Stats 369 A1

Richard Choi

23 July 2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

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

```
library(dplyr)
```

```
# Question 1
# If you try to convert the cycle count data to tidy format (which you don't have to do for this assignm
```

When you try to convert the cycle count data to tidy format it would be computationally inefficient. It would take lot of time to convert it to the tidy format. Also, lot of the names are actually the same place but they are inconsistent. It is really difficult to cover all the patterns so we would need to change the name manually.

```
# Question 2
# Compute the total number of cyclists counted for each day, and a suitable summary of the rainfall for
bike_files<- list.files("/cloud/project/Data",pattern=".csv",full=TRUE)
bike_data<-map(bike_files,read_csv)
```

```
## Rows: 366 Columns: 33

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Date
## dbl (32): Beach Road Cyclists, Carlton Gore Cycle Counter Cyclists, Curran S...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 365 Columns: 40

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Date
## dbl (39): Beach Road Cyclists, Carlton Gore Cycle Counter Cyclists, Curran S...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 366 Columns: 44
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Date
## dbl (43): Archibald Park Cyclists, Beach Road Cyclists, Carlton Gore Cycle C...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
rain_data <- list.files("/cloud/project/Data",pattern="txt",full=TRUE) %>%
  map(~ read_csv(., skip=9))
```

```
## Rows: 35096 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Station, Time(NZST), Freq
## dbl (3): Date(NZST), Amount(mm), Period(Hrs)
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 17310 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Station, Time(NZST), Freq
## dbl (3): Date(NZST), Amount(mm), Period(Hrs)
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# function for removing columns with na
notallNA<-function(x) !all(is.na(x))

bike_data <- map(bike_data, select_if, notallNA)
rain_data <- map(rain_data, select_if, notallNA)
```

```
## Warning: One or more parsing issues, see `problems()` for details

## Warning: One or more parsing issues, see `problems()` for details
```

```
bike_data <- bind_rows(bike_data)
rain_data <- bind_rows(rain_data)

# adding separator in the date
bike_data = bike_data %>%
  mutate(Format_Date = as.Date(bike_data$Date, format="%a %d %b %Y"))

cyclist_amount = bike_data %>%
  rowwise(Format_Date) %>%
  summarise(total_cyclist = sum(c_across(where(is.numeric)), na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'Format_Date'. You can override using the `.groups` argument.
```

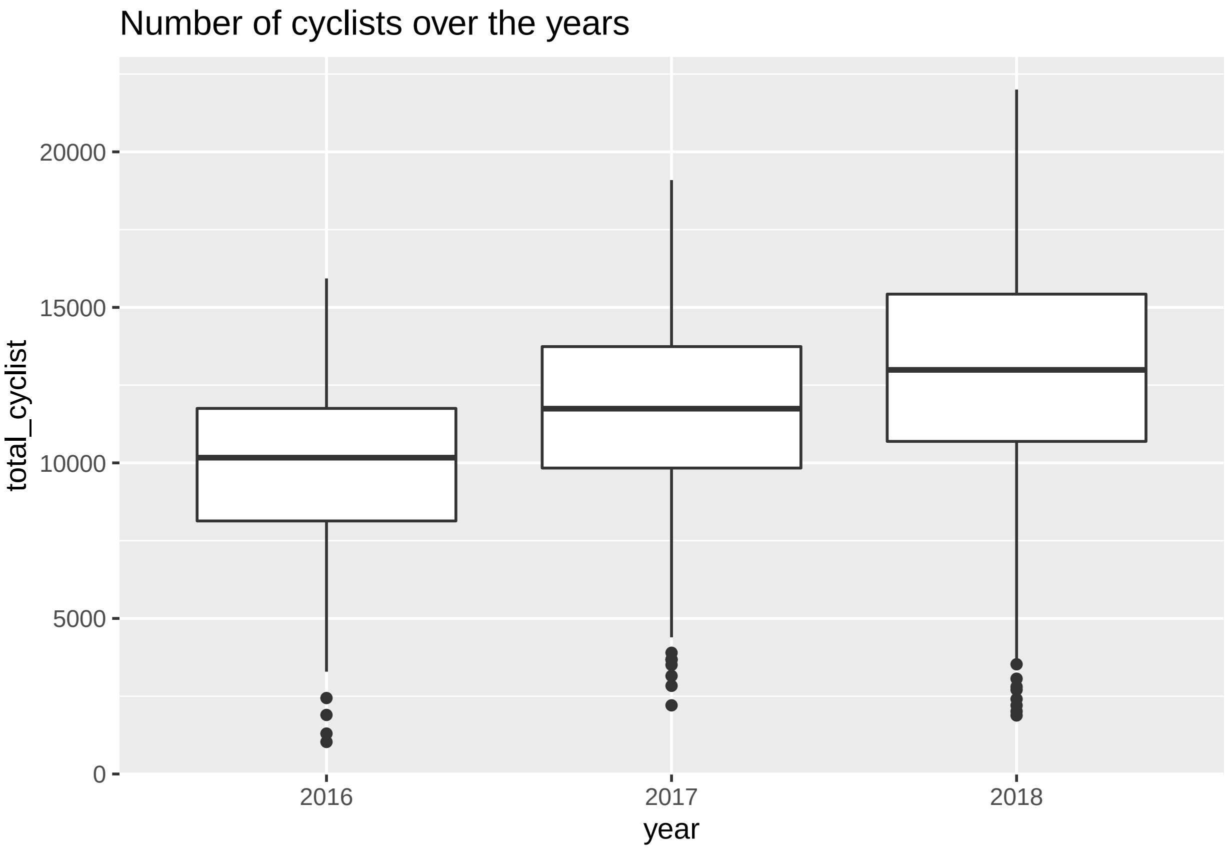
```
lct <- Sys.getlocale("LC_TIME"); Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
# adding separator in the date
rain_data = rain_data %>%
  mutate(Format_Date = as.Date(as.character(rain_data$`Date(NZST)`), "%Y%m%d"))

rain_amount = rain_data %>%
  group_by(Format_Date = Format_Date) %>%
  summarise(Amount = sum(`Amount(mm)`))

both = inner_join(cyclist_amount, rain_amount, by="Format_Date")
both
```

