Analysis of New York Bus Data in December 2017

Yaya Liu

1 Introduction

The dataset is about New York bus running record from December 1, 2017 through December 31, 2017 (Stone, 2017), which is under business domain. This data can be used to measure the performance of each bus line and provide thoughts on improving the performance of bus network.

The name of the dataset is “mta_1712.csv”. It requires 1.24 GB to store. It has 6462122 rows and 17 columns. This dataset contains historical records for all buses running in New York in December,2017. Each record contains the time of observation, the name of the bus line, bus location, bus route, next stop, distance from that stop, and other variables described below (Stone, 2017).

Following is the briefly data type analysis of each column (MTA, n.d.).

- RecordedAtTime: the date and time of the observation. The format is "YYYY -MM-DD HH: MM: SS". It provides the ranked order of when the information has been recorded. It is ordinal data. Data mode is character.
- Direction Ref: the direction that the bus is going. The value is either 0 or 1, which indicates two directions of each bus line. It is binary data. The values of mean, median and quantiles are meaningless.
- PublishedLineName: the name of the bus line. There are 206 different bus lines in the dataset. It is nominal data. Data mode is character.
- OriginName: the name of the starting stop. It is nominal. Data mode is character.
- OriginLat: the latitude of the starting stop. Data type is ratio.
- OriginLong: the longitude of the starting stop. Data type is ratio.
- DestinationName: the name of the destination. It is nominal data. Data mode is character.
- DestinationLat: the latitude of the destination. Data type is ratio.
- DestinationLong: the longitude of the destination. Data type is ratio.
- VehicleRef: the 4-digit ID of the bus. It is nominal data. Data mode is character.
- VehicleLocationLatitude: the current latitude of the bus. Data type is ratio.
- VehicleLocationLongitude: the current longitude of the bus. Data type is ratio.

- NextStopPointName: the next stop the bus will serve. It is nominal data. Data mode is character.
- ArrivalProximityText: text about the arrival proximity. It is nominal data. Data mode is character.
- DistanceFromStop: the distance of the bus (in meters) from that next stop. Data type is ratio.
- ExpectedArrivalTime: the expected arrival time to the next stop. The format is "YYYY-MM-DD HH: MM: SS". It is ordinal data. Data mode is character.
- ScheduledArrivalTime: the scheduled arrival time to the next stop. The format is "HH: MM: SS". It is ordinal data. Data mode is character.

```
> summary.default(df)
                          Length  Class  Mode
RecordedAtTime            6462122 -none- character
DirectionRef              6462122 -none- numeric
PublishedLineName         6462122 -none- character
OriginName                6462122 -none- character
OriginLat                 6462122 -none- numeric
OriginLong                6462122 -none- numeric
DestinationName           6462122 -none- character
DestinationLat            6462122 -none- numeric
DestinationLong           6462122 -none- numeric
VehicleRef                6462122 -none- character
VehicleLocation.Latitude  6462122 -none- numeric
VehicleLocation.Longitude 6462122 -none- numeric
NextStopPointName         6462122 -none- character
ArrivalProximityText      6462122 -none- character
DistanceFromStop          6462122 -none- character
ExpectedArrivalTime       6462122 -none- character
ScheduledArrivalTime      6462122 -none- character
```

2 About the organization collecting data

Metropolitan Transportation Authority

This dataset was originally collected and released by Metropolitan Transportation Authority(MTA). It is “North America's largest transportation network, serving a population of 15.3 million people in the 5,000-square-mile area fanning out from New York City through Long Island, southeastern New York State, and Connecticut.” (MTA, n.d.)

MTA's role is to provide safe, clean, efficient public transportation to the New York City area and "help ensure New York's place as a world center of finance, commerce, culture, and education." (MTA, n.d.)

In addition, I found this dataset from Kaggle, which is "a platform for the world's largest community of data scientists and machine learners" (Kaggle, n.d.). With this platform, users can share high quality datasets, and all users can explore and build models with the datasets. Thus, data scientists and machine learners can connect and learn from each other about new methods. Besides, enterprises are also allowed to host contests in the platform. Data scientists and machine learners can join and challenge their skills. Finding interesting data and transforming raw data into problem-solving insights can be difficult. This is why Kaggle's founder Anthony John Goldbloom created this web-based data-science platform (Kaggle, n.d.).

3 The purpose of collecting the data

MTA generated this data for MTA Mobility application quest. The other purposes of collecting this data is to measure bus performance and provide data to developers who is interested in creating their own apps based on this data.

4 Potential questions could be answered by studying this data

The potential questions could be answered by studying this data are:

- How many bus lines in each borough of New York city?
- What is the average delay time in each borough?
  Which borough has the worst bus service according to the delay time?
- How long will the average delay be for each bus line?
  Was each bus line running to schedule during December,2017?
  Which one has the longest delay in each borough?
- How many bus lines use the Select Bus Service(SBS) route? Do these bus lines are more punctual than bus lines that do not use the SBS route? (The definition about "Select Bus Service(SBS)" can be found in Chapter 15, Technical Terms)
- What are the possible factors related to the average delay time?

5 Hardware requirements

At a minimum, the processing power of a recent model laptop is required for the analysis of this dataset. All analysis for this project was performed in Laptop: Dell Inspiron 13 5000 series. Following are the characteristics of this laptop:

- Processor: Intel Core i7-8550U CPU
- Processor Speed: 1.8GHz
- Number of Processor: 1
- Total number of cores: 8
- L2 Cache (per Core): 1 MB
- L3 Cache: 8MB
- Memory: 8 GB
- Hard Drive: 250 GB

6 Software Requirements

- RStudio 1.1.456 (RStudio Team, 2016) for data exploration, transforms, analysis and visualization. Following packages were used:
    - tidyverse 1.2.1 (Wickham, 2017)
    - data.table 1.11.8 (Dowle & Srinivasan, 2018)
    - chron 2.5-53 (James & Hornik, 2018)
    - geosphere 1.5-7 (Hijmans, 2017)

7 Issues with the dataset

- Privacy:
    - There is no privacy issue about this dataset. It is open to the public and it is free to download.
- Quality:
    - There are a lot of missing values in these columns: the name of the starting stop; the latitude and longitude of the starting stop; the latitude and longitude of the destination and the scheduled arrival time. After dropping all the rows with missing values, the number of rows decreased from 6462122 to 4581067. There are not enough records for express routes that run across different boroughs in New York. Therefore, the following analytics focuses on the bus lines that provide local service in 5 boroughs of New York.

    - In “the scheduled arrival time” column, a few time records are wrong. These time records are not within 00:00:00 to 24:00:00.

- Other issues:
    - There are no records on actual arrival time in this dataset. The delay time has to be calculated from the expected arrival time and the scheduled time. The calculated delay time may not be the same as the actual delay time.
    - There is no data about the actual bus route length and the distance from starting point to the current location. This information has to be calculated by the coordinates, which may not be 100% accurate, especially when the bus route is a loop.

8 Data exploration

RStudio was used for data exploration.

Step1: importing the dataset, changing 2 column names, dropping missing values and getting the statistical summary of the dataset.

