Citi Bike NYC Trip Segmentation and Station Network Analysis

Jessica M. Rudd, Shashank Hebbar, and Soujanya Mandalapu, Kennesaw State University

INTRODUCTION

Citi Bike is a privately-owned bike sharing system that operates in New York City and Jersey City, New Jersey. The system launched in May 2013 with 332 stations and 6,000 bikes. Today, Citi Bike is the largest bike share program in the United States with 10,000 bikes utilizing 600 stations in Manhattan, Brooklyn, Queens, and Jersey City (https://www.citibikenyc.com/). Users can purchase short-term passes with unlimited rides up to 30 minutes for $12 per day, or an annual pass at $163/year with unlimited rides up to 45-minutes. The annual subscriber option for the entire year costs less than two monthly subway passes, so it is a cheaper, and often faster, option than public transportation and taxis for people living and/or working in New York City. With nearly 40,000 average trips daily, ensuring that bikes and stations are operational and readily available requires a fleet of box trucks redistributing over 3,000 bikes per day between stations, and engaging in daily maintenance requirements. The objective of this project is to segment types of riding days and provide recommendations for optimal bike redistribution and maintenance.

DATA

The team obtained publicly available datasets from the Citi Bike system data webpage (https://www.citibikenyc.com/system-data). There are two types of files available: trip histories, and daily ridership and membership data. Since the objective of this project was to segment trips and riding days and make recommendations based on the network of trips, we decided to use the trip histories dataset. Trip history provided information for every Citi Bike trip since the beginning of the program through September 2017, with datasets provided for each month. Each month of data contained over 1 million trips, and 15 variables: BIKEID, END_STATION_ID, END_STATION_LATITUDE, END_STATION_LONGITUDE, END_STATION_NAME, ENDTIME, START_STATION_ID, START_STATION_LATITUDE, START_STATION_LONGITUDE, START_STATION_NAME, STARTTIME, TRIPDURATION, BIRTH_YEAR, GENDER, USERTYPE. The public data was preprocessed in advance to remove trips taken by staff for service and inspection, trips taken between “test” stations, and trips below 60 seconds in length (false starts).

Considering the size of the datasets and various expansions of the system since its inception, we opted to analyze the 2016 trip data since it represented the most recent, complete year of data, and the size of the system in 2016 is comparable to its current operation. Additional analysis variables were created to better understand the timing and seasonality of trips, and comparison between yearly subscribers and short-term pass customers. This included partitioning the start and end time variables into days of the week, month, hour of day, weekday vs weekend, and whether rush hour or not. With approximately 1 million records per month, our computing resources required us to take a 10% sample of the 2016 data. Distributions of rides by day of the week, month, gender, and user type were compared between the full dataset and the 10% sample to ensure the sample was representative of 2016 Citi Bike trips (Table 2, Table 3).

To provide additional insight for segmenting trips and riding days, a publicly available dataset of weather in the NYC area was merged with the trip data. Weather data was obtained from the national climatic data center (https://www.ncdc.noaa.gov/) and included variables such as temperature, snow, precipitation, and wind speed. These observations were merged by latitude and longitude into the Citi bike trip history dataset.

PROBLEM

The project goal was to segment Citi Bike trips using daily weather information and trip characteristics, and provide business recommendations based on segmented clusters for potential improvements to bike redistribution and bike and station maintenance. After segmenting the trips, a network analysis can be performed within each segment to provide insight into which stations are most and least heavily utilized on distinct types of trip days. The thought process is that, if Citi Bike management knows, for example, tomorrow is going to be a warm sunny day, then they can predict which stations will encounter the most traffic, and potentially require additional redistribution of bikes. We predict that individual riding days will segment according to various weather patterns and system management can be planned accordingly.

DATA CLEANING, VALIDATION, AND EXPLORATION

In preparation for data analysis, the following actions were taken:

1. The team chose to use all available Citi Bike Trip History data from 2016, with reasons explained in Data section.
2. 12 months of Citi Bike trip history for 2016 was imported using a SAS ® macro and merged.
3. Analysis variables created using DATEPART, TIMEPART, WEEKDAY, and MONTH functions: START_DATE, END_DATE, START_TIME, END_TIME, WEEKDAY, MONTH, HOD (hour of day of start of trip), WORKDAY (1 = M-F, 0 = Weekend), RUSH (1 = AM rush hour 7am-10am, 2 = PM rush hour 4pm-7pm, 0 = all other hours). The rush hour variable was created based on increased subscriber rides (presumably people who live/work in NYC) during workdays (Figure 3, Figure 4).
4. Since each month had nearly 1 million records, we took a 10% sample of the complete 2016 data due to limitations in our computing resources.
5. Distributions of WEEKDAY, MONTH, GENDER, and USERTYPE compared between full dataset and 10% sample to ensure sample representative of the full dataset (Table 2, Table 3).
6. Publicly available weather data merged with Citi Bike sample dataset, including SNOW, TMAX(max daily temperature), TMIN (min daily temperature), PRCP (precipitation in inches), and AWND (average wind speed).
7. Exploratory analysis completed on merged weather attributes:
   a) Values of skewness and kurtosis for TMIN and TMAX indicate approximately normal distribution with few outliers (Figure 5, Figure 6). As such, these variables are appropriate for inclusion in a segmentation model.
   b) PRCP is highly skewed but is effectively binary (Figure 7), and can be binned into no precipitation (PRCP = 0) and any precipitation (PRCP > 0). We predict that trips on rainy days will segment differently than trips on clear days, for example.
   c) AWND, and SNOW are sparse, highly skewed, and effectively unary, indicating they are not appropriate to include in analysis (Figure 8, Figure 9).
8. Exploratory Analysis completed on trip-related characteristics:
   a) Visualization of stations and number of trips at each station using SAS® Visual Statistics GeoMap procedure (Figure 10).
   b) Yearly subscribers represent 89% of all trips (Figure 11). In this case we assume that subscribers are population living and/or working in NYC area, while short-term customers are likely to be NYC tourists.
   c) Subscribers take more trips during weekdays, while customers take more trips on weekends (Figure 3). During all of 2016, number of trips for subscribers' spikes during each work week and dramatically decreases during weekends and holidays (Figure 12). In contrast, trips for customers spike during weekends and holidays and decreases during weekdays (Figure 13). This is logical assuming subscribers use Citi Bike for commuting and customers use Citi Bike for tourist activity.
   d) Both subscribers and customers ride more in warmer months (Figure 14).
   e) Most popular start (Figure 15, Figure 16) and end (Figure 17, Figure 18) stations over all trips identified for subscribers and customers, respectively. Most popular start and end stations for subscribers are in business districts. Most popular start and end stations for customers are near Central Park and tourist attractions.

f) Assessment of number of rides per day and maximum temperature (Figure 19). Lower temperatures coincide with decreases in Citi Bike activity.

g) Based on exploratory data analysis, variables retained for segmentation model (Table 1: List of Variables Considered for Clustering):

Variable	Variable Description
AWND	Average Wind Speed
HoD	Hour of Day
PRCP	Precipitation
SNOW	Snow (Indicator)
Tmax	Maximum Temperature
Tmin	Minimum Temperature
age	Age of rider
gender	Gender of Rider
month	Month of year
no_of_trips	No of trips per day
rush	nominal variable for rush hour of the day
trip_duration	Trip duration of each ride
weekday	Week day of the week
workday	Binary indicator for weekday/weekend

Table 1: List of Variables Considered for Clustering

h) Final analysis dataset exported into SAS® Enterprise Miner ™ for clustering analysis.

DATA CLEANING, VALIDATION, AND EXPLORATION

Considering the results of exploratory analysis, we decided to complete the cluster analysis on trips completed by subscribers only. Subscribers complete 89% of all Citi Bike trips, are the most consistent daily users of the system, and trips completed by subscribers include additional descriptive information that can augment the ride segmentation analysis, including age and gender for example. To divide the observations into groups with certain distinctive characteristics, K means clustering was implemented. Here the value of K=4 yielded the most balanced clusters.

Figure 1: Cubic Clustering Criterion vs Number of Clusters

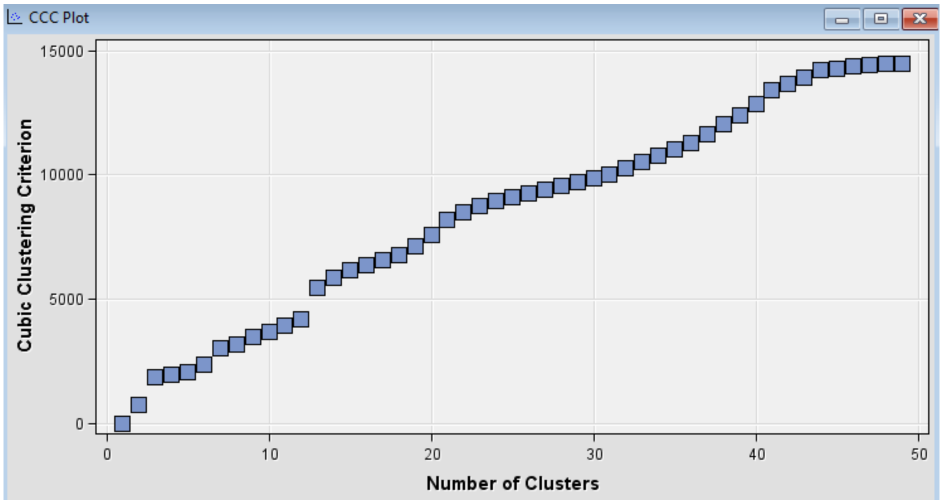


Figure 1 above shows the variation of the cubic clustering criterion(CCC) vs the number of clusters. The CCC is a statistic that measures the difference in error by applying the clustering algorithm to the training data set and a reference distribution, which is usually a hyper cube. Here, the CCC increases to 4 and then flattens out before trending upward again. In the interest of parsimony, 4 is the best number of clusters since it is much more balanced and has a high CCC.   Cluster size and statistics are shown in the appendix (Figure 20, Figure 21). The SAS® Enterprise Miner Diagram is shown in Figure 22. To consider the characteristics of the clusters, a segment profile node was added and the results of which are shown in Figure 2: Profile of Individual Clusters.

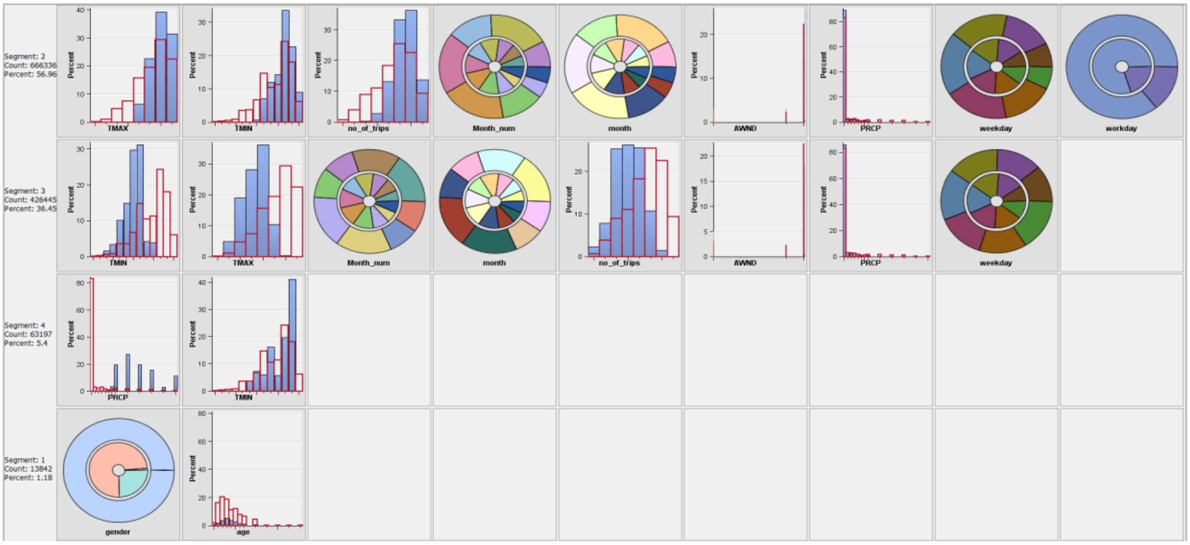


Figure 2: Profile of Individual Clusters

As shown above, segment 1 is mainly characterized by age group, (i.e. younger people), hence these subscribers travel around local college campuses like the new school and NYU. The other three segments are centered around business districts. Segment 2 is characterized by elevated temperatures with little or no precipitation, i.e. summer months. Segment 3 contains those observations which are low temperature and slightly windy days, i.e. winter months. Segment 4 consists of the spring and fall months with high precipitation and moderate temperatures.

Each segment was converted to a graph data structure G=(V,E) where the stations are the vertices(V) and each trip between stations is an edge (E), where the number of trips per day between stations is the weight on that edge. SAS® OPTGRAPH procedure was run independently on these 4 segments to obtain the centrality of the stations. Centrality is defined as the total number of edges incident on the vertices and is an indicator of the importance of a station. The centrality measures are shown on maps of each segment (Figure 23, Figure 24, Figure 25, Figure 26).

GENERALIZATION

We discovered 4 distinct clusters of Citi Bike trips from the cluster analysis:

1. Trips completed by younger population, close to college campuses
2. Trips on warmer days with no precipitation
3. Trips on colder, windy days
4. Trips on median temperature days with high precipitation

Using SAS® PROC OPTGRAPH, we completed a network analysis of all trips within each discovered segment. Degree centrality was used as a proxy measurement for amount of Citi Bike user traffic at each station. In this case, degree centrality measures an important station if there are many trips coming or going from the station to many other stations. From this analysis we have identified the busiest stations within each ride segment (Figure 23, Figure 24, Figure 25, Figure 26).

