Chicago Taxi Data

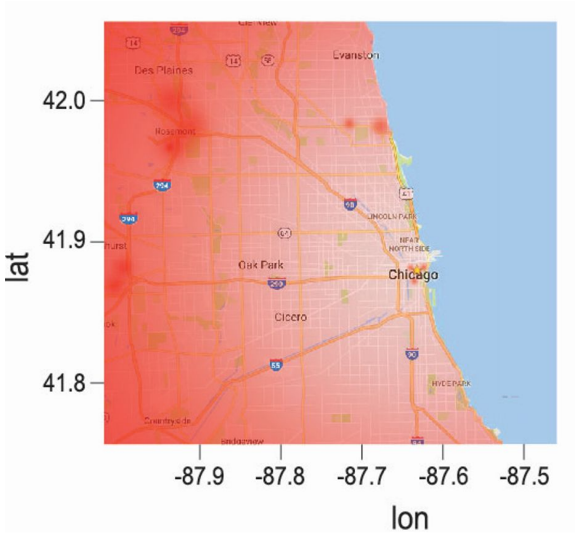
Jonggoo Kang, Becky Jacob, Don Crowley,
Ryan An, James Cooper

I. Non Technical Summary

This summary outlines the major findings of an analysis performed on taxi ride data in the city of Chicago. The data includes relevant information, such as categorical variables like Pick-up Neighborhood, Drop-off Neighborhood, Payment Type, and Company, as well as many numeric variables, such as: Date, Time, Trip Seconds, Trip Miles, Pick-up and Drop-off Census Tracts and Community Areas, Longitude/Latitude, Fare, Tips, Tolls, Extras, Trip Total, and Distance. The goal of this analysis was to help cab drivers maximize their tip earnings. This seemed like an especially relevant topic, as more and more people opt for ride share services over traditional taxis. Therefore, our analysis focused on predicting tip, as well as identifying key trends in the categorical variables that could help cab drivers make informed decisions while on the job.

We used Principal Component Analysis to examine the numerical variables with the hope of coming up with a model that could help taxi drivers best maximize their tips. First, we used PCA to identify hidden groups of variables that could be used to help simplify our analysis. The PCA led to four groups of variables. First was a group that could be described as intuitive factors of a trip cost such as trip length. The next component could be described as pickup location. This tells us that as rides get picked up further North and further West, tips tend to increase. Third was a component for drop-off location, again showing the same pattern. Finally, our fourth component was tolls, which is also positively correlated with tips.

Using these four groups of variables we then wanted to create and evaluate a model that could be useful for the taxi drivers. After running a regression, we created a model that accounted for about 64% of the reason that tips rise or fall, based on our grouped variables from PCA. Even though this doesn't tell the whole story, it could be useful for cab drivers as a piece of our larger analysis. We also attempted to use the numerical variables to calculate tips as a percentage of overall fare, but the model we came up with was not effective.

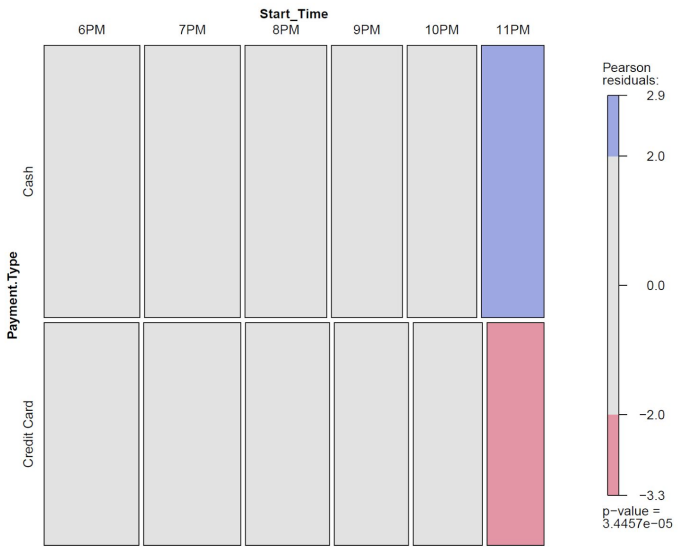


Our graph to the right shows a heat map of tips from rides taken from downtown. The darker the red, the larger the tip.

Because we had categorical variables such as neighborhood, payment type, cab company, and time of day, we wanted to explore Correspondence Analysis to find any insights into the data as to when taxi companies can get the most business, exactly where in the city they can get this business, and when are the best times to receive cash as opposed to credit card. This was important in conjunction with the other methods being used by our team to find where tip percentages tend to be the highest as well as what specific factors lead to higher tips. Running a Correspondence Analysis on pickups and dropoffs throughout the city by the hour, we were able to put together a nice picture of exactly where the most business occurs throughout the day. Because our Principal Component

Analysis had given us some good insight into where tips tend to increase, we could now tell from the Correspondence Analysis what times of day would give you the most business in these areas.

We also used Correspondence analysis to explore the data further to include payment types frequently used in each neighborhood as well as what cab companies tend to spend the most time in which neighborhoods. The purpose of this analysis was to give a cab company a nice profile of the city from a competitive standpoint as well as what time or neighborhood is more associated with cash tips vs. credit if either is preferred. The graph to the right indicates when payments in cash are most significant as opposed to credit card. The purple shows us an abundance of observed cash transactions and a scarcity of credit card transactions at 11pm across the city.



The last bit of analysis performed, was an attempt to predict if a customer would pay with credit card, or cash. Since tip amount was not provided for customers paying by cash, we have to assume that each cab driver has an idea of whether cash or credit card riders tip better. Based on their domain knowledge, they could use this model to find customers who are more likely to tip well. Linear Discriminant Analysis was used to create a model that predicted payment type with over a 90% accuracy rate. Unfortunately, this model could not be used by actual cab drivers, as it took into account all information provided in the dataset (including variables the cab driver would not know). A second model was created to see if cab drivers could predict payment type of a customer simply based on the location of the pick up. Though this model proved to be only slightly better than a random guess, the creation of the LDA model did allow us to see that pickup latitude and longitude were the most important factors and best predictors.

Though we were unable to build a model that would allow cab drivers to predict payment type, we were able to see some interesting trends in location, date and time. In addition, being able to explain 64% of the variance in tips could help a cab driver choose rides more intelligently, and perhaps increase their average tip amount.

II. Technical Summary

A. Data Exploration and Cleaning

The initial dataset was quite large, and had to be trimmed down as our computers could not handle the computational cost of such a large dataset. Therefore, a random 20% sample was taken of the original data, making the final dataset we worked with 192924 rows, with 24 features. To be detailed with categorical variables, they are all nominal. Also, we have 6 discrete variables of Start Date, End Date, Pick-up Census Tract, Drop-off Census Tract, Pick-up Community Area, Drop-off Community Area, and other numerical variables are continuous. In general, our data set is positively skewed, so we took a log normalization for the columns. After taking a normalization, our data set become slightly skew to the left.

A majority of the data cleaning was performed in python, by utilizing pandas and numpy. Some of the columns had unclear names, so they were renamed appropriately. Two of the columns: Pickup Centroid Location and Dropoff Centroid Location, were simply duplicates of the Latitude and Longitude columns, so they were removed. There were dollar signs and other special characters sprinkled throughout the data, which did not allow us to correctly import numeric variables. Regular expressions in python were used to remove these, and further clean the data to make it as consistent as possible.

There were a few features which had only a couple missing variables, and so these were deleted. However, others were missing quite a few, and needed to be filled appropriately. To examine missing categorical variables, we were able to plot the values using the a jitter graph to see which variables matched up. There were more values missing for the dropoff location than pickup location in the data. Using the jitter graph we were able to see clearly which values of pickup location correlated heavily to the dropoff location. In this way we replaced missing values for drop off area by associating the most common pickup location.

B. Multiple Regression Analysis

For our first analysis, we used regression to predict tip amount and identify how much each variable affected the tip amount. To build the regression model, we used six numerical explanatory variables. We took a random sample of 20% of the original data, and compared the means and IQRs of the reduced dataset to the original to make sure we weren't corrupting the data. Upon examining the data, several of the overall fares were in the hundreds of dollars. These seemed more likely to be mistaken entries than actual rides, so we decided to remove any lines of data with overall fares that were three standard deviations away from the mean.

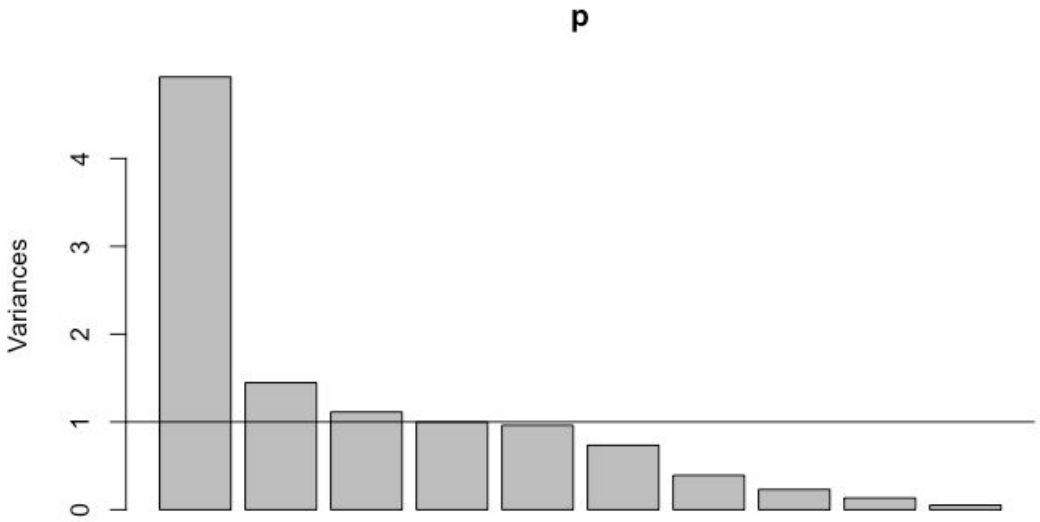
After selecting the variables to use and removing lines of data with Fare outliers, we made several attempts at creating a regression model. We used forward, backward, and both regression techniques. After cleaning our data and trying three different regression techniques, the best regression model we could produce only explained 36.3% of the variance. The screenshot below shows the best seven models produced by our regressions. Model three explained the most variance, using Trip_Seconds, Trip_Miles, and Fare variables to predict tips.

Model	Trip_Seconds	Trip_Miles	Fare	Distance (Long & Lat)	Extras	Trip_Sec x Distance (Tolls)	Classification Success*
0	< 2e-16	X	X	X	X	X	39.7%
1	X	< 2e-16	X	X	X	X	13.9%
2	X	X	< 2e-16	X	X	X	24.2%
3	4.40e-06	1.13e-02	< 2e-16	X	X	X	36.3%
4	1.45e-05	X	< 2e-16	0.39	X	X	19.5%
5	< 2e-16	0.00024	< 2e-16	X	0.65	0.72	14.5%
6	< 2e-16	X	< 2e-16	X	X	< 2e-16	21.5%

C. Principal Component Analysis

With eleven numerical variables, our dataset was well suited for a principal component analysis. The idea was to use the analysis to predict either tip or tip as a percentage of total fare with the numerical values. Because we only have tip values for payments made with credit cards, we had to reduce the data further by taking out cash payments.

First, we split the data into test and training sets with a 75%-25% split. We did an initial analysis with prcomp and created a scree plot to determine how many factors to use in the analysis. Five factors took us to about 86% and that was also where the "knee" appeared to be in our scree plot. The principal components from prcomp didn't appear to tell a story, so we decided to use a varimax rotation to rotate the components and see if any underlying groupings could be found. After looking at the varimax rotation, the last two components were single variables, so we decided to further reduce to four principal components.



The varimax rotation of the components worked well to help us interpret the components. Component one contained trip seconds, fare, trip total, distance, and pickup longitude. Except for longitude, these are the variables that would be most likely to go into a straightforward calculation of the cost of a taxi trip. The second component is the pickup location (latitude and longitude). The third component is the drop-off location (latitude and longitude). The fourth component is the tolls. We can also note that the variables 'Trip.Miles', and 'extras' were removed entirely.

Next, we created a model using summated scaling. This model had an adjusted r squared of .524 and all the variables were statistically significant. We ran a cross validation with 75% of the data in the training set and 25% of the data in the testing set. The residual standard errors were 1.98 and 3.43 respectively.

We created a model using the lm function in R to come up with a regression to predict tips from our eleven numerical variables. This model had an adjusted R squared of about .63, meaning it was better at predicting tips than our summated scales model. Again, we ran a cross validation to make sure that our residual standard errors were comparable. All our variables were statistically significant, as was the model as a whole.

