

# Case Study 2 for MSDS 6306-402

*James Hosker and Samuel Coyne*

*April 23, 2017*

## Introduction

This report answers 5 critical questions asked in Case Study 2 for MSDS 6306-402 (Thursday, 6:30 - 8:00 PM).

First, we will create and print out Matrix X in R, Python and SAS. Second, we will calculate and plot the Automatic Data Processing (ADP) volatility decayed over 10, 30 and 100 days. Third, we will calculate the mean and median circumference of five different trees; plot a scatterplot of trunk circumference vs. the age of the tree; and plot a boxplot of the tree type versus its circumferences. Fourth, we will look at the maximum and minimum monthly average temperature over a period of years by Country ("TEMP.csv") and by City ("Citytemp.csv") and compare them. We will examine the average monthly and yearly temperatures for the United States (US), converted from Celsius to Fahrenheit. In addition, we will look at the average temperature difference between years. Finally, we will plot using polar coordinates.

## Folder Description

The URL with the code and our analysis is on the GitHub website: <https://github.com/jjhosker/CaseStudy2>. It contains the following important files:

1. README: This file provides instructions on how to run the R code and provides information in detail of each of the files in the GitHub repository.
2. RMakefile\_CaseStudy2.R: This file can be found in the Analysis sub-directory under the CaseStudy2 GitHub repository. It is the main R makefile with the function calls.
3. Makefile\_GatherData.R: This file can be found in the Analysis/Data sub-directory under CaseStudy2 GitHub repository. This R makefile installs all required R-libraries and then loads the R-libraries. Second, this R makefile uploads the TEMP.csv and CityTemp.csv data files with monthly average temperature data by country and by city, respectively. This makefile also creates the raw data, and creates the cleaned datasets. In cleaning the data, we read in the csv files; we remove NAs and blanks in the data; and we reformat the date data. This makefile then saves all these datasets to the Analysis/Data sub-directory.
4. RFunctions\_CaseStudy2.R: This file can be found in the Analysis sub-directory under the CaseStudy2 GitHub repository. This file contains functions that are loaded and used the main R makefile(RMakefile\_CaseStudy2.R).
5. In addition to the two raw data sets, there are three data files created below and three PNG files which are loaded into the PDF file on creation. In addition, we provide a text file (SessionsInfo.txt) containing all the R session information including the current R version and all the libraries and packages being utilized. All the files below can be found in the Analysis/Data sub-directory under this GitHub repository.
  - Raw: TEMP.csv
  - Raw: CityTemp.csv
  - Cleaned: TEMPclean.csv
  - Cleaned: CityTempclean.csv
  - R version: SessionInfo.txt

- PNG: SAS\_MatrixCode.PNG
  - PNG: SAS\_MatrixOutput.PNG
  - PNG: Python\_Matrix.PNG
6. Finally, the CaseStudy2PDF.Rmd file runs the main RMakefile\_CaseStudy2.R and extracts code to generate this “CaseStudy2PDF.pdf” report. This report uses code that that is extracted from the main RMakefile\_CaseStudy2.R and then run.
- To use this Rmd file to create a PDF, the user has to install MikTeX (the non-basic version). The URL with instructions to install MikTeX is [MikTeX Install website](#).

## Running the R-Code

Before beginning please review the README.md file in GitHub website <https://github.com/jjhosker/CaseStudy2>. In short, download and clone the GitHub repository and the sub-directories. RMakefile\_CaseStudy2.R should be executed in the Analysis sub-directory of this GitHub repository; and CaseStudy2PDF.Rmd can be executed from this GitHub repository.

1. In the Analysis sub-directory in this repository, execute the main R source code RMakefile\_CaseStudy2.R, which performs the analysis for this report.
  - The RMakefile\_CaseStudy2.R loads function in the RFunctions\_CaseStudy2.R and executes the makefile Makefile\_GatherData.R in the Analysis/Data directory. Makefile\_GatherData.R installs the packages, loads the libraries and cleans the raw data. A message indicating that the libraries loaded may appear after executing the R code that follows.
  - If a failure occurs, please check the function “packagelibrary.check” in the Makefile\_GatherData.R and/or the SessionInfo.txt file, both found in the Analysis/Data sub-directory of the CaseStudy2 GitHub repository.
2. To generate this PDF report, execute CaseStudy2PDF.Rmd in this GitHub repository. Please note the required installations of software to generate the PDF is detailed in the prior section, Folder Description.

Here, we execute RMakefile\_CaseStudy2.R to generate all the analysis used in this report.

```
#####
## Execute Main RMakefile_CaseStudy2.R Makefile
#####
source("Analysis/RMakefile_CaseStudy2.R")
```

## Five Major Questions for CaseStudy2:

Question #1: Create the X matrix and print it from SAS, R, and Python.

### R-Code and Output of Matrix X

- a. The R code below prints the necessary information for creation of the matrix. For more details in R, see R-code section for Question #1 in RMakefile\_CaseStudy2.R.

```
## See R-code in RMakefile_CaseStudy2.R
xlist <- c(4,5,1,2,1,0,3,5,2,1,8,2)
X <- matrix(xlist,nrow=3,ncol=4,byrow=TRUE)
print(X)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    5    1    2
## [2,]    1    0    3    5
## [3,]    2    1    8    2
```

- b. We read in a PNG file below showing the execution of the code in Python using Jupyter.

