Predicting Uber and Lyft Prices

BA810 - TEAM 8A

YING ZHANG, Jiahui Xu
JIAZHENG LI, YUJIA CAO
CHRISTIAN LAWRENCE

Table of Contents

Problem:

There is no clear indication of what contributes to successful dynamic pricing in ride-shares / public vehicular transportation.

- Customers might be interested in understanding the determinants of dynamic pricing so that they have better understanding of their commute options.
- Competitors (other ride-share or taxi companies) may want to optimize their pricing through machine learning

Solution:

Through predictive modeling, we will attempt to identify the important attributes that determine pricing/surges in demand.

About the Data

Data Source: Uber & Lyft Cab prices

- **Simulated** Uber and Lyft rides (https://github.com/ravi72munde/scala-spark-cab-rides-predictions)
  - Queried ride information every 5 minutes
  - Simulated in Nov/Dec 2018
  - About a week's worth of data
- Captured features on Uber/Lyft rides like price, distance, time/date (timestamp), origin, destination, type of ride (e.g. UberBlack vs UberX), etc.
- Captured weather information associated with location and time of simulated rides
  - Collected every hour

Weather Data
Size: 342KB
Shape: 6276 rows x 8 columns

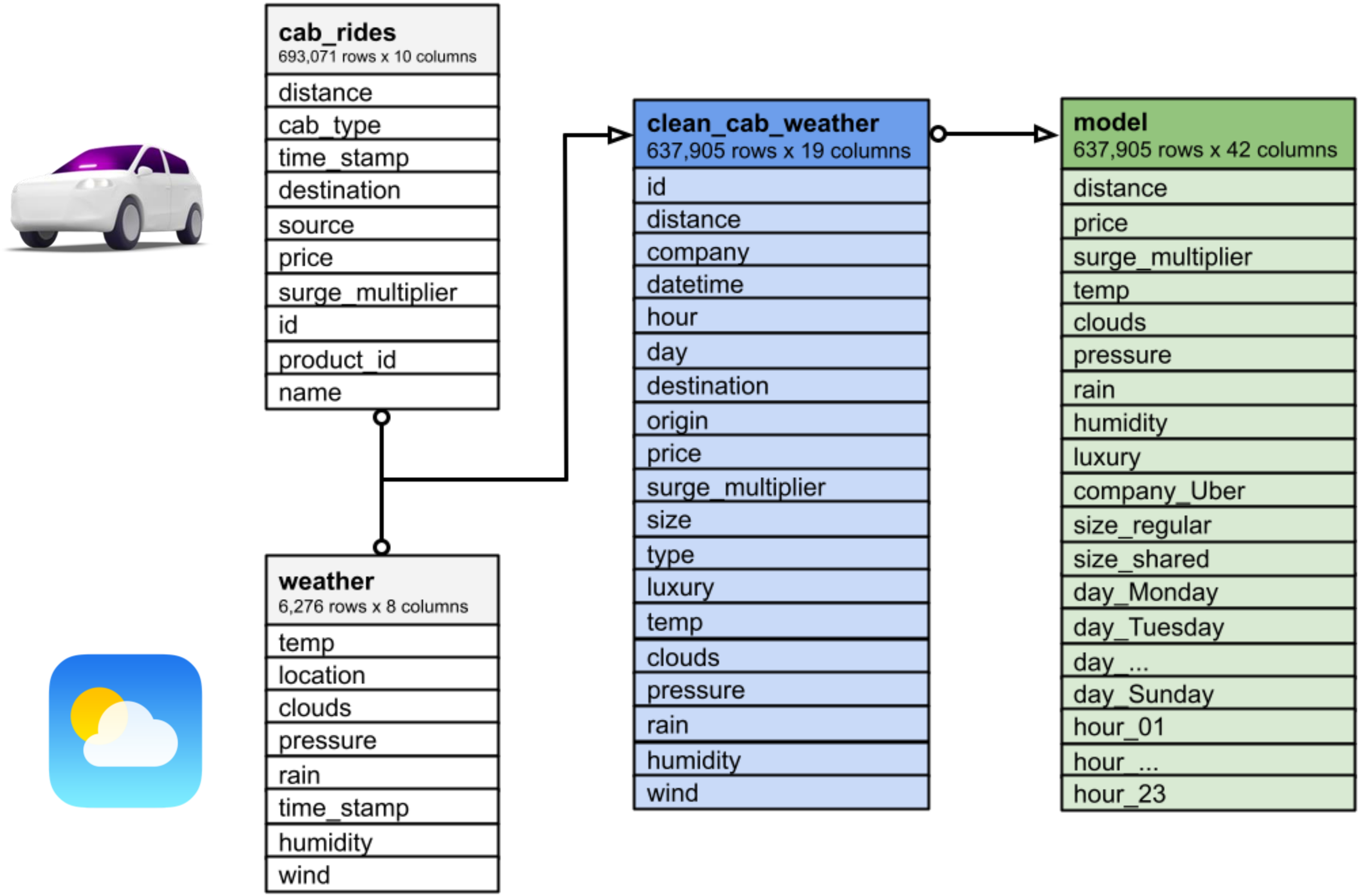
Uber/Lyft Data
Size: 85MB
Shape: 693,071 rows x 10 columns

