

New York City School Bus Analysis

Avinav Pandey, Aishwarya Badlani, Yi Ying Lin, Yu-Yang Hung

Table of Contents:

- **Dataset Link**
- **Dataset Description**
- **Reason for choosing this particular dataset**
- **Content**
- **Background**
- **Objective**
- **Loading Dataset into R**
- **Viewing the Dataset**
- **Dimension Reduction**
- **Data Exploration**
- **Data Cleaning**
- **Algorithm Application**
- **Algorithm Analysis**
 1. Lift Chart
 2. Decile-wise Chart
 3. Residual
- **Results**
- **Reccomondations**
- **References**

Dataset Link:

<https://www.kaggle.com/new-york-city/ny-bus-breakdown-and-delays>

Dataset Description:

The New York city school bus dataset has 268018 observations of 21 variables
This is highly categorical dataset with multiple levels present for different attributes.
Dataset Characteristics: Multivariate
Attribute Characteristics: Categorical, Integer
Missing Values: Yes

Reason for choosing this particular dataset:

The reason for choosing this particular dataset is that we wanted to work with a real-life large dataset which is complex and has missing values so that in analyzing it we would get hands-on experience with working such data and learn to apply what we have learnt in class.

Content:

The Bus Breakdown and Delay system collects information from school bus vendors operating out in the field in real time. Bus staff that encounter delays during the route are instructed to radio the dispatcher at the bus vendor's central office. The bus vendor staff are then instructed to log into the Bus Breakdown and Delay system to record the event and notify OPT. OPT customer service agents use this system to inform parents who call with questions regarding bus service. The Bus Breakdown and Delay system is publicly accessible and contains real time updates. All information in the system is entered by school bus vendor staff. [Taken from Kaggle Content of the dataset]

Background:

Students throughout New York City rely on a fleet of thousands of school buses to arrive safely at school in the morning, and at home each evening.
Students miss important class time and may be delayed for extensive periods while the source of the delay is resolved. So, we wanted to get to the bottom of the reasons for these delays and predict the time by which the future delays can happen.

Objective:

The objective of our analysis is that we wanted to get a deeper insight into the reason for the delays, find trends between the reason for delay and the area where the maximum delays occurred and ultimately be able to make futuristic prediction about the delay time.

The project is divided into the following parts:

1. Understanding the dataset and the meaning of each variables
2. Data Exploration and Data cleaning
3. Application of algorithm

4. Results: Analysis of the algorithm used and its fitness.

Loading dataset into R:

```
df<-read.csv("bus-breakdown-and-delays.csv")
```

Viewing the dataset:

```
View(df)
str(df)
summary(df)
```

```
> str(df)
'data.frame': 268018 obs. of 21 variables:
 $ School_Year      : Factor w/ 5 levels "2015-2016","2016-2017",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ Busbreakdown_ID  : int  1212699 1212700 1212701 1212703 1212704 1212705 1212708 1212709 1212710 1212711 ...
 $ Run_Type         : Factor w/ 11 levels "","General Ed AM Run",...: 9 9 9 9 9 9 9 9 9 9 ...
 $ Bus_No           : Factor w/ 13387 levels "","-", "#", "...: 6780 4136 3768 3150 6768 962 1483 9112 6085 3241 ...
 $ Route_Number     : Factor w/ 12985 levels "","0","01","01 AM",...: 6266 4007 1805 2126 6295 10697 4256 2810 2923 5585 ...
 $ Reason           : Factor w/ 11 levels "","Accident",...: 8 7 8 8 7 7 5 9 8 5 ...
 $ Schools_Serviced : Factor w/ 14971 levels "","(07684","(32274), (16669)",...: 14604 10203 8286 10065 14604 14658 8033 8011 7563 758 ...
 $ Occurred_On      : Factor w/ 121496 levels "2015-09-01T06:12:00",...: 19 18 20 21 21 23 24 25 26 27 ...
 $ Created_On       : Factor w/ 130406 levels "2015-09-01T06:16:00",...: 18 19 20 21 22 23 24 25 26 27 ...
 $ Boro             : Factor w/ 12 levels "","All Boroughs",...: 7 4 4 4 7 12 6 4 4 6 ...
 $ Bus_Company_Name : Factor w/ 117 levels "","1967","1992",...: 16 91 76 32 16 66 91 17 16 54 ...
 $ How_Long_Delayed : Factor w/ 1880 levels "","-----", "-----",...: 918 1 1205 702 1098 1 1 1805 716 626 ...
 $ Number_Of_Students_On_The_Bus : int  0 0 0 1 0 0 0 9 0 2 ...
 $ Has_Contractor_Notified_Schools: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 ...
 $ Has_Contractor_Notified_Parents: Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 1 2 ...
 $ Have_You_Alerted_OPT : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 2 1 2 ...
 $ Informed_On      : Factor w/ 130406 levels "2015-09-01T06:16:00",...: 18 19 20 21 22 23 24 25 26 27 ...
 $ Incident_Number   : Factor w/ 6746 levels "","?", "0","000000",...: 1 1 1 1 1 1 1 1 1 ...
 $ Last_Updated_On   : Factor w/ 160971 levels "1900-01-01T00:00:00",...: 20 21 31 22 23 24 25 26 27 28 ...
 $ Breakdown_or_Running_Late : Factor w/ 2 levels "Breakdown","Running Late": 2 1 2 2 2 1 2 2 2 ...
 $ School_Age_or_PreK : Factor w/ 2 levels "Pre-K","School-Age": 2 2 2 2 2 2 2 2 2 ...
> |
```

Dimension reduction:

After analysis of each variable and the description given on each variable in the dataset, it was decided that the following variables were not useful in our analysis and hence dropped. These are the following variables dropped:

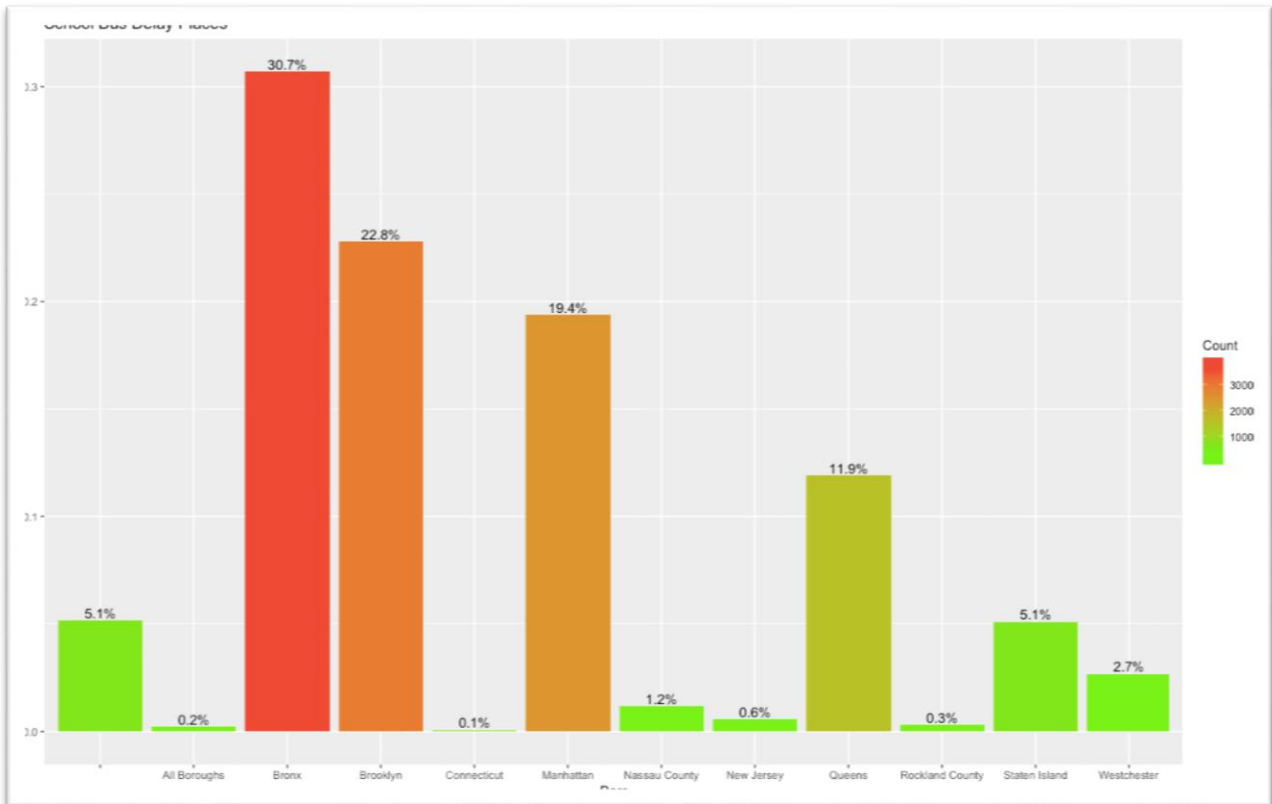
1. Busbreakdown_ID: The bus breakdown id is not very significant and did not provide much insight into our analysis.
2. Created_On: This variable is the Time/date the record was created in the OPT Breakdown and Delay system and not the time when the delay happened. Hence this variable is also not very useful in our analysis.
3. Has_Contractor_Notified_Schools: This variable indicates whether the contractor has notified the school. Indicator status as reported by the staff employed by the reporting bus vendor. OPT does not systematically monitor the contents of this field in real time. Hence, this was also not very useful in our analysis and thus dropped.
4. Has_Contractor_Notified_Parents: This variable indicates whether the contractor has notified the parents. Indicator status as reported by the staff employed by the reporting bus vendor. OPT does not systematically monitor the contents of this field in real time. This also does not give much insight into our analysis of predicting the delay times.
5. Have_You_Alerted_OPT: This variable indicates whether the OPT has been alerted by the staff employed by the reporting bus vendor. Hence not useful in our analysis.
6. Incident_Number: Some reports of bus breakdowns or delays originate from calls to the OPT Customer Service line who records incidents. When this happens, the record will have the Incident reference number.
7. Last_Updated_On: Time/date the record was last edited in the OPT Breakdown and Delay system.

Code for dropping these columns:

```
df <- df[c(-2,-9,-14,-15,-16,-17,-18,-19)]
```

Data Exploration:

The percentage of 10 places School Buses Delayed

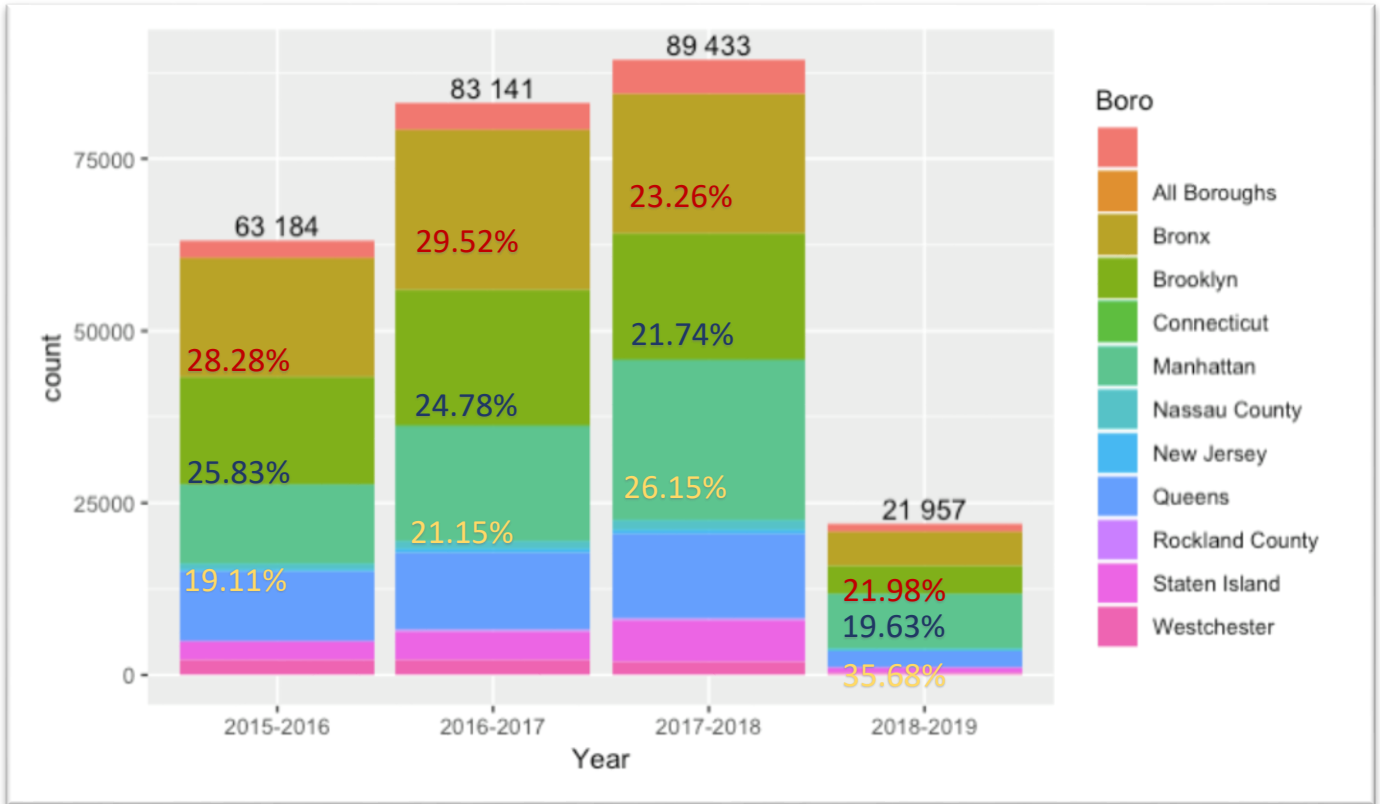


According to the graphs, we can see the highest percentage for delays area is Bronx. It occupies 30.7%. The second one is 22.8%, in Brooklyn, and the third one is 19.4%, in Manhattan. As we know, The Bronx is the northernmost of New York City. The residents of Bronx are mainly African and Latin American. It's the famous slum of the City. Since 1914, the borough has had the same boundaries as Bronx County, the third-most densely populated county in the United States. These components cause the highest percentage for delays area is here.

Manhattan is the geographically smallest and most densely populated borough; is the symbol of New York City which has most of the city's skyscrapers and prominent landmarks. Also, it contains the headquarters of many major multinational corporations, the United Nations Headquarters, Wall Street. Brooklyn, on the western tip of Long Island, is the city's most populous borough. Downtown Brooklyn is the largest central core neighborhood in the outer boroughs.

Brooklyn, Manhattan and Queens are populous boroughs. Many people in the whole world desire to visit here. They are prosperous and bustling.

The percentage of 10 places School Buses Delayed during 2015-2019

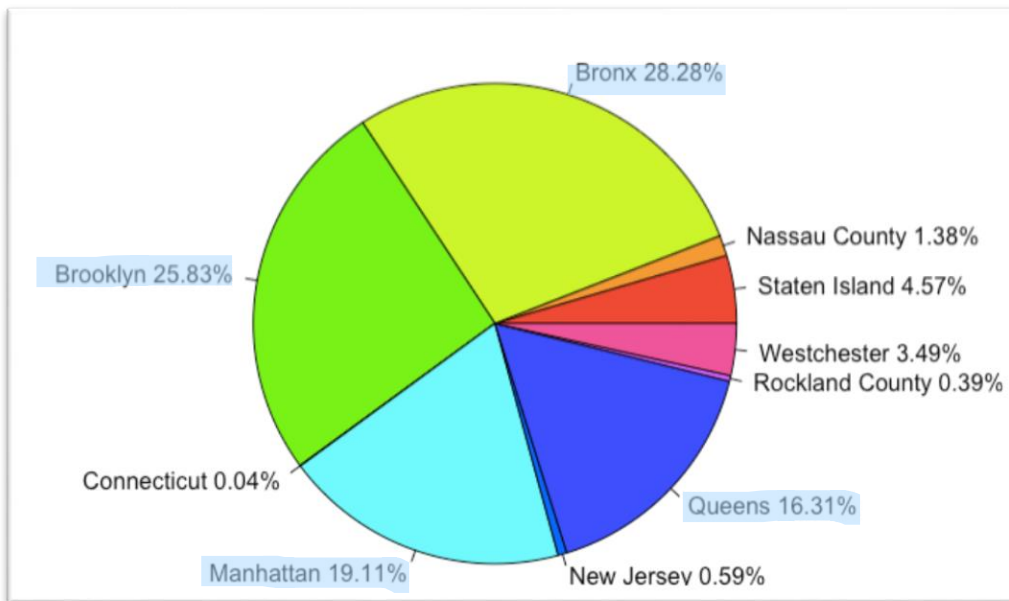


Looking in past years, the areal distribution of delay does not have obvious change. The borough Bronx is on top in 2015 and 2016, but then drops to 21.98% in 2018. In Brooklyn, the percentage of delays situation decreased gradually. Maybe government did some strategies to improve the situation of delays. The main reasons of delay are below the report. Some area improved the issue, but the delays in Manhattan goes up gradually year by year, which increases 18.65 percentiles from 19.11 to 37.76% during 2015 to 2018/10.

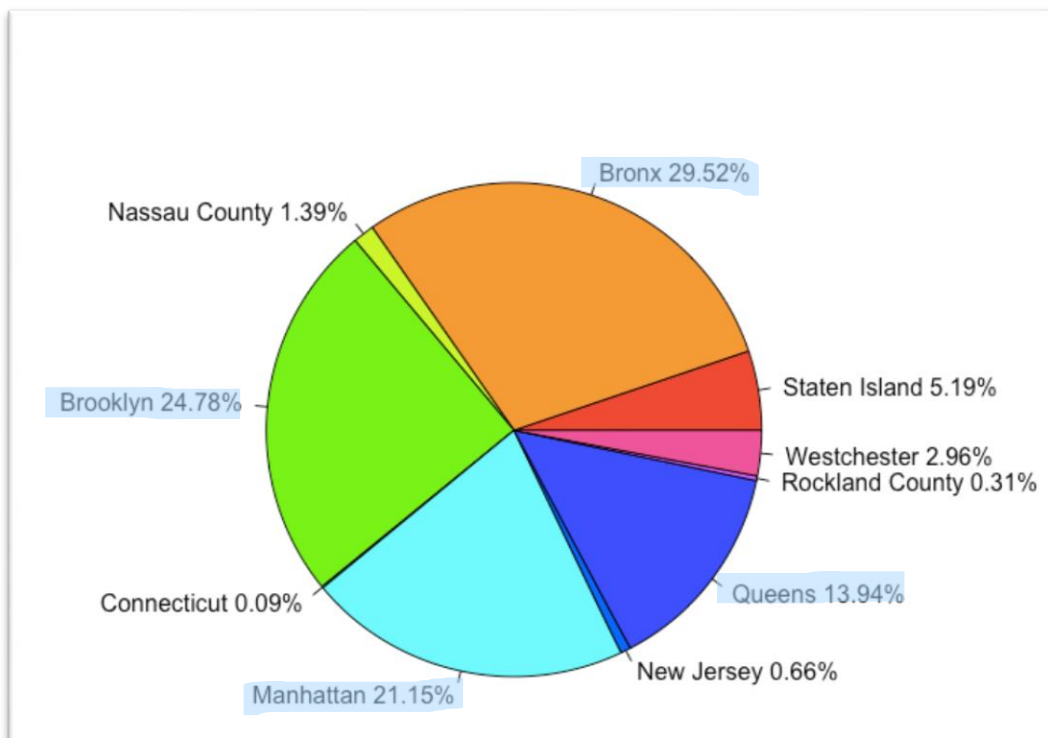
Overall, we can see the delay situation in these years does not improve so much. The amount of delays increases 26,249 from 63,184 to 89,433, which is 40%. However, the latest data show 21,957 until October 2018. If the data is reliable, it's not expected that last three months of this year will over amount in past years, which drops a lot.

The pie charts of details in each year is below:

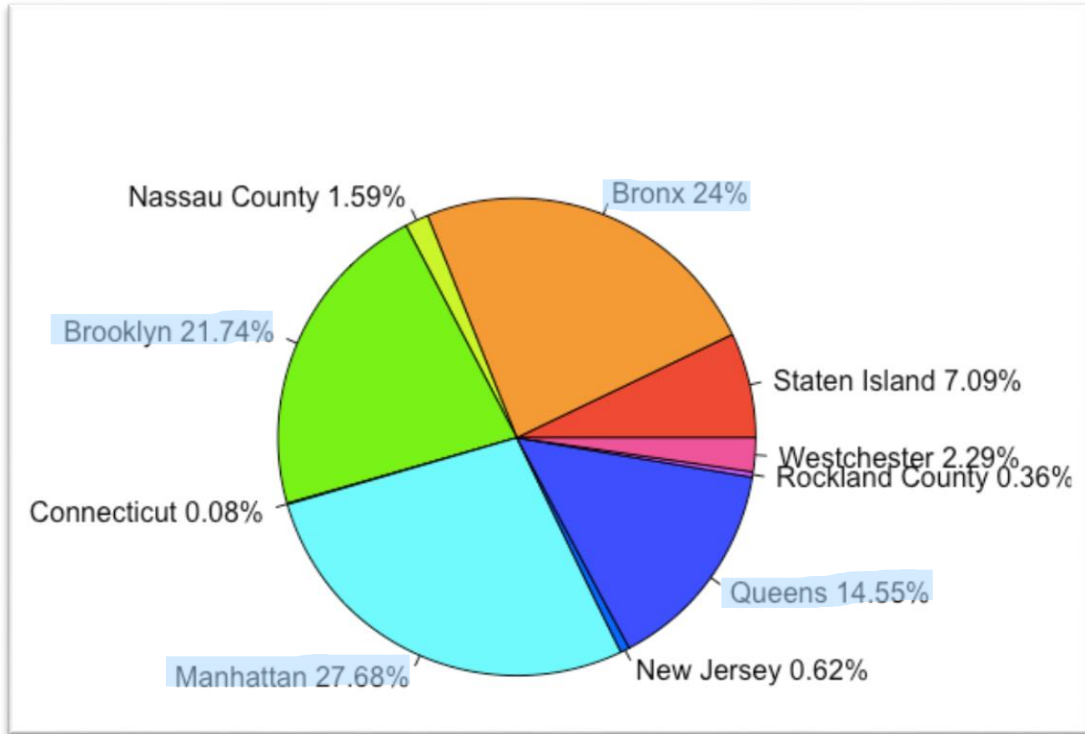
The 10 Places School Buses were Delayed during 2015- 2016



