Lab 6: A Data Analysis in R

Lab Goals

The goal of this lab is to walk you through the process of performing a data analysis in R. We will review some of the commands we have encountered so far in the context of analyzing an interesting data set. We will also see how to create a new `.Rmd` file from scratch (rather than modifying a pre-existing one). The data has to do with speed dating and can be found in the `SpeedDating.csv`. The source of the `SpeedDating` data is "Gelman, A. and Hill, J., *Data analysis using regression and multilevel/hierarchical models*, Cambridge University Press: New York, 2007."

R Functions

R Function	Description
`barplot()`	Makes a bar graph
`c()`	Combines values into a vector
`cbind()`	Combines columns of data frames, matrices, or vectors
`describe()`	Gives summary information for all variables in a data frame
`dim()`	Finds the dimensions of an object
`head()`	Finds the first part of a vector, matrix, table, data frame or function
`hist()`	Creates a histogram
`legend()`	Creates a legend for a plot
`length()`	Finds the length of a vector
`names()`	Finds the names of the variables in a data frame
`mean()`	Finds the mean of the input values
`plot()`	Makes a scatterplot
`read.csv()`	Reads a file in a table format and creates a data frame from it
`summary()`	Reports the minimum, 25th percentile, mean, median, 75th percentile, and maximimum of a numerical input variable
`table()`	Tabulates counts of categorical data
`tail()`	Finds the last part of a vector, matrix, table, data frame or function
`title()`	Adds a title to a plot

Speed Dating Data

Between 2002 and 2004 a series of speed dating experiments were conducted at Columbia University. Participants were students at Columbia's graduate and professional schools. Each participant attended one speed dating session in which they met with each participant of the opposite sex for four minutes. After each speed date, participants filled out a form rating their date on a scale from 1 to 10 on the following attributes: *Attractive*, *Sincere*, *Intelligent*, *Fun*, *Ambitious*, and *Shared Interests*. They also made a *Decision* indicating whether they would like to see the date partner again. *Like* rated how much they liked the partner overall. *PartnerYes* indicated the respondent's opinion of how likely it was that the date respondent would say yes to them. There is also data on the *Age* and *Race* of each participant. In the data, each row corresponds to one speed date. Responses by the female have an *F* after the variable name. Responses by the male have an *M* after the variable name.

Investigation of the Speed Dating Data

Part 1 - Reading in the Data

How do we load in the dataset?

1. Use the `read.csv()` command in the console, creating a data frame named `SpeedDating` in the workspace for the console. A *comma-separated values*(.csv) file is a file that stores tabular data in plain text form where different fields are separated by a comma or some other character.

In practice, you should never just read in the data file and assume the data has been properly loaded. You should always check and double check that the data has been read in *correctly*! Remember: if you read garbage in, you get garbage out.

2. In the console, type `head(SpeedDating)`. Then open the raw `.csv` file in RStudio using the path *File > Open File.* Do the first few rows of the data frame match those of the actual data? You should also do the same for the last few rows of the data set. What R command can be used for this step? Also, does the last row number of `SpeedDating` correspond to the number of observed speed dates? (The idea is to think of as many "sanity checks" as possible.)

3. One thing to always be careful about is whether there is a header or not in the file. Looking at `?read.csv`, you will see that by default `header=TRUE`, which means that it assumes the first line gives the names of the columns. While this is appropriate for `SpeedDating.csv`, sometimes a data file will not start with a row of column names. We have included another file called `SpeedDatingWithoutHeader.csv` in which we have deleted the column names so that we can see what would have happened if there were no column names. Try `read.csv` on this file and see what happens when `header=TRUE` versus `header=FALSE`. What do you notice?

Part 2 - Creating a R Markdown File

Loading in data properly takes time and it is important to have a clear record of what we have done rather than just typing commands in the console and assuming we will later remember what we have done. This is especially important for published work since you want to make sure that someone else (or yourself in 5 years) will be able to exactly reproduce your analysis, resulting in the identical results.

