BIOSTATISTICS LABORATORY PART 1:

INTRODUCTION TO DATA ANALYSIS WITH STATA: EXPLORING AND SUMMARIZING DATA

Learning objectives:
Getting data ready for analysis:
  1) Learn several methods of exploring the contents of a dataset.
  2) Learn the commands for labeling variables and values.
First steps of data analysis:
  3) Learn the commands for describing continuous and categorical variables.
  4) Learn the commands to create new variables.

FIRST THINGS FIRST:
- The general format of these labs combines an illustrative example followed by a “Task” that allows you to use the command again on your own. For each example, type in the commands and view the output in STATA’s results window. Compare your output to the laboratory text to ensure your commands were correct.
- Throughout the labs there will be “Notes of Caution” that explain common errors or problems with the commands.
- STATA commands are shown in the courier font – as they actually appear when typed in the STATA program or when viewed in the reference manuals. Each command is also enclosed in a box for easy reference.
- When referring to a generic variable name in a command, we will use varname. However, when typing the command, you should insert the actual name of your variable.

CAVEAT:
- Keep in mind, modern statistical computer packages are a double-edged sword: on one hand, it’s very easy to perform complex analyses; on the other hand, it’s also very easy to do them wrong – leading to incorrect answers. Make sure you understand the assumptions underlying each statistical test and, if you don’t, get help from an epidemiologist or statistician.
EXPLORING DATA:

Before performing any analysis it is important to get to know your data. STATA can help you with this task in several ways.

The general goal of exploring your data is to make sure it was imported without errors and to become familiar with the general layout. Some specific goals include looking for missing values (how they are coded) and unrealistic values (e.g., age of 235) that may influence your results.

Look in the Documentation Accompanying the Data

When you receive a dataset, whether a national sample of hospital discharges or a homegrown case-series, read any accompanying documentation. Specifically, look for the definition of each variable, what the values of each variable mean, and how missing values are encoded. To create your course dataset, I have already done this for the Maryland CABG dataset and added appropriate labels to the STATA version of the dataset on your CD.

More on Missing Values

Missing values are often coded as large numbers (e.g., 999) or large negative numbers (e.g., -999) so it is important to recognize this before doing any analyses.

STATA codes missing values as a single period ("."), and will exclude observations that are missing when performing analyses. We have already converted missing values from the Maryland dataset to the STATA convention (".").

Using the Data Editor

The best place to start looking at your data is to open your Data Editor and directly examine the format and content of your dataset.

Task: Open your Maryland CABG dataset in STATA format (the one provided on the CD) open the Data Editor. Scroll down and across and examine how each variable is coded (dichotomous, continuous, etc...).

Looking in the Data Editor is a good way to ensure all the variables, observations, and variable names were imported without problems. With large datasets it is impossible to examine every observation. But you can establish that the labels were brought in and the data have a uniform appearance within each column.

Useful commands in STATA for Exploring Data:

One useful command to make sure you have all the observations imported is to type *summarize*, which will give you a summary of all your variables.

Type the command:

```
summarize
```
Examining this output will allow you to check the number of observations of each variable. Also, you make sure that the minimum and maximum values are within the plausible range for that variable. For example, a value of age of 325 and a value for death of 999 (coded “0” or “1”) are not plausible. In the case of 999, it is most likely a missing value.

One thing you will notice about this dataset is that the values for most numbers are numeric, i.e., the variable race is coded 1-6. When you look in the Data Editor, however, these appear as words (white, black, etc.). Using the codebook command, you can learn what numbers go with what descriptions (see output below).

The codebook command also provides other useful information about each variable, including the number of total observations, number of missing observations, and the range of values. When typed with an individual variable name after codebook, STATA will tell you about that single variable only. Otherwise, it will list all the variables in your dataset.

Type the command:

```
codebook race
```
SUMMARIZING DATA:

As described in lecture, the first step in any analysis is to generate summary statistics of the variables in your dataset. For continuous variables some measure of central tendency (mean or median) and a measure of spread (standard deviation, range, or inter-quartile range) should be determined. For dichotomous or categorical variables the proportion in each category should be calculated.

**Summarizing continuous variables**

Normally distributed variables – when variables are normally distributed the mean is the preferred measure of central tendency. (I’m using the term “Normal” loosely here – to mean any symmetric or bell-shaped distribution.)

The best way to make sure a variable has a normal distribution is to plot a frequency histogram looking for the bell-shaped curve. In STATA, you can plot the histogram with a superimposed normal curve to make the comparison easy.

General command structure:

```
histogram varname, frequency normal
```

The first part of this command (histogram varname) will make a graph of the variable. The second part, after the comma (frequency and normal), modifies the output from the command. Adding the word “frequency” makes the y-axis a number (rather than “frequency density”) and adding the word “normal” adds the superimposed normal curve. We will first make a histogram of patient age (usually normally distributed).

