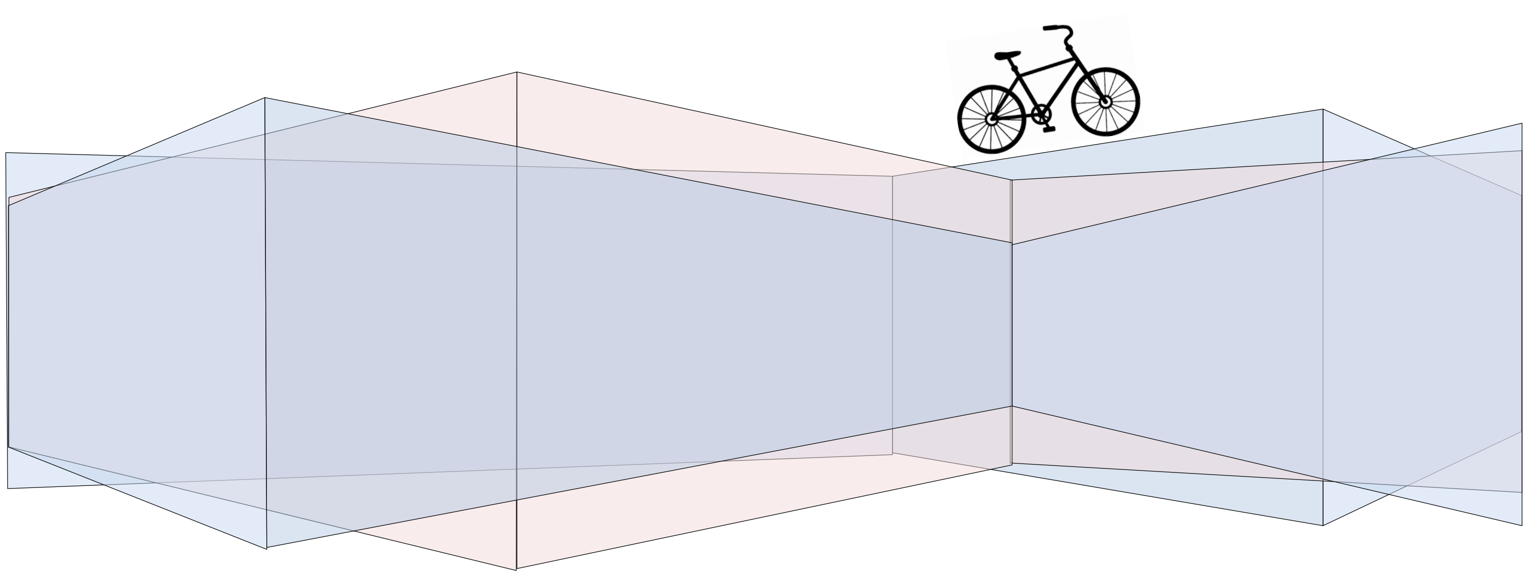
Forecasting Bike Availability

A study on bike sharing in Washington

David Albrecht



Capital Bikeshare and Bikesharing

In many cities across the world you might find bicycles for public, daily use. Specifically, in the United States there are at least 119 bikesharing systems that, in total, share about 4,800 docking stations as of January 2017. These numbers are incredibly impressive considering that it has been less than 10 years since the first, large-scale bikesharing system was created in the United States.[1] The 3rd largest bikesharing platform in the nation is operated by Capital Bikeshare in Washington, and is the subject of the following analysis. Replacing SmartBike DC in September 2010, Capital Bikeshare is a public taxpayer-supported program of both the District of Columbia and Arlington County.[2]

Bikesharing, in general, is the practice of placing docks of bikes or "stations" in ideal locations across cities such that anyone can pay to use the bikes to travel to and from any destination. With Capital Bikeshare's approximately 3,700 bikes and 440 stations across 5 jurisdictions, there is a lot of biking opportunity for residents or visitors of the Washington area. Capital Bikeshare makes it easy and affordable for anyone to pick up a bike and ride with a single $2 trip under 30 minutes, an $8 24-hour pass for trips under 30 minutes, or an $85 annual membership that allows for unlimited 30-minute rides. Going the extra mile, the company even supports a mobile application with up-to-the-minute system information. Popular bikesharing platforms are not always easy to maintain though, and that is exactly what we explore in the subsequent paragraphs.

The Issue with Bikeshare Platforms

In our infinitely expanding world, it's hard to imagine a scenario in which something you need simply *runs out* and there's nothing you can easily do about it. Allow me to explain just that scenario: Imagine you live in the bustling capital of the United States and you work in downtown. You get up every morning at 6am, hit the treadmill for half an hour, clean up, eat some breakfast, and are quickly on your way to the office that, conveniently, is about a mile away. Everything goes pretty smoothly for the most part on most days. Then, seemingly out of the blue, a public bike transit system has taken roots in the city. There are bike stations all over the place, and even one outside of your apartment! Cool. You decide to start off the week by purchasing a daily pass and

[1] https://ggwash.org/view/62137/all-119-us-bikeshare-systems-ranked-by-size
[2] https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems

planning to bike to work the next day. It's Monday: You start your morning routine and accidentally spill some coffee on your favorite white shirt. Thanks, Monday. You get out the door a few minutes later than usual and hustle on down to the bike station… But there's something wrong… There are no bikes. Outraged, you vow to never use the bikeshare service again. "What caused this mess?!" you might ask yourself. Turns out, there *is* a good explanation.

Bikeshare platforms suffer from cyclical travel patterns. We, as humans who need to get places at specific times during the working week, tend to use the same travel routes at the same times on the same days. It's the reason freeways get crowded and why 9am and 5pm are affectionately known as "traffic hour". It's the same reason why bike stations in popular locations tend to fill up at certain times (and prohibit users from ending a session) and empty out at other times, and why you were late to work on the Monday to end all Mondays. There's a silver lining to all of this though: It's a problem that the bikesharing company might be able to solve if it so chooses. Known as rebalancing, it is the practice of moving bikes around to different stations to keep them from being too empty or too full for too long. Optimizing this practice, I aimed to forecast the percent of total capacity of bike stations so that bikes can be rebalanced in a predictive fashion.