1. Create a new R Markdown document by clicking on this arrow in the upper left corner of RStudio  . Then choose the second option, *R Markdown...*. A window will pop up in which you can change the title of your document to *Lab 6* and denote yourself as the author. Also, you can choose a default knitting format. Note, all knitting options will still be available regardless of the chosen default.

2. This new `.Rmd` document needs to be renamed and saved in your folder for Lab 6. Choose the path *File > Save As..* to save this document as `LastF-Lab6.Rmd`.

3. This file has some default text already included. You can knit this document to see the effect of the default text.

4. Now, delete all of the default text except for the header with the document title, author, date, and default knitting option.

5. Under the document header write a sentence or two that includes a description of the data set, where you found the data set (from blackboard), and some information on who collected the data.

6. Next, insert a code chunk to read the data into the workspace for the lab document.

Part 3 - Getting to Know Your Data

In the console, let's get some basic information on the `SpeedDating` data frame.

1. Type `dim(SpeedDating)` to determine the number of observations in these data as well as the number of variables.

The function `describe()` in the package `Hmisc` will give detailed information for each variable in the data including:

a) The number of filled in values

b) The number of missing values

c) The number of unique observations

d) The mean of the numerical observations

e) The 5th through the 95th quantiles for numerical data

f) A table including frequency (counts) and percent of total for each level of a factor or over the range of numerical variables

2. The function `describe` is in the R package `Hmisc`. To load this package into the console's environment, type `library(Hmisc)`. If you get an error saying `there is no package called 'Hmisc'`, then you must first install the package. To do so, choose *Tools > Install Packages...* from the menu for RStudio. A window will pop up in which you will type `Hmisc` in the field for "Packages."

3. After installing the `Hmisc` package, load it into your workspace by typing `library(Hmisc)` into the console. Once the `Hmisc` package is loaded into your workspace, you can type `?describe` into the console to get the documentation for the function `describe`.

4. Now type `describe(SpeedDating)` into the console. Are there any missing values? Give a couple of examples of numerical variables in this data frame. Which variables are categorical?

5. Now, let's return to our R Markdown document for this lab. Add a sentence in this document explaining that we will start by looking at the data before proceeding. Follow this by adding a code chunk that does the following:

   a) Finds the dimension of `SpeedDating`

   b) Uses `library(Hmisc)` to load `Hmisc` into the R Markdown document workspace (recall that when you knit your R Markdown file, a fresh R environment is spawned separate from the console)

   c) Describes the variables in `SpeedDating`

6. After the code chunk, write an observation about the data such as a note indicating that `Decision` has been read in as a quantitative variable.

Part 4 - More Detailed Investigation of the Variables in `SpeedDating`

`DecisionM` and `DecisionF`

The variable `DecisionM` indicates whether the male participant wishes another date. `DecisionF` indicates whether the female participant wishes to have another date.

Decision	Description
0	Do not want a second date
1	Do want a second date

1. Add a code chunk to your R Markdown document that uses the `table()` function to create a table of counts of `DecisionM` by `DecisionF`. Put your cursor on this line of code in your R Markdown document and press the "Run" button, which will run it in the console (this is faster than the alternative of copying and pasting it into the console and hitting enter).

2. Below this code chunk include a brief description of the significance the four combinations of the levels of `DecisionM` and `DecisionF`.

3. It might be interesting to look at which gender was more likely to want a second date when their "date" was not interested in another date. Let us restrict our attention to only those dates in which `DecisionM + DecisionF ==1` since that means that one person was interested and the other was not. On this smaller data frame, which we'll call `onlyOne`, we can look at the relative frequency with which it was the male who was interest (`DecisionM` being 1) as opposed to the female being interested (`DecisionM` being 0). The following code gives us these proportions and saves it to a variable named `SympMale`:

```
onlyOne <- subset(SpeedDating, DecisionF + DecisionM == 1)
SympMale <- table(onlyOne$DecisionM) / nrow(onlyOne)
```

4. Now create a barchart of `SympMale` using the console. In the case that only one person was interested, was it more commonly the male or female who was interested?

