Lab #2: Access Data

Jacob Jameson

It is expected you watch the Module 2 material, here prior to this lab.

In this lab, you will work with 3 data sets from the ISLR textbook, which you can download here.

General Guidelines:

You will encounter a few functions we did not cover in the lecture video. This will give you some practice on how to use a new function for the first time. You can try following steps:

1. Start by typing `?new_function` in your Console to open up the help page
2. Read the help page of this `new_function`. The description might be too technical for now. That's OK. Pay attention to the Usage and Arguments, especially the argument `x` or `x`,`y` (when two arguments are required)
3. At the bottom of the help page, there are a few examples. Run the first few lines to see how it works
4. Apply it in your lab questions

It is highly likely that you will encounter error messages while doing this lab Here are a few steps that might help get you through it.

1. Locate which line is causing this error first
2. Check if you may have a typo in the code. Sometimes another person can spot a typo faster than you.
3. If you enter the code without any typo, try googling the error message
4. Scroll through the top few links see if any of them helps

Warm-up

1. Create a new Rmd and add code to load the `tidyverse` package.

2. Your classmate comes to you and says they can't get data to load after restarting their R session. You see the code:

```
install.packages("haven")
awesome_data <- read_dta("awesome_data.dta")

Error in read_dta("awesome_data.dta") : could not find function "read_dta"
```

Diagnose the problem.

Note: If they say the code worked before, it's likely they had loaded haven in the console or perhaps in an earlier script. R packages will stay attached as long as the R session is live.

3. In general, once you have successfully used `install.packages(pkg)` for a "pkg", you won't need to do it again. Install `haven` and `readxl` using the console.

4. In your script, load `haven` and `readxl`. Notice that if you had to restart R right now. You could reproduce the entire warm-up by running the script. We strive for reproducibility by keeping the code we want organized in scripts or Rmds.

5. It’s good practice when starting a new project to clear your R environment. This helps you make sure you are not relying on data or functions you wrote in another project. After you `library()` statements add the following code `rm(list = ls())`.

6. `rm()` is short for remove. Find the examples in `?rm` and run them in the console.

ISLR Chapter 2 Q8

This exercise relates to the College data set, which can be found in the file `College.csv`. It contains a number of variables for 777 different universities and colleges in the US. The variables are

Variable	Description
`Private`	Public/private indicator
`Apps`	Number of applications received
`Accept`	Number of applicants accepted
`Enroll`	Number of new students enrolled
`Top10perc`	New students from top 10 % of high school class
`Top25perc`	New students from top 25 % of high school class
`F.Undergrad`	Number of full-time undergraduates
`P.Undergrad`	Number of part-time undergraduates
`Outstate`	Out-of-state tuition
`Room.Board`	Room and board costs

Variable	Description
Books	Estimated book costs
Personal	Estimated personal spending
PhD	Percent of faculty with Ph.D.'s
Terminal	Percent of faculty with terminal degree
S.F.Ratio	Student/faculty ratio
perc.alumni	Percent of alumni who donate
Expend	Instructional expenditure per student
Grad.Rate	Graduation rate

Before reading the data into R, it can be viewed in Excel or a text editor. Make sure that you have the directory set to the correct location for the data.

a. Use the base R `read.csv()` function to read the data into R with option `stringsAsFactors=T` (this is needed later on for plotting figures). Call the loaded data `college`.

b. Look at the data using the `View()` function. You should notice that the first column is just the name of each university. Load your data and then try the following commands:

```
#set your working directory ,fill in your code after this line

#read in the file College.csv using read.csv() with option `stringsAsF
        actors=T`
college <- read.csv('College.csv', stringsAsFactors = T)

#Give data frame college rownames
rownames(college) <- college[,1]
```

```
# Please comment out View function after using it. Otherwise you'll se
        e some error when knit.
# View(college)
```

c. You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will **not** try to perform calculations on the row names. Next, we will remove the first column in the data where the names are stored. Try

```
#Use a negative number to generate a subset with all but one column
# college[, -c(1, 2, 3)]  will generate a subset with all but the firs
        t three columns
college <- college[,-1]
# as.factor can turn a character column to a factor column so that we
         can use it to plot later on
college$Private <- as.factor(college$Private)
#View(college)
```

