Data Analytics HW3_Part A

Tanmay Nema

10/27/2020

PART A: Exploring Base R Features

Question 1: Initial data processing and exploratory data analysis

1.a: Read in the raw CSV file, and create a data frame called raw_data that contains only the date (dt or dt_iso), city_id, and five other variables (temp, humidity, wind_speed, rain_1h, and weather_id). As needed, change the variable classes, and specify factors, to make sure the rest of your analysis will work correctly!

Ans: The raw data is imported into the R program using the read.table function while retaining the column headings. There were 253915 records and 28 attributes. The data.frame function was used to create the required data frame. New column names are also assigned at the beginning to avoid any confusion at later stages.

```
# 1.a
readfile <- read.table("raw_data.csv",TRUE,",")
raw_data <- data.frame(readfile$dt_iso,readfile$city_id,readfile$temp,readfile$humidity,readfile$wind_speed,readf
ile$rain_1h,readfile$weather_id)
colnames(raw_data)<-c("Date","City_ID","Temperature","Humidity","Wind_Speed","Rain_1hr","Weather")
raw_data$Date <- as.POSIXct(raw_data$Date, tz="UTC")
raw_data$City_ID <- as.factor(raw_data$City_ID)
raw_data$Weather <- as.factor(raw_data$Weather)
head(raw_data)
```

```
##                  Date City_ID Temperature Humidity Wind_Speed Rain_1hr Weather
## 1 2012-10-10 16:00:00 2643743      286.10       66          4       NA     803
## 2 2012-10-10 17:00:00 2643743      285.34       71          4       NA     803
## 3 2012-10-10 18:00:00 2643743      284.48       71          4       NA     803
## 4 2012-10-10 19:00:00 2643743      284.16       91          0        0     800
## 5 2012-10-10 20:00:00 2643743      283.78       76          2       NA     802
## 6 2012-10-10 21:00:00 2643743      283.41       96          0        0     800
```

1.b: How many observations are there overall in the raw data? Also, make a ‘table’ to show how many raw observations are there for each of the city_id’s specified.

Ans: The str() function provides information about the structure of the data frames. We have 253915 number of records and have selected 7 key attributes to focus on. The frequency statistic for the number of records for each city is also shown below using the table() function in Base R.

```
# 1.b
str(raw_data) #Gives number of records
```

```
## 'data.frame':    253915 obs. of  7 variables:
##  $ Date       : POSIXct, format: "2012-10-10 16:00:00" "2012-10-10 17:00:00" ...
##  $ City_ID    : Factor w/ 6 levels "2643743","2950159",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Temperature: num  286 285 284 284 284 ...
##  $ Humidity   : int  66 71 71 91 76 96 85 85 85 85 ...
##  $ Wind_Speed : int  4 4 4 0 2 0 2 2 2 2 ...
##  $ Rain_1hr   : num  NA NA NA 0 NA 0 NA NA NA NA ...
##  $ Weather    : Factor w/ 43 levels "200","201","202",..: 42 42 42 39 41 39 39 39 39 39 ...
```

```
table(raw_data$City_ID)
```

```
##
## 2643743 2950159 2988507 3143244 3169070 6458923
##   49979   43886   35965   41964   40006   42115
```

1.c: Perform a high-level EDA of the data frame created in 1a. As always, describe the dataset, and identify potential data problems from the EDA results (but do not fix/clean the data yet). Discuss which of the weather variables have the least or most NA (missing) values.

