Question 1)

#SAS Code

proc iml;

x = {4 5 1 2, 1 1 4 5, 2 1 0 2};

create num_matrix from x;

append from x;

close num_matrix;

run;

proc print data = num_matrix; run;

Obs	COL1	COL2	COL3	COL4
1	4	5	1	2
2	1	1	4	5
3	2	1	0	2

#R

#cell_1 <- c(4, 5, 1, 2)

#cell_2 <- c(1, 1, 4, 5)

#cell_3 <- c(2, 1, 0, 2)

#my_matrix_2 <- rbind(cell_1, cell_2, cell_3)

cells <- c(4, 5, 1, 2, 1, 1, 4, 5, 2, 1, 0, 2)

my_matrix <- matrix(cells, nrow = 3, ncol = 4, byrow = T)

my_matrix

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

#Python

import numpy as np

x = np.matrix([[4, 5, 1, 2], [1, 1, 4, 5], [2, 1, 0, 2]])

x

matrix([[4, 5, 1, 2],

[1, 1, 4, 5],

[2, 1, 0, 2]])

Question 2)

#The following calculates log returns and volatility measures of time series data on Automatic Data Processing, Inc. (ADP) stock performance using three different decay factors.

library(tseries)

options("getSymbols.warning4.0"=FALSE)

options("getSymbols.yahoo.warning"=FALSE)

ADPdata <- get.hist.quote('ADP', quote = 'Close')

head(ADPdata)

```
                Close
1991-01-02 5.402749
1991-01-03 5.303616
1991-01-04 5.204483
1991-01-07 5.154917
1991-01-08 5.055784
1991-01-09 5.055784
```

##Lagged Log Returns

ADPret <- log(lag(ADPdata)) - log(ADPdata)

head(ADPret)

```
                  Close
1991-01-02 -0.018519046
1991-01-03 -0.018868483
1991-01-04 -0.009569353
1991-01-07 -0.019418082
1991-01-08  0.000000000
1991-01-09 -0.004914163
```

