Prediction modeling for bicycle-sharing system

- A case study on NYC Citi Bike

Different names

Ola-Pedal: India

Ofo: China

Santander: London

Citi Bikes-NYC

https://techcrunch.com/2017/12/01/ola-pedal-bike-sharing-india/

Motivation

- **Bicycle-sharing system (BSS)** is really popular worldwide**.**
- But,
- How to **balance the number of bikes** at each station that can satisfy the demand while making full use of resources.

Demand
from people

Availability
in station

For any day d, any time t, the number of bikes under/over demand

$$T(d,t) = |\boldsymbol{Availability} - \boldsymbol{Demand}|$$

Goals

1. **Correlation Analysis** for the demand and availability data in each station
2. **Clustering** the stations that have the same changing pattern
3. **Demand prediction**: $D(d, t)$
4. **Daily availability prediction**: $A(t)$

Raw Data Structure

Real-Time Station Status

https://feeds.citibikenyc.com/stations/stations.json

Variable	Format
id	Number
stationName	String
availableDocks	Number
totalDocks	Number
latitude	Number
longitude	Number
statusValue	String
statusKey	Number
availableBikes	Number
stAddress1	String
lastCommunicationTime	Timestamp

{"executionTime":"2017-12-14 10:31:19 PM","stationBeanList":[{"id":72,"stationName":"W 52 St & 11
Ave","availableDocks":36,"totalDocks":39,"latitude":40.76727216,"longitude":-73.99392888,"statusValue":"In Service","statusKey":1,"availableBikes":2,"stAddress1":"W 52 St &
11 Ave","stAddress2":"","city":"","postalCode":"","location":"","altitude":"","testStation":false,"lastCommunicationTime":"2017-12-14 10:31:11 PM","landMark":""},
{"id":79,"stationName":"Franklin St & W Broadway","availableDocks":29,"totalDocks":33,"latitude":40.71911552,"longitude":-74.00666661,"statusValue":"In
Service","statusKey":1,"availableBikes":4,"stAddress1":"Franklin St & W
Broadway","stAddress2":"","city":"","postalCode":"","location":"","altitude":"","testStation":false,"lastCommunicationTime":"2017-12-14 10:29:48 PM","landMark":""},
{"id":82,"stationName":"St James Pl & Pearl St","availableDocks":20,"totalDocks":27,"latitude":40.71117416,"longitude":-74.00016545,"statusValue":"In
Service","statusKey":1,"availableBikes":7,"stAddress1":"St James Pl & Pearl

Raw Data Structure

- **Historical trip data** https://s3.amazonaws.com/tripdata/index.html

Variable	Format
Trip Duration	In hour, minute and second format
Start Time	Timestamp
Stop Time	Timestamp
Start Station ID	Number
Start Station Name	String
Start Station Latitude	Number
Start Station Longitude	Number
End Station ID	Number
End Station Name	String
End Station Latitude	Number
End Station Longitude	Number
Bike ID	Number
User Type	Casual or Registered
Birth Year	Number
Gender	String

	Trip Duration	Start Time	Stop Time	Start Station ID	Start Station Name	Start Station Latitude	Start Station Longitude	End Station ID	End Station Name	End Station Latitude	End Station Longitude	Bike ID	User Type	Birth Year	Gender
0	704	7/1/2016 0:00	7/1/2016 0:11	459	W 20 St & 11 Ave	40.746745	-74.007756	347	Greenwich St & W Houston St	40.728846	-74.008591	17431	Customer	NaN	0
1	492	7/1/2016 0:00	7/1/2016 0:08	293	Lafayette St & E 8 St	40.730287	-73.990765	466	W 25 St & 6 Ave	40.743954	-73.991449	24159	Subscriber	1984.0	1
2	191	7/1/2016 0:00	7/1/2016 0:03	3090	N 8 St & Driggs Ave	40.717746	-73.956001	3107	Bedford Ave & Nassau Ave	40.723117	-73.952123	16345	Subscriber	1986.0	2
3	687	7/1/2016 0:00	7/1/2016 0:11	459	W 20 St & 11 Ave	40.746745	-74.007756	347	Greenwich St & W Houston St	40.728846	-74.008591	25210	Customer	NaN	0
4	609	7/1/2016 0:00	7/1/2016 0:10	284	Greenwich Ave & 8 Ave	40.739017	-74.002638	212	W 16 St & The High Line	40.743349	-74.006818	15514	Customer	NaN	0

Final Data Structure

Variate	Format
Station_id	Number
Date	String
Hour	Number
Minute	Number
Avail_bikes	Number
Avail_docks	Number
Tot_docks	Number
In_service	String
Status_key	Number

Station Status

Variable	Format
Station ID	String
Date	String
Hour	Number
Weekday	Number
Dock_demand/Bike_demand	Number