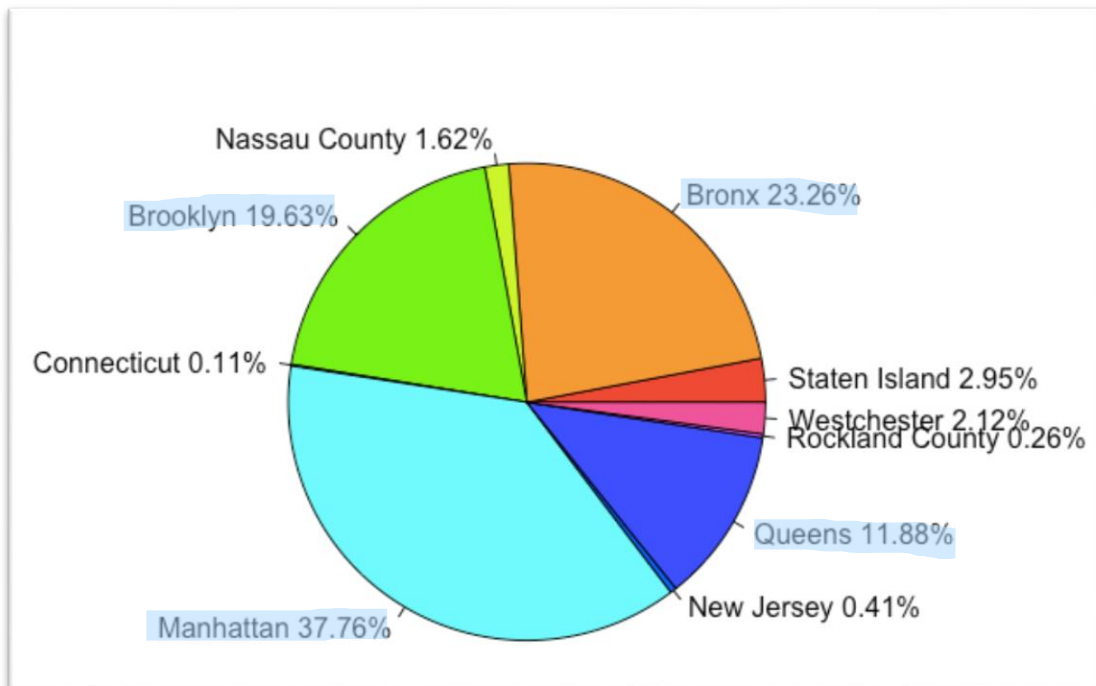
10 Places school buses were delayed during 2016-2017



10 Places school buses were delayed during 2017 - 2018



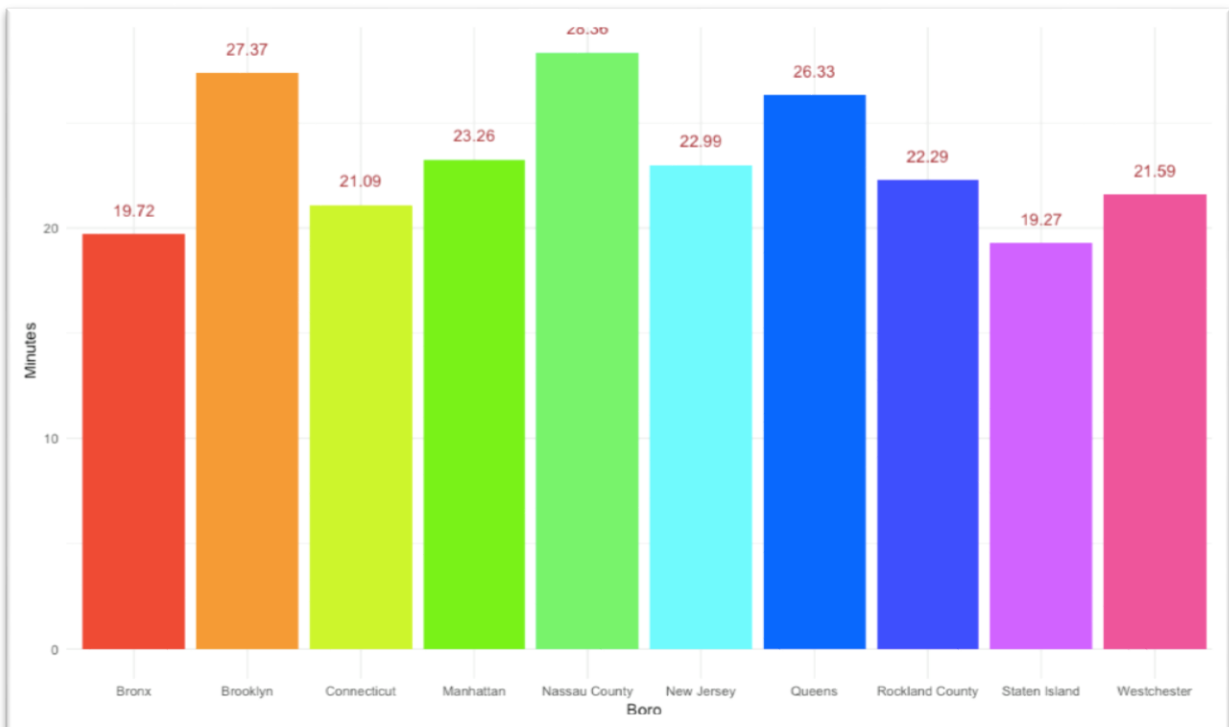
10 Places school buses were delayed during 2018-2019



The Average of Delay Times in each 10 Places (Boro):

| | Boro | delay |
|----|-----------------|----------|
| 1 | Nassau County | 28.35933 |
| 2 | Brooklyn | 27.36945 |
| 3 | Queens | 26.32866 |
| 4 | Manhattan | 23.25895 |
| 5 | New Jersey | 22.99045 |
| 6 | Rockland County | 22.29219 |
| 7 | Westchester | 21.58710 |
| 8 | Connecticut | 21.09000 |
| 9 | Bronx | 19.71734 |
| 10 | Staten Island | 19.27290 |

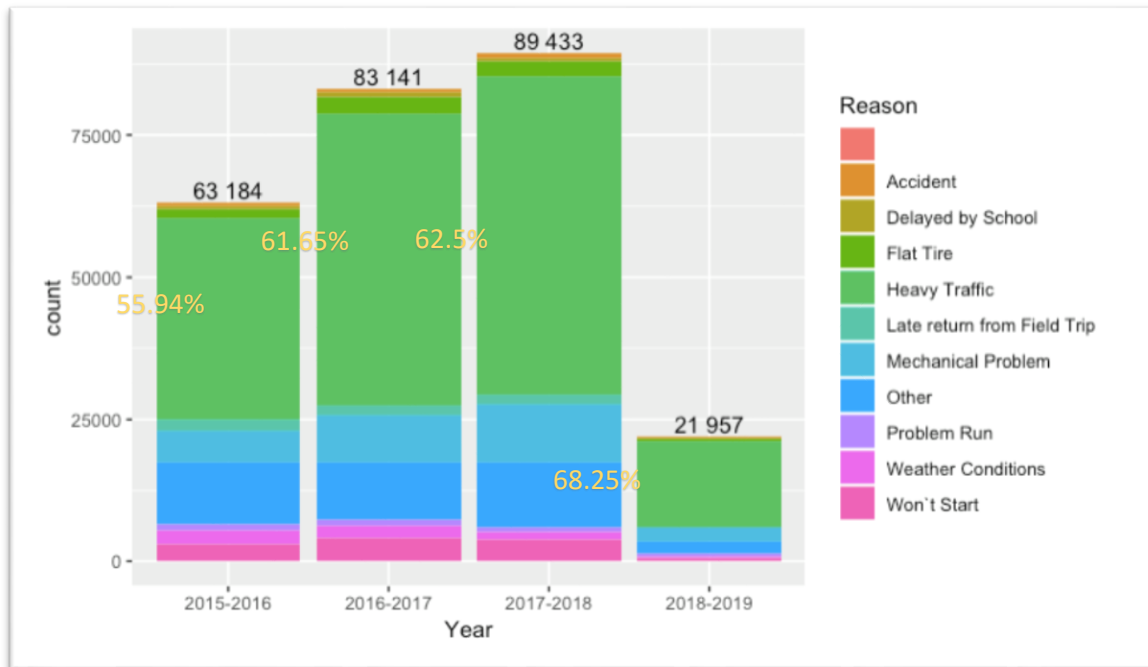
The Relationship between Boro and Average Delay Time



Students face different delay times depending on the school's location. When delays occurred, those in Nassau County, for example, faced an average of 28.36 minutes over the past three

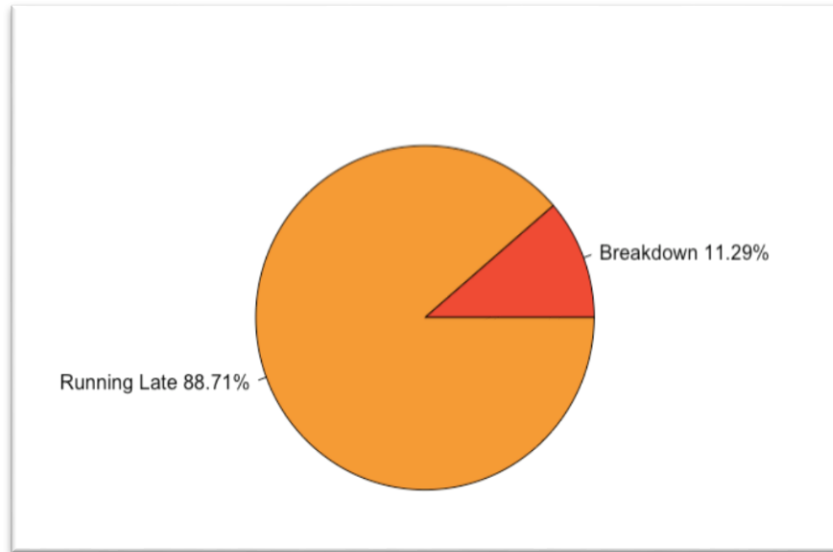
years. Students in Brooklyn and in Queens were delayed an average of 27.37 minutes and 26.33 minutes. Both areas are prosperous and crowded. Comparing Staten Island, which is in a relatively remote location in New York, the average of delays time is only 19.27 minutes over the same period.

The Reason of Delay during year 2015 to 2019



According to OPT data, heavy traffic is the most common reason for delays, which accounted for over 50 percent of delays in the past three years 2015, 2016 and 2017. Next reason is mechanical problem.

Percentage of Delay Reasons as reported by Bus Vendor



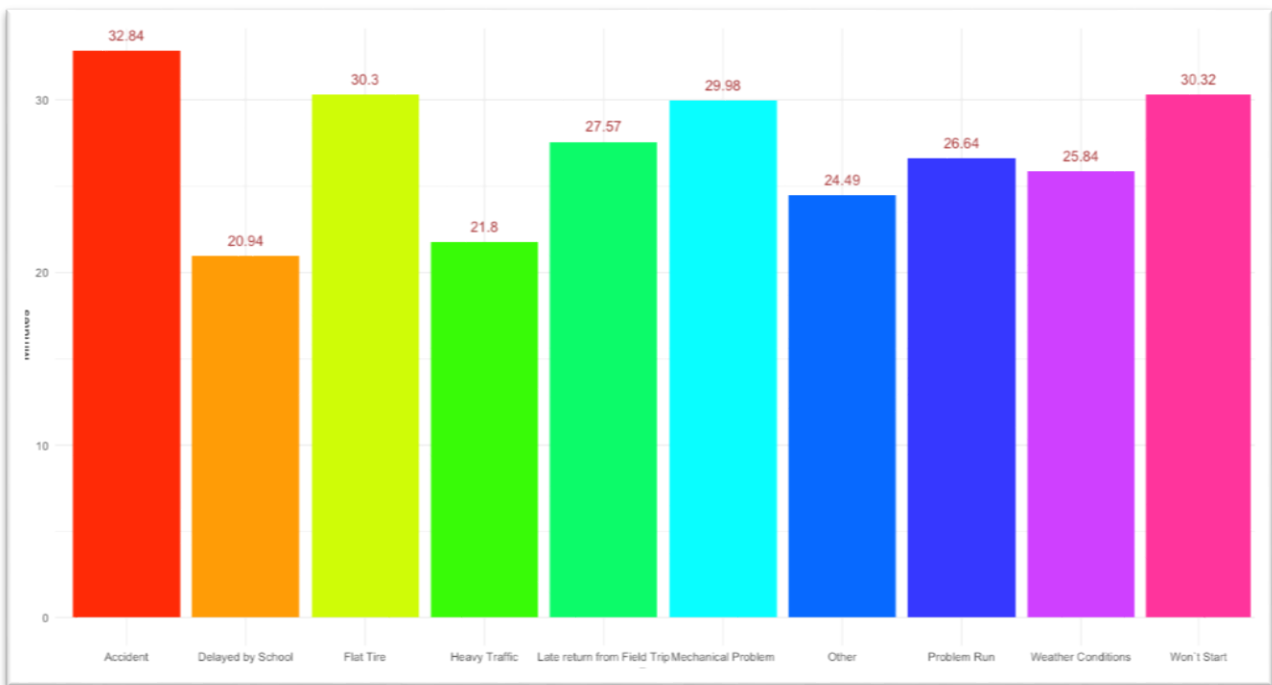
The bus vendors report the reasons for delay by themselves. The reason only includes two: breakdown and running late. A bus is broken down which means require another vehicle to be dispatched to finish the route and is delayed which means may not require another vehicle. The data shows the reason: running late is 88.7%.

