

Chapter 2

Data Description

Aim: Upon completing the chapter, the learner should be able to describe a database with the specific commands of Stata.

2.1 Most Useful Commands

Although specific reference is made to the use of the menus and dialog windows of the program, it is important to understand how to manage the different conditions and options that are available for each Stata command. Most Stata commands follow the same basic sequence:

```
<command> <variable or variables list> <condition or use of  
if>, <options>
```

A list containing several of the commands and their corresponding descriptions follows below:

list	Lists the values of the variables
describe	Produces a summary of the dataset in memory or of the data stored in a Stata-format dataset
codebook	Examines the variable names, labels, and data to produce a codebook for describing the dataset
generate	Creates a new variable in the dataset
recode	Changes the values of the numeric variables according to the rules specified

replace	Changes the contents of an existing variable
label	Can be used for several purposes, but is mainly used for attaching labels to data, variables, or values
drop	Eliminates variables or observations from the data that are in the memory
summarize	Calculates and displays a variety of univariate summary statistics
tabulate	Produces one-way or two-way tables of frequency counts
table	Calculates and displays tables of statistics

2.2 list Command

The *list* command displays the values of all the components of the database requested in the command line. If the user wants to view a specific variable of the database, the user must first write the word *list*, followed by the condition *in* and then write the number of observations to be viewed on the screen, as shown in the following:

Output

```
. list in 5/10
+-----+
|  var1  | var2 | var3 | var4 | var5 |
+-----+
5. |    5   |   45  |   56  |  1.52 |    1  |
6. |    6   |   36  |   87  |  1.46 |    1  |
7. |    7   |   30  |   78  |  1.44 |    1  |
8. |    8   |   29  |   77  |  1.56 |    1  |
9. |    9   |   27  |   67  |  1.52 |    0  |
+-----+
10. |   10   |   29  |   63  |  1.52 |    1  |
+-----+
```

Only observations 5–10 of the active database are displayed.

2.3 Mathematical and Logical Operators

To carry out different mathematical or logical operations with numbers or variables of the active database, the following symbols are available:

Symbol	Definition
<code>==</code>	Equal to (double equal sign)
<code>!=</code> or <code>~=</code>	Not equal to
<code>></code>	Greater than
<code>>=</code>	Greater than or equal to
<code><</code>	Less than
<code><=</code>	Less than or equal to
<code>+</code>	Addition (e.g., $2 + 3$)
<code>-</code>	Subtraction (e.g., $2 - 3$)
<code>*</code>	Multiplication (e.g., $2 * 3$)
<code>/</code>	Division (e.g., $2/3$)
<code>^</code>	Exponentiation (e.g., $2 ^ 3$)
<code>&</code>	<i>and</i> (assuming two conditions, the symbol <code>&</code> is used to indicate that both conditions are occurring simultaneously.)
<code> </code>	<i>or</i> (assuming two conditions, the symbol <code> </code> is used to indicate that one or the other or both conditions are occurring; that is, that at least one of them is occurring.)

Usually, these operators are associated with the conditional command *If* for specific variables. For example, to display only those observations in which the age is below 30, the command line is as follows:

```
list id age weikg heimt if age<30
```

Output

	id	age	weikg	heimt
1.	1	28	59	1.55
3.	3	25	76	1.6
4.	4	26	65	1.78
8.	8	29	77	1.56
9.	9	27	67	1.52
10.	10	29	63	1.52

The symbol of asterisk (*) is also used to make any comment during the Stata programming; for example:

```
*Displaying observations in which the age is below 30
list id age weikg heimt if age<30
```

2.4 generate Command

The *generate* command (or *gen*) is used to define new variables in an existing database. For example, let us suppose that you have a database of anthropometric measurements (corresponding to the hypothetical participants of your study), such as weight in kilograms (*weikg*) and height in meters (*heimt*), and you want to calculate the body mass index (*bmi*) of each participant, with *bmi* being defined as the ratio of *weikg* over *heimt* squared. Suppose, further, that the following database is active in Stata:

	id	age	weikg	heimt	sex
1.	1	28	59	1.55	0
2.	2	32	35	1.35	0
3.	3	25	76	1.6	0
4.	4	26	65	1.78	0
5.	5	45	56	1.52	1
6.	6	36	87	1.46	1
7.	7	30	78	1.44	1
8.	8	29	77	1.56	1
9.	9	27	67	1.52	0
10.	10	29	63	1.52	1