Type the command: (this may take a few seconds to run)

```
histogram age, frequency normal
```

STATA output:

The age variable distribution is close to the normal curve, and we can summarize it using the mean and standard deviation.
Type the command

```
summarize age
```

STATA output:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>4668</td>
<td>65.78535</td>
<td>10.73579</td>
<td>16</td>
<td>94</td>
</tr>
</tbody>
</table>

Mean age of 65.8 years
Standard deviation of 10.7 years

Variables without a normal distribution – when variables do not have a normal distribution the best measure of central tendency is the median (50\textsuperscript{th} percentile).

Task: Make a histogram of length of stay – is it normally distributed? No, most often length of stay (and costs) have right-skewed distributions (long right-sided “tails”).

Using the following command you can get the median (along with a lot of other information).

Type the command:

```
summarize los, detail
```

STATA output:

<table>
<thead>
<tr>
<th>Percentiles</th>
<th>Smallest</th>
<th>Largest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>3</td>
<td>102</td>
</tr>
<tr>
<td>5%</td>
<td>3</td>
<td>108</td>
</tr>
<tr>
<td>10%</td>
<td>4</td>
<td>108</td>
</tr>
<tr>
<td>25%</td>
<td>5</td>
<td>112</td>
</tr>
<tr>
<td>50%</td>
<td>6</td>
<td>Mean</td>
</tr>
<tr>
<td>75%</td>
<td>9</td>
<td>Std. Dev.</td>
</tr>
<tr>
<td>90%</td>
<td>14</td>
<td>Variance</td>
</tr>
<tr>
<td>95%</td>
<td>20</td>
<td>Skewness</td>
</tr>
<tr>
<td>99%</td>
<td>42</td>
<td>Kurtosis</td>
</tr>
</tbody>
</table>

Note that the mean is significantly larger than the median – this is because of the right-skewed distribution

From this output, you can obtain the median and inter-quartile range (25\textsuperscript{th} to 75\textsuperscript{th} percentile) for the length of stay after CABG in Maryland: median length of stay, 6 days (IQR, 5 to 9 days).
Tabulating dichotomous or categorical variables

The “tab” command in STATA is very useful for finding the proportion of observations in a certain category.

Consider death in the hospital (0 = alive, 1 = dead) – a dichotomous variable. The proportion dead can be found using the following generic command:

```
tab varname
```

What was the mortality rate after CABG in Maryland?

Type the command:

```
tab died
```

STATA Output:

<table>
<thead>
<tr>
<th>Died during hospitalization</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alive</td>
<td>4530</td>
<td>97.19</td>
<td>97.19</td>
</tr>
<tr>
<td>Dead</td>
<td>131</td>
<td>2.81</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>4661</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Task: Use the above command to determine the proportion of patients with Medicare as the primary payer? [Hint: replace the variable “died” in the above command with the variable of interest, “pay1”].
**EXTRA EXERCISES:**

Continue on and perform these extra exercises if you finish the lab early or you can perform these exercises on your own after the course.

**LABELING VARIABLES:**

Labeling variables in STATA is tricky. For some reason, it requires two steps: 1) defining the labels and 2) attaching the labels to the actual values in your dataset.

I have already created labels for most variables. But as an example we will create a label for variable “died”, where 1=dead and 0=alive.

The first step is to create—or define—the label. To keep it simple, when creating label names I usually add the word “label” to the front of the variable name. But the label name could be any word.

Type the command:

```
label define labeldied 1"Dead" 0"Alive"
```

Now, the second step is to label the values of the variable died.

Type the command:

```
label values died labeldied
```

Let’s tabulate the variable so we can see our new labels:

Type the command:

```
tab died
```

**STATA Output:**

```
<table>
<thead>
<tr>
<th></th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alive</td>
<td>4,530</td>
<td>97.19</td>
<td>97.19</td>
</tr>
<tr>
<td>Dead</td>
<td>131</td>
<td>2.81</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>4,661</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
```

We can clearly see that the values are now labeled. But we can also add a label to the variable name as follows:

Type the command:

```
label variable died “Died during hospitalization”
```
**Task:** Tabulate the variable again to make sure the label has been added.