Pre-processing Steps

1. Timestamp/datetime Conversion + feature engineering
   - Added: Hour (e.g. 09:00, 16:00), Day (e.g. Monday, Tuesday, etc.)
2. Missing data: `rain` and `price`
   - `price`: dropped missing rows since this was the target variable (~8% of dataset)
   - `rain`: filled with 0 since null values meant there was no rain during that day/time
3. Join data: `.mergeas_of()`
   - Merged on: closest datetime value and location/ride origin
4. Dummified Categorical Variables (after EDA)

Feature Engineering

1. Adding categorical variables `size` and `luxury`, replacing `product_id` & `type`
2. Adding `day` and `hour` variables to more easily conduct EDA and model on datetime object


cab_rides
693,071 rows x 10 columns
distance
cab_type
time_stamp
destination
source
price
surge_multiplier
id
product_id
name
weather
6,276 rows x 8 columns
temp
location
clouds
pressure
rain
time_stamp
humidity
wind
clean_cab_weather
637,905 rows x 19 columns
id
distance
company
datetime
hour
day
destination
origin
price
surge_multiplier
size
type
luxury
temp
clouds
pressure
rain
humidity
wind
model
637,905 rows x 42 columns
distance
price
surge_multiplier
temp
clouds
pressure
rain
humidity
luxury
company_Uber
size_regular
size_shared
day_Monday
day_Tuesday
day_...
day_Sunday
hour_01
hour_...
hour_23

Exploratory Data Analysis

Executive Summary of EDA

Our goal is to familiarize ourselves with the dataset and set hypotheses around which features may have the highest predictive value. We usually looked into one of the three aspects in our EDA:

- Distribution of variables’ values
- Relationship to price
- Potential collinearity with other features

Ultimately, we hypothesize that the following features carried the most predictive power:

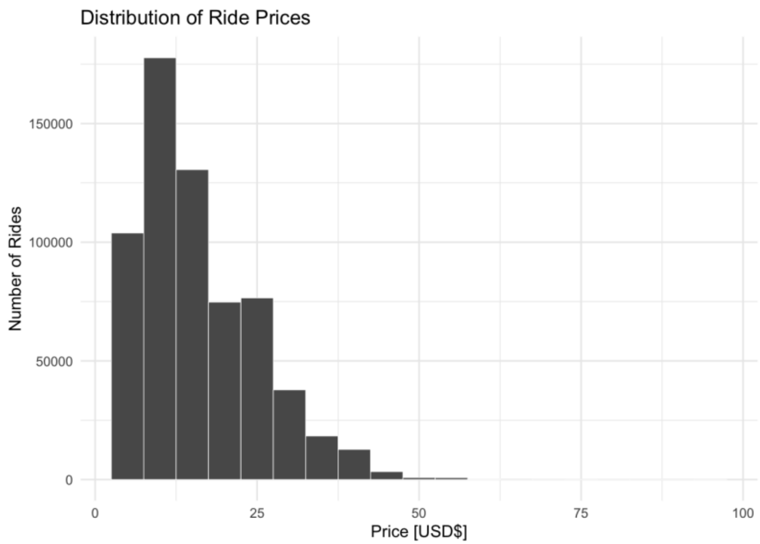
1. Distance
2. Ride type (size and luxury)

`price` - The distribution of ride prices was normally distributed with a right-skew

Key Findings:

- Normally distributed with a right skew
- High-priced ‘outliers’
    - Only made up of large, luxury rides (e.g Lyft Lux Black XL)

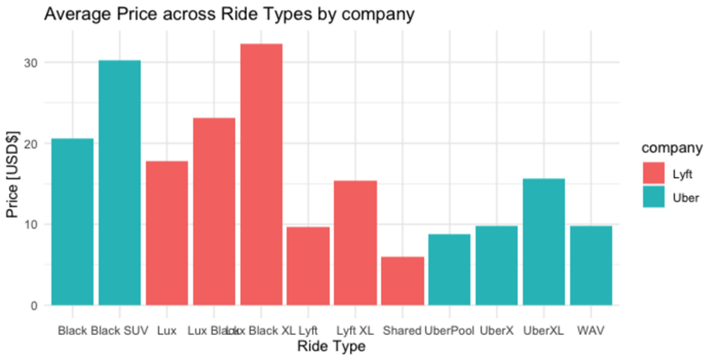
```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.50    9.00   13.50   16.55   22.50   97.50
```

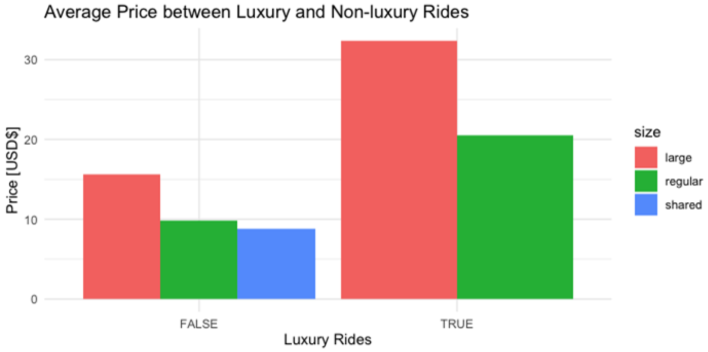


Luxury rides and larger vehicles are more expensive than their counterparts

Key Findings:

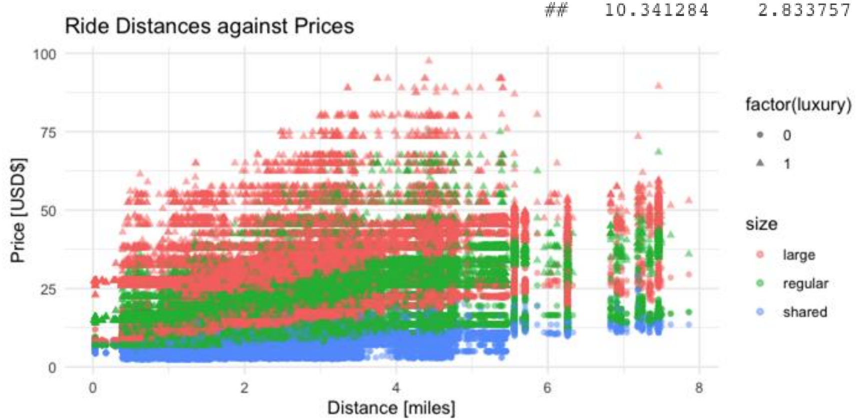
- Luxury rides are more expensive than non-luxury rides
- Larger rides are more expensive than regular rides, which are more expensive than shared rides.
- Uber's WAV (wheelchair-accessible vehicles) were priced very similarly to the regular non-luxury rides (UberX).





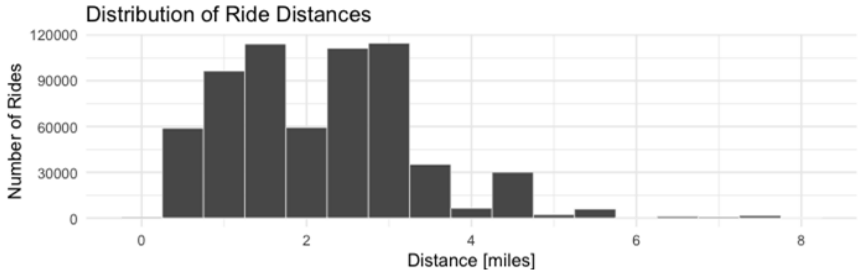
Correlation between price and distance

```
## (Intercept)    distance
##   10.341284    2.833757
```



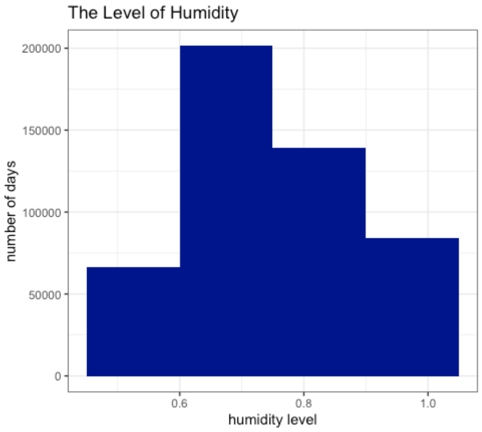
Key Findings:

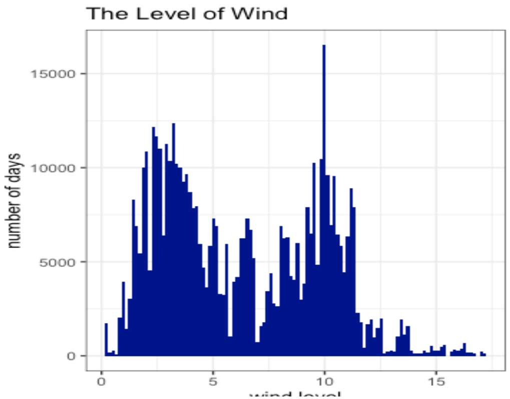
- Positive correlation between variables
- Distribution doesn't really matter since its a simulated dataset

