Lyft and Uber Ride Price Prediction via Multiple Linear Regression

Jingcheng Chu, Qingyang Fu, Yaoxin Liu

MDA9159: Statistical Modeling

December 2, 2022

Jingcheng Chu	jchu268@uwo.ca	251004521
Qingyang Fu	qfu32@uwo.ca	250963843
Yaoxin Liu	yliu4685@uwo.ca	251324891

Contents

1 Introduction

As a group, we wish to investigate a real-life problem in more depth which can provide more insights. One of the most common ways of transportation nowadays is online ride-hailing. As regular users of Uber and Lyft, two of the most popular platforms for rides, we wish to gain a further understanding of the algorithms for predicting the price of rides.

The purpose of our study is to predict the price of each ride based on given conditions such as time, type of vehicle selected, and weather indexes. In this project, the dataset we have selected contains more than 10000 records of rides. Then, we decided to reduce our dataset into a randomly selected subset with 4042 observations. Within the dataset, we have one target variable, *price*, and 58 predictors.

Upon examining the dataset, we have an initial assumption of which of the predictor variables are most likely to be significant, which includes time, distance, temperature, and cab type. We will investigate this hypothesis by conducting further analysis utilizing different models.

We choose to construct multivariate linear models for this dataset due to the model's ability to explain the relationship between one response variable and many predictor variables[1]. As figure 1 shows that *distance* has a linear relationship with the target variable *price*. Multilinear models may also have certain limitations, which we will disclose and discuss in the Limitation section of our report.

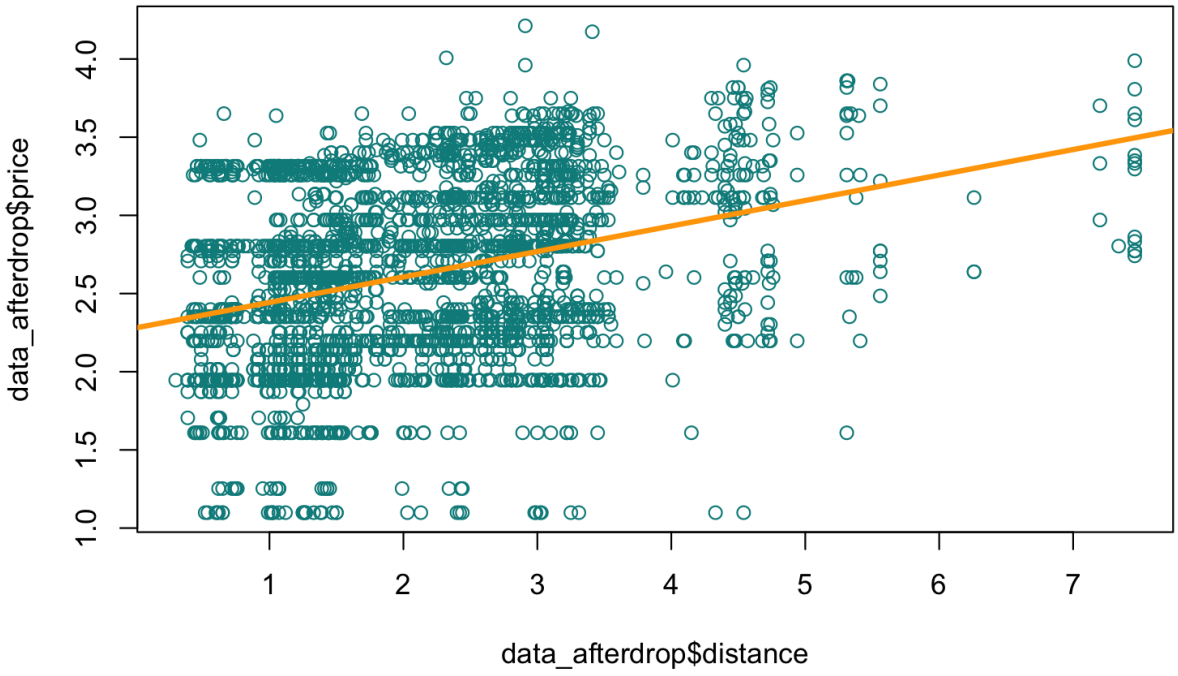


Figure 1: Relationship between price and distance

This study aims to construct the optimal multivariate linear regression model to predict each ride's price, and we build our models from different selection algorithms. Forward, backward, and bidirectional selection algorithms will be utilized to generate several models for comparisons. Within the selection algorithms, we also utilized various measurements to select models, such as AIC and BIC.

After building the models, we choose our best and final model based on the lowest RMSE from cross-validation and the highest score of adjusted R^2 to avoid multicollinearity. By constructing and validating our best and final model, we will give a thorough interpretation of predictor variables and address any possible limitations or drawbacks that our final model may have. As achieving the ability to predict the price of each ride is our goal, the prediction power of our final model will also be assessed.

2 Dataset and Pre-processing

The raw dataset contained over 10000 instances of rides in the United States from both Uber and Lyft. There are 58 variables in this dataset, such as *price*, *distance*, *surge multipler*, and many other ones describing weather and temperature, which are all related to ride shares. We decide to randomly select a subset containing 4042 observations to perform our analysis. To improve the efficiency and potential accuracy of the classifier, data cleaning is an essential step. Handling missing values, predictor selection, handling categorical variables, response transformation will be evaluated and discussed in the following sections.

2.1 Handling Missing Value

Upon checking the missing values from the dataset, we found 239 Nah values, which all located in the column of the target variable, *price*. After dropping the observations which contained Nah values, we reduced our dataset to 3803 instances.