After dropping missing values, there are not enough records for express routes, so I dropped the records for express routes and kept the records for bus lines which provide local service in 5 boroughs of New York. Since I am interested in how many bus lines provided local service, I counted the number of bus lines and printed out the name of the bus lines.

```r
library(tidyverse)
library(chron)
library(geosphere)

##Import dataset, total 6462122 rows and 17 columns.
df <- read.csv("C:\\Rdata\\Bussiness_Analytics\\mta_1712.csv", stringsAsFactors = FALSE)
names(df)[names(df) == "VehicleLocation.Latitude"] <- "VehicleLocationLatitude"
names(df)[names(df) == "VehicleLocation.Longitude"] <- "VehicleLocationLongitude"

summary(df)
summary.default(df)
nrow(df)    #6462122 rows

TableLineName <- table(df$PublishedLineName)
AllPublishedLineNames <-  names(TableLineName)
length(AllPublishedLineNames)   #329 bus Lines

##Drop rows including missing values,
MTA_Dec1712 <- na.omit(df)

##Drop records for express routes
MTA_Dec1712 <- dplyr::filter(MTA_Dec1712, !grepl("BM|BxM|M Shuttle Bus|QM|X", MTA_Dec1712$PublishedLineName))

summary(MTA_Dec1712)
nrow(MTA_Dec1712)   #4856508 rows

##Get all published Line names
TableLineName1 <- table(MTA_Dec1712$PublishedLineName)
AllPublishedLineNames1 <-  names(TableLineName1)
length(AllPublishedLineNames1)   #206 bus Lines
```

```
> ##Get all published Line names
> TableLineName1 <- table(MTA_Dec1712$PublishedLineName)
> AllPublishedLineNames1 <-  names(TableLineName1)
> length(AllPublishedLineNames1)   #206 bus Lines
[1] 206
```

```
> AllPublishedLineNames1
  [1] "B1"       "B11"      "B12"      "B13"      "B14"      "B15"      "B16"      "B17"      "B2"
 [10] "B20"      "B24"      "B25"      "B26"      "B3"       "B31"      "B32"      "B35"      "B36"
 [19] "B37"      "B38"      "B39"      "B4"       "B41"      "B42"      "B43"      "B44"      "B44-SBS"
 [28] "B45"      "B46"      "B46-SBS"  "B47"      "B48"      "B49"      "B52"      "B54"      "B57"
 [37] "B6"       "B60"      "B61"      "B62"      "B63"      "B64"      "B65"      "B67"      "B68"
 [46] "B69"      "B7"       "B70"      "B74"      "B8"       "B82"      "B83"      "B84"      "B9"
 [55] "Bx1"      "Bx10"     "Bx11"     "Bx12"     "Bx12-SBS" "Bx13"     "Bx15"     "Bx16"     "Bx17"
 [64] "Bx18"     "Bx19"     "Bx2"      "Bx20"     "Bx21"     "Bx22"     "Bx24"     "Bx26"     "Bx27"
 [73] "Bx28"     "Bx29"     "Bx3"      "Bx30"     "Bx31"     "Bx32"     "Bx33"     "Bx34"     "Bx35"
 [82] "Bx36"     "Bx38"     "Bx39"     "Bx4"      "Bx40"     "Bx41"     "Bx41-SBS" "Bx42"     "Bx46"
 [91] "Bx4A"     "Bx5"      "Bx6"      "Bx6-SBS"  "Bx7"      "Bx8"      "Bx9"      "M1"       "M10"
[100] "M100"     "M101"     "M102"     "M103"     "M104"     "M106"     "M11"      "M116"     "M12"
[109] "M14A"     "M14D"     "M15"      "M15-SBS"  "M2"       "M20"      "M21"      "M22"      "M23-SBS"
[118] "M3"       "M31"      "M34-SBS"  "M34A-SBS" "M35"      "M4"       "M42"      "M5"       "M50"
[127] "M55"      "M57"      "M60-SBS"  "M66"      "M7"       "M72"      "M79-SBS"  "M8"       "M86-SBS"
[136] "M9"       "M96"      "M98"      "Q1"       "Q12"      "Q13"      "Q15"      "Q15A"     "Q16"
[145] "Q17"      "Q2"       "Q20A"     "Q20B"     "Q24"      "Q26"      "Q27"      "Q28"      "Q3"
[154] "Q30"      "Q31"      "Q32"      "Q36"      "Q4"       "Q42"      "Q43"      "Q44-SBS"  "Q46"
[163] "Q48"      "Q5"       "Q54"      "Q55"      "Q56"      "Q58"      "Q59"      "Q76"      "Q77"
[172] "Q83"      "Q84"      "Q85"      "Q88"      "S40"      "S42"      "S44"      "S46"      "S48"
[181] "S51"      "S52"      "S53"      "S54"      "S55"      "S56"      "S57"      "S59"      "S61"
[190] "S62"      "S66"      "S74"      "S76"      "S78"      "S79-SBS"  "S81"      "S84"      "S86"
[199] "S89"      "S90"      "S91"      "S92"      "S93"      "S94"      "S96"      "S98"
>
```

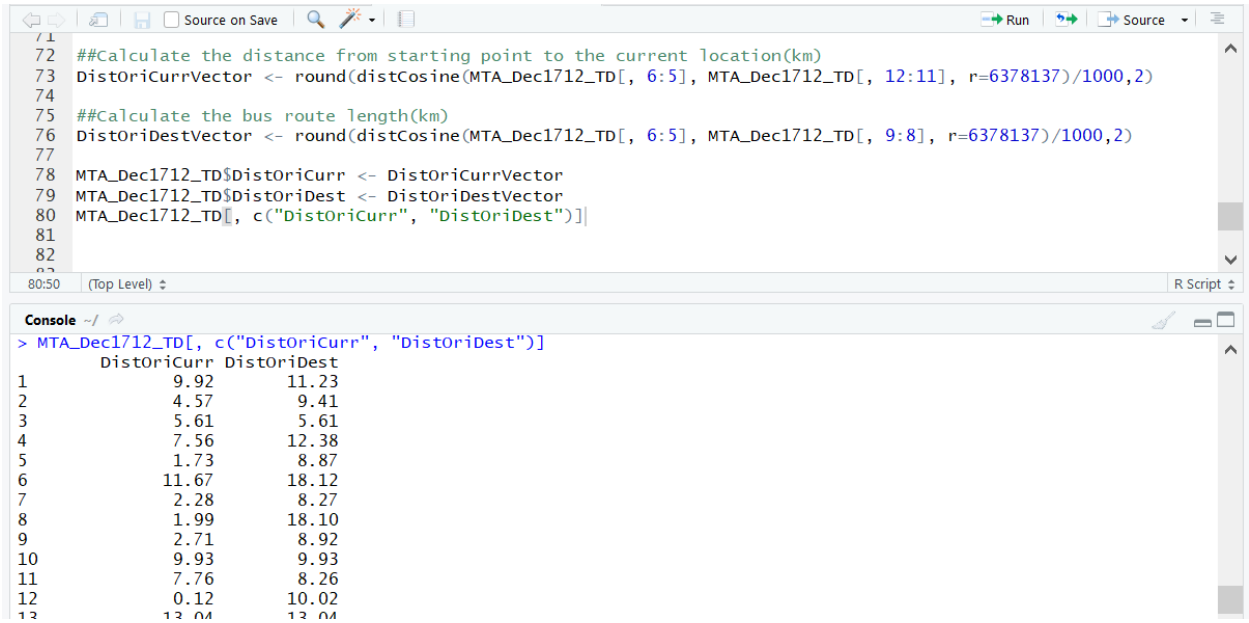
Step2: Because I am interested in the delay time from 3:00 am to 23:00 pm, I filtered out the records during this time, calculated the delay time (seconds) for each record based on the expected arrival time and the scheduled arrival time.