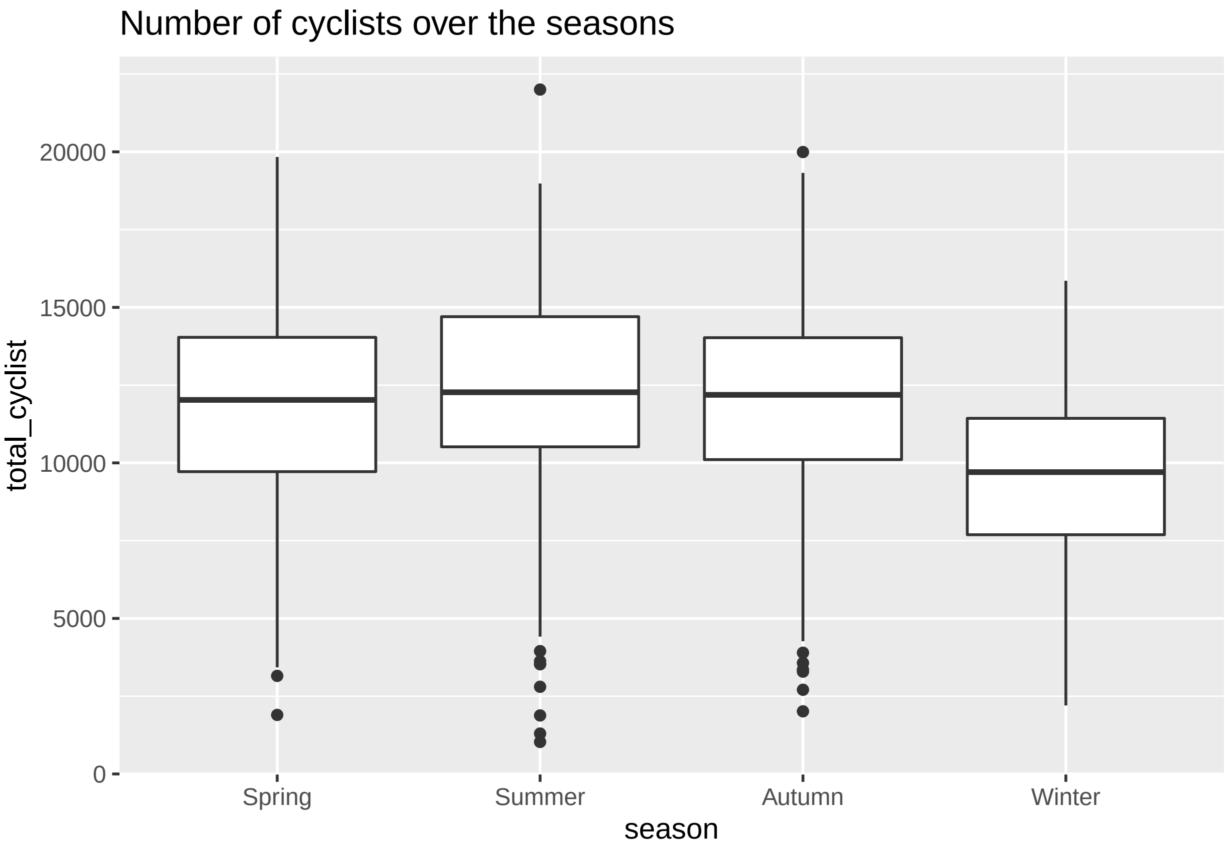
```
## # A tibble: 1,097 x 3
## # Groups:   Format_Date [1,097]
##    Format_Date total_cyclist Amount
##    <date>              <dbl>  <dbl>
##  1 2016-01-01           1299   40.5
##  2 2016-01-02           1030   38.3
##  3 2016-01-03           7423   13.6
##  4 2016-01-04          11956    0.1
##  5 2016-01-05          10167    0
##  6 2016-01-06          10387    0
##  7 2016-01-07           9573    0
##  8 2016-01-08           3535   73.1
##  9 2016-01-09           8998    0.2
## 10 2016-01-10          10429    0
## # ... with 1,087 more rows
```

```
# Question 3
# Draw suitable graphs to display how the number of cyclists varies over time, over season, over day of

all_data = both %>%
  mutate(weekday = weekdays(Format_Date)) %>%
  mutate(weekday = factor(weekday, levels =
           c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))) %>%
  separate(Format_Date, into=c("year", "month", "day"), sep="-")  %>%
  mutate(season = case_when(month=="09" | month=="10" | month=="11" ~ "Spring",
                            month=="12" | month=="01" | month=="02" ~ "Summer",
                            month=="03" | month=="04" | month=="05" ~ "Autumn",
                            month=="06" | month=="07" | month=="08" ~ "Winter")) %>%
  mutate(season = factor(season, levels = c("Spring", "Summer", "Autumn", "Winter")))

# time
time.df = all_data %>%
  filter(!is.na(year))

ggplot(aes(year, total_cyclist), data=time.df) + geom_boxplot() +
  ggtitle("Number of cyclists over the years")
```



