Introduction to Data Analysis with R

Michelle Norris, Dept of Mathematics and Statistics

April 6, 2018

Introduction

In this lesson, we will import, format, summarize and graph data from the Young People Survey available at the url `www.kaggle.com/miroslavsabo/young-people-survey`. Let's navigate to the website to learn about this data.

Some key information from the data description at the above URL:

- "The data file (`responses.csv`) consists of 1010 rows and 150 columns (139 integer and 11 categorical)."
- "For convenience, the original variable names were shortened in the data file. See the `columns.csv` file if you want to match the data with the original names."

The survey contains the responses from 1010 Slovakian young people aged 15-30 on the following categories of variables: Music preferences (19 items), Movie preferences (12 items), Hobbies & interests (32 items), Phobias (10 items), Health habits (3 items), Personality traits, views on life, & opinions (57 items), Spending habits (7 items), and Demographics (10 items).

Install R and RStudio

If you haven't already done so, install R and RStudio from these websites: `www.r-project.org` and `www.rstudio.com`. Launch RStudio.

Getting Help in R

If you don't know the command you want to use, you can google it and will likely find lots of discussion boards with answers. For help with the syntax of known commands type `?<command name>`, i.e. for help with the `mean` function:

```
> ?mean
```

Warmup: Creating a vector object in R

The following code create a vector consisting of the values 6,8,1 stored in an object named `myvector`.

```
> myvector <- c(6,8,1)
> myvector  # look to see what the object holds

[1] 6 8 1

> sum(myvector)  # add the elements in myvector

[1] 15

> mean(myvector)  # calculate the average the the values in myvector

[1] 5
```

Importing the Data

You can import the data using the `read.csv()` command as shown below.

```
> responses <- read.csv("http://www.csus.edu/indiv/n/norrisa/responses.csv")
> head(responses)  # shows first 6 rows of the dataframe
> str(responses)   # shows type of each variable and first 10 values
> View(responses) # spreadsheet type view of responses
```

The data are stored in an object named `responses`. You can also use the Import Dataset drop-down menu in the Environment pane of RStudio. Click `Import Dataset | From CSV...`. Use your web browser to navigate to my website `http://www.csus.edu/indiv/n/norrisa/` where I have placed an unzipped copy of the data. Copy and Paste the URL for the data in the box at the top of the next dialog box, click the UPDATE button, review the options, then click IMPORT at the bottom right of the dialog box. (Alternately, if you download the original data to your local drive and unzip it, you can navigate to the location on your local drive where the data file is stored using the Browse button.) The corresponding R code will be automatically written to the console. Note this method uses the `read_csv` command instead of `read.csv`. I prefer `read.csv` since it automatically imports character variables as factor/categorical type variables.

Vectors and Dataframes in R

Vectors and dataframes are two important types of objects in R. A vector is a one-dimensional list of elements. All elements in a vector must be of the same type, i.e. integer, numeric, character, logical, etc.

```
> x <- c(2.5, 8, pi) # a numeric vector
> x
```

```
[1] 2.500000 8.000000 3.141593
```

```
> y <- c('cat', 'dog', 'hamster', 'horse')  # a character vector
> y
```

```
[1] "cat"     "dog"     "hamster" "horse"
```

A dataframe consists of columns and rows. For example, we may collect data from 4 students on their height, favorite food and number of siblings.

```
> height <- c(66, 70, 62, 67)
> food <- c('pizza', 'candy', 'broccoli', 'broccoli')
> numSiblings <- c(0,3, 30, 2)
> mydf <- data.frame(height=height, food = food, numSib=numSiblings)
> mydf
```

```
  height     food numSib
1     66    pizza      0
2     70    candy      3
3     62 broccoli     30
4     67 broccoli      2
```

```
> str(mydf)
```

```
'data.frame':        4 obs. of  3 variables:
 $ height: num  66 70 62 67
 $ food  : Factor w/ 3 levels "broccoli","candy",..: 3 2 1 1
 $ numSib: num  0 3 30 2
```

```
> summary(mydf)  # summary of all variables in the dataframe
```

```
     height            food       numSib
 Min.   :62.00   broccoli:2   Min.   : 0.00
 1st Qu.:65.00   candy   :1   1st Qu.: 1.50
 Median :66.50   pizza   :1   Median : 2.50
 Mean   :66.25                Mean   : 8.75
 3rd Qu.:67.75                3rd Qu.: 9.75
 Max.   :70.00                Max.   :30.00
```

Each column is a vector and must have the same length.

Subsetting

We can extract elements from a vector by specifying their location:

```
> x <- c(3,5,4,7)
> x[3]  # extract element in 3rd location1

[1] 4

> x[1:3]  # elements 1,2, and 3

[1] 3 5 4
```

In a dataframe, we can extract by specifying the row and column locations of our subset.

```
> mydf[2, ] # extract 2nd row
> mydf[ , 3] # extract 3rd column
> mydf[ , 2:3]
> mydf[2:4, 3]
> mydf$food #extract column by name
```

Analysis of Young People Survey Data

Let's summarize the survey data.

```
> summary(responses) # Too many variables, let's focus on the fears category
> varNames <- read.csv('http://csus.edu/indiv/n/norrisa/columns.csv')
> View(varNames) # looking in spreadsheet view columns 64-73 deal with fears
```

Let's find out what the respondents feared the most and the fear of Flying differs by gender.