Demand data

Data Preparation

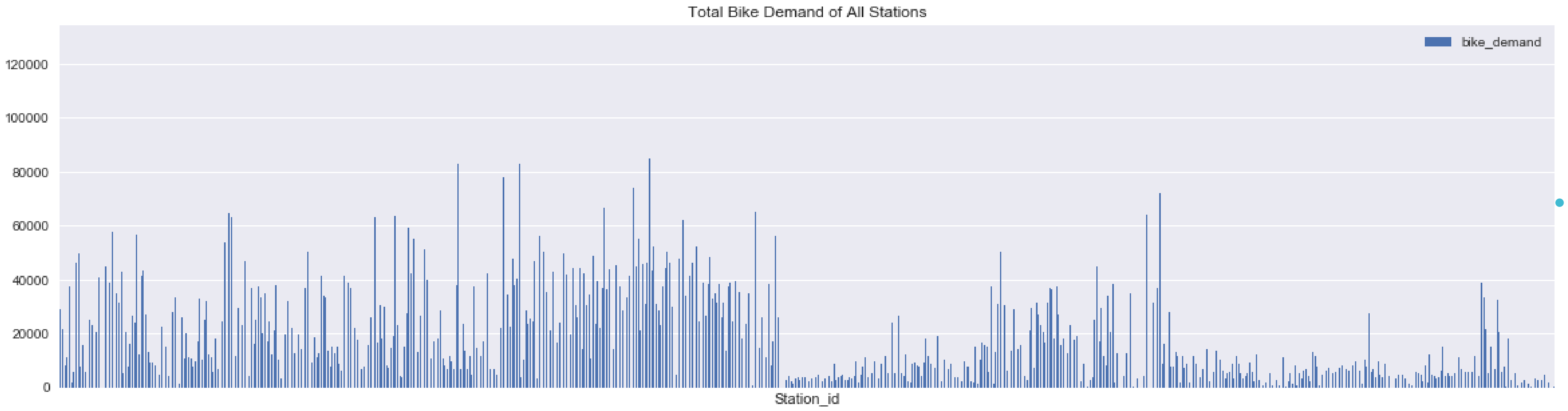
- **Station Status**

	dock_id	date	hour	minute	avail_bikes	avail_docks	tot_docks	in_service	status_key
0	72	16-07-01	1	0	19	19	39	1	1
1	72	16-07-01	1	29	19	19	39	1	1
2	72	16-07-01	1	59	19	19	39	1	1
3	72	16-07-01	2	29	19	19	39	1	1
4	72	16-07-01	2	58	20	18	39	1	1
5	72	16-07-01	3	33	21	17	39	1	1
6	72	16-07-01	4	2	22	16	39	1	1
7	72	16-07-01	4	31	22	16	39	1	1
8	72	16-07-01	5	1	22	16	39	1	1
9	72	16-07-01	5	31	22	16	39	1	1

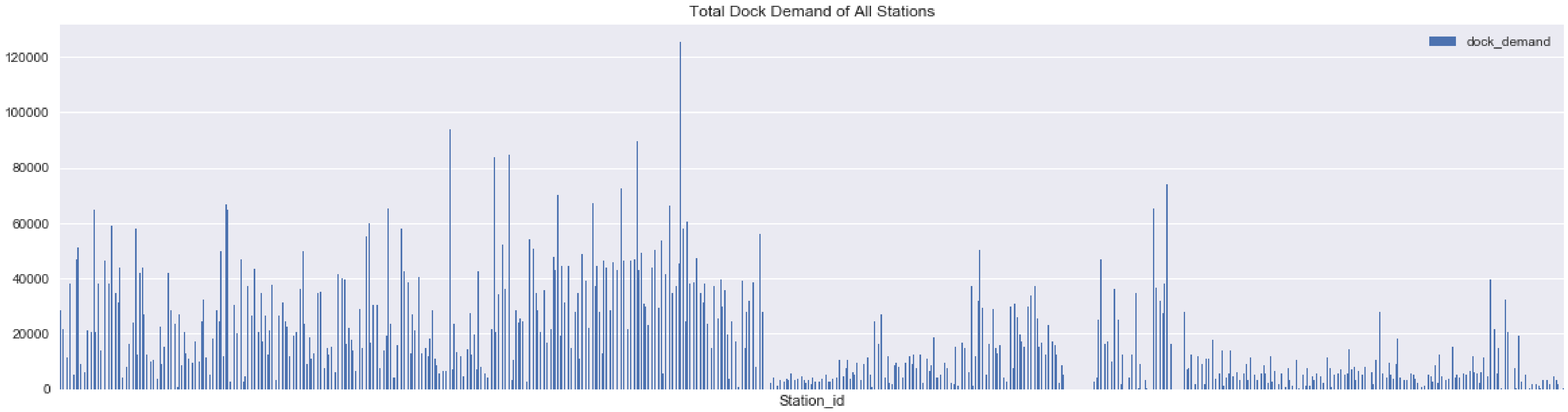
- **Demand data**

			minute
Start Station ID	**date**	**hour**	
72	**2016-07-01**	**0**	3
		6	4
		7	4
		8	19
		9	11
		10	3
		11	15
		12	1
		13	3
		14	5
		15	6
		16	6
		17	2
		18	2
		19	3
		20	4
		23	3

