STATS 369 Assignment 1 2019

Pavan Mani (6288156)

06/08/2019

```
knitr::opts_chunk$set(echo = TRUE)

# Load relevant libraries
library(tidyverse)

## -- Attaching packages ---------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(dplyr)
library(ggplot2)
library(s20x)

# getSeason is a user-defined R function that returns the season a month belo
ngs to. It takes an abbreviation of the month e.g. 'Jan' or 'Apr' as it's inp
ut and returns the corresponding season e.g. 'Summer' or 'Autumn'.
# https://www.newzealand.com/int/seasons-in-new-zealand/
# Author: Pavan Mani
getSeason <- function(x){
  if (x == 'Sep'|x == 'Oct'|x == 'Nov'){
    return('Spring')
  } else if (x== 'Dec'|x == 'Jan'|x == 'Feb') {
    return('Summer')
  } else if (x == 'Mar'|x == 'Apr'|x == 'May'){
    return('Autumn')
  } else if (x == 'Jun'|x == 'Jul'|x == 'Aug'){
    return('Winter')
  } else {
    return('')
  }
}
```

Question 1

When observing and comparing the 2016, 2017 and 2018 cycle count datasets, one obstacle we will encounter is that in some instances the location variable headings differ even though they are counting the same observation e.g. Curran St Total (2016), Curran Street Total (2017) and Curran Street Total Cyclists (2018). Furthermore, there is additional locational data for the year 2018 not found in years 2016 and 2017 i.e. the dimensions of the datasets are different.

Question 2

```
# Read data
cyclist_data_2016 <- read.csv("C:/Users/pavan/Downloads/dailyakldcyclecountdata2016_updated.csv")

cyclist_data_2017 <- read.csv("C:/Users/pavan/Downloads/dailyakldcyclecountdata2017_1.csv")

cyclist_data_2018 <- read.csv("~/STATS 369/dailyakldcyclecountdata2018.csv")

# Remove data not needed for analysis (i.e. last row in 2018 cycle data)
cyclist_data_2018 <- cyclist_data_2018[-c(366), ]

# Extract date information and total sum of columns for each row from 2016, 2017 and 2018 cycle data.
cyclist_data_2016 <- cyclist_data_2016 %>%
  mutate(total_cyclists = rowSums(cyclist_data_2016[ ,2:33], na.rm = TRUE)) %>%
  subset(select = c(1,34))

cyclist_data_2017 <- cyclist_data_2017 %>%
  mutate(total_cyclists = rowSums(cyclist_data_2017[ ,2:40], na.rm = TRUE)) %>%
  subset(select = c(1,41))

cyclist_data_2018 <- cyclist_data_2018 %>%
  mutate(total_cyclists = rowSums(cyclist_data_2018[ ,2:44], na.rm = TRUE)) %>%
  subset(select = c(1,45))

# Combine datasets containing all dates from Jan 2016 to Dec 2018 and total cyclists observed for each date.
cyclist_data_merge_1 <- rbind(cyclist_data_2016, cyclist_data_2017)

cyclist_data <- rbind(cyclist_data_merge_1, cyclist_data_2018)

# Check size of dataset (for later use when indexing or checking we have the
```

```
right number of rows for the rainfall data)
dim(cyclist_data)

## [1] 1096    2

# read rainfall data
rain_2016_2017 <- read.csv("C:/Users/pavan/Documents/STATS 369/rain2016-17.tx
t")

rain_2018 <- read.csv("C:/Users/pavan/Documents/STATS 369/rain2018.txt")

# Merge rainfall datasets vertically
rain_df <- rbind(rain_2016_2017, rain_2018)
dim(rain_df)

## [1] 52394     6

# Remove row corresponding to 2019
rain_df <- rain_df[-c(52394), ]
# Check size of rainfall dataset to see it contains the same number of rows a
s the cyclist dataset
dim(rain_df)

## [1] 52393     6

# Calculate amount of total rainfall for each day
amount_rain_per_day <- rain_df %>%
  group_by(Date.NZST.) %>%
  summarise(rainfall_mm = sum(Amount.mm.))
# Check size of dataset to ensure number of rows matches cyclist data
dim(amount_rain_per_day)

## [1] 1096    2

# Merge cyclist and rain datasets horizontally
df <- merge(cyclist_data,amount_rain_per_day,by ="row.names")
dim(df)

## [1] 1096    5

# Order dataset from oldest date to most recent
cycle_rain_df <- with(df, df[order(df$Date.NZST.), ])
# Get Date, Total Cyclists, Date (from rainfall dataset) and amount of rainfu
ll columns
cycle_rain_df <- subset(cycle_rain_df, select = c(2,3,4,5))
cycle_rain_df %>% head()

##               Date total_cyclists Date.NZST. rainfall_mm
## 1   Fri 1 Jan 2016           1299   20160101        40.5
## 209 Sat 2 Jan 2016           1030   20160102        38.3
## 320 Sun 3 Jan 2016           7423   20160103        13.6
## 431 Mon 4 Jan 2016          11956   20160104         0.1
```