Ans: Summary() function was used to perform high-level EDA of the data frame. The EDA shows that the data is not refined and highly skewed due to the presence of severe outliers. Various data types and categories are misclassified as well for. eg. City_Id and dt_iso etc. The extreme outlier can be seen in Temperature, wind speed and also rainfall attributes, which suggest interference in the sensor signal at the time of accumulation. We can see that many records of rainfall are missing.

```
# 1.c
summary(raw_data)
```

```
##       Date                        City_ID       Temperature       Humidity
##  Min.   :2012-10-01 12:00:00   2643743:49979   Min.   :  0.0   Min.   :  0.0
##  1st Qu.:2014-01-29 19:00:00   2950159:43886   1st Qu.:280.1   1st Qu.: 63.0
##  Median :2015-06-13 22:00:00   2988507:35965   Median :285.7   Median : 80.0
##  Mean   :2015-05-17 14:16:16   3143244:41964   Mean   :285.4   Mean   : 74.4
##  3rd Qu.:2016-10-04 05:00:00   3169070:40006   3rd Qu.:290.7   3rd Qu.: 88.0
##  Max.   :2017-11-13 00:00:00   6458923:42115   Max.   :315.8   Max.   :203.0
##
##    Wind_Speed        Rain_1hr           Weather
##  Min.   : 0.000   Min.   :    0.00   800    :82583
##  1st Qu.: 1.000   1st Qu.:    0.25   803    :33299
##  Median : 3.000   Median :    0.51   500    :24987
##  Mean   : 3.129   Mean   :   37.16   801    :23649
##  3rd Qu.: 4.000   3rd Qu.:    1.27   802    :18955
##  Max.   :58.000   Max.   :41050.50   701    :15838
##                   NA's   :237300     (Other):54604
```

1:d Find the mean and standard deviation of temperature and wind speed using all of the raw data.

Ans: I used the mean() and sd() function to calculate the respective quantities. These were used in an lapply() loop mechanism instead of calculating individually. Base R offers efficient methods of performing baisc mathematic operations.

```
# 1.d
qty <- list(raw_data$Temperature,raw_data$Wind_Speed)
avg <- lapply(qty, mean)
print("Mean")
```

```
## [1] "Mean"
```

```
unlist(avg)
```

```
## [1] 285.379484   3.128701
```

```
sdv <- lapply(qty, sd)
print("Standard Deviation")
```

```
## [1] "Standard Deviation"
```

```
unlist(sdv)
```

```
## [1] 8.124850 2.138859
```

1.e: Calculate again the means of the two variables but after ‘trimming’ 2% of data from each end. Describe the results and what they tell you about data cleaning needed.

Ans: The trim parameter of the mean() function is used to exclude some proportion of the data from either ends. I used (trim = 0.02) to trim 2% of data from each end. The result of this trimming can be seen from the change in the mean values. The change in the average wind speed is more pronounced than compared to temperature.

```
# 1.e
qty <- list(raw_data$Temperature,raw_data$Wind_Speed)
avg <- lapply(qty,mean, trim=0.02)
unlist(avg)
```

```
## [1] 285.437303   3.049044
```

1.f: Consider the following table that matches the ids and names of the cities: i) Create a dataframe named city_df that contains the information provided in the table above. Name the columns “Id” and “city_name” as shown.

Ans: A new matrix is created consisting the details as provided - the city id and the city name.

```
# 1.f(i)
city_factors <- c(levels(raw_data$City_ID),1111111)
cities <- c("London","Berlin","Paris","Oslo","Rome","Lisbon","Pittsburgh")
city_df = data.frame("Id"=city_factors,"City_Name"=cities)
city_df
```

```
##        Id  City_Name
## 1 2643743     London
## 2 2950159     Berlin
## 3 2988507      Paris
## 4 3143244       Oslo
## 5 3169070       Rome
## 6 6458923     Lisbon
## 7 1111111 Pittsburgh
```

ii. Join raw_data and city_df data frames to create a data frame called joined_data that has an additional column with the city name associated with each observation (column must be called “city_name”).

