Uber Data Analysis Project

Sooraj Kumaar

8/7/2021

For this assignment I use RMarkdown, and we first install the package using the code *install.packages(“rmarkdown”)*. once the package is installed, I load the package for use by using *library(“rmarkdown”)*.

library(rmarkdown)

# Overview:

In this project i follow the steps listed in the source code <https://data-flair.training/blogs/r-data-science-project-uber-data-analysis/>. I use the data to give intuitive visualizations using ggplot2.

# Installing and loading Packages:

Install all visualization, data manipulation and date/time packages(ggplot2, ggthemes, lubridate, dplyr, tidyr, DT, scales) below using function call *install.packages()*. Loading required libraries:

library(ggplot2)

Warning: package 'ggplot2' was built under R version 4.0.5

library(ggthemes)

Warning: package 'ggthemes' was built under R version 4.0.5

library(lubridate)

Warning: package 'lubridate' was built under R version 4.0.5

Attaching package: 'lubridate'

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

library(dplyr)

Warning: package 'dplyr' was built under R version 4.0.5

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(tidyr)

Warning: package 'tidyr' was built under R version 4.0.5

library(DT)

Warning: package 'DT' was built under R version 4.0.5

library(scales)

# Discuss the business problem/goal:

The Goal of the Uber R project is to determine factor influencing Uber pickups during 2014 April to September to gain an understanding of the data collected through different visualizations.

# Identify where the dataset was retrieved from:

The data was derived and retrieved from DataFlair website <https://data-flair.training/blogs/r-data-science-project-uber-data-analysis/> which illustrates the code for this project. The website is using the data from Uber and the data is only from April through September.

<https://drive.google.com/file/d/1emopjfEkTt59jJoBH9L9bSdmlDC4AR87/view> - dataset.

# identify the code that imported and saved your dataset in R:

Throughout the code lines below, the code used to import and load the data set is *“read.csv”* from the “utils” package. The data is saved or assigned dataframes such as: *apr\_data, may\_data, jun\_data, jul\_data, aug\_data, sep\_data* and all of these are together combined into one dataframe : *data\_2014*

The data and outputs are stored in a word document as the RMarkdown YAML front matter dictates the output to be in *WORD* format.

## Reading Data

Create dataframes for each of the CSV files, combine all of them in one dataframe and add new columns for day, month, year, day of the week, hour, minute and second:

apr\_data <- read.csv("uber-raw-data-apr14.csv")  
may\_data <- read.csv("uber-raw-data-may14.csv")  
jun\_data <- read.csv("uber-raw-data-jun14.csv")  
jul\_data <- read.csv("uber-raw-data-jul14.csv")  
aug\_data <- read.csv("uber-raw-data-aug14.csv")  
sep\_data <- read.csv("uber-raw-data-sep14.csv")  
  
data\_2014 <- rbind(apr\_data,may\_data, jun\_data, jul\_data, aug\_data, sep\_data)  
  
data\_2014$Date.Time <- as.POSIXct(data\_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S")  
  
data\_2014$Time <- format(as.POSIXct(data\_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S"), format="%H:%M:%S")  
  
data\_2014$Date.Time <- ymd\_hms(data\_2014$Date.Time)  
  
data\_2014$day <- factor(day(data\_2014$Date.Time))  
data\_2014$month <- factor(month(data\_2014$Date.Time, label = TRUE))  
data\_2014$year <- factor(year(data\_2014$Date.Time))  
data\_2014$dayofweek <- factor(wday(data\_2014$Date.Time, label = TRUE))  
data\_2014$hour <- factor(hour(hms(data\_2014$Time)))  
data\_2014$minute <- factor(minute(hms(data\_2014$Time)))  
data\_2014$second <- factor(second(hms(data\_2014$Time)))

# Describe your data set:

The data set contains Uber pickups during 2014 April to September which is combined together in a dataframe *data\_2014*. The data also contains the date and time of pickups and also the Base of the pickups.

summary(data\_2014)

