Gmail

PMC H <pmchalvorsen@gmail.com>

A dataset of Citibikes in NYC
1 message

Per <pmchalvorsen@gmail.com>
To: Per <pmchalvorsen@gmail.com>

Sun, Sep 15, 2019 at 7:34 PM

A dataset of Citibikes in NYC

Friday, September 13, 2019
14:34

Hi, and welcome to this analysis of the CitiBikes in New York City. The main goal of this report is to explain and answer the following three questions:

- ***What distribution does the trip durations follow?***
- ***What was the most popular route?***
- ***Did any new stations appear in the middle of the set? How do you know?***

The data set consisted of 16 attributes with over 30 million entries. The attributes that are most relevant for the questions this report will focus on are the `tripduration, start_station_name` and `end_station_name.` We'll start by looking at the shape of the distribution of trip durations.

The shape:

The only attribute needed for analyzing the distribution of time spent on a bike is the column titled `tripduration`.

This column contains an integer datatype with the amount of seconds each trip took. Since this data set was so big, I decided to split the durations into intervals of 15, 30 and 60 seconds. This resulted in smaller .csv, without losing too much of the valuable information needed to find which distribution the data follows. An example of the SQL query I used to select and sort the `tripduration` into intervals of quarter-minutes is shown below.
SQL query:

```
SELECT
  (tripduration - MOD(tripduration, 15)) / 15 as qmin_interval,
  COUNT(*) as num_trips


FROM
  `bigquery-public-data.new_york.citibike_trips`


GROUP BY
  1
ORDER BY
  qmin_interval ASC
 LIMIT
  15000
```

On the Google Cloud Platform, visualization of data can easily be done in Google's Data Studio. A comma-separated-values file can also be exported if you want to plot your own graph using other languages, like Python or R. I chose to do the latter in order to find a curve that best fit my data.

Python:

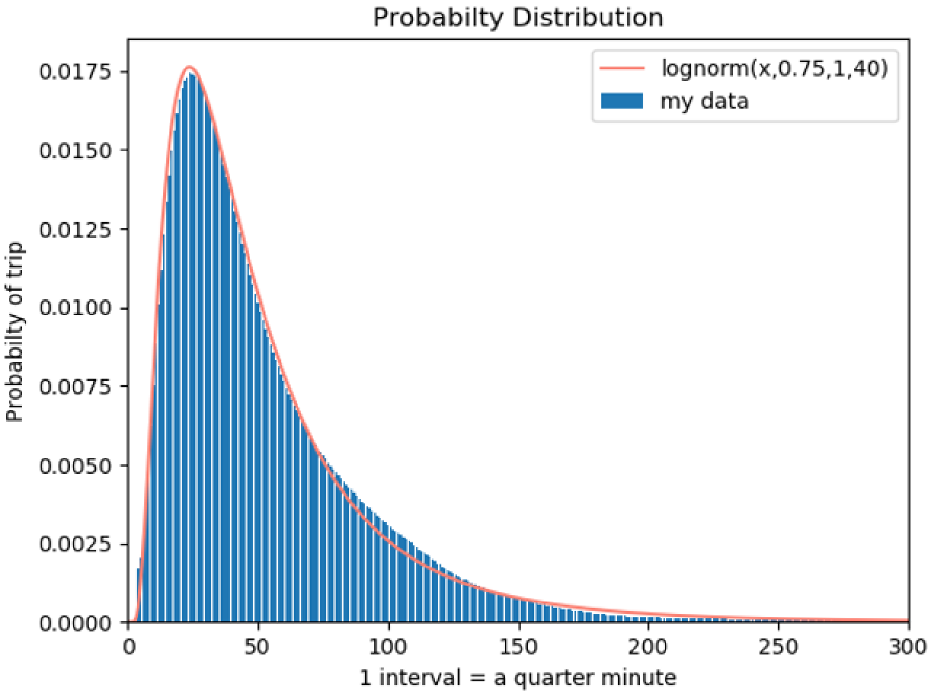
There were a few steps to get from .csv-file to distribution plot analysis. First, the data from the .csv file needed to be read in. To do this, I used pandas `read_csv()` function. I also pulled the time and corresponding number of trips columns into their own respective variables `qmin_interval` and `num_trips`.

```
trip_data = pandas.read_csv("trip_per_quarter_minute.csv")

qmin_interval = trip_data["qmin_interval"]

num_trips = trip_data["num_trips"]
```

