Economy of the Big Data World

Final Project – Uber rides prediction

Given Data Base:

Raw table (csv format) that contains data of UBER Rides in New York between the months Apr-Jul 2014. The variables are as following:

- Date & time
- Latitude
- Longitude

Each row represents one instance; an executed UBER ride.

The Task:

Create a linear regression model that will predict the sum of UBER Rides that will be executed during every 15 minutes, between 5-12 PM, within 1km of the New York Stock Exchange.

Part I – Preparing the Data according to the task's frame

Step 1 – Filtering irrelevant Rides & Grouping by 15 minutes:

Hours -

We will filter out all rides that were not executed between 5-12 PM:

```
DB$Date <- as.POSIXct(DB$Date.Time,tz='GMT', format = "%m/%d/%Y %H:%M")
DB$Hour <- hour(DB$Date) # Extracting the Hour from the Date&Time variable
DB <- subset(DB,DB$Hour >= 17)
```

Within 1km from NYSE –

1. We will save 2 variables; Latitude & Longitude of the NYSE.
2. We will use 'distm' function to calculate the distance between each ride and the NYSE and save it under 'Distance' variable.
3. We will filter out irrelevant rides by filtering all >1km distances.

```
NYlat <- 40.706913
NYlong <- -74.011322
DB$Distance <- (distm(cbind(DB$Lon,DB$Lat),cbind(NYlong,NYlat),fun = distHaversine)) / 1000
DB <- subset(DB, DB$Distance <= 1)
```

Sum of Rides in every 15 minutes –

1. We will add 'count' variable for counting rides.
2. We will round down every Date&Time by to 15 minutes point using 'floor_date' function.
3. We will group by Adjusted Date&&Time and sum the 'count' variable using 'dplyr' package.

```
DB$Rides <- 1
DB$Date <- floor_date(DB$Date, '15 minutes')
DS <- DS %>%
  group_by(Date, .drop=FALSE) %>%
  summarise(Rides = sum(Rides))
```

Part II – Adding Relevant Variables

Weather –

After getting Weather data, that includes Hourly data of Temperature, Wind speed, Humidity and General Condition. we will merge it with our DB. Because it is an Hourly data, we will need to multiply each row by 4 in order to match the existing 15 minutes instances (4 times in 1 hour):

1. We will multiply each instance by 4, using ‘slice’ function.
2. We will merge the Weather data with our DB using ‘cbind’ function

```
DW <- DW %>% slice(rep(1:n(), each = 4))
DB <- cbind(DB,DW)
```

Date & Time variables –

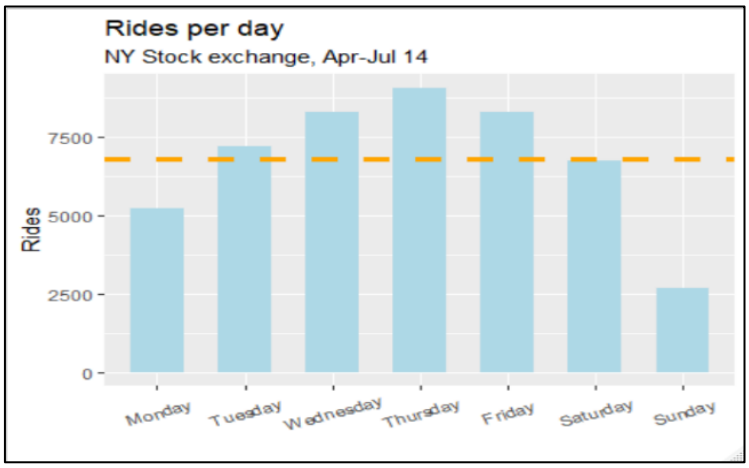
We will split the Date&Time variable into; Months, Weekday (using weekdays function), Hour and Minutes (00 / 15 / 30 / 45).

```
DB$Hour <- as.factor(DB$Hour)
DB$Month <- as.factor(MONTH(DB$Date.Time))
DB$Wday <- weekdays(DB$Date.Time)
DB$Wday <- factor(DB$Wday,levels = c("Monday", "Tuesday", "Wednesday", "Thursday",
"Friday","Saturday", "Sunday"))
```

Part III – Explanatory analysis

Time effects on UBER Rides –

- **Rides per Day:**
  We can see clearly that the number of UBER rides is affected by which day is it during the week. During the weekend, the number of UBER rides is decreasing, probably because NYSE area is much less active during the weekends.