Two of graphs indicate the main problem for delay is not "bus" itself. Mostly, it's external factors.

Delay Trends for delay reasons:

| | Reason | delay |
|----|-----------------------------|----------|
| 1 | Accident | 32.84447 |
| 2 | Won`t Start | 30.32483 |
| 3 | Flat Tire | 30.30040 |
| 4 | Mechanical Problem | 29.97665 |
| 5 | Late return from Field Trip | 27.56712 |
| 6 | Problem Run | 26.63613 |
| 7 | Weather Conditions | 25.83951 |
| 8 | Other | 24.49038 |
| 9 | Heavy Traffic | 21.79512 |
| 10 | Delayed by School | 20.93632 |

The Relationship between Reason and Average Delay Time



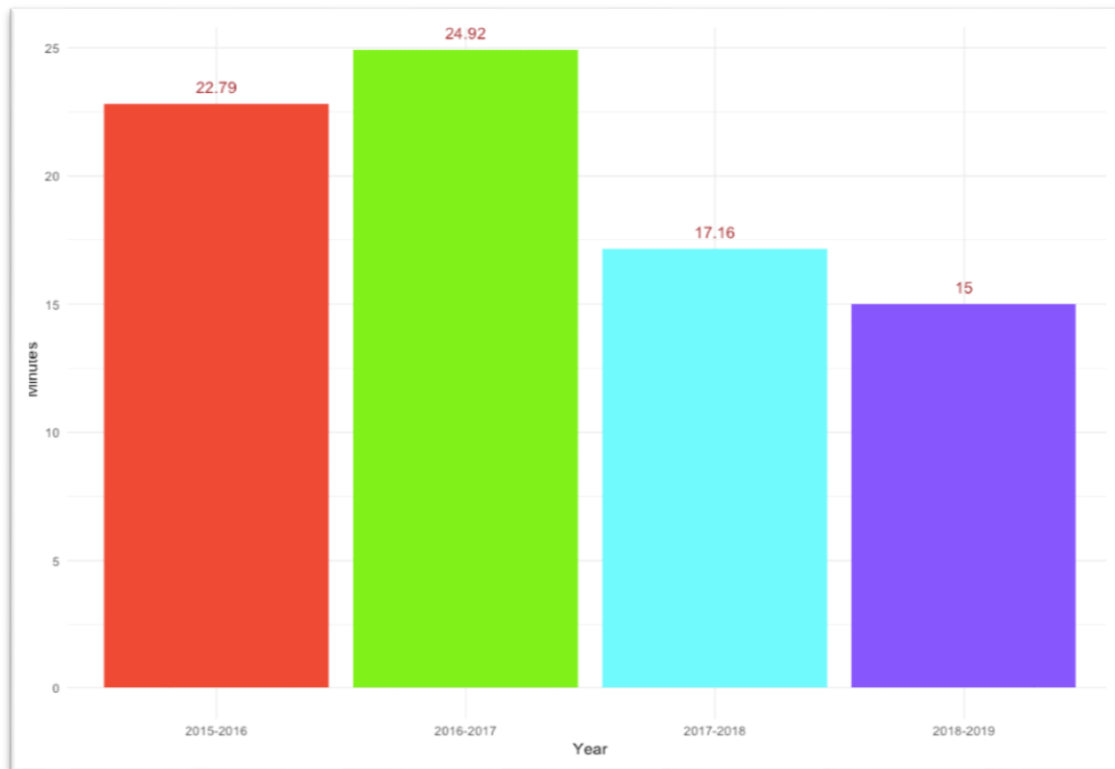
The following analysis explored different reasons for the delays occurred between September 2015 and September 2018, and the average delay each reason has contributed with schools contributing least to the delay reasons.

While traffic accidents are relatively rare, they cause the longest delays. They resulted in delays averaging 39 minutes and heavy traffic causes average 21 minutes in delays.

Delay Times

| | School_Year | delay |
|---|-------------|----------|
| 1 | 2016-2017 | 24.91545 |
| 2 | 2015-2016 | 22.79141 |
| 3 | 2017-2018 | 17.15967 |
| 4 | 2018-2019 | 15.00000 |

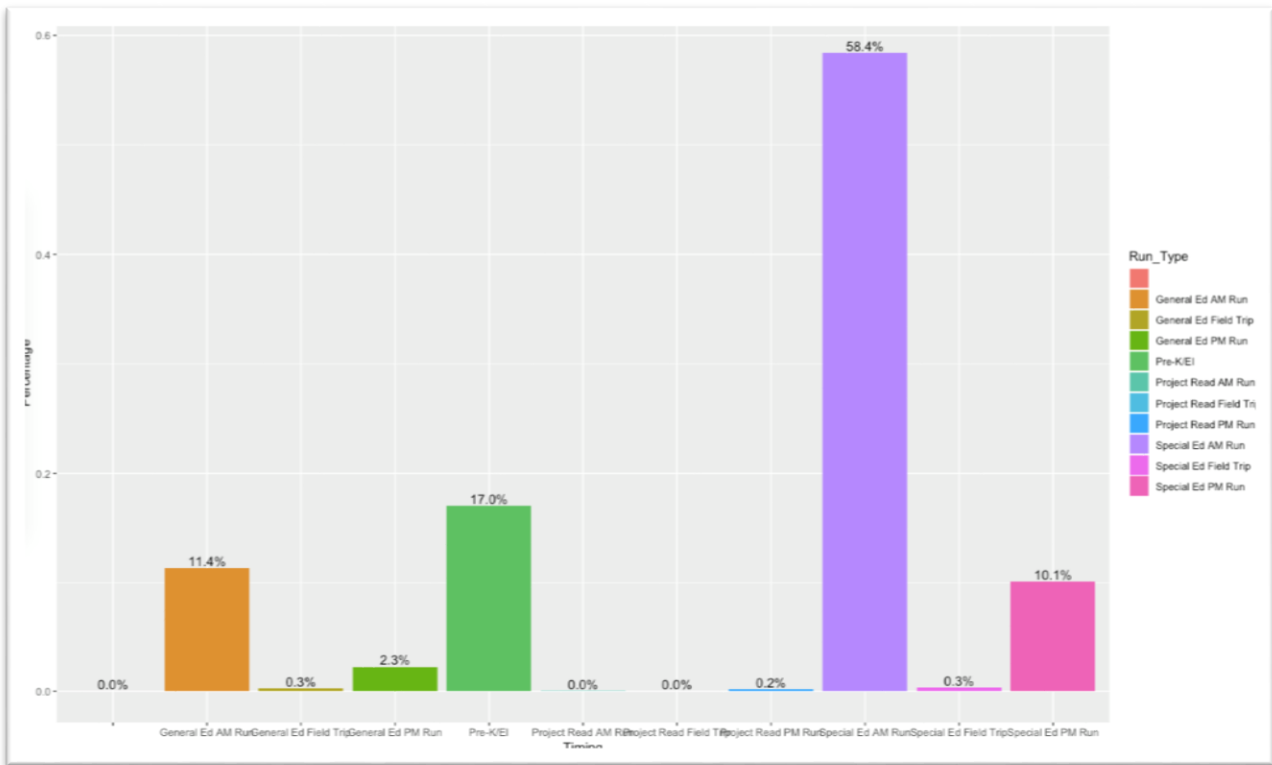
The Relationship between Year and Average Delay Time



In 2015, on average, a student's riding a delayed school buses could expect to be 22.79 minutes late. This is an increase from 2016, when the average delay was 24.92 minutes, but a decrease from 2017, when the average delay time was 17.6 minutes. Also, a decrease 2 minutes, it delays 15 minutes in 2018.

We cannot sure it will keep decrease in the future, even latest two years show it decrease. Since there are only 4 years data and are just slightly ups and down, it's hard to say the situation for delay is improved.

Breakdown of Delay occurred on a specific category of bus services:



OPT also classifies the campus bus in different purpose: General Ed (general education) run, General Ed Field Trip, Special Ed Run and so on. We pick obvious 4 types among of all to discuss: Special Ed AM Run 58.4%, Pre-K/EI 17%, General Ed AM Run 11.4%, and Special Ed PM Run 10.1%.

The definition of these types:

Special Ed AM Run- curb-to-curb service in the morning with pick-ups at residences and drop-offs at school(s)

Pre-K/EI- curb-to-curb service. It's specially to Pre-Kindergarten or Early Intervention.

General Ed AM Run - stop-to-school service in the morning with pick-ups at bus stops and drop-offs at school(s).

Special Ed PM Run - curb-to-curb service in the afternoon with pick-ups at school(s) and drop-offs at residences.

Special Ed AM Run occupied over a half of all, 58.4%; Pre-K/EI rank number two, but it only takes 17%. If the whole data from the same population. We can know actually the general campus buses don't usually have problem for delays. Therefore, the government can choose the designated types of bus in each time to improve delay issues.

Data Cleaning:

The dataset is highly categorical with multiple levels present and lot of missing values. Hence cleaning the dataset and preparing it for the application of algorithm is very crucial. It was one of the most difficult part of the project as we had never dealt with such a complex dataset with so many categorical variables with a lot of missing data and many of the variables were character. So, to make it suitable for linear regression, a lot of cleaning had to be done. And in this process, we learned a lot. Below mentioned are the data cleaning processes.

1. Firstly, the **Boro** variable which is the Borough or the county where the delay occurred is categorical and has many missing values. Hence had to firstly convert the Null values to NA and then had to convert the column to character and then converted the NA values to readable FF character. Then created a temporary dataframe with all FF as Boro values. Then converted the Boro column into dummy variables using `model.matrix` on the original dataset and then dropped the Boro column as the dummy variables had been created.
2. The **Route_Number** column had to be leveled as it was also categorical with multiple levels. Created a new Column `Route.No.leveled` in the dataframe and levelled the actual `Route_Number` and after levelling them, dropped the `Route_Number`.
3. For **School_Serviced** column, firstly filtered out schools with 0 and ` values taking the length of values for schools serviced column and then levelled the column based on the length and number of alphabets in the column. *Understood pattern recognition from the below mentioned site:*
http://www.jdatalab.com/data_science_and_data_mining/2017/03/20/regular-expression-R-part11.html
4. Then derived the time component and the AM/PM component from the column `Occurred_On` and then levelled the AM/PM component and dropped the original `Occurred_On` component.
5. Then created dummy variables for `School_Year`.
6. Then Coming to the dependent Variable `How_Long_Delayed` which is a character, so converted it to numeric to get only good numbers and to get good amount of rows to train and test data.
7. Taking only the last part of bus company name to easily dummify the data
Source from
http://www.jdatalab.com/data_science_and_data_mining/2017/03/20/regular-expression-R-part11.html
8. Similarly had to clean all the other columns based on whether they were categorical or not and whether they had missing values or not.

Algorithm Application:

Model: LINEAR REGRESSION

Our target variable was How Long Delayed. Since we wanted to predict how long the next bus would be delayed.

We had to first find out then least contributing variables and then remove them from the model but instead of doing so we used STEP AIC where the model itself chooses the best subset of the independent variables to build the model.

Performing the linear modelling on the data set

We have applied linear regression that is effective for Prediction.

Traditional Statistics measures

Residual standard error: 8.155 on 91049 degrees of freedom

Multiple R-squared: 0.2899, *Adjusted R-squared:* 0.2895

F-statistic: 640.9 on 58 and 91049 DF, *p-value:* < 2.2e-16

Residual Standard Error tells us that 8.155 data is not explained or is unexplained by our model on a scale of 91049 degrees of freedom.

