MSDS 6306 401: Case Study 2

Yao Yao, Robert Flamenbaum

April 18, 2017

Contents

Introduction:

This case study is an exercise of programming in SAS, python, and R for number 1. Question 2, 3, and 4 are exercises in cleaning and analyzing data using tables and ggplot. R markdown was used to source different data files and the creation of the paper file.

The following code is an exploratory analysis on three data sets which will demonstrate the various ways of transforming and analyzing data in R. The results of the analysis are meant to provide a clear way to view the data sets and obtain key observations in the data.

Question 2 is an exploration of stock data and time series analysis of volatility, trends, and log returns.

Question 3 uses a data set called Orange, which consists of 35 observations and 3 variables. The first variable, Tree, simply identifies the tree being measured. The second variable, age, describes the age of the tree in days after 12/31/1968. The third variable, circumference, is a measurement in millimeters of the tree circumference at breast height. The data consists of 5 trees which are assumed to be the same age. The trees were measured on 7 different occasions at the same time interval. The measurements, taken at each time interval, provide a detailed view of how the trees in the study vary in growth over time.

Question 4 parts i and ii uses a data set called Temp, which consists of a total of 574,223 observations and 4 variables. Each record measures average monthly temperature in degrees Celsius for countries spanning a time range from 1838 to 2013. There is also a variable called Monthly.AverageTemp.Uncertainty, which notes the level of error in each of the observations. There is no treatment or other descriptive attributes, so no conclusions will be able to be made from this data set. The data can however be used to observe any discernable patterns that exist in the data set.

Question 4 parts iii and iv uses a data set called CityTemp, which consists of 237,200 observations of 7 variables. The data consists of all the same variables as Temp, but with the addition of 3 variables which narrow down the geographic region of the measured data to cities within the countries. Also, a latitude and longitude are given for each observation. It is unclear as to where the latitude and longitude reside within the confines of the city in question. The coordinates could point to the geographic center of the city, or perhaps, it may point to an actual sampling point within the city. For the purposes of this study, which does not include a geographically acute analysis, it is not important to know the exact location where the coordinates point to.

Question 5 is an exploration of polar coordinates for sine and consine.

All data sets are stored in .csv format where headers are imported directly from the original data set, where the columns that are not used to answer the questions in the analysis are then eliminated. The columns had their names changed accordingly per analysis, where the full data frame is exported in its own .csv file.

Directions to run the code:

```
InstallLoadMultPackage <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}
InstallLoadMultPackage(c("plyr", "ggplot2", "dplyr","data.table","pander","knitr","tseries","zoo"))
```

```
##       plyr    ggplot2      dplyr data.table     pander      knitr
##       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
##    tseries        zoo
##       TRUE       TRUE
```

```
opts_knit$set(root.dir = 'C:\\Users\\Yao\\Documents\\GitHub\\DDS-Case-Study-2\\Data')
```

Question 2 (15 points)

Please watch videos1 and 2 in week 11 lecture assignment. You can download the code which used for S&P from files tab.

Please do the following with your assigned stock. Yao and Robert: AGIO

Download the data.

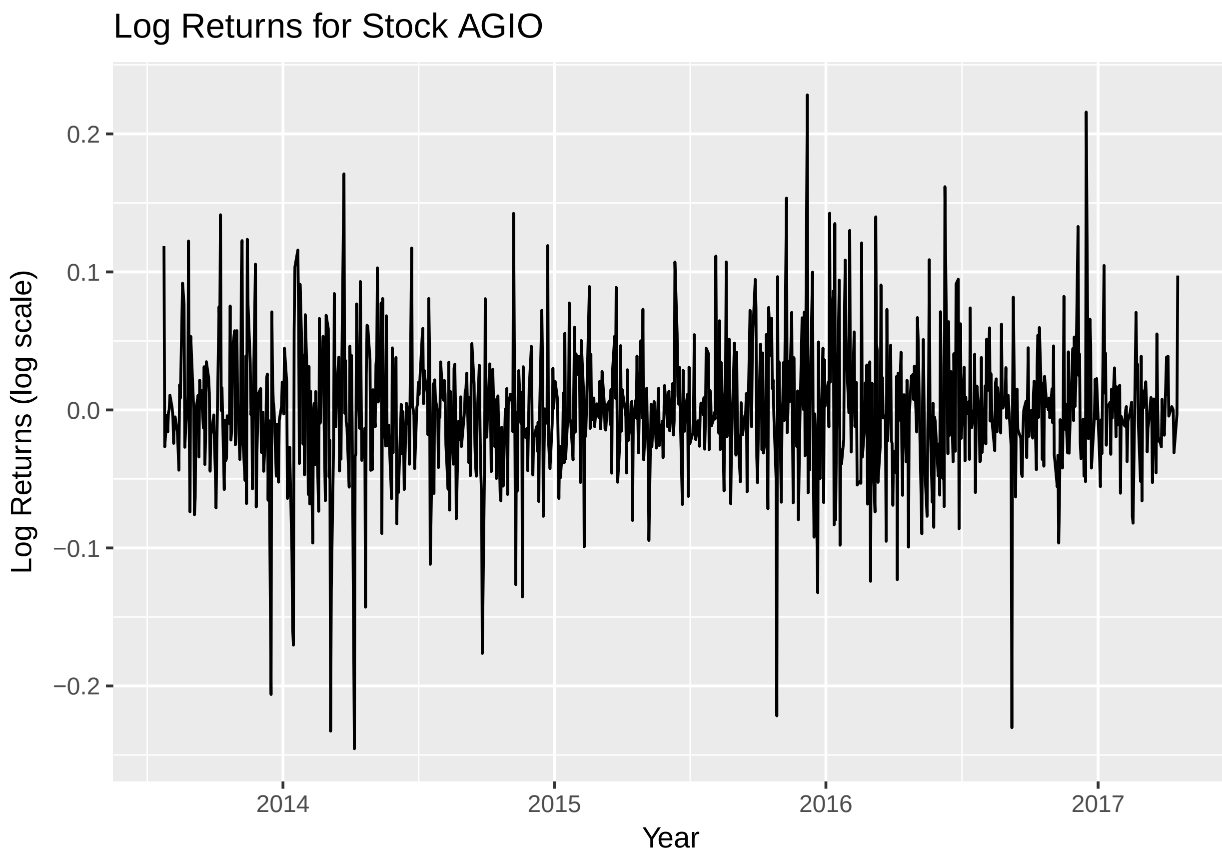
```
#Function to download the historical data and calculates log returns and historical volatility
#Returns dataframe of 8 sets of data
AnalyzeStock <- function(ticker){
  DailyClosingPrice <- get.hist.quote(ticker,quote="Close",quiet=TRUE)
  DailyClosingPrice<-na.trim(DailyClosingPrice, sides = "both")
  begindt <- start(DailyClosingPrice)
  enddt <- end(DailyClosingPrice)
  obs=length(DailyClosingPrice)
  logreturns <- log(lag(DailyClosingPrice))-log(DailyClosingPrice)
  logreturns <- na.trim(logreturns, sides = "both")
  # 252 trading days per year
  vol <- round(sd(logreturns) * sqrt(252)*100,2)
  plottitle <- paste("Historical Closing price for ",ticker," (",begindt, " to ", enddt,")\n",
                     " Obs = ",obs,"  ","Historical Volatility = ",vol)
  list(Ticker = ticker,
       Data=DailyClosingPrice,
       Obs=obs,
       LogReturns=logreturns,
       BeginDate=begindt,
       EndDate=enddt,
       Volatility=vol,
       PlotTitle = plottitle
  )
}
```

```r
#plots the stock with crucial information in the title
stockAGIO <- AnalyzeStock("AGIO")
write.csv(stockAGIO$Data, "DailyClosingPrice.csv")
autoplot.zoo(stockAGIO$Data) +
  ggtitle(stockAGIO$PlotTitle) + xlab("Year") + ylab("Closing Price (U.S. Dollars)")
```

Historical closing prices for Agios Pharmaceuticals Inc (AGIO) in US dollars are graphed from starting date to current day. AGIO is a recent stock that started in 2013 with peak prices in 2015.

Calculate log returns.

```
#plots the log returns and writes it
write.csv(stockAGIO$LogReturns, "LogReturns.csv")
autoplot.zoo(stockAGIO$LogReturns) +
  ggtitle("Log Returns for Stock AGIO") + xlab("Year") + ylab("Log Returns (log scale)")
```



The log returns on the log scale for Agios Pharmaceuticals Inc (AGIO) in US dollars are graphed from starting date to current day. The log returns hover slightly above 0 for positive investment.

Calculate volatility measure.

```r
paste("Historical Volatility =",stockAGIO$Volatility)
```

```
## [1] "Historical Volatility = 78.66"
```

The historical volatitlity measure of Agios Pharmaceuticals Inc (AGIO) is 78.66, which used 252 as the number of trading days in the calculation.