Data and Description of Features

I used three different datasets that describe bikeshare availability for the Capital Bikeshare platform in the Washington area:

1. Historical Trip Patterns[3]
2. Live XML Feed[4]
3. Historical System Outage[5]

Historical Trip Patterns

This data by Capital Bikeshare shows trips taken from all stations from the start of 2015 Q3 to the end of 2016 Q3. It follows a transaction format in that each instance shows the location and time for both the start and ending stations of any particular trip. Specifically, it contains the following features:

1. The duration of the trip in milliseconds.

[3] https://www.capitalbikeshare.com/system-data
[4] https://feeds.capitalbikeshare.com/stations/stations.xml
[5] http://www.cabitracker.com/outage_history.php

2. The start date and time in the datetime format: m/d/y h:m
3. The end date and time in the datetime format: m/d/y h:m
4. The starting station name and unique ID number.
5. The ending station name and unique ID number.
6. The unique ID number of the bike used for the trip.
7. Whether the user was a "registered" member (Annual Member, 30-Day Member or Day Key Member) or a "casual" rider (Single Trip, 24-Hour Pass, or 5-Day Pass).

It is important to note that this dataset comes to the public in a scrubbed format. Capital Bikeshare states: “This data has been processed to remove trips that are taken by staff as they service and inspect the system, trips that are taken to/from any of our “test” stations at our warehouses and any trips lasting less than 60 seconds (potentially false starts or users trying to re-dock a bike to ensure it's secure).” This dataset does not contain every single trip taken over this time period, but the hope is that it is fairly close and that it still shows true trends in bike usage.

Live XML Feed

This data, also by Capital Bikeshare, gives a snapshot view of all stations in operation, but only at the present. From the feed, I used the following features:

1. The unique ID number of the station.
2. The latitude coordinate of the station.
3. The longitude coordinate of the station.
4. The current number of bikes at the station.
5. The current number of empty docks at the station.

I used the latitude and longitude features to help visualize station popularity and added the current number of bikes at the station to the current number of empty docks at the station to obtain station capacities.

The above datasets tell us where bikes are going and station capacity information, but it is still difficult to accurately back out station percent capacities down the line because there is no way to know how many bikes are at any station at the beginning of the dataset in 2015 Q3. To help estimate how many bikes are at any station at the first time interval, I combined the above data with CaBi Tracker’s data below.

Historical System Outage

This data by CaBi Tracker leverages Capital Bikeshare's live data feed. It contains full/empty notifications for every station over any time period. As we will match this data with the beginning of Capital Bikeshare's data, I chose data from the beginning of the 2015 Q3 period. This data, unlike the first set, has not been scrubbed (but may be based on scrubbed data depending on the state of the Live XML Feed). It contains the following features:

1. The unique ID number of the station.
2. The name of the station.
3. Whether the station is full or empty.
4. The start time of the full/empty state in datetime format: y/m/d h:m:s
5. The end time of the full/empty state in datetime format: y/m/d h:m:s
6. The duration of the full/empty state in minutes.

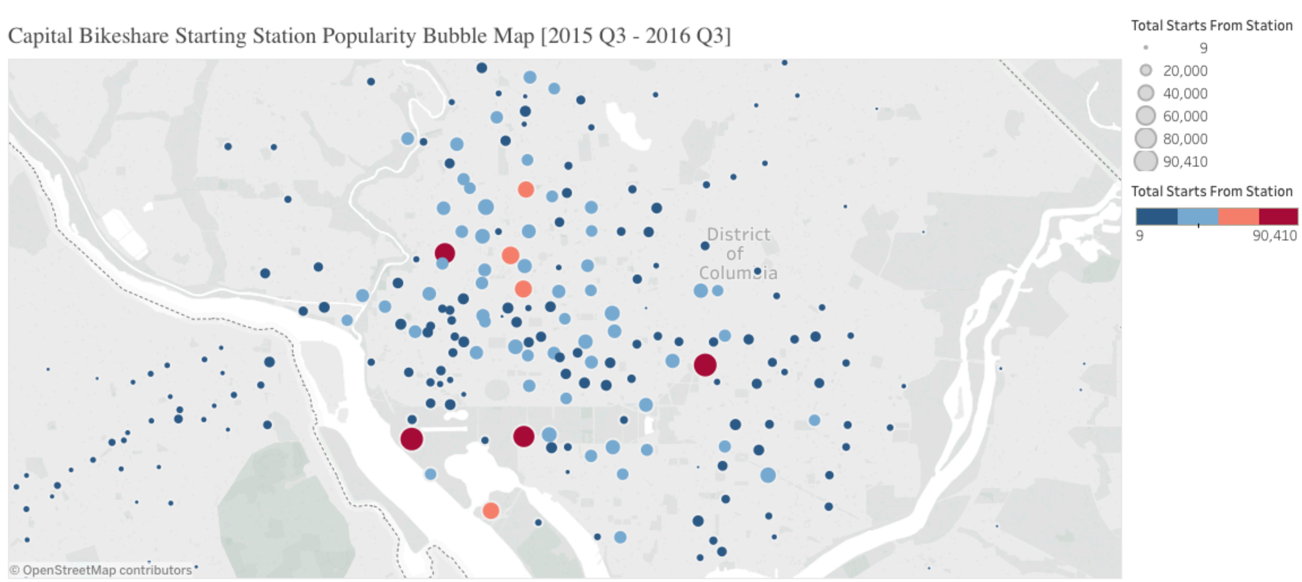
The idea with combining these three datasets is that if we know 1) where trips are being taken to and from at any time (Capital Bikeshare's Historical Trip Patterns data), 2) station capacity (Capital Bikeshare's Live XML Feed data), and 3) the beginning state of any station (CaBi Trackers's Historical System Outage data), then we can back out the percent of total capacity for any station at any given time. Now that we have an idea of what data we want to use, let's move on to exploration.

Exploratory Data Analysis and Transformations

There were many questions asked of the data, and, with questions, came answers and then, inevitably, more questions. Most exploration was done during the combinations and transformations of the above datasets. Let's examine the key findings and nuances while exploring the data transformation process.

