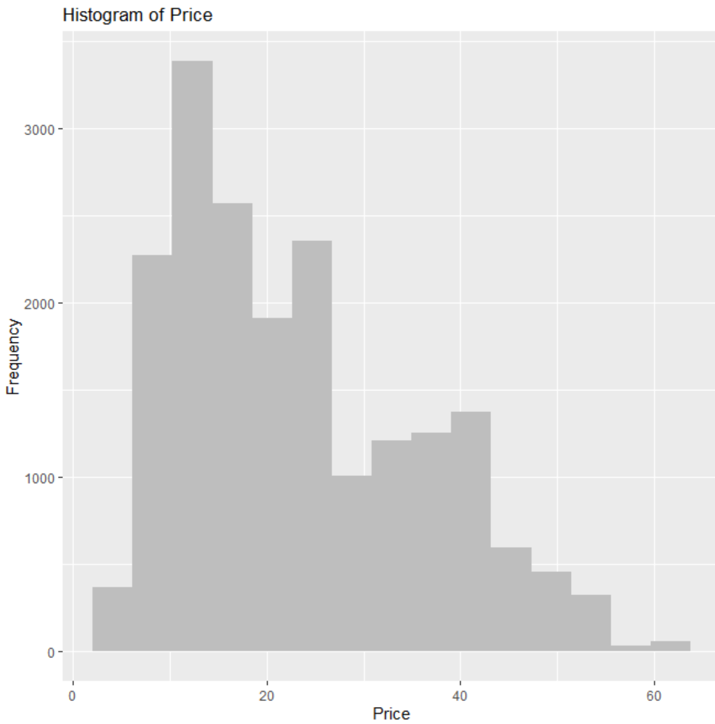
The case data file has 19,160 records representing hypothetical rideshares from Lyft and Uber in the Boston area. There is no missing value in each of the column, which is shown at below table.

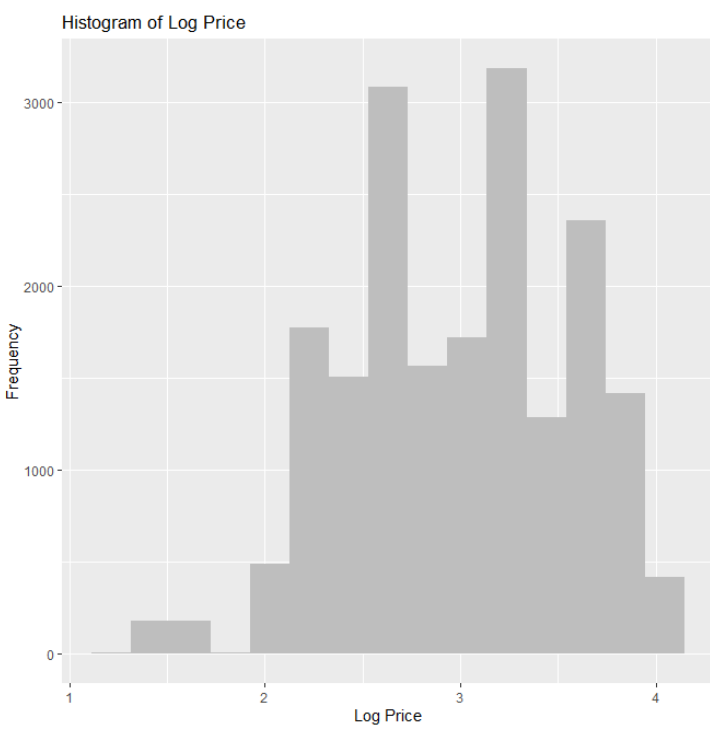
```
> colSums(is.na(mydata))
         id     date_time           hour          day        month        weekday     source
          0             0              0            0            0              0          0
destination     rideshare  ride_category        price     distance surge_multiplier   weather
          0             0              0            0            0              0          0
temperature precip_probability  humidity   wind_speed    wind_gust          ozone
          0             0              0            0            0              0
```

```
> summary(mydata)
      id              date_time                      hour            day          month      weekday        source
 Length:19160       Min.   :2020-08-01 04:57:00.00   Min.   : 0.00   Min.   : 1.00   8 :   71   Mon:1448   Length:19160
 Class :character   1st Qu.:2020-11-02 00:02:00.00   1st Qu.: 6.00   1st Qu.:11.00   9 :  300   Tue:1178   Class :character
 Mode  :character   Median :2020-11-15 00:27:30.00   Median :12.00   Median :15.00   10: 1133   Wed:3201   Mode  :character
                    Mean   :2020-11-13 19:00:03.99   Mean   :11.61   Mean   :16.19   11:17508   Thu:3387
                    3rd Qu.:2020-11-25 23:54:00.00   3rd Qu.:17.00   3rd Qu.:26.00   12:  148   Fri:2763
                    Max.   :2020-12-28 19:53:00.00   Max.   :23.00   Max.   :28.00              Sat:2433
                                                                                                Sun:4750
 destination         rideshare     ride_category      price          distance     surge_multiplier   weather             temperature
 Length:19160       Lyft:9284   Lyft      :6423   Min.   : 3.60   Min.   :0.024   Min.   :1.000   Length:19160       Min.   :20.31
 Class :character   Uber:9876   Black     :2693   1st Qu.:12.96   1st Qu.:1.500   1st Qu.:1.000   Class :character   1st Qu.:37.67
 Mode  :character               Black SUV:2681   Median :19.44   Median :2.484   Median :1.000   Mode  :character   Median :41.87
                                UberPool :1415   Mean   :23.20   Mean   :2.560   Mean   :1.013                      Mean   :40.87
                                WAV      :1408   3rd Qu.:32.40   3rd Qu.:3.432   3rd Qu.:1.000                      3rd Qu.:44.97
                                UberX    :1405   Max.   :61.20   Max.   :8.952   Max.   :2.500                      Max.   :58.62
                                (Other)  :3135
 precip_probability    humidity        wind_speed       wind_gust         ozone          hour_sin          hour_cos           month_day
 Min.   :0.0000     Min.   :0.3600   Min.   : 0.380   Min.   : 0.78   Min.   :267.3   Min.   :-1.0000   Min.   :-1.000000   11-1   :3285
 1st Qu.:0.0000     1st Qu.:0.6200   1st Qu.: 3.340   1st Qu.: 4.04   1st Qu.:289.0   1st Qu.:-0.7071   1st Qu.:-0.707107   11-26  :1961
 Median :0.0000     Median :0.6900   Median : 5.850   Median : 7.54   Median :306.3   Median : 0.0000   Median : 0.000000   11-25  :1959
 Mean   :0.1457     Mean   :0.7205   Mean   : 6.147   Mean   : 8.48   Mean   :311.9   Mean   :-0.0225   Mean   :-0.001048   11-27  :1576
 3rd Qu.:0.0000     3rd Qu.:0.8600   3rd Qu.: 8.340   3rd Qu.:11.96   3rd Qu.:330.1   3rd Qu.: 0.7071   3rd Qu.: 0.707107   11-14  :1142
 Max.   :1.0000     Max.   :0.9400   Max.   :14.930   Max.   :27.23   Max.   :376.8   Max.   : 1.0000   Max.   : 1.000000   11-15  :1131
                                                                                                                             (Other):8106
 month_day_freq                                    route       avgRoute_distance
 Min.   :   1   North Station-Fenway                 :  388   Min.   :0.5123
 1st Qu.:1023   North Station-North End              :  369   1st Qu.:1.4574
 Median :1142   Theatre District-Haymarket Square    :  331   Median :2.4561
 Mean   :1559   Theatre District-North End           :  316   Mean   :2.5602
 3rd Qu.:1961   Northeastern University-Theatre District:  311   3rd Qu.:3.4319
 Max.   :3285   Back Bay-Boston University           :  310   Max.   :6.0968
                (Other)                              :17135
```

