Uber pickups in NYC analysis

Nicola Vanoli

24/5/2020

Motivation and Introduction

The city of New York is without any doubt one of the most chaotic and busy city in the world: many citizens do not own a car and the public means of transport are their only way to travel. In the last decade a new ride-hailing company, called **Uber**, has established itself on the transport market and has started to offer new services such as Uber Eats (food delivery system) and other micromobility systems with electric bikes and scooters.

In this work I try to give a visualization of the uber pickup system in NYC, by analyzing the times the services were delivered, the duration of trips and the starting and ending points.

The goal of this report is trying to track the habits of the newyorkers: what is the time of day when uber services are most in demand? What about the month? How long does in average a trip last?

With the help of our data, we will be able to rise hypothesis that can answer these and other questions.

Data preparation

explain download system

Once the dataset is downloaded, we can start prepare it for its analysis. In this work we only focus on the data collected in the year 2014 by Uber: since the dataset we want to use is split into different files, firstly unify them in a single dataset.

```r
#setwd("C:/Users/39339/Desktop/Analisi Dati/project/Uber-dataset")
apr <- read.csv("uber-raw-data-apr14.csv")
may <- read.csv("uber-raw-data-may14.csv")
jun <- read.csv("uber-raw-data-jun14.csv")
jul <- read.csv("uber-raw-data-jul14.csv")
aug <- read.csv("uber-raw-data-aug14.csv")
sep <- read.csv("uber-raw-data-sep14.csv")

# single dataset creation
data14 <- rbind(apr,may,jun,jul,aug,sep)

head(data14)
```

```
##          Date.Time     Lat      Lon   Base
## 1 4/1/2014 0:11:00 40.7690 -73.9549 B02512
## 2 4/1/2014 0:17:00 40.7267 -74.0345 B02512
## 3 4/1/2014 0:21:00 40.7316 -73.9873 B02512
## 4 4/1/2014 0:28:00 40.7588 -73.9776 B02512
## 5 4/1/2014 0:33:00 40.7594 -73.9722 B02512
## 6 4/1/2014 0:33:00 40.7383 -74.0403 B02512
```

```
# number of rows
nrow(data14)
```

```
## [1] 4534327
```

As we can see from the above table the dataset is charactherized by nrow(data14) observations, each one consisting of 4 parameters: Date&Time, Latitude, Longitude and base location.

The first variable contains a lot of informations (day, month and hour of the trip): let us split them into new variables

```
#loading packages we use in this project

# we use the package "lubridate" to separate day/month/year
library(lubridate)

#load library to summarize
library(dplyr)

#loading libraries for better plot
library(scales)
library(ggplot2)
library(gridExtra)

#loading spatial libraries
library(maps)
library(rgdal)
library(sp)
library(leaflet)
```

```
data14$Date.Time <- as.POSIXct(data14$Date.Time, format = "%m/%d/%Y %H:%M:%S")
data14$Time <- format(as.POSIXct(data14$Date.Time, format = "%m/%d/%Y %H:%M:%S"), format="%H:%M:%S")

# we now divide the first variable into multiple ones
data14$Date.Time <- ymd_hms(data14$Date.Time)
data14$day <- factor(day(data14$Date.Time))
data14$month <- factor(month(data14$Date.Time,label = TRUE))
data14$dayofweek <- factor(wday(data14$Date.Time, label = TRUE))

data14$hour <- factor(hour(hms(data14$Time)))

data14 <- select(data14,- Base)
uber_raw_2014 = data14
head(data14)
```

```
##             Date.Time     Lat      Lon     Time day month dayofweek hour
## 1 2014-04-01 00:11:00 40.7690 -73.9549 00:11:00   1   apr       mar    0
## 2 2014-04-01 00:17:00 40.7267 -74.0345 00:17:00   1   apr       mar    0
## 3 2014-04-01 00:21:00 40.7316 -73.9873 00:21:00   1   apr       mar    0
## 4 2014-04-01 00:28:00 40.7588 -73.9776 00:28:00   1   apr       mar    0
## 5 2014-04-01 00:33:00 40.7594 -73.9722 00:33:00   1   apr       mar    0
## 6 2014-04-01 00:33:00 40.7383 -74.0403 00:33:00   1   apr       mar    0
```

