

Answers to Application Activities in Chapter 1

1.1.3 Application Activities for Getting Started with SPSS

1 DeKeyser (2000)

The names of the variables are: Age, GJTScore, and Status. These can be read from the “Data View” spreadsheet. To find out what values are assigned to the numbers in the “Status” variable, tab to the “Variable View” on the bottom of the spreadsheet. In the row for “Status” under the column for “Values,” click on this box and a grey button appears. Click the button, and you’ll see that 1 = under15 and 2 = over15.

2 Make a new file

Open a new file by going to FILE>NEW>DATA. A spreadsheet opens with the “Variable View” already open. To label the variables, in the “Variable View” enter the names for your variable in the rows under the “Name” column. To give the variables “Group” and “Gender” value labels, go to the “Values” column in the “Variable View” and click on the box pertaining to the correct row. A grey box appears and when you open it you will now be able to input labels that link to each number.

3 Customize Variable View

With a file open, go to EDIT > OPTIONS and choose the DATA tab. Click the CUSTOMIZE VARIABLE VIEW button. I clicked off the last five choices (from MISSING to ROLE) as I do not customarily use these. Note: You can do this in version 15 and change a couple of settings, but

there is no CUSTOMIZE VARIABLE VIEW tab and there are not as many options as in version 16 and higher.

1.1.6 Application Activity for Importing and Saving Files

1 read.txt

If you have downloaded the SPSS files to your desktop, go to FILE > READ TEXT DATA, and find the file on your computer. A window titled “Open Data” will start. Pull down the menu for “File of Type” and choose .txt files. Open the .txt file, and a “Text Import Wizard” will start.

For Step 1, leave the radio button as it is (“No”) and click the NEXT button. In Step 2, variables are delimited (separated) by a space, so leave the default checked. Variable names ARE found in the top row of the file, so change the second radio button to Yes and click NEXT. In Step 3, since the first case of the data starts on line 2 (since line 1 was variable names), leave the first box that says “2” alone. Each line of the data represents a case, and we want to import all cases, so we’ll leave everything alone on this step and click NEXT. In Step 4, the Wizard has automatically detected that we’re using a space to separate (delimit) variables, so leave that box checked. In this step you can preview what the SPSS file will look like, so if it looks crazy you may have to back up (by the way, if the original file hasn’t separated the data out in a consistent way, there’s nothing you can do in the Text Import Wizard to make it turn out! For example, if some columns are separated with 1 space but others with 2 spaces, the Wizard cannot handle that and you’ll have to go to the source file and fix it before you can successful import it).

My file looks fine, so I click NEXT. Step 5 lets you change variable names but the file came with some already that I won’t change. If you want to change names, click on the names of the variable in the “Data preview” and the new name can be entered in the “Variable name” box. I

see that variables Ran2T and Speed are defined as Strings, so I will change them to Numeric in the “Data format” box. Then I click NEXT. I don’t want to save syntax for future use, so I click FINISH. A new Data Editor comes up.

2 Save Data

To save your Data Editor, go to FILE > SAVE or click the icon of the floppy on the menu bar. Navigate to the place you want to save it, and notice that it will be saved as a .sav file. Do the same thing with the Viewer for the output files, but notice now that the file is a .spv file (in version 15 and earlier this is a .spo file).

1.3.7 Application Activity Practicing Entering Data into R

1 Create New Dataset

Using R Commander, follow the menu sequence DATA > NEW DATA SET (remember, if you only have the R console open, to get R Commander open you will need to type the sequence `library(Rcmdr)` first). Enter the name **Count** in the first box that appears. A spreadsheet pops up. Click in the box that says “V1” and type **Score**. Press the “Add column” button to get another column. Name this column **Group**. Press the “Add row” button enough times to get 10 rows. Enter your numbers as explained in the exercise.

To create a new dataset in the R console, type:

```
Count=data.frame()
```

```
fix(Count) #this will open up an empty spreadsheet
```

Another command which can open an already-created dataset which has some data in it is:

`editDataset(Count)`

This command opens up a fancier-looking spreadsheet than the previous command.

2 read.txt

Using R Commander, follow the menu sequence DATA > IMPORT DATA > FROM TEXT FILE, CLIPBOARD OR URL. Replace the default name for the file with `read`. Keep the check mark on the box “Variable names in file,” but change the Field Separator to “White space.” Click OK. Navigate to where you have placed the file, and open it. Use the “View data set” button on R Commander to view it. It should have opened with 12 columns, all topped by the appropriate name.

3 DeKeyser2000.sav

Using R Commander, follow the menu sequence DATA > IMPORT DATA > FROM SPSS DATA SET. Replace the default name with `dekeyser` and then press OK. Navigate to where the file exists on your computer and open it. Go back to R Commander and verify that the dataset looks good.

1.4.2 Using R as a Calculator Practice Activities

1 Answer:

`88+2689+331+389+2`

`[1] 3499`

2 Answer:

`25338-89.32`

[1] 25248.68

3 Answer:

$(59 \cdot 26) / \sqrt{24}$

[1] 313.1264

4 Answer:

$2500 - (800 + 500 + 150 + 136 + 55 + 260 + 35 + 225)$

[1] 339

5 Answer:

$((275 + 38)^3) / 6$

[1] 5110716

6 Answer:

$((258367 / 268) + 245 + 897) \cdot (4^{20})$

[1] 2.315633e+15

(which means take the decimal point and move it over to the right 15 places, adding zeros so that you can! The result is 23156330000000000)

7 Answer:

Hit the arrow up button to go back to your original calculation in #4 and easily change the 800 to 850, then add the 3 more expenses to the end of the calculation (this might be faster than retyping it!).

