now, as we are focused on univariate statistics, you should view the help information for tabulate oneway (we work with tabulate twoway below, when we consider bivariate relationships).

Part V: Creating and Recoding Variables

Destination

By the end of this section, you will be able to:

- *explain why you should never recode original variables;*
- *create new variables;*
- *recode variables; and*
- *rename variables and add or alter variable labels.*

As we have already observed, it is often necessary to recode variables before you can work them. As a rule, we recommend never altering original variables within a dataset. We will repeat this, in case you are reading quickly: **never alter original variables in a dataset**. Instead, you should generate a new variable from the original and then make the transformations you need to your new variable. There are two reasons for this: (1) it allows you to check your work by comparing the recoded variable against the original one, and (2) maintaining the original variable allows you to use a different transformation processes if you need to do so at a later time.

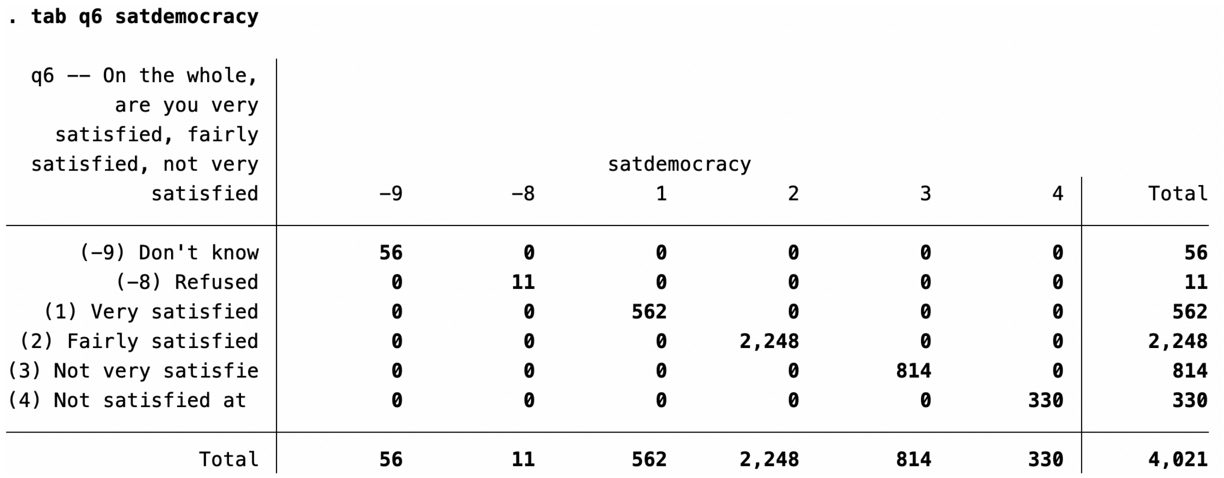
Creating New Variables

To create a new variable, we use ***generate*** **(*gen*)**. We select a new variable name for the new variable, and then command Stata to create (generate) a new variable from the existing variable. Let's consider this with variable *q6* from the 2019 CES. You are going to create a new variable named *satdemocracy* (remember that the variable looks at satisfaction with democracy).

1. In your Do-file, type the following: ***gen*** *satdemocracy* = *q6*
2. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)

When you generate a new variable, you can confirm your work by comparing the original variable and the new variable in a cross-tabulation (we discuss cross-tabulation in more detail below). Note that using a cross-tabulation to check your recoding is only advisable when working with nominal or ordinal level variables. For interval/ratio level variables, we use a frequency distribution, as demonstrated below. To produce a cross-tabulation, you simply add the two variables after the tab command:

1. In your Do-file, type the following: ***tab*** *q6 satdemocracy*
2. Select the line and then click on the "Do" icon in the top right corner of the Do-file screen. (Reminder: you can also use the short-cut keys: Shift – Command -D on a Mac or CTRL – D with a Windows operating system.)
3. Compare the original values in the row and the new values in the column.



```
. tab q6 satdemocracy

q6 -- On the whole, |
       are you very |
   satisfied, fairly|
satisfied, not very |                          satdemocracy
          satisfied |        -9         -8          1          2          3          4 |     Total
--------------------+------------------------------------------------------------------+----------
     (-9) Don't know |        56          0          0          0          0          0 |        56
       (-8) Refused |         0         11          0          0          0          0 |        11
  (1) Very satisfied |         0          0        562          0          0          0 |       562
(2) Fairly satisfied |         0          0          0      2,248          0          0 |     2,248
(3) Not very satisfie|         0          0          0          0        814          0 |       814
(4) Not satisfied at |         0          0          0          0          0        330 |       330
--------------------+------------------------------------------------------------------+----------
              Total |        56         11        562      2,248        814        330 |     4,021
```

The distribution of the two variables should be identical (values only on the diagonal line), although you will note that the value labels are not attached to the new variable. You will need to add these, which we return to shortly.

There are a number of more advanced options for the ***generate*** command, including the option of combining multiple variables to generate a single measure. This is part of a host of mathematical calculations that can be used with ***generate***. We do not cover these more advanced procedures in this introductory handbook but encourage interested users to seek out additional information on the various expressions that can be used with ***generate*** (along with other ways that the ***generate*** command can be used**)** via the Stata help files.

Recoding Variables

Once you have created the new variable, you can begin making transformations to meet your research needs. For example, let’s say you want to transform *satdemocracy* to remove cases that report “Don’t know” or “Refused”. Recall from the last section that the range for *q6* was nonsensical given the inclusion of these responses, which are coded as -9 and -8, respectively. Given that we cannot be sure how individuals who answered “Don’t know” or “Refused” feel about the way democracy works in Canada, we want to exclude these cases from our new variable for our analysis.

In Stata, missing cases are denoted as “.” To transform the existing -9 and -8 codes for our new *satdemocracy* variable into “.” in the dataset, we use the recode command. We can do this for individual values (option A) or alternatively, we can use a range instead of noting each value separately (option B):

Option A: ***recode*** *satdemocracy* -9=. -8=.

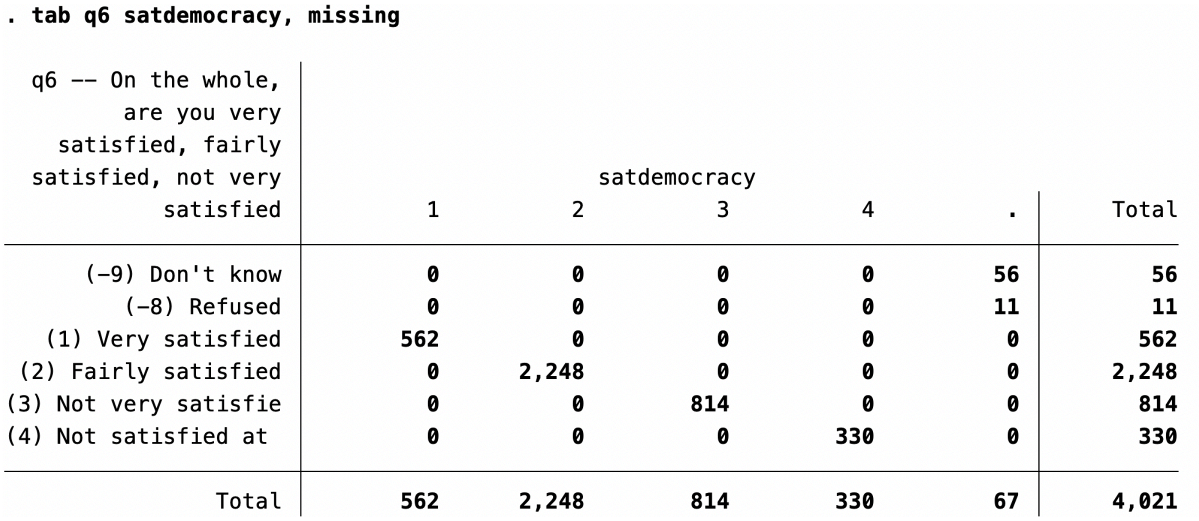
Option B: ***recode*** *satdemocracy* -9/-8=.

