City Bike Demand Prediction and Analysis

Group 3

Zeyang Lin, Xiao Wan, Karan Kandhaswamy, Nitika Puri

Instructor: Zac Wentzell

Introduction

Citi Bike is New York City's bike share system, and the largest in the nation. Citi Bike launched in May 2013 and has become an essential part of the NY transportation network. It's fun, efficient and affordable – not to mention healthy and good for the environment. The bikes can be unlocked from one station and returned to any other station in the system, making them ideal for one-way trips. Many people say that is the best way to travel around in NYC. We can pick up a bike at one of hundreds of stations around Manhattan, Brooklyn, Queens and Jersey City and see bike availability on the Station Map or mobile app. Take as many short rides as you want while your pass is active. The user can end the ride where ever he wants to if there is a city bike station. Slide your bike firmly into an empty dock and wait for the green light to make sure it's locked. City bike keeps sponsoring different events and plans to offer more discount to their customers like bike month and more. The most popular bike rides are in Central park, Hudson River Gateway and Brooklyn Bridge Park.

People use bike share to commute to work or school, run errands, get to appointments or social engagements, and more. Citi Bike is available for use 24 hours/day, 7 days/week, 365 days/year, and riders have access to thousands of bikes at hundreds of stations across Manhattan, Brooklyn, Queens and Jersey City." Such kind of massive commuter force also generates massive amounts of data.

Problem Statement

After having a look at the data, we observed that most of the bikes are used by people for their work. Hence the start station may not always be the same as the stop station. This is where the problem arises.

The number of bikes that leave the station are more than the number of bikes that return back by the end of the day. This causes less or no bikes at few popular bike stations. And also many times we notice that people do not find a dock to place their bike because the number of bikes returned are more than the number of bikes taken out.

We used the data from https://s3.amazonaws.com/tripdata/index.html and inferred few important things about this study. One of them is to predict the demand of bikes at each station on a daily basis and the availability of bikes.

Data Exploration

The data got from website “https://s3.amazonaws.com/tripdata/index.html” was in the form of csv, which had datasets of all the rides made using Citi bike in New York City in a period of 1 month. We used the dataset for the month of December.

By using Web Scraping technique, we gathered the Total No. of Docks present at each station (how many bikes the stations have facilities to hold the bikes).

The Dataset had some missing values (half completed rides). So we filled the missing values with the mean of that current field (column) and plotted a bar Graph with x and y coordinates as “Total No. of docks the station has” and “No. of stations” to represent the how many stations have similar No. of Docks.

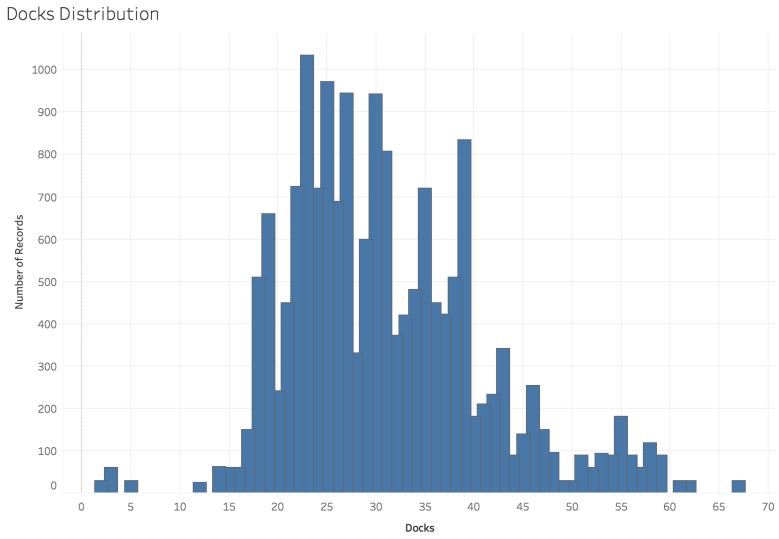


Figure 1

Now we took some random stations and wanted to see how the filtered data is by plotting a graph with x and y axis values as “Bike_Flow” and the “date” values respectively. The variable Bike_Flow means the approximation(difference) between the No. of bikes which were taken out of a station to the No. of bikes which were brought in to the station.

