Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

**Course: ALY6010 - Probability Theory and Introductory Statistics**

**Assignment:**

Final Project

**Submitted on:**

December 16, 2022

**Submitted to:**  **Submitted by:**

Professor: BEHZAD AHMADI NIKSHITA RANGANATHAN

**INTRODUCTION**

Analyzing massive data sets to find patterns, trends, and correlations is the process of data analytics. It is used in organizations to gain insights into their operations and performance, as well as to make better decisions. It is used in a variety of ways in organizations to identify areas of improvement, discover new opportunities, and optimize processes. For example, data analytics can be used to analyze customer data to identify patterns in customer behavior or to analyze financial data to identify areas of cost savings.

The four primary subcategories of data analysis are descriptive, predictive, diagnostic and prescriptive analytics.

* The technique of summarizing data to better understand the past is known as descriptive analytics. The use of descriptive analytics can provide an explanation for questions such as "What happened?" and "How did that happen?"
* Analyzing data to produce future predictions is the process of predictive analytics. Predictive analytics can be used to answer questions such as “What will happen next?” and “What should we do to prepare?”
* Analyzing data to find a problem's underlying causes is the process of diagnostic analytics Diagnostic analytics can be used to answer questions such as “Why did it happen?” and “What caused it?”
* Prescriptive analytics is the process of analyzing data to recommend actions. Prescriptive analytics' use can offer solutions to problems like "What should we do?" and “How can we optimize our operations?”

**About Dataset: Uber/ Lyft Rideshare**

The way people travel has changed dramatically because of companies like Uber and Lyft. With just a few taps on a smartphone, users can request a ride from a driver and be taken to their destination. This on-demand service is convenient, reliable, and often cheaper than taking a taxi or other form of transportation. They provide a variety of services, such as shared rides and luxury cars, that make it easier and more affordable for customers to get around.

Uber and Lyft both employ dynamic pricing as a pricing approach. It is based on the idea of adjusting prices based on the demand for a particular service at a given time. Prices are typically higher during peak times, such as rush hour, and lower during off-peak times. This allows ride-sharing companies to maximize their profits and helps to reduce congestion on the roads, as it encourages people to use ride-sharing services during off-peak times.

**Purpose and Source of the dataset:**

This project aims to explore the Uber and Lyft dataset is to identify the factors that affect cab prices and to predict cab fares based on these factors. We will also generate insights and visualizations from the data. The data used in this project is an open-source dataset obtained from Kaggle by the data owner, BM.

**Data description :**

This dataset contains information about Uber and Lyft rides in Boston, MA from November 26, 2018, to December 18, 2018, including the source and destination locations, type of cab, product ID, price, distance, date, time, humidity, and surge multiplier. Both categorical and numerical data are included in the dataset. The dataset has 693071 observations and 58 attributes.

***Rideshare dataset:***

|  |  |  |
| --- | --- | --- |
| **No** | **Feature** | **Dictionary** |
| 1 | id | Unique Identification Number |
| 2 | timestamp | The specific point in time (seconds) |
| 3 | Hour | Hour of the day when the cab ride occurred |
| 4 | day | Day when the cab ride occurred |
| 5 | Day | Day of the week(Monday to Sunday) when the cab ride occurred |
| 6 | month | The month when the cab ride occurred |
| 7 | datetime | Date and time of the ride |
| 8 | timezone | The time zone in which the ride occurred |
| 9 | source | Start location of the ride |
| 10 | destination | End location of the ride |
| 11 | cab\_type | Uber/ Lyft |
| 12 | product\_id | An identifier for Cab-type (Uber/Lyft) |
| 13 | name | Name of ride options |
| 14 | price | Price Estimate of the Ride (USD) |
| 15 | distance | Total distance covered |
| 16 | surge\_multiplier | Multiplier to standard rates;dynamic pricing method used by ride-sharing services like Uber and Lyft |
| 17 | latitude | Latitude details of the ride |
| 18 | longitude | Longitude details of the ride |
| 19 | temperature | Temperature (F) |
| 20 | apparentTemperature | The temperature which shows the effects of both air temperature and humidity (F) |
| 21 | short\_summary | Brief description of the weather condition |
| 22 | long\_summary | Detailed overview of the weather condition |
| 23 | precipIntensity | Level of rainfall |
| 24 | precipProbability | Chance of rainfall |
| 25 | humidity | Humidity (%) |
| 26 | windSpeed | Wind Speed (mph) |
| 27 | windGust | The sudden increase in wind speed |
| 28 | windGustTime | Time of wind gust (mph) |
| 29 | visibility | The degree of clearness |
| 30 | temperatureHigh | Highest temperature expected to occur (F) |
| 31 | temperatureHighTime | Time at which the Highest temperature expected to occur |
| 32 | temperatureLow | Lowest temperature expected to occur (F) |
| 33 | temperatureLowTime | Time at which the Lowest temperature expected to occur |
| 34 | apparentTemperatureHigh | High apparent temperature for a given time period of the day |
| 35 | apparentTemperatureHighTime | Time of apparentTemperatureHigh |
| 36 | apparentTemperatureLow | Low apparent temperature for a given time period of the day |
| 37 | apparentTemperatureLowTime | Time of apparentTemperatureLow |
| 38 | icon | Represents the weather conditions |
| 39 | dewPoint | The temperature at which water vapor is completely vaporized in the air. |
| 40 | pressure | Pressure |
| 41 | windBearing | Direction from which the wind is blowing |
| 42 | cloudCover | Amount of sky or atmosphere covered by clouds |
| 43 | uvIndex | The measure of the strength of the sun's ultraviolet radiation |
| 44 | visibility | The maximum distance one can see clearly in the atmosphere |
| 45 | ozone | Index for air quality |
| 46 | sunriseTime | Time of day when the sun rises |
| 47 | sunsetTime | Time of day when the sun sets |
| 48 | moonPhase | The illuminated portion of the moon's visible surface |
| 49 | precipIntensityMax | Maximum Level of rainfall |
| 50 | uvIndexTime | Time of day when the UV index is at its highest |
| 51 | temperatureMin | The lowest temperature that is recorded |
| 52 | temperatureMinTime | Time at which the Lowest temperature is recorded |
| 53 | temperatureMax | The highest temperature that is recorded |
| 54 | temperatureMaxTime | Time at which the Highest temperature is recorded |
| 55 | apparentTemperatureMin | Minimum of the apparent temperature for a given day |
| 56 | apparentTemperatureMinTime | Time of apparentTemperatureMin |
| 57 | apparentTemperatureMax | Maximum of the apparent temperature for a given day |
| 58 | apparentTemperatureMaxTime | Time of apparentTemperatureMax |

*Table 1: Dictionary*

**DATA CLEANING AND MANIPULATION**

* Importing the credit CSV file

All the journeys in Boston between November 26, 2018, and December 18, 2018, are in the "ride" vector.