Ans: The raw_data and the matrix created above were combined or merged together to generated the required data frame. Inner joint was performed based on the city_id column name.

```
# 1.f(ii)
city_df = data.frame("City_ID"=city_factors,"City_Name"=cities)
joined_data<-merge(raw_data,city_df, by = "City_ID", all =TRUE)
head(joined_data)
```

```
##   City_ID                Date Temperature Humidity Wind_Speed Rain_1hr Weather
## 1 2643743 2012-10-10 16:00:00      286.10       66          4       NA     803
## 2 2643743 2012-10-10 17:00:00      285.34       71          4       NA     803
## 3 2643743 2012-10-10 18:00:00      284.48       71          4       NA     803
## 4 2643743 2012-10-10 19:00:00      284.16       91          0        0     800
## 5 2643743 2012-10-10 20:00:00      283.78       76          2       NA     802
## 6 2643743 2012-10-10 21:00:00      283.41       96          0        0     800
##   City_Name
## 1    London
## 2    London
## 3    London
## 4    London
## 5    London
## 6    London
```

Question 2: Data Manipulation and Turning data into information

2.a: Modify the joined_data data frame by removing: (1) all duplicate records from the data frame, (2) any records where the temperature is equal to zero and (3) any records where the humidity is greater than 100. Name the resulting data frame removed_joined_data.

Ans: I used logical operators and unique() to perform the data cleaning requirements. Removed_joined_data was thus created. A total of 20313 record were thus cleared, making the data frame for realistic.

```
# 2.a
joined_data <- unique(joined_data)
removed_joined_data <- subset(joined_data, Temperature != 0 & Humidity <100)
str(removed_joined_data)
```

```
## 'data.frame':    233603 obs. of  8 variables:
##  $ City_ID    : Factor w/ 7 levels "2643743","2950159",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date       : POSIXct, format: "2012-10-10 16:00:00" "2012-10-10 17:00:00" ...
##  $ Temperature: num  286 285 284 284 284 ...
##  $ Humidity   : int  66 71 71 91 76 96 85 85 85 85 ...
##  $ Wind_Speed : int  4 4 4 0 2 0 2 2 2 2 ...
##  $ Rain_1hr   : num  NA NA NA 0 NA 0 NA NA NA NA ...
##  $ Weather    : Factor w/ 43 levels "200","201","202",..: 42 42 42 39 41 39 39 39 39 39 ...
##  $ City_Name  : chr  "London" "London" "London" "London" ...
```

2.b: What proportion of the observations in removed_joined_data (overall, not at city level) is the temperature below 0 degrees Celsius AND the wind speed is greater than 5?

Ans: Subset of data was created from original data frame which included the required filters. The number of rows of this filtered subset was found using “nrow” and divided by nrow of original data frame to get the proportion value in percentage i.e. 0.245 %

```
# 2.b
cold_wildy <- subset(removed_joined_data, Temperature < 273.15 & Wind_Speed > 5)
val <- nrow(cold_wildy)
val2 <- nrow(removed_joined_data)
proportion <- (val/val2)*100
proportion
```

```
## [1] 0.2448599
```

2.c: Make PivotTable-like summaries (for each city, and showing city names) for the following using removed_joined_data: i) Average temperature and wind speed ii) Standard deviation of the temperature and wind speed ii) Minimum and maximum temperature and wind speed iv) Frequency of ‘clear’ (weather ID=800) conditions