For this study, our goal is to predict the price by time, type of ride, distance, weather. Therefore, we need to look at the distribution of the price. The price has mean $23.20, median19.44 and standard deviation 12.47. The following plot is the histogram of the price, and it shows that the price is skew to right and most instance's price are gathered between $5 and $25.



Since we are going to do linear regression so as the assumption of linear regression, response variable needs to be normal distributed. We will do linear transformation on log the price to make it more like a normal distribution as bell shape shows below. And applied in our model.

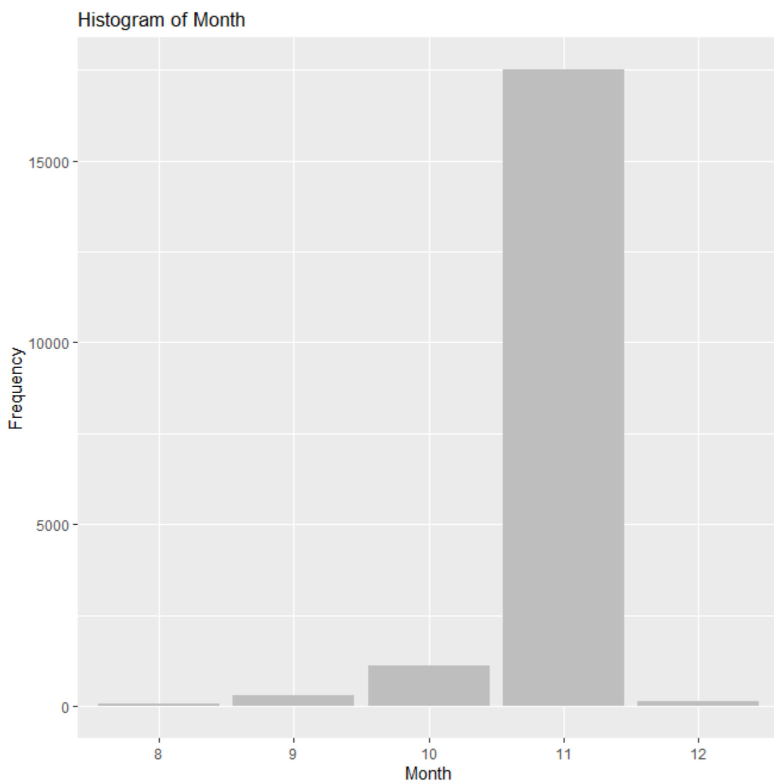


Let's break down to three different parts in our explanatory variables: time data, ride-type data, and other numerical data.