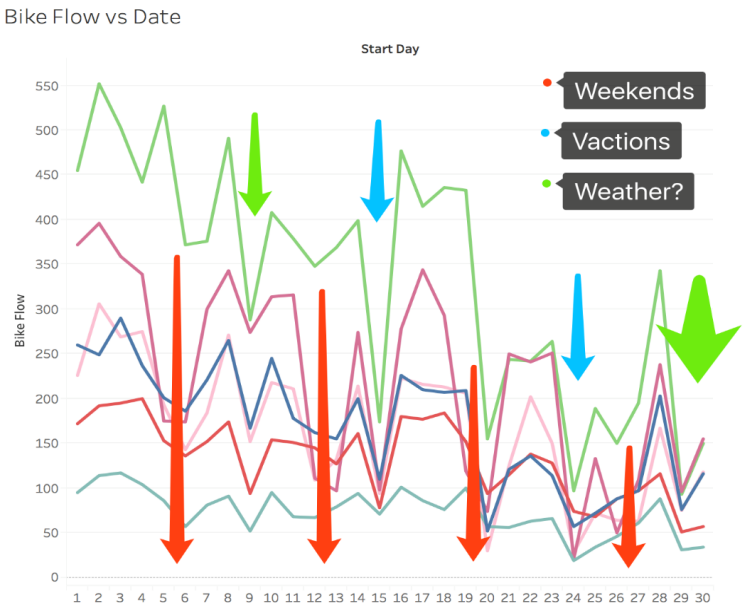


Figure 2

As you can see in the above graphs, the flow rate decreased in some days. The Red arrow represents that the bike flow decreased on Weekends and the Blue arrow represents the

decrease in bike flow during holidays (Vacation holidays). Those decreases represented by green arrows are what we try to find out, which might be explained by a bad weather.

Organized & Cleaned Data

Now that the Filtered data is done, we need to organize and clean the data and add some more variable to it so that it can we used for regression and predicting the data.

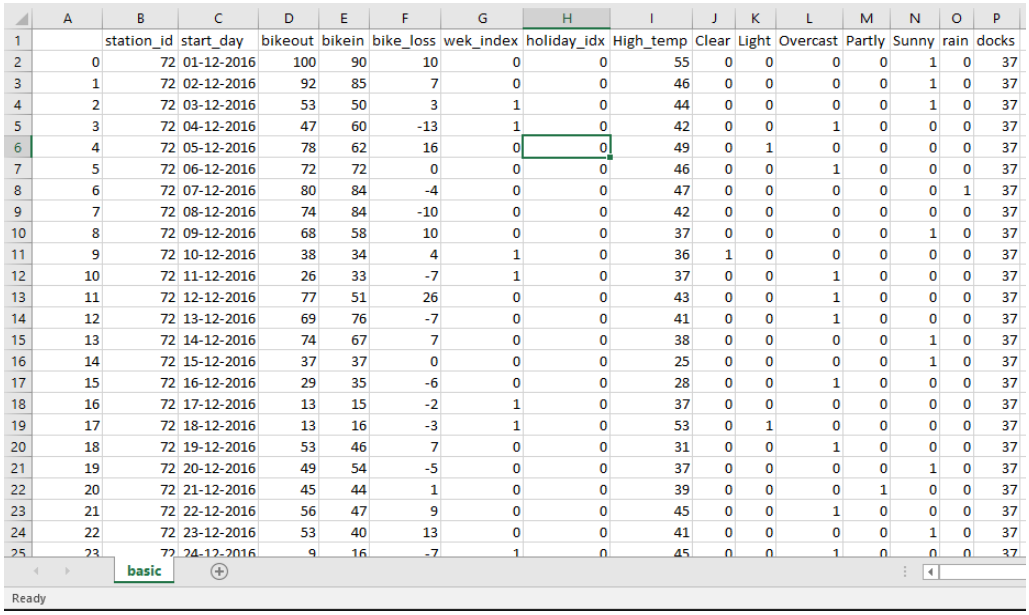
We used the world weather online API and gathered data for the different days the filtered data used. The data gathered are :

- Max temperature for the date
- The average type of Climate for that date

The average type of Climate for a particular day was calculated by taking the values that appeared most frequently on that day. i.e., The api gives values on the Climate type in a hourly manner. So we had to take the frequently appeared climate type for that day.

The different climate types were Clear, Light, Overcast, Partly, Sunny and rain.