Then I created a new column (“Delay”) in the dataset to store the delay time for later use.

```
33
34 ##Calculate the scheduled Hours, Minutes and Seconds.
35 MTA_Dec1712_TD <- filter(MTA_Dec1712, hours(RecordedAtTime) >= 3 & hours(RecordedAtTime) < 23)
36
37 Bus_scheduled_hours <- hours(strptime(MTA_Dec1712_TD$ScheduledArrivalTime,
38                                        format = "%H:%M:%S",
39                                        tz = "America/New_York"))
40 Bus_scheduled_minutes <- minutes(strptime(MTA_Dec1712_TD$ScheduledArrivalTime,
41                                           format = "%H:%M:%S",
42                                           tz = "America/New_York"))
43 Bus_scheduled_seconds <- seconds(strptime(MTA_Dec1712_TD$ScheduledArrivalTime,
44                                           format = "%H:%M:%S",
45                                           tz = "America/New_York"))
46
47
48 ##Store the the scheduled Hours, Minutes and Seconds to the data frame temporarily
49 MTA_Dec1712_TD$ScheduledHours <- Bus_scheduled_hours
50 MTA_Dec1712_TD$ScheduledMinutes <- Bus_scheduled_minutes
51 MTA_Dec1712_TD$ScheduledSeconds <- Bus_scheduled_seconds
52
53 MTA_Dec1712_TD <- na.omit(MTA_Dec1712_TD)
54
55 ##Get expected arrival hours, minutes and seconds
56 Bus_expected_hours <- hours(MTA_Dec1712_TD$ExpectedArrivalTime)
57 Bus_expected_minutes <- minutes(MTA_Dec1712_TD$ExpectedArrivalTime)
58 Bus_expected_seconds <- seconds(MTA_Dec1712_TD$ExpectedArrivalTime)
59
60 ##Calculate delay time based on the expected arrival time and the scheduled arrival time
61 MTA_Dec1712_TD$Delay <- 3600 * (Bus_expected_hours - MTA_Dec1712_TD$ScheduledHours) +
62   60 * (Bus_expected_minutes - MTA_Dec1712_TD$ScheduledMinutes) +
63   (Bus_expected_seconds - MTA_Dec1712_TD$ScheduledSeconds)
64
65 MTA_Dec1712_TD$Delay <- round(MTA_Dec1712_TD$Delay, 0)
66
67 ##Remove temporial columns
68 MTA_Dec1712_TD[ ,c('ScheduledHours', 'ScheduledMinutes', 'ScheduledSeconds')] <- list(NULL)
69
```

```
Console ~/
>
> MTA_Dec1712_TD$Delay
   [1]   -92   342   167   -13    71    43   -38    -5  -135   345   521    59   931   165   729   165     3    67   314
  [20]   -19   405  -250   129   201   103   111   365   -60   490   950    -8   133   292    81   348   337    86   -15
  [39]   173   204   -15   432   364  -257   -28  -211   -53    19   126   405  -195  -171   115   -39   130   289  -194
  [58]   324    24   541    38  -129   -15  -324   113  1339    50   645   116   122   224   109  -155    87  1402    89
  [77]     7  -141   -77   213   220    91   125    97   -44   -84   -51   384  -231   -72    74    -4   130   927   -87
  [96]   166   -75    -7    80  -151   -75   222   225    25   -24   -32   -44   587   216   -82   -46    27   -75  -112
 [115]   220   107   -18   165    11    51   119    90   357   502    78   302   405  -201   140   -75   223   319  -236
 [134]   607  -157   -48   230   287    45   861   -21  -205   186   449  -101   -54   243   126   -52   -66   -60   -91
 [153]     3  1026   196   121    40   745   786   187  -248   350   546  1039   162  -110   126    -5    71   384  -177
 [172]    56   522   948    18   438    36   232     6   284  -206   208    19   159   123   172   -34   234  -275    65
 [191]   -76  -138  -303   246    66   126  -174   275    36  -254   163   206   846   445   -98   318  -109    33   306
 [210]  1266   385    31   -55   122   -25  -123   -39    64    48    56   126  -174   -74  -167   186    95    82   147
 [229]    50   570   315   -83  -109    39    11  -835   110    45    73  1038  1746   154    94    31   158  -114  -196
```

Step3: For each record, I used the function distCosine() to calculate the distance from the starting point to the current location based on coordinates of the starting point and the current location, and the route of the bus length based on the coordinates of the starting point and the destination. After calculating the results, I created 2 new columns (“DistOriCurr” and “DistOriDest”) in the dataset to store them respectively.



```
##Calculate the distance from starting point to the current location(km)
DistOriCurrVector <- round(distCosine(MTA_Dec1712_TD[, 6:5], MTA_Dec1712_TD[, 12:11], r=6378137)/1000,2)

##Calculate the bus route length(km)
DistOriDestVector <- round(distCosine(MTA_Dec1712_TD[, 6:5], MTA_Dec1712_TD[, 9:8], r=6378137)/1000,2)

MTA_Dec1712_TD$DistOriCurr <- DistOriCurrVector
MTA_Dec1712_TD$DistOriDest <- DistOriDestVector
MTA_Dec1712_TD[, c("DistOriCurr", "DistOriDest")]
```

```
> MTA_Dec1712_TD[, c("DistOriCurr", "DistOriDest")]
        DistOriCurr DistOriDest
1              9.92       11.23
2              4.57        9.41
3              5.61        5.61
4              7.56       12.38
5              1.73        8.87
6             11.67       18.12
7              2.28        8.27
8              1.99       18.10
9              2.71        8.92
10             9.93        9.93
11             7.76        8.26
12             0.12       10.02
```

Step4: I generated a sequence from 1 to 4581067, then stored this sequence in the first column of the dataset. It will serve as the primary key to create a table by using SQL.

```
PrimaryKey <- seq(1, nrow(MTA_Dec1712_TD), by = 1)
MTA_Dec1712_TD$PrimaryKey <- PrimaryKey
MTA_Dec1712_TD <- MTA_Dec1712_TD[, c(21, 1:20)]
```

9 Metadata definitions and statistical summary

After data exploration, the new dataset has 4581067 rows and 21 columns. It requires 1.07 GB to store. Following is the data dictionary and statistical summary for each column.

- PrimaryKey: serves as the primary key. A sequence from 1 to 4581067. It is ordinal data. The values of mean, median and quantiles are meaningless.

- RecordedAtTime: the date and time of the observation. The format is "YYYY -MM-DD HH: MM: SS". It provides the ranked order of when the information has been recorded. It is ordinal data. Data mode is character.

- Direction Ref: the direction that the bus is going. The value is either 0 or 1, which indicates two directions of each bus line. It is binary data. The values of mean, median and quantiles are meaningless.

- PublishedLineName: the name of the bus line. There are 206 different bus lines in the dataset. It is nominal data. Data mode is character.
  Bus lines with the Top10 frequency :

Bus line	B6	B41	Q58	B35	Q44-SBS	Bx36	M101	Q27	B82	M15-SBS
Frequency	88367	78544	74486	67849	66646	64754	60075	59795	56010	53364

- OriginName: the name of the starting stop. It is nominal. Data mode is character.

- OriginLat: the latitude of the starting stop. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
40.50894	40.66465	40.71784	40.73 377	40.81295	40.91236

- OriginLong: the longitude of the starting stop. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
-74.24806	-73.98235	-73.93124	-73.92552	-73.87883	-73.70187

- DestinationName: the name of the destination. It is nominal data. Data mode is character.

- DestinationLat: the latitude of the destination. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
40.50894	40.66184	40.71639	40.73285	40.81001	40.91238

- DestinationLong: the longitude of the destination. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
-74.24819	-73.98295	-73.93114	-73.92606	-73.87833	-73.70146

- VehicleRef: the 4-digit ID of the bus. It is nominal data. Data mode is character.

- VehicleLocationLatitude: the current latitude of the bus. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
40.50288	40.66326	40.72926	40.73253	40.81046	40.91239

- VehicleLocationLongitude: the current longitude of the bus. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
-74.25234	-73.97374	-73.93229	-73.92582	-73.88008	-73.70149

- NextStopPointName: the next stop the bus will serve. It is nominal data. Data mode is character.

- ArrivalProximityText: text about the arrival proximity. It is nominal data. Data mode is character.

- DistanceFromStop: the distance of the bus (in meters) from that next stop. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
0	30	100	-163.5	202	11224

- ExpectedArrivalTime: the expected arrival time to the next stop. The format is "YYYY-MM-DD HH: MM: SS". It is ordinal data. Data mode is character.

- ScheduledArrivalTime: the scheduled arrival time to the next stop. The format is "HH: MM: SS". It is ordinal data. Data mode is character.

- Delay: delay time equals the expected arrival time minus the scheduled arrival time. The unit is the second. Data type is ratio. (Negative numbers mean the bus arrived before the scheduled arrival time.)