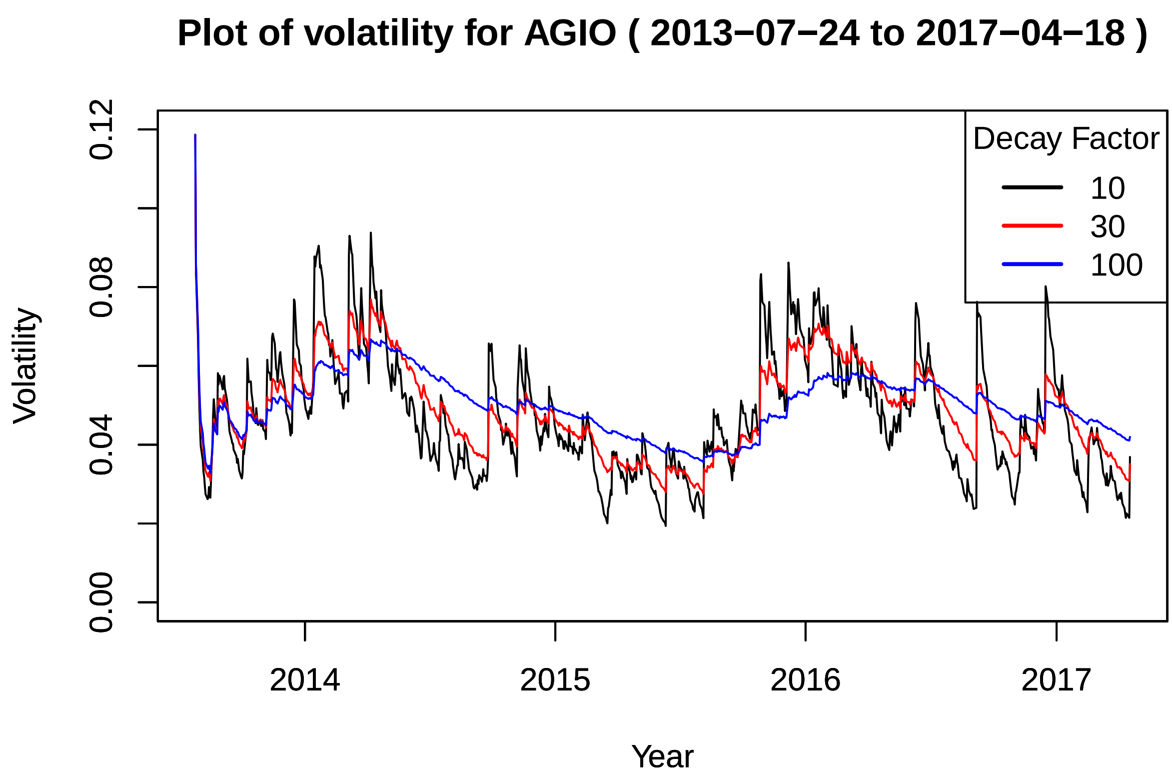
Calculate volatility over entire length of series for various three different decay factors.

```r
# volatility function
Vol <- function(d, logreturns)
{
  var = 0
  lam = 0
  varlist <- c()
  for (r in logreturns) {
    lam = lam*(1 - 1/d) + 1
    var = (1 - 1/lam)*var + (1/lam)*r^2
    varlist <- c(varlist, var)
  }
  sqrt(varlist)
}
#retrieve volatility for decays 10, 30, and 100
stockAGIO$LogReturns$vol10 <- Vol(10,stockAGIO$LogReturns$Close)
stockAGIO$LogReturns$vol30 <- Vol(30,stockAGIO$LogReturns$Close)
stockAGIO$LogReturns$vol100 <- Vol(100,stockAGIO$LogReturns$Close)
write.csv(stockAGIO$LogReturns, "LogReturnsVolatility.csv")
```

The volatility over the entire length of Agios Pharmaceuticals Inc (AGIO) for 10, 30, and 100 decay factors are calculated.

Plot the results, overlaying the volatility curves on the data, just as was done in the S&P example.

```
#plotting the overlay of 3 volatility functions over the same period
plot(stockAGIO$LogReturns$vol10,type="l",xlab="Year",ylab="Volatility",ylim= c(0,0.12))
par(new=TRUE)
plot(stockAGIO$LogReturns$vol30,type="l",xlab="Year",ylab="Volatility",ylim= c(0,0.12),col="red")
par(new=TRUE)
plot(stockAGIO$LogReturns$vol100,type="l",xlab="Year",ylab="Volatility",ylim= c(0,0.12),col="blue")
title(main=paste("Plot of volatility for",stockAGIO$Ticker,"(",
                 stockAGIO$BeginDate,"to",stockAGIO$EndDate,")"))
legend("topright",c("10","30","100"),lty=c(1,1,1),lwd=c(2,2,2),
       col=c("black","red","blue"),title="Decay Factor")
```



The volatility over the entire length of Agios Pharmaceuticals Inc (AGIO) for 10, 30, and 100 decay factors plotted. All three volatility curves hover around 0.05.

Question 3 (20 points)

The built-in data set called Orange in R is about the growth of orange trees. The Orange data frame has 3 columns of records of the growth of orange trees.

```r
#shows dataframe of Orange
pander(head(Orange), caption = "Orange Trees Types by Age and Circumference")
```

Table 1: Orange Trees Types by Age and Circumference

Tree	age	circumference
1	118	30
1	484	58
1	664	87
1	1004	115
1	1231	120
1	1372	142

Variable description

Tree: an ordered factor indicating the tree on which the measurement is made. The ordering is according to increasing maximum diameter.

age: a numeric vector giving the age of the tree (days since 1968/12/31)

circumference: a numeric vector of trunk circumferences (mm). This is probably "circumference at breast height", a standard measurement in forestry.

a) Calculate the mean and the median of the trunk circumferences for different size of the trees. (Tree)

```
#the dplyr package was used to summarise the mean and circumference of the Orange trees by type
pander(ddply(Orange, .(Tree), summarize,  MeanCircumference=mean(circumference),
             MedianCircumference=median(circumference)),
       caption = "Mean and Median of Trunk Circumferences for Different Size of Orange Trees")
```

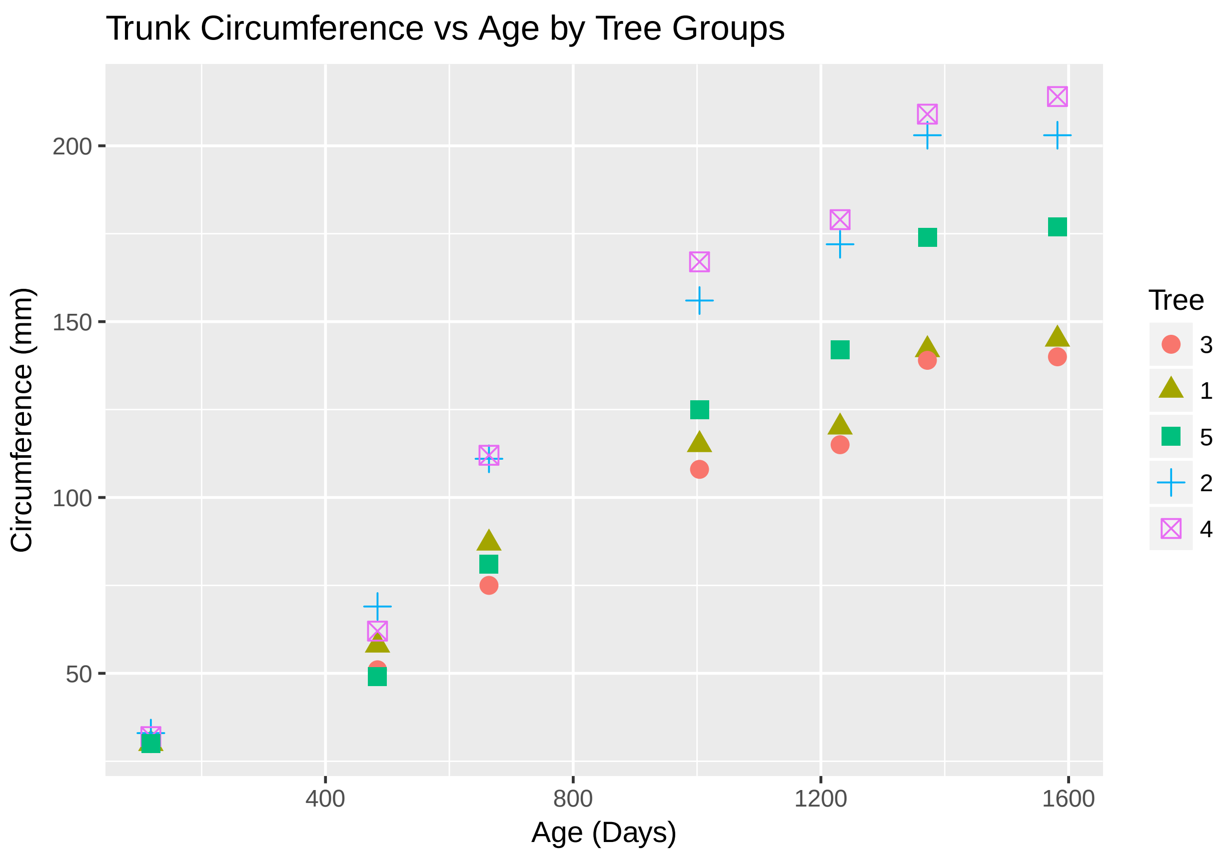
Table 2: Mean and Median of Trunk Circumferences for Different Size of Orange Trees

Tree	MeanCircumference	MedianCircumference
3	94	108
1	99.57	115
5	111.1	125
2	135.3	156
4	139.3	167

The mean and median circumferences are calculated above for each of the five orange tree types. Tree 3 had the smallest mean and median circumferences and Tree 4 had the largest mean and median circumferences.