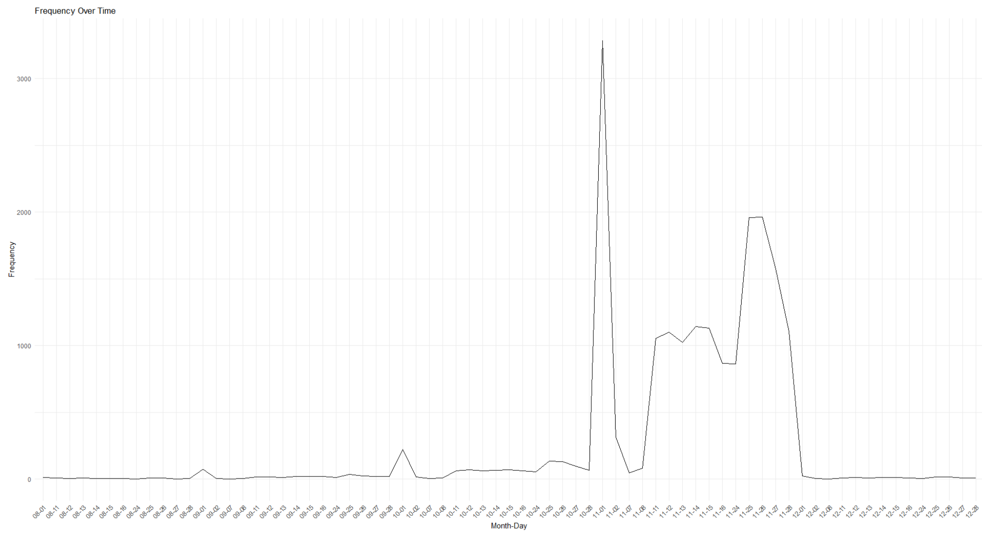
1. Time data:

In the data frame, we have hour, day, month, and weekday. We are going to interpret them separately and do the feature engineer to extract their value and use in our model.

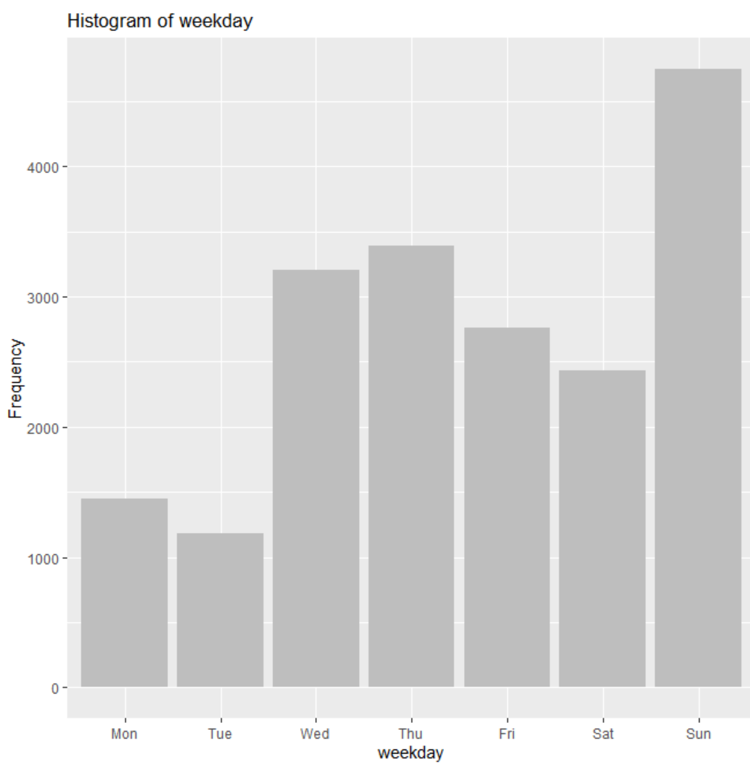
Following plot is the month frequence barplot, month data seems very unbalance since most instances are collect in November and less in August, September, October, and December. Therefore, we cannot identify month and day separately.



The method that I implement to solve this unbalance issues is to combine month and day together as "month-day." The coming issue of the month-day is that we create too many categorical data. Therefore, as the following time trend plot, we convert the categorical data "month-day" to frequency of each "month-day". Advantages are that we create the numerical data to represent each date and we also solve the issue of unbalance month data while doing train-test in model (this will explain more detail in next section.



For weekday, we discover that there are more users have ride on Sunday and less users on Monday and Tuesday, which is show as following graph.