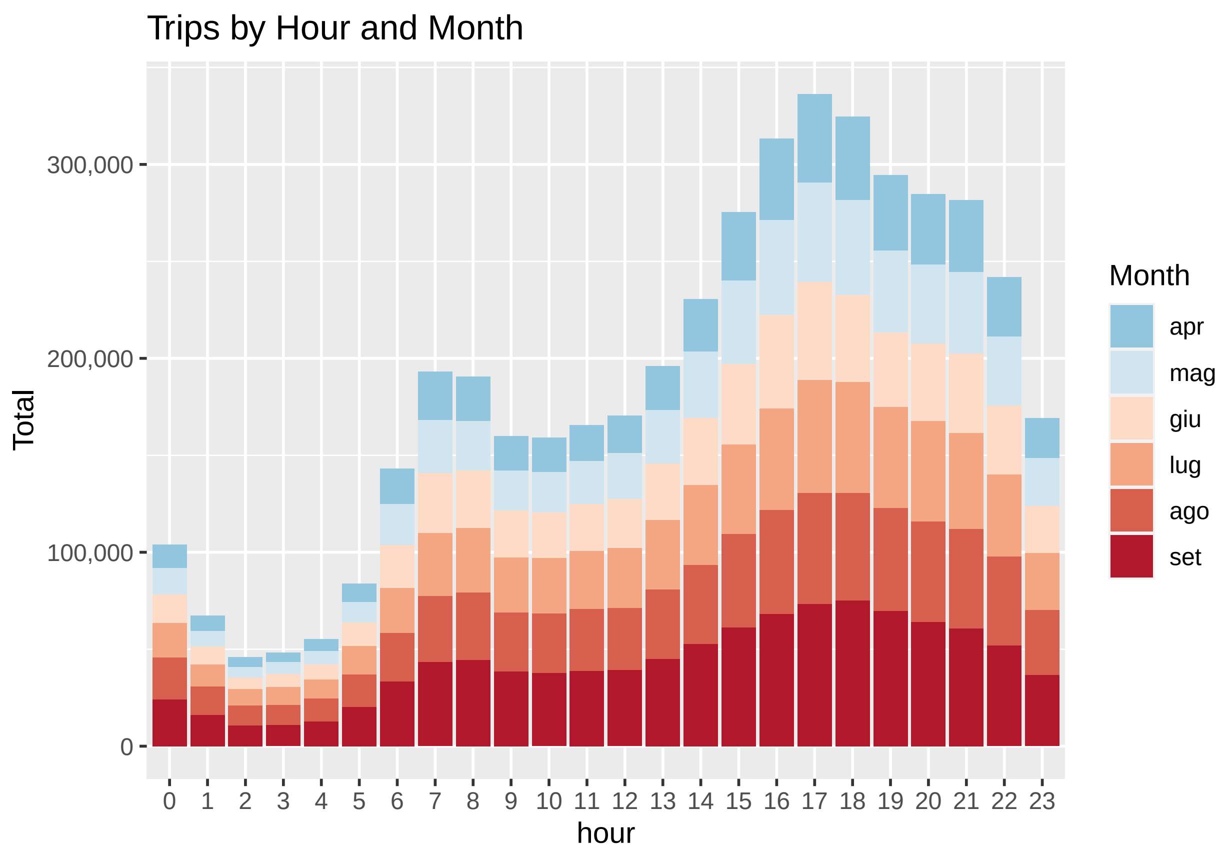
Data Visualization

We are now ready to visualize the information contained in our Dataset. Let's explore the trips distribution by hours in a day.