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
-13796	-11	164	319	470	19659

- DistOriCurr: the distance from the starting point to the current location. This value is calculated by using the longitude/latitude of the starting point

and the longitude/latitude of the current location. The unit is the kilometer. Data type is ratio.

Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
0	1.93	4.13	4.695	6.87	21.72

- DistOriDest: the bus route length. This value is calculated by using the longitude/latitude of the starting point and the longitude/latitude of the destination. The unit is the kilometer. Data type is ratio.

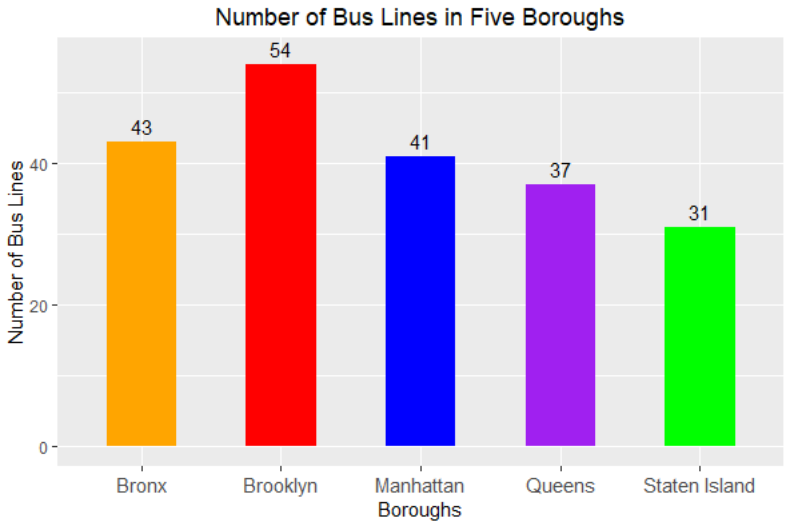
Minimum	1st quantile	Median	Mean	3rd quantile	Maximum
0	6.49	8.15	8.587	10.93	18.12

10 Data visualization

RStudio was used to do data visualization.

Graph1

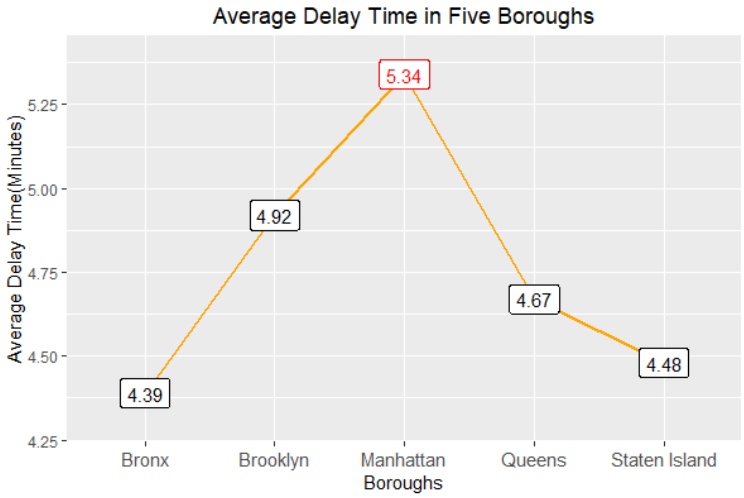
This bar graph shows the number of bus lines in each borough. In Brooklyn, there are 54 bus lines, which is more than other boroughs.



Five Boroughs in New York City

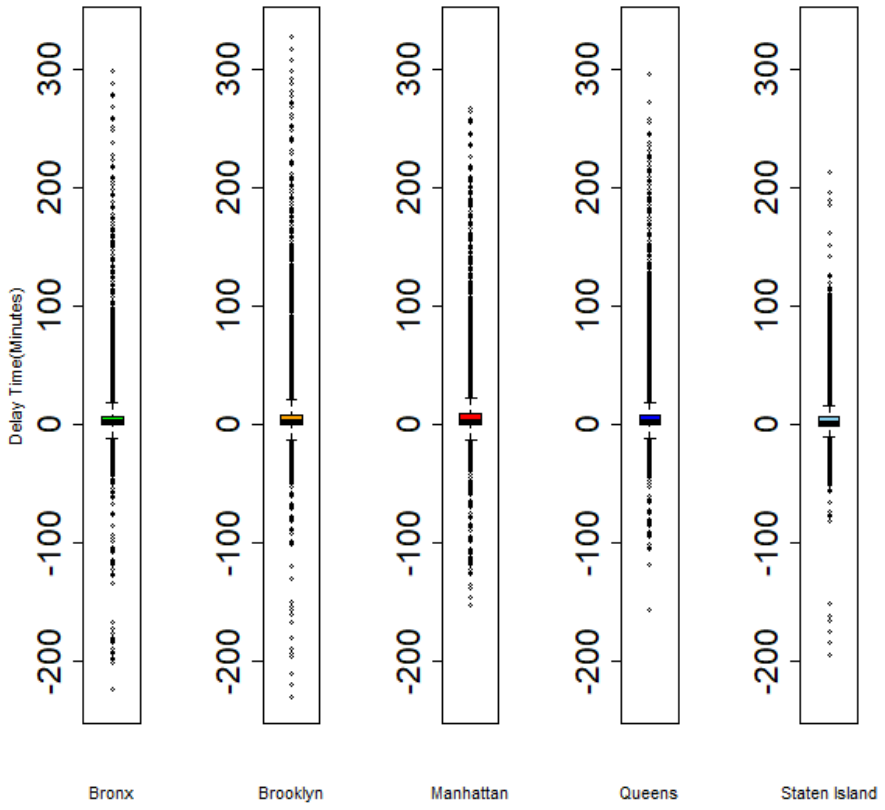
Graph2

This plot shows the average delay time in each borough. The average delay time in Manhattan is 5.34 minutes, which is longer than other boroughs.



Graph3

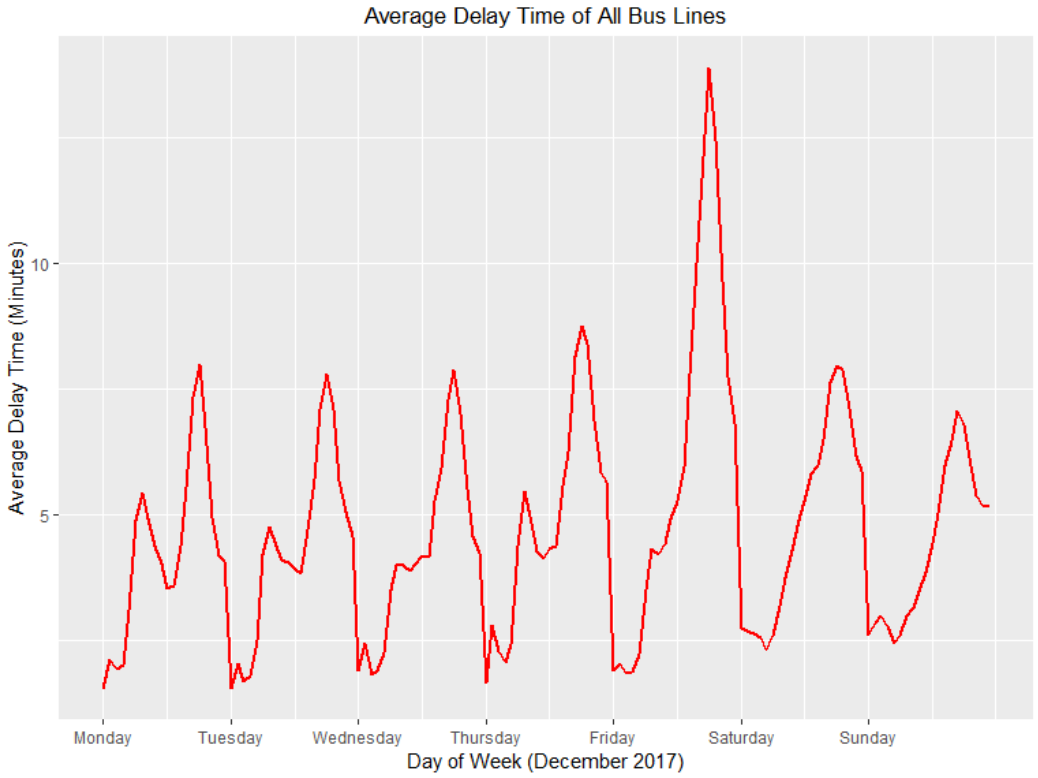
This boxplot shows the delay time in 5 boroughs through quantiles. Although the average delay time showed in graph 2 is between 4.39 to 5.34 minutes, the range of delay time is quite wide in each borough, especially in Brooklyn.

