A1_cyclists_and_rain_in_Auckand

ray_wang

2020.05.18

Use data on the number of people counted cycling in Auckland, and the amount of rain to answer the following questions.

1. If you try to convert the cycle count data to tidy format (which you don't have to do for this assignment, because it's hard), what is one obstacle you will encounter?

First, read the data into R.

```
#import packages
options(warn = -1)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages ---------------------------------------------
------------ tidyverse 1.3.0 --

## √ ggplot2 3.3.0     √ purrr   0.3.3
## √ tibble  3.0.1     √ stringr 1.4.0
## √ tidyr   1.0.2     √ forcats 0.4.0
## √ readr   1.3.1

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

library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

library(ggplot2)
library(s20x)
#read the data into R
cycle2016.df = read.csv(file= "dailyakldcyclecountdata2016_updated.csv",
 header = TRUE)
cycle2017.df = read.csv(file = "dailyakldcyclecountdata2017_1.csv", hea
der = TRUE)
cycle2018.df = read.csv(file = "dailyakldcyclecountdata2018.csv", heade
r = TRUE)
#we can see the brief information about these three dataframes
#head(cycle2018.df)
#head(cycle2017.df)
#head(cycle2016.df)
ncol(cycle2018.df)

## [1] 44

ncol(cycle2017.df)

## [1] 40

ncol(cycle2016.df)

## [1] 33

rain2018.df = read.table(file = "rain2018.txt", header = TRUE, sep = ",
")
rain16.17.df = read.table(file = "rain2016-17.txt", header = TRUE, sep
= ",")
ncol(rain2018.df)

## [1] 6

ncol(rain16.17.df)

## [1] 6
```

I find the columns of cyclist data are different in 2016,2017,2018.
So I can't bind these three years data into one directly. There are differnt names for same counters in these threee years data.

If I try to convert the cycle count data to tidy format, the obstacle is the difficulty in counter names change.

2. Compute the total number of cyclists counted for each day, and a suitable summary of the rainfall for each day, in the same data frame.

Before I start to manipulate the data, I must transform the NA into 0.

```
cycle2018.df[is.na(cycle2018.df)]<-0
cycle2017.df[is.na(cycle2017.df)]<-0
cycle2016.df[is.na(cycle2016.df)]<-0
cycle2018.df$Total.number.Cyclists <- rowSums(cycle2018.df[,2:ncol(cycl
e2018.df)])
cycle2017.df$Total.number.Cyclists <- rowSums(cycle2017.df[,2:ncol(cycl
e2017.df)])
cycle2016.df$Total.number.Cyclists <- rowSums(cycle2016.df[,2:ncol(cycl
e2016.df)])
```

Then get the sum of cyclists each day.

```
cycle2018.df$Total.number.Cyclists <- rowSums(cycle2018.df[,2:ncol(cycl
e2018.df)])
cycle2017.df$Total.number.Cyclists <- rowSums(cycle2017.df[,2:ncol(cycl
e2017.df)])
cycle2016.df$Total.number.Cyclists <- rowSums(cycle2016.df[,2:ncol(cycl
e2016.df)])
```

It is necessary to tidy the data. We need seprate the date into four variables: day of week, day of month, month and year.

And it helps to have numeric values stored as numeric values rather than as strings. We can mutate them.