BUSINESS RECOMMENDATIONS BASED ON SEGMENTATION AND NETWORK ANALYSIS

- **Segment 1**: This segment represents a younger population with popular stations centered around several university centers. For example, the busiest station within this segment is near Union Square, an area surrounded by several buildings and dormitories of The New School and New York University. When the universities are in session we recommend augmenting busy stations in university areas with temporary mobile stations to increase the availability of bikes. In addition, we recommend maintenance of bikes and stations in these areas during university holidays when there will be the least disruption to the system.
- **Segment 2**: This segment represents rides taken on warmer days (summer) with no precipitation. The busiest stations are those near commuting and business districts such as Grand Central Terminal and Midtown and lower Manhattan. This segment experiences the busiest trip activity in general with highest degree centrality of its' stations. During nicer weather months we recommend maintenance of bikes and these stations during weekends and off-peak hours when subscribers/commuters are not utilizing the system as heavily.
- **Segment 3**: This segment represents colder windy days (winter). Similar stations as segment 2 (near commuting and business centers) experience the most activity in comparison to other stations, however their overall activity is less than rides within in segment 2 due to decreased usage of CitiBike system in general during winter months. Winter months, in general, may represent the best opportunity for system maintenance, and less priority for bike redistribution activities.
- **Segment 4**: This segment represents mild temperature days, but with rain. Once again, the commuting and business centers experience the most activity, but overall this segment sees the less activity per station than the other segments. If rainy days do not present a challenge for Citi Bike managers to maintain bikes and stations, or engage bike redistribution, then we recommend any planned system disruptions on these types of days.

SUGGESTIONS FOR FUTURE STUDIES

- Building a predictive model to assess how adding bikes to existing stations or adding new stations will redistribute the network analysis.
- Using a weighted K-Nearest-Neighbor model to predict station pick-up demand based on historical trip data around high-demand events, i.e. St. Patrick's Day parade, Macy's Thanksgiving parade.

CONCLUSION

Citi Bike trips among subscribers to the system segmented into 4 distinct clusters of rides. Based on this information, a network analysis of trip activity within each segment provided insights into which stations experienced the most Citi Bike usage. Knowing this information moving forward, Citi Bike management can plan bike and station maintenance, and bike redistribution, around predicted station activity for each type of trip day or activity center, i.e. stations near universities.

REFERENCES

Christie, Peter, et al. 32Aug2015. *Applied Analytics Using SAS® Enterprise Miner™ Course Notes*. Cary, NC: SAS Institute Inc.

Citi Bike. 2017. "System Data." Accessed November 2017. https://www.citibikenyc.com/system-data.

ACKNOWLEDGMENTS

We'd like to thank Dr. Sherry Ni for her guidance on this project.

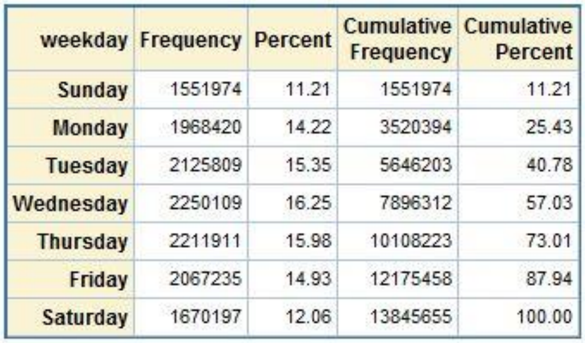
APPENDIX

TABLE OF CONTENTS

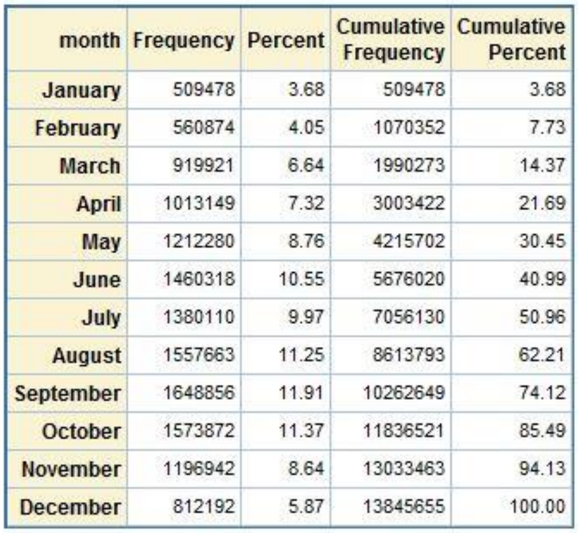
Tables

Figures

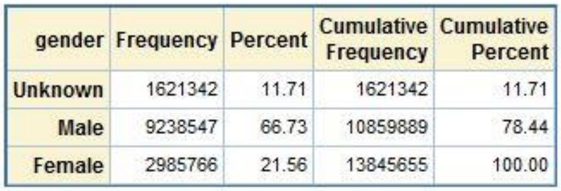
Table 2: Distribution of Descriptive Variables in 2016 Citi Bike Trip History



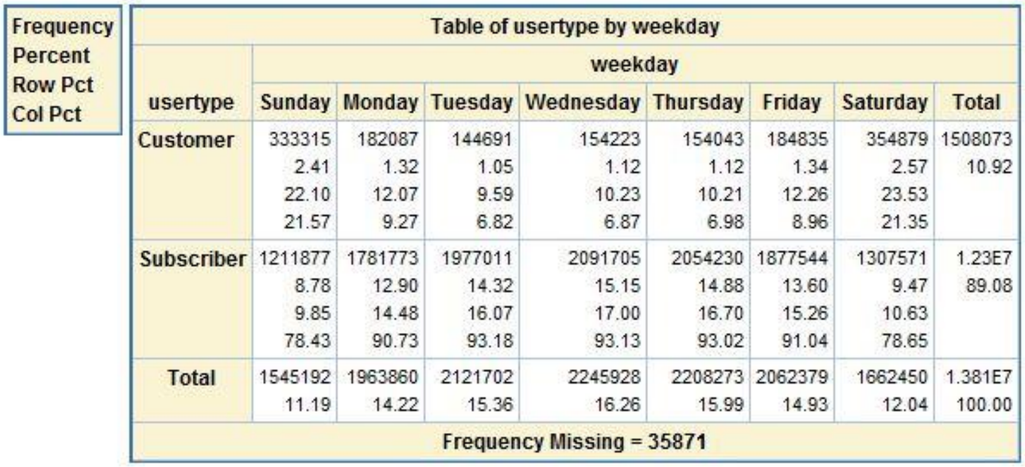
weekday	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Sunday	1551974	11.21	1551974	11.21
Monday	1968420	14.22	3520394	25.43
Tuesday	2125809	15.35	5646203	40.78
Wednesday	2250109	16.25	7896312	57.03
Thursday	2211911	15.98	10108223	73.01
Friday	2067235	14.93	12175458	87.94
Saturday	1670197	12.06	13845655	100.00



month	Frequency	Percent	Cumulative Frequency	Cumulative Percent
January	509478	3.68	509478	3.68
February	560874	4.05	1070352	7.73
March	919921	6.64	1990273	14.37
April	1013149	7.32	3003422	21.69
May	1212280	8.76	4215702	30.45
June	1460318	10.55	5676020	40.99
July	1380110	9.97	7056130	50.96
August	1557663	11.25	8613793	62.21
September	1648856	11.91	10262649	74.12
October	1573872	11.37	11836521	85.49
November	1196942	8.64	13033463	94.13
December	812192	5.87	13845655	100.00



gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Unknown	1621342	11.71	1621342	11.71
Male	9238547	66.73	10859889	78.44
Female	2985766	21.56	13845655	100.00



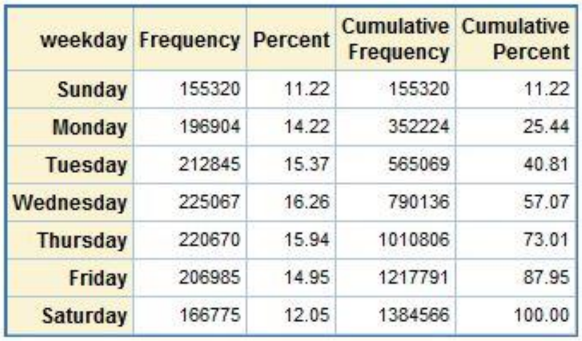
Frequency
Percent
Row Pct
Col Pct

Table of usertype by weekday

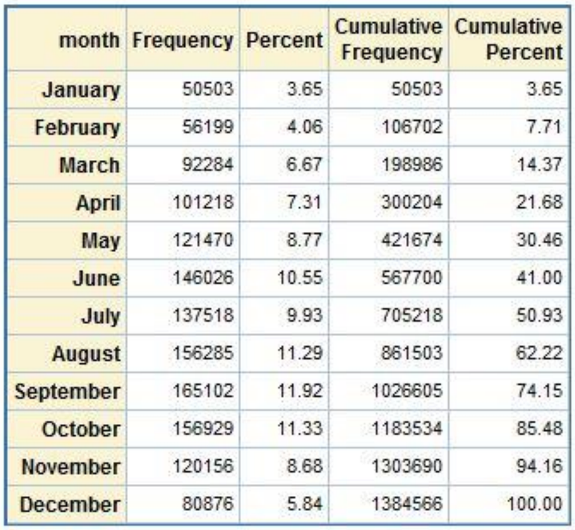
	weekday							
usertype	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Total
Customer	333315 2.41 22.10 21.57	182087 1.32 12.07 9.27	144691 1.05 9.59 6.82	154223 1.12 10.23 6.87	154043 1.12 10.21 6.98	184835 1.34 12.26 8.96	354879 2.57 23.53 21.35	1508073 10.92
Subscriber	1211877 8.78 9.85 78.43	1781773 12.90 14.48 90.73	1977011 14.32 16.07 93.18	2091705 15.15 17.00 93.13	2054230 14.88 16.70 93.02	1877544 13.60 15.26 91.04	1307571 9.47 10.63 78.65	1.23E7 89.08
Total	1545192 11.19	1963860 14.22	2121702 15.36	2245928 16.26	2208273 15.99	2062379 14.93	1662450 12.04	1.381E7 100.00

Frequency Missing = 35871

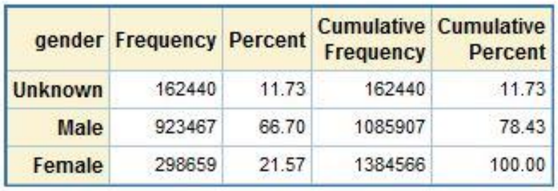
Table 3: Distribution of Descriptive Variables in 10% Sample of 2016 Citi Bike Trip History



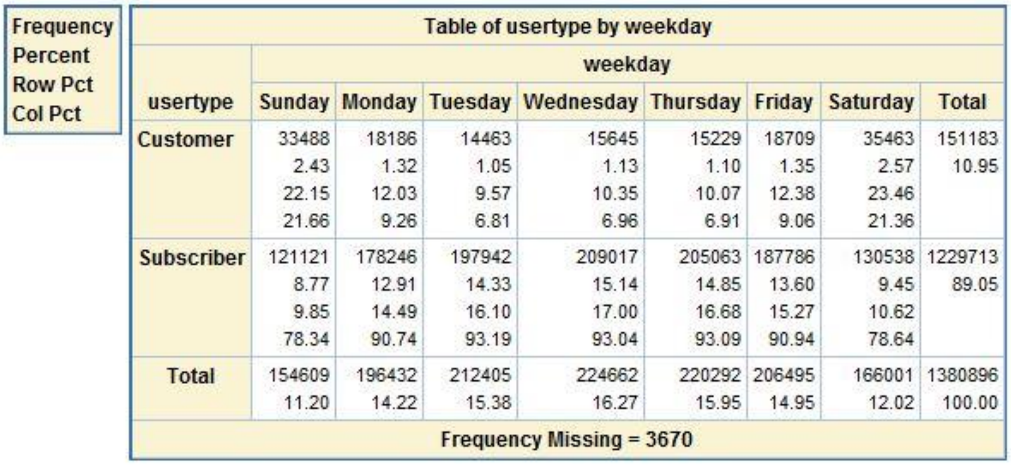
weekday	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Sunday	155320	11.22	155320	11.22
Monday	196904	14.22	352224	25.44
Tuesday	212845	15.37	565069	40.81
Wednesday	225067	16.26	790136	57.07
Thursday	220670	15.94	1010806	73.01
Friday	206985	14.95	1217791	87.95
Saturday	166775	12.05	1384566	100.00



month	Frequency	Percent	Cumulative Frequency	Cumulative Percent
January	50503	3.65	50503	3.65
February	56199	4.06	106702	7.71
March	92284	6.67	198986	14.37
April	101218	7.31	300204	21.68
May	121470	8.77	421674	30.46
June	146026	10.55	567700	41.00
July	137518	9.93	705218	50.93
August	156285	11.29	861503	62.22
September	165102	11.92	1026605	74.15
October	156929	11.33	1183534	85.48
November	120156	8.68	1303690	94.16
December	80876	5.84	1384566	100.00



gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Unknown	162440	11.73	162440	11.73
Male	923467	66.70	1085907	78.43
Female	298659	21.57	1384566	100.00



Frequency
Percent
Row Pct
Col Pct

Table of usertype by weekday

	weekday							
usertype	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Total
Customer	33488 2.43 22.15 21.66	18186 1.32 12.03 9.26	14463 1.05 9.57 6.81	15645 1.13 10.35 6.96	15229 1.10 10.07 6.91	18709 1.35 12.38 9.06	35463 2.57 23.46 21.36	151183 10.95
Subscriber	121121 8.77 9.85 78.34	178246 12.91 14.49 90.74	197942 14.33 16.10 93.19	209017 15.14 17.00 93.04	205063 14.85 16.68 93.09	187786 13.60 15.27 90.94	130538 9.45 10.62 78.64	1229713 89.05
Total	154609 11.20	196432 14.22	212405 15.38	224662 16.27	220292 15.95	206495 14.95	166001 12.02	1380896 100.00