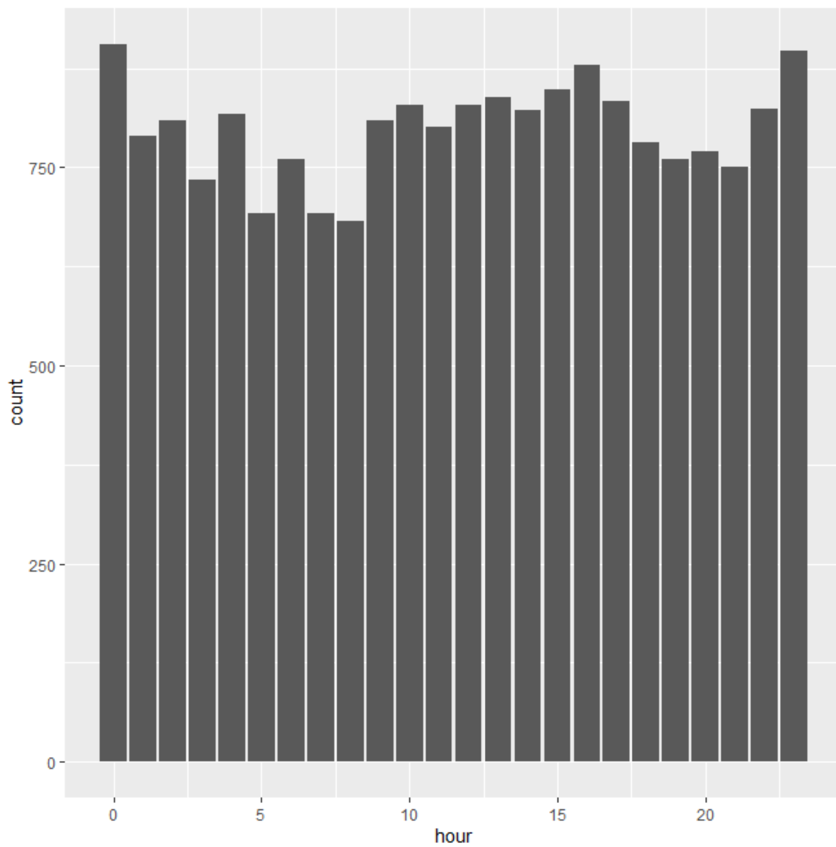


For hour, it is surprise that the hour distributed uniformly. And since hour has Cyclical Features so we transform hour into two features as following:

hour_sin = sin(2 * pi * hour / 24)

hour_cos = -cos(2 * pi * hour / 24)

which is the common method to use sin and -cos which are the cyclical transformation hour and minute.