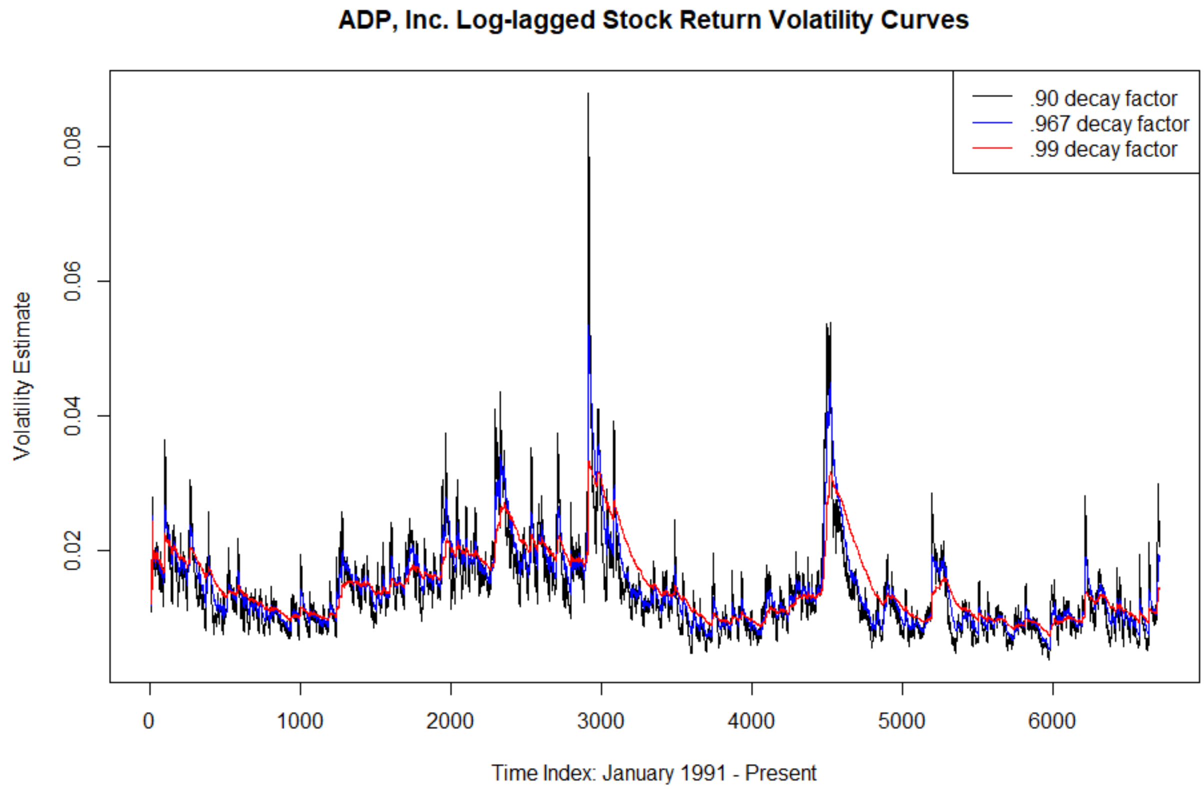
##Calculated Volatility Measure

ADPvol <- sd(ADPret) * sqrt(250) * 100

ADPvol

```
[1] 24.69478
```

```
##Exponentially Downweighted Continuous Lookback Window

Vol <- function(d, logrets){
 var = 0
 lam = 0
 varlist <- c()
 for (r in logrets){
  lam = lam*(1 - 1/d) + 1
  var = (1 - 1/lam)*var + (1/lam)*r^2
  varlist <- c(varlist, var)
 }
 sqrt(varlist)
}

volest <- Vol(10, ADPret)
head(volest)
[1] 0.01851905 0.01870377 0.01595408 0.01703414 0.01480899 0.01332875
volest2 <- Vol(30, ADPret)
head(volest2)
[1] 0.01851905 0.01869754 0.01614490 0.01706619 0.01513238 0.01385249
volest3 <- Vol(100, ADPret)
head(volest3)
[1] 0.01851905 0.01869546 0.01620716 0.01707929 0.01523756 0.01402262

##Plotted Volatility Curves

plot(volest, type = 'l', col = 'black', xlab = 'Time Index: January 1991 - Present', ylab = 'Volatility Estimate', main = 'ADP, Inc. Log-lagged Stock Return Volatility Curves')
lines(volest2, type = 'l', col = 'blue')
lines(volest3, type = 'l', col = 'red')
legend("topright", c('.90 decay factor', '.967 decay factor', '.99 decay factor'), lty=c(1, 1, 1), col = c('black', 'blue', 'red'))
```



#The volatility curve with decay factor 0.99 demonstrates substantially smoother features.

Question 3)

attach(Orange)

Mean_Circ_by_Tree <- aggregate(circumference ~ Tree, data = Orange, FUN = mean)

Mean_Circ_by_Tree

```
  Tree circumference
1    3      94.00000
2    1      99.57143
3    5     111.14286
4    2     135.28571
5    4     139.28571
```

Median_Circ_by_Tree <- aggregate(circumference ~ Tree, data = Orange, FUN = median)

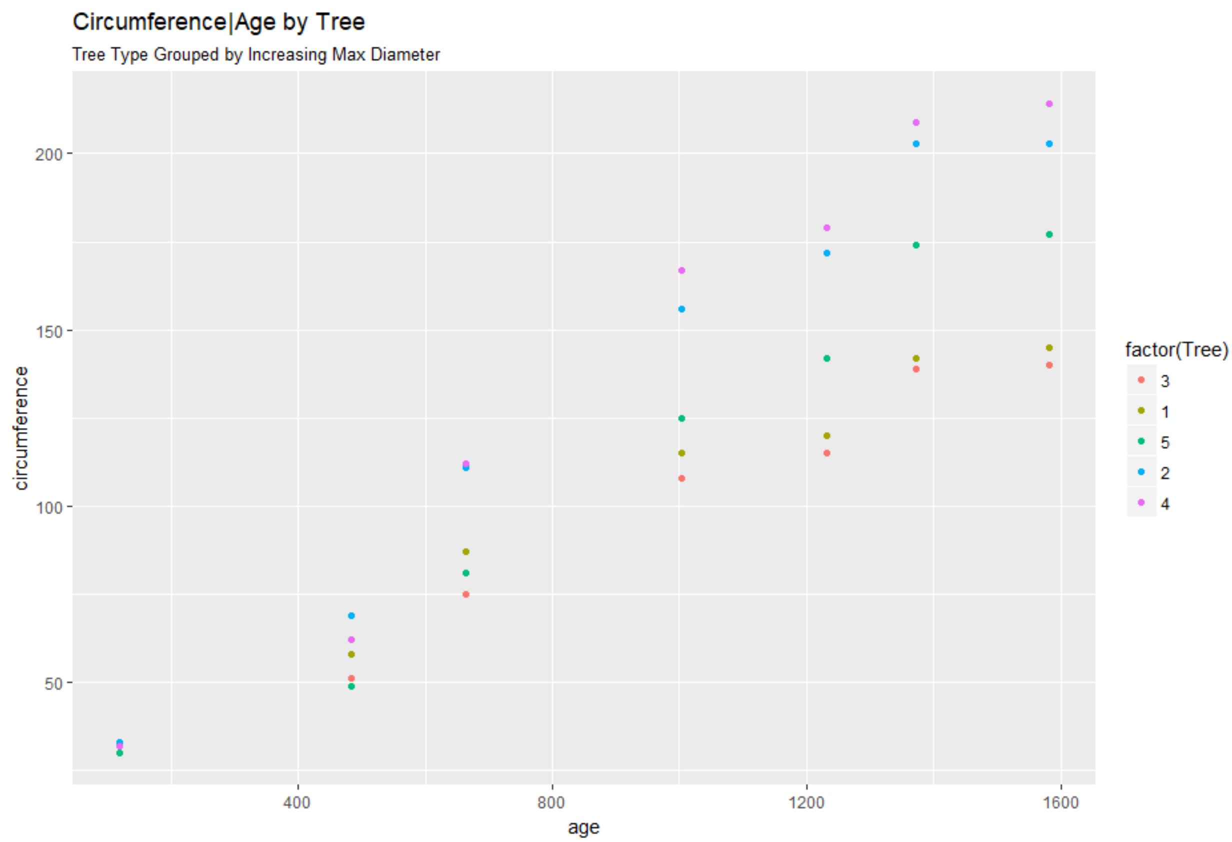
Median_Circ_by_Tree

```
  Tree circumference
1    3           108
2    1           115
3    5           125
4    2           156
5    4           167
```