Trip Distribution by Hours and Month

In this paragraph we are interested in showing how the number of uber trips change throughout the day, and how it changes in relation to the month.

```
month_hour = dplyr::summarize(group_by(data14,month, hour),Total = n())

ggplot(month_hour, aes(hour, Total, fill = month)) +
scale_fill_manual(values = c("#92C5DE", "#D1E5F0","#FDDBC7" ,"#F4A582", "#D6604D", "#B2182B"),
                  name = "Month") + geom_bar( stat = "identity") + ggtitle("Trips by Hour and Month") +
```



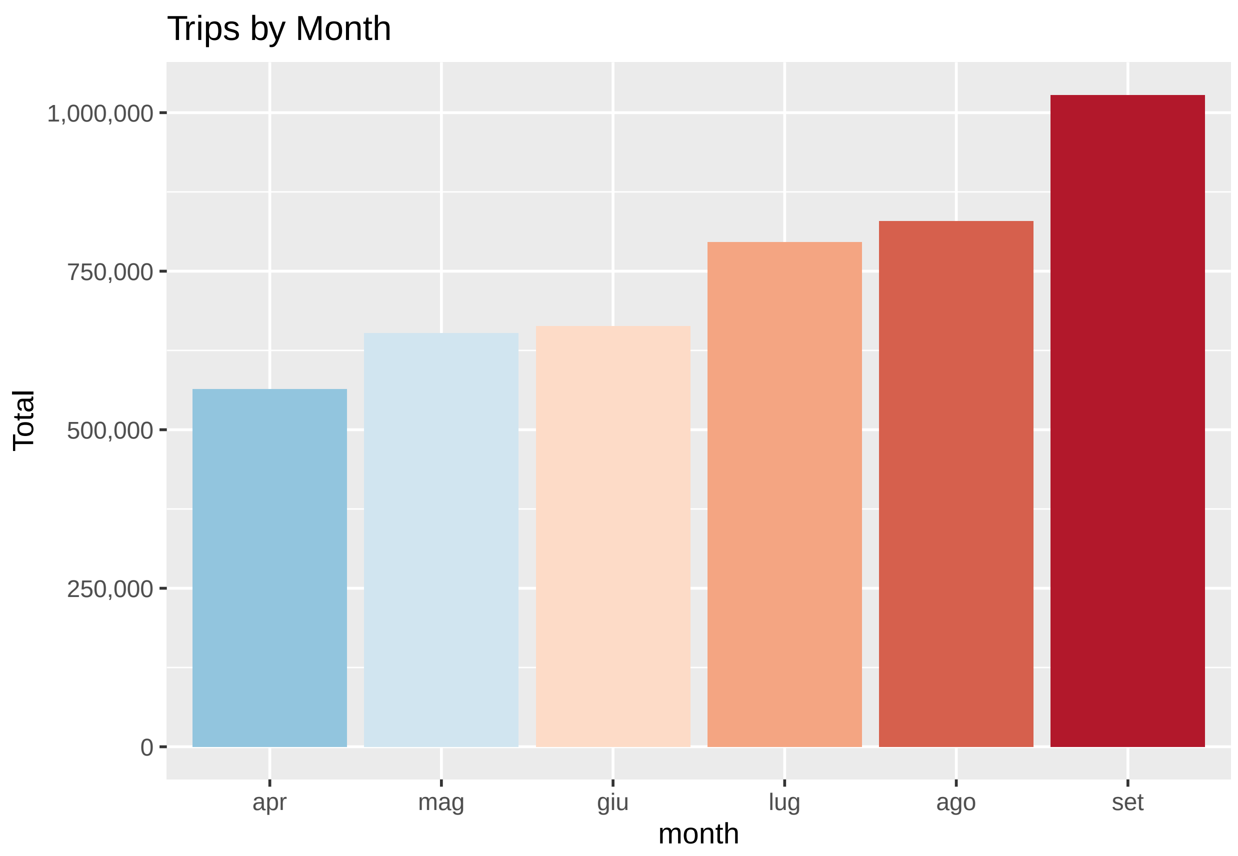
As we can see from the above graph, the most intense hours for uber drivers are sround 17-18 PM, and it looks like this trend is mantained during all six observations' months.

we may assume that this peak is related to the fact that people usually finish working around that time, and since many people in NYC do not own a car, they have to relief on public transoprt systems.