We then assigned dummy variables (flags of (0's and 1's) for the type of climate for each day and represented the climate types as columns



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		station_id	start_day	bikeout	bikein	bike_loss	wek_index	holiday_idx	High_temp	Clear	Light	Overcast	Partly	Sunny	rain	docks
2	0	72	01-12-2016	100	90	10	0	0	55	0	0	0	0	1	0	37
3	1	72	02-12-2016	92	85	7	0	0	46	0	0	0	0	1	0	37
4	2	72	03-12-2016	53	50	3	1	0	44	0	0	0	0	1	0	37
5	3	72	04-12-2016	47	60	-13	1	0	42	0	0	1	0	0	0	37
6	4	72	05-12-2016	78	62	16	0	0	49	0	1	0	0	0	0	37
7	5	72	06-12-2016	72	72	0	0	0	46	0	0	1	0	0	0	37
8	6	72	07-12-2016	80	84	-4	0	0	47	0	0	0	0	0	1	37
9	7	72	08-12-2016	74	84	-10	0	0	42	0	0	0	0	0	0	37
10	8	72	09-12-2016	68	58	10	0	0	37	0	0	0	0	1	0	37
11	9	72	10-12-2016	38	34	4	1	0	36	1	0	0	0	0	0	37
12	10	72	11-12-2016	26	33	-7	1	0	37	0	0	1	0	0	0	37
13	11	72	12-12-2016	77	51	26	0	0	43	0	0	1	0	0	0	37
14	12	72	13-12-2016	69	76	-7	0	0	41	0	0	1	0	0	0	37
15	13	72	14-12-2016	74	67	7	0	0	38	0	0	0	0	1	0	37
16	14	72	15-12-2016	37	37	0	0	0	25	0	0	0	0	1	0	37
17	15	72	16-12-2016	29	35	-6	0	0	28	0	0	1	0	0	0	37
18	16	72	17-12-2016	13	15	-2	1	0	37	0	0	0	0	0	0	37
19	17	72	18-12-2016	13	16	-3	1	0	53	0	1	0	0	0	0	37
20	18	72	19-12-2016	53	46	7	0	0	31	0	0	1	0	0	0	37
21	19	72	20-12-2016	49	54	-5	0	0	37	0	0	0	0	1	0	37
22	20	72	21-12-2016	45	44	1	0	0	39	0	0	0	1	0	0	37
23	21	72	22-12-2016	56	47	9	0	0	45	0	0	1	0	0	0	37
24	22	72	23-12-2016	53	40	13	0	0	41	0	0	0	0	1	0	37
25	23	72	24-12-2016	9	16	-7	1	0	45	0	0	1	0	0	0	37

Figure 3

Variables

The variable which we used for regression (Dependent Variable) are

Y-Axis = "bike_loss" which is found but getting the difference between the out bikes and in bikes for each station on a particular day (positive values denote the bikes lost and negative values denote the bike gained on that day)

The variables which we used for regression (Independent Variables) are

X-Axis = comprises of columns below:

- **wek_index**= which ranges from 0-1 where 0 denotes Weekday and 1 denotes Weekend.
- **holiday_idx =** which ranges from 0-1 where 1 denotes holidays and 0 denotes it's not.
- **High_temp =** this column has values of highest temperature in F, on that day.
- **Clear, Light, Overcast, Partly, Sunny and rain** = These columns has range 0-1 where 1 denotes that climate for that day and 0 denotes it's not.

Regression

1. Grid Search (Inside Each Model)

Grid search is an approach to parameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid. In our case, we do grid search for each station, aiming to find out best parameter for the model used.

We decided to compare 4 models, which are: Linear Regression, Lasso Regression, Random Forest, and Neural Network. Among those, only linear regression has no parameter to tune, so we used grid search to tune the parameters of Lasso Regression, Random Forest and Neural Network separately.

In detail, Figure4 shows all parameters we used for each model. For example, we try to tune number of estimators & maximum features for Random Forest and remain other parameters of RF default. So we provided 3 options for number of estimates('n_estimates') and 5 options for maximum features('max_features'), then for each unique station, try all combinations of them (3*5=15). Cross validation counted error for each one of the 15, and chose the one for the least error.

```
RF:
reg_type = RandomForestRegressor()
param_grid = [
    {'n_estimators': [3, 10, 30], 'max_features': [2, 4, 6, 8, 9]}
  ]

Lasso:
reg_type = linear_model.Lasso()
param_grid = [{'alpha':[0,000.1,0.01,0.1,1,2,5,10.2,12.4,15,18]}]


# MLPRegressor -- Neural Network
reg_type = MLPRegressor()
param_grid = [{ 'hidden_layer_sizes':[(5),(50),(50,100),(100,50,100)],
        'activation':['relu','identity', 'logistic', 'tanh'] ,
        'max_iter':[50,200,500],
        }]
```

Figure4

After we finish one station for one model, we did same model for different stations. After finishing all stations for one model, we changed to another model. Partial Grid Search Result is shown in Figure5. (Index is station ID, so here we only show station id from 72-164).