We pick the useful columns from the data which includes the date and the sum of cyclist each day. After we get three years data tidy, we can bind them together.

```
cycle_18 <-cycle2018.df %>%
  separate(col=Date, into=c("dow", "day", "month", "year")) %>%
  mutate(dayno=as.numeric(day), yearno=as.numeric(year),
         wday=factor(dow,levels=c("Mon","Tue","Wed","Thu","Fri","Sat","
Sun")))
#tail(cycle_18)
#there is a 2019.0101 data which is useless in this assignment, we shou
ld delete this row.
cycle_18 = cycle_18[-366,]
#pick the useful columns from the data which includes the date and the
sum of cyclist each day.
cycle_18 = cycle_18[,-c(5:(ncol(cycle_18)-4))]
#tail(cycle_18)

#the procedure is same as above
cycle_17 <-cycle2017.df %>%
```

```
  separate(col=Date, into=c("dow", "day", "month", "year")) %>%
  mutate(dayno=as.numeric(day), yearno=as.numeric(year),
         wday=factor(dow,levels=c("Mon","Tue","Wed","Thu","Fri","Sat","
Sun")))
cycle_17 = cycle_17[,-c(5:(ncol(cycle_17)-4))]
#tail(cycle_17)

#the procedure is same as above
cycle_16 <-cycle2016.df %>%
  separate(col=Date, into=c("dow", "day", "month", "year")) %>%
  mutate(dayno=as.numeric(day), yearno=as.numeric(year),
         wday=factor(dow,levels=c("Mon","Tue","Wed","Thu","Fri","Sat","
Sun")))
cycle_16 = cycle_16[,-c(5:(ncol(cycle_16)-4))]
#tail(cycle_16)

cycle = rbind(cycle_16,cycle_17,cycle_18)
summary(cycle)

##      dow                 day                month
##  Length:1096        Length:1096        Length:1096
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      year           Total.number.Cyclists     dayno           yearno

##  Length:1096        Min.   : 2060         Min.   : 1.00   Min.   :20
16
##  Class :character   1st Qu.:18404         1st Qu.: 8.00   1st Qu.:20
16
##  Mode  :character   Median :23081         Median :16.00   Median :20
17
##                     Mean   :22650         Mean   :15.73   Mean   :20
17
##                     3rd Qu.:27109         3rd Qu.:23.00   3rd Qu.:20
18
##                     Max.   :44004         Max.   :31.00   Max.   :20
18
##

##   wday
##  Mon:157
##  Tue:156
##  Wed:156
##  Thu:156
##  Fri:157
```

```
##  Sat:157
##  Sun:157
```

```
head(cycle)
```

```
##   dow day month year Total.number.Cyclists dayno yearno wday
## 1 Fri   1   Jan 2016                  2598     1   2016  Fri
## 2 Sat   2   Jan 2016                  2060     2   2016  Sat
## 3 Sun   3   Jan 2016                 14846     3   2016  Sun
## 4 Mon   4   Jan 2016                 23912     4   2016  Mon
## 5 Tue   5   Jan 2016                 20334     5   2016  Tue
## 6 Wed   6   Jan 2016                 20774     6   2016  Wed
```

```
tail(cycle)
```

```
##      dow day month year Total.number.Cyclists dayno yearno wday
## 1091 Wed  26   Dec 2018                 16628    26   2018  Wed
## 1092 Thu  27   Dec 2018                 25604    27   2018  Thu
## 1093 Fri  28   Dec 2018                 24204    28   2018  Fri
## 1094 Sat  29   Dec 2018                 22984    29   2018  Sat
## 1095 Sun  30   Dec 2018                 24542    30   2018  Sun
## 1096 Mon  31   Dec 2018                 21206    31   2018  Mon
```

Then tidy the rain data. Aggregate the rain amount each hour to get the rain amount each day.

Similarly, we We need seprate the date into three variables: day of month, month and year.

And it helps to have numeric values stored as numeric values rather than as strings. We can mutate them.

