Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

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

**Assignment:**

Final Project – Milestone 2

**Submitted on:**

December 4, 2022

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

Professor: BEHZAD AHMADI ARCHIT BARUA

NIKSHITA RANGANATHAN

**ABSTRACT**

Two of the most well-known ride-sharing businesses in existence today are Uber and Lyft. Both companies have made it easier than ever for people to get around, by providing on-demand transportation at the push of a button. They provide a reliable and convenient way to connect drivers with passengers. Passengers can request a ride from a driver with the press of a button, and the driver can then pick them up and drop them off at their destination. The services make it easier for people to travel, and they can often be cheaper than taking a taxi or other form of transportation.

These companies have revolutionized the transportation industry. They offer a variety of services, such as shared rides and luxury cars, and they make it easy for customers to get around. Customers can save money, time, and hassle by using these services, as well as have the convenience of a reliable ride whenever they need it. In addition, Uber and Lyft have made it easier for drivers to earn money. Drivers can sign up to be Uber or Lyft drivers and make money by providing rides to passengers. For drivers, this can be a wonderful way to boost their income or even make a livelihood.

Hypothesis testing is a form of statistical inference used to determine whether or not a certain hypothesis is true. Based on a sample, it is used to form judgments about a population. The test then involves collecting data from a sample and using statistical analysis to determine the probability that the null hypothesis is true. The process of hypothesis testing involves four steps: (1) stating the hypothesis, (2) selecting a test statistic, (3) collecting data, and (4) interpreting the results.

**INTRODUCTION**

Dynamic pricing refers to a pricing strategy in which prices are modified in real-time in accordance with the condition of the market, consumer behavior, and various external factors. It enables businesses to maximize their profits and optimize their resources. This approach can be used in a variety of industries including retail, travel, hospitality, and entertainment.

This important tool is used by Uber and Lyft to optimize their business operations. Dynamic pricing allows companies to adjust the cost of their services based on current market conditions such as demand, supply, and competition. For example, during peak times (such as a rush hour or peak summer months) demand for rides increases, so dynamic pricing allows Uber and Lyft to increase their prices in order to meet the higher demand. This helps the companies to maximize their profits and ensure that drivers are adequately compensated for their time. Dynamic pricing also helps protect riders from excessive prices by ensuring that prices remain competitive, even during periods of high demand.

* What is the purpose of the dataset? What is the source of the data?

This project aims to provide a detailed exploratory analysis of the Uber and Lyft dataset. The purpose of our project is to find the variables that affect cab prices and to predict the price of cab rides based on the identified factors. We also want to comprehend some insights and visualizations. The project uses an open-source dataset that was obtained from Kaggle by the data owner, BM.

* What kind of data is included? Is it all text data, or is it numerical?

Both categorical and numerical data are present in the dataset.

* Character type – Day, Source, Destination, Cab type, Product ID, Name, Short and Long summary.
* Numerical type – Hour, Month, Price, Distance, Surge Multiplier, Latitude, Longitude, temperature, humidity, wind speed, etc.
* Describe the data fields including the title, the data type, the data description, etc
* The dataset includes information on the source and destination locations, the type of cab, the product id, the price, the distance, the date, the time, the humidity, the surge multiplier, etc.
* Data from November 26, 2018, to December 18, 2018, are included in the dataset.
* The title of the dataset is “Uber and Lyft Dataset Boston, MA”.
* How many rows are there in the data? The number of fields?
* The dataset has 693071 observations and 58 attributes.

Below are the data descriptions of each variable of the data that briefly describe the contents of the data set. The dataset's features are as follows:

***Rideshare dataset:***

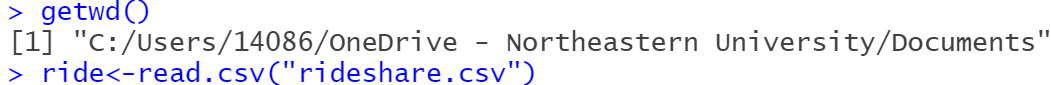
|  |  |  |
| --- | --- | --- |
| **No** | **Feature** | **Dictionary** |
| 1 | id | Unique Identification Number |
| 2 | timestamp | The specific point in time (seconds) |
| 3 | Hour | Hour of the day when the cab ride occurred |
| 4 | day | Day when the cab ride occurred |
| 5 | Day | Day of the week(Monday 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 timezone of the ride |
| 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

<ride> vector contains the details of all the trips in Boston between 26th Nov 2018 to 18th Dec 2018.