* Cleaning the rideshare dataset
  + Dropping columns

The dataset was cleaned up by removing all extra columns like visibility, sunrise and sunset times, etc. There are 24 columns now instead of 58 columns.

* Visualization of <NA> values

We found 55095 NA values in the price column and no null values. NA values are removed from the analysis because they can lead to inaccurate results and can create bias in the analysis.

Graphical user interface, application, table

Description automatically generatedChart

Description automatically generated with low confidence

Table

Description automatically generated

A picture containing logo

Description automatically generated

**Figure 1-<NA> values**

* Eliminating the <NA>s

The total number of rows dropped from 693071 to 637976.

* Revising the column names

Renaming column names during data cleaning is important for a few reasons. First, it can help make the data more readable and understandable.

* Removing duplicate rows

Removing duplicate rows helps ensure that the data is accurate and reliable. We used the anyDuplicated() function in R to check for duplicate rows in the dataset. This function returned the index of the first duplicate encountered, confirming that the dataset contained some duplicate values. After removing these duplicates, the dataset contained 634237 records.

A picture containing text

Description automatically generated

**Figure 2- anyDuplicated()**

* Separating the date and time columns

We split the date and time into two different columns to facilitate better analysis.

* + Inserting columns

Two new columns, "price\_per\_distance" and "price\_without\_surge", can be added by dividing the price by the distance and the surge multiplier, respectively.

* Removing the outliers

Outliers are numbers that are significantly different from the rest of the data, and boxplots are frequently used to find these values. In this case, we have one extreme outlier that has been removed from the price column..

Chart, box and whisker chart

Description automatically generated

**Figure 3- Boxplot for identifying Outliers**

* Replacing column values

Using the mutate and recode functions, the column values of Hours have been replaced with categories of Morning, Afternoon, Evening, and Night, with 0-5 hours being categorized as Night, 6-11 hours as Morning, 12-17 hours as Afternoon, and 18-23 as Evening.

* Dropping day column

As we already have a Day column with days of the week, we removed another day column.

* Changing datatypes

The columns (pickup location, drop location, conditions, cab type and cab category) were converted from one data type to factor using as.factor().

**DATA ANALYSIS**

* Analyzing the dataset
  + The dataset "ride1" contains 634236 rows and 26 columns, with data types including factors, characters, numbers, and integers.
  + There are two cab types – Lyft and Uber and 12 pickup and drop locations in the dataset.
* Text, letter

  Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 4 – str() and summary()**

Calendar

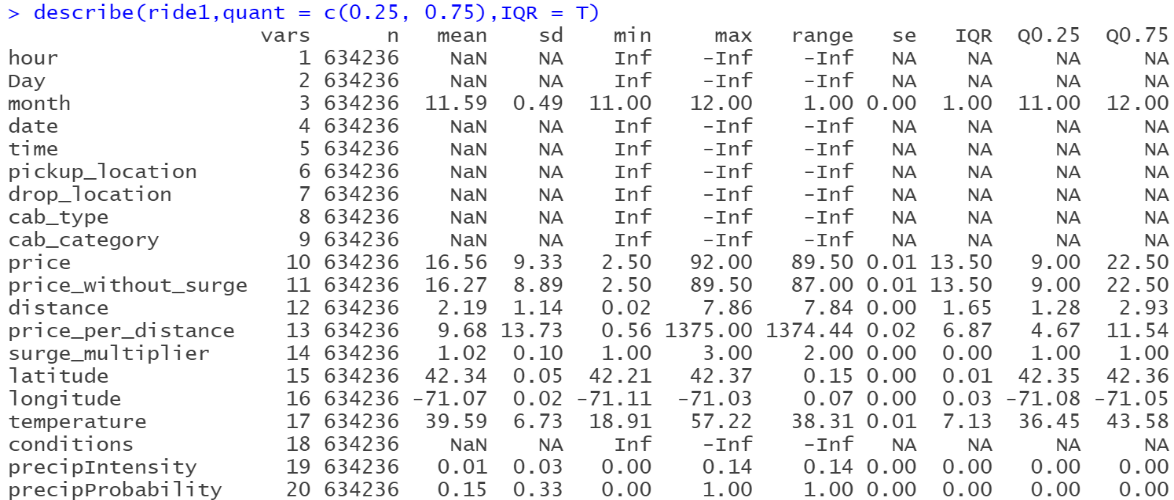
Description automatically generated with low confidence