After we get the rain data tidy, we can bind the rain data andy cyclist data together.

```
#head(rain2018.df)
#head(rain16.17.df)
summary(rain2018.df)
```

```
##     Station        Date.NZST.         Time.NZST.     Amount.mm.
##  Min.   :22719   Min.   :20180101   Min.   :   0   Min.   : 0.0000
##  1st Qu.:22719   1st Qu.:20180401   1st Qu.: 500   1st Qu.: 0.0000
##  Median :37852   Median :20180630   Median :1100   Median : 0.0000
##  Mean   :30381   Mean   :20180662   Mean   :1150   Mean   : 0.1655
##  3rd Qu.:37852   3rd Qu.:20180928   3rd Qu.:1700   3rd Qu.: 0.0000
##  Max.   :37852   Max.   :20190101   Max.   :2300   Max.   :61.3000
##   Period.Hrs. Freq
##  Min.   :1    H:17304
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
```

```r
summary(rain16.17.df)
```

```
##     Station        Date.NZST.          Time.NZST.     Amount.mm.
##  Min.   :22719   Min.   :20160101   Min.   :   0   Min.   : 0.0000
##  1st Qu.:22719   1st Qu.:20160701   1st Qu.: 500   1st Qu.: 0.0000
##  Median :30286   Median :20161231   Median :1100   Median : 0.0000
##  Mean   :30286   Mean   :20165662   Mean   :1150   Mean   : 0.1359
##  3rd Qu.:37852   3rd Qu.:20170702   3rd Qu.:1700   3rd Qu.: 0.0000
##  Max.   :37852   Max.   :20180101   Max.   :2300   Max.   :27.4000
##   Period.Hrs. Freq
##  Min.   :1    H:35090
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
```

```r
rain_tmp = rbind(rain16.17.df, rain2018.df)
# Aggregate the hours rain amount to get the rain amount each day.
rain_sum<-aggregate(Amount.mm. ~  Date.NZST.,data=rain_tmp,sum)

#tidy the rain data
rain <- rain_sum %>%
  select(Date=Date.NZST., Rainfall=Amount.mm.) %>%
  separate(Date, into=c("year","month","day"),sep=c(4,6)) %>%
  mutate(dayno=as.numeric(day))

#there is a 2019.0101 data which is useless in this assignment, we shou
ld delete this row.
rain = rain[-1097,]
tail(rain)
```

```
##      year month day Rainfall dayno
## 1091 2018    12  26        0    26
## 1092 2018    12  27        0    27
## 1093 2018    12  28        0    28
## 1094 2018    12  29        0    29
## 1095 2018    12  30        0    30
## 1096 2018    12  31        0    31
```

```r
Rainfall =  rain$Rainfall
monthno = rain$month
#confirm that the number of rows is equal in these two dataframe
nrow(rain)
```

```
## [1] 1096
```

```r
nrow(cycle)
```

```
## [1] 1096
```

```
#After we get the rain data tidy, we can bind the rain data andy cyclis
t data together.
cycle_rain <- cbind(cycle , Rainfall, monthno)%>%
  mutate(monthno =as.numeric(monthno))%>%
  #transfrom month to season
  mutate(season = quarter(monthno))
```

We achieve the goal: compute the total number of cyclists counted for each day, and a suitable summary of the rainfall for each day, in the same data frame.

```
summary(cycle_rain)
```

```
##      dow                 day                month
##  Length:1096        Length:1096        Length:1096
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      year           Total.number.Cyclists     dayno           yearno

##  Length:1096        Min.   : 2060         Min.   : 1.00   Min.   :20
16
##  Class :character   1st Qu.:18404         1st Qu.: 8.00   1st Qu.:20
16
##  Mode  :character   Median :23081         Median :16.00   Median :20
17
##                     Mean   :22650         Mean   :15.73   Mean   :20
17
##                     3rd Qu.:27109         3rd Qu.:23.00   3rd Qu.:20
18
##                     Max.   :44004         Max.   :31.00   Max.   :20
18
##

##   wday        Rainfall          monthno           season
##  Mon:157   Min.   :  0.000   Min.   : 1.000   Min.   :1.000
##  Tue:156   1st Qu.:  0.000   1st Qu.: 4.000   1st Qu.:2.000
##  Wed:156   Median :  0.500   Median : 7.000   Median :3.000
##  Thu:156   Mean   :  6.964   Mean   : 6.522   Mean   :2.508
##  Fri:157   3rd Qu.:  6.000   3rd Qu.:10.000   3rd Qu.:4.000
##  Sat:157   Max.   :139.200   Max.   :12.000   Max.   :4.000
##  Sun:157
```

```
head(cycle_rain)
```