5. Now that we have found something interesting, it would be worthwhile to include this barchart in our R Markdown document. Create a code chunk that:

a) Creates the table `SympMale`
b) Creates a barchart for `SympMale` where you set `names.arg = c('F','M')` and `main = 'Who Desired the Second Date When One of Them Did Not?'`

6. Write a brief analysis of this plot under the code chunk in your R Markdown document.

`PartnerYesM` and `PartnerYesF`

The `PartnerYesM` and `PartnerYesF` indicate on a scale of 1 to 10 on whether their partner would say "Yes" to a second date. `PartnerYesM` are the responses from the males, and `PartnerYesF` are the responses from the females. Here we will investigate whether females or males in the study are better able to tell how likely their "date" would see them again.

How well did females and males predict that their partner would say "No" to a second date? Include a code chunk in your R Markdown document that creates a boxplot for `PartnerYesF` using only data where the male did not desire another date and a boxplot for `PartnerYesM` using only data where the female did not desire another date. Use the following steps to create these boxplots:

a) We would like the boxplots to be side by side for comparison. Hmmm... how can we do this? Sometimes the best way to determine the correct code for something we would like to do in R is to "Google" it. Google 'create side by side boxplots in R' to determine what code to use to set our boxplots side by side.

b) Using (a), include a code chunk that creates side by side boxplots of `PartnerYesF` where only the data for `DecisionM==0` is included and `PartnerYesM` where only the data for `DecisionF==0` is included. Title this plot by including the option `main="Guess about partner's opinion when answer was no"`.

c) For the sake of comparison, it would be best to have each box plot displayed on the same scale. Include the option `ylim=c(0,10)` for the boxplots created in (b).

d) We also want each boxplot labeled by gender. In the console type `?boxplot` to figure out how to add names to your plots. The first boxplot should be labeled `Female Guess`. The second plot should be labeled `Male Guess`. Edit your code chunk for (b) to create these labels.

It also may be interesting to see how well males and females could predict that their partner would say "Yes" to a second date. Repeat steps (a)-(d) using only data where `DecisionM==1` for `PartnerYesF` and `DecisionF==1` for `PartnerYesM`. Title these plots "Guess about partner's opinion when answer was yes."

After the code chunk for these two sets of boxplots, include a short summary of what you see in the boxplots. First, describe the variable being plotted. Then, include answers to the following questions.

i) Was there a difference between the ability of males versus females to detect when their partner would say "Yes" to another date?

ii) Was there a difference between the ability of males versus females to detect when their partner would say "No" to another date?

Relationship Between Attractiveness and Intelligence

Was the level of intelligence or the attractiveness of a "date" more important in determining whether the participant wanted to see their date again? It may be interesting to look at this question for females and males separately.

a) We'll first look at the female responses. Using the console, create a scatterplot of `SpeedDating$IntelligentF` by `SpeedDating$AttractiveF`.

b) Notice that many of the responses are identical and fall on a grid. Therefor we cannot see how many there are. The `jitter()` function will spread these points out slightly. Now in the console plot `jitter(SpeedDating$IntelligentF)` by `jitter(SpeedDating$AttractiveF)`.

c) Add the following options to this plot:

   i) `main = 'Female Assessment of Male'`
   ii) `ylim=c(1,10)` and `xlim=c(1,10)`
   iii) `xlab="Attractiveness"` and `ylab="Intelligence"`

d) Let's add color to get at our original question. Notice that `DecisionM` is 0 or 1. If we add one to this we get 1 and 2 which as colors are interpreted as "black" and "red" in R. Use the `col` option in the code for the scatterplot to color observations red when the male desired a second date.

e) Add a legend to include a description of the significance of the different colored observations. Use the following code to create this legend, `legend("bottomright",legend=c("Doesn't want 2nd date", "Does want 2nd date"), col=1:2, pch=1)`.

f) Create similar plots for the males. Include a code chunk in your R Markdown document that creates both scatterplots as directed.

g) Do you see any differences in the importance of intelligence versus attractiveness for females versus males in determining whether they would choose to see their "date" again? Comment on this in your R Markdown document.