Ans: Aggregate() function is used to generate pivot table-like summaries in Base R. In addition to that, basic statistic criterion such as mean, standard deviation, minimum, maximum were utilized. Min and Max were found separately and then merged togethe rin one table. Table() command was used to get the frequency distribution i.e. count of “Clear” weather condition.

```
# 2.c(i)
temp <- data.frame(Temperature=removed_joined_data$Temperature,Wind_Speed=removed_joined_data$Wind_Speed)
temp2 <- list(City_Name = removed_joined_data$City_Name)
avg_temp_wind <- aggregate(x=temp,by=temp2, FUN=mean)
avg_temp_wind
```

```
##   City_Name Temperature Wind_Speed
## 1    Berlin    284.4252   3.275068
## 2    Lisbon    289.2962   3.873132
## 3    London    284.6329   3.618207
## 4      Oslo    280.4589   2.384263
## 5     Paris    285.2311   2.958942
## 6      Rome    290.4557   2.618917
```

```
# 2.c(ii)
sd_temp_wind <-aggregate(x=temp,by=temp2, FUN=sd,na.rm=TRUE)
sd_temp_wind
```

```
##   City_Name Temperature Wind_Speed
## 1    Berlin    8.120269   2.080968
## 2    Lisbon    6.068757   2.240174
## 3    London    5.929117   2.125766
## 4      Oslo    8.680038   1.858452
## 5     Paris    7.180012   1.885770
## 6      Rome    7.104061   1.991264
```

```
# 2.c(iii)
max_temp_wind <- aggregate(x=temp,by=temp2, FUN = max)
min_temp_wind <- aggregate(x=temp,by=temp2, FUN = min)
min_max_temp_wind <- merge(max_temp_wind,min_temp_wind,by="City_Name")
colnames(min_max_temp_wind)<-c("City_Name","Max_Temp","Max_Wind","Min_Temp","Min_Wind")
min_max_temp_wind
```

```
##   City_Name Max_Temp Max_Wind Min_Temp Min_Wind
## 1    Berlin   311.50       15  259.900        0
## 2    Lisbon   315.81       16  268.150        0
## 3    London   306.74       18  266.563        0
## 4      Oslo   306.21       13  251.150        0
## 5     Paris   311.21       58  262.608        0
## 6      Rome   310.55       17  266.370        0
```

```
# 2.c(iv)
temp4 <- subset(removed_joined_data,Weather=="800",na.rm=TRUE)
count_clear <- data.frame(table(temp4$City_Name))
colnames(count_clear)<-c("City_Names","Clear Weather Counts")
count_clear
```

```
##   City_Names Clear Weather Counts
## 1     Berlin                14548
## 2     Lisbon                12733
## 3     London                 8918
## 4       Oslo                14536
## 5      Paris                11470
## 6       Rome                17059
```

Question 3 Data visualization in Base R:

3.a: Assume you want to understand whether there are data gaps in the hourly records for each city (e.g., days/weeks/months with no records due to sensor failure). Make a new variable that tracks ‘time between consecutive records’ for removed_joined_data AND for each city create a table of the ‘time between’ to try to find whether any or all of the cities have gaps. Discuss whether they are generally day/week/month long type gaps.

