R Workshop - Exercise 1

Debby Kermer & Maya Pham

2016-03-10

Table of Contents

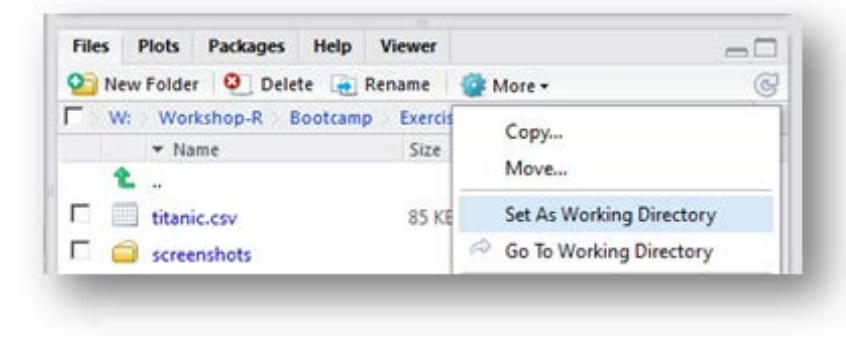
Before starting this exercise, please:

1. Install R & RStudio. see: http://dataservices.gmu.edu/files/Installing-R-RStudio.pdf
2. Download the zip file for this workshop from the Bootcamp website and unzip it.

Step 1: Set your Working Directory

On the Files tab in RStudio, find the folder with the fles for this workshop. To pick a different drive, click the elipses (...) box.

You should see the **titanic.csv** file when you have found the proper folder. Then, click "More" and choose **Set As Working Directory**.



If you know the path to the files, you can use the **setwd** function, like this:

```
setwd("C:/Users/Myname/Documents")
```

Confirm that you have set the Working Directory. In RStudio, check the top of the Console:

Or, you can type the following in the Console and press Enter:

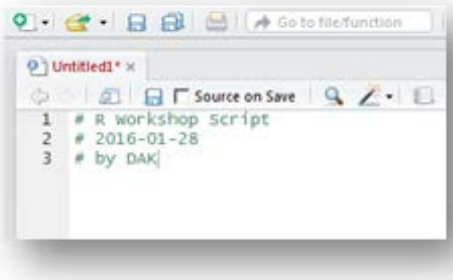
```
getwd()

## [1] "W:/Workshop-R/Bootcamp/Exercises"
```

Step 2: Start a Script File

If you do not already have a blank script file open, do one of the following:

- Click File, New File | R Script
- Click the new script button, then R Script
- Press Ctrl+Shift+N (on Mac: Cmd+Shift+N)

Start documenting your work by including the purpose of the document, the date, and your name. To write a comment, begin the line with a hash mark (#).



A note on "running" R syntax

In the following instructions, to "run" something means that you should type it in your script file, then do one of the following;

- Select the text and click the Run button.
- Click somewhere in the line and press *Ctrl+Enter* (for Mac: *Cmd+Enter*)
- [unless otherwise specified] type it in the Console next to the ">" and press Enter.

Step 3: Install and Load Packages

Later, you will use the package "ggplot2", so you first must install and load the package.

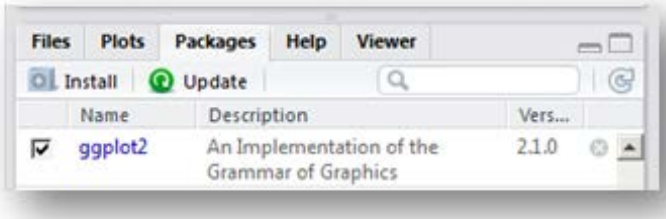
You can use the Packages tab in RStudio to do this. First, click Install and type ggplot in the box labeled "Packages". Then, click the checkbox next to ggplot2 when it appears in the list.

Note: It may take a while to install the package since it installs several other packages as well. Ignore any warning message related to mismatched version.

Or, you can type the following syntax in your script file and run it:

```
install.packages("ggplot2")
library(ggplot2)
```

Confirm that *ggplot2* is installed and loaded by checking the Packages tab in RStudio: ggplot2 should now be listed and checked.



Step 4: Import Data

For this exercise, we will use the data **titanic.csv**, which is a listing of the passengers of the ill-fated Titanic and some information about them. Since the file that you want to import is in your working directory, you should only have to give R the name of the file.