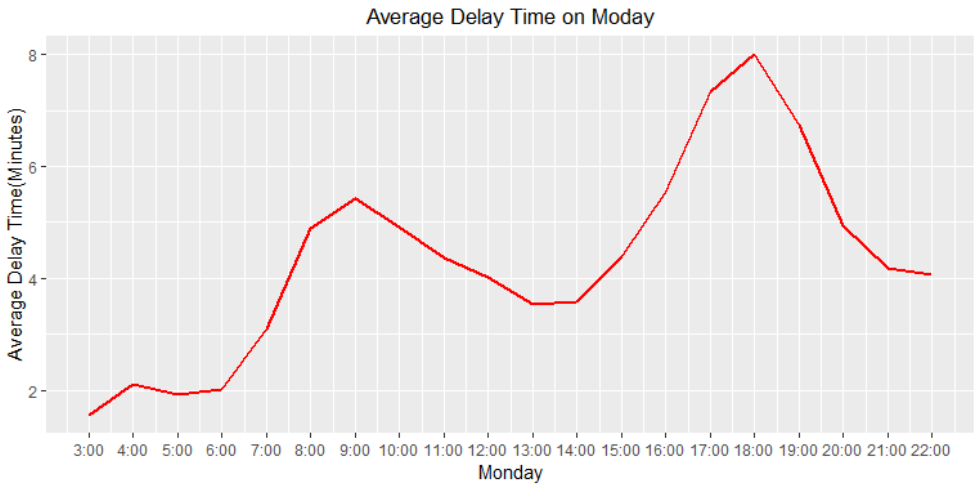


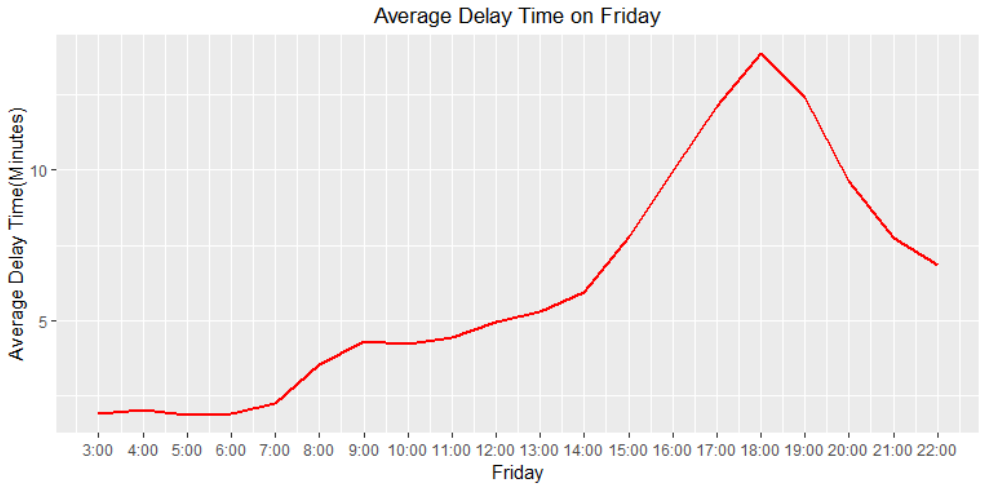
Graph4

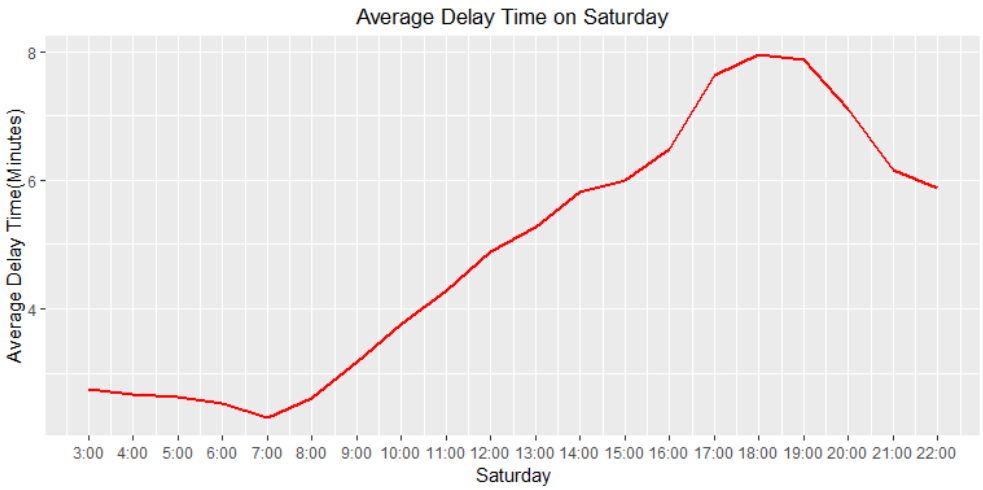
- The first line chart shows the average delay time in a daily cycle during Dec. 2017. We can see the same up-and-down pattern through Monday to Friday. There is a small peak in the morning (between 9:00 am to 10:00 am) and a big peak in the evening (between 18:00 pm to 19:00 pm), but the peak in Friday's evening is much higher than other days. For this reason, I just plotted the second and third line chart presenting the average delay time on Monday and Friday.
- Similarly, there is the same pattern on Saturday and Sunday. A big peak exists in the evening. But on Saturday, the big peak appeared between

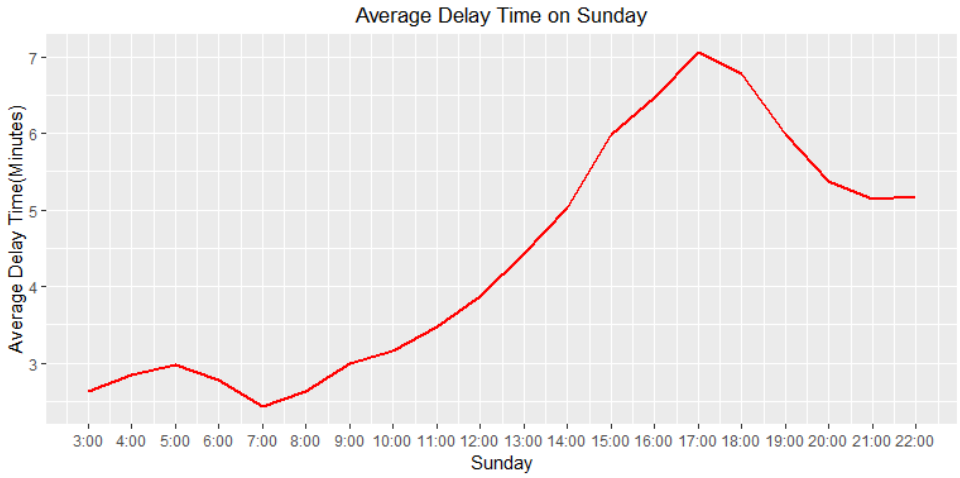
18:00 pm to 19:00 pm. On Sunday, the big peak appeared between 17:00 pm to 18:00 pm.