library(ggplot2)

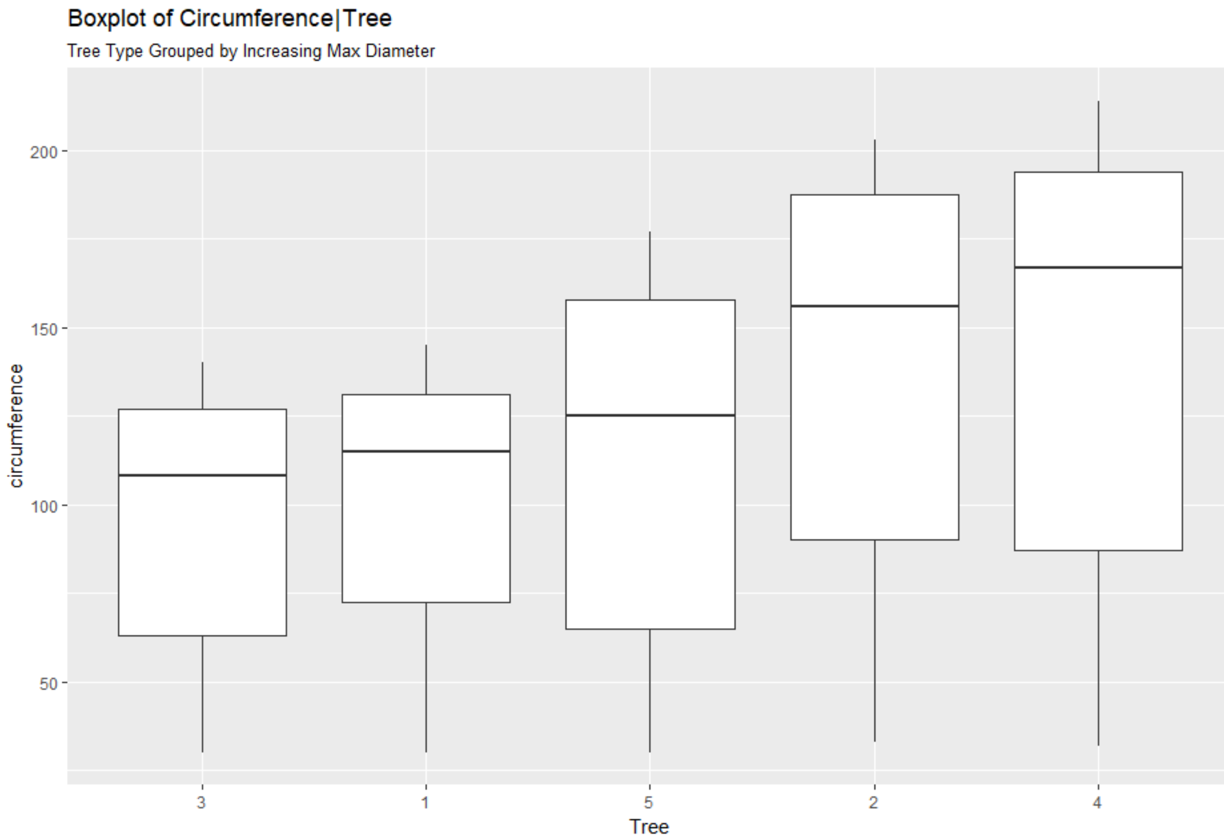
Circ_v_Age_by_Tree <- ggplot(Orange, aes(age, circumference)) + geom_point(aes(color = factor(Tree))) + labs(title = "Circumference|Age by Tree", subtitle = "Tree Type Grouped by Increasing Max Diameter")