Visualization

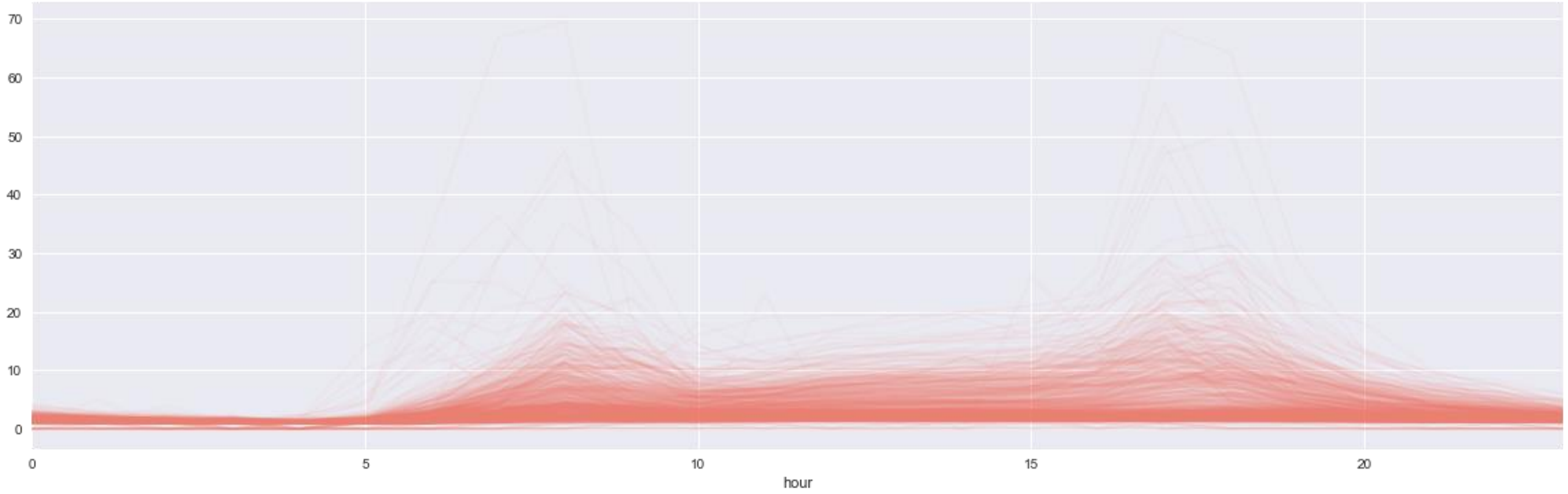


- Bike Demand

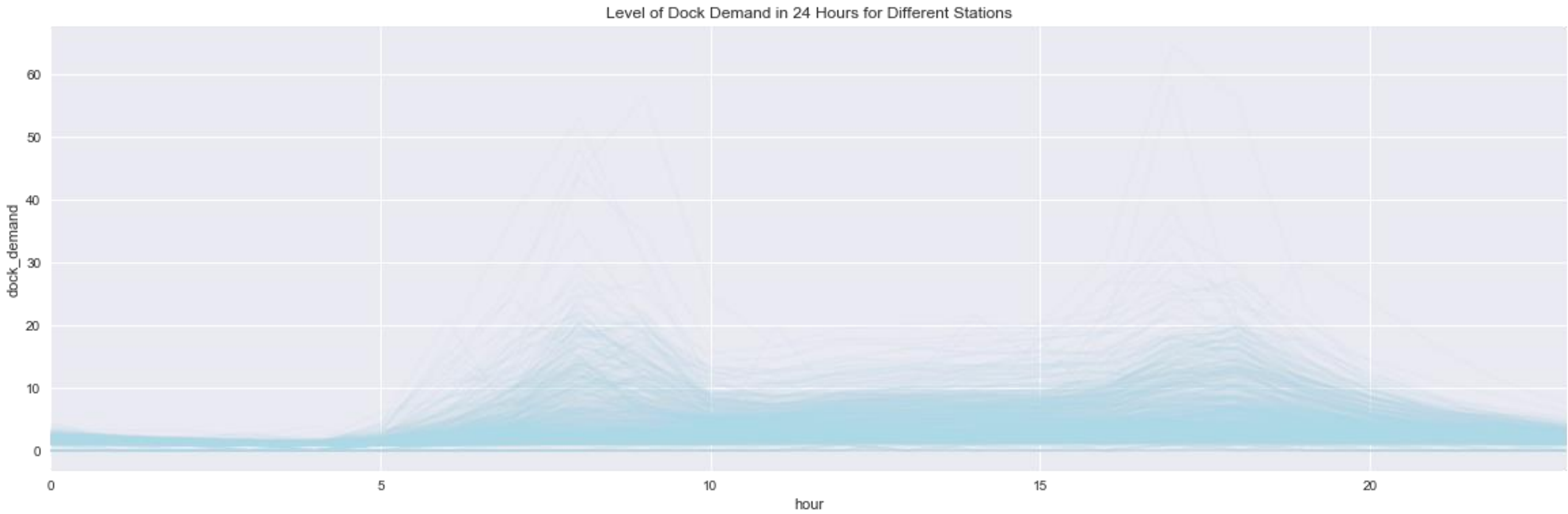


- Dock Demand

Visualization

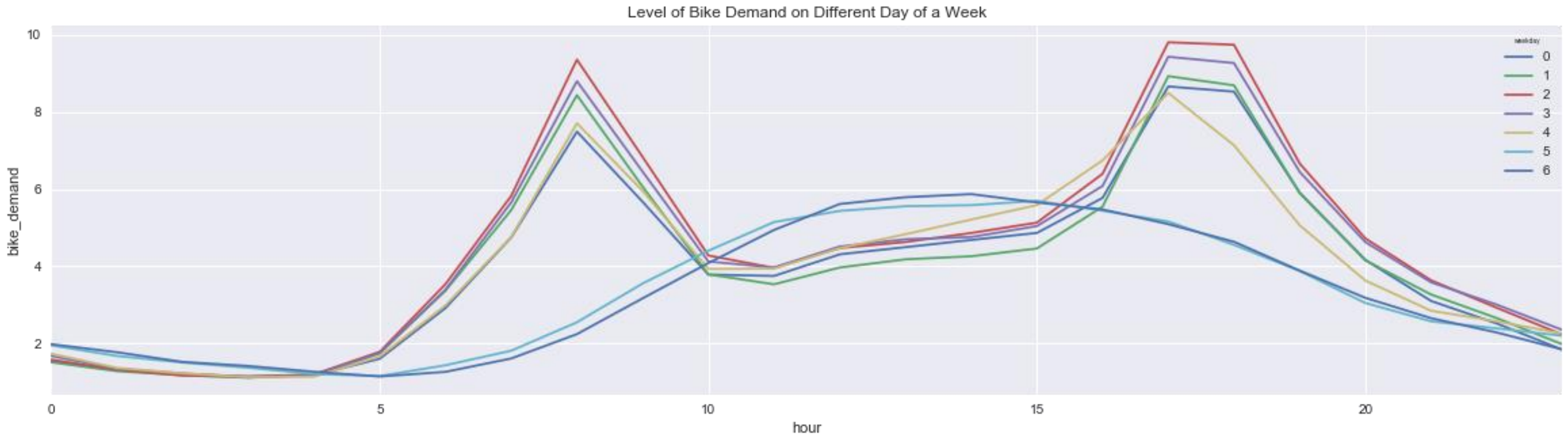