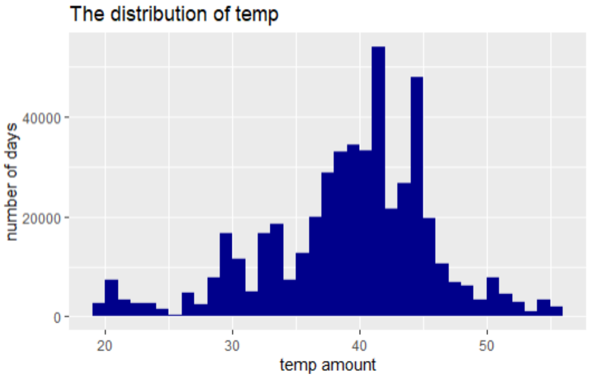


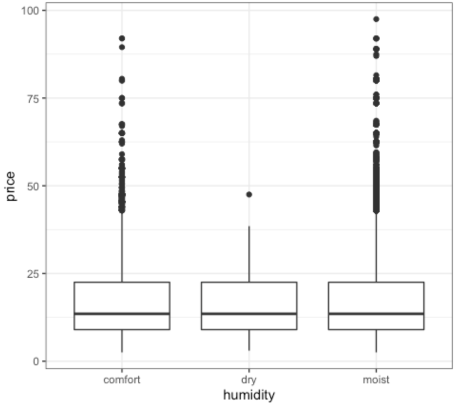
Correlation between price and weather

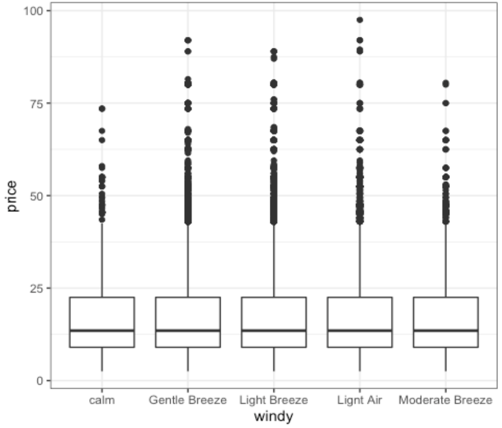
There is no significant correlation between price and the humidity, wind and temperature during any given time.

