EcoBici 2017 Exploratory Data Analysis

Emanuel Becerra Soto

August 29, 2018

Introduction

This is an analysis of the Mexico City EcoBici individual trips data.

EcoBici is a bicycle sharing system, launched in February 2010. It is managed by a private company but with an initial investment by the government of 75 million pesos.

The source data was obtained from ecobici web page https://www.ecobici.cdmx.gob.mx/
and was made public online thanks to policies for improving transparency on government programs.

The data being analyze is made of trips made over 2017. It was previously transformed in order to facilitate its processing. As this is an exploratory analysis only 20% of the total trips made during 2017 will be analyzed, an approximation of 2,000,000 data points.

The data set is interesting as it could help to answer things like:

1. Does the system is socially inclusive?
2. Was the service a success?
3. Is the service still a successful one?
4. Which are some general patterns of mobility using the service?
5. Does the system integrate well with the other transportation options that the city offers?

Probably some of the answers are known but an analysis may help to get a deeper insight over the topics.

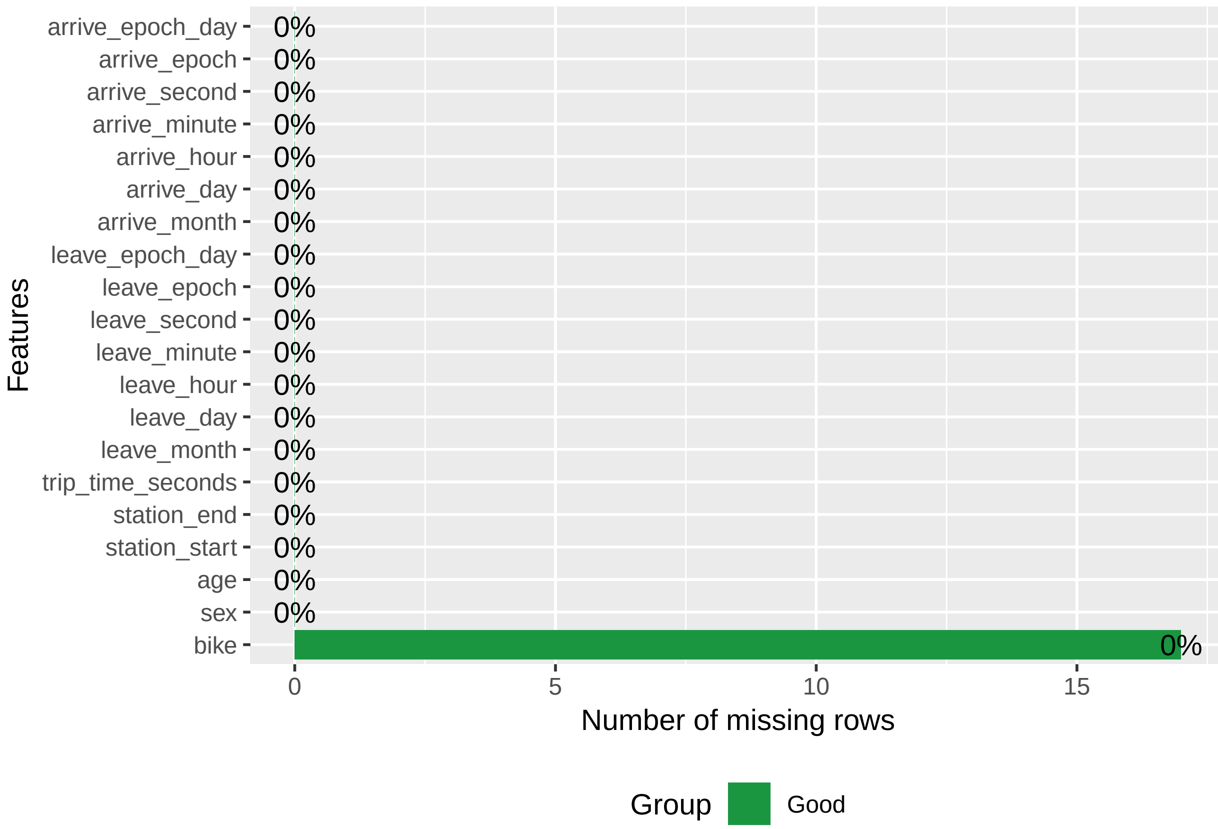
Additionally as an exploratory analysis it would serve as a starting point for further more specific future analysis.

Cleaning the data

Before going right away into the analysis it is necessary to asses the quality of the data and clean it up a little bit.

```
bikes_missing <- plot_missing(bikes)
```

Figure 1:




```
# That is about 0.000895% of the total
filter(bikes_missing, num_missing > 0)
```

```
## # A tibble: 1 x 4
##   feature num_missing pct_missing group
##   <fct>         <int>       <dbl> <chr>
## 1 bike             17  0.00000895 Good
```

```
# So let's Remove them
bikes <- ( bikes[ !is.na(bikes$bike), ] )
```

Plotting the missing data we can observe that our data has 17 missing values That is about 0.000895% of the total so let's remove the missing values.

Other problem that I've run into is that some trips oddly have negative times. They're 6019 trips with negative times. They only account for 0.32 %.

In order to observe if something strange is going on I will be plotting the 6019 negative times.

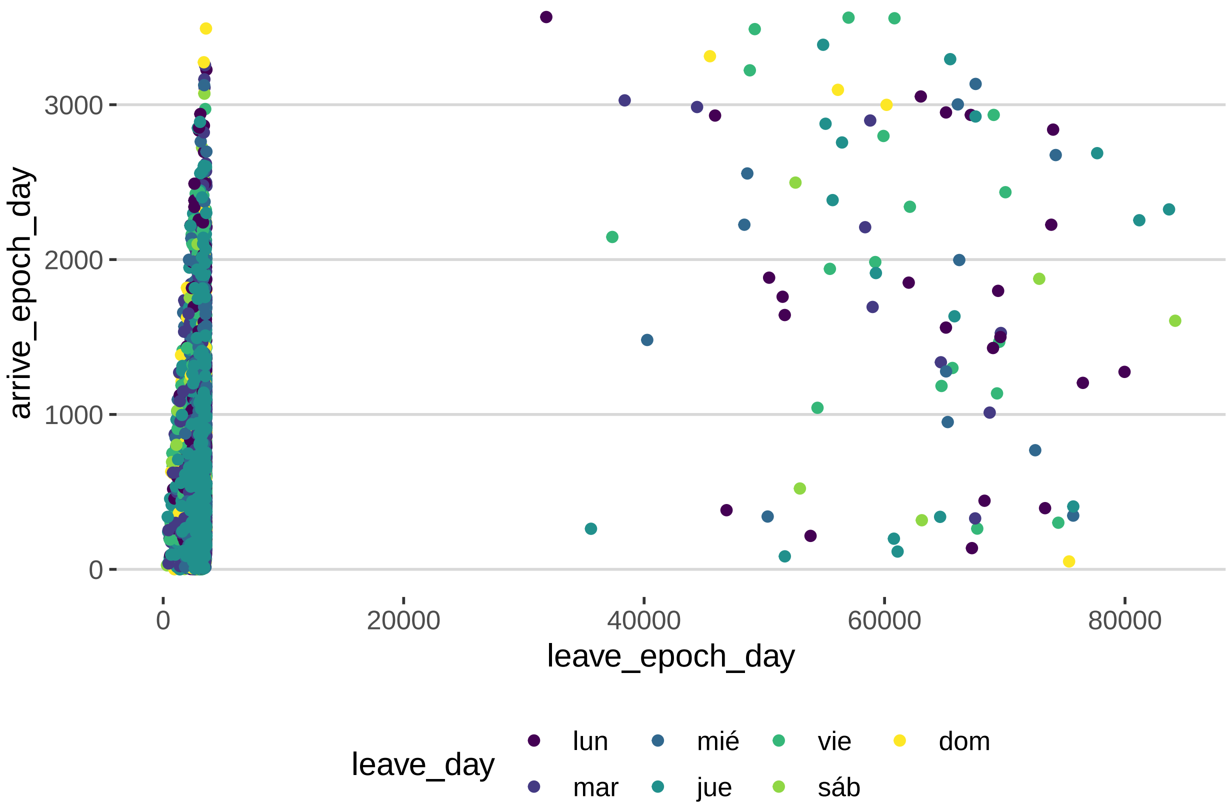
```
###### Removing problematic Data  ######

# Some trips oddly give negative times
neg_times_idxs <- which( bikes$trip_time_seconds < 0 )

length(neg_times_idxs)
```

```
## [1] 6019
```

```
length(neg_times_idxs) / nrow(bikes)
```

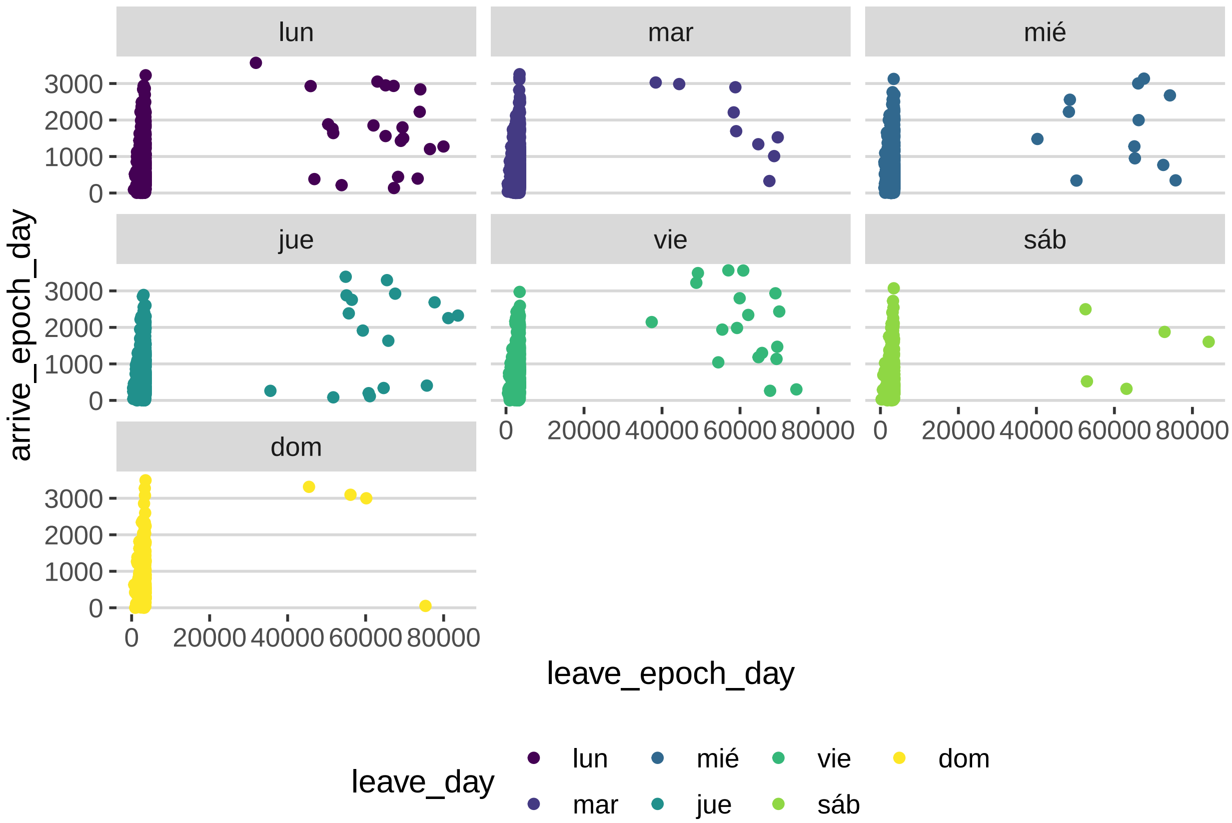
```
## [1] 0.003168575
```

```r
# Visualizing the problematic trips
ggplot(filter(bikes, trip_time_seconds < 0),
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day)) +
  geom_point()+theme_hc()
```



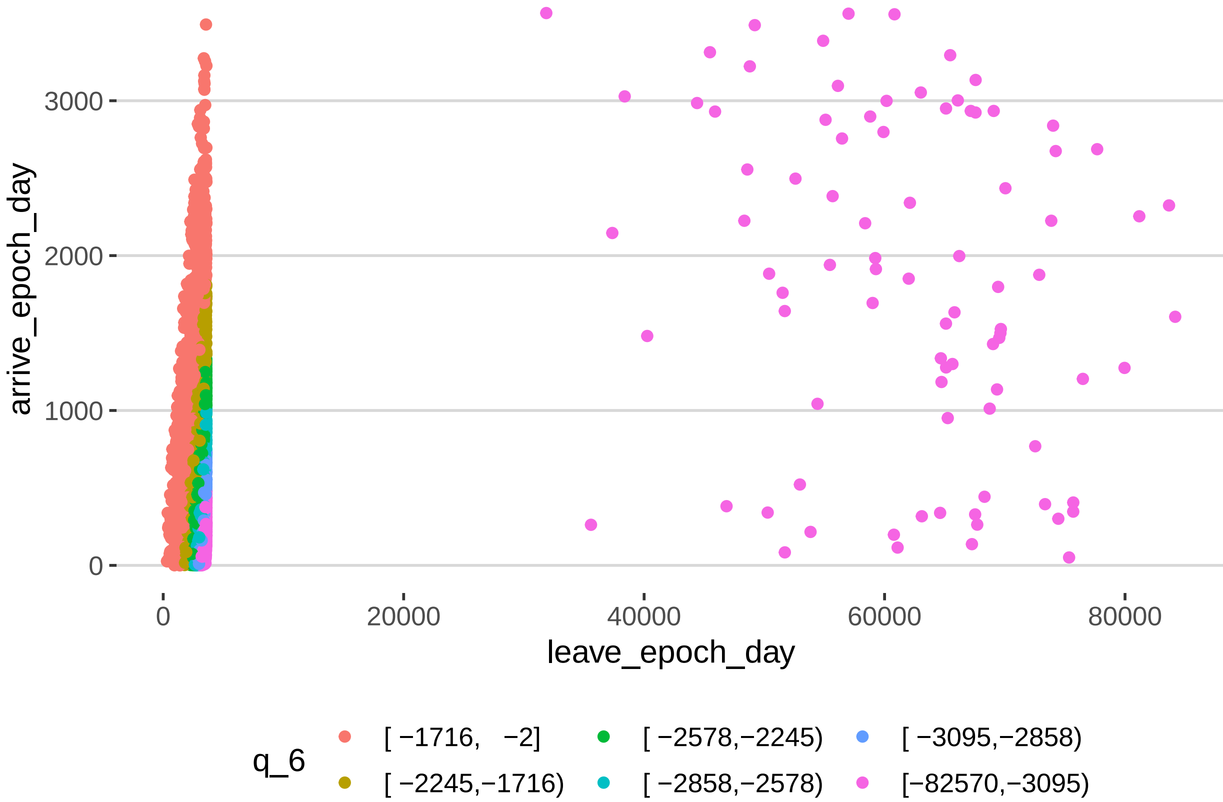
The x axis is the leave time (seconds since midnight) and the y axis is the arrive time, the color is the week days. We could see that the negative time trips are clustered near 0 for leaving time this means that almost all of this trips are started in the night near 12:00 am We cannot see a pattern over any week day.

```r
# Visualizing the problematic trips
ggplot(filter(bikes, trip_time_seconds < 0),
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day)) +
  geom_point()+facet_wrap(.~leave_day)+theme_hc()
```

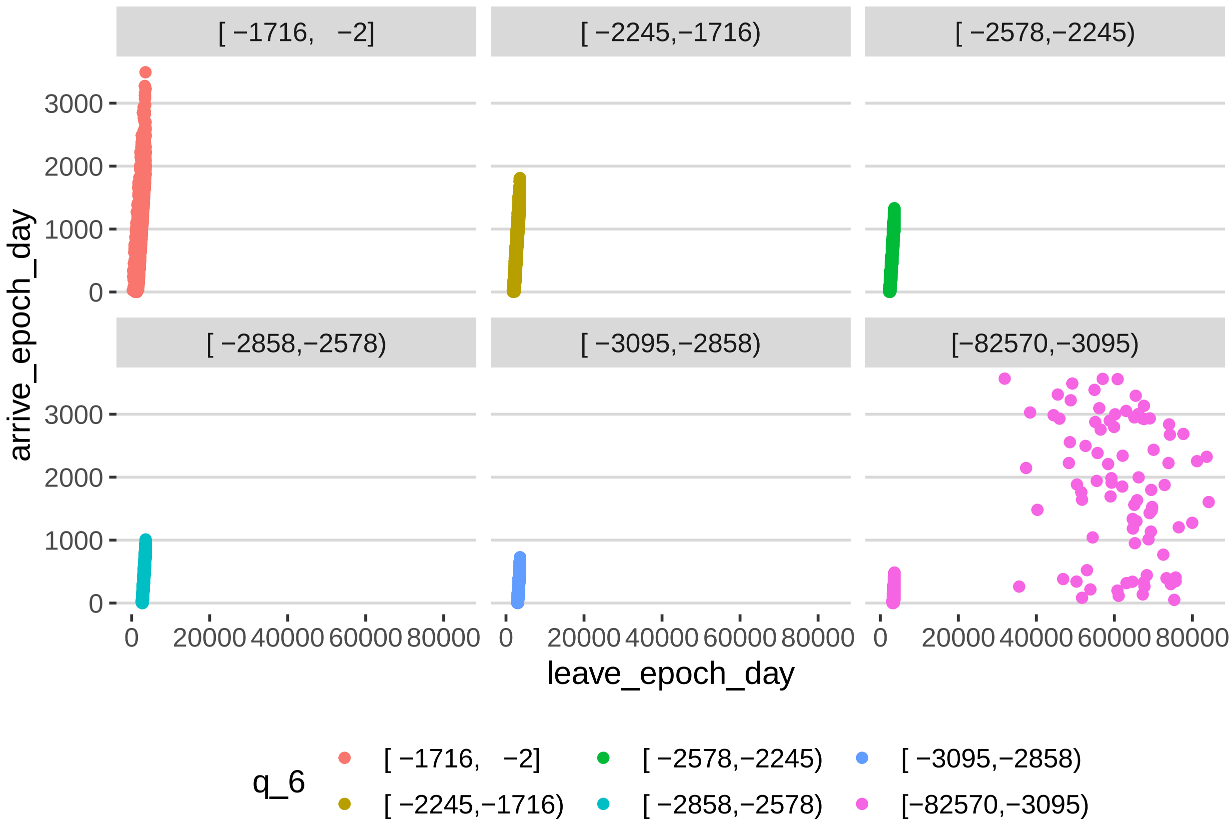


Because in the last plot there were a lot of points, maybe we missed a pattern over week days. So I've plotted the problematic time trips in 7 plots each per day the point distribution is similar, so there is not an specific day that the problematic trips occurs more or in a different fashion.

```
q_6 <- as.character( cut2(bikes$trip_time_seconds[ bikes$trip_time_seconds < 0], g = 6 ) )
# Visualizing the problematic trips
ggplot(filter(bikes, trip_time_seconds < 0),
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=q_6)) +
  geom_point()+theme_hc()
```




```
bikes2 <- filter(bikes, trip_time_seconds < 0)
bikes2$q_6 <- as.factor(q_6)
# Visualizing the problematic trips
ggplot(bikes2,
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=q_6)) +
  geom_point()+facet_wrap(.~q_6)+theme_hc()
```



Probably there is a pattern between the problematic trips and its duration. We could see that the trips lasting more than -3095 seconds or -51 min were made only between the 11:00 am (40,000) and 4:00 pm (80,000) of leaving time.

The pattern of the negative times of the intervals:

0min-28min

28min-37min

37min-42min

42min-47min

47min-51min

Is similar.

The interval between. 51min-22hours has a different pattern.

We don't know why this errors in the total trip time arrive. The providers of the service should check their bike stations to correct the issues with the time keeping systems Regardless this problem tends to happens early in the morning between the 0:00 hrs and 1:00 hrs

Because the problematic times happen very little and they account for less than 1% (0.32 %) percent of the travels let's remove them.

```r
# Removing negtive time trips
bikes <- filter(bikes, trip_time_seconds >= 0)
```

EcoBici system set prices (MXN) for exceeding time tips as follows:

From 0min-45min No extra cost. From 45min-60min $12.00. From Each extra hour $39.00. From More than a day 24 hrs. $5485.00.

Analyzing the data by these divisions seems natural.