Use the **read.csv** function to import the data into the object **mydata** by typing the following in your Script file and running it:

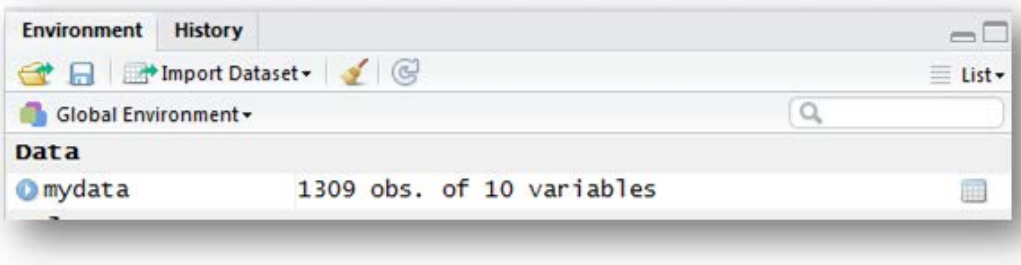
```
mydata <- read.csv("titanic.csv")
```

Import from the Web If you are having trouble loading the downloaded file, try importing the file directly from the web by running the following syntax. The first line creates an object that simply stores the location (be sure to copy this exactly), the second line actually loads the file.

```
datasource <- 'http://dataservices.gmu.edu/files/titanic.csv'
mydata <- read.csv(datasource)
```

Step 5: View Data

Check the RStudio Environment tab to confirm that "mydata" is listed. You can switch between List and Grid view using the dropdown on the right.



See your data table by clicking anywhere in the row. Or, run:

```
View(mydata)
```

It is also common to view only a few of the first or last rows in the dataset using **head** and **tail** functions, such as:

```
head(mydata)
```

```
##   id                              name survived age gender sibsp parch
## 1  1              Abbing, Mr. Anthony     Died  42   male     0     0
## 2  2    Abbott, Master. Eugene Joseph     Died  13   male     0     2
## 3  3      Abbott, Mr. Rossmore Edward     Died  16   male     1     1
## 4  4 Abbott, Mrs. Stanton (Rosa Hunt) Survived  35 female     1     1
## 5  5      Abelseth, Miss. Karen Marie Survived  16 female     0     0
## 6  6    Abelseth, Mr. Olaus Jorgensen Survived  25   male     0     0
##   pclass  fare embarked
## 1    3rd  7.55        S
## 2    3rd 20.25        S
## 3    3rd 20.25        S
## 4    3rd 20.25        S
## 5    3rd  7.65        S
## 6    3rd  7.65        S
```

Note: R automatically wraps the output to fit your screen. If it looks bad, make the console wider by dragging the vertical divider to the right. Then, re-run the syntax.

The Environment tab lists all the variables and their details (e.g., data types). But, you can also use the **str** function.

```
str(mydata)
```

```
## 'data.frame':    1309 obs. of  10 variables:
##  $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ name    : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ survived: Factor w/ 2 levels "Died","Survived": 1 1 1 2 2 2 1 2 2 2 ...
```

```
##  $ age     : num  42 13 16 35 16 25 30 28 20 18 ...
##  $ gender  : Factor w/ 2 levels "female","male": 2 2 2 1 1 2 2 1 2 1 ...
##  $ sibsp   : int  0 0 1 1 0 0 1 1 0 0 ...
##  $ parch   : int  0 2 1 1 0 0 0 0 0 0 ...
##  $ pclass  : Factor w/ 3 levels "1st","2nd","3rd": 3 3 3 3 3 3 2 2 3 3 ...
##  $ fare    : num  7.55 20.25 20.25 20.25 7.65 ...
##  $ embarked: Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 2 2 4 2 ...
```

In this output, you want to confirm that **mydata** is a **'data.frame'** and that **gender** and **pclass** are listed as **Factors**.

Step 6: Descriptive Statistics

You can look at your data Using the **summary** function. You will be working with these three variables:

- **gender**: Male or Female
- **fare**: Cost of the ticket
- **pclass**: Passenger Class (1st Class as the fanciest)

For this function, as with many built-in functions, remember to use the $ to specify that the variable **gender** is inside th data frame **mydata**.

```
summary(mydata$gender)
```

```
## female   male
##    466    843
```

To see a crosstabulation of gender and pclass, use the **table** function. This displays the number of cases (people) that have each combination of characteristics. You can see that the gender balance is much more equal for 1st and 2nd class than for 3rd class.

```
table(mydata$pclass, mydata$gender)
```

```
##
##       female male
##   1st    144  179
##   2nd    106  171
##   3rd    216  493
```