2.2 Reducing Predictors

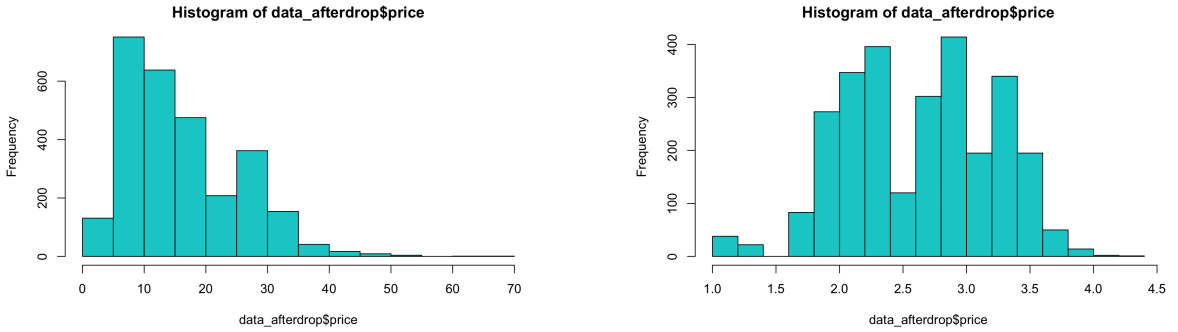
After examining each predictor, we discovered that there were a significant number of redundant predictors, such as *short summary*, *precipProbability*, *winGustTime*, etc. We removed these repetitious predictors, reducing the number of predictor variables from 58 to 36.

2.3 Handling Categorical Variables

Within the 36 remaining predictor variables, there are two categorical variables, *cab_type* and *name*, which indicate the platform of choice and the vehicle type, respectively. To perform regression on these two categorical variables, we need to convert them to factors. After applying `as.factor()`, we successfully converted them from `<char>` to `<fatr>`, after which we can continue to generate our models.

2.4 Response Variable

After examining our response variable, *price*, by plotting the histogram, the graph shows that the *price* is skewed to the left; therefore, we decided to perform the log transformation to the response variable. As a result, the *price* is distributed normally, as shown in figure 2.

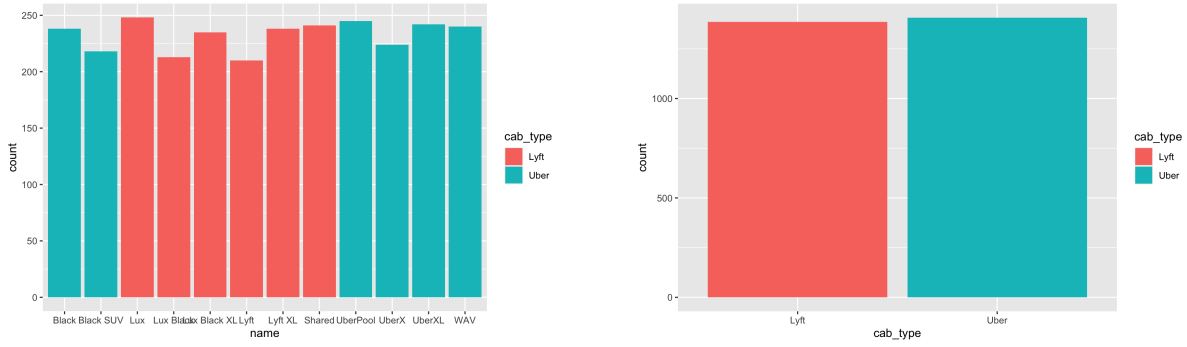


(a) Distribution before Transformation (b) Distribution after Transformation

Figure 2: Distribution of *price* before and after Transformation

2.5 Balance of the Dataset

We also examine the balance of the data in two aspects, the distribution of the platform of choice and the vehicle types. As shown in figure 3, the data are balanced between the two platforms. The total number of rides for Uber is around 1400, and 1398 for Lyft. As for different types of vehicle selection, e.g. (Uber XL, Uber Green, and Lyft XL), the number of rides also seems to be balanced, by examining figure 3. Therefore, our model will have less bias regarding platform and vehicle selection since we have an overall balanced dataset.



(a) Balance of Predictor *name* (b) Balance of Predictor *cab_type*

Figure 3: Balance of Predictors *name* and *cab_type*

3 Methodology

3.1 Full Model Analysis

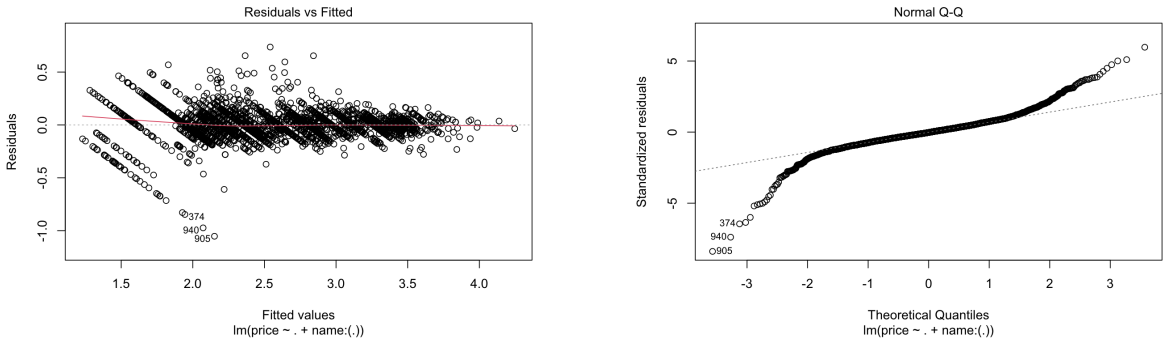
Using the data we have cleaned and analyzed in the previous step, we choose to first fit the full model with all of the predictors. From the summary of the full model (Appendix B figure 6), predictors like *cab_type* (Uber, Lyft), *name* (Lux, XL, and X), *distance*, and *surge_multiplier* are most significant, as shown their p-value is smaller than $\alpha = 0.05$. This aligns with our initial speculation of what influences the response variable price the most.