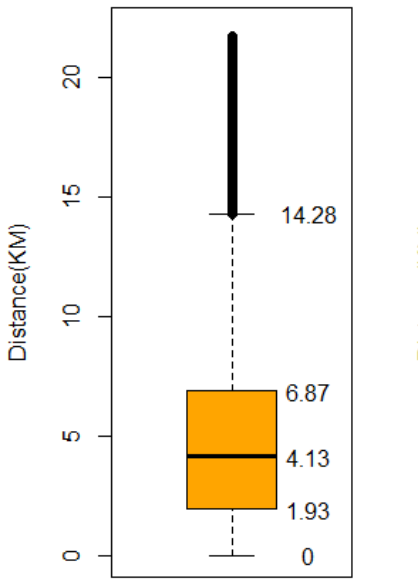




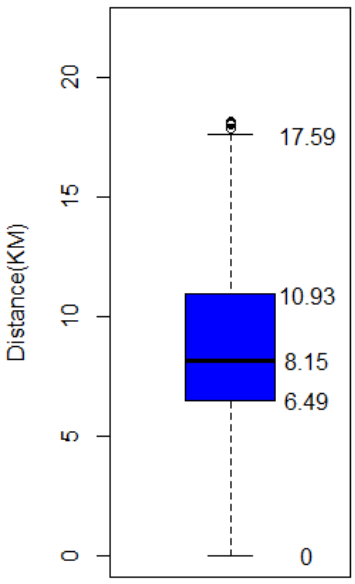
Graph5

- The left boxplot shows the distance from the starting point to the current location through quantiles.

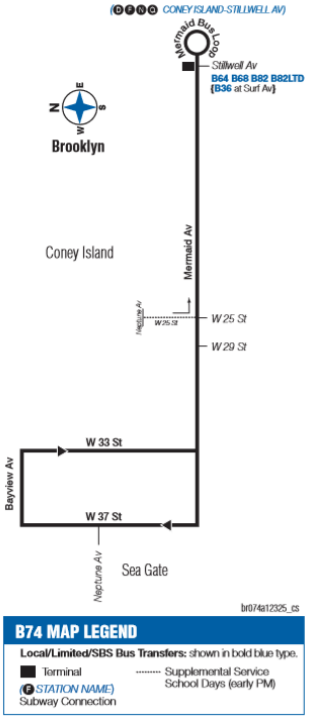
- The middle boxplot shows the bus route length through quantiles.
- The right graph explains why the minimum bus route length is 0. This is because the starting point and the destination for B74 are very close.



The Distance from the Starting Point to the Current Location



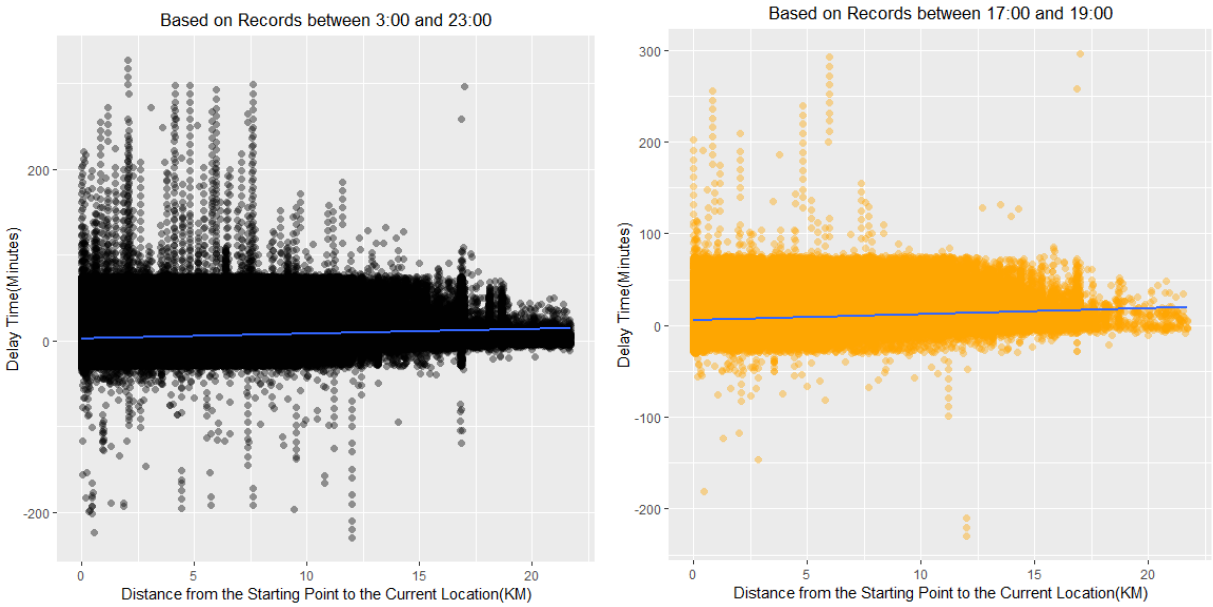
The Bus Route Length



(MTA, n.d.)

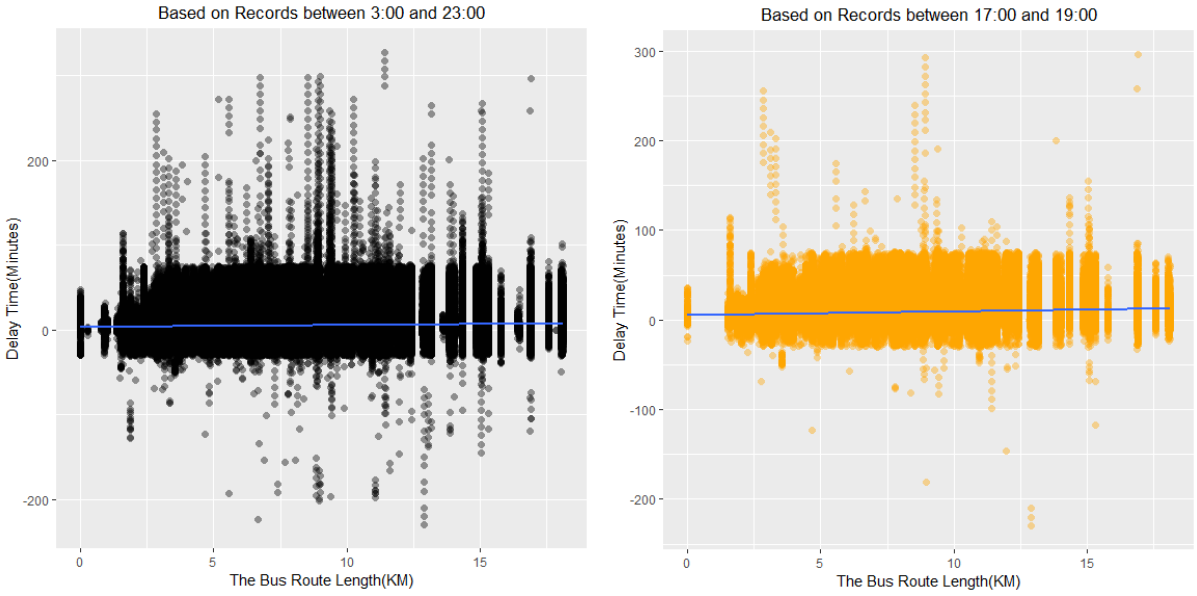
Graph6

- These two scatterplots shows the relationship between the “Delay time” and “The distance from the starting point to the current location”. The only difference is the left one is based on the data recorded between 3:00 am and 23:00 pm, the right one is based on the data recorded between 17:00 pm and 19:00 pm. According to graph 4, 17:00 pm to 19:00 pm is when the longest average delay time appeared.
- After fitting a smooth line (the blue line) with “lm” metod in each plot, I found the line has a small slope, which suggests that the “Delay time” and “The distance from the starting point to the current location” have a weak relationship.