We also attempted the same analysis with tips as a percentage of total fare. The idea was that knowing how to find the largest tips in percentage terms would help the taxi drivers maximize their behavior. Unfortunately, our model only had an R squared of about 3%, rendering the model useless. Using the log of fare also didn't help to increase our R squared.

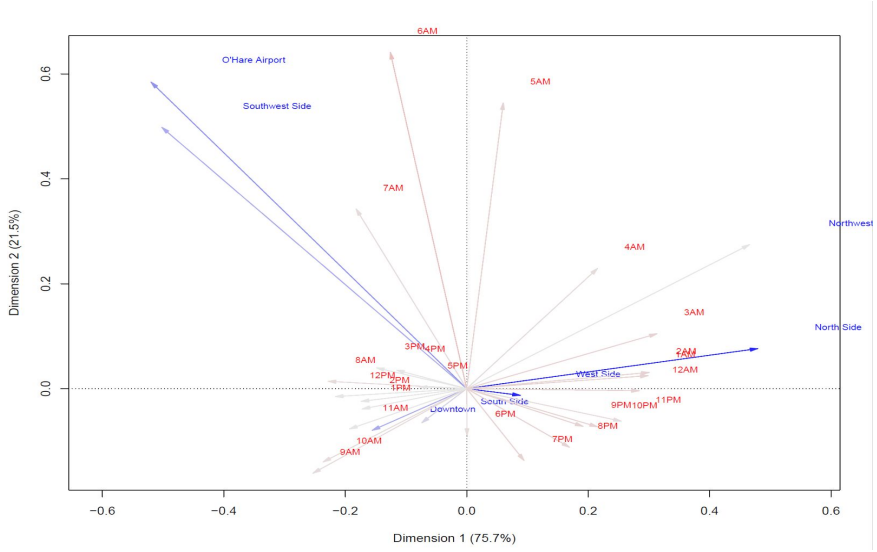
Overall, our PCA model should prove moderately useful to the cab drivers. 63% of the variance is explained by our model, and it provides a good starting point to be used in conjunction with the analysis of our non-numerical data. Our model shows that tips increase as pickups and drop-offs occur further from the lake and further North. Tips also increase as time, distance, fare, and tolls increase. The major weakness in terms of usability is that we did not have any tip data for cash purchases and were therefore unable to model these types of purchases.

D. Correspondence Analysis

Our data also contained categorical variables that when paired with analysis of the numerical variables could be very useful for a cab driver or company. Plotting the data and exploring the time series gave an early indication of where we would expect to see the most pickups and dropoffs. The neighborhoods were broken up into North Side, Northwest Side, Downtown, South Side, Southwest Side and West Side using the community area codes in the data, the second analysis used all the community codes giving every specific neighborhood in Chicago. This early analysis gave some unsurprising results, cabs most frequent downtown during the hours of 8-5, but some other trends seemed to pop out as well, and a specific breakdown was

also obtained using all 77 neighborhoods in Chicago with the hours of the day.

After this early analysis, the Correspondence Analysis was performed on the data. For pickup and dropoff neighborhood, the analysis gave us a specific neighborhood profile to work with as well as the more general North/West/South/Downtown profile. Plotting dimension 1 and dimension 2 for pickup neighborhood gave 51% and 23.8% of the variance respectively and 75.7% and 21.5% of the variance in dimension 1 and 2 for drop off neighborhood. We could see a clear picture of the busiest airport times as well as the North Side and Downtown with the variance being captured very well for these levels in the data that of . The seventy-seven neighborhoods plot gave 72.9% and 10.5% in dimension 1 and 2 for pickups and 70.9% and 17.5% for dropoffs.
Drop off Times in North/South/West/Downtown areas:



Moving on to the more in depth Correspondence Analysis with all 77 neighborhoods and looking at the row coordinates and column coordinates in the data, we can pinpoint which specific areas are most associated with which times. The most positive values in dimension 1 are associated with the late night/early morning hours. Looking at the seventy-seven neighborhoods, we can see for pickups that the North and Northwest Side neighborhoods fit into this category - that of Logan Square, West Town, Humboldt Park, Lakeview, Lincoln Park, and Lincoln Square all correspond heavily to late night and early morning hours. Humboldt Park was the most positive for the row coordinates and 3AM the most positive for the column coordinates. Since the PCA gave us the areas of the city tending to generate more tips (North and West Sides), we can now say with confidence what time cabs should focus on these neighborhoods. Combining these analyses also gives the possibility that late night riders tend to be a little more generous with their tipping.

Also examined was the payment type and the time of day. When using a mosaic plot, we can see a divergence in cash vs. credit, specifically in the early morning around 5am and at night around 11pm when cash is more giving more of an abundance of observations. Looking at the dimension 1 'Payment Type' coordinate we can see that Cash is slightly negative while Credit is

slightly positive. In the time coordinate we can see that the many of the values are slightly positive and slightly negative, but a bit of a trend develops in that the majority of the early morning hours are positive and the business hours are negative.

E. Linear Discriminant Analysis

As a final step to this project, we utilized linear discriminant analysis to try to predict the payment type. Since the payment type variable had several different levels, this seemed to be the best route. Two separate LDAs were completed. The first included all numeric variables available from the dataset, while the second focused on a more realistic application wherein the cab drivers know only the pickup location variables.

Both LDAs began the same way, with data cleaning and splitting between training and test sets. First a subset of the data was selected based on the appropriate variables. For the first LDA this included all numeric variables, for the second LDA it included only the Pickup Community Area, Pickup Longitude, Pickup Latitude and the Pickup Census Tract. After creating these intial subsets, the data was then split again based on a random selection of 75% for the training data, and 25% for the testing data. Boxplots were used to visualize the dependent variable for both sets in order to be certain that a random sample had been chosen.

After creating clean and appropriate datasets, it was important to decide which variables should be included in the model. Unlike some other multivariate analyses, LDA does not have built in variable selection options such as step-wise, forward, backward, etc. Instead of conducting a forward selection by hand, we chose to utilize ANOVA tests to gauge significance of each independent variable. An ANOVA test was performed by turning each independent variable in the main model, into a dependent variable and using the dependent variable as the independent variable in the ANOVA. After each was conducted, variables that proved not to be statistically significant were removed. The first LDA found that both Fare and Tip were insignificant and so they were removed. The second LDA found all four variables to be significant, so the dataset was not altered.

After thoroughly preparing the data, the LDA models were created. The first model performed quite well, and performed very similarly on the training and test set, so there was little concern of overfitting. Below you can see the accuracy results for each payment type, as well as the overall accuracy. Cash was predicted correctly nearly 100% of the time, while Credit Card was predicted correctly just over 80% of the time. This lead to an overall accuracy of 91.44%.

```
> diag(prop.table(ct, 1))
      Cash Credit Card     Dispute   No Charge       Pcard      Prcard     Unknown
 0.9994899   0.8208455   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
> sum(diag(prop.table(ct)))
[1] 0.9144403
```

The first model gave us quite a bit of hope, that payment type would be easily predictable for cab drivers. Unfortunately, after removing so many important variables, we lost most of our accuracy. Below you can see the test set results (again, they were very similar to the training set results). Being correct only 58% of the time, is not much of an improvement on randomly

guessing. It would seem that our second model was quite good at predicting cash payments, but regularly classified rides as cash when they were in fact credit cards.

```
            p
              Cash Credit Card
Cash         25487        1424
Credit Card 18458        2509

> diag(prop.table(ct, 1))
       Cash Credit Card
  0.9470848   0.1196642
> sum(diag(prop.table(ct)))
[1] 0.5847362
```

Interestingly, in this second LDA the most important factors in the discriminant was longitude and latitude. This gives us some certainty in our model, as it reinforces what was seen in the PCA analysis. Clearly, location within the city has a large effect on tips, and has at least a minor effect on payment type.

F. Conclusion

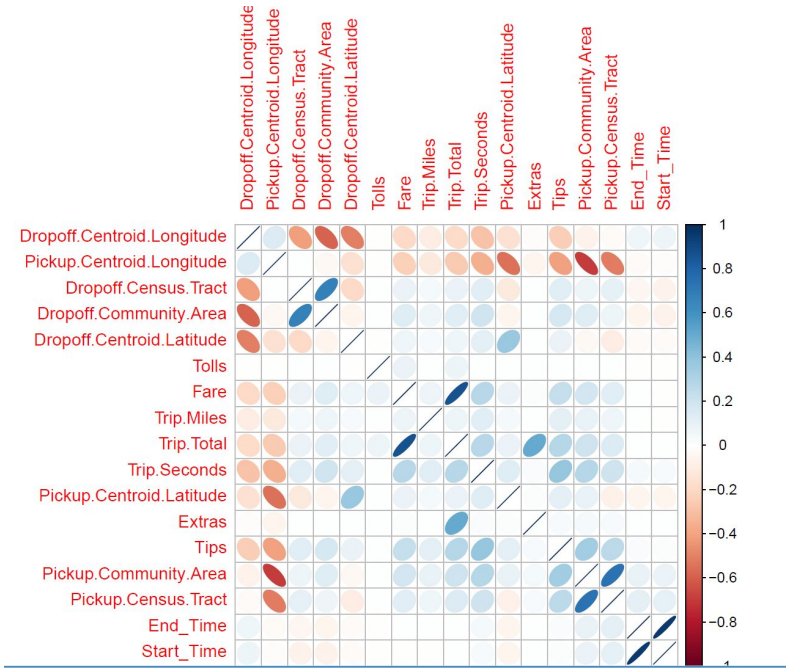
We began our analysis with the hypothesis that we could help the cab drivers of Chicago maximize their tips. After our very first attempt at a basic linear regression, we realized it might be harder than expected. Taking advantage of PCA to limit the number of variables, allowed us to account for 64% of the variance, a large increase from the initial regression. While this may be helpful to cab drivers, we wanted to see what other insights could be drawn from the categorical variables. Correspondence analysis showed us important relationships between time of the ride and neighborhoods throughout Chicago. Insights such as: late night riders being more generous, as well as an increase in tip when riders were in the north or northwest areas of the city, could prove to be beneficial to cab drivers. Our LDA model, though it did not lead to high accuracy rate, did show us that even in predicting the type of payment, location of drop off and pick up are key predictors. The above analysis may not lead to millionaire cab drivers in Chicago, but it can hopefully shed some light on trends in the industry.

G. Appendices

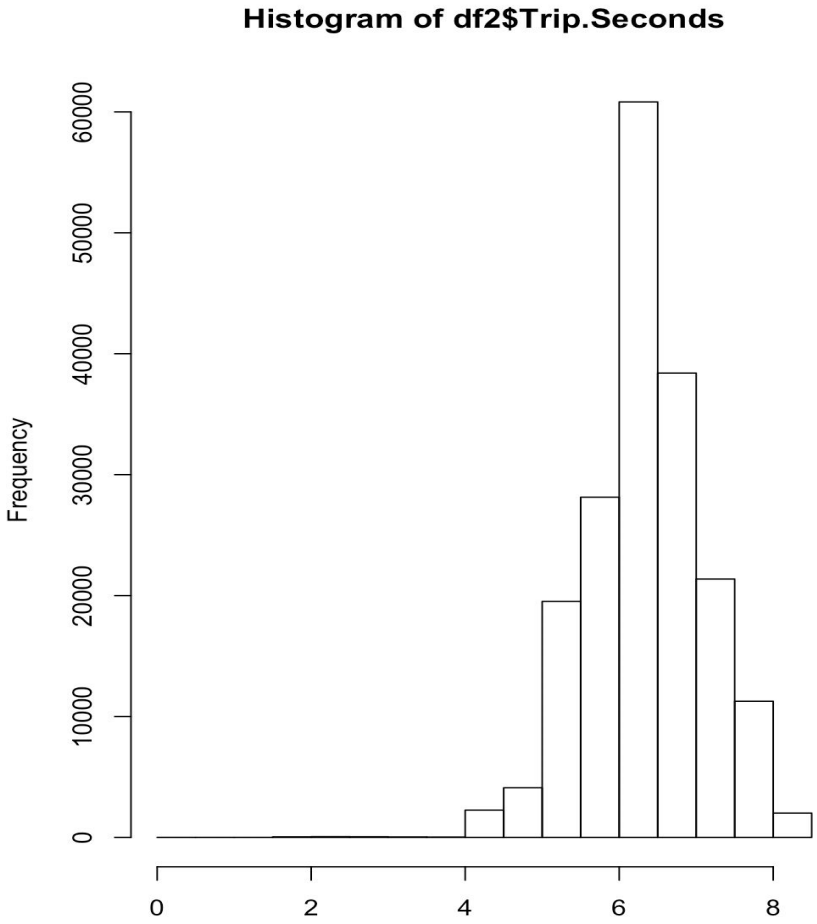
1. Visualizations
2. Individual Reports
3. Code

