# Case Study II

Various Coding Problems

*Trace Smith & Damon Resnick*

*December 1, 2016*

### Question 1

**Create the following X matrix and print it from SAS, R, and Python.**

$$X = \begin{pmatrix} 4 & 5 & 1 & 2 \\ 1 & 0 & 3 & 5 \\ 2 & 1 & 8 & 2 \end{pmatrix}$$

Figure 1:

- **SAS Code**

```
proc iml;
/*create 3x4 matrix*/
reset print;
x={4 5 1 2,
   1 0 3 5,
   2 1 8 2};
quit;
```

- SAS output for X matrix shown below:

| | | | |
|---|---|---|---|
| 4 | 5 | 1 | 2 |
| 1 | 0 | 3 | 5 |
| 2 | 1 | 8 | 2 |

Figure 2:

- **R Code**

```
mymatrix <- matrix(c(4, 1, 2, 5, 0, 1, 1, 3, 8, 2, 5, 2), nrow = 3, ncol = 4)
print(mymatrix)
```

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

- **Python Code**

```
import numpy as np
x = np.matrix([[4,5,1,2],[1,0,3,5],[2,1,8,2]])
print x
```

- Python output (Ipython Notebook):

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

Figure 3:

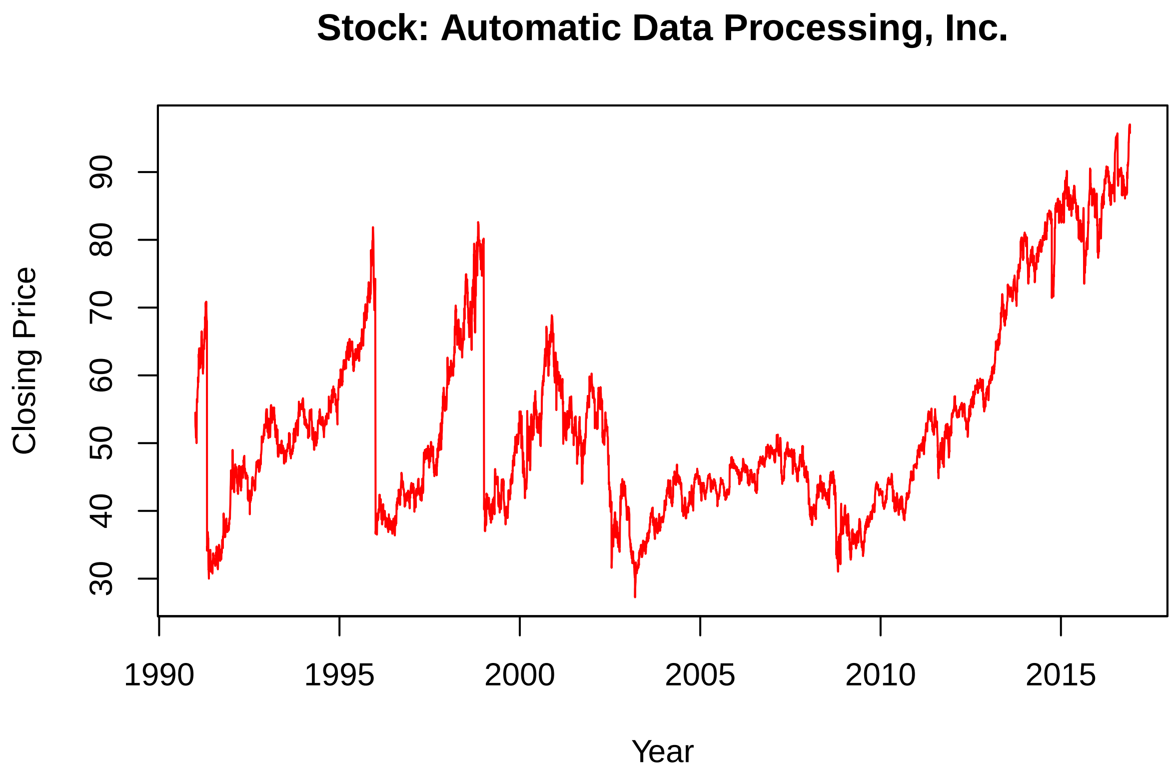
**Question 2**

- **Answer the following questions for Automatic Data Processing, Inc. stock (symbol = `ADP`):**

ADP, LLC., is an American provider of human resources management software and services. This file contains analysis of the ADP stock price from 1900-2016.

- **1.) Download the data.**

```r
library(tseries)
# SNPdatahist <- get.hist.quote('^gspc',quote='Close')
SNPdata <- get.hist.quote("adp", quote = "Close")
plot(SNPdata, col = "red", main = "Stock: Automatic Data Processing, Inc.", xlab = "Year",
    ylab = "Closing Price")
```




- **2.) Calculate log returns.**

```
SNPret <- log(lag(SNPdata)) - log(SNPdata)
SNPret <- SNPret[!(is.na(SNPret)), ]
# plot(SNPret,col='red',main='Stock: Automatic Data Processing,
# Inc.',xlab='Index',ylab='log(Returns)')
```

- **3.) Calculate volatility measure.**

```
SNPvol <- sd(SNPret) * sqrt(250) * 100
SNPvol
```

```
## [1] 34.28743
```

- **4.) Calculate volatility over entire length of series for various three different decay factors.**

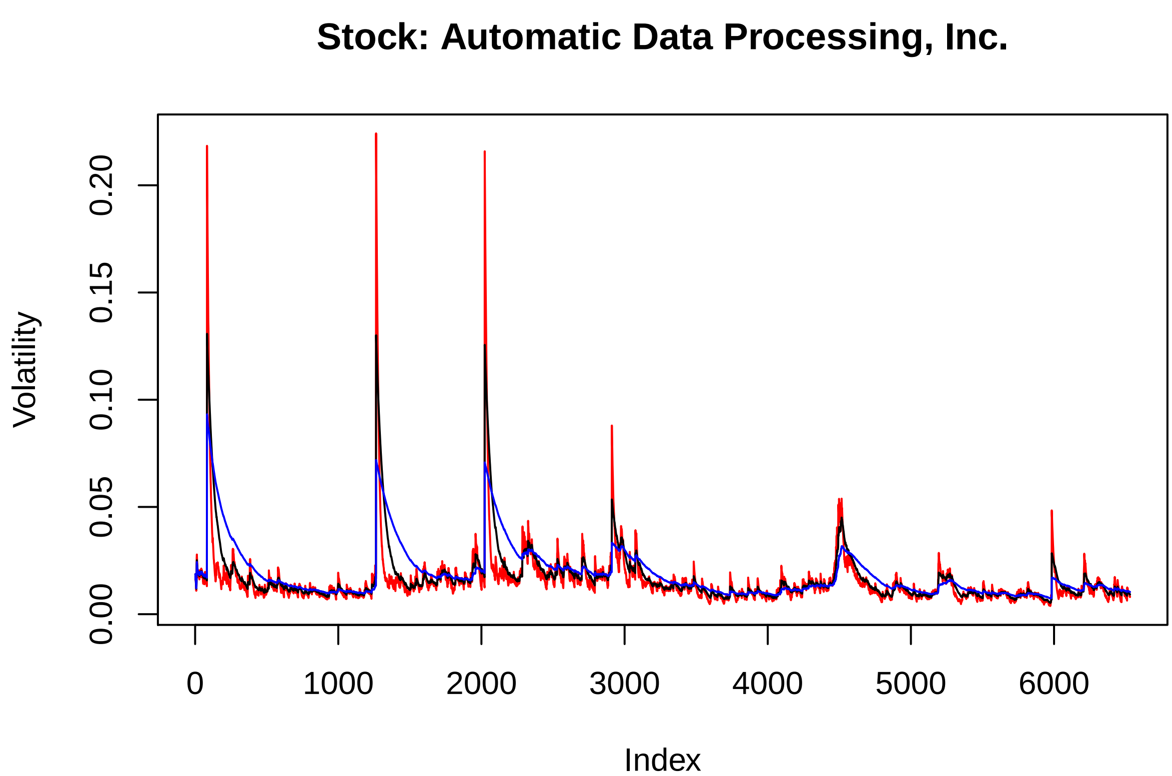
```
## volatility
get
```

```
## function (x, pos = -1L, envir = as.environment(pos), mode = "any",
##     inherits = TRUE)
## .Internal(get(x, envir, mode, inherits))
## <bytecode: 0x00000000135e70e0>
## <environment: namespace:base>
```

```
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, SNPret)
volest2 <- Vol(30, SNPret)
volest3 <- Vol(100, SNPret)
```

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

```
plot(volest, type = "l", col = "red", main = "Stock: Automatic Data Processing, Inc.",
    xlab = "Index", ylab = "Volatility")
lines(volest2, type = "l", col = "black")
lines(volest3, type = "l", col = "blue")
```



**Volatility for the ADP stock for the three different decay factors, 10 (red), 30(black), and 100(blue).**

### Question 3

- 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.