Fig. 1: Python-code and Output of Matrix X

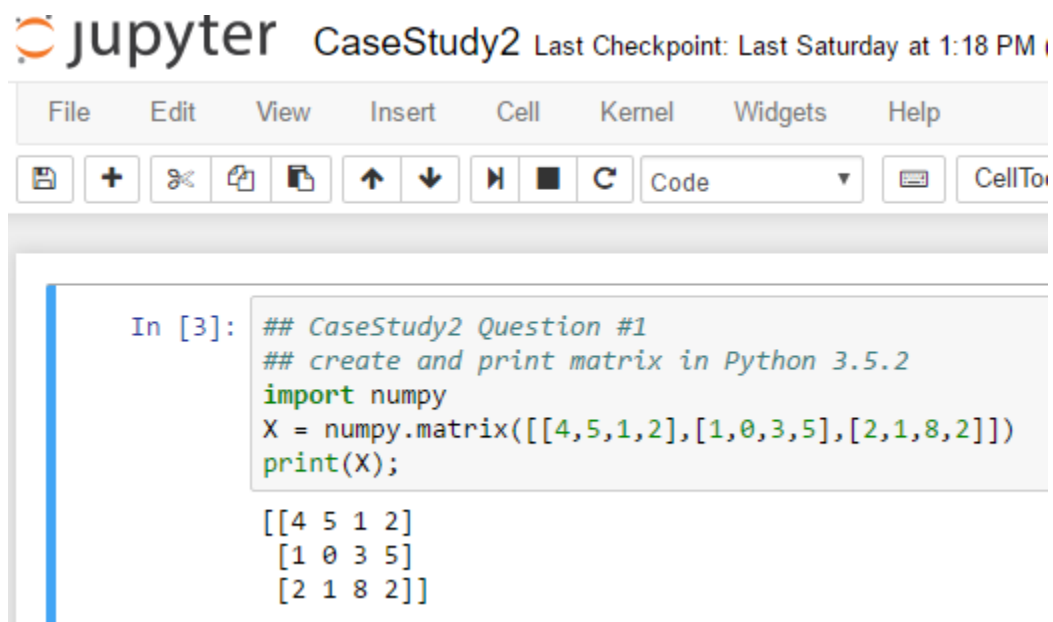
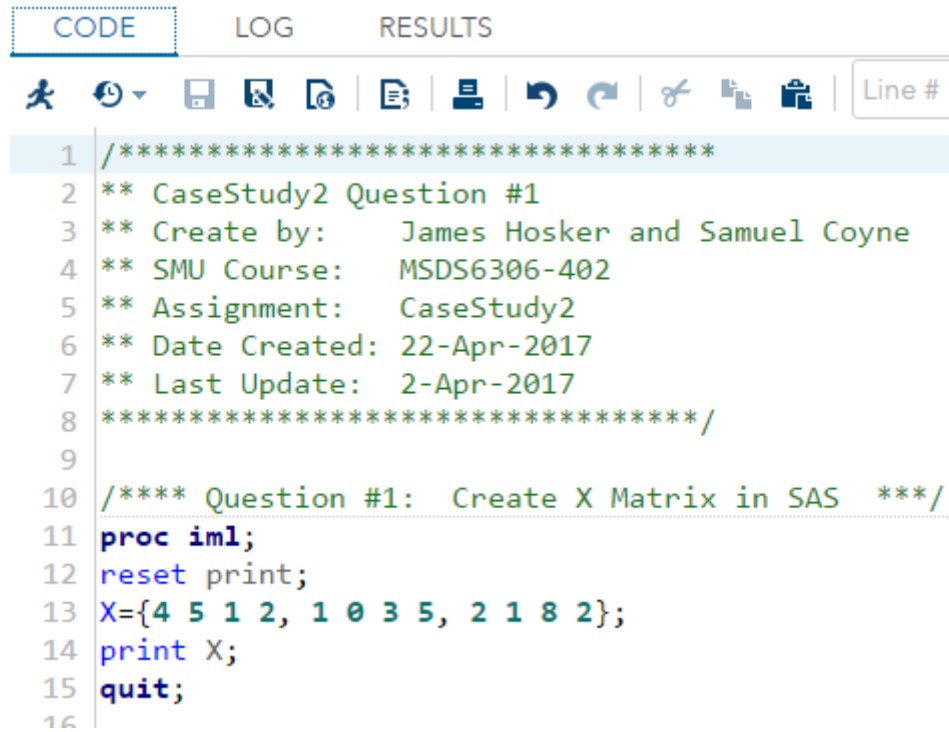


Figure 1: Python Jupyter: Matrix Output.

- c. We read in the two PNG files below showing the execution of the code in SAS using Jupyter on the next page.

Fig.2: SAS-code and Output of Matrix X



```
1 /*****  
2 ** CaseStudy2 Question #1  
3 ** Create by: James Hosker and Samuel Coyne  
4 ** SMU Course: MSDS6306-402  
5 ** Assignment: CaseStudy2  
6 ** Date Created: 22-Apr-2017  
7 ** Last Update: 2-Apr-2017  
8 *****/  
9  
10 /**** Question #1: Create X Matrix in SAS ****/  
11 proc iml;  
12 reset print;  
13 X={4 5 1 2, 1 0 3 5, 2 1 8 2};  
14 print X;  
15 quit;  
16
```

Figure 2: SAS Code for Matrix Output.



4	5	1	2
1	0	3	5
2	1	8	2

4	5	1	2
1	0	3	5
2	1	8	2

Figure 3: SAS Output Printing Matrix.

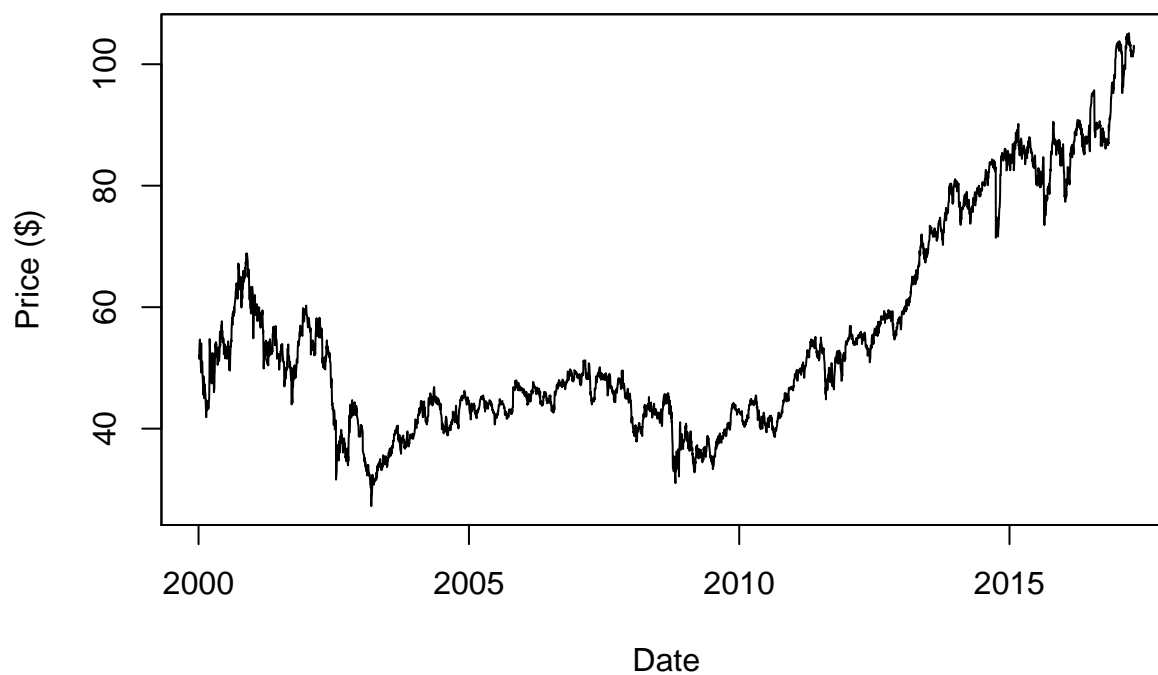
## Question #2: We do the following with your assigned stock: ADP.