Other predictors are less significant, but their p-value is still smaller than $\alpha = 0.05$. These predictors are mostly weather-related, such as *temperatureMax*, *apparentTemperatureMax*, *apparentTemperatureLowTime*, *apparentTemperatureHigh*, and *precipProbability*.

We also take interaction terms into consideration given that there is chances the predictors have different impact on Uber and Lyft. We have included all interaction terms between *name* and all remaining predictors in our model. After which, we conduct an F-test, with the null hypothesis of all interaction terms being insignificant. As a result, the p-value is 0.0001 which enables us to reject the null hypothesis. So at least one of the interaction terms is significant. Hence, we conduct our further model and variable selection based on the full model with interaction terms.

3.2 Model Assumptions

Fitting our full model with interaction, we check if all model assumptions hold. From the residual vs Fitted and Normal Q-Q plot(figure 4), the equal variance, linearity and normality assumptions are all violated. The results of BP-test and Shapiro-Wilk test are both close to 0 which also support our graphical findings.



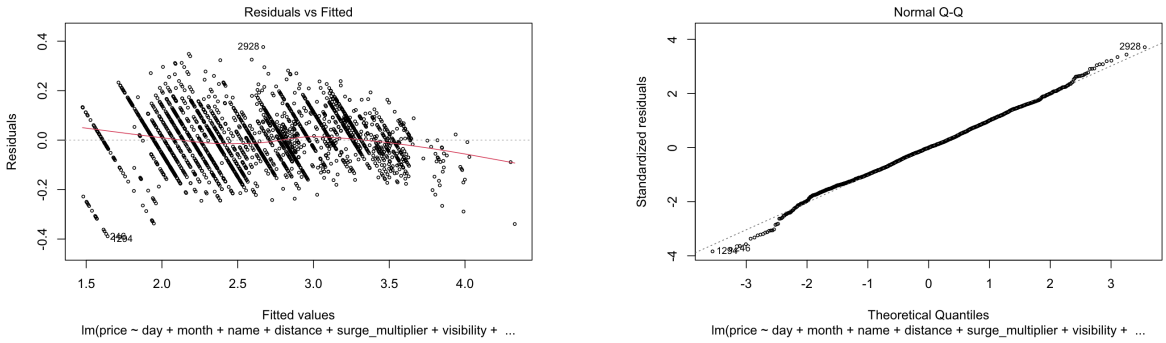
(a) Residual V.S. Fitted Values

(b) Normal Q-Q Plot

Figure 4: Model Assumption Check

As model assumptions for all three of our models are all violated, we choose to review and eliminate the influential points from our dataset, utilizing Cook's constant. Fitting our models again after removing the influential points, we discover some minor improvements

regarding the model assumptions, especially on the normality of the error term; as shown in figure 5, the departure of extreme values is significantly reduced.



(a) Residual V.S. Fitted Values

(b) Normal Q-Q Plot

Figure 5: Model Assumption Check

Running BP-test and Shapiro-Wilk normality tests again, the p-value of BP-test is still close to 0, however, Shapiro-Wilk test shows considerable improvement (increased from 0 to 8.104e-05).

3.3 Stepwise Regression

Assuring the significance of each predictor, stepwise regression can help us select significant predictors using different directions of the selection algorithm, such as forward selection, backward selection and bidirectional elimination. We use all three approaches with two different measures, AIC and BIC. As AIC and BIC have different penalty terms, selection algorithms will generate distinct models. BIC, compared to AIC, will penalize the model complexity harder when many predictors are present.

3.4 Model Comparison

After comparing the backward, forward, and bidirectional stepwise selection using AIC, ***Fit_Both_AIC*** which is the model after bidirectional selection with AIC is the better model, as it has the lowest AIC and the highest adjusted R^2 (0.9432). Similarly, for using BIC as a penalty, we choose ***Fit_Both_BIC*** as the optimal one with adjusted R^2 0.9430 while the other two using forward and backward stepwise selection. The summary could be found in table 1 and table 2.

Before comparing ***Fit_Both_AIC*** with ***Fit_Both_BIC***, we want to investigate if collinearity is present in each model. Upon examining the VIF, variance inflation factor, we found significant multicollinearity in ***Fit_Both_AIC***, as well as in the ***Fit_Both_BIC*** model, since both models' coefficients have VIF above 5, meaning that there is collinearity present between predictors. Meanwhile, there various coefficients showing VIF exceeding 10 in ***Fit_Both_AIC*** (Appendix B figure 7), whereas no VIF is above 10 in ***Fit_Both_BIC*** (Appendix B figure 8). From the multicollinearity standpoint, we prefer the ***Fit_Both_BIC***.

Table 1: Model Selection with BIC Summary

Model	**BIC**	**Adjusted R^2**
Fit_Back_BIC	-3001.5415	0.943272
Fit_For_BIC	-518.6882	0.9414452
Fit_Both_BIC	-3001.5415	0.943272

Table 2: Model Selection with AIC Summary

Model	**AIC**	**Adjusted R^2**
Fit_Back_AIC	-3158.853	0.9430293
Fit_For_AIC	-2756	0.9414452
Fit_Both_AIC	-3158.853	0.9430293

Given the fact that the interaction terms are the product of two predictors, so a relatively large VIF is reasonable, which should not lead to a biased prediction.