Ans: I used the attribute “dt_iso” to get information regarding the date and time of the recordings. In addition difftime() function was used to get the difference between two consecutive recording in hours. Each city subset was created to gather information regarding the data accumulation.

```
#London
london <- subset(removed_joined_data,City_Name == "London")
london2 <- subset(joined_data,City_Name == "London")

iter = c(1:nrow(london)-1)
for (i in iter){
  london$Time_Gap[i+1] <- as.numeric(difftime(london$Date[i+1],london$Date[i],tz="UTC",units="hours"))
}
london$Time_Gap <- as.factor(london$Time_Gap)
op_london <- data.frame(table(london$Time_Gap))
colnames(op_london) <- c("Time Difference (hrs)","Frequency")
op_london
```

```
##    Time Difference (hrs) Frequency
## 1                      0      8719
## 2                      1     34733
## 3                      2      2925
## 4                      3       294
## 5                      4       167
## 6                      5        54
## 7                      6        23
## 8                      7        22
## 9                      8        15
## 10                     9        10
## 11                    10         5
## 12                    11         3
## 13                    12         2
## 14                    13         4
## 15                    14         1
## 16                    15         1
## 17                    17         3
## 18                    20         1
## 19                    24         2
## 20                    25         1
## 21                    30         1
## 22                    40         1
## 23                    47         1
## 24                    74         1
## 25                   211         1
## 26                   215         1
## 27                   244         1
## 28                   526         1
```

```
#Berlin
berlin <- subset(removed_joined_data,City_Name == "Berlin")

iter = c(1:nrow(berlin)-1)
for (i in iter){
  berlin$Time_Gap[i+1] <- as.numeric(difftime(berlin$Date[i+1],berlin$Date[i],tz="UTC",units="hours"))
}

berlin$Time_Gap <- as.factor(berlin$Time_Gap)
op_berlin <- data.frame(table(berlin$Time_Gap))
colnames(op_berlin) <- c("Time Difference (hrs)","Frequency")
op_berlin
```

```
##    Time Difference (hrs) Frequency
## 1                      0      2925
## 2                      1     33656
## 3                      2      2890
## 4                      3       347
## 5                      4       168
## 6                      5        77
## 7                      6        51
## 8                      7        37
## 9                      8        31
## 10                     9        16
## 11                    10        10
## 12                    11         6
## 13                    12        12
## 14                    13         9
## 15                    14         4
## 16                    15         5
## 17                    16         6
## 18                    17         2
## 19                    19         1
## 20                    22         1
## 21                    24         2
## 22                    25         2
## 23                    26         1
## 24                    30         1
## 25                    32         1
## 26                    33         1
## 27                    43         1
## 28                    47         1
## 29                   118         1
## 30                   211         1
## 31                   214         1
## 32                   245         1
## 33                   526         1
```

```
#Paris
paris <- subset(removed_joined_data,City_Name == "Paris")

iter = c(1:nrow(paris)-1)
paris$Time_Gap <- 0
for (i in iter){
  paris$Time_Gap[i] <- (as.numeric(difftime(paris$Date[i+1],paris$Date[i],tz="UTC",units="hours")))
}
paris$Time_Gap <- as.factor(paris$Time_Gap)
op_paris <- data.frame(table(paris$Time_Gap))
colnames(op_paris) <- c("Time Difference (hrs)","Frequency")
op_paris
```

```
##    Time Difference (hrs) Frequency
## 1                      0      2858
## 2                      1     28434
## 3                      2      2489
## 4                      3       309
## 5                      4       126
## 6                      5        55
## 7                      6        43
## 8                      7        22
## 9                      8        20
## 10                     9        15
## 11                    10        19
## 12                    11         6
## 13                    12         3
## 14                    13         3
## 15                    14         7
## 16                    15         3
## 17                    16         1
## 18                    17         1
## 19                    18         1
## 20                    19         1
## 21                    21         2
## 22                    22         4
## 23                    24         4
## 24                    25         2
## 25                    30         2
## 26                    33         1
## 27                    34         1
## 28                    47         1
## 29                    65         1
## 30                    93         1
## 31                   117         1
## 32                   118         1
## 33                   244         1
## 34                  7387         1
```

```
#Oslo
oslo <- subset(removed_joined_data,City_Name == "Oslo")

iter = c(1:nrow(oslo)-1)
for (i in iter){
  oslo$Time_Gap[i+1] <- as.numeric(difftime(oslo$Date[i+1],oslo$Date[i],tz="UTC",units="hours"))
}
oslo$Time_Gap <- as.factor(oslo$Time_Gap)
op_oslo <- data.frame(table(oslo$Time_Gap))
colnames(op_oslo) <- c("Time Difference (hrs)","Frequency")
op_oslo
```

```
##    Time Difference (hrs) Frequency
## 1                      0      1442
## 2                      1     32626
## 3                      2      2990
## 4                      3       390
## 5                      4       228
## 6                      5        84
## 7                      6        50
## 8                      7        34
## 9                      8        26
## 10                     9        13
## 11                    10        20
## 12                    11         8
## 13                    12         6
## 14                    13        10
## 15                    14         4
## 16                    15         4
## 17                    16         3
## 18                    17         4
## 19                    18         1
## 20                    19         1
## 21                    20         1
## 22                    21         3
## 23                    22         3
## 24                    23         1
## 25                    24         1
## 26                    25         2
## 27                    26         3
## 28                    29         1
## 29                    30         1
## 30                    31         2
## 31                    35         2
## 32                    38         1
## 33                    47         1
## 34                    62         1
## 35                    98         1
## 36                   119         1
## 37                   211         1
## 38                   214         1
## 39                   283         1
## 40                   526         1
```

```
#Rome
rome <- subset(removed_joined_data,City_Name == "Rome")

iter = c(1:nrow(rome)-1)
for (i in iter){
  rome$Time_Gap[i] <- as.numeric(difftime(rome$Date[i+1],rome$Date[i],tz="UTC",units="hours"))
}
rome$Time_Gap <- as.factor(rome$Time_Gap)
op_rome <- data.frame(table(rome$Time_Gap))
colnames(op_rome) <- c("Time Difference (hrs)","Frequency")
op_rome
```

```
##    Time Difference (hrs) Frequency
## 1                      0      2292
## 2                      1     30368
## 3                      2      3062
## 4                      3       302
## 5                      4       113
## 6                      5        76
## 7                      6        32
## 8                      7        24
## 9                      8        28
## 10                     9        16
## 11                    10        12
## 12                    11         5
## 13                    12         6
## 14                    13         9
## 15                    14         7
## 16                    15         1
## 17                    16         6
## 18                    19         2
## 19                    21         1
## 20                    22         1
## 21                    24         1
## 22                    25         2
## 23                    26         1
## 24                    29         1
## 25                    30         1
## 26                    43         2
## 27                    44         2
## 28                    47         1
## 29                    55         1
## 30                    67         2
## 31                    85         1
## 32                    94         1
## 33                   156         1
## 34                   158         1
## 35                   167         1
## 36                   174         1
## 37                   212         1
## 38                   214         1
## 39                   248         1
## 40                   429         1
## 41                   511         1
## 42                   526         1
## 43                   741         1
## 44                   949         1
```

```
#Lisbon
lisbon <- subset(removed_joined_data,City_Name == "Lisbon")

iter = c(1:nrow(lisbon)-1)
for (i in iter){
  lisbon$Time_Gap[i] <- as.numeric(difftime(lisbon$Date[i+1],lisbon$Date[i],tz="UTC",units="hours"))
}
lisbon$Time_Gap <- as.factor(lisbon$Time_Gap)
op_lisbon <- data.frame(table(lisbon$Time_Gap))
colnames(op_lisbon) <- c("Time Difference (hrs)","Frequency")
op_lisbon
```

```
##    Time Difference (hrs) Frequency
## 1                      0      2016
## 2                      1     31545
## 3                      2      3187
## 4                      3       372
## 5                      4       153
## 6                      5        77
## 7                      6        43
## 8                      7        28
## 9                      8        17
## 10                     9        17
## 11                    10        11
## 12                    11         5
## 13                    12        13
## 14                    13        10
## 15                    14         3
## 16                    15         2
## 17                    16         4
## 18                    17         1
## 19                    19         3
## 20                    20         3
## 21                    21         1
## 22                    24         1
## 23                    30         2
## 24                    32         1
## 25                    37         1
## 26                    40         1
## 27                    41         1
## 28                    43         1
## 29                    48         1
## 30                    57         1
## 31                    64         1
## 32                    66         1
## 33                    80         1
## 34                    92         2
## 35                   112         1
## 36                   118         1
## 37                   140         1
## 38                   211         1
## 39                   214         1
## 40                   227         1
## 41                   246         1
## 42                   607         1
## 43                   676         1
```