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

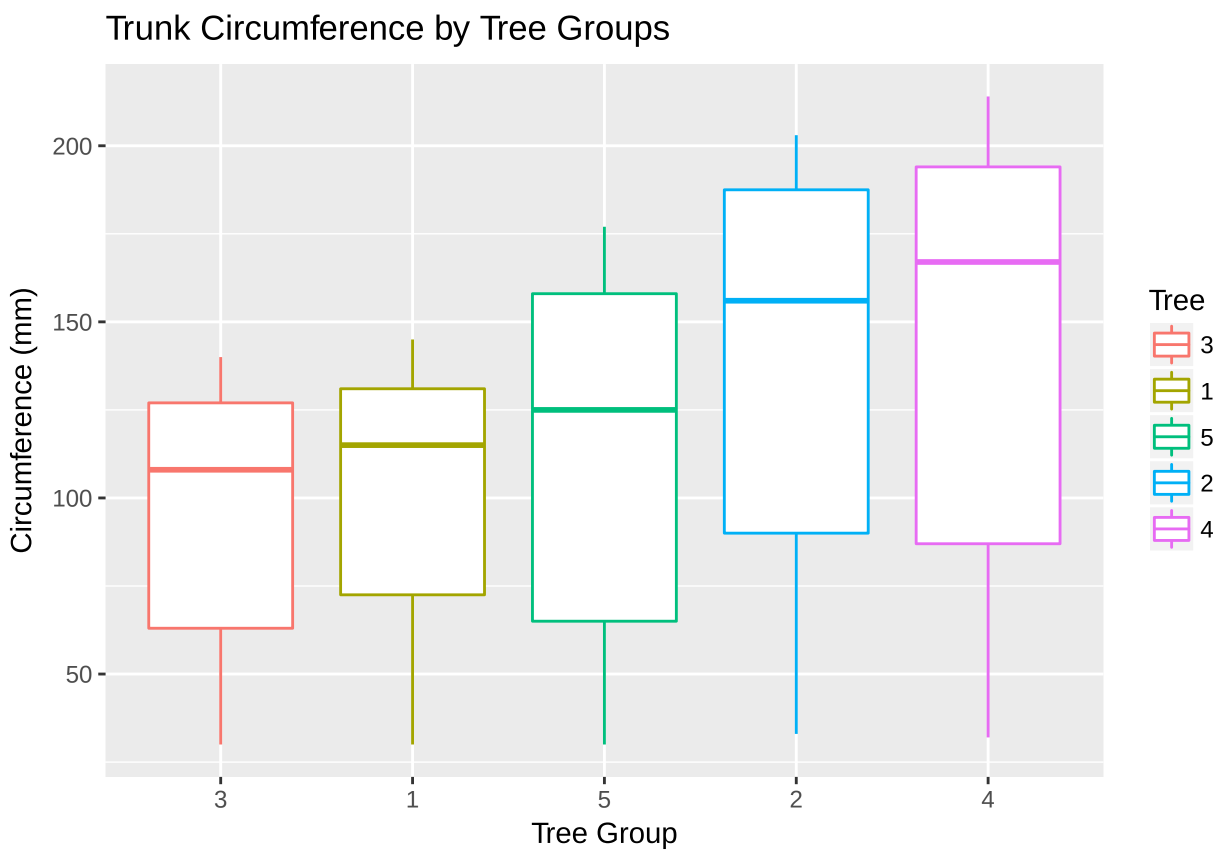
```r
#ggplot was used to plot circumference vs age by tree type, with the legend having different colors
#and symbols for orange tree types. Geom_point was used for the scatterplot
ggplot(data=Orange,aes(x=age,y=circumference,group=Tree))+
  geom_point(aes(shape=Tree, color=Tree), size = 3)+
  labs(x="Age (Days)",y="Circumference (mm)",title="Trunk Circumference vs Age by Tree Groups")
```



When the five orange tree circumferences are plotted against their age, the circumference for younger trees are all about the same while the circumference for older trees have a larger deviation.

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

```
#ggplot was used to plot the distribution of circumference by tree type, with each boxplot having
#different colors for tree types. Geom_boxplot was used for the box plots by ascending max diameter
ggplot(data=Orange,aes(x=Tree,y=circumference,group=Tree))+
  geom_boxplot(aes(shape=Tree, color=Tree))+
  labs(x="Tree Group",y="Circumference (mm)",title="Trunk Circumference by Tree Groups")
```



The median circumferences are calculated above for each of the five orange tree types by max circumference. Tree 3 had the smallest median circumferences and Tree 4 had the largest median circumferences. This agrees with the table from part a.

Observations of the Orange data demonstrate that all 5 trees started out at roughly the same circumference at the first measurement marked by 118 days since 12/31/1968. The circumferences of the 5 trees at this first observational time vary by only 2 mm. At each time interval, the variance in circumference steadily increases until it reaches a maximum of 74 mm at the last measurement of 1,582 days since 12/31/1968. The reasons for the difference of circumferences between trees at each time interval cannot be explained as there is no treatment given in the data set. Causes ranging from environmental conditions, diseases contracted throughout the life of the tree, genetic differences, and many others are all possible causes for the varying tree circumferences, but without any indication of a controlled treatment in the given data set, no conclusions can be made about this data.

Question 4 (45 points)

Download "Temp" data set (check your SMU email)

```
#Retrieving the country temperature files
Temperature <- read.csv("C:\\Users\\Yao\\Documents\\GitHub\\DDS-Case-Study-2\\Data\\TEMP.csv",
                        stringsAsFactors = FALSE)
str(Temperature)
```

```
## 'data.frame':    574223 obs. of  4 variables:
##  $ Date                         : chr  "1838-04-01" "1838-05-01" "1838-06-01" "1838-07-01" ...
##  $ Monthly.AverageTemp          : num  13 NA 23.9 26.9 24.9 ...
##  $ Monthly.AverageTemp.Uncertainty: num  2.59 NA 2.51 2.88 2.99 ...
##  $ Country                      : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
```

There are 574k observations in Temperature and the date column needs to changed to date type. Dates prior to 1900 were formatted as YYYY-MM-DD, while dates after 1900 were formatted as MM/DD/YYYY. We are interested in the dates after 1900; therefore, the import format is %m/%d/%Y.

```
#formatting the date files to extract the data 1900 and after
Temperature$Date <- as.Date(Temperature$Date, format="%m/%d/%Y")
Temperature2<-Temperature[rowSums(is.na(Temperature[,1:4]))==FALSE,]
str(Temperature2)
```

```
## 'data.frame':    327454 obs. of  4 variables:
##  $ Date                         : Date, format: "1900-01-01" "1900-02-01" ...
##  $ Monthly.AverageTemp          : num  -3.43 1.23 10.54 13.35 20.26 ...
##  $ Monthly.AverageTemp.Uncertainty: num  0.936 1.135 0.933 0.536 0.524 ...
##  $ Country                      : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
```

```
row.names(Temperature2) <- NULL
write.csv(Temperature2, "Temperature2.csv")
pander(head(Temperature2), caption = "Monthy Temperatures of Countries Cleaned")
```

Table 3: Monthy Temperatures of Countries Cleaned

Date	Monthly.AverageTemp	Monthly.AverageTemp.Uncertainty	Country
1900-01-01	-3.428	0.936	Afghanistan
1900-02-01	1.234	1.135	Afghanistan
1900-03-01	10.54	0.933	Afghanistan
1900-04-01	13.35	0.536	Afghanistan
1900-05-01	20.26	0.524	Afghanistan
1900-06-01	24.45	0.944	Afghanistan

(i) Find the difference between the maximum and the minimum monthly average temperatures for each country and report/visualize top 20 countries with the maximum differences for the period since 1900.