Frequency Missing = 3670

Figure 3: Citi Bike Usage by Day of Week: Customers and Subscribers

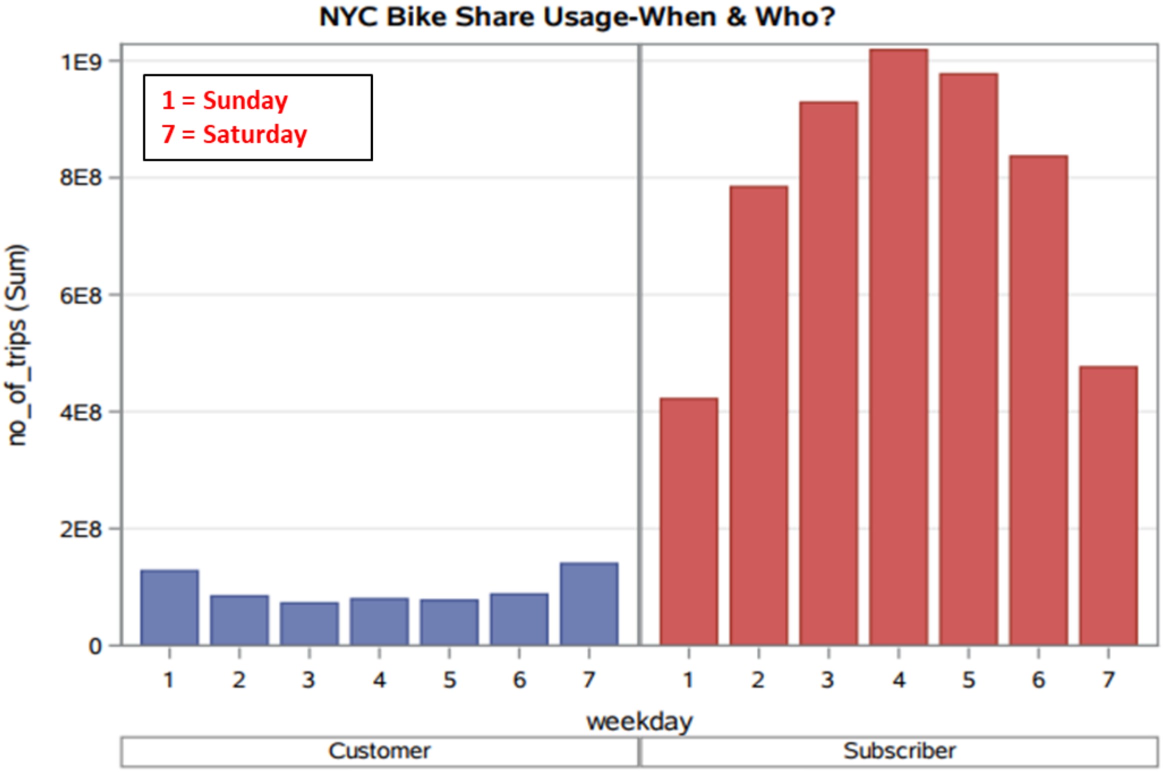


Figure 4: Citi Bike Usage by Hour of Day: Customers and Subscribers

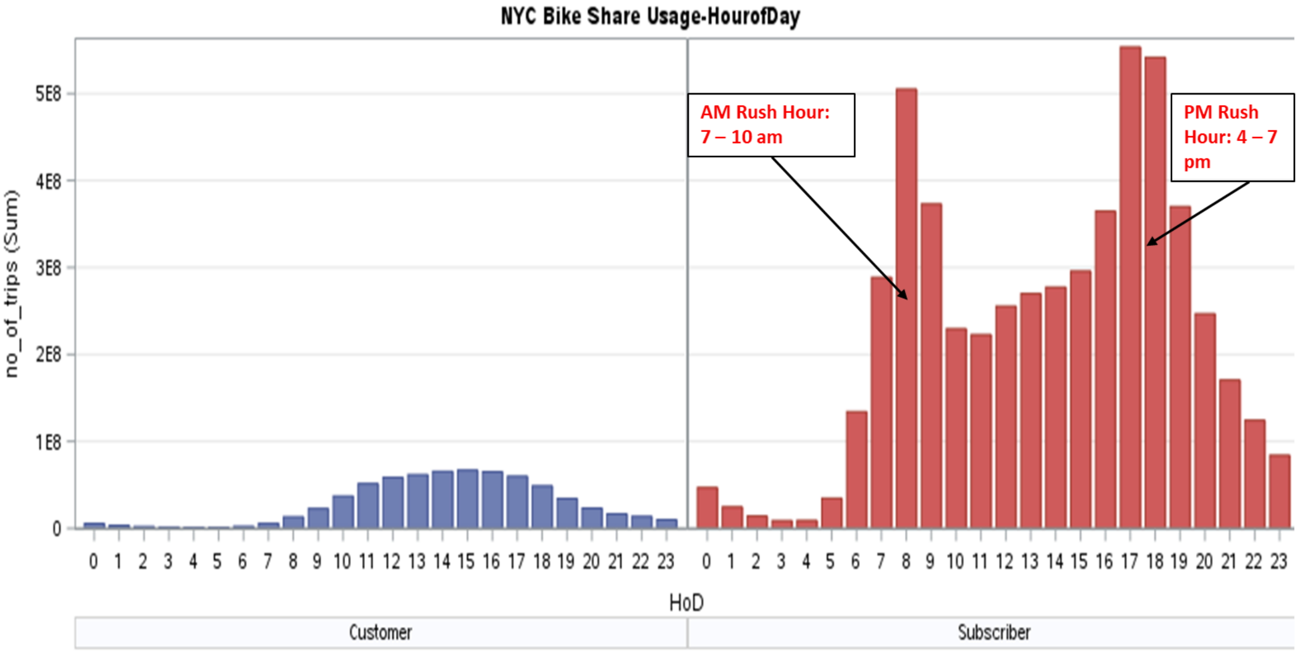


Figure 5: Distribution of Minimum Temperature (F)

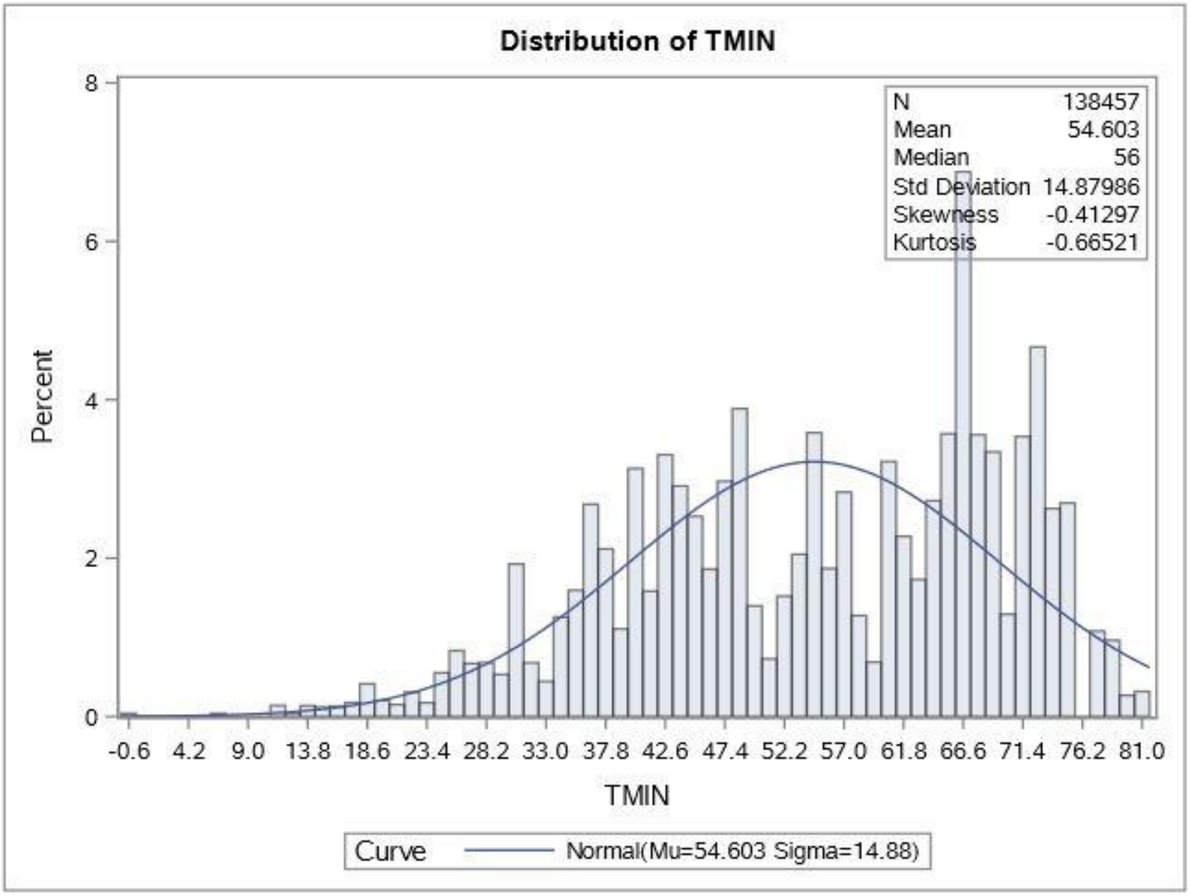


Figure 6: Distribution of Maximum Temperature (F)

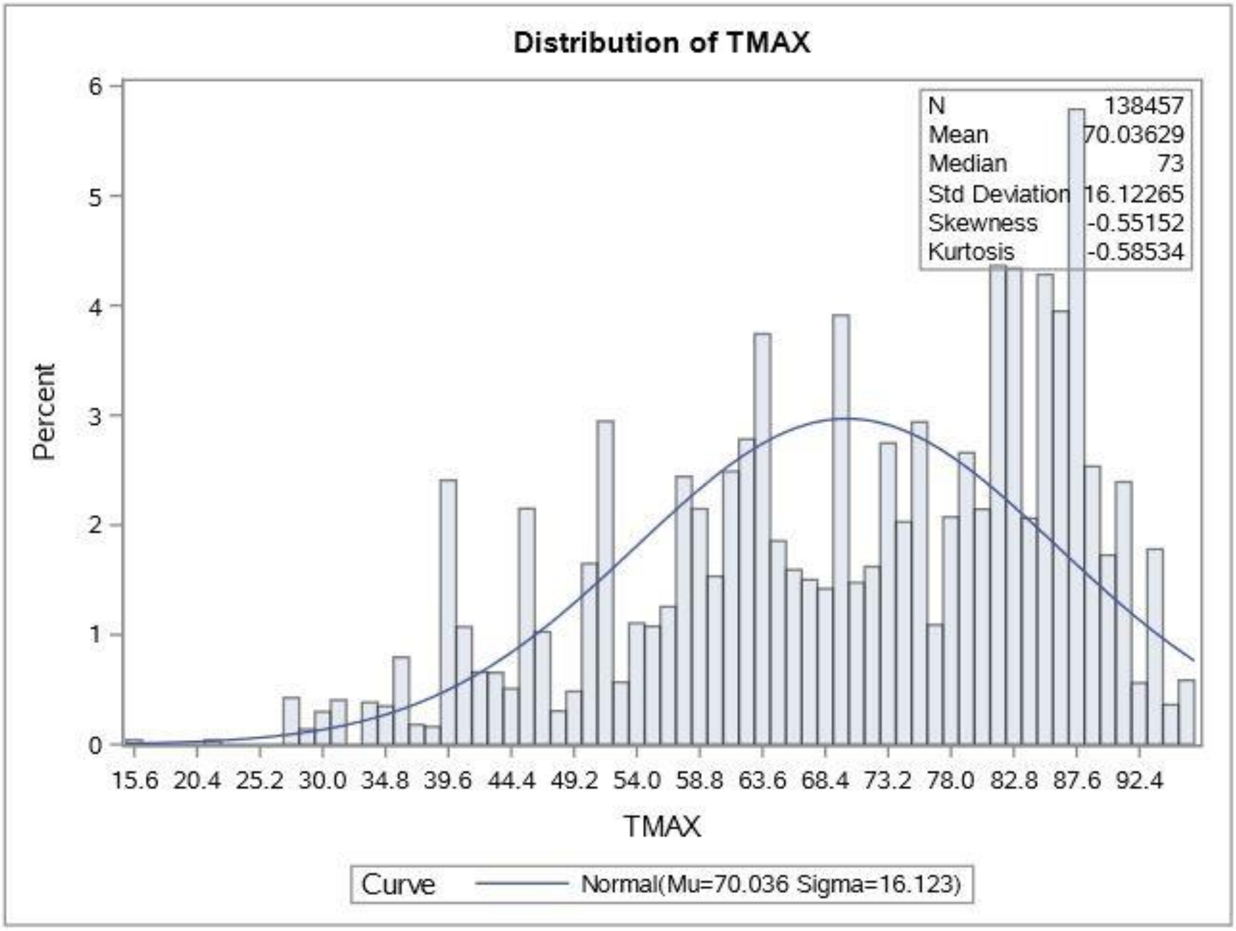


Figure 7: Distribution of Precipitation (in inches)

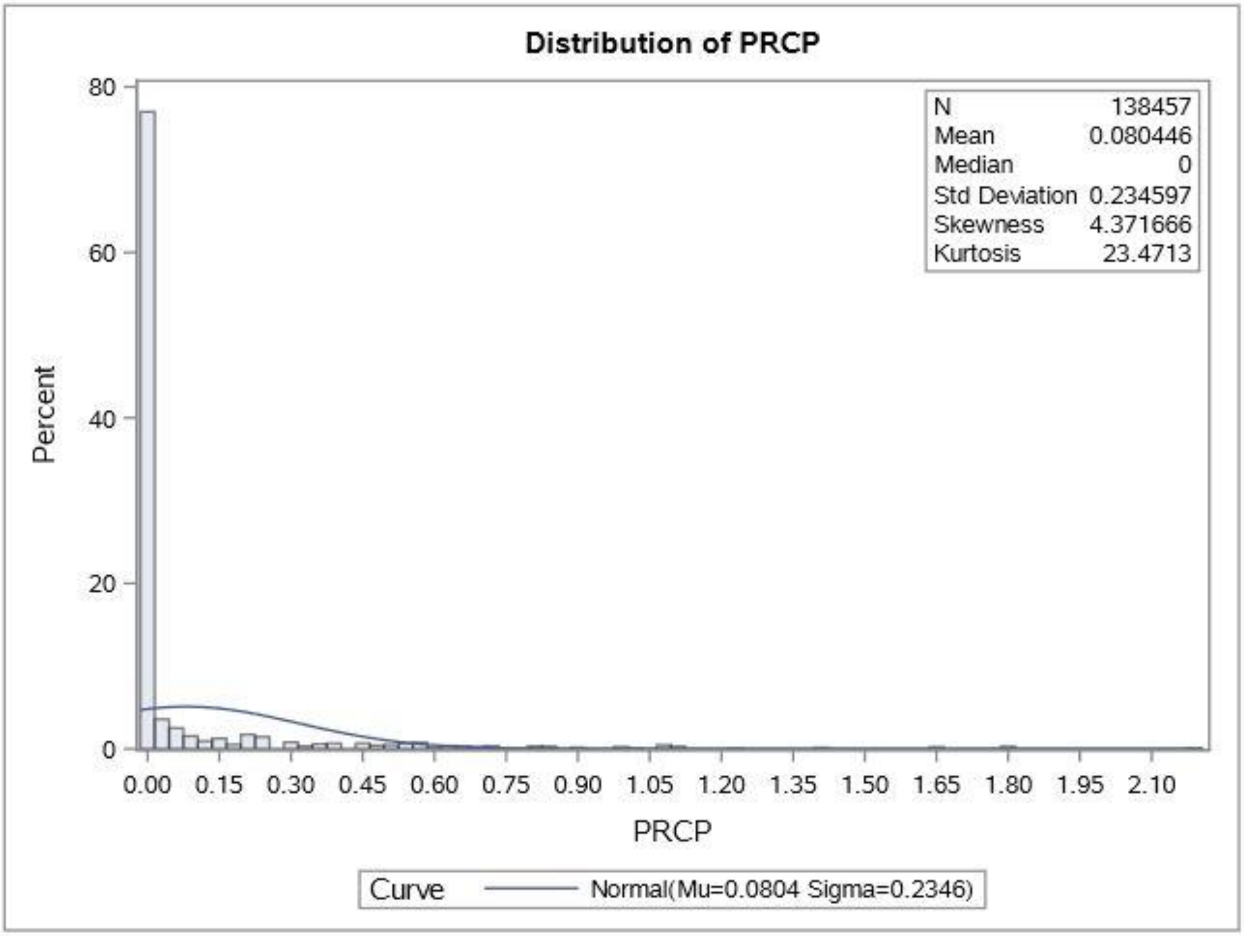


Figure 8: Distribution of AWND (Average Wind Speed miles/hour)