Visualizations

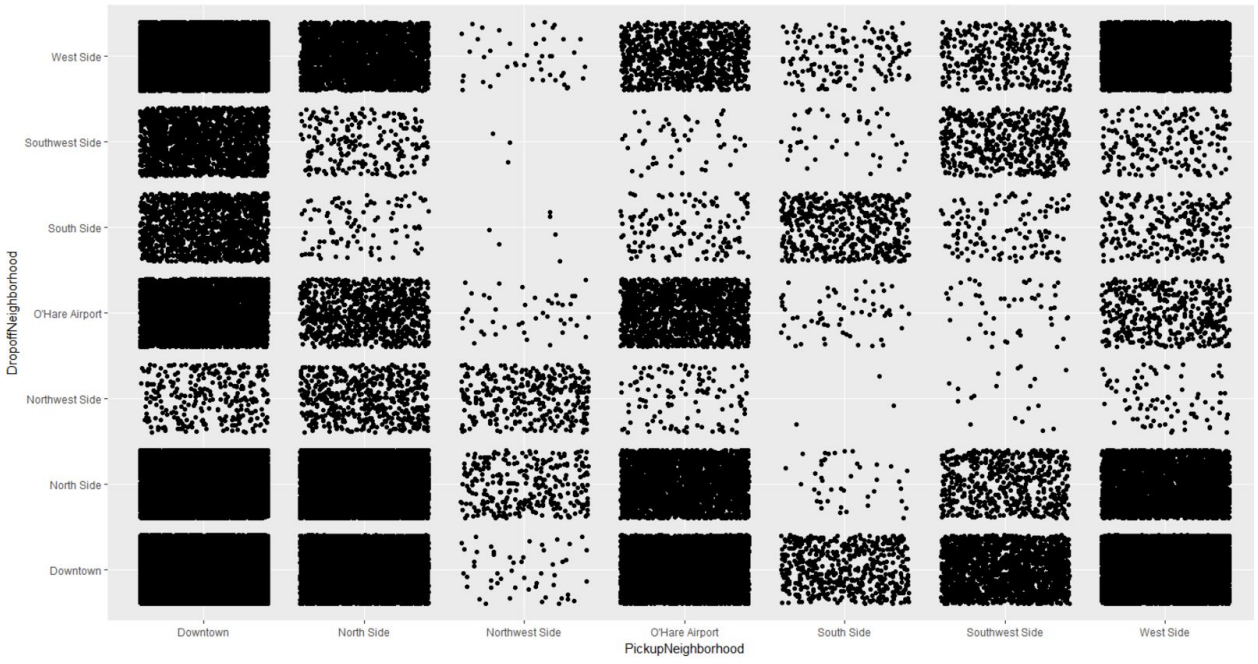
Exploratory:



skew to the left after log normalization

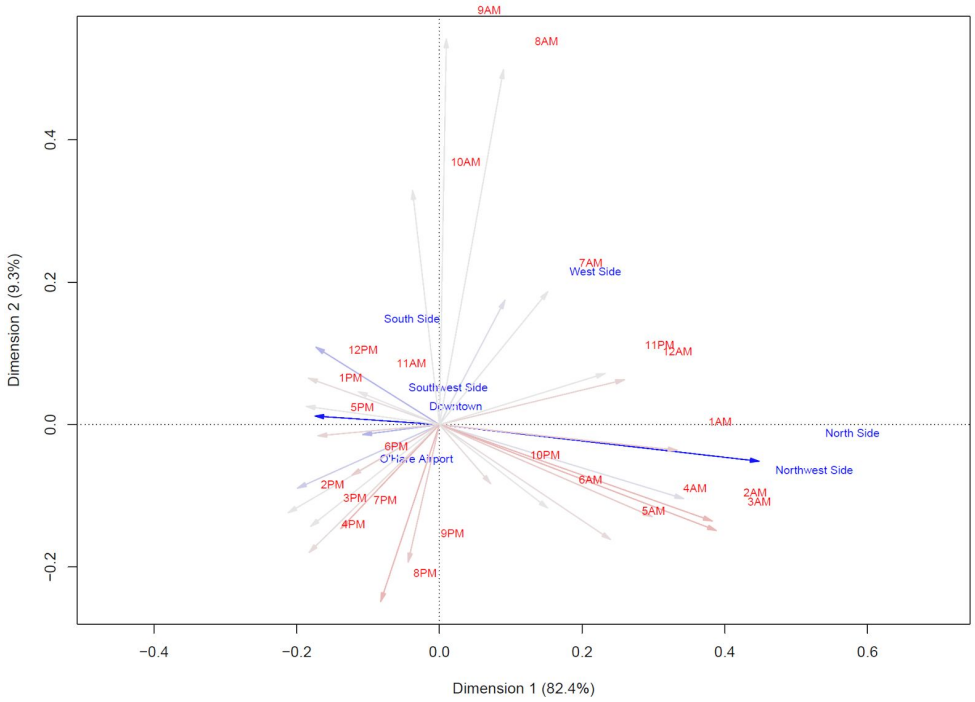


Missing Values

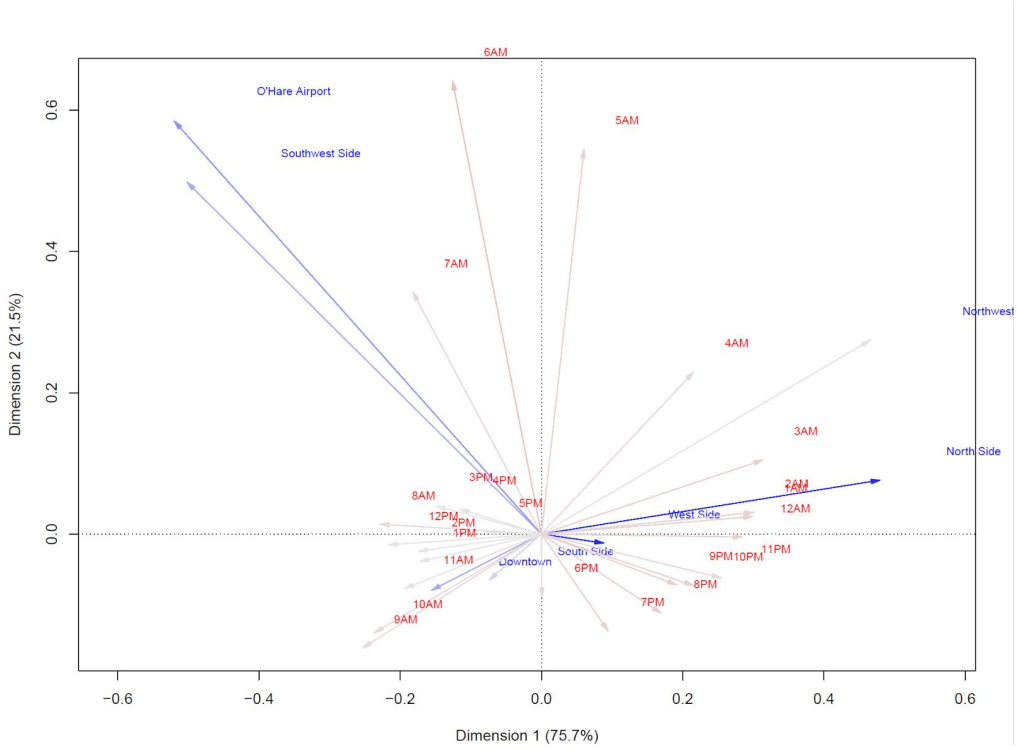


Correspondence Analysis:

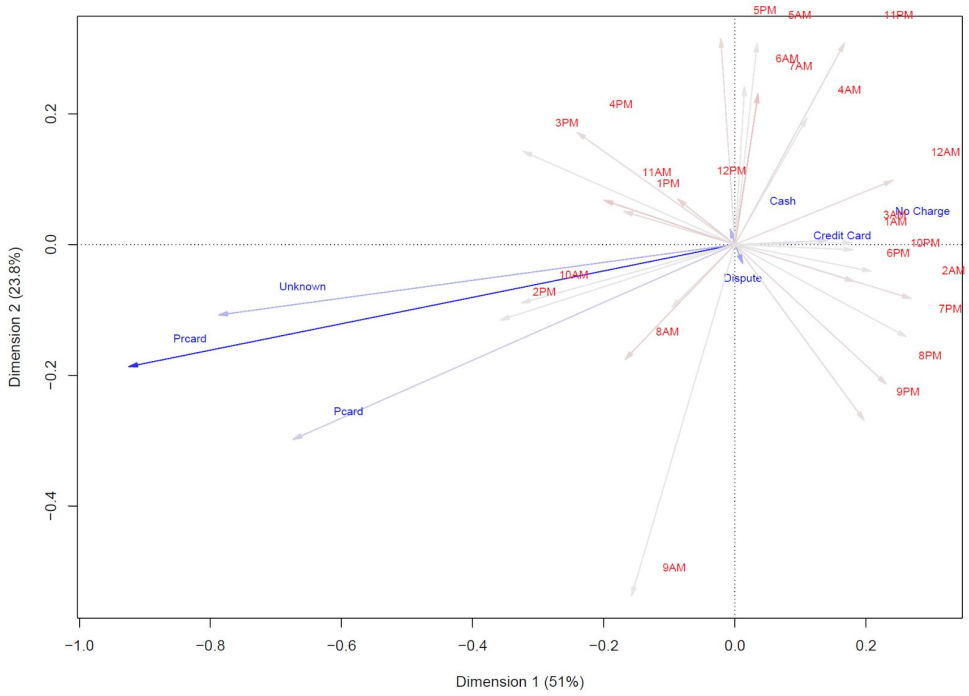
Pickup Neighborhoods and Time of Day

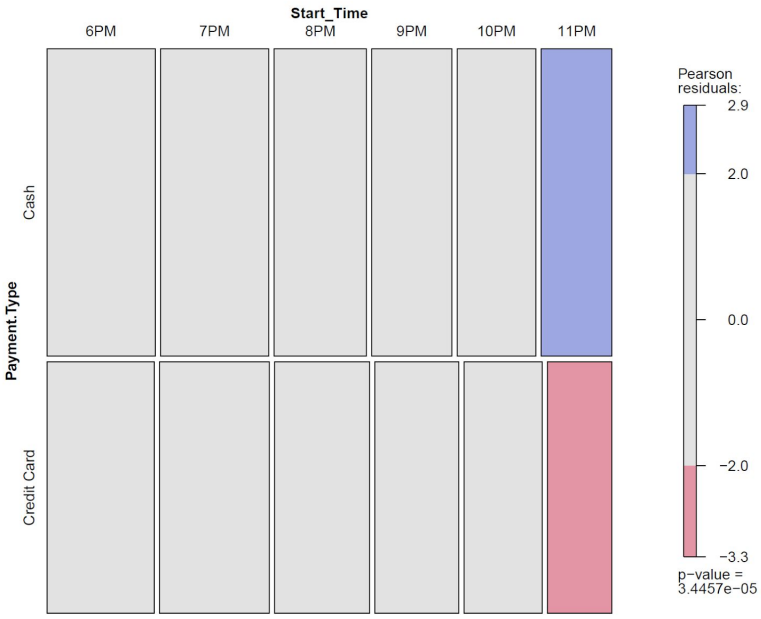


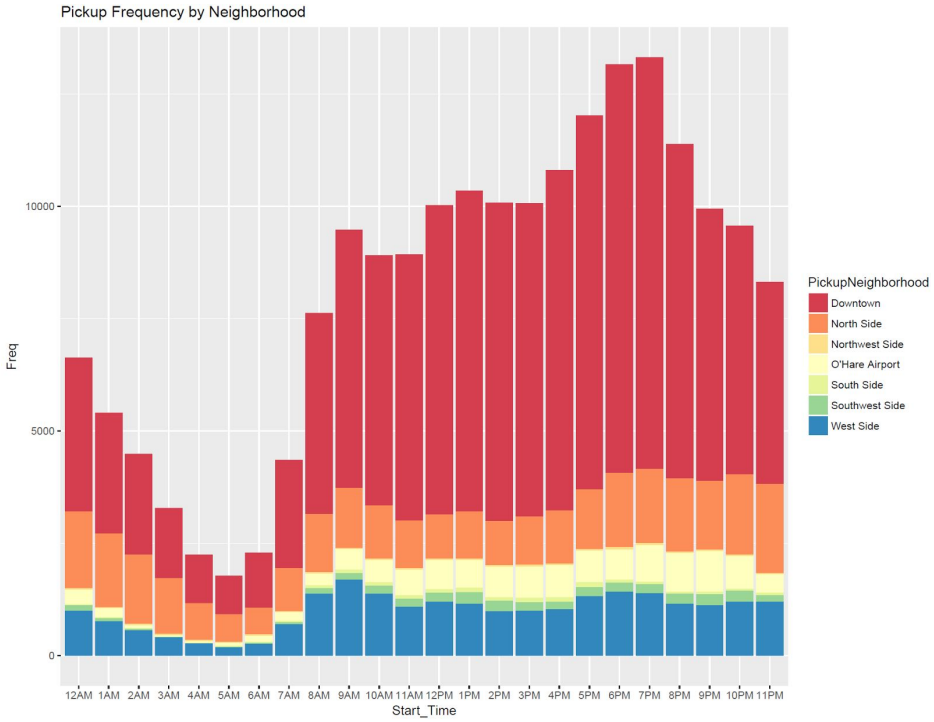
Dropoff neighborhoods and time of day

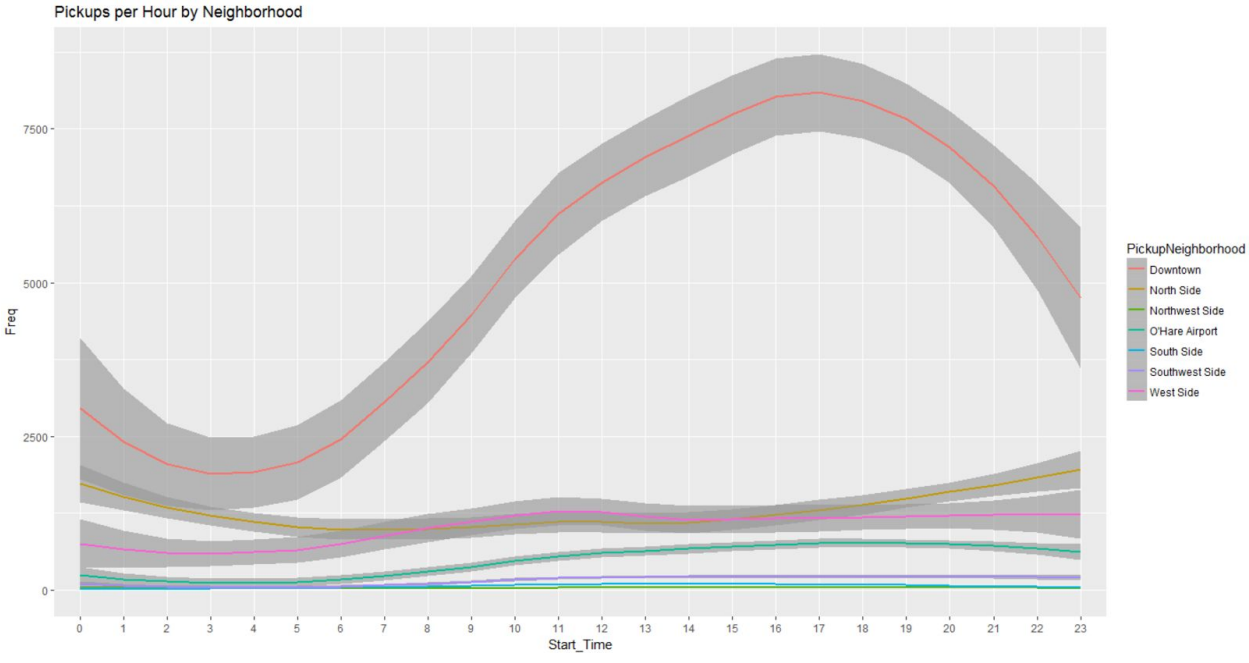


Payment type and time of day






Pickup Frequency by Neighborhood
Freq
10000
5000
0
12AM 1AM 2AM 3AM 4AM 5AM 6AM 7AM 8AM 9AM 10AM 11AM 12PM 1PM 2PM 3PM 4PM 5PM 6PM 7PM 8PM 9PM 10PM 11PM
Start_Time
PickupNeighborhood
Downtown
North Side
Northwest Side
O'Hare Airport
South Side
Southwest Side
West Side


Pickups per Hour by Neighborhood
Freq
7500
5000
2500
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Start_Time
PickupNeighborhood
Downtown
North Side
Northwest Side
O'Hare Airport
South Side
Southwest Side
West Side

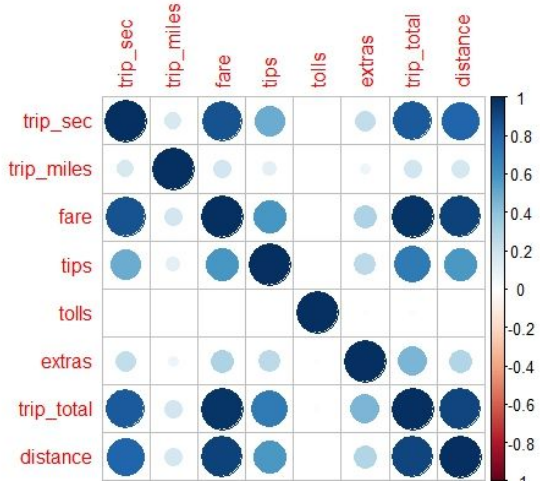
Data Conversion for missing values

Trip	Start	Timestamp	End
2/5/2016	11:00:00	PM	11:15:00
2/5/2016	11:00:00	PM	11:15:00
2/5/2016	11:00:00	PM	11:00:00

start_hours	start_mins	end_hours	end_mins	start_min	end_min	min_diff
23	00	23	15	1380	1395	15
23	00	23	15	1380	1395	15
23	00	23	00	1380	1380	0
23	00	23	00	1380	1380	0

<Correlation Matrix for Multiple Regression>

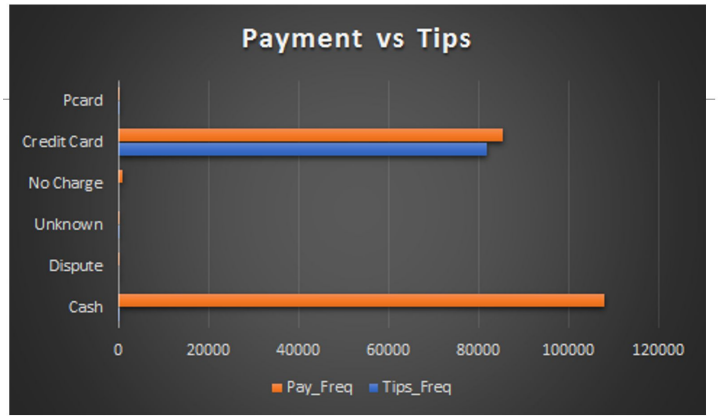
- Correlation Matrix of sub dataset for regression models
- Calculating mathematically the real distance using longitude and latitude.




```
In [49]: lat1 =df['Pickup Centroid Latitude']
         lat2 = df['Dropoff Centroid Latitude']
         long1 = df['Pickup Centroid Longitude']
         long2 = df['Dropoff Centroid Longitude']
```

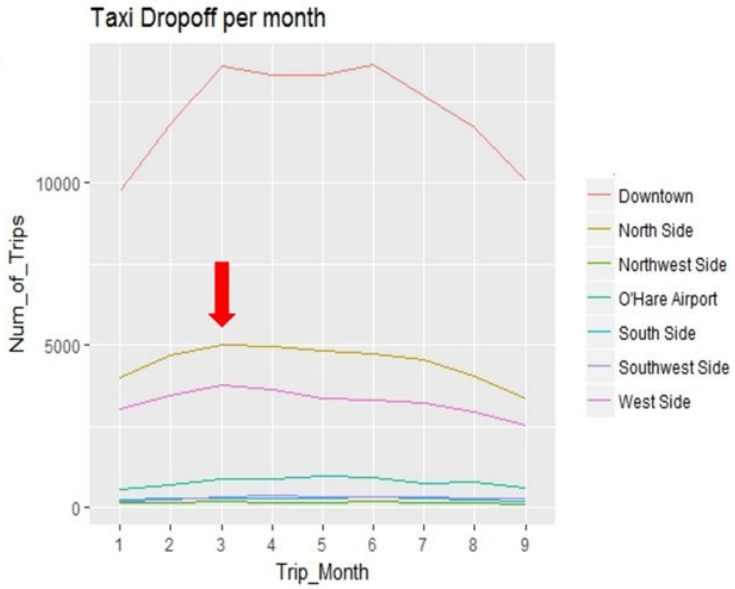
```
In [54]: dist = []
         for i in range(len(lat2)):

             p = 0.017453292519943295
             a = 0.5 - cos((lat2[i] - lat1[i])*p)/2 + cos(lat1[i] * p) * cos(lat2[i] * p) * (1 - cos((long2[i] - long1[i]) * p)) / 2
             dist.append(12742 * asin(sqrt(a)))
```

```
In [58]: df['Distance'] = pd.DataFrame(dist)
```



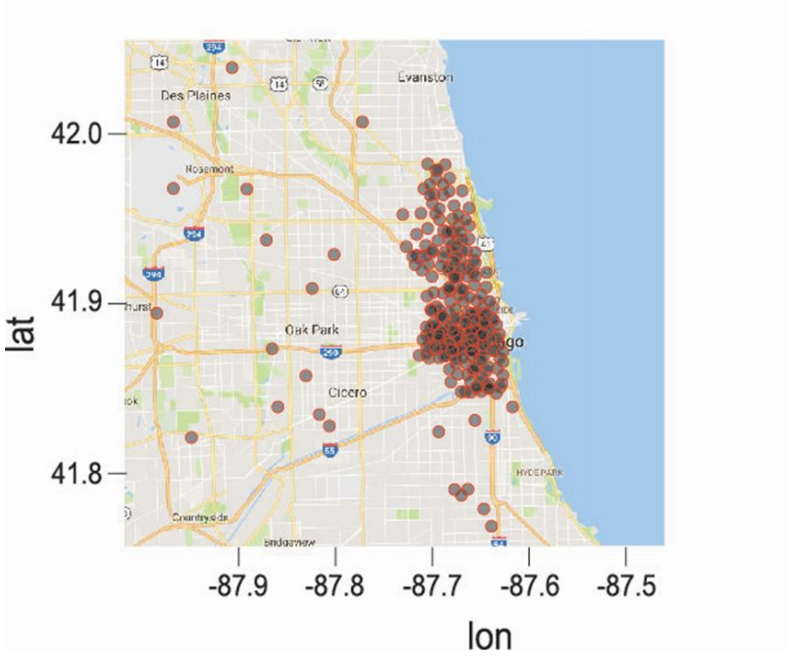
<Other Exploration>

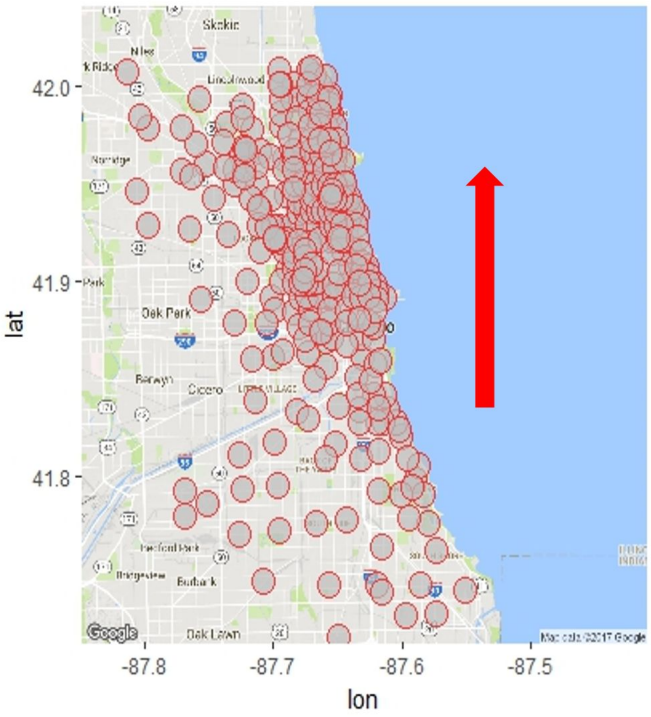
- Pick and Drop off per month



<Plotting the Seasonal Effect on Google Map>

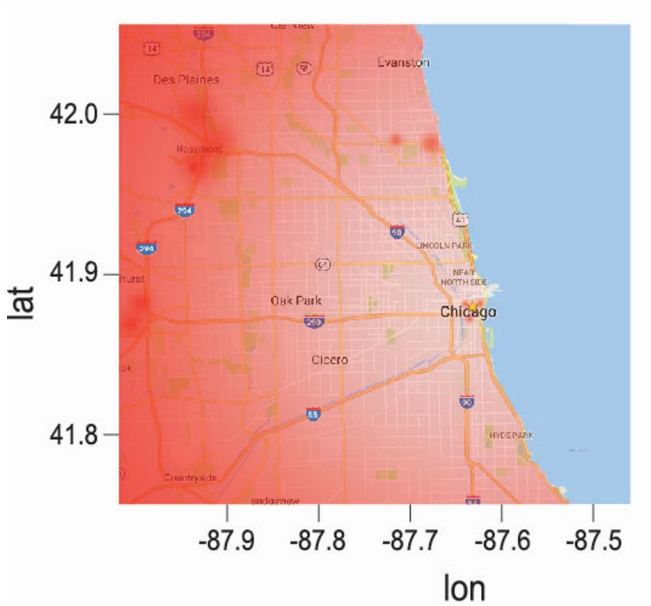
- Between March and August, the number of drop off increased around Northside (Wrigley Field)
- This job was done with longitude and latitude