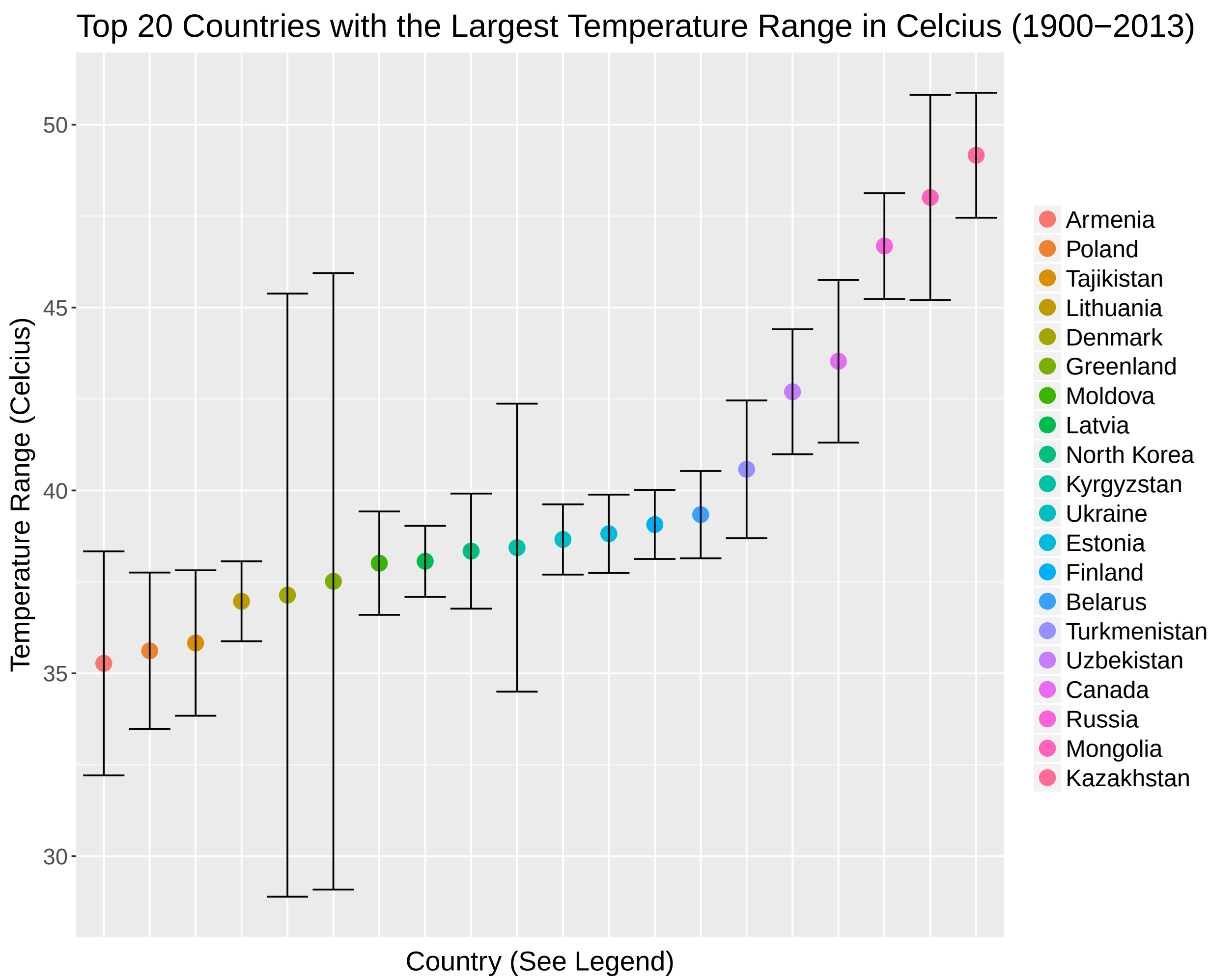
```
#In order to find the range, a function was used in the aggregate function to retrieve temp range and
#temp range uncertainty. The full file is written as DescRangeStdevTemp1900.csv with the top 20 shown
Temp1900 <- subset(Temperature2, Date >= as.Date("1900-01-01"))
write.csv(Temp1900, "Temp1900.csv")
RangeTemp1900 <- aggregate(Monthly.AverageTemp ~ Country, Temp1900, FUN = function(i)max(i) - min(i))
RangeTemp1900Stdev <- aggregate(Monthly.AverageTemp.Uncertainty ~ Country, Temp1900, max)
RangeStdevTemp1900 <- merge(y = RangeTemp1900Stdev, x = RangeTemp1900, by ='Country', all=TRUE)
DescRangeStdevTemp1900 <- RangeStdevTemp1900[order(-RangeStdevTemp1900$Monthly.AverageTemp),]
row.names(DescRangeStdevTemp1900) <- seq(length=nrow(DescRangeStdevTemp1900))
DescRangeStdevTemp1900 <- setnames(DescRangeStdevTemp1900,
                                   old = c('Monthly.AverageTemp','Monthly.AverageTemp.Uncertainty'),
                                   new = c('TempRange','TempRange.Uncertainty'))
write.csv(DescRangeStdevTemp1900, "DescRangeStdevTemp1900.csv")
TopDescRangeStdevTemp1900 <- DescRangeStdevTemp1900[1:20,]
pander(TopDescRangeStdevTemp1900,
       caption = "Top 20 Countries with the Largest Temperature Range (1900-2013)")
```

Table 4: Top 20 Countries with the Largest Temperature Range (1900-2013)

Country	TempRange	TempRange.Uncertainty
Kazakhstan	49.16	1.709
Mongolia	48.01	2.804
Russia	46.68	1.446
Canada	43.53	2.222
Uzbekistan	42.7	1.708
Turkmenistan	40.58	1.882
Belarus	39.34	1.192
Finland	39.07	0.941
Estonia	38.81	1.071
Ukraine	38.66	0.96
Kyrgyzstan	38.44	3.936
North Korea	38.34	1.573
Latvia	38.06	0.969
Moldova	38.01	1.413
Greenland	37.52	8.425
Denmark	37.14	8.243
Lithuania	36.97	1.093
Tajikistan	35.83	1.988
Poland	35.62	2.14
Armenia	35.27	3.063

The top 20 countries with the largest range of temperatures since 1900 are located in the northern hemisphere with Canada, Russia, Mongolia, and Kazakhstan. This could be caused by better reporting for some countries over others, with some countries having larger uncertainty measurements than that of others.

```r
#The top 20 countries with the max temperature ranges are graphed in order of ascending range
#by rainbow color, each with error bars for uncertainty. Use legend for identification
ggplot(data=TopDescRangeStdevTemp1900,aes(x=reorder(Country, TempRange),y=TempRange, group = Country))+
  geom_point(aes(color=reorder(Country, TempRange)), size = 4)+
  labs(x="Country (See Legend)",y="Temperature Range (Celcius)",
       title="Top 20 Countries with the Largest Temperature Range in Celcius (1900-2013)")+
  theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  geom_errorbar(aes(ymin = TempRange-TempRange.Uncertainty,
                    ymax = TempRange+TempRange.Uncertainty))+
  theme(legend.title=element_blank(), legend.text=element_text(size=14), text = element_text(size=16))
```



The graphical representation of the table shows a better way to visualize the data of the top temperature ranges in ascending order by country.

Denmark owns Greenland; thus, they have the same error bars despite different climate for those two territories. It might be governmental that the error bars be that large for those 2 countries.

The plot that displays the top 20 countries with the largest temperature ranges does not show many surprises. Many of the top 20 countries reside in the notoriously harsh climates of Mongolia the Russian Steppes and the Stan's. Many of these countries have very little of their land masses exposed to large bodies of water which contributes to regulate land temperatures. Also of note is that none of the top 20 countries reside near the equator. Land masses near the equator typically do not display large variation in temperature over the course of a year.

(ii) Select a subset of data called "UStemp" where US land temperatures from 01/01/1990 in Temp data. Use UStemp dataset to answer the followings.

```r
#The subset where country is United States and date starts at 1990 is created
UStemp <- subset(Temp1900, Country == "United States" & Date >= as.Date("1990-01-01"))
row.names(UStemp) <- NULL
str(UStemp)
```

```
## 'data.frame':    285 obs. of  4 variables:
##  $ Date                         : Date, format: "1990-01-01" "1990-02-01" ...
##  $ Monthly.AverageTemp          : num  -1.12 -1.75 4.46 9.38 13.77 ...
##  $ Monthly.AverageTemp.Uncertainty: num  0.195 0.107 0.24 0.08 0.112 0.255 0.175 0.218 0.203 0.159 .
##  $ Country                      : chr  "United States" "United States" "United States" "United Sta
```

The subset of data for UStemp is gathered from Temp1900, where the filtered data only has United States as the country and 1990 as the starting year.

a) Create a new column to display the monthly average land temperatures in Fahrenheit (F).

```r
#Both the temperature and uncertainty are converted from Celcius to Fahrenheit and shown in a table
UStemp$Monthly.AverageTemp.F <- UStemp$Monthly.AverageTemp*1.8 + 32
UStemp$Monthly.AverageTemp.F.Uncertainty <- UStemp$Monthly.AverageTemp.Uncertainty*1.8
write.csv(UStemp, "UStemp.csv")
pander(head(UStemp),
       caption = "United States monthly average land temperatures in Celcius and Fahrenheit (1990 - 2013
```

Table 5: United States monthly average land temperatures in Celcius and Fahrenheit (1990 - 2013) (continued below)

Date	Monthly.AverageTemp	Monthly.AverageTemp.Uncertainty	Country
1990-01-01	-1.123	0.195	United States
1990-02-01	-1.747	0.107	United States
1990-03-01	4.465	0.24	United States
1990-04-01	9.38	0.08	United States
1990-05-01	13.77	0.112	United States
1990-06-01	19.78	0.255	United States

Monthly.AverageTemp.F	Monthly.AverageTemp.F.Uncertainty
29.98	0.351
28.86	0.1926
40.04	0.432
48.88	0.144
56.79	0.2016
67.6	0.459

United States monthly average land temperatures in Celcius and Fahrenheit along with uncertainty calculations from 1990 to 2013.

b) Calculate average land temperature by year and plot it.

The original file has the average land temperature by month.