Step 7: Create Graphs

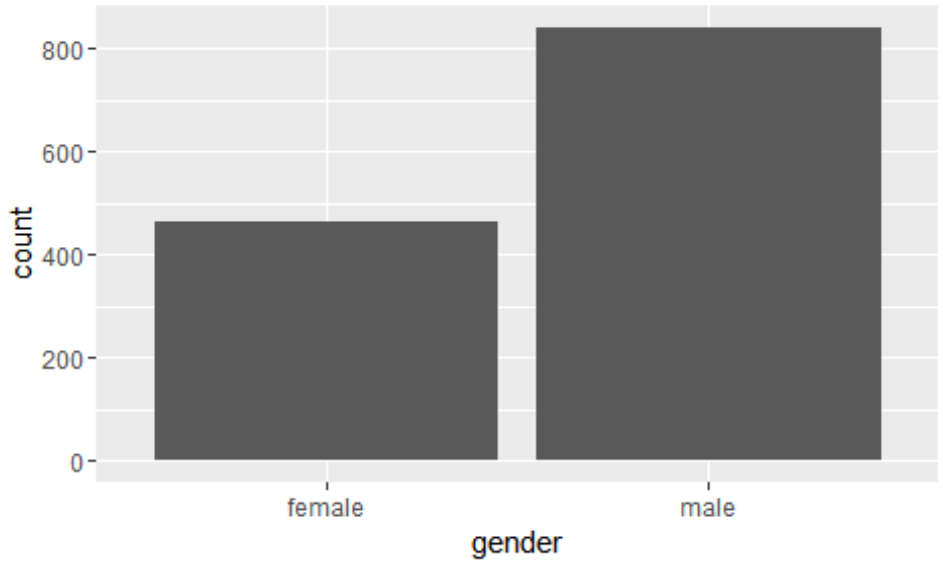
You can use the built-in **plot** function to get bar charts for the variables **gender**, **fare**, and **pclass**. Just replace the word summary with plot:

```
plot(mydata$gender)
```

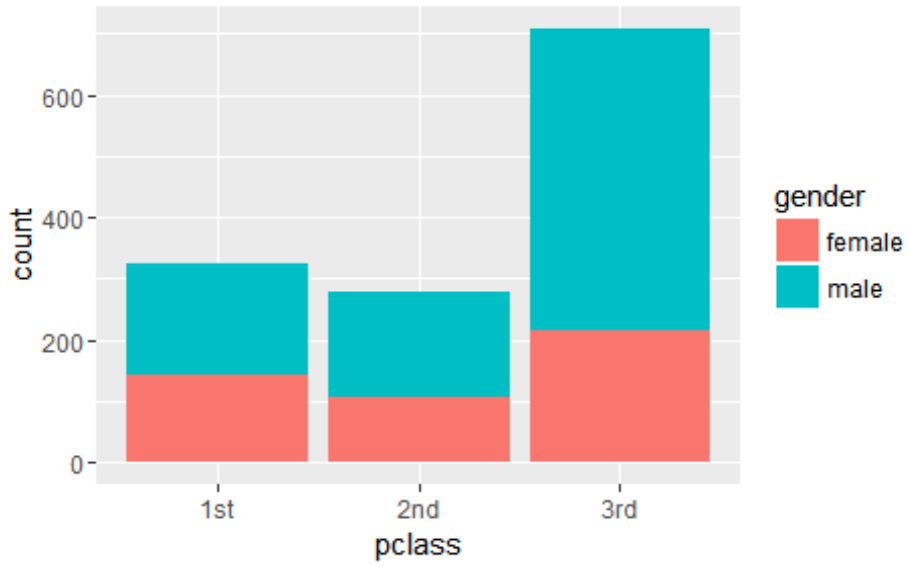
The **plot** function has many limitations. In particular, it does not allow the "data" argument. This is not a problem with simple graphs, but is useful when dealing with more than one variable.

The **qplot** function, from the package **ggplot2**, is similar to **plot**, but allows the data argument. Make sure you have completed Step 3 to install and load **ggplot2** before running this syntax.

```
qplot(gender, data=mydata)
```




```
qplot(pclass, fill=gender, data=mydata)
```



It is great that you do not even have to specify the type of graph you want: ggplot2 chose the correct graph based on your variable types. Note that **qlot** is a simpler function of this package. The main function **ggplot** allows more customization to create advanced graphs, but it requires a sharp learning curve.

Step 8: Run a Statistical Analysis (optional)

Use the **aov** function to a run two-way ANOVA (Analysis of Variance) to test whether class (**pclass**) and **gender** predicts the **fare**.

You must first create an analysis object and then use functions on that object. We name our analysis object tt.anova, but you can use another object name if you prefer. Replace the plus sign with an asterisk (*) to also include the interaction.

```
tt.anova <- aov(fare ~ pclass + gender, data=mydata )
summary(tt.anova)
```

```
##               Df  Sum Sq Mean Sq F value   Pr(>F)
## pclass         2 1272986  636493  380.85  < 2e-16 ***
## gender         1   49114   49114   29.39 7.05e-08 ***
## Residuals   1304 2179300    1671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

Interpretation: The average fare for each passenger's class is different significantly which reflects the fact that first-class passengers paid more than the second class and third class.

To further examine the results of the ANOVA, you can use other functions on the analysis object, such as **coefficients** or **residuals**.

```
coefficients(tt.anova)
```

```
## (Intercept)   pclass2nd   pclass3rd  gendermale
##    94.65548   -65.51545   -72.39123   -12.89563
```

End of Exercise 1