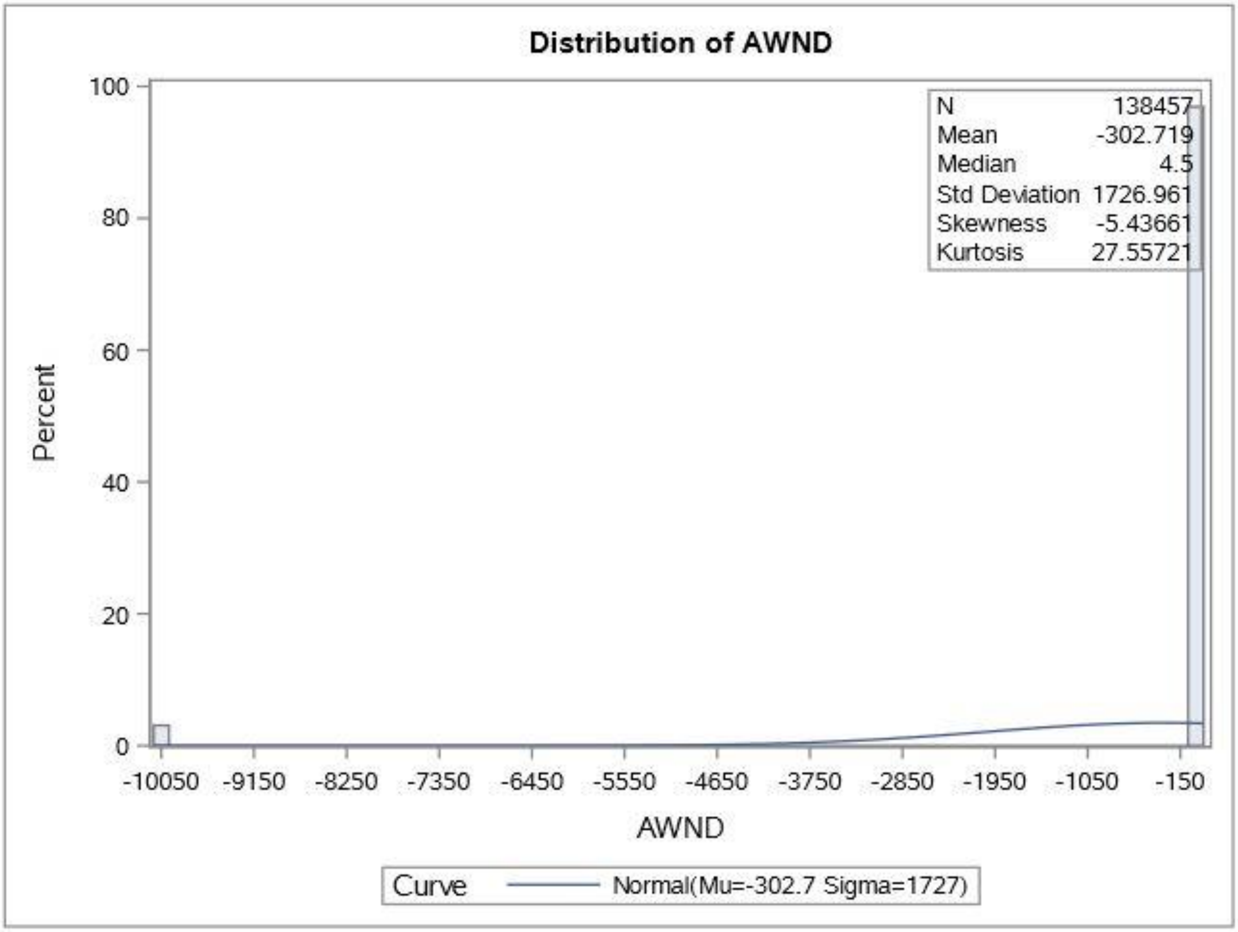


Figure 9: Distribution of Snowfall (in inches)

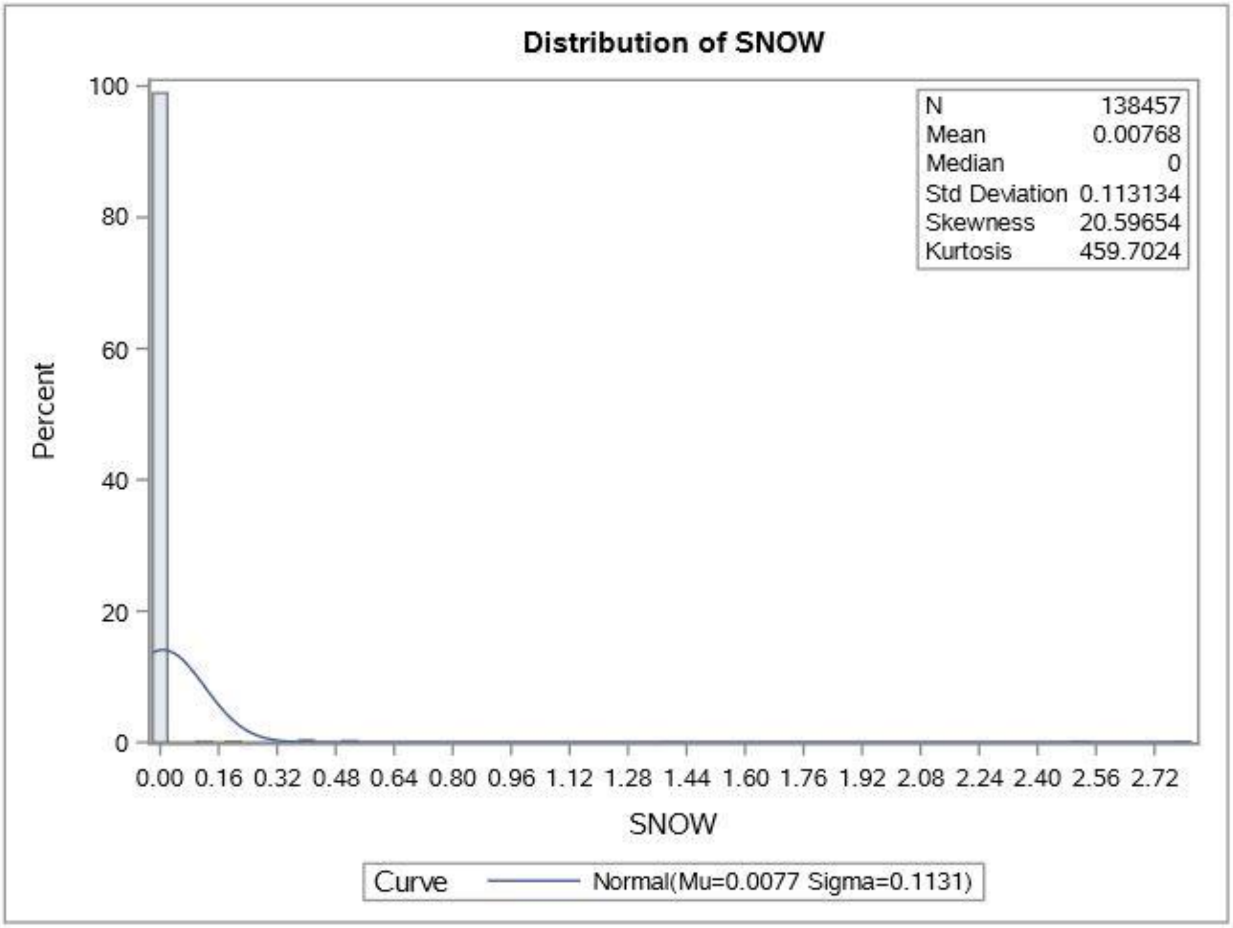


Figure 10: Selection of Station Locations and Relative Number of Trips (bubble size)