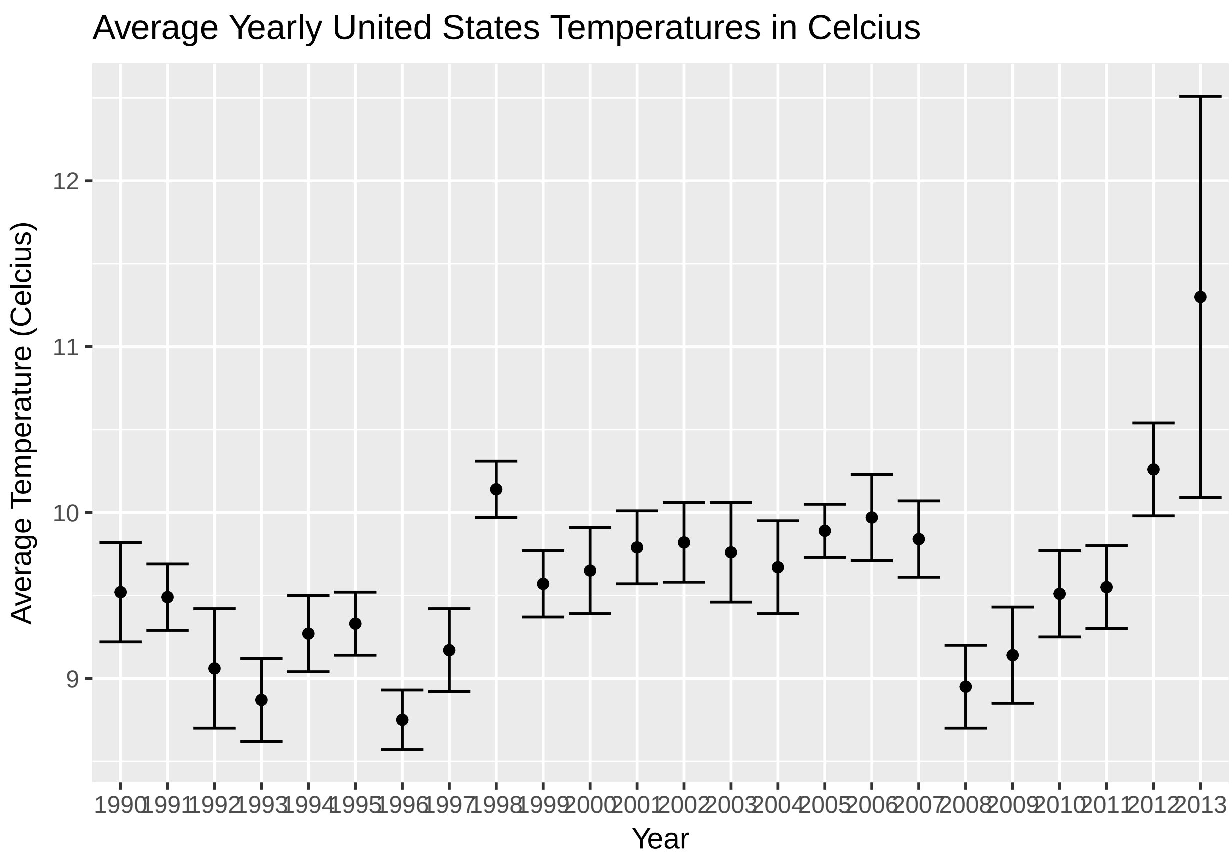
```r
#The table calculations of the average US temperature by year is created
UStemp$Year <- format(UStemp$Date,format="%Y")
UStempYear<- ddply(UStemp, .(Year), summarize, AvgTemp.C=round(mean(Monthly.AverageTemp), digits = 2),
                    AvgTemp.C.Uncertainty=round(max(Monthly.AverageTemp.Uncertainty), digits = 2),
                    AvgTemp.F=round(mean(Monthly.AverageTemp.F), digits = 2),
                    AvgTemp.F.Uncertainty=round(max(Monthly.AverageTemp.F.Uncertainty), digits = 2))
pander(UStempYear, caption = "Average Yearly United States Temperatures in Celcius and Fahrenheit")
```

Table 7: Average Yearly United States Temperatures in Celcius and Fahrenheit

Year	AvgTemp.C	AvgTemp.C.Uncertainty	AvgTemp.F	AvgTemp.F.Uncertainty
1990	9.52	0.3	49.14	0.54
1991	9.49	0.2	49.09	0.36
1992	9.06	0.36	48.3	0.64
1993	8.87	0.25	47.96	0.44
1994	9.27	0.23	48.69	0.41
1995	9.33	0.19	48.8	0.34
1996	8.75	0.18	47.76	0.33
1997	9.17	0.25	48.51	0.44
1998	10.14	0.17	50.25	0.31
1999	9.57	0.2	49.22	0.37
2000	9.65	0.26	49.37	0.47
2001	9.79	0.22	49.61	0.39
2002	9.82	0.24	49.67	0.43
2003	9.76	0.3	49.56	0.54
2004	9.67	0.28	49.4	0.5
2005	9.89	0.16	49.81	0.3
2006	9.97	0.26	49.95	0.47
2007	9.84	0.23	49.71	0.41
2008	8.95	0.25	48.11	0.45
2009	9.14	0.29	48.45	0.52
2010	9.51	0.26	49.11	0.46
2011	9.55	0.25	49.19	0.46
2012	10.26	0.28	50.47	0.5
2013	11.3	1.21	52.33	2.18

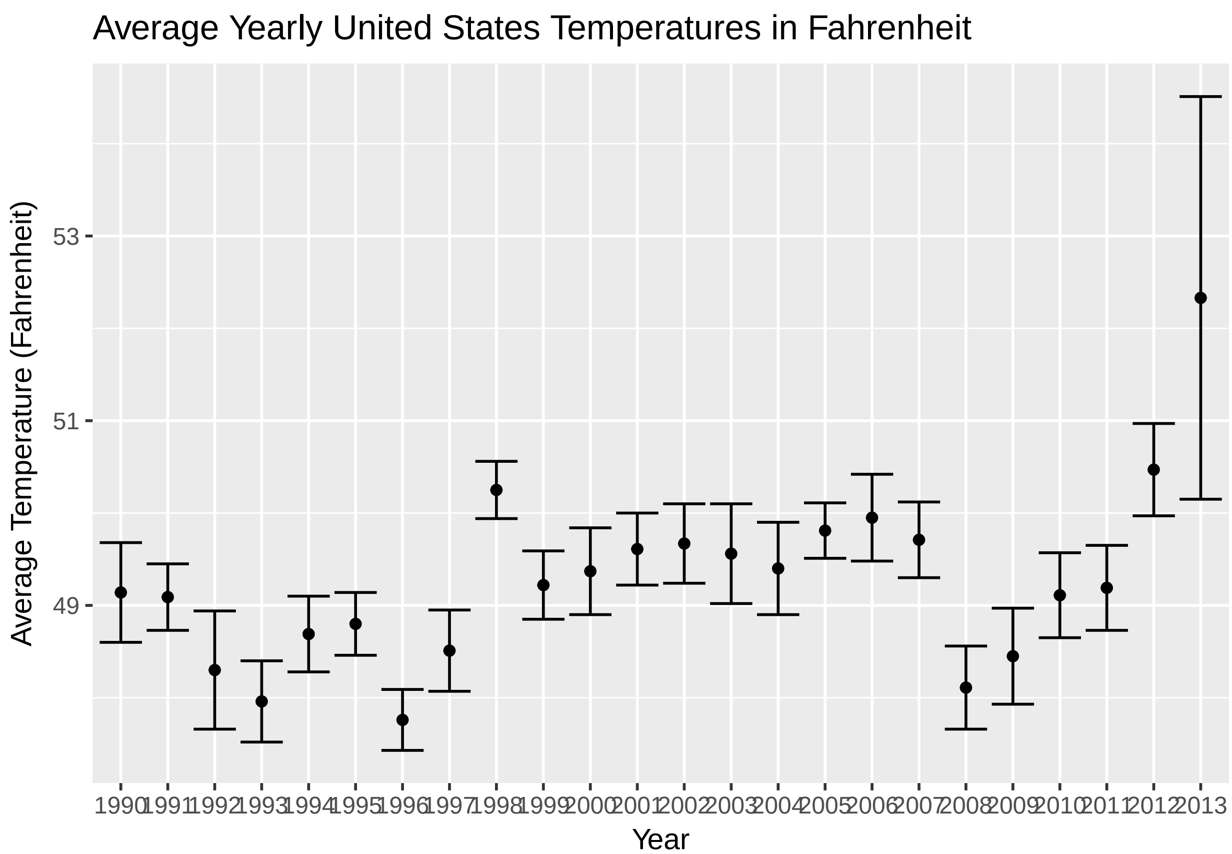
The average US temperature by year in celcius and fahrenheit along with uncertainty in reporting.

```r
#The graphical representation of average yearly US temperature in Celius with error bars
ggplot(data=UStempYear,x=Year,y=AvgTemp.C)+
  geom_point(aes(x=Year,y=AvgTemp.C))+
  labs(x="Year",y="Average Temperature (Celcius)",
       title="Average Yearly United States Temperatures in Celcius")+
  geom_errorbar(aes(x=Year, ymin = AvgTemp.C-AvgTemp.C.Uncertainty,
                    ymax = AvgTemp.C+AvgTemp.C.Uncertainty))
```



Average yearly United States temperatures in Celcius. There are bigger errors at year 2013. The temperature is possibly increasing due to global warming.

```r
#The graphical representation of average yearly US temperature in Fahrenheit with error bars
ggplot(data=UStempYear,x=Year,y=AvgTemp.F)+
  geom_point(aes(x=Year,y=AvgTemp.F))+
  labs(x="Year",y="Average Temperature (Fahrenheit)",
       title="Average Yearly United States Temperatures in Fahrenheit")+
  geom_errorbar(aes(x=Year, ymin = AvgTemp.F-AvgTemp.F.Uncertainty,
                    ymax = AvgTemp.F+AvgTemp.F.Uncertainty))
```



Average yearly United States temperatures in Fahrenheit. This matches the shape of the Celcius graph exactly on a different temperature scale.

c) Calculate the one year difference of average land temperature by year and provide the maximum difference (value) with corresponding two years.