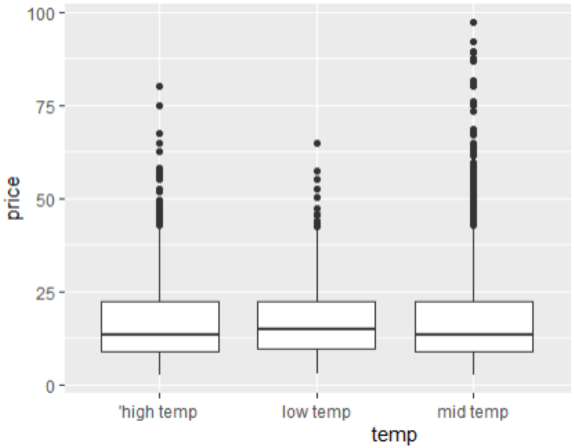






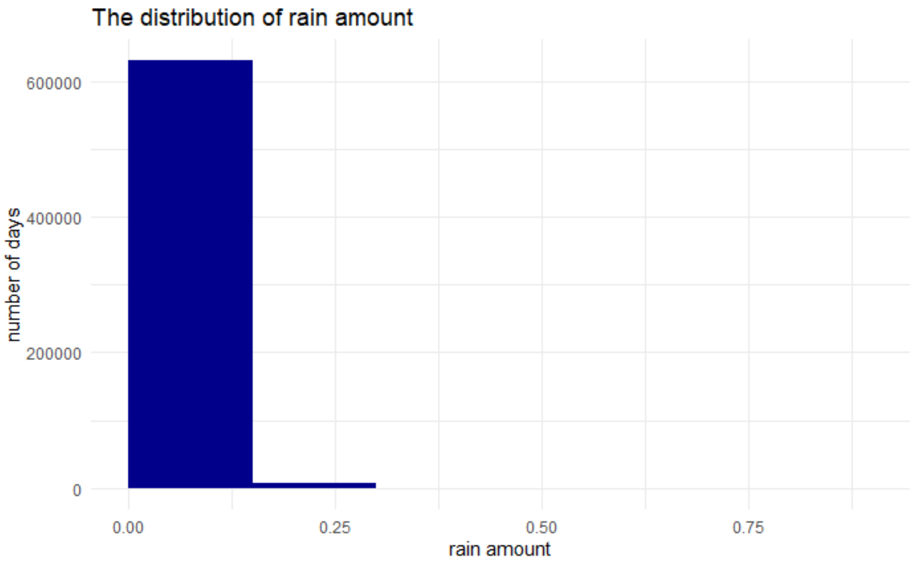


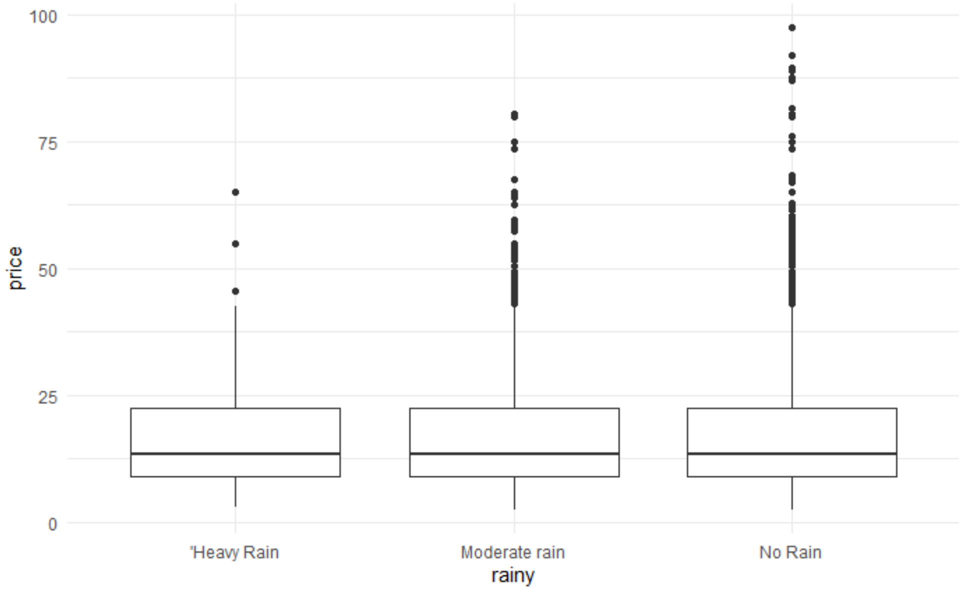




Rain

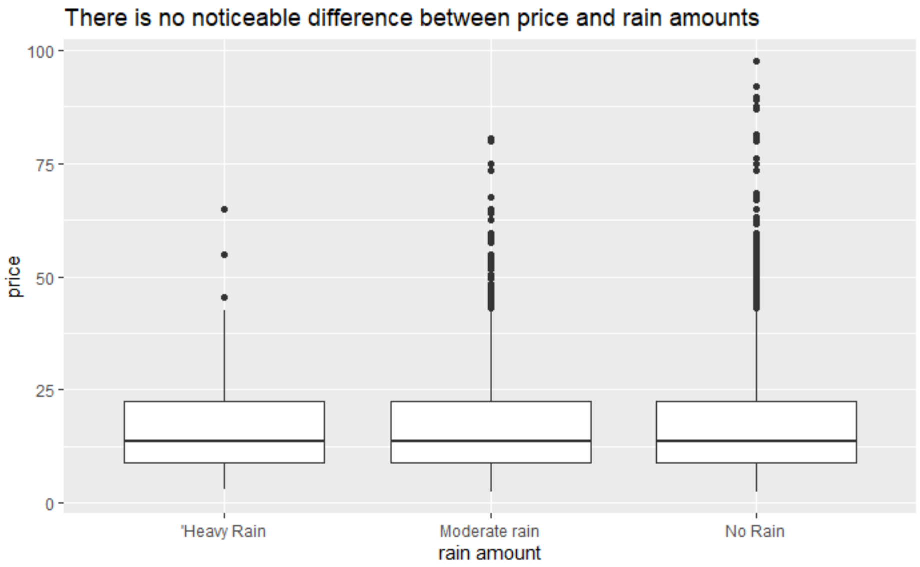
There is no significant correlation between price and the amount of rain during any given time.

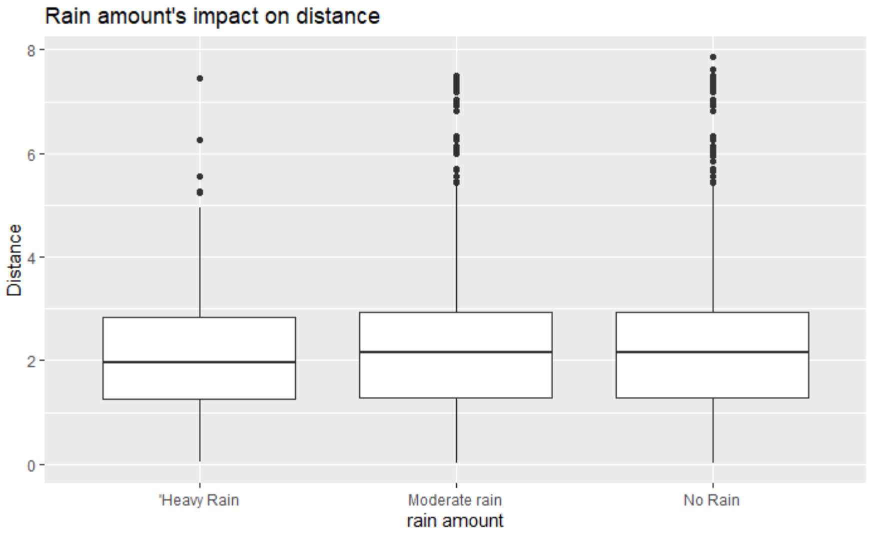




Rain

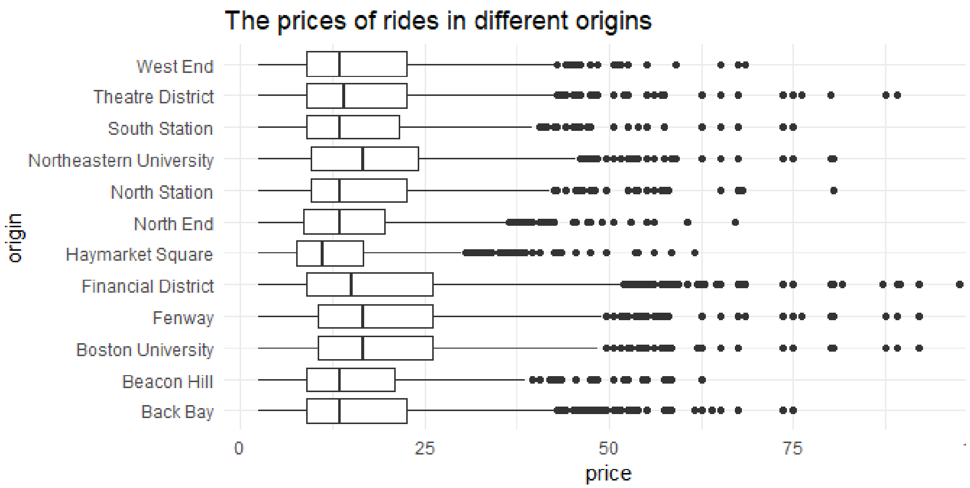
There is no significant correlation between price and the amount of rain during any given time. But rain amount has more or less some impact on the distance.