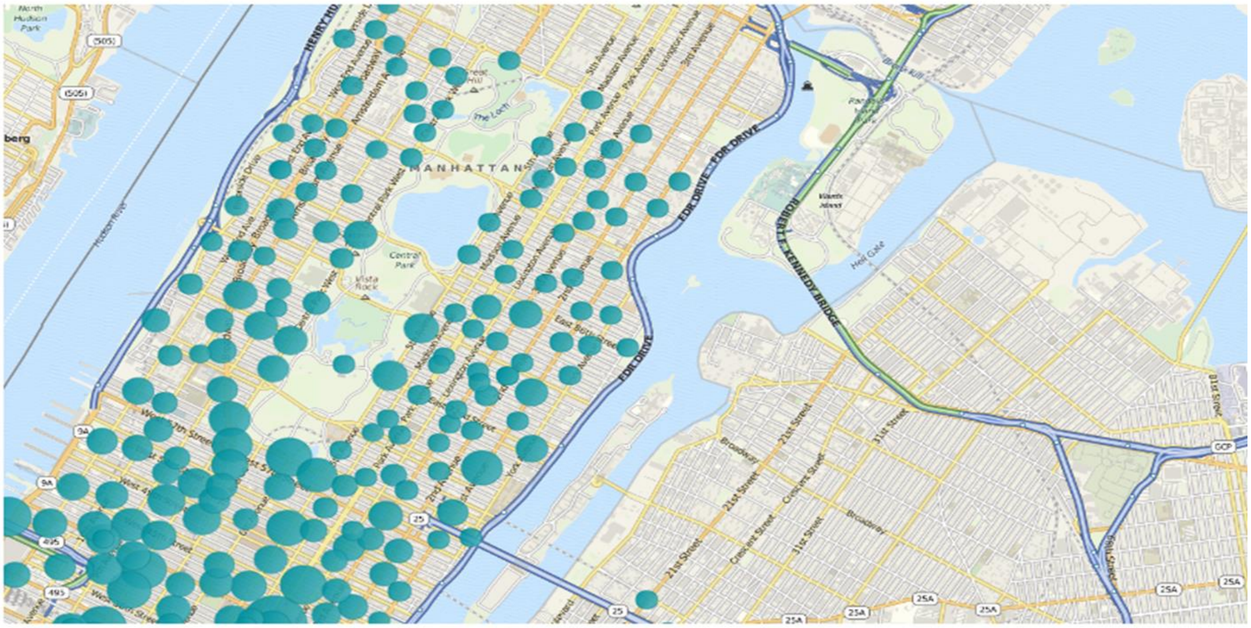


Figure 11: Pie Chart of User Type

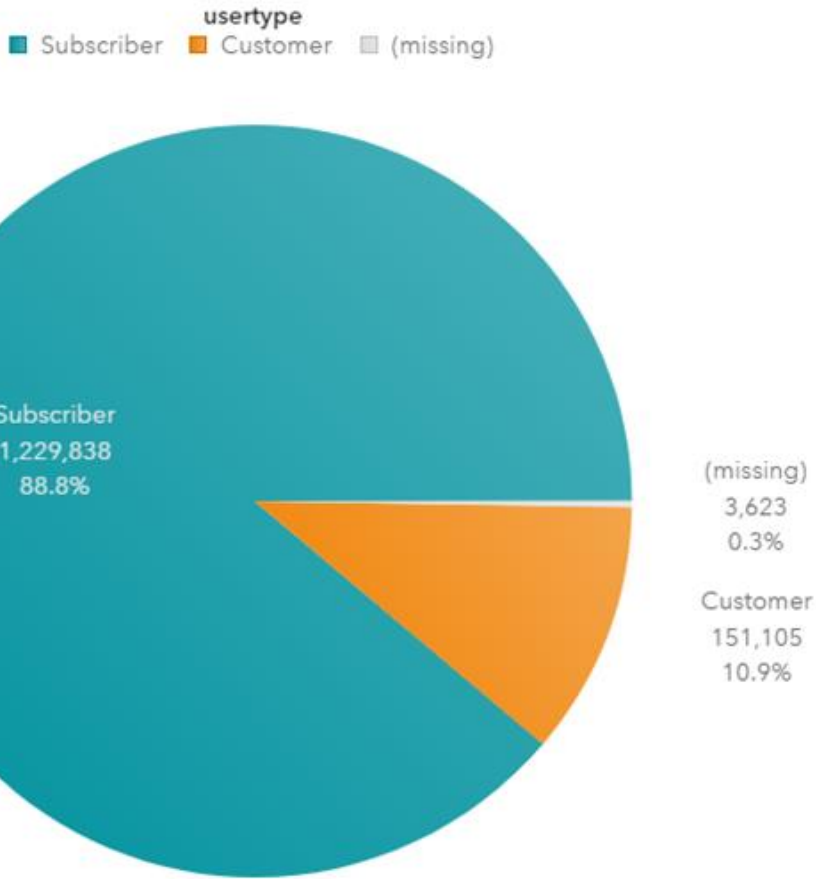


Figure 12: Time Series of Citi Bike Trips, 2016, Subscribers

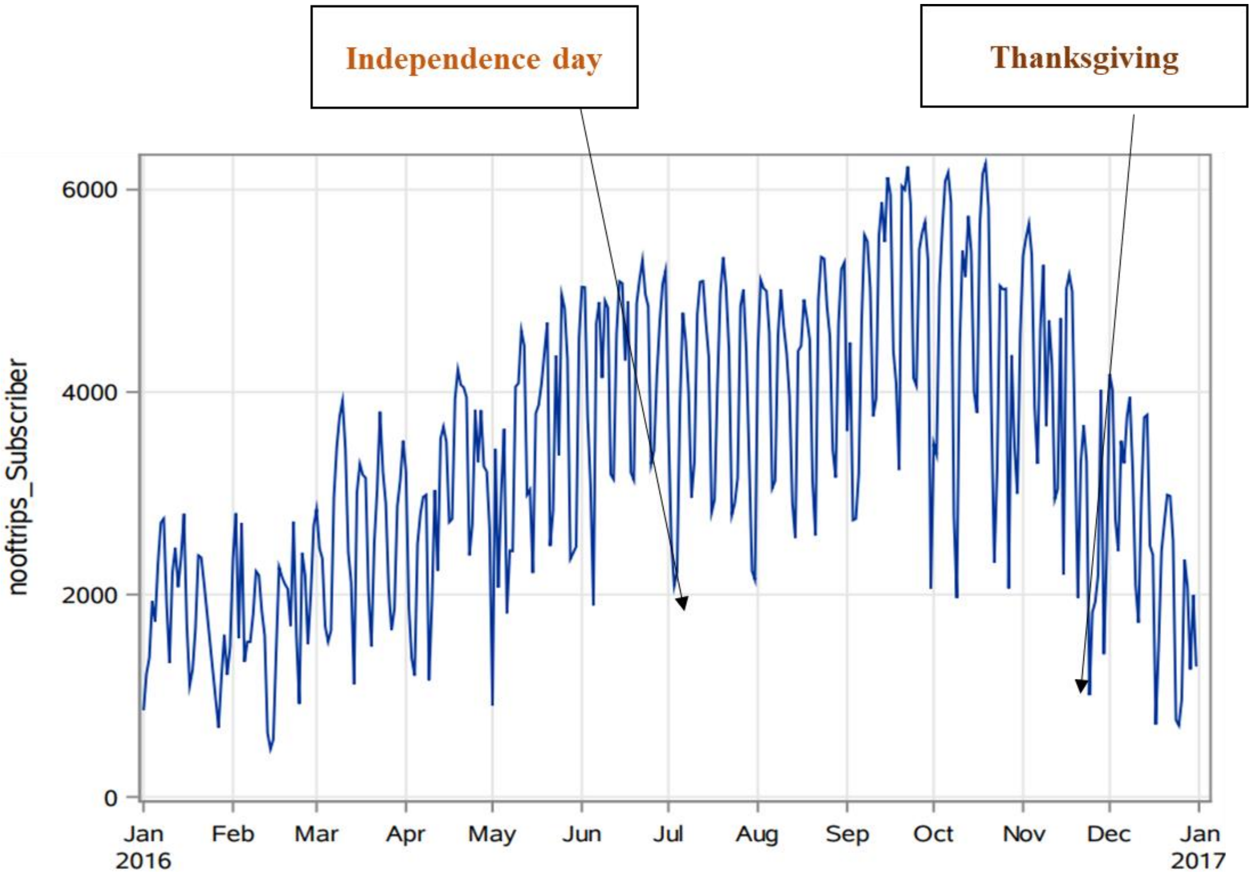


Figure 13: Time Series of Citi Bike Trips, 2016, Customers

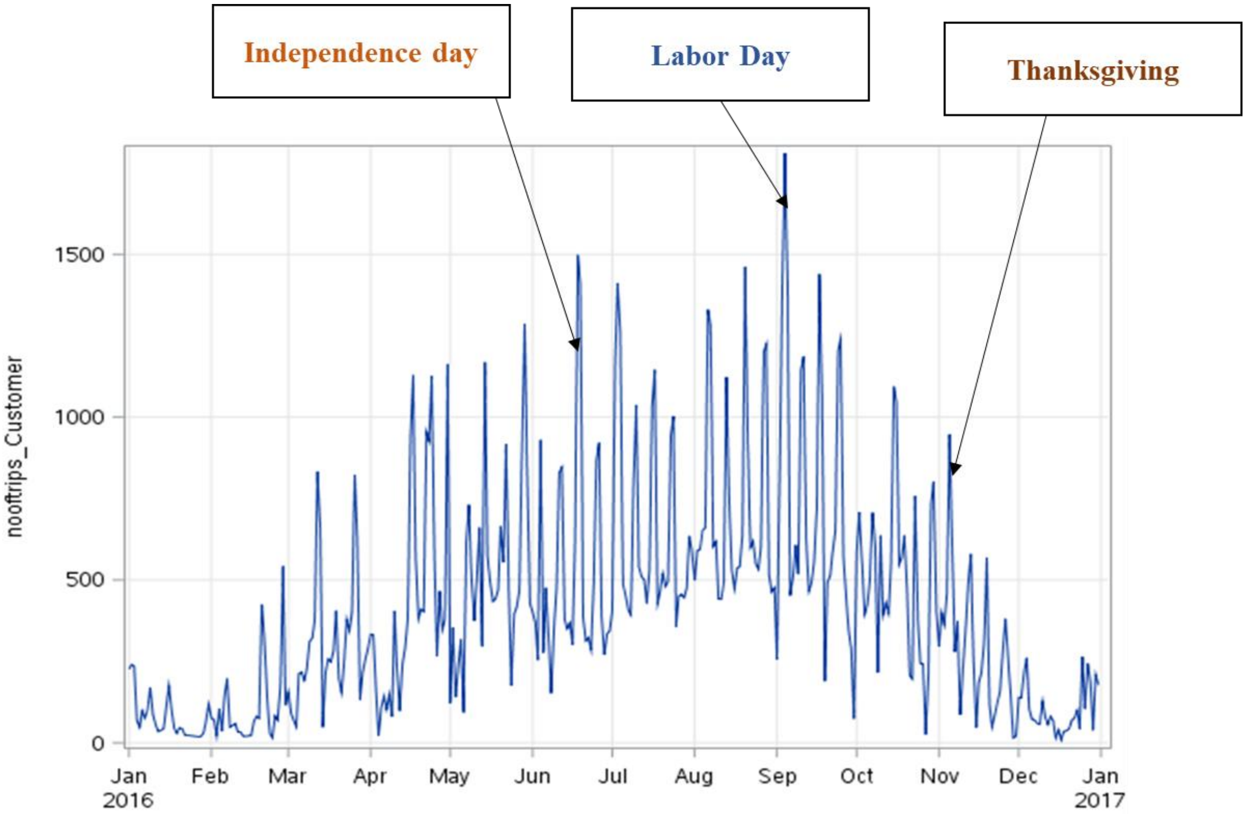


Figure 14: Citi Bike Trips by Month, 2016

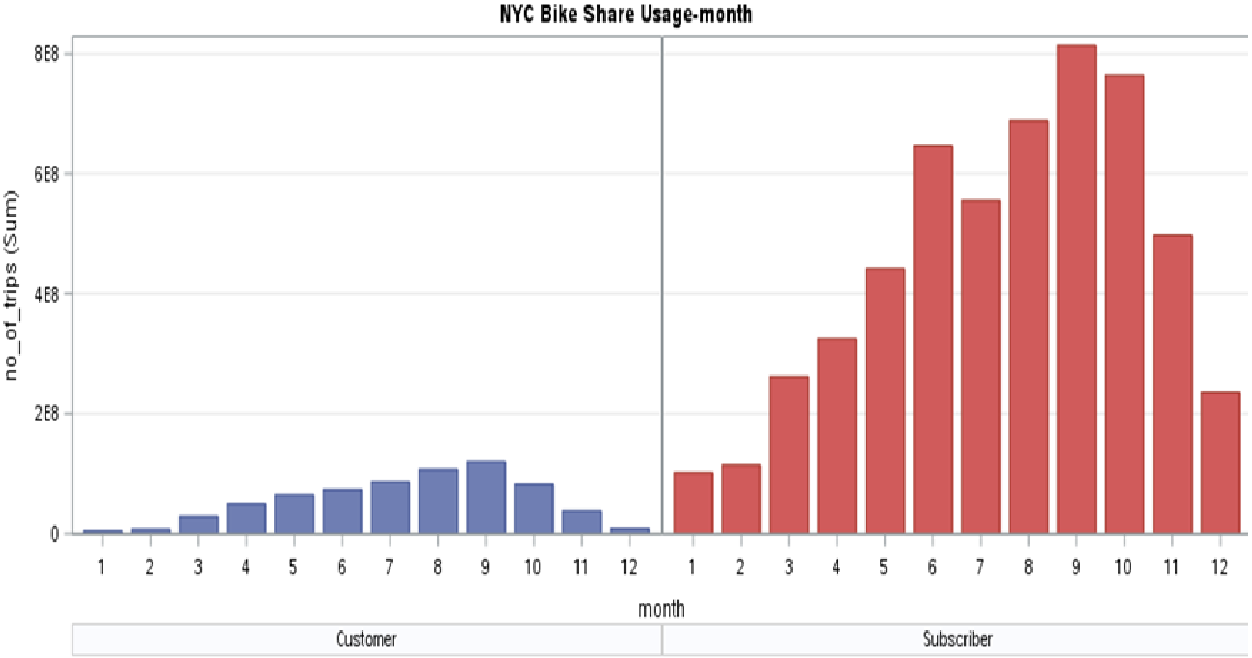


Figure 15: Most Popular Start Stations for Subscribers

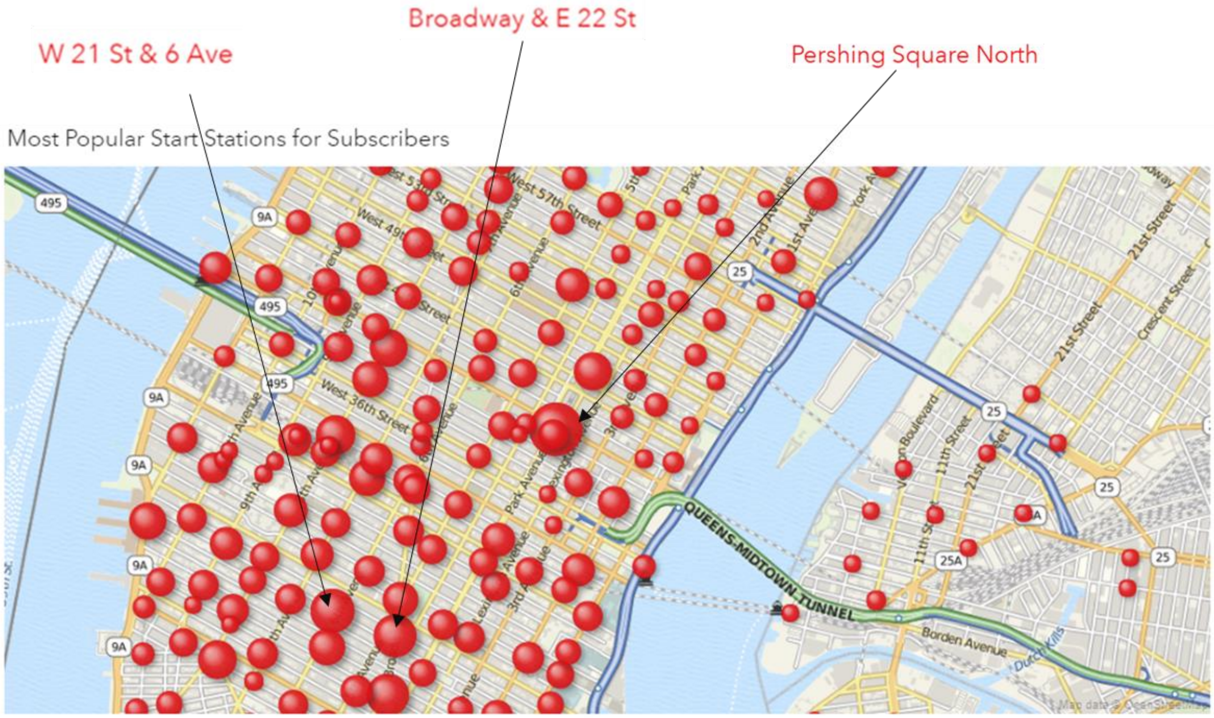


Figure 16: Most Popular Start Stations for Customers

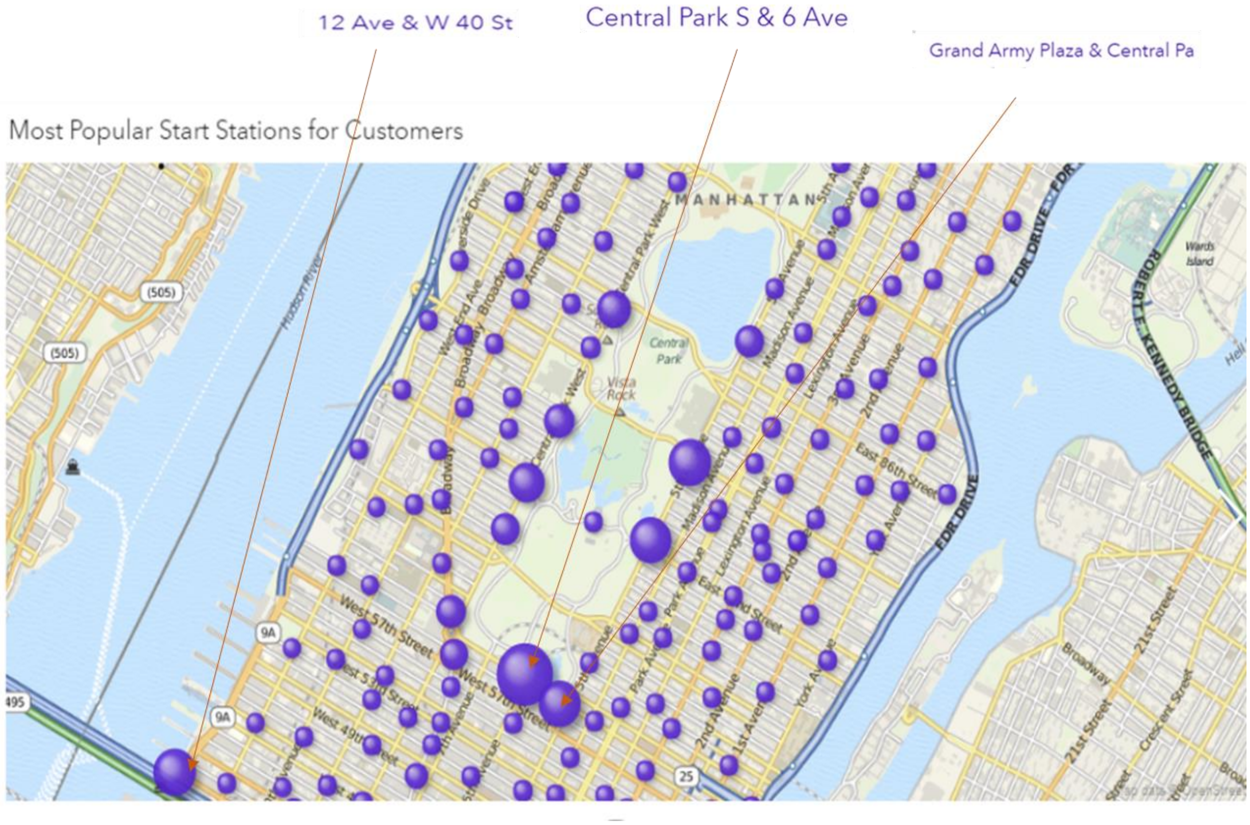


Figure 17: Most Popular End Stations for Subscribers

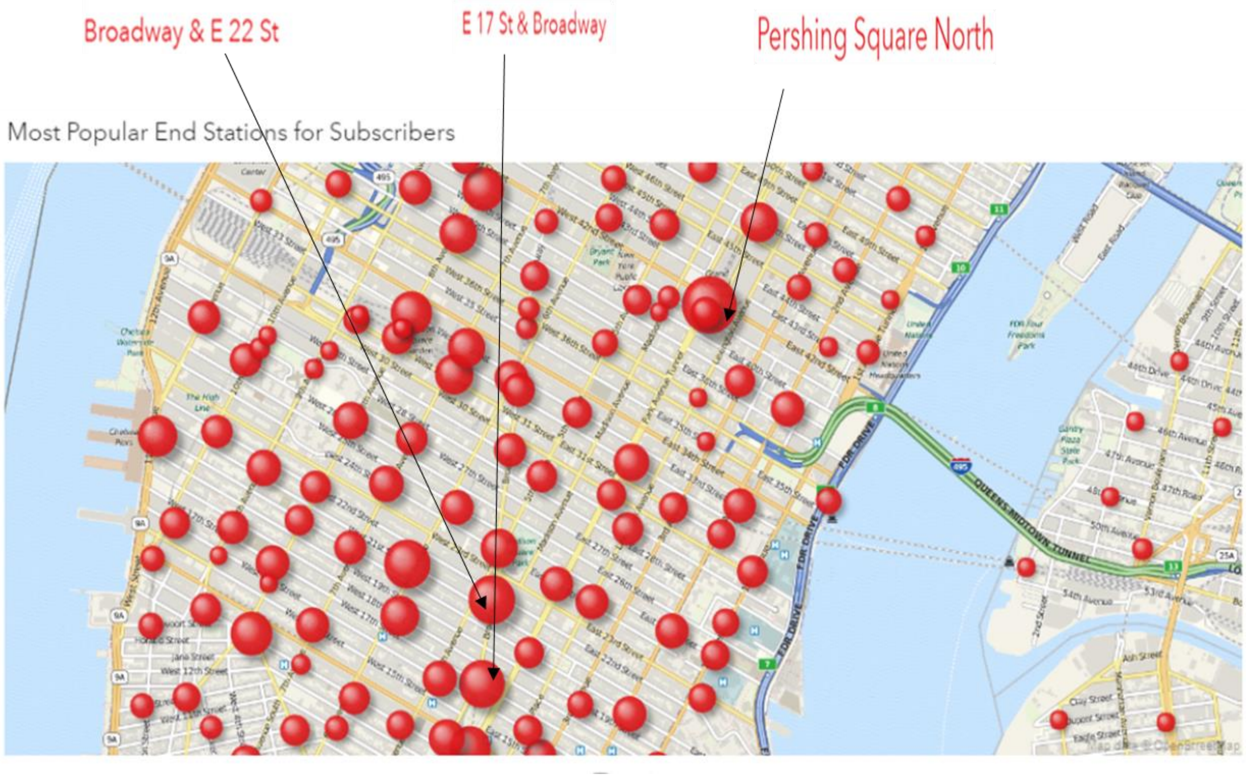


