Getting Started With R

1 Importing and Viewing Data

R is capable of importing many different types of data files into your workspace for analysis. To keep things as simple as possible in this class, we will work with data files that are saved in CSV format (Note than any file that is not in CSV format can be converted into CSV format using excel or other tools). On your computer, these will appear as something like dataexample1.csv. You can import these files manually by clicking File,Import Dataset,From Text, readr Browse and then clicking on the data file you want. A better way is to type something like the following:

```
setwd('directory in which the data file is stored')
mydata <- read.csv('dataexample1.csv')
```

A directory is just a folder on your computer. For this class I recommend you make a folder called ec102. In this case your directory will be something like "/Users/Name/Documents/ec102. In your Global Environment you will now see something called mydata. You can click on it and view the dataset you have just uploaded. Throughout this tutorial, we will be using a dataset with macroeconomic data for the US from 1871-2015. This can be downloaded at https://www.calvinackley.com/teaching/. Once this file is put into the folder called Teaching/EC102_Sum18 in my dropbox, I can upload it by typing

```
setwd("~/Dropbox/Teaching/EC102_Sum18")
mydata <- read.csv('us_macrodata_1871_2015.csv')
```

There is now an item called mydata in my environment box. If you click on this you can view the dataset. To look at specific rows and columns of your data you can use the format mydata[row,column]. So, to examine the first observation in column 2 of your data you type

```
mydata[1,2]

## [1] 11.84
```

To establish a new object in your environment from part of your main dataset you simply pick a name for this new object and use the ¡- as an equals sign.

```
c <- mydata[12,3]
```

Creates a new object called c which is equal to the value in the 12th row and 3rd column of my dataset. You can view multiple rows or columns by using a colon for consecutive rows/columns i.e 1:10 and by typing a list in the format c(1,4,9). To view the first 10 observations in the 2nd, 4th, and 5th columns, for example, you can type

```
mydata[c(1:10), c(2,4,5)]
```

```
##      cpi population realgdp_percap
## 1  11.84      41010        3098.76
## 2  11.84      42066        3273.76
## 3  11.60      43225        3457.76
## 4  11.04      44429        3425.25
## 5  10.64      45492        3339.25
## 6  10.39      46459        3405.10
## 7  10.15      47400        3503.57
## 8   9.67      48319        3547.37
## 9   9.67      49264        3884.85
## 10  9.91      50262        4123.29
```

You can also reference columns by their names rather than their numbers. You can see what your columns are titled by clicking on the data set or by typing colnames(mydata). In the dataset we are working with there is a column called year and another called cpi. To see the first 10 observations for these columns you can type

```
mydata[c(1:10), c('year','cpi')]
```

```
##    year   cpi
## 1  1871 11.84
## 2  1872 11.84
## 3  1873 11.60
## 4  1874 11.04
## 5  1875 10.64
## 6  1876 10.39
## 7  1877 10.15
## 8  1878  9.67
## 9  1879  9.67
## 10 1880  9.91
```

Excercise 1) Find the year and the president in the 20th and 57th row

1.1 SUBSETTING

We often want to examine only part of our dataset that meets certain conditions i.e only the females or only the years 1900-1950 for this we use the subset command in R.

To examine the cpi and unemployment from 1900-1910 you can type

```
subset(mydata, year %in% c(1900:1910), select = c('year','cpi','unemployment'))
```

The second part of the subset command (after the first comma) is the condition we want to use and the third part select= c() specifies which variables we want to keep (if the third part is left empty then all variables are kept). The condition takes the general form variable ¡, ¿, ==, %in%. If we only want to keep the rows where war = 1 (indicating wartime) and create a new dataset called wardat, we type

```
wardat <- subset(mydata, war==1,
                 select =  c('year','cpi','unemployment','president','war_name'))
```

Excercise 2) Create a subset of data for the years 1950-1975 that inlcludes the president and unemployment rate

You can specify multiple conditions to be met in the subset using the & sign between conditions. The command below creates a new dataset with only the years between 1900-1950 in which there's a war

```
wardat <- subset(mydata,year %in% c(1900:1950) & war==1,
                 select = c('year','cpi','unemployment','president','war_name'))
```

Excercise 3) create a subset of data called newdat for the years 1898-1945 that includes president, cpi, realgdppercap, war, and warname and only includes years that there is a war //

1.2 Averages

You can specify entire columns of data by typing mydata$column-name. The command below Will return every cpi value in the dataset

```
mydata$cpi
```

In order to compute the average we use the mean() command

```
mean(mydata$cpi)

## [1] 54.86669
```

We often want to compare averages across different subsets. The easiest way to do this is to create two different subsets and compute the average of interest for each one. Suppose we want to compare the mean sp_composite between Obama and Bush administrations

```
obama <- subset(mydata, president == 'Obama')
mean(obama$unemployment)

## [1] 0.07714286

bush <- subset(mydata, president == 'GWBush')
mean(bush$unemployment)

## [1] 0.055
```

Excercise 4) Compare the average ten_year_rate between Reagan, Clinton, and Truman.