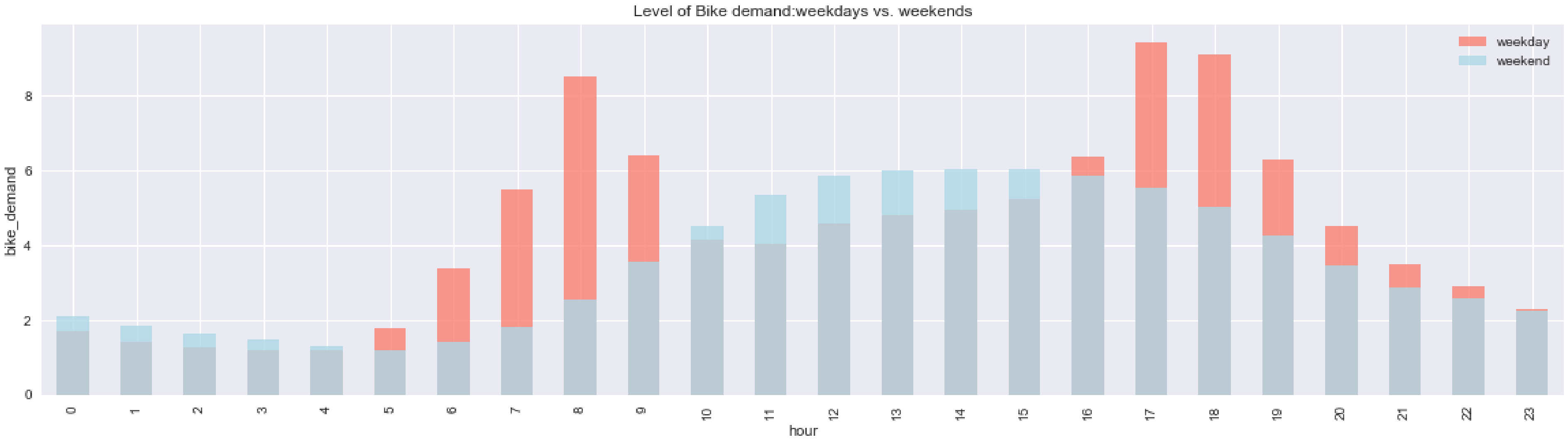
Demand of bikes
in each station



Demand of docks
in each station

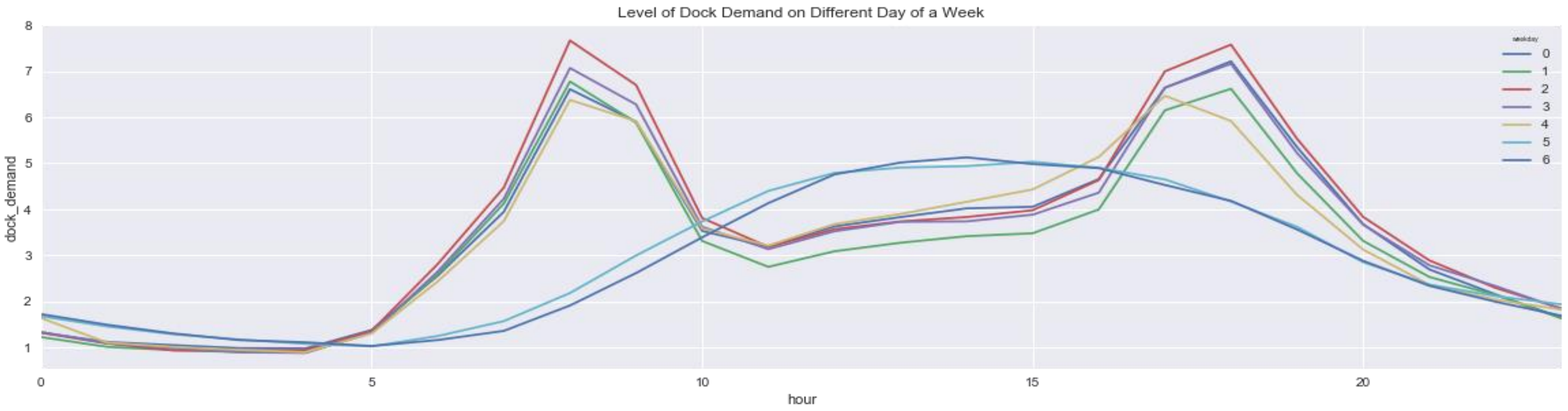
Visualization

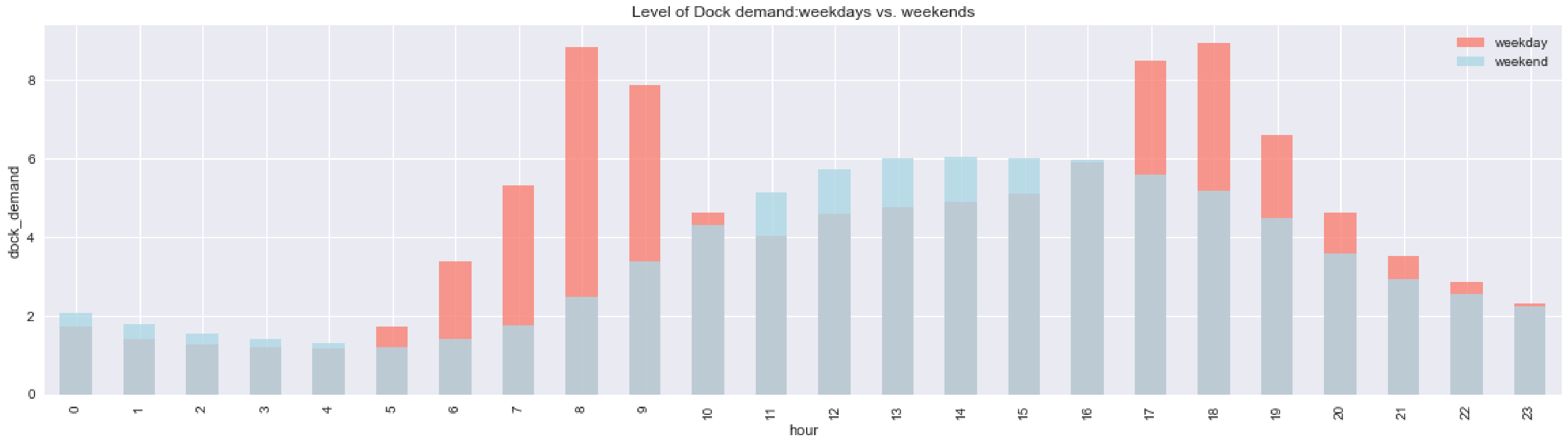