A picture containing text

Description automatically generated

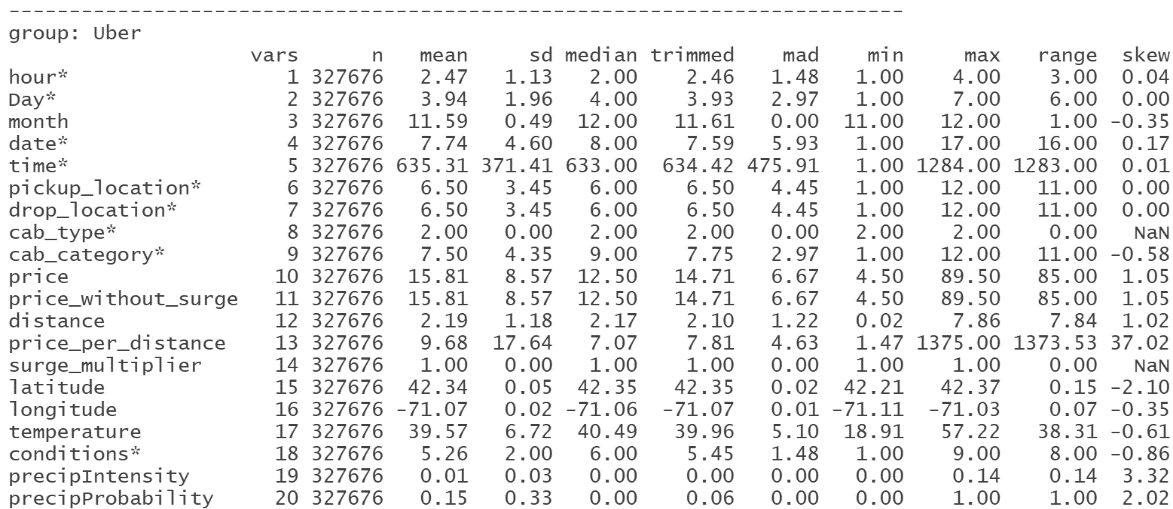
**Figure 5 – headTail() and dim()**

* The mean price for Uber rides is 16.56 USD, with a standard deviation of 9.33, and a minimum of 2.5 USD, and a maximum of 92 USD.
* The mean distance for Uber rides is 2.19 miles, with a standard deviation of 1.14, and a minimum of 0.02 miles, and a maximum of 7.86 miles.
* The function describeBy() provides a statistical summary of the data grouped by cab type, with 327676 records in the Uber category and 306560 in the Lyft category.
* The price feature in Lyft has a skew of 0.98 indicating that the data is slightly skewed to the right whereas the price feature in Uber has a skew of 0.98 indicating that the data is slightly skewed to the left.
* The distance feature in Lyft has a skew of 0.54 (negatively skewed distribution). On the other hand, The distance feature in Uber has a skew of 1.02 (positively skewed).



Table

Description automatically generated



**Figure 6 – describe() and describeBy()**

* Normality tests

These tests are used to determine whether a data set is normally distributed and can be used to identify outliers or other patterns in the data.

* + Density plots

If the data is normally distributed, the density plot will be bell-shaped, with the highest peak at the center and the values tapering off symmetrically on either side. The density plot will be skewed or have several peaks if the data are not normally distributed.

Chart, histogram

Description automatically generated

**Figure 7 – Density plots of features – Price, Distance, Surge multiplier and Price per distance**

The density plot helps us to understand the magnitude of the variable and its usage. The price range of the variable is between 5-30, with most people commuting via Lyft or Uber between 1-4 miles. The surge multiplier charge indicates the most commonly used multiplier, while the price per distance provides a further analysis of the price and distance variable, confirming the visual from the price variable and showing the average price consumers pay per distance traveled.

* + Q-Q plots

QQ plots are used to compare two sets of data to determine if they come from the same distribution. If all of the points on a QQ plot fall along a straight line, a normal distribution will be seen.

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

**Figure 8 – QQplots of features – Price and Distance**

The data for both price and distance variables appears to follow a right-skewed distribution.

* + Box Plots

In a boxplot, the normal distribution can be seen by looking at the median (the middle line of the box), the quartiles (the top and bottom lines of the box. The quartiles should be uniformly spaced apart, and the median should be in the center of the box.

Chart, box and whisker chart

Description automatically generated**Chart, box and whisker chart

Description automatically generated**

**Figure 9 – Boxplots – Price vs Cab Type, Distance vs Cab type**

The box plot of price reveals that most of the prices charged for Uber and Lyft rides are below $25, with the median prices for both cab types varying. Additionally, the average distance covered by both cab types is mainly within the 1-3 mile range, although Uber is more likely to be chosen for trips over 6 miles.

* Hypothesis testing

A statistical method known as hypothesis testing is used to examine a hypothesis on a population parameter. Hypothesis testing is used to make decisions about population parameters, such as the mean, median, or standard deviation. It is also used to assess whether there is a substantial difference between two or more groups of data by comparing them. Hypothesis testing is used in many areas of research, including medical research, economics, and psychology.

They assist in figuring out whether the two samples are statistically significantly different from one another. For example, a two sample test might be used to compare the average test scores of two different classes to see if one class is performing better than the other.

* **Questions we would like to ask and help answer through hypothesis testing**
* Is there a significant difference in the surge multiplier used by Uber and Lyft? Which app can the consumers use in order to save money and avoid paying high surge multipliers?
* Does Uber or Lyft have more coverage in terms of distance traveled?
* Uber or Lyft, Which is more value for money in terms of price paid per distance traveled?

**Test 1**

**Null Hypothesis H0:** The difference in means of surge multiplier between Uber and Lyft is equal to 0.

**Alternate Hypothesis H1:** The difference in means of surge multiplier between Uber and Lyft is not equal to 0.

Text, letter

Description automatically generated

**Figure 10 – Two sample t-test 1**

**Degree of Freedom: 306559**

**Confidence Interval: 95 %**

We reject the null hypothesis of the test because the p-value is below the level of significance of 0.05. There is sufficient evidence to say that Uber and Lyft do not have the same mean surge multiplier.

Uber- 1.0

Lyft- 1.03

**Test 2**

**Null Hypothesis H0:** The difference in means of the distance between Uber and Lyft is equal to 0.

**Alternate Hypothesis H1:** The difference in means of the distance between Uber and Lyft is not equal to 0.

Text, letter

Description automatically generated

**Figure 11 – Two sample t-test 2**

**Degree of Freedom: 634089**

**Confidence Interval: 95 %**

We reject the null hypothesis of the test because the p-value is below the level of significance of 0.05. There is sufficient evidence to say that Both Uber and Lyft do not have the same mean distance covered.

Uber- 2.19

Lyft- 2.18

**Test 3**

**Null Hypothesis H0:** The difference in means of the price per distance between Uber and Lyft is equal to 0.

**Alternate Hypothesis H1:** The difference in means of the price per distance between Uber and Lyft is not equal to 0.

Text

Description automatically generated

**Figure 12 – Two sample t-test 3**

**Degree of Freedom: 450880**

**Confidence Interval: 95 %**

We fail to reject the test's null hypothesis since the p-value is more than the significance level(α) of 0.05. There is insufficient evidence to disprove that both Uber and Lyft have the same mean price per distance.

Uber- 9.6822

Lyft- 9.6819

A statistical test called a one-sample t-test is used to compare a sample mean to an estimated or known population mean.

**Test 1**

**Null Hypothesis H0:** True mean price is greater than 20

**Alternate Hypothesis H1:** True mean price is less than equal to 20

Text

Description automatically generated

**Figure 13 – One sample t-test 1**

**Degree of Freedom: 634235**

**Confidence Interval: 95 %**

We reject the null hypothesis of the test because the p-value is below the level of significance of 0.05. There is sufficient evidence to say that the mean price is less than or equal to 20.