Circ_v_Age_by_Tree



Circ_v_Tree_boxplot <- ggplot(Orange, aes(Tree, circumference)) + geom_boxplot() + labs(title = "Boxplot of Circumference|Tree", subtitle = "Tree Type Grouped by Increasing Max Diameter")

Circ_v_Tree_boxplot



Question 4)

#Question 4.I

library(tidyverse)

TEMP <- read_csv("C:/Users/jkras/Desktop/SMU/DDS/Case Study 2/TEMP.csv")

TEMP_2 <- subset(TEMP, grepl("^.+(19|20)",Date))

TEMP_new <- aggregate(TEMP_2$`Monthly AverageTemp`, by = list(TEMP_2$Country), FUN = min, na.rm = T)

TEMP_new2 <- aggregate(TEMP_2$`Monthly AverageTemp`, by = list(TEMP_2$Country), FUN = max, na.rm = T)

TEMP_min <- subset(TEMP_new, select= x)

colnames(TEMP_min) <- "Minimum"

colnames(TEMP_new2) <- c("Country", "Maximum")

TEMP_Difference <- cbind(TEMP_new2, TEMP_min)

TEMP_Difference$Max_Min_Difference <- (TEMP_Difference$Maximum - TEMP_Difference$Minimum)

TEMP_20Max_Differences <- TEMP_Difference[order(TEMP_Difference$Max_Min_Difference, decreasing = T)[1:20],]

TEMP_20Max_Differences

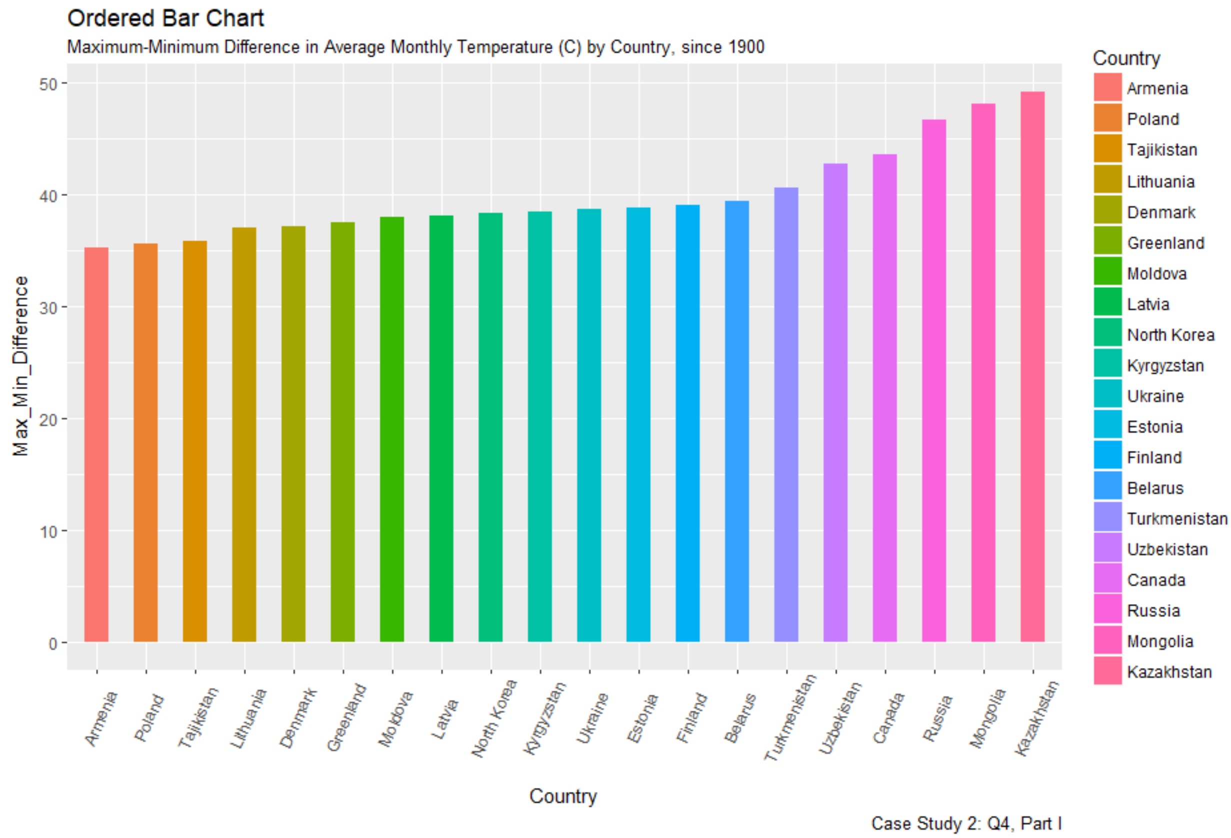
```
        Country Maximum Minimum Max_Min_Difference
116   Kazakhstan  25.562 -23.601             49.163
145     Mongolia  20.716 -27.294             48.010
181       Russia  16.893 -29.789             46.682
40        Canada  14.796 -28.736             43.532
235   Uzbekistan  30.375 -12.323             42.698
226 Turkmenistan  32.136  -8.443             40.579
23       Belarus  22.811 -16.527             39.338
76       Finland  18.967 -20.101             39.068
69       Estonia  22.332 -16.483             38.815
229      Ukraine  23.936 -14.724             38.660
121   Kyrgyzstan  19.275 -19.161             38.436
161  North Korea  23.952 -14.390             38.342
123       Latvia  22.279 -15.784             38.063
143      Moldova  25.231 -12.781             38.012
89     Greenland   0.339 -37.177             37.516
59       Denmark   0.699 -36.439             37.138
129    Lithuania  21.791 -15.179             36.970
217   Tajikistan  19.363 -16.466             35.829
175       Poland  22.509 -13.107             35.616
12       Armenia  25.291  -9.982             35.273
```