Now you should see that the first data column is Private. Note that another column labeled row.names now appears before the Private column. However, this is not a data column but rather the name that R is giving to each row.

i. Use the `summary()` function to produce a numerical summary of the variables in the data set. Hint: `summary()` takes in an object such as data.frame and return the summery results

ii. Use the `pairs()` function to produce a scatterplot matrix of the first five columns or variables of the data. Recall that you can reference the first five columns of a data frame dat using `dat[,1:5]`

iii. Use the `plot()` function to produce side-by-side boxplots of Outstate versus Private. Hint: `plot()` takes two arguments one vector for x axis and one vector for y axis. Try `plot(dat$col_name, dat$col_name)`.

```
# replicate "No" for the same times as the number of colleges using re
        p()
Elite <- rep("No",nrow(college))
```

```
# change the values in Elite for colleges with proportion of students
# coming from the top 10% of their high school classes
# exceeds 50 % to "Yes"
Elite[college$Top10perc >50] <- "Yes"
# as.factor change ELite, a character vector to a factor vector
# (we will touch on factors later in class)
Elite <- as.factor(Elite)
# add the newly created vector to the college data frame
college <- data.frame(college ,Elite)
```

iv. Use the `summary()` function to see how many elite universities there are. Now use the `plot()` function to produce side-by-side boxplots of Outstate versus Elite.

Continue exploring the data, and provide a brief summary of what you discover.

ISLR Chapter 2 Q9

This exercise involves the Auto data set. `na.omit()` removes the missing values from the data and returns a new data frame.

```
#load the Auto.csv into a variable called auto using read_csv()
```

```
# remove all rows with missing values using na.omit()
auto <- na.omit(auto)
```

We can use `class()` to check which of the columns are quantitative (numeric or integer), and which are qualitative(logical or character). And `sapply()` function takes in a data frame and a function (in this case `class()`), apply the class function to each column. Try the following commands:

```
#apply the class() function to each column of auto data frame
sapply(auto, class)
```

a. What is the range of each quantitative columns? You can answer this using the `range()` function. *Hint: You can call `range()` function individually on each column. You can also subset the quantitative columns by creating a variable `quant_cols` equal to all columns with a numeric mode, then use `sapply` the function `range()` with the data frame with only quantitative columns. This is not required.*

b. Using the functions `mean()` and `sd()`. Find out what is the mean and standard deviation of each quantitative columns?

c. Now remove the 10th through 85th observations (rows). What is the range, mean, and standard deviation of each column in the subset of the data that remains? *Hint: We've seen removing columns in question 8. To remove the rows, we can use the negative sign - again. For example, `auto[-c(1,3),]` removes the first and third row*

d. Using the full data set, investigate the columns graphically, using scatterplots (pairs or plot) or other tools of your choice. Create some plots highlighting the relationships among the columns Comment on your findings.

e. Suppose that we wish to predict gas mileage (mpg) on the basis of the other numerical variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

ISLR Chapter 2 Q10

This exercise involves the Boston housing data set.

To begin, load in the Boston data set. The Boston data set is part of the `MASS` library in R. You may need to install the package using `install.packages()` function if you haven't done so.

```r
# install.packages(MASS)
library(MASS)
```

Now the data set is contained in the object Boston.

```
Boston
```

Read about the data set:

```
?Boston
```

a. How many rows are in this data set? How many columns? What do the rows and columns represent?

b. Make some pairwise scatterplots of the columns in this data set. Describe your findings. *Hint: Use function* `pairs()`

c. How many of the suburbs in this data set bound the Charles river? Hint: Subset the data using a logical vector to check if variable `chas==1`, then use `nrow()` to see the number of suburbs.

d. Using `median()`, find out what is the median pupil-teacher ratio among the towns in this data set?

Well done! You've learned how to work with R to read in data and perform some simple analysis and exploration!

Want to improve this tutorial? Report any suggestions/bugs/improvements on here! We're interested in learning from you how we can make this tutorial better.