Since I wanted to find a matching probability distribution for my data, I needed to convert the values in my `num_trips` column to percentages. This was done by simply dividing the `num_trips` array by the sum of the same column.

```
prob_trips = num_trips/sum(num_trips)
```

The next step was to plot the data and see what trends show up. After some trail and error, I decided the trend followed a **log-normal distribution** with σ ≈ 0.75. The plot of `scipy.stats' lognorm(x, s, loc, scale)` function on top of the data from my .csv file is shown below.



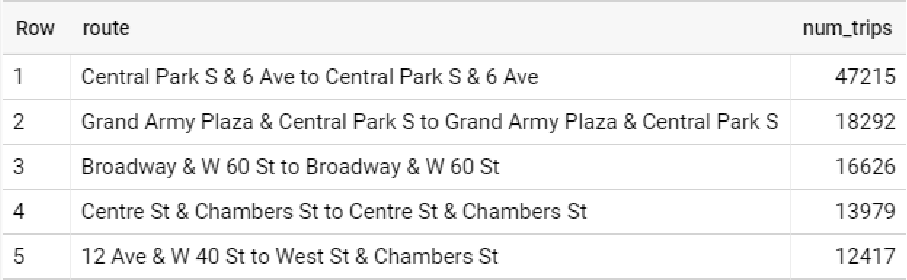
As you can see, the lognormal-distribution fits surprisingly well to these completely natural data entries. A peak occurs at 6 minutes (remember, this plot shows intervals of 15 seconds). The plot also shows a truncated version of the actual dataset. I chose to cut off the values past 75 minutes, since there were relatively so few entries with this high of a value.

Most popular route:

The next objective was to find the most popular route among the city bikes users. The columns needed to find this are the `start_station` and `end_station` columns. A UNION or CONCAT operation is needed in order to count the number of trips per route. I chose to join the starting station with the ending station using the CONCAT() function, in order to include a whitespace between the two station names. Here is the SQL-query used:

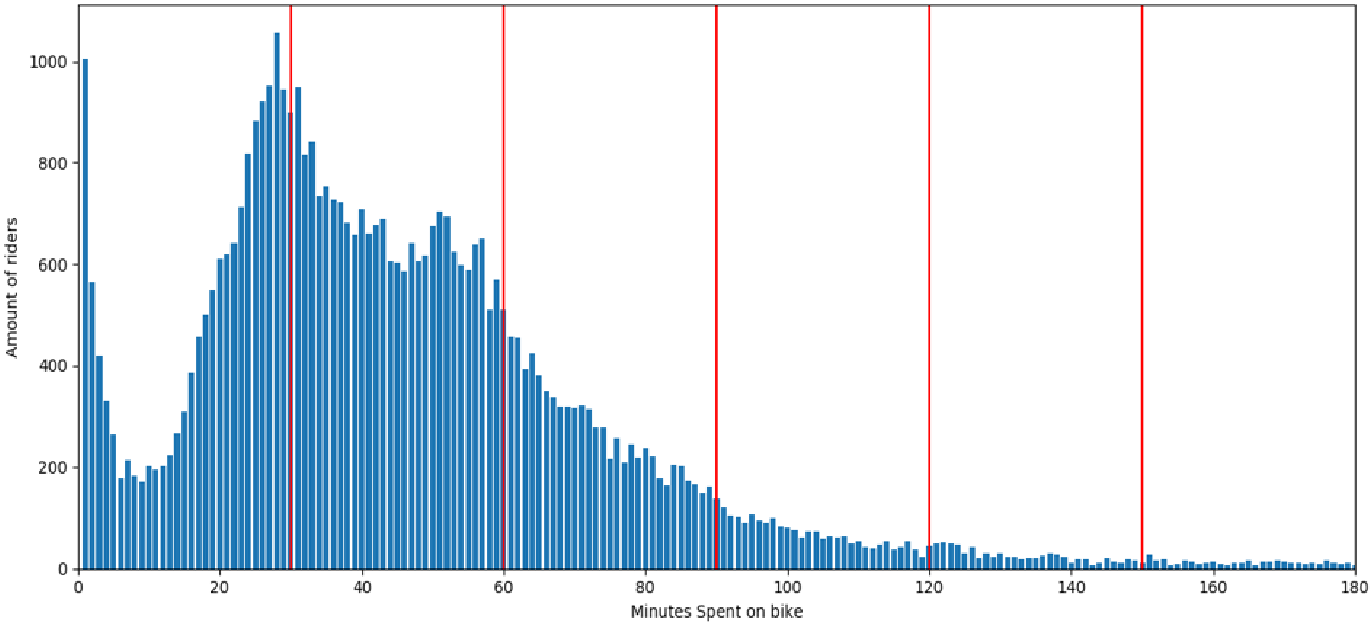
```
SELECT
  CONCAT(start_station_name, ' to ', end_station_name) as route,
  COUNT(*) as num_trips

FROM
  `bigquery-public-data.new_york.citibike_trips`

GROUP BY
  start_station_name,
  end_station_name

ORDER BY
  num_trips DESC
 LIMIT
  1000
```