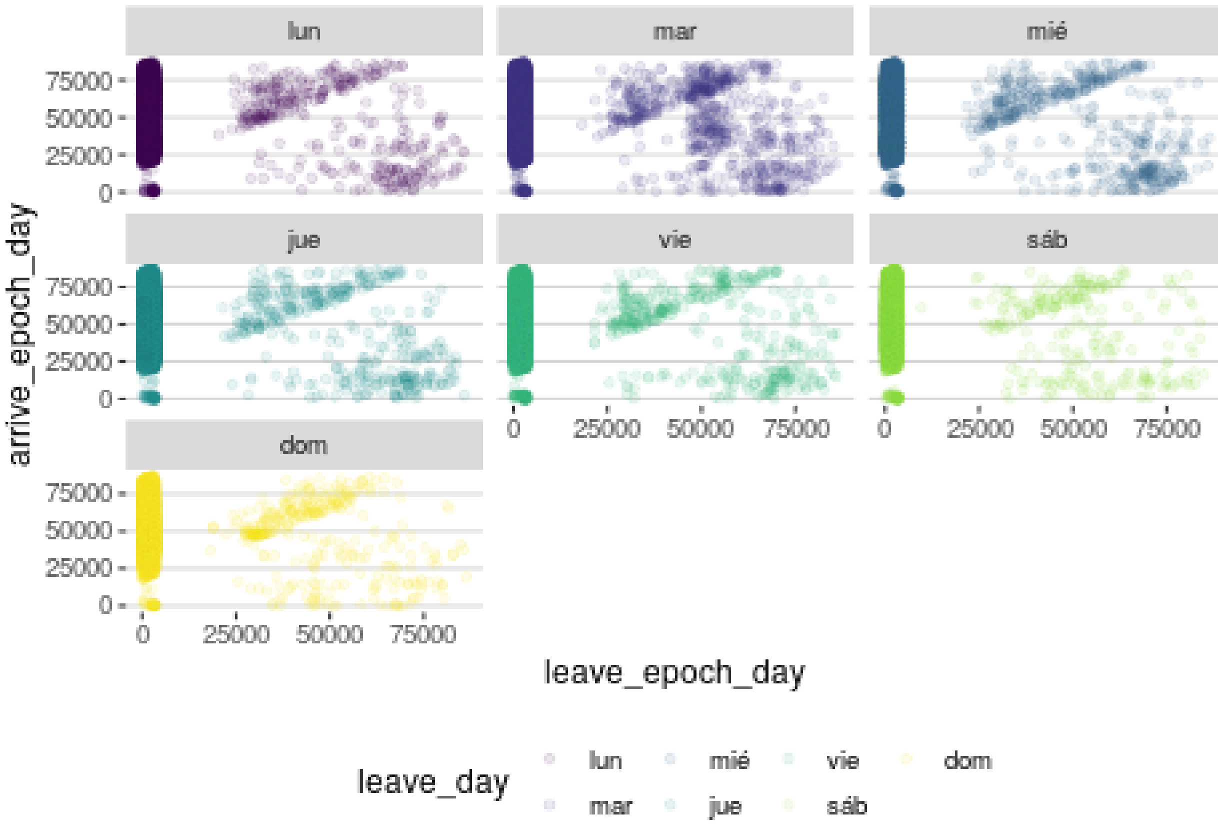
As the principal interest over the data set is in finding general trends outliers need to be removed. I've set the outlier threshold on 4 hours, there isn't a specific reason, any time more than 45 min or 1 hour could be chosen as the bulk of the trips are below 1 hour.

```
# Set outlier thereshold for 4 hours
outlier_thereshold <- 4 * 3600
exceeding_time <- 45 * 60
exceeding_time_hour <- 1 * 3600

# Adding a vector of all the exeding time trips
bikes$exceeding <- bikes$trip_time_seconds > exceeding_time
# Adding the 1 hour exeding trips
bikes$exceeding_hour <- bikes$trip_time_seconds > exceeding_time_hour

# Extracting outliers (more than 4 hours)
out_bikes <- filter( bikes, trip_time_seconds > outlier_thereshold )
# Extracting trips with exeding time more than 45 min
exceeding_bikes <- filter( bikes, trip_time_seconds > exceeding_time )

# Removing the outliers
bikes <- filter( bikes, trip_time_seconds <= outlier_thereshold )

# Quick View of the outliers
# They account for the 7.51%
n <- nrow(bikes)
n_out <- nrow(out_bikes)
n_out/n
```

```
## [1] 0.08125888
```

Visualizing the outliers.

```
# Visualizing the outliers
ggplot( out_bikes,
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day) ) +
  geom_point(alpha=1/10)+facet_wrap(.~leave_day)+theme_hc()
```

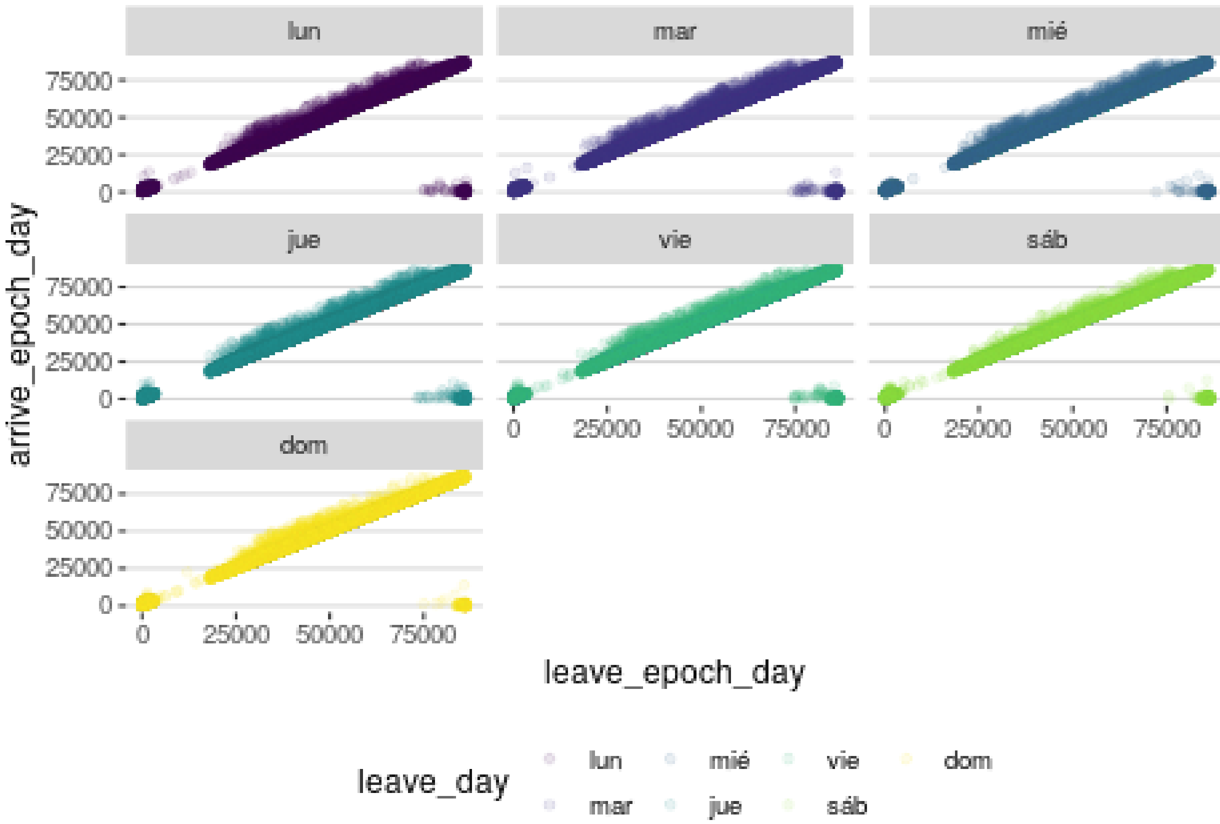



```
# A lot of outliers have the pattern of startiing the trip close to midnigth and
# Returning the bike several hours later.
```

A lot of outliers have the pattern of startiing the trip close to midnigth and Returning the bike several hours later.

Versus the data with out the outliers.

```
ggplot( bikes,
        aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day) ) +
  geom_point(alpha=1/10)+facet_wrap(.~leave_day)+theme_hc()
```



With out the outliers we can see that most people uses the bike in more day time hours and returned the bike a little after that, so the an almost perfect line gets formed between leave and arrive time there is a small cluster of points at the end of the day on leaving time and the start of the day on arriving time, that cluster is the people that started their trip close to midnigth and returned the bike the next day, a couple of minutes after midnigth.

Describing the data attributes

Printing 20 random trips from the data.

```
n <- nrow(bikes)
knitr::kable(
bikes[ sample(1:n, size = 20), c('sex', 'age', 'station_start', 'station_end', 'trip_time_seconds', 'lea
caption = "20 random trips"
)
```

Table 1: 20 random trips

sex	age	station_start	station_end	trip_time_seconds	leave_month	leave_day	leave_hour
M	40	21	274	1138	mar	lun	16
M	34	320	63	707	jul	sáb	9
F	33	77	54	420	dic	lun	14
M	27	250	226	305	nov	mié	8
M	46	87	91	426	may	mar	10
F	34	8	87	722	feb	jue	8
M	29	351	69	1336	jun	jue	18
M	33	134	123	344	abr	dom	11

sex	age	station_start	station_end	trip_time_seconds	leave_month	leave_day	leave_hour
M	67	74	53	510	nov	vie	20
M	29	139	405	1362	feb	jue	6
M	51	161	36	611	may	jue	11
F	27	279	315	524	ene	mar	13
M	28	35	174	777	oct	jue	17
F	63	18	4	317	may	vie	19
M	32	32	222	929	oct	mié	9
M	28	314	390	667	oct	lun	19
F	21	392	347	486	jun	jue	18
M	45	87	93	900	feb	vie	9
M	29	182	72	526	ene	dom	11
M	29	11	85	311	jul	mié	19

```r
# Describe the data attributes
col_type <- sapply(bikes, class)
col_type <- col_type %>%
  names() %>%
  sapply( FUN = function(i) { col_type[[i]][1] } )

categorical_col  <- names(col_type[ col_type == 'character'
                                    | col_type == 'factor'
                                    | col_type == 'ordered' ])

numeric_col  <- names(col_type[ col_type != 'character'
                                & col_type != 'factor'
                                & col_type != 'ordered' ])

length(col_type)
```

```
## [1] 22
```

```r
#plot_str(bikes)
col_type
```

```
##               sex               age     station_start       station_end
##          "factor"         "integer"          "factor"          "factor"
##              bike trip_time_seconds       leave_month         leave_day
##          "factor"         "numeric"         "ordered"         "ordered"
##        leave_hour      leave_minute      leave_second       leave_epoch
##         "ordered"         "integer"         "integer"         "integer"
##   leave_epoch_day      arrive_month        arrive_day       arrive_hour
##         "integer"         "ordered"         "ordered"         "ordered"
##     arrive_minute     arrive_second      arrive_epoch  arrive_epoch_day
##         "integer"         "integer"         "integer"         "integer"
##         exceeding    exceeding_hour
##         "logical"         "logical"
```

The data has 22 attributes A lot of them derived from the source table from https://www.ecobici.cdmx.gob.mx/

The original table have only 9 attributes:

1. User gender
2. User age

3. Bike number
4. Station where the trip started
5. Station where the trip ended
6. Start Date
7. Start Time
8. End Date
9. End Time

Carninality of data and counts of categoricals

The cardinality of the data is: 22 variables and 1,751,267.

```r
# Cardinality
dim(bikes)
```

```
## [1] 1751267      22
```

```r
# Counting the categorical variables
categorical_counts <- lapply( bikes[ categorical_col ] ,
        function(df_col) { if( class(df_col)[1] == 'ordered'){
          return(table(df_col))
        } else { sort(table(df_col), decreasing = TRUE) } } )

categorical_counts$sex
```

```
## df_col
##       M       F
## 1322466  428801
```

```r
( n_stations_start <- length(categorical_counts$station_start) )
```

```
## [1] 461
```

```r
( n_stations_end <- length(categorical_counts$station_start) )
```

```
## [1] 461
```

```r
# Because there are 461 stations
# Just showing the 50 more visited
categorical_counts$station_start[1:50]
```

```
## df_col
##    27   271     1    18    21    15    36    25    43    23    64    41
## 21185 16194 14295 14077 12925 10990 10939 10874 10537 10433 10421 10377
##   217    47   182    19    74    16   266    86   208    28    24    10
##  9870  9837  9786  9738  9720  9709  9664  9387  9225  9216  8989  8847
##   146    32    38   174    17    84   211   136    20   134   194   261
##  8676  8653  8651  8597  8518  8382  8319  8188  8113  8096  7899  7888
##    54   158    53    14    56   242    13   272   270    51    46    63
##  7883  7875  7866  7865  7834  7788  7778  7665  7661  7660  7592  7504
##    85   116
##  7346  7335
```

```r
categorical_counts$station_end[1:50]
```

```
## df_col
##    27   266     1    18   271    43    21   217    64    25   182    36
## 20992 15216 15101 14585 13490 12684 12034 11349 11112 11073 10979 10850
##    47    16    15    74    23   267    38   174   134   146    19   116
## 10838 10698 10327 10243 10086  9923  9514  9377  9362  9205  9013  8971
##    17   136    28    51   295    24    32    41    56    46   141    52
##  8916  8833  8827  8630  8548  8496  8433  8430  8399  8365  8267  8111
##    54    63    14   270    29   208    59    53   158     7   261    10
##  8084  8009  8008  7976  7941  7939  7829  7827  7801  7783  7777  7722
##    84    20
##  7657  7651
```

```r
# Because there are 6894 bikes, just showing the first 50 places
( n_bikes <- length(categorical_counts$bike) )
```

```
## [1] 6894
```

```r
categorical_counts$bike[1:50]
```

```
## df_col
## 9359 4229 2264 9212 9217 4155 9237 8479 9312 9434 9504 2494 7377 8432 9369
##  421  416  412  408  407  406  406  405  404  404  404  403  403  402  402
## 1534 2758 2333 3290 4352 6942 8076 2698 3960 7243 8937 2693 4314 9274 2565
##  401  401  399  399  399  399  399  398  398  398  398  397  397  397  396
## 9372 1832 9342 2100 2954 3806 9261 1897 9354 3124 3288 7690 9050 9315 1561
##  396  395  395  394  394  394  394  393  393  392  392  392  392  391  390
## 1604 1722 2686 2697 2591
##  390  390  390  390  389
```

```r
categorical_counts$leave_month
```

```
## df_col
##    ene    feb    mar    abr    may    jun    jul    ago    sep    oct
## 162178 159218 175539 140949 174554 170627 154423  22808 141517 164753
##    nov    dic
## 159214 125487
```

```r
categorical_counts$leave_day
```

```
## df_col
##    lun    mar    mié    jue    vie    sáb    dom
## 281142 313370 313694 305100 296650 131195 110116
```

```r
categorical_counts$leave_hour
```

```
## df_col
##      0      1      2      3      4      5      6      7      8      9
##  27855     17     12      8      8   7725  33711  85448 164878 128535
##     10     11     12     13     14     15     16     17     18     19
##  85733  77114  81493 100035 124238 122955 101783 116712 163123 136642
##     20     21     22     23
##  84593  56268  33922  18459
```

```r
categorical_counts$arrive_month
```

```
## df_col
```

```
##    ene    feb    mar    abr    may    jun    jul    ago    sep    oct
## 162170 159219 175535 140960 174543 170638 154426  22802 141514 164754
##    nov    dic
## 159206 125500
```

```
categorical_counts$arrive_day
```

```
## df_col
##    lun    mar    mié    jue    vie    sáb    dom
## 281022 313306 313639 305009 296617 131355 110319
```

```
categorical_counts$arrive_hour
```

```
## df_col
##      0      1      2      3      4      5      6      7      8      9
##  30323    305     34     15     12   5448  27135  68852 149994 146027
##     10     11     12     13     14     15     16     17     18     19
##  90853  76303  79851  95652 122650 124003 104843 109033 154404 147201
##     20     21     22     23
##  96879  62234  38232  20984
```

```
# Summary over all the variables
summary_bikes <- summary(bikes)
sd_bikes <- apply(bikes, 2, sd)
# Printing summaries
summary_bikes
```

```
##  sex              age         station_start      station_end
##  F: 428801   Min.   : 16.0   27      :  21185   27      :  20992
##  M:1322466   1st Qu.: 27.0   271     :  16194   266     :  15216
##              Median : 32.0   1       :  14295   1       :  15101
##              Mean   : 34.7   18      :  14077   18      :  14585
##              3rd Qu.: 40.0   21      :  12925   271     :  13490
##              Max.   :117.0   15      :  10990   43      :  12684
##                              (Other):1661601   (Other):1659199
##       bike          trip_time_seconds  leave_month     leave_day
##  9359   :    421   Min.   :    1     mar    :175539   lun:281142
##  4229   :    416   1st Qu.:  398     may    :174554   mar:313370
##  2264   :    412   Median :  640     jun    :170627   mié:313694
##  9212   :    408   Mean   :  812     oct    :164753   jue:305100
##  9217   :    407   3rd Qu.: 1033     ene    :162178   vie:296650
##  4155   :    406   Max.   :14397     feb    :159218   sáb:131195
##  (Other):1748797                     (Other):744398   dom:110116
##    leave_hour      leave_minute    leave_second    leave_epoch
##  8      :164878   Min.   : 0.00   Min.   : 0.00   Min.   :1.483e+09
##  18     :163123   1st Qu.:14.00   1st Qu.:15.00   1st Qu.:1.490e+09
##  19     :136642   Median :29.00   Median :29.00   Median :1.497e+09
##  9      :128535   Mean   :29.08   Mean   :29.49   Mean   :1.498e+09
##  14     :124238   3rd Qu.:44.00   3rd Qu.:44.00   3rd Qu.:1.507e+09
##  15     :122955   Max.   :59.00   Max.   :59.00   Max.   :1.515e+09
##  (Other):910896
##  leave_epoch_day  arrive_month    arrive_day    arrive_hour
##  Min.   :    0   mar    :175535   lun:281022   18     :154404
##  1st Qu.:35652   may    :174543   mar:313306   8      :149994
##  Median :52745   jun    :170638   mié:313639   19     :147201
##  Mean   :51258   oct    :164754   jue:305009   9      :146027
```

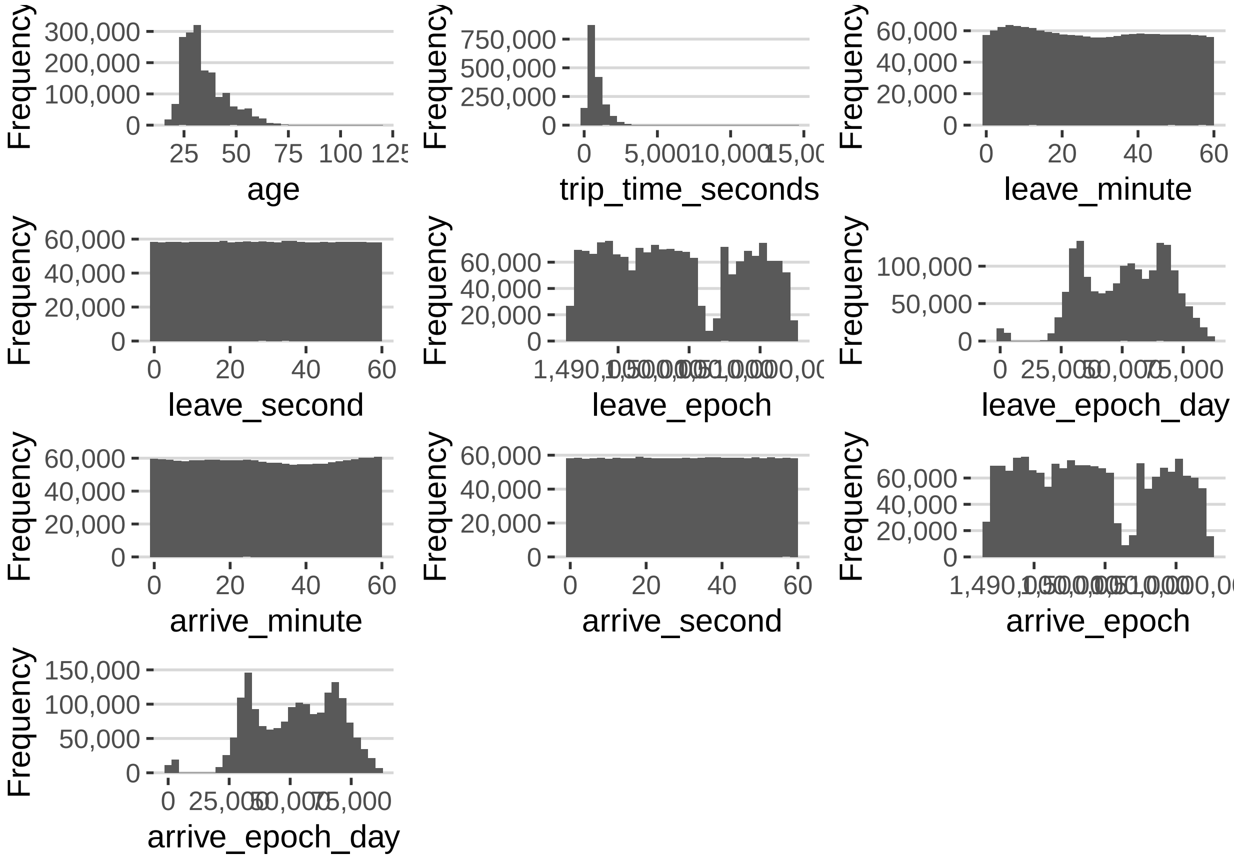
```
##  3rd Qu.:65918   ene     :162170   vie:296617   15      :124003
##  Max.   :86399   feb     :159219   sáb:131355   14      :122650
##                  (Other):744408   dom:110319   (Other):906988
##  arrive_minute   arrive_second    arrive_epoch       arrive_epoch_day
##  Min.   : 0.00   Min.   : 0.00   Min.   :1.483e+09   Min.   :    0
##  1st Qu.:14.00   1st Qu.:15.00   1st Qu.:1.490e+09   1st Qu.:36299
##  Median :29.00   Median :30.00   Median :1.497e+09   Median :53470
##  Mean   :29.47   Mean   :29.51   Mean   :1.498e+09   Mean   :51932
##  3rd Qu.:45.00   3rd Qu.:44.00   3rd Qu.:1.507e+09   3rd Qu.:66815
##  Max.   :59.00   Max.   :59.00   Max.   :1.515e+09   Max.   :86397
##
##  exceeding       exceeding_hour
##  Mode :logical   Mode :logical
##  FALSE:1730873   FALSE:1743428
##  TRUE :20394     TRUE :7839
##
##
##
##
```

```
sd_bikes
```