- **Bike Demand**

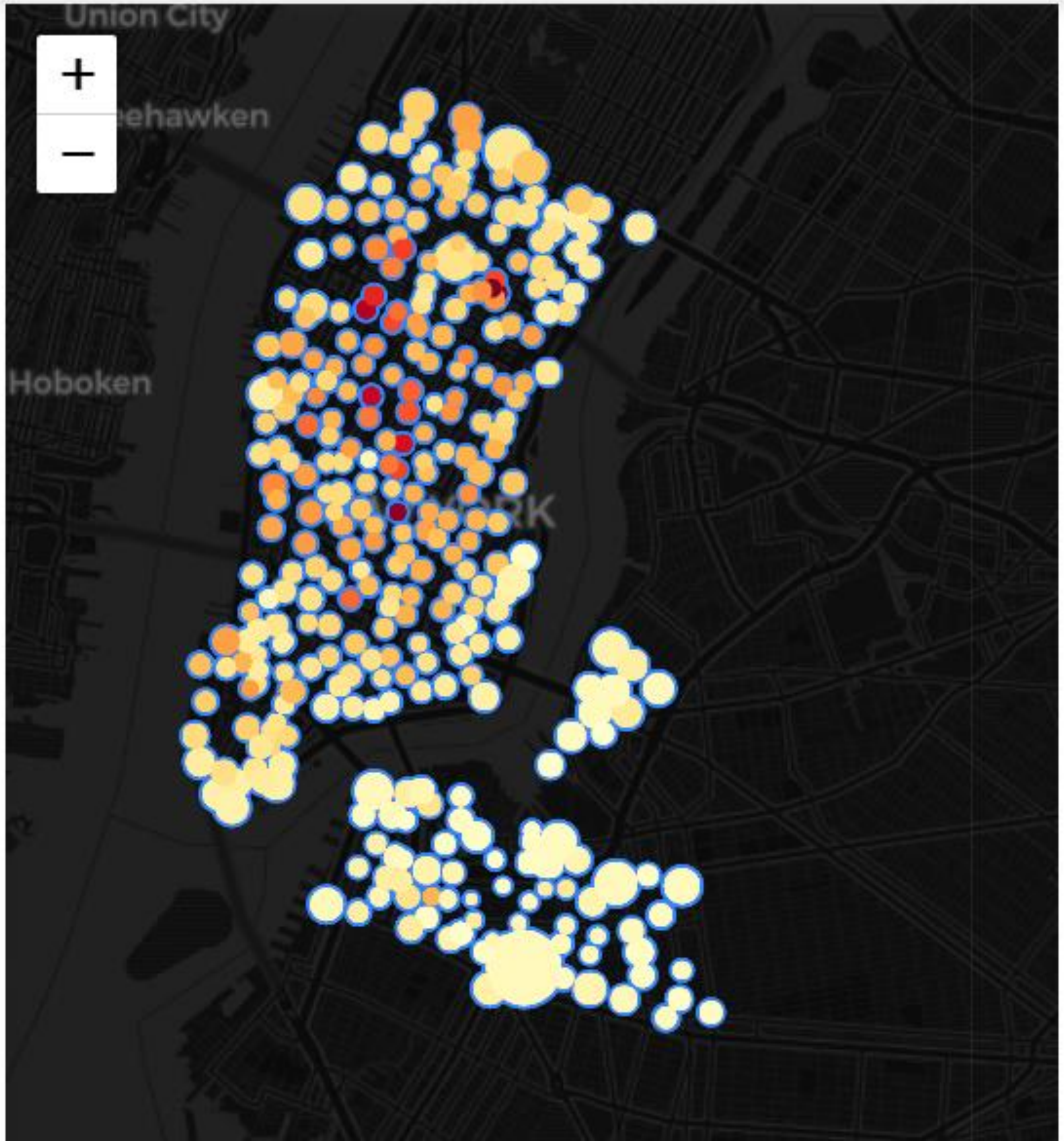
Visualization



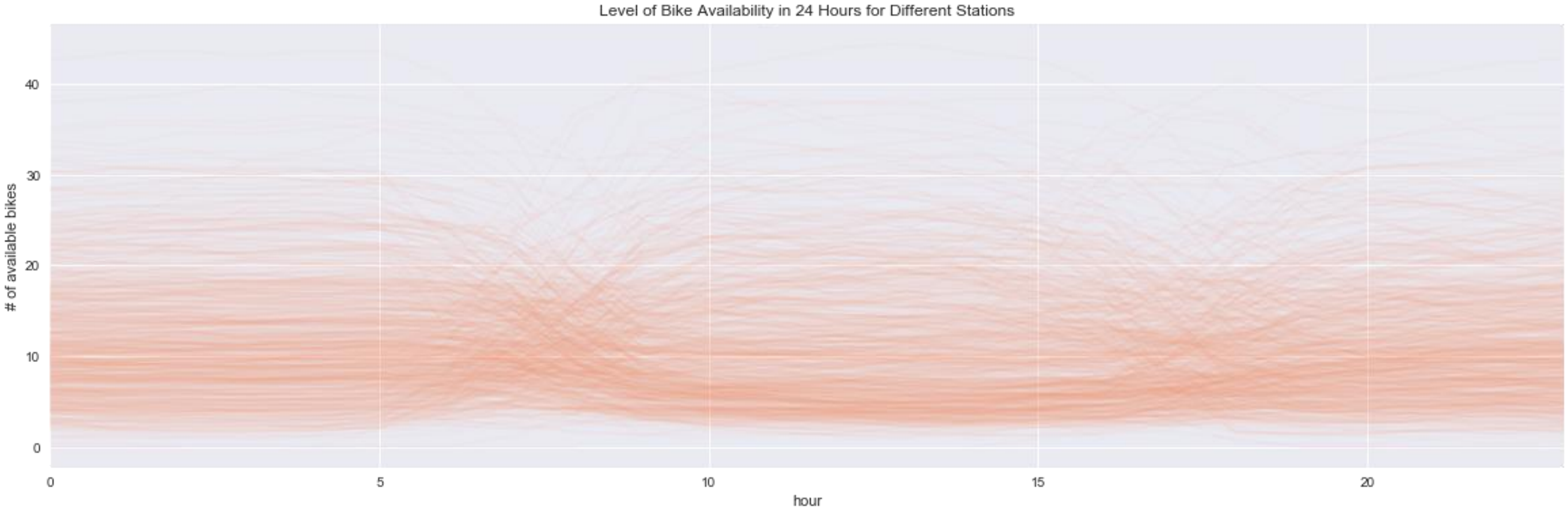


- **Dock Demand**

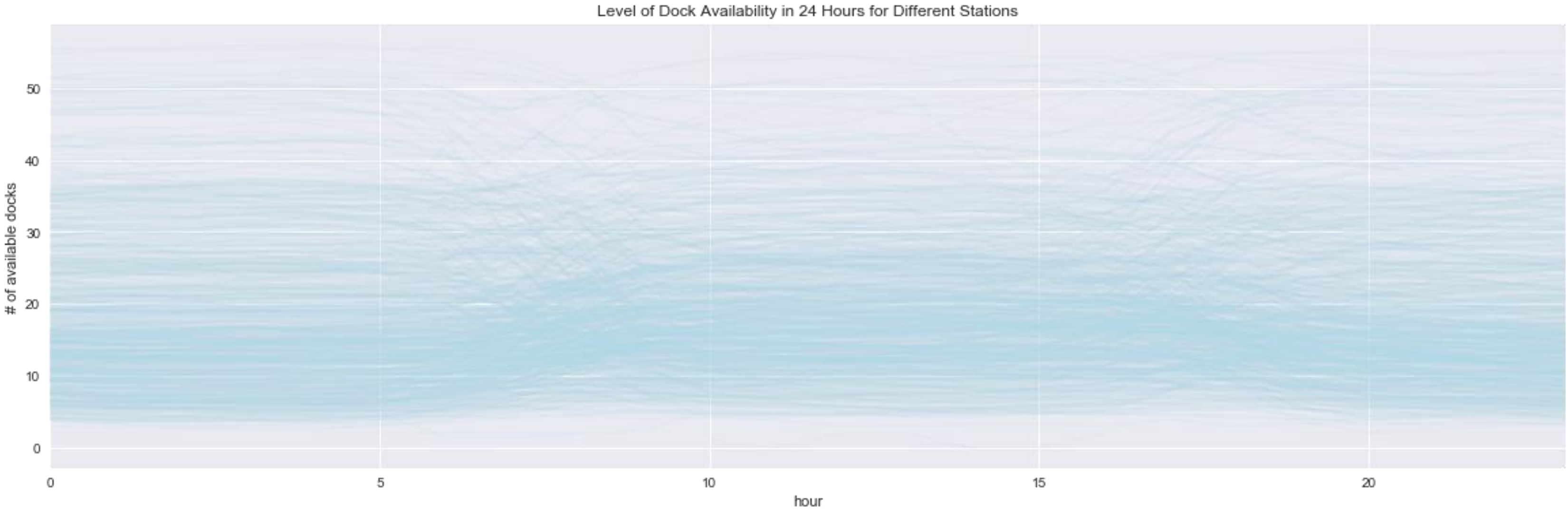
Out[65]:



Visualization



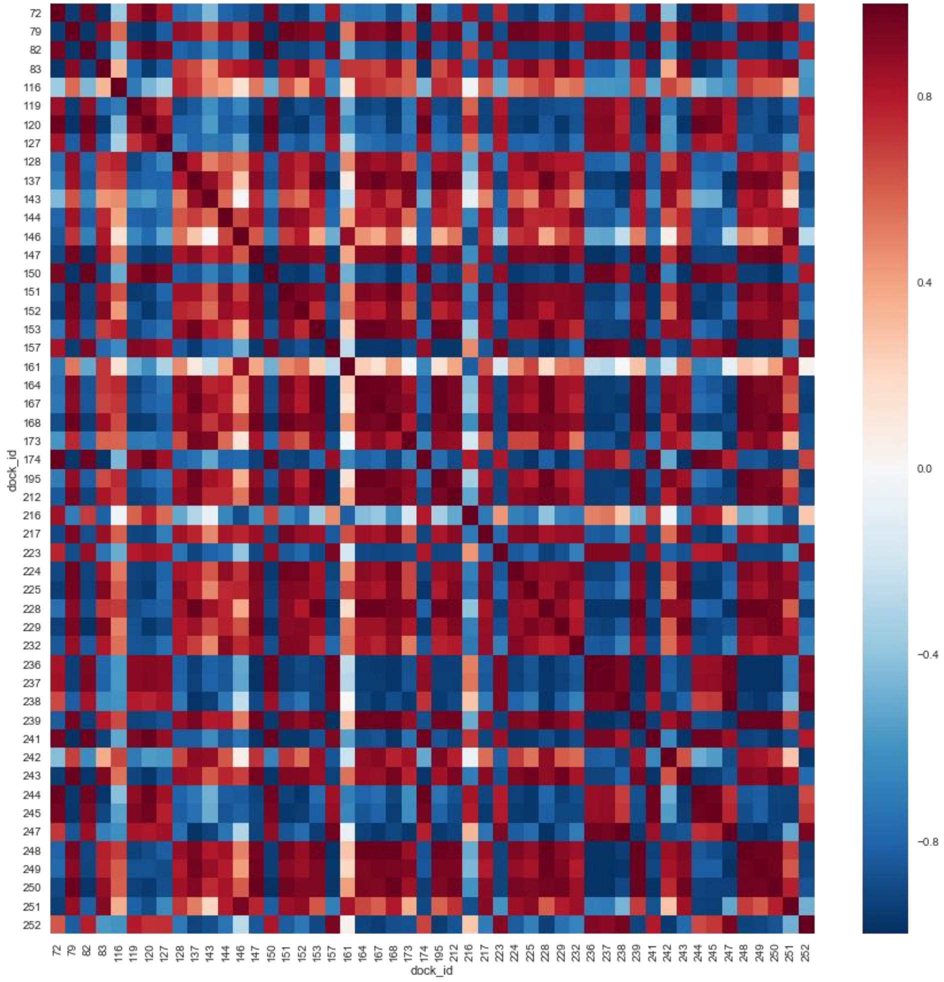
Available bikes
in each station



Available docks
in each station

Correlation Analysis **-Available bikes** in each station

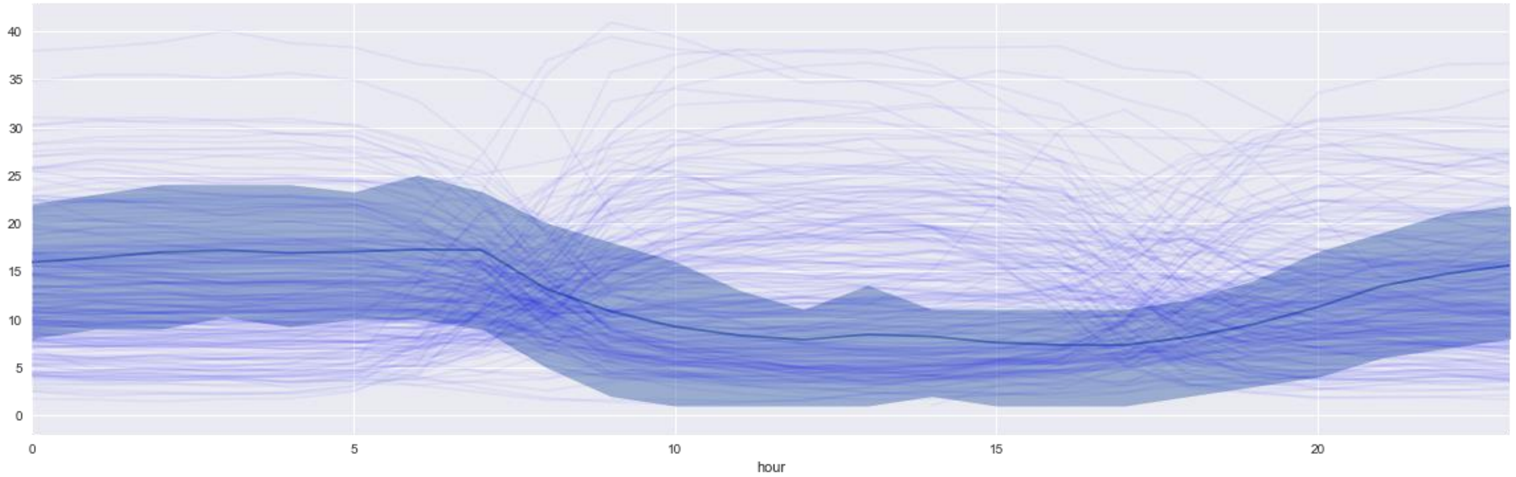
For each station, find all stations that have high correlation with it. (correlation coefficient >=0.9)