For comparing the goodness-of-fit between ***Fit_Both_AIC*** with ***Fit_Both_BIC***, we utilize adjusted R^2, as this measure also takes account of collinearity. Surprisingly, the adjusted R^2 of all above 0.94 which is impressive. Besides, they are very close to each other, so it is difficult to pick one model only based on adjusted R^2. In this situation, a simpler model would be preferred. As a result, ***Fit_Both_BIC*** is also preferable to ***Fit_Both_AIC***, as it has a slightly higher adjusted R^2 achieved with less predictors.

We also performed cross-validation on our final model to examine its predictive power. Root Mean Squared Error (RMSE) is calculated to compare the forecasting errors of different models for training and testing datasets. Both ***Fit_Both_AIC*** and ***Fit_Both_BIC*** have similar RMSE (0.09537 and 0.09543); hence, strictly based on RMSE, ***Fit_Both_BIC*** has higher prediction power.

4 Results

4.1 Final Model

After comparison, our best and final model is the ***Fit_Both_BIC***. This model contains 24 predictors, *name*, *distance*, and *surge multiplier*, along with the interaction term between *name* and *distance*.

$$price \sim name + distance + surge\_multiplier + name \times distance$$

name indicates which type of vehicle the user selected. Luxurious or larger-sized vehicles have a higher pricing strategy than shared rides and regular vehicle models. The coefficients of predictor *UberPool* and *UberXL* is -0.767 and -0.277, meaning that *UberPool* has a greater effect on *price* than *UberXL*. Considering the coefficients are negative, when other predictors

are fixed, the price with category *UberPool* will be lower than that with *UberXL*. In reality, the price of UberPool ride is actually cheaper than UberXL assuming other factors remain the same. *distance* is the most self-explanatory attribute; the longer the trip, the higher the fare. The positive 0.1725 coefficient of *distance* precisely reflects this relationship. However, the level of impact of *distance* is lower than excepted. The potential reason is that the the effect of distance is relatively stable and consistent, and the type of vehicle has more significant impact on price. *Surge multiplier* indicates a multiplier of the price of each ride based on the increasing demand for rides; the higher the demand, the higher the multiplier. Lastly, the interaction terms between *name* and *distance* are also significant, as different types of vehicles may have different pricing strategies per unit distance traveled; hence, such interaction can heavily influence the prediction of the target variable, *price*. Overall, this model and the coefficients of predictors are performing in the logical sense. The summary of this model can be found in the Appendix B figure 10.

Reassessing our initial assumption regarding important predictors, many of our assumed key attributes are missing, such as time and weather indexes. Therefore, we suspect that the *surge multiplier* might reflect other vital attributes, which will be validated in the later sections.

4.2 Prediction

Performing prediction, we have split our data into a training set containing 3031 instances, and a testing set containing 1011 instances. After fitting our best model with training data, ***Fit_back_BIC***, we calculated two MSE measures for both training and testing set for comparison. The MSE of the training set was 0.0185587, which is similar to the MSE of our testing set, 0.02022274. Since the MSE of the training and testing set are close, we believe that our model does not have the situation of overfiting and has strong prediction power.

4.3 Further Analysis on Surge Multiplier

We are curious about how temperature and time factors become insignificant. Therefore, we found that the surge multiplier might be an indicator that reflects other vital factors.

We then fit another linear model, using the *surge multiplier* as the target variable and the rest as predictor variables, excluding *price*. The result coincides with our assumption that *percip Intensity*, *precip Probability*, *Apparent Temperature Min Time*, and *Apparent Temperature High time* showing a significant effect on the response *surge multiplier* (Appendix B figure 9). Besides, we also discover that for Lyft, their *surge multiplier* also depend on the cabin type, e.g. Lyft Lux, Black, and XL, whereas Uber does not.

However, the adjusted R^2 is not high, only 0.038. Therefore, to confirm that the *surge multiplier* is well explained by the predictors which were significant, we decide to fit again with only significant predictors.

As figure **??** shows that the R^2 is still around only 0.37, which indicates that the *surge multiplier* has a non-linearly relationship with the *name* (Uber XL, Lux), *Temperature*, *Precip Intensity*, and *Precip Probability*. There might be a polynomial or other relationship

among them, and it will be beyond the scope of our project. Hence, *surge multiplier* does not linearly reflect other predictor variables as we assumed.

5 Limitations

Limitations of our study lie in several aspects of our analysis. Firstly, the data we collected is only from one specific month; the location was also limited to the United States. Since it is difficult to use a single region to represent the entire global market, as different areas or continents might have utterly opposite ride habits, our final model's prediction power may not be able to generalize.

Secondly, the model assumptions for our full, reduced, or optimal models are all violated. "In real-life examples (as opposed to computer simulations), the linearity, homoskedasticity and normality assumptions are pretty much guaranteed to be violated" [2]. Simply said, there are many more possibilities for a relationship to be nonlinear than linear. Also, It must be extremely unlikely that the distribution of outcome values having a different mean but the exact same standard deviation at different predictor values. Meanwhile, the normal distribution range is from negative to positive infinity and can take any real value. In contrast, real-life data sets have theoretical or practical minimum and maximum values and don't have infinite precision.

Moreover, there might be predictors among the 58 variables which do not have a linear relationship with our response variable price. Although multivariate linear regression is one of the best options to explain the relationship between the independent and dependent variables in real life, polynomials or higher powers of regression might be a better choice.