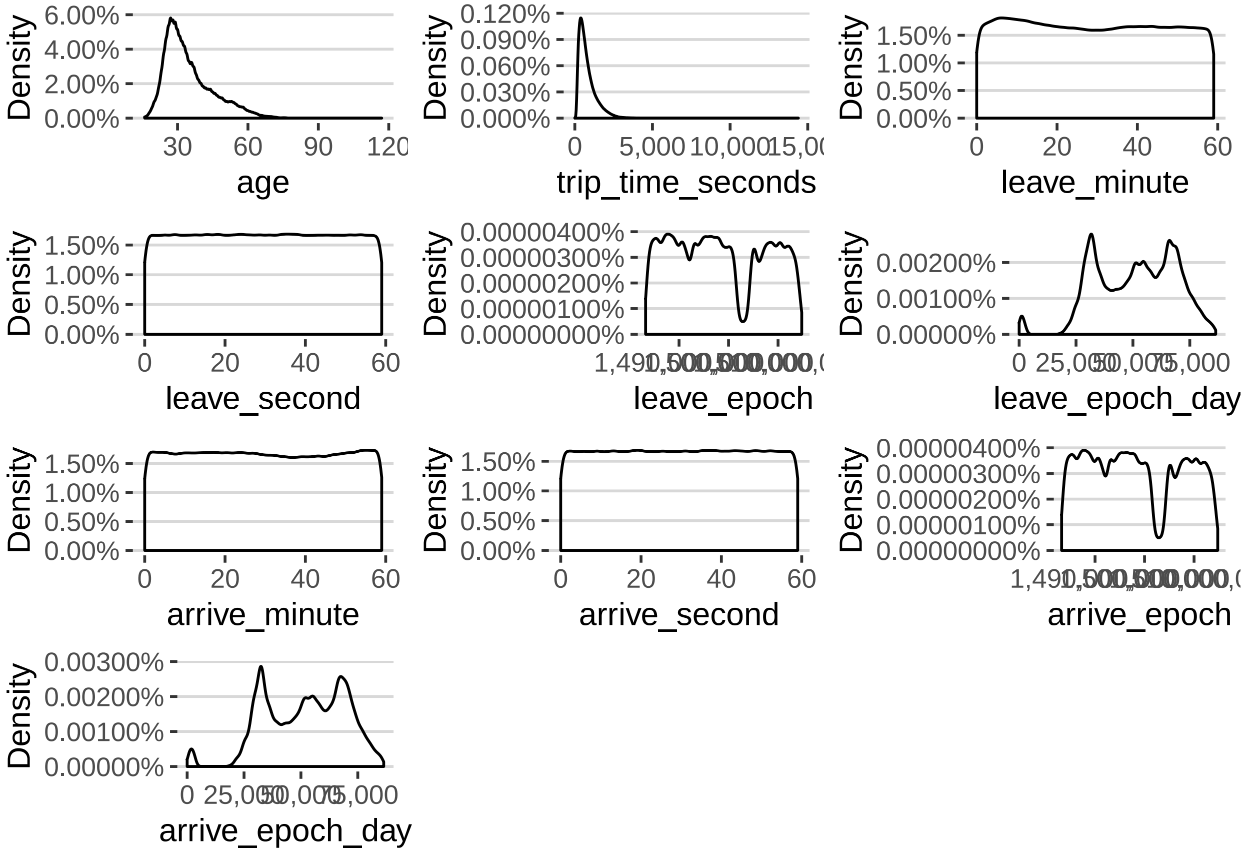
```
##              sex              age    station_start      station_end
##               NA     1.028721e+01     1.218290e+02     1.220127e+02
##             bike trip_time_seconds     leave_month        leave_day
##     3.055649e+03     6.709530e+02               NA               NA
##       leave_hour      leave_minute     leave_second      leave_epoch
##     4.840833e+00     1.739925e+01     1.730555e+01     9.202912e+06
##  leave_epoch_day      arrive_month       arrive_day      arrive_hour
##     1.739096e+04               NA               NA     4.868537e+00
##    arrive_minute     arrive_second     arrive_epoch arrive_epoch_day
##     1.742911e+01     1.730927e+01     9.202926e+06     1.750149e+04
##        exceeding    exceeding_hour
##               NA               NA
```

Distributions

```
######  Distributions  ######

plot_histogram(bikes, ggtheme = theme_hc())
```



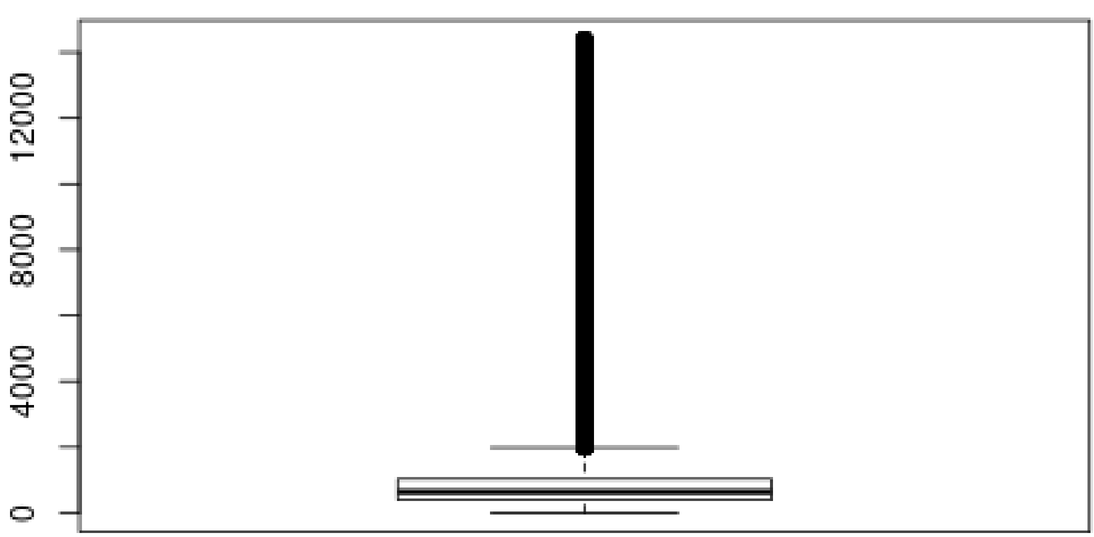

```r
plot_density(bikes, ggtheme = theme_hc())
```



In the approximation of the probability density functions of the numeric variables we can see some patterns. For example people tend to start or end their trips near exactly complete hours, as instance near the 0 or 60 minutes mark. The distribution on the seconds is uniform as expected, there isn't a preferred second to start trips.

Boxplots

```
p1 <- boxplot(bikes$trip_time_seconds)
```

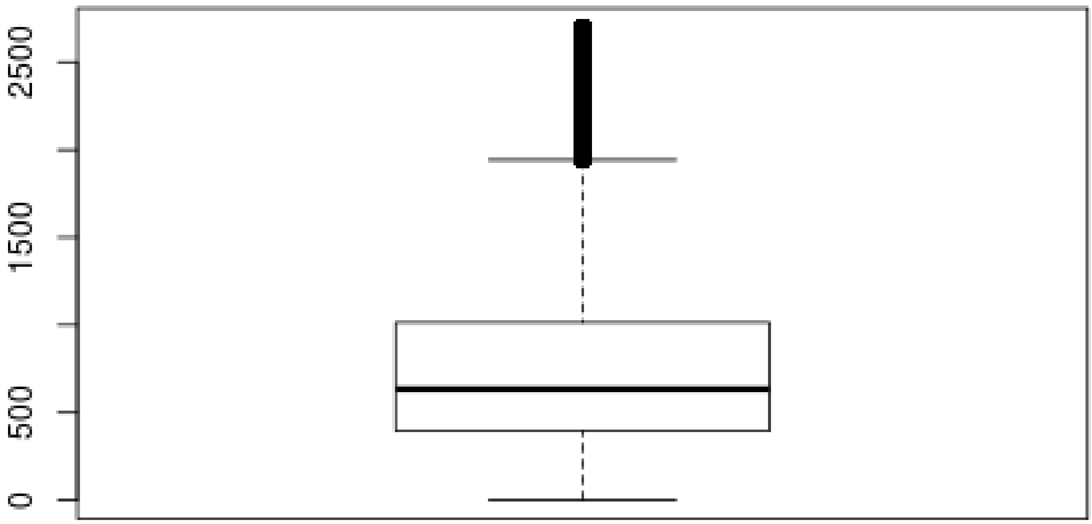


Box plot of

trip time we can see a lot of outliers people that exceeded the 45 min tolerance

What happens if we remove the 45 min exceeding trips?

```
boxplot(bikes$trip_time_seconds[ !bikes$exceeding ])
```




```
summary(bikes$trip_time_seconds[ !bikes$exceeding ])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   395.0   633.0   771.8  1014.0  2700.0
```

Still removing the exceeded time trips see some outliers, most people is far away from the 45 min mark and makes on average **10 minutes** the median is actually 3267 seconds equal to 10 min with 30 seg

```
# Make categories over trip time
# 6 equal size categories
q_trip <- cut2(bikes$trip_time_seconds, g = 6 )
levels(q_trip)
```

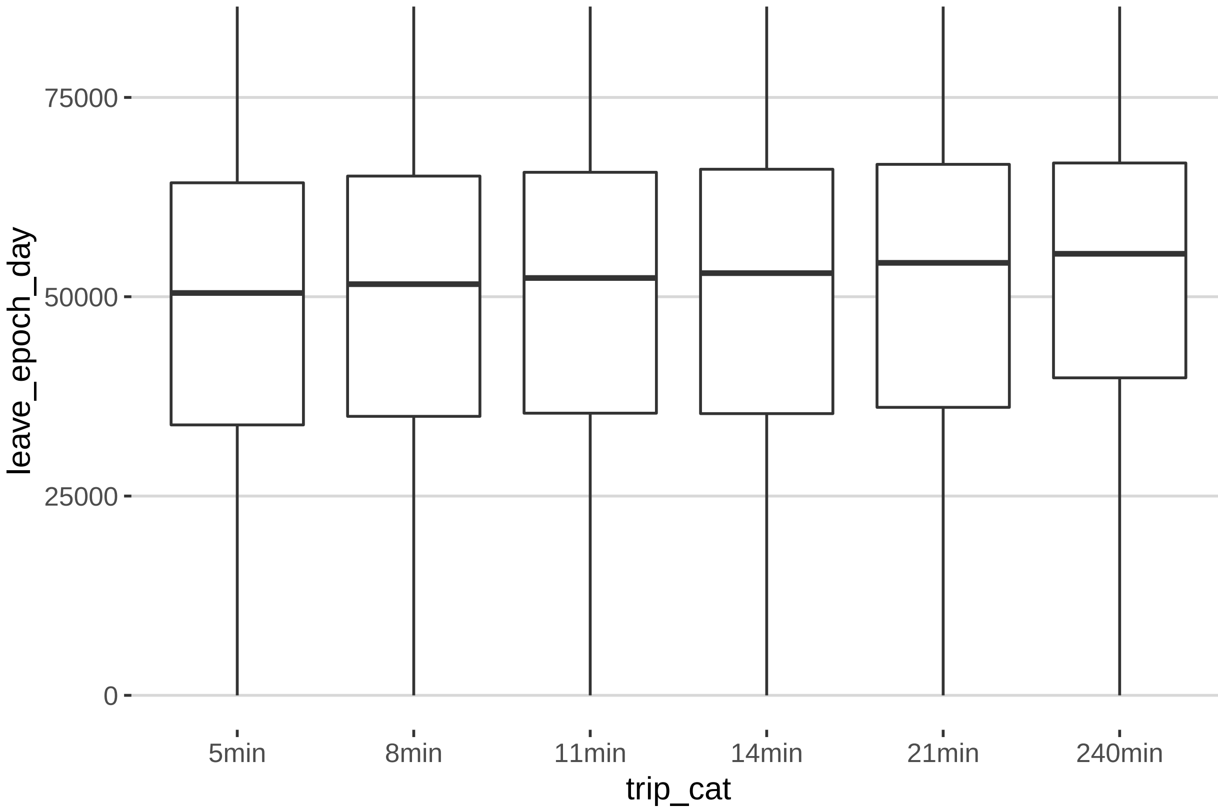
```
## [1] "[   1,  326)" "[ 326,  473)" "[ 473,  641)" "[ 641,  870)"
## [5] "[ 870, 1272)" "[1272,14397]"
```

```
q_sec <- c(326,473,641,870,1272,14397)
round(q_sec / 60)
```

```
## [1]   5   8  11  14  21 240
```

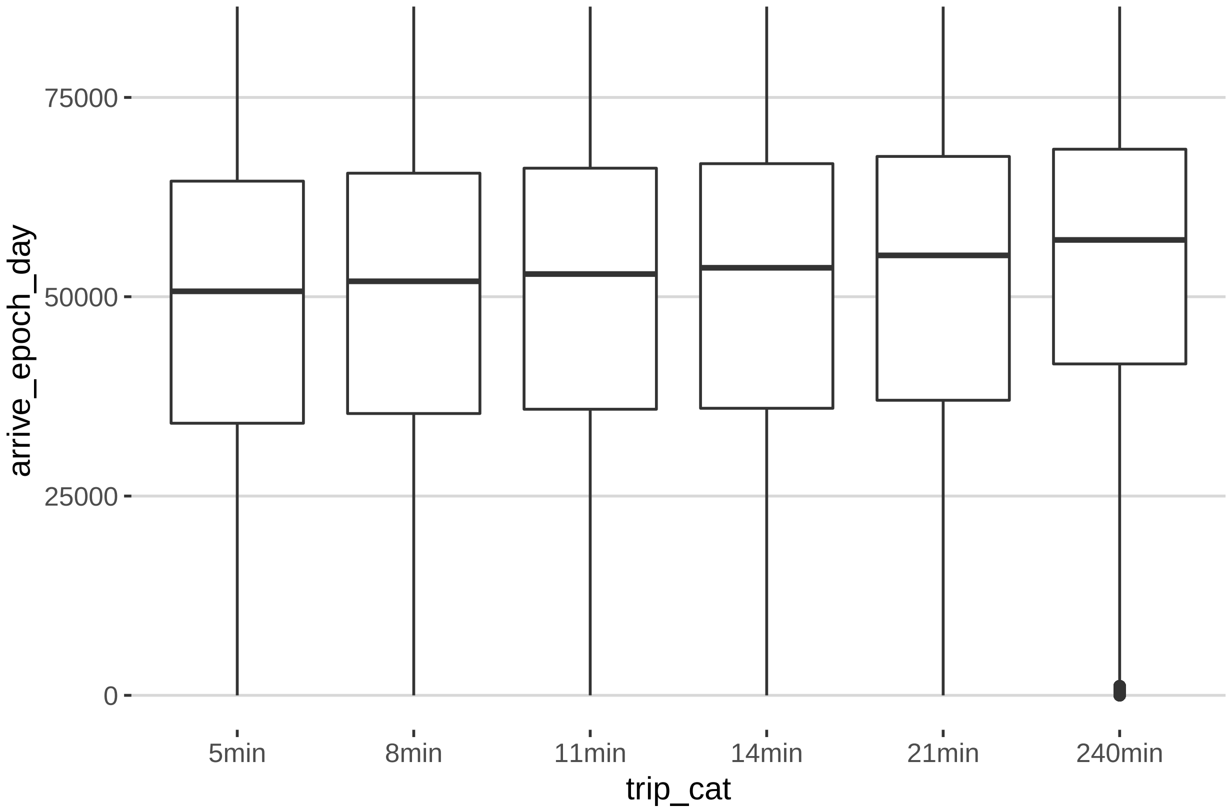
```
old_levels <- levels(q_trip)
new_levels <- paste0( round(q_sec / 60) , 'min')
levels(q_trip) <- new_levels

bikes$trip_cat <- q_trip
```

```
ggplot(bikes, aes(x=trip_cat, y=leave_epoch_day) )+
  geom_boxplot()+theme_hc()
```



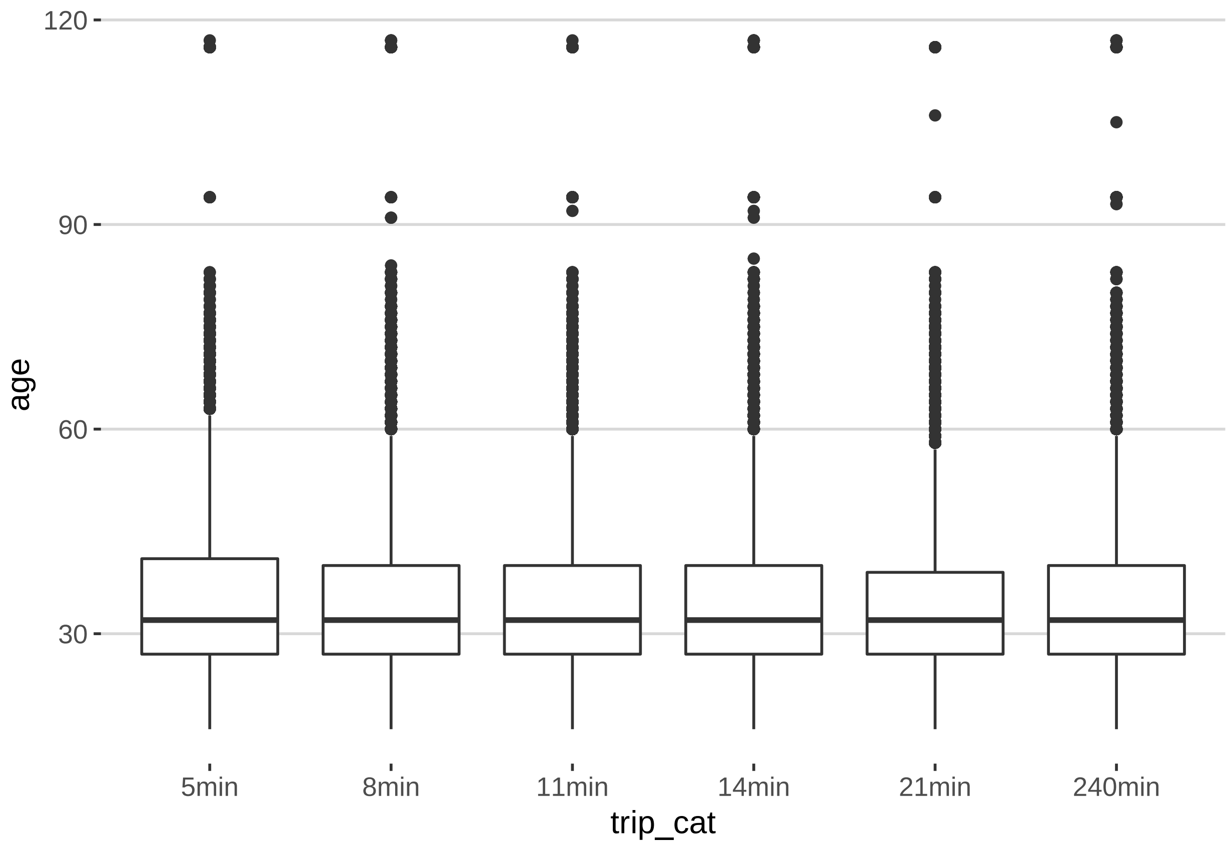
Box plot of leave time, most people tend to start theirs trips at the same hour in the day regardless of how much they would take to complete it.

```
ggplot(bikes, aes(x=trip_cat, y=arrive_epoch_day) )+
  geom_boxplot()+theme_hc()
```



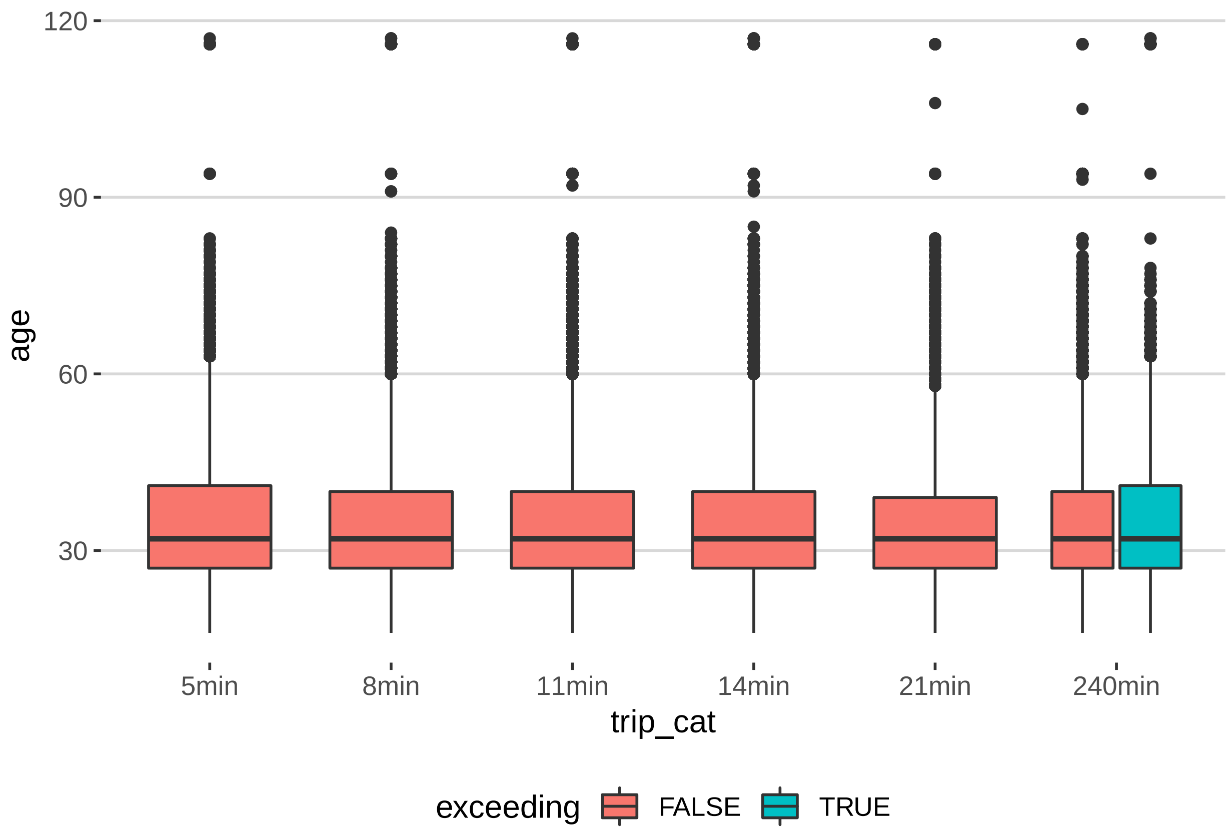
Box plot of arrive time, most people tend to end theirs trips at the same hour in the day regardless of how much they have taken to complete it.

```
ggplot(bikes, aes(x=trip_cat, y=age) )+
  geom_boxplot()+theme_hc()
```



