R Notebook, Data Frames, and swirl

In this lesson you will learn how to combine text, R code, and results into a single file. Data sets are usually stored in so-called data frames. Here you will learn how to use a data frame in your analysis. Finally, I'll point you to some resources to learn more about data types in R.

Reproducible Research

Suppose that you want to reproduce a result published by some other researchers. This may be difficult because the result involves certain data sets and software code that may not be provided. Even if they are, you may not have access to the particular software that the authors were using. Wouldn't it be great if the authors would provide the data and the statistical analysis in a way that would make it easy for you to check and reproduce their results? This idea of reproducibility is supported by the document formats of *Quarto*, *R Notebook*, *R Markdown*, and *R Sweave*. Which of these four document types you should use depends on the type of report that you want to create. For this class I suggest that you use R Notebook since it is the easiest to learn and will be sufficient for our purposes.

When you create an R Notebook document, RStudio creates an R Notebook template for you. To get a sense for how this works, go ahead and click on the *Preview* button. If you haven't saved the R Notebook document yet, you will be prompted to do so. Then the software will convert the R Notebook document into an html document and display it. You could email this html file and the recipient could view it with a browser. You could also email your R Notebook document and it could be used to recreate your html file. Instead of html you can also create other formats, e.g. PDF or Word, but this does require additional software. For simplicity we will just use html in this class.

Data Frames

Typically you will read some data into R and then process it. As an example consider the data file College.csv. It contains statistics for a large number of US Colleges from the 1995 issue of US News and World Report. Download this

data file, read it into R, and store it in the variable *us.colleges*. Remember to set the working directory to the folder containing the data.

```
us.colleges <- read.csv("College.csv")
```

Another way in RStudio, perhaps easier, is to use the “Import Dataset” button. You can use the R function *str()* to find out about the structure of an R object.

```
str(us.colleges)
```

We learn that the variable *us.colleges* contains a so-called *data frame* containing 777 observations of 19 variables. A data frame is like a spreadsheet with rows and columns. You can pick out certain rows and columns as follows:

```
us.colleges[3, 4]          # 3rd row and 4th column
us.colleges[3, 'Accept']   # 3rd row and column labelled Accept
us.colleges[3, c(1, 3)]    # 3rd row and columns 1 and 3
us.colleges[3, ]           # 3rd row and all columns
us.colleges[3, -4]         # 3rd row and all columns except the 4th column
us.colleges[3, -c(1, 4)]   # 3rd row and all columns except the 1st and the 4th column
```

Suppose you want to modify your data set. Let's store the first four and the last column of the data set in the variable *us.colleges2*. What are the variable names of *us.colleges2*?

```
us.colleges2 <- us.colleges[,c(1:4,19)]
names(us.colleges2)
```

The variable names are not very descriptive. Let's change the variable names of us.colleges2 to “college”, “private”, “applied”, “accepted”, “graduation.rate”. Here, the variable “applied” records the number of applications received and the variable “accepted”, the number of applicants accepted.

```
names(us.colleges2) <- c("college", "private", "applied", "accepted", "graduation.rate")
```

Now let's add another column to us.colleges2 called “acception.rate‘ which records the proportion of accepted students.’ ”

```
us.colleges2$acception.rate <- us.colleges2$accepted/us.colleges2$applied
```

Let's save the new data set. Create a folder on your computer called "Example". Change your working directory to this folder and write the data of us.colleges2 to the file Colleges2.csv using the function *write.csv()*.

```
write.csv(us.colleges2, file="Colleges2.csv")
```

Did you perhaps forget to set your working directory and now you are wondering where your file is? You can find the location of your working directory as follows.

```
getwd()
```

You can also write your file into a specific location or folder by specifying not just the file name but its url. Here is an example.

```
write.csv(us.colleges2, file="~/Desktop/Colleges2.csv")
```

Is there a difference between the acceptance rates for private and public colleges? Find the average acceptance rate for private and public colleges separately.

Here is a solution using logical vectors to select the desired colleges.

```
ind.private <- us.colleges2$private == "Yes"
mean(us.colleges2$acception.rate[ind.private])
ind.public <- us.colleges2$private == "No"
mean(us.colleges2$acception.rate[ind.public])
```

A more elegant solution would be to use the R function by().

```
by(us.colleges2$acception.rate, us.colleges2$private, mean)
```

It gets tedious to have to type us.colleges2$college. To avoid this you could use the with() function.

```
with(us.colleges2, by(acception.rate, private, mean))
```

Another approach is to attach and then detach the data frame.

```
attach(us.colleges2)
```

```
by(acception.rate, private, mean)
detach(us.colleges2)
```

How many colleges accept at least 90% of all applicants?

```
sum(us.colleges2$acception.rate>=0.9)
```

Find the ten colleges with the highest acceptance rates

```
attach(us.colleges2)
ord <- order(acception.rate, decreasing=TRUE)
college[ord][1:10]
detach(us.colleges2)
```

Find all public colleges that have an acception rate of at least 80% and a graduation rate of at least 80%.

```
attach(us.colleges2)
ind <- acception.rate >= 0.8 & graduation.rate >= 80
college[ind]
detach(us.colleges2)
```

swirl

As you know, there are thousands of R libraries available. In fact, there is one that teaches you R programming interactively at your own pace. Install the R package swirl as follows. Open R or RStudio and type

```
install.packages("swirl")
```

into the R console and press the Enter key. Then start this learning environment as follows. Type the following, pressing Enter after each line.

```
library("swirl")
swirl()
```

Choose the lesson "R Programming" and complete the following seven sections:

1. Basic Building Blocks
2. Workspace and Files

3. Sequences of Numbers
4. Vectors
5. Missing Values
6. Subsetting Vectors
7. Matrices and Data Frames

At the end of each section you might be asked if you want to send an email to your instructor. Please don't.

For additional help I recommend the lectures given by Mike Marin (see here: https://www.youtube.com/user/marinstatlectures). Most of them are short (less than 10 minutes) and to the point.