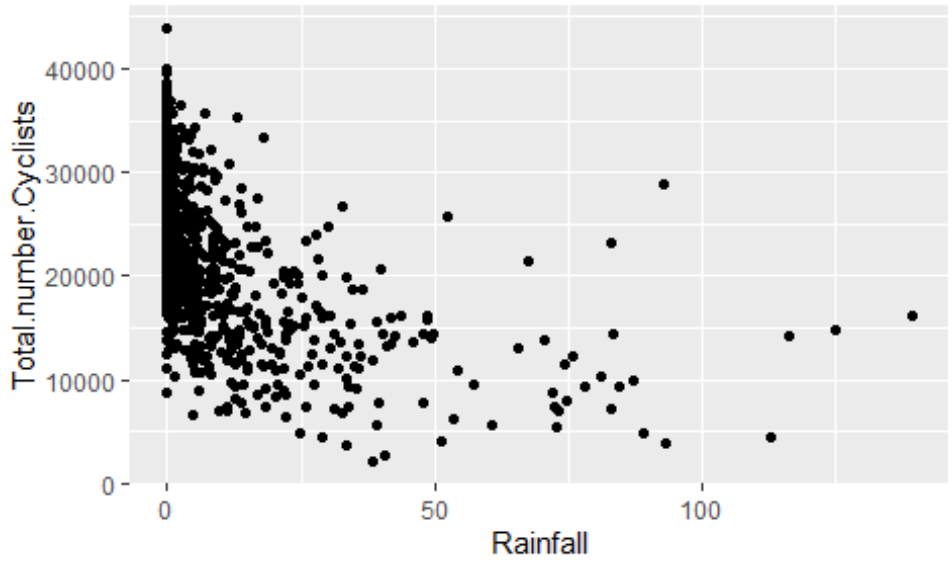
```
##   dow day month year Total.number.Cyclists dayno yearno wday Rainfal
l
## 1 Fri   1   Jan 2016                  2598     1   2016  Fri     40.
5
```

```
## 2 Sat    2    Jan 2016                  2060      2    2016  Sat      38.
3
## 3 Sun    3    Jan 2016                 14846      3    2016  Sun      13.
6
## 4 Mon    4    Jan 2016                 23912      4    2016  Mon       0.
1
## 5 Tue    5    Jan 2016                 20334      5    2016  Tue       0.
0
## 6 Wed    6    Jan 2016                 20774      6    2016  Wed       0.
0
##   monthno season
## 1       1      1
## 2       1      1
## 3       1      1
## 4       1      1
## 5       1      1
## 6       1      1
```

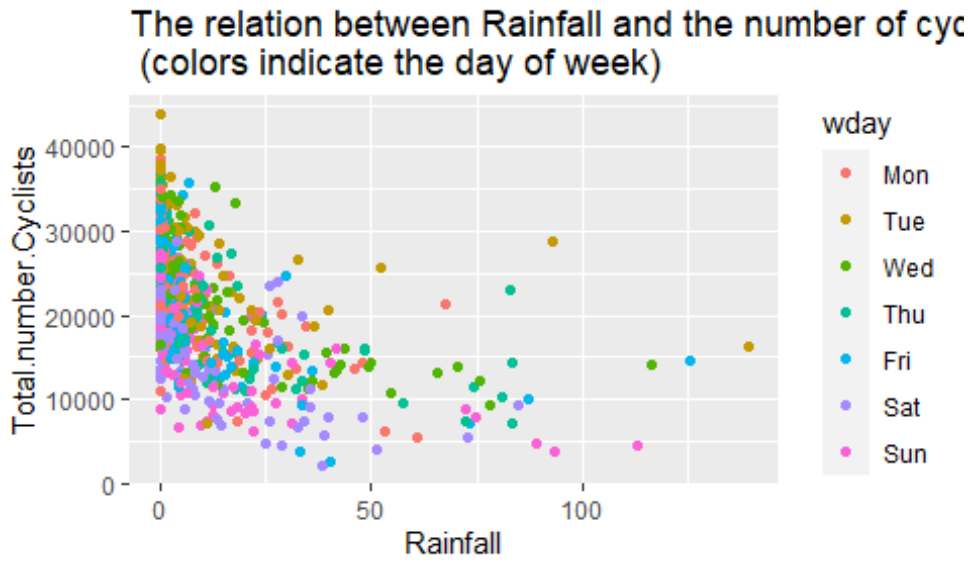
3. Draw suitable graphs to display how the number of cyclists varies over time, over season, over day of the week, and with rain