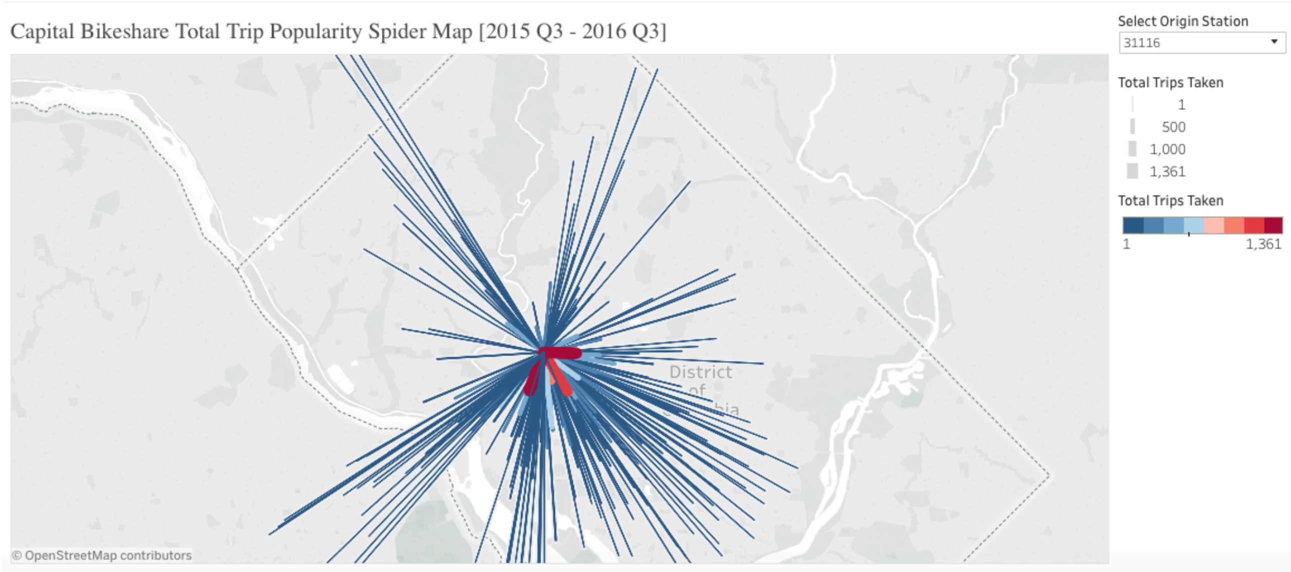
We begin with the Historical Trip Patterns (HTP) dataset and join the Live XML Feed (LXF) dataset to it using station ID features as a key between datasets. *This is the first step toward creating our target variable, percent capacity, and allows us to map station ID numbers to station capacity.* In addition, we are able to map station ID numbers to latitude and longitude coordinates. Taking this information, we find that a certain station, ID #31709 located on 34th St & Minnesota Ave SE, is missing mappings to both coordinates and capacity values. We see this mismatch because the LXF is current while the HTP is about 6 months behind (at the time of running), and so this station may have been discontinued. Luckily, this is easily fixed by finding the appropriate values and inputting them manually. Next, we see that there is an odd pattern with station names, ID numbers, and geographical coordinates: The station name features have 8 more unique values than both the coordinate and station ID features when we expect them to have the same number of unique values. The station ID numbers and coordinates are matched up equally because we used station ID numbers as a key when we joined the two datasets, so we need to find the issue with the station names feature. We find the problem and it turns out that, over time, the same station IDs are assigned to new, slightly different station names. For example, station ID #31039 is marked "N Quincy St & Wilson Blvd" in 4,315 cases and "Wilson Blvd & N Quincy St" in 1,363 cases. This seems to be a function solely of time and so nothing too unusual is going on with the data. Since we have station ID as a reliable identifier feature, we drop the station name feature entirely.

Now we have reliable mappings between stations and geographic coordinates over this time period, and can create visualizations in Tableau:



This bubble map[6] shows the start station popularity of all stations over the beginning of 2015 Q3 to the end of 2016 Q3. The more popular stations to begin a trip from are redder and larger than other stations.

[6] https://public.tableau.com/profile/dpalbrecht#!/vizhome/CapitalBikeshareStartingStatio

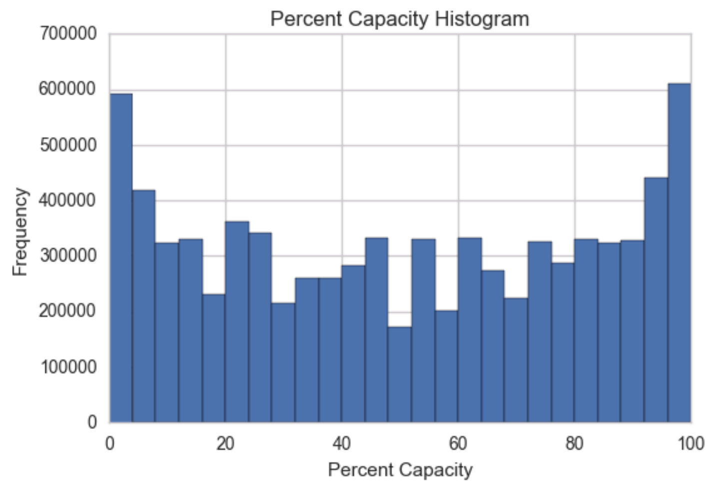


This spider map[7] shows the most popular routes taken to each station. For example, when traveling to station ID #31116, most people start at the stations that are connected by redder, larger lines.