As a result, I found the most frequently used route was picking up from Central Park S & 6th Avenue and delivering to the same station. The top 5 most popular routes are shown in the table below.



Row	route	num_trips
1	Central Park S & 6 Ave to Central Park S & 6 Ave	47215
2	Grand Army Plaza & Central Park S to Grand Army Plaza & Central Park S	18292
3	Broadway & W 60 St to Broadway & W 60 St	16626
4	Centre St & Chambers St to Centre St & Chambers St	13979
5	12 Ave & W 40 St to West St & Chambers St	12417

Since the start and stop stations were the same for the top four choices, I decided to look a little deeper into this specific route. I wanted to see how long people usually spent on these rolls through Central Park on CitiBikes, so I a pulled a query selecting the trip durations for this specific route. It turns out, most people used between 20-40 minutes on these trips, with an absolute peak at 28 minutes. The plot below shows the distribution for the first three hours used on this specific route.

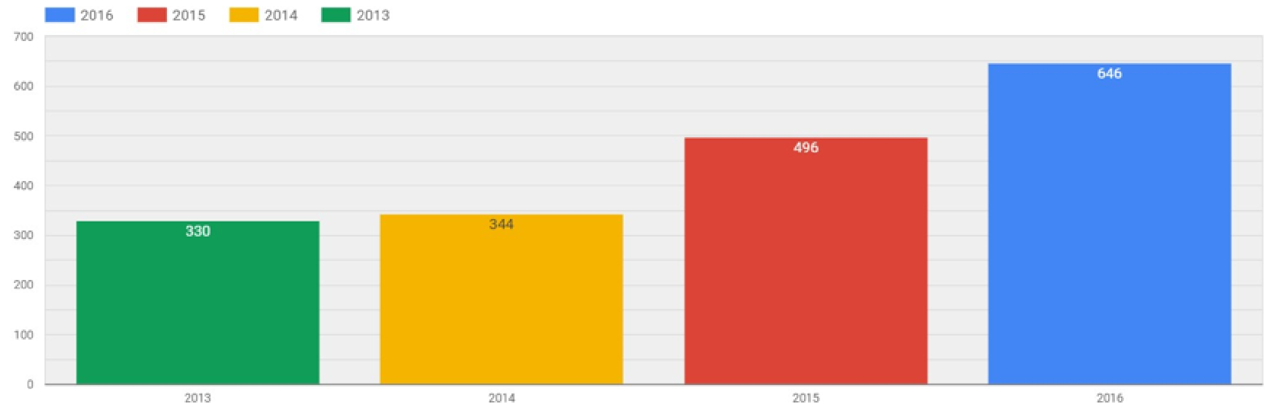


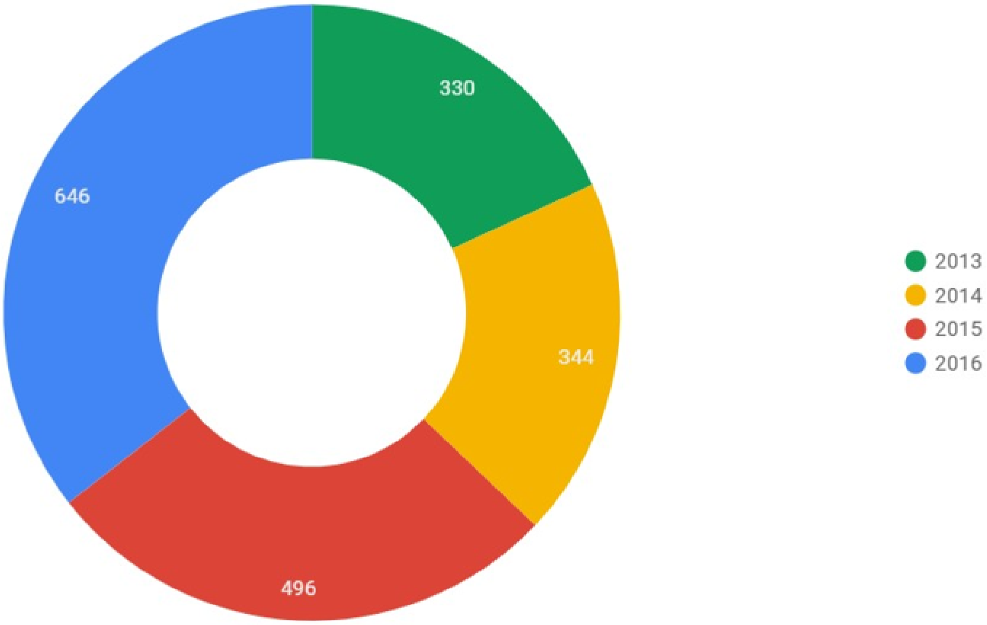
New stations:

To find out if any new stations were added to the CitiBike system, I decided to pull the names of the all of the start stations for each year. If there is an increase in the total number of start stations as the years increase, then there has to have been new stations added into the system. Here's the SQL-query I used:

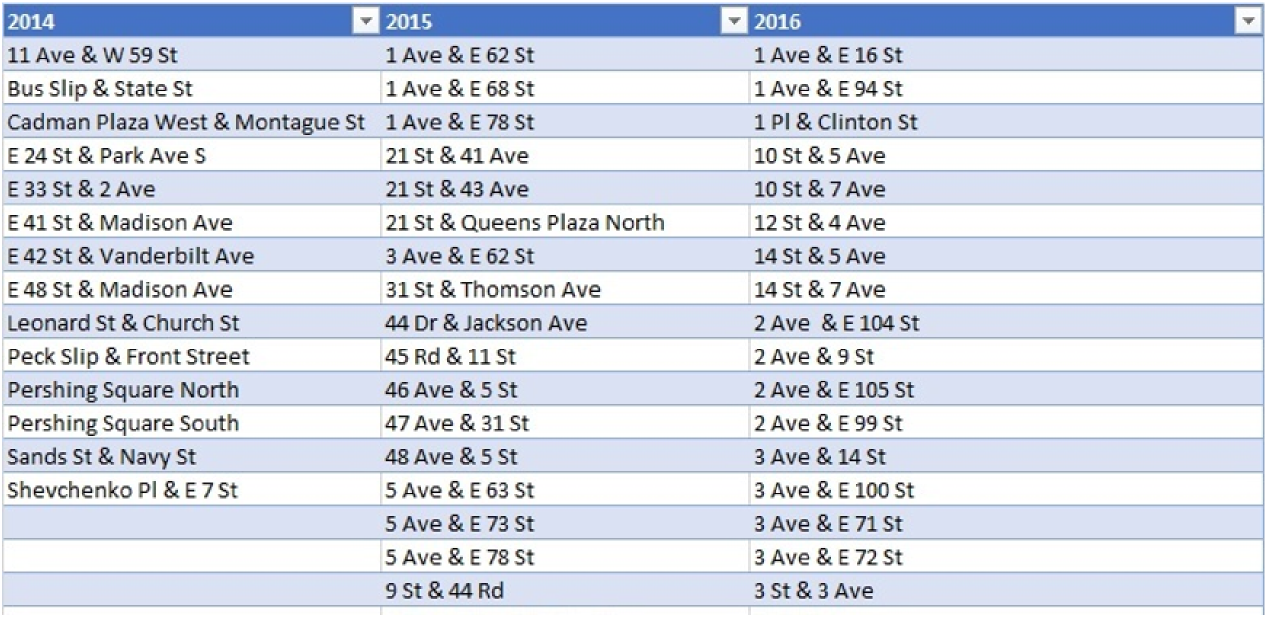
```
SELECT
  start_station_name as names,
  SUBSTR(STRING(TIMESTAMP_TRUNC(starttime, YEAR)), 0, 4) as years,
  COUNT(*) as num_trips

FROM
  `bigquery-public-data.new_york.citibike_trips`

GROUP BY
  years,
  names
ORDER BY
  names ASC,
  years ASC
 LIMIT
  2000
```