**Figure 1-read.csv()- credit**

* Cleaning the rideshare dataset
  + Dropping columns

All the unnecessary columns like UV index, visibility, sunrise, sunset time, etc were removed from the dataset. The number of columns dropped from 58 to 24.

Text

Description automatically generated

**Figure 2- names() function and ! operator**

* Visualization of <NA> values

gg\_miss\_which() creates a plot of the missing values in a data frame that helps quickly identify which rows or columns have the missing values.

On the other hand, gg\_miss\_var() creates a bar chart that displays the number of missing values in each variable of the data frame.

There are 55095 NA values in the price column and no null values.

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 3-<NA> values**

* Eliminating the <NA>s

The number of rows reduced from 693071 to 637976.

**Table

Description automatically generated**

**A picture containing text, receipt

Description automatically generated**

****

**Figure 4 – Obtaining the clean rideshare dataset free of "NA" values**

* Revising the column names

Renaming a column can help make the data easier to understand and work with.

Shape

Description automatically generated with medium confidence

**Figure 5**  **–names()**

* Deleting duplicate rows

We checked for identical rows in the dataset. The anyDuplicated() function in R returns the index of the first duplicate encountered. Using this, we can verify that the dataset contains some duplicate values. After removal, there are 634237 records.

Table

Description automatically generated

A picture containing text

Description automatically generated

**Figure 6- duplicated() and anyDuplicated()**

* Splitting the date and time into two different columns

We wanted the date and time columns to be separate in order to help us analyze better.



**Figure 7 – separate()**

* + Adding columns

Inserting a new column “price\_per\_distance” by dividing price by distance. Another column “price\_without\_surge” is calculated by dividing the price by the surge multiplier.

* Removing the outliers

Values that deviate from the pattern the rest of the dataset follows are known as outliers.

Boxplots are frequently used to identify outliers. We can eliminate the extreme outliers from the price column. In this case, we have only 1 extreme outlier.

Chart, box and whisker chart

Description automatically generated

**Figure 8- Boxplot for identifying Outliers**

* Replacing column values

Hours have been replaced by categories Morning, Afternoon, Evening, and Night using mutate and recode function.

1. 0-5 hours as Night
2. 6-11 hours as Morning
3. 12-17 hours as Afternoon
4. 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 data type of the columns is changed from one datatype to factor using as.factor().

**Text

Description automatically generated**

**Figure 9- as.factor()**

**DATA ANALYSIS**

* Analyzing the dataset

The clean dataset “ride1” consists of 634236 rows and 26 columns. It consists of data types - factors, characters, numbers, and integers.