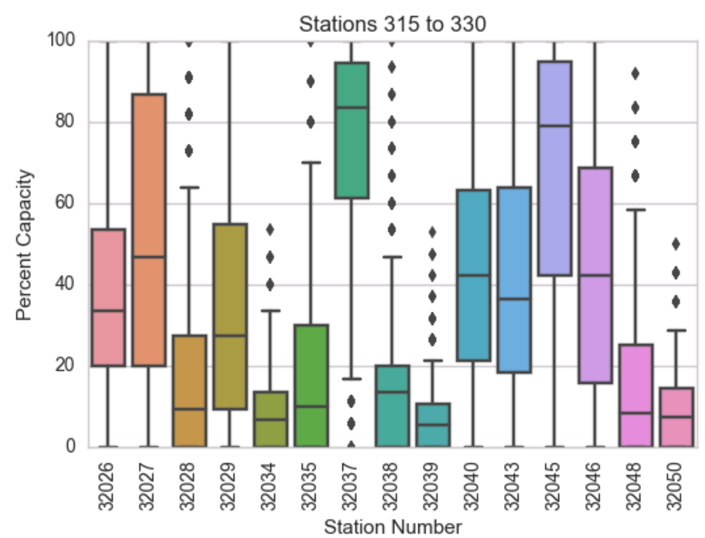
Now we move on to loading in the Historical System Outage (HSO) dataset. *Using the first empty or full notification for each station, we now have the third, and final, piece that allows us to estimate the percent capacity feature for each station*. Without this information, we might have had to blindly assume that each station is empty or full to begin with and neither assumption is very good when considering that all other stations should follow the same assumption. Inspecting this data compared to the combined dataset, though, we find that there is no outage data for 77 stations. One plausible explanation is that we might expect that the less popular stations do not experience full or empty states (especially over a relatively short time period). I decided to exclude these stations from the analysis and we are left with 330 stations. Taking the reduced, combined dataset, we append the single full or empty flag and determine, for each station, whether the station is most likely closer to full or empty and continue with that assumption as we estimate percent capacity values. We iterate over each instance and determine whether the percent capacity increases or decreases based on whether a trip began at the station or ended at the station. *We have finally engineered our best estimate for the target feature, percent capacity, and can continue to explore the completely combined dataset*.

[7] https://public.tableau.com/profile/dpalbrecht#!/vizhome/CapitalBikeshareTotalTripPopul

We start with a histogram over all stations to determine where percent capacity values tend to fall:



We see that values fall between 0% and 100%, as we would expect for percent capacity values. We also notice that percent capacity exhibits a bimodal distribution (keeping in mind that it is truncated), which supports the idea that there are stations that stay too full or too empty. Do individual stations have patterns we might expect as well? Below we see the boxplots for a sample of the last 15 stations:



We see that each station shows values we might expect depending on the popularity of each station, and the percent capacities do not seem to be too clumped at any value or show too many outliers.

The target feature follows values that we would expect, which is a good sign, but we are not too sure how much data there is for any particular station. We already experienced this potential issue when looking at the HSO dataset when we noticed that 77 stations were without outage flags. Since we are attempting to optimize the bike rebalancing process, we want to focus on stations with enough data for analysis as these are the stations with the most user activity. We pick two thresholds for determining whether a station has enough data:

1. Are there stations with less than 90% of the date range of the station with the largest date range?
    - For example, if the most popular station has 100 days worth of data (there are 100 days between the start and end of the time series), we exclude any stations with less than 90 days worth of data.
2. Are there stations with less than 4 instances per hour per day, on average?
    - For example, if a station has at least 90 days worth of data (satisfies condition #1 above), we would expect any station, in this scenario, to have at least 8,640 instances (90 days * 24 hours/day * 4 instances/hour).

If we find a station satisfies these two conditions, we can be confident that the station is 1) popular enough to possibly need bike rebalancing and 2) has enough data for a time series forecast. When testing these two thresholds, we find that 6 stations do not satisfy condition #1 and 257 stations do not satisfy condition #2. After discarding these stations from analysis we are left with 67 stations. At this point, we have reduced our dataset by about 44% (approximately 8.1 million instances down to 4.5 million instances).

We now have all of the stations that we want to forecast, so we can continue by transforming it into a state that a time series algorithm will do well with. We recall that the main dataset we base this analysis on, HTP, was actually in a network format like this:

Duration	**Start Date**	**End Date**	**Start Station Number**	**Start Station**	**End Station Number**	**End Station**	**Bike Number**	**Member Type**
257866	7/1/2015 0:00	7/1/2015 0:04	31116	California St & Florida Ave NW	31117	15th & Euclid St NW	W21516	Registered

To transform this dataset initially, we had to essentially split the dataset down the middle so that, for any station ID, if it was in the "Start Station Number" feature above then we know that this equates to a reduction in the percent capacity feature and, conversely, if the station ID was in the "End Station Number" feature above then this must equate to an increase in the percent capacity feature. We were left with a dataset that looks like:

Date	**Station Number**	**Count Percent Capacity**
2015-07-01 08:25:00	31116	5.263158
2015-07-01 08:26:00	31116	0.000000
2015-07-01 09:42:00	31116	5.263158
2015-07-01 09:55:00	31116	0.000000
2015-07-01 10:38:00	31116	5.263158

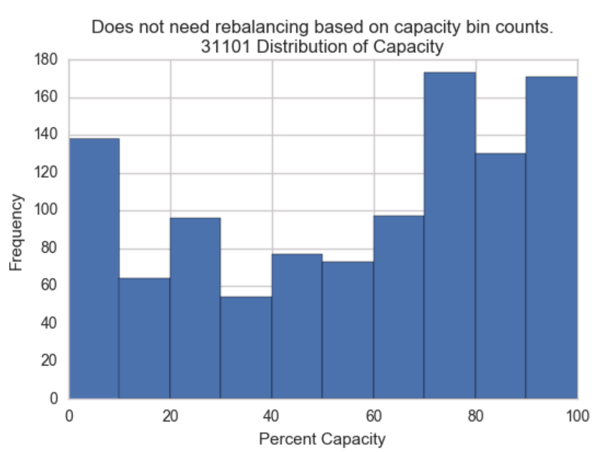
Clearly, while the dataset has been transformed immensely, the data points come in very irregularly. To account for this, I decided to upsample the data to every hour and linearly interpolate the new percent capacity values. We now have a dataset that is much more consistent, in terms of timestamps, both within and between each station:

Date	**Station Number**	**Count Percent Capacity**
2015-07-01 08:25:01	31116	2.631579
2015-07-01 09:25:01	31116	2.631579
2015-07-01 10:25:01	31116	2.631579
2015-07-01 11:25:01	31116	1.315789
2015-07-01 12:25:01	31116	2.631579