The results I found showed 14 news stations added after one year, and additional 152 stations added the next year. By 2016, the total number of stations nearly doubled from 2013, to a whopping 642. To visualize this information, made a simple bar graph and a donut chart in Google's Data Studio.





I wrote a simple Python code to get a list of the new stations added each year. These lists are shown below.



2014	2015	2016
11 Ave & W 59 St	1 Ave & E 62 St	1 Ave & E 16 St
Bus Slip & State St	1 Ave & E 68 St	1 Ave & E 94 St
Cadman Plaza West & Montague St	1 Ave & E 78 St	1 Pl & Clinton St
E 24 St & Park Ave S	21 St & 41 Ave	10 St & 5 Ave
E 33 St & 2 Ave	21 St & 43 Ave	10 St & 7 Ave
E 41 St & Madison Ave	21 St & Queens Plaza North	12 St & 4 Ave
E 42 St & Vanderbilt Ave	3 Ave & E 62 St	14 St & 5 Ave
E 48 St & Madison Ave	31 St & Thomson Ave	14 St & 7 Ave
Leonard St & Church St	44 Dr & Jackson Ave	2 Ave & E 104 St
Peck Slip & Front Street	45 Rd & 11 St	2 Ave & 9 St
Pershing Square North	46 Ave & 5 St	2 Ave & E 105 St
Pershing Square South	47 Ave & 31 St	2 Ave & E 99 St
Sands St & Navy St	48 Ave & 5 St	3 Ave & 14 St
Shevchenko Pl & E 7 St	5 Ave & E 63 St	3 Ave & E 100 St
	5 Ave & E 73 St	3 Ave & E 71 St
	5 Ave & E 78 St	3 Ave & E 72 St
	9 St & 44 Rd	3 St & 3 Ave

Analyzing these results showed that not only were there new stations that were added as the years increased, but there must have also been some stations that were closed and removed from the system. I know this because the above graphs only show the total number of stations per year, whereas the generated lists hold the names of the newly added stations per year. Since the lengths of the lists were longer than the differences we see in the graphs above, there must be some names that were taken away in graph totals. In 2015 and 2016 were there respectively 13 and 20 stations removed from the system. The numbers from 2014 match perfectly, so no stations were removed this year.

Summary:

We've now seen that the data follows a log-normal distribution with a σ ≈ 0.75. We also found that the 4 most popular routes for the entire 4 year period the dataset covers were picked up and dropped off at the same stations, with Central Park S and & 6th Avenue as the most popular. Finally, not only were there new stations added each year, but there were also stations that were removed from the system. The dataset used to answer these questions also contained information on the ages and membership types of each usage, along with the exact longitude and latitude for each stations. There is plenty more information to pull from these data, but I'll save that for the next guy. Thanks for paying attention until this point! I hope this report fulfilled all your expectations!

Sent from Mail for Windows 10