### 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.
- First, let's load the `Orange` data set into a data frame and examine the structure of the data:

```r
# Read in Orange dataset from R into data.frame
df <- data.frame(Orange)
```

```r
# Return first 6 rows of Orange df
head(df)
```

```
##   Tree  age circumference
## 1    1  118            30
## 2    1  484            58
## 3    1  664            87
## 4    1 1004           115
## 5    1 1231           120
## 6    1 1372           142
```

```r
# get summary of Orange dataset
summary(df)
```

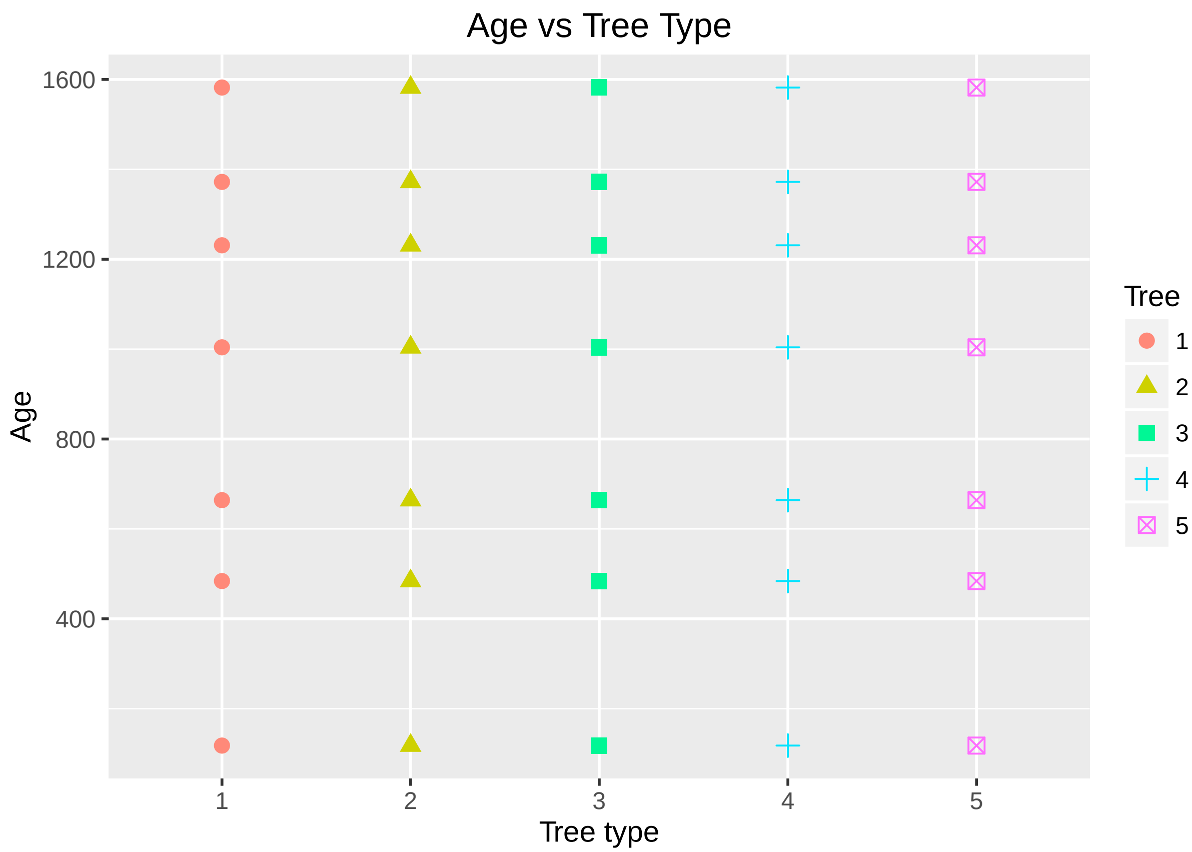
```
##  Tree       age         circumference
##  3:7   Min.   : 118.0   Min.   : 30.0
##  1:7   1st Qu.: 484.0   1st Qu.: 65.5
##  5:7   Median :1004.0   Median :115.0
##  2:7   Mean   : 922.1   Mean   :115.9
##  4:7   3rd Qu.:1372.0   3rd Qu.:161.5
##        Max.   :1582.0   Max.   :214.0
```

```r
# get structure of each columns
str(df)
```

```
## 'data.frame':    35 obs. of  3 variables:
##  $ Tree         : Ord.factor w/ 5 levels "3"<"1"<"5"<"2"<..: 2 2 2 2 2 2 2 4 4 4 ...
##  $ age          : num  118 484 664 1004 1231 ...
##  $ circumference: num  30 58 87 115 120 142 145 33 69 111 ...
```

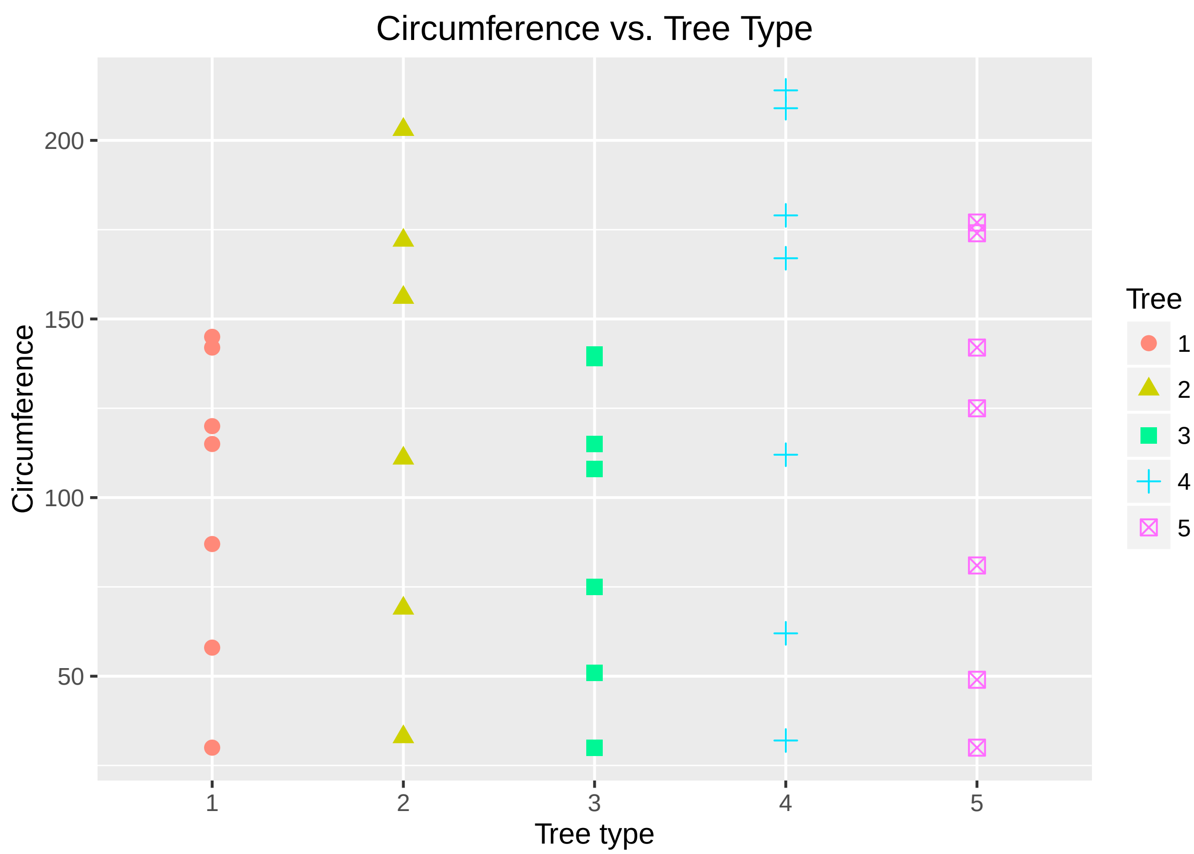
```r
df$Tree <- as.character(df$Tree)
```

```r
# Take a look at the data
p <- ggplot(df) + geom_point(aes(y = age, x = Tree, colour = Tree, shape = Tree),
    size = 2.5) + scale_colour_hue(l = 80, c = 150)
p + labs(title = "Age vs Tree Type", x = "Tree type", y = "Age", colour = "Tree") +
    theme(plot.title = element_text(hjust = 0.5))
```