Text, letter

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 10 – str() and summary()**

Calendar

Description automatically generated with low confidence

A picture containing text

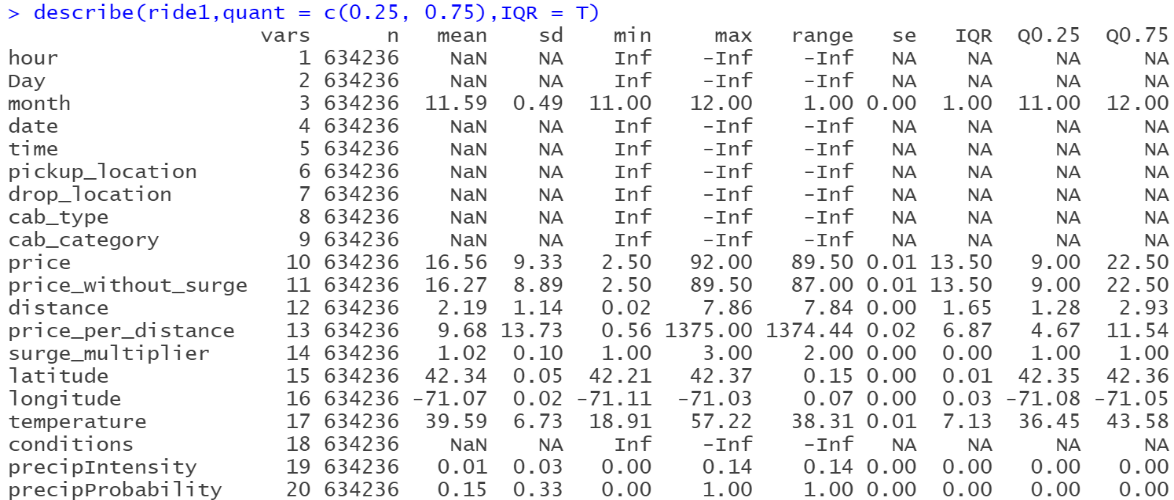
Description automatically generated

**Figure 11 – headTail() and dim()**

The mean of price is approximately 16.56, with a standard deviation of 9.33. The minimum and maximum values are 2.5 USD and 92 USD respectively.

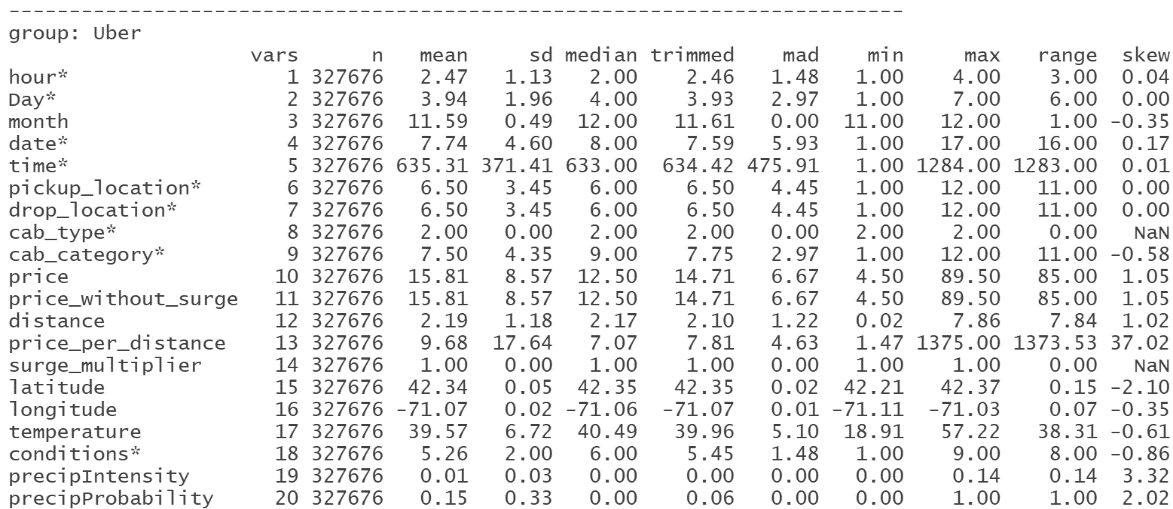
The mean distance is approximately 2.19, with a standard deviation of 1.14. The minimum and maximum values are 0.02 miles and 7.86 miles respectively.

describeBy() returns the statistical summary of grouped data. In this case, it is by Cab type (Uber and Lyft). There are 327676 records in the Uber category and 306560 in the Lyft category.



Table

Description automatically generated



**Figure 12 – describe() and describeBy()**

* **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?
* Which of the two charges a surge multiplier more frequently and what is the difference without the multiplier?
* Normality tests
  + Density plots

Chart, histogram

Description automatically generated

**Figure 13 – Density plots of features – Price, Distance, Surge multiplier and Price per distance**

The above density plot helps us understand the magnitude of the variable as per its usage.

The price range of the price variable is between 5-30.

Most people commute via Lyft or uber between 1-4 miles range.