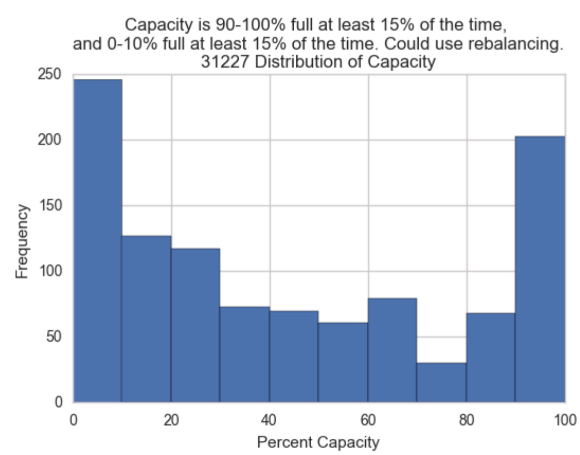
At this point, we have reduced our dataset by about 84% (approximately 4.5 million instances down to 735,000 instances). This technique has fundamentally changed our data and is less representative of the reality for each station, but it is a necessary step because it distributes the data points more evenly over time.

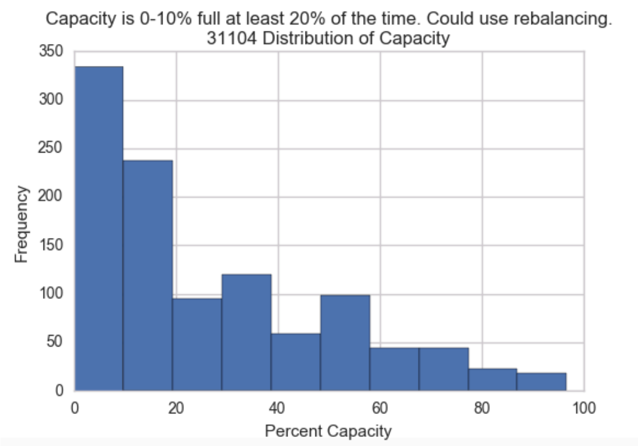
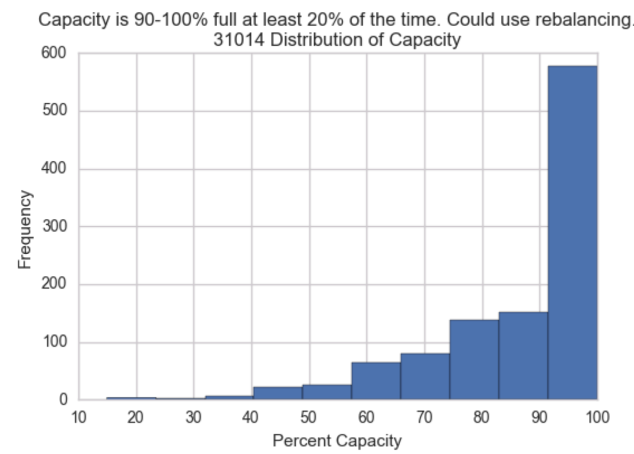
Now we have narrowed down our list of stations for rebalancing to 67, but does each station *need* to be rebalanced? We have already proven that these stations have enough data to study rebalancing, but if a station fluctuates enough over the period and does not stay too empty or too full for too long then rebalancing is probably not needed. We can take a look at station percent capacity distributions through histograms. Below are the four different types of distributions we find (keeping in mind that they are truncated):

30 stations take this distribution:



6 stations take this distribution:

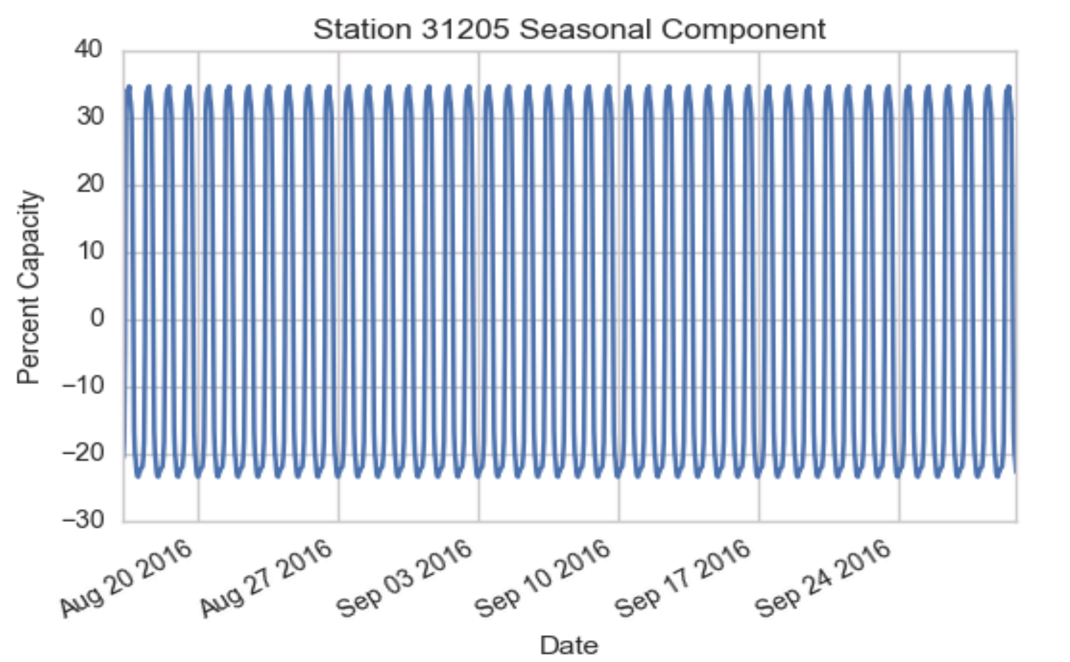


15 stations take this distribution:	16 stations take this distribution:
	

We can see that 30 of the 67 stations fluctuate enough so that they most likely would not benefit from rebalancing, and so we are left with 37 stations for forecasting purposes.