Use qpplot() to see the trend over time, over day of the week, over season and with rain.

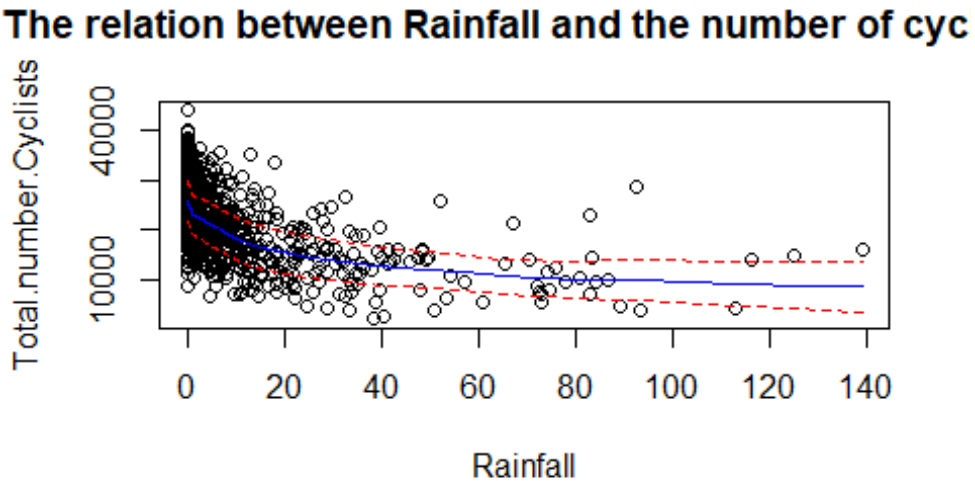
```
ggplot(cycle_rain, aes(Rainfall,Total.number.Cyclists)) + geom_point()
```



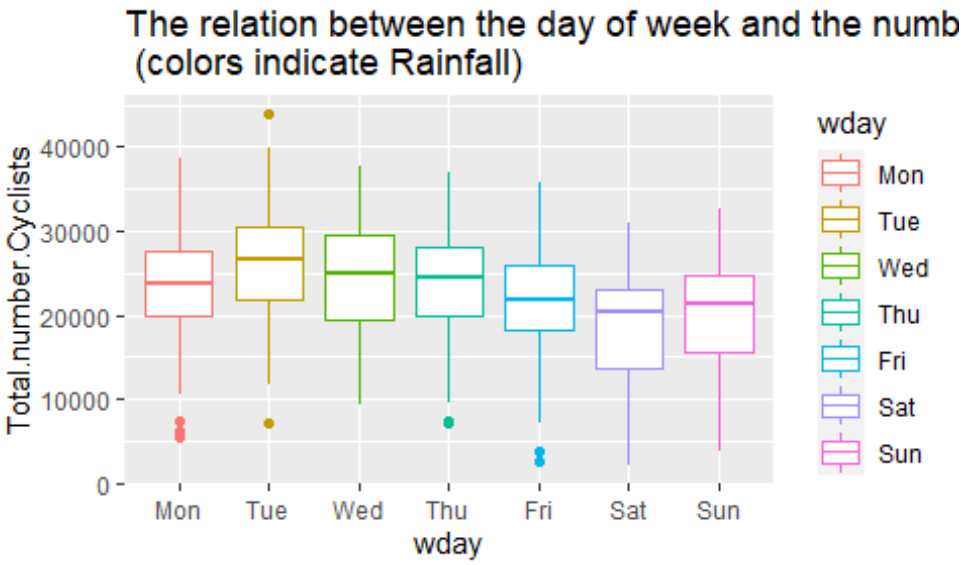

```
qplot(Rainfall, Total.number.Cyclists,data=cycle_rain, col=wday,
      main="The relation between Rainfall and the number of cyclists\n
(colors indicate the day of week)")
```