We will later investigate this hypothesis by plotting an heat map.

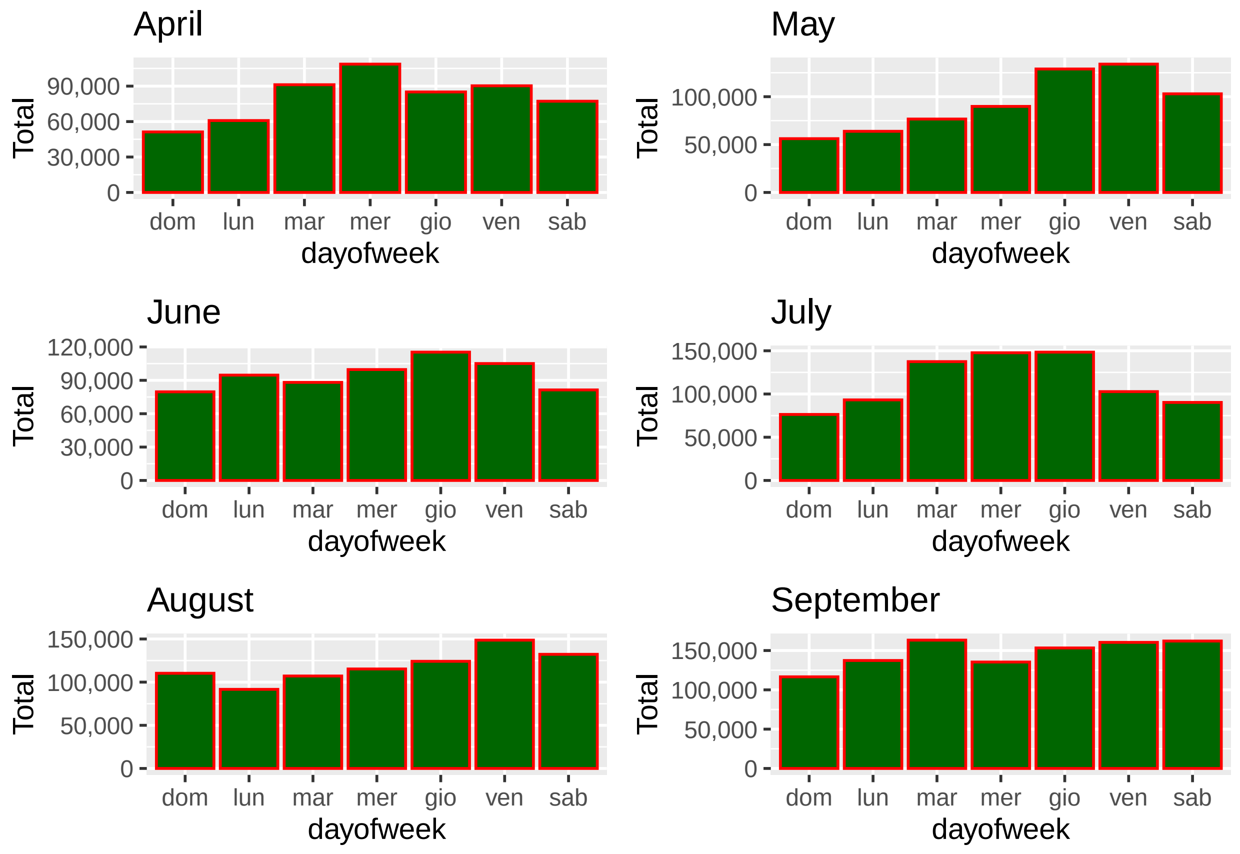
We are now interested in see what is the most productive month in terms of Uber pickups.

```
month_group <-dplyr::summarize(group_by(data14,month),Total = n())
ggplot( month_group, aes(month, Total, fill = month)) +
geom_bar( stat = "identity") +
ggtitle("Trips by Month") +
theme(legend.position = "none") +
scale_y_continuous(labels = comma) + scale_fill_manual(values = c("#92C5DE", "#D1E5F0","#FDDBC7"
```