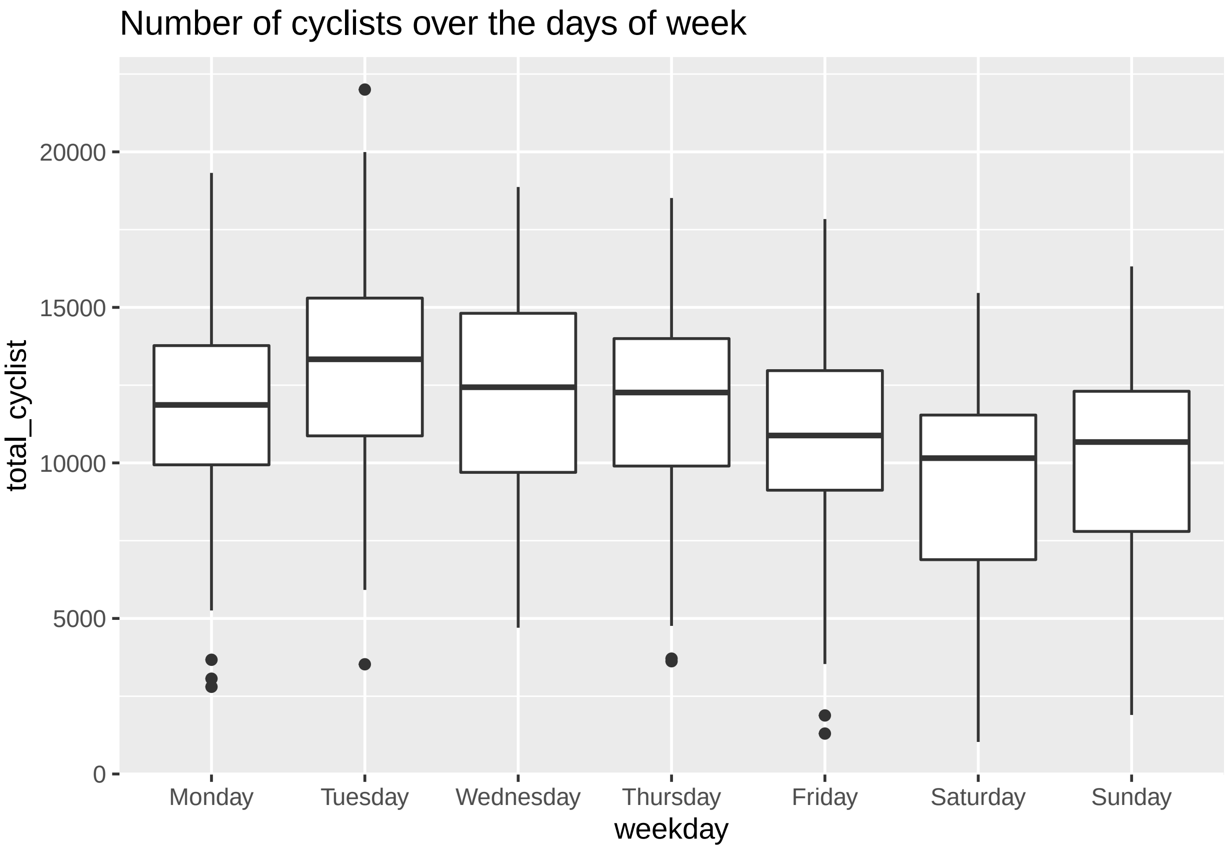
For the number of cyclists over the years I have plotted box plot. We can see the median of total cyclists is gradually increasing over the years. The range of the boxplots are also increasing. This could mean that there are more people cycling and it's getting popular.

```
# season
season.df = all_data %>%
  filter(!is.na(year))

ggplot(aes(season, total_cyclist), data=season.df) + geom_boxplot() +
  ggtitle("Number of cyclists over the seasons")
```