Recode your missing values now with either option by entering the syntax command into your Do-file, selecting the text, and clicking “Do.” It does not matter which option you select; there are often a number of ways to achieve the same outcome in Stata. We outline some of the more simplistic procedures within this handbook but recognize that there are other commands Stata users may use to achieve the same results.

With the newly recoded variable, you should confirm that you have not made any errors by generating a cross-tabulation to compare the new variable to the original variable. To include the missing cases (those cases you set to “.”) you add the **missing** option to the ***tabulate*** command as follows:

```
*Check original and new variable distributions using a cross-tabulation

tab q6 satdemocracy, missing
```

Compare your results to our own:



```
. tab q6 satdemocracy, missing

q6 -- On the whole, │
      are you very  │
  satisfied, fairly │
satisfied, not very │                 satdemocracy
          satisfied │         1          2          3          4          . │     Total
────────────────────┼───────────────────────────────────────────────────────┼──────────
     (-9) Don't know │         0          0          0          0         56 │        56
       (-8) Refused │         0          0          0          0         11 │        11
  (1) Very satisfied │       562          0          0          0          0 │       562
(2) Fairly satisfied │         0      2,248          0          0          0 │     2,248
(3) Not very satisfie│         0          0        814          0          0 │       814
(4) Not satisfied at │         0          0          0        330          0 │       330
────────────────────┼───────────────────────────────────────────────────────┼──────────
              Total │       562      2,248        814        330         67 │     4,021
```

From the results, we can confirm that we have not made any errors by comparing the number of cases in each category of the new variable against those of the original variable. You should note that the number of missing cases includes the 56 respondents who responded “Don’t Know” and the 11 respondents who refused to answer.

What would you do if you discovered that you made a mistake? (For example, say you accidently set the value of 4 as missing, when it is in fact not missing). First, you would congratulate yourself for creating new variables rather than transforming an original variable, as your problem can be easily fixed. Second, you can simply delete a mis-transformed variable in Stata using the ***drop*** command and then re-create it with the corrected syntax. For the example above, you would use the following syntax to delete the *satdemocracy* variable:

```
drop satdemocracy
```

(Please don’t actually do this – we are going to continue working with *satdemocracy!*)

Renaming variables and adding or altering value labels

The majority of the variables in the 2019 CES dataset include variable and value labels – descriptions of each variable, and descriptions of the values for categorical variables. However, for any new variables that you generate, you will need to either create or modify variable names and labels. The most useful commands for doing so are listed below.

To change the name of an existing variable:

```
rename old_varname new_varname
```

To add a variable label to an existing variable:

```
label variable varname ["label"]
```

To add value labels, you must define the labels (first line), and then attach those labels to the variable:

```
label define lblname # "label" [# "label"…]
label values varname [lblname]
```

For example, we can rename our newly generated variable, *satdemocracy*, and add the appropriate variable and value labels:

```
rename satdemocracy satdem
```

Next, we will add variable and value labels:

```
label variable satdem "Satisfaction with democracy"
label define satdem 1 "Very satisfied" 2 "Fairly Satisfied" 3 "Not very satisfied" 4 "Not satisfied at all"
label values satdemoc satdemoc
```

You should once again check your work by comparing the new variable against the original variable.

```
tab q6 satdem, missing
```

Recall from above that you can use the ***label list*** command to check the numeric values assigned to the value labels:

```
label list labels15 satdem
```

Repeat the steps outlined in Part IV to see how the removal of "don't know" and "refused" response categories affects the frequency distribution and summary statistics.

There are other commands that may be used with labels such as ***drop*** to delete a label set, or the **replace** option that can be used to modify a label set. As with the other commands we have introduced, additional information on labels can be found using the Stata help files ***(help label***).

Check-In Point

We covered a bit of ground in this section: you now know the reasons why you should never recode original variables (and thus will avoid future despair when discovering recoding errors). You know how to create new variables, and then recode, rename, and add variable and value labels to those new variables. These are likely to be some of the most frequently used procedures when working with data.

Part VI: Creating Bar Graphs and Pie Charts

Destination

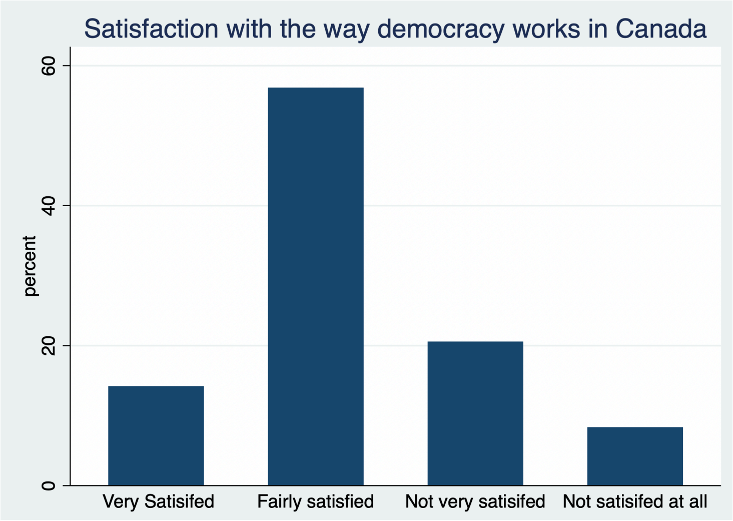
By the end of this section, you will be able to:

- *create bar graphs and pie charts.*

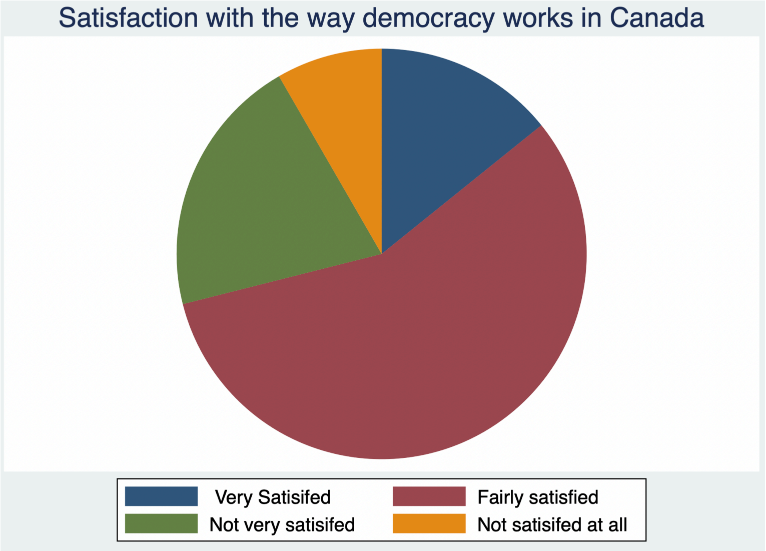
Stata offers a range of graphing options. While the numerous options available for graphing within Stata are beyond the scope of this handbook, we do outline the steps to create basic graphs and charts that you may use to display univariate frequency data.