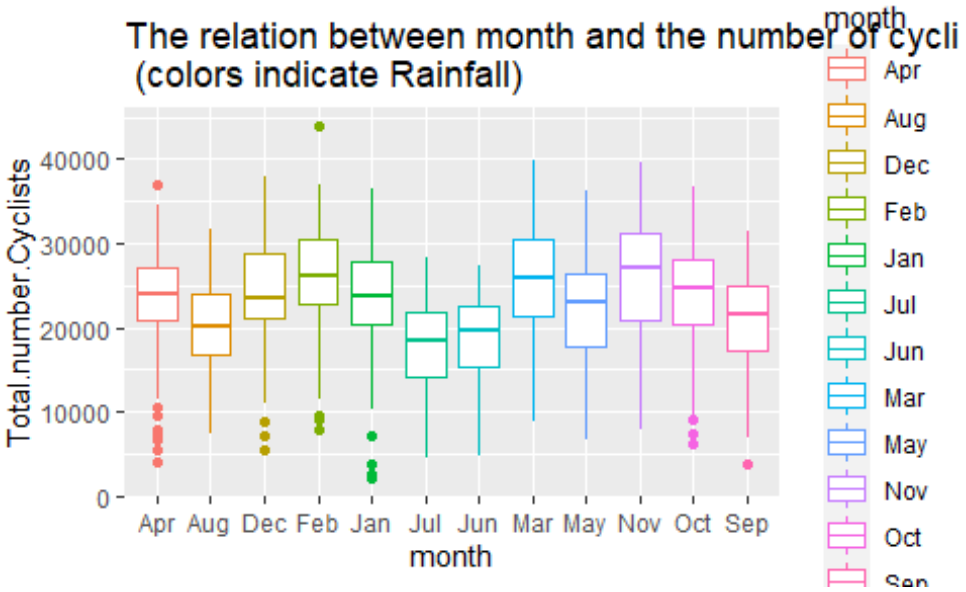

```
trendscatter(Total.number.Cyclists ~ Rainfall,main = "The relation betw
een Rainfall and the number of cyclists" ,data=cycle_rain)
```



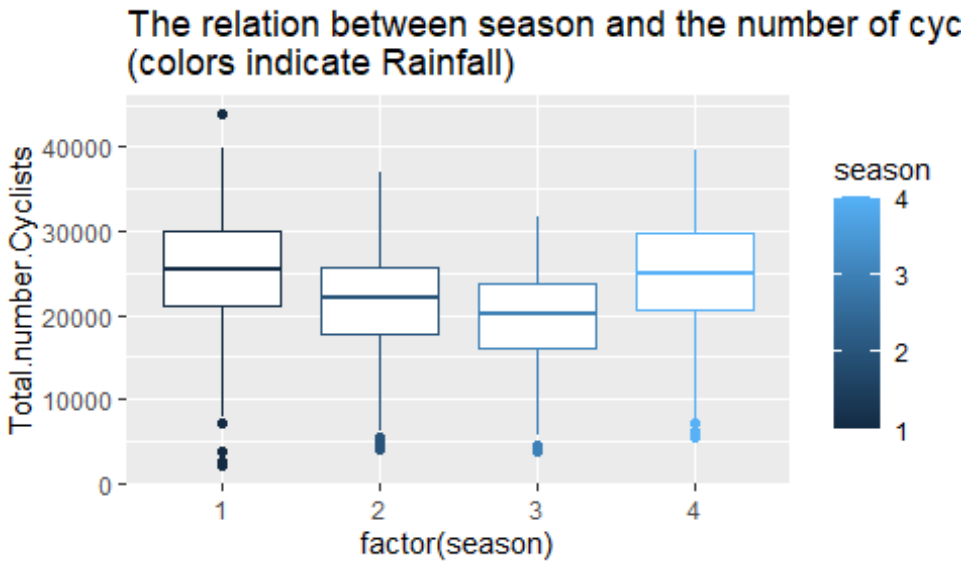

```
qplot(wday,Total.number.Cyclists,data=cycle_rain, col=wday, geom= "boxp
lot",
      main="The relation between the day of week and the number of cycl
ists\n (colors indicate Rainfall)")
```