**There are seven different ages in years:(118, 484, 664, 1004, 1231, 1372, 1582) and five tree types (1-5).**

```r
p2 <- ggplot(df) + geom_point(aes(y = circumference, x = Tree, colour = Tree, shape = Tree),
    size = 2.5) + scale_colour_hue(l = 80, c = 150)
p2 + labs(title = "Circumference vs. Tree Type", x = "Tree type", y = "Circumference",
    colour = "Tree") + theme(plot.title = element_text(hjust = 0.5))
```




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

```
# aggregate data.frame by Tree and compute mean circumference
circum.mean <- aggregate(df$circumference, by = list(df$Tree), FUN = mean)
colnames(circum.mean) <- c("Tree", "Mean Circ.")
circum.mean
```

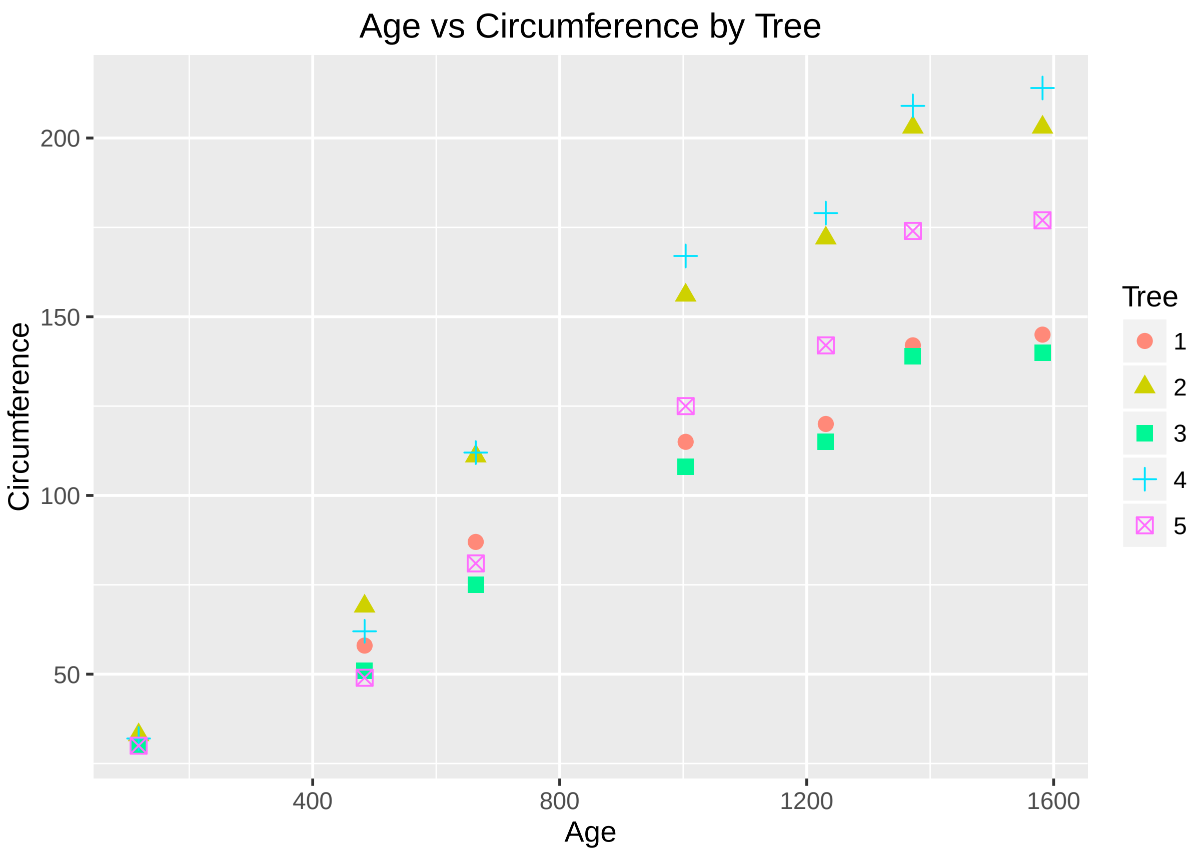
```
##   Tree Mean Circ.
## 1    1   99.57143
## 2    2  135.28571
## 3    3   94.00000
## 4    4  139.28571
## 5    5  111.14286
```

```
# aggregate data.frame by Tree and compute median circumference
circum.median <- aggregate(df$circumference, by = list(df$Tree), FUN = median)
colnames(circum.median) <- c("Tree", "Median Circ.")
circum.median
```

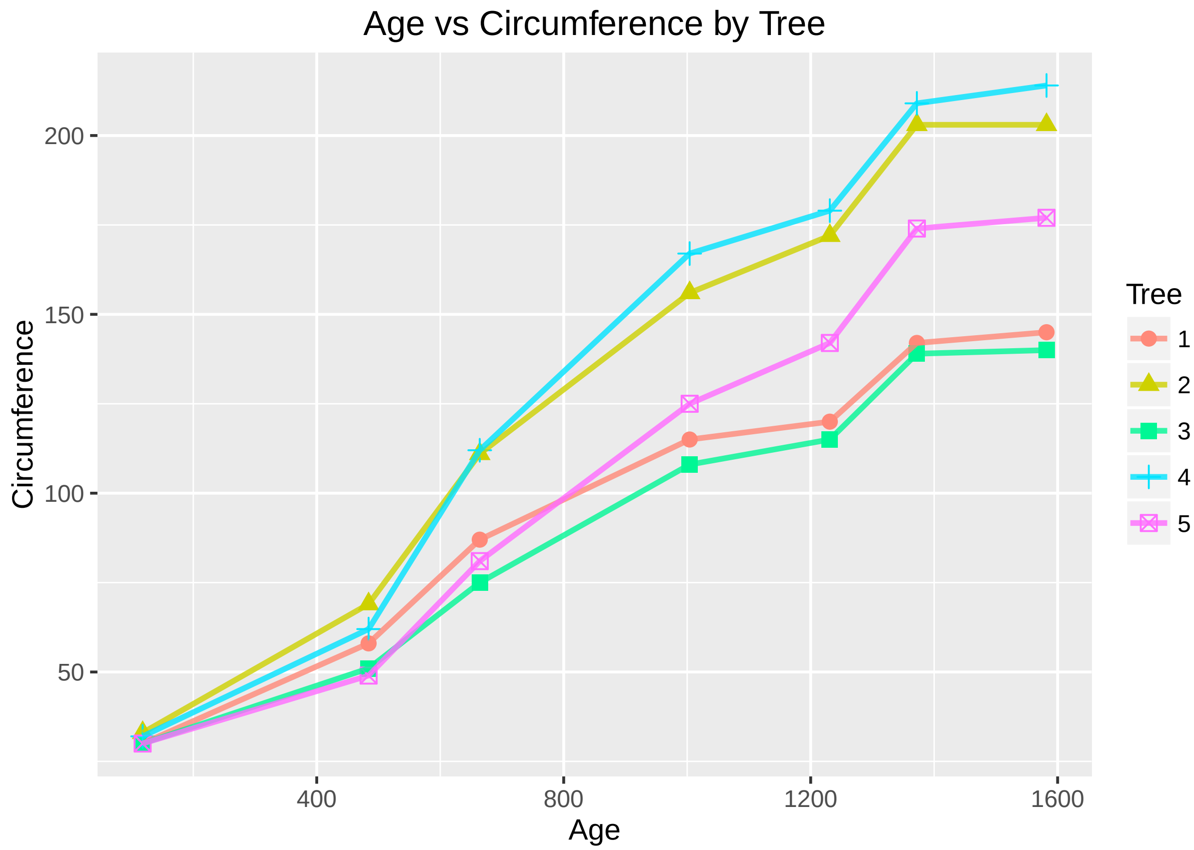
```
##   Tree Median Circ.
## 1    1          115
## 2    2          156
## 3    3          108
## 4    4          167
## 5    5          125
```