Let's report the frequency distribution of our recoded satisfaction with democracy in a bar chart and a pie chart with the following syntax:

graph bar, over(*satdem*) title(Satisfaction with the way democracy works in Canada)

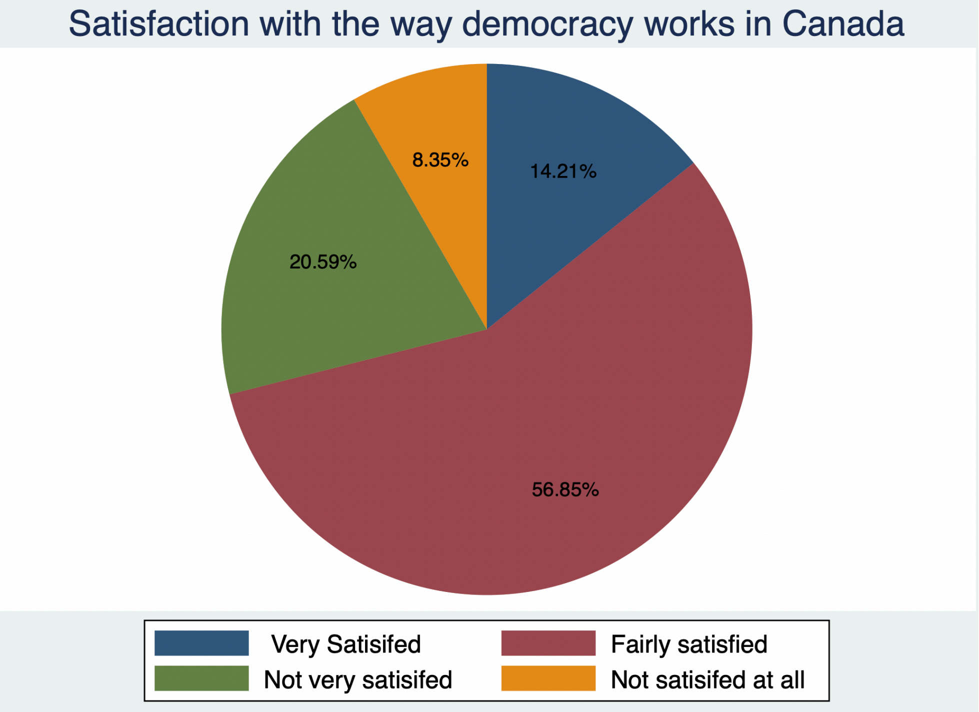


graph pie, over(*satdem*) title(Satisfaction with the way democracy works in Canada)

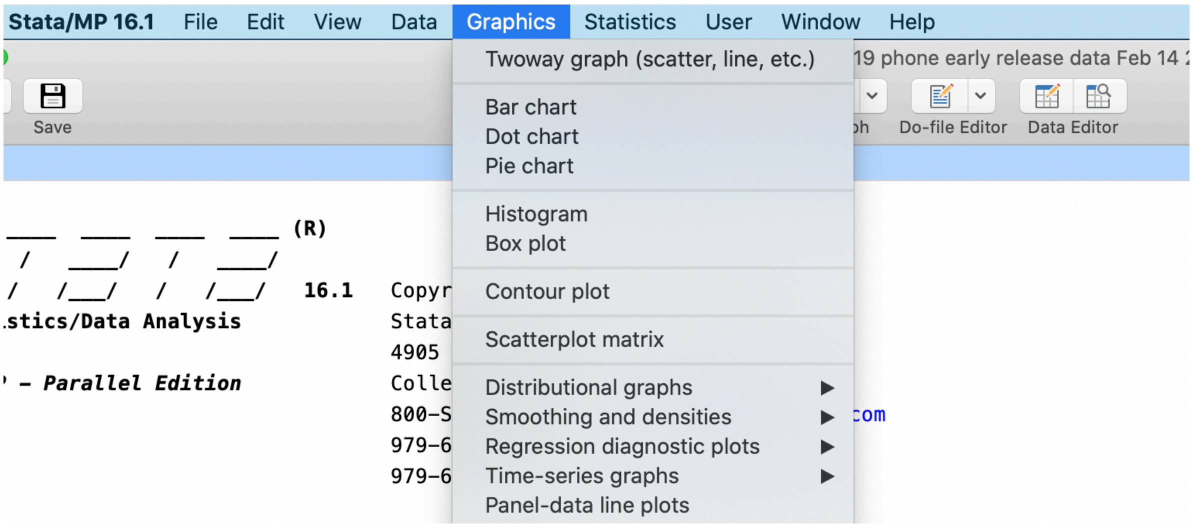


You will notice that these graphs include the title that you gave the graph and the variable's value labels. We can include additional information in the graph by adding additional instructions in the syntax. Let's add the percentage of the sample within each category to the pie chart with the following syntax:

graph pie, over(*satdem*) plabel(_all percent, format(%9.2f)) title(Satisfaction with the way democracy works in Canada)

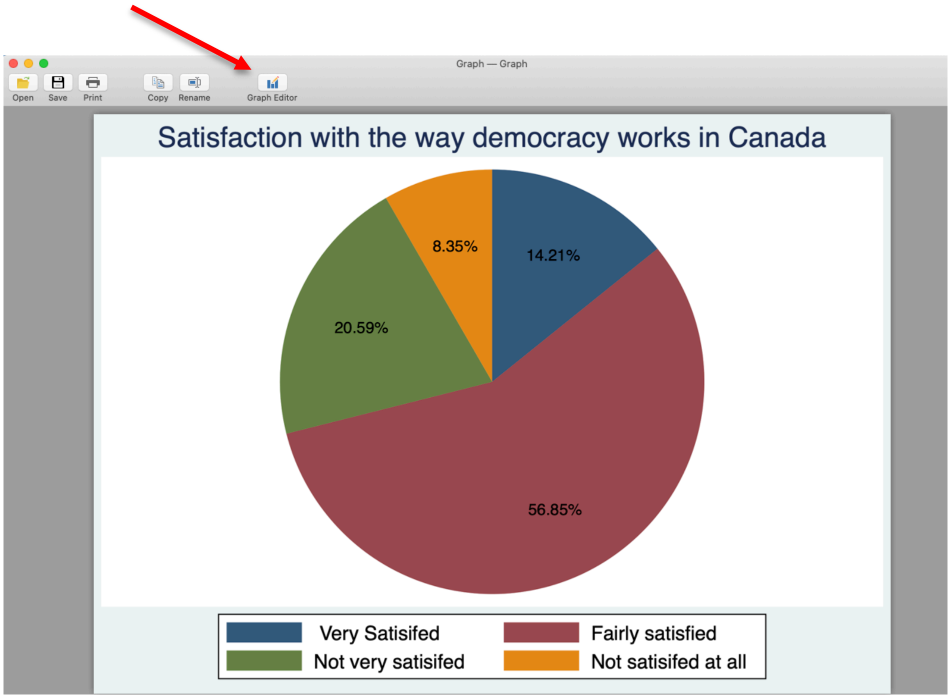


There are a number of ways to modify how you graph displays and what information is included. In addition to the Stata help file for graphing, you can also work with the drop down **Graphics** menu on the top Stata toolbar.



```
 ____  ____  ____  ____ (R)
 /    ____/  /    ____/
/   /___/  /   /___/   16.1   Copyr
stics/Data Analysis           Stata
                              4905
' - Parallel Edition          Colle
                              800-S
                              979-6
                              979-6
```

In addition, you can make changes to the graph after it has been created within the **Graph window** by using the **Graph Editor** feature.



Check-In Point

Congratulations – you are now able to create basic univariate graphs. If you have an interest in more advanced graphing features, be sure to review the Stata graph help files and other online resources.

Part VII: Comparing Two Independent Samples

Destination

By the end of this section, you will be able to:

- *use a t-test to assess differences of means between two independent samples.*

In Chapter 13, we consider how we often wish to compare the means of two independent groups to see if they differ. To assess differences of means between two groups, we can use a t-test. To do this, we need a variable with our two groups of interest (for example, a treatment group and a control group from an experimental study) and an interval/ratio variable for which we expect a difference between the two groups. (Recall that means should only be used with interval/ratio level variables.)