Forecasting and Evaluation

As mentioned while initially introducing the rebalancing issue, I compared bikeshare usage to traffic hour in the way that they both show cyclical trends. This comparison points to a sense of seasonality in the data that needs to be captured by a prospective model. Below is an example of a seasonal decomposition of a station's time series over the training timeframe:



As one might expect, we see there is clear seasonality that follows an approximately daily trend. This sort of seasonality is also present in various forms among the other 36 stations. Taking this into account, it seems reasonable to forecast all stations with a Seasonal Autoregressive Integrated Moving Average (SARIMA) model, which is a variation of the Autoregressive Integrated Moving Average (ARIMA) model specifically for modeling time series that are suspected of having seasonal effects. A SARIMA, in general, comprises of a non-seasonal Autoregressive (AR or p) value, non-seasonal differencing (d) value, non-seasonal Moving Average (MA or q) value, seasonal AR (or P) value, seasonal differencing (or D) value, seasonal MA (or Q) value, and periodicity (s) value in the series. For each station, we will need to find the optimal p, d, and q parameters that are required by the model and leave default values of P, D, Q, and s. We automate the process to iterate over the last ~10% of each station's series and find the best parameter values. The process iterates over all values of p and q between 1 and 10, for each station, and finds the best p and q values based on the Akaike Information Criterion (AIC) where their sum is not greater than 15. The d value, as mentioned above, is the differencing of each series, and is always set to 0 in this case because the time series for these stations are rather stationary as tested by an Augmented Dickey-Fuller (ADF) test. All p-values related to each station's ADF test can be found in Appendix A: Stationarity P-Values.

Training on the same portion of the series (~10% as we did above when finding parameter values) and then forecasting on the next 24 hours, I took into account both the forecasted plot of percent capacity and the associated distribution flag (90-100% full 20%+ of the time, 0-10% full 20%+ of the time, or 0-10% full 15%+ and 90-100% full 15%+ of the time), as discovered at the end of the previous section, to outline a rebalancing schedule for the next 24 hours of each station. Since the forecasts ranged in quality, I evaluated them on two qualitative scales and then a single quantitative score:

<u>Qualitative</u>:

1. **Schedule Appropriateness**: Does the proposed schedule alleviate potential problems seen in the actual data?
   - Values: Poor, Fair, Good, Very good.
2. **Model Fit**: Does the model fit the actual data well and follow its trends well?
   - Takes the same values as Schedule Appropriateness.

<u>Quantitative</u>:

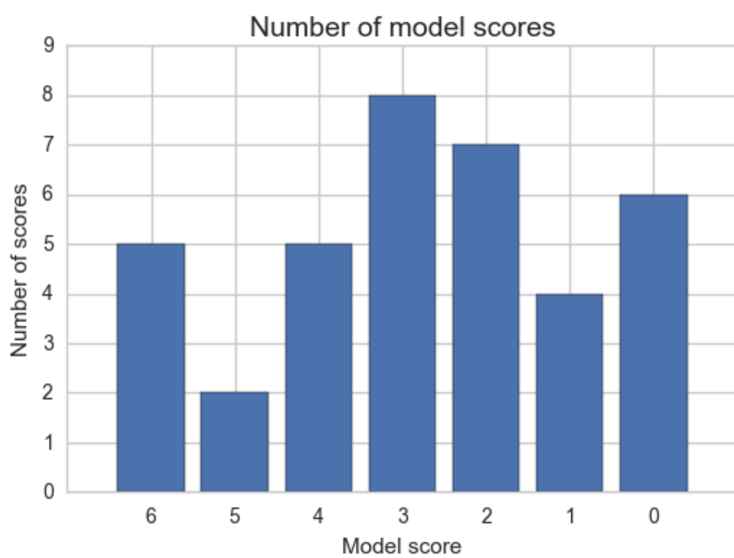
3. **Model Score**: The addition of each model's qualitative ratings and recorded quantitatively where Poor = 0, Fair = 1, Good = 2, and Very good = 3.

- For example, a model that leads us to make an appropriate schedule (Schedule Appropriateness: Very good = 3), but does not seem to follow the observed data very well (Model Fit: Poor = 0) receives a score of 3 + 0 = 3.

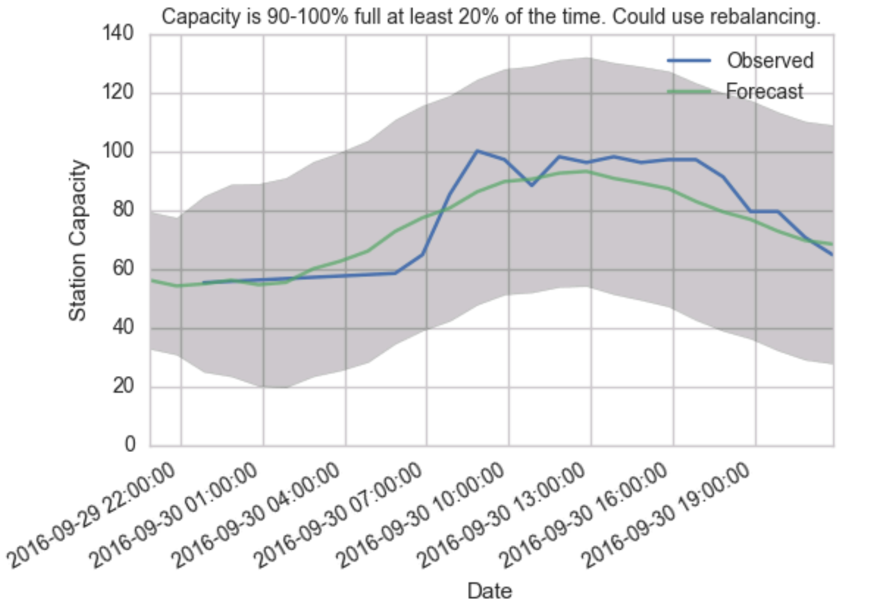
The rebalancing schedule in its entirety can be found in Appendix B: 24-Hour Rebalancing Schedule. In the next section, we take a closer look at the results of the forecasting process.