The R code below prints the necessary information for creation of plot on volatility. For more details in R, see R-code section for Question #2 in RMakefile\_CaseStudy2.R.

- We downloaded the data for the stock Automatic Data Processing (ADP) from January 2000 to the current date and print out the graph below.

```
plot(ADPdata,xlab="Date",ylab="Price ($)"),  
     main= "Fig. 3: Price of Automatic Data Processing (ADP)")
```

**Fig. 3: Price of Automatic Data Processing (ADP)**



- We calculated the daily log returns for ADP over the same period.
- Calculate volatility measure over the entire series from Jan 2000 to the current date.

```
print(paste("ADP volatility of series from 01-Jan-2000 to 23-Apr-2017: ",round(ADPvol,2),"%"))
```

```
## [1] "ADP volatility of series from 01-Jan-2000 to 23-Apr-2017: 25.04 %"
```

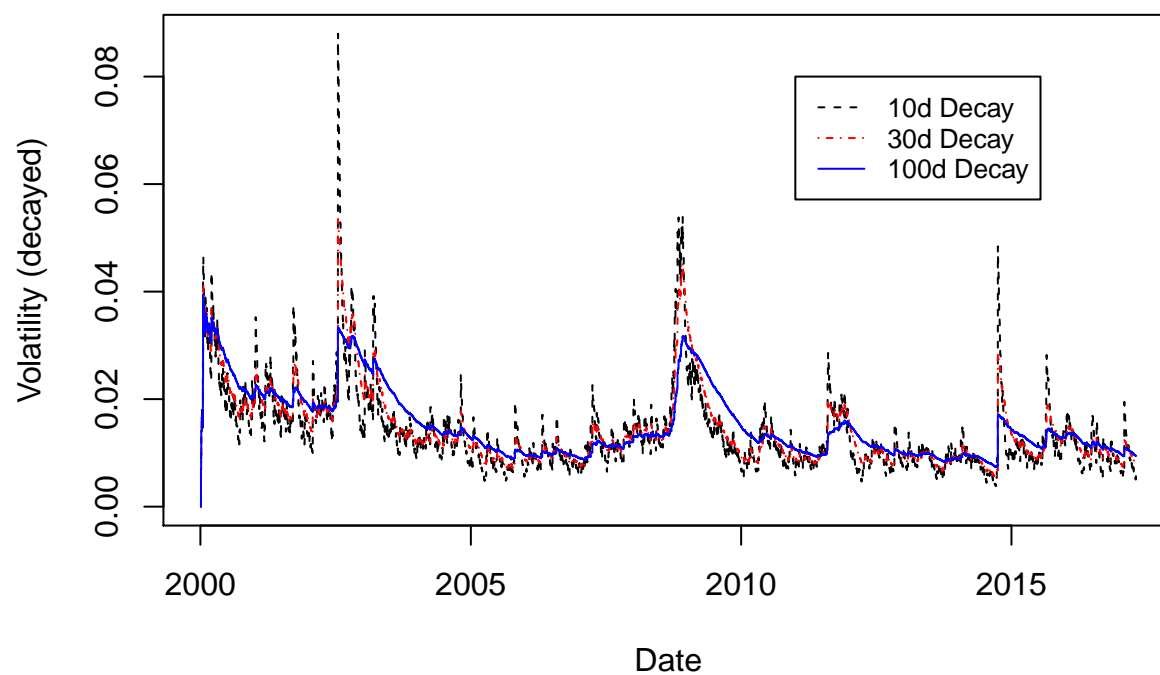
- We calculate volatility over entire length of series for various three different decay factors: 10, 30 & 100 days.
- Now we plot the results, overlaying the volatility curves on the data, just as was done in the S&P example but now for ADP

```
## Below in CaseStudy2PDF.Rmd file
```

```
plot(x=ddates,y=ADPvol_10d,type="s",xlab="Date", ylab="Volatility (decayed)",  
     main="Fig. 4: ADP Volatility using 10, 30 and 100 Day Decay", lty=2)  
lines(x=ddates,y=ADPvol_30d,type="s", col="red",lty=4)  
lines(x=ddates,y=ADPvol_100d,type="s", col="blue",lty=1)
```

```
legend(as.Date("2011-01-02"), 0.08, legend=c("10d Decay", "30d Decay", "100d Decay"),  
      col=c("black", "red", "blue"), lty=c(2,4,1), cex=0.8)
```

**Fig. 4: ADP Volatility using 10, 30 and 100 Day Decay**



### Question #3: We perform the following on the Orange data set.

The R code below prints the necessary information for creation of plots. For more details in R, see R-code section for Question #3 in RMakefile\_CaseStudy2.R.

**Q#3.a: Calculate the mean and the median of the trunk circumferences for different size of the trees. (Tree)**

```
## calculate the mean of trunk circumference for different Tree sizes
print(paste("Mean Trunk Circumference:"))
```

```
## [1] "Mean Trunk Circumference:"
```

```
print(TrunkMeanCir)
```

```
##      3      1      5      2      4
## 94.00000 99.57143 111.14286 135.28571 139.28571
```

```
## calculate the median of trunk circumference for different Tree sizes
print(paste("Median Trunk Circumference:"))
```