To compute and display the *bmi* of each participant, the following commands are executed:

```
gen bmi = weikg/(heimt^2)
list id bmi
```

You can see that a new variable, named *bmi*, has been created as a result of using the *list* command:

Output

	id	bmi
1.	1	24.55775
2.	2	19.20439
3.	3	29.6875
4.	4	20.51509
5.	5	24.23823
6.	6	40.81441
7.	7	37.61574
8.	8	31.64037
9.	9	28.99931
10.	10	27.26801

2.5 *recode* Command

The *recode* command allows the user to change or regroup the values of any variable. For example, let us assume that a user wants to regroup the values of the *bmi* variable of the previous database in the following three groups: (1) Group 1 contains the values ranging from 18.5 to 24.9, (2) Group 2 are those that range from 25 to 29.9, and (3) Group 3 are the values that are 30 or greater. The commands sequence is as follows:

```
gen bmig=bmi
recode bmig 18.5/24.9=1 25/29.9=2 30/max=3
list id bmig
```

Output

	id	bmig
1.	1	1
2.	2	1
3.	3	2
4.	4	1
5.	5	1
6.	6	3
7.	7	3
8.	8	3
9.	9	2
10.	10	2

2.6 *drop* Command

The *drop* command allows us to eliminate from one to several variables from our active database. For example, to eliminate the *bmi* variable, the following command must be used:

```
drop bmi
```

2.7 *replace* Command

This command allows you to change the value of an existing variable according to the rule specified. For example, let us assume that a user wants to categorize *bmi* into the following groups: Group 1 (persons with a *bmi* from 18.5 to 24.49), Group 2 (persons with a *bmi* from 24.5 and 29.9), and Group 3 (persons with a *bmi* from 30 onward). The command sequence to create these categories of *bmi*, assuming that *bmi* was not already created, is as follows:

```
gen bmi= wei/kg/(heimt^2)
gen bmig=bmi
replace bmig=1 if bmi >= 18.5 & bmi < 24.5
replace bmig=2 if bmi >= 24.5 & bmi < 30
replace bmig=3 if bmi >= 30
list id bmig
```

After the *list* command, the results will be the same as that reported with the *replace* command.

2.8 *label* Command

The *label* command defines a name to the variables of the active database; for example, the *label variable* command assigns a label to the variable *bmig* as follows:

```
label variable bmig "body mass index categories"
```

In addition, the *label* command decodes the categories of the variables, combining *label define* and *label value* commands. The *label define* command is used to create a label for different codes to be attached to a legend. Then, the *label value* command is used to relate the categories of 1 variable to the labels defined in *label define* command. For example, the command lines that are used to label the codes of the variables *sex* and *bmig* are as follows:

```
label define sexc 0 "Male" 1 "Female"
label value sex sexc
```

```
label define bm 1 "Normal" 2 "Overweight" 3 "Obese"  
label value bmig bm  
list id sex bmig
```

After using the *list* command, the following output will be displayed:

	id	sex	bmig
1.	1	Male	Overweight
2.	2	Male	Normal
3.	3	Male	Overweight
4.	4	Male	Normal
5.	5	Female	Normal
6.	6	Female	Obese
7.	7	Female	Obese
8.	8	Female	Obese
9.	9	Male	Overweight
10.	10	Female	Overweight

If you want to eliminate a label that was previously assigned to a variable, the *drop* command must be used, as follows:

```
label drop sex
```

And to eliminate all of the assigned labels, write the following:

```
label drop all
```

2.9 summarize Command

If we want to summarize the variables in a database, we must write *summarize* (or *sum*) in our command window. After we do this, a table containing a summary of all the variables in our database appears. It is recommended that this command be used with quantitative variables. For example, in the previous database, age, weight (*weikg*), height (*heimt*), and bmi are defined as quantitative variables; therefore, the command would be as follows:

```
sum age weikg heimt bmi
```