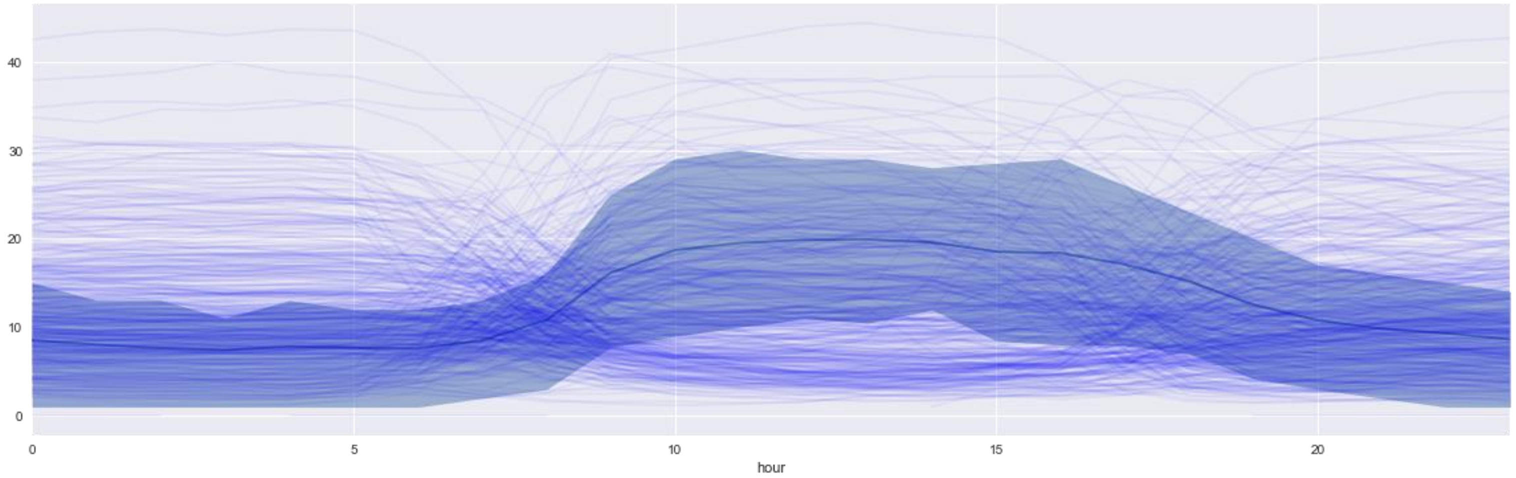

```
{72: Int64Index([  72,   79,   82,   83,  120,  144,  150,  152,  174,  216,
            ...
             3411, 3412, 3413, 3421, 3422, 3423, 3430, 3445, 3449, 3454],
           dtype='int64', name='dock_id', length=226),
 79: Int64Index([  72,   79,   82,   83,  119,  120,  144,  147,  150,  151,
            ...
             3430, 3434, 3438, 3440, 3445, 3449, 3452, 3454, 3461, 3462],
           dtype='int64', name='dock_id', length=416),
 82: Int64Index([  72,   79,   82,   83,  120,  127,  144,  147,  150,  152,
            ...
             3423, 3424, 3427, 3430, 3434, 3440, 3445, 3449, 3454, 3461],
           dtype='int64', name='dock_id', length=361),
 83: Int64Index([  72,   79,   82,   83,  120,  144,  150,  152,  174,  216,
            ...
             3411, 3412, 3413, 3421, 3422, 3423, 3430, 3445, 3449, 3454],
           dtype='int64', name='dock_id', length=235),
```

Correlation Analysis **-Available bikes** in each station

For a certain station, plot the availability of bikes for all the stations that are identified having high correlation with it.
Example below: stations 72 and 79



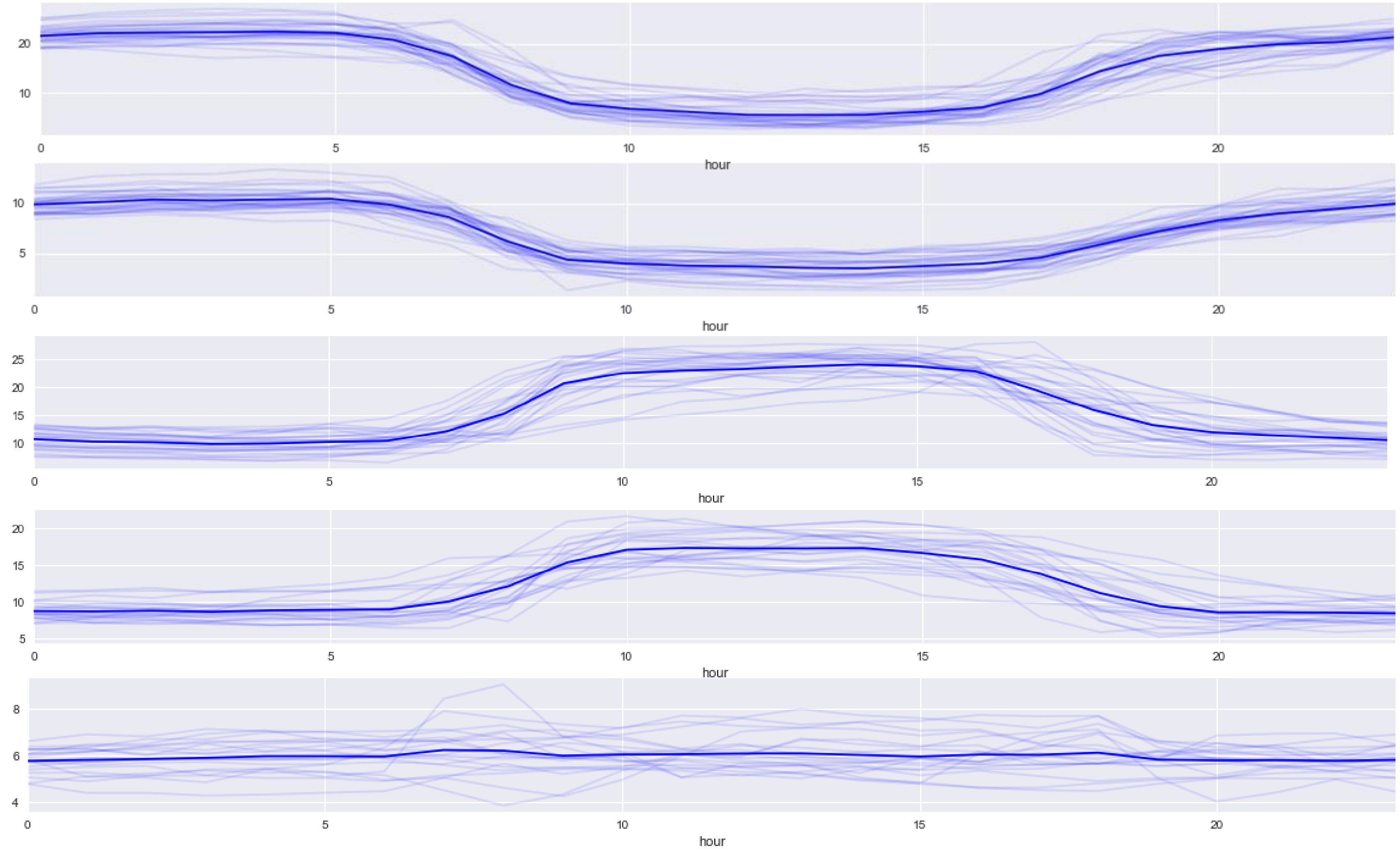
Station #72



Station #79

Clustering -**Available bikes** in each station

K-means Clustering with Dynamic Time Warping (DTW)