<Heat map for the relationship between tips and (longitude and latitude)>

- The origin is downtown (where most people pick up and drop off the taxi)
- Latitude has positive relationship with Tips (going further west)
- Longitude has negative relationship with Tips (going further north)



Principal Component Analysis

components after varimax rotation:

```
Loadings:
                           RC1    RC3    RC2    RC4
Trip.Seconds                0.880
Fare                        0.936
Trip.Total                  0.936
Distance                    0.896
Pickup.Centroid.Latitude           0.883
Pickup.Centroid.Longitude  -0.554 -0.743
Dropoff.Centroid.Latitude                0.882
Dropoff.Centroid.Longitude              -0.768
Tolls                                           0.993
Trip.Miles
Extras
```

Model Performance(predicting tips):

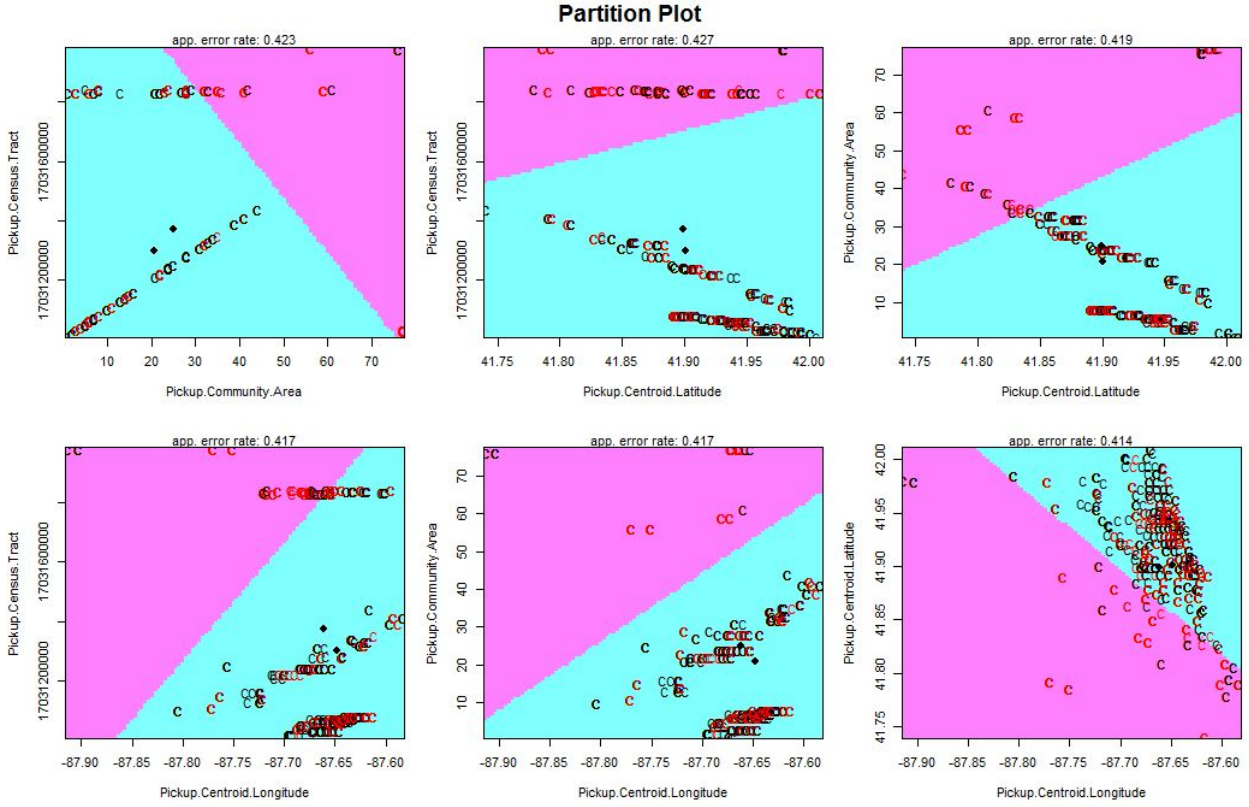
```
lm(formula = Tips ~ ., data = recast)

Residuals:
    Min      1Q  Median      3Q     Max
-12.185  -0.455   0.062   0.547 178.380

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.272217   0.006759  484.11   <2e-16 ***
RC1         2.165146   0.006759  320.32   <2e-16 ***
RC3         0.695592   0.006759  102.91   <2e-16 ***
RC2         0.322654   0.006759   47.73   <2e-16 ***
RC4         0.126674   0.006759   18.74   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.698 on 63131 degrees of freedom
Multiple R-squared:  0.6472,    Adjusted R-squared:  0.6472
F-statistic: 2.896e+04 on 4 and 63131 DF,  p-value: < 2.2e-16
```

Model Performance(predicting tips as a percentage of total):

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.20705 -0.02830  0.00111  0.03604  0.82435