For the number of cyclists over the seasons, I have also plotted box plots. We can observe that there are highest number of cyclists in summer and the lowest number of cyclists in winter.

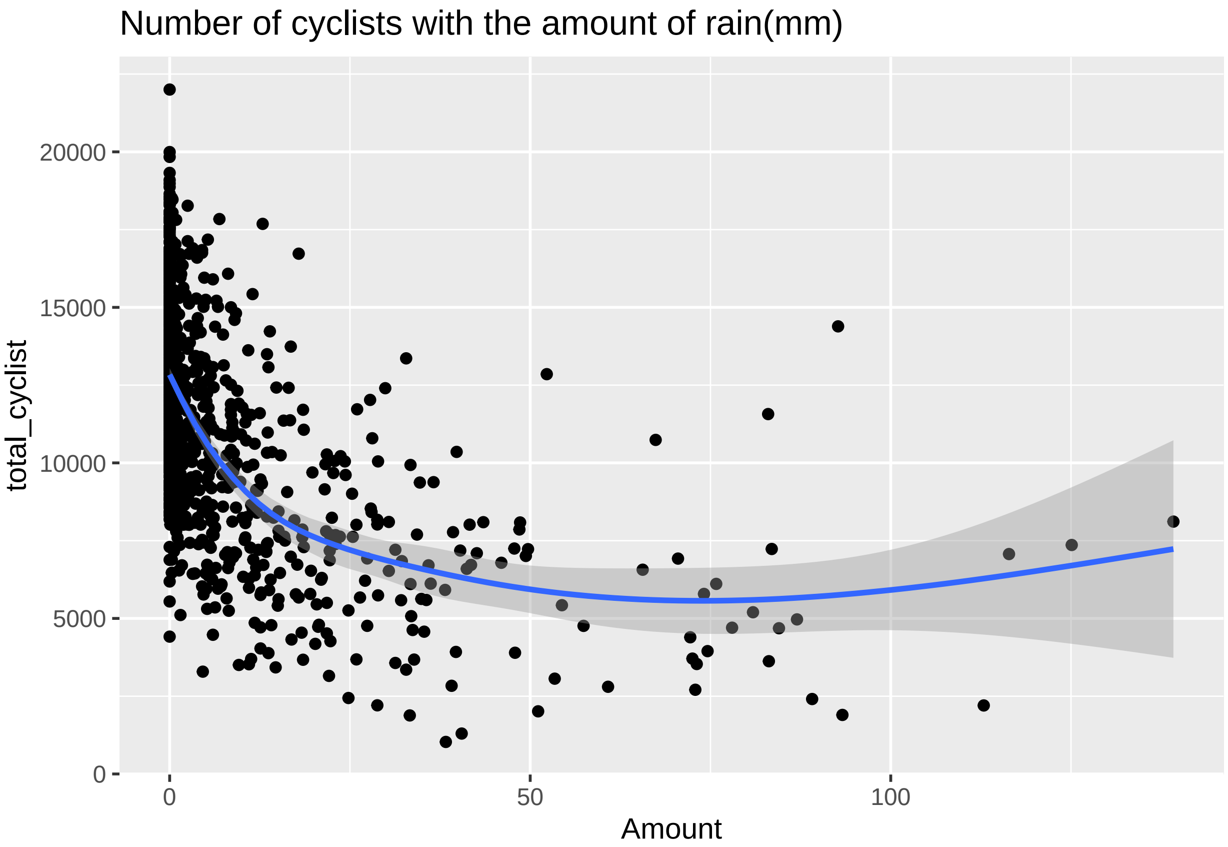
```
#day of week
day.df = all_data %>%
  filter(!is.na(year))

ggplot(aes(weekday, total_cyclist), data=day.df) + geom_boxplot() +
  ggtitle("Number of cyclists over the days of week")
```



