

# A Very Gentle Introduction to Statistics

Ethan Weed

17 April, 2020

## Contents

1 Getting Data Into R.	1
------------------------	---

## 1 Getting Data Into R.

One of the most useful variable types for data analysis in R is the dataframe. A dataframe is a little bit like a spreadsheet, in that it has rows and columns, but we interact with it a little differently than we do with a spreadsheet like Excel. In Excel, you can just go in and click on a cell and change whatever you want. You can add plots on top of your data, and mess around however you please. This can be great for taking a quick look at data, but it is no good for actual analysis, because there is too much room for human error. It is too easy to accidentally change something and mess up all your data. Dataframes in R are a little more structured than a spreadsheet, and because we interact with them *programmatically*, by writing code, even if we mess something up, we can always trace back our steps, and figure out what went wrong.

The easiest way to get your data into R is by importing it from a .csv file. Here we look at data from students carrying out a version of the Stroop color-naming task. The command `sep = ","` tells R that the data uses commas to separate the columns. You could use other characters to separate the columns. As an example, Excel sometimes uses semicolons instead of commas. In this case, we can just write `sep = ";"`. If you don't use the `sep =` command, R will just assume it is a comma, so if your data is comma-separated you can just leave this command off.

```
df <- read.csv("/Users/ethan/Documents/GitHub/ethanweed.github.io/r-tutorials/data/Stroop-row-over-the-")
```

There are many rows of data, and maybe we only want to work with the data from 2015. We can easily make a new dataframe, with only the 2015 data:

```
df_2015 <- subset(df, Year == 2015)
df_2015
```

##	Reading_NoInt	Naming_Int	Naming_NoInt	Reading_Int	Year
## 91	4.16	6.76	4.45	4.65	2015
## 92	4.35	7.73	4.78	4.46	2015
## 93	3.60	7.00	4.00	3.50	2015
## 94	3.90	9.03	4.60	6.30	2015
## 95	4.22	9.98	6.83	6.24	2015
## 96	4.31	6.51	5.78	4.08	2015
## 97	4.58	11.57	6.22	4.49	2015
## 98	3.93	7.33	5.50	4.26	2015
## 99	4.48	9.78	5.04	5.33	2015
## 100	3.56	7.86	4.16	3.81	2015
## 101	3.10	10.49	5.34	3.62	2015

```
## 102      4.11      9.17      5.22      3.97 2015
## 103      3.17      7.51      3.69      3.26 2015
## 104      3.85      9.92      6.12      4.28 2015
```

You can use the \$ symbol to select a single column from the dataframe, using the column header:

```
Reading_no_interference <- df_2015$Reading_NoInt
Reading_no_interference
```

```
## [1] 4.16 4.35 3.60 3.90 4.22 4.31 4.58 3.93 4.48 3.56 3.10 4.11 3.17 3.85
```