For example, let's compare the average income (interval/ratio variable) of men and women (dichotomous variable).[4] We used the following syntax to find, examine, recode, and check the variables that we will use for this analysis:

```
*Use the lookfor command to find the gender variable

lookfor gender
```

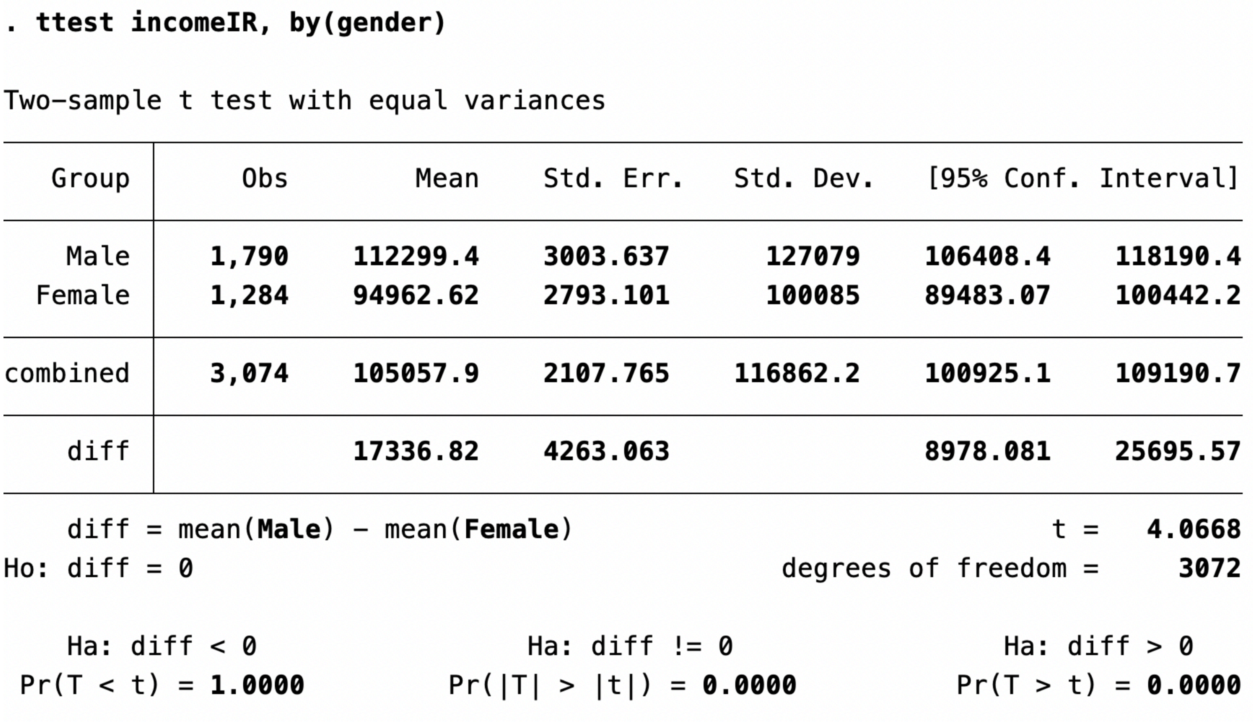
[4] Note that the CES asks about household income. We have used this as a proxy for personal income in this example.

```
*Look at the value labels and distribution of the original variable

des q3
label list labels13
tab q3

*Generate, recode and label the new variable[5]

gen gender = q3
recode gender 3=.
lab var gender "Dichotomous gender variable"
lab define gender 1 "Male" 2 "Female"
lab values gender gender

*Check original and new variable distributions using a cross-tabulation

tab q3 gender, missing

*Use the lookfor command to find the income variable

lookfor income

*Look at the value labels and distribution of the original variable

des q69
label list labels69
tab q69

*Generate, recode and label the new variable (note that we have added "IR" to the end of the
new variable name to indicate interval/ratio variable)

gen incomeIR = q69
recode incomeIR -9/-8=.
lab var incomeIR "Household income in dollars'"

*Check original and new variable distributions. Note that we use the frequency distribution
command when working with interval/ratio level variables. In this example, we check that
“don’t know” and “refused” responses have been set to missing

tab1 q69 incomeIR, missing
```

You can then use Stata’s ***ttest*** command to compare the average income of the two groups. Note that the **by** option indicates the variable to use for the two groups; men and women, in this example:

```
ttest incomeIR, by(gender)
```

[5] Note that we have set the single respondent that identified as “Other” to missing since there is not enough cases for a meaningful analysis of this category.



```
. ttest incomeIR, by(gender)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
    Male |   1,790    112299.4    3003.637      127079    106408.4    118190.4
  Female |   1,284    94962.62    2793.101      100085    89483.07    100442.2
---------+--------------------------------------------------------------------
combined |   3,074    105057.9    2107.765    116862.2    100925.1    109190.7
---------+--------------------------------------------------------------------
    diff |            17336.82    4263.063                8978.081    25695.57
------------------------------------------------------------------------------
    diff = mean(Male) - mean(Female)                              t =   4.0668
Ho: diff = 0                                     degrees of freedom =     3072

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

Do the incomes of men and women differ? The results suggest that they do. From the table above we see that the mean income for men is $112,299.40 compared to a mean income of $94,962.62 for women, a difference of $17,336.82 (the value reported as "diff" in the table).[6] In other words, the results indicate that, on average, men earn $17,336.82 a year more than women. We provide a more detailed discussion of this test and the results reported here in the text (see Chapter 13).

In Chapter 13, we also discuss statistical significance. You will recall that researchers use a one-tailed test when they hypothesize a specific direction for a relationship and use a two-tailed test when they do not hypothesize a direction. The Stata ***ttest*** automatically reports the level of statistical significance of the difference of the two means, for both a two-tailed t-test (Ha: diff != 0) and a one-tailed test (Ha: diff < 0 or Ha: diff > 0). Given that we did not make any assumptions about the direction of the relationship in advance, we would use the results from the two-tailed test, a result that indicates a statistically significant relationship at p<0.001.

Check-In Point

If you are following along with the examples by running your own data, you should now be able to use a t-test to assess differences of means between two independent samples. This is a useful skill, and we encourage you to practice by exploring other income differences between other dichotomous groups in the CES dataset. For example, do the average incomes of university and non-university graduates differ?

Part VIII: Examining Bivariate Relationships for Nominal and/or Ordinal Variables

Destination

By the end of this section, you will be able to:

- *create crosstabulations;*
- *calculate measures of association; and*

[6] Note that the t-test results reported in Chapter 13 differ from those reported here due to the use of weighted data to produce the results reported in the text.

- *calculate Chi Square.*

In Chapter 12, we introduce you to the four questions we must answer when assessing whether there is a relationship between two variables:

1. What is the form/direction of the relationship?
2. How strong is the relationship?
3. Is the relationship statistically significant?
4. What happens to the relationship when we control for other variables?

In this section, we will focus on how to use Stata to help answer the first three of these questions for nominal and/or ordinal variables. In the section that follows, we will look at interval/ratio variables. The final section of this handbook will consider the fourth and final question.

To examine the relationship between two nominal and/or ordinal variables, we create a cross-tabulation (contingency table), calculate the appropriate measure of association, and calculate the appropriate inferential statistic. With Stata, we can do all of this with a single line of syntax.

For example, let's test the hypothesis that those with higher levels of income will also be more interested in politics. Before we can do so, we must recode our variables. For this example, we will recode our dependent variable (DV), political interest, into a three-point measure ranging from low to high (terciles), as follows:

*Use the lookfor command to find the interest variable

```
lookfor interest
```

*Look at the value labels and distribution of the original variable

```
des q9
label list labels17
tab q9
```

*Generate, recode and label the new variable into terciles (note that the tercile divisions are based on the cumulative frequency)

```
gen polinterest = q9
recode polinterest -9=. -8=.
```

*Check original and new variable distributions using a cross-tabulation

```
tab q9 polinterest, missing
```

To transform this variable into terciles, we generate a frequency distribution for the new variable and use the cumulative frequency column to determine the values for approximately 33%, 66% and 99% of the sample. In this case the ranges would be 0-6, 7-8, and 9-10, respectively.[7] We will use these values to create a political interest tercile variable:

[7] An alternative way to recode this variable would be to set respondents that choose 0-4 as low interest, 5 as the mid-point, and all responses over 5 as high interest. We opt to use the cut-off points reported in the cumulative percent to produce three roughly equal sized groups.