**Test 2**

**Null Hypothesis H0:** True mean distance covered is less than 1.5

**Alternate Hypothesis H1:** True mean distance covered is greater than equal to 1.5

Text

Description automatically generated

**Figure 14 – One sample t-test 2**

**Degree of Freedom: 634235**

**Confidence Interval: 95 %**

We reject the null hypothesis of the test because the p-value is below the level of significance of 0.05. There is sufficient evidence to say that the mean distance covered is greater than or equal to 1.5.

**Test 3**

**Null Hypothesis H0:** True mean surge multiplier is less than 2

**Alternate Hypothesis H1:** True mean surge multiplier covered is greater than equal to 2

Text, letter

Description automatically generated

**Figure 15 – One sample t-test 3**

**Degree of Freedom: 634235**

**Confidence Interval: 95 %**

We fail to reject the test's null hypothesis since the p-value is more than the significance level(α) of 0.05. There is insufficient evidence to disprove that the mean surge multiplier is less than 2.

* Regression Models

Regression is a statistical method for determining how one or more independent variables relate to a dependent variable. In R, regression is done by using the lm() function, which takes a formula and a data frame as its arguments.

The lm() function allows categorical variables to be incorporated into regression models. For each level of the categorical variable, this function will produce a dummy variable. The regression model can then employ the dummy variables as independent variables.

**Model 1 – Price with distance and Day**

**Table

Description automatically generated**

**Figure 16 –Model 1**

The R2 value of 0.1192 indicates that the model does not explain a significant amount of variance in the data. This shows that other factors may be affecting the results and that the model is not a good fit for the data.

**Model 2 – Price with distance and surge multiplier**

**Text

Description automatically generated**

***Figure 17 –Model 2***

The R2 value of 0.173 indicates that the model explains only 17.3% of the variance in the data. This is a relatively low R2 value, which suggests that the model is not a good fit for the data.

**Model 3 – Price with distance, surge multiplier, and price per distance**

**Table

Description automatically generated**

***Figure 18 –Model 3***

The R2 of 0.2891 indicates that the model explains 28.91% of the variance in the data. This is a relatively low R2, suggesting that the model is not a good fit for the data.

**Model 4 – Price with distance, surge multiplier, and price per distance**

Chart

Description automatically generated

Table

Description automatically generated

***Figure 19 –Model 4***

The R2 value of 0.9272 indicates that the model is able to explain 92.72% of the variance in the data. This is a very good result, indicating that the model is able to accurately predict the outcome of the data.

**Model 5 – Price with distance, surge multiplier, and price per distance**

**Chart, table

Description automatically generated with medium confidence**

**Table

Description automatically generated**

**\**

***Figure 20 –Model 5***

The model is a good fit for the data. This means that the model is able to accurately predict the data with a high degree of accuracy. The R2 value of 0.9919 indicates that the model is able to explain 99.19% of the variability in the data.

**DATA VISUALIZATION**

* **GRAPH 1**

Chart, bar chart

Description automatically generated

***Figure 21 –Grouped Bar graph***

* Please copy and paste the link on your web browser to see the interactive version [https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R charts/myChart.html](https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R%20charts/myChart.html) (DO NOT click the link directly)
* We can see that the majority of the consumers prefer to take cabs during overcast conditions.
* Consumers prefer Uber over Lyft.
* During foggy and rainy conditions consumers do not prefer to travel and that is evident by looking at the drop in the graph.
* **GRAPH 2**

**Chart, bar chart

Description automatically generated**

***Figure 22 –Stacked Bar graph***

* Please copy and paste the link on your web browser to see the interactive version [**https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R charts/myChart1.html**](https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R%20charts/myChart1.html)(DO NOT click the link directly)

Map

Description automatically generated

***Figure 23 –Map***

Map

Description automatically generated

***Figure 24 –Density Map***

* This graph (Figure 22) helps us visualize the number of rides in the given locations based on the cab type.
* Financial District in Boston is a popular location which is located in the downtown area, and it tends to have the most number of rides.
* The map (Figure 23) along with the stacked bar graph helps us visualize the prime locations in real-time by comparing and contrasting between the two graph types.
* The density map (Figure 24) helps visualize the density based on the given location.
* **GRAPH 3**

Chart, sunburst chart

Description automatically generated

***Figure 25 –Sunburst chart***

* Please copy and paste the link on your web browser to see the interactive version  [https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R charts/myChart2.html](https://6dp9jllhcaquubqcx8xzqg.on.drv.tw/R%20charts/myChart2.html) (DO NOT click the link directly)
* This sunburst chart gives us an interactive experience of the count with the cab type, cab category and hour of the day.
* **GRAPH 4**

Chart, surface chart

Description automatically generated

***Figure 26 – Multiple density charts***

* Uber pool and Wav have the highest density. Uber black has the lowest density.
* Consumers generally prefer cheaper options such as share and normal rides.
* Shared and normal Lyft have the highest density while Lux black XL and Lyft XL have the lowest density.
* **GRAPH 5**

Chart, bar chart

Description automatically generated

***Figure 27 – Jitter plots***

* Lyft charges the highest price regardless of the hour of the day as we can see charges upward of 75+ in our graph charged frequently by Lyft.
* Uber is comparatively cheaper.
* Morning and night hours have the highest density and are the busiest.
* **GRAPH 6**

Background pattern

Description automatically generated

***Figure 28 – Heatmaps***

* Uber pool and shared have similar rates regardless of the conditions except during overcast conditions.
* The luxury ride types have high prices regardless of the conditions.
* Clear conditions have the cheapest fare amongst all conditions.
* **GRAPH 7**

Chart, box and whisker chart

Description automatically generated

***Figure 29 – Boxplots***

* Lyft has a higher median than Uber.
* Both the mean price range is between 0-25.
* Lyft has a higher number of counts for charges of 70+.
* **GRAPH 8**

Chart, radar chart

Description automatically generated

***Figure 30 – Circular bar graph A***

* This graph shows us the number of rides during the hour of the day along with the surge multiplier.
* 1.25 has the highest number of counts in a day.
* Nighttime has the highest number of surge multipliers.
* **GRAPH 9**

Diagram

Description automatically generated with low confidence

***Figure 31 – Bubble graph***

* Lyft charges the highest price for shorter distances compared to Uber.
* Lyft also charges higher surge charges than Uber.
* Consumers also prefer Uber or Lyft for longer distances because of the cheaper price options.
* Chart, radar chart