	Lasso-Alpha	MLP-Activation	MLP-HiddenLayerSize	MLP-MaxIter	RF-MaxFeatures	RF-NEstimates
72	12.4	relu	50	200	4	10
79	1.0	tanh	50	200	9	30
82	1.0	relu	5	200	9	30
83	1.0	relu	50	50	2	30
116	15.0	relu	50	200	4	30
119	10.2	relu	50	500	2	10
120	1.0	logistic	5	500	2	30
127	1.0	tanh	(50, 100)	500	8	10
128	10.2	relu	50	50	4	30
143	0.1	tanh	50	500	6	30
144	10.2	relu	(100, 50, 100)	200	2	10
146	15.0	identity	50	200	2	30
147	10.2	logistic	(50, 100)	50	8	10
150	1.0	tanh	50	500	8	10
151	18.0	tanh	5	200	6	30
152	10.2	logistic	50	50	2	10
153	18.0	logistic	(100, 50, 100)	200	2	10
157	10.2	tanh	5	50	6	10
161	0.0	tanh	50	500	8	3
164	10.2	tanh	(100, 50, 100)	50	6	30

Figure 5

2. Models Comparison (Between Models)

For model comparison, as we mentioned above, we tried to compare 4 different regression models: Linear Regression, Lasso Regression, Random Forest, and Neural Network.

Since we have best parameters for each model, the accuracy counted for each model is its best, now all we need is to find out a unified measurement to compare the accuracy of these four models.

Based on the features of each model, we decided to use RMSE (Root Mean Squared Error) to be the measurement tool for each model. Hence, lower RMSE means better accuracy. Figure 6 shows the formula of RMSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Figure 6

After we counted RMSE for each of the four models, we found that lasso regression has the lowest RMSE. However, in this case the coefficients of lasso regression are all 0, so it is overfitting for lasso regression. So we chose the one has the second lowest model to be the best model, which is Neural Network. The Cumulative RMSE plot for LR, Lasso, RF & NN is shown in Figure 7.

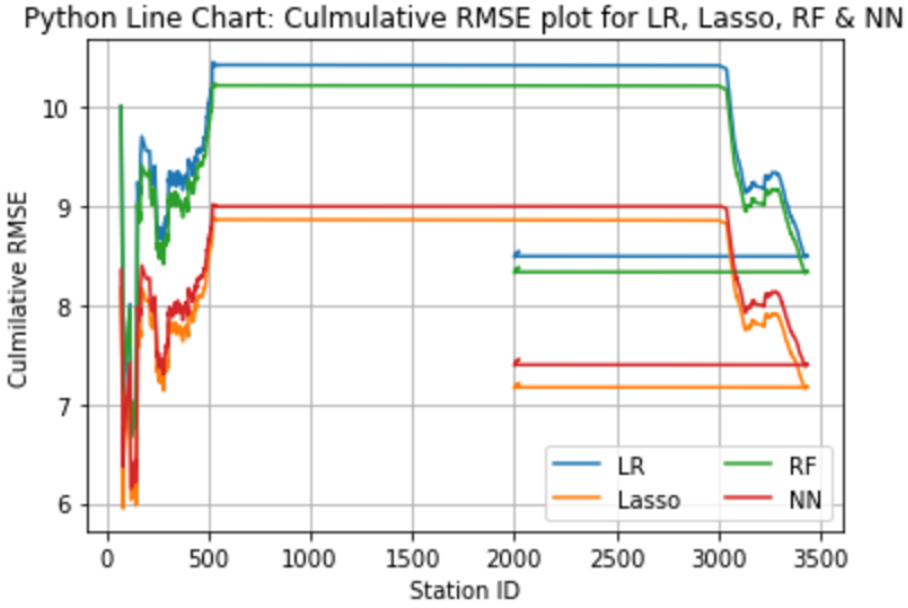


Figure 7

3. Prediction Using the Best Model

We used winter data to be the training data, and 11/2016 data to be the testing data, the model we used is the best model: Neural Network.

The prediction is saved as csv file for visualization. Top 20 rows are shown in Figure 8.