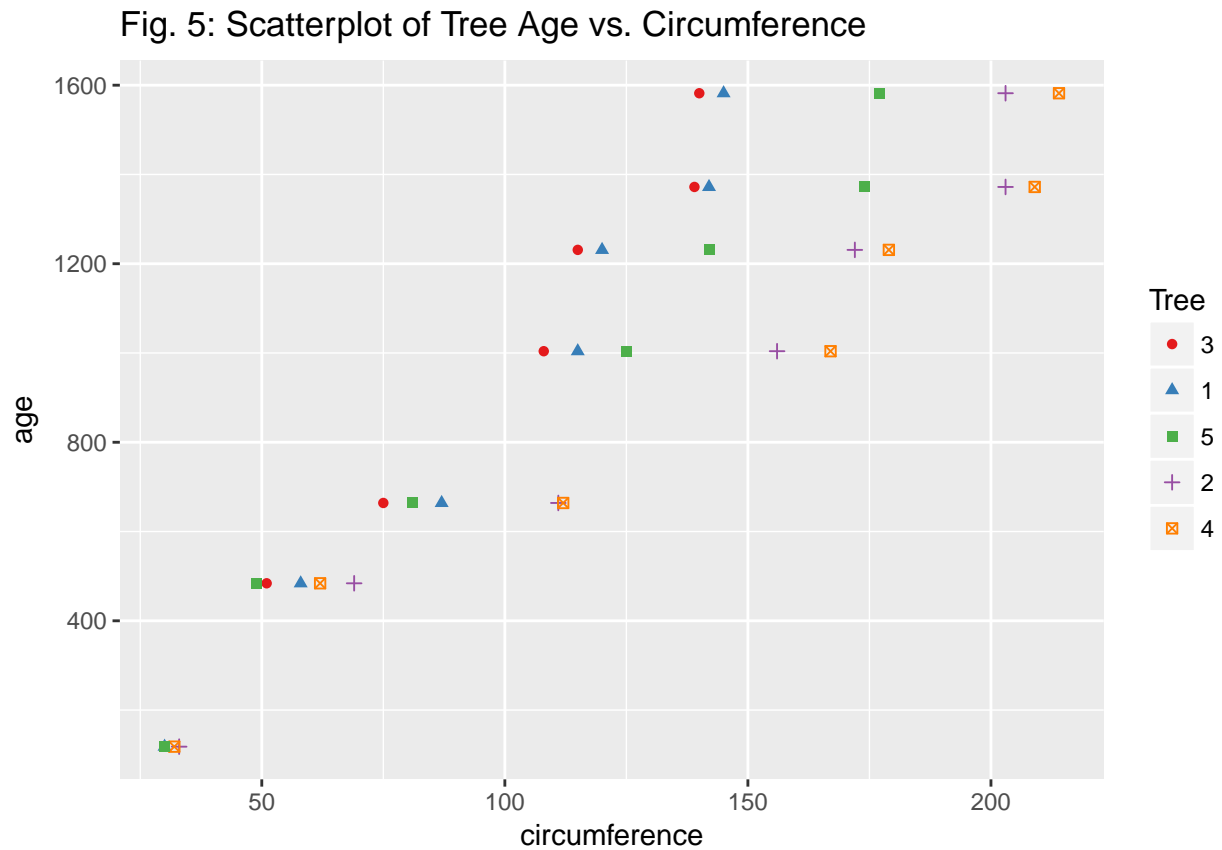
```
## [1] "Median Trunk Circumference:"
```

```
print(TrunkMedCir)
```

```
##      3      1      5      2      4
## 108 115 125 156 167
```

Q#3.b: Make a scatter plot of the trunk circumferences against the age of the tree. Use different plotting symbols for different size of trees.

```
## make a scatterplot of age vs. circumference with different plotting symbols
ggplot(Orange, aes(x=circumference, y=age, shape=Tree, colour=Tree)) +
  geom_point() + scale_colour_brewer(palette="Set1") +
  ggtitle("Fig. 5: Scatterplot of Tree Age vs. Circumference")
```

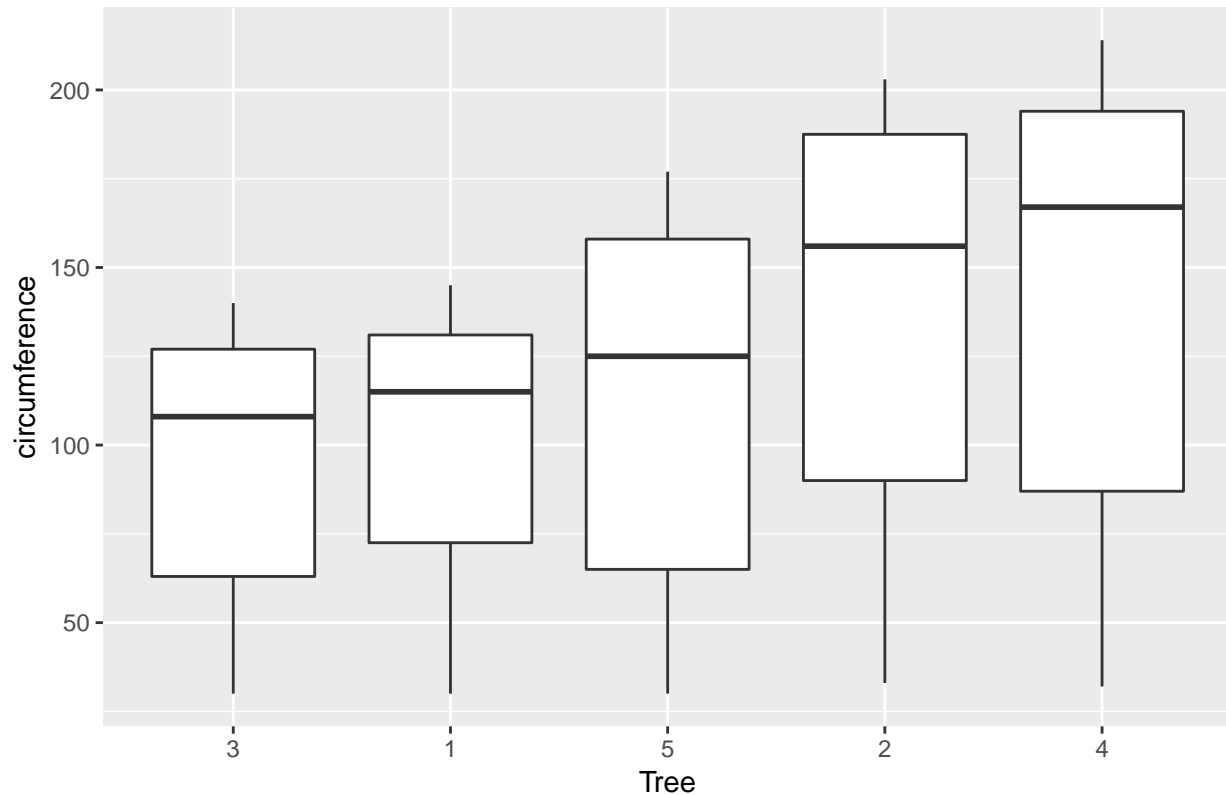




Q#3.c: Display the trunk circumferences on a comparative boxplot against tree. Be sure you order the boxplots in the increasing order of maximum diameter.

```
## make a comparative boxplot of tree vs. circumference, displayed  
## in increasing order of max circumference  
ggplot(Orange, aes(x=Tree, y=circumference)) + geom_boxplot() +  
  ggtitle("Fig. 6: BoxPlot of Tree vs. Circumference")
```

Fig. 6: BoxPlot of Tree vs. Circumference



## Question #4: Analyze the TEMP.csv data

The R code below prints the necessary information for creation of plots. For more details in R, see R-code section for Question #4 in RMakefile\_CaseStudy2.R.