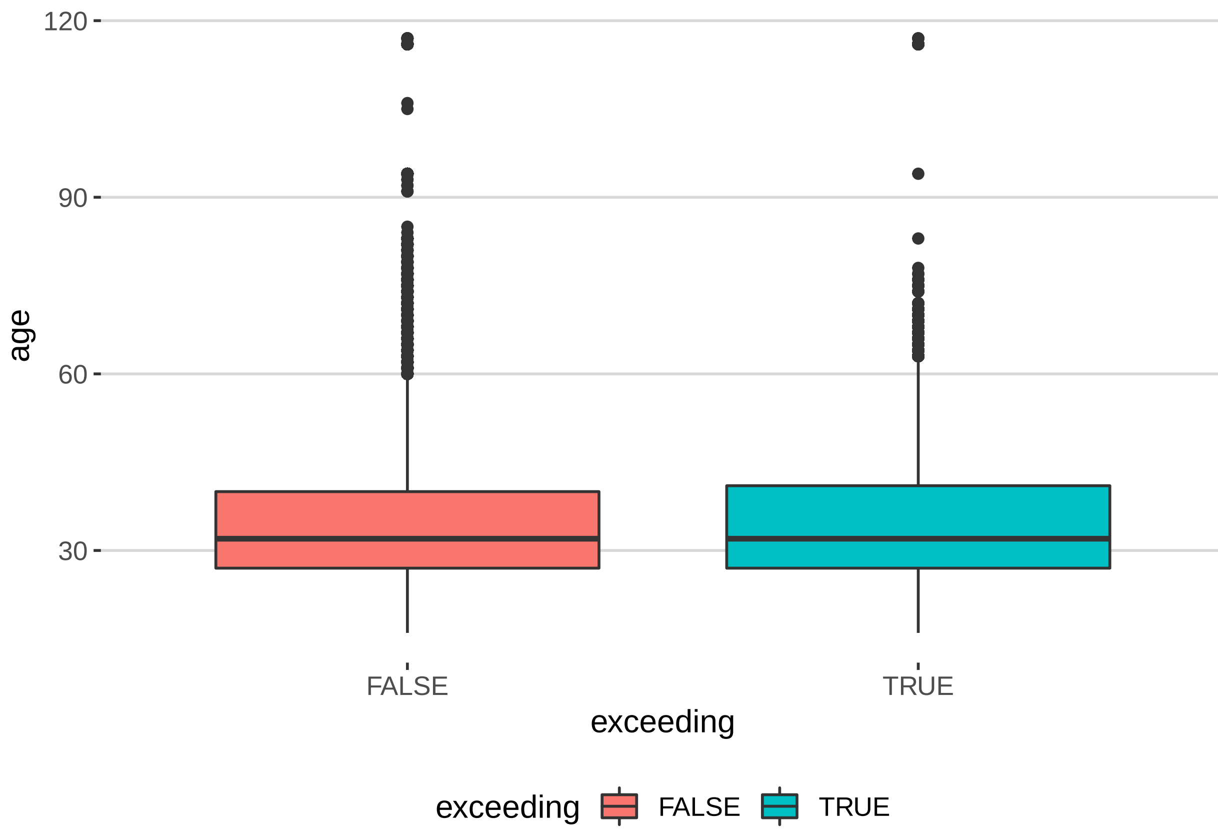
Box plot of age divided by trip time. It doesn't matter to much your age in your trip time

```
ggplot(bikes, aes(x=trip_cat, y=age, fill = exceeding) )+
  geom_boxplot()+theme_hc()
```



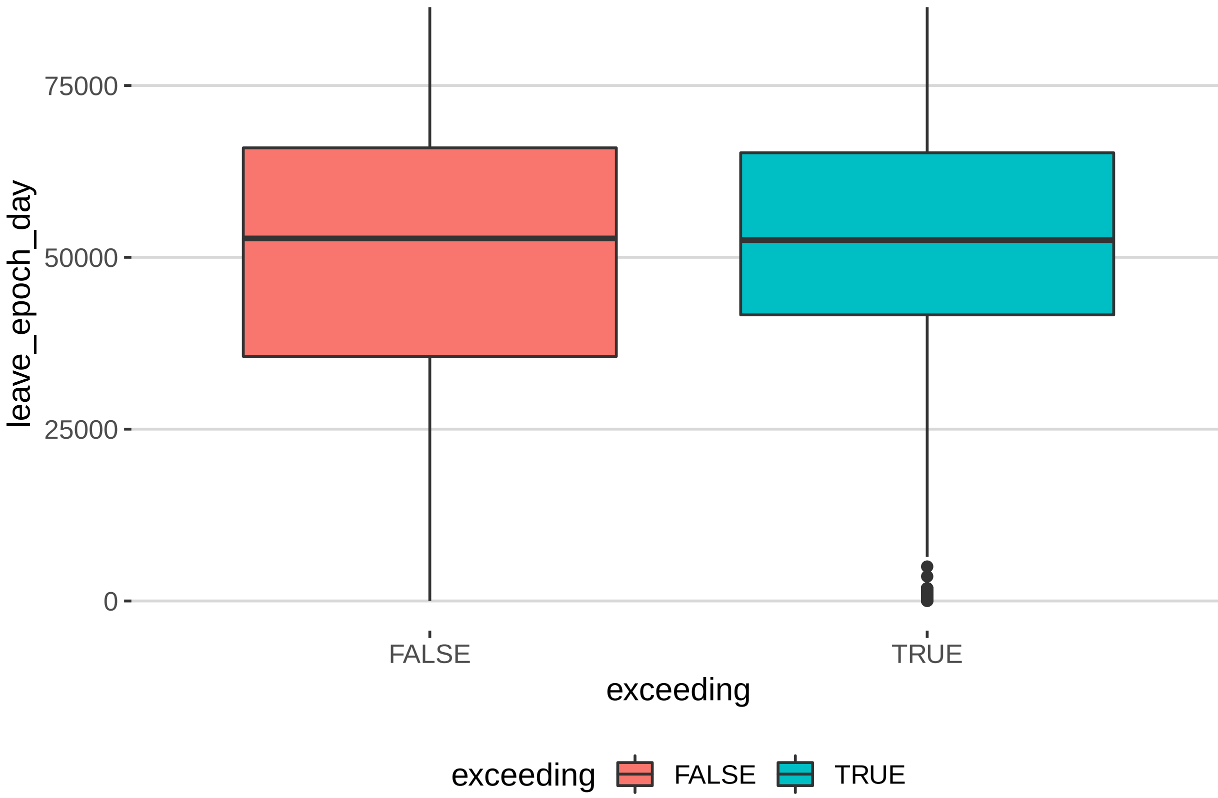
Box plot of age divided by trip time. It doesn't matter to much your age in your trip time even when separated by exceeding time trips.

```
ggplot(bikes, aes(x=exceeding, y=age, fill = exceeding) )+
  geom_boxplot()+theme_hc()
```



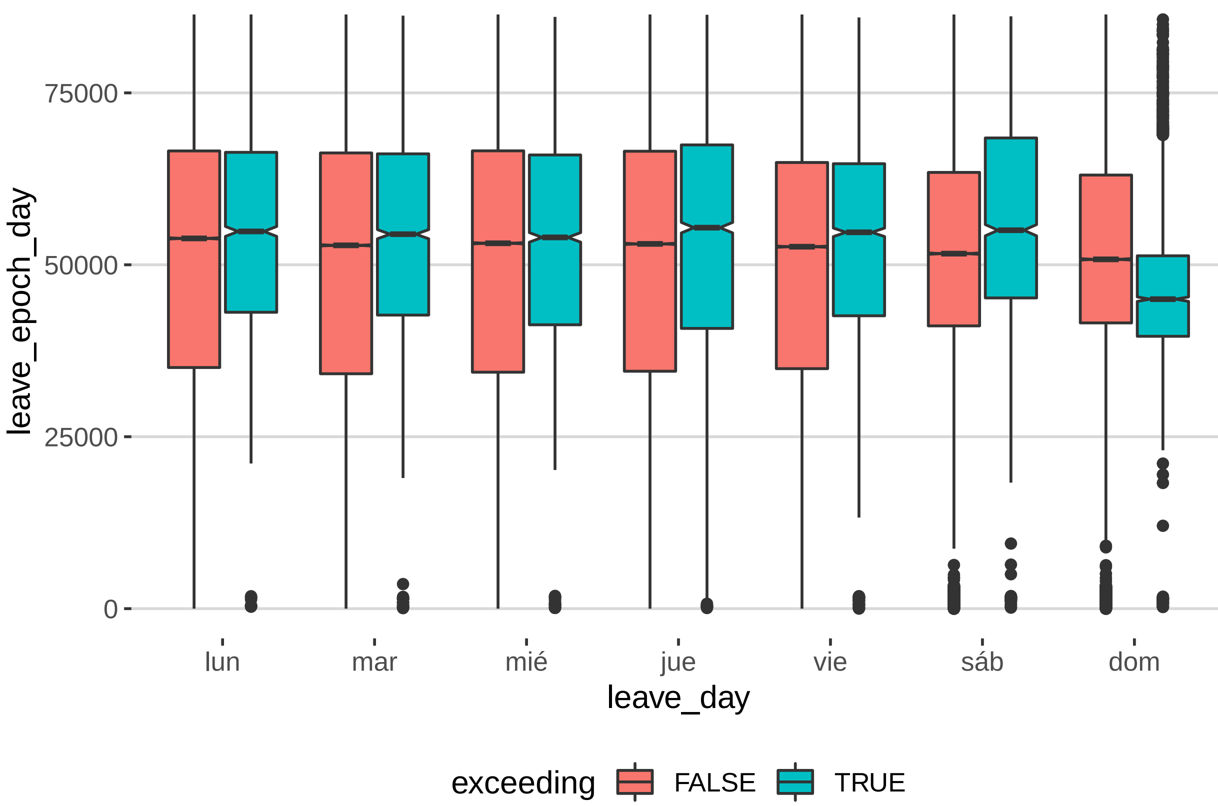
The distribution is similar of the people that exceeded the time limit return vs age.

```
ggplot(bikes, aes(x=exceeding, y=leave_epoch_day, fill = exceeding) )+
  geom_boxplot()+theme_hc()
```



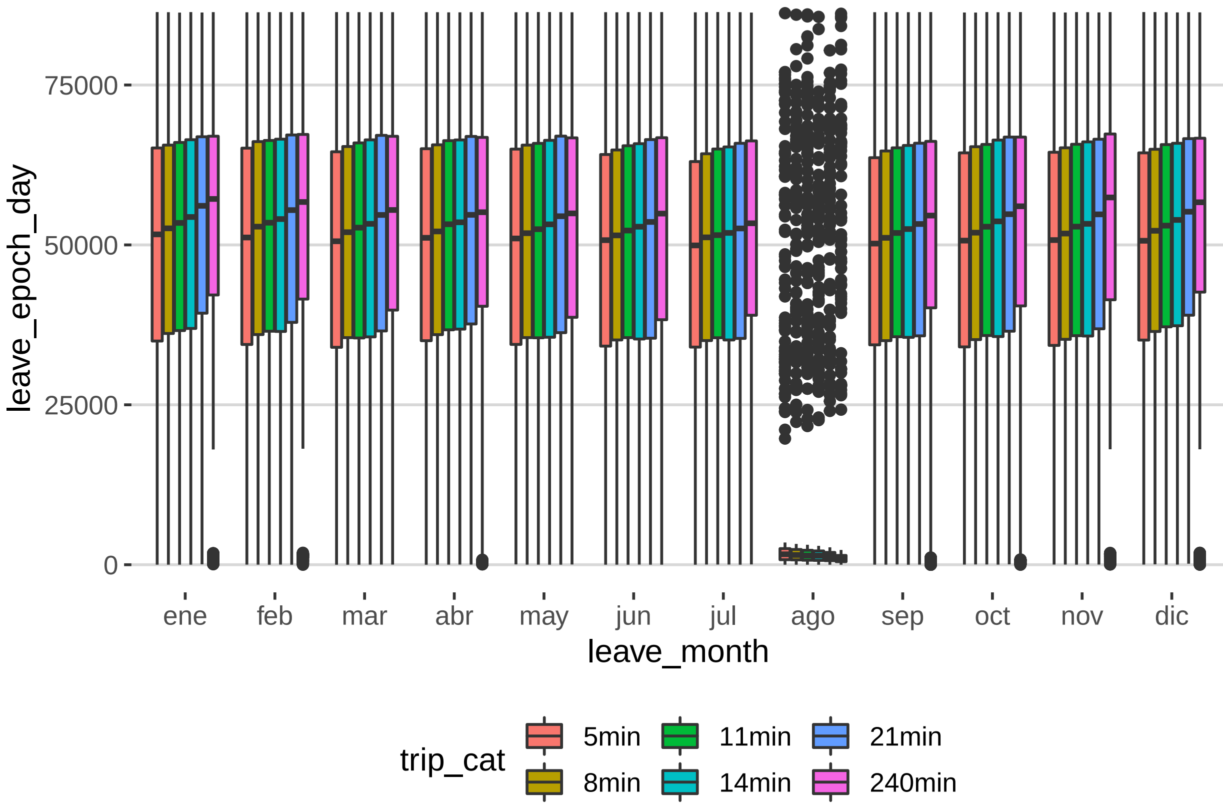
The distribution is similar of the people that exceeded the time limit return vs their start time

```
ggplot(bikes, aes(x=leave_day, y=leave_epoch_day, fill = exceeding) )+
  geom_boxplot(notch = TRUE)+theme_hc()
```



The medians on Sunday are a little bit different probably because Sunday is the day with fewer trips, so more variance estimating the median could be expected.

```
ggplot(bikes, aes(x=leave_month, y=leave_epoch_day, fill = trip_cat) )+
  geom_boxplot()+theme_hc()
```

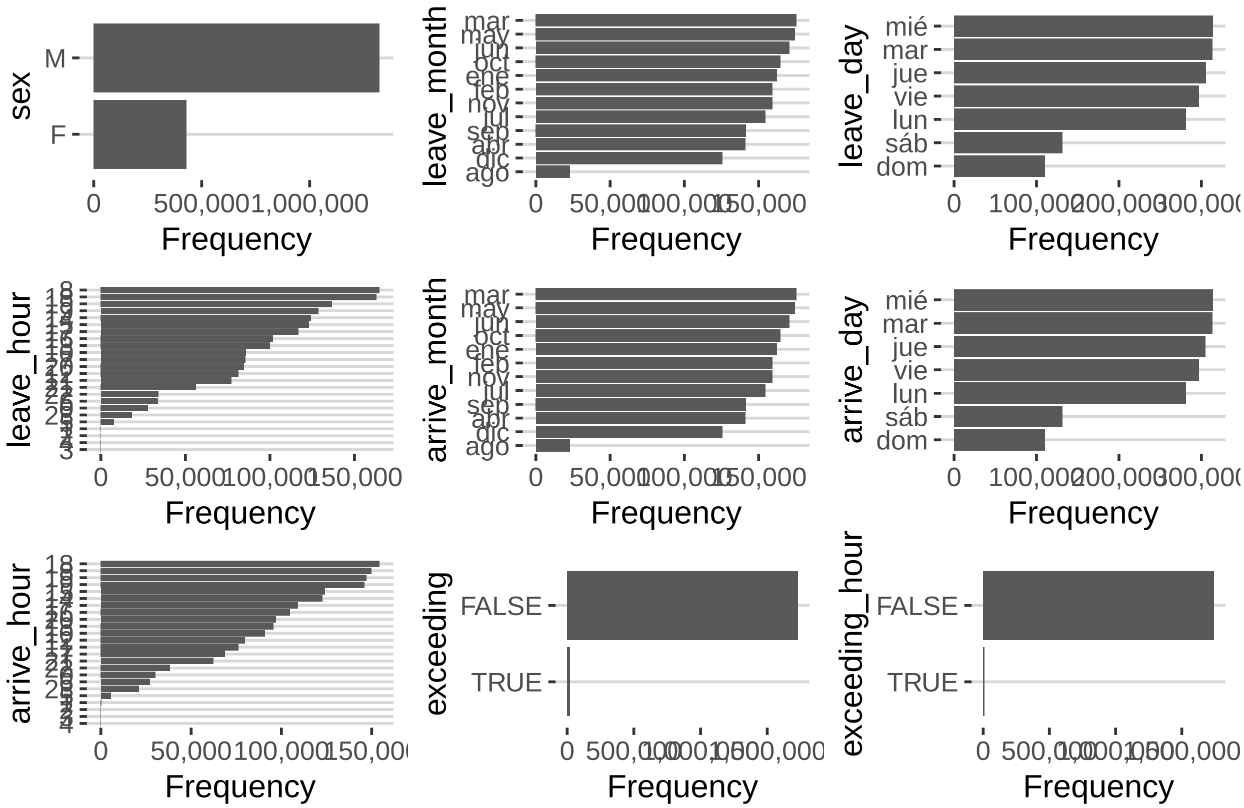


August has a strange pattern, is different from all the other months, and the majority of trips were made at night. I was expecting seeing some change in September due to the 2017 earthquake, but September looks similar to the other months.

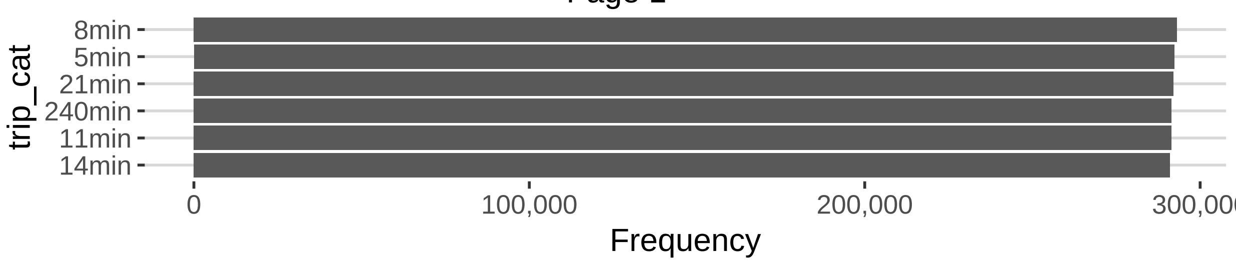
Bar plots

```
plot_bar(bikes, ggtheme = theme_hc())
```

```
## 3 columns ignored with more than 50 categories.
## station_start: 460 categories
## station_end: 462 categories
## bike: 6893 categories
```

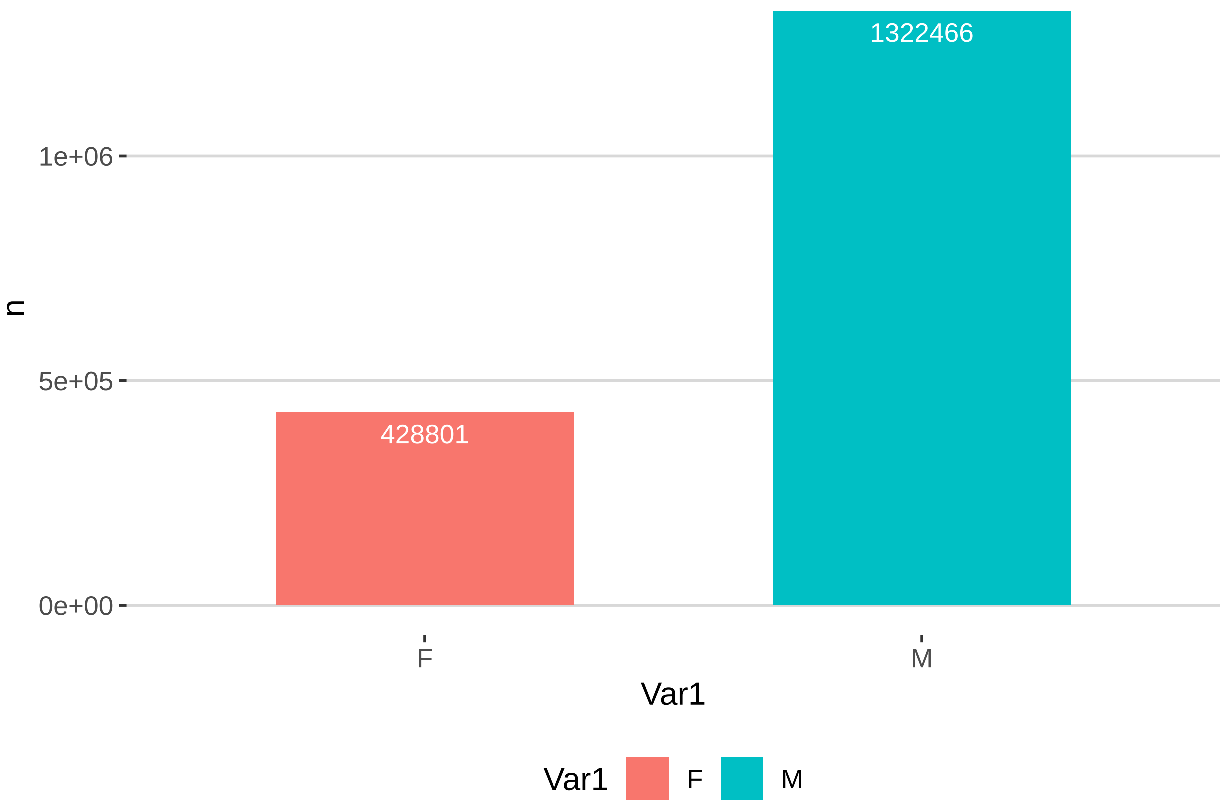

sex
M
F
Frequency
leave_month
mar
may
jun
oct
ene
feb
nov
jul
sep
abr
dic
ago
leave_day
mié
mar
jue
vie
lun
sáb
dom
leave_hour
arrive_month
arrive_day
arrive_hour
exceeding
FALSE
TRUE
exceeding_hour