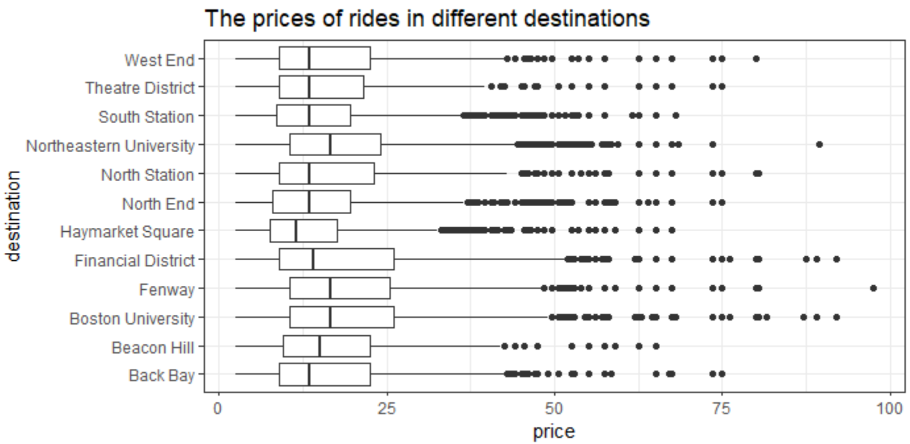




Origins & Destinations

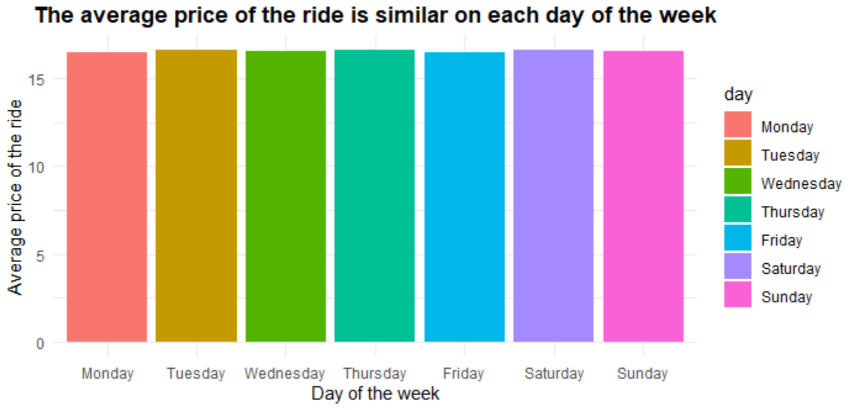
Comparing the distributions of price in different locations