Forecasting Results

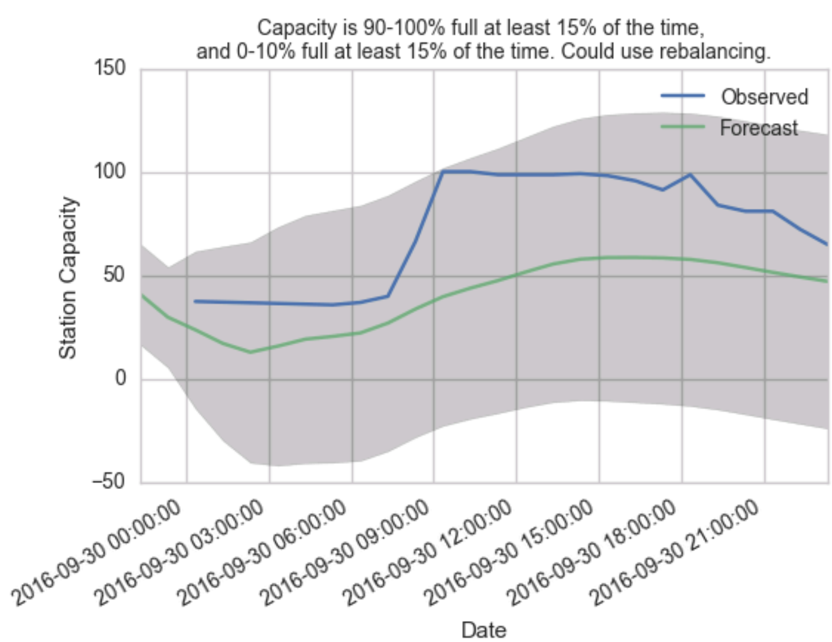
As touched on above, each individual SARIMA model behaved differently on its respective time series, and possible explanations for this are explored in the subsequent section. To make it easier to judge the number of potentially useful models, we can visualize the model scores in a bar graph:



Scoring a 3 and above is generally indicative of a useful model, so we can see that about 20 out of the 37 forecasts are expected to help us accurately and predictively rebalance bikes. More specifically, let's explore a forecast that exhibits a model score of 6:



Knowing that this station, ID #31215, is 90-100% full at least 20% of the time in the training set, we can infer that upticks in the forecast may be indicative of a need to subtract bikes and rebalance the station. From this forecast we might suggest a schedule to subtract bikes around 1:00pm on September 30th, 2016, which, when compared to the observed data, shows that it would alleviate the issue that the station remains very full until around 5:00pm the same day. This corresponds to a schedule appropriateness of "Very good", or a 3. Comparing the forecast overall to the observed data, we see that it captures trends very well for a model fit of "Very good", or a 3. Adding these two scores, we come to a model score of 6. How does a less suitable forecast fare – one that exhibits a model score of 3? Let's explore further:



Knowing that this station, ID #31205, is 90-100% full at least 15% of the time and 0-10% full at least 15% of the time in the training set, we can infer that upticks and downticks in the forecast may be indicative of a need to subtract and add bikes to the station, respectively. From this forecast we might suggest a schedule to add bikes around 3:00am and subtract bikes around 3pm on September 30th, 2016, which, when compared to the observed data, shows that it would help alleviate the issue that the station remains very full until around 6:00pm the same day. This proposed schedule, however, overstates the low point around 3:00am and would cause us to rebalance when there would be no need, which is why this station's forecast received a schedule appropriateness of "Fair", or a 1. Comparing the forecast overall to the observed data, we see that it captures trends rather well for a model fit of "Good", or a 2. Adding these two scores, we come to a model score of 3.

Focusing on the stations that modeled well and provided helpful rebalancing schedules in Appendix B, we need to specify how many bikes we might add or subtract at any given time. We have to remember that the stations generally fluctuate between 0-100% at some point, and changing the number of bikes at a station affects *the remainder* of the series. This is a tough question and really depends on how often the station oscillates between empty and full. It is likely the safest rule of thumb to keep each station from being completely full or empty by a few bikes in lieu of more extreme changes.

Trial and Error

Several iterations and a few different models were used when attempting to forecast each station. In order of attempt, below are the different methods used and rationale for discarding them:

- Using Partial Autocorrelation Function (PACF) and Autocorrelation Function (ACF) plots, I determined the appropriate p and q parameters for an ARIMA model related to station ID #31116. I then used the same parameters for the remainder of the stations. This posed two issues: 1) It does not make sense to generalize a single station's model order to all stations and 2) the ARIMA model tended to converge to the mean very quickly in a long-term forecast.
- Using the PACF and ACF plots found previously, I tested several seasonal order parameters (P,D,Q) in order to model a proper SARIMA for station ID #31116. This posed two issues: 1) Finding the seasonal order parameters proved very time consuming and prone to overfitting (and not very statistically minded) and 2) each forecast is done on a specific

section of time, which means that each subsequent forecast may require a different model even for the same station.

- In an effort to find general non-seasonal parameters for different types of stations, I attempted to cluster stations in a supervised matter using hierarchical clustering. I clustered on:
    - Mean capacity values over a 24-hour time period for the entire year
    - Full PACF/ACF vectors
    - Sparse PACF/ACF vectors containing only certain values in the plots

  While the clusters created through these techniques appeared plausible, generalization did not improve substantially compared to previous attempts.
- Using WEKA's forecast package was straightforward and fast to implement, but issues arose when forecasts ventured outside of the 0% or 100% capacity bounds.
- Using an LSTM Neural Network in an effort to use a less statistical forecasting technique also did not lend itself to forecasting over a future 24-hour time period. The benefit of the LSTM model was much faster model selection, and, as with the ARIMA model, forecasts were suitable one step ahead but quickly converged to the mean when forecasting more than a few steps into the future.
- Turning back to the SARIMA model, I implemented R's auto_arima and timeseries (ts) functions to instill frequencies of 7 and 24, and effectively find seasonal as well as non-seasonal parameters. A frequency of 7 did not fit the data well and often returned erratic and extreme forecasts, and a frequency of 24 took far too long to optimize and train to be usable in a business setting. Further, I implemented different trend polynomials during training, but this did not improve forecasts. Dropping seasonal parameters and trend polynomials led me to the most successful models that are outlined in previous sections.

Open Questions and Conclusions

At this point we might ask: Why didn't all of the stations forecast well? This might be the case for several reasons:

- The data was artificial to some extent since it was created using three different datasets, so it is possible that true trends were washed out and too much signal was lost.