Page 1


trip_cat
8min
5min
21min
240min
11min
14min
0
100,000
200,000
Frequency

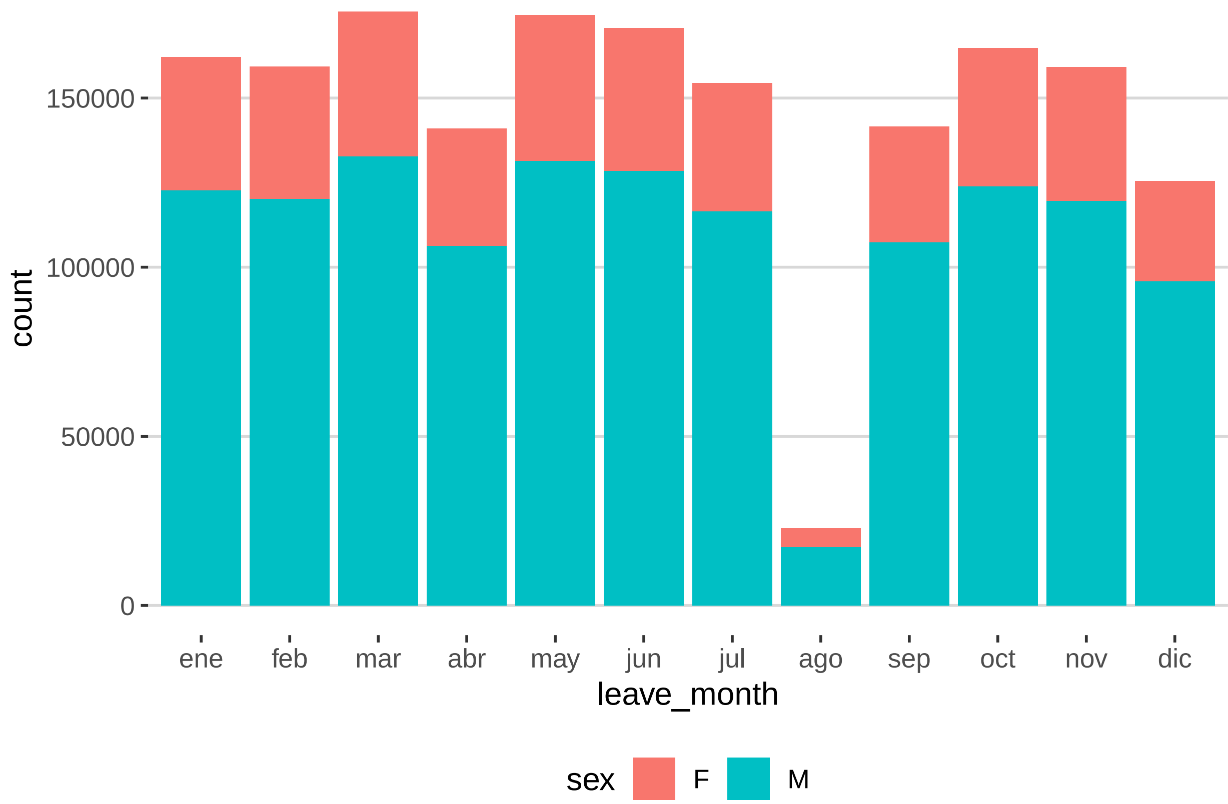
Page 2

```r
# Almost are the users are men
d <- as.tibble( table(bikes$sex) )
ggplot( d, aes( x =  Var1, y = n, fill = Var1) )+
  geom_bar(stat = 'identity', width = 0.6)+
  geom_text(aes(label=n), vjust=1.6, color="white", size=3.5)+
  theme_hc()
```



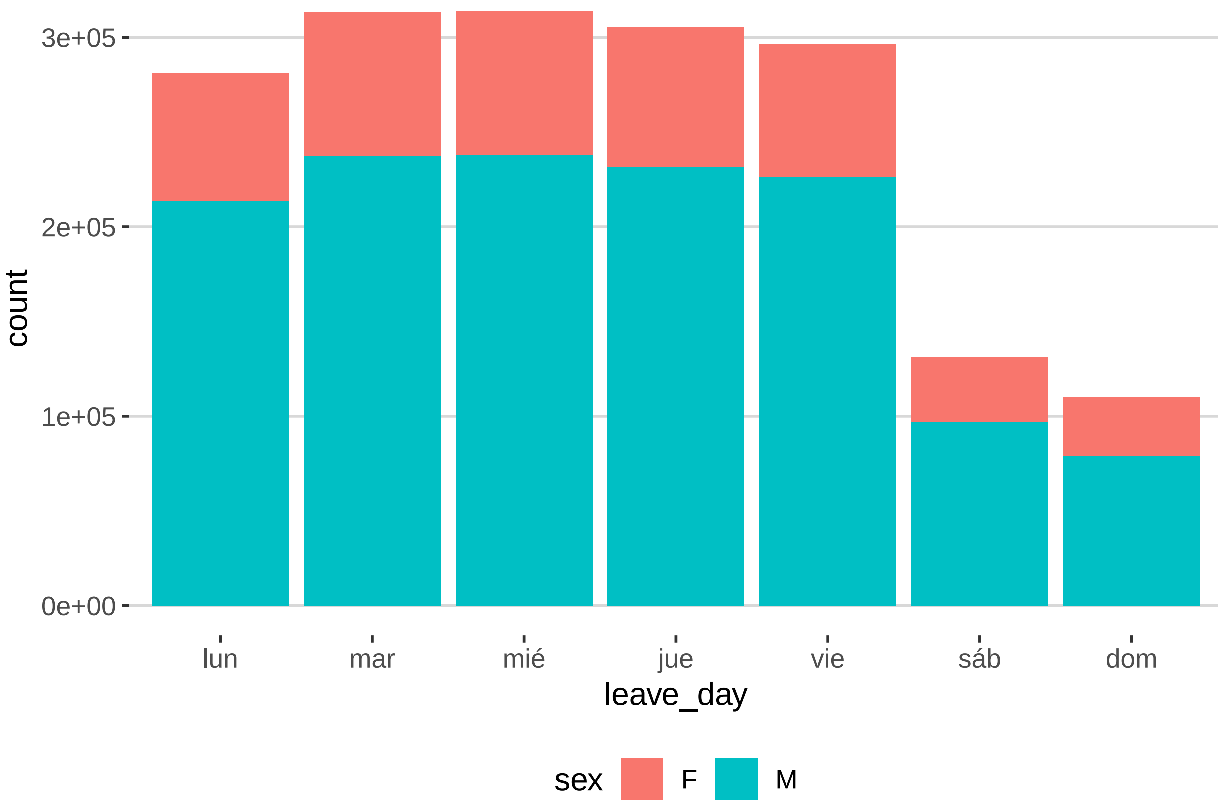
Almost are the users are men

```r
# August had a very few trips
ggplot( bikes, aes( x = leave_month, fill = sex  ) )+
  geom_bar()+theme_hc()
```



August is the month with the lowest number of trips, and in comparison to the other months is one order of magnitude below.

```
# As expected the traffic on the weekends is less
ggplot( bikes, aes( x = leave_day, fill = sex  ) )+
  geom_bar()+theme_hc()
```



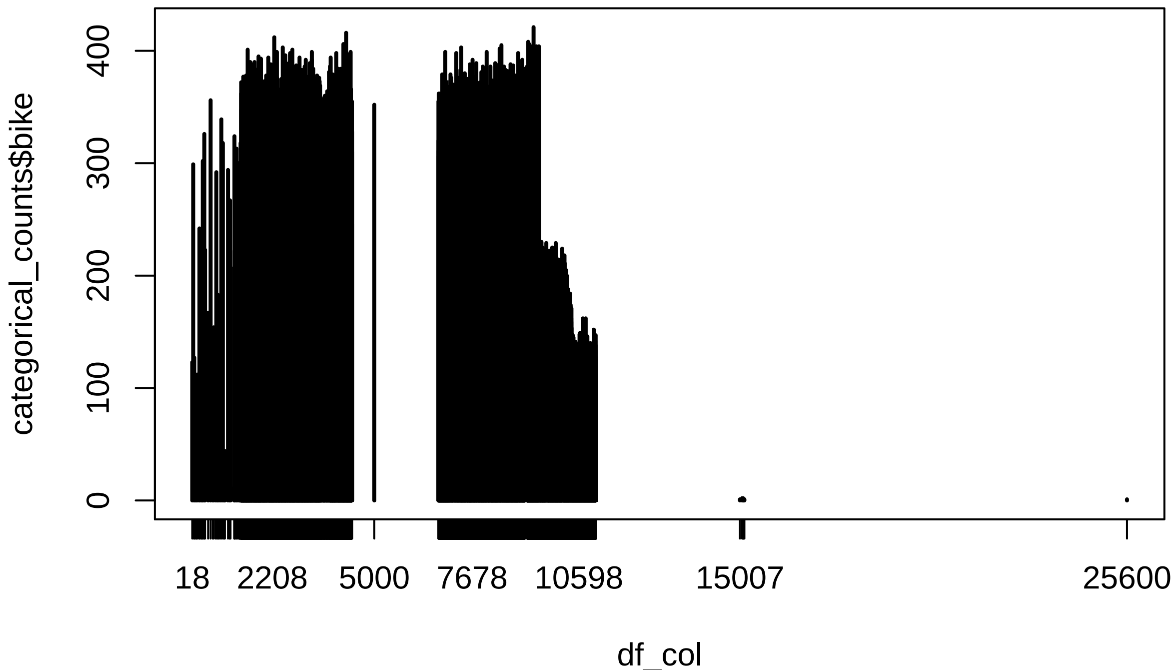
As expected the traffic on the weekends is less.

```
# The most of the trips are either at eigth in the morning or at six in the afternoon
# This is consistent with the typical day work
# Also the count is 0 over 1,2,3,4 hours, when the service is closed.
ggplot( bikes, aes( x = bikes$leave_hour, fill = sex  ) )+
  geom_bar()+theme_hc()
```

Most of the trips are either at eight in the morning or at six in the afternoon.

This is consistent with the typical day work. Also the count is almost 0 over 1,2,3,4 hours, when the service is closed.

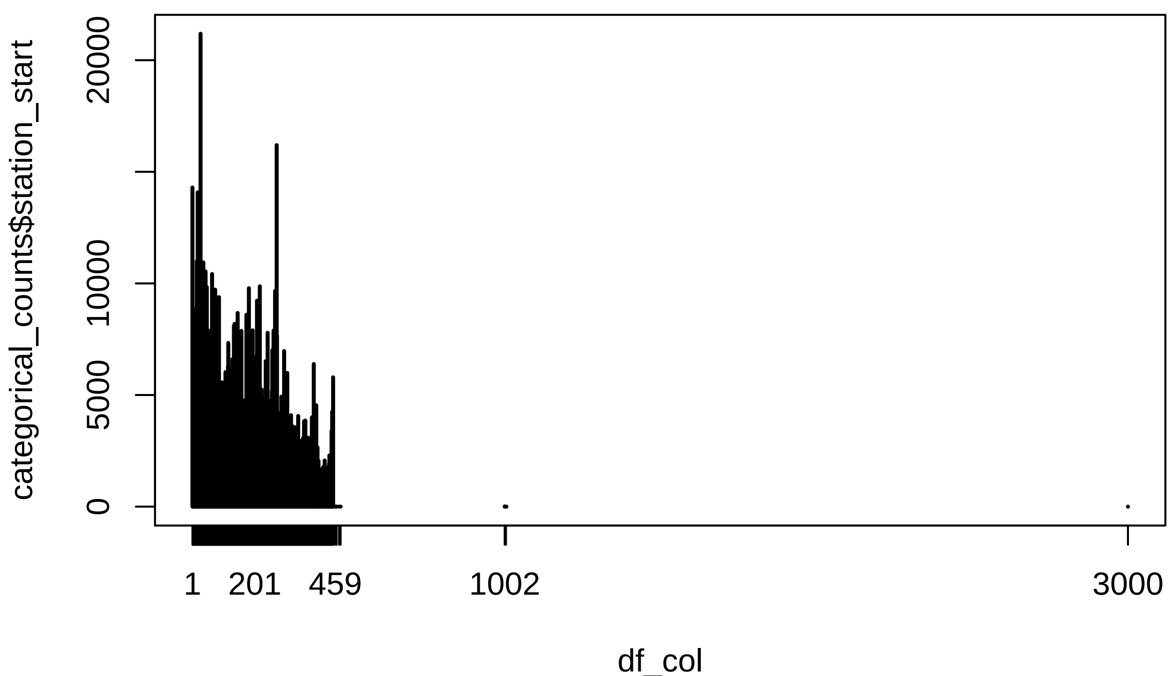
```
# Bar plots of bikes
# there is a group bikes that is under used
# could be due to mechanical problems
plot(categorical_counts$bike)
```



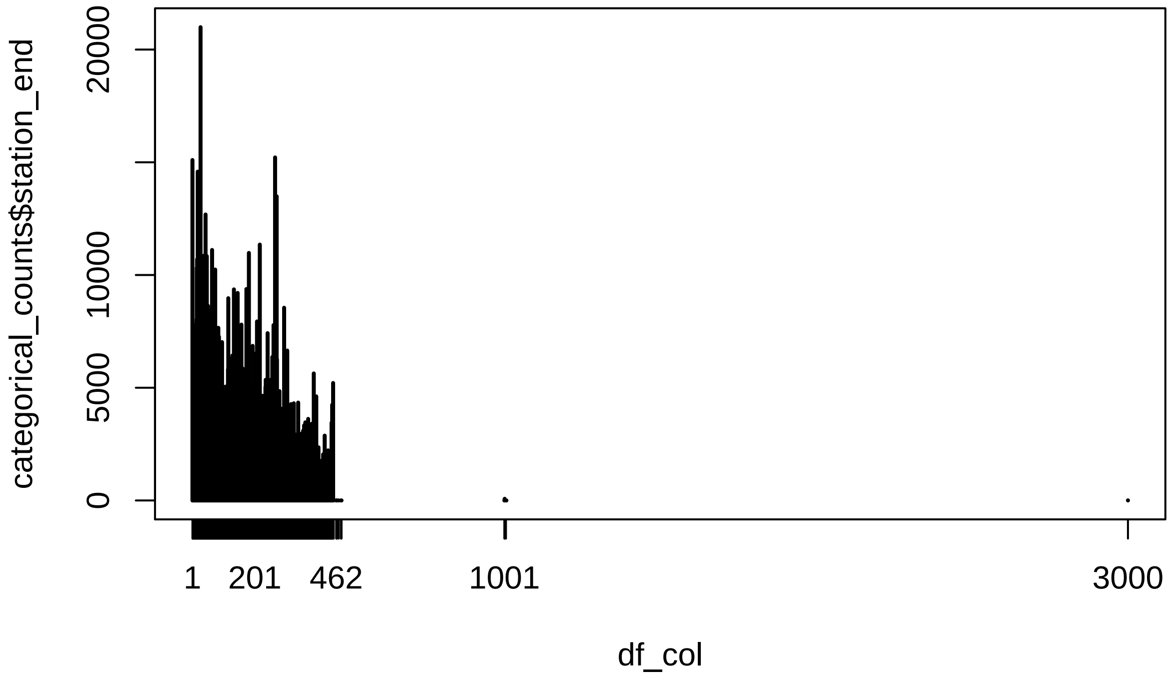
There is a group bikes that is under used could be due to mechanical problems on some bikes.

Plotting station usage:

```r
# Plotting stations
plot(categorical_counts$station_start)
```

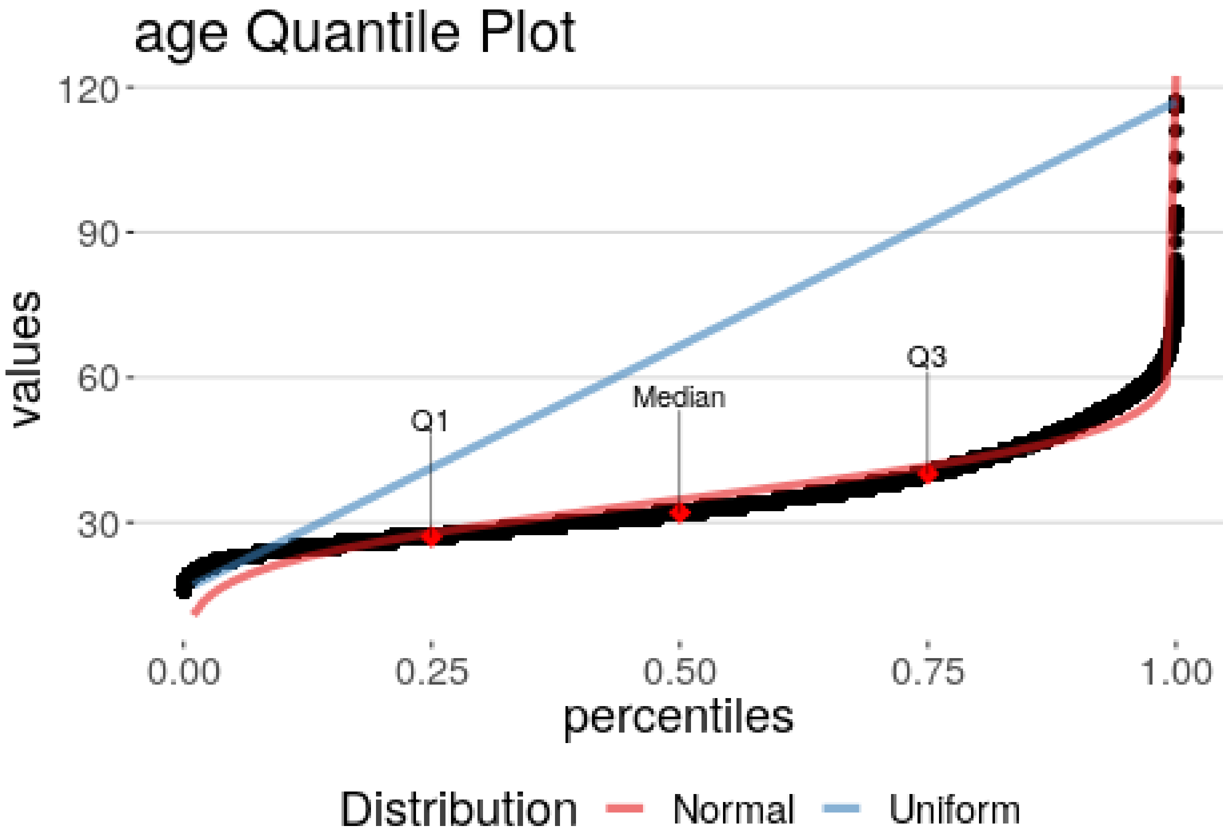



```r
plot(categorical_counts$station_end)
```



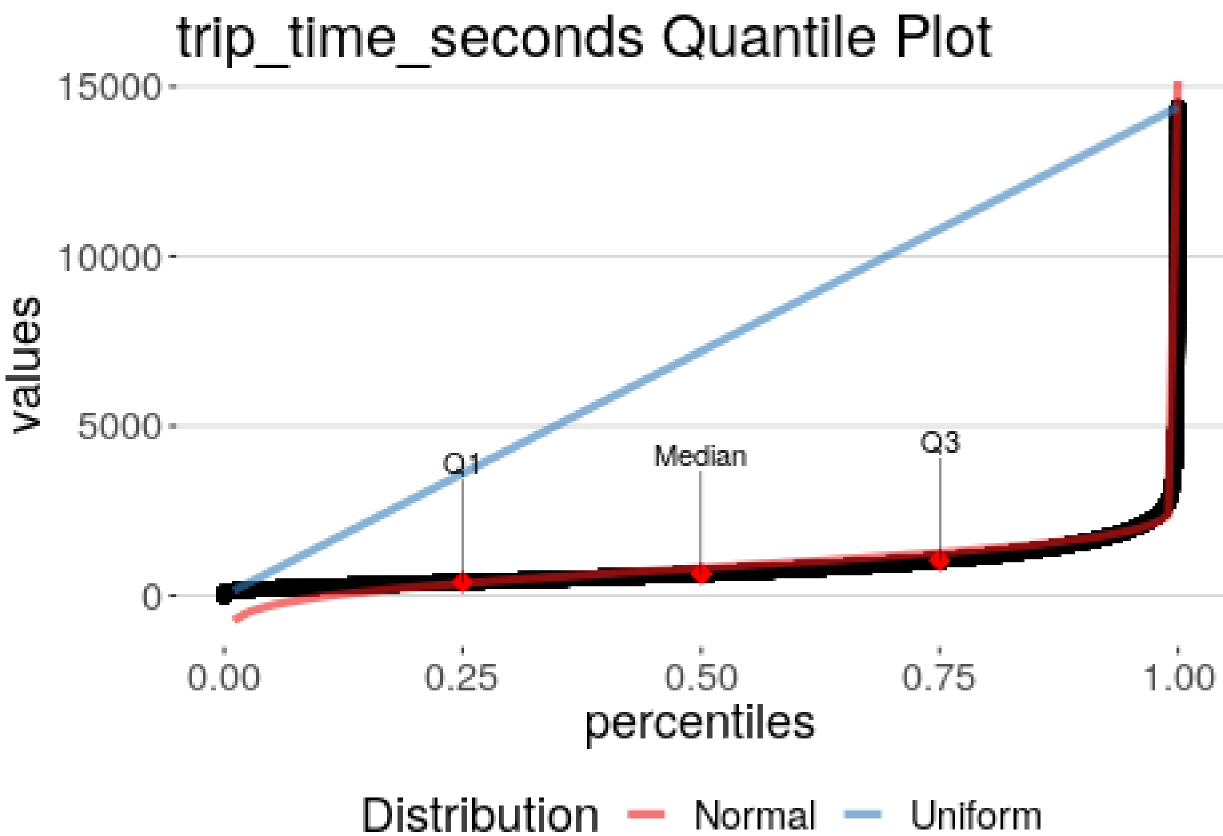
Quantile plots

```r
# Age distribution looks like a normal distribution
quantile_plot(bikes, 'age')
```



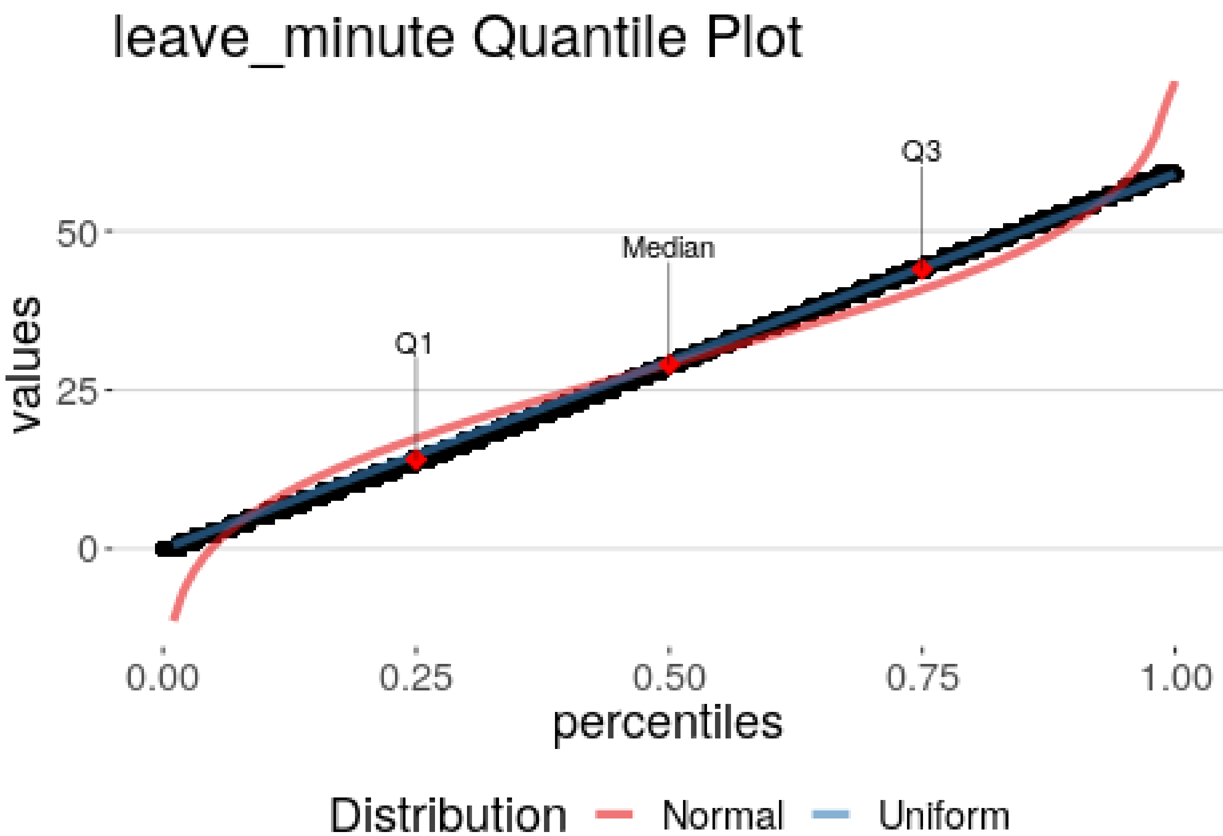
Age distribution, is a bit skewed to the right.

```
# The distribution of trip time
# has outliers to the left
# that explains the look of the quantil plot
quantile_plot(bikes, 'trip_time_seconds')
```