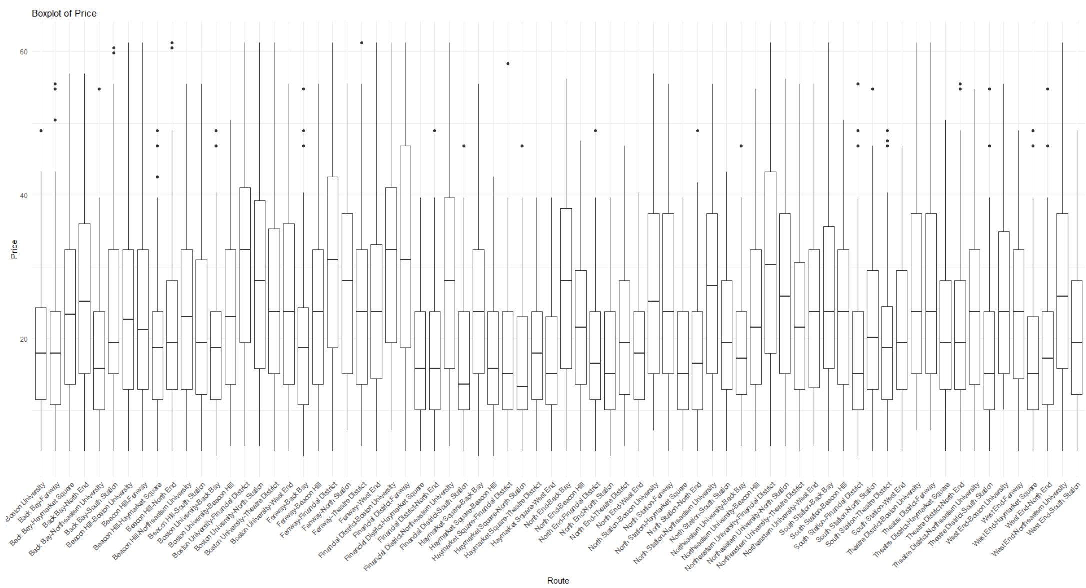


2. Travel route and distance:

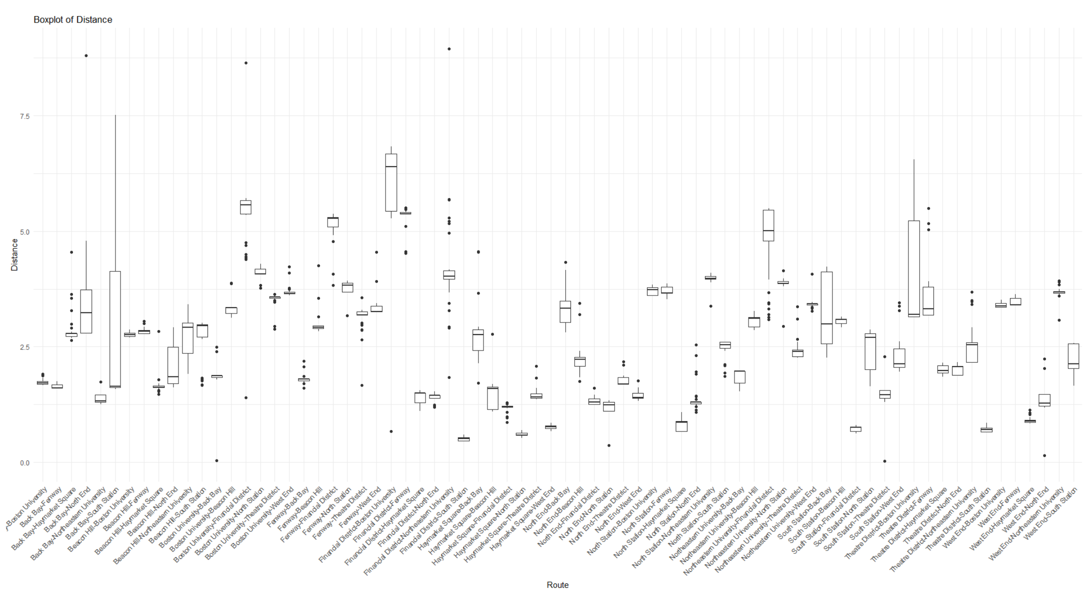
Following are the table are the unique pickup and drop off locations. We create the new variable: route which is the "pickup location – drop off location." The advantage of this feature is that this can be related with the distance variables, and it become more specific of each instance's route of trip.

```
> unique(mydata$source)
 [1] "Boston University"      "Haymarket Square"       "South Station"          "Fenway"                 "North End"
 [6] "Back Bay"               "North Station"          "Financial District"     "Beacon Hill"            "West End"
[11] "Theatre District"       "Northeastern University"
> unique(mydata$destination)
 [1] "Back Bay"               "Beacon Hill"            "Financial District"     "North Station"          "Northeastern University"
 [6] "West End"               "Fenway"                 "North End"              "Haymarket Square"       "Theatre District"
[11] "South Station"          "Boston University"
```

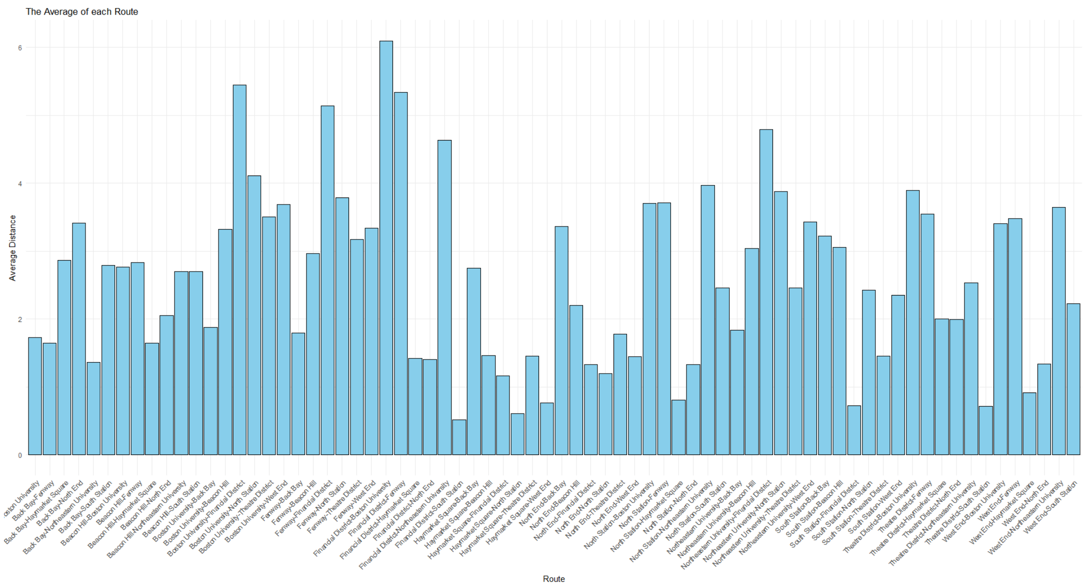
Following is the boxplot of price of each of the route. We can see that each route distributed differently, and it is not necessary to cluster them to small groups. But similar as the "month-day," route has too many categories.