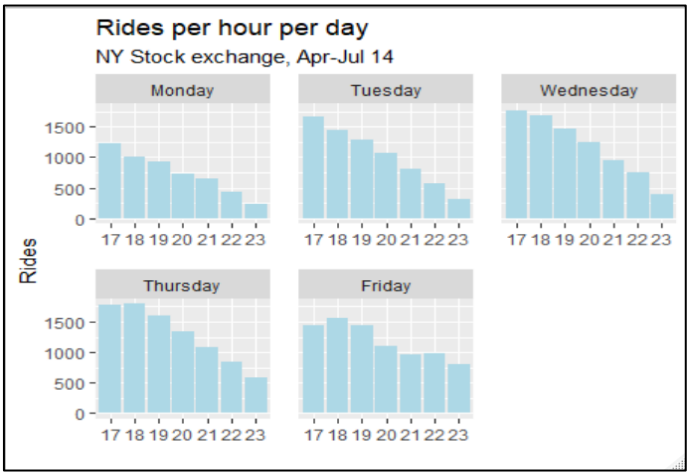



```
DB %>%
  group_by(Wday) %>%
  summarise(Rides = sum(Rides))

mRides = sum(DS$Rides) / 7

ggplot(DB, aes(x=Wday,y=Rides)) +
  geom_bar(stat = 'identity', width = 0.6, fill = 'light blue') +
  labs(x='',title='Rides per day', subtitle = 'NY Stock exchange, Apr-Jul 14') +
  theme(axis.text.x = element_text(angle=20, vjust=0.6)) +
  geom_hline(yintercept= mRides, linetype = 'dashed', color = 'orange', size = 1.5)
```

- **Rides per Hour:**
  When we look only on the workdays, we can see that the distribution of the rides is similar across the week – declining towards the late-night hours.

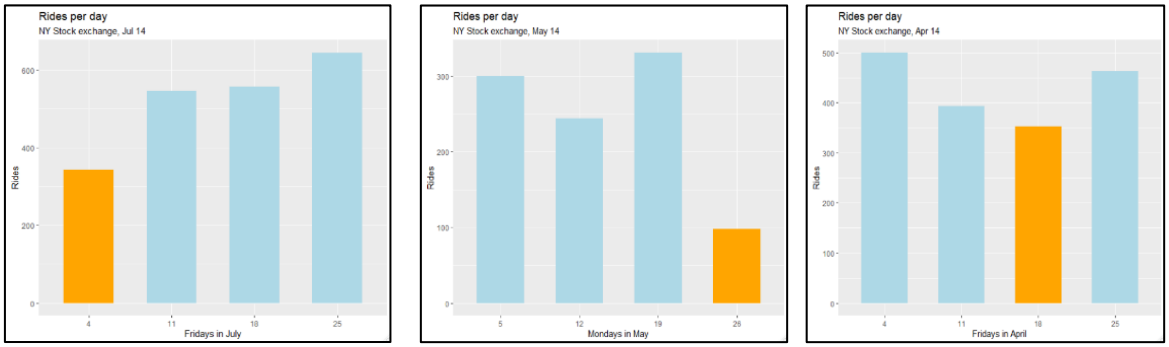


```
DB %>%
  group_by(Hour) %>%
  summarise(Rides = sum(Rides))

DB_Filter <- DB %>%
  filter(DB$Wday != 'Saturday' & DB$Wday != 'Sunday')


ggplot(DB_Filter, aes(x=Hour,y=Rides)) +
  geom_bar(stat = 'identity', width = 0.9, fill = 'light blue') +
  labs(x=' ',title='Rides per hour per day', subtitle = 'NY Stock exchange, Apr-Jul 14')+
  facet_wrap(DB_Filter$Wday, scales = 'free_x') +
  theme(panel.spacing = unit(1, "lines"))
```

- **Holidays:**
  During Apr-Jul 2014, there were 3 Holidays; Good Friday (18 Apr), Memorial Day (26 May) and Independence Day (4 July). As we can see, the number of UBER rides during these dates are lower, compared to the same weekdays.




```
July:
DS_Filter <- subset(DS, Month == 7 & Wday == 'Friday')
ggplot(DS_Filter, aes(x=Day,y=Rides)) +
  geom_bar(stat = 'identity', width = 0.6, fill = ifelse(DS_Filter$Day == '4' ,'orange','light blue')) +
  labs(x='Fridays in July',title='Rides per day', subtitle = 'NY Stock exchange, Jul 14')
```