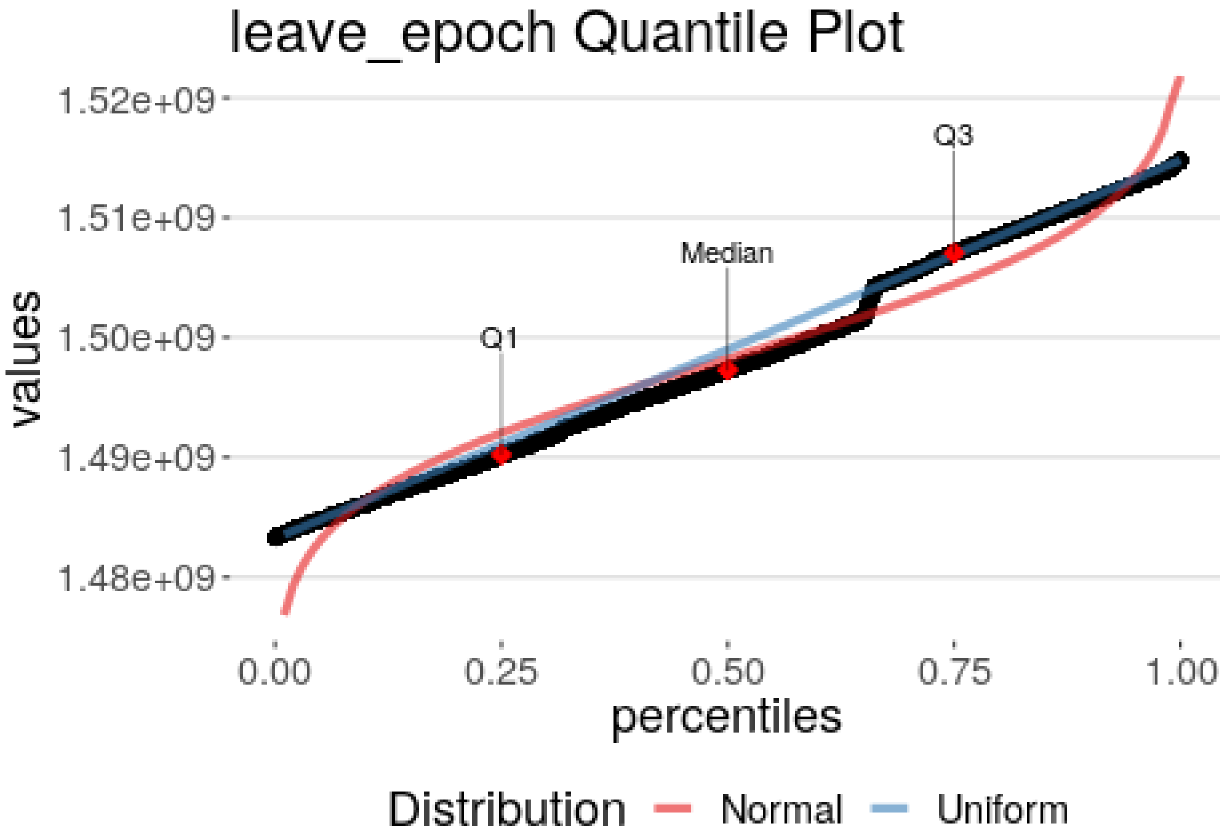
The distribution of trip time has outliers to the left that explains the look of the quantile plot.

```
quantile_plot(bikes, 'leave_minute')
```



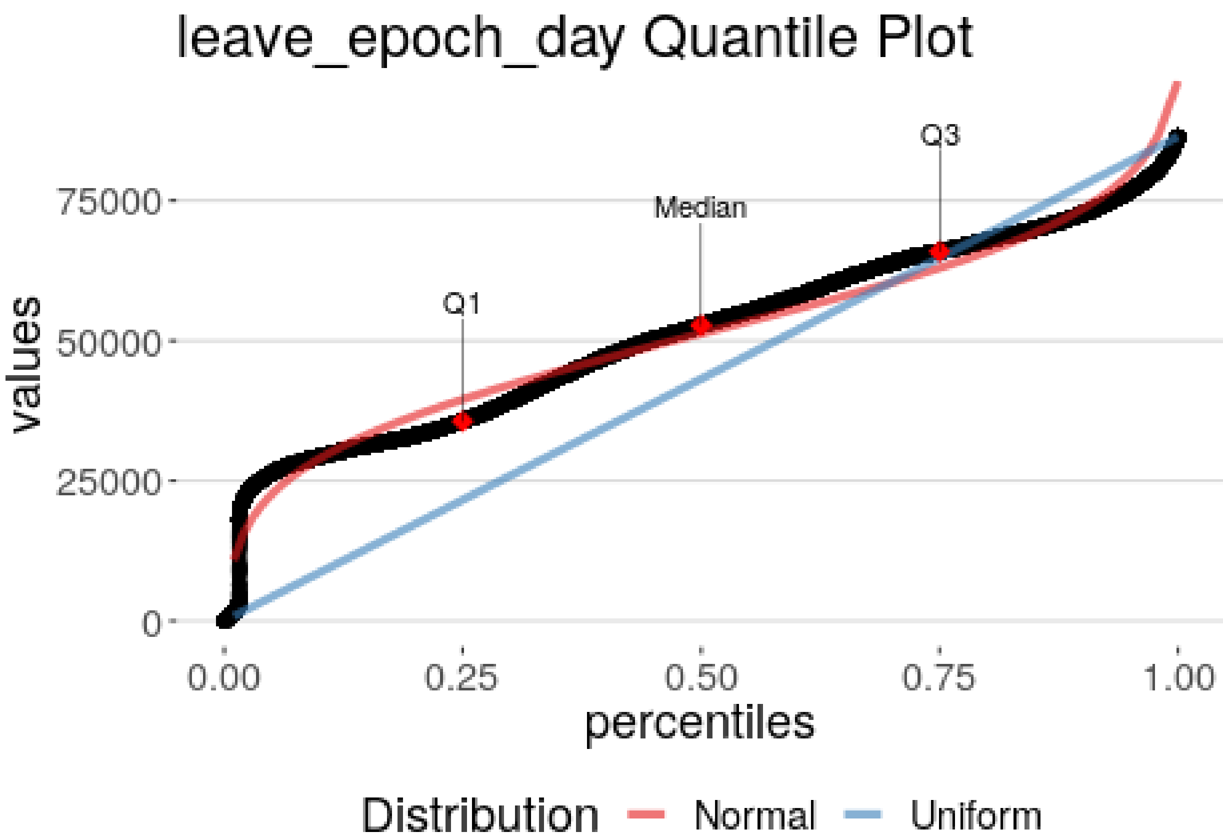
Leave minute quantile plot.

```
# The trips are progressively happening
quantile_plot(bikes, 'leave_epoch')
```



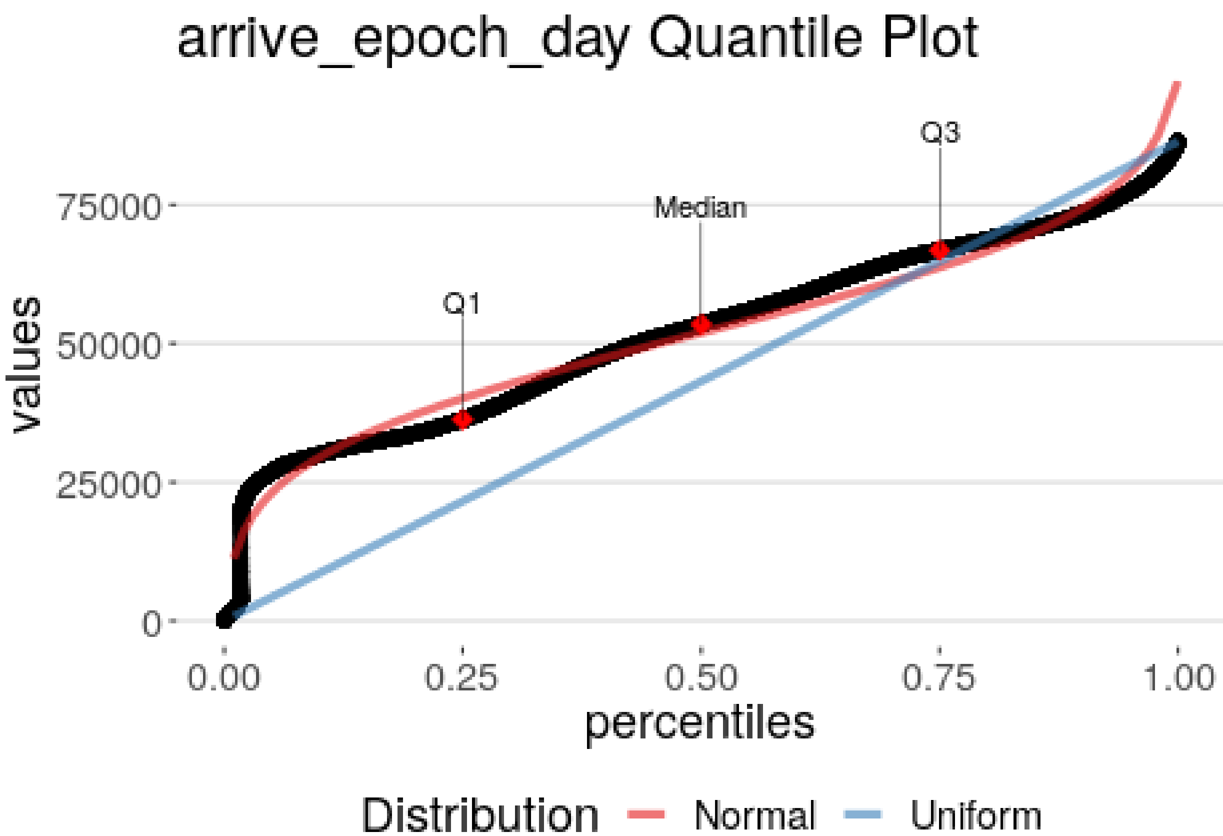
The trips are progressively happening through the year that explains the almost uniform distribution the break between the median and Q3 could be due to the few August trips.

```r
# The quantile plot is step near the 0:00 hrs,
# that is cause from 1 to 4 hours theservice is closed
quantile_plot(bikes, 'leave_epoch_day')
```



Leave second since midnigth quantile plot. The quantile plot is stepper near the 0:00 hrs, that is cause from 1 to 4 hours the service is closed.

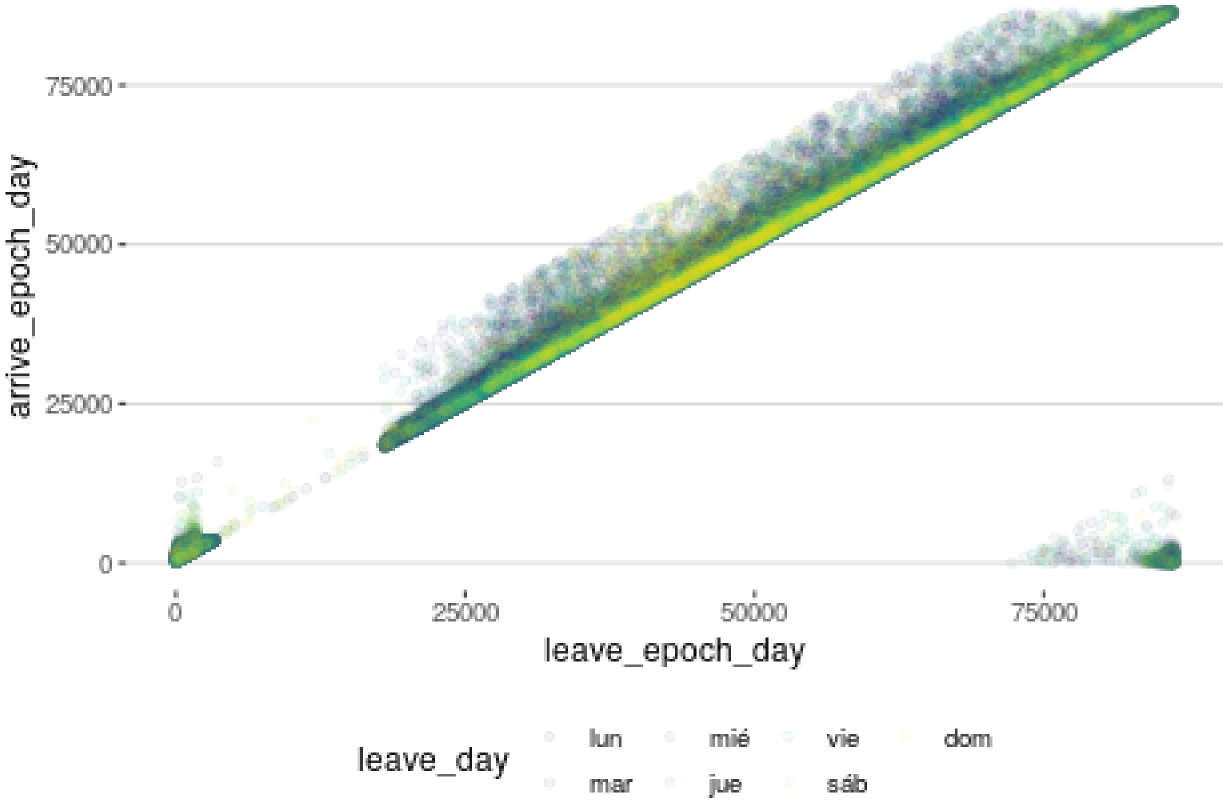
```
# Pattern similar to leave time
quantile_plot(bikes, 'arrive_epoch_day')
```



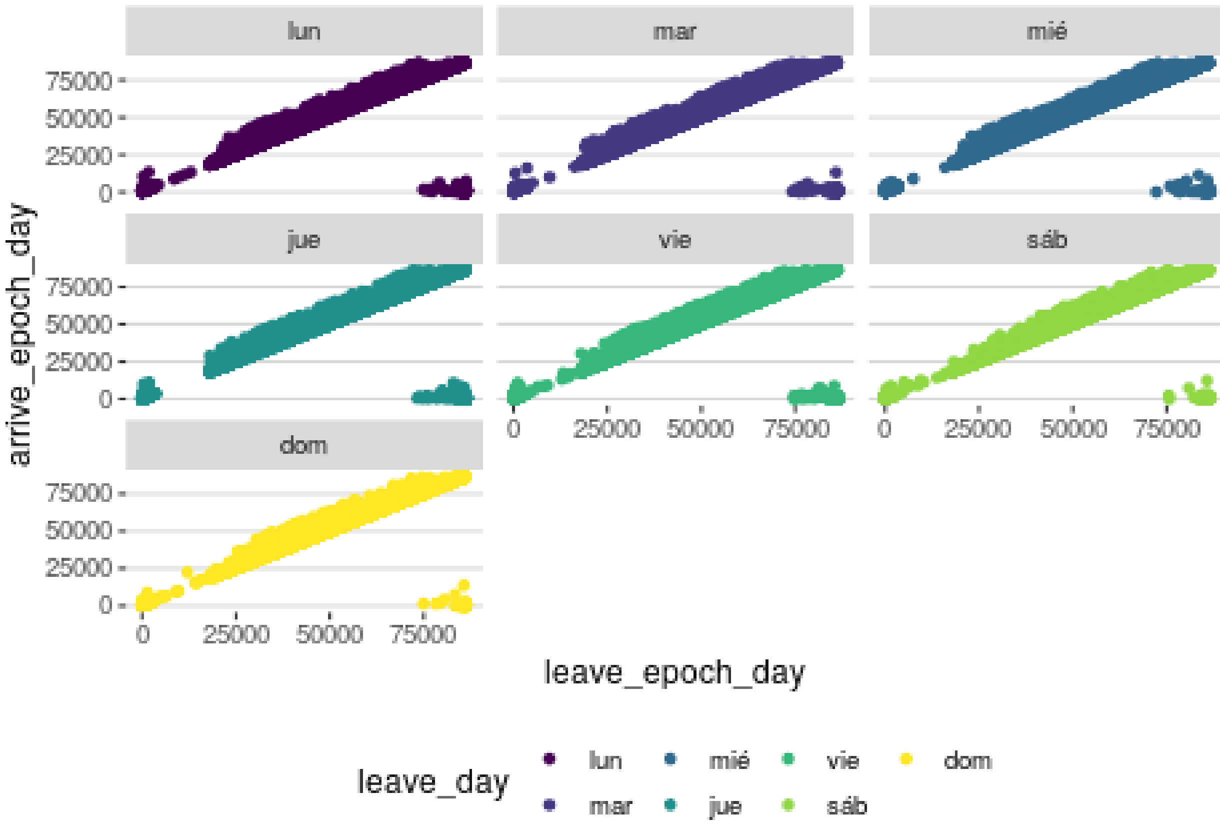
Arrive second since midnigth quantile plot. Pattern similar to leave time.

Scatter plots

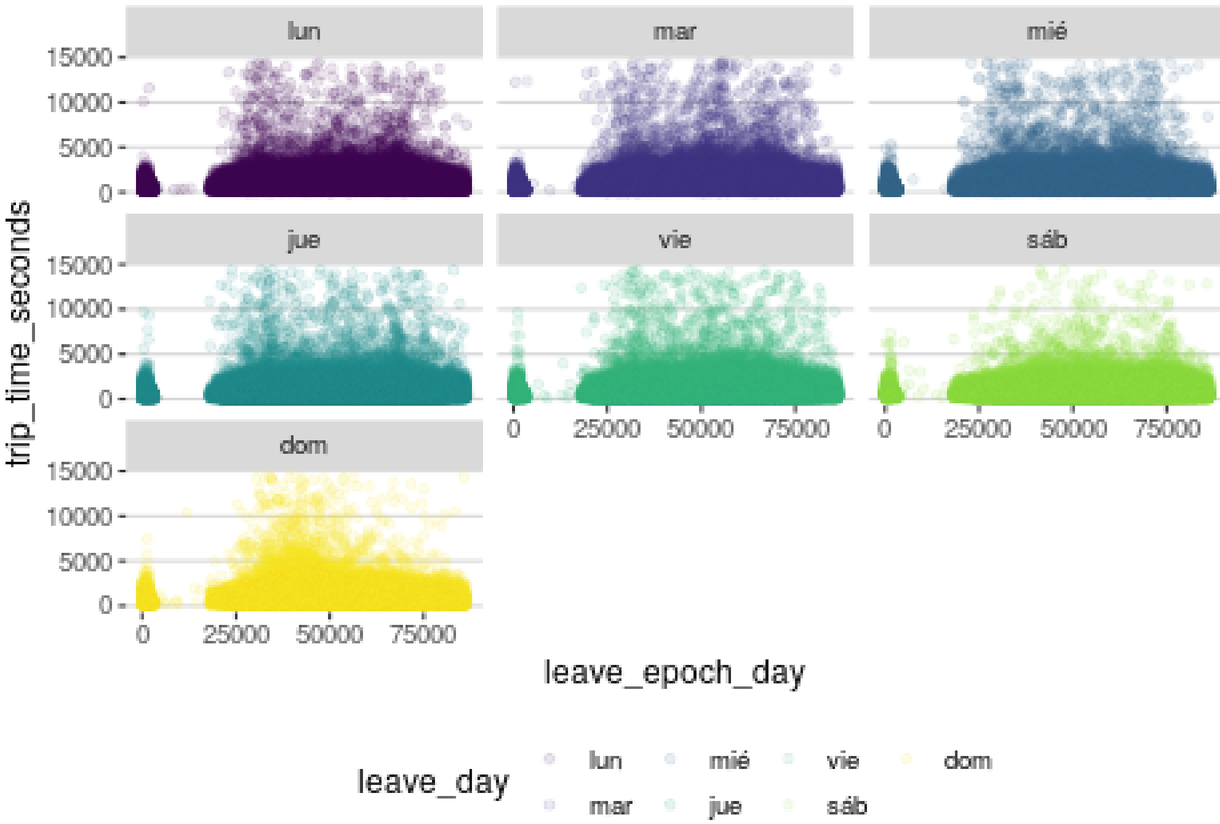
```
ggplot( bikes,
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day)) +
  geom_point(alpha = 1/20)+theme_hc()
```