TEMP_Sorted_Max_Differences <- TEMP_20Max_Differences[order(TEMP_20Max_Differences$Max_Min_Difference),]

TEMP_Sorted_Max_Differences$Country <- factor(TEMP_Sorted_Max_Differences$Country, levels = TEMP_Sorted_Max_Differences$Country)

library(ggplot2)

```
ggplot(TEMP_Sorted_Max_Differences, aes(x=Country, y=Max_Min_Difference)) +
 geom_bar(stat="identity", width=.5, aes(fill = Country)) +
 labs(title="Ordered Bar Chart",
     subtitle="Maximum-Minimum Difference in Average Monthly Temperature (C) by Country, since 1900",
     caption="Case Study 2: Q4, Part I") +
 theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



#Question 4.III

CityTemp <- read_csv("C:/Users/jkras/Desktop/SMU/DDS/Case Study 2/CityTemp.csv")

CityTemp_2 <- subset(CityTemp, grepl("^.+(19|20)",Date))

CityTemp_new <- aggregate(CityTemp_2$`Monthly AverageTemp`, by = list(CityTemp_2$City), FUN = min, na.rm = T)

CityTemp_new2 <- aggregate(CityTemp_2$`Monthly AverageTemp`, by = list(CityTemp_2$City), FUN = max, na.rm = T)

CityTemp_min <- subset(CityTemp_new, select= x)

colnames(CityTemp_min) <- "Minimum"

colnames(CityTemp_new2) <- c("City", "Maximum")

CityTemp_Difference <- cbind(CityTemp_new2, CityTemp_min)

CityTemp_Difference$Max_Min_Difference <- (CityTemp_Difference$Maximum - CityTemp_Difference$Minimum)

CityTemp_20Max_Differences <-
CityTemp_Difference[order(CityTemp_Difference$Max_Min_Difference, decreasing = T)[1:20],]

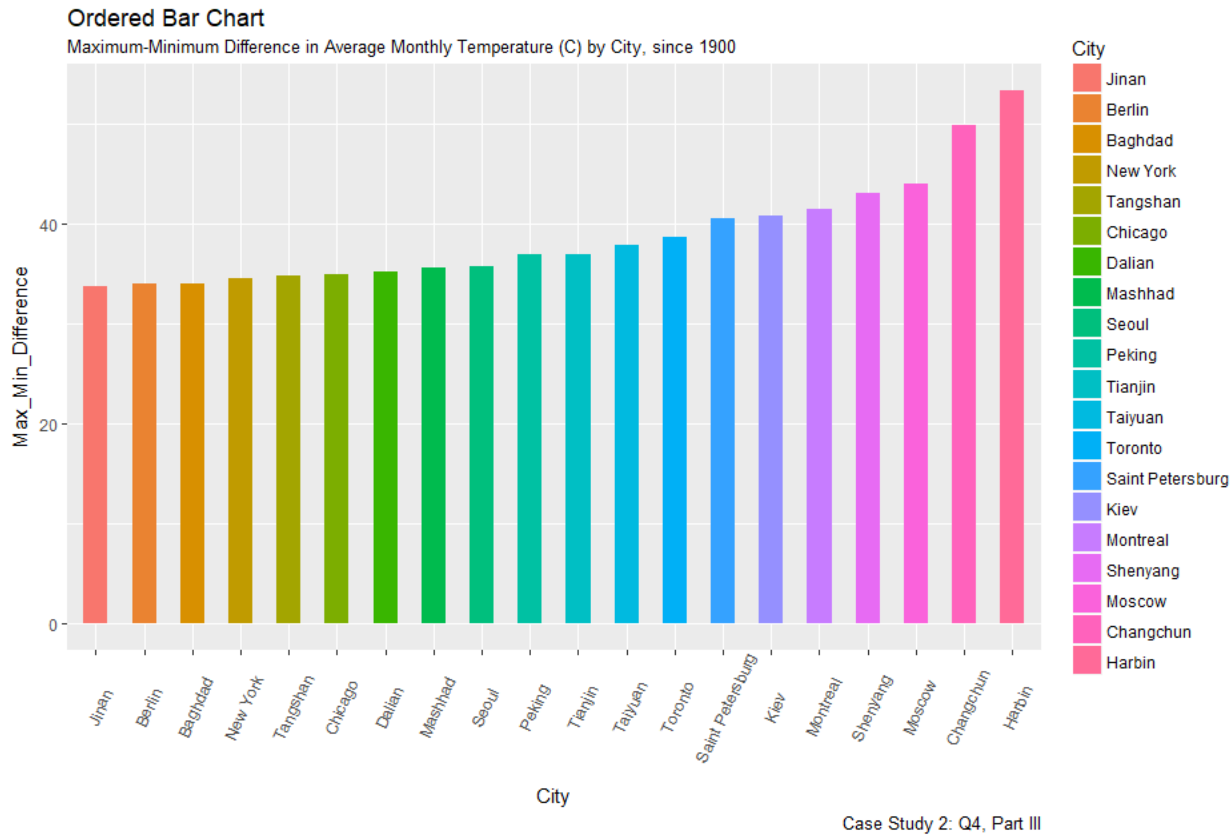
CityTemp_20Max_Differences