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1857653  0.0002860 649.560  < 2e-16 ***
RC1         -0.0201276  0.0002860 -70.380  < 2e-16 ***
RC3         -0.0027985  0.0002860  -9.785  < 2e-16 ***
RC2         -0.0042900  0.0002860 -15.001  < 2e-16 ***
RC5         -0.0013005  0.0002860  -4.547 5.44e-06 ***
RC4          0.0002171  0.0002860   0.759    0.448
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.07186 on 63128 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.07739,   Adjusted R-squared:  0.07732
F-statistic:  1059 on 5 and 63128 DF,  p-value: < 2.2e-16
```

Linear Discriminant Analysis:

This graph illustrates the second LDA model, which only used four variables. Each plot shows a combination of two of these four variables. The line formed between the pink and blue areas illustrates where LDA drew it's best line in trying to differentiate the two classes. The red dots represent incorrectly labeled data points, and the black dots are correctly labeled data points.



Individual Reports

Don Crowley Individual Write-Up

For the last homework, we decided to reduce the size of the dataset to make it easier to work with. We took a random sampling of 20% of the data and performed our preliminary analyses from there. I wanted to make sure that our data was not fundamentally changed, so I found the means and IQRs of the original data set and compared them to the smaller dataset. After calculating these values, I was reassured that at least for the numerical values our smaller dataset was very similar to the initial dataset.

Most of my focus has been on the principal component analysis. The idea was to use the analysis to predict either tip or tip as a percentage of total fare with the numerical values. Because we only have tip values for payments made with credit cards, we had to reduce the data further.

I did an initial analysis with prcomp and created a scree plot to determine how many factors to use in the analysis. Five factors took us to about 86% and that was also where the "knee" appeared to be in our scree plot. The principal components from prcomp didn't appear to tell a story, so I decided to use a varimax rotation to rotate the components and see if any underlying groupings could be found.

The varimax rotation of the components worked well to help us interpret the components. Component one contained trip seconds, fare, trip total, distance, and pickup longitude. Except for longitude, these are the variables that would be most likely to go into a straightforward calculation of the cost of a taxi trip. The second component is the pickup location (latitude and longitude). The third component is the drop-off location (latitude and longitude). The fourth component is the trip in miles, and the fifth component is tolls. We can also note that the variable 'extras' was removed entirely.

Next I created a model using summated scaling. This model had an adjusted r squared of .524 and all the variables were statistically significant. I ran a cross validation with 75% of the data in the training set and 25% of the data in the testing set. The residual standard errors were 1.98 and 3.43 respectively.

It was interesting to see the real-world decisions needed to be made for data cleansing. Initially we missed that cash transactions were missing tips. We discussed the idea of trying to fill in tips based on a similar data from credit card transactions. Because tips were our y-variable, this didn't seem like the right decision. There was really no good decision here, and our analysis was only done on credit transactions. As far as the real world application, we would have to hope that the taxi drivers had some idea of if and how cash-paying customers tipping behavior differed from that of the credit card paying customers. One of the main weaknesses of this model is that we are leaving out all of the categorical variables.

Becky Jacob - Individual Write-Up

The following brief report will outline two main areas. First, the summary of the work I personally completed as a part of the final project, including the major roles I played as well as the specific analysis I performed. Secondly, I will summarize the aspects of multivariate analysis and statistics that I am taking away from this class.

Summary of Work

Major roles played

There were two main roles I played in this group project. The first being, data discovery and cleaning. When looking for a dataset, I tried to find something that could apply to the variety of skillsets we learned in this class, while also being a relevant and interesting topic, and not too overwhelming as we were on a deadline. I recommended five different datasets, one from kaggle, and four from the Chicago City data portal. I was also heavily involved in the initial cleaning of the data. I feel most comfortable in Python, so I used that to remove dollar signs from all monetary columns and convert them to numeric values, verify and change other data types that imported incorrectly, evaluate nulls and remove some rows based on this evaluation, drop unnecessary columns, and finally remove extreme outliers based on Trip Seconds being outside of 3 standard deviations from the mean.

Specific Analyses

Initially I performed a CFA on the taxi dataset, but it was not particularly helpful and showed us something very similar to what the PCA team discovered, so it was dropped from the project. As a follow up I was tasked with performing an LDA to try and predict the Payment Type (Cash, Credit, Dispute, etc.) for each taxi trip. Initially, I was curious just how accurately LDA could predict this metric, with all available data. However, after completing this initial LDA, I took a step back to look at the analysis from a real life perspective. If a cab driver was going to use this model, she/he would not have access to all of the post-ride data. I therefore, performed a second LDA, which only utilized the features available at the point of the pickup. This, not surprisingly, had a much lower performance, and would only slightly outperform using the base percentage split.

Summary of Takeaways

I think my biggest takeaways from this class have been the many different ways of dealing with categorical variables. Prior to this class I had a general idea of the purpose of PCA, and it is quite useful, but I was already capable of producing a model to try to predict a numeric dependent variable. However, if my dependent variable was anything other than a binary categorical variable I had no real way of trying to model and predict. Now I have several tools available to me as well as a better understanding of dealing with very large datasets and how to make them more manageable.

In addition, learning to appreciate the bias and variance tradeoff was crucial. Especially, when you move from a very complicated but accurate model, to a slightly less accurate but interpretable model. Having worked prior to attending graduate school, I can recall experiences, where accuracy was less important, but being able to tell a customer concisely about their product, was crucial.

James Cooper - Individual Write-Up

The data contained over 1 million rows with variables including time of pickup and drop off, location data (latitude and longitude as well as community area), payment type, as well as metric variables relating to payment amounts and cab fare. Our goal was to look at this data from a number of different perspectives to find interesting results, but with an overarching theme of finding patterns in tip amounts and predicting what factors will lead to the largest tips. In order to do this, we needed to come at the data from a number of different ways utilizing the methods we learned over the course of the quarter.

The original dataset was a bit messy with a number of missing values as well as some redundant variables such as community centroid area - these were variables that were duplicated in other variables. We decided to remove these first, then move on to missing values. Since the dataset contained so many rows, we decided to take a smaller sample of the data in order to give us a set that was a little bit easier to work with. We took a random 20% sample from the data. When we had the final set, we split up some of the cleaning amongst the group. I specifically took a look at parsing the time data and turning the times into a 24 hour cycle from 0-23. We then had both the date and specific hour of the day for all rows. Another area of missing values was the drop off area. For this I plotted the pickup area against the drop off area using a jitter method to compare where the most correlated drop off areas were with pickup areas (see plot in graph section). In this way, I was able to replace the missing drop off areas with the darkest corresponding plots.

We also had missing values for payment type. I used the same method to examine the relationships with a variable that we had near no missing values for (Pickup Neighborhood). The result was a little bit less clear but served as somewhat of an insight into where the missing values may be replaced, though I decided against this since the correlations were not nearly as clear. There was a near even split for Downtown, North Side, and West Side with Cash and Credit Card.

For the analysis portion, I decided on examining some relationships with the categorical variables using Correspondence Analysis. Because we wanted to focus on tips and fare, I was mainly interested in what times of day people use taxis the most as well as what neighborhoods these times are most associated with. I also thought it might be useful for drivers, especially for tips, whether people used cash or credit the most during certain times of day. Overall, these analyses would provide a guideline for companies to concentrate cabs during certain times of the day.

The first analysis I did divided the neighborhoods into more broad areas – Downtown, North Side, South Side, West Side, Northwest Side, Southwest Side, and O'Hare. This was to get a more general idea of the community area in order to have a broad idea where the cabs were concentrating. From there I could further examine the most popular areas and break down the neighborhoods more specifically. Much of the rides were unsurprisingly Downtown during the

day, but there was a little more variation during later hours. I then examined community areas 1-77 to get an idea of which specific neighborhoods (Logan Square, Albany Park, Lincoln Park, etc.) would give the most pickups/drop offs.

The correspondence analysis gave us a good picture of where cabs most frequent – or more specifically where the demand is highest throughout the city of Chicago on a given day. The first dimension gave us the most variance and we could easily confirm from the data that the downtown area is most active during business hours, roughly 8am-5pm. The North Side area has the most demand in the later hours, specifically evening and early morning, the plot gives the 11pm – 6am as the lines with the most acute angles to the North and Northwest Side lines. The South and Southwest Sides were more acute to the morning/afternoon hours as well as O'Hare. It was interesting how strong the correlation to these times appears for the airports.

I also performed a correspondence analysis on the times of day vs. the payment type to examine when tips might be maximized, as well as neighborhoods vs. cab company. These showed both the times of day when cash is preferred and the neighborhoods that most taxi company's frequent or reside in. The analysis for the payment type give us a few significant times for cash occurring early in the morning and late at night, whereas the credit card seems to be a preferred method around 9am. The cab company analysis gave a snapshot of where different cab companies of Chicago most congregate. This was meant as more of an exploratory analysis, my thinking was that it could be useful for cab companies to examine where competition makes most of their profit in order to strategize location.

The graphs I used to illustrate this analysis used the rowcoord and colcoord fields from the 'ca' package giving the dimensions of the data. I plotted multiple mosaic plots for the significant number of observations for times of day vs. pickups, drop-offs, and payment type, as well as the 'caplot' that gave the lines of the variables in order to examine the acuteness of the angle giving the relationship between pickup and dropoff neighborhood vs. time, as well as cab company and neighborhood. The last part of my analysis involved time series plots in which I used the 'ts' function in R to plot the busiest times of day and the neighborhoods with most demand. I used ggplot to also graph a bar plot showing this as well. The 'ts' function in R also gives a breakdown of the most significant times using the auto regression 'as' as well as the maximum likelihood estimate 'mle' functions.

The biggest takeaway I had from this class is that I now have a large number of tools to deal with data that contains both metric and categorical variables, multicollinearity, and the fact that there are so many different ways to deal with issues in data cleaning and analysis. I think this really opened up my eyes to examining data using many different methods and comparing the results. Using linear algebra in multivariate regression to handle multicollinearity was an extremely valuable thing to learn with PCA and CFA. Also dealing with categorical data with correspondence analysis and LDA gave me a way to deal with categorical variable prediction that was somewhat of a mystery to me before this class. Overall, this class exposed me to a number of methods that will be invaluable to multivariate statistics.

Jonggoo Kang Individual Write-Up

For the data cleaning part, I detected missing values in each column using python, and drop them. Thus, I got new data set with 35000 rows from the row data set with 1 million rows. Also, I realized that zero values in the columns of 'Fare', 'Trip Second', and 'Trip Total' ruined our hypothesis which is predicting 'Tips'. Therefore, I removed the zero values in the dataset too. Furthermore, I add another numerical variable called 'Distance' to check real distances. I wrote a program to calculate the distances from the values of 'Latitude' and 'Longitude' using python.

For the analyses, I checked both 'Multiple Regressions' and 'CFA' at the first time with two other members. The result from the Regressions was good, except R-squared scores. As I mentioned first paragraph, zero values in the three variables were making troubles for multiple regressions. At the same time, I realized it did not make sense that the driver received tips even though he did never go trip. Therefore, we decided to drop zero values in those three variables too. I also helped other member to conduct 'log normalization' for some variables for Multiple Regression. For my second analysis, I conducted CFA. I did not meet with a good result from CFA, but it was not the terrible result. I met with two main factors. One factor gave me groups of 'Pickup longitude and latitude' and 'drop-off longitude and latitude', which could be a Location factor. Other factor was a group of 'Fare, Tolls, and Extras' which could be a factor represented as Money. After our group decided to drop CFA, I started PCA for my next analysis. In PCA, the scree plot and cumulative proportion of variance suggested five principal components. However, five PCs were not good options to draw a good conclusion. In the first component, I had 'Trip second, fare, trip total, and Distance' with 'pickup longitude'. For me, 'pickup longitude' should not be in this component. Also, it is included the other component at the same time. To solve this problem, I chose four PCs as professor recommended. It gave me better results. In the first component, it describes a Taxi trip and 'Trip second, fare, trip total, and Distance' are included in the component. Second component describes a pickup latitude and longitude, and third one shows a dropoff latitude and longitude. Lastly, last one shows a Tolls. These four components finally much more make sense.

The takeaway from this class is that I am not afraid of dealing with a large dataset. When I first faced with the original dataset, I had no idea what to do because R even did not work with it. Now, I achieved knowledges of how to manipulate large numbers of dataset. Furthemore, I realized that it is very important to understand domain knowledge and knowing what my dataset consists of. Our group had hard time to clean the dataset. Each member applied so many ways to clean it for almost more than 6 weeks. Therefore, I could say dealing with datasets is my first takeaway. In addition, I think CFA and PCA are very useful. When you do not know about your dataset, I think these two are first step you should conduct. After you catch the factors, you could develop your analysis with other statistical method for your next step. Lastly, reviewing eigenvectors and eigenvalues are very helpful for me. I did not know how important they are. By understanding them, my narrow views of interpreting the results are now getting wider, which I am very happy with.

Ryan Yong Woo An Individual Write-Up

<Data Cleansing>
The goal of our project was to crack, explore, and visualize the interesting facts from the dataset in different perspectives. I have involved data cleansing, analyzing, and discovering the interesting facts from our dataset. Initially, our dataset had more than 1M rows (exactly, 1,261,589 rows) with a lot of missing values, especially the data related to money: Fare, Total Amount, Tips, Tolls, and Extras.

To keep the data as much as possible and fill the missing values, the first thing I did was data cleansing. I removed the dollar sign from amount data and converted to numeric, and pull the hour and min data from the trip start time and end time to convert to minutes using SQL. After that, I added 12 if the trip was in the afternoon to make the trip time scale between 1-24. Secondly, I added a new column trip distance after calculating the real distance mathematically using longitude and latitude.

I randomly pulled 70% of non-missing data from original dataset with replacement to see if there's any delta linear relationship or with convexity relationship between fare, trip distance, and trip time each other. I repeated couple times but could not find the right coefficient for each variable, so I filled the N/A values based on the information from Chicago Taxi website (https://www.cityofchicago.org/city/en/depts/bacp/supp_info/2012_passenger_information.html). Thus, our team decided to remove the data using one of two statistical method. One way was to use IQR and another way was to use standard deviation. However, when we compared the number of rows we need remove using of those methods, standard deviation was better to keep more data. Finally, we narrowed down to around 20% from original dataset, which is 215,441 rows.

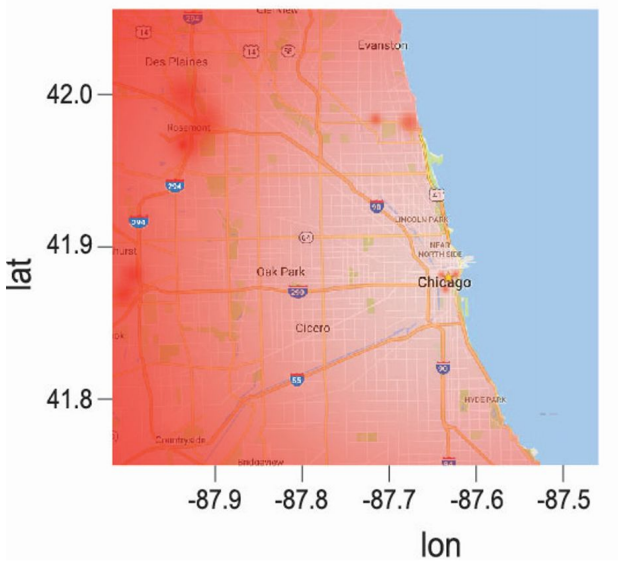
<Analysis>
For Analysis, I have worked on both Multiple regression and PCA but more on regression. I was trying to predict the percentage of the tip amount and identify which variables are correlated. To build a regression model, I used six factors as explanatory variables which are the most relevant to tip amount: Fare, Tolls, Extras, Travel Time, Travel Miles, distance using longitude and latitude.
To do this, I created a sub dataset by pulling the data within three standard deviation from the median instead of mean. It was because the data was strongly positively skewed especially from fare, tips, and extra (Min: 0, Max: 124,200, Mean: 64.47, Median: 17.4). I also created correlation matrix to see if there's any multicollinearity from the sub-dataset. I reached the conclusion after making several attempts using different models and mutation of the variables. Cleaning our data and adding, removing, or replacing other explanatory variables from our models still produced no significant results. The explanatory variables for the best model were Trip Seconds, Trip Miles, and Fare with 0.36298 (around 36.3%) R-squared.
For better analysis, I tried to find the reasons why goodness of fit wasn't great, then I found that there were couple different facts that could not be quantified. Refer to the image (payment vs Tips) below, people pay with cash the most, but it looks like the only tips paid by credit card were reported. I guessed that the taxi drivers didn't report honestly or there might

be data entry issues due to legacy which could mislead to find the relationship between tips and other factors.

	New_payment	Tips_Paid	Payment
1	Cash	71	108036
2	Dispute	0	138
3	Unknown	111	339
4	No Charge	0	858
5	Credit Card	81937	85395
6	Pcard	6	90

<Other Exploration (Unsupervised)>
I believe that there should be any trend or seasonal effect, so rather than removing all the categorical data, I tried to explore anything interesting. First, I wanted to know if there is any relationship between tip amount and pickup longitude with latitude, so I created a sub dataset by pulling tip amount, pickup longitude and latitude, and drop-off longitude and latitude data only to create heat map to visualize it. The conclusion was that the latitude was positively, and the longitude was negatively correlated with tip amount, and more tips can be predicted if the destination is further West and North from downtown area.

Furthermore, after James's time series, I wanted to dig the ground deeper. I created the time series the pickup and drop off change per month and per day. I found that there was a seasonal effect between March and August. To do this, I got the delta of the number of drop-off of each month by community areas and sliced the dataset by every 6 months to calculate the average number of drop off. The number of taxi pickup and drop-off in downtown was the most, but the delta of drop-off between March and August was the highest around Northside (around Wrigley Field). I concluded that there was seasonal effect which is Cubs home games.



<Lesson>
The first lesson I learned that the R squared for real data set cannot be as high as the examples we used in the class. In my opinion in those situations, even there was quantified data, the metric is somewhat meaningless. Stochasticity probably dominates the underlying behavior rather than common cause variation. However, since we are analyzing the real-world data, I could guess that R squared isn't really a practical metric because it only gives us information about goodness of fit, not goodness of prediction. Also, if there is human behavior is involved in the dataset, the prediction may not be accurate and cannot be explained numerically.

Code

A. Multiple Regression Analysis

```
IF OBJECT_ID('tempdb..#new_taxi') IS NOT NULL
DROP TABLE #new_taxi


SELECT
ID = [Column 0],
Trip = LEFT(Trip, LEN(Trip)-5),
Start = [Start],
Timestamp = [Timestamp],
Timestamp_num = case when timestamp = 'AM' then 0 else 1 end,
[End],
Trip_Seconds = [Trip Seconds],
Trip_Miles = [Trip Miles],
Fare = CAST(REPLACE(REPLACE(ISNULL(Fare,0),',',''),'$','') AS float),
Tips= CAST(REPLACE(REPLACE(ISNULL(Tips,0),',',''),'$','') AS float),
Tolls = CAST(REPLACE(REPLACE(ISNULL(Tolls,0),',',''),'$','') AS float),
Extras= CAST(REPLACE(REPLACE(ISNULL(Extras,0),',',''),'$','') AS float),
Total_Cost = CAST(REPLACE(REPLACE(ISNULL([Trip Total],0),',',''),'$','') AS float),
Payment_Type = [Payment Type],
Pay_Type = case when [Payment Type] = 'cash' then 0 else 1 end,
[Company]
into #new_taxi
FROM [dbo].newtaxi