- A 24-hour time period for forecasting is considered to be a relatively long-term window, and forecasts tend to degrade the further they are extended. Given the nature of the problem, a 12-hour time period might be more accurate but less useful when used in a business setting.
- Capital Bikeshare scrubbed the original data, HTP, and so it is possible that some signal was lost in the process.
- Capital Bikeshare may already be engaging in bike rebalancing, which would keep us from finding the stations that truly need rebalancing.
- The original data, HTP, was in a network format, which, when transformed into a time series format, resulted in each station having different timestamps for each instance of the target feature. Upsampling helped reduce the effect of this issue, but each station also did not have the same number of days of data. These two factors led me to pursue a univariate time series forecast that assumes all stations operate independently when, in reality, stations can be highly correlated. A multivariate time series forecast may have been a better solution given proper data.

Overall, the dataset that we finally used for forecasting was built using three different datasets and transformed into a time series problem. We reduced the original dataset with 407 stations by about 84% through inspection of data volume and upsampling transformations, and then we further reduced the number of stations through inspection of distributions down to 37 stations. Further, of those 37 stations, about 20 stations are expected to model fairly well or better over a 24-hour time window using a SARIMA model.

The process of rebalancing by using useful forecasts could potentially alleviate the issue of stations running too full or too empty, and allow commuters to be more confident that they can start or end a trip at any station (especially if it is popular). In all, the benefits are three-fold: The bikeshare company will 1) be highly regarded for proactively stocking all stations and can more readily guarantee that riders will not encounter empty or full stations; 2) benefit from lowered real-time tracking costs such as in a mobile application; and 3) benefit from reducing costs associated with reactively and inefficiently allocating resources to rebalance bikes after stations are already full or empty.

Appendix A: Stationarity P-Values

Station	P-Value*	99% Confidence Threshold	95% Confidence Threshold	90% Confidence Threshold
31014	-6.7795	-3.4364	-2.8642	-2.5682
31104	-4.8473	-3.4365	-2.8642	-2.5682
31108	-3.7657	-3.4365	-2.8642	-2.5682
31114	-2.7085	-3.4365	-2.8642	-2.5682
31116	-3.4669	-3.4365	-2.8642	-2.5682
31119	-3.0714	-3.4365	-2.8642	-2.5682
31121	-9.8123	-3.4365	-2.8642	-2.5682
31205	-3.2475	-3.4365	-2.8642	-2.5682
31213	-3.3367	-3.4365	-2.8642	-2.5682
31214	-3.4692	-3.4365	-2.8642	-2.5682
31215	-5.8329	-3.4365	-2.8642	-2.5682
31219	-3.0927	-3.4365	-2.8642	-2.5682
31223	-3.6577	-3.4365	-2.8642	-2.5682
31225	-5.5177	-3.4365	-2.8642	-2.5682
31227	-3.2511	-3.4365	-2.8642	-2.5682
31228	-3.6594	-3.4366	-2.8643	-2.5682
31229	-3.2121	-3.4365	-2.8642	-2.5682
31230	-3.6545	-3.4365	-2.8642	-2.5682
31231	-3.4002	-3.4365	-2.8642	-2.5682
31238	-3.3128	-3.4365	-2.8642	-2.5682
31239	-2.9841	-3.4365	-2.8642	-2.5682
31241	-3.1881	-3.4365	-2.8642	-2.5682
31244	-3.8333	-3.4365	-2.8642	-2.5682
31246	-6.8061	-3.4364	-2.8642	-2.5682
31258	-8.4821	-3.4364	-2.8642	-2.5682
31263	-5.5857	-3.4365	-2.8642	-2.5682
31266	-4.567	-3.4365	-2.8642	-2.5682
31269	-7.0763	-3.4364	-2.8642	-2.5682
31277	-3.4007	-3.4365	-2.8642	-2.5682
31281	-3.3072	-3.437	-2.8644	-2.5683
31600	-3.8455	-3.4365	-2.8642	-2.5682
31602	-3.4226	-3.4365	-2.8642	-2.5682
31603	-4.0983	-3.4365	-2.8642	-2.5682
31613	-8.1876	-3.4364	-2.8642	-2.5682
31616	-3.9039	-3.4365	-2.8642	-2.5682
31627	-4.2399	-3.4365	-2.8642	-2.5682
31628	-3.2699	-3.4365	-2.8642	-2.5682

* We are 99% confident that 22 stations are stationary, 95% confident that 14 stations are stationary, and 90% confident that 1 station is stationary. We are at least 90% confident in each individual station's series stationarity.

Appendix B: 24-Hour Rebalancing Schedule

Station	Proposed Schedule	Model Score
31014	Subtract bikes near the beginning of the period.	4
31104	Add bikes around 9am.	6
31108	Subtract bikes near the beginning and end of the period.	4
31114	Subtract bikes near the middle of the period.	4
31116	Subtract bikes around midnight.	6
31119	No need for rebalancing.	0
31121	No need for rebalancing.	0
31205	Add bikes around 3am and subtract bikes around 3pm.	3
31213	Add bikes near the beginning of the period.	4
31214	Subtract bikes near the beginning and end of the period.	3
31215	Subtract bikes around 1pm.	6
31219	No need for rebalancing.	2
31223	Subtract bikes near the beginning and end of the period. Add bikes near the middle of the period.	1
31225	No need for rebalancing.	6
31227	No good model.	0
31228	Subtract bikes around 1pm.	2
31229	Subtract bikes near the beginning of the period and add bikes near the middle of the period.	1
31230	Add bikes around midnight and subtract bikes around 12pm.	6
31231	Add bikes around 2am and subtracts bikes around 11am.	2
31238	No need for rebalancing.	3
31239	No need for rebalancing.	0
31241	No need for rebalancing.	3
31244	No need for rebalancing.	5
31246	No need for rebalancing.	3
31258	No need for rebalancing.	3
31263	No need for rebalancing.	4
31266	No need for rebalancing.	0
31269	No need for rebalancing.	2
31277	No need for rebalancing.	2
31281	Subtract bikes around 2am and 8pm.	3
31600	No need for rebalancing.	2
31602	No need for rebalancing.	0
31603	Subtract bikes around 6am.	5
31613	No need for rebalancing.	2
31616	Subtract bikes around 2am.	1
31627	No need for rebalancing.	1
31628	No need for rebalancing.	3