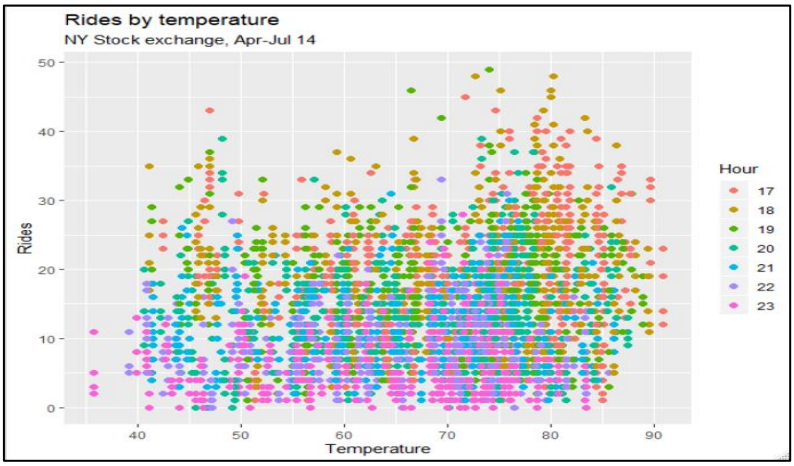
Weather effects on UBER Rides –

According the correlation table with the weather's variables, the most correlated variable with UBER rides is Temperature:

	Rides	Temperature	Wind speed	Relative humidity
Rides	1	0.16	0.08	0.11

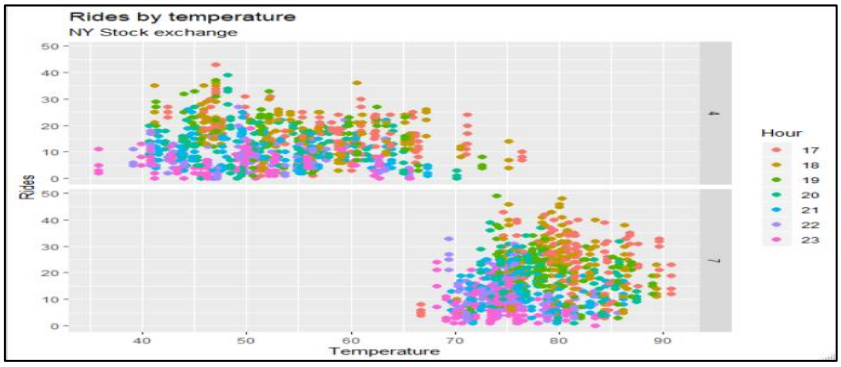
```
cor_weather <- cor(DB, use = 'everything', method = 'pearson')
```

- **Temperature:**
  By Scattering the relation between Temperature and number of rides, we can see that there is kind of ‘V’ shape correlation. That means that there are more rides when the Temperature is extreme (high or low).




```
ggplot(DB, aes(x=Temperature,y=Rides)) +
  geom_point(aes(color = Hour),size = 2) +
  labs(title = 'Rides by temperature', subtitle = 'NY Stock exchange, Apr-Jul 14')
```

By comparing April (relatively cold weather) and July (relatively warm weather), we can see that in April, the correlation is negative and in July, the correlation is positive. Meaning, there are less rides when the temperature is “comfortable”.




```
ggplot(DB_Filter, aes(x=Temperature,y=Rides)) +
  geom_point(aes(color = Hour),size = 2) +
  labs(title = 'Rides by temperature', subtitle = 'NY Stock exchange') +
  facet_grid(DB$Month)
```

Part IV – Prediction Model

Linear Regression -

$$Rides_i = \beta_0 + \beta_1 * Wday * Hour * Minutes + \beta_2 * Temperatue + u_i$$

As we explained above, there is an effect between the different Time Variables. Meaning, 7 PM on a Sunday will be different than 7 PM on a Tuesday. Therefor, we decided to apply interaction on all the Time Variables. From the Weather data, we chose to include only the Temperature variable as it was the most statistically significant variable.

Testing -

We divided the Data into training and test sets as follows:
Training set – April, June
Test set – September (wasn't given until submission).

The results:
R^2 – 0.69
P-Value – 2.2e-16