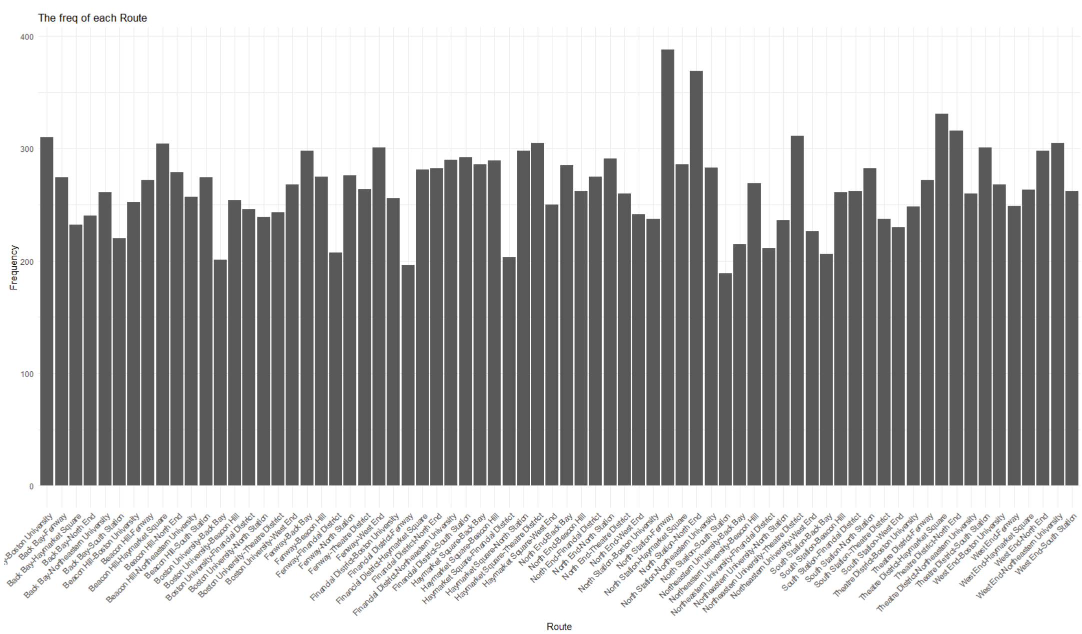


We transform the rout into the new variables, which is discover by the following boxplot.



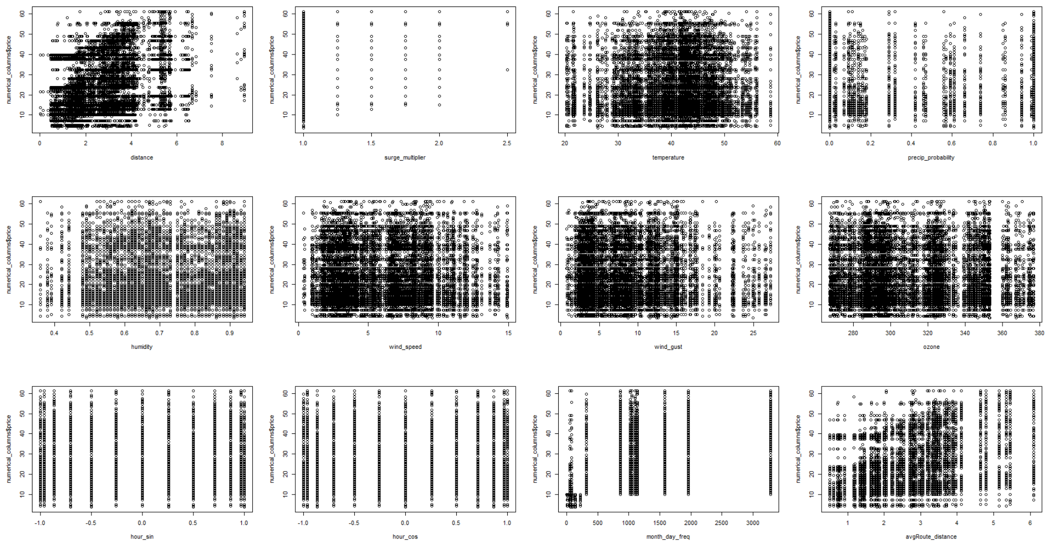
From the above boxplot we know each route has different mean, standard deviation, and frequency. We create mean and deviation distance of each route and the frequency of each route. Following is the graph of these:





3. Numerical Analysis:

Lastly, we have distance, surge multiplier and other weather information data. However, all of these numerical does not have obvious linear relationship with price which are show as below plots.



From those 12 graphs, there is not any quadratic or exponential relationship with price so that we can not do linear transformation to improve the linear relationship. Following table is the correlation table between features and price and it verify that they have week linear relationship to price.

	price
price	1.000000000
distance	0.307389258
surge_multiplier	0.145627307
temperature	0.018205198
precip_probability	0.008757253
humidity	0.008693709
wind_speed	0.008559312
wind_gust	0.007899999
ozone	-0.008297300
hour_sin	-0.004443729
hour_cos	0.004810034
month_day_freq	0.186669580
avgRoute_distance	0.303804758

We conduct the muti-linear regression model to answer questions to help the business in predicting what impacts prices. There are 4 steps for my model:

1) We implement the stepwise from both directions to do feature selection.
2) We Split the data with 80% training set and 20% testing set.
3) We conduct the 10-fold validation to resolve the overfitting problem with the explanatory variables that chosen from the first step.
4) We summarize our result by residual plot and compare the training and testing set by MAE/MAPE measure.

Feature Selection:

From EDA, we create new variables from original variables, and we also discovered that there are many uncorrelated variables with price. We already transform “month-day” and “route” into numerical data so that we won’t have too many coefficients for our model. But it is still necessary to implement the stepwise method to choose the right features and limit the number of features so that we can have more explanatory power for our analysis.