Output

Variable	Obs	Mean	Std. Dev.	Min	Max
age	10	30.7	5.9264	25	45
weikg	10	66.3	14.62912	35	87
heimt	10	1.53	.1127435	1.35	1.78
bmi	10	28.45408	6.925241	19.20439	40.81441

As a result, in the command window, a table is displayed containing the number of observations, mean (*Mean*), standard deviation (*Std. Dev.*), minimum (*Min*), and maximum (*Max*) for each variable. If the user is interested in displaying descriptive statistics for certain conditions, the conditional command *if* can be used. For example, a statistical description of the *bmi* in subjects less than 30 years old, the following procedure can be used:

`sum bmi if age < 30`

Output

Variable	Obs	Mean	Std. Dev.	Min	Max
bmi	6	27.11134	4.01918	20.51509	31.64037

The *detail* command can be written at the end of the command line to obtain information, which is more detailed, about quantitative variables in the database. For example, assuming we want the detailed information of the distribution of the variable *bmi*, the following command line can be used:

`sum bmi, detail`

Output

bmi				

Percentiles		Smallest		
1%	19.20439	19.20439		
5%	19.20439	20.51509		
10%	19.85974	24.23823	Obs	10
25%	24.23823	24.55775	Sum of Wgt.	10
50%	28.13366		Mean	28.45408
		Largest	Std. Dev.	6.925241
75%	31.64037	29.6875		
90%	39.21507	31.64037	Variance	47.95897
95%	40.81441	37.61574	Skewness	.4458304
99%	40.81441	40.81441	Kurtosis	2.272217

2.10 do-file Editor

A do-file is a set of Stata commands that can be stored for later use. In the Stata toolbar, click on the icon  to create a do-file. A window will open (Figure 2.1), and you can either type or paste a series of commands and then save this as a file for later use. To execute these commands, totally or partially, click on the *Do* icon, , in this editor, which is located in the extreme right corner.

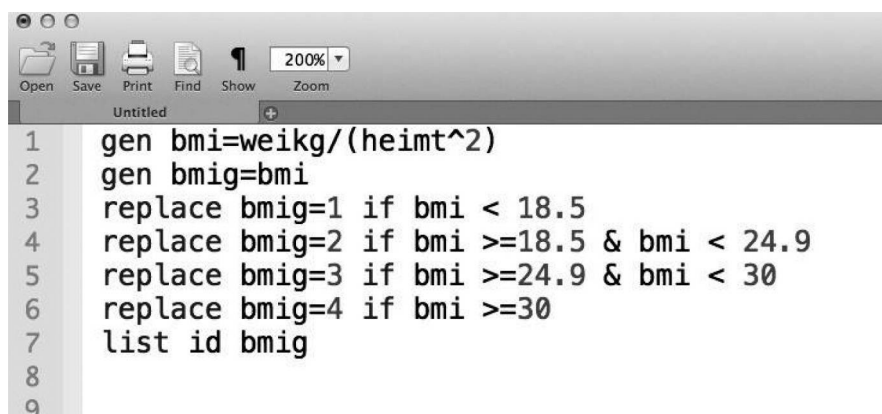
Once the sequence of Stata commands is defined for the first time, a name has to be assigned to this do-file for later use. Stata assigns the extension `.do` to this file name.

2.11 Descriptive Statistics and Graphs

To generate a table of descriptive statistics, on the main taskbar, click **Statistics** → **Summaries, tables, and tests** → **Other tables** → **Compact table of summary statistics** (Figure 2.2).

When you click this sequence, a window opens that lets you select the variable of the active database that will be analyzed and choose the statistical procedure of interest. Within this window, we can assess (in terms of mean, standard deviation, coefficient of variation, 25th percentile [p25], 75th percentile [p75], and interquartile range [p75–p25]) the statistical distribution of a quantitative variable, as shown for *bmi* from the previous database (Figure 2.3).