We can see that the median of cyclists in the weekday higher than the median of cyclists in the weekends.

```
#rain
rain.df = all_data %>%
  filter(!is.na(year))

ggplot(aes(Amount, total_cyclist), data=rain.df) + geom_point() +  geom_smooth() +
  ggtitle("Number of cyclists with the amount of rain(mm)")
```

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

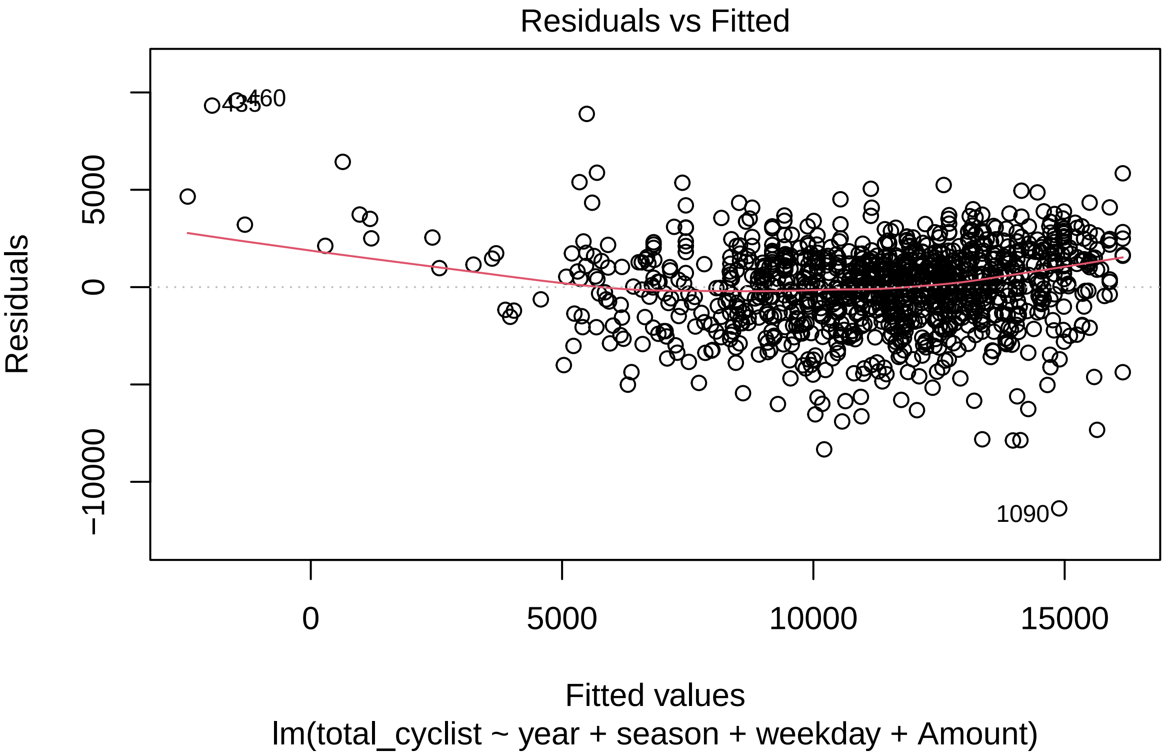


We can see a decreasing trend between rain and cyclists. The number of cyclists decrease significantly as the rain amount increases.