Similarly, is with adjusted r square and multiple r squared.

AIC to find the best fit model

AIC number: is that is helpful for comparing models as it includes measures of both how well the model fits the data and how complex the model is.

Call:

```
lm(formula = How_Long_Delayed ~ Bus_No.sk + Number_Of_Students_On_The_Bus +  
  School_Age_or_Prek + `BoroAll Boroughs` + BoroBronx + BoroBrooklyn +  
  BoroFF + `BoroNassau County` + `BoroNew Jersey` + BoroQueens +  
  `BoroRockland County` + `BoroStaten Island` + ReasonAccident +  
  `ReasonDelayed by School` + `ReasonFlat Tire` + `ReasonHeavy Traffic` +  
  `ReasonLate return from Field Trip` + `ReasonMechanical Problem` +  
  ReasonOther + `ReasonProblem Run` + `ReasonWon't Start` +  
  Route.No.levelled8 + `Route.No.levelledstop-to-school` + `Schools.Srvcd.lvlMultiple Sites` +  
  `Run_TypeProject Read Field Trip` + `Run_TypeSpecial Ed AM Run` +  
  `abvtd_bcmpnme(` + `abvtd_bcmpnme(B2` + `abvtd_bcmpnme(B2192)` +  
  `abvtd_bcmpnme(B23` + `abvtd_bcmpnme(B232` + `abvtd_bcmpnme(B2321)` +  
  `abvtd_bcmpnme(SCH` + abvtd_bcmpnmeADDIES + `abvtd_bcmpnmeAGE)` +  
  abvtd_bcmpnmeBUS + abvtd_bcmpnmeC + abvtd_bcmpnmeCO + `abvtd_bcmpnmeCO.,INC.` +  
  abvtd_bcmpnmeCORP + abvtd_bcmpnmeI + abvtd_bcmpnmeIN + abvtd_bcmpnmeINC +  
  abvtd_bcmpnmeINC. + `abvtd_bcmpnmeINC.(B2192)` + abvtd_bcmpnmeLLC +  
  abvtd_bcmpnmeLTD. + abvtd_bcmpnmePICK + abvtd_bcmpnmeSERVICE +  
  abvtd_bcmpnmeSYST + abvtd_bcmpnmeSYSTEMS + abvtd_bcmpnmeTRANSIT +  
  abvtd_bcmpnmeTRANSPORTATION, data = df.train)
```

Residuals:

| Min | 1Q | Median | 3Q | Max |
|---------|--------|--------|-------|--------|
| -35.451 | -5.570 | -1.170 | 4.998 | 29.958 |

The below is the lm output. The section of output labeled 'Residuals' gives the difference between the experimental and predicted. Estimates for the model's coefficients are provided along with their standard deviations ('Std Error'), and a t-value and probability for a null hypothesis.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.971e+01  3.098e-01  63.614 < 2e-16 ***
Bus_No.sk    2.589e-05  1.188e-05   2.179 0.029338 *
Number_Of_Students_On_The_Bus -1.975e-03  4.941e-04 -3.998 6.38e-05 ***
School_Age_or_Prek  3.318e+00  1.447e-01  22.932 < 2e-16 ***
`BoroAll Boroughs`  6.913e+00  6.110e-01  11.314 < 2e-16 ***
BoroBronx    -3.054e+00  1.131e-01 -27.006 < 2e-16 ***
BoroBrooklyn  1.337e+00  1.032e-01  12.946 < 2e-16 ***
BoroFF       -1.532e+00  1.574e-01 -9.730 < 2e-16 ***
`BoroNassau County`  2.366e+00  2.701e-01  8.760 < 2e-16 ***
`BoroNew Jersey`   8.872e-01  3.830e-01  2.317 0.020528 *
BoroQueens     1.806e+00  1.154e-01  15.650 < 2e-16 ***
`BoroRockland County` -8.632e-01  5.073e-01 -1.701 0.088868 .
`BoroStaten Island` -2.929e-01  1.746e-01 -1.678 0.093415 .
ReasonAccident  7.411e+00  4.001e-01  18.524 < 2e-16 ***
`ReasonDelayed by School` -3.671e+00  3.094e-01 -11.867 < 2e-16 ***
`ReasonFlat Tire`    3.741e+00  2.750e-01  13.607 < 2e-16 ***
`ReasonHeavy Traffic` -3.129e+00  1.628e-01 -19.224 < 2e-16 ***
`ReasonLate return from Field Trip` 1.145e+00  2.477e-01  4.623 3.78e-06 ***
`ReasonMechanical Problem` 2.129e+00  2.153e-01  9.890 < 2e-16 ***
ReasonOther    -2.093e+00  1.770e-01 -11.822 < 2e-16 ***
`ReasonProblem Run`  4.299e-01  2.764e-01  1.555 0.119893
`ReasonWon\\`t Start` 1.823e+00  2.630e-01  6.930 4.24e-12 ***
Route.No.level8  5.559e+00  1.301e+00  4.272 1.94e-05 ***
`Route.No.level8stop-to-school` 4.230e-01  1.198e-01  3.530 0.000416 ***
`Schools.Srvcd.lvldMultiple Sites` -2.333e+00  1.096e-01 -21.292 < 2e-16 ***
`Run_TypeProject Read Field Trip` 1.572e+01  8.351e+00  1.882 0.059808 .
`Run_TypeSpecial Ed AM Run` -2.603e-01  9.369e-02 -2.778 0.005469 **
`abvtd_bcmprnme(`  1.496e+00  2.335e-01  6.408 1.48e-10 ***
`abvtd_bcmprnme(B2`  6.968e+00  1.392e-01  50.052 < 2e-16 ***
`abvtd_bcmprnme(B2192)` -2.317e+00  1.282e-01 -18.068 < 2e-16 ***
`abvtd_bcmprnme(B23` -7.396e+00  2.789e+00 -2.652 0.008010 **
`abvtd_bcmprnme(B232` -1.864e+00  1.247e-01 -14.956 < 2e-16 ***
`abvtd_bcmprnme(B2321)` -3.510e+00  1.941e-01 -18.084 < 2e-16 ***
`abvtd_bcmprnme(SCH`  2.832e+00  2.325e-01  12.180 < 2e-16 ***
abvtd_bcmprnmeADDIES 1.223e+01  2.954e+00  4.139 3.49e-05 ***
`abvtd_bcmprnmeAGE)` -6.568e+00  1.743e+00 -3.768 0.000165 ***

abvtd_bcmprnmeBUS    4.591e+00  3.517e-01  13.052 < 2e-16 ***
abvtd_bcmprnmeC      1.070e+00  3.549e-01  3.016 0.002564 **
abvtd_bcmprnmeCO     -5.053e+00  1.542e-01 -32.780 < 2e-16 ***
`abvtd_bcmprnmeCO.,INC.` 6.313e+00  1.555e+00  4.061 4.89e-05 ***
abvtd_bcmprnmeCORP   -4.341e+00  2.151e-01 -20.180 < 2e-16 ***
abvtd_bcmprnmeI      3.073e+00  2.719e-01  11.300 < 2e-16 ***
abvtd_bcmprnmeIN     -1.847e+00  3.021e-01 -6.116 9.65e-10 ***
abvtd_bcmprnmeINC    -3.778e+00  1.698e-01 -22.245 < 2e-16 ***
abvtd_bcmprnmeINC.   1.346e+00  1.158e-01  11.628 < 2e-16 ***
`abvtd_bcmprnmeINC.(B2192)` -3.786e+00  2.264e-01 -16.723 < 2e-16 ***
abvtd_bcmprnmeLLC    -1.550e+00  8.922e-01 -1.737 0.082428 .
abvtd_bcmprnmeLTD.   4.202e+00  1.514e-01  27.757 < 2e-16 ***
abvtd_bcmprnmePICK   9.711e+00  6.991e-01  13.891 < 2e-16 ***
abvtd_bcmprnmeSERVICE 4.561e+00  1.955e-01  23.326 < 2e-16 ***
abvtd_bcmprnmeSYST   4.810e+00  1.191e+00  4.038 5.39e-05 ***
abvtd_bcmprnmeSYSTEMS -5.416e+00  3.159e+00 -1.714 0.086497 .
abvtd_bcmprnmeTRANSIT 6.965e+00  3.412e+00  2.042 0.041181 *
abvtd_bcmprnmeTRANSPORTATION 1.729e+00  4.952e-01  3.491 0.000482 ***