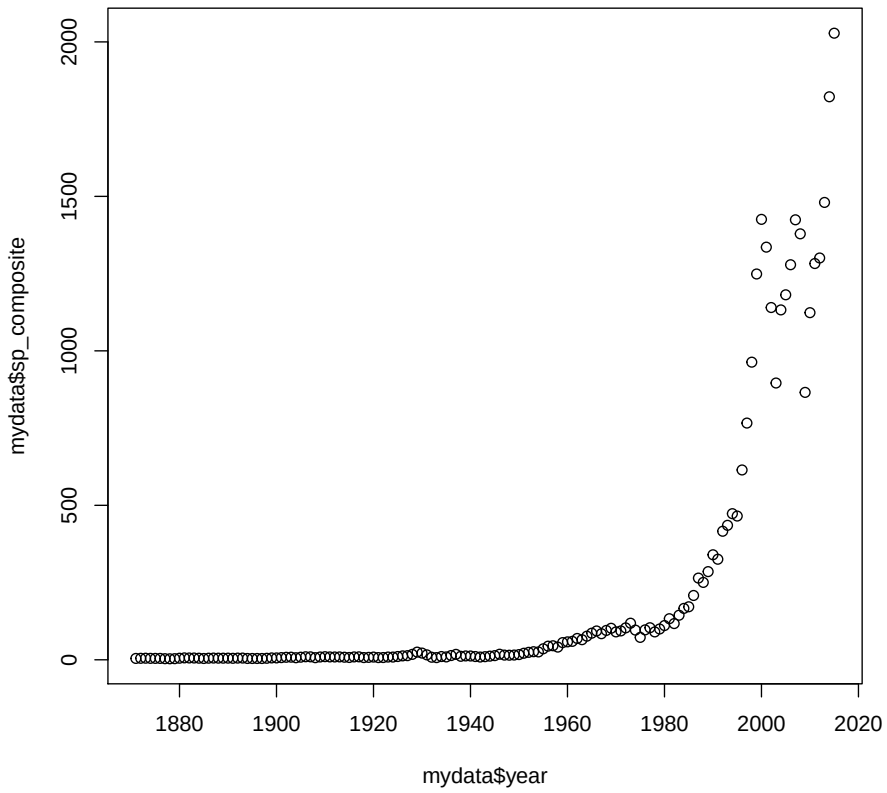
Excercise 5) Compare the mean unemployment rate between wartime and non wartime. Explain

1.3 Plots

The most basic plotting command in R is plot(). It is useful to type plot into the help window to see the different arguments In general, the format is plot(x,y, options) where x is the x-axis variable, y is the y-axis variable, and options allow you to specify different formats and label the axis and create a title

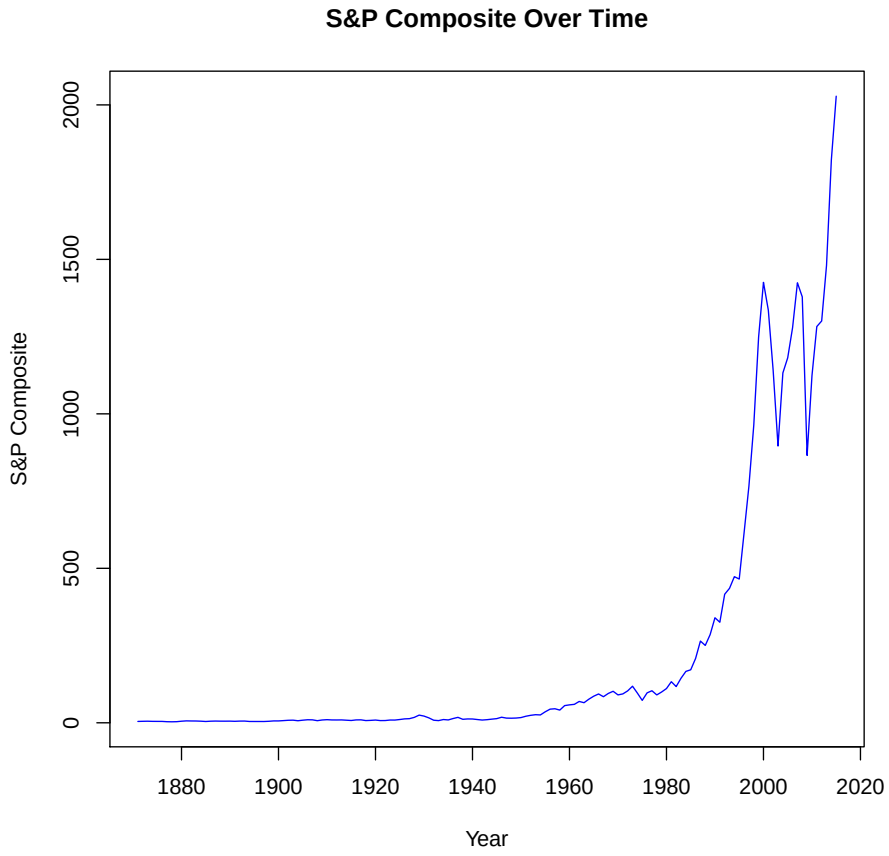
The following command plots the sp_composite for every year in the dataset

```
plot(mydata$year, mydata$sp_composite)
```