IF OBJECT_ID('tempdb..#new_taxi2') IS NOT NULL
DROP TABLE #new_taxi2

select
*,
start_hours = case when timestamp = 'PM' then Substring(start, 1,Charindex(':', start)-1) +12 else Substring(start, 1,Charindex(':', start)-1) end,
start_mins = Substring(start, Charindex(':', start)+1,2),

end_hours = case when timestamp = 'PM' then Substring([end], 1,Charindex(':', [end])-1) +12 else  Substring([end], 1,Charindex(':', [end])-1) end,
end_mins = Substring([end], Charindex(':', [end])+1,2)
into #new_taxi2
from #new_taxi



IF OBJECT_ID('tempdb..#new_taxi3') IS NOT NULL
DROP TABLE #new_taxi3

select
*,
start_min = start_hours * 60 + start_mins,
end_min = end_hours * 60 + end_mins,
min_diff = (end_hours * 60 + end_mins) - (start_hours * 60 + start_mins)

--into #new_taxi3
from #new_taxi2
```

```
data = read.csv('Taxi_V10_credit_reg.csv', header = T)

#Z-Score (Standardized)
new_data = as.data.frame(scale(data))
new_data

#Building Regression
model = lm(tips ~ ., data = new_data)
summary(model)

#Stepwise - Forward
step_f = step(model, direction = 'forward')
summary(step_f)
#Stepwise - Backward
step_b = step(model, direction = 'backward')
summary(step_b)
step_b2 = step(model, direction = 'both')
```

```
summary(step_b2)


vif(model)

#Correlation
corrplot(cor(new_data), method= 'ellipse', order='AOE')

#Correlation Test
corrtest = corr.test(new_data,adjust = 'none')
corrtest
a = cor(new_data)
corrplot(a, method = 'number')
test = ifelse(corrtest$p < 0.1, T, F)
(colSums(test)-1)
```

Plotting on the Map

```
library('ggmap')
library('ggplot2')
library('ggmap')
library('ggmap')
library('ggthemes')
library('plotGoogleMaps')

setwd('C:/Users/RADK/Desktop/School/Spring17/CSC424/Project')

lat_long = read.table('long_lat2.csv', sep = ',', header = T)
new_taxi = read.table('New_taxi2.csv', sep = ',', header = T)

head(new_taxi)

lat = c(lat_long[1])
long = c(lat_long[2])
value = c(lat_long[3])

#get Chicago map
chicago <- get_map(location = 'chicago', zoom = 11)
ggmap(chicago)

map_data
#vary the color of the data points
chicago <- get_map(location = 'chicago', zoom = 11, style = 'satellite')
map_data_coloured <-
  ggmap(chicago) +
  geom_point(data=lat_long,
             aes(x = as.numeric(as.character(long)), y = as.numeric(as.character(lat)), size = sqrt(value)), size=5,
alpha=I(0.5)
             ,xlab = 'longitude', ylab = 'latitude')
 # scale_y_continuous(limits=c(0,1000))+
 # scale_x_continuous(limits=c(0,1000))+
  + geom_smooth(method="lm")

map_data_coloured
```

Time Series

```
##Pickup
time = read.csv('time_series.csv')
head(time)
time$Trip_Month <- factor(time$Trip_Month, levels = c( 1,2,3,4,5,6,7,8,9 ) )
pick_mon = qplot(Trip_Month, Num_of_Trips, data = time, geom = 'line', group = PickupNeighborhood,
    color = PickupNeighborhood, main = 'Taxi Pickup per month')


#Dropoff
drop = read.csv('drop_time.csv')
head(drop)
drop$Trip_Month <- factor(drop$Trip_Month, levels = c( 1,2,3,4,5,6,7,8,9 ) )
drop_mon = qplot(Trip_Month, Num_of_Trips, data = drop, geom = 'line', group = dropoff,
          color = dropoff, main = 'Taxi Dropoff per month')

#dropoff by day
drop2 = read.csv('drop_time_day.csv', header = T)
head(drop2)

drop_day = qplot(days, Num_of_Trips, data = drop2, geom = 'line', group = drop,
    color = drop, main = 'Taxi per day')

drop2$days <- factor(drop2$days, levels = c( "Mon", "Tue", "Wed","Thurs","Fri","Sat", "Sun" ) )
drop_day = qplot(days, Num_of_Trips, data = drop2, geom = 'line', group = drop,
    color = drop, main = 'Taxi dropoff per day')

#pickup by day
pick = read.csv('pick_time_day.csv', header = T)
head(pick)
pick$days <- factor( pick$days, levels = c( "Mon", "Tue", "Wed","Thurs","Fri","Sat", "Sun" ) )
pick_day = qplot(days, Num_of_Trips, data = pick, geom = 'line', group = pickup,
    color = pickup, main = 'Taxi Pickup per day')


drop_day
```

B. Principal Component Analysis

```
data = read.csv("Taxi_V10_credit.csv",  header = T)
head(data)
# narrowing down to the numerical data
#seconds, miles, fare, tip, toll, extras, pickup lat, pickup long, dropoff lat, dropoff long
c4 = c( 10,11, 19, 20, 21, 22, 23, 26, 27, 28, 29,30)
data2 = data[c4]
head(data2)
data2$Trip.Total = data2$Fare + data2$Tips + data2$Extras
#replacing any division by zero with actual zeroes
data2$TipsPerTotal = if (data2$Trip.Total == 0) {
 0
```

```
} else {
  data2$Tips/data2$Trip.Total
}

c5 = c(1,2,3,5,6,7,8,9,10,11,12  )
data3 = (data2[c5])
sapply(data3, class)
head(data3)
cor(data3)
#latitude is type 'factor' for some reason
data3$Pickup.Centroid.Latitude = as.numeric(data3$Pickup.Centroid.Latitude)

is.nan.data.frame <- function(x)
  do.call(cbind, lapply(x, is.nan))
data3[is.nan(data3)] <- 0
library(psych)
p = prcomp(na.omit(data3), scale = T)
p
summary(p)
plot(p)
abline(1,0)

data3 = read.csv("Taxi_V10_credit.csv", header = T)

Total_row = nrow(data3)
Train = round(nrow(data3) * 0.75, 0)
size = sample(Total_row, size = Train)
xtrain = data3[size,]
ytrain = data3[size,]
xtest = data3[-size,]
ytest = data3[-size,]

head(xtrain)
cx = c( 10,11, 19, 21, 22, 23, 26, 27,28,29, 30)
cy = c(20,23)
xtrain = xtrain[cx]
xtest = xtest[cx]
ytrain = ytrain[cy]
ytest = ytest[cy]
head(xtrain)
head(ytrain)

summatedtrain = data.frame(ytrain[1],xtrain)

head(summatedtrain)

head(ytrain)
head(xtrain)
p2 = psych::principal(xtrain, rotate = "varimax", nfactors = 4, scores = T, covar = F)

print(p2$loadings, cutoff = .5, sort = T)

fit2 = psych::principal(xtrain, rotate = "varimax", nfactors = 4, scores = T)
```

```
recast = as.data.frame(fit2$scores[,1:4])
nrow(recast)
nrow(summatedtrain)
recast$Tips = summatedtrain$Tips
model = lm(Tips~., data = recast)
summary(model)

head(recast)

model = lm(Tips~., data = recast)
summary(model)
yhat = predict(model, data = xtrain)
rse_train = sqrt(sum((ytrain - yhat)^2) / (length(ytrain)))
rse_train

ypredict = predict(model, data = xtest)
rse_test = sqrt(sum((ytest - ypredict)^2) / (length(ytest)))
rse_test
```

C. Correspondence Analysis

```
############ PAYMENT TYPE AND TIME OF DAY ###################

install.packages("wesanderson")
library(wesanderson)
install.packages("circular")
library(circular)
library(ggplot2)
library(corrplot)
library(Amelia)
library(RColorBrewer)
library(dplyr)
library(ca)

Taxiset2 = read.csv("Taxi_V10 - FINAL", header = TRUE, sep = ",")
head(Taxiset2)
summary(Taxiset2)

#Make Contingency Table from Taxiset2
ctable = with(Taxiset2, table(Payment.Type, Start_Time))
ctable2 = as.table(ctable)

colnames(ctable2) = c("12AM", "1AM", "2AM", "3AM", "4AM", "5AM", "6AM", "7AM", "8AM","9AM",
            "10AM", "11AM", "12PM", "1PM", "2PM", "3PM", "4PM", "5PM", "6PM",
            "7PM", "8PM", "9PM", "10PM", "11PM")
ctable2

z = as.data.frame(ctable2)
z

#Plot Times vs. Payment Type
plot1 = ggplot(z, aes(x = Start_Time, y = Freq, fill = Payment.Type)) +
  labs(title = "Payment Type per Hour") + geom_bar(stat = "identity")
plot1 + scale_fill_brewer(palette = "Spectral")
```

```
#Barplot for CA table
barplot(ctable, legend = T, beside = F)

#Mosaic plot for Payment Type vs. Time of Day
mosaic(ctable2[1:2,1:6], shade = TRUE, legend = TRUE)
mosaic(ctable2[1:2,7:12], shade = TRUE, legend = TRUE)
mosaic(ctable2[1:2,13:18], shade = TRUE, legend = TRUE)
mosaic(ctable2[1:2,19:24], shade = TRUE, legend = TRUE)

#Transpose table so times are on the side
ctranspose = t(ctable2)
ctranspose
mosaic(ctranspose, shade = TRUE, legend = TRUE)

#Write the Contingency Table to a .csv file
write.csv(ctable, file = "ctable.csv")

cataxi = ca(ctable2)
cataxi$N
cataxi$rowcoord
cataxi$colcoord

#Plots using ca dimensions 1 and 2
plot(cataxi)

plot(cataxi, mass=T, contrib="absolute",
    map="rowgreen")

plot(cataxi, mass=T, contrib="absolute",
    map="rowgreen", arrows=c(T, T))


   ############  CORRESPONDENCE COMMUNITY AREA AND TIME OF DAY  ##############

Taxihood = read.csv("Taxi_Neighb.csv", header = TRUE, sep = ",")


Taxihood2 = read.csv("Taxi_V10 - FINAL", header = TRUE, sep = ",")
head(Taxihood2)
glimpse(Taxihood2)
summary(Taxihood2)
names(Taxihood2)
summary(Taxihood2)
range(Taxihood2$Tips)

plot(Taxihood2$Trip.Miles, Taxihood$Fare, ylim = c(0,100))

#Create Correspondence Table for Start Times

taxitable = with(Taxihood2, table(PickupNeighborhood, Start_Time))
names(taxitable)
head(taxitable)
class(taxitable)
```

```
colnames(taxitable) = c("12AM", "1AM", "2AM", "3AM", "4AM", "5AM", "6AM", "7AM", "8AM","9AM",
          "10AM", "11AM", "12PM", "1PM", "2PM", "3PM", "4PM", "5PM", "6PM",
          "7PM", "8PM", "9PM", "10PM", "11PM")
taxitable
x = as.data.frame(taxitable)
head(x)

barplot(taxitable, legend = T, beside = F)

#Barplot for neighborhoods and times
plot1 = ggplot(x, aes(x = Start_Time, y = Freq, fill = PickupNeighborhood)) +
 labs(title = "Pickup Frequency by Neighborhood") + geom_bar(stat = "identity")
plot1 + scale_fill_brewer(palette = "Spectral")
mosaic(taxitable, shade = TRUE, legend = TRUE)

cahood = ca(taxitable)
cahood$N
cahood$rowcoord
cahood$colcoord

#Plots with ca dimensions 1 and 2 pickup time vs. pickup neighborhood
plot(cahood)
plot(cahood, mass=T, contrib="absolute",
   map="rowgreen")

plot(cahood, mass=T, contrib="absolute",
   map="rowgreen", arrows=c(T, T))


#Create Correspondence Table for End Times
taxitable2 = with(Taxihood2, table(DropoffNeighborhood, End_Time))
taxitable2
colnames(taxitable2) = c("12AM", "1AM", "2AM", "3AM", "4AM", "5AM", "6AM", "7AM", "8AM","9AM",
                 "10AM", "11AM", "12PM", "1PM", "2PM", "3PM", "4PM", "5PM", "6PM",
                 "7PM", "8PM", "9PM", "10PM", "11PM")
taxitable2