Date.Time Lat Lon   
 Min. :2014-04-01 00:00:00 Min. :39.66 Min. :-74.93   
 1st Qu.:2014-05-28 15:18:00 1st Qu.:40.72 1st Qu.:-74.00   
 Median :2014-07-17 14:45:00 Median :40.74 Median :-73.98   
 Mean :2014-07-11 18:50:50 Mean :40.74 Mean :-73.97   
 3rd Qu.:2014-08-27 21:55:00 3rd Qu.:40.76 3rd Qu.:-73.97   
 Max. :2014-09-30 22:59:00 Max. :42.12 Max. :-72.07   
   
 Base Time day month   
 Length:4534327 Length:4534327 30 : 167160 Apr: 564516   
 Class :character Class :character 12 : 160606 May: 652435   
 Mode :character Mode :character 16 : 158921 Jun: 663844   
 13 : 156892 Jul: 796121   
 23 : 156032 Aug: 829275   
 9 : 155135 Sep:1028136   
 (Other):3579581   
 year dayofweek hour minute second   
 2014:4534327 Sun:490180 17 : 336190 10 : 77757 0:4534327   
 Mon:541472 18 : 324679 14 : 77161   
 Tue:663789 16 : 313400 15 : 77124   
 Wed:696488 19 : 294513 13 : 76957   
 Thu:755145 20 : 284604 12 : 76849   
 Fri:741139 21 : 281460 8 : 76719   
 Sat:646114 (Other):2699481 (Other):4071760

Total number of observations - **4534327**. Total number of variables - **12**. The summary stats for date, time, lat and longitude does not make much sense except for the max and min which just shows the range of the data set between the time frame of April and September. The 30th of day of each month has the highest Uber pick ups. September has the highest Uber pick ups. Thursday and Friday of the week has the highest Uber pick ups as well. During the coarse of the day, between 4 to 6 pm you have the highest Uber pick ups.

# Discuss any data preparation and errors:

There are 6 excel files(monthly files from April to September) which is combined together into one single dataframe. We add 8 more columns/variables: *data\_2014Time* *data\_2014day* *data\_2014year* *data\_2014hour* *data\_2014second* Dataframes are created by grouping values like hour and day for plotting graphs. you will find these codes before plotting each graph. There were no errors found with the dataset itself. there were few errors found with the code and are noted throughout the code.

## Creating colored vectors:

Create color vector to be used in plots of this project:

### error found during this run -

*the original code contains additional double quotes at the start and end of the vector.* *colors = c("“#CC1011”, “#665555”, “#05a399”, “#cfcaca”, “#f5e840”, “#0683c9”, “#e075b0”")*

corrected below:

colors = c("#CC1011", "#665555", "#05a399", "#cfcaca", "#f5e840", "#0683c9", "#e075b0")

#Discuss the data Modeling: In this project the data is prepped by using join function calls to combine the 6 excel files to one dataframe. After combining the dataset is then prepared for analyzing visualizations of Uber pick ups by Month, hour, day and Bases. this is basically gruped for plotting. There is no modelling through regression or other statistical modeling tools.

# Produce and discuss the output, and provide explanation with any visuals:

Below the plots and explanations are provided after every paired visualization.

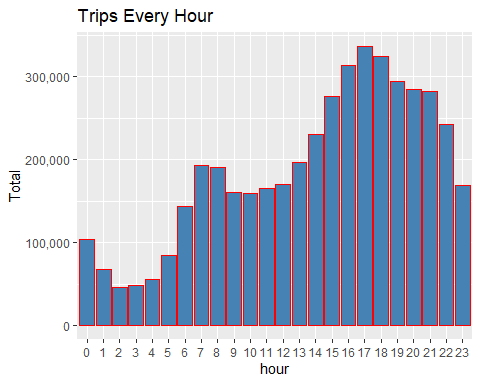
## Analyzing trip by hours in a day:

Create a dataframe that uses dplyr to group by the hour of the day and summarize by the number of trips. Output this in a table. Plot this data and also group the hourly trip data by months; month\_hour:

hour\_data <- data\_2014 %>%  
 group\_by(hour) %>%  
 dplyr::summarize(Total = n())   
datatable(hour\_data)

PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

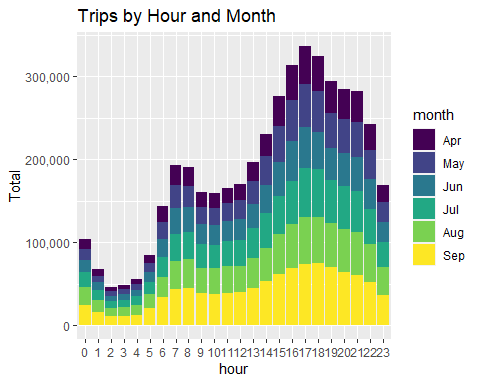
ggplot(hour\_data, aes(hour, Total)) +   
 geom\_bar( stat = "identity", fill = "steelblue", color = "red") +  
 ggtitle("Trips Every Hour") +  
 theme(legend.position = "none") +  
 scale\_y\_continuous(labels = comma)



month\_hour <- data\_2014 %>%  
 group\_by(month, hour) %>%  
 dplyr::summarize(Total = n())

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

ggplot(month\_hour, aes(hour, Total, fill = month)) +   
 geom\_bar( stat = "identity") +  
 ggtitle("Trips by Hour and Month") +  
 scale\_y\_continuous(labels = comma)



**We observe that the number of trips are higher in the evening around 5:00 and 6:00 PM.**

### error found during this run -

*Error: Functions that produce HTML output found in document targeting docx output. Please change the output type of this document to HTML. Alternatively, you can allow HTML output in non-HTML formats by adding this option to the YAML front-matter of your rmarkdown file: always\_allow\_html: true*

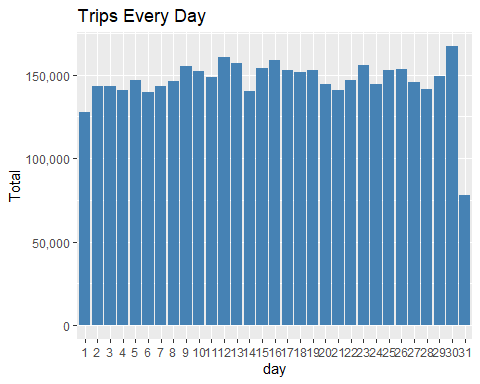
**moving forward we try to plot metrics by day, month, day&month, Bases, Bases&month, Bases&Dayofweek as shown below:**

## Trips Every Day and every day of the month:

same process repeated of grouping and summarize the number of trips by day and every day of month:

day\_group <- data\_2014 %>%  
 group\_by(day) %>%  
 dplyr::summarize(Total = n())   
datatable(day\_group)

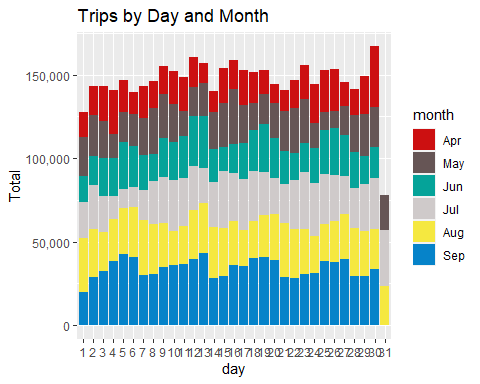
ggplot(day\_group, aes(day, Total)) +   
 geom\_bar( stat = "identity", fill = "steelblue") +  
 ggtitle("Trips Every Day") +  
 theme(legend.position = "none") +  
 scale\_y\_continuous(labels = comma)



day\_month\_group <- data\_2014 %>%  
 group\_by(month, day) %>%  
 dplyr::summarize(Total = n())

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

ggplot(day\_month\_group, aes(day, Total, fill = month)) +   
 geom\_bar( stat = "identity") +  
 ggtitle("Trips by Day and Month") +  
 scale\_y\_continuous(labels = comma) +  
 scale\_fill\_manual(values = colors)