```
* Generate a frequency distribution for the new variable
tab polinterest

*Note new variable for political interest terciles
gen polinteresttercile = polinterest
recode polinteresttercile 0/6=1 7/8=2 9/10 =3
lab define polinteresttercile 1 " Low interest" 2 "Middle interest" 3 "High interest" , replace
lab values polinteresttercile polinteresttercile

*Check original and new variable distributions using frequency distribution

tab1 polinterest polinteresttercile,missing
```

We also will recode our independent variable (IV), income, into terciles (low, middle, and high income). It is important to note that many respondents (approximately 25% in the 2019 CES) did not report their actual income. To help reduce the number of non-responses, individuals who refuse or report that they do not know their actual income are asked a follow-up question that provides income categories for the respondent to choose instead of stating their actual income. In this example, we combine the responses from the two questions to generate a new income category variable. Note that we use the income variable that we generated for the t-test above in this example:

```
*Use the lookfor command to find the interest variable

lookfor income

*Look at the value labels and distribution of the original variable

des q70
label list labels70
tab1 incomeIR q70

*Generate, recode and label the new variable into terciles (note that this combines the two income
measures included in the CES)

gen incomegrptemp =incomeIR

recode incomegrptemp 0=1 1/30000=2 30001/60000=3 60001/90000=4 90001/110000 = 5
110001/150000=6 150001/200000=7 200001/2130000=8

lab var incomegrptemp "Temp income grp variable'"
tab1 incomeIR incomegrptemp, missing

*Merge two income group variables

gen incomegrpmerged = incomegrptemp
replace incomegrpmerged = q70 if incomegrptemp==.
recode incomegrpmerged -9/-8=.
lab define labels70 3 "(3) $30,001 to $60,000", modify
lab values incomegrpmerged labels70
tab incomegrpmerged
```

```
*Create new income tercile variable

gen incometercile=incomegrpmerged
recode incometercile  1/3 = 1 4/5=2 6/8=3
lab var incometercile "Income terciles"
lab define incometercile 1 " low Income" 2 "Middle Income" 3 "High Income"
lab values incometercile incometercile

*Check original and new variable distributions using frequency distribution

tab1 incomegrpmerged incometercile , missing
```

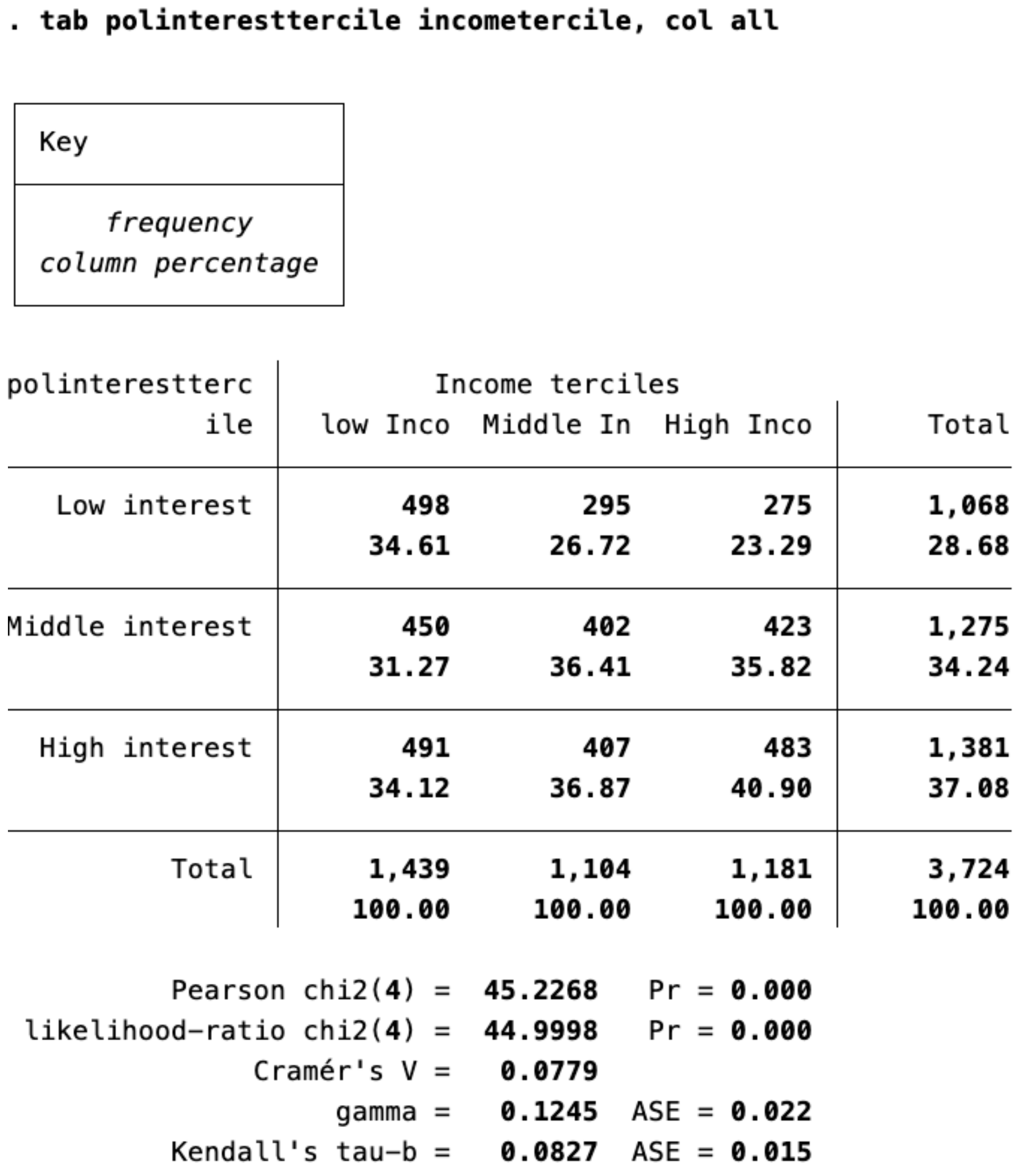
Note that we corrected an error in the value label “labels70” using the *modify* option. Also note that we used a conditional statement (“**if**”) when we combined two variables to create *incomegrpmerged*. This if-then statement tells Stata to only apply the command to cases that meet the criteria we have included. In this example, only do this if the variable *incomegrptemp* is equal to (==) missing (.). You can use other expressions besides equal to with the if command, such as greater than (>) and less then (<). Use the ***help*** **if** command to learn more.

With our variables prepared for analysis, we can now use the following syntax to test the relationship:

```
tab polinteresttercile incometercile, col all
```

Note that in creating crosstab tables, it is important to be clear on the order of your dependent and independent variables. We have followed the format DV then IV (*polinteresttercile incometercile*), so we ask for the column percentages option (**col**) in order to compare values of the DV across categories of the IV. If we had instead followed the format IV then DV, we would need to ask for row percentages. Note as well that we have used the **all** option to request that all available statistics are reported. This produced the following results:[8]

[8] Note that these results differ from those reported in Chapter 13. This is due to weighting. Stata does not allow the use of non-integer weights to estimate measures of association and statistical significance. As such, we have proceeded without weights for this example.



```
. tab polinteresttercile incometercile, col all

+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

polinteresttterc |          Income terciles
             ile |  low Inco  Middle In  High Inco |     Total
-----------------+---------------------------------+----------
    Low interest |       498        295        275 |     1,068 
                 |     34.61      26.72      23.29 |     28.68 
-----------------+---------------------------------+----------
 Middle interest |       450        402        423 |     1,275 
                 |     31.27      36.41      35.82 |     34.24 
-----------------+---------------------------------+----------
   High interest |       491        407        483 |     1,381 
                 |     34.12      36.87      40.90 |     37.08 
-----------------+---------------------------------+----------
           Total |     1,439      1,104      1,181 |     3,724 
                 |    100.00     100.00     100.00 |    100.00 