If the *Statistics* icon is used, Stata displays the command and the results. The output above indicates that the value of the sd (*standard deviation*) is only 24.3% of the mean, which might suggest a relatively moderate variability in the *bmi*



```

1  gen bmi=weikg/(heimt^2)
2  gen bmig=bmi
3  replace bmig=1 if bmi < 18.5
4  replace bmig=2 if bmi >=18.5 & bmi < 24.9
5  replace bmig=3 if bmi >=24.9 & bmi < 30
6  replace bmig=4 if bmi >=30
7  list id bmig
8
9

```

Figure 2.1 do-file editor window.

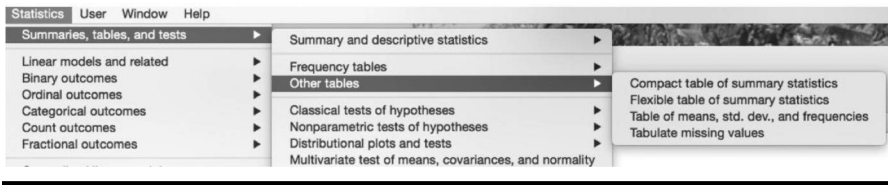


Figure 2.2 Creating a table to display summary statistics.

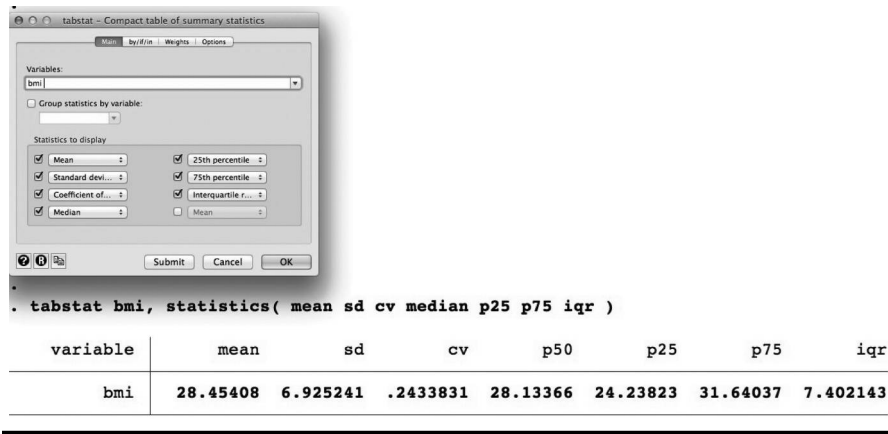


Figure 2.3 Window for displaying a table of summary statistics.

distribution. Based on the *iqr* (*interquartile range*), the output indicates that 50% of the *bmi* around the median value is not greater than 7.4.

2.12 *tabulate* Command

The *tabulate* (or *tab*) command provides a table with the frequency values of the corresponding variable. For example, to obtain the frequency distribution of the grouped *bmi* (*bmig*), the user needs to write the following:

`tab bmig`

Output

<i>bmig</i>	<i>Freq.</i>	<i>Percent</i>	<i>Cum.</i>
Normal	3	30.00	30.00
Overweight	4	40.00	70.00
Obese	3	30.00	100.00
Total	10	100.00	

In this example, 30% of the study group was categorized as being obese and 40% as being normal.

The *tab* command can be used to report contingency tables that, in turn, can be used to report the frequency distribution, with the option of including percentages by column and row. For example, to describe the association between the variables *bmig* and *sex* (see the previous database), use the *tab* command, as follows:

```
tab bmig sex, co
```

Output

+-----+			
Key			
frequency			
column percentage			
+-----+			
bmig	sex		Total
	Male	Female	
Normal	2 40.00	1 20.00	3 30.00
Overweight	3 60.00	1 20.00	4 40.00
Obese	0 0.00	3 60.00	3 30.00
Total	5 100.00	5 100.00	10 100.00

The results show that 80% of women are categorized as being either overweight or obese, while 40% of men are categorized as being overweight, with none being categorized as being obese. Only 30% of the subjects (both sexes) are categorized as being of normal weight.