The limitation also lies inside the objective of our study; we are trying to mimic the well-developed price algorithm, which Uber and Lyft engineers have already constructed. Before choosing which predictor has the most significant effect on ride price, the response variable itself has already been well-explained by all those 58 predictors. We are doing a reverse engineering process for the ride price, which is why our full model, before we use backward, forward selection, already has an adjusted R^2 of 0.935, where the goodness-of-fit is nearly perfect. Even when we have done all those processes to choose the optimal model, the adjusted R^2 only increases to around 0.943.

6 Conclusion

In this project, raw data was collected and processed. Stepwise model selection technique and cross-validation were utilized to select the predictors of the final optimal model, where critical measures like AIC, BIC and VIF are being closely examined. The most influential predictors chosen for price prediction are *name*, *distance*, *surge multiplier*, and the iteraction term between *name* and other predictors, which corresponds to the initial expectation. The basic architecture and processing steps of data analysis and price prediction were investigated. The result of the final model shows an adjusted R^2 score of 0.943, which means it explains the data very well, and multivariate linear regression is sufficient to predict the ride price

in the United States region. However, as the Limitation section expressed, the dataset can only represent the situation of the specific month and the US only, so the final model is not generic enough to predict either in another country or another month of the year. A larger dataset consisting of the prices in different months and countries is needed to conduct further analysis and develop a more generic model.

References

[1] Michelle Sergent, Didier Mathieu, Roger Phan-Tan-Luu, and Giuliana Drava. Correct and incorrect use of multilinear regression. *Chemometrics and Intelligent Laboratory Systems*, 27(2):153–162, 1995.

[2] Jan Vanhove. Before worrying about model assumptions, think about model relevance.

Appendix A Code

```r
#load library
library(ggplot2)
library(lmtest)
library(glmnet)
library(MASS)
library(faraway)

################## Data Pre processing ####################
#Load Data
data_v1 = read.csv("reduced_data.csv")
head(data_v1)
# missing value: 239 Missing Value in price
sum(is.na(data_v1))
# drop NAH
data_nomissing = na.omit(data_v1)
#check omit work or not
sum(is.na(data_nomissing))

# Drop Columns
drop <- c("id","datetime" , "timezone" , "timestamp", "source" ,"
    destination" , "short_summary" , "latitude" , "longitude", "product_id"
     , "short_summary", "long_summary", "humidity" , "windGust" , "
    windGustTime" , "icon", "dewPoint" , "pressure" , "windBearing" , "
    cloudCover" , "ozone" , "moonPhase")
data_afterdrop = data_nomissing[,!(names(data_nomissing) %in% drop)]
head(data_afterdrop)

# to factors
data_afterdrop$cab_type <- as.factor(data_afterdrop$cab_type)
data_afterdrop$name <- as.factor(data_afterdrop$name)
head(data_afterdrop)

#Check the response value distribution
hist(data_afterdrop$price, col = "cyan3")

#Transform data
data_afterdrop$price = log(data_afterdrop$price)

#histogram after transformation
hist(data_afterdrop$price, col = "cyan3")

#balanced of the Lyft and Uber
ggplot(data = data_afterdrop, aes(cab_type,fill = cab_type)) + geom_bar()
ggplot(data = data_afterdrop, aes(name, fill = cab_type)) + geom_bar()


#Relaitonship between distance and price, and its correlation
m1=lm(price~distance,data = data_afterdrop)
cor(data_afterdrop$distance, data_afterdrop$price)
```

```r
plot(data_afterdrop$distance,data_afterdrop$price,pch=1,col="cyan4")
abline(coef=m1$coefficients,c="orange",lwd=3)

###################### Models ##########################

#full model regression
lm_full_v1 = lm(price ~. , data = data_afterdrop)
summary(lm_full_v1)

#Cab_type Interaction
lm_full = lm(price ~. + name:(.), data = data_afterdrop)
summary(lm_full)

# with interaction, the R2 become larger, from 0.93 to 0.94.
anova(lm_full_v1, lm_full)

#Surge multiplier
lm_multiplier = lm( surge_multiplier ~. -price ,data = data_afterdrop)
summary(lm_multiplier)

#Without those non_significant predictors, R2 performance
lm_multiplier_without_nonsig =  lm(surge_multiplier  ~ name +
   precipIntensity + precipProbability +  apparentTemperatureHighTime +
   apparentTemperatureMinTime, data = data_afterdrop)
summary(lm_multiplier_without_nonsig)

#Model Assumptions
plot(lm_full)
bptest(lm_full)
shapiro.test(resid(lm_full))

############## Model Selections (Backward / Forward) ##########

#backward step model using AIC
fit_back_aic = step(lm_full, direction = "backward", trace = 0)

#null model
lm_for_null = lm(price ~ 1 , data = data_afterdrop)

#forward step model using AIC

lm_for_aic = step(lm_full,direction = "forward",
                  trace = 0)

#Stepwise Both Direction Model using AIC
lm_both_aic = step(lm_full,
                   direction = "both",
                   trace = 0)

#Compare models for AIC group
AIC(fit_back_aic , lm_for_aic , lm_both_aic)
summary(fit_back_aic)$adj.r.squared
```

```r
summary(lm_for_aic)$adj.r.squared
summary(lm_both_aic)$adj.r.squared

#Model Assumptions for best of three
plot(fit_back_aic, pch = 1, cex = 0.5)
bptest(fit_back_aic)
shapiro.test(resid(fit_back_aic))

#using BIC as penalty
n = nrow(data_afterdrop)

#backward step model using BIC
fit_back_bic = step(lm_full, direction = "backward", trace = 0, k = log(n)
    )

# null model
lm_for_null = lm(price ~ 1 , data = data_afterdrop)

#forward step model using BIC
lm_for_bic = step(lm_full,
                  direction = "forward",
                  trace = 0, k = log(n) )

#Stepwise Both Direction Model using BIC
lm_both_bic = step(lm_full,
                   trace = 0,
                   k = log(n) )

#Model Comparison
BIC(fit_back_bic, lm_for_bic , lm_both_bic)

#adj r2
summary(fit_back_bic)$adj.r.squared
summary(lm_for_bic)$adj.r.squared
summary(lm_both_bic)$adj.r.squared

#Collinearity
vif(lm_both_aic)
vif(lm_both_bic)

#Model Assumptions
plot(fit_back_bic)
bptest(fit_back_bic)
shapiro.test(resid(fit_back_bic))

#The last obs is an influential point
out_i = which(cooks.distance(fit_back_aic) > 4 / length(cooks.distance(fit
   _back_aic)))
data_new = data_afterdrop[-out_i,]

#refit the both and check model assumption
lm_both_aic_without_inf = lm(price ~ day + month + name + distance + surge
```

```
    _multiplier +
                                  visibility + temperatureHighTime +
    temperatureLowTime + apparentTemperatureHigh +
                                  apparentTemperatureLow + temperatureMin +
    temperatureMinTime + apparentTemperatureMinTime,
                               data = data_new)

#Model Assumpiton check again
plot(lm_both_aic_without_inf, cex = 0.5)
bptest(lm_both_aic_without_inf)
shapiro.test(resid(lm_both_aic_without_inf))

n = nrow(data_new)

#Create k equally size folds
k = 5
folds <- cut(1:n,breaks=k,labels=FALSE)

RMSE_kcv_both_aic = RMSE_kcv_both_bic = numeric(k)

#Perform a k-fold cross validation
for(i in 1:k)
{
  # Find the indices for test data
  test_index = which(folds==i)

  # Obtain training/test data
  test_data = data_new[test_index, ]
  training_data = data_new[-test_index, ]



  kcv_both_aic = lm(price ~ name + distance + surge_multiplier +
    temperature + hour + windSpeed + apparentTemperature + month + name:
    distance, data = training_data)

  kcv_both_bic  = lm(price ~ name + distance + surge_multiplier + name:
    distance, data = training_data)

  # Obtain RMSE on the 'test' data

  resid_both_aic = test_data[,6] - predict(kcv_both_aic, newdata=test_data
    )
  RMSE_kcv_both_aic[i] = sqrt(sum(resid_both_aic^2)/nrow(test_data))

  resid_both_bic = test_data[,6] - predict(kcv_both_bic, newdata=test_data
    )
  RMSE_kcv_both_bic[i] = sqrt(sum(resid_both_bic^2)/nrow(test_data))
}

# Chooses fit_quad
mean(RMSE_kcv_both_aic)
```

```r
mean(RMSE_kcv_both_bic)


y_pred_train = predict(lm_both_bic, newdata = data_afterdrop)
mse_train = mean((y_pred_train - data_afterdrop$price)^2)
mse_train

##clean the test data
data_test = read.csv("Test data.csv")
data_test$price = log(data_test$price)
data_test$name = as.factor(data_test$name)

data_newtest = data_test[-c(which(data_test$name == "Taxi")),]


#################### Prediction ######################

#predict test
y_pred_test = predict(lm_both_bic, newdata = data_newtest)
mse_test = mean((y_pred_test - data_newtest$price)^2)
mse_test

#plot prediction
plot(y_pred_test,data_newtest$price,pch=10,col="cyan4")
abline(coef=c(0,1),c="orange",lwd=3)
plot(y_pred_train ,data_afterdrop$price,pch=10,col="cyan4")
abline(coef=c(0,1),c="orange",lwd=3)
```