          Pearson chi2(4) =  45.2268   Pr = 0.000
 likelihood-ratio chi2(4) =  44.9998   Pr = 0.000
               Cramér's V =   0.0779
                    gamma =   0.1245  ASE = 0.022
          Kendall's tau-b =   0.0827  ASE = 0.015
```

Is there a relationship between income and political interest? Recall from Chapter 12 that the first step when assessing the results from a contingency table with two ordinal level variables is to look for a consistent increase/decrease in the percentage of respondents across categories of the IV in the top row and the opposite pattern in the bottom row. In this example, reading across the top row ("Low interest"), we find that the percentages decrease as we move from left to right (low to high income): high income earners are approximately 11 percentage points less likely to indicate low political interest compared to their low income counterparts. Looking at the bottom row ("High interest") we find the reverse pattern, with high income earners approximately 7 percentage points more likely to indicate high political interest relative to those in the low income category.

As noted in Chapter 12, our next step is to consider the correct correlation coefficient. Given that both variables are ordinal, we can assess the strength of the association by looking at the Gamma or the Tau value. Since gamma tends to inflate the strength of the relationship, we will opt for the more conservative Tau estimate. In this example, we find an extremely weak, positive (as income increases, political interest increases) relationship with a Tau value of 0.08.

Finally, recall from Chapter 13 that we can assess whether or not the relationship is statistically significant by looking at the Pearson Chi-square value. The results indicate that the relationship is statistically significant at p<0.001. As such, we would conclude that our results support our hypothesis: those with higher levels of income appear to be more interested in politics.

Check-In Point

This section covered an incredible amount of information – creating crosstabulations, how to calculate measures of association, and calculating Chi Square – all with a single line of syntax (after the variables were recoded!). This is powerful, but it is always critical to keep in mind that your decisions with respect to recoding have great influence on the results, so always check your work carefully before assessing relationships.

Part IX: Examining Bivariate Relationships between Interval/Ratio Variables

Destination

By the end of this section, you will be able to:

- *create scatterplots;*
- *calculate Pearson's Correlation Coefficient; and*
- *conduct basic linear regression.*

To assess the relationship between two continuous (interval/ratio) variables, we continue to ask the same four questions noted in the last section (and, of course, in Chapter 12), but we use different statistical techniques.

For example, we might theorize that younger individuals are more apt to like the Green party. To test this, we first recode the variables for analysis. To do this, you can use the following syntax:

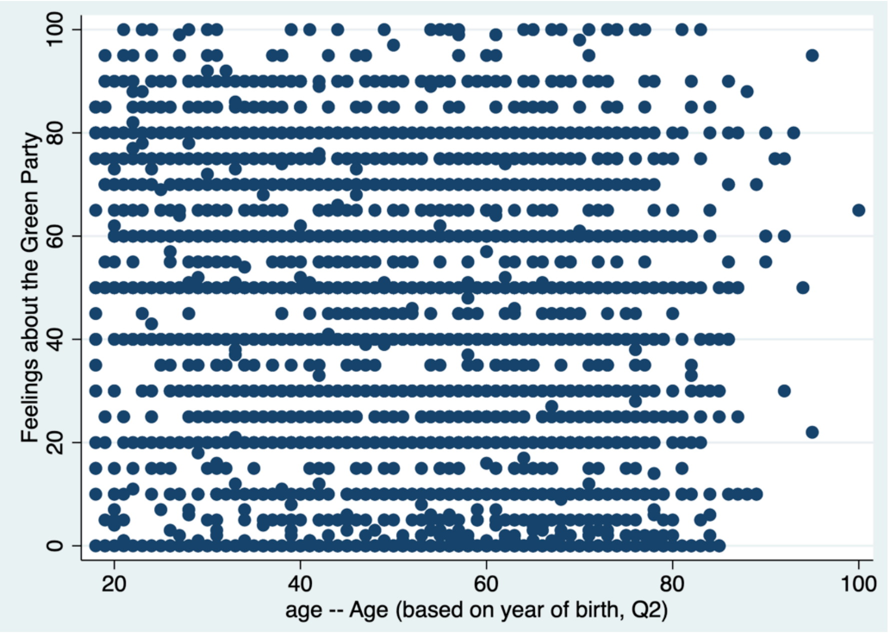
```
*Use the lookfor command to find the interest variable

lookfor green

*Look at the value labels and distribution of the original variable

des q18
label list labels26
tab q18

*Generate, recode and label the new variable

gen greenfeelings = q18
recode greenfeelings -9/-6=.
lab var greenfeelings "Feelings about the Green Party"
lab define greenfeelings 0 "Really dislike" 100 "Really like" , replace
lab values greenfeelings greenfeelings

*Check original and new variable distributions using frequency distribution
```

```
tab1 q18 greenfeelings, missing

*Use the lookfor command to find the interest variable

lookfor age

*Look at the value labels and distribution of the original variable

des age
tab age
```

NOTE: The age variable does not require recoding.

With our variables ready for analysis, we can produce a scatterplot to visually inspect whether or not there appears to be a linear relationship between age and feelings towards the Green party using the following syntax to produce the scatterplot shown below:

```
twoway (scatter  greenfeelings age)
```



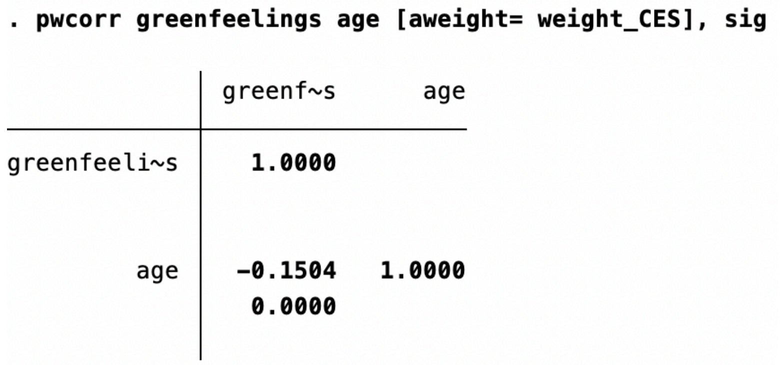
Now, we assess the scatter plot. What do you see? Don't panic – we don't see anything either. Based on the graph, it is difficult to interpret any type of relationship! It may be that age is *not* associated with feelings about the Green party. To be sure, we need to look to further, either using Pearson's *r* or basic linear regression.

Pearson's *r*

We can find the measure of association between the two variables, Pearson's *r*. To estimate this value, we will use the command for a pairwise correlation, ***pwcorr,*** and we will add the command ***sig*** so that the output includes the level of statistical significance for the relationship. We will also apply the sample weight with this command. All together, our syntax is as follows:

```
pwcorr greenfeelings age [aweight= weight_CES], sig
```

Compare your results to our own.



```
. pwcorr greenfeelings age [aweight= weight_CES], sig

             | greenf~s      age
-------------+------------------
greenfeeli~s |   1.0000 
             |
             |
         age |  -0.1504   1.0000 
             |   0.0000
             |
```

In interpreting the results, we will start with the correlation coefficient. The output includes the correlation between each variable with itself (a value of 1 since they are obviously perfectly correlated) and the measure of association between age and feelings about the Green party. The results indicate a weak, negative relationship between age and feelings about the Green party (-0.15): as age increases, feelings about the Green party decrease.

We next turn to the inferential statistic to see if this weak, negative relationship is statistically significant. The value below the correlation coefficient is the probability of observing a relationship of this strength in the sample if a similar relationship ***did not*** exist in the population from which the sample was drawn. In this case, the relationship is found to be statistically significant at p<0.001.

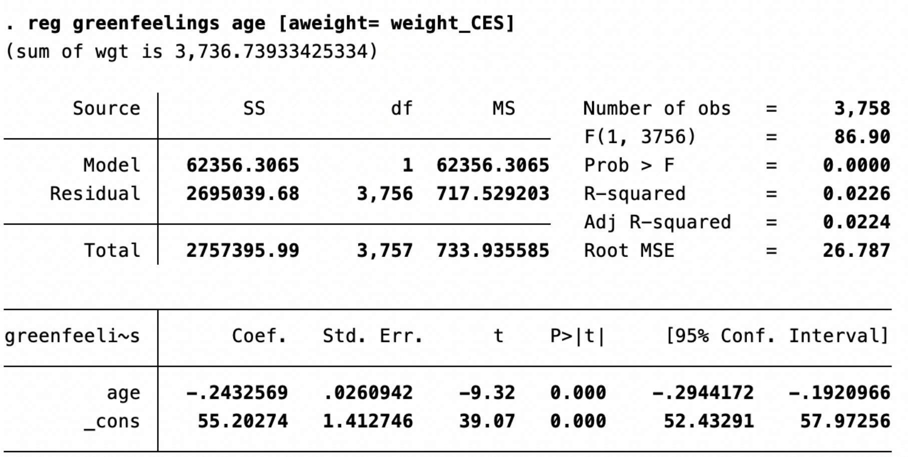
Basic linear regression

While the measure of association and the strength of the relationship between two variables is informative, we can also use information about the independent variable to predict scores on the dependent variable using basic linear regression. Stata allows us to easily produce regression models with the use of the ***regress*** (***reg***) command.

To continue with our example, we can estimate how feelings for the Green party changes for every year increase in age with the following syntax:[9]

reg *greenfeelings age* [**aweight**= *weight_CES*]

[9] Note that we have continued to use the aweight for consistency. We could have also used a pweight with the ***regression*** command, which would have produced the same results.