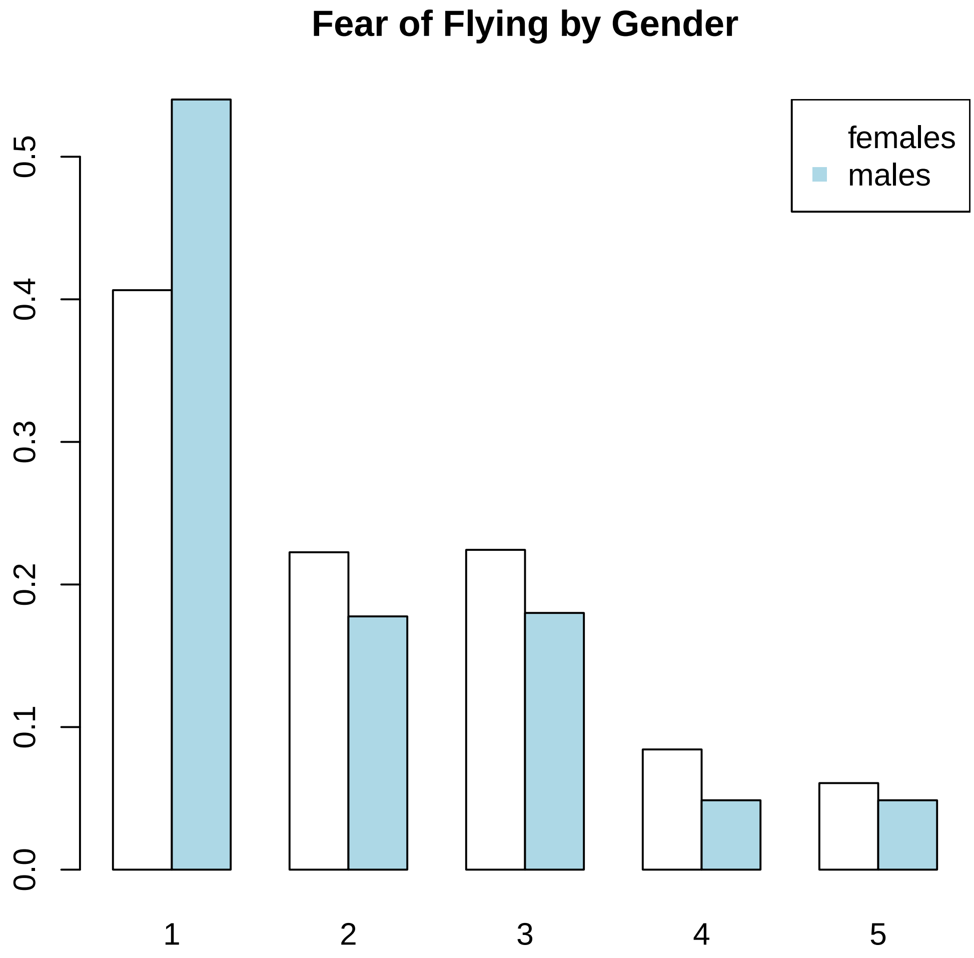
```
> responses <- read.csv("http://www.csus.edu/indiv/n/norrisa/responses.csv") # you need not reload the data,
> fearsData <- responses[, c(64:73, 145)]
> summary(fearsData)  # messy summary
> apply(fearsData, MARGIN = 2, FUN=mean, na.rm=TRUE)
```

Is there a difference by gender in fear of Flying?

```
> responses <- read.csv("http://www.csus.edu/indiv/n/norrisa/responses.csv")
> totByGender <- table(responses$Gender)
> totByGender

       female   male
     6    593    411

> FearFlyByGender <- table(responses$Flying, responses$Gender)  # counts
> PropFemales <- FearFlyByGender[,2]/totByGender[2] # proportions for Females
> PropMales <- FearFlyByGender[,3]/totByGender[3] # proportions for Males
> barplot(rbind(PropFemales, PropMales),
+         beside = TRUE,
+         col = c("white","lightblue"),
+         names.arg = 1:5)
> legend("topright", c("females","males"), pch=15,
+        col=c("white","lightblue"))
> title("Fear of Flying by Gender")
```



Exercise: Change the above code to compare another fear by gender.

Some Graphics

Well do a histogram, scatterplot and boxplot. More tomorrow from 1-2.

```
> hist(responses$Height)
> plot(x=responses$Height, y=responses$Weight)  #scatterplot
> boxplot(responses$Flying ~ responses$Adrenaline.sports, xlab ='Adrenaline Sports', ylab = 'Fear of Flying')
```