  Description automatically generated**GRAPH 10**

***Figure 32 – Circular bar chart***

* Monday and Tuesday have the highest counts out of the week.
* Consumers prefer Uber over Lyft.
* Wednesday has the lowest count out of the week.
* **GRAPH 11**

Chart, bar chart

Description automatically generated

***Figure 33 – Histograms***

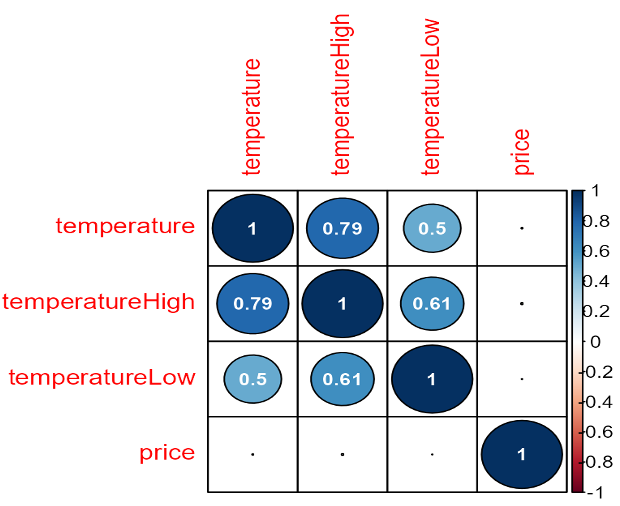
* The majority concentration for the price range is between 0.25.
* The average distance traveled is between 1-4 miles range.
* The average temperature is between 35-45 F.
* **GRAPH 12**

Chart

Description automatically generated with medium confidenceChart

Description automatically generated

***Figure 34 – Correlation matrix (1)***

* Distance and surge multipliers have a positive weak level of correlation with price
* All climate-related features also have a weak correlation with our target price.
* **GRAPH 13**

**Chart, treemap chart

Description automatically generated**

***Figure 34 – Correlation matrix (2)***

* All temperature-related features have a weak correlation with our target price.
* Geographic coordinates have a low correlation with price.
* **GRAPH 14**

**Chart

Description automatically generated**

***Figure 35 – Scatterplot***

* Price and distance are positively correlated, and the majority of the data plots lay close to the regression line.
* **Chart, scatter chart

  Description automatically generatedGRAPH 15**

***Figure 36 – 3D Plot***

* Lyft charges a higher surge multiplier sometimes as high as 3x.
* **GRAPH 16**

**A picture containing logo

Description automatically generated**

**Text

Description automatically generated**

***Figure 37– Wordclouds***

* Most frequently used words in the data set for the location and cab category variables.

**CONCLUSION**

We picked this data set to help make commuting easier for consumers by predicting fare prices. This was possible because of the Hypothesis testing we conducted to understand the relationship between the variables in detail and create regression models by using the variables given in the data set and creating new ones for better adaptability. The 2 new variables created (price without surge multiplier and price per distance) play a vital role in understanding the price matrix between Uber and Lyft in depth. We created a number of regression models by playing around with the variables to deploy the best possible model that fits the data set. Model 5 with the variables price ~ distance, price without surge multiplier, surge multiplier, and cab category gave us a R2 of 0.9919, which means that the model is able to explain 99% of the variances.

The key takeaways from the analysis

* People prefer Uber over Lyft.
* Uber covers more distance than Lyft.
* Lyft charges a higher surge price.
* Nighttime is the most expensive time to take a cab.
* The busiest days of the week are Monday and Tuesday.
* Relative to other days, Wednesday is the less busy.
* Surge multiplier and distance traveled are positively correlated with the price-output.
* Consumers avoid carbs during rainy and foggy conditions.
* Overcast conditions are expensive to travel on.
* Clear conditions are the cheapest.
* Temperature does not play a key role in determining the price of the fare.
* Lyft has a very high upper bound of charging rates of 75+ and a surge multiplier of 3x.

**REFERENCES**

Bluman, A. G. (2018). Elementary Statistics, 10th ed. McGraw Hill.

Kabacoff, R. I. (2011). R in action: Data analysis and graphics with R. Manning Publications Co.

Uber and Lyft Dataset Boston, MA. (2019, October 13). Kaggle.

[https://www.kaggle.com/datasets/brllrb/Uber-and-Lyft-dataset-boston-ma](https://www.kaggle.com/datasets/brllrb/uber-and-lyft-dataset-boston-ma)

Holtz, Y. (n.d.). The R Graph Gallery – Help and inspiration for R charts. The R Graph Gallery.

https://r-graph-gallery.com/index.html

**APPENDIX**

#---------------------- Week\_4\_Module\_4 R Script ----------------------#

print("Final Assignment Milestone 2")

print("Course Name - ALY6010: Probability Theory and Introductory Statistics")

# Importing rideshare dataset containing 693071 observations with 58 variables

getwd()

ride<-read.csv("rideshare.csv")

# Installing and loading the libraries

library(visdat)

install.packages("tidyr")

library("tidyr")

library(car)

library(dplyr)

library(psych)

install.packages("skimr")

library(skimr)

library(naniar)

library(ggplot2)

install.packages("hrbrthemes")

library(hrbrthemes)

library(tm)

library(SnowballC)

library(RColorBrewer)

library(wordcloud)

library(dplyr)

install.packages("scatterplot3d")

library("scatterplot3d")

# Dropping columns

drop<-c("id","timestamp","product\_id","apparentTemperature","long\_summary","windGustTime",

"temperatureHighTime","temperatureLowTime","apparentTemperatureHigh","apparentTemperatureHighTime",

"apparentTemperatureLow","apparentTemperatureLowTime","icon","dewPoint","pressure","windBearing","cloudCover",

"uvIndex","visibility.1","ozone","sunriseTime","sunsetTime","moonPhase","precipIntensityMax",

"uvIndexTime","temperatureMin","temperatureMinTime","temperatureMax","temperatureMaxTime",

"apparentTemperatureMin","apparentTemperatureMinTime","apparentTemperatureMax","apparentTemperatureMaxTime","timezone")

ride<-ride[,!(names(ride) %in% drop)]

# Visualization and Checking NA values

gg\_miss\_which(ride)

gg\_miss\_var(ride)

miss\_var\_summary(ride)

sum(is.na(ride))

sum(is.null(ride))

complete.cases(ride)

which(!complete.cases(ride))

na<-which(!complete.cases(ride))

ride1<- ride[-na,]