```
. reg greenfeelings age [aweight= weight_CES]
(sum of wgt is 3,736.73933425334)

      Source |       SS           df       MS      Number of obs   =     3,758
-------------+----------------------------------   F(1, 3756)      =     86.90
       Model |  62356.3065         1  62356.3065   Prob > F        =    0.0000
    Residual |  2695039.68     3,756  717.529203   R-squared       =    0.0226
-------------+----------------------------------   Adj R-squared   =    0.0224
       Total |  2757395.99     3,757  733.935585   Root MSE        =    26.787

------------------------------------------------------------------------------
greenfeeli~s |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.2432569   .0260942    -9.32   0.000    -.2944172   -.1920966
       _cons |   55.20274   1.412746    39.07   0.000     52.43291    57.97256
------------------------------------------------------------------------------
```

Be sure to check your results against ours.

There is considerably more information presented here than in the Pearson's *r* results. Let's walk through some of it:

- Age coefficient. The age coefficient is -0.24. This indicates that for every year increase in age, feelings about the Green party decrease by 0.24 points. We also find that there is a statistically significant relationship between age and feelings about the Green party based on the value reported under the P> |t| column (0.000). We would report this a $p<0.001$ in our written interpretation of the results.
- Intercept: The constant (intercept), 55.20, is the value on the feelings about the Green party variable when age is equal to 0 (an impossible value given that respondents for the CES are a minimum of 18 years of age).
- r^2: How much of the variance in the dependent variable does our independent variable explain? Not much. Recall from Chapter 12 that we can determine the proportion of the dependent variable that can be explained by the independent variable by squaring the Pearson's *r* value, producing a result known as r^2. This value is reported with the Stata output as "R-squared", 0.02 in this example. In other words, using age as a predictor of feelings about the Green party reduces our prediction errors by two percent.

Check-In Point

If you are continuing to follow along by running all of the examples in your own dataset, you now have the ability to create a scatter plot, calculate Pearson's *r*, and conduct basic linear regression with Stata. You have come a long way!

Part X: Assessing Relationships Using Control Variables

Destination

By the end of this section, you will be able to:

- *add control variables to your analyses.*

The final question when assessing a relationship between two variables is to consider what happens to the relationship once other important variables are controlled. We discuss this question fully in Chapter 13, and in this section of the handbook we direct you to the appropriate Stata syntax.

Cross tabs and control variables

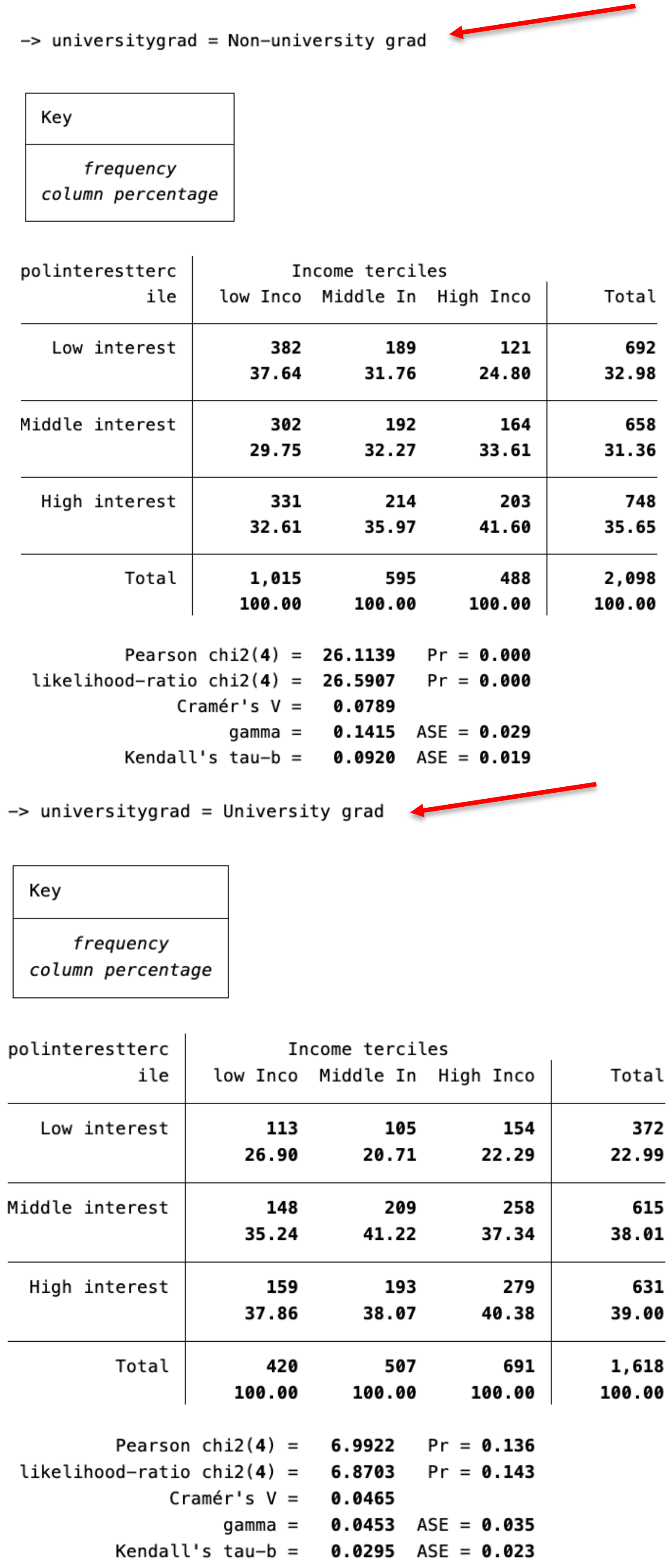
Recall from Chapter 13 that to test a control variable using a crosstabulation, you assess the IV -DV relationship separately for each category of the control and compare these results against those obtained in the original (full) model.

Let's consider the relationship between income and political interest (recall that we recoded these variables previously), controlling for education (university graduate versus non-university graduate). We can use the ***bysort*** command, which instructs Stata to run the subsequent command separately for each sub-group (category) of the variable indicated.[10] To test the income-political interest relationship while controlling for education, we instruct Stata to create crosstabulations for all categories of education. Try it by first generating the dichotomous education variable and then using the ***bysort*** command as follows:

```
*Use the lookfor command to find the interest variable

lookfor education

*Look at the value labels and distribution of the original variable

des q61
label list labels60
tab q61

*Generate, recode and label the new variable

gen universitygrad= q61
recode universitygrad -9/-8=. 1/8=0 9/11=1
lab var universitygrad "University graduate versus non-university graduate"
lab define universitygrad 0 "Non-university grad" 1 "University grad"
lab values universitygrad universitygrad