(for example, year 2000: add all 12 monthly averages and divide by 12 to get average temperature in 2000. You can do the same thing for all the available years. Then you can calculate the one year difference as 1991-1990, 1992-1991, etc)

```
#The difference of yearly average temperatures for the US in F and C
UStempYear$AvgTemp.C.Diff<- c(NA, round(diff(UStempYear$AvgTemp.C), digits = 2))
UStempYear$AvgTemp.F.Diff<- c(NA, round(diff(UStempYear$AvgTemp.F), digits = 2))
pander(UStempYear[,c(1,2,4,6,7)],
       caption = "Difference in Yearly Average United States Temperatures in Celcius and Fahrenheit")
```

Table 8: Difference in Yearly Average United States Temperatures in Celcius and Fahrenheit

Year	AvgTemp.C	AvgTemp.F	AvgTemp.C.Diff	AvgTemp.F.Diff
1990	9.52	49.14	NA	NA
1991	9.49	49.09	-0.03	-0.05
1992	9.06	48.3	-0.43	-0.79
1993	8.87	47.96	-0.19	-0.34
1994	9.27	48.69	0.4	0.73
1995	9.33	48.8	0.06	0.11
1996	8.75	47.76	-0.58	-1.04
1997	9.17	48.51	0.42	0.75
1998	10.14	50.25	0.97	1.74
1999	9.57	49.22	-0.57	-1.03
2000	9.65	49.37	0.08	0.15
2001	9.79	49.61	0.14	0.24
2002	9.82	49.67	0.03	0.06
2003	9.76	49.56	-0.06	-0.11
2004	9.67	49.4	-0.09	-0.16
2005	9.89	49.81	0.22	0.41
2006	9.97	49.95	0.08	0.14
2007	9.84	49.71	-0.13	-0.24
2008	8.95	48.11	-0.89	-1.6
2009	9.14	48.45	0.19	0.34
2010	9.51	49.11	0.37	0.66
2011	9.55	49.19	0.04	0.08
2012	10.26	50.47	0.71	1.28
2013	11.3	52.33	1.04	1.86

The fluctuations of yearly average temperature of the US looks normal until we reach 2013, where there was a large increase in yearly average temperature.

(iii) Download "CityTemp" data set (check your SMU email). 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.

```
#Retrieveal of the citytemp file
CityTemp <- read.csv("C:\\Users\\Yao\\Documents\\GitHub\\DDS-Case-Study-2\\Data\\CityTemp.csv",
                 row.names = NULL,
                 stringsAsFactors = FALSE)
str(CityTemp)
```

```
## 'data.frame':    237200 obs. of  7 variables:
##  $ Date                            : chr  "1850-01-01" "1850-02-01" "1850-03-01" "1850-04-01" ...
##  $ Monthly.AverageTemp             : num  16 18.3 18.6 18.2 17.5 ...
##  $ Monthly.AverageTemp.Uncertainty: num  1.54 1.53 2.16 1.69 1.24 ...
##  $ City                            : chr  "Addis Abeba" "Addis Abeba" "Addis Abeba" "Addis Abeba" ...
##  $ Country                         : chr  "Ethiopia" "Ethiopia" "Ethiopia" "Ethiopia" ...
##  $ Latitude                        : chr  "8.84N" "8.84N" "8.84N" "8.84N" ...
##  $ Longitude                       : chr  "38.11E" "38.11E" "38.11E" "38.11E" ...
```

There are 237k observations for citytemp and the date column needs to changed to date type. Dates prior to 1900 were formatted as YYYY-MM-DD, while dates after 1900 were formatted as MM/DD/YYYY. We are interested in the dates after 1900; therefore, the import format is %m/%d/%Y.

```
#formatting the date files to extract the data 1900 and after
CityTemp$Date <- as.Date(CityTemp$Date, format="%m/%d/%Y")
CityTemp2<-CityTemp[rowSums(is.na(CityTemp[,1:5]))==FALSE,]
CityTemp2<-CityTemp2[,1:5]
CityTemp1900 <- subset(CityTemp2, Date >= as.Date("1900-01-01"))
CityTemp1900$CityCountry <- paste(CityTemp1900$City,",",CityTemp1900$Country)
row.names(CityTemp1900) <- NULL
write.csv(CityTemp1900, "CityTemp1900.csv")
RangeCityTemp1900 <- aggregate(Monthly.AverageTemp ~ CityCountry, CityTemp1900,
                               FUN = function(i)max(i) - min(i))
RangeCityTemp1900Stdev <- aggregate(Monthly.AverageTemp.Uncertainty ~ CityCountry,
                                    CityTemp1900, max)
RangeCityStdevTemp1900 <- merge(y = RangeCityTemp1900Stdev, x = RangeCityTemp1900,
                                by ='CityCountry', all=TRUE)
RangeCityStdevTemp1900 <- setnames(RangeCityStdevTemp1900,
  old = c('Monthly.AverageTemp','Monthly.AverageTemp.Uncertainty'),
  new = c('TempRange','TempRange.Uncertainty'))

DescCityRangeStdevTemp1900 <- RangeCityStdevTemp1900[order(-RangeCityStdevTemp1900$TempRange),]
row.names(DescCityRangeStdevTemp1900) <- seq(length=nrow(DescCityRangeStdevTemp1900))
write.csv(DescCityRangeStdevTemp1900, "DescCityRangeStdevTemp1900.csv")
```

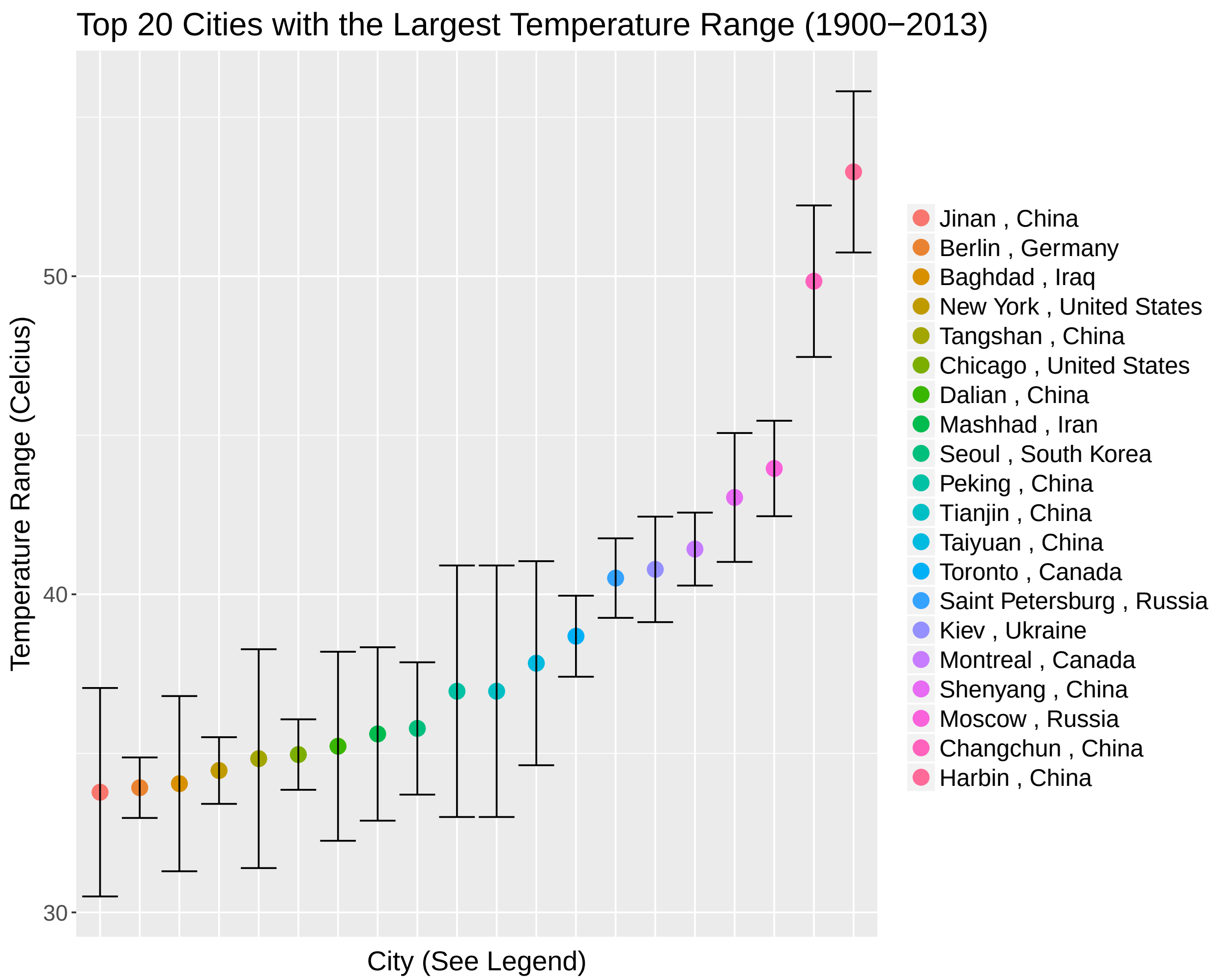
```r
#In order to find temp range, a function was used in the aggregate function to retrieve temp range
#and temp range uncertainty. The full file is in DescCityRangeStdevTemp1900.csv with the top 20 shown
TopDescCityRangeStdevTemp1900 <- DescCityRangeStdevTemp1900[1:20,]
pander(TopDescCityRangeStdevTemp1900,
       caption = "Top 20 Cities with the Largest Temperature Range (1900-2013)")
```