**Q#4.i:** Below is table and graph of difference between the maximum and the minimum monthly average temperatures for each country, for which we report and visualize the top 20 countries with the maximum differences for the period since 1900 from TEMP.csv.

```
## R Code for CaseStudy2PDF.Rmd for Summary Table output
kable(top_20_country, format="pandoc", align='c', row.names = FALSE,
      col.names = TableNames,
      caption = "Top 20 Countries with Largest Monthly Avg Max - Min Temp. (1900 TO 2013)")
```

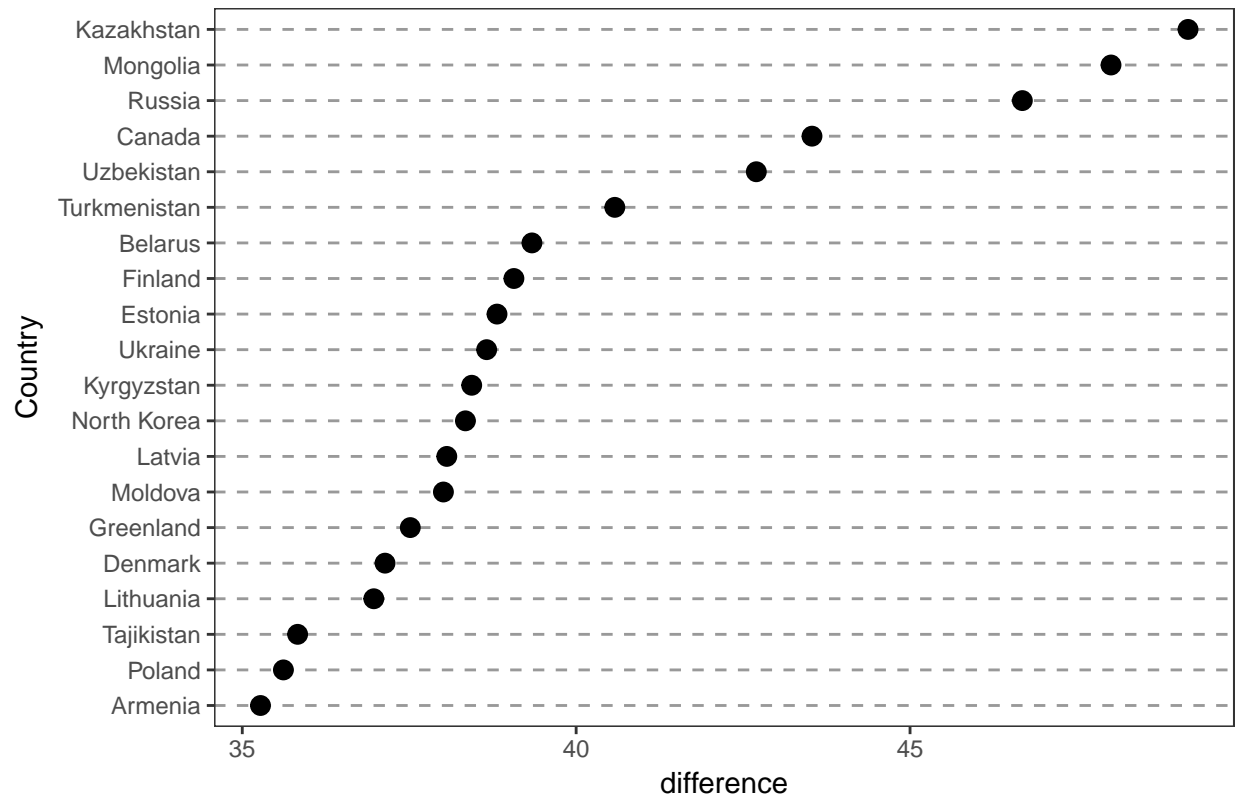
Table 1: Top 20 Countries with Largest Monthly Avg Max - Min Temp. (1900 TO 2013)

Country	Temp. Min	Temp. Max	Diff (Max-Min)
Kazakhstan	-23.601	25.562	49.163
Mongolia	-27.294	20.716	48.010
Russia	-29.789	16.893	46.682
Canada	-28.736	14.796	43.532
Uzbekistan	-12.323	30.375	42.698
Turkmenistan	-8.443	32.136	40.579
Belarus	-16.527	22.811	39.338
Finland	-20.101	18.967	39.068
Estonia	-16.483	22.332	38.815
Ukraine	-14.724	23.936	38.660
Kyrgyzstan	-19.161	19.275	38.436
North Korea	-14.390	23.952	38.342
Latvia	-15.784	22.279	38.063
Moldova	-12.781	25.231	38.012
Greenland	-37.177	0.339	37.516
Denmark	-36.439	0.699	37.138
Lithuania	-15.179	21.791	36.970
Tajikistan	-16.466	19.363	35.829
Poland	-13.107	22.509	35.616
Armenia	-9.982	25.291	35.273

```
#make a visual of top 20 countries and their temp difference
visual1 <- ggplot(top_20_country, aes(x=difference, y=reorder(Country, difference))) +
  geom_point(size=3) + ylab("Country") + theme_bw() +
  theme(panel.grid.major.x=element_blank(), panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(colour="grey60", linetype="dashed")) +
  ggtitle("Fig. 7: Top 20 Monthly Avg Max-Min Temp by Country (1900 to 2013)")

#show the plot
visual1
```