	A	B	C	D	E	F	G	H
1		station_id	bike_loss	docks	75%docks	left_bike	latitude	longitude
2	0	72	4	37	27	23	40.7672722	-73.993929
3	1	72	2	37	27	25	40.7672722	-73.993929
4	2	72	4	37	27	23	40.7672722	-73.993929
5	3	72	1	37	27	26	40.7672722	-73.993929
6	4	72	3	37	27	24	40.7672722	-73.993929
7	5	72	2	37	27	25	40.7672722	-73.993929
8	6	72	3	37	27	24	40.7672722	-73.993929
9	7	72	3	37	27	24	40.7672722	-73.993929
10	8	72	3	37	27	24	40.7672722	-73.993929
11	9	72	1	37	27	26	40.7672722	-73.993929
12	10	72	0	37	27	27	40.7672722	-73.993929
13	11	72	2	37	27	25	40.7672722	-73.993929
14	12	72	4	37	27	23	40.7672722	-73.993929
15	13	72	3	37	27	24	40.7672722	-73.993929
16	14	72	3	37	27	24	40.7672722	-73.993929
17	15	72	4	37	27	23	40.7672722	-73.993929
18	16	72	3	37	27	24	40.7672722	-73.993929
19	17	72	4	37	27	23	40.7672722	-73.993929
20	18	72	1	37	27	26	40.7672722	-73.993929

Figure 8

We saved csv file for training data as well. The visualization comparison of testing data and prediction result is presented below. Left plot is generated by real data, and right plot is by prediction.

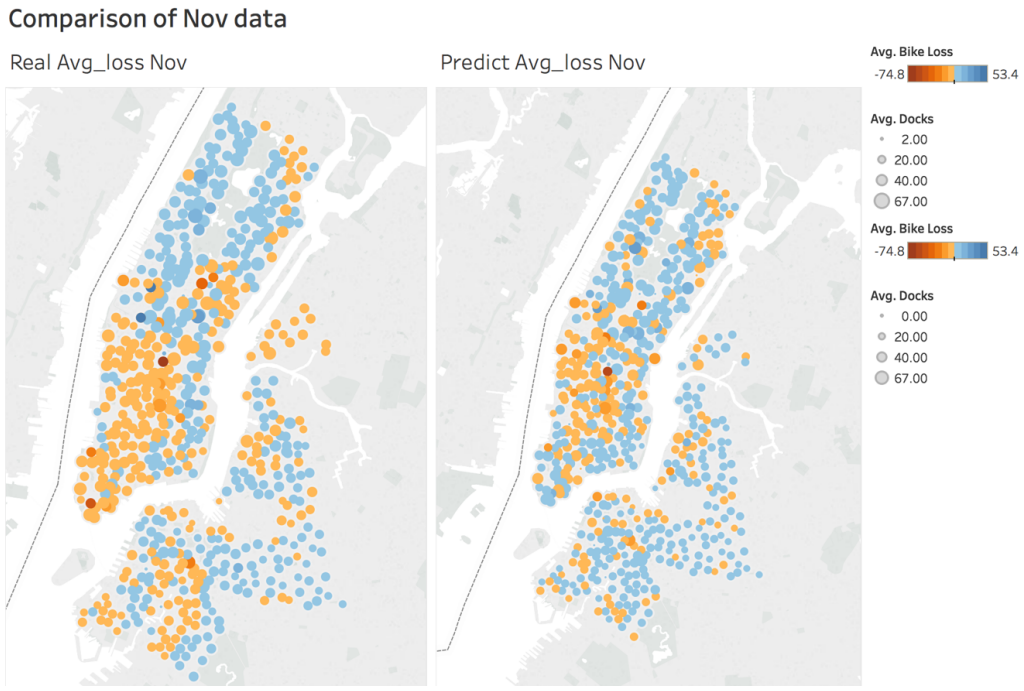


Figure 9

In this graph, each spot represents a station, the size of spot represents the No. of Docks of each station, and the color shows the average daily loss of bike of each station in Nov 2016. Yellow means shortage of bikes, and blue means redundancy.

We can see that prediction on most stations are close to the real states, while some spots get lighter color, which means our model is conservative. In general, our model can give us the trend of each station, and it can be more accurate on a given date.

Analysis

Our next step is to explore the bike_loss under different situations in a week. So we need to calculate the average of bike loss on weekdays and weekends in a week separately.