$2500 - (850 + 500 + 150 + 136 + 55 + 260 + 35 + 225 + 63.24 + 35.10 + 180)$

[1] 10.66

8 Answer:

$((42 \cdot 84) / (90 + 266 + 35))^3$

9 [1] 734.6077

1.4.4 Creating objects in R practice activity

- 1 Answer: No, the years are in the same order you entered them in!

```
Years=c(1976 ,2003,2009, 1900)
```

```
Years
```

- 2 Answer: Names are marked with double quotes, no matter how you entered them.

(for example)

```
MyFamily=c('Jenifer', 'Calvin')
```

```
MyFamily
```

```
[1] "Jenifer" "Calvin"
```

- 3 (for example)

```
Salary=c(2500)
```

```
Salary
```

```
[1] 2500
```

```
Expenses=c(700, 350, 225, 96, 42, 33)
```

```
Budget=c(Salary-sum(Expenses))
```

```
Budget
```

```
[1] 1054
```

1.4.6 Types of Data Practice Activities

1 EQ.dominance

First, import `beq` using R Commander sequence DATA > IMPORT DATA > FROM SPSS DATA SET and call it `beq`.

```
class(beq)
```

```
[1] "data.frame"
```

2 BEQ.dominance

```
names(beq)
```

```
[1] "catdominance" "numberoflang" "sex" "age"
```

```
class(beq$catdominance)
```

Yes, it is a factor!

3 BEQ.dominance

```
summary(beq$catdominance) #There are 4 levels
```

```
YES NO YESPLUS NA's
```

```
558 105 373 542
```

4 Classes of objects

```
class(MyFamily)
```

```
[1] "character"
```

```
class(Salary)
```

```
[1] "numeric"
```

Answer: Character vectors have quotes around them and are categorical variables. They are not used in statistical calculations. Numeric vectors contain numbers that can be added, subtracted, etc.

1.5.8 Functions in R practice activities

1 DeKeyser2000.sav

First, import the DeKeyser.2000 file using R Commander sequence DATA > IMPORT DATA > FROM SPSS DATA SET and call it `dekeyser`.

```
var(dekeyser$gjtsscore)
```

```
[1] 876.3127
```

```
sd(dekeyser$gjtsscore)
```

```
[1] 29.60258
```

```
29.60258^2
```

```
[1] 876.3127
```

2 DeKeyser2000.sav

```
mean(dekeyser$age)
```

```
[1] 21.56140
```


3 DeKeyser2000.sav

```
max(dekeyser$age)
```

```
[1] 40
```

4 LarsonHallPartial.sav

Import LarsonHallPartial.sav using R Commander sequence DATA > IMPORT DATA > FROM SPSS DATA SET and name this file `partial`. If you try to use the command `length()` on the entire dataframe you'll get the answer 5, which is the number of variables. You'll want to use the command for *one particular vector* to get the N size for this file. Use the `names()` function to see what the variables are.

```
names(partial)
```

```
[1] "id" "age" "LOR" "aptitude" "R_L_accuracy"
```

```
length(partial$id)
```

```
[1] 15 # 15 participants
```

```
mean(partial$r_l_accuracy)
```

```
[1] 3.783389
```

```
sd(partial$r_l_accuracy)
```

```
[1] 0.722244
```

1.5.2 Missing data application activity in SPSS

1 ClassTest

Go to FILE > NEW > DATA and type in the numbers. Then go to the VariableView tab and write in the names of the variables. Go to FILE > SAVE and save it as CLASSTEST.

2 ClassTest Imputation

Go to ANALYZE > MISSING VALUE ANALYSIS and put all 3 variables into the “Quantitative Variables” window. Press the “Patterns” button. Tick on “Cases with missing values” and press Continue, then OK. 14.3% is missing out of the last 2 variables (0% in the first).

To impute the data, go to ANALYZE > MULTIPLE IMPUTATION > IMPUTE MISSING DATA VALUES. Move all the variables over to the “Variables in Model” window. Then name the new dataset. I named mine CLASSTESTIMPUTED. Press OK. A new window should open with this file and you’ll see the imputed data in every imputation iteration from 1–5. You can easily see which values have been imputed by looking for the highlighted cells.

1.5.4 Missing data application activity in R

1 ClassTest creation

Using R Commander, follow the menu sequence DATA > NEW DATASET (remember, if you only have the R console open, to get R Commander open you will need to type the sequence `library(Rcmdr)` first). Enter the name “ClassTest” in the first box that appears. A spreadsheet pops up. Create three columns, and name the first “Pretest”, the second “Posttest”, and the last

“DelayedPosttest.” Create enough rows to enter the data in the spreadsheet. When you are finished, close the window. To check for missing data, in the R Console type:

```
is.na(ClassTest)
```

You should see two entries that say “TRUE”.

To save the file, using R Commander: First, make sure your **ClassTest** is the active dataset by looking at the window in R Commander. Then follow the menu sequence DATA > ACTIVE DATASET > EXPORT ACTIVE DATASET. Click on “Commas” for Field Separator. Press OK. Now you will be able to name the file and place it where you want. Add .csv after the name of the file. Click OK. If you now navigate to where you saved your file, you will see your file marked as an Excel spreadsheet and that is what it will open up as if you click it.

2 ClassTest Imputation

Open AmeliaView in R Console:

```
library(Amelia)
```

```
AmeliaView()
```

You can now load the .csv file you saved called **ClassTest**. Press the Missingness Map button on the AmeliaView GUI to see the map.

To impute the data, open the `mice` package:

```
library(mice)
```

```
imp<-mice(ClassTest, 5)
```

```
complete(imp)
```

```
impClassTest<-complete(imp)
```

Look at the new file with the AmeliaView Missingness Map again.