Table 9: Top 20 Cities with the Largest Temperature Range (1900-2013)

CityCountry	TempRange	TempRange.Uncertainty
Harbin , China	53.28	2.534
Changchun , China	49.84	2.382
Moscow , Russia	43.96	1.501
Shenyang , China	43.05	2.026
Montreal , Canada	41.42	1.147
Kiev , Ukraine	40.78	1.658
Saint Petersburg , Russia	40.51	1.25
Toronto , Canada	38.68	1.274
Taiyuan , China	37.83	3.208
Peking , China	36.95	3.954
Tianjin , China	36.95	3.954
Seoul , South Korea	35.78	2.08
Mashhad , Iran	35.61	2.726
Dalian , China	35.22	2.972
Chicago , United States	34.96	1.108
Tangshan , China	34.83	3.44
New York , United States	34.46	1.048
Baghdad , Iraq	34.05	2.754
Berlin , Germany	33.92	0.951
Jinan , China	33.78	3.277

The top 20 cities with the largest range of temperatures since 1900 are located in China, Russia, and the United States. This could be caused by better reporting for cities in those countries than that in other countries.

```r
#The top 20 cities with the max temperature ranges are graphed in order of ascending range
#by rainbow color, each with error bars for uncertainty. Use legend for identification
ggplot(data=TopDescCityRangeStdevTemp1900,aes(x=reorder(CityCountry, TempRange),
                                              y=TempRange, group = CityCountry))+
  geom_point(aes(color=reorder(CityCountry, TempRange)), size = 4)+
  labs(x="City (See Legend)",y="Temperature Range (Celcius)",
       title="Top 20 Cities with the Largest Temperature Range (1900-2013)")+
  theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  geom_errorbar(aes(ymin = TempRange-TempRange.Uncertainty,
                    ymax = TempRange+TempRange.Uncertainty))+
  theme(legend.title=element_blank(), legend.text=element_text(size=14), text = element_text(size=16))
```



The graphical representation of the table shows a better way to visualize the data of the top temperature ranges in ascending order by city.

Some cities having larger uncertainty measurements than that of others and Chineses cities dominate this plot, possibly due to better reporting in those cities.

(iv) Compare the two graphs in (i) and (iii) and comment it.

```
#The top city ranges are converted into country ranges from part (iii) and then compared with
#the top country ranges from part (i) by tables
RangeCCTemp1900 <- aggregate(Monthly.AverageTemp ~ Country, CityTemp1900, FUN = function(i)max(i) - min
RangeCCStdevTemp1900 <- aggregate(Monthly.AverageTemp.Uncertainty ~ Country, CityTemp1900, max)
RangeCCStdevTemp1900 <- merge(y = RangeCCStdevTemp1900, x = RangeCCTemp1900, by ='Country', all=TRUE)
colnames(RangeCCStdevTemp1900) <- c("Country", "AvgTempRange.ByCity", "StdevTempRange.ByCity")
DescRangeCCStdevTemp1900 <- RangeCCStdevTemp1900[order(-RangeCCStdevTemp1900$AvgTempRange.ByCity),]
row.names(DescRangeCCStdevTemp1900) <- seq(length=nrow(DescRangeCCStdevTemp1900))
pander(DescRangeCCStdevTemp1900[1:20,],
       caption = "Top 20 Countries with the Largest Temperature Range from City Data (1900-2013)")
```

Table 10: Top 20 Countries with the Largest Temperature Range from City Data (1900-2013)

Country	AvgTempRange.ByCity	StdevTempRange.ByCity
China	58	4.706
Russia	43.96	1.501
Canada	41.54	1.274
Ukraine	40.78	1.658
United States	36.48	1.524
South Korea	35.78	2.08
Iran	35.61	2.726
Turkey	35.15	1.828
Iraq	34.05	2.754
Germany	33.92	0.951
Syria	31.54	1.708
Japan	30.19	1.12
Afghanistan	29.66	2.582
Italy	27.39	1.803
Saudi Arabia	27.36	4.399
France	27.14	1.078
Pakistan	27	2.577
Spain	24.71	1.829
India	24.63	2.195
United Kingdom	23.2	0.801

From city temperature data converted into country temperature range (iii), the top 3 countries with max temperature ranges are China, Russia, and Canada. This could be caused by better reporting for some cities over others, with some cities having larger uncertainty measurements than that of others, which caused the whole country to have a larger temperature range.

```
colnames(DescRangeStdevTemp1900) <- c("Country", "AvgTempRange.ByCountry", "StdevTempRange.ByCountry")
row.names(DescRangeStdevTemp1900) <- seq(length=nrow(DescRangeStdevTemp1900))
pander(DescRangeStdevTemp1900[1:20,],
       caption = "Top 20 Countries with the Largest Temperature Range from Country Data (1900-2013)")
```

Table 11: Top 20 Countries with the Largest Temperature Range from Country Data (1900-2013)

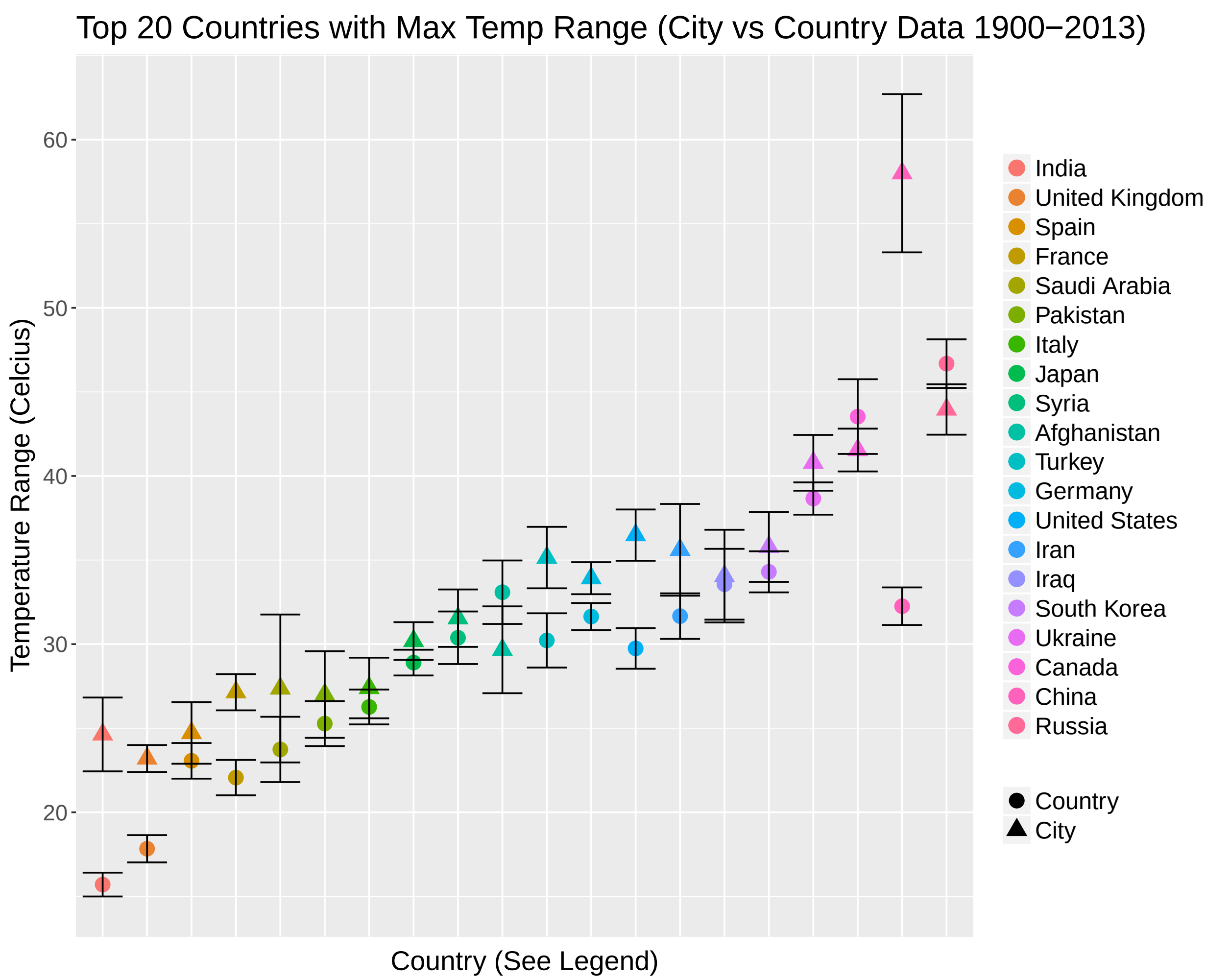
Country	AvgTempRange.ByCountry	StdevTempRange.ByCountry
Kazakhstan	49.16	1.709
Mongolia	48.01	2.804
Russia	46.68	1.446
Canada	43.53	2.222
Uzbekistan	42.7	1.708
Turkmenistan	40.58	1.882
Belarus	39.34	1.192
Finland	39.07	0.941
Estonia	38.81	1.071
Ukraine	38.66	0.96
Kyrgyzstan	38.44	3.936
North Korea	38.34	1.573
Latvia	38.06	0.969
Moldova	38.01	1.413
Greenland	37.52	8.425
Denmark	37.14	8.243
Lithuania	36.97	1.093
Tajikistan	35.83	1.988
Poland	35.62	2.14
Armenia	35.27	3.063