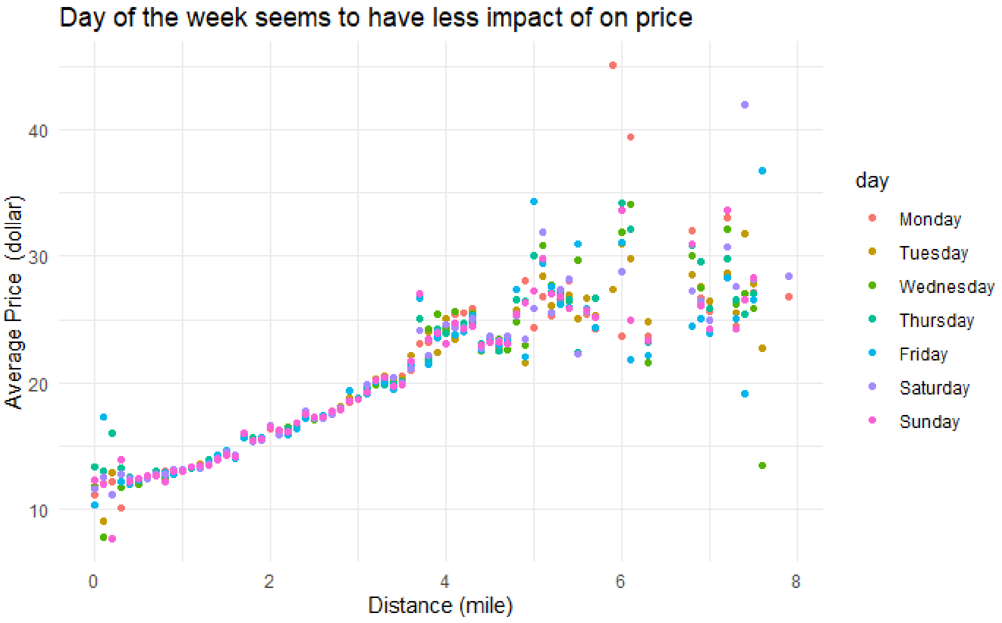




Day

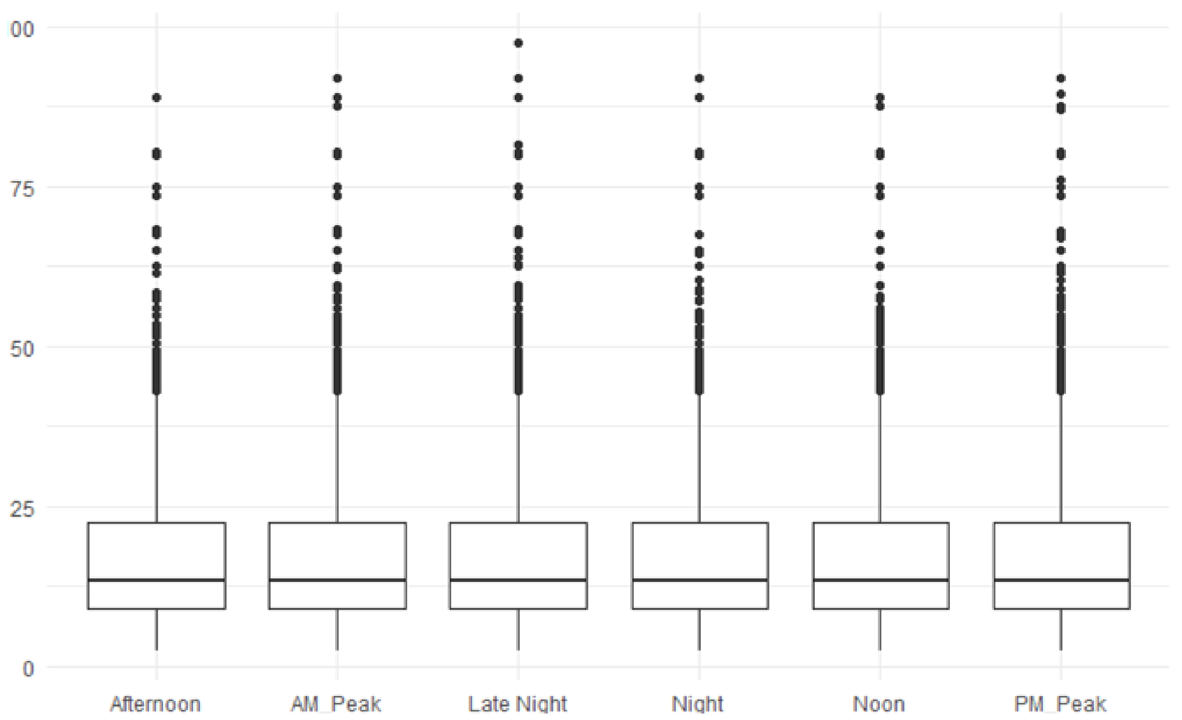
There is no significant correlation between price and the day of the week





Hour

There is no significant variance in price among the different ride times.



Supervised ML Model Development

```
# train test split
set.seed(810)
test_index <- sample(nrow(data), (nrow(data)*0.2)) # 80-20 split
data.test <- data[test_index]
data.train <- data[!test_index]

y.train <- data.train$price
y.test <- data.test$price
```

Splitting the Dataset

Linear Regression

Variables	MSE_{train}	MSE_{test}	Adjusted R^2
distance***	76.58	76.65	0.1196
Distance*** + Luxury*** + Size***	12.12	12.22	0.8606
Distance + Luxury + Size + Weather (rain, humidity, etc.) + Hour + Day + Company	12.04	12.13	0.8615

```
Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  11.6091287  0.8918947   13.016   <2e-16 ***
distance      2.8389599  0.0042766  663.841   <2e-16 ***
luxury       12.8667546  0.0108231 1188.827   <2e-16 ***
company_Uber  0.5694978  0.0098888   57.590   <2e-16 ***
size_regular -8.2881512  0.0108597 -763.202   <2e-16 ***
size_shared  -9.5161190  0.0155494 -611.991   <2e-16 ***
```

```
temp          -0.0015414  0.0012366  -1.246  0.2126
clouds        -0.0068687  0.0219221  -0.313  0.7540
pressure      -0.0010162  0.0008332  -1.220  0.2226
rain          -0.1544434  0.1413346  -1.093  0.2745
humidity      -0.0901787  0.0698692  -1.291  0.1968
wind           0.0026900  0.0020170   1.334  0.1823

day_Monday     0.0136423  0.0225512   0.605  0.5452
day_Saturday   0.0412823  0.0210141   1.965  0.0495 *
day_Sunday    -0.0056045  0.0211762  -0.265  0.7913
day_Thursday  -0.0203086  0.0219977  -0.923  0.3559
day_Tuesday   -0.0006005  0.0268491  -0.022  0.9822
day_Wednesday -0.0402010  0.0315165  -1.276  0.2021
hour_01       -0.0455677  0.0330971  -1.377  0.1686
hour_02        0.0105907  0.0329648   0.321  0.7480
hour_03       -0.0300100  0.0332306  -0.903  0.3665
hour_04       -0.0443747  0.0331112  -1.340  0.1802
```

```
hour_05  -0.0146953  0.0344083  -0.427  0.6693
hour_06  -0.0691161  0.0336624  -2.053  0.0401 *
hour_07  -0.0508847  0.0347341  -1.465  0.1429
hour_08  -0.0438177  0.0351310  -1.247  0.2123
hour_09  -0.0378896  0.0336998  -1.124  0.2609
hour_10  -0.0500848  0.0335240  -1.494  0.1352
hour_11   0.0450968  0.0334593   1.348  0.1777
hour_12  -0.0083098  0.0329385  -0.252  0.8008
hour_13   0.0217358  0.0325729   0.667  0.5046
hour_14  -0.0377382  0.0324703  -1.162  0.2451
hour_15  -0.0662235  0.0324616  -2.040  0.0413 *
hour_16  -0.0585604  0.0326945  -1.791  0.0733 .
hour_17   0.0109671  0.0331417   0.331  0.7407
hour_18  -0.0779638  0.0334574  -2.330  0.0198 *
hour_19  -0.0414647  0.0344062  -1.205  0.2281
hour_20  -0.0019465  0.0345646  -0.056  0.9551
hour_21  -0.0015978  0.0338629  -0.047  0.9624
hour_22  -0.0023238  0.0330936  -0.070  0.9440
hour_23  -0.0197782  0.0321692  -0.615  0.5387
```