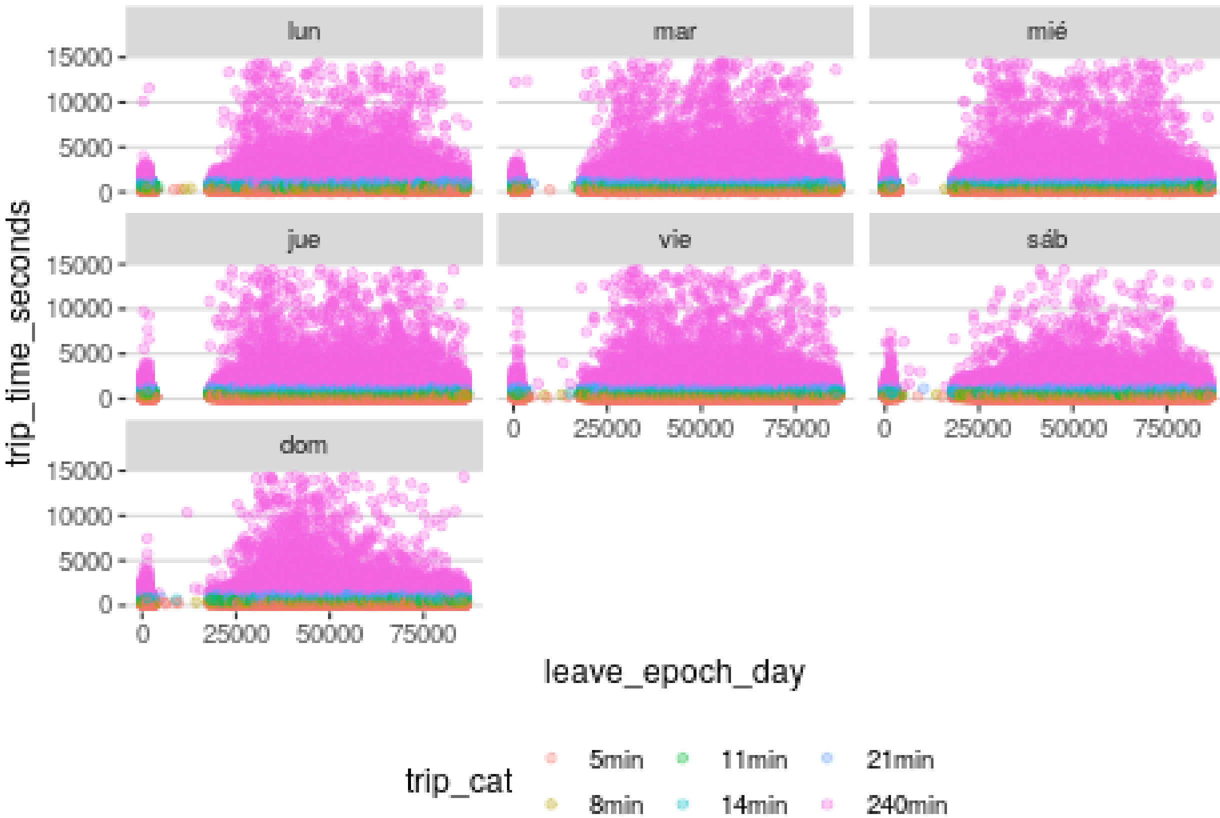

```r
ggplot( bikes,
       aes(x=leave_epoch_day, y=arrive_epoch_day, color=leave_day)) +
  geom_point()+facet_wrap(.~leave_day)+theme_hc()
```



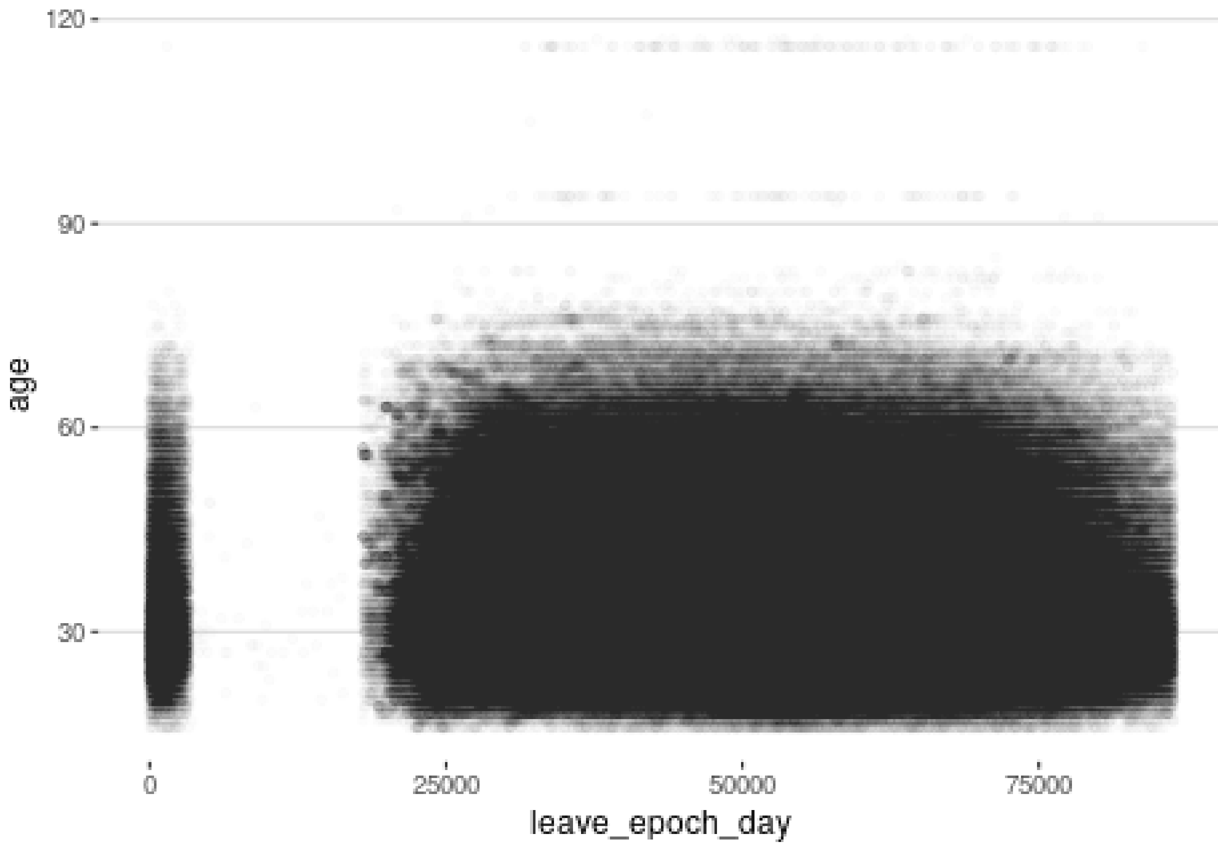

```r
ggplot( bikes,
       aes(x=leave_epoch_day, y=trip_time_seconds, color=leave_day)) +
  geom_point(alpha=1/10)+facet_wrap(.~leave_day)+theme_hc()
```



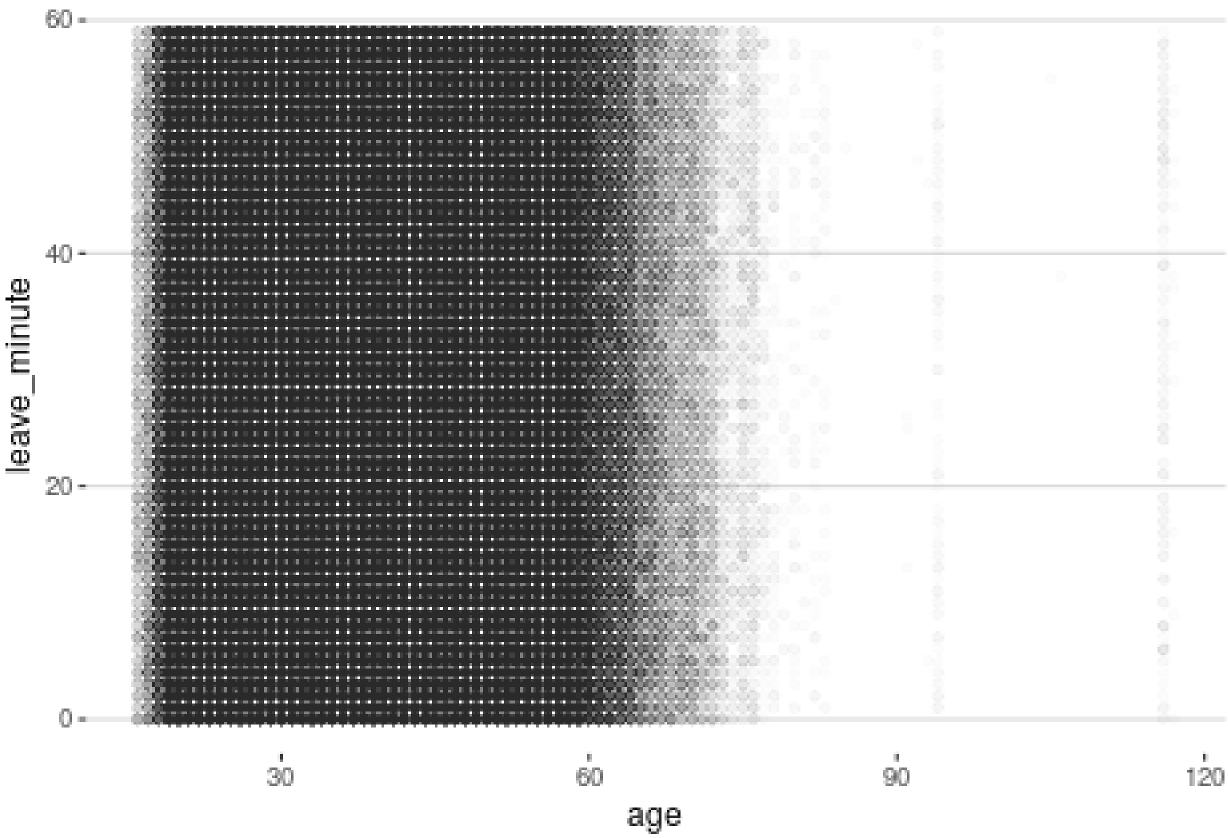

```
ggplot( bikes,
       aes(x=leave_epoch_day, y=trip_time_seconds, color=trip_cat)) +
  geom_point(alpha=1/3)+facet_wrap(.~leave_day)+theme_hc()
```




```r
ggplot( bikes,
       aes(x=leave_epoch_day, y=age) ) +
  geom_point(alpha=1/100)+theme_hc()
```




```
ggplot( bikes,
       aes(x=age, y=leave_minute) ) +
  geom_point(alpha=1/100)+theme_hc()
```

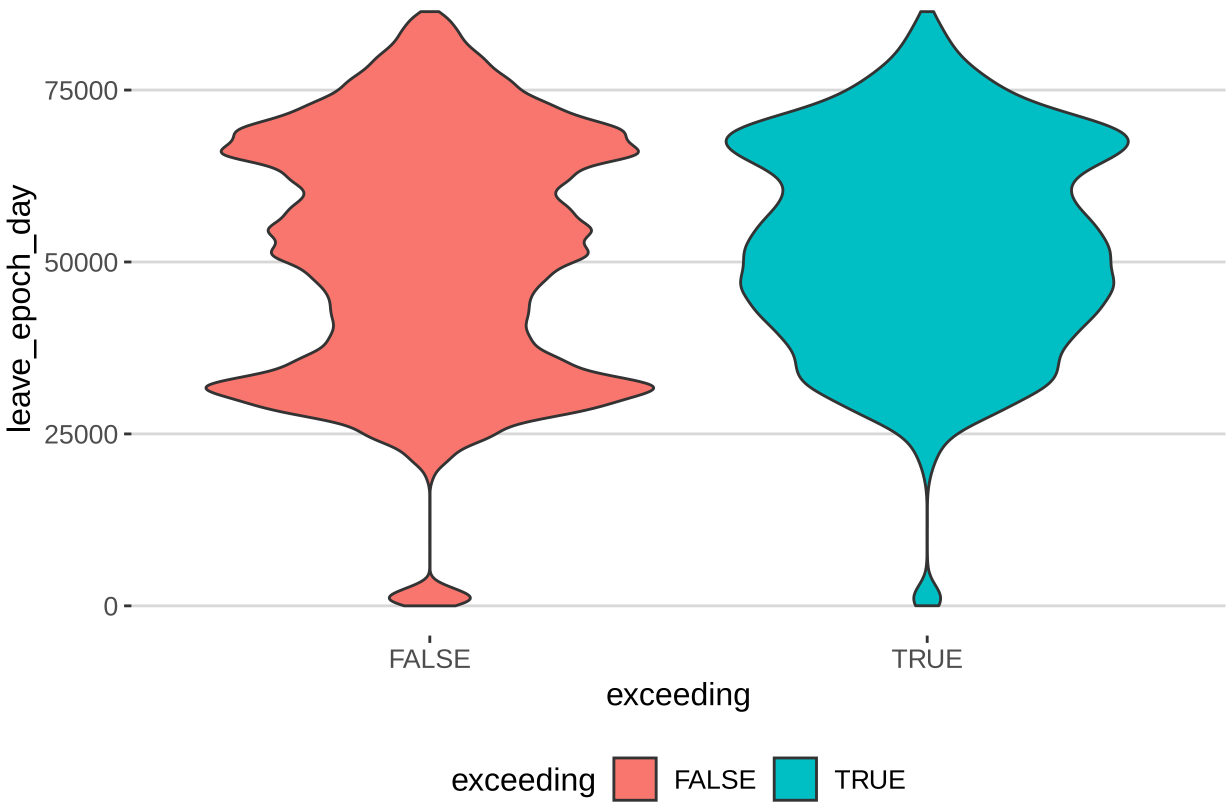


Variables to drop

From the source data I dropped the dates attributes since I've captured that information in other new columns, I've separated it over month, day, hour, minute and second. Because the original data has only few (9) attributes.

Other plots

```
# The distribution is similar between exceeding trips and not exceeding trips
ggplot( bikes, aes(x=exceeding, y=leave_epoch_day, fill = exceeding ) )+
  geom_violin()+theme_hc()
```



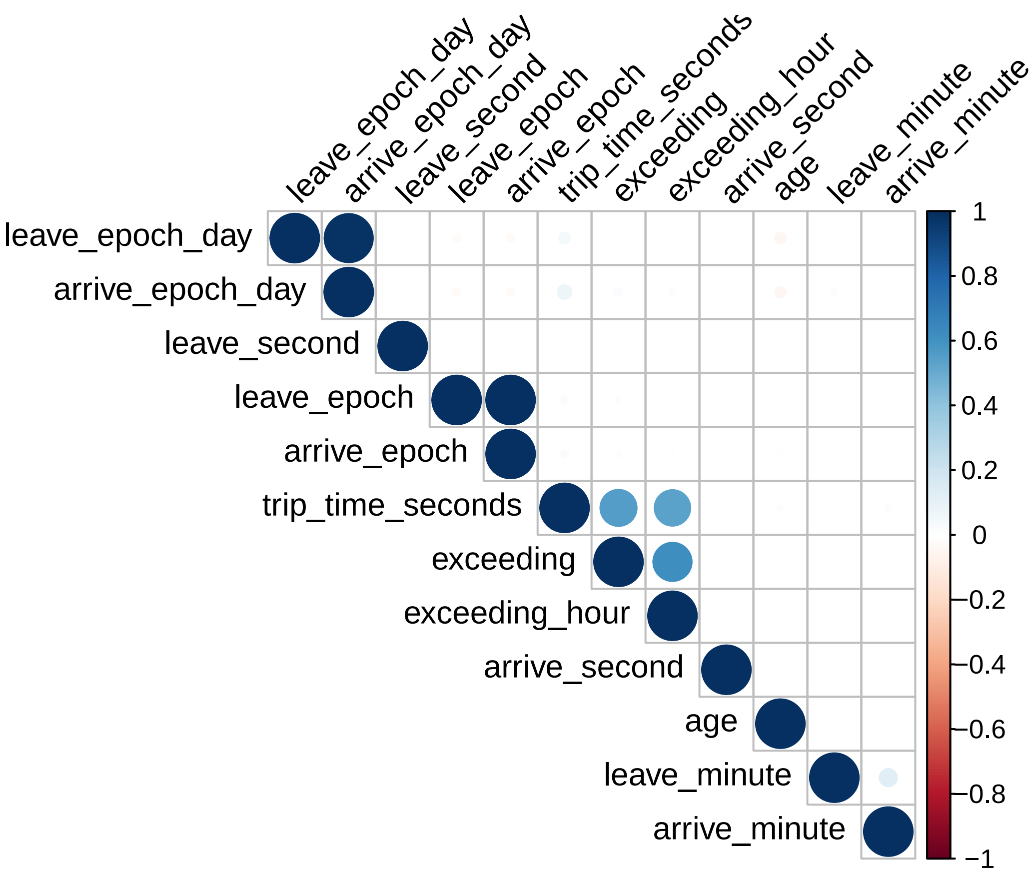
Trips started in the morning tend not to exceed the 45 min mark in comparison to other times, still the distribution between people in time or not in time looks kind similar.

```r
# Calculating the correlation on the numeric variables.
cor_bikes <- cor(bikes[ numeric_col ], method = 'pearson')
round(cor_bikes, 2)
```

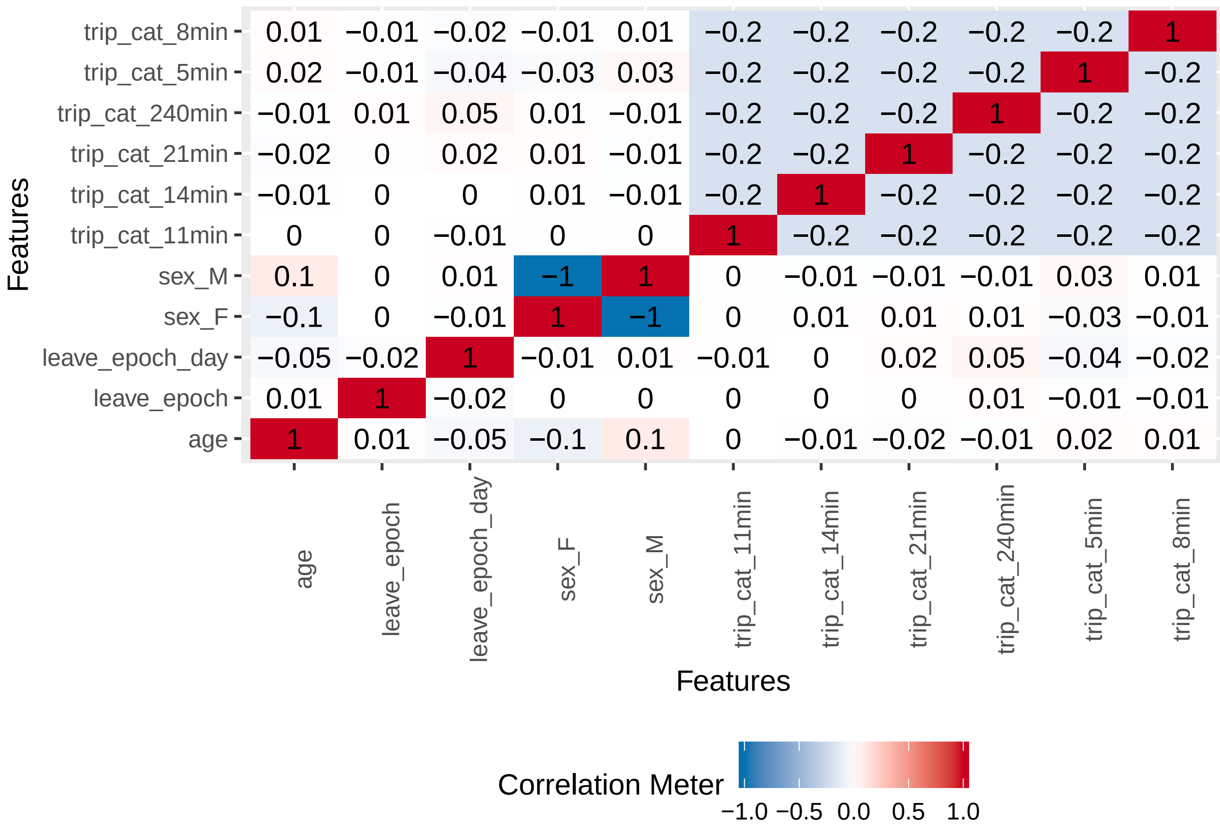
```
##                     age trip_time_seconds leave_minute leave_second
## age                1.00             -0.01         0.00            0
## trip_time_seconds -0.01              1.00        -0.01            0
## leave_minute       0.00             -0.01         1.00            0
## leave_second       0.00              0.00         0.00            1
## leave_epoch        0.01              0.02         0.00            0
## leave_epoch_day   -0.05              0.05         0.00            0
## arrive_minute      0.00              0.02         0.12            0
## arrive_second      0.00              0.00         0.00            0
## arrive_epoch       0.01              0.02         0.00            0
## arrive_epoch_day  -0.04              0.08        -0.01            0
## exceeding          0.00              0.55         0.00            0
## exceeding_hour     0.00              0.53         0.00            0
##                   leave_epoch leave_epoch_day arrive_minute arrive_second
## age                      0.01           -0.05          0.00             0
## trip_time_seconds        0.02            0.05          0.02             0
## leave_minute             0.00            0.00          0.12             0
## leave_second             0.00            0.00          0.00             0
## leave_epoch              1.00           -0.02          0.00             0
## leave_epoch_day         -0.02            1.00          0.00             0
## arrive_minute            0.00            0.00          1.00             0
## arrive_second            0.00            0.00          0.00             1
## arrive_epoch             1.00           -0.02          0.00             0
```

```
## arrive_epoch_day         -0.02            0.98         0.01              0
## exceeding                 0.01            0.01         0.00              0
## exceeding_hour            0.01            0.00         0.00              0
##                    arrive_epoch arrive_epoch_day exceeding exceeding_hour
## age                        0.01            -0.04      0.00           0.00
## trip_time_seconds          0.02             0.08      0.55           0.53
## leave_minute               0.00            -0.01      0.00           0.00
## leave_second               0.00             0.00      0.00           0.00
## leave_epoch                1.00            -0.02      0.01           0.01
## leave_epoch_day           -0.02             0.98      0.01           0.00
## arrive_minute              0.00             0.01      0.00           0.00
## arrive_second              0.00             0.00      0.00           0.00
## arrive_epoch               1.00            -0.02      0.01           0.01
## arrive_epoch_day          -0.02             1.00      0.02           0.01
## exceeding                  0.01             0.02      1.00           0.62
## exceeding_hour             0.01             0.01      0.62           1.00
```

```r
corrplot(cor_bikes, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```




```r
plot_correlation( bikes[ c('sex', 'age',
                           'leave_epoch',
                           'leave_epoch_day',
                           'trip_cat') ] )
```



The correlation found are expected.

Leave and arrive time are correlated, as trip time and exceeding times (45 min and 1 hour)