From country temperature data (i), the top 3 countries with max temperature ranges are Kazakhstan, Mongolia, and Russia. This could be caused by better reporting for some countries over others, with some countries having larger uncertainty measurements than that of others.

```r
#The countries from (i) are then paired with matching countries from city data from part (iii) to do a
#comparison of the country ranges
CompareTemp1900 <- merge(y = DescRangeCCStdevTemp1900, x = DescRangeStdevTemp1900,
                         by ='Country', all=TRUE)
CompareMTemp1900<-CompareTemp1900[rowSums(is.na(CompareTemp1900[,1:4]))==FALSE,]
CITYcompareMtemp1900 <- CompareMTemp1900[order(-CompareMTemp1900$AvgTempRange.ByCity),]
TopCITYcompareMtemp1900 <- CITYcompareMtemp1900[1:20,]
#The data are then bound together with a new column that describes city data from country data for plot
TopCITYcity <- subset(TopCITYcompareMtemp1900,
                      select = c("Country","AvgTempRange.ByCity", "StdevTempRange.ByCity"))
TopCITYcity2 <- cbind(TopCITYcity, Type="City")
TopCITYcity3 <- setnames(TopCITYcity2, old = c('AvgTempRange.ByCity','StdevTempRange.ByCity'),
                         new = c('AvgTempRange','StdevTempRange'))

TopCITYcountry <- subset(TopCITYcompareMtemp1900,
                         select = c("Country","AvgTempRange.ByCountry", "StdevTempRange.ByCountry"))
TopCITYcountry2 <- cbind(TopCITYcountry, Type="Country")
TopCITYcountry3 <- setnames(TopCITYcountry2, old = c('AvgTempRange.ByCountry','StdevTempRange.ByCountry
                            new = c('AvgTempRange','StdevTempRange'))
TopCITY2 <- rbind(TopCITYcountry3,TopCITYcity3)
```

```
#Both the city and country temperature max ranges are plotted on the same graph, with cities
#in the same country converted into country data for comparison
ggplot(data=TopCITY2,aes(x=reorder(Country, AvgTempRange),y=AvgTempRange, group = Type))+
  geom_point(aes(color=reorder(Country, AvgTempRange), shape=Type), size = 4)+
  labs(x="Country (See Legend)",y="Temperature Range (Celcius)",
       title="Top 20 Countries with Max Temp Range (City vs Country Data 1900-2013)")+
  theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  geom_errorbar(aes(ymin = AvgTempRange-StdevTempRange, ymax = AvgTempRange+StdevTempRange))+
  theme(legend.title=element_blank(), legend.text=element_text(size=14), text = element_text(size=16))
```



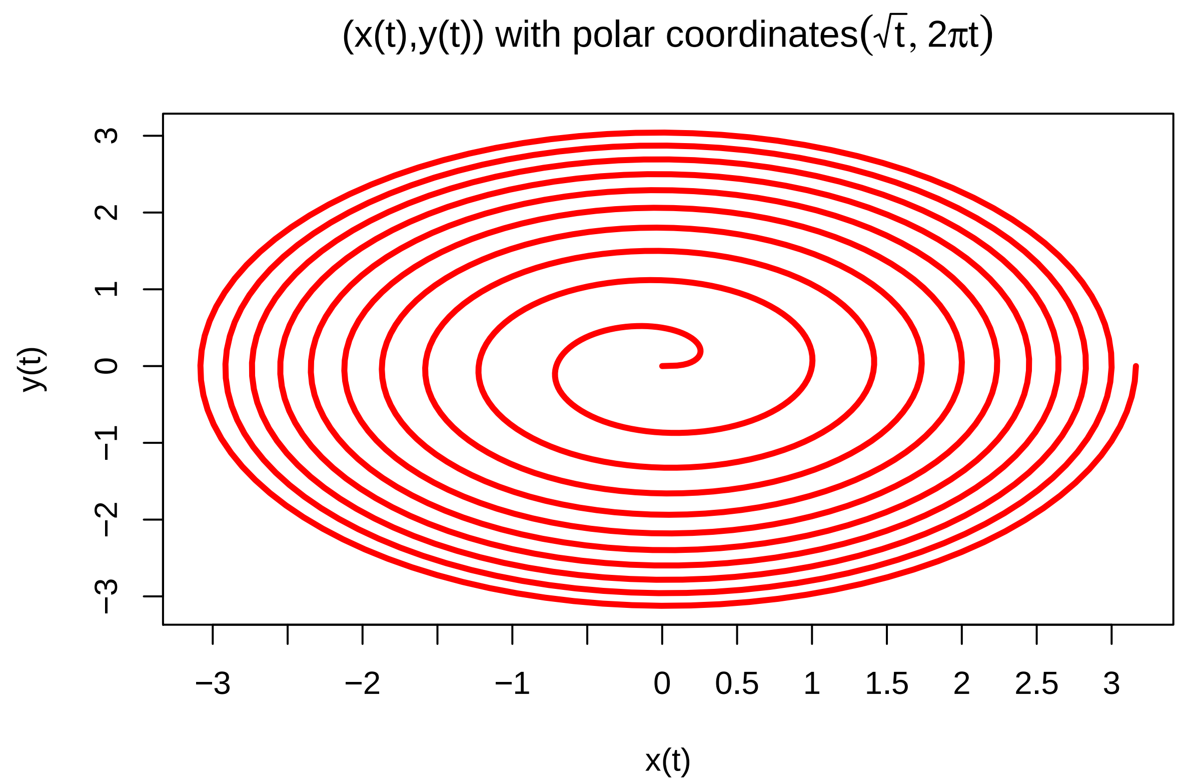
For the scatterplot, there are discrepencies between the city and country data that the uncertainty errors cannot account for when plotted and cross compared with each other. This could be caused by average country temperatures vs average city temperatures, where some of the countries occupy different climate zones as depicted by city temperature ranges to account for the larger country temperature fluctuations and range.

Observations show that when using the finer geographic area of city as opposed to country, many of the areas that were present in the top 20 temperature ranges for countries are absent from the top 20 temperature ranges for cities.

Question 5 (10 points)

Semester End Bonus: Run the following code in R and paste the final graph.

```
#plots sine and cosine of t in polar coordinates
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("(x(t),y(t)) with polar coordinates",(list(sqrt(t),2*pi*t)))))
```



Very cool graph of a swirl when cosine and sine are plotted against each other in polar coordinates.

Conclusion:

2) AGIO is a recent stock that started in 2013 with peak prices in 2015 with its log returns hover slightly above 0 for positive investment. The historical volatitlity was 78.66, with the volatility curves hovering around 0.05.

3)a) The order from smallest to largest mean and median circumferences is tree: 3, 1, 5, 2, 4.

3)b) The circumference for younger trees are about the same while that for older trees have a larger deviation from each other.

3)c) The data had it so that the median circumference had a direct correlation to the maximum circumference per tree type.

4)i) Countries located in the northern hemisphere dominate the top 20 countries with the largest temperature range, possibly due to better reporting in these countries with some governments having larger uncertainty in their measurements than for others.

4)ii)a) When converting temperature uncertainty from Celcius to Fahrenheit, do not add 32 and instead only multiply by 1.8.

4)ii)b) The temperature is possibly increasing in the US due to global warming. The Celcius and Fahrenheit plots have the same shape with a different temperature scale.

4)ii)c) The fluctuations of yearly average temperature of the US looks normal until we reach 2013, where there was a large increase in yearly average temperature.

4)iii) Cities located in China dominate the top 20 cities with the largest temperature range, possibly due to better reporting in these cities. Also, Bagdad made the top 20 list despite it not being too far north in latitude as the other cities.

4)iv) From city temperature data, the top 3 countries with max temperature ranges are China, Russia, and Canada. From country temperature data, the top 3 countries with max temperature ranges are Kazakhstan, Mongolia, and Russia. There are discrepencies between the city and country data that the uncertainty errors cannot account for when plotted and cross compared with each other. This could be caused by average country temperatures vs average city temperatures, where some of the countries occupy different climate zones as depicted by city temperature ranges to account for the larger country temperature fluctuations and range.

5) The capabilities of R to produce graphs in different coordinates makes it very useful.

Further Work:

Q2: We can examine the AGIO against the S&P 500 to see if the stock better or worse than the average top 500 stocks. We can explore the other features of time series for stocks and time sensitive data.

Q3: We can do summary reports and graphical analysis for all the built-in data sets for R.

Q4: We could increase the data set to include all the dates prior to 1900 to plot how the trend is global warming is affecting the temperature trends of every country and city. We can plot the trend of global warming and how that increases or reduces temperature range of countries and cities over time. We can use the temperature trend up to 2013 to predict how the average temperature and temperature range is going to increase for this year by comparing them with the actual data.

Q5: We could use different coordinate systems to plot more graphs in 2D and 3D.