

Reading, Writing and Exploring Data

Exercise files

Please download all exercise files for this exercise at <http://georgejmount.com/hillsdaler>.

Loading our libraries

It's a good idea to call the libraries you're going to use in your script at the beginning of the program.

We will be using two libraries, `tidyverse` and `readxl`. Load them now:

```
library(tidyverse)
library(readxl)
```

If you see an error message, check your packages are installed:
`install.packages()`.

DRILL

1. Install and load the caret package to your machine.

Reading in Files from R

While R has its own file extension for storing data, it's more common to reading and writing to outside file types. We will cover how to read and write to .csv, .txt and .xlsx files.

we will be importing in the famous iris dataset which is popular in data science training.

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres.

Working Directories in R

An active R session always has a working directory associated with it. This is where, by default, all files are read and write to.

Check the location of your current directory with the `getwd` function:

```
getwd()
```

```
getwd
```

Let's say that instead I had set my working directory to another folder, "C:/Iris".

In this case a relative reference will not work.

```
setwd("C:/Iris")  
iris<-read_csv("iris.csv")
```

Instead, we need to use the *absolute* file path because the file is not in the working directory.

For the remainder of this unit, I will save all files to C:/RFiles which will also be my working directory:

```
setwd("C:/RFiles")
```

Reading and Writing to CSV Files

To *read* files we will use `read_csv()`

Make sure to assign this function to an object! If not, you will simply have read the file in R without any way to call it!

To *write* files we will use (you guessed it!) `write_csv()`.

This function will take one extra argument in that we first type the object where the dataset is pointed to and then write to the file we want.

However, we will not need to assign it to an object, since we are exporting it anyway.

```
#write the iris dataframe to iriswrite.csv  
write_csv(iris,"iriswrite.csv")
```

This file will now appear in our *working directory* (C:/RFiles).

Reading and writing to .txt files

This will be very similar to the read_csv and write_csv functions:
We use read_tsv and write_tsv.

```
#assign the iris.txt dataset to iris variable  
iris<-read_tsv("iris.txt")
```

```
#write the iris dataset to Iriswrite.txt  
iris<-write_tsv("iriswrite.txt")
```

Reading and Writing Excel Files

We will now read in the Iris.xlsx file using the read_excel function:

Questions?