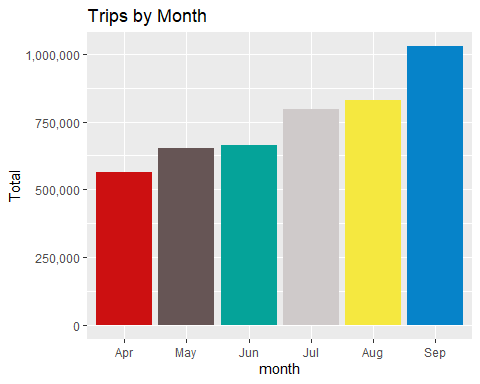
**We observe from the resulting visualization that 30th of the month had the highest trips in the year which is mostly contributed by the month of April**

## Trips by months in a year and by month&dayofeweek:

same process repeated of grouping and summarize the number of trips by months:

month\_group <- data\_2014 %>%  
 group\_by(month) %>%  
 dplyr::summarize(Total = n())   
datatable(month\_group)

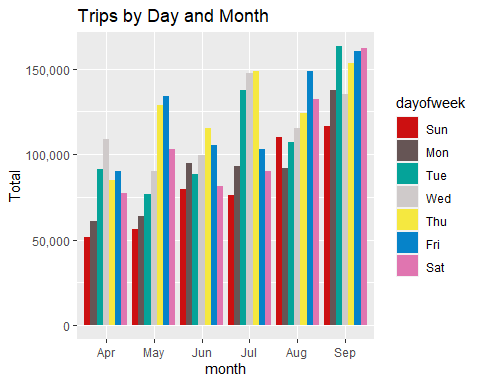
ggplot(month\_group , aes(month, Total, fill = month)) +   
 geom\_bar( stat = "identity") +  
 ggtitle("Trips by Month") +  
 theme(legend.position = "none") +  
 scale\_y\_continuous(labels = comma) +  
 scale\_fill\_manual(values = colors)



month\_weekday <- data\_2014 %>%  
 group\_by(month, dayofweek) %>%  
 dplyr::summarize(Total = n())

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

ggplot(month\_weekday, aes(month, Total, fill = dayofweek)) +   
 geom\_bar( stat = "identity", position = "dodge") +  
 ggtitle("Trips by Day and Month") +  
 scale\_y\_continuous(labels = comma) +  
 scale\_fill\_manual(values = colors)

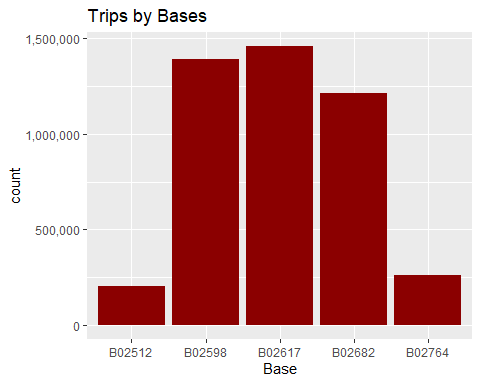


**we observe that most trips were made during the month of September**

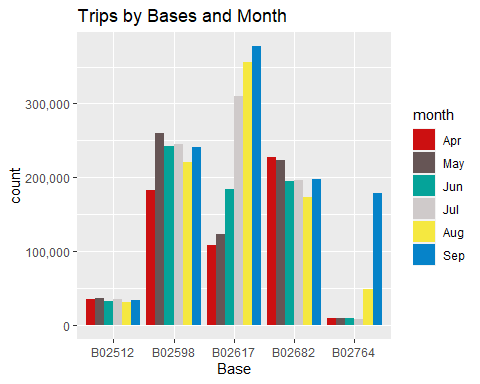
## Trips by Bases, by bases&month and by bases&dayofweek:

same process repeated of grouping and summarize the number of trips by bases:

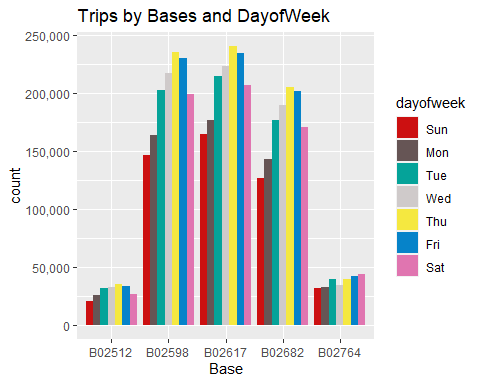
ggplot(data\_2014, aes(Base)) +   
 geom\_bar(fill = "darkred") +  
 scale\_y\_continuous(labels = comma) +  
 ggtitle("Trips by Bases")



ggplot(data\_2014, aes(Base, fill = month)) +   
 geom\_bar(position = "dodge") +  
 scale\_y\_continuous(labels = comma) +  
 ggtitle("Trips by Bases and Month") +  
 scale\_fill\_manual(values = colors)



ggplot(data\_2014, aes(Base, fill = dayofweek)) +   
 geom\_bar(position = "dodge") +  
 scale\_y\_continuous(labels = comma) +  
 ggtitle("Trips by Bases and DayofWeek") +  
 scale\_fill\_manual(values = colors)



**There are five bases in all out of which, we observe that B02617 had the highest number of trips. Furthermore, this base had the highest number of trips in the month September. Thursday observed highest trips in the three bases – B02598, B02617, B02682.**

## Geo-plot:

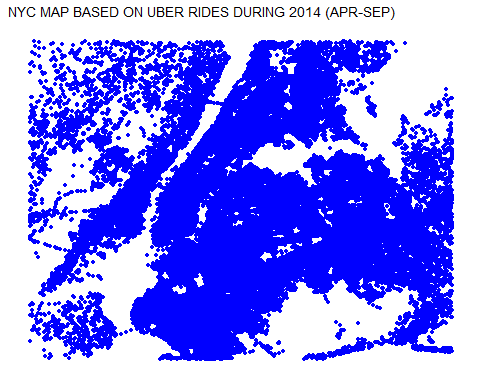
geo-plot of the trips taken by the different bases during 2014(apr-sept)

min\_lat <- 40.5774  
max\_lat <- 40.9176  
min\_long <- -74.15  
max\_long <- -73.7004

first defining the are of teh map to show new york.

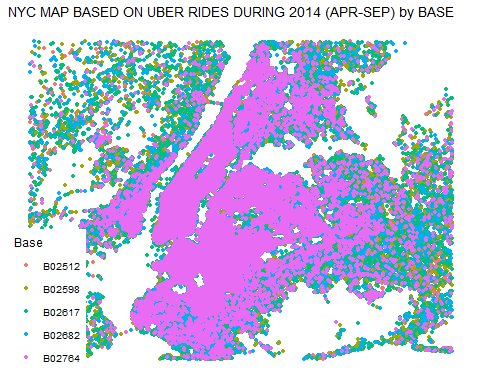
ggplot(data\_2014, aes(x=Lon, y=Lat)) +  
 geom\_point(size=1, color = "blue") +  
 scale\_x\_continuous(limits=c(min\_long, max\_long)) +  
 scale\_y\_continuous(limits=c(min\_lat, max\_lat)) +  
 theme\_map() +  
 ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)")

Warning: Removed 71701 rows containing missing values (geom\_point).



ggplot(data\_2014, aes(x=Lon, y=Lat, color = Base)) +  
 geom\_point(size=1) +  
 scale\_x\_continuous(limits=c(min\_long, max\_long)) +  
 scale\_y\_continuous(limits=c(min\_lat, max\_lat)) +  
 theme\_map() +  
 ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by BASE")

Warning: Removed 71701 rows containing missing values (geom\_point).



# Conclusion:

The intention of this project was to show the use of datavisualization in r and how informative it is. **We conclude that time does affect customer trips**.