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

Residual standard error: 8.349 on 89451 degrees of freedom
Multiple R-squared:  0.2573,    Adjusted R-squared:  0.2568
F-statistic: 584.6 on 53 and 89451 DF,  p-value: < 2.2e-16

```

Various variables like p values shows the significance level of different features or predictors in our dataset .The result above shows the significance level of different features or predictors in our dataset .For example: - Number of students on the bus ,school_age or Prek , BoroAll Boroughs has 0 level of significance .While on the other hand variables like Run_TypeSpecial AM run, abvtd_bcmepme etc which has significance level od 0.001. Similarly different variables have different level of significance as shown in the output.

The following table shows the predicted values and the actual values. We want to know how well the model predicts new data, not how well it fits the data it was trained with.

| | pred | real |
|----|----------|------|
| 6 | 25.43974 | 25 |
| 11 | 26.67662 | 30 |
| 14 | 26.67662 | 30 |
| 16 | 27.70566 | 20 |
| 17 | 25.33999 | 20 |
| 21 | 26.67662 | 45 |
| 24 | 23.14085 | 15 |
| 25 | 26.67662 | 45 |
| 35 | 27.70566 | 20 |
| 38 | 26.67662 | 30 |
| 39 | 26.67662 | 22 |
| 40 | 25.04711 | 30 |
| 47 | 26.67662 | 30 |
| 50 | 26.67662 | 30 |
| 55 | 23.80835 | 15 |
| 58 | 26.67662 | 45 |
| 61 | 28.26475 | 15 |
| 63 | 23.14085 | 15 |
| 66 | 26.67662 | 30 |

Strength and Weakness of Linear Regression

1. Is Sensitive to Outliners

Mostly all our dataset features had outliers, which effected our model performance. Outliers can be univariate (base on one variable) or multivariate. Outliers can have huge effects on the regression.

2. Data Must Be Independent

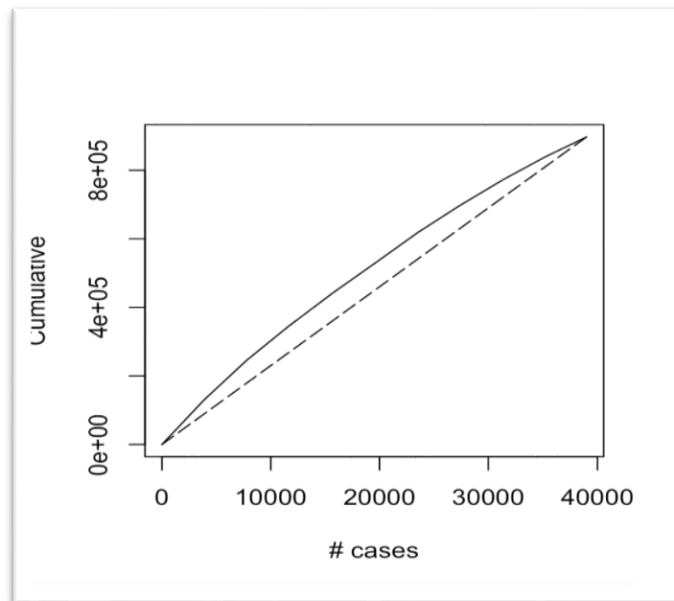
Linear regression assumes that the data are independent. That means that the scores of one subjects (such as a person) have nothing to do with those of another. This is often, but not always, sensible.

3. More than one independent variable is correlated with the dependent variable.

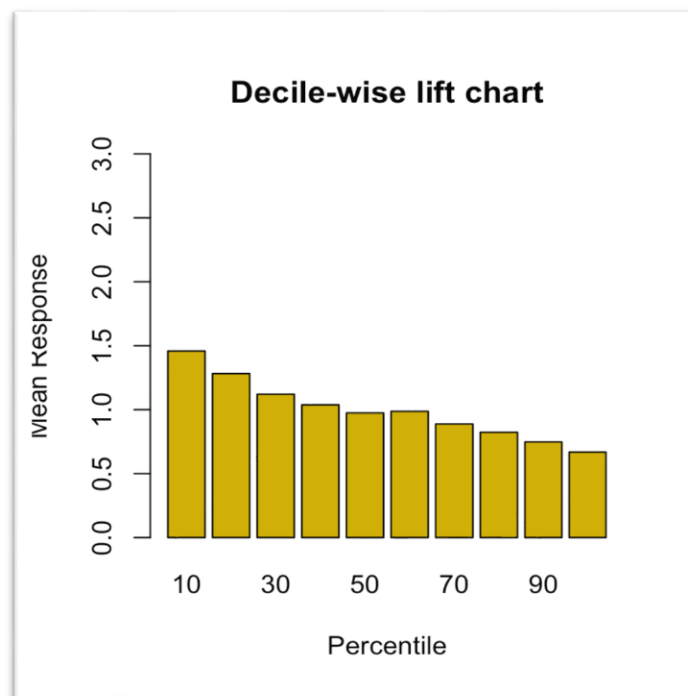
Algorithm Analysis:

Lift Chart:

Looking the lift chart, we can say that the model is doing well. As we can see from the lift chart that our model is doing better than the naïve model.



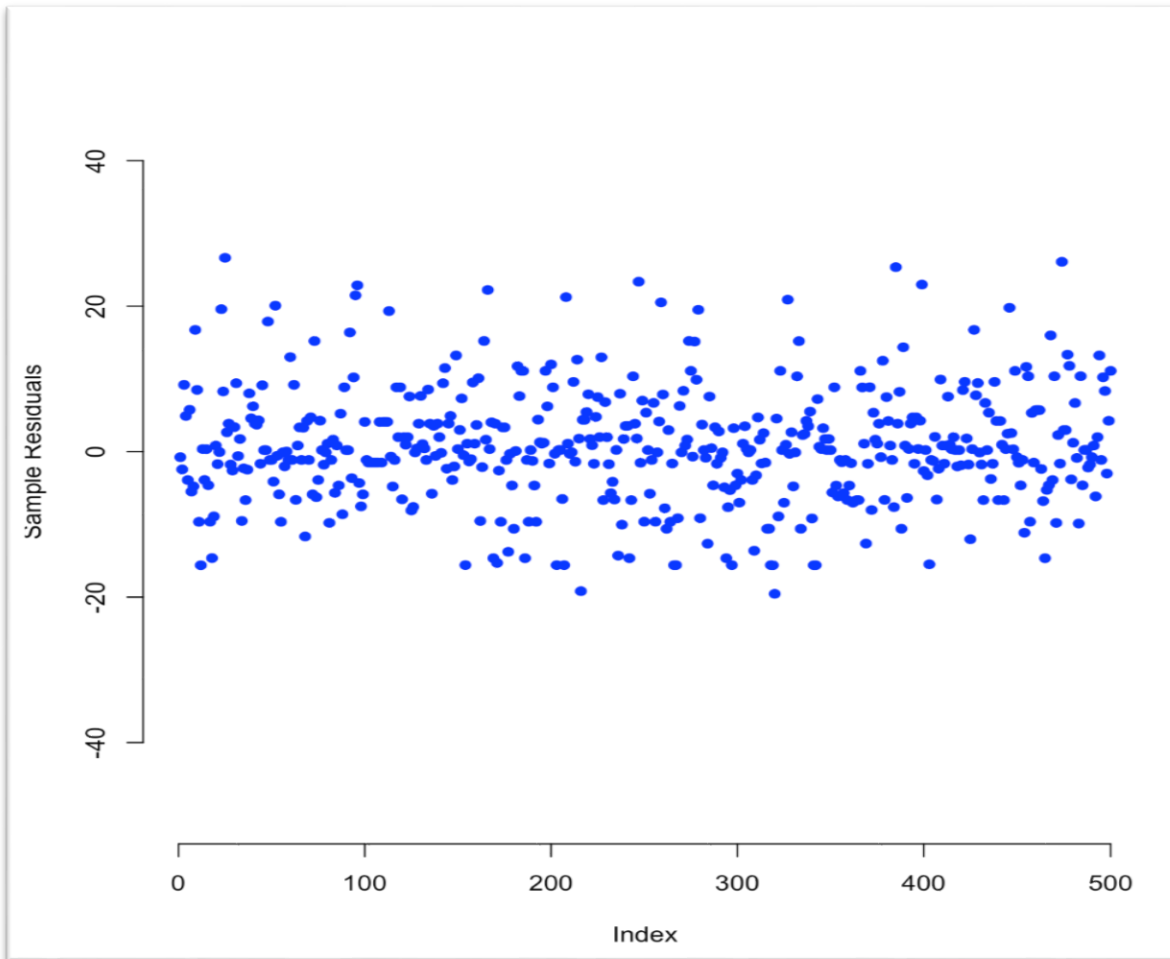
Plotting decile-wise chart:



Analysis of Decile Chart:

Our model is doing well as the top decile is able to capture 1.5 times better than the average and it is descending in nature. So, it is good but there is a scope of improvement for our model as all the decile are not in a proper descending order. Our model is performing good when using 40% of data is used.

Residual:



Analysis of residuals:

We have made residual of 500 observations. In our dataset the residuals are horizontally spread which tells us about our data that our linear model is appropriate in terms of unexplained data.

Results:

```
library(caret)
> RMSE(df.predict,df.test$How_Long_Delayed)
[1] 8.150196
```

RMSE = 8.71 which is not good. But as it is a relative measure and our dataset is a highly categorical dataset. Since our RMSE was coming 8.71

```
Step: AIC=378952.6
How_Long_Delayed ~ Bus_No.sk + Number_Of_Students_On_The_Bus +
  `BoroAll Boroughs` + BoroBronx + BoroBrooklyn + BoroFF +
  BoroManhattan + `BoroNassau County` + `BoroNew Jersey` +
  BoroQueens + ReasonAccident + `ReasonDelayed by School` +
  `ReasonFlat Tire` + `ReasonHeavy Traffic` + `ReasonLate return from Field Trip` +
  ReasonOther + `ReasonProblem Run` + `ReasonWeather Conditions` +
  Route.No.leveled6 + Route.No.leveled7 + `Route.No.leveledPre-K/EI Route` +
  `School_Year2015-2016` + `Run_TypeGeneral Ed Field Trip` +
  `Run_TypeGeneral Ed PM Run` + `Run_TypePre-K/EI` + `Run_TypeProject Read Field Trip` +
  `Run_TypeProject Read PM Run` + `Run_TypeSpecial Ed Field Trip` +
  `Run_TypeSpecial Ed PM Run` + `abvtd_bcmpnme(` + `abvtd_bcmpnme(B` +
  `abvtd_bcmpnme(B2` + `abvtd_bcmpnme(B2192)` + `abvtd_bcmpnme(B23` +
  `abvtd_bcmpnme(B232` + `abvtd_bcmpnme(B2321)` + `abvtd_bcmpnme(SCH` +
  abvtd_bcmpnmeADDIES + `abvtd_bcmpnmeAGE)` + abvtd_bcmpnmeBUS +
  abvtd_bcmpnmeC + abvtd_bcmpnmeCO + `abvtd_bcmpnmeCO.,INC.` +
  abvtd_bcmpnmeCORP + abvtd_bcmpnmeCORP. + abvtd_bcmpnmeI +
  abvtd_bcmpnmeIN + abvtd_bcmpnmeINC + abvtd_bcmpnmeINC. +
  `abvtd_bcmpnmeINC.(B2192)` + abvtd_bcmpnmeLLC + abvtd_bcmpnmeLTD. +
  abvtd_bcmpnmePICK + abvtd_bcmpnmeSERVICE + abvtd_bcmpnmeSTEPS +
  abvtd_bcmpnmeSYST + abvtd_bcmpnmeSYSTEMS + abvtd_bcmpnmeTRANSIT
```

We used step AIC to find goodness of fit. The AIC number also came pretty high that is 378952.6.

Dimension Reduction Section, we didn't use PCA because there were only 14 variables left after dropping the unwanted variables.

It means that there is no absolute good or bad threshold, however you can define it based on your DV. For a datum which ranges from 0 to 1000, an RMSE of 0.7 is small, but if the range goes from 0 to 1, it is not that small anymore. However, although the smaller the RMSE, the better, you can make theoretical claims on levels of the RMSE by knowing what is expected from your DV in your field of research

The charts are not as good because this is real world data and it is difficult to achieve high accuracy.

Which variables are significant?

When one sees the output of linear regression one can see the lower the p value more significant is the variable. So, the most significant variables according to our linear model are Number of students on the bus, BoroAllBoroughs, BoroBrooklyn, ReasonAccident.

R-Code:

```
```{r NYbus}
library(ggplot2)
library(dplyr)
library(caTools)
library(MASS)
library(plyr)
library(hflights)

nybuss <- "/Users/Amy/R_Homework/PROJECT/bus-breakdown-and-delays.csv"
df <- read.csv(nybuss)
View(df)
```

```{r NYbus_1}
School Bus Delay Places(Boro)
ggplot(data=df, aes(x = df$Boro)) +
 geom_bar(aes(fill=..count.., y = (..count..)/sum(..count..))) +
 geom_text(aes(y = ((..count..)/sum(..count..)), label = scales::percent((..count..)/sum(..count..)), stat =
"count", vjust = -0.25) +
 scale_fill_gradient("Count", low="green", high="red")+labs(title = "School Bus Delay Places", y =
"Percentage", x = "Boro")
```

```{r NYbus_2}
A breakdown or delay occurred on a specific category of busing service(Run_Type)
ggplot(data=df, aes(x = df$Run_Type)) +
 geom_bar(aes(fill= Run_Type, y = (..count..)/sum(..count..))) +
 geom_text(aes(y = ((..count..)/sum(..count..)), label = scales::percent((..count..)/sum(..count..)), stat =
"count", vjust = -0.25) +
 labs(title = "A Breakdown or Delay Occurred on a Specific Category of Busing Service", y =
"Percentage", x = "Timing")
```

```{r NYbus_3}
the relationship between year and Boro
df1 <- tbl_df(df)
class(df1)
df1

b_df1 <- filter(df1, School_Year == "2015-2016", Boro == "Staten Island")
b_df2 <- filter(df1, School_Year == "2015-2016", Boro == "Bronx")
b_df3 <- filter(df1, School_Year == "2015-2016", Boro == "Nassau County")
b_df4 <- filter(df1, School_Year == "2015-2016", Boro == "Brooklyn")
b_df5 <- filter(df1, School_Year == "2015-2016", Boro == "Connecticut")
b_df6 <- filter(df1, School_Year == "2015-2016", Boro == "Manhattan")
```

```

b_df7 <- filter(df1, School_Year == "2015-2016", Boro == "New Jersey")
b_df8 <- filter(df1, School_Year == "2015-2016", Boro == "Queens")
b_df9 <- filter(df1, School_Year == "2015-2016", Boro == "Rockland County")
b_df10 <- filter(df1, School_Year == "2015-2016", Boro == "Westchester")

count(b_df1)
count(b_df2)
count(b_df3)
count(b_df4)
count(b_df5)
count(b_df6)
count(b_df7)
count(b_df8)
count(b_df9)
count(b_df10)

The 10 Places of School Buses Delayed during 2015-2016
slices <- c(2765, 17120, 835, 15639, 26, 11570, 358, 9874, 235, 2113)
lbls <- c("Staten Island", "Bronx", "Nassau County", "Brooklyn", "Connecticut", "Manhattan", "New
Jersey", "Queens", "Rockland County", "Westchester")
pct <- round(slices/sum(slices)*100, digits = 2)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls, col=rainbow(10),
 main="The 10 Places of School Buses Delayed during 2015-2016")
...