Also, we think not only large stations should be focused, attention should also be paid on small stations with high rate of bike shortage (because this would disappoint customers and drive them away too). So, instead of absolute value of bike loss, we use the quotient of bike loss over number of docks for each station.

Now compare the bike loss of weekends with weekdays in the graph below. The left plot is generated by weekdays data, and right plot is by weekends. The size of spot represents the No. of docks of each station, and the color shows the average daily bike loss divided by No. of docks of each station. Red means shortage of bikes, and blue means redundancy.

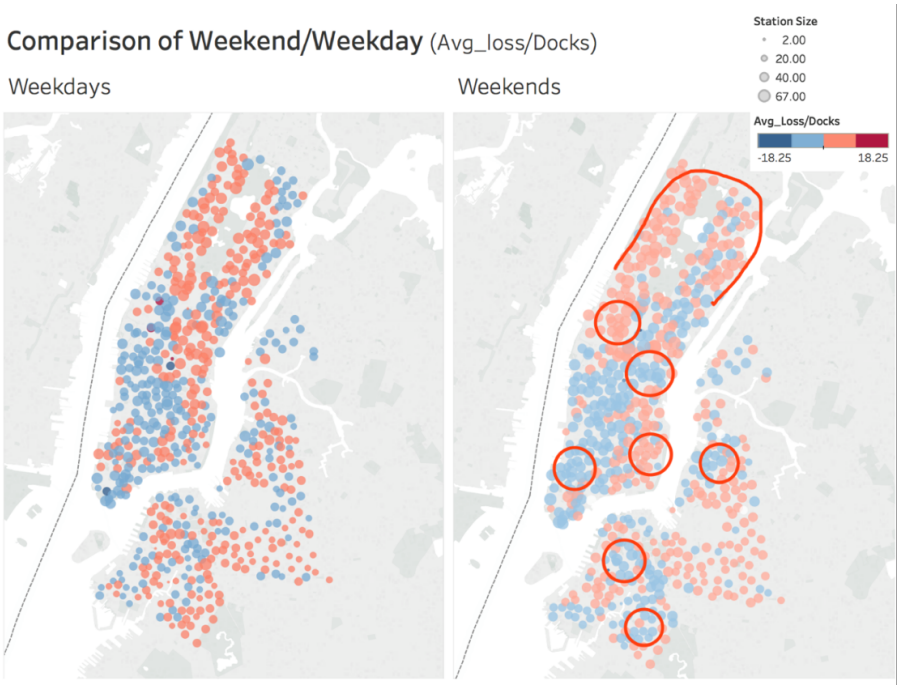


Figure 10

There are two main conclusions. First, most stations are busier in weekdays than in weekends. Second, there are oscillation of bike in the areas marked by red circles. Here are some representative examples:

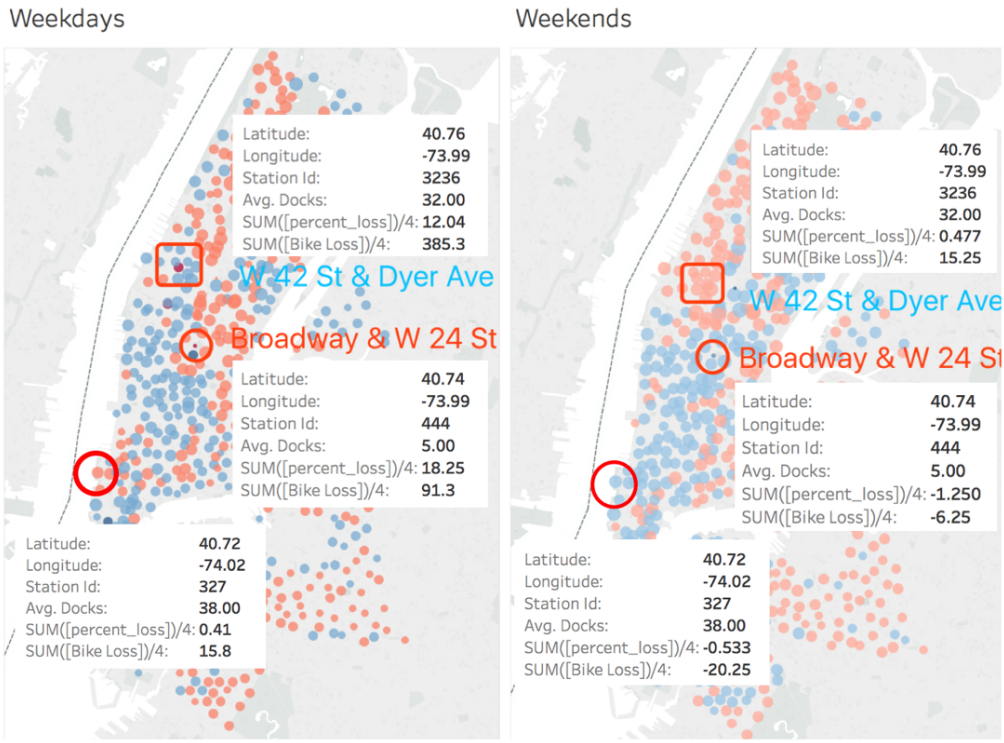


Figure 11

- Station 327: bikes flow out on weekdays but flow in on weekends.
- Station 444: a small station with large bike flows on weekdays, can take a breath on weekends.
- Station 3236: persistent losing of bikes results in consistent demand of bike supplement.

Since the trends of bike reserves of each station is periodically, we consider calculating the cumulative bike loss in a week instead of daily loss. Besides, we should pay more attention of busy stations like station 444, not only the large stations with many docks. Now in the graph presented below, the size of spot represents the bike_flow of each station, and the color shows the average weekly bike_loss divided by No. of docks of each station.

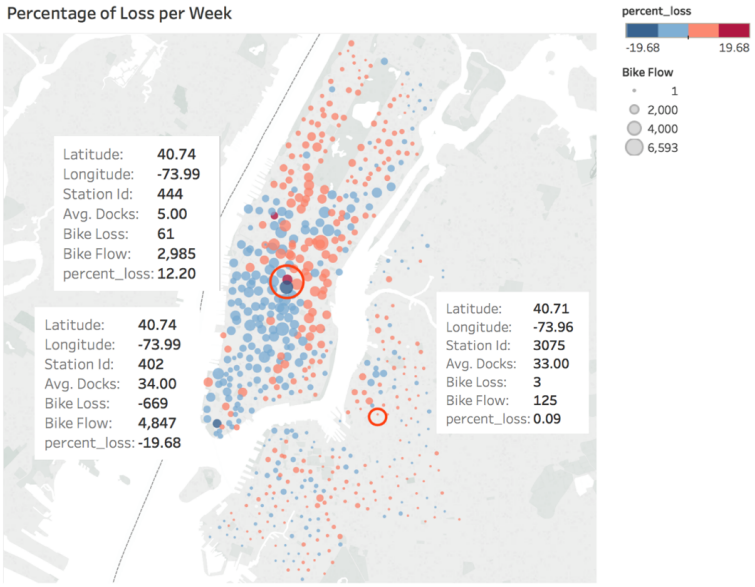
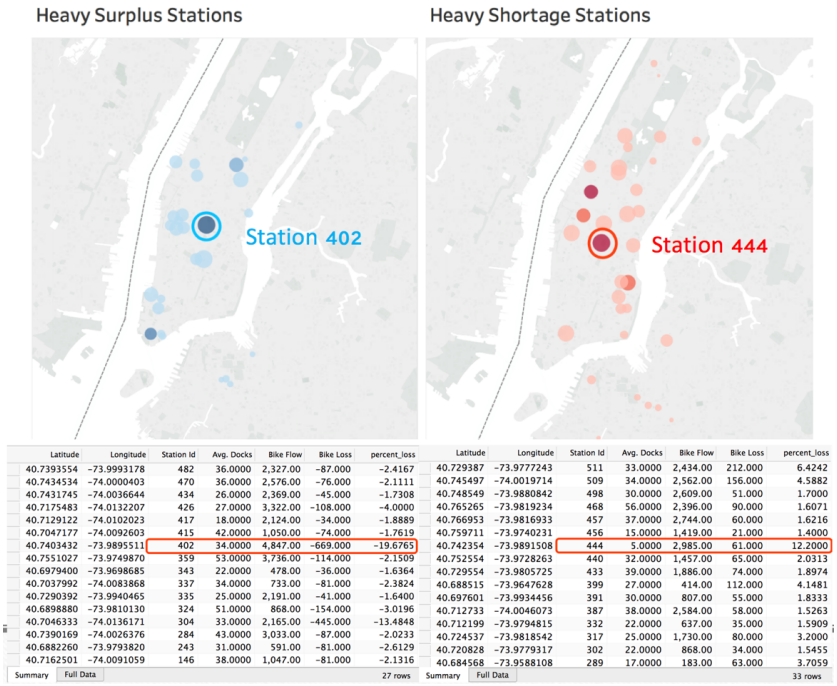


Figure 12

We can see that size of many stations become small, but stations locate on the central of the city are generally busier than others. And some stations in deep color before are now much lighter because of weekly self-balance. Therefore, now we can generate a list of stations which needs urgent bike supplement every week, and some of them might require daily supplement (whose color is deeper in the visualization below). Also stations with larger bike_flow get priority when we consider rebalance of bike in the system.