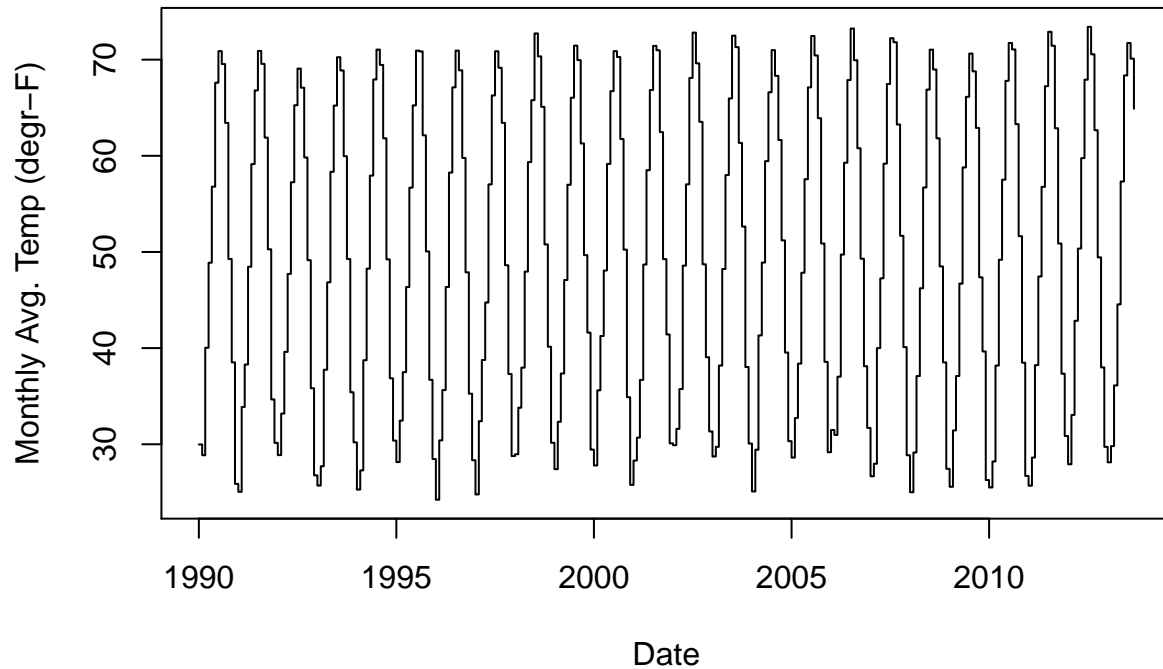
Fig. 7: Top 20 Monthly Avg Max–Min Temp by Country (1900 to 201



Q#4.ii.a: Create a new column to display the monthly average land temperatures in Fahrenheit (°F).

```
## plot UStemp monthly avg temp for USA
plot(UStemp$date2, UStemp$MthAvgTemp.Fahrenheit,type="s",xlab="Date",
      ylab="Monthly Avg. Temp (degr-F)",
      main="Fig. 8: Monthly Average Temperature for USA (Fahrenheit)")
```

**Fig. 8: Monthly Average Temperature for USA (Fahrenheit)**



Q#4.ii.b: Calculate average land temperature by year and plot it. The original file has the average land temperature by month.

```
## make a table of yearly temp in US
kable(TableFrame, format="pandoc", align='c',row.names = FALSE,
      col.names = c("Year", "Yearly Temperature Avg. (Fahrenheit)"),
      caption = "Yearly Avg. Temperature for USA (1990-2013)")
```

Table 2: Yearly Avg. Temperature for USA (1990-2013)

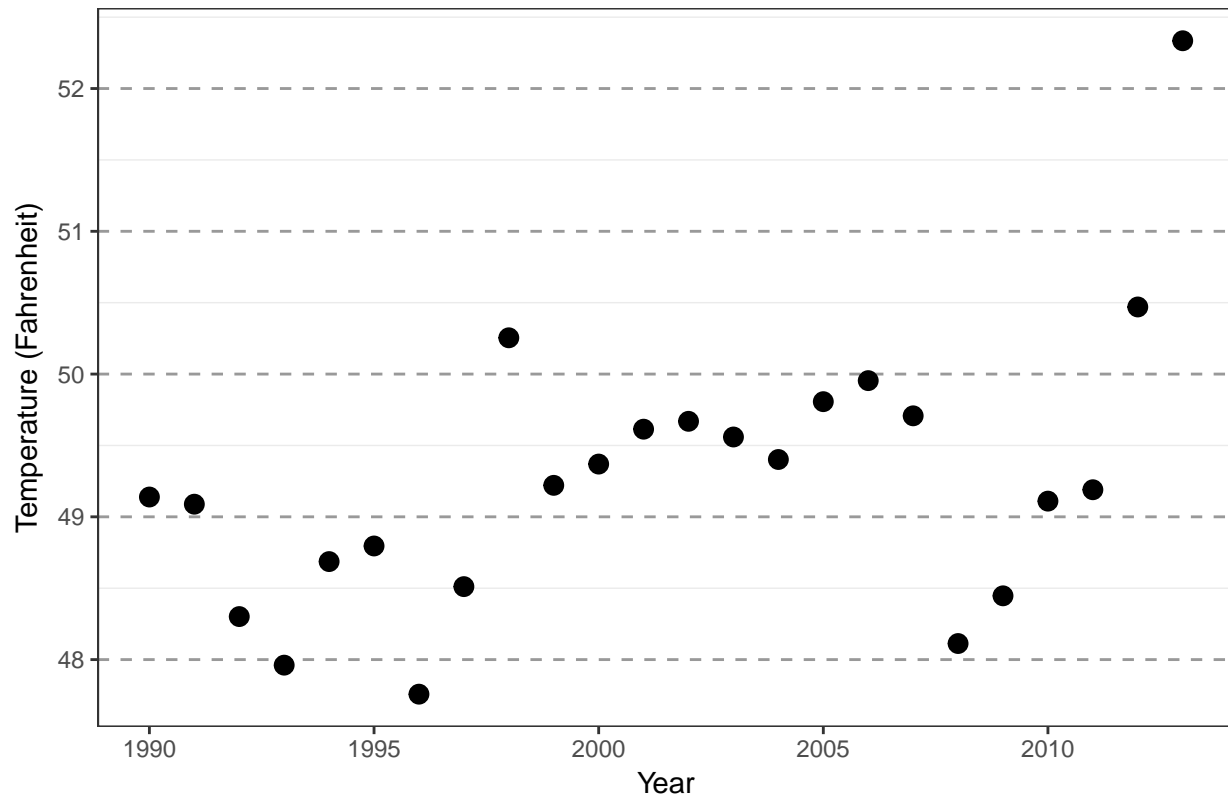
Year	Yearly Temperature Avg. (Fahrenheit)
1990	49.13885
1991	49.08860
1992	48.30140
1993	47.96105
1994	48.68660

Year	Yearly Temperature Avg. (Fahrenheit)
1995	48.79580
1996	47.75780
1997	48.51125
1998	50.25410
1999	49.22060
2000	49.36955
2001	49.61405
2002	49.66910
2003	49.55915
2004	49.40150
2005	49.80680
2006	49.95410
2007	49.70750
2008	48.11255
2009	48.44675
2010	49.11005
2011	49.18985
2012	50.46995
2013	52.33480

```
## make a visual of yearly temp in US
visual3 <- ggplot(TableFrame, aes(x=Year, y=YearlyAvg)) +
  geom_point(size=3) + ylab("Temperature (Fahrenheit)") + xlab("Year") + theme_bw() +
  theme(panel.grid.major.x=element_blank(), panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(colour="grey60", linetype="dashed")) +
  ggtitle("Fig. 9: Yearly Avg. Temperature for USA (1990-2013)")

#show the plot
visual3
```