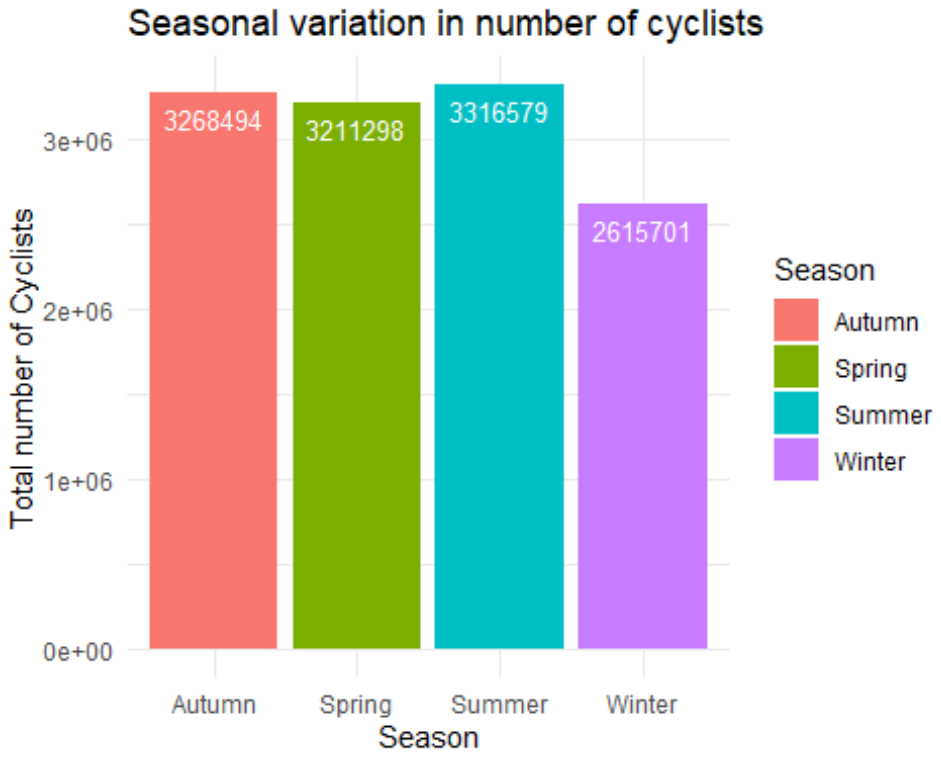
```
## 542 Tue 5 Jan 2016          10167   20160105         0.0
## 653 Wed 6 Jan 2016          10387   20160106         0.0

# Create new columns for Day, Year and Month in dataframe
cycle_rain_df <- cycle_rain_df %>%
  mutate(Day = substring(Date, first = 1, last = 3)) %>%
  mutate(Year = substring(Date, first = nchar(as.character(Date))-4, last = n
char(as.character(Date)))) %>%
  mutate(Month = substring(Date, first = nchar(as.character(Date))-7, last =
nchar(as.character(Date))-5))

# Get size of dataframe
dim(cycle_rain_df)

## [1] 1096    7

# Iterate through rows and assign a season depending on the month data was co
llected
for (i in 1:1096) {
  cycle_rain_df$Season[i] <- getSeason(cycle_rain_df$Month[i])
}

# Convert integer date to date-time format for aesthetic plotting purposes
cycle_rain_df <- cycle_rain_df %>%
  mutate(DATE = as.Date(as.character(Date.NZST.), format = '%Y%m%d'))
cycle_rain_df %>% head()

##             Date total_cyclists Date.NZST. rainfall_mm Day  Year Month
## 1 Fri 1 Jan 2016           1299   20160101        40.5 Fri  2016   Jan
## 2 Sat 2 Jan 2016           1030   20160102        38.3 Sat  2016   Jan
## 3 Sun 3 Jan 2016           7423   20160103        13.6 Sun  2016   Jan
## 4 Mon 4 Jan 2016          11956   20160104         0.1 Mon  2016   Jan
## 5 Tue 5 Jan 2016          10167   20160105         0.0 Tue  2016   Jan
## 6 Wed 6 Jan 2016          10387   20160106         0.0 Wed  2016   Jan
##   Season       DATE
## 1 Summer 2016-01-01
## 2 Summer 2016-01-02
## 3 Summer 2016-01-03
## 4 Summer 2016-01-04
## 5 Summer 2016-01-05
## 6 Summer 2016-01-06

# Store date, total cyclists count and rainfall amount columns in separate da
taframe.
date_cycle_rain <- subset(cycle_rain_df, select = c(1,2,4))
date_cycle_rain %>% head()

##             Date total_cyclists rainfall_mm
## 1 Fri 1 Jan 2016           1299        40.5
## 2 Sat 2 Jan 2016           1030        38.3
## 3 Sun 3 Jan 2016           7423        13.6
```

```
## 4 Mon 4 Jan 2016          11956         0.1
## 5 Tue 5 Jan 2016          10167         0.0
## 6 Wed 6 Jan 2016          10387         0.0
```

The total number of cyclists counted for each day along with the corresponding amount of rainfall for that day are now stored in date_cycle_rain.df.