Latitude	Longitude	Station Id	Avg. Docks	Bike Flow	Bike Loss	percent_loss
40.7393554	-73.9993178	482	36.0000	2,327.00	-87.000	-2.4167
40.7434534	-74.0000403	470	36.0000	2,576.00	-76.000	-2.1111
40.7431745	-74.0036644	434	26.0000	2,369.00	-45.000	-1.7308
40.7175483	-74.0132207	426	27.0000	3,322.00	-108.000	-4.0000
40.7129122	-74.0102023	417	18.0000	2,124.00	-34.000	-1.8889
40.7047177	-74.0092603	415	42.0000	1,050.00	-74.000	-1.7619
40.7403432	-73.9895511	402	34.0000	4,847.00	-669.000	-19.6765
40.7551027	-73.9749870	359	53.0000	3,736.00	-114.000	-2.1509
40.6979400	-73.9698685	343	22.0000	478.00	-36.000	-1.6364
40.7037992	-74.0083868	337	34.0000	733.00	-81.000	-2.3824
40.7290392	-73.9940465	335	25.0000	2,191.00	-41.000	-1.6400
40.6898880	-73.9810130	324	51.0000	868.00	-154.000	-3.0196
40.7046333	-74.0136171	304	33.0000	2,165.00	-445.000	-13.4848
40.7390169	-74.0026376	284	43.0000	3,033.00	-87.000	-2.0233
40.6882260	-73.9793820	243	31.0000	591.00	-81.000	-2.6129
40.7162501	-74.0091059	146	38.0000	1,047.00	-81.000	-2.1316

Summary | Full Data — 27 rows

Latitude	Longitude	Station Id	Avg. Docks	Bike Flow	Bike Loss	percent_loss
40.729387	-73.9777243	511	33.0000	2,434.00	212.000	6.4242
40.745497	-74.0019714	509	34.0000	2,562.00	156.000	4.5882
40.748549	-73.9880842	498	30.0000	2,609.00	51.000	1.7000
40.765265	-73.9819234	468	56.0000	2,396.00	90.000	1.6071
40.766953	-73.9816933	457	37.0000	2,744.00	60.000	1.6216
40.759711	-73.9740231	456	15.0000	1,419.00	21.000	1.4000
40.742354	-73.9891508	444	5.0000	2,985.00	61.000	12.2000
40.752554	-73.9728263	440	32.0000	1,457.00	65.000	2.0313
40.729554	-73.9805725	433	39.0000	1,886.00	74.000	1.8974
40.688515	-73.9647628	399	27.0000	414.00	112.000	4.1481
40.697601	-73.9934456	391	30.0000	807.00	55.000	1.8333
40.712733	-74.0046073	387	38.0000	2,584.00	58.000	1.5263
40.712199	-73.9794815	332	22.0000	637.00	35.000	1.5909
40.724537	-73.9818542	317	25.0000	1,730.00	80.000	3.2000
40.720828	-73.9779317	302	22.0000	868.00	34.000	1.5455
40.684568	-73.9588108	289	17.0000	183.00	63.000	3.7059

Summary | Full Data — 33 rows

Figure 13

Conclusion and Next Step

The result of this study can be deployed to get an early warning of bike exhaustion. For a given day in following week, we can gather the information like weekdays, holidays and weather status and predict the demand of each station. So Citi bike can utilize it for preparation of bike supplement.

In the long term, this study can also be helpful as a suggestion of stations extension project, since we can find out those stations baring consistent shortage.

For next step, we can try to do optimization study for the systematic bike rebalance plan. Given weekday/weather/seasonal status, we can estimate shortage and redundancy of each station, and then find the best arrangement of rebalance by minimizing the total distance of bike transport.

References

[1] Junming Liu, Leilei Sun, Weiwei Chen, Hui Xiong. *Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization*. 2016.

[2] Vladimir Batagelj, Anuˇska Ferligoj. *Symbolic network analysis of bike sharing data*. 2016.