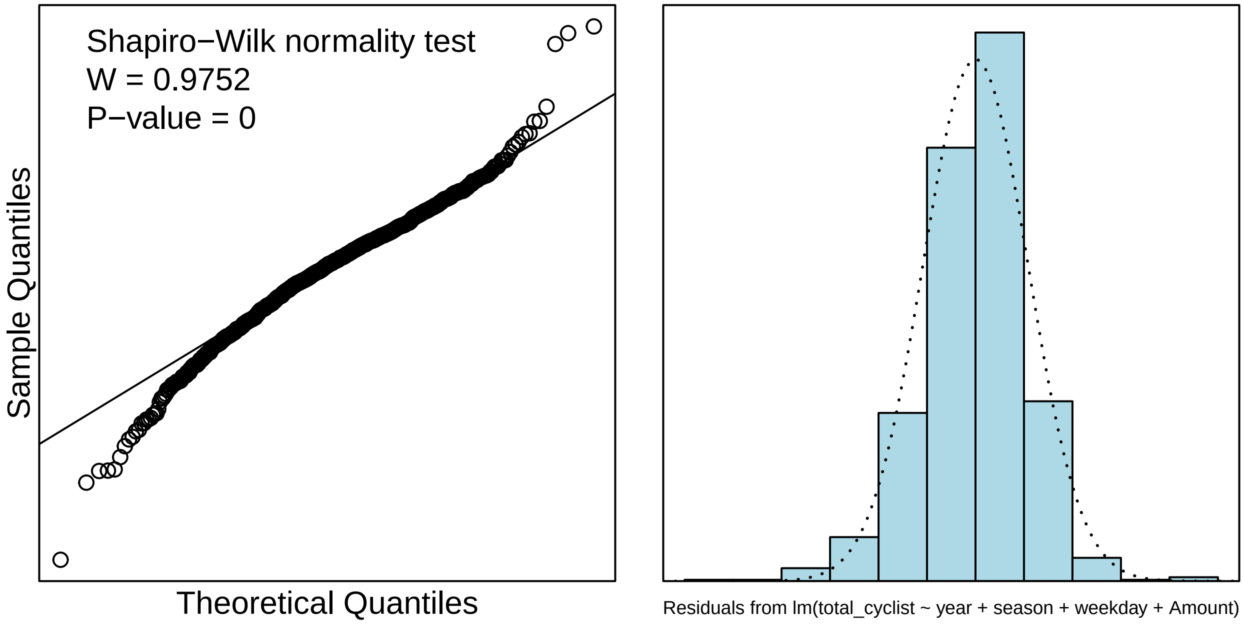
```
# Question 4
# Fit a regression model to predict the number of cyclists from year, season, day of the week, and rain
library(s20x)
cyclist.glm = lm(total_cyclist ~ year + season + weekday + Amount, data = all_data)
summary(cyclist.glm)
```

```
##
## Call:
## lm(formula = total_cyclist ~ year + season + weekday + Amount,
##     data = all_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11364.0  -1232.1    193.8   1361.3   9588.5
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10981.651    228.572  48.045  < 2e-16 ***
## year2017          1719.841    161.238  10.666  < 2e-16 ***
## year2018          3077.388    161.369  19.071  < 2e-16 ***
## seasonSummer       659.099    186.958   3.525 0.000441 ***
## seasonAutumn       399.916    186.420   2.145 0.032155 *
## seasonWinter     -1961.424    186.411 -10.522  < 2e-16 ***
## weekdayTuesday    1438.449    246.365   5.839 6.95e-09 ***
## weekdayWednesday   927.151    246.662   3.759 0.000180 ***
## weekdayThursday    525.327    246.503   2.131 0.033304 *
## weekdayFriday     -671.732    245.965  -2.731 0.006417 **
## weekdaySaturday  -2199.306    246.026  -8.939  < 2e-16 ***
## weekdaySunday    -1557.861    246.048  -6.332 3.55e-10 ***
## Amount            -115.053      4.242 -27.123  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2179 on 1083 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.6078, Adjusted R-squared:  0.6035
## F-statistic: 139.9 on 12 and 1083 DF,  p-value: < 2.2e-16
```