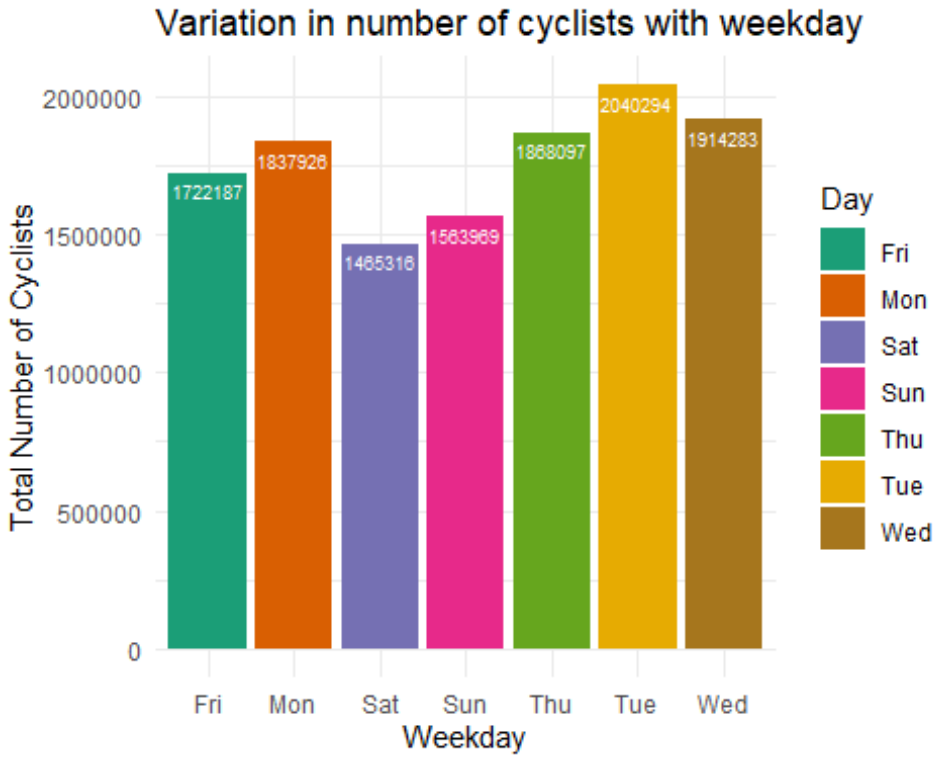
Question 3

```
# Group data by season and compute total number of observed cyclists.
cyclists_by_season <- cycle_rain_df %>%
  group_by(Season) %>%
  summarise(Total_Cyclists = sum(total_cyclists))
# Generate bar graph showing total number of cyclists observed in each season
ggplot(data = cyclists_by_season, aes(x = Season, y = Total_Cyclists, fill =S
eason))+geom_bar(stat = "identity")+theme_minimal()+labs(x = "Season", y = "T
otal number of Cyclists", title='Seasonal variation in number of cyclists')+g
eom_text(aes(label=Total_Cyclists),vjust=1.6,color="white" ,size = 3.5)
```

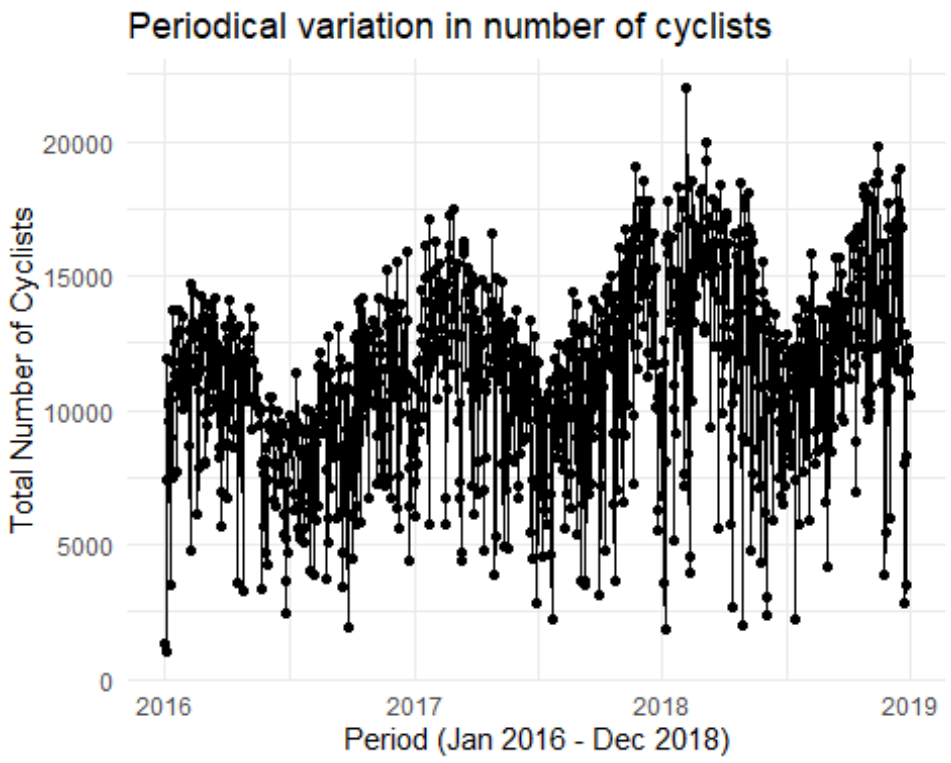



```
# Group by day in the week and compute total number of cyclists observed
cyclists_by_day <- cycle_rain_df %>%
  group_by(Day) %>%
  summarise(Total_Cyclists = sum(total_cyclists))

# Generate bar chart showing number of cyclists observed for each day in the
week
```

```
ggplot(data = cyclists_by_day, aes(x = Day, y = Total_Cyclists, fill =Day)) +
geom_bar(stat = "identity")+theme_minimal()+scale_fill_brewer(palette="Dark2"
)+labs(x='Weekday', y = 'Total Number of Cyclists',title = 'Variation in numb
er of cyclists with weekday')+geom_text(aes(label=Total_Cyclists),vjust=1.6,c
olor="white" ,size = 2.5)
```