Figure 18: Most Popular End Stations for Customers

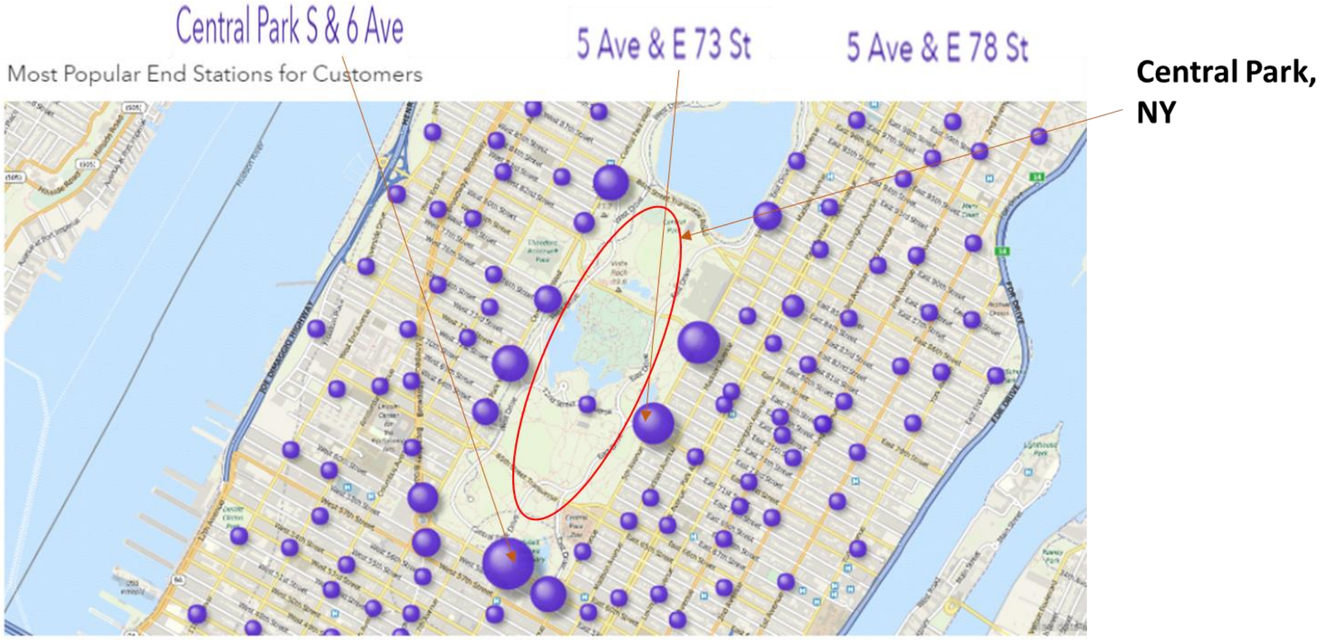


Figure 19: Time Series of Number of Citi Bike Trips and Maximum Daily Temperature (F), 2016

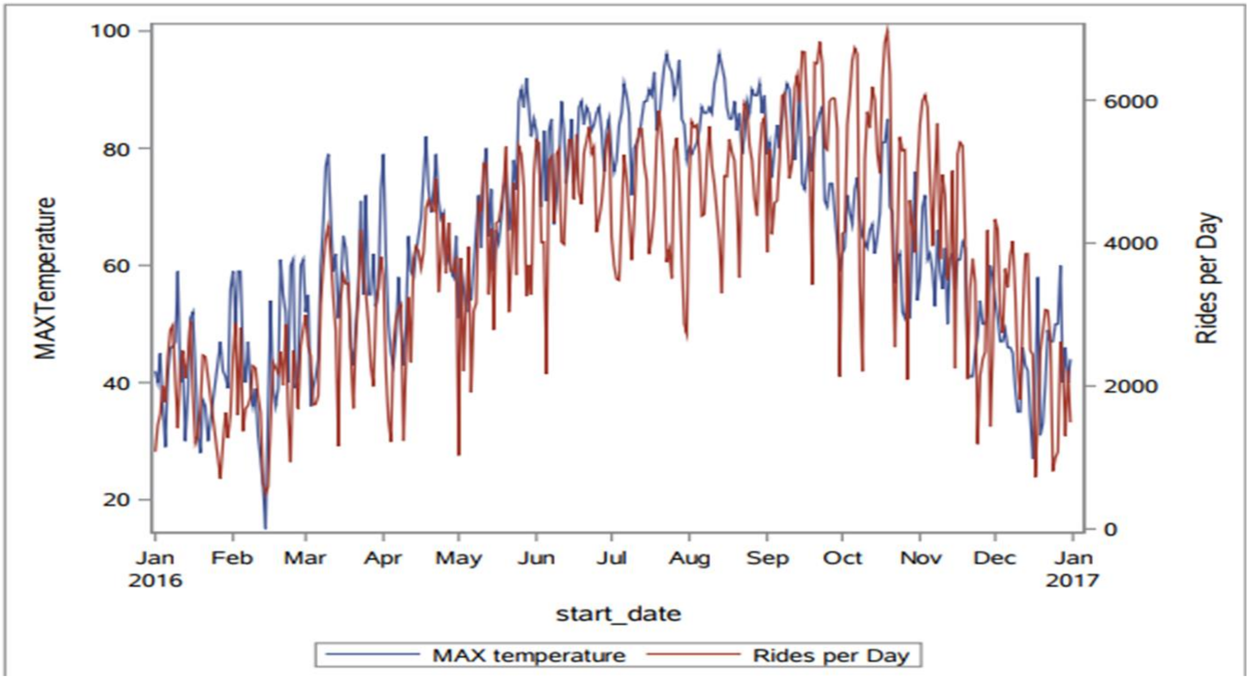


Figure 20: Sizes of individual clusters

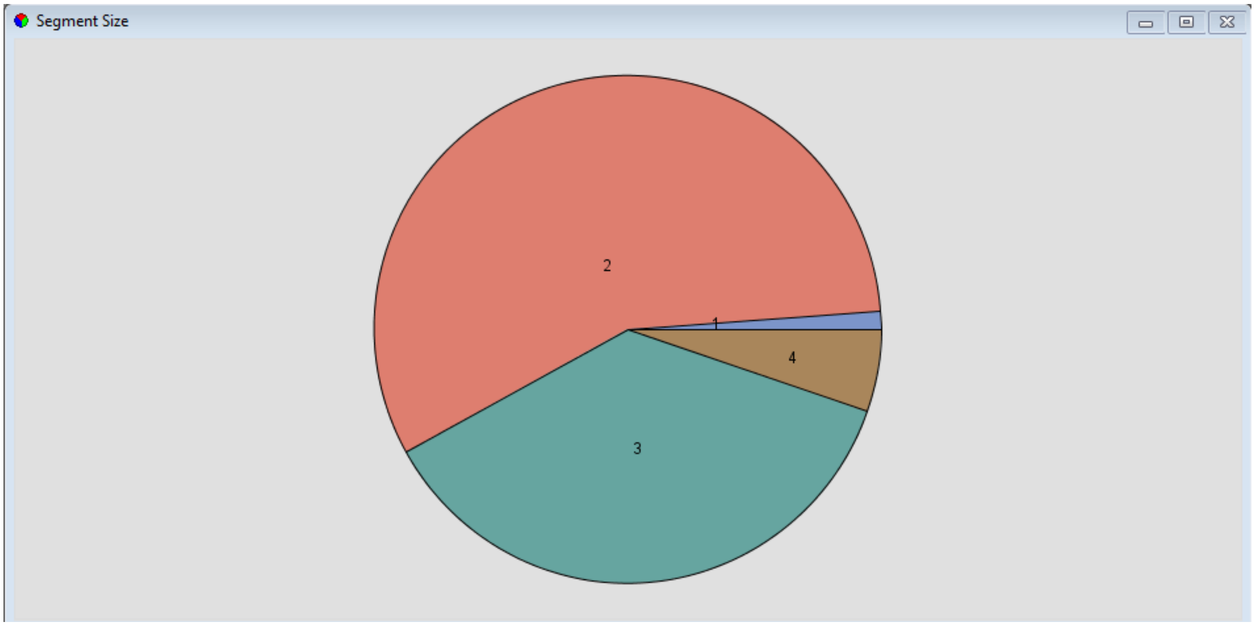
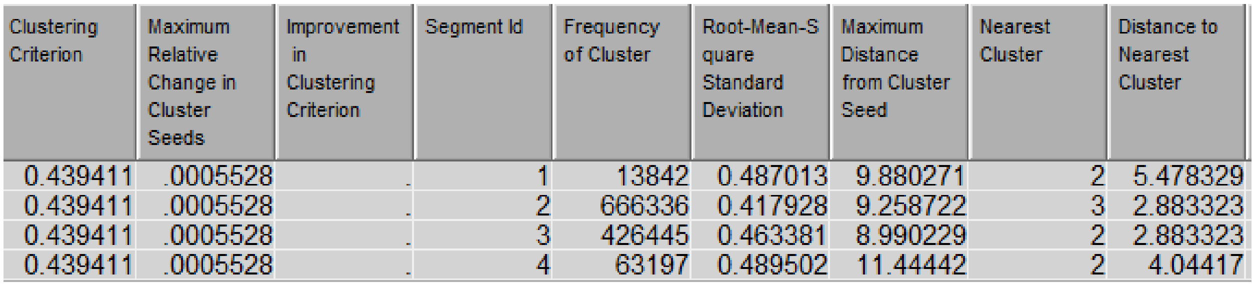


Figure 21: Cluster Statistics



Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster
0.439411	.0005528	.	1	13842	0.487013	9.880271	2	5.478329
0.439411	.0005528	.	2	666336	0.417928	9.258722	3	2.883323
0.439411	.0005528	.	3	426445	0.463381	8.990229	2	2.883323
0.439411	.0005528	.	4	63197	0.489502	11.44442	2	4.04417