Fig. 9: Yearly Avg. Temperature for USA (1990–2013)



Q#4.ii.c: Calculate the one year difference of average land temperature by year and provide the maximum difference (value) with corresponding years.

```
kable(TableFrame2, format="pandoc", align='c', row.names = FALSE,
      col.names = c("Range of Years", "Yearly Avg. Temp. Diff of Year - Prior Year (Fahrenheit)"),
      caption = "Yearly Avg. Temperature Between Years for USA (1990-2013)")
```

Table 3: Yearly Avg. Temperature Between Years for USA (1990-2013)

Range of Years	Yearly Avg. Temp. Diff of Year - Prior Year (Fahrenheit)
1991 - 1990	-0.05
1992 - 1991	-0.787
1993 - 1992	-0.34
1994 - 1993	0.726
1995 - 1994	0.109
1996 - 1995	-1.038
1997 - 1996	0.753
1998 - 1997	1.743
1999 - 1998	-1.034
2000 - 1999	0.149
2001 - 2000	0.244
2002 - 2001	0.055
2003 - 2002	-0.11

Range of Years	Yearly Avg. Temp. Diff of Year - Prior Year (Fahrenheit)
2004 - 2003	-0.158
2005 - 2004	0.405
2006 - 2005	0.147
2007 - 2006	-0.247
2008 - 2007	-1.595
2009 - 2008	0.334
2010 - 2009	0.663
2011 - 2010	0.08
2012 - 2011	1.28
2013 - 2012	1.865

```
print(paste("Largest avg. yearly temp. difference (+/-) for USA is:",
  SortBtwYrAvg$Diff[1], " Fahrenheit (", SortBtwYrAvg$YearDiff[1], ")"))
```

```
## [1] "Largest avg. yearly temp. difference (+/-) for USA is: 1.865 Fahrenheit ( 2013 - 2012 )"
```

**Q#4.iii: Download “CityTemp” data set (check your SMU email). We find the difference between the maximum and the minimum temperatures for each major city and report/visualize top 20 cities with maximum differences for the period since 1900.**

```
## R Code for CaseStudy2PDF.Rmd for Summary Table output
kable(top_20_cities, format="pandoc", align='c', row.names = FALSE,
  col.names = TableNames3,
  caption = "Top 20 Cities with Largest Monthly Avg Max - Min Temp. (1900 TO 2013)")
```

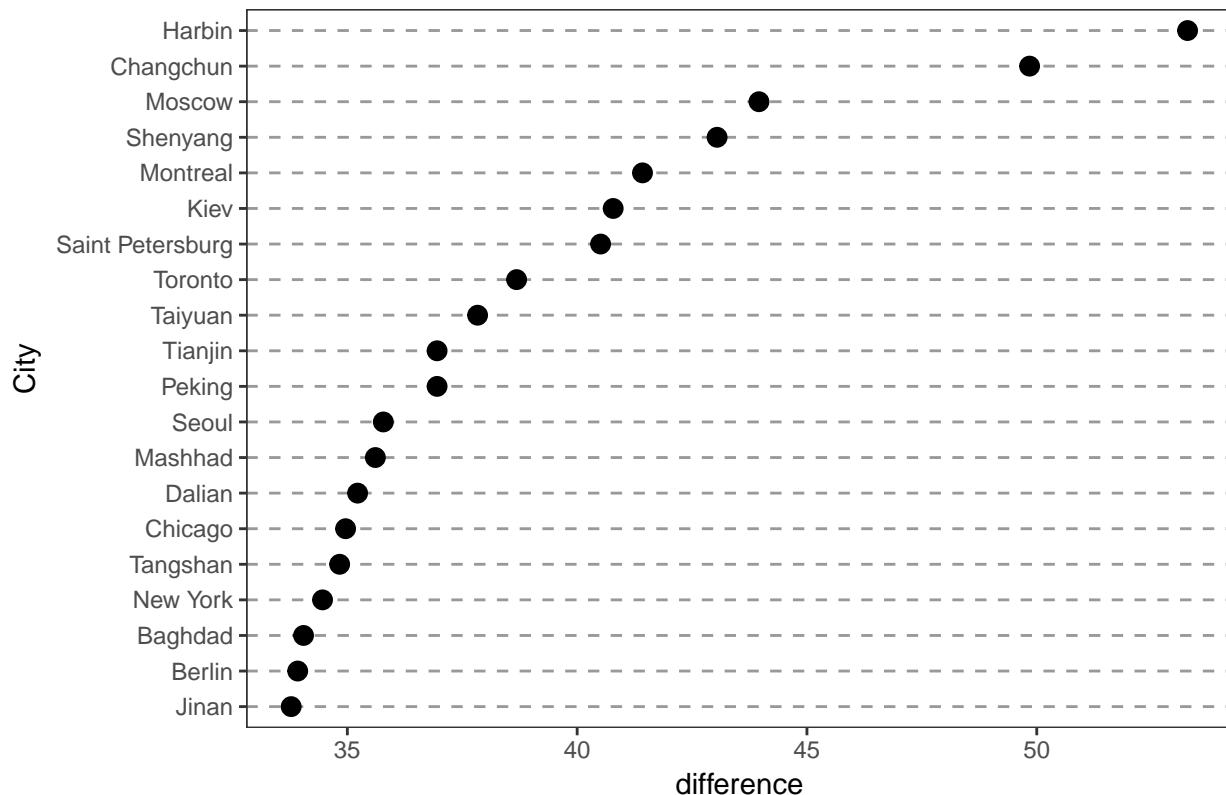
Table 4: Top 20 Cities with Largest Monthly Avg Max - Min Temp.  
(1900 TO 2013)

City	Country	Temp. Min	Temp. Max	Diff (Max-Min)
Harbin	China	-26.772	26.509	53.281
Changchun	China	-23.272	26.572	49.844
Moscow	Russia	-19.376	24.580	43.956
Shenyang	China	-17.035	26.010	43.045
Montreal	Canada	-18.363	23.059	41.422
Kiev	Ukraine	-16.191	24.593	40.784
Saint Petersburg	Russia	-18.589	21.921	40.510
Toronto	Canada	-15.502	23.181	38.683
Taiyuan	China	-13.116	24.718	37.834
Peking	China	-8.017	28.936	36.953
Tianjin	China	-8.017	28.936	36.953
Seoul	South Korea	-8.992	26.791	35.783
Mashhad	Iran	-8.384	27.226	35.610
Dalian	China	-9.348	25.875	35.223
Chicago	United States	-8.590	26.372	34.962
Tangshan	China	-7.487	27.346	34.833
New York	United States	-9.147	25.313	34.460
Baghdad	Iraq	4.236	38.283	34.047
Berlin	Germany	-10.125	23.795	33.920
Jinan	China	-5.389	28.389	33.778

```
#make a visual of top 20 cities and their temp difference
visual2 <- ggplot(top_20_cities, aes(x=difference, y=reorder(City, difference))) +
  geom_point(size=3) + ylab("City") + theme_bw() +
  theme(panel.grid.major.x=element_blank(), panel.grid.minor.x = element_blank(),
  panel.grid.major.y = element_line(colour="grey60", linetype="dashed")) +
  ggtitle("Fig. 10: Top 20 Monthly Avg Max-Min Temp by City (1900 to 2013)")

#show the plot
visual2
```