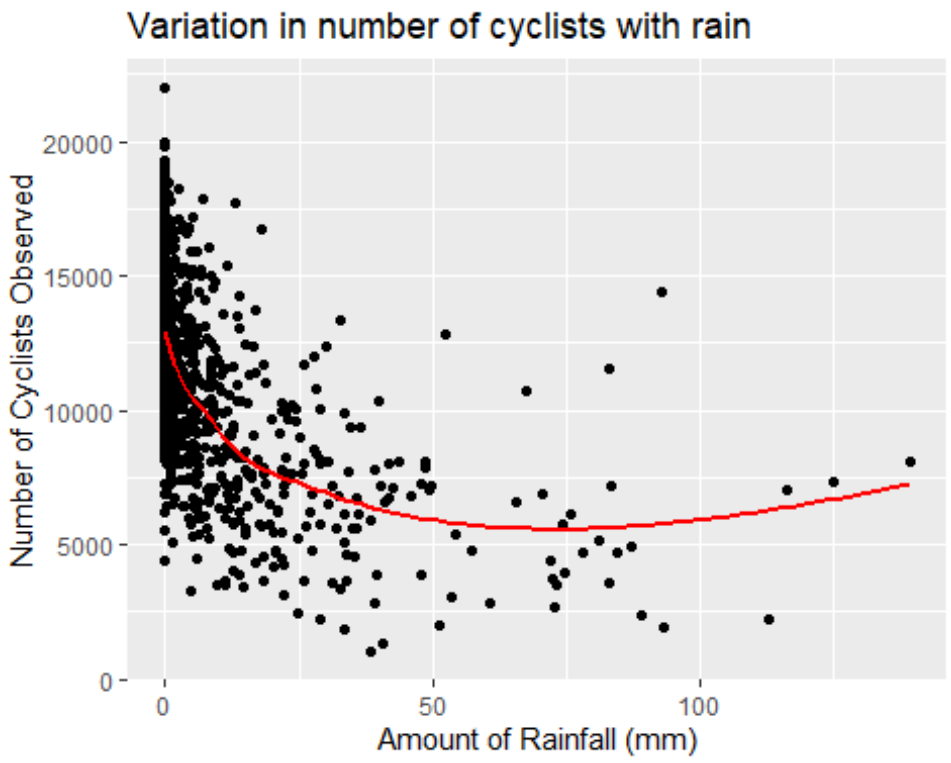



```
# Generate line plot of variation in observed cyclists with time
ggplot(data = cycle_rain_df, aes(x = DATE, y = total_cyclists))+geom_point()+
geom_line()+labs(x="Period (Jan 2016 - Dec 2018)", y = "Total Number of Cycli
sts",title = 'Periodical variation in number of cyclists')+theme_minimal()
```