# Changing the column names

colnames(ride1)[6]<-"pickup\_location"

colnames(ride1)[7]<-"drop\_location"

colnames(ride1)[9]<-"cab\_category"

colnames(ride1)[16]<-"conditions"

# Checking for duplicated rows and removing them

duplicated(ride1)

anyDuplicated(ride1)

ride1<-ride1[!duplicated(ride1), ]

# Seperating date and time into different columns

ride1<-ride1 %>%

separate(datetime, c("date", "time"), " ")

# Adding columns

ride1$price\_per\_distance<-round(ride1$price/ride1$distance,2)

ride1<-ride1 %>% relocate(price\_per\_distance,.after = distance)

ride1$price\_without\_surge<-round(ride1$price/ride1$surge\_multiplier,1)

ride1<-ride1 %>% relocate(price\_without\_surge,.after = price)

# checking outliers

boxplot(ride1$price)

ride1<-subset(ride1,price!=97.5)

# Replacing hours to Morning, Afternoon,Evening and Night

#0-5 night

#6-11 morning

#12-17 afternoon

#18-24 evening

ride1<-ride1 %>% mutate(hour = recode(hour, '0' = 'Night', '1' = 'Night', '2' = 'Night','3' = 'Night','4' = 'Night','5' = 'Night','6' = 'Morning','7' = 'Morning','8' = 'Morning','9' = 'Morning','10' = 'Morning','11' = 'Morning','12' = 'Afternoon','13' = 'Afternoon','14' = 'Afternoon','15' = 'Afternoon','16' = 'Afternoon','17' = 'Afternoon','18' = 'Evening','19' = 'Evening','20' = 'Evening','21' = 'Evening','22' = 'Evening','23' = 'Evening'))

# Dropping day column (numeric)

drop2<-c("day")

ride1<-ride1[,!(names(ride1) %in% drop2)]

# Changing the datatypes

ride1$hour<-as.factor(ride1$hour)

ride1$pickup\_location<-as.factor(ride1$pickup\_location)

ride1$drop\_location<-as.factor(ride1$drop\_location)

ride1$conditions<-as.factor(ride1$conditions)

ride1$cab\_type<-as.factor(ride1$cab\_type)

ride1$cab\_category<-as.factor(ride1$cab\_category)

# Analysis

headTail(ride1)

str(ride1)

summary(ride1)

dim(ride1)

skim(ride1)

describe(ride1,quant = c(0.25, 0.75),IQR = T)

glimpse(ride1)

describeBy(ride1,group=ride1$cab\_type,quant = c(0.25, 0.75), IQR = T)

# Normal QQplots

qqnorm(ride1$price, pch = 1, frame = FALSE,main="Q-Q Plot (Price)")

qqline(ride1$price, col = "tomato", lwd = 2)

qqnorm(ride1$distance, pch = 1, frame = FALSE,main="Q-Q Plot (Distance)")

qqline(ride1$distance, col = "blue", lwd = 2)

#Density plots

normality\_price <- ggdensity(ride1$price, main = "Density plot of Price", xlab = "Price", fill = "#baf54c")

normality\_distance <- ggdensity(ride1$distance, main = "Density plot of Distance", xlab = "Distance", fill = "#4cf5bd")

normality\_surge.multiplier <- ggdensity(ride1$surge\_multiplier,xlim=c(1,1.25), main = "Density plot of surge Multiplier", xlab = "Surge Multiplier", fill = "#40c7f7")

normality\_price.per.distance <- ggdensity(ride1$price\_per\_distance,xlim=c(0,100), main = "Density plot of Price Per Distance", xlab = "Price Per Distance", fill = "#d27afa")

grid.arrange(normality\_price, normality\_distance , normality\_surge.multiplier, normality\_price.per.distance)

# Boxplots

ggplot(ride1, aes(x=cab\_type,y=price, fill=cab\_type)) + geom\_boxplot() +scale\_fill\_brewer(palette = "Spectral") + theme(text = element\_text(size = 20))+ ylab("Price")+ggtitle("Boxplot")

ggplot(ride1, aes(x=cab\_type,y=distance, fill=cab\_type)) + geom\_boxplot() + scale\_fill\_brewer(palette = "Dark2") + theme(text = element\_text(size = 20))+ylab("Distance") + ggtitle("Boxplot")

#Hypothesis testing

# Subgroups

Uber<-subset(ride1,cab\_type=="Uber")

Lyft<-subset(ride1,cab\_type=="Lyft")

#Two sample T-test

#Test 1

#Both Uber and Lyft have the same mean surge multiplier

t.test(Uber$surge\_multiplier ,Lyft$surge\_multiplier,var.equal = FALSE)

#Test 2

#Both Uber and Lyft have the same mean price

t.test(Uber$price ,Lyft$price,var.equal = FALSE)

#Test 3

#Both Uber and Lyft have the same mean distance covered

t.test(Uber$distance ,Lyft$distance,var.equal = FALSE)

#Test 4

#Both Uber and Lyft have the same mean price per distance

t.test(Uber$price\_per\_distance ,Lyft$price\_per\_distance,var.equal = FALSE)

#Test 5

#Both Uber and Lyft have the same mean price without surge.

t.test(Uber$price\_without\_surge ,Lyft$price\_without\_surge,var.equal = FALSE)

# One sample T-test

#Test 1

#Null Hypothesis: mean price is greater than 20.

#Alternate hypothesis: mean price is < than or = 20

t.test(ride1$price, mu=20, alternative = "less")

#Test 2

#Null Hypothesis: mean distance covered is lesser than 1.5

#Alternate hypothesis: mean distance covered is > than or = 1.5

t.test(ride1$distance, mu=1.5, alternative = "greater")

#Test 3

#Null Hypothesis: mean surge multiplier is lesser than 2

#Alternate hypothesis: mean distance covered is > than or = 2

t.test(ride1$surge\_multiplier, mu=2, alternative = "greater")

#Test 4

#Null Hypothesis: mean price per distance is greater than 5

#Alternate hypothesis: mean price per distance is < than or = 5

t.test(ride1$price\_per\_distance, mu=5,alternative = "less")

#Test 5

#Null Hypothesis: mean temperature is = 45

#Alternate hypothesis: mean temperature is != 45

t.test(ride1$temperature, mu=45)