The following is the result of stepwise selected with response variable log(price):

log(price) ~ ride_category + distance + month_day_freq + weekday + surge_multiplier + ozone + weather + avgRoute_distance + temperature + rideshare + wind_gust + hour_cos + hour_sin + humidity

```
> stepboth$anova
                  Step  Df   Deviance Resid. Df Resid. Dev       AIC
1                       NA         NA     19159   5954.973 -22388.34
2      + ride_category -13 607.476050     19146   5347.497 -24423.92
3           + distance  -1 475.805837     19145   4871.691 -26207.39
4     + month_day_freq  -1 210.867602     19144   4660.823 -27053.20
5            + weekday  -6  91.474997     19138   4569.348 -27420.98
6   + surge_multiplier  -1  77.074038     19137   4492.274 -27744.92
7              + ozone  -1  12.931036     19136   4479.343 -27798.15
8            + weather  -6  15.807583     19130   4463.536 -27853.89
9  + avgRoute_distance.x -1   9.353562     19129   4454.182 -27892.08
10       + temperature  -1   8.674608     19128   4445.507 -27927.43
11         + rideshare  -1   8.304081     19127   4437.203 -27961.26
12         + wind_gust  -1   5.446108     19126   4431.757 -27982.79
13          + hour_cos  -1   1.302253     19125   4430.455 -27986.42
14          + hour_sin  -1   0.984350     19124   4429.471 -27988.68
15          + humidity  -1   1.428758     19123   4428.042 -27992.86
```

We have 18 explanatory variables originally and it reduces to 14 variables which are shown above table.

precip_probability

numRoute

sdRoute_distance

wind_speed

There are the 4 features that be removed by stepwise AIC test. We will implement the rest features to our multi-linear regression model.

Splitting training and testing:

This is necessary to split data so that our model have ability to predict the future values. From EDA part, we mentioned that unbalance of the month, this is essential to convert the "month-day" variable into numerical if we split the data since we may have no instances of other months than November in test set. Route of trips also have similar concern.

10-fold validation training:

To avoid the overfitting, we conduct the 10-fold validation training with the model that has:

Response variable: log(price)

Explanatory variables: ride_category, distance,month_day_freq, weekday, surge_multiplier, ozone, weather, avgRoute_distance, Temperature, rideshare, wind_gust, hour_cos, hour_sin, humidity

For each of the numerical data, we implement the standard scale before the cv regression to normalize them, so we have better performance for the result, equalizing the scale and it is handling the outlier well.

The standard scale's formula is:

(Xi-X_mean) / sd(X)

The following result are our final model that maximize the company's revenues and shows which features impact the price the most.

The table below is the summary of the final model. We can compare the coefficient of each feature since we have standardized.

1. Uber has less price than Lyft.
2. Among the rides category, WAV is the most expensive since it has largest coefficient among the rides category. Other than WAV, taxi, uber pool,

UberX, UberXL, and Uber, and Black SUV also have higher price. On the over side, Ride Shared has lowest price.