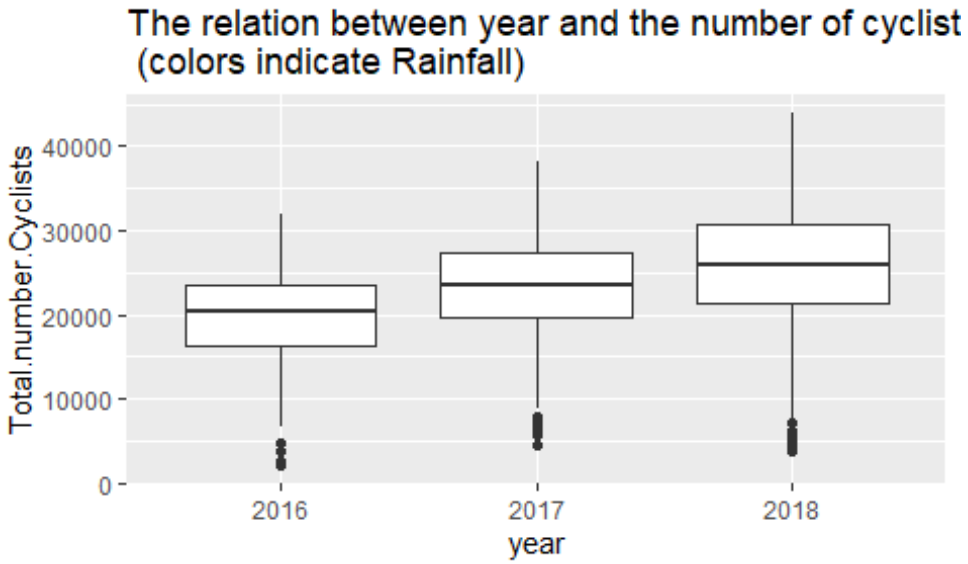

```
qplot(month,Total.number.Cyclists,data=cycle_rain, col=month, geom= "bo
xplot",
      main="The relation between month and the number of cyclists\n (co
lors indicate Rainfall)")
```




```
qplot(factor(season),Total.number.Cyclists,data=cycle_rain, col=season,
 geom= "boxplot",
      main="The relation between season and the number of cyclists \n(c
olors indicate Rainfall)")
```




```
qplot(year,Total.number.Cyclists,data=cycle_rain, col=Rainfall, geom= "
boxplot",
      main="The relation between year and the number of cyclists\n (col
ors indicate Rainfall)")
```



It is obvious that the number of cyclists decrease when the rain amount increase. And people like go to cycle at Tuesday most while they don't like go to cycle at Saturday.

What's more, people like go to cycle in December most while they don't like go to cycle in April, July and June. In other words, they perfer go to cycle in season 4 (month 10-12) while don't like go to cycle in season 3 (7-9) and season 2 (3-6).

In long term, the number of cyclists increase from 2016 to 2018.