Graph7

- These two scatterplots shows the relationship between the “Delay time” and “The bus route length”. The only difference is the left one is based on the data recorded between 3:00 am and 23:00 pm, the right one is based on the data recorded between 17:00 pm and 19:00 pm. According to graph 4, 17:00 pm to 19:00 pm is when the longest average delay time appeared.
- After fitting a smooth line(the blue line) with lm method in each plot, I found the line is quite flat, which suggests that the “Delay time” and “The bus route length” may not related.



11 Correlation Analysis

I set “Delay Time” as the dependent variable, “The distance from the starting point to the current location” and “The bus route length” as the independent variables. I got the following correlation coefficients between the dependent and independent variables.

Based on the correlation coefficients below, there is a very weak relationship between the “Delay Time” and “The distance from the starting point to the current location”, and there is no relationship between the “Delay Time” and “The bus route length”.

Dependent ~ Independent Variable	Correlation Coefficient
Delay Time ~ The distance from the starting point to the current location	0.21
Delay Time ~ The bus route length	0.07

12 Regression Analysis

I set “Delay Time” as the dependent variable, “The distance from the starting point to the current location” and “The bus route length” as the independent variables. Then I fitted them to a linear regression model; I got the following results:

```
Console ~/
> MTA_Dec1712_New.lmfit <- lm(Delay ~ DistOriCurr + DistOriDest, data = MTA_Dec1712_New)
> summary(MTA_Dec1712_New.lmfit)

Call:
lm(formula = Delay ~ DistOriCurr + DistOriDest, data = MTA_Dec1712_New)

Residuals:
     Min       1Q   Median       3Q      Max
-14355.1   -314.5   -127.0    143.5  19446.1

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 192.12836    0.75181  255.56   <2e-16 ***
DistOriCurr  35.35731    0.09175  385.37   <2e-16 ***
DistOriDest  -4.50708    0.09437  -47.76   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 583.3 on 4581064 degrees of freedom
Multiple R-squared:  0.03722,   Adjusted R-squared:  0.03722
F-statistic: 8.856e+04 on 2 and 4581064 DF,  p-value: < 2.2e-16
```

The beta coefficients for distance from the starting point to the current location and "The bus route length" are 35.36 and -4.5. The p-values of these independent variables are close to zero (2.2e-16). The significant p-values suggest that these two beta coefficients are real.

The adjusted R-squared is 0.037. This low value shows that only 3.7% of the variance of delay time has been explained by the independent variables. It cannot be used to predict the "Delay Time" based on "the distance from the starting point to the current location" and "the bus route length". This model is useless.

13 Null hypothesis testing

This section shows the null hypothesis on Pearson's r:

Step1: define null and alternative hypothesis

- H0: ρ= 0, "Delay Time" and "The distance from the starting point to the current location" are not related.
- Ha: ρ ≠ 0, "Delay Time" and "The distance from the starting point to the current location" are related.

Step2: define significance alpha = 0.05.

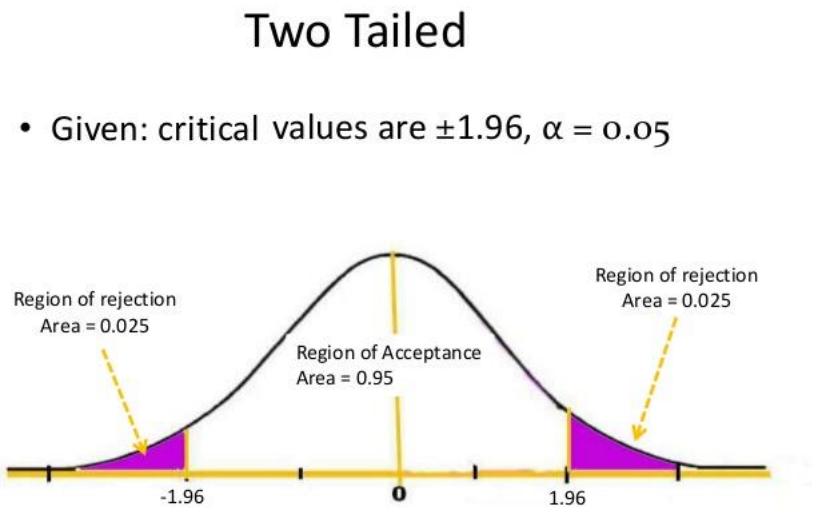
Step3: calculate test statistic.

- According to formula of T-test: $t = r\sqrt{\frac{n-2}{1-r^2}}$, the test statistic is 460.73.

  n: the number of observations, which is 4581067.
  r: correlation coefficient, which is 0.21 according to chapter 11, correlation analysis.

Step4: find the critical value

- According to the T-distribution table, the 2-tailed critical value region is from -1.96 to +1.96. (alpha = 0.05, the degrees of freedom = n-2 = 4581065). If the test statistics is out of this region, we reject the null hypothesis, otherwise, we fail to reject the null hypothesis.



Step5: conclusion

- Since the test statistic falls out of the critical value region, we reject the null hypothesis. “Delay Time” and “The distance from the starting point to the current location” are related.

14 Results Interpretation

Two important results have been figured out by using RStudio.

- The first one is the average delay time for each borough and each bus line.
- The second one is shown in graph 4: the delay time is highly related to the recording time (local time).

How many bus lines in each borough of New York city?

According to Graph 1, there are 43, 54, 41, 37 and 31 bus lines in Bronx, Brooklyn, Manhattan, Queens and Staten Island, respectively. Brooklyn has 54 bus lines, which is more than other boroughs. State Island has 31 bus lines, which is lesser than other boroughs.

This result gives a big picture about the density of bus lines in each Borough. The bus company can allocate human resources and budget based on the density.

Borough	Number of Bus Lines
Bronx	43
Brooklyn	54
Manhattan	41
Queens	37
Staten Island	31

What is the average delay time in each borough? Which borough has the worst bus service according to the delay time?

- Based on graph 2, the delay time in each borough can be found in the table below.

Borough	Average Delay Time(Minutes)
Bronx	4.39
Brooklyn	4.92
Manhattan	5.34
Queens	4.67
Staten Island	4.48

- The average delay time in Manhattan is 5.34 minutes, which is longer than other boroughs. Therefore, Manhattan provides the worst bus service based on the longest average delay time.

For the bus company, this result can be used to evaluate their performance in each borough. Also, this data tells the bus company they need to pay more attention in the Manhattan area to reduce delay time.

<u>How long will the average delay be for each bus line? Was each bus line running to schedule in Dec, 2017? Which one has the longest delay in each borough?</u>

- According to graph 3, I found the range of delay time is quite wide in each borough. I further calculated the average delay time for each bus line as below.
- **On average, no bus was running to schedule in these 5 boroughs.**
- In the Bronx, Bx15 had the longest delay time of 7.99 minutes. In Brooklyn, B35 had the longest delay time of 9.83 minutes. In Manhattan, M60-SBS had the longest delay time of 11.40 minutes. In Queens, Q56 had the longest delay time of 10.05 minutes. In Staten Island, S86 had the longest delay time 12.75 minutes.

The following table shows the average delay time for each bus line in a descending order. For bus company, they can easily find out which bus lines has longer delay time than others, and which bus lines should be improved first to be more punctual. For customers who plan to take a certain bus line, the result can give them a hint how long they may need to wait based on the scheduled arrival time.