```
                 City Maximum Minimum Max_Min_Difference
34             Harbin  26.509 -26.772             53.281
19          Changchun  26.572 -23.272             49.844
65             Moscow  24.580 -19.376             43.956
86           Shenyang  26.010 -17.035             43.045
64           Montreal  23.059 -18.363             41.422
48               Kiev  24.593 -16.191             40.784
79 Saint Petersburg  21.921 -18.589             40.510
96            Toronto  23.181 -15.502             38.683
92            Taiyuan  24.718 -13.116             37.834
73             Peking  28.936  -8.017             36.953
94            Tianjin  28.936  -8.017             36.953
84              Seoul  26.791  -8.992             35.783
60            Mashhad  27.226  -8.384             35.610
24             Dalian  25.875  -9.348             35.223
21            Chicago  26.372  -8.590             34.962
93           Tangshan  27.346  -7.487             34.833
71           New York  25.313  -9.147             34.460
6             Baghdad  38.283   4.236             34.047
10             Berlin  23.795 -10.125             33.920
43              Jinan  28.389  -5.389             33.778
```

CityTemp_Sorted_Max_Differences <-
CityTemp_20Max_Differences[order(CityTemp_20Max_Differences$Max_Min_Difference),]

CityTemp_Sorted_Max_Differences$City <- factor(CityTemp_Sorted_Max_Differences$City, levels =
CityTemp_Sorted_Max_Differences$City)

```
ggplot(CityTemp_Sorted_Max_Differences, aes(x=City, y=Max_Min_Difference)) +
 geom_bar(stat="identity", width=.5, aes(fill = City)) +
 labs(title="Ordered Bar Chart",
     subtitle="Maximum-Minimum Difference in Average Monthly Temperature (C) by City, since 1900",
     caption="Case Study 2: Q4, Part III") +
 theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



#Question 4.IV

#Since 1900, the ranges of the maximum-minimum average monthly temperature difference by country and by city are roughly comparable.

#Of note is the larger spread in the top twenty cities (~20 degrees compared to ~14 degrees across the top twenty countries).