##4. Fit a regression model to predict the number of cyclists from year, season, day of the week, and rain.

Finally, we can fit a model to predict the number of cyclists from year, season, day of the week, and rain.

Accroding to the anaylsis of graphs above, I choose total numberof cyclists each day as the dependent variable and the amount of rain, the day of weeks and season as independent variables.

By the way, the variable month is useful too, but it has the similar impact as the variable season, so I didn't choose month as one of independent variables.

```
model <- cycle_rain %>%
  lm(Total.number.Cyclists~Rainfall + factor(year) + factor(wday)+facto
r(season),data=.)
coef(summary(model))
```

```
##                     Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)      23925.6852 457.000704  52.353716 6.025379e-299
## Rainfall          -229.6557   8.460875 -27.143256 3.423148e-124
## factor(year)2017  3441.0818 321.015060  10.719378  1.497792e-25
## factor(year)2018  6155.8501 321.276896  19.160575  1.075182e-70
## factor(wday)Tue   2871.6663 490.484934   5.854749  6.327547e-09
## factor(wday)Wed   1814.3807 491.064381   3.694792  2.310684e-04
## factor(wday)Thu   1011.1459 490.744004   2.060435  3.959545e-02
## factor(wday)Fri  -1348.6619 489.698743  -2.754065  5.984554e-03
## factor(wday)Sat  -4365.0846 489.828721  -8.911451  2.091162e-18
## factor(wday)Sun  -3082.2972 489.872122  -6.292045  4.540394e-10
## factor(season)2  -3388.2688 372.042111  -9.107218  3.974277e-19
## factor(season)3  -5357.4098 371.053105 -14.438391  2.388713e-43
## factor(season)4   -915.4808 372.262844  -2.459232  1.407894e-02
```

```
summary(model)
```

```
##
## Call:
## lm(formula = Total.number.Cyclists ~ Rainfall + factor(year) +
##     factor(wday) + factor(season), data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22457.5  -2210.5    380.5   2733.8  21343.9
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      23925.685    457.001  52.354  < 2e-16 ***
## Rainfall          -229.656      8.461 -27.143  < 2e-16 ***
## factor(year)2017  3441.082    321.015  10.719  < 2e-16 ***
## factor(year)2018  6155.850    321.277  19.161  < 2e-16 ***
```

```
## factor(wday)Tue   2871.666    490.485   5.855 6.33e-09 ***
## factor(wday)Wed   1814.381    491.064   3.695 0.000231 ***
## factor(wday)Thu   1011.146    490.744   2.060 0.039595 *
## factor(wday)Fri  -1348.662    489.699  -2.754 0.005985 **
## factor(wday)Sat  -4365.085    489.829  -8.911  < 2e-16 ***
## factor(wday)Sun  -3082.297    489.872  -6.292 4.54e-10 ***
## factor(season)2  -3388.269    372.042  -9.107  < 2e-16 ***
## factor(season)3  -5357.410    371.053 -14.438  < 2e-16 ***
## factor(season)4   -915.481    372.263  -2.459 0.014079 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4339 on 1083 degrees of freedom
## Multiple R-squared:  0.6114, Adjusted R-squared:  0.607
## F-statistic:   142 on 12 and 1083 DF,  p-value: < 2.2e-16
```

Almost very variables' p-value is good which means variables in this model are significant. The model is acceptable.

The model support the analysis of graphs above that the number of cyclists decrease when the rain amount increase.

And people prefer go to cycle at Tuesday and in season 4 most while they don't like go to cycle at Saturday, Sunday and in season 3 and season 2.

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

Yes, I think rain have a big impact on the number of people cycling in Auckland.

First, the p-value of the variable 'Rainfall' is very close to zero which proves its significance.

What's more, the cofficient of variable 'Rainfall' is -229.6 which means the number of cyclists decrease about 230 when the rain amount increase 1 mm in Auckland.

In summary, rain have a big impact on the number of people cycling in Auckland.