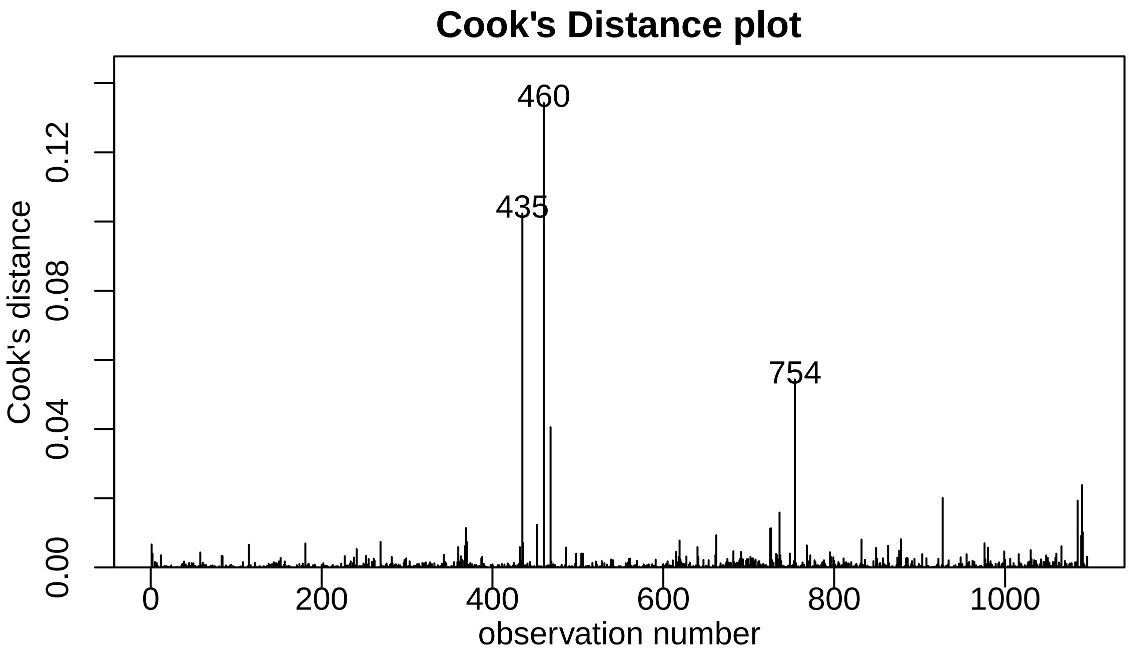
```
plot(cyclist.glm, which=1)
```



```
normcheck(cyclist.glm, shapiro.wilk=TRUE)
```



```
cooks20x(cyclist.glm)
```



```
confint(cyclist.glm)
```

```
##                         2.5 %     97.5 %
## (Intercept)      10533.15790 11430.1445
## year2017          1403.46709  2036.2157
## year2018          2760.75713  3394.0184
## seasonSummer       292.25685  1025.9403
## seasonAutumn        34.13155   765.7005
## seasonWinter     -2327.19066 -1595.6570
## weekdayTuesday     955.04178  1921.8563
## weekdayWednesday   443.16047  1411.1413
## weekdayThursday     41.65013  1009.0035
## weekdayFriday    -1154.35373  -189.1112
## weekdaySaturday  -2682.04822 -1716.5635
## weekdaySunday    -2040.64566 -1075.0758
## Amount            -123.37591  -106.7294
```

We have fitted a simple linear model with year, season, weekdays, and rain amount. All the variables are statistically significant (p - value less than 0.05). The residual plot shows a slight upwards trend around the low end of fitted values. However, most of the residuals are pattern less and shows constant scatter after 5000 fitted values. The normality is ok and the cook's plot is also fine. The R squared is 60.78% which isn't great for prediction.

Question 5

Based on your graphs and model, does rain have a big impact on the number of people cycling in Auckland?

Both the graph and model indicates that the rain has a big impact on the number of people cycling in Auckland. The graph shows a decreasing trend between rain amount and total cyclists. Likewise, from the model, we estimate that every millilitre increase in rain amount is associated with a decrease in the mean of cyclists of between 107 and 123 people. The rain term in the model is -115 and the p - value is 2e-16 which shows statistically significant.