To add a title, label axis, and make a line graph that is colored blue, you can type

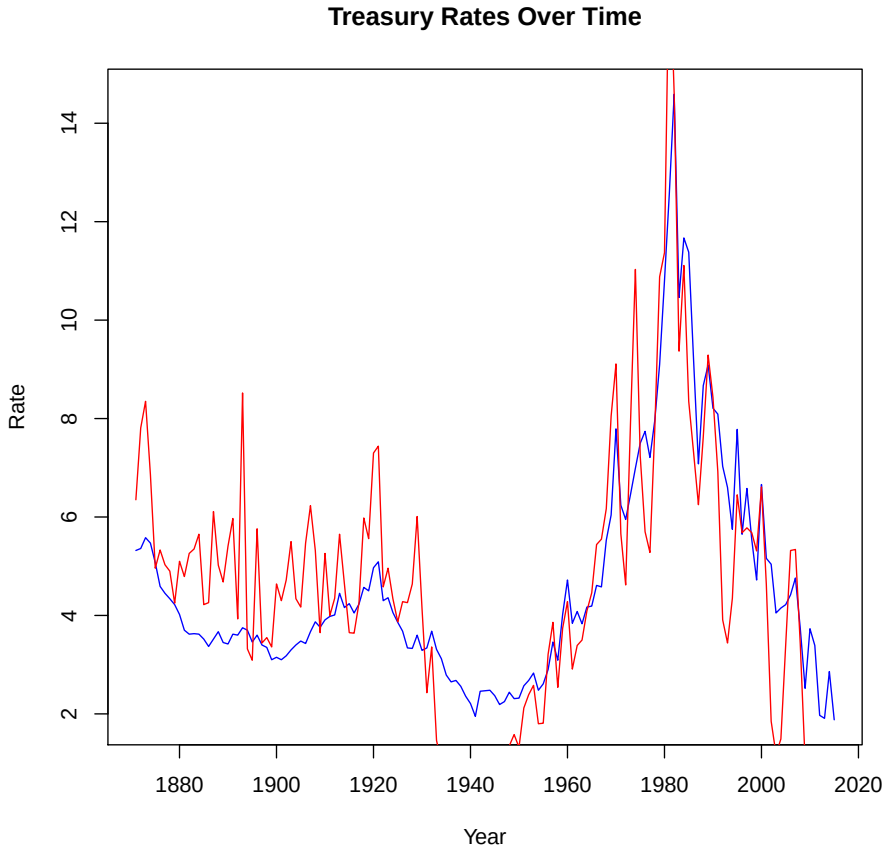
```
plot(mydata$year, mydata$sp_composite, type = 'l',
     main = 'S&P Composite Over Time',xlab = 'Year',ylab = 'S&P Composite',
     col = 'blue')
```



Excercise 6) Using only the years 1950-2015 plot the inlfation rate over time

To display multiple lines on one plot, first establish a plot using the commands above. Next use the lines() command to add a new line with a different series to the plot. The following code plots the ten_year_rate first, and then adds the one_year_rate to the plot using the lines command.

```
plot(mydata$year, mydata$ten_year_rate, type = 'l',
     main = 'Treasury Rates Over Time',xlab = 'Year',ylab = 'Rate',
     col = 'blue')
lines(mydata$year,mydata$one_year_rate, col = 'red')
```


Treasury Rates Over Time
Rate
2
4
6
8
10
12
14
1880
1900
1920
1940
1960
1980
2000
2020
Year