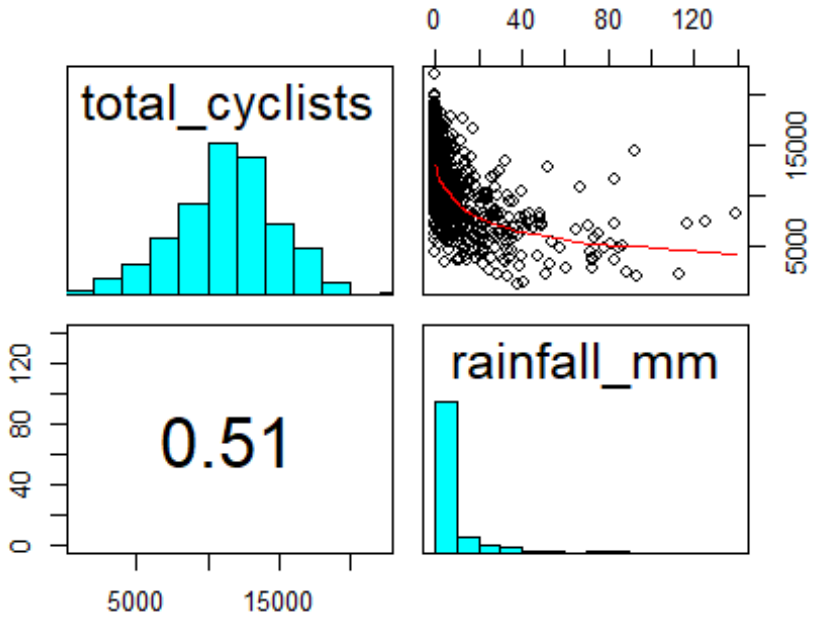

```
# Generate scatter plot of variation in number of cyclists with rainfall
ggplot(data = cycle_rain_df, aes(x = rainfall_mm, y = total_cyclists)) + geom
_point() + labs(x = 'Amount of Rainfall (mm)',y = 'Number of Cyclists Observe
d',title = 'Variation in number of cyclists with rain')+geom_smooth(se=FALSE,
color="red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

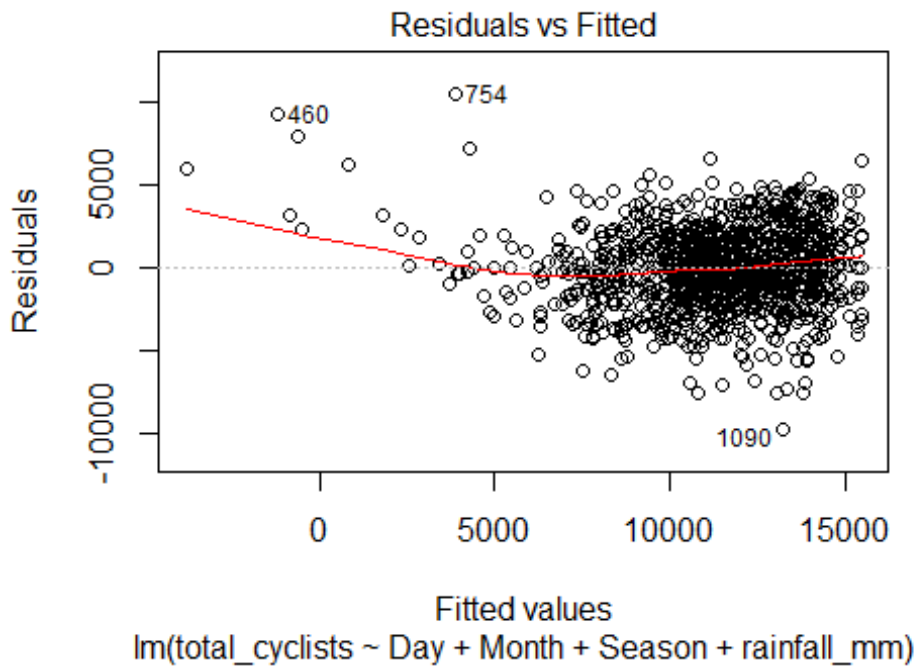


Question 4

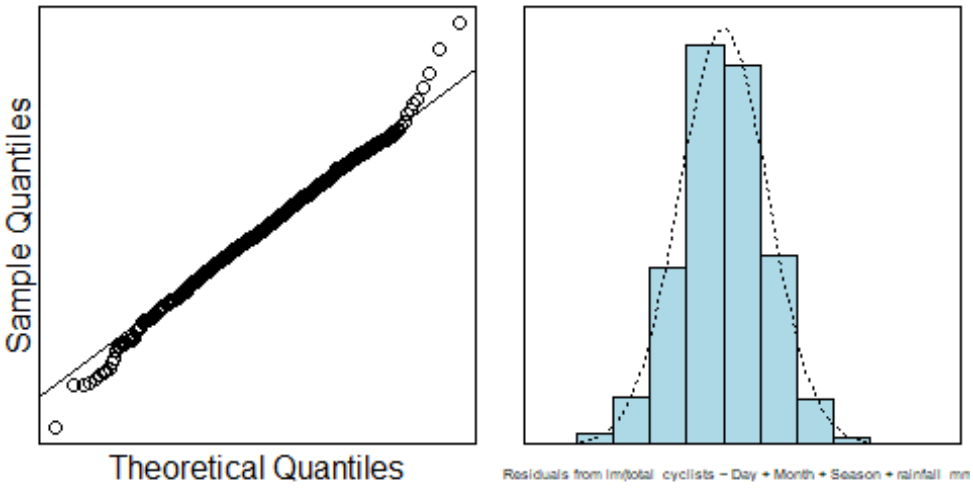
```
# Observe correlation between number of cyclists and amount of rainfall
pairs20x(cycle_rain_df[c(2,4)])
```



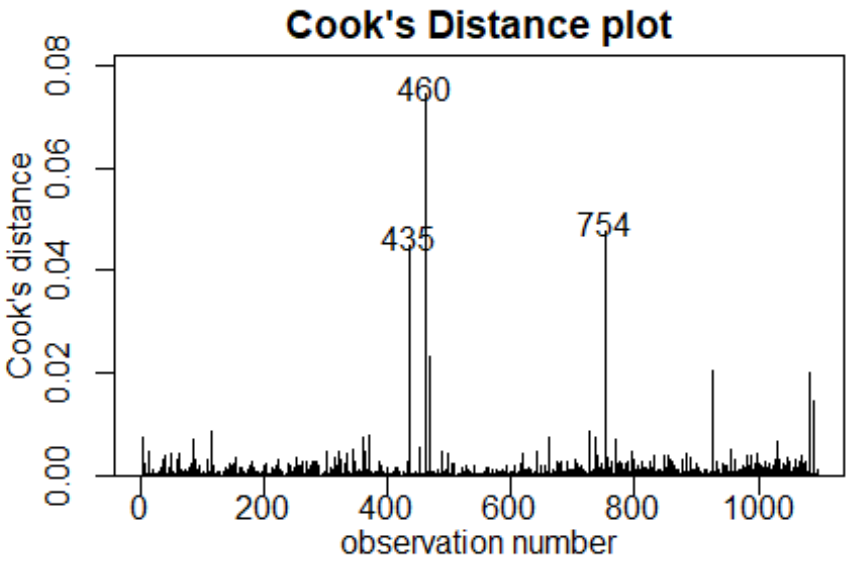