Figure 22: EM Flow Diagram

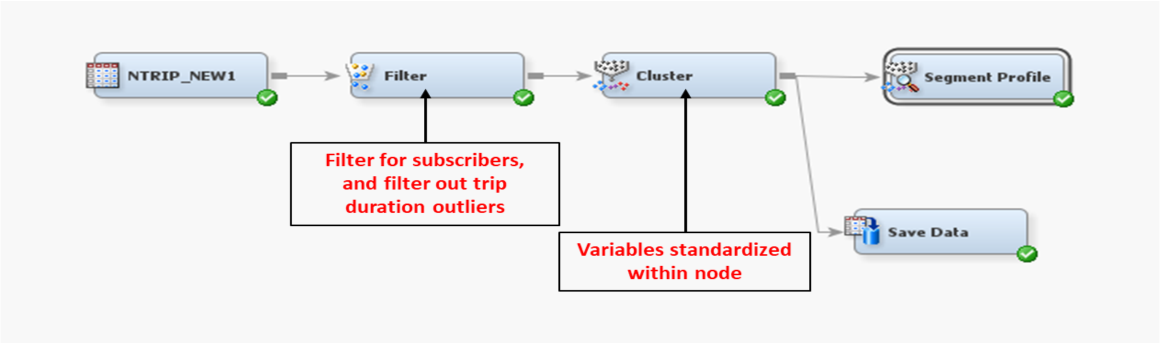
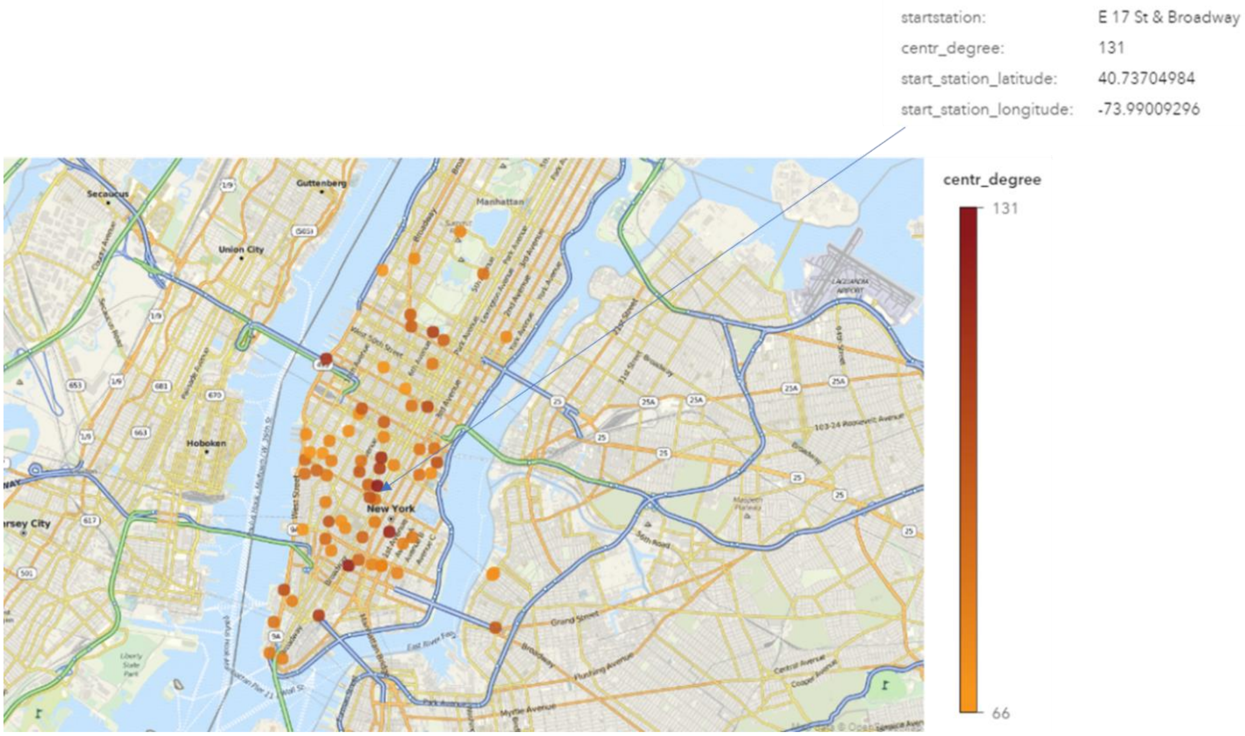
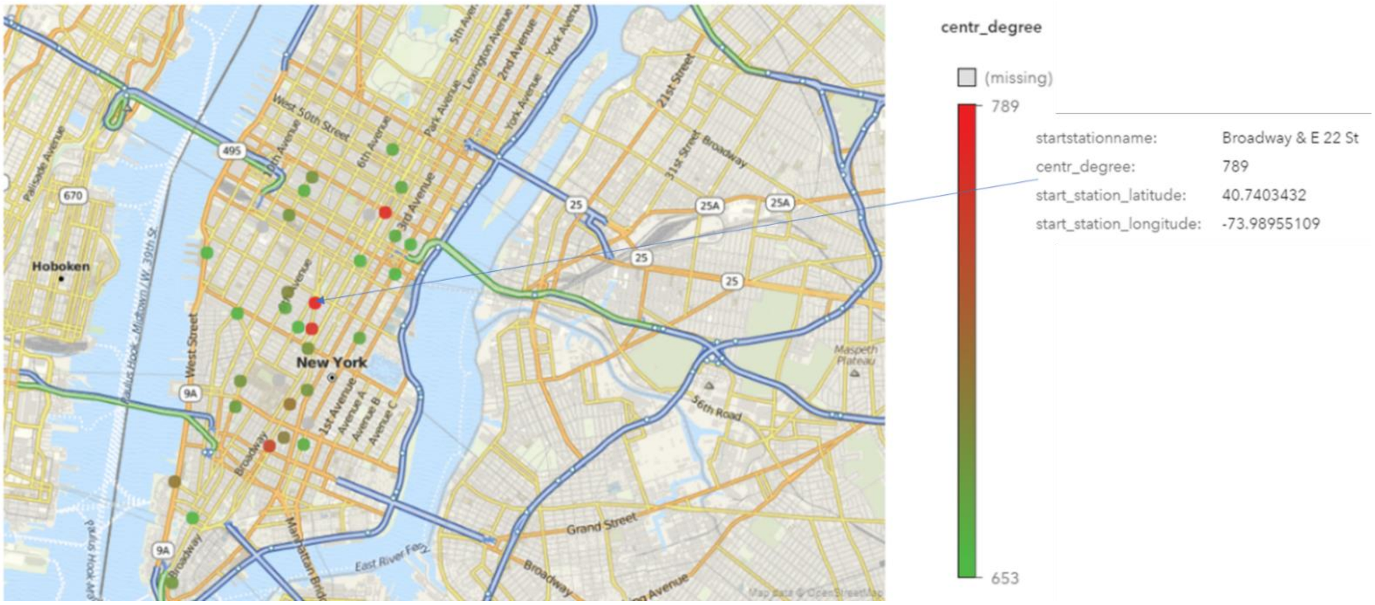

NTRIP_NEW1
Filter
Cluster
Segment Profile
Save Data
Filter for subscribers, and filter out trip duration outliers
Variables standardized within node

Figure 23: Map of Busiest Stations by Degree Centrality, Citi Bike Trip Cluster 1



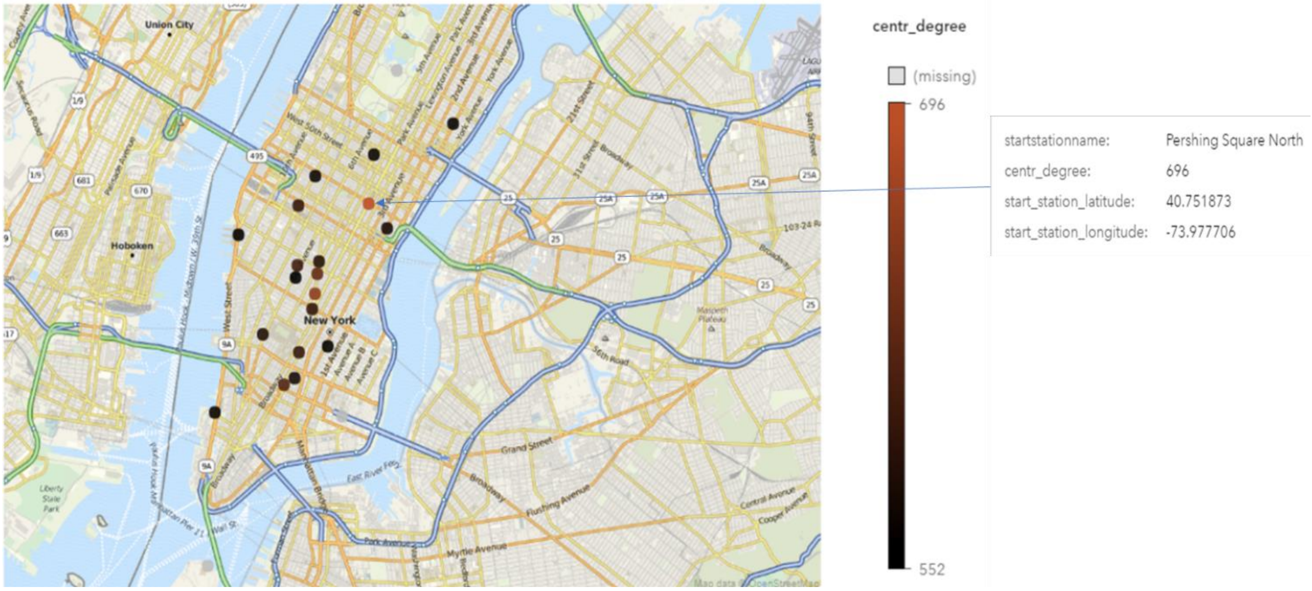
startstation: E 17 St & Broadway
centr_degree: 131
start_station_latitude: 40.73704984
start_station_longitude: -73.99009296

Figure 24: Map of Busiest Stations by Degree Centrality, Citi Bike Trip Cluster 2



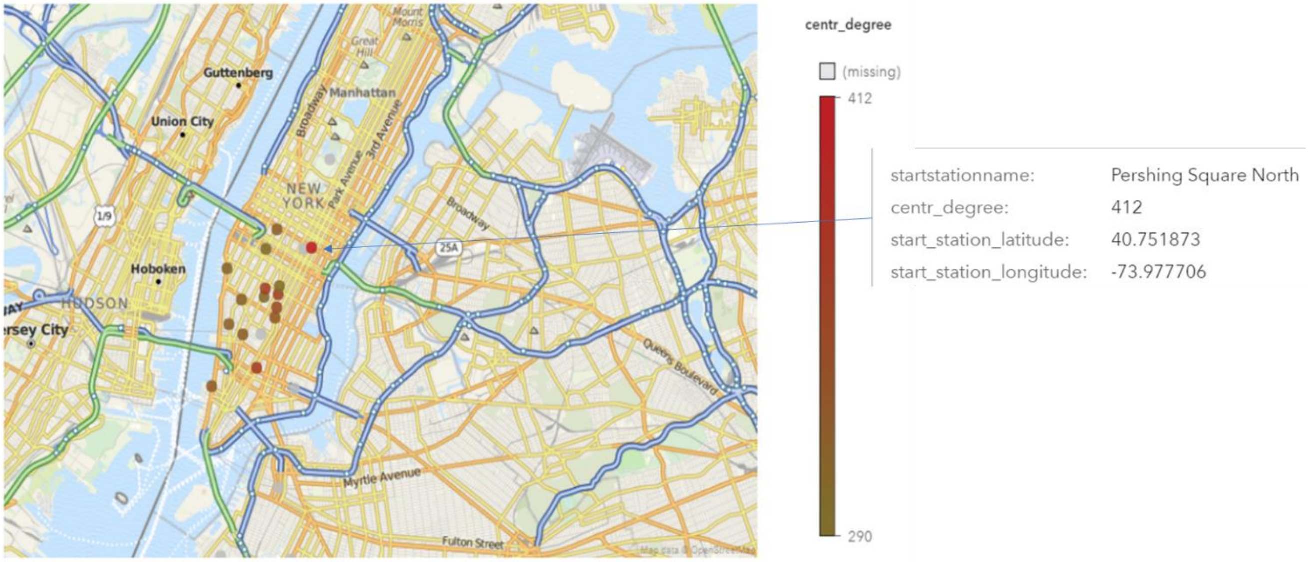
startstationname: Broadway & E 22 St
centr_degree: 789
start_station_latitude: 40.7403432
start_station_longitude: -73.98955109

Figure 25: Map of Busiest Stations by Degree Centrality, Citi Bike Trip Cluster 3



startstationname: Pershing Square North
centr_degree: 696
start_station_latitude: 40.751873
start_station_longitude: -73.977706

Figure 26: Map of Busiest Stations by Degree Centrality, Citi Bike Trip Cluster 4



startstationname: Pershing Square North
centr_degree: 412
start_station_latitude: 40.751873
start_station_longitude: -73.977706