Interestingly, the months of august and september are the most profitable for uber drivers, with almost a x2 number of trips compared too april. This could be due to the fact that in these periods there is the highest number of tourists. To support this hypothesis, let us track the Uber pickups by day of the week and month

```
day_group = dplyr::summarize(group_by(data14,month, dayofweek),Total = n())

plt1 = ggplot(filter(day_group, month == levels(data14$month)[1]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("April") +
scale_y_continuous(labels = comma)

plt2 = ggplot(filter(day_group, month == levels(data14$month)[2]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("May") +
scale_y_continuous(labels = comma)

plt3 = ggplot(filter(day_group, month == levels(data14$month)[3]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("June") +
scale_y_continuous(labels = comma)

plt4 = ggplot(filter(day_group, month == levels(data14$month)[4]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("July") +
scale_y_continuous(labels = comma)

plt5 = ggplot(filter(day_group, month == levels(data14$month)[5]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("August") +
scale_y_continuous(labels = comma)

plt6 = ggplot(filter(day_group, month == levels(data14$month)[6]), aes( dayofweek, Total)) +
  geom_bar( stat = "identity",fill = "#006600", col = "red") + ggtitle("September") +
scale_y_continuous(labels = comma)

grid.arrange(plt1, plt2, plt3, plt4, plt5, plt6, ncol=2)
```