```r
# Fit a linear model
cycle_model.fit <- lm(total_cyclists~Day+Month+Season+rainfall_mm, data = cyc
le_rain_df)

# Equal variability, normality and cook's distance assumption checks.
plot(cycle_model.fit, which = 1)
```




```
normcheck(cycle_model.fit)
```




```
cooks20x(cycle_model.fit)
```



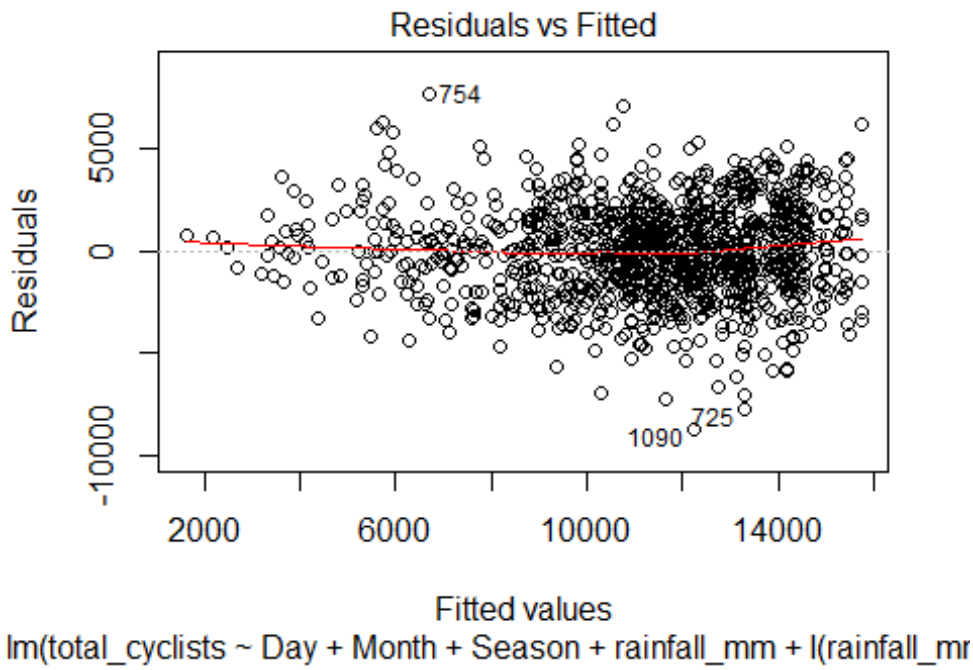

```
# Get summary outputs and confidence intervals
summary(cycle_model.fit)

##
## Call:
## lm(formula = total_cyclists ~ Day + Month + Season + rainfall_mm,
##     data = cycle_rain_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9684  -1505    -20   1593  10519
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12142.315    315.447  38.492  < 2e-16 ***
## DayMon         705.642    274.866   2.567  0.01039 *
## DaySat       -1512.482    274.809  -5.504 4.65e-08 ***
## DaySun        -853.306    274.896  -3.104  0.00196 **
## DayThu        1174.612    275.323   4.266 2.16e-05 ***
## DayTue        2123.502    275.301   7.713 2.78e-14 ***
## DayWed        1575.794    275.547   5.719 1.39e-08 ***
## MonthAug     -1877.139    360.209  -5.211 2.25e-07 ***
## MonthDec       164.862    360.557   0.457  0.64759
## MonthFeb      1178.183    368.203   3.200  0.00142 **
## MonthJan      -119.863    360.013  -0.333  0.73924
## MonthJul     -2576.823    359.919  -7.159 1.50e-12 ***
## MonthJun     -2252.928    362.926  -6.208 7.66e-10 ***
```