#Quick barplot for contingency table
barplot(taxitable2, legend = T, beside = F)

g = as.data.frame(taxitable2)
g


#Barplot for neighborhoods and times
plot2 = ggplot(g, aes(x = End_Time, y = Freq, fill = DropoffNeighborhood)) +
 geom_bar(stat = "identity") + labs(title = "Dropoff Frequency by Neighborhood")
plot2 + scale_fill_brewer(palette = "Spectral")
mosaic(taxitable2, shade = TRUE, legend = TRUE)

cahood_end = ca(taxitable2)
cahood_end$N
cahood_end$rowcoord
cahood_end$colcoord
```

```
#Plots with ca dimensions 1 and 2 dropoff times and dropoff neighborhoods
plot(cahood_end)
plot(cahood_end, mass=T, contrib="absolute",
    map="rowgreen")

plot(cahood_end, mass=T, contrib="absolute",
    map="rowgreen", arrows=c(T, T))


 ##### CORRESPONDENCE WITH ALL NEIGHBORHOODS ######

taxitable_all = with(Taxihood2, table(Pickup.Community.Area, Start_Time))

cafull = ca(taxitable_all)
cafull$N
cafull$rowcoord
cafull$colcoord

#Plots with ca dimensions 1 and 2
plot(cafull)
plot(cafull, mass=T, contrib="absolute",
    map="rowgreen")

plot(cafull, mass=T, contrib="absolute",
    map="rowgreen", arrows=c(T, T))


taxitable_all2 = with(Taxihood2, table(Dropoff.Community.Area, End_Time))

cafull2 = ca(taxitable_all2)
cafull2$N
cafull2$rowcoord
cafull2$colcoord

#Plots with ca dimensions 1 and 2
plot(cafull2)
plot(cafull2, mass=T, contrib="absolute",
    map="rowgreen")

plot(cafull2, mass=T, contrib="absolute",
    map="rowgreen", arrows=c(T, T))


##### TIME SERIES #####

taxitable
transpose = t(taxitable)
transpose
tstaxi = as.ts(transpose)
tstaxi
class(tstaxi)
is.ts(tstaxi)
cycle(tstaxi)
```

```
plot(tstaxi)
boxplot(tstaxi~cycle(tstaxi))
acf(tstaxi)
decompose(tstaxi)

#Graphing Start Times and End Times

qplot(y=Taxiset2[,5], x=1:length(Taxiset2[,5]), fill=factor(1:length(Taxiset2[,5])),
    stat='identity', geom='bar') + coord_polar()

#Time Series for Start Time
qplot(Start_Time, Freq, data = x, geom = "smooth", group = PickupNeighborhood,
    color = PickupNeighborhood, main = "Pickups per Hour by Neighborhood") + geom_smooth()
ggplot(x, aes(x = Start_Time, y = Freq, col = PickupNeighborhood)) + geom_line()


 #Time Series for End Time
qplot(End_Time, Freq, data = x, geom = "smooth", group = DropoffNeighborhood,
    colour = DropoffNeighborhood, main = "Dropoffs per Hour by Neighborhood") +
 geom_smooth()


#write.csv(taxitable, file = "tstaxi.csv")

qplot(End_Time, Freq, data = tstaxi, geom = "line", group = DropoffNeighborhood,
    colour = DropoffNeighborhood, main = "Dropoffs per Hour by Neighborhood")


#Graphing Start Times and End Times
circlegraph = rose.diag(Tcomplete$Start_Time, bins=24, main="Trip Times")

qplot(y=Taxiset2[,5], x=1:length(Taxiset2[,5]), fill=factor(1:length(Taxiset2[,5])),
    stat='identity', geom='bar') + coord_polar()


Not used in final project:

 ###### CORRESPONDENCE FOR TIPS ######

Correspondence for Tips not used in final project:


taxitips = with(Taxihood2, table(Tips, PickupNeighborhood))
taxitips
cataxitips = ca(taxitips)
cataxitips$N
cataxitips$rowcoord
cataxitips$colcoord

plot(taxitips)
plot(taxitips, mass=T, contrib="absolute",
   map="rowgreen")

plot(taxitips, mass=T, contrib="absolute",
```

```
    map="rowgreen", arrows=c(T, T))
```

Exploratory plots and missing values plots:

```
missmap(Taxiset2)

#Pickup Neighborhood vs. Miles travelled
ggplot(Taxihood2, aes(x = PickupNeighborhood, y = Trip.Miles )) + geom_jitter()

#Payment Types by neighborhood
ggplot(Taxihood2, aes(x = Payment.Type)) + geom_histogram(stat = "count", fill = PickupNeighborhood) +
  labs(x = "Payment Type", y = "Count")

#Payment Types by Neighborhood to examine missing values
ggplot(Taxihood2, aes(x = PickupNeighborhood, y = Payment.Type )) + geom_jitter()

#Payment Types by Hour to examine missing values
names(Taxihood2)
ggplot(Taxihood2, aes(x = Start_Time, y = Payment.Type )) + geom_jitter()

#Pickup vs. Dropoff Neighborhood to replace missing values
ggplot(Taxihood2, aes(x = PickupNeighborhood, y = DropoffNeighborhood)) +
  geom_jitter()

#Plot Times vs. Payment Type
plot1 = ggplot(z, aes(x = Start_Time, y = Freq, fill = Payment.Type)) +
  labs(title = "Payment Type per Hour") + geom_bar(stat = "identity")
plot1 + scale_fill_brewer(palette = "Spectral")
```

D. LDA Analysis

```
setwd('C:\\Users\\rebec\\OneDrive - DePaul University\\DePaul\\Multi\\Project')
taxi = read.csv('Taxi_V10 - FINAL.csv',sep=',',header=T)
head(taxi)
tail(taxi)

#select all numeric and the dependent variables
numTaxi = taxi[, c(24,10:14,16,19:23,26:30)]

#Dividing the data set into training and testing datasets
Total_row = nrow(numTaxi)
Train = round(nrow(numTaxi) * 0.75, 0)
size = sample(Total_row, size = Train)
length(size)

trainTaxi = numTaxi[size, ]
testTaxi = numTaxi[-size, ]

#variables being used
varlist = names(trainTaxi[2:17])
```

```
#Run ANOVAs to find which vars should be included
F6 = aov(Trip.Seconds ~ Payment.Type, data=trainTaxi)
F7 = aov(Trip.Miles ~ Payment.Type, data=trainTaxi)
F8 = aov(Pickup.Census.Tract ~ Payment.Type, data=trainTaxi)
F9 = aov(Dropoff.Census.Tract ~ Payment.Type, data=trainTaxi)
F10 = aov(Pickup.Community.Area ~ Payment.Type, data=trainTaxi)
F11 = aov(Dropoff.Community.Area ~ Payment.Type, data=trainTaxi)
F12 = aov(Fare ~ Payment.Type, data=trainTaxi)
F13 = aov(Tips ~ Payment.Type, data=trainTaxi)
F14 = aov(Tolls ~ Payment.Type, data=trainTaxi)
F15 = aov(Extras ~ Payment.Type, data=trainTaxi)
F16 = aov(Trip.Total ~ Payment.Type, data=trainTaxi)
F17 = aov(Pickup.Centroid.Latitude ~ Payment.Type, data=trainTaxi)
F18 = aov(Pickup.Centroid.Longitude ~ Payment.Type, data=trainTaxi)
F19 = aov(Dropoff.Centroid.Latitude ~ Payment.Type, data=trainTaxi)
F20 = aov(Dropoff.Centroid.Longitude ~ Payment.Type, data=trainTaxi)
F21 = aov(Distance ~ Payment.Type, data=trainTaxi)

summary(F6)  ##
summary(F7)  ##
summary(F8)  ##
summary(F9)  ##
summary(F10) ##
summary(F11) ##
summary(F12) ##
summary(F13) ##
summary(F14)
summary(F15)
summary(F16) ##
summary(F17) ##
summary(F18) ##
summary(F19) ##
summary(F20) ##
summary(F21) ##

#plot(trainTaxi, col=trainTaxi$Payment.Type)

library(MASS)
taxiLDA <- lda(Payment.Type ~ . - Extras - Tolls, data=trainTaxi)
taxiLDA

taxiLDA.values = predict(taxiLDA)

#ldahist(data=RedtaxiLDA.values$x[, 1], g=trainTaxi$Payment.Type)
#ldahist(data=RedtaxiLDA.values$x[, 2], g=taxiTrain$RATING)
#plot(RedtaxiLDA.values$x[, 1], RedtaxiLDA.values$x[, 2], col=taxiTrain$Payment.Type)

#How did it perform on training
p = predict(taxiLDA, trainTaxi[2:17])$class
table(trainTaxi$Payment.Type, p)

# Assess the accuracy of the prediction
# percent correct for each category of Payment Type
ct <- table(trainTaxi$Payment.Type, p)
```

```
diag(prop.table(ct, 1))
# total percent correct
sum(diag(prop.table(ct)))

#How did it perform on validation
p = predict(taxiLDA, testTaxi[2:17])$class
table(testTaxi$Payment.Type, p)

#prop prints x for each LD, where: LD explains x% of the between-group variance in the dataset.
prop = CVtaxiLDA$svd^2/sum(CVtaxiLDA$svd^2)
prop

# Assess the accuracy of the prediction
# percent correct for each category of Payment Type
ct <- table(testTaxi$Payment.Type, p)
diag(prop.table(ct, 1))
# total percent correct
sum(diag(prop.table(ct)))

##Graphs
library(klaR)
partimat(Payment.Type ~.,data=testTaxi,method="lda", col.correct='black',col.wrong='red')

~~Second LDA
taxi = read.csv('Taxi_V10 - FINAL.csv',sep=',',header=T)

#Use only Cash and Credit Card
ntaxi <- taxi[taxi$Payment.Type == 'Cash'|taxi$Payment.Type == 'Credit Card',]

#select all numeric variables that we know at beginning of trip and the dependent variables
numTaxi = ntaxi[, c(24,10:14,16,19:23,26:30)]
redTaxi = numTaxi[,c(1,4,6,13,14)]
#Dividing the data set into training and testing datasets
Total_row = nrow(redTaxi)
Train = round(nrow(redTaxi) * 0.75, 0)
size = sample(Total_row, size = Train)
length(size)

trainTaxi = redTaxi[size, ]
testTaxi = redTaxi[-size, ]

#variables being used
varlist = names(trainTaxi[2:5])
varlist

#Run ANOVAs to find which vars should be included
F8 = aov(Pickup.Census.Tract ~ Payment.Type, data=trainTaxi)
F10 = aov(Pickup.Community.Area ~ Payment.Type, data=trainTaxi)
F17 = aov(Pickup.Centroid.Latitude ~ Payment.Type, data=trainTaxi)
F18 = aov(Pickup.Centroid.Longitude ~ Payment.Type, data=trainTaxi)

summary(F8)
summary(F10)
summary(F17)
```

```
summary(F18)

#How did it perform on training
p = predict(taxiLDA, trainTaxi[2:17])$class
table(trainTaxi$Payment.Type, p)

# Assess the accuracy of the prediction
# percent correct for each category of Payment Type
ct <- table(trainTaxi$Payment.Type, p)
diag(prop.table(ct, 1))
# total percent correct
sum(diag(prop.table(ct)))

#How did it perform on validation
p = predict(taxiLDA, testTaxi[2:17])$class
table(testTaxi$Payment.Type, p)

#prop prints x for each LD, where: LD explains x% of the between-group variance in the dataset.
prop = CVtaxiLDA$svd^2/sum(CVtaxiLDA$svd^2)
prop

# Assess the accuracy of the prediction
# percent correct for each category of Payment Type
ct <- table(testTaxi$Payment.Type, p)
diag(prop.table(ct, 1))
# total percent correct
sum(diag(prop.table(ct)))

##Graphs
library(klaR)
partimat(Payment.Type ~.,data=testTaxi,method="lda", col.correct='black',col.wrong='red')
```

E. CFA Analysis (Not used in our final project)

```
taxi = read.table('Taxi_v7.csv',sep=',',header=TRUE)
head(taxi)

##CFA with Tip removed (Tip as DV)
TaxiSub1 <- taxi[,c(5,7:9,14,16:18)]
head(TaxiSub1)
fit = factanal(TaxiSub1, 4)
print(fit$loadings, cutoff=.4, sort=T)
summary(fit)


##CFA on Tip as Percentage of Fare
TaxiSub <- taxi[,c(5,7:9,14:18)]
head(TaxiSub)
fit = factanal(TaxiSub, 3)
print(fit$loadings, cutoff=.4, sort=T)
summary(fit)
```