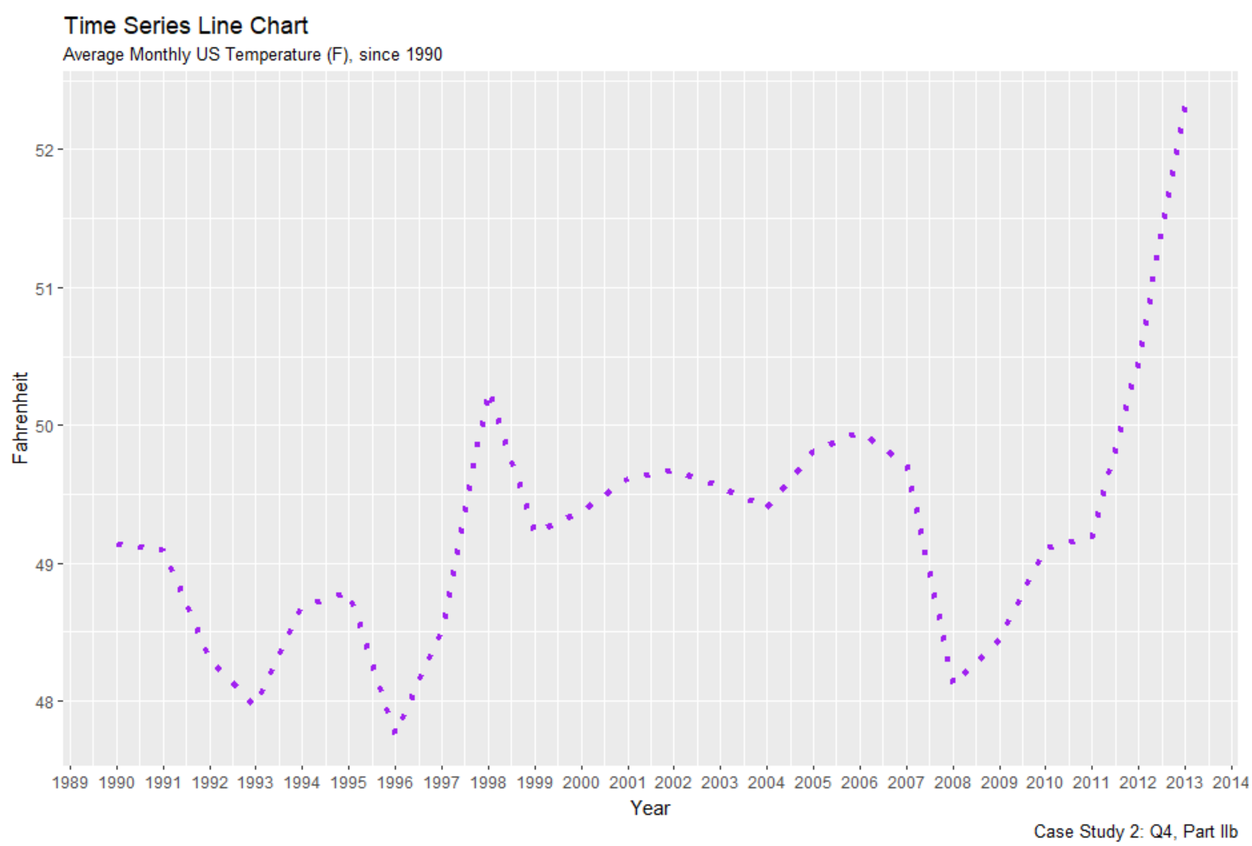
#Differences across countries are largely attributable to geographic and climatological differences. Arid, semi-arid, and landlocked humid continental regions with charcteristically large diurnal and seasonal fluctuations are especially prevalent on the list.

#Differences across cities are also partially attributable to the environmental impact of human activity in densely populated regions.

#As some of the listed cities are not situated in the list of countries, one can reasonably conclude that the diversity of climate types in countries with larger geographic landmass is also a factor.

#Question 4.II

```
US_TEMP <- subset(subset(TEMP, grepl("^.+(199|200|201)",Date)), Country == "United States")

US_TEMP$Fahrenheit <- US_TEMP$`Monthly AverageTemp` * (9/5) + 32

head(US_TEMP$Fahrenheit)

[1] 29.9786 28.8554 40.0370 48.8840 56.7896 67.6040

US_TEMP$Date <- as.Date(US_TEMP$Date, "%m/%d/%Y")

US_TEMP$Year <- as.Date(cut(US_TEMP$Date, breaks = "year"))

library(scales)

ggplot(data = US_TEMP, aes(Year, Fahrenheit)) +
 stat_summary(fun.y = mean, geom = "line", size = 1.5, linetype = "dotted", color = "purple") +
 scale_x_date(labels = date_format("%Y"), date_breaks = "1 year") +
 labs(title="Time Series Line Chart",
     subtitle="Average Monthly US Temperature (F), since 1990",
     caption="Case Study 2: Q4, Part IIb")
```