The surge multiplier charge shows us the most commonly used multiplier.

The price per distance is a further deeper analysis of the price and distance variable and we can see that it confirms the visual from the price variable. It shows us the average price consumers pay per distance they travel.

* + Q-Q plots

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

**Figure 14 – QQplots of features – Price and Distance**

The data above for both price and distance variables appear to be curved and takes the shape of right-skewed data.

* + Box Plots

**Chart, box and whisker chart

Description automatically generated**

**Figure 15 – Boxplots – Price vs Cab Type**

The box plot above shows most of the prices charged for both Uber and Lyft are under the 25 range while the median for both cab types varies.

Chart, box and whisker chart

Description automatically generated

**Figure 16 – Boxplots – Distance vs Cab Type**

The box plot above shows the average distance covered by both the cab types lie between the 1-3 range while uber is shown to be selected more while traveling above the 6-mile range.

* Two-Sample T-test

**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 17 – Two sample t-test 1**

Considering the p-value is very low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, Both 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 price between Uber and Lyft is equal to 0.

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

Text, letter

Description automatically generated

**Figure 18 – Two sample t-test 2**

Considering the p-value is very low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, Both Uber and Lyft do not have the same mean price.

Uber- 15.81

Lyft- 17.36

**Test 3**

**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 19 – Two sample t-test 3**

Considering the p-value 0.02 low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, Both Uber and Lyft do not have the same mean distance covered.

Uber- 2.19

Lyft- 2.18

**Test 4**

**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 20 – Two sample t-test 4**

Considering the p value 0.99 is above the 0.05 level, we can confidently accept the null hypothesis. Therefore, Both uber and lyft have the same mean price per distance.

Uber- 9.6822

Lyft- 9.6819

**Test 5**

**Null Hypothesis H0:** The difference in means of the price without surge between Uber and Lyft is equal to 0.

**Alternate Hypothesis H1:** The difference in means of the price without surge between Uber and Lyft is not equal to 0.

Text

Description automatically generated

**Figure 21 – Two sample t-test 5**

Considering the p-value is low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, Both Uber and Lyft do not have the same mean price without surge.

Uber- 15.81

Lyft- 16.76

* One-Sample T-test

**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 22 – One sample t-test 1**

Considering the p-value is low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, 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 23 – One sample t-test 2**

Considering the p-value is low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, 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 24 – One sample t-test 3**

Considering the p-value is 1 and above the 0.05 level, we can confidently accept the null hypothesis. Therefore, the mean surge multiplier is less than 2.

**Test 4**

**Null Hypothesis H0:** True mean price per distance is greater than 5

**Alternate Hypothesis H1:** True mean price per distance is less than equal to 5

Text, letter

Description automatically generated

**Figure 25 – One sample t-test 4**

Considering the p-value is 1 and above the 0.05 level, we can confidently accept the null hypothesis. Therefore, the mean price per distance is greater than 5.

**Test 5**

**Null Hypothesis H0:** True mean temperature is equal to 45

**Alternate Hypothesis H1:** True mean temperature is not equal to 45

Text, letter

Description automatically generated

**Figure 26 – One sample t-test 5**

Considering the p-value is low and below the 0.05 level, we can confidently reject the null hypothesis. Therefore, the mean temperature is not equal to 45.

**CONCLUSION**

The hypothesis testing has helped us understand the relationship between the key variables and how they play a part in bringing out the similarities and differences between the two cab types: Uber and Lyft. We are able to find answers to some of the key questions we have posted about the data set with the help of the two sample T-test. The testing has played a vital role in understanding the characteristics of the target price variable along with the other dependent variables of the cab types, it proved to be a crucial step before entering into the final steps of the project with regression model and analysis.

Key Points :

* Uber charges a lesser surge multiplier compared to Lyft.
* Uber has a better coverage distance than Lyft; the box plot for the distance variable also shows Uber being preferred to Lyft for covering longer distances.
* The value for price per distance covered is similar for both Lyft and Uber.
* Uber is cheaper than Lyft while commuting during normal hours and not peak business hours.

**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

**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)