# Data Visualizations

# Graph 1

df1<-ride1 %>% group\_by(conditions,cab\_type) %>% summarize(count=n())

ggplot(df1, aes(conditions,count, fill = cab\_type)) +

geom\_bar(stat = "identity", position = "dodge")+

scale\_fill\_brewer(palette="Accent")+ggtitle("Grouped Bar chart")+xlab("Conditions")+ylab("Count")+labs(fill = "Cab Type")+theme(axis.text.x = element\_text(angle = 90, hjust = 1))

# Graph 2

require(ochRe)

library(ochRe)

df2<-ride1 %>% group\_by(pickup\_location,cab\_type) %>% summarize(count=n())

ggplot(df2, aes(count,pickup\_location, fill = cab\_type)) +

geom\_bar(stat = "identity",colour = "black")+

scale\_fill\_ochre(palette="mccrea")+xlim(0,60000)+ggtitle("Stacked Bar chart")+xlab("Count")+ylab("Location")+labs(fill = "Cab Type")+theme(axis.text.x = element\_text( hjust = 1))

# Graph 3

library(gridExtra)

Uber<-subset(ride1,cab\_type=="Uber")

Lyft<-subset(ride1,cab\_type=="Lyft")

Den1<-ggplot(Uber, aes(x=price, group=cab\_category, fill=cab\_category)) +

geom\_density(adjust=4, alpha=.4) +xlim(0,60)+xlab("Price")+ylab("Density")+labs(fill = "Cab Category")+theme\_ipsum()

Den2<-ggplot(Lyft, aes(x=price, group=cab\_category, fill=cab\_category)) +

geom\_density(adjust=4, alpha=.4) +xlim(0,60)+xlab("Price")+ylab("Density")+labs(fill = "Cab Category")+theme\_ipsum()

grid.arrange(Den1,Den2,top=textGrob("Multiple density charts of Price for various Cab Categories"))

# Graph 4

cols <- c("dark blue", "orange")

with(ride1, scatterplot3d(price,distance,surge\_multiplier,

main="3 D Plot",xlab = "Price",ylab = "Distance",zlab = "Surge Multiplier",pch = 16, color=cols[as.numeric(ride1$cab\_type)]))

legend("top",inset = c(0, 1), legend = levels(ride1$cab\_type),col =cols, pch =16,xpd=TRUE,horiz=TRUE,bty = "n")

# Graph 5

library(wesanderson)

ggplot(ride1, aes(hour,price, col=cab\_type))+ scale\_color\_manual(values= wes\_palette("GrandBudapest1", n = 2))+

geom\_point(size=0.5,position = position\_jitter(width = .15, height = 4))+ggtitle("Jitter plot")+xlab("Hour")+ylab ("Price")+labs(color = "Cab Type")

# Graph 6

ride1$Day <- factor(ride1$Day, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

ggplot(ride1, aes(x=`Day`, y=`price`, fill=cab\_type)) +

geom\_boxplot() + facet\_wrap(~cab\_type)+

theme\_ipsum() +scale\_fill\_ochre("qalah")+labs(fill = "Cab Type")+ggtitle("Boxplots")+xlab("Days")+ylab("Price")+

theme(axis.text.x = element\_text(angle=90, hjust=1))

# Graph 7

ggplot(ride1, aes(`conditions`, `cab\_category`, fill= `price`)) +

geom\_tile() +theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))+

scale\_fill\_distiller(palette = "Spectral") + ggtitle("Heat Map")+xlab("Conditions")+ylab("Cab Category")+labs(fill = "Price")

# Graph 8

df5<-ride1 %>% group\_by(Day,cab\_type) %>% summarize(count=n())

ggplot(df5, aes(Day,count,fill=cab\_type)) +

geom\_col(position = "dodge")+coord\_polar()+

scale\_fill\_ochre(palette="parliament")+ggtitle("Circular Bar chart")+xlab("Days of the week")+ylab("Count")+labs(fill = "Cab Type")+theme(axis.text.x = element\_text( hjust = 1))

# Graph 9

install.packages("highcharter")

library(highcharter)

library(dplyr)

df4<-ride1 %>% group\_by(cab\_type,hour,cab\_category) %>% summarize(count=n())

dout<-data\_to\_hierarchical(df4,c(hour,cab\_type,cab\_category),count)

hchart(dout,type="sunburst")

# Graph 10

df3<-ride1 %>% group\_by(surge\_multiplier,distance,price,cab\_type) %>% summarize(count=n())

library(viridis)

colours = c( "#F5A2A2", "#4682B4")

ggplot( df3,aes(x=distance, y=price,size=surge\_multiplier, color=cab\_type)) +

geom\_point(alpha=0.7) +scale\_color\_manual(values = colours) +scale\_size(range = c(.1, 2))+

ggtitle("Price vs Distance for both cab types") +

theme\_ipsum() +ylab("Price")+xlab("Distance")

# Graph 11

df6 <- ride1 %>%

filter(surge\_multiplier > 1.00) %>%

group\_by(hour,surge\_multiplier) %>%

dplyr::summarize(total\_rides=n())

p<-ggplot(data = df6, aes(x = hour, y = total\_rides, fill = factor(surge\_multiplier))) +

geom\_bar(stat = "identity", color = "black",lwd = 1, show.legend = TRUE)

p + coord\_polar(theta = "y")

# Regression chart

model<-subset(ride1,surge\_multiplier<=1)

Var1 <- model %>%group\_by(price, distance)

Var2 <- Var1 %>%group\_by(cab\_type, distance) %>%summarize\_at(vars(price), list(cab\_category = mean))

col3<-c("brown1","royalblue4")

Reg1<-ggplot(Var2, aes(distance, cab\_category, color=cab\_type)) +

xlab("Distance") + ylab("Price")+geom\_point(size=.5,shape=4)+scale\_color\_manual(values = col3)+geom\_smooth(method = "lm",se=FALSE)

Reg2<-ggplot(Var2, aes(distance, cab\_category)) +xlab("Distance") + ylab("Price")+geom\_point(size=.5,shape=4)+scale\_color\_manual(values = col3)+geom\_smooth(method = "lm",se=FALSE)

grid.arrange(Reg1,Reg2)