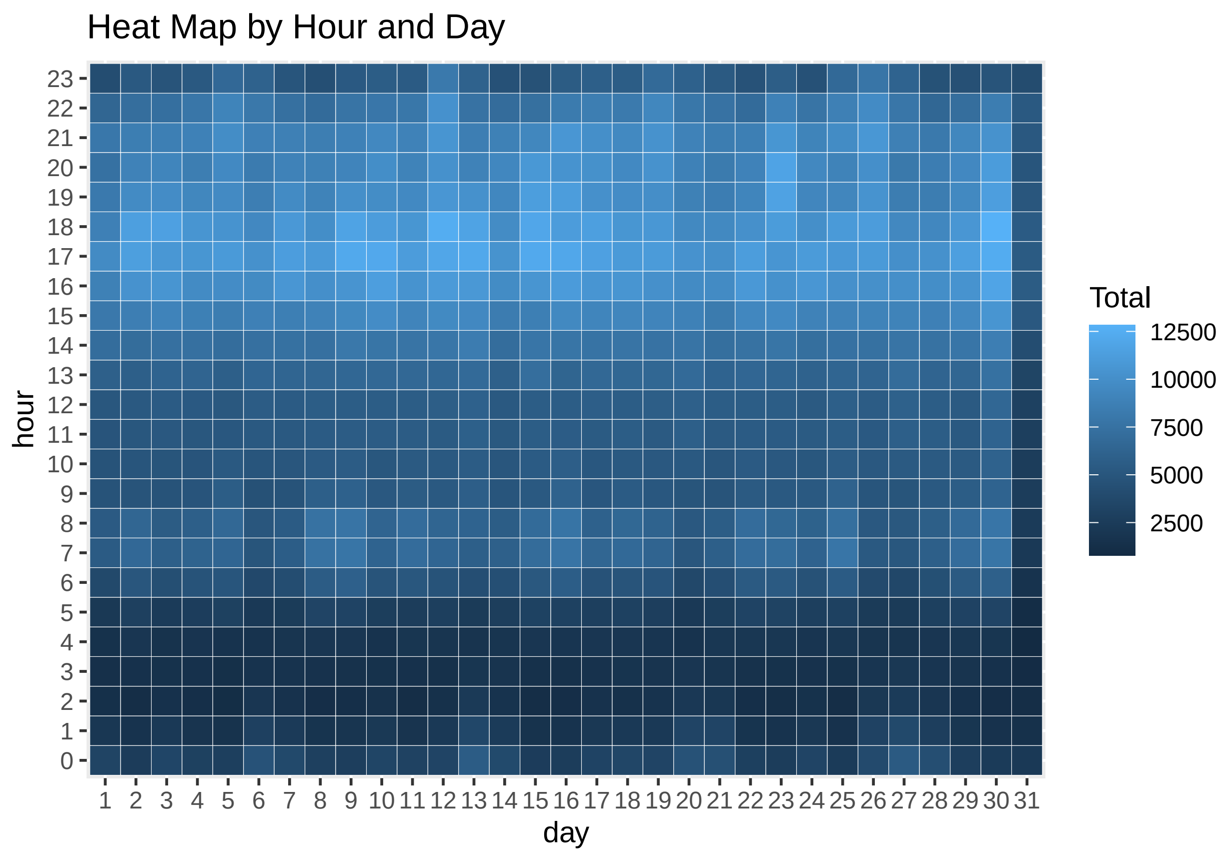


The tourist hypthesis seems to be partially true: In fact, especially in september, the "Saturday" column is higher than it is during the others months.

Heat Map of Trips by Hour

We previously discussed how the "work hypothesis" could explain the trips' peak at 17-18 PM. Let us draw a heat map by hour and day

```r
day_hour = dplyr::summarize(group_by(data14,day, hour),Total = n())

ggplot(day_hour, aes(day, hour, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Hour and Day")
```



Two brighter horizontal sripes are distinguishable: the first one is around 17-18 PM and remarks what we discussed above, the other one is around 7-8 AM, the typical hours people go to work. We can now assume that our hypothesis is valid and might be the right explanation for this phenomenon.

Events in NYC with Clustering

In this chapter we try to spot private and public events (i.e. parties) that occured during the observations period in NYC.

In order to do that, we perform density clustering on a specific subset of the original dataset. Before coding, we make a couple of assumptions that will significantly reduce the size of the dataset: 1.The party Weekend extends from Friday night to Sunday 2. People leave party places between 12am and 3 am

Splitting data in boroughs

Before clustering, we associate to each data a specific borough: in NYC we count 5 boroughs that are Manhattan, Bronx, Queens, Brooklyn and Staten Island. That is because we want to cluster indipendently inside each borough instead of clustering on the whole NYC area at once.

```
head(uber_raw_2014)
```

```
##             Date.Time     Lat      Lon     Time day month dayofweek hour
## 1 2014-04-01 00:11:00 40.7690 -73.9549 00:11:00   1   apr       mar    0
## 2 2014-04-01 00:21:00 40.7316 -73.9873 00:21:00   1   apr       mar    0
## 3 2014-04-01 00:28:00 40.7588 -73.9776 00:28:00   1   apr       mar    0
## 4 2014-04-01 00:33:00 40.7594 -73.9722 00:33:00   1   apr       mar    0
## 5 2014-04-01 00:39:00 40.7223 -73.9887 00:39:00   1   apr       mar    0
## 6 2014-04-01 00:45:00 40.7620 -73.9790 00:45:00   1   apr       mar    0
##     Borough
## 1 Manhattan
## 2 Manhattan
## 3 Manhattan
## 4 Manhattan
## 5 Manhattan
## 6 Manhattan
```

```
nrow(uber_raw_2014)
```

```
## [1] 4414138
```

As you can notice, the number of data we are now working with is lower than before: this is due to the fact that all observations outside the interested boroughs are not considered.