Dynamic time warping finds the optimal non-linear alignment between two time series.
The Euclidean distances between alignments are then much less susceptible to pessimistic similarity measurements due to distortion in the time axis.
There is a price to pay for this, however, because dynamic time warping is quadratic in the length of the time series used

Prediction **-Available bikes** in each station

Index	X			y
date	Hour	A(t-1)	A(t-2)	A(t)
		Available bikes in the given station at time (t-1) and (t-2)		

Basic: Target station

Index	X							y
date	Hour	A(t-1)	A(t-2)	A1(t-1)	A2(t-1)	……	An(t-1)	A(t)
				Available bikes in the correlated stations at time (t-1)				

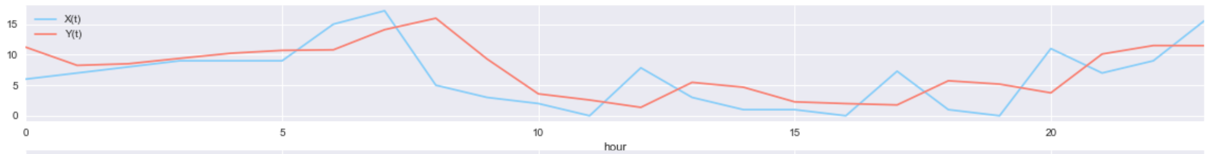
Basic+corr:
Target station
+Correlated Stations

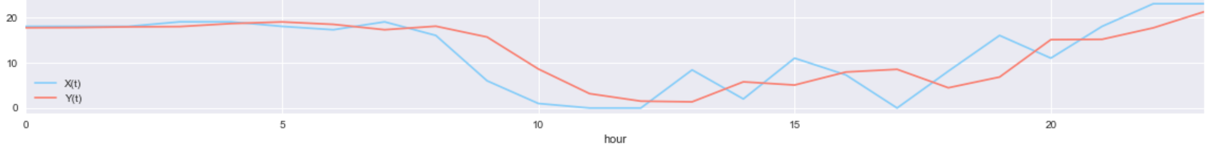
Index	X							y
date	Hour	A(t-1)	A(t-2)	B1(t-1)	B2(t-1)	……	Bn(t-1)	A(t)
				Available bikes in the stations that are in the same cluster at time (t-1)				

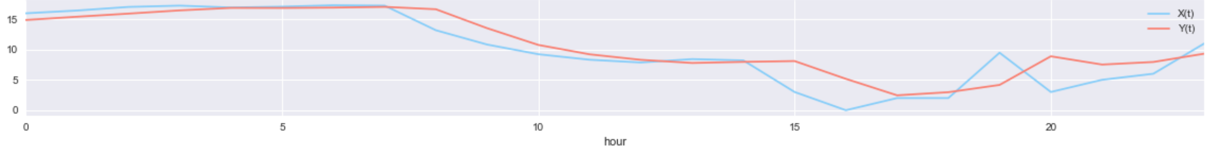
Basic+cluster:
Target station
+Clustered Stations

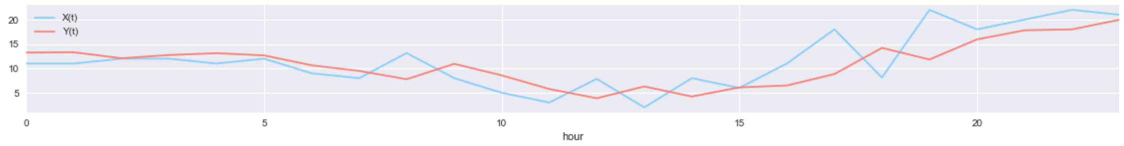
Prediction Result **-Available bikes**

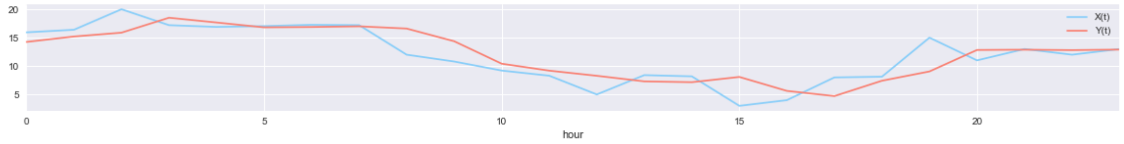
		Linear Regression	SVR	Decision Tree Regressor	Random Forest Regressor	Neural Network
Basic	MSE	18.45	19.58	20.96	16.73	16.59
	R-square	0.5468	0.5191	0.4852	0.5892	0.5923
Basic+corr:	MSE	17.77	18.90	27.00	15.67	18.25
	R-square	0.5634	0.5357	0.3368	0.6152	0.5517
Basic+cluster	MSE	17.40	18.40	27.39	17.85	19.69
	R-square	0.5724	0.5480	0.3274	0.5615	0.5165

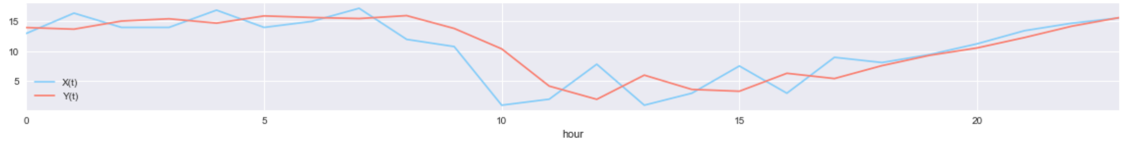












Prediction Result –Demand

- **Bike Demand**

	Linear Regression	SVR	Decision Tree Regressor	Random Forest Regressor	Neural Network
R-square	0.0300	-0.0720	0.0399	0.9352	0.2053
MSE	90135.8693	34333.0470	31729.5360	2076.1104	25451.6440

- **Dock Demand**

	Linear Regression	SVR	Decision Tree Regressor	Random Forest Regressor	Neural Network
R-square	0.0309	-0.0784	0.0414	0.9489	0.2196
MSE	80647.5477	35425.9728	31392.8499	1677.8532	25637.6504

Questions?