SAS® Code

```
/* Import csv files by month */
%Macro loop;
%LOCAL I;
%LET I = 201601;/* %TO 201612 %by 1; /* Update here when new
datasets become available*/
filename bike&I
'\\cdc.gov\private\L317\icj2\SAS\SASGF_symposium\Data\&I.-
citibike-tripdata.csv';*/
data citibike12;
%let _EFIERR_ = 0; /* SET THE ERROR DETECTION MACRO VARIABLE */
infile '/gpfs/sasdata1/bikeride\&I-citibike-tripdata.csv'
delimiter = ','
missover dsd lrecl=32767 firstobs=2;
     informat tripduration BEST32.;
     informat starttime ANYDTDTM40.;
     informat endtime ANYDTDTM40.;
     informat start_station_id BEST32.;
     informat start_station_name $29.;
     informat start_station_latitude BEST32.;
     informat start_station_longitude BEST32.;
     informat end_station_id BEST32.;
     informat end_station_name $29.;
     informat end_station_latitude BEST32.;
     informat end_station_longitude BEST32.;
     informat bikeid BEST32.;
     informat usertype $10.;
     informat birth_year BEST32.;
     informat gender BEST32.;

     format tripduration BEST12.;
     format starttime datetime.;
     format endtime datetime.;
     format start_station_id BEST12.;
     format start_station_name $29.;
     format start_station_latitude BEST12.;
     format start_station_longitude BEST12.;
     format end_station_id BEST12.;
     format end_station_name $29.;
     format end_station_latitude BEST12.;
     format end_station_longitude BEST12.;
     format bikeid BEST12.;
     format usertype $10.;
     format birth_year BEST12.;
     format gender BEST12.;

     input tripduration starttime endtime start_station_id
```

```
start_station_name $
          start_station_latitude start_station_longitude
end_station_id end_station_name $ end_station_latitude
          end_station_longitude bikeid usertype $ birth_year
gender;

     if _ERROR_ then call symput('_EFIERR_',1); /* set ERROR
detection macro variable */
     run;

/*%END;
%MEND LOOP;
%LOOP;
QUIT;*/

Proc contents data=citibike01;
run;

/*Combining all months of data*/
Data allbike2016;
set CITIBIKE01 CITIBIKE02 CITIBIKE03 CITIBIKE04 CITIBIKE05
CITIBIKE06 CITIBIKE07
     CITIBIKE08 CITIBIKE09 CITIBIKE10 CITIBIKE11 CITIBIKE12;
run;

/*Checking contents*/
Proc contents data=allbike2016;
run;

/*Checking some frequencies*/
Proc freq data=allbike2016;
tables usertype gender;
run;

Proc univariate data=allbike2016;
var tripduration;
histogram;
run;

/*Create some formats*/
Proc format;
     value dow 1 = "Sunday"
                      2 = "Monday"
                      3 = "Tuesday"
                      4 = "Wednesday"
                      5 = "Thursday"
                      6 = "Friday"
                      7 = "Saturday";
     value mth   1 = "January"
                      2 = "February"
                      3 = "March"
                      4 = "April"
                      5 = "May"
                      6 = "June"
                      7 = "July"
                      8 = "August"
                      9 = "September"
                      10 = "October"
                      11 = "November"
                      12 = "December";
     value gender          0 = "Unknown"
                      1 = "Male"
                      2 = "Female";
     value yn    0= 'No'
                      1= 'Yes';
     value rush       1='AM rush hour'
                      2='PM rush hour'
                      0='Not rush hour';
run;


/*Creating some analysis variables*/
Data allbike2016_a;
set allbike2016;
age = 2016 - birth_year;
start_date = datepart(starttime);
start_time = timepart(starttime);
end_date = datepart(endtime);
end_time = timepart(endtime);
weekday = weekday(start_date);
month = month(start_date);
format start_date end_date mmddyy10. start_time end_time time8.
weekday dow. month mth. gender gender.;
run;

/*Check distributions of complete dataset*/
Proc freq data=allbike2016_a;
tables weekday month gender usertype*weekday;
run;

/*Take 10% sample for use in Kennesaw SAS grid*/
Proc Surveyselect data=allbike2016_a out= allbike2016_samp
method=srs samprate=0.1;
run;
```

```
/* Check distributions of sample data same as full dataset*/
Proc freq data=allbike2016_samp;
table weekday month gender;

/* SAS grid administrator put the 10% sample onto the grid */

/* Importing NYC weathe datafile */
PROC IMPORT OUT= work.weather DATAFILE=
"/gpfs/user_home/shebbar/sasuser.v94/weather.csv"
            DBMS=csv REPLACE;
            RUN;


data work.bike;
set bikeride.citibike2016_samp;
run;


data bike;
set bike;
date_part = datepart(starttime);
time_part= timepart(starttime);
format date_part date9.;
hour=hour(time_part);
run;


/*Merging bike trip data with weather data by day */
PROC SQL;
CREATE TABLE merged1 AS
SELECT *
FROM bike, weather
WHERE bike.date_part=weather.date1 ;
QUIT;


proc sql;
create table ntrip as
SELECT count(*),tripduration , starttime , endtime ,
start_station_id , start_station_name , start_station_latitude ,
start_station_longitude , end_station_id , end_station_name ,
end_station_latitude , end_station_longitude , bikeid , usertype
, birth_year , gender , age , start_date , start_time , end_date
, end_time , weekday , 'month'n , date_part , VAR1 , STATION ,
STATION_NAME , 'DATE'n , PRCP , SNWD , SNOW , TMAX , TMIN , AWND
, 'day'n , Month_num , Date1
```

```
FROM merged1
GROUP BY date1;
quit;

data ntrip;
set ntrip;
rename _TEMG001=no_of_trips;
run;

data bikeride.ntrip;
set ntrip;
run;
run;

libname bike '/gpfs/sasdata1/bikeride';

/* Create hour of day var (HoD), workday( 1= yes, 0 = no), rush
hour (rush, 1= am, 2 = pm, 0 = no)*/
Data bike.ntrip_new1;
set bike.ntrip;
HoD = hour(start_time);
if weekday in (2,3,4,5,6) then workday = 1;
else workday = 0;
if HoD in (7,8,9) and workday = 1 then rush = 1;
else if HoD in (16,17,18,19) and workday =1 then rush = 2;
else rush = 0;
run;

/* NYC Bike Share Usage-When & Who?*/
ods graphics / reset imagemap;
title 'NYC Bike Share Usage-When & Who?';
proc sgpanel data=BIKERIDE.NTRIP_NEW1;
   where usertype ="Subscriber" | usertype ="Customer";
  panelby usertype / layout=columnlattice
                 colheaderpos=bottom rows=1 novarname;
  vbar weekday/ response=no_of_trips group=usertype
groupdisplay=cluster clusterwidth=0.8;
  colaxis display=ALL ;
  rowaxis grid;
run;
ods graphics / reset;

/* NYC Bike Share Usage-HourofDay*/
ods graphics / width=25cm height=10cm imagename="test200";
title 'NYC Bike Share Usage-HourofDay';
proc sgpanel data=BIKERIDE.NTRIP_NEW1;
   where usertype ="Subscriber" | usertype ="Customer";
  panelby usertype / layout=columnlattice
                 colheaderpos=bottom rows=1 novarname;
  vbar HoD/ response=no_of_trips group=usertype
groupdisplay=cluster clusterwidth=0.8;
   colaxis display=ALL ;
  rowaxis grid;
run;
ods graphics / reset;

/* NYC Bike Share Usage-month*/
ods graphics / width=25cm height=10cm imagename="test200";
title 'NYC Bike Share Usage-month';
proc sgpanel data=BIKERIDE.NTRIP_NEW1;
   where usertype ="Subscriber" | usertype ="Customer";
  panelby usertype / layout=columnlattice
                 colheaderpos=bottom rows=1 novarname;
  vbar month/ response=no_of_trips group=usertype
groupdisplay=cluster clusterwidth=0.8;
     colaxis display=ALL ;
  rowaxis grid;
run;
ods graphics / reset;

/* Creating Duration categorical variable to analyse over
specified time intervals */
data NTRIP_NEW;
set BIKERIDE.NTRIP_NEW1;
tripdurationinminutes=round(tripduration/60,1);
run;

/*Applying format for trip duation*/
proc format;
value mytripduration
0 - 5 = '0-5'
6 - 10 = '6 - 10'
11 - 15 = '11 - 15'
16 - 30 = '16 - 30'
31 - 60 = '31 - 60'
61-high = '60+'
;
run;

data NTRIP_NEW1;
set NTRIP_NEW;
format tripdurationinminutes mytripduration.;
run;
```

```
/*NYC Bike Share Usage-Duration over specified intervals*/
ods graphics / width=25cm height=10cm imagename="test200";
title 'NYC Bike Share Usage-Duration';
proc sgpanel data=WORK.NTRIP_NEW1;
   where usertype ="Customer" | usertype ="Subscriber";
  panelby usertype / layout=columnlattice
                 colheaderpos=bottom rows=1 novarname;
vbar tripdurationinminutes /group=usertype;
  colaxis display=ALL ;
  rowaxis grid label="Number of Rides";
run;
ods graphics / reset;

/*creating most popular routes*/
data bike.test2;
  set BIKERIDE.NTRIP_NEW1;
  Route= catx("  -  ", of start_station_name end_station_name);
run;
proc print data = bike.test2(obs=20);
run;
/*separating dataset for Customers and Subscribers*/
proc sql;
create table bike.bike_data_Subscriber
as
select * from bike.test2 where usertype='Subscriber';
run;

proc sql;
create table bike.bike_data_Customer
as
select * from bike.test2 where usertype='Customer';
run;


/*144980 distinct routes available out of 1384566.*/
 proc sql;
SELECT count(DISTINCT Route) as distinct_route FROM
bike.bike_data_Subscriber;
run;

 proc sql;
 create table bike.bike_analysis_Subscriber as
select Route, count(Route) as CountOfRoute from
bike.bike_data_Subscriber group by Route;
run;

proc sort data=bike.bike_analysis_Subscriber
```

```
out=bike.bike_analysis_Subscriber_sort;
by descending CountOfRoute;
run;

proc print data=bike.bike_analysis_Subscriber_sort(obs=20);
run;

proc sql;
 create table bike.bike_analysis_Customer as
select Route, count(Route) as CountOfRoute from
bike.bike_data_Customer group by Route;
run;

proc sort data=bike.bike_analysis_Customer
out=bike.bike_analysis_Customer_sort;
by descending CountOfRoute;
run;

proc print data=bike.bike_analysis_Customer_sort(obs=20);
run;

data NTRIP_NEW5;
set NTRIP_NEW1;
format month mth. weekday dow. start_date end_date mmddyy10.
start_time end_time time8.;
run;

proc contents data=NTRIP_NEW1;
run;

proc contents data=NTRIP_NEW5;
run;

proc sql;
select count (*) as nooftrips,start_date,usertype from
ntrip_new5 group by start_date,usertype;
run;

/*Creating no of trips/day variable separately for customers and
subscribers */
proc sql;
create table bikeride.bike_data_Final1 as
select count (*) as nooftrips_Subscriber,* from ntrip_new5 where
usertype='Subscriber' group by start_date,usertype ;
run;
/*151105*/
proc sql;
```

```
create table bikeride.bike_data_Final2 as
select count (*) as nooftrips_Customer,* from ntrip_new5 where
usertype='Customer' group by start_date,usertype ;
run;

/*merging both datasets to get all the variables including newly
created 2 variables*/

data bikeride.bike_data_Final;
set bikeride.bike_data_Final1 bikeride.bike_data_Final2;
by start_date;
run;
/*
 Time Series data preparation
 This is the code that I have taken from sas studio
 *
 */

ods noproctitle;

proc sort data=BIKERIDE.BIKE_DATA_FINAL1
out=Work.preProcessedData;
     by start_date;
run;

proc timedata data=Work.preProcessedData seasonality=7
out=WORK._tsoutput;
     id start_date interval=day setmissing=missing;
     var nooftrips_Subscriber / accumulate=average
transform=none;
run;

data WORK.Time_Series_Data_Prep_Sub(rename=());
     set WORK._tsoutput;
run;

proc print data=WORK.Time_Series_Data_Prep_Sub(obs=10);
     title "Subset of WORK.Time_Series_Data_Prep_Sub";
run;

title;

proc delete data=Work.preProcessedData;
run;

proc delete data=WORK._tsoutput;
run;
```

```
/*Time Series Exploration*/

ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TIME_SERIES_DATA_PREP_SUB
out=Work.preProcessedData;
     by start_date;
run;

proc timeseries data=Work.preProcessedData seasonality=7
plots=(series corr);
     id start_date interval=day;
     var nooftrips_Subscriber / accumulate=none transform=none
dif=0 sdif=0;
run;

proc delete data=Work.preProcessedData;
run;

/*Creating timeseries plot with temperature into account*/

proc sgplot data=bikeride.bike_data_Final;
   series x=start_date y=TMAX/ legendlabel="MAX temperature";
   series x=start_date y=No_of_trips / y2axis legendlabel="Rides
per Day";
   yaxis label="MAXTemperature";
   y2axis label="Rides per Day";
run;


/* tried this code to create animation plot
%macro ClassBubbleAnim(start=, end=, steps=);
  %do i=0 %to &steps-1;
    data bike_data_Final;
       set bikeride.bike_data_Final;
       frac=&start + &i*(&end - &start)/&steps;
       start_station_name=start_station_name*frac;
       end_station_name=end_station_name*frac;
       no_of_trips=no_of_trips*frac;
     run;

     proc sgplot data=bike_data_Final;
      title 'Number of rides per day';
      footnote j=l 'Using gif animation with SGPLOT procedure';
      bubble x= start_station_name y=end_station_name
```

```
size=no_of_trips/ group=usertype dataskin=sheen;
       inset "Step &i of &Steps" / textattrs=(size=12)
position=bottomright;
      keylegend / location=inside position=topleft;
    run;
  %end;
%mend ClassBubbleAnim;

options papersize=('4 in', '3 in') printerpath=gif
animation=start animduration=0.05 animloop=yes noanimoverlay;
ods printer file='ClassBubbleAnim.gif';

ods graphics / width=4in height=3in imagefmt=GIF;
%ClassBubbleAnim(start=0.5, end=1.0, steps=50);
%ClassBubbleAnim(start=1.0, end=0.5, steps=50);

options printerpath=gif animation=stop;
ods printer close;
*/

/* Cluster analysis completed in SAS EM and output datasets
imported back into SAS Studio */
/* Continue with optgraph analysis of each cluster */

/*splitting dataset by cluster*/
data bikeride.cluster1;
set bikeride.clustered;
if _SEGMENT_=1
then output;
run;


data bikeride.cluster2;
set bikeride.clustered;
if _SEGMENT_=2
then output;
run;


data bikeride.cluster3;
set bikeride.clustered;
if _SEGMENT_=3
then output;
run;

data bikeride.cluster4;
set bikeride.clustered;
```

```
if _SEGMENT_=4
then output;
run;

/*graph subset data for optgraph*/

%MACRO optclus;
%do i=1 %to 4;
data bikeride.optclus&i(keep=from to);
      set bikeride.cluster&i(rename=(start_station_id=from
end_station_id=to));
       run;
%end;
%mend;

%optclus;

/*to find centrality*/
%macro central;
%do i=1 %to 4;
proc optgraph
graph_direction = directed
data_links = bikeride.optclus&i
out_nodes = bikeride.NodeSetOut&i;
centrality
degree = both;
run;
%end;
%mend;

%central;

/*merging centrality output with the other variables*/
PROC SQL;
CREATE TABLE bikeride.inter AS
SELECT start_station_id , start_station_name ,
start_station_latitude , start_station_longitude FROM
BIKERIDE.NTRIP_NEW1;
RUN;
QUIT;

PROC SORT DATA=bikeride.inter
 OUT=bikeride.inter1
 NODUPRECS ;
 BY start_station_id ;
RUN ;
```

```
data bikeride.inter1;
set bikeride.inter1;
rename start_station_id=node;
run;


%macro combine;
%do i=1 %to 4;
proc sort data=bikeride.nodesetout&i;
by node;
run;
data bikeride.merged&i;
merge bikeride.inter1 bikeride.nodesetout&i;
by node;
run;
%end;
%mend;

%combine;
```