# Histograms

library(ggpubr)

hist1<-ggplot( ride1,aes(x=price)) +geom\_histogram(binwidth = 5,fill= "#00A9FF",colour = "black")

hist2<-ggplot( ride1,aes(x=distance)) +geom\_histogram(binwidth = 0.5,fill= "#A54657",colour = "black")

hist3<-ggplot( ride1,aes(x=surge\_multiplier)) +geom\_histogram(binwidth = 0.15,fill= "#97ce4c",colour = "black")

hist4<-ggplot( ride1,aes(x=temperature)) +geom\_histogram(binwidth = 2.5,fill= "#00C19A",colour = "black")

hist5<-ggplot( ride1,aes(x=humidity)) +geom\_histogram(binwidth = 0.04,fill= "#F39B7FFF",colour = "black")

hist6<-ggplot( ride1,aes(x=precipIntensity)) +geom\_histogram(binwidth = 0.01,fill= "#938dd2",colour = "black")

hist7<-ggplot( ride1,aes(x=windSpeed)) +geom\_histogram(binwidth = 1,fill= "#F8766D",colour = "black")

hist8<-ggplot( ride1,aes(x=windGust)) +geom\_histogram(binwidth = 1.5,fill= "#00BFC4",colour = "black")

hist9<-ggplot( ride1,aes(x=visibility)) +geom\_histogram(binwidth = 0.75,fill= "#FF68A1",colour = "black")

grid.arrange(hist1,hist2,hist3,hist4,hist5,hist6,hist7,hist8,hist9,top=textGrob("Histograms of Variables"))

# Wordcloud 1

ride2=as.character(ride1$drop\_location)

word.corpus<-Corpus(VectorSource(ride2))

word.corpus<-word.corpus%>% tm\_map(removePunctuation)%>%

tm\_map(removeNumbers)%>%

tm\_map(stripWhitespace)%>%

tm\_map(tolower)%>%

tm\_map(stopwords("english"))

countofwords<-as.matrix(TermDocumentMatrix(word.corpus))

freqofwords<-sort(rowSums(countofwords), decreasing=TRUE)

head(freqofwords)

library(wesanderson)

dev.new(width=10, height=5, unit="in")

wordcloud(words=names(freqofwords), freq=freqofwords,scale=c(4, .5),max.words = 100, random.order = FALSE,color=wes\_palette("GrandBudapest1"))

# Wordcloud 2

ride3=as.character(ride1$cab\_category)

word.corpus2<-Corpus(VectorSource(ride3))

word.corpus2<-word.corpus2%>% tm\_map(removePunctuation)%>%

tm\_map(removeNumbers)%>%

tm\_map(stripWhitespace)%>%

tm\_map(tolower)%>%

tm\_map(stopwords("english"))

countofwords2<-as.matrix(TermDocumentMatrix(word.corpus2))

freqofwords2<-sort(rowSums(countofwords2), decreasing=TRUE)

head(freqofwords2)

dev.new(width=10, height=5, unit="in")

wordcloud(words=names(freqofwords2), freq=freqofwords2,scale=c(4, .5),max.words = 100, random.order = FALSE,color=wes\_palette("Zissou1"))

# Maps

lat1<-c(42.364,42.3661,42.352,42.3661,42.3398,42.3647,42.3503,42.3588,42.3505,42.3559,42.3505,42.3519)

lon1<-c(-71.060,-71.0892,-71.065,-71.1054,-71.0810,-71.0551,-71.1054,-71.0707,-71.0542,-71.0550,-71.0631,-71.0631)

area<-c("North Station","North End","West End","Financial District","Beacon Hill","Fenway", "South Station",

"Theatre District","Back Bay", "Boston University","Northeastern University","Haymarket Square")

map1<-data.frame(lat1,lon1,area)

bostonmap<-get\_stamenmap(bbox = c(left=-71.26,right=-70.87,top= 42.45,bottom=42.25),

maptype = "terrain",zoom=12)

ggmap(bostonmap) +geom\_point(data=map1, aes(x=lon1, y=lat1),size=0.5)+

labs(x = 'Longitude', y = 'Latitude') +

geom\_label\_repel(data=map1, aes(x = lon1, y= lat1,

label = area), fill = "grey",

label.size = 0, family="mono",

min.segment.length = 0,

max.overlaps = Inf,

box.padding = unit(.6, "lines"),

label.padding = unit(.10, "lines"),

segment.color = "black", segment.size = .5)

theme\_map()

# Ubermap

ggmap(bostonmap) + stat\_density\_2d(data=Uber, aes(x=longitude, y=latitude,fill=..level..), geom="polygon",bins=150,alpha=.1) +

ggtitle("Density Map of Uber rides in Boston")+scale\_fill\_viridis\_c(option = "inferno")

grid.arrange(Boston,Uber)

# LyftMap

ggmap(bostonmap) + stat\_density\_2d(data=Lyft, aes(x=longitude, y=latitude,fill=..level..),bins=150,alpha=.1, geom="polygon") +

ggtitle("Density Map of Lyft Rides in Boston")+ scale\_fill\_viridis\_c(option = "turbo")

# Correlation Matrix

corr<-ride1 %>% select(temperature,temperatureHigh,temperatureLow,price)

corrplot(cor(corr),addCoef.col = "white",number.cex = 0.8,

number.digits = 2,bg="white",outline = "black",addgrid.col = "black")

corr1<-ride1 %>% select(precipIntensity,precipProbability,humidity,windSpeed,windGust,visibility,price)

corrplot(cor(corr1),addCoef.col = "white",number.cex = 0.8,

number.digits = 2,bg="grey",outline = "black",addgrid.col = "white",method="shade")

corr2<-ride1 %>% select(latitude,longitude,price)

corrplot(cor(corr2),addCoef.col = "white",number.cex = 0.8,

number.digits = 2,bg="grey",outline = "black",addgrid.col = "white",method = 'color')

corr3<-ride1 %>% select(distance,surge\_multiplier,price)

corrplot(cor(corr3),addCoef.col = "red",number.cex = 0.8,

number.digits = 2,bg="white",outline = "black",addgrid.col = "grey",method = 'pie')

# Regression Models

model1<-lm(formula = price ~ distance+Day, data = ride1)

summary(model1)

model2<-lm(formula = price ~ distance+surge\_multiplier, data = ride1)

summary(model2)

model3<-lm(formula = price ~ distance+surge\_multiplier+price\_per\_distance,data = ride1)

summary(model3)

model4<-lm(formula = price ~ distance+surge\_multiplier+cab\_category, data = ride1)

summary(model4)

model5<-lm(formula = price ~ distance+surge\_multiplier+price\_without\_surge+cab\_category, data = ride1)

summary(model5)