Converting data in UTM coordinates

We now convert the coordinate system in UTM coordinates. This trick will allow us to perform better density clustering.

```
uber_utm <- uber_raw_2014

coordinates(uber_utm) <- c("Lon", "Lat")
proj4string(uber_utm) <- CRS("+proj=longlat +datum=WGS84")

res <- spTransform(uber_utm, CRS("+proj=utm +zone=18 ellps=WGS84"))
```

Density cluster

When trying to cluster geo data, the first idea may be to use Hierarchical Clustering. To do this you have to make a distance matrix first. However our dataset contains ~4 milions rows, which implies that we would have to built a 4M x 4M matrix (yes it is symmetric, but still it would contain too many elements!).

A smarter idea is to use use the UTM coordinates (expressed in meters) and cluster data in 50x50 meters block. Basically we divide each bourough in blocks of size 50 m x 50 m and count the number of pickups occured at the interested time in each of the blocks.

By doing that we cluster the density of observations inside of it (it is a good approximation since the average size of a New York block is 80 m × 274 m).

```
head(uber_raw_2014)
```

```
##              Date.Time     Lat      Lon     Time day month dayofweek hour
## 1 2014-04-01 00:11:00 40.7690 -73.9549 00:11:00   1   apr       mar    0
## 2 2014-04-01 00:21:00 40.7316 -73.9873 00:21:00   1   apr       mar    0
## 3 2014-04-01 00:28:00 40.7588 -73.9776 00:28:00   1   apr       mar    0
## 4 2014-04-01 00:33:00 40.7594 -73.9722 00:33:00   1   apr       mar    0
## 5 2014-04-01 00:39:00 40.7223 -73.9887 00:39:00   1   apr       mar    0
## 6 2014-04-01 00:45:00 40.7620 -73.9790 00:45:00   1   apr       mar    0
##     Borough
## 1 Manhattan
## 2 Manhattan
## 3 Manhattan
## 4 Manhattan
## 5 Manhattan
## 6 Manhattan
```

```
head(res)
```

```
##            coordinates           Date.Time     Time day month dayofweek hour
## 1 (588201.4, 4513640) 2014-04-01 00:11:00 00:11:00   1   apr       mar    0
## 2 (585514.8, 4509456) 2014-04-01 00:21:00 00:21:00   1   apr       mar    0
## 3 (586298.8, 4512485) 2014-04-01 00:28:00 00:28:00   1   apr       mar    0
## 4 (586753.8, 4512557) 2014-04-01 00:33:00 00:33:00   1   apr       mar    0
## 5 (585408.5, 4508422) 2014-04-01 00:39:00 00:39:00   1   apr       mar    0
## 6 (586176.5, 4512839) 2014-04-01 00:45:00 00:45:00   1   apr       mar    0
##     Borough
## 1 Manhattan
## 2 Manhattan
## 3 Manhattan
## 4 Manhattan
## 5 Manhattan
## 6 Manhattan
```

```
uber_utm <- data.frame(uber_raw_2014$Lon, uber_raw_2014$Lat,
                            res$Lon, res$Lat,

                            res$Borough, res$hour, res$dayofweek)%>%
               mutate(Lon = uber_raw_2014.Lon,
           Lat = uber_raw_2014.Lat,
           Lon_utm = res.Lon,
           Lat_utm = res.Lat,

           Borough = res.Borough,
           Hour = res.hour,
           Weekday = res.dayofweek )  %>%
    select(Lon:Weekday) %>%
    mutate(Lon50m = Lon_utm %/% 50,
           Lat50m = Lat_utm %/% 50) %>%
    group_by(Lon50m, Lat50m) %>%
    mutate(Lon = mean(Lon),
           Lat = mean(Lat),
           Lon_utm = mean(Lon_utm),
           Lat_utm = mean(Lat_utm)) %>% ungroup()

head(uber_utm)
```

```
## # A tibble: 6 x 9
##      Lon   Lat Lon_utm  Lat_utm Borough   Hour  Weekday Lon50m Lat50m
##    <dbl> <dbl>   <dbl>    <dbl> <fct>     <fct> <ord>    <dbl>  <dbl>
## 1 -74.0  40.8 588223. 4513633. Manhattan 0     mar      11764  90272
## 2 -74.0  40.7 585516. 4509460. Manhattan 0     mar      11710  90189
## 3 -74.0  40.8 586279. 4512479. Manhattan 0     mar      11725  90249
## 4 -74.0  40.8 586767. 4512574. Manhattan 0     mar      11735  90251
## 5 -74.0  40.7 585408. 4508418. Manhattan 0     mar      11708  90168
## 6 -74.0  40.8 586174. 4512827. Manhattan 0     mar      11723  90256
```