US_YR_mean <- aggregate(US_TEMP$Fahrenheit, by = list(US_TEMP$Year), FUN = mean)

colnames(US_YR_mean) <- c("Year", "Yr_Mean_Fahrenheit")

US_YR_Diff <- diff(US_YR_mean$Yr_Mean_Fahrenheit)

US_YR_Difference <- as.numeric(c("NA", US_YR_Diff))

US_Yearly_Mean_Diff <- cbind(US_YR_mean, US_YR_Difference)

colnames(US_Yearly_Mean_Diff)[3] <- "Yearly_Mean_Difference"

head(US_Yearly_Mean_Diff)

```
        Year Yr_Mean_Fahrenheit Yearly_Mean_Difference
1 1990-01-01           49.13885                     NA
2 1991-01-01           49.08860               -0.05025
3 1992-01-01           48.30140               -0.78720
4 1993-01-01           47.96105               -0.34035
5 1994-01-01           48.68660                0.72555
6 1995-01-01           48.79580                0.10920
```

US_Yearly_Mean_Diff[which.max(US_Yearly_Mean_Diff$Yearly_Mean_Difference),]

```
         Year Yr_Mean_Fahrenheit Yearly_Mean_Difference
24 2013-01-01            52.3348                1.86485
```

Question 5)

```
def convert_to_fahrenheit(celsius):

   temp = celsius.split()

   c = [float(i) for i in temp]

   f = [(i * 9 / 5 + 32) for i in c]

   print('The conversions from Celsius to Fahrenheit are: ', *f)


def convert_to_celsius(fahrenheit):

   temp = fahrenheit.split()

   f = [float(i) for i in temp]

   c = [((i - 32) * 5 / 9) for i in f]

   print('The conversions from Fahrenheit to Celsius are: ', *c)
```

```
def convert():
  while True:
    try:

      user_input = input('From what do you want to convert (C, F, or Q to quit)?: ')

      if user_input == 'celsius' or user_input == 'c' or user_input == 'C' or user_input == 'Celsius':
        print('To convert a temperature from Celsius to Fahrenheit:')
        celsius = input('CELSIUS as a (sequence of space-separated) number(s): ')
        convert_to_fahrenheit(celsius)

      elif user_input == 'fahrenheit' or user_input == 'f' or user_input == 'F' or user_input == 'Fahrenheit':
        print('To convert a temperature from Fahrenheit to Celsius:')
        fahrenheit = input('FAHRENHEIT as a (sequence of space-separated) number(s): ')
        convert_to_celsius(fahrenheit)

      elif user_input == 'q' or user_input == 'Q' or user_input == 'Quit' or user_input == 'quit':
        print('Quitting now')
        break

      else:
        raise ValueError

    except ValueError:
      print('Choose C, F, or Q. Insert temperatures as space-separated numbers.')
      continue

convert()
```

[Trial Run]

runfile('C:/Users/jkras/Desktop/Fahrenheit-Celsius Converter.py', wdir='C:/Users/jkras/Desktop')

From what do you want to convert (C, F, or Q to quit)?: c

To convert a temperature from Celsius to Fahrenheit:

CELSIUS as a (sequence of space-separated) number(s): 34 35 36

The conversions from Celsius to Fahrenheit are: 93.2 95.0 96.8

From what do you want to convert (C, F, or Q to quit)?: f

To convert a temperature from Fahrenheit to Celsius:

FAHRENHEIT as a (sequence of space-separated) number(s): 35 36 37

The conversions from Fahrenheit to Celsius are: 1.6666666666666667 2.2222222222222223 2.7777777777777777

From what do you want to convert (C, F, or Q to quit)?: ooga booga

Choose C, F, or Q. Insert temperatures as space-separated numbers.

From what do you want to convert (C, F, or Q to quit)?: " "

Choose C, F, or Q. Insert temperatures as space-separated numbers.

From what do you want to convert (C, F, or Q to quit)?:

Choose C, F, or Q. Insert temperatures as space-separated numbers.

From what do you want to convert (C, F, or Q to quit)?: celsius

To convert a temperature from Celsius to Fahrenheit:

CELSIUS as a (sequence of space-separated) number(s): 24 25 26

The conversions from Celsius to Fahrenheit are: 75.2 77.0 78.8

From what do you want to convert (C, F, or Q to quit)?: fahrenheit

To convert a temperature from Fahrenheit to Celsius:

FAHRENHEIT as a (sequence of space-separated) number(s): 57 58 59

The conversions from Fahrenheit to Celsius are: 13.88888888888889 14.444444444444445 15.0

From what do you want to convert (C, F, or Q to quit)?: quit

Quitting now