```
## MonthMar      1089.242    360.034   3.025  0.00254 **
## MonthMay      -716.164    360.139  -1.989  0.04700 *
## MonthNov       871.697    363.614   2.397  0.01669 *
## MonthOct      -223.616    360.562  -0.620  0.53527
## MonthSep     -1436.063    362.990  -3.956 8.11e-05 ***
## SeasonSpring        NA         NA      NA       NA
## SeasonSummer        NA         NA      NA       NA
## SeasonWinter        NA         NA      NA       NA
## rainfall_mm   -110.854      4.747 -23.352  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2434 on 1077 degrees of freedom
## Multiple R-squared:  0.5134, Adjusted R-squared:  0.5053
## F-statistic: 63.13 on 18 and 1077 DF,  p-value: < 2.2e-16
```

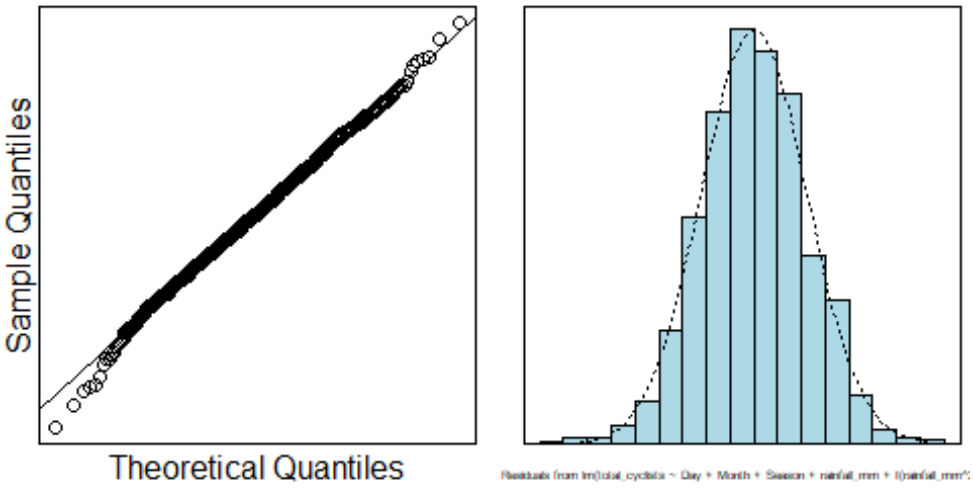
```
confint(cycle_model.fit)
```

```
##                   2.5 %       97.5 %
## (Intercept)  11523.3552 12761.275800
## DayMon         166.3073  1244.976128
## DaySat       -2051.7024  -973.260862
## DaySun       -1392.6980  -313.913838
## DayThu         634.3813  1714.843125
## DayTue        1583.3149  2663.690051
## DayWed        1035.1233  2116.464585
## MonthAug     -2583.9296 -1170.348104
## MonthDec      -542.6127   872.335993
## MonthFeb       455.7072  1900.659114
## MonthJan      -826.2691   586.544069
## MonthJul     -3283.0445 -1870.601516
## MonthJun     -2965.0514 -1540.805070
## MonthMar       382.7936  1795.690606
## MonthMay     -1422.8176    -9.510053
## MonthNov       158.2251  1585.169398
## MonthOct      -931.0984   483.867081
## MonthSep     -2148.3106  -723.814965
## SeasonSpring         NA           NA
## SeasonSummer         NA           NA
## SeasonWinter         NA           NA
## rainfall_mm   -120.1683  -101.539503
```

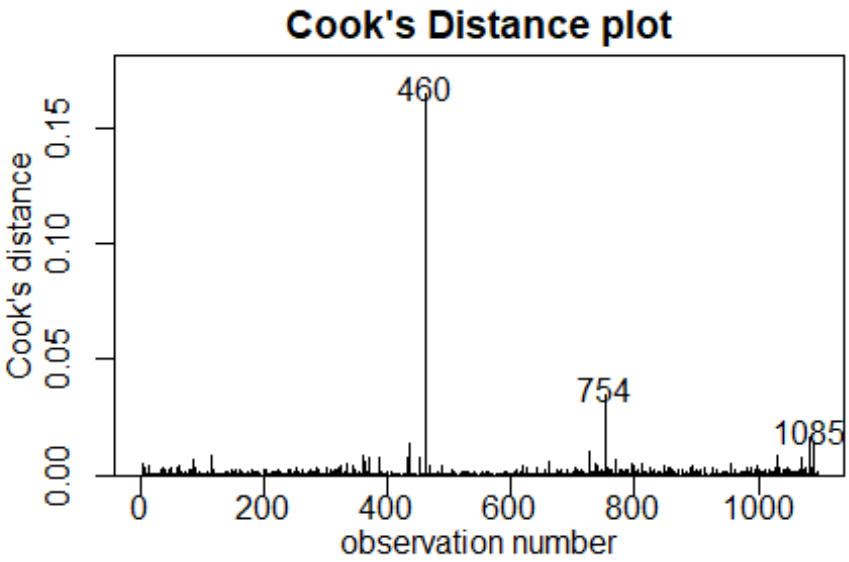
```
# Refit model with a quadratic term added for amount of rainfall variable.
cycle_model.fit1 <- lm(total_cyclists~Day+Month+Season+rainfall_mm+I(rainfall
_mm^2), data = cycle_rain_df)

