

An Introduction to R

Stat 217

Getting Started

The R-studio interface is divided into four regions:

1. The top left is where you go to open a *script*. Click on **File, New File, R Script**. When you are working on a project you should begin by opening a script for that project. Scripts can be saved, so you can put your project down and come back to it later. To run a line of code in a script, type Ctrl+Enter (PC) or Command+Enter (Mac).
2. In the bottom left is the *Console*. You can run code directly in the console, or you can run it from your script and the output will show up in the console.
3. In the upper right is the *Environment*. It is a list of all the different R objects you currently have defined, along with brief descriptions of those things.
4. The bottom right space has a few functions. It is the space where the plots you make will appear (see the *Plots* tab. It is also where you can go to find R help pages. See the *Help* tab. Don't worry too much about the other tabs.

Testing Out R

To get a little bit of a feel for what it's like to work with R, let's start with something super simple. Open a script and use it to add the numbers 14 and 37. Then do the same addition directly in the console.

```
14 + 37 ## Adding the numbers 14 and 37 together.
```

Also, notice how I used the `##` symbol to make a comment about what I was doing in my R script. Commenting your code is a good programming practice.

Creating Variables

In R we use the symbol `<-` (there's no space in there) for assignment. For example, let's assign the value 3 to the letter x and then add x and 4.

```
x <- 3
x + 4

## [1] 7
```

That line at the bottom is displaying the R output that you should see in your console when you run those two lines of code.

An Error in R

R is really smart, but sometimes it doesn't quite get what we're trying to tell it. When this happens, it will produce an error. Type `X + 4` into the console and see what you get.

```
X + 4
```

```
## Error in eval(expr, envir, enclos): object 'X' not found
```

See what happened? R is case sensitive, and we defined `x` to be equal to 3, but we didn't define `X` as anything. So R doesn't know what to do.

Data

Strictly speaking, R is not only for performing statistical analysis. However, R really does excel in this realm, and in statistics we need data. So let's go get some. Start by installing the package `heplots`. Go to **Tools, Install Packages**, then type in `heplots` and hit the `install` button. In your script type the following:

```
require(heplots) ## tells R to load this package
data(MockJury)  ## tells R to get the MockJury data set and put it in your Environment
```

Type `MockJury` into the help tab to read about the data set. We need to do a few things to the data before we can start working with it. You can just copy and paste the following code into your script, then run it.

```
MockJury <- subset(MockJury, Attr != "Beautiful", select = c(Attr, Crime,
  Years, Serious, independent, sincere))
rownames(MockJury) <- 1:nrow(MockJury)
```

The data set called `MockJury` is now ready to go. You can view it by running

```
MockJury
```

Whoa! That was a lot of output. Here are two handy commands to just look at the first six lines and the last six lines of a data set.

```
head(MockJury) ## prints the first six lines of the dataset MockJury
```

```
##      Attr   Crime Years Serious independent sincere
## 1 Unattractive Burglary    1      3          6      5
## 2 Unattractive Burglary    4      8          2      9
## 3 Unattractive Burglary    3      5          9      7
## 4 Unattractive Burglary    2      2          7      1
## 5 Unattractive Burglary    8      5          5      1
## 6 Unattractive Burglary    8      9          2      8
```

```
tail(MockJury) ## prints the last six lines of the dataset MockJury
```

```
##      Attr   Crime Years Serious independent sincere
## 70 Average Swindle    3      2          9      9
## 71 Average Swindle    2      1          8      8
## 72 Average Swindle    7      4          9      1
## 73 Average Swindle    6      3          5      2
## 74 Average Swindle   12      9          9      1
## 75 Average Swindle    8      8          1      5
```

Another nice way to begin our exploration of the data is to use the `summary` function.

```
summary(MockJury)
```

```
##           Attr      Crime      Years      Serious      independent
## Beautiful   : 0   Burglary:38   Min.    : 1.00   Min.    :1.00   Min.    :1.00
## Average     :38   Swindle :37   1st Qu.: 2.00   1st Qu.:3.00   1st Qu.:5.00
## Unattractive:37           Median : 3.00   Median :5.00   Median :6.00
##           Mean   : 4.88   Mean   :4.87   Mean   :5.99
##           3rd Qu.: 7.50   3rd Qu.:6.00   3rd Qu.:8.00
##           Max.   :15.00   Max.   :9.00   Max.   :9.00
## sincere
## Min.    :1.00
## 1st Qu.:2.00
## Median :4.00
## Mean   :4.51
## 3rd Qu.:7.00
## Max.   :9.00
```

Talk with your group members about what they output is telling you. If you're not sure, ask!

Selecting Variables

You can extract columns of a data set using the \$ symbol. For example, we can ask R to print only the Years variable.

```
MockJury$Years
```

Maybe we want to know the mean and standard deviation of sentence lengths given to defendants.

```
mean(MockJury$Years)
```

```
## [1] 4.88
```

```
sd(MockJury$Years)
```

```
## [1] 3.76
```

Plots

R can display data in some truly amazing ways. But we're going to start with some simple box plots and histograms.

```
par(mfrow = c(1, 3)) ## This tells R to put the following 3 plots in 1 row
hist(MockJury$Years)
boxplot(MockJury$Years)
plot(density(MockJury$Years))
```