*Check original and new variable distributions using frequency distribution

tab1 q61 universitygrad, missing

*Generate crosstab with control variable

bysort universitygrad: tab polinteresttercile incometercile, col all
```

This will produce the following results for non-graduates and university graduates, respectively:

[10] Note that Stata will also produce a set of results for cases that do not have a value (".") for the control variable.



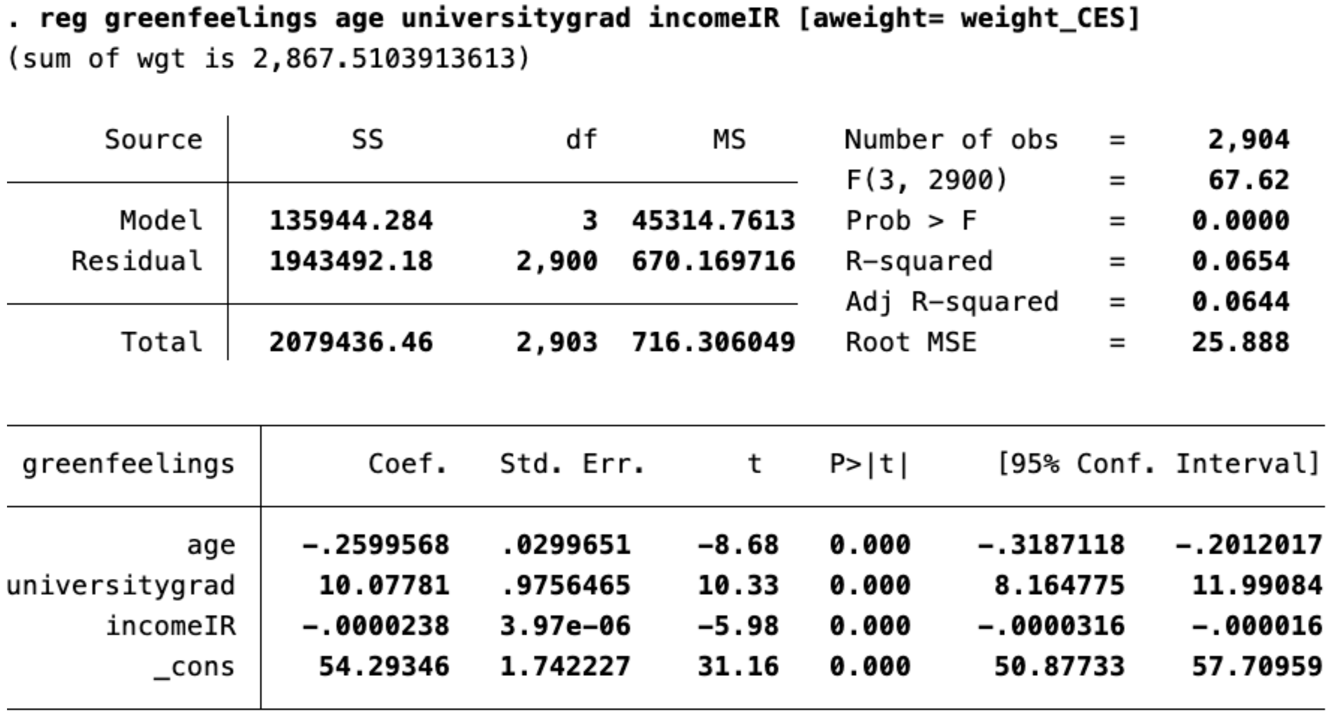
```
-> universitygrad = Non-university grad

+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

polinterestterc |          Income terciles
            ile |  low Inco  Middle In  High Inco |     Total
----------------+---------------------------------+----------
   Low interest |       382        189        121 |       692
                |     37.64      31.76      24.80 |     32.98
----------------+---------------------------------+----------
Middle interest |       302        192        164 |       658
                |     29.75      32.27      33.61 |     31.36
----------------+---------------------------------+----------
  High interest |       331        214        203 |       748
                |     32.61      35.97      41.60 |     35.65
----------------+---------------------------------+----------
          Total |     1,015        595        488 |     2,098
                |    100.00     100.00     100.00 |    100.00

          Pearson chi2(4) =  26.1139   Pr = 0.000
 likelihood-ratio chi2(4) =  26.5907   Pr = 0.000
               Cramér's V =   0.0789
                    gamma =   0.1415  ASE = 0.029
          Kendall's tau-b =   0.0920  ASE = 0.019

-> universitygrad = University grad

+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

polinterestterc |          Income terciles
            ile |  low Inco  Middle In  High Inco |     Total
----------------+---------------------------------+----------
   Low interest |       113        105        154 |       372
                |     26.90      20.71      22.29 |     22.99
----------------+---------------------------------+----------
Middle interest |       148        209        258 |       615
                |     35.24      41.22      37.34 |     38.01
----------------+---------------------------------+----------
  High interest |       159        193        279 |       631
                |     37.86      38.07      40.38 |     39.00
----------------+---------------------------------+----------
          Total |       420        507        691 |     1,618
                |    100.00     100.00     100.00 |    100.00

          Pearson chi2(4) =   6.9922   Pr = 0.136
 likelihood-ratio chi2(4) =   6.8703   Pr = 0.143
               Cramér's V =   0.0465
                    gamma =   0.0453  ASE = 0.035
          Kendall's tau-b =   0.0295  ASE = 0.023
```

The interpretation of the relationship with the inclusion of a control variable is the same as the process you followed to interpret the original relationship, only you need to do so for each category of the control variable. You then compare the results from each of the control variable categories to that of the original relationship to assess whether or not the control variable affects the relationship as anticipated (see Chapter 13 for a full interpretation of the results with the addition of the control variable). Recall that we observed an extremely weak, statistically significant relationship (Tau = 0.08; p<0.001) in the original model, with levels of political interest increasing with income.[11] When we control for education, we find that the relationship is essentially replicated for non-university graduates and disappears when we assess the university graduate group (Tau = 0.03; p=0.14). Accordingly, education does not appear to be a source of spuriousness.

Multivariate linear regression

The syntax for multivariate linear regression in Stata is the same as that for basic linear regression, only we add the additional independent and/or control variables to the model. For example, in addition to age, we can assess how education and income influence feelings about the Green party with the following syntax:

reg greenfeelings age universitygrad incomeIR [aweight= weight_CES]



```
. reg greenfeelings age universitygrad incomeIR [aweight= weight_CES]
(sum of wgt is 2,867.5103913613)

      Source |       SS           df       MS      Number of obs   =     2,904
-------------+----------------------------------   F(3, 2900)      =     67.62
       Model |  135944.284         3  45314.7613   Prob > F        =    0.0000
    Residual |  1943492.18     2,900  670.169716   R-squared       =    0.0654
-------------+----------------------------------   Adj R-squared   =    0.0644
       Total |  2079436.46     2,903  716.306049   Root MSE        =    25.888

------------------------------------------------------------------------------
greenfeelings |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          age |  -.2599568   .0299651    -8.68   0.000    -.3187118   -.2012017
universitygrad |   10.07781   .9756465    10.33   0.000     8.164775    11.99084
     incomeIR |  -.0000238   3.97e-06    -5.98   0.000    -.0000316    -.000016
        _cons |   54.29346   1.742227    31.16   0.000     50.87733    57.70959
------------------------------------------------------------------------------
```

The results show that, holding education and income equal, for every year increase in age, feelings about the Green party decrease by 0.26 points (p<0.001). In the case of education, given that we are using a dichotomous variable, we would interpret the results as indicating that university graduates are more likely (10 points) to have more positive feelings about the Green party than those who have not completed university, net of age and income. While the coefficient for income is statistically significant, the impact on feelings about the Greens is marginal, a decrease of 0.00002 points for every unit increase in income, holding age and education constant. How much of the variance in the dependent variable does our model explain? We can use the value reported as the adjusted R-squared, which takes into account the number of variables in the model, to determine the proportion of the

[11] Note that these results differ from those reported in Chapter 13. This is due to weighting. Stata does not allow the use of non-integer weights to estimate measures of association and statistical significance. As such, we have proceeded without weights for this example.

dependent variable that can be explained by the independent variables. In this example, our model reduces our prediction errors by 6 percent.

Check-In Point and Conclusion

As we come to the end of this introductory Stata handbook, we hope that the procedures outlined here have provided you with the basic skills necessary to conduct your own statistical analyses. We also hope that this introduction has encouraged you to learn more about the many possibilities to use this type of statistical program for your research. As we have noted, this is only a very small sampling of the many options available in Stata, which we hope serves as the starting point for your continued exploration of the possibilities that this software has to offer.