# Re-check Equal variability, Normality and Cook's Distance assumptions
plot(cycle_model.fit1, which = 1)
```




```
normcheck(cycle_model.fit1)
```




```
cooks20x(cycle_model.fit1)
```




```
# Get summary output and confidence intervals
summary(cycle_model.fit1)

##
## Call:
## lm(formula = total_cyclists ~ Day + Month + Season + rainfall_mm +
##     I(rainfall_mm^2), data = cycle_rain_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8715.7 -1510.0   -28.6  1484.1  7691.0
##
## Coefficients: (3 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      12149.5028   291.7369  41.645  < 2e-16 ***
## DayMon             814.3755   254.3330   3.202 0.001405 **
## DaySat           -1339.4034   254.4741  -5.263 1.71e-07 ***
## DaySun            -833.9363   254.2373  -3.280 0.001071 **
## DayThu            1224.6199   254.6555   4.809 1.73e-06 ***
## DayTue            2107.4330   254.6110   8.277 3.71e-16 ***
## DayWed            1686.5307   254.9672   6.615 5.86e-11 ***
## MonthAug         -1466.0450   334.5156  -4.383 1.29e-05 ***
## MonthDec           339.4156   333.7053   1.017 0.309329
## MonthFeb          1478.8128   341.2504   4.334 1.60e-05 ***
## MonthJan            66.3186   333.2368   0.199 0.842290
## MonthJul         -2032.2598   335.2885  -6.061 1.87e-09 ***
## MonthJun         -1849.8460   336.9658  -5.490 5.02e-08 ***
```

```
## MonthMar          1199.9723   333.0730   3.603 0.000329 ***
## MonthMay          -284.9237   334.5898  -0.852 0.394647
## MonthNov          1155.1752   336.9346   3.428 0.000630 ***
## MonthOct           125.0142   334.4534   0.374 0.708637
## MonthSep         -1038.4951   336.9886  -3.082 0.002111 **
## SeasonSpring             NA         NA      NA       NA
## SeasonSummer             NA         NA      NA       NA
## SeasonWinter             NA         NA      NA       NA
## rainfall_mm       -231.6986     9.9497 -23.287  < 2e-16 ***
## I(rainfall_mm^2)     1.6121     0.1191  13.534  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2251 on 1076 degrees of freedom
## Multiple R-squared:  0.5842, Adjusted R-squared:  0.5769
## F-statistic: 79.57 on 19 and 1076 DF,  p-value: < 2.2e-16
```

```
confint(cycle_model.fit1)
```

```
##                          2.5 %       97.5 %
## (Intercept)      11577.065024 12721.940508
## DayMon             315.330563  1313.420387
## DaySat           -1838.725050  -840.081720
## DaySun           -1332.793417  -335.079174
## DayThu             724.942303  1724.297490
## DayTue            1607.842657  2607.023413
## DayWed            1186.241314  2186.820016
## MonthAug         -2122.421923  -809.668047
## MonthDec          -315.371338   994.202544
## MonthFeb           809.221023  2148.404589
## MonthJan          -587.549045   720.186220
## MonthJul         -2690.153162 -1374.366480
## MonthJun         -2511.030711 -1188.661384
## MonthMar           546.426127  1853.518555
## MonthMay          -941.446188   371.598859
## MonthNov           494.051812  1816.298547
## MonthOct          -531.240570   781.269041
## MonthSep         -1699.724505  -377.265777
## SeasonSpring               NA          NA
## SeasonSummer               NA          NA
## SeasonWinter               NA          NA
## rainfall_mm       -251.221651  -212.175630
## I(rainfall_mm^2)     1.378374     1.845807
```

From the pairwise plot of total cyclists against amount of rainfall, we can observe a non-linear, decreasing relationship between the two variables. When we fitted a linear model, there appeared to be some doubts with the equal variability assumption. In response we refitted a linear model, adding a quadratic term for rainfall, and saw that the equal variability and normality assumptions appeared to be satisifed. We used this new prediction model for our analysis.

Question 5

We are interested in whether rain has a big impact on the number of people cycling in Auckland.

From our plot showing the seasonal variation in the number of cyclists, we can clearly see that less cyclists were observed in the Winter season which is when wet weather is most likely to occur. Furthermore, from our line chart of the variation of cyclists with time there is a noticeable dip occuring around the same time as when winter season occurs in Auckland for each year we have data for. Lastly, when observing the scatter plot of cyclists against rainfall, we can see that the relationship between the two variables is non-linear. The density of scatter is higher at low amounts of rainfall, indicating the presence of more cyclists. Overall, these trends imply that rainfall has an impact on the number of cyclists but they don't tell us if the impact is significant or how big the impact is.

From our fitted linear regression model with an added quadratic term, we can see that rainfall has a significant effect on the number of cyclists (p-value of 2E-16 is much smaller than 0.05). For each 1 mm increase in rainfall, we estimate that the average number of cyclists decreases by between 212 to 251. Our model explains 58% of the variation in the cyclist count data and is therefore not very good for prediction.