**CREATING AND LABELING NEW VARIABLES:**

You will often need to create new variables using STATA. Next we will create a new variable dividing age (a continuous variable) into discrete categories. The command “generate” is used to create a new variable. For the new age variable, we will create five categories: <50 years, 50-59 years, 60-69 years, 70-80 years, and >80 years. We will call the new variable “age5cat” (since there are 5 categories). Type the commands in the box below: (after each line of text hit enter – you will see some text in the results window but ignore it unless it’s red—which means an error).

```
generate age5cat = 1 if age<50  
replace age5cat = 2 if age>=50 & age<60  
replace age5cat = 3 if age>=60 & age<70  
replace age5cat = 4 if age>=70 & age<80  
replace age5cat =5 if age>=80
```

To make sure the variable was created correctly, tabulate the new variable:

Type the command:

```
tab age5cat
```

**STATA output:**

```
<table>
<thead>
<tr>
<th>age5cat</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>362</td>
<td>7.75</td>
<td>7.75</td>
</tr>
<tr>
<td>2</td>
<td>990</td>
<td>21.21</td>
<td>28.96</td>
</tr>
<tr>
<td>3</td>
<td>1,398</td>
<td>29.95</td>
<td>58.91</td>
</tr>
<tr>
<td>4</td>
<td>1,498</td>
<td>32.09</td>
<td>91.00</td>
</tr>
<tr>
<td>5</td>
<td>420</td>
<td>9.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>4,668</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
```

What do the values 1-5 mean? You may remember now, since you just typed them, but after a few weeks you may forget. Thus, we will generate labels for each category.

Remember, STATA does this in two steps: 1) define the label itself and 2) attach the labels to the values of a variable.

Type the commands: (this will create the label)

```
label define lage5cat 1"<50 years" 2"50-59 years" 3"60-69 years" 4"70-80 years" 5">80 years"
```

Type the commands: (this will attach the labels to the values)

```
label values age5cat lage5cat
```
Also, we will label the actual variable (as opposed to the values):

```
label variable age5cat "Five age categories"
```

Now, test the labeling by making a table by typing the following command:

```
tab age5cat
```

STATA output:

<table>
<thead>
<tr>
<th>Five age categories</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;50 years</td>
<td>362</td>
<td>7.75</td>
<td>7.75</td>
</tr>
<tr>
<td>50-59 years</td>
<td>990</td>
<td>21.21</td>
<td>28.96</td>
</tr>
<tr>
<td>60-69 years</td>
<td>1,398</td>
<td>29.95</td>
<td>58.91</td>
</tr>
<tr>
<td>70-80 years</td>
<td>1,498</td>
<td>32.09</td>
<td>91.00</td>
</tr>
<tr>
<td>&gt;80 years</td>
<td>420</td>
<td>9.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>4,668</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Your output should appear as above. Note that both the variable and the values now have labels. Learning how to label values and variables in STATA is very tricky. Remember that labeling the values (e.g., 1,2,3,4,etc…) requires two steps: 1) defining the label and then 2) adding the label to the values in your dataset. Labeling the variable itself—as opposed to the values—requires only one step. Labeling the variable allows you to give it a more descriptive name (such as “Five age categories” as in the example below).

Now we are done with Part 1 of the laboratory. You can save any changes to your STATA dataset by going to “File” in the menu bar and scrolling down to “Save”.

**IMPORTING DATA INTO STATA:**

We have provided data for all coronary artery bypass surgery performed in Maryland during 2001 on a CD. The dataset is provided in both an Excel and STATA format. We will introduce two basic methods for getting a dataset from Excel into STATA.

**Copy and Paste Data:**

1) Open the provided Excel file “CABG.Maryland”. To select all the data click in the upper-left hand box (above the “1” and to the left of the “A”). Or you can manually select the entire dataset. Then go to the Edit menu and Copy the data.

2) Open STATA and the Data Editor (An empty dataset – not the STATA file that already has the data). Click on the upper left hand cell (so it turns blue) and choose Edit→Paste from the menu. The data should appear in the data editor along with variable names. (Remember to close the Data Editor before trying any commands).

3) Close STATA and do not save this file – you might confuse this file with the STATA file used in the lab. (The STATA file provided on the CD already has variable labels.)
Using the `insheet` command:

The `insheet` command is typed into the STATA command window and brings data directly into the Data Editor. The tricky thing about the `insheet` command is typing the “path” to the file on your computer – luckily STATA can make this easy.

1) Open the Excel file and, using the Save As option, save it as a text (tab delimited) file. Save this somewhere on your computer where you’ll be able to find it easily. You will encounter two dialog boxes when you save your dataset. Click “OK” and “Yes” when they appear.

2) Open STATA and type the following in your command window:

```stata
insheet using "your filename will go here"
```

3) To insert the name of your file, go to the File option in the drop-down menu of STATA. Choose Filename and you should now be able to browse to find your new text file.

4) Use the window to find your stored datasets (CABG.Maryland.txt). Remember to change the file type to All files, so your text file will be visible.

5) Double click (or use the Open option) on your data file and the complete path will appear in your STATA command window. It should look something like the following: (But, of course, the exact path will be different for every computer.)

```stata
insheet using "C:\Documents and Settings\CABG.Maryland.txt"
```

6) Now hit enter and STATA should import (or insheet) your text file. To make sure, open the data editor and visualize the data. Now close STATA and **do not save** this dataset either.