```
station_usage_start <- arrange( as.tibble( table(bikes$station_start) ), desc(n) )
station_usage_start$perct_start <- ( station_usage_start$n / sum(station_usage_start$n) )

station_usage_end <- arrange( as.tibble( table(bikes$station_end) ), desc(n) )
station_usage_end$perct_end <- ( station_usage_end$n / sum(station_usage_end$n) )

station_usage <- inner_join(station_usage_start, station_usage_end, by = 'Var1')
station_usage <- rename(station_usage, st_id = Var1 , n_start = n.x, n_end = n.y)
station_usage <- arrange(station_usage, desc(n_start))

dim(station_usage)
```

```
## [1] 458   5
```

```
length( cumsum(station_usage$perct_start) )
```

```
## [1] 458
```

```
head(station_usage)
```

```
## # A tibble: 6 x 5
##   st_id n_start perct_start n_end perct_end
##   <chr>   <int>       <dbl> <int>     <dbl>
## 1 27      21185      0.0121  20992    0.0120
## 2 271     16194      0.00925 13490    0.00770
## 3 1       14295      0.00816 15101    0.00862
## 4 18      14077      0.00804 14585    0.00833
```

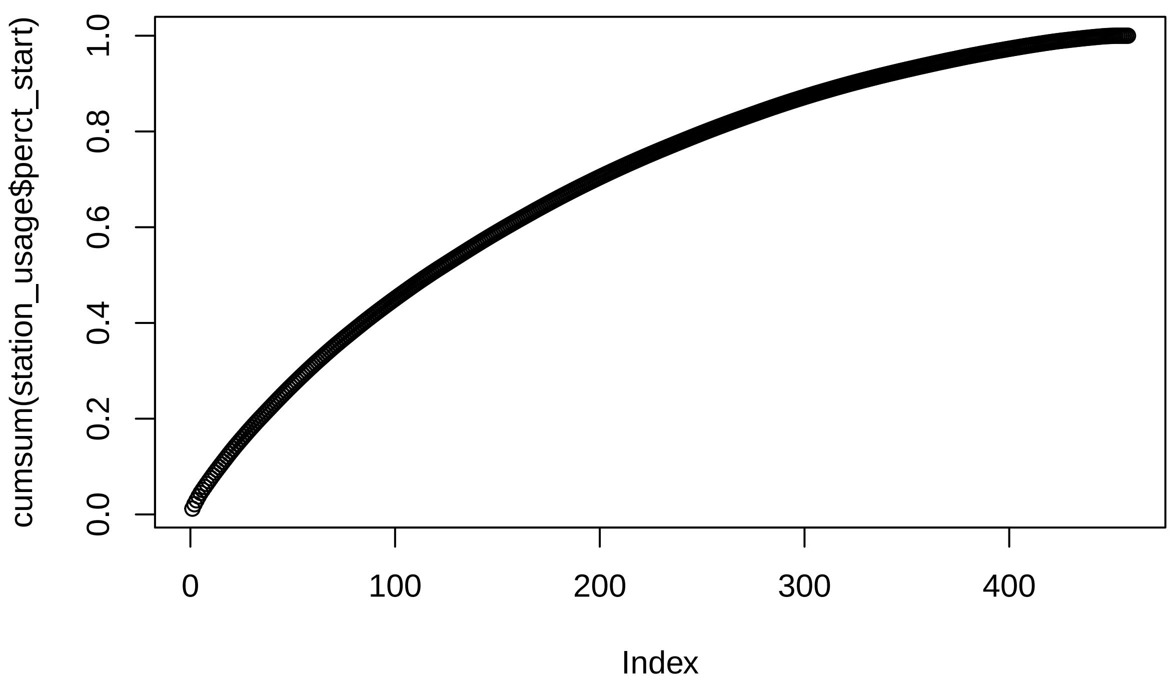
```
## 5 21       12925      0.00738 12034    0.00687
## 6 15       10990      0.00628 10327    0.00590
```

```r
knitr::kable(
station_usage[ 1:10, ],
caption = "10 most used leave stations"
)
```

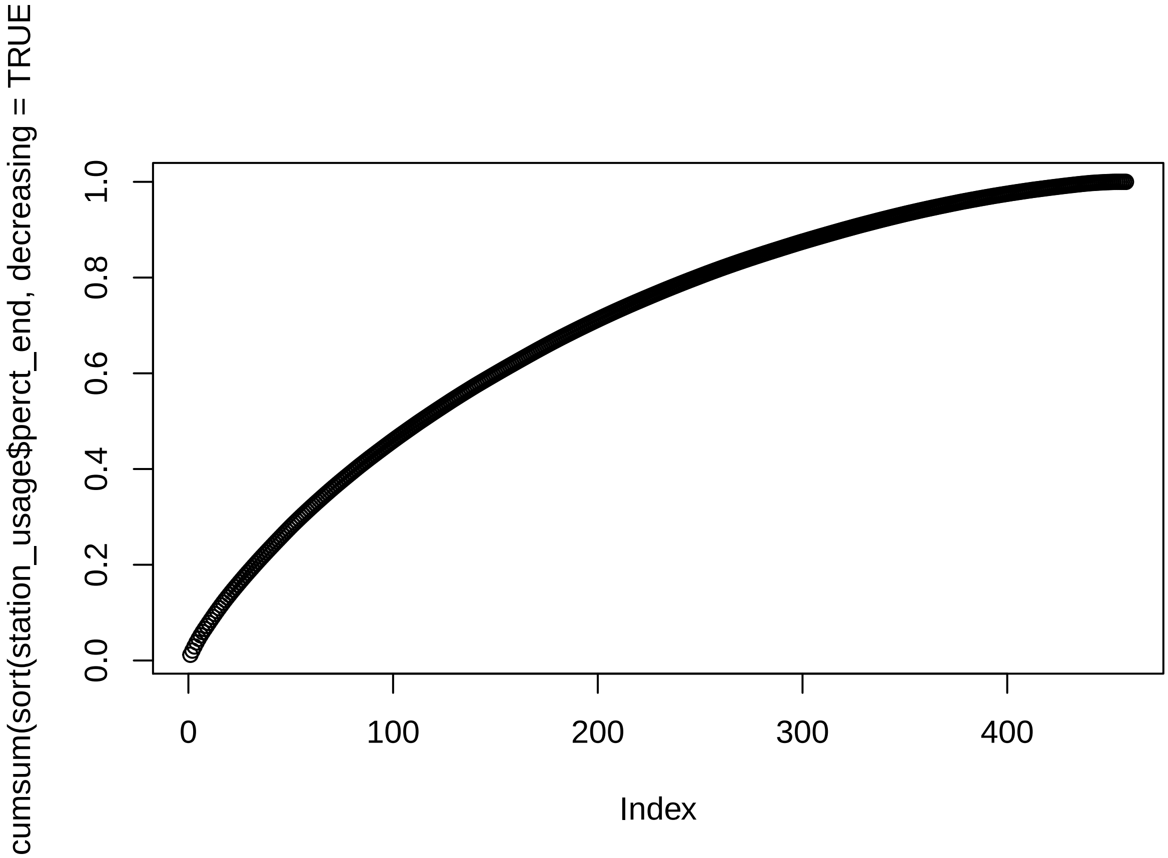
Table 2: 10 most used leave stations

st_id	n_start	perct_start	n_end	perct_end
27	21185	0.0120970	20992	0.0119868
271	16194	0.0092470	13490	0.0077030
1	14295	0.0081627	15101	0.0086229
18	14077	0.0080382	14585	0.0083283
21	12925	0.0073804	12034	0.0068716
15	10990	0.0062755	10327	0.0058969
36	10939	0.0062463	10850	0.0061955
25	10874	0.0062092	11073	0.0063229
43	10537	0.0060168	12684	0.0072428
23	10433	0.0059574	10086	0.0057593

```r
# Cumulative percents over station usage
plot(cumsum(station_usage$perct_start))
```




```r
# Cumulative percents over station usage
plot(cumsum( sort(station_usage$perct_end, decreasing = TRUE) ))
```

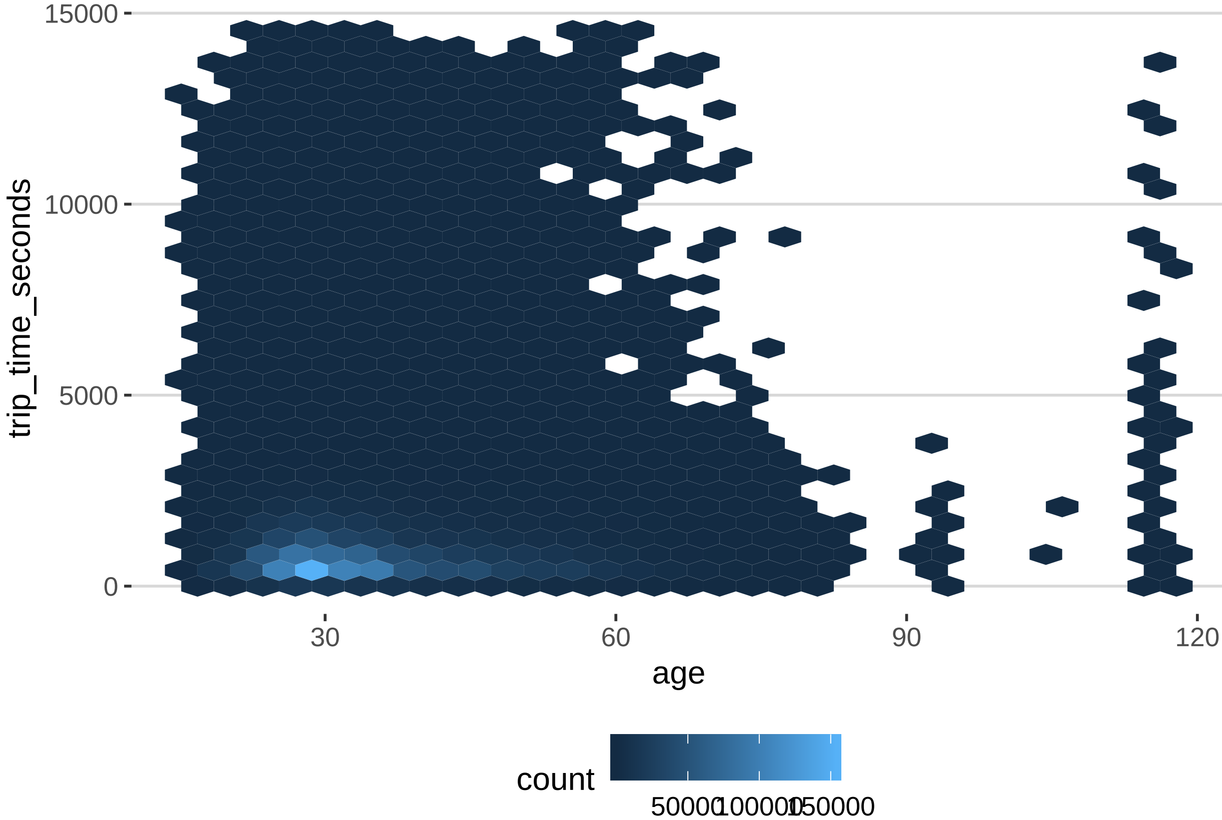


Cumulative sum of station usage.

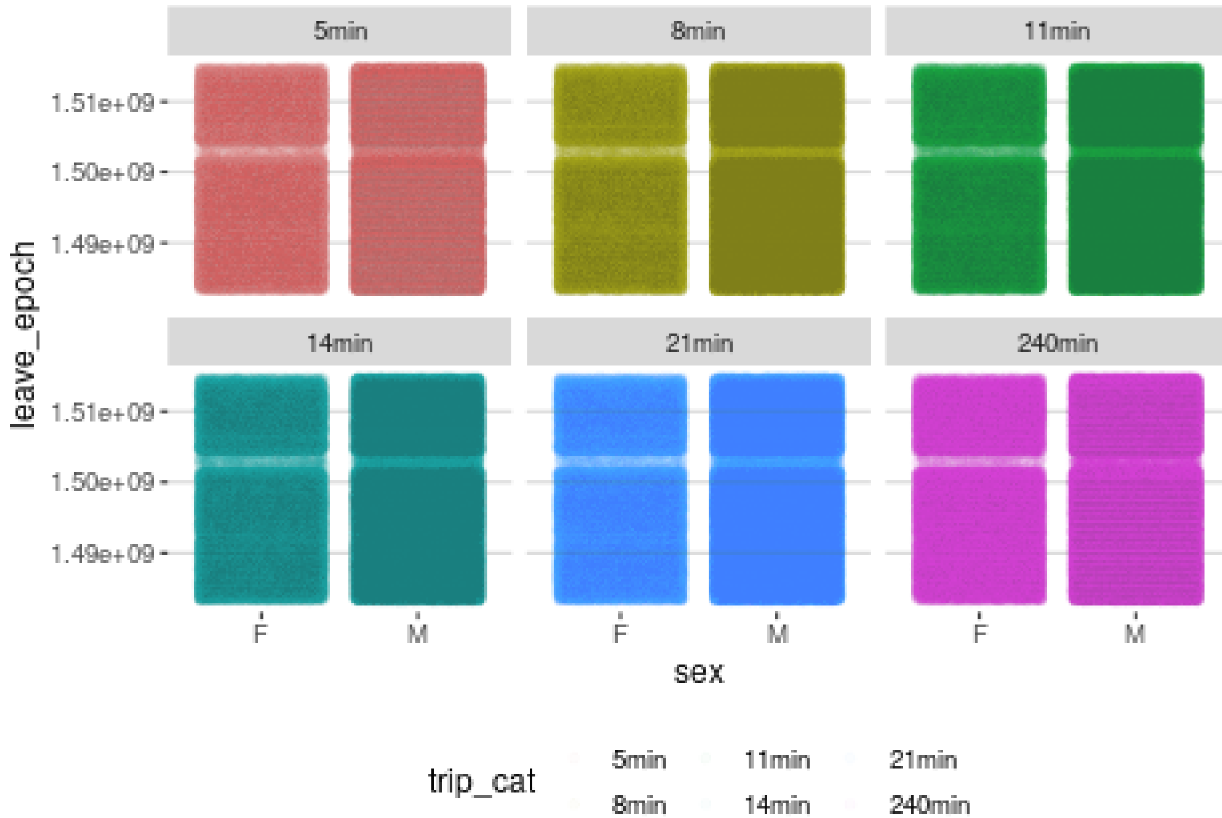
1. For leaving stations.
2. For starting stations.

Approximately the 50% percent of traffic comes from the top 100 (per use) stations

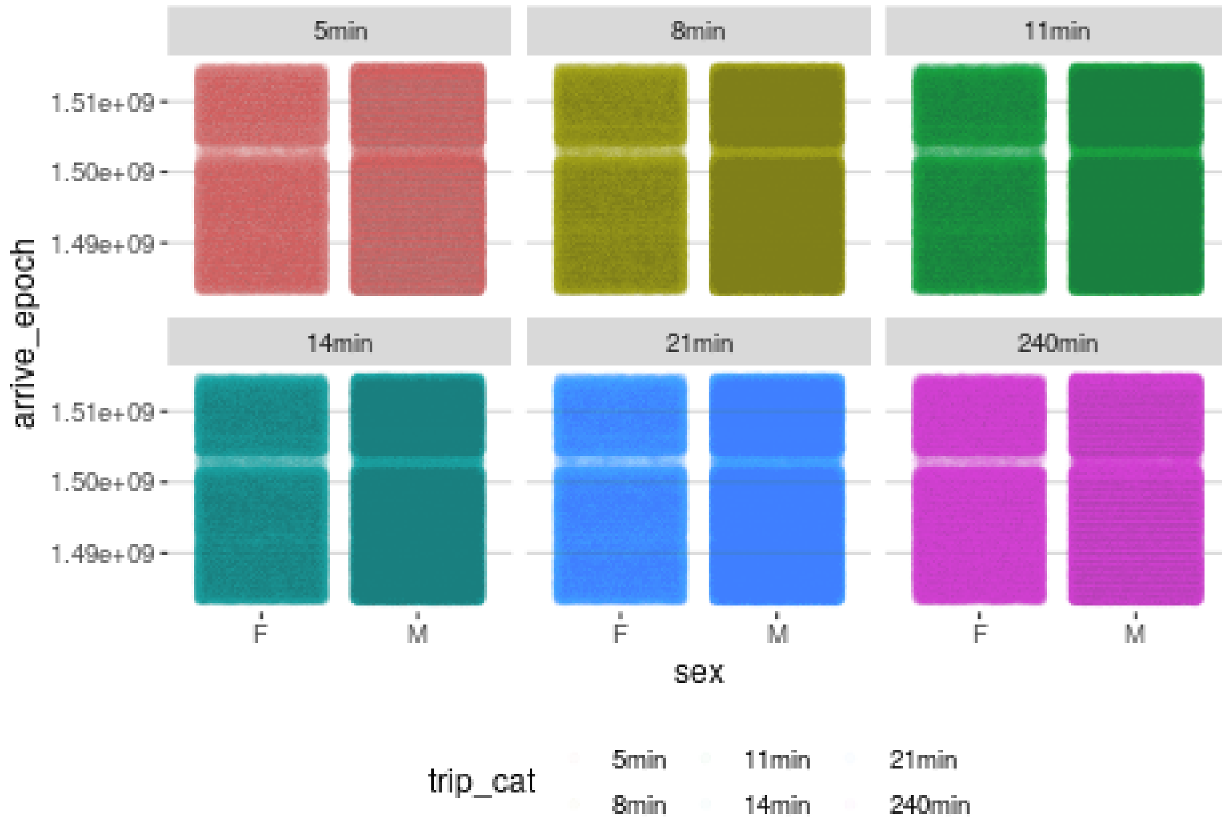
```
ggplot( bikes,
       aes(x=age, y=trip_time_seconds) ) +
  geom_hex()+theme_hc()
```



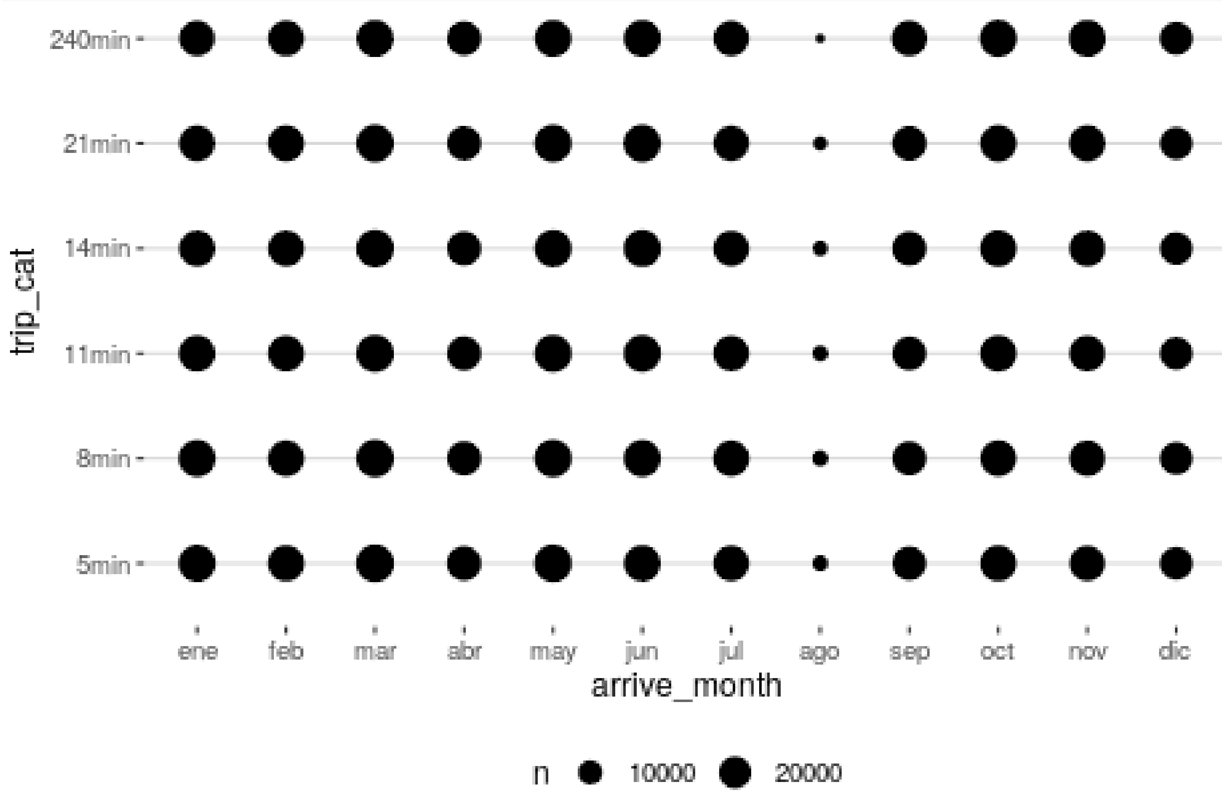

```r
ggplot( bikes,
       aes(x=sex, y=leave_epoch, color=trip_cat)) +
  geom_point(alpha=1/50, position = 'jitter')+facet_wrap(.~trip_cat)+theme_hc()
```




```
ggplot( bikes,
       aes(x=sex, y=arrive_epoch, color=trip_cat)) +
  geom_point(alpha=1/50, position = 'jitter')+facet_wrap(.~trip_cat)+theme_hc()
```




```r
ggplot( bikes,
        aes(x=arrive_month, y=trip_cat)) +
  geom_count()+theme_hc()
```



Comments over interesting patters

It doesn't make a lot of difference your age over where you're going to exceed the 45 min mark an incur on the penalization for using to much time.

The august month is strange a very few people traveled on that month and mainly on the midnigth.

The problematic times cluster around the 0:00 hours.

The outlier trip time cluster around 0:00 hours, probably they missed the service closing time and they had to wait until the next day.

Ideas of data mining about your data set

Geo tag the bike stations to calculate traveled distance.

Analyze the patterns of mobilization through the morning rush and the afternoon rush.

Find good candidate places to expand the service and put new stations.

Comment about the issues of your data and its useful transformations

The data was in csv format so it was easy to load and start working with it right away. However it contained some NAs and some data didn't make sense for example some trips give negative total time.

The problematic trips were removed also specialized libraries were used to analyze the date time attributes.