3. Saturday most likely are the most expensive weekday and Sunday are the cheapest weekday. We can also compare the frequency plot of each weekday in the EDA. We find out that Sunday has the most usages, but the lowest price and Saturday has relatively low usages but the highest price.
4. Fog and cloud night are the two weathers with the lowest price and raining day has the highest price. And for temperature, higher temperature tends to have lower price.
5. Distance and month_day_freq are the two most significant numerical feature that impact the price. This means that the distance of travelling, and the day has large demand of ride can affect the price at most. On the other hand, temperature, humidity, wind do not really affect that much as a numerical feature.

```
Residuals:
    Min      1Q  Median      3Q     Max 
-1.5743 -0.4073 -0.0252  0.3670  1.9550 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  3.1946936  0.0297482 107.391  < 2e-16 ***
`ride_categoryBlack SUV`    -0.0009966  0.0147552  -0.068 0.946148    
ride_categoryLux            -0.9841319  0.0661745 -14.872  < 2e-16 ***
`ride_categoryLux Black`    -1.0145283  0.0661380 -15.340  < 2e-16 ***
`ride_categoryLux Black XL` -0.9540604  0.0692123 -13.785  < 2e-16 ***
ride_categoryLyft           -0.0035819  0.0146197  -0.245 0.806457    
`ride_categoryLyft XL`      -0.9765180  0.0697806 -13.994  < 2e-16 ***
ride_categoryshared         -1.4843187  0.3409103  -4.354 1.35e-05 ***
ride_categoryShared         -1.0824508  0.0750801 -14.417  < 2e-16 ***
ride_categoryTaxi           -0.0009970  0.0190098  -0.052 0.958175    
ride_categoryUberPool       -0.0003092  0.0190117  -0.016 0.987023    
ride_categoryUberX          -0.0092246  0.0191168  -0.483 0.629431    
ride_categoryUberXL         -0.0052422  0.0191286  -0.274 0.784049    
ride_categoryWAV             0.0063044  0.0190505   0.331 0.740701    
distance                     0.0933396  0.0119379   7.819 5.68e-15 ***
month_day_freq               0.1959597  0.0059299  33.046  < 2e-16 ***
weekdayTue                  -0.0270829  0.0255739  -1.059 0.289615    
weekdayWed                  -0.0864907  0.0211132  -4.097 4.22e-05 ***
weekdayThu                  -0.1054903  0.0198042  -5.327 1.01e-07 ***
weekdayFri                   0.0298772  0.0195101   1.531 0.125700    
weekdaySat                   0.0503555  0.0217597   2.314 0.020672 *  
weekdaySun                  -0.2421997  0.0233591 -10.369  < 2e-16 ***
surge_multiplier             0.0602497  0.0039026  15.438  < 2e-16 ***
ozone                        0.0467682  0.0061569   7.596 3.23e-14 ***
`weatherclear-night`        -0.0292078  0.0275740  -1.059 0.289501    
weathercloudy               -0.0459627  0.0243229  -1.890 0.058819 .  
weatherfog                  -0.1172191  0.0435911  -2.689 0.007173 ** 
`weatherpartly-cloudy-day`  -0.0844445  0.0236991  -3.563 0.000367 ***
`weatherpartly-cloudy-night` -0.1002966  0.0246138  -4.075 4.63e-05 ***
weatherrain                  0.0003602  0.0289956   0.012 0.990089    
avgRoute_distance            0.0575933  0.0119009   4.839 1.31e-06 ***
temperature                 -0.0335145  0.0058989  -5.681 1.36e-08 ***
rideshareUber               -0.0759343  0.0148644  -5.108 3.29e-07 ***
wind_gust                   -0.0230848  0.0057788  -3.995 6.51e-05 ***
hour_cos                     0.0094612  0.0047605   1.987 0.046891 *  
hour_sin                    -0.0121493  0.0057046  -2.130 0.033209 *  
humidity                     0.0113553  0.0074310   1.528 0.126511    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

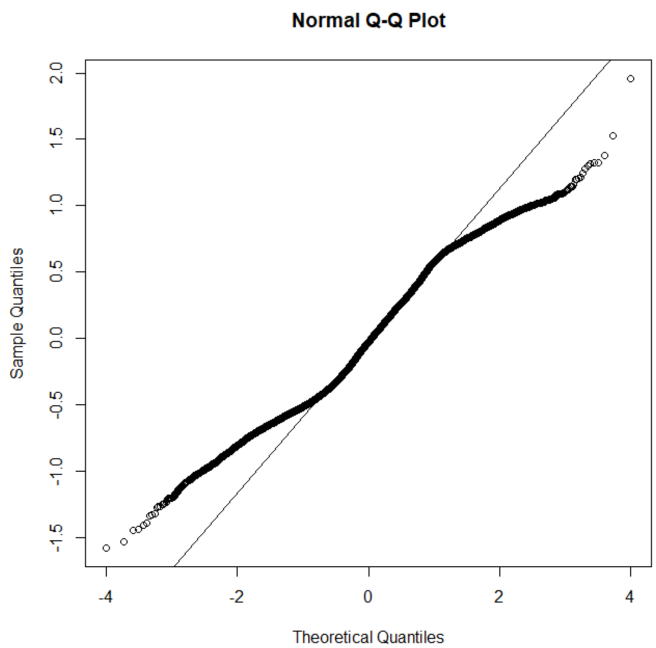
From the table below, our Residual standard error is equal to 0.4813, which mean the observed values deviate from the predicted values by approximately 0.4813% (the unit is in percentage since we use log transformation on price. The adjusted R-squared is equal to 0.2559 and it is low since there are no feature is highly linear correlated with price (verified with the scatter plots and correlation table in EDA)

```
Residual standard error: 0.4813 on 15291 degrees of freedom
Multiple R-squared:  0.2577,	Adjusted R-squared:  0.2559 
F-statistic: 147.4 on 36 and 15291 DF,  p-value: < 2.2e-16
```

Next is the 10-fold result that shows the RMSE, R_Squared, and MAE for each fold. This sures that we have sufficient data point so each have fold has similar error; consequently, our model is really robust that it can use to predict the future value with same error.

```
> modelfold$resample
        RMSE  Rsquared       MAE Resample
1  0.4827174 0.2597988 0.4091249   Fold01
2  0.4841912 0.2679893 0.4077940   Fold02
3  0.4752826 0.2899666 0.4027919   Fold03
4  0.4866425 0.2198234 0.4114463   Fold04
5  0.4782918 0.2788276 0.4063161   Fold05
6  0.4851265 0.2169502 0.4103517   Fold06
7  0.4807879 0.2479254 0.4073850   Fold07
8  0.4824535 0.2582119 0.4124445   Fold08
9  0.4810393 0.2588347 0.4054912   Fold09
10 0.4824225 0.2467682 0.4106705   Fold10
```

The QQ-plot shown below is expected since the R_Squared is low so we may not have normal distributed residuals. From the QQ-plot, we can see that both end tails do not stick on the line, because we have more residuals at both the side which means that most of the data points are far away from the regression line. We may need to conduct more complex model (like polynomial regression or any other Machine learning model to train better).



From the train-test test, our model does not overfitting by there is no huge difference between the train MAE, MAPE and test MAE, MAPE since we have conducted the 10-fold to receive the robustness model. As mentioned, since we do not have overfitting and our model's performance still has some space to improve, we may try more complex model in the future to get better model.

```
> mae(trainset$price, exp(pred))
[1] 9.087572
> mape(trainset$price, exp(pred))
[1] 0.4261101
```

```
> mae(testset$price, exp(pred))
[1] 9.113972
> mape(testset$price, exp(pred))
[1] 0.4232503
>
```