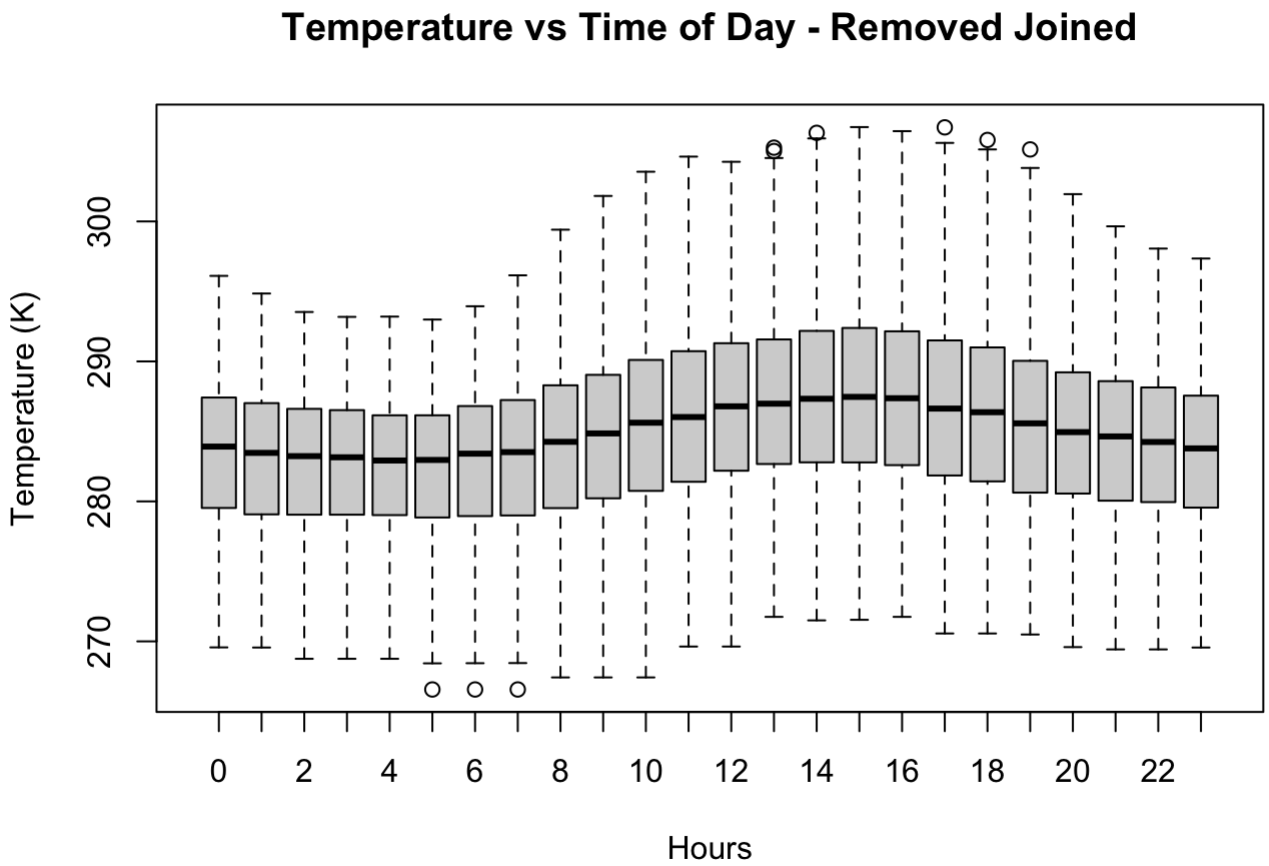
3.b: For each of the following, create boxplots for the hourly data for London, one using joined_data and the other using removed_joined_data: (i) temperature (variable temp) (ii) hourly rainfall (rain_1h).