Events' locations

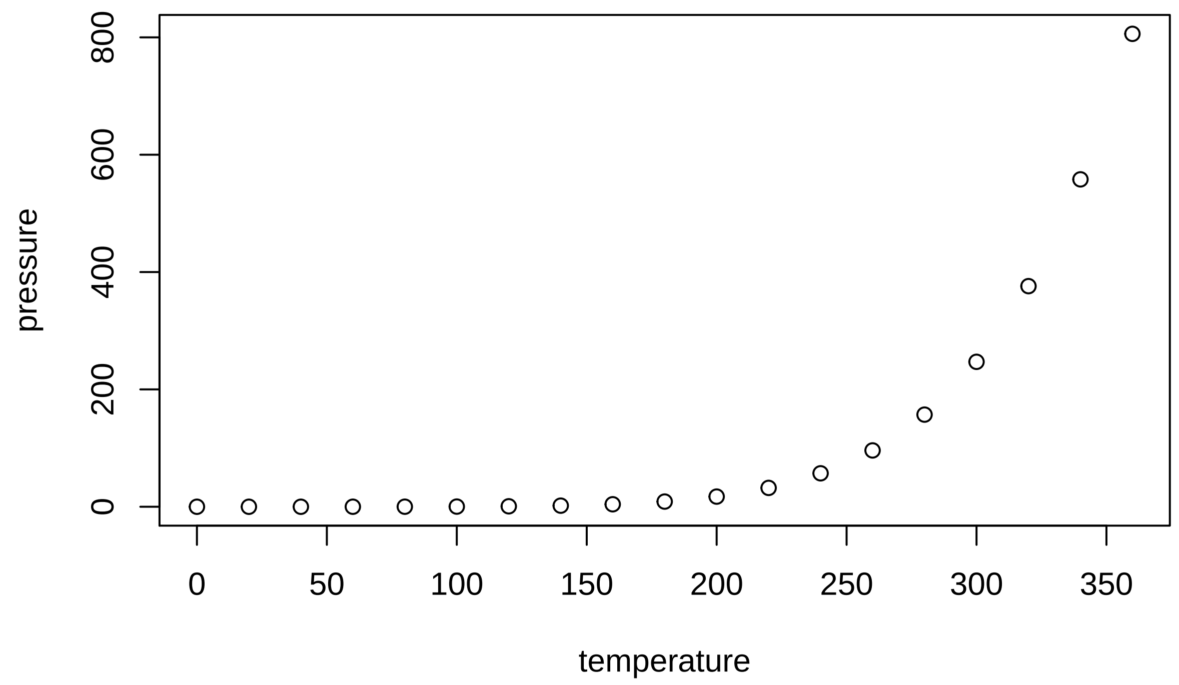
We are now almost ready to plot the final locations of events occured during night weekends in NYC. The variables "Lon" and "Lat" now represents the centroid's coordinates associated to each data item. We just have to group the dataset by "Lon" and "Lat" and count the number of items.

We will only consider the top 30 spots and plot them using the "leaflet" package.

```
Event_place <- uber_utm %>%
    filter( Hour %in% c ("0","1","2") ,
            Weekday %in% c ("sab", "dom", "lun")) %>%
    group_by(Lon, Lat) %>%
    tally() %>%
    arrange(desc(n))
```

lol

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.