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

```
# Scatter plot
p <- ggplot(df) + geom_point(aes(y = circumference, x = age, colour = Tree, shape = Tree),
    size = 2.5) + scale_colour_hue(l = 80, c = 150)
p + labs(title = "Age vs Circumference by Tree", x = "Age", y = "Circumference",
    colour = "Tree") + theme(plot.title = element_text(hjust = 0.5))
```




```
# Line plot
p <- ggplot(df, aes(y = circumference, x = age, colour = Tree)) + geom_point(aes(shape = Tree),
    size = 2.5) + geom_line(size = 1, alpha = 0.8) + scale_colour_hue(l = 80, c = 150)
p + labs(title = "Age vs Circumference by Tree", x = "Age", y = "Circumference",
    colour = "Tree") + theme(plot.title = element_text(hjust = 0.5))
```



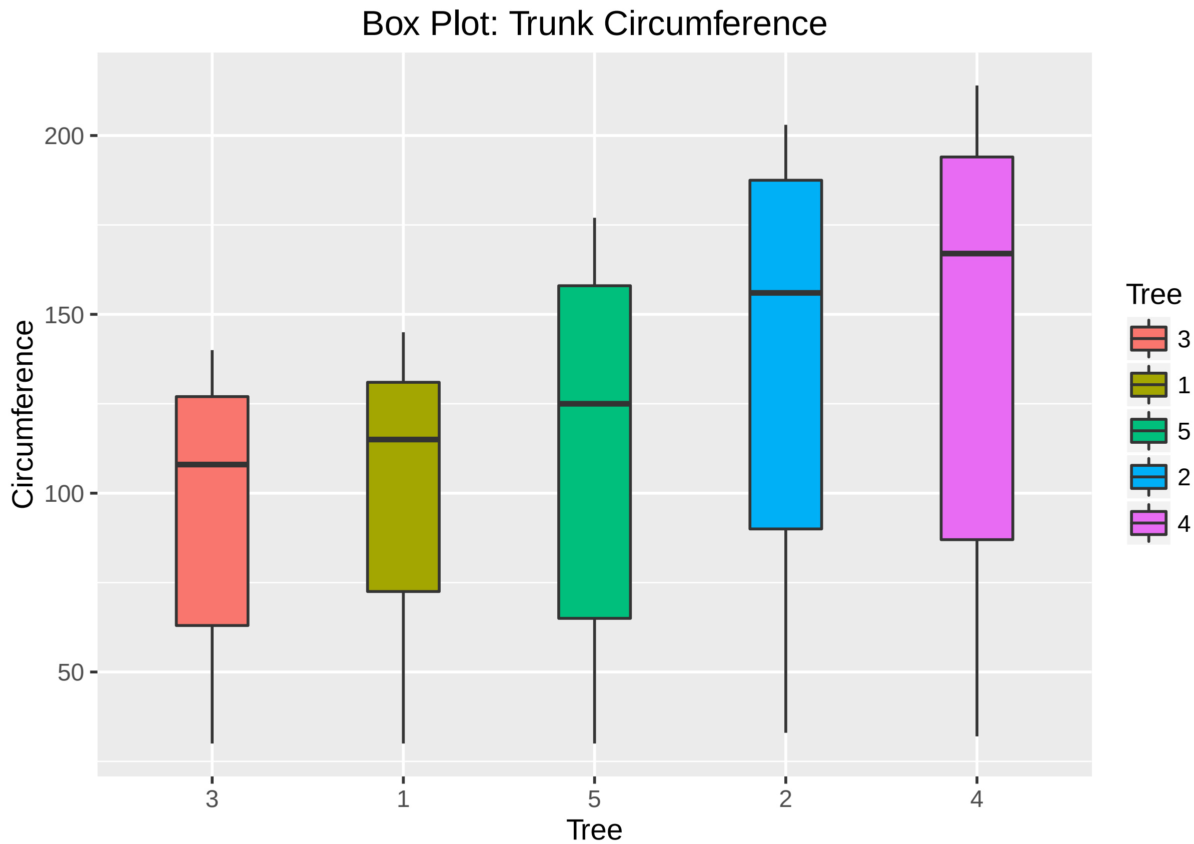

- **c) Display the trunk circumferences on a comparative boxplot against tree. Be sure you order the boxplots in the increasing order of maximum diameter.**

```
# Determine the max circum by each group and reorder the levels accordingly
circum.max <- aggregate(df$circumference, by = list(df$Tree), FUN = max)  #aggregate for max circum
colnames(circum.max) <- c("Tree", "Max Circum.")  #rename columns
circum.max
```

```
##   Tree Max Circum.
## 1    1         145
## 2    2         203
## 3    3         140
## 4    4         214
## 5    5         177
```

```
df$Tree <- factor(df$Tree, c("3", "1", "5", "2", "4"))  #reorder the boxplot for max circum. by tree
```

```
p <- ggplot(df, aes(x = Tree, y = circumference)) + geom_boxplot(aes(fill = Tree),
    width = 0.5)  # ggplot: boxplot
p + labs(title = "Box Plot: Trunk Circumference", y = "Circumference", x = "Tree") +
    theme(plot.title = element_text(hjust = 0.5))
```




**Box-plots for Tree Circumference vs. Tree type. You can see that tree type 4 has the largest average circumference while type 3 has the smallest.**

**Question 4**

**(i) First, download a 'Temp' data set. 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.****

**Load data**

```r
# Create new data.frame to join the two aggregated list'
tempraw <- read.csv("Data/Temp.csv", header = TRUE)
temp <- tempraw
head(temp)
```

```
##         Date Monthly.AverageTemp Monthly.AverageTemp.Uncertainty
## 1 1838-04-01              13.008                           2.586
## 2 1838-05-01                  NA                              NA
## 3 1838-06-01              23.950                           2.510
## 4 1838-07-01              26.877                           2.883
## 5 1838-08-01              24.938                           2.992
## 6 1838-09-01              18.981                           2.538
```

```
##       Country
## 1 Afghanistan
## 2 Afghanistan
## 3 Afghanistan
## 4 Afghanistan
## 5 Afghanistan
## 6 Afghanistan
```

### Preprocessing and Exploring the Data

```
# Need to make Date column into a character in order to use grepl to extract out
# other date format
temp$Date <- as.character(temp$Date)

# Deletes all the dates below 1900 because all of those dates are in a different
# format with '-' and not '/'
temp <- temp[!grepl("-", temp$Date), ]

# Remove any columns with 'NA' just to be careful
temp1 <- temp[!(is.na(temp$Date)), ]

# Make Country column a character
temp1$Country <- as.character(temp1$Country)

# return all the rows (i.e. margin=1) with NA
row.with.na <- apply(temp, 1, function(x) {
    any(is.na(x))
})

# Sum all of the rows containing NA
sprintf("Number of Rows Deleted that contained NA's: %s", sum(row.with.na))
```

```
## [1] "Number of Rows Deleted that contained NA's: 1049"
```