Appendix B Figures

```
Call:
lm(formula = price ~ ., data = data_afterdrop)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97092 -0.07807  0.00004  0.07294  0.83073

Coefficients: (3 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                -8.824e+01  1.923e+02  -0.459  0.64641
hour                       -1.448e-03  7.150e-04  -2.025  0.04299 *
day                        -3.845e-03  1.118e-02  -0.344  0.73087
month                      -1.232e-01  3.322e-01  -0.371  0.71069
cab_typeUber                1.301e+00  1.303e-02  99.850  < 2e-16 ***
nameBlack SUV               4.128e-01  1.335e-02  30.925  < 2e-16 ***
nameLux                     1.111e+00  1.295e-02  85.840  < 2e-16 ***
nameLux Black               1.391e+00  1.349e-02 103.155  < 2e-16 ***
nameLux Black XL            1.744e+00  1.311e-02 133.017  < 2e-16 ***
nameLyft                    5.165e-01  1.348e-02  38.314  < 2e-16 ***
nameLyft XL                 9.661e-01  1.309e-02  73.825  < 2e-16 ***
nameShared                         NA         NA      NA       NA
nameUberPool               -8.422e-01  1.295e-02 -65.025  < 2e-16 ***
nameUberX                  -7.282e-01  1.327e-02 -54.885  < 2e-16 ***
nameUberXL                 -2.838e-01  1.300e-02 -21.838  < 2e-16 ***
nameWAV                    -7.412e-01  1.301e-02 -56.985  < 2e-16 ***
distance                    1.656e-01  2.368e-03  69.908  < 2e-16 ***
surge_multiplier            7.153e-01  2.886e-02  24.784  < 2e-16 ***
temperature                -5.428e-03  3.805e-03  -1.427  0.15379
apparentTemperature         5.980e-03  3.240e-03   1.846  0.06506 .
precipIntensity            -2.050e-01  2.216e-01  -0.925  0.35501
precipProbability          -9.969e-03  2.108e-02  -0.473  0.63638
windSpeed                   9.009e-03  2.797e-03   3.221  0.00129 **
visibility                 -1.509e-03  2.070e-03  -0.729  0.46618
temperatureHigh             1.418e-01  1.335e-01   1.063  0.28807
temperatureHighTime         1.568e-06  1.798e-06   0.872  0.38332
temperatureLow             -4.612e-03  3.750e-03  -1.230  0.21886
temperatureLowTime         -3.446e-07  1.120e-06  -0.308  0.75834
apparentTemperatureHigh    -5.173e-02  6.903e-02  -0.749  0.45368
apparentTemperatureHighTime 4.297e-06  3.168e-06   1.356  0.17506
apparentTemperatureLow      3.120e-03  2.318e-03   1.346  0.17838
apparentTemperatureLowTime -2.490e-08  1.366e-06  -0.018  0.98546
uvIndex                    -7.037e-04  6.501e-03  -0.108  0.91381
visibility.1                       NA         NA      NA       NA
sunriseTime                 4.051e-06  3.672e-06   1.103  0.27008
sunsetTime                         NA         NA      NA       NA
precipIntensityMax          2.501e-01  1.966e-01   1.272  0.20346
uvIndexTime                -4.245e-06  3.243e-06  -1.309  0.19063
temperatureMin             -1.377e-03  3.389e-03  -0.406  0.68448
temperatureMinTime          4.234e-07  5.281e-07   0.802  0.42283
temperatureMax             -1.449e-01  1.322e-01  -1.096  0.27306
temperatureMaxTime         -6.341e-07  6.706e-07  -0.946  0.34446
apparentTemperatureMin      2.849e-03  4.080e-03   0.698  0.48505
apparentTemperatureMinTime -4.926e-07  3.257e-07  -1.512  0.13058
apparentTemperatureMax      5.413e-02  6.838e-02   0.792  0.42868
apparentTemperatureMaxTime -4.539e-06  3.103e-06  -1.463  0.14363
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1416 on 2749 degrees of freedom
Multiple R-squared:  0.9399,    Adjusted R-squared:  0.939
F-statistic:  1024 on 42 and 2749 DF,  p-value: < 2.2e-16
```