Make sure the plots are properly formatted and labeled (don’t worry about color!). Discuss outliers identified by plots in terms of number of data points and how much the outliers differ.

Ans: Time information was stored in hours for each record by using strptime() function. After storing the hours as factors, boxplot was created to get the distribution of the temperature as a function of hour of day. Only single boxplot was generated as per the new update on canvas.

The temperature plot shows that the temperature is highest during the afternoon hours, which is very much intuitive. We can also see the outlier as the circular points.

```
t <- strptime(london$Date, format= "%Y-%m-%d %H:%M:%S")
t <- as.numeric(format(t, "%H"))
london$Hours <- as.factor(t)
boxplot(london$Temperature~london$Hours,data=london, main="Temperature vs Time of Day - Removed Joined",
   xlab="Hours", ylab="Temperature (K)")
```



The rain plot shows that there exists severe skewness in the data. The plot is limited to 500 mm of rainfall to avoid extreme outliers from skewing the plot and resulting in very limited information. We can see that the majorly, rainfall is evenly distributed throughout the day.

```
london3 <- subset(london2,Rain_1hr<500)
t <- strptime(london3$Date, format= "%Y-%m-%d %H:%M:%S")
t <- as.numeric(format(t, "%H"))
london3$Hours <- as.factor(t)
boxplot(london3$Rain_1hr~london3$Hours,data=london3, main="Rain_1hr vs Time of Day - Removed Joined",
   xlab="Hours", ylab="Precipitation (mm)")
```