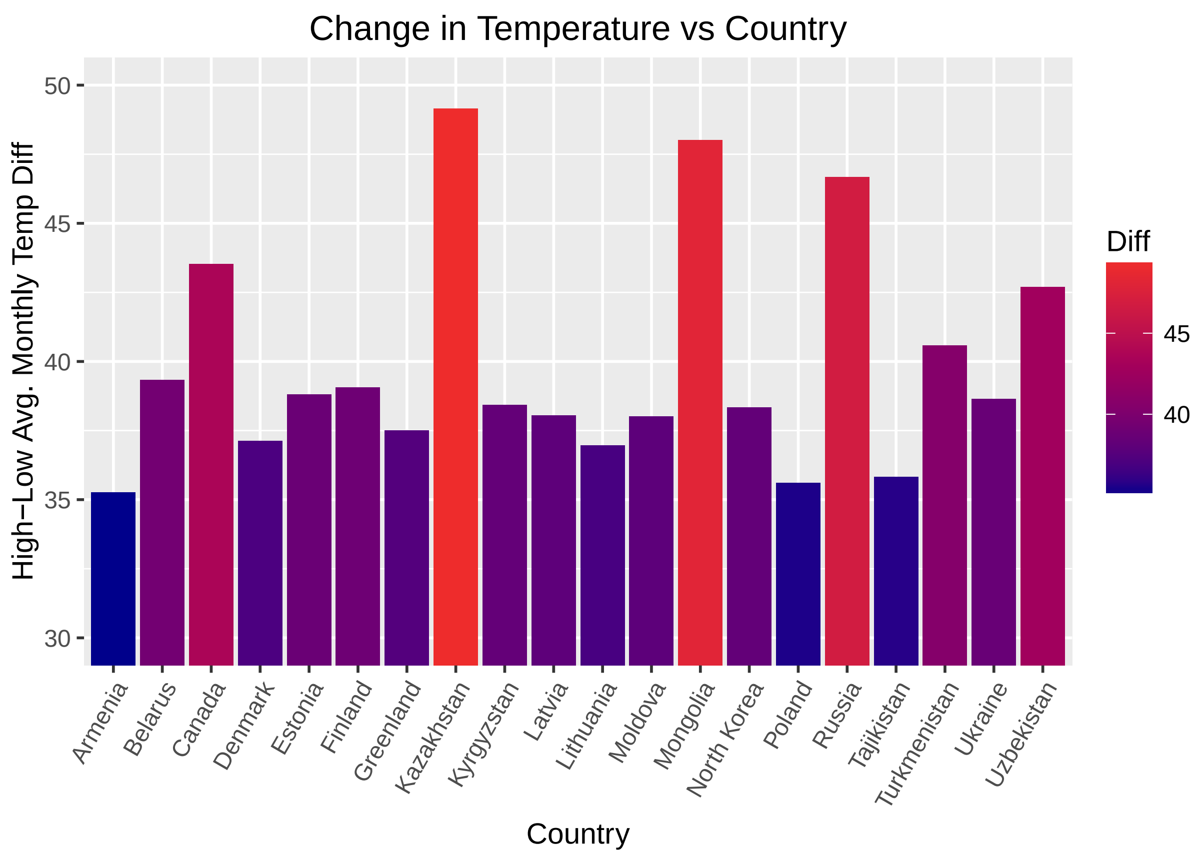
```
# Remove the Rows with NA's
temp1 <- temp[!row.with.na, ]
```

```
# Aggregate for max and min average temps
temp.max <- aggregate(temp1["Monthly.AverageTemp"], by = temp1["Country"], FUN = max)
temp.min <- aggregate(temp1["Monthly.AverageTemp"], by = temp1["Country"], FUN = min,
    na.rm = TRUE)

# Create new data.frame to join the two aggregated list
data <- data.frame(temp.max, temp.min)

# Drop extra Country column
data$Country.1 <- NULL

# Rename column
colnames(data) <- c("Country", "Max Avg. Temp", "Min Avg. Temp")

# Take difference between max and min avg. temp columns
data$Diff <- data$"Max Avg. Temp" - data$"Min Avg. Temp"
```

```r
# Sort the dataframe by decreasing Diff
data <- data[order(data$Diff, data$Country, decreasing = TRUE), ]
head(data, 20)
```

```
##          Country Max Avg. Temp Min Avg. Temp   Diff
## 115   Kazakhstan        25.562       -23.601 49.163
## 144     Mongolia        20.716       -27.294 48.010
## 180       Russia        16.893       -29.789 46.682
## 39        Canada        14.796       -28.736 43.532
## 234   Uzbekistan        30.375       -12.323 42.698
## 225 Turkmenistan        32.136        -8.443 40.579
## 22       Belarus        22.811       -16.527 39.338
## 75       Finland        18.967       -20.101 39.068
## 68       Estonia        22.332       -16.483 38.815
## 228      Ukraine        23.936       -14.724 38.660
## 120   Kyrgyzstan        19.275       -19.161 38.436
## 160  North Korea        23.952       -14.390 38.342
## 122       Latvia        22.279       -15.784 38.063
## 142      Moldova        25.231       -12.781 38.012
## 88     Greenland         0.339       -37.177 37.516
## 58       Denmark         0.699       -36.439 37.138
## 128    Lithuania        21.791       -15.179 36.970
## 216   Tajikistan        19.363       -16.466 35.829
## 174       Poland        22.509       -13.107 35.616
## 11       Armenia        25.291        -9.982 35.273
```

```r
# Subset the data to only take the first 20 columns with highest temp diff.
data.sub <- data[1:20, ]

# plot Country vs Temp Diff
p <- ggplot(data.sub, aes(Country, Diff, fill = Diff)) + geom_bar(stat = "identity") +
    scale_fill_gradientn(colours = c("dodgerblue1", "darkblue", "firebrick2"), values = scale(c(35,
        40, 45)))
p + labs(title = "Change in Temperature vs Country", x = "Country", y = "High-Low Avg. Monthly Temp Dif
    theme(plot.title = element_text(hjust = 0.5)) + theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + coord_cartesian(ylim = c(30, 50))
```




**(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.****

```
temp.usa <- subset(temp1, temp1$Country == "United States")
# pander(head(temp.usa))
which(temp.usa$Date == "1/1/90")
```

```
## [1] 1081
```

```
temp.usa <- temp.usa[-c(1:1080), ]
temp.usa$Date <- as.Date(temp.usa$Date, format = "%m/%d/%y")
```

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

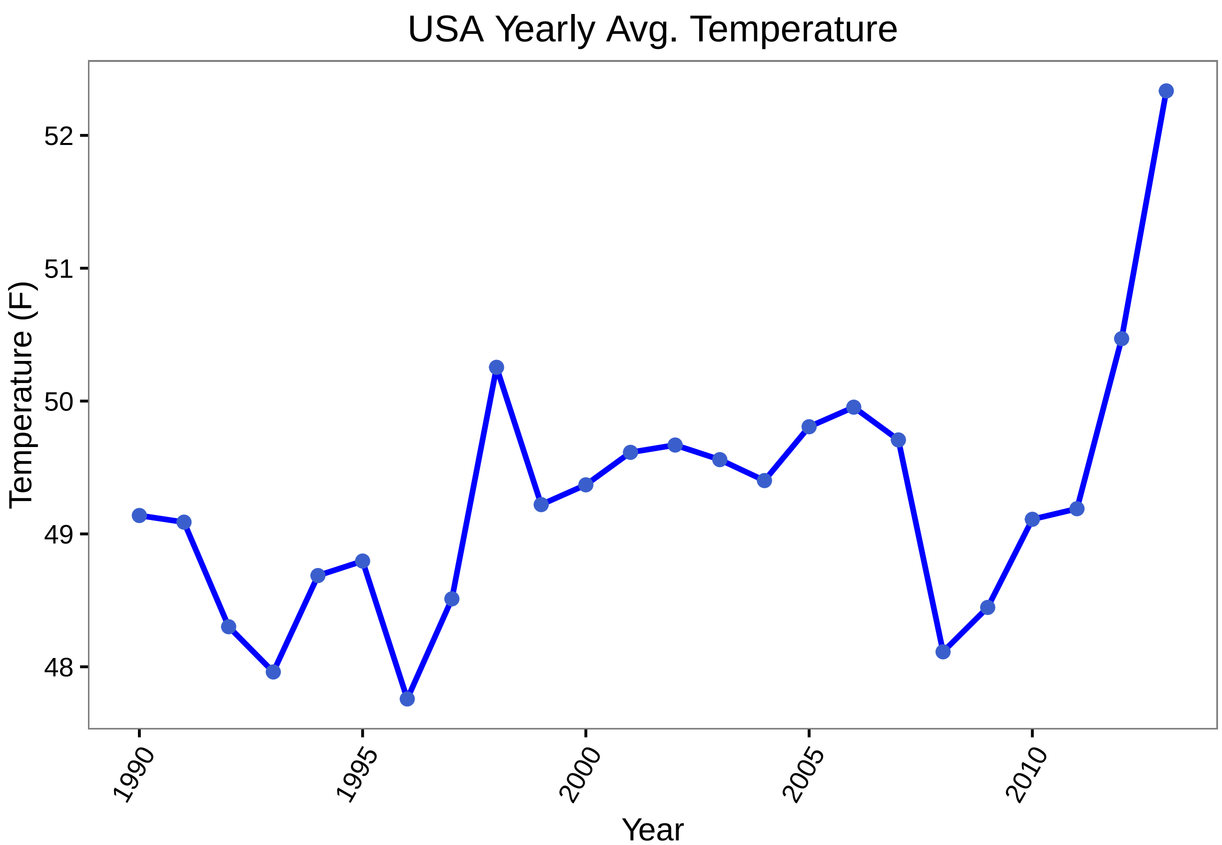
```
temp.usa$Temp_F <- ((temp.usa$Monthly.AverageTemp * (9/5)) + 32)
head(temp.usa["Temp_F"])
```

```
##          Temp_F
## 554298 29.9786
## 554299 28.8554
## 554300 40.0370
## 554301 48.8840
## 554302 56.7896
## 554303 67.6040
```