Figure 6: Full Model Regression Summary

hour	nameBlack SUV	nameLux	nameLux Black
1.392158e+00	7.892083e+00	8.738304e+00	8.313095e+00
nameLux Black XL	**nameLyft**	**nameLyft XL**	**nameShared**
8.806241e+00	8.416156e+00	8.985597e+00	8.816721e+00
nameUberPool	**nameUberX**	**nameUberXL**	**nameWAV**
8.596115e+00	7.941553e+00	7.943647e+00	8.014477e+00
distance	**surge_multiplier**	**apparentTemperature**	**precipIntensity**
1.226380e+01	1.047407e+00	3.861170e+00	1.549829e+00
windSpeed	**temperatureHigh**	**apparentTemperatureHighTime**	**temperatureMax**
3.272250e+00	3.050562e+03	4.147169e+05	2.742163e+03
apparentTemperatureMax	**apparentTemperatureMaxTime**	**nameBlack SUV:distance**	**nameLux:distance**
2.512063e+01	4.110186e+05	8.056074e+00	8.686552e+00
nameLux Black:distance	**nameLux Black XL:distance**	**nameLyft:distance**	**nameLyft XL:distance**
8.376031e+00	8.860270e+00	7.928380e+00	9.155084e+00
nameShared:distance	**nameUberPool:distance**	**nameUberX:distance**	**nameUberXL:distance**
8.968435e+00	8.890595e+00	8.584693e+00	8.654848e+00
nameWAV:distance			
8.672694e+00			

Figure 7: VIF summary for ***Fit_Both_AIC***

nameBlack SUV	nameLux	nameLux Black	nameLux Black XL
7.863271	8.692547	8.292358	8.787548
nameLyft	**nameLyft XL**	**nameShared**	**nameUberPool**
8.384569	8.981209	8.778402	8.582693
nameUberX	**nameUberXL**	**nameWAV**	**distance**
7.909257	7.916121	7.995245	12.210261
surge_multiplier	**nameBlack SUV:distance**	**nameLux:distance**	**nameLux Black:distance**
1.044603	8.035478	8.623049	8.359485
nameLux Black XL:distance	**nameLyft:distance**	**nameLyft XL:distance**	**nameShared:distance**
8.837605	7.905808	9.148760	8.934026
nameUberPool:distance	**nameUberX:distance**	**nameUberXL:distance**	**nameWAV:distance**
8.865565	8.545576	8.631274	8.653991