```{r NYbus_4}
b1_df1 <- filter(df1, School_Year == "2016-2017", Boro == "Staten Island")
b1_df2 <- filter(df1, School_Year == "2016-2017", Boro == "Bronx")
b1_df3 <- filter(df1, School_Year == "2016-2017", Boro == "Nassau County")
b1_df4 <- filter(df1, School_Year == "2016-2017", Boro == "Brooklyn")
b1_df5 <- filter(df1, School_Year == "2016-2017", Boro == "Connecticut")
b1_df6 <- filter(df1, School_Year == "2016-2017", Boro == "Manhattan")
b1_df7 <- filter(df1, School_Year == "2016-2017", Boro == "New Jersey")
b1_df8 <- filter(df1, School_Year == "2016-2017", Boro == "Queens")
b1_df9 <- filter(df1, School_Year == "2016-2017", Boro == "Rockland County")
b1_df10 <- filter(df1, School_Year == "2016-2017", Boro == "Westchester")

count(b1_df1)
count(b1_df2)
count(b1_df3)
count(b1_df4)
count(b1_df5)
count(b1_df6)

```



```

count(b1_df7)
count(b1_df8)
count(b1_df9)
count(b1_df10)

# The 10 Places of School Buses Delayed during 2016-2017
slices_1 <- c(4114, 23389, 1103, 19637, 75, 16761, 526, 11050, 246, 2343)
lbls <- c("Staten Island", "Bronx", "Nassau County", "Brooklyn", "Connecticut", "Manhattan", "New
Jersey", "Queens", "Rockland County", "Westchester")
pct <- round(slices_1/sum(slices_1)*100, digits = 2)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices_1,labels = lbls, col=rainbow(10),
    main="The 10 Places of School Buses Delayed during 2016-2017")
```



```

```{r NYbus_5}
b2_df1 <- filter(df1, School_Year == "2017-2018", Boro == "Staten Island")
b2_df2 <- filter(df1, School_Year == "2017-2018", Boro == "Bronx")
b2_df3 <- filter(df1, School_Year == "2017-2018", Boro == "Nassau County")
b2_df4 <- filter(df1, School_Year == "2017-2018", Boro == "Brooklyn")
b2_df5 <- filter(df1, School_Year == "2017-2018", Boro == "Connecticut")
b2_df6 <- filter(df1, School_Year == "2017-2018", Boro == "Manhattan")
b2_df7 <- filter(df1, School_Year == "2017-2018", Boro == "New Jersey")
b2_df8 <- filter(df1, School_Year == "2017-2018", Boro == "Queens")
b2_df9 <- filter(df1, School_Year == "2017-2018", Boro == "Rockland County")
b2_df10 <- filter(df1, School_Year == "2017-2018", Boro == "Westchester")

count(b2_df1)
count(b2_df2)
count(b2_df3)
count(b2_df4)
count(b2_df5)
count(b2_df6)
count(b2_df7)
count(b2_df8)
count(b2_df9)
count(b2_df10)

The 10 Places of School Buses Delayed during 2017-2018
slices_2 <- c(5994, 20282, 1343, 18367, 68, 23390, 521, 12294, 305, 1931)
lbls <- c("Staten Island", "Bronx", "Nassau County", "Brooklyn", "Connecticut", "Manhattan", "New
Jersey", "Queens", "Rockland County", "Westchester")
pct <- round(slices_2/sum(slices_2)*100, digits = 2)

```


```

```

lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices_2,labels = lbls, col=rainbow(10),
    main="The 10 Places of School Buses Delayed during 2017-2018")
```

```{r NYbus_6}
b3_df1 <- filter(df1, School_Year == "2018-2019", Boro == "Staten Island")
b3_df2 <- filter(df1, School_Year == "2018-2019", Boro == "Bronx")
b3_df3 <- filter(df1, School_Year == "2018-2019", Boro == "Nassau County")
b3_df4 <- filter(df1, School_Year == "2018-2019", Boro == "Brooklyn")
b3_df5 <- filter(df1, School_Year == "2018-2019", Boro == "Connecticut")
b3_df6 <- filter(df1, School_Year == "2018-2019", Boro == "Manhattan")
b3_df7 <- filter(df1, School_Year == "2018-2019", Boro == "New Jersey")
b3_df8 <- filter(df1, School_Year == "2018-2019", Boro == "Queens")
b3_df9 <- filter(df1, School_Year == "2018-2019", Boro == "Rockland County")
b3_df10 <- filter(df1, School_Year == "2018-2019", Boro == "Westchester")

count(b3_df1)
count(b3_df2)
count(b3_df3)
count(b3_df4)
count(b3_df5)
count(b3_df6)
count(b3_df7)
count(b3_df8)
count(b3_df9)
count(b3_df10)

# The 10 Places of School Buses Delayed during 2018-2019
slices_3 <- c(613, 4828, 336, 4073, 22, 7836, 86, 2465, 54, 441)
lbls <- c("Staten Island", "Bronx", "Nassau County", "Brooklyn", "Connecticut", "Manhattan", "New
Jersey", "Queens", "Rockland County", "Westchester")
pct <- round(slices_3/sum(slices_3)*100, digits = 2)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices_3,labels = lbls, col=rainbow(10),
    main="The 10 Places of School Buses Delayed during 2018-2018/10")
```

```{r NYbus_7}
# the relationship between year and Reason
df3 <- df[-35580, ]
View(df3)

```

```

ggplot(data=df3, aes(x = School_Year)) +
  geom_bar(aes(fill=Reason, y = ..count..))+
  geom_text(aes(y = (..count..), label = scales::number(..count..), stat = "count", vjust = -0.25) +
  labs(title = "The Reason of Delay during 2015-Present", x = "Year")
```

```{r NYbus_8}
# Breakdown_or_Running_Late
f_df11 <- filter(df1, Breakdown_or_Running_Late == "Breakdown")
f_df12 <- filter(df1, Breakdown_or_Running_Late == "Running Late")
count(f_df11)
count(f_df12)

slices_4 <- c(29103, 228613)
lbls <- c("Breakdown", "Running Late")
pct <- round(slices_4/sum(slices_4)*100, digits = 2)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices_4,labels = lbls, col=rainbow(10),
  main="Percentage of Delay Reasons by Bus Vendor")
```

```{r NYbus_9}
## the relationship between year and delay mean
library(dplyr, warn.conflicts = F)
library(hflights)
df$How_Long_Delayed <- gsub("MINUTES","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MIN","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MINS","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("minutes","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mins","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("min","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Minutes","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Min","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("S","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mns","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MN","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mn","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mnis","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Mn","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Mns","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-15 MIIN","12.5",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("15 to 20m","17",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("m","",df$How_Long_Delayed)

```

```

df$How_Long_Delayed <- gsub("MIIN","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("M","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-12","11",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("20-30","25",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("20-25","22",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-15","12",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("25-30","27",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-20","15",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("15-20","17",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("30-60","45",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("45-60","52",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 hour","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 HR","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 hr","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("2HR","120",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1hr","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 1/2HR","90",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1HOUR","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 Hour","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("i","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("I","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("u","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("U","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("t","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("T","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("e","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("E","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("s","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("S","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("0.", "0",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("0 .", "0",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("5 .", "5",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("2 .", "2",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("5.", "5",df$How_Long_Delayed)
##
df$How_Long_Delayed <- as.numeric(df$How_Long_Delayed)
df6 <- filter(df,How_Long_Delayed!="NA")
df6 <- filter(df,How_Long_Delayed <=48 & How_Long_Delayed > 0.55)
...

```{r NYbus_9.1}
df_2 <- tbl_df(df6)
class(df_2)
df_2

```

```

bus <- group_by(df_2, School_Year)
bus

delaybus <- dplyr::summarise(bus, count = n(), delay = mean(How_Long_Delayed, na.rm = T))
delaybus

df.delaybus <- arrange(delaybus, desc(delay))
df.delaybusN <- df.delaybus[,-2]
df.delaybusN
...

```{r NYbus_9.2}
df.delaybus_1 <- round(df.delaybus$delay, digits = 2)

ggplot(data=df.delaybus, aes(x= df.delaybus$School_Year, y = df.delaybus$delay)) +
  geom_bar(fill = rainbow(4), stat="identity", position=position_dodge())+
  geom_text(aes(label= df.delaybus_1, vjust=-0.75, color="brown")+
  scale_fill_brewer(palette="Paired")+
  theme_minimal()+
  labs(title = "The Relationship between Year and Average Delay Time", y = "Minutes", x = "Year")
...

```{r NYbus_10}
the relationship between Boro and delay mean
borogroup <- group_by(df_2, Boro)
borogroup

borogrouppdelay <- dplyr::summarise(borogroup, count = n(), delay = mean(How_Long_Delayed, na.rm =
T))
borogrouppdelay
borogrouppdelay1 <- borogrouppdelay[-1:-2,]

df.borogrouppdelay <- arrange(borogrouppdelay1, desc(delay))
df.borogrouppdelayN <- df.borogrouppdelay[,-2]
df.borogrouppdelayN
...

```{r NYbus_10.1}
df.borogrouppdelay_1 <- round(df.borogrouppdelayN$delay, digits = 2)

ggplot(data=df.borogrouppdelay, aes(x= df.borogrouppdelay$Boro, y = df.borogrouppdelay$delay)) +
  geom_bar(fill= rainbow(10), stat="identity", position=position_dodge())+
  geom_text(aes(label= df.borogrouppdelay_1, vjust=-0.75, color="brown")+
  theme_minimal()+
  labs(title = "The Relationship between Boro and Average Delay Time", y = "Minutes", x = "Boro")
...

```

```

```{r NYbus_11}
the relationship between Reason and delay mean
reasongroup <- group_by(df_2, Reason)
reasongroup

reasongroupdelay <- dplyr::summarise(reasongroup, count = n(), delay = mean(How_Long_Delayed,
na.rm = T))
reasongroupdelay

df.reasongroupdelay <- arrange(reasongroupdelay, desc(delay))
df.reasongroupdelayN <- df.reasongroupdelay[, -2]
df.reasongroupdelayN
```

```{r NYbus_11.1}

df.reasongroupdelay_1 <- round(df.reasongroupdelay$delay, digits = 2)

ggplot(data=df.reasongroupdelay, aes(x= df.reasongroupdelay$Reason, y = df.reasongroupdelay$delay))
+
 geom_bar(fill=rainbow(10), stat="identity", position=position_dodge())+
 geom_text(aes(label= df.reasongroupdelay_1, vjust=-0.75, color="brown")+
 theme_minimal()+
 labs(title = "The Relationship between Reason and Average Delay Time", y = "Minutes", x = "Reason")
```

```{r NYbus_12}
df <- read.csv(nybuss)
str(df)

#dimension reduction
df <- df[c(-2,-9,-14,-15,-16,-17,-18,-19)]
str(df)

##dividing the data into two groups with how_long_delayed as NULL and one with NOT NULL
##Cant use how_long_delayed as NULL in test/train, but can be used as final test dataset

df.delay <- filter(df, How_Long_Delayed == "")
df <- filter(df, How_Long_Delayed != "")

#setting the Null values to NA
df[df == ""] = NA

getting the sum of NA values