- **b) Calculate average land temperature by year and plot it. The original file has the average land temperature by month.**

```
# Average Land Temperatue by Year:
temp.usa$year <- substr(temp.usa$Date, 1, 4)
df.temp.usa <- do.call(data.frame, aggregate(Temp_F ~ year, data = temp.usa, FUN = mean))
df.temp.usa$year <- as.numeric(as.character(df.temp.usa$year))
# str(df.temp.usa$year)
```

```
# plot USA yearly avgerage temp
p <- ggplot(df.temp.usa) + geom_line(aes(x = year, y = Temp_F), stat = "identity",
    lwd = 1, colour = "blue") + geom_point(aes(x = year, y = Temp_F), color = "royalblue3",
    size = 2)
p + labs(title = "USA Yearly Avg. Temperature", x = "Year", y = "Temperature (F)") +
    theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    theme(axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "none") +
    theme(plot.title = element_text(hjust = 0.5))
```



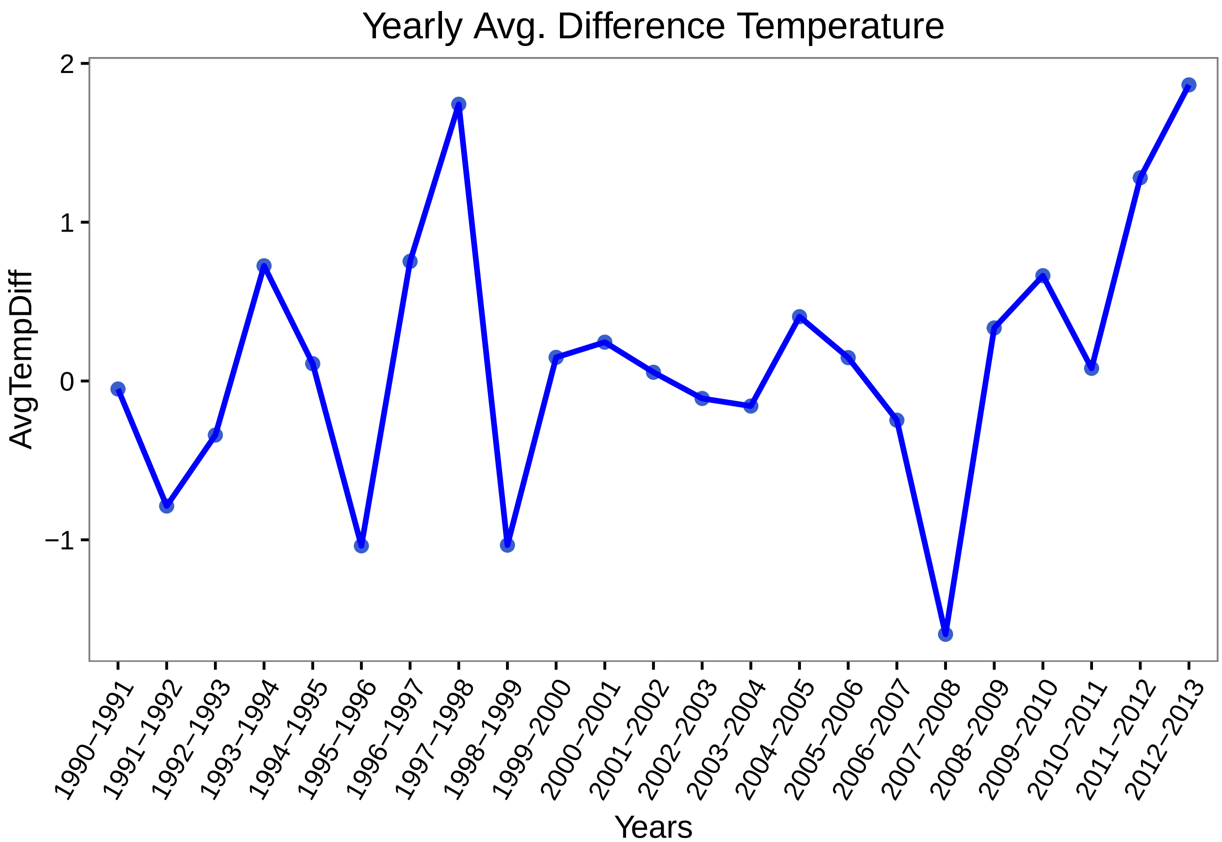

- **c) Calculate the one year difference of average land temperature by year**

```
temp.usa.year.diff <- df.temp.usa$Temp_F[2:24] - df.temp.usa$Temp_F[1:23]
```

```
# Create a function that returns a character vector for the difference in years
# from 1990 to 2013 pass the sequence of dates
```

```r
diff.year <- function(y) {
    date.char <- as.character(y)  #convert dates to strings
    date.str <- c()  # initialize vector
    for (i in 1:length(date.char)) {
        # iterate from 1 to length of vector
        date.str[i] <- paste0(date.char[i], "-", date.char[i + 1])  #concat date(n) and date(n+1)
    }  # returns date: (i.e. 1990-1991,1992-1992,etc..)
    # Remove the last date: (i.e. NA-2013)
    date.str <- date.str[-length(date.str)]
    return(date.str)
}
# Create new object calling the sequence method from the date class
one.year <- seq(1990, 2013, 1)
# call the diff.year function and pass the sequence
temp.usa.year.diff.year <- diff.year(one.year)

temp.usa.ydiff <- data.frame(temp.usa.year.diff.year, temp.usa.year.diff)
colnames(temp.usa.ydiff) <- c("Years", "AvgTempDiff")

temp.usa.ydiff$Years2 <- as.integer(temp.usa.ydiff$Years)
```

```r
p <- ggplot(temp.usa.ydiff) + geom_point(aes(x = Years, y = AvgTempDiff), size = 2,
    colour = "royalblue3")
p + labs(title = "Yearly Avg. Difference Temperature") + theme_bw() + theme(panel.grid.major = element_
    panel.grid.minor = element_blank()) + theme(axis.text.x = element_text(angle = 60,
    hjust = 1), legend.position = "none") + theme(plot.title = element_text(hjust = 0.5)) +
    geom_line(aes(x = Years2, y = AvgTempDiff), colour = "Blue", lwd = 1)
```




**As you can see from the above two graphs it looks like the difference between the average temperature of the last two years of the data set 2012 and 2013 is the largest between any consecutive years. That value is easy to calculate and is 1.86485.**

```
# Max difference
max(temp.usa.ydiff$AvgTempDiff)
```

```
## [1] 1.86485
```

**Again the maximum yearly difference was seen in the last two years 2012-2013.**

**(iii) Download 'CityTemp' data set. 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.**

```
citytempraw <- read.csv("./Data/CityTemp.csv", header = TRUE)
citytemp <- citytempraw
head(citytemp)
```

```
##         Date Monthly.AverageTemp Monthly.AverageTemp.Uncertainty
## 1 1850-01-01              15.986                           1.537
## 2 1850-02-01              18.345                           1.527
## 3 1850-03-01              18.632                           2.162
```

```
## 4 1850-04-01              18.154                             1.693
## 5 1850-05-01              17.480                             1.237
## 6 1850-06-01              17.183                             1.252
##          City  Country Latitude Longitude
## 1 Addis Abeba Ethiopia    8.84N    38.11E
## 2 Addis Abeba Ethiopia    8.84N    38.11E
## 3 Addis Abeba Ethiopia    8.84N    38.11E
## 4 Addis Abeba Ethiopia    8.84N    38.11E
## 5 Addis Abeba Ethiopia    8.84N    38.11E
## 6 Addis Abeba Ethiopia    8.84N    38.11E
```

```
# Preprocessing the Data:

# Convert the Date column into a character in order to use grepl to extract out
# other date format
citytemp$Date <- as.character(citytemp$Date)
# Delete all dates below 1900 because all of those dates are in a different
# format with '-' and not '/'
citytemp <- citytemp[!grepl("-", citytemp$Date), ]

row.with.na <- apply(citytemp, 1, function(x) {
    any(is.na(x))
})
sprintf("Number of rows deleted with NA's: %s", sum(row.with.na))
```

```
## [1] "Number of rows deleted with NA's: 92"
```

```
citytemp1 <- citytemp[!row.with.na, ]
```

```
# Identify which columns are strings
cols = c(4, 5, 6, 7)
# convert these columns to characters using the apply function
citytemp1[, cols] = apply(citytemp1[, cols], 2, function(x) as.character(x))
# test if worked correctly
str(citytemp1$City)
```