Figure 8: VIF summary for ***Fit_Both_BIC***

```
Call:
lm(formula = surge_multiplier ~ . - price, data = data_afterdrop)

Residuals:
     Min       1Q   Median       3Q      Max
-0.07978 -0.03177 -0.00741  0.00367  1.44184

Coefficients: (3 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 7.453e+01  1.271e+02   0.587  0.55752
hour                        2.628e-04  4.724e-04   0.556  0.57800
day                         4.859e-03  7.384e-03   0.658  0.51058
month                       1.477e-01  2.195e-01   0.673  0.50114
cab_typeUber               -1.730e-03  8.609e-03  -0.201  0.84076
nameBlack SUV               1.957e-03  8.820e-03   0.222  0.82443
nameLux                     4.428e-02  8.513e-03   5.201 2.13e-07 ***
nameLux Black               4.622e-02  8.867e-03   5.213 2.00e-07 ***
nameLux Black XL            2.462e-02  8.648e-03   2.847  0.00444 **
nameLyft                    2.770e-02  8.892e-03   3.115  0.00186 **
nameLyft XL                 4.449e-02  8.605e-03   5.170 2.51e-07 ***
nameShared                         NA         NA      NA       NA
nameUberPool                2.648e-03  8.557e-03   0.309  0.75701
nameUberX                   2.457e-03  8.766e-03   0.280  0.77931
nameUberXL                  1.598e-03  8.588e-03   0.186  0.85238
nameWAV                     6.067e-04  8.594e-03   0.071  0.94372
distance                    9.741e-04  1.565e-03   0.623  0.53361
temperature                -1.194e-04  2.514e-03  -0.048  0.96211
apparentTemperature         1.504e-04  2.141e-03   0.070  0.94399
precipIntensity             3.875e-01  1.462e-01   2.650  0.00809 **
precipProbability          -2.898e-02  1.392e-02  -2.082  0.03745 *
windSpeed                  -2.431e-03  1.848e-03  -1.316  0.18830
visibility                 -1.562e-03  1.368e-03  -1.142  0.25351
temperatureHigh            -7.048e-02  8.817e-02  -0.799  0.42414
temperatureHighTime         6.217e-07  1.188e-06   0.523  0.60077
temperatureLow             -2.043e-03  2.477e-03  -0.825  0.40965
temperatureLowTime         -3.597e-07  7.399e-07  -0.486  0.62691
apparentTemperatureHigh     2.438e-02  4.561e-02   0.535  0.59299
apparentTemperatureHighTime -3.526e-06  2.092e-06  -1.686  0.09199 .
apparentTemperatureLow      1.395e-03  1.531e-03   0.911  0.36225
apparentTemperatureLowTime  6.563e-07  9.025e-07   0.727  0.46719
uvIndex                     3.166e-04  4.295e-03   0.074  0.94125
visibility.1                       NA         NA      NA       NA
sunriseTime                 3.732e-06  2.425e-06   1.539  0.12400
sunsetTime                         NA         NA      NA       NA
precipIntensityMax          7.870e-02  1.299e-01   0.606  0.54461
uvIndexTime                -2.970e-06  2.142e-06  -1.386  0.16573
temperatureMin             -8.544e-04  2.239e-03  -0.382  0.70278
temperatureMinTime          2.491e-07  3.489e-07   0.714  0.47529
temperatureMax              7.004e-02  8.734e-02   0.802  0.42267
temperatureMaxTime          4.293e-07  4.430e-07   0.969  0.33257
apparentTemperatureMin      3.551e-03  2.695e-03   1.318  0.18776
apparentTemperatureMinTime -5.622e-07  2.149e-07  -2.616  0.00896 **
apparentTemperatureMax     -2.603e-02  4.518e-02  -0.576  0.56453
apparentTemperatureMaxTime  1.681e-06  2.050e-06   0.820  0.41223
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09355 on 2750 degrees of freedom
Multiple R-squared:  0.05224,   Adjusted R-squared:  0.03811
F-statistic: 3.697 on 41 and 2750 DF,  p-value: 4.113e-14
```

Figure 9: Surge Multiplier Linear Regression Summary

```
Call:
lm(formula = price ~ name + distance + surge_multiplier + name:distance,
    data = data_afterdrop)

Residuals:
     Min       1Q   Median       3Q      Max
-1.01433 -0.06808 -0.00181  0.06458  0.85726

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.901484   0.033720  56.391  < 2e-16 ***
nameBlack SUV               0.559317   0.027068  20.664  < 2e-16 ***
nameLux                    -0.299660   0.026839 -11.165  < 2e-16 ***
nameLux Black               0.045981   0.028094   1.637 0.101809
nameLux Black XL            0.540815   0.027652  19.558  < 2e-16 ***
nameLyft                   -0.796409   0.028434 -28.009  < 2e-16 ***
nameLyft XL                -0.422662   0.027794 -15.207  < 2e-16 ***
nameShared                 -1.323752   0.027323 -48.448  < 2e-16 ***
nameUberPool               -0.766860   0.026816 -28.597  < 2e-16 ***
nameUberX                  -0.688728   0.026812 -25.687  < 2e-16 ***
nameUberXL                 -0.277450   0.025898 -10.713  < 2e-16 ***
nameWAV                    -0.680730   0.026125 -26.057  < 2e-16 ***
distance                    0.172519   0.007944  21.718  < 2e-16 ***
surge_multiplier            0.707653   0.027756  25.496  < 2e-16 ***
nameBlack SUV:distance     -0.068963   0.011218  -6.148 9.00e-10 ***
nameLux:distance            0.053746   0.011276   4.766 1.97e-06 ***
nameLux Black:distance      0.022004   0.011694   1.882 0.059994 .
nameLux Black XL:distance  -0.044918   0.011542  -3.892 0.000102 ***
nameLyft:distance           0.006687   0.012395   0.540 0.589584
nameLyft XL:distance        0.040475   0.011513   3.516 0.000446 ***
nameShared:distance         0.009762   0.011344   0.861 0.389581
nameUberPool:distance      -0.034289   0.011063  -3.100 0.001958 **
nameUberX:distance         -0.017882   0.010839  -1.650 0.099098 .
nameUberXL:distance        -0.003418   0.010483  -0.326 0.744379
nameWAV:distance           -0.028109   0.010591  -2.654 0.008002 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1368 on 2767 degrees of freedom
Multiple R-squared:  0.9435,    Adjusted R-squared:  0.943
F-statistic:  1926 on 24 and 2767 DF,  p-value: < 2.2e-16
```

Figure 10: Final Model Summary