```

```

sum(is.na(df))

#converting the Boro column to character

df$Boro <- as.character(df$Boro)

#converting NA values to readable character FF
df$Boro[which(is.na(df$Boro))] <- "FF"

Eliminate the NULL replaced columns
#making a temp dataframe with all the boro with FF
df.temp <- filter(df,Boro == "FF")
View(df.temp)
summary(df.temp)

#converting the route number to character and levelling them as per description of details about this
dataset

df$Route_Number <- as.character(df$Route_Number)

df$Route.No.veled <- nchar(df$Route_Number)
df$Route.No.veled <- gsub("4","curb-to-curb",df$Route.No.veled)
df$Route.No.veled <- gsub("5","stop-to-school",df$Route.No.veled)
df$Route.No.veled <- gsub("1","Pre-K/EI Route",df$Route.No.veled)
df$Route.No.veled <- gsub("2","Pre-K/EI Route",df$Route.No.veled)
df$Route.No.veled <- gsub("3","Pre-K/EI Route",df$Route.No.veled)

#Removing Route_Number column as its levelled
df <- df[,-4]

#filtering out schools with 0 and ` values

df <- filter(df,Schools_Serviced!="0")
df <- filter(df,Schools_Serviced!="")

#taking the length of values for schools serviced column
df$Schools_Serviced <- as.character(df$Schools_Serviced)

df$Schools.Srvcd.length <- nchar(df$Schools_Serviced)

df$num.char <- substr(df$Schools_Serviced,1,1)

#levelling the schools serviced based on the length and number of alphabets in schools serviced column
understood pattern recognition from the below mentioned site

```

```

#got code from http://www.jdatalab.com/data_science_and_data_mining/2017/03/20/regular-expression-R-part11.html for pattern used within grepl
df$Schools.Srvcd.lvlId <- ifelse(grepl("[0-9]+[.]?[0-9]|[-]?[0-9]+[L]?|[-]?[0-9]+[.]?[0-9][eE][0-9]+",df$num.char)==T,"school-aged service","Pre-K/El service")

df$Schools.Srvcd.lvlId <- ifelse(df$Schools.Srvcd.length>5,"Multiple Sites",df$Schools.Srvcd.lvlId)

df <- df[c(-5,-14,-15)]

Getting the time component(with AM/PM) of bus breakdown occurrence time

df$timecomponent <- substr(df$Occurred_On,12,13)
df$AMPMcomp <- substr(df$Occurred_On,21,22)
df$timecomponent <- as.numeric(df$timecomponent)

#creating a dummy variable for Boro using model.matrix and then dropping boro

dummy_Boro<- model.matrix(~Boro -1, data=df)
df<- cbind(df,-6],dummy_Boro)

#creating a dummy variable for Reason

df$Reason <- as.character(df$Reason)
df$Reason[is.na(df$Reason)] <- "OTHER"
df$Reason=as.factor(df$Reason)
df$Reason
dummy_Reason<- model.matrix(~Reason -1, data=df)
df<- cbind(df,-4],dummy_Reason)

#Taking only the last part of bus companyname to easily dummify the data later
#source from http://www.jdatalab.com/data_science_and_data_mining/2017/03/20/regular-expression-R-part11.html
ptn <- "(.*?)"
df$abvtd_bcmpnme <- gsub(ptn, "", df$Bus_Company_Name)
df$abvtd_bcmpnme <- as.factor(df$abvtd_bcmpnme)

#Levelling the below mentioned column for linear regression
levels(df$School_Age_or_PreK) <- c(1,2)
levels(df$Breakdown_or_Running_Late) <- c(1,2)
df$Breakdown_or_Running_Late <- as.numeric(df$Breakdown_or_Running_Late)

Removing factored columns
df <- df[c(-4,-5)]

```



```

Levelling AM/PM component
df$AMPMcomp <- as.factor(df$AMPMcomp)
levels(df$AMPMcomp) <- c(1,2)

Creating dummy variabe for route number levelled through model.matrix
dummy_Route.No.Lvld<- model.matrix(~Route.No.levelled -1, data=df)
df<- cbind(df[, -8], dummy_Route.No.Lvld)
#View(df)
Creating dummy variabe for school service levelled through model.matrix
df$Schools.Srvcd.lvld <- as.factor(df$Schools.Srvcd.lvld)
dummy_Schools.Srvcd.Lvld<- model.matrix(~Schools.Srvcd.lvld -1, data=df)
df<- cbind(df[, -8], dummy_Schools.Srvcd.Lvld)

df$timecomponent <- as.numeric(df$timecomponent)

#creating dummy variable for school year
dummy_School.Year<- model.matrix(~School_Year -1, data=df)
df<- cbind(df[, -1], dummy_School.Year)

options(na.action='na.pass')
#View(df)
Creating dummy variabe for Run_Type levelled through model.matrix
dummy_Run.Type<- model.matrix(~Run_Type -1, data=df)
df<- cbind(df[, -1], dummy_Run.Type)
df <- df[, -42] # removing 2018-2019

#how long should be numeric, but it is character, therefore removing characters to get only numbers
#and to get good number of rows to test & train our dataset

df$How_Long_Delayed <- gsub("MINUTES", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MIN", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MINS", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("minutes", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mins", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("min", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Minutes", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Min", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("S", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mns", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MN", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mn", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("mnis", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Mn", "", df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("Mns.", "", df$How_Long_Delayed)

```

```

df$How_Long_Delayed <- gsub("10-15 MIIN","12.5",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("15 to 20m","17",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("m","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("MIIN","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("M","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-12","11",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("20-30","25",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("20-25","22",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-15","12",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("25-30","27",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("10-20","15",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("15-20","17",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("30-60","45",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("45-60","52",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 hour","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 HR","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 hr","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("2HR","120",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1hr","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 1/2HR","90",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1HOUR","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("1 Hour","60",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("i","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("I","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("u","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("U","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("t","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("T","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("e","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("E","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("s","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("S","",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("0.", "0",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("0 .", "0",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("5 .", "5",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("2 .", "2",df$How_Long_Delayed)
df$How_Long_Delayed <- gsub("5.", "5",df$How_Long_Delayed)

```

```

df$How_Long_Delayed <- as.numeric(df$How_Long_Delayed)
df <- filter(df,How_Long_Delayed!="NA")
df <- filter(df,How_Long_Delayed <=48 & How_Long_Delayed > 0.55)

```

```

df$School_Age_or_PreK <- as.numeric(df$School_Age_or_PreK)

```

```

df$AMPMcomp <- as.numeric(df$AMPMcomp)

dummy_abvtd_bcmprnme<- model.matrix(~abvtd_bcmprnme -1, data=df)
df<- cbind(df[, -30],dummy_abvtd_bcmprnme)

df <- df[complete.cases(df[,]),]

skgendf <- data.frame(skgen=1:9924)
dfjoin <- data.frame(Bus_No=sort(unique(df$Bus_No)))
dfjoiner <- cbind(skgendf,dfjoin)
df <- merge(dfjoiner,df)
df <- df[,-1]
names(df)[names(df)=="skgen"] <- "Bus_No.sk"

library(dplyr)
df %>% select_if(~!is.numeric(.x)) %>% head()
df <- df[,sapply(df, is.numeric)]
df.corr <- cor(df)
df.corr
df <- df[c(-5,-32,-33,-35,-36,-41,-44)]
...

```{r NYbus_12.1}
library(caTools)
set.seed(101)
#splitting the dataset into test/train dataset
sample <- sample.split(df$How_Long_Delayed,SplitRatio = 0.7)
df.train <- subset(df,sample == TRUE)
df.test <- subset(df,sample == FALSE)
View(df.test)
#Linear regression model application.
model.df <- lm(How_Long_Delayed ~., df.train)
library(MASS)
model.df.step <- stepAIC(model.df,direction="both")
class(model.df.step)
summary(model.df.step)
#summary(model.df)
...

```{r NYbus_12.2}
df.predict <- predict(model.df.step,df.test)
#View(df.predict)
results <- cbind(df.predict,df.test$How_Long_Delayed)
colnames(results) <- c('pred','real')

```

```

results <- as.data.frame(results)
results
```

```{r NYbus_13}
chooseCRANmirror(graphics=FALSE, ind=1)
knitr::opts_chunk$set(echo = TRUE)
Generate Lift Chats
library(gains)
install.packages("gains")
df.predict <- predict(model.df.step,df.test)
View(df.predict)

gain <- gains(df.test$How_Long_Delayed, df.predict)

#Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(df.test$How_Long_Delayed))~c(0,gain$cume.obs),
 xlab = "# cases", ylab = "Cumulative", main = "", type = "l")
lines(c(0,sum(df.test$How_Long_Delayed))~c(0, dim(df.test)[1]), lty = 5)
```

```{r NYbus_14}
Plot decile-wise chart
heights <- gain$mean.resp/mean(df.test$How_Long_Delayed)
decile_lift <- barplot(heights, names.arg = gain$depth, ylim = c(0,3), col = "gold3",
 xlab = "Percentile", ylab = "Mean Response",
 main = "Decile-wise lift chart")
```

```{r NYbus_15}
#residual
some.residuals <- results$real[1:500] - results$pred[1:500]

plot(some.residuals, type = "p", pch = 16,
 col = "blue1",
 ylab = "Sample Residuals",
 ylim = c(-50, 50), bty = "n"
)
```

```{r NYbus_16}
install.packages("caret")
library(caret)

RMSE(df.predict, df.test$How_Long_Delayed)

```

### Recommendations:

In near future queuing algorithm could have been applied to predict delay times of buses. There were too many missing values these are human errors and could be improved in the near future for better analysis of delay times.

Try finding alternate route of traversing to and from the schools especially in Bronx and Brooklyn and Manhattan according to our data exploratory analysis.

One more thing while manually entering the delay times a fixed format could have being used.

### Reference:

1. Krzanowsk WJ. Principles of multivariate analysis: a user's perspective. Oxford, England: Clarendon, 1988.
2. Goldman RN, Weinberg JS. Statistics: an introduction. Upper Saddle River, NJ: Prentice Hall, 1985; 72–98.
3. Seber GAF. Linear regression analysis. New York, NY: Wiley, 1997; 48–51.
4. Fisher RA. Frequency distributions of the values of the correlation coefficient Biometrika 1915; 10:507–521.
5. Carroll RJ, Ruppert D. Transformation and weighting in regression. New York, NY: Chapman & Hall, 1988; 2
6. Mudholkar GS, McDermott M, Scrivastava DK. A test of p-variate normality. Biometrika 1992; 79:850–854

## Individual report

by Aishwarya Badlani

---

### Team members:

1. *Avinav Pandey*
2. *Aishwarya Badlani*
3. *Yu-Yang Hung*
4. *Yi-Ying Lin*

### Group Dynamics

The group project was a journey of learning, challenges and some conflicts. Learning has been there right from the project's initial phase, which was choosing the dataset to work on. I gained knowledge on how to pick a dataset, which is challenging, at the same time suitable. After hours of discussion, as per our team's priority we decided to work on the NY Bus breakdown and delays dataset, which was sourced by Kaggle. We later realized that the dataset we selected was full on missing values but we decided to take this challenge and learn from this experience. The dataset was fun as well as challenging. Since we were people from different background we even learned about different perspective and different approach. We were also able to practically implement things learned in class to a practice problem.

### Success and Challenges

We were successful in predicting how long the buses are delayed and understanding which factors plays important role in predicting the delays. With regards to challenges, it was difficult to prepare the data for further use but we collectively accepted the challenge without changing the dataset which consumed a lot of time. We tried applying different algorithms like knn, clustering, decision trees on target variables but due to limitation of algorithms like in random forest only 53 variables or elements are allowed. In decision tree the fully-grown tree was not giving any good rules. So similar problems came with decision trees and knn. Due to our complex dataset we were unable to do so. We also thought of doing classification on different variables but we concluded linear model will be the best for our dataset and should stick to prediction. We had a long discussion over what model to apply apart from linear regression to predict the delays. We searched the Internet and came to know that queuing could be applied. We tried to implement it, but failed so choose against it and used linear model, as our dataset was highly categorical in nature.

### Experience and Contribution

YU-YANG HUAG and I contributed towards the group by streamlining the flow of the meeting, Exploratory Data analysis by plotting graphical representations each, working on algorithm part by applying Linear regression and Analysis of the algorithm which included lift chart, decile chart and residuals graph and results section of the group report. The coding part of these was easy compared to analyzing the output of the regression. We tried to interpret the P values, linear regression coefficients for which I took use of the class notes and presentations. I went through various examples online to understand the various interpretations. Avinav & YI-YING contributed toward data cleaning, Dimension reduction, and exploratory data analysis. We divided the data visualization part among all team members. All the team members contributed towards report creation in professional quality.