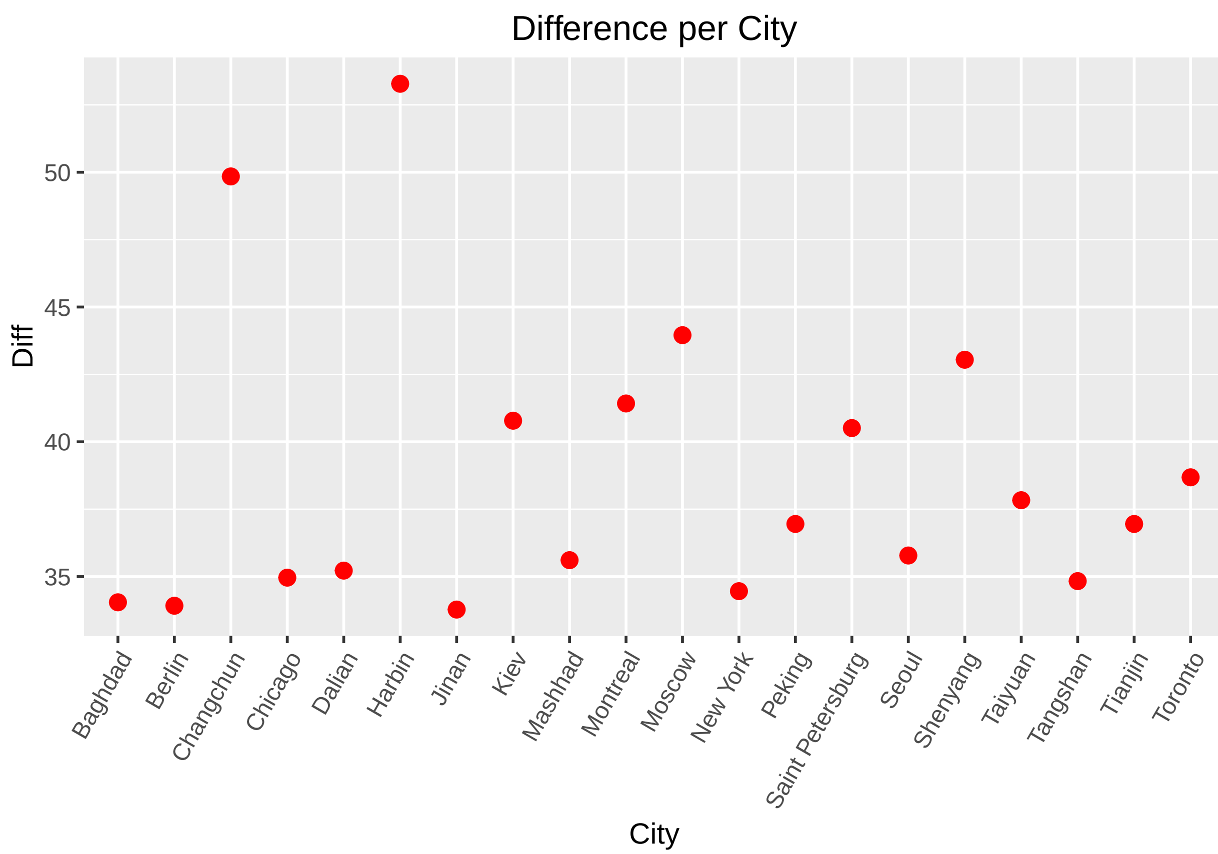
```
##  chr [1:135043] "Addis Abeba" "Addis Abeba" "Addis Abeba" ...
```

```
# Aggregate for max and min average temps
citytemp.max <- aggregate(citytemp1["Monthly.AverageTemp"], by = citytemp1["City"],
    FUN = max)
citytemp.min <- aggregate(citytemp1["Monthly.AverageTemp"], by = citytemp1["City"],
    FUN = min, na.rm = TRUE)

# Create new data.frame to join the two aggregated list
citydata <- data.frame(citytemp.max, citytemp.min)

# Drop extra Country column
citydata$City.1 <- NULL

# Rename column
colnames(citydata) <- c("City", "Max Avg. Temp", "Min Avg. Temp")
```

```r
# Take difference between max and min avg. temp columns
citydata$Diff <- citydata$"Max Avg. Temp" - citydata$"Min Avg. Temp"

# Sort the dataframe by decreasing Diff
citydata <- citydata[order(citydata$Diff, citydata$City, decreasing = TRUE), ]
head(citydata, 20)
```

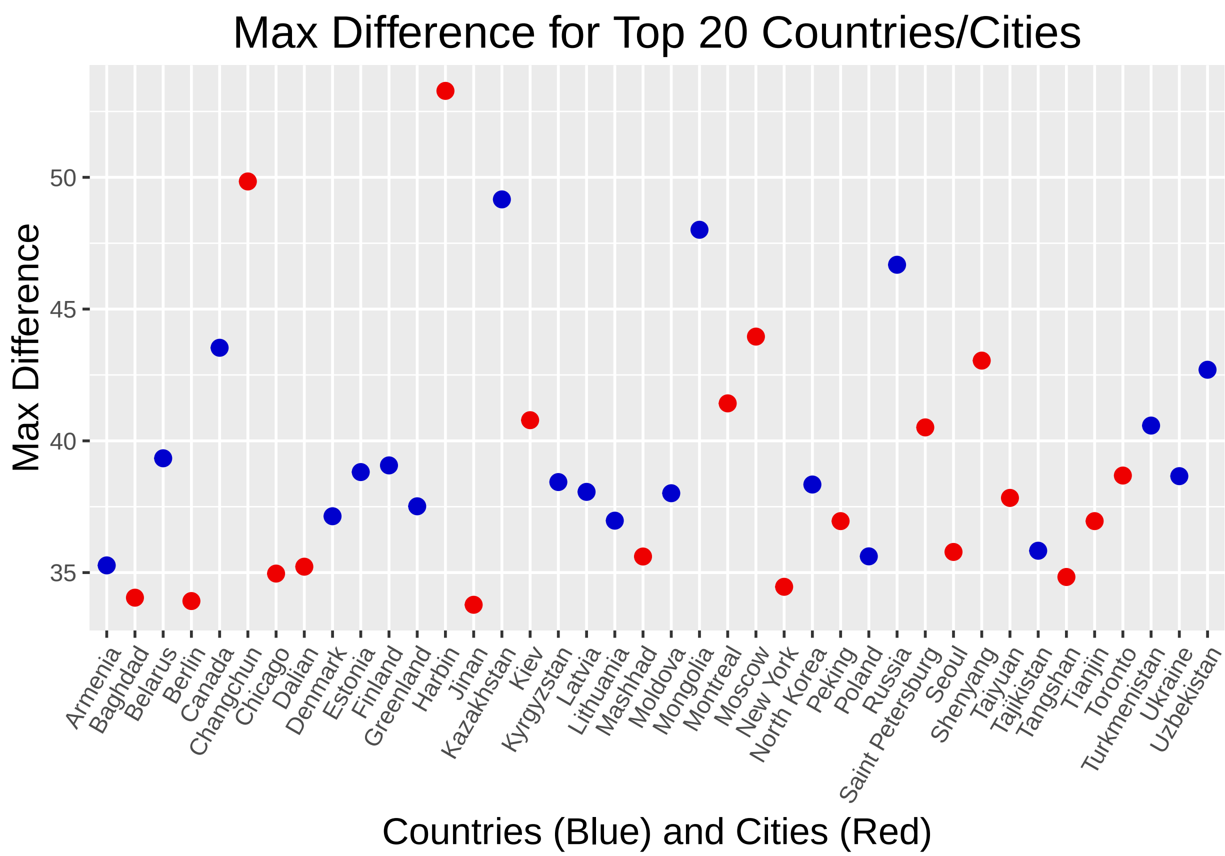
```
##                City Max Avg. Temp Min Avg. Temp   Diff
## 34           Harbin        26.509       -26.772 53.281
## 19        Changchun        26.572       -23.272 49.844
## 65           Moscow        24.580       -19.376 43.956
## 85         Shenyang        26.010       -17.035 43.045
## 64         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
## 94          Tianjin        28.936        -8.017 36.953
## 73           Peking        28.936        -8.017 36.953
## 83            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
```

```r
# Subset the data to only take the first 20 columns with highest temp diff.
citydata.sub <- citydata[1:20, ]

p <- ggplot(citydata.sub) + geom_point(aes(x = City, y = Diff), size = 2.5, colour = "Red")
p + labs(title = "Difference per City") + theme(axis.text.x = element_text(angle = 60,
    hjust = 1), legend.position = "none") + theme(plot.title = element_text(hjust = 0.5))
```