Bronx	Average Delay Time (Minutes)	Brooklyn	Average Delay Time (Minutes)	Manhattan	Average Delay Time (Minutes)	Queens	Average Delay Time (Minutes)	Staten Island	Average Delay Time (Minute)
Bx15	7.99	B35	9.83	M60-SBS	11.40	Q56	10.05	S86	12.75
Bx39	7.29	B41	9.27	M5	10.95	Q32	9.93	S84	11.24
Bx32	7.22	B25	8.51	M1	10.29	Q59	8.74	S81	7.50
Bx20	7.18	B24	7.65	M7	10.20	Q54	8.18	S92	6.88
Bx21	6.93	B12	7.58	M2	9.17	Q58	7.56	S96	6.66
Bx41 -SBS	6.40	B60	7.04	M3	8.35	Q88	7.00	S66	5.32
Bx12 -SBS	6.09	B38	6.70	M4	8.09	Q24	6.76	S76	5.22

Bx18	5.85	B45	6.62	M9	7.86	Q44-SBS	6.33	S94	4.84
Bx30	5.78	B82	6.57	M102	7.55	Q5	5.93	S98	4.68
Bx17	5.75	B57	6.29	M101	7.09	Q17	5.88	S46	4.56
Bx41	5.49	B43	6.19	M103	7.06	Q31	5.87	S59	4.38
Bx36	5.34	B46	6.18	M55	6.72	Q55	5.83	S74	4.27
Bx4A	5.21	B7	6.02	M50	6.64	Q30	5.68	S54	4.21
Bx35	5.05	B47	5.91	M15	5.78	Q13	5.63	S93	4.16
Bx11	4.86	B44	5.67	M57	5.71	Q76	4.98	S62	4.11
Bx1	4.81	B11	5.64	M42	5.67	Q85	4.72	S52	4.10
Bx28	4.46	B15	5.64	M12	5.66	Q43	4.70	S48	4.09
Bx10	4.30	B37	5.58	M31	5.52	Q28	4.25	S57	4.09
Bx4	4.30	B20	5.51	M20	5.50	Q77	4.17	S40	4.05
Bx2	4.26	B16	5.47	M15-SBS	5.41	Q27	3.92	S44	3.98
Bx5	4.24	B1	5.43	M34A-SBS	5.37	Q48	3.87	S79-SBS	3.94
Bx3	4.06	B62	5.36	M11	5.18	Q16	3.80	S90	3.87
Bx9	4.04	B4	5.22	M22	5.11	Q4	3.62	S53	3.59
Bx26	3.88	B44-SBS	5.16	M104	4.95	Q3	3.52	S78	3.11
Bx16	3.85	B6	5.07	M8	4.28	Q36	3.41	S51	3.00
Bx6-SBS	3.72	B13	5.00	M66	3.98	Q20A	3.36	S91	2.65
Bx38	3.70	B84	4.87	M106	3.94	Q1	3.21	S61	2.26
Bx19	3.64	B83	4.86	M14A	3.81	Q2	3.16	S89	1.79
Bx33	3.61	B54	4.68	M23-SBS	3.63	Q46	2.74	S42	1.77
Bx13	3.59	B69	4.62	M35	3.63	Q83	2.66	S56	1.60
Bx6	3.55	B48	4.45	M86-SBS	3.30	Q12	2.50	S55	0.20
Bx27	3.50	B52	4.40	M116	3.23	Q20B	2.46		
Bx22	3.46	B8	4.39	M14D	3.02	Q84	2.18		
Bx7	3.43	B14	4.31	M72	2.44	Q42	2.12		
Bx34	3.27	B17	4.20	M100	2.42	Q15	1.46		
Bx42	3.16	B49	4.13	M96	2.10	Q15A	1.38		
Bx40	2.82	B67	4.12	M79-SBS	1.97	Q26	1.35		
Bx31	2.69	B26	4.08	M21	1.94				
Bx8	2.60	B46-SBS	4.00	M98	1.46				
Bx12	2.53	B70	3.91	M10	1.33				
Bx24	2.48	B61	3.84	M34-SBS	1.29				
Bx29	2.34	B9	3.83						

Bx46	0.22	B65	3.80						
		B3	3.69						
		B68	3.60						
		B32	3.14						
		B63	3.06						
		B74	3.05						
		B64	2.80						
		B36	2.69						
		B2	2.00						
		B31	1.92						
		B42	1.71						
		B39	0.21						

How many bus lines use the Select Bus Service(SBS) route? Do these bus lines are more punctual than bus lines that do not use SBS route on average?

- There are 14 bus lines use the Select Bus Service(SBS) route.
- No. Following table shows the average and median delay time for bus lines that use or do not use the SBS route. Both the average and median delay time indicate bus lines use the SBS route have longer delay than bus lines that do not use the SBS route.

According to the definition of SBS, "SBS improves speed and reliability through dedicated bus lanes, off-board fare payment, station spacing and transit signal priority" (MTA, n.d.). However, the result shows SBS did not help to improve the speed. The bus company should pay attention to these 14 bus lines to improve the punctuality. The results also can be used to remind customers. Even though they take the bus lines that use SBS route, the risk of delay is high.

	Average Delay Time(Minutes)	Median Delay Time(Minutes)
Bus lines use the SBS route	4.86	4.58
Bus lines DO NOT use the SBS route	4.78	4.3

What are the possible factors related to the average delay time?

- Based on graph 4, we can see the delay time is highly related to the recording time (local time).
    - From Monday to Friday, the delay time had a small peak in the morning between 9:00 am to 10:00 am and a big peak in the evening between 18:00 pm to 19:00 pm.
    - On Saturday, the delay time had only one big peak which appeared between 18:00 pm to 19:00 pm.
    - On Sunday, the delay time had only one big peak which appeared between 17:00 pm to 18:00 pm.
- Based on graph 6 and the correlation analysis in chapter 11, we can see the delay time has a weak relationship with "The distance from the Starting point to the current location".

The delay time is highly correlated to the recording time (local time). This gives a clue to the bus company that they need to focus on the peak hours to optimize their bus line service system in order to shorten the delay time. Especially in Friday around 18:00 pm to 19:00 pm, the average delay time is about 13.75 minutes, which is much longer than any other time.

The result also gives customers a hint that they may need to go the bus stops much earlier during the peak hours to keep their schedule.

15 Technical Terms

Bus lines that use the "**Select Bus Service**" route will have "-SBS" in their published names, For example, Q44-SBS.

"**Select Bus Service(SBS)** provides a complementary service to the subway system by connecting neighborhoods to subway stations and major destinations. The goal of SBS is to bring faster, more reliable and quality bus service to high ridership corridors. SBS improves speed and reliability through dedicated bus lanes, off-board fare payment, station spacing and transit signal priority (TSP)." (MTA, n.d.)

References

Dowle, M., & Srinivasan, A. (2018). *data.table: Extension of `data.frame`*, 1.11.8. Retrieved November 23, 2018, from https://CRAN.R-project.org/package=data.table

Hijmans, R. J. (2017). *geosphere: Spherical Trigonometry*, 1.5-7. Retrieved November 17, 2018, from https://CRAN.R-project.org/package=geosphere

James, D., & Hornik, K. (2018). *chron: Chronological Objects which Can Handle Dates and Times*, 2.3-53. Retrieved November 17, 2018, from https://CRAN.R-project.org/package=chron

Kaggle. (n.d.). *Kaggle*. Retrieved November 17, 2018, from Kaggle is the place to do data science projects: https://www.kaggle.com/

MTA. (n.d.). *About Us*. Retrieved November 17, 2018, from MTA: http://web.mta.info/mta/network.htm#buscostats

MTA. (n.d.). *Brooklyn Bus Schedules* . Retrieved December 12, 2018, from MTA: http://web.mta.info/nyct/bus/schedule/bkln/b074cur.pdf

MTA. (n.d.). *MTA Bus Time*. Retrieved November 17, 2018, from http://www.bustime.mta.info/wiki/Developers/ArchiveData

MTA. (n.d.). *New York City Transit SBS* . Retrieved December 14, 2018, from MTA: http://web.mta.info/mta/planning/sbs/aboutUs.htm

RStudio Team. (2016). *RStudio: Integrated Development Environment for R*, 1.1.456. (RStudio, Inc.) Retrieved November 17, 2018, from http://www.rstudio.com/

Stone, M. (2017). *New York City Bus Data*. Retrieved November 14, 2018, from Kaggle: https://www.kaggle.com/stoney71/new-york-city-transport-statistics

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*, 1.2.1. Retrieved November 17, 2018, from https://CRAN.R-project.org/package=tidyverse