Fig. 10: Top 20 Monthly Avg Max–Min Temp by City (1900 to 2013)



Q#4.iv: Compare the two graphs in (4.i) and (4.iii) and comment on it below:

1. We notice that cities in China (e.g. Harbin, Changchun, Shenyang, Taiyuan, Peking, Tianjin and others) are in the top 20 cities with the largest monthly average temperature difference in Celsius between the maximum and minimum from 1900 to 2013 but China as a country is not in the to 20 countries with the largest monthly difference between the maximum and minimum temperature. However, nearby countries like Mongolia are.
2. We notice that the cities of Montreal and Toronto in Canada are in the top 20 cities with the largest difference in monthly average temperature and Canada is one of the top 20 countries with the largest monthly difference between the maximum and minimum temperature.
3. We notice that the cities of Moscow and Saint Petersburg in Russia are in the top 20 cities with the largest difference in monthly average temperature and Russia is one of the top 20 countries with the largest monthly difference between the maximum and minimum temperature. In addition countries nearby such as Kazakhstan, Uzbekistan, Estonia, Ukraine



and other are on the top countries with the largest monthly difference between the maximum and minimum temperature.

4. We notice that the cities of New York and Chicago in the United States are in the top 20 cities with the largest difference in monthly average temperature but overall the United States is not in the top 20 countries with the largest monthly difference between the maximum and minimum temperature.

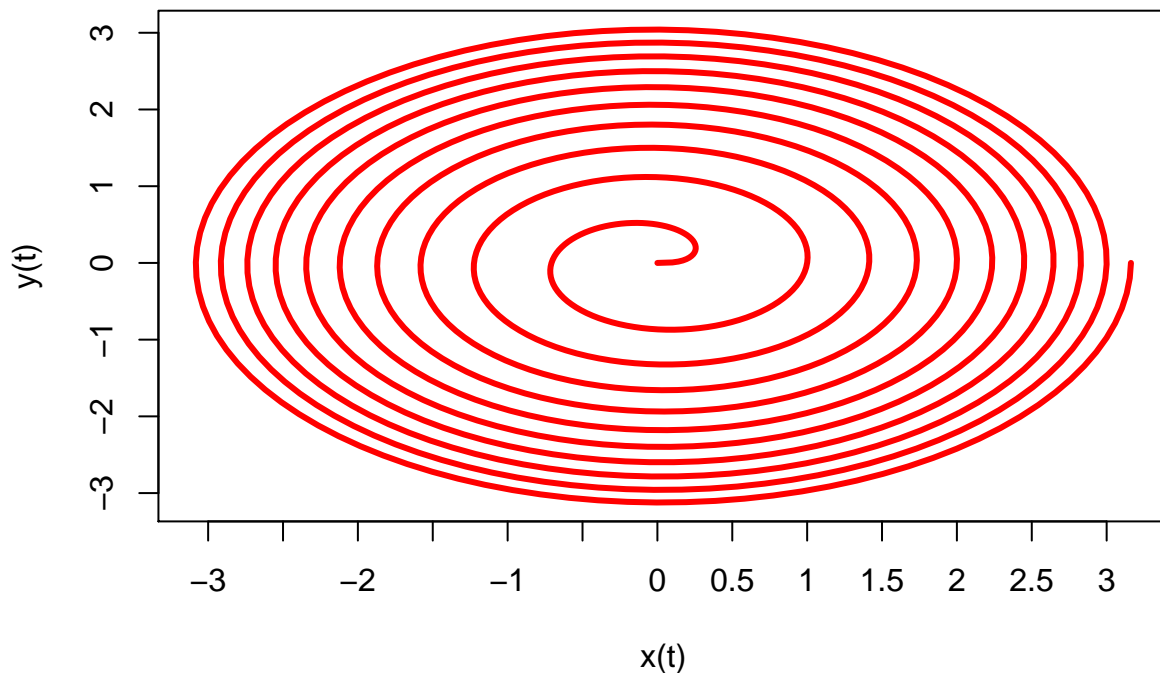
### Question #5: Semester Bonus Question

We Run the R-code below and print out the graph. For more details in R, see R-code section for Question #5 in RMakefile\_CaseStudy2.R.

```
## Question #5: Semester End Bonus Plot
t<-seq(0,10,length=1000)
x<-sqrt(t)*cos(2*pi*t)
y<-sqrt(t)*sin(2*pi*t)
plot(x,y,axes=F,type="l",lwd=3,xlab="x(t)",ylab="y(t)",col="red")

axis(1,at=seq(-3,3,by=0.5),labels=seq(-3,3,by=0.5))
axis(2)
box()
title(main=expression(
  paste("Fig. 11: (x(t),y(t)) with polar coordinates",
    (list(sqrt(t),2*pi*t))
  )))
```

Fig. 11:  $(x(t),y(t))$  with polar coordinates  $(\sqrt{t}, 2\pi t)$



## Conclusions:

In this report, we answer five questions/tasks and have provided five answers.

1. We created and printed out Matrix X in R, Python and SAS.
2. We downloaded the daily stock prices for Automatic Data Processing (ADP) from Yahoo finance and plotted it. We then calculated and plotted the ADP volatility decayed over 10, 30 and 100 days.
3. We analyzed the native "Orange" data set and calculated the mean and median circumference of five different trees; plotted a scatterplot of trunk circumference vs. the age of the tree; and plotted a boxplot of the tree type versus its circumferences.
4. We uploaded the "TEMP.csv" and "CityTemp.csv" data set and found the difference in the maximum and minimum monthly average temperature over a period of years by Country ("TEMP.csv") and by City ("Citytemp.csv"). In addition, we compared the two outputs by City versus Country. Moreover, we selected a subset of average monthly temperatures for the United States (US); converted them from Celsius to Fahrenheit; calculated the average yearly temperature; and calculated the difference in maximum and minimum temperatures by year.
5. Finally, we plotted a graph with x and y polar coordinates.