The coefficient estimates around `weather`, `hour`, and `day` are either statistically insignificant or close to zero (or both).

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Final Linear Regression & Lasso Regression Model

Variables	MSE_{train}	MSE_{test}	Adjusted R^2
Distance*** + Luxury*** + Size*** + Company***	12.04	12.13	0.8615
Lasso: Distance + Luxury + Size + ~~Weather (rain, humidity, etc.) + Hour + Day +~~ Company	12.05	12.13	-

The MSEs and Adjusted R^2 **did not change** with the exclusion of the weather and time/day variables

Lasso Coefficients

```
(Intercept)    10.5978714
distance        2.7461901
luxury         12.6641518
company_Uber    0.3247694
size_regular   -7.9078942
size_shared    -9.1052035
```

```
temp          .
clouds        .
pressure      .
rain          .
humidity      .
wind          .
```

```
day_Monday     .
day_Saturday   .
day_Sunday     .
day_Thursday   .
day_Tuesday    .
day_Wednesday  .
hour_01        .
hour_02        .
hour_03        .
hour_04        .
hour_05        .
hour_06        .
hour_07        .
hour_08        .
```

```
hour_09        .
hour_10        .
hour_11        .
hour_12        .
hour_13        .
hour_14        .
hour_15        .
hour_16        .
hour_17        .
hour_18        .
hour_19        .
hour_20        .
hour_21        .
hour_22        .
hour_23        .
```

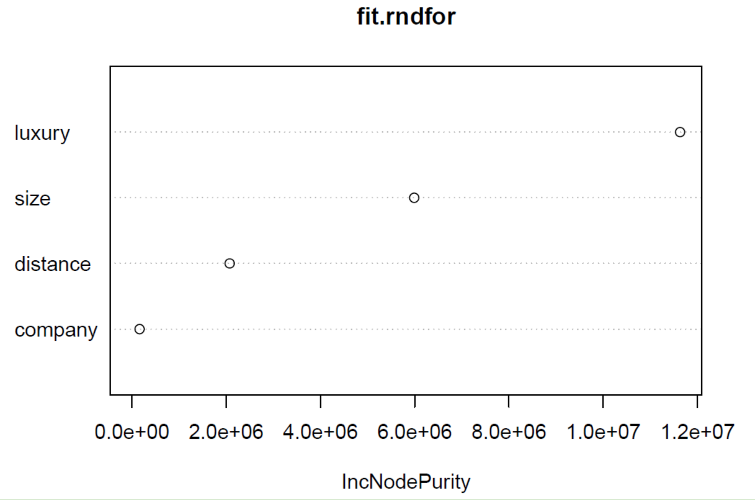
Our lasso model has decreased the coefficients for `weather`, `hour`, and `day` variables to 0.

Random Forest

```
f1 <- as.formula(price ~ distance + luxury + size + company )
```

```
dd.test <- cab[test_index]
dd.train <- cab[!test_index]
x.train <- model.matrix(f1, dd.train)[, -1]
y.train <- data.train$price
```

```
x.test <- model.matrix(f1, dd.test)[, -1]
y.test <- data.test$price
fit.rndfor <- randomForest(f1,
dd.train,
ntree=200,
do.trace=F)
```



Train MSE : **18.66631**

Test MSE: **18.62814**

Boosting Trees

- Splitting the data into train and test for 80% and 20% respectively
- Setting the equation: the predictor as price and the response variable; Distance, luxury, size and company as response variables.

```
f_dum <- as.formula(price ~ distance + luxury + size_regular + size_shared + company_Uber)
```

- Setting the lambda equals to 1 and the number of trees equals to 500, we had a really good result on this same equation we had.

Train MSE : **8.297794**

Test MSE: 8.371822

Challenges & Solutions

- Data pre-processing (joins and datetime parsing)

  ⟶ DataCamp courses and trial & error

- Random forest memory requirement

  ⟶ borrowing friend's computer with higher computational capabilities

- (Not) using the surge multiplier in our analysis
- Underfitting due to use of sample from training set (Subset of subset)

Conclusion

1. For competitors trying to optimize their dynamic pricing model based on Uber/Lyft's pricing ~ **Boosted forest model** is the most accurate model (lowest MSE)

2. For consumers trying to understand what goes into dynamic pricing ~ **Linear/lasso regression model** provided relatively low MSEs and high interpretability

(Intercept)	10.5978714
distance	2.7461901
luxury	12.6641518
company_Uber	0.3247694
size_regular	-7.9078942
size_shared	-9.1052035