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

```
p4 <- ggplot() + geom_point(data = data.sub, aes(x = data.sub$Country, y = data.sub$Diff),
    color = "Blue3", size = 2.5) + geom_point(data = citydata.sub, aes(x = citydata.sub$City,
    y = citydata.sub$Diff), color = "Red2", size = 2.5)
p4 + labs(title = "Max Difference for Top 20 Countries/Cities", x = "Countries (Blue) and Cities (Red)"
    y = "Max Difference") + theme(title = element_text(size = 14), axis.title = element_text(size = 14)
    axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "none") +
    theme(plot.title = element_text(hjust = 0.5))
```

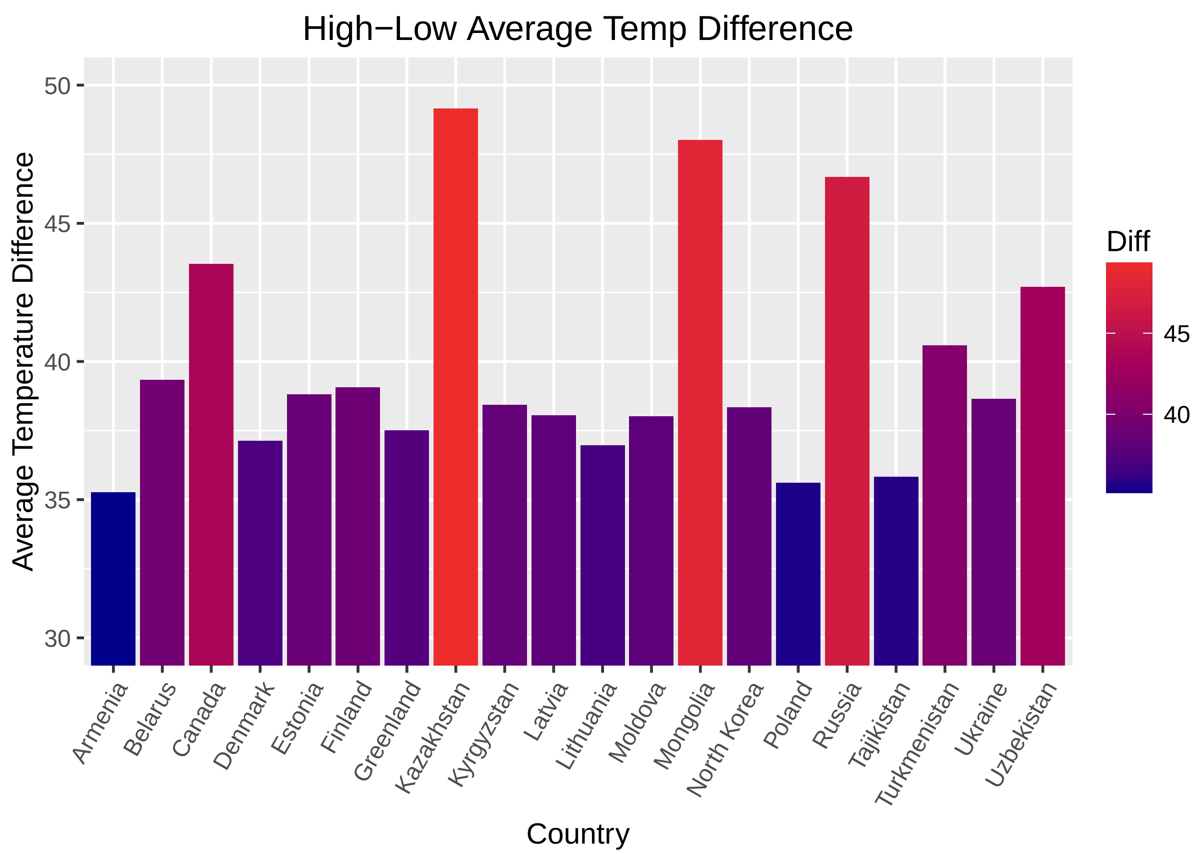



**Looking at the top 20 countries and cities in the world for the temperature swing during a year we see that the City of Harbin has the largest temperature swing, but by and large the countries have a wider temperature swing than the major cities. It is intersting to note that Russia has the 3rd largest temperature swing for a country and has two major cities in the top 20. Canada ranked 4th, also has two top 20 cities while the US is not ranked in the top 20 but has two cities in the top 20.**

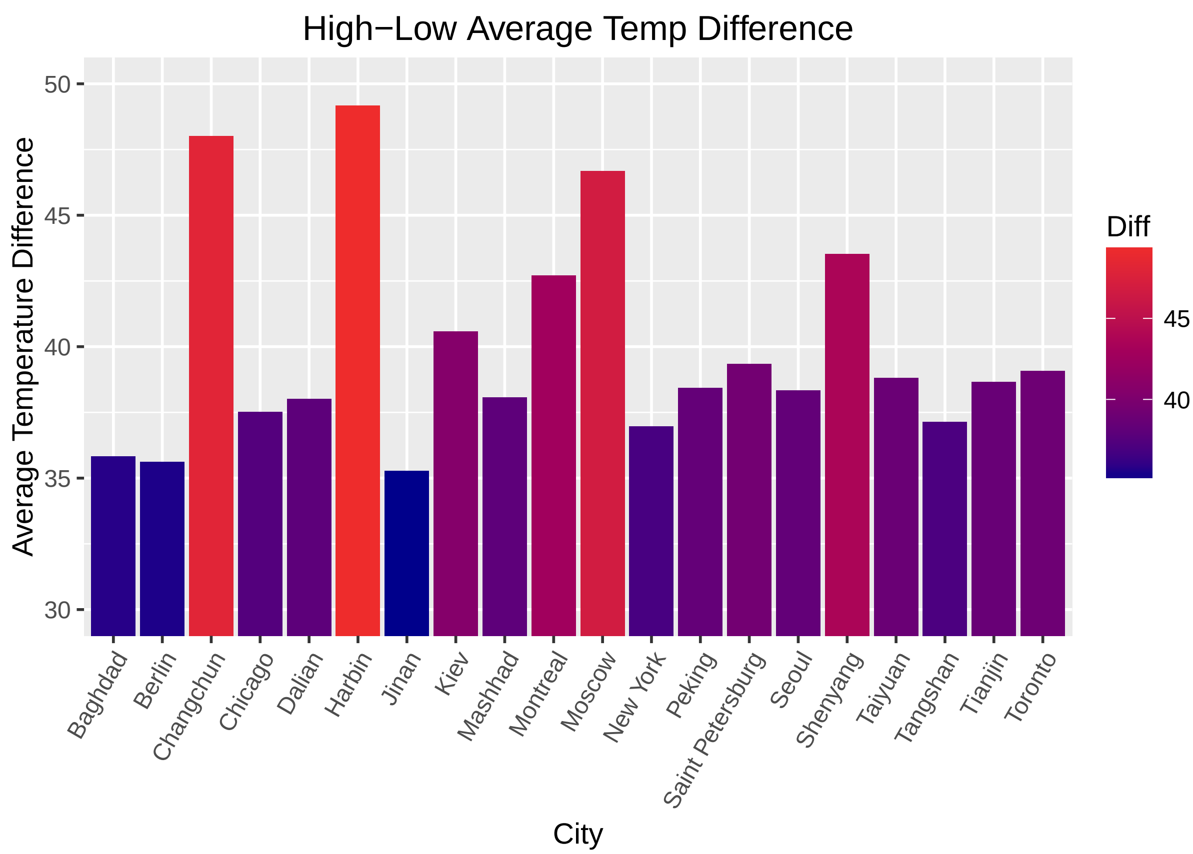
**Extra Exploratory Analysis**

```
hilow <- function(df, x, name) {
    p <- ggplot(data = df, aes(x, Diff, fill = Diff)) + geom_bar(stat = "identity") +
        coord_cartesian(ylim = c(30, 50)) + scale_fill_gradient(low = "darkblue",
        high = "firebrick2")
    p + labs(title = "High-Low Average Temp Difference", x = name, y = "Average Temperature Difference")
        theme(axis.text.x = element_text(angle = 60, hjust = 1)) + theme(plot.title = element_text(hjus
}
```

```
# Difference in Average Temp Per Country
country_var <- data.sub$Country
hilow(data.sub, country_var, name = "Country")
```




```r
# Difference in Average Temp Per Country
city_var <- citydata.sub$City
hilow(data.sub, city_var, name = "City")
```




**Question 05**

**Christmas Bonus**

Figure 4: