CAPSTONE PROJECT FOR THE MASTER OF SCIENCE IN APPLIED URBAN SCIENCE AND INFORMATICS

BUS RELIABILITY METRICS USING PUBLIC MTA BUS TIME DATA

July 25, 2016

Jiaxu Zhou
Yuqiao Cen
Matthew Urbanek
Bonan Yuan
Sara Arango-Franco

Advisors:
Dr. Kaan Ozbay
Dr. Huy T. Vo

Sponsor Agency:
Department of Transportation, City of New York

Center for Urban Science and Progress
New York University

Contents

BRIEF

Despite a growing demand for public transportation in New York City, bus ridership levels are declining. This can be explained by drops in vehicle speeds and customer perceptions of dependability. The *New York City Department of Transportation* (NYC DOT) wished to engage the *New York University Center for Urban Science and Progress* (CUSP) to explore the use of public vehicle location data from the *Metropolitan Transit Authority* (MTA) Bus Time to generate operational data relevant to the DOT's planning decisions. This information is provided in the form of reliability metrics for bus service.

Based on the MTA *Automated Vehicle Location* (AVL) data and its public Bus Time API, the team performed a data assessment analysis for the data generating process and the data collection process. We also deliver methods for estimating bus travel and stop times, measuring reliability with different metrics, and settles the ground for the DOT to identify the distribution of reliability measurements as a function of factors regarded as relevant to their practice.

ACKNOWLEDGEMENTS

We would like to express our sincere appreciation to all who have lent us hands during this time. First of all, we would like to show our sincere gratitude to our sponsor agency, the New York City Department of Transportation, for giving us an opportunity to focus on this important city issue. We would specially like to to thank Jeremy Safran, who gave us plenty of insight into the way the DOT wants to treat and understand bus data analytics. Secondly, we would like to express our gratitude to our advisors, Dr. Kaan Ozbay and Dr. Huy T. Vo., not only for their technical expertise but for their generosity and willingness to lead us through this process. Invaluable consultation on use of the Bus Time API for performance analysis was provided by Nathan Johnson and Kuan Butts, primary authors of the NYC Bus Transit Data project (http://busdataapi1.cloudapp.net/); and Chris Pangilinan and Zak Accuardi at Transit Center. Last but not least, we would like to thank Kai Zhao, Abdullah Kurkcu, Ender Morgul, and our classmates in CUSP who gave us assistance and advice during this process. We will always be thankful.

0.1 OVERVIEW AND BACKGROUND

0.1.1 Statement and Description

This project focuses on identifying these two factors along the city's MTA operated bus routes, in the form of reliability metrics that are relevant to the DOT as the municipal traffic authority.

The DOT receives batch files from MTA containing archived basic AVL data, but the agency lacks a formal process for compiling and analysing it according to their decision making capabilities (such as those related with road design and traffic management), which substantially differ from the MTA's purposes and reach. Additionally, the basic AVL data differs significantly in density and reported elements compared to those from the Bus Time API. Although the MTA, being the agency in charge of operations, has internally defined metrics used for scheduling, planning and analysis of the bus service, this their attention is on the bus level instead of the whole system level, which is more of the interest of the DOT.

To help the DOT improve their planning efficiency, this project aims to assess the usefulness of Bus Time data for measurement of the dependent variable metrics related to bus performance and reliability, enabling the agency to, in a further analysis, understand the effect of independent variables that affect service quality.

0.1.2 Goals

The goals of the project can be summarized as follows:

1. Perform a thorough data quality assessment on the MTA Bus Time data.
2. Implement metrics related to the performance of the buses with respect to customer experience, which in part considers their planned schedule.
3. Document the entire process and deliver flexible code, so both data quality assessment and reliability measurements can be reproduced by the agency.

0.2 PROJECT OFFERINGS

This Capstone project hopes to address the core urban challenge of transportation planning with the following contributions:

- The conclusion that public, self-service data can be applied to evaluate transit operations performance, independent of the operating agency
- A framework for a new, more efficient data interface between agencies and for processing of the data
- Demonstration of the potential for implementation of the framework using only a personal computer and open source software
- New value realized from a public service originally implemented with a narrower scope in mind

0.3 Approach

In this section we discuss the data retrieval, processing and reliability measurement techniques. Because MTA does not continually publish archived AVL data, data was retrieved from the Bus Time API in real-time and stored on NYU CUSP servers. Parsing is required both to store data efficiently and to decrease processing time of subsequent methods. We then propose a variety of transit system performance metrics that can be applied using these data before discussing the results.

0.3.1 Bus Time data extraction

MTA's AVL data is acquired using a ”get” request to a web service offered by the MTA to the public. The service uses a standard protocol, originally developed in Europe, called *Service Interface for Real-Time Information* (SIRI).

Per MTA's recommendation, the response data is requested in JSON format (*Javascript Object Notation*). JSON offers a flexible structure, for example allowing elements to be stored in hierarchies, or a combination of named and unnamed elements. JSON does not transform directly to a tabular layout and as a result cannot be imported by typical data analysis tools like Microsoft Access or Excel without first parsing it.

Parsing can be performed using a variety of approaches. A small program that can be written on almost any personal computer (a macro, or command-line script) may be acceptable for parsing one JSON response, requiring in the order-of-magnitude one second for each JSON. However this is likely unacceptable for reading and aggregating any meaningful amount of archived JSON response files (entire days, or entire lines over multiple days). Advanced techniques requiring additional software or hardware can significantly improve processing time by distributing the data across multiple processing units. Regardless of the technique, the extracted data has a straightforward structure, containing text and numerical elements (which can be stored as text), appropriate for storage in a comma-separated values (CSV) file.

The SIRI standard calls for date-time representation according to the ISO 8601 standard. While this can be read directly and used for many typical calculations (for example, elapsed time), it poses a problem when performing any analysis requiring date or time data from the planned schedule for the buses. Those analysis require both the ”true” date-time element as well as the trip reference date. Here are two example comparisons of a bus trip's estimated departure from its first stop and the corresponding scheduled departure time. Such a comparison may be used to explain whether bus reliability can be attributed to operational issues such as late departures from the depot. Note the conversion and that time zone information is dropped; in the data for this project, both SIRI response and schedule data time elements are written in the local time zone.

trip_id	Time stamp	Converted time stamp	Scheduled time
OH_B6WeekdaySDon077600_M101_100	20160613T13:09:32.00004:00	0d13:09:32	13:08:55
MV_B6WeekdaySDon041500_M5_206	20160614T01:20:42.00004:00	1d01:20:42	25:12:15

0.3.2 Big Data Techniques

Our data extraction step picks the useful information from the original json and dumps the redundancies. This step shrinks file size by 30 times and facilitates future analysis by creating a human readable table that can easily be analysed through Excel by anyone.

However, due to the time required to process and parse each JSON, we decided to apply big data techniques for the data. Each JSON file is originally between one and three megabytes, before parsing; the entire set of collected data is about 3 terabytes..We apply Apache Spark along with some Spark SQL techniques for manipulating the data in large scale. It takes around 30 to 45 minutes for the processing the entire year data depending on the performance of the server. Without the use of this technique, processing only one day of data takes 10 to 15 minutes.

JSON ELEMENT(schema)	**Column NAME**	**Explanation**
LineRef	ROUTE_ID	Name of bus line(B42)
VehicleLocation.Latitude	latitude	latitude of record
VehicleLocation.Longitude	longitude	longitude of record
RecordedAtTime	recorded_time	What time it gets recorded
VehicleRef	vehicle_id	ID of vehicle
FramedVehicleJourneyRef.DatedVehicleJourneyRef	TRIP_ID	Same as trip_id in GTFS*
FramedVehicleJourneyRef.DataFrameRef	trip_date	Date of the trip
JourneyPatternRef	SHAPE_ID	Same as shape_id in GTFS*
StopPointRef	STOP_ID	Id of next stop,Same as stop_id in GTFS*
Extensions.Distances.DistanceFromCall	distance_stop	Distance to next stop
Extensions. Distances.CallDistanceAlongRoute	distance_shape	Stop_s total distance along the shape
Extensions. Distances.PresentableDistance	status	Report the current status of bus to next stop [1]
DestinationRef	destination	Headsign of bus

0.3.3 Schedule data extraction

Schedule data is published by MTA according to the General Transit Feed Specification, a standard established in 2006 and now widely used by transit agencies and developers. One transit feed is essentially a small relational database, containing a minimum of six tables and any of seven optional tables. Basic required data in a transit feed file are routes, trips, stops, stop times, and effective date ranges. MTA does not include optional metadata that can be used to distinguish multiple publications covering the same schedule period.

0.3.4 Data quality assessment

We implement three general approaches to assessing the completeness and validity of Bus Time data. Schedule data feeds from MTA are presumed to be complete and accurate, given their widespread use.

1. Analysis of time-series patterns
2. Correlation of Bus Time data length to planned level of bus activity, as informed by schedule data
3. Validation of individual elements

0.3.5 Arrival time estimation techniques

Three methods are investigated for estimating a vehicle's arrival time at a certain location, such as a bus stop. Arrival time estimates are the basis for most measurement and identification related to bus reliability, such as running times and headway. Calculation of headway, which is in turn the basis for many performance metrics, also requires estimates of all vehicles' arrival times at a certain location. Missing estimates (whether due to the data generating process or the chosen estimation algorithm) must be identified; without them, the resulting headway will be the difference in arrival times of non-sequential vehicles.

The first method is use to use a spatial algorithm to identify any data points within a certain radius of the location, and make an estimation from within that subset. Possible calculations on that subset are to take earliest or latest time recorded, the median time recorded, or apply an interpolation to generate a point-estimate at the exact location. The second method is to apply an interpolation algorithm to all observations reported for a given vehicle's trip. The tradeoff is that this may be more computation than required if few points along the trip are to be estimated. The risk with linear interpolation is that it assumes the bus is traveling at uniform speed while in reality the bus speed changes a lot due the complicated traffic condition and traffic lights, especially in New York.

The third method is to apply the interpolation only for stops that are reported in Bus Time as a ”Monitored Call,” that is, the upcoming stop. The advantage of only interpolating the reported stops is that it avoids the risk of reporting stops that the bus never made (for example, due to a detour) and the risk of high inaccuracy in the event the bus did make the stop (due to the long distance over which times must be interpolated).The tradeoff for the third method is that since many trips are not completely recorded, we may lose a lot of information for the missing part.

0.3.6 Reliability metrics

Wait Assessment is a metric used by New York City Transit, defined in the Transit Capacity and Quality of Service Manual as the percentage of actual headways between successive vehicle arrivals that are less than or equal to a given standard. The wait assessment for bus is only measured on weekdays. It is defined as the percentage of observed service intervals that are no more than the scheduled interval plus 3 minutes during peak (7 a.m. - 9 a.m., 4 p.m. - 7 p.m.) and plus 5 minutes during off-peak (12 a.m - 7 a.m., 9 a.m. - 4 p.m, 7 p.m. - 12 a.m.)

Wait Assessment is a simple calculation that can be performed after all headway calculations have been performed for a given location.

OTP (On Time Performance) is defined as the positive difference between actual arrival time and schedule arrival time. Different from the measurement of on-time performance percentage by MTA, which is the percentage reflects the number of buses that arrive within a certain time before or after the published schedule, we use the distribution of a group of data to describe the on-time performance for a single trip. The main reason is the criteria is not provided by MTA for buses. However, During low-frequency period, on-time performance is more important while during the high frequency period, the headways matter more.

Running Time Adherence (measured in %) is defined as the average difference between the actual and the scheduled running times relative to the scheduled running time. When the actual running time is shorter than the schedule, the measure is called shorter running time and otherwise longer running time.

Similarly, *Headway Regularity* (measured in %) is defined as the average difference between the actual and the scheduled headways relative to the scheduled headway. The Headway Regularity describes how evenly distributed actual bus is in relation to scheduled service. If two consecutive buses are further from (or closer to) each other than the scheduled headway, the difference is called a longer (or shorter) headway difference. Bus bunching is an extreme example of short headway.

0.4 RESULTS AND IMPLICATIONS FOR PRACTICE

0.4.1 Data extraction and comparison to current DOT process

Daily files of stored response content from the Bus Time API, totalling approximately 3 terabytes, were successfully parsed and converted into one table containing only the useful elements, totalling approximately 63 gigabytes. It is important to highlight that data extracted from Bus Time is different than the data contained in flat files used by DOT, and not purely supplemental. Table 1 lists the differences along several metadata dimensions [2].

Table 1: Differences between DOT and Bus Time data.

	DOT flat file	**Bus Time API**
Source database	Archived	Real-time
Sample frequency	30 seconds	Limited by reliability of interface (max 30 seconds)
Spatiotemporal elements	Raw NMEA, including speed*	Only time and location (projected onto shapeline)
Trip elements	Route and status only	Includes inferred elements, like Next Stop and Trip ID

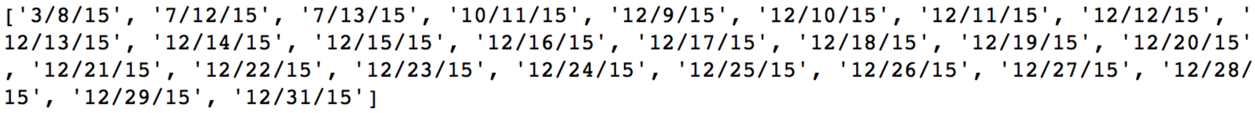
0.4.2 Data quality assessment

The following is an overview of the reliability of the whole dataset. By visualizing the size of the data (for example, in terms of JSON length or number of vehicle records) as a time series and correlating with data from the schedule, we can identify unreasonable variations. Finally, some tactical examination of data elements reveals irregularities that must be considered during the performance measurement phase.

Missing data

The data covers a total number of 318 different bus lines and 340 days in a year. So, although this project tried to focus on the whole year data (from 2015-01-01 to 2015-12-31), some days (listed in Figure 1) are entirely missing.

Figure 1: Missing days in the dataset.



```
['3/8/15', '7/12/15', '7/13/15', '10/11/15', '12/9/15', '12/10/15', '12/11/15', '12/12/15', '
12/13/15', '12/14/15', '12/15/15', '12/16/15', '12/17/15', '12/18/15', '12/19/15', '12/20/15'
, '12/21/15', '12/22/15', '12/23/15', '12/24/15', '12/25/15', '12/26/15', '12/27/15', '12/28/
15', '12/29/15', '12/31/15']
```

From the list of missing days, December is the month with the most missing data, which contains 21 days without data. It is necessary to find out what factors cause this problem. If

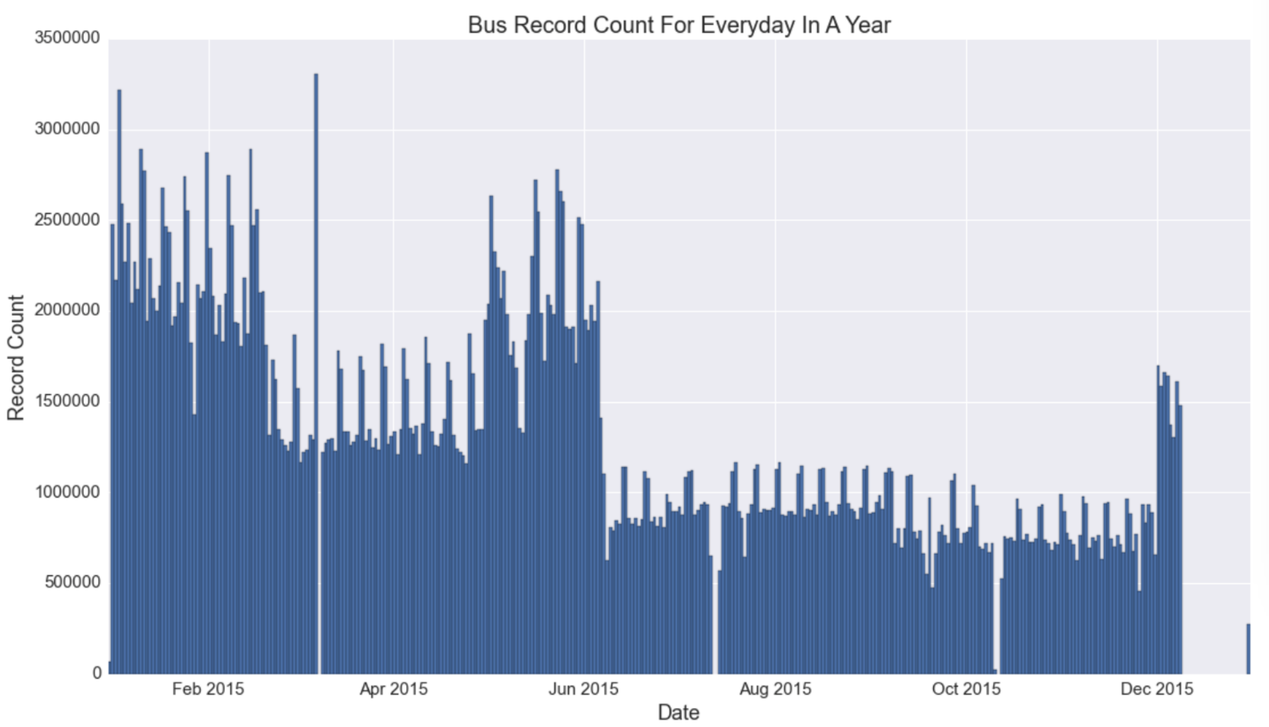
[2] Unclear if reported speed is instantaneous or averaged since last transmission.

it is caused by some uncontrollable factors such as weather, some mitigation plan may be devised. But if it is caused by some human factors of systems factors, it should be avoided in the future.

Visualization of records throughout the year

The total number of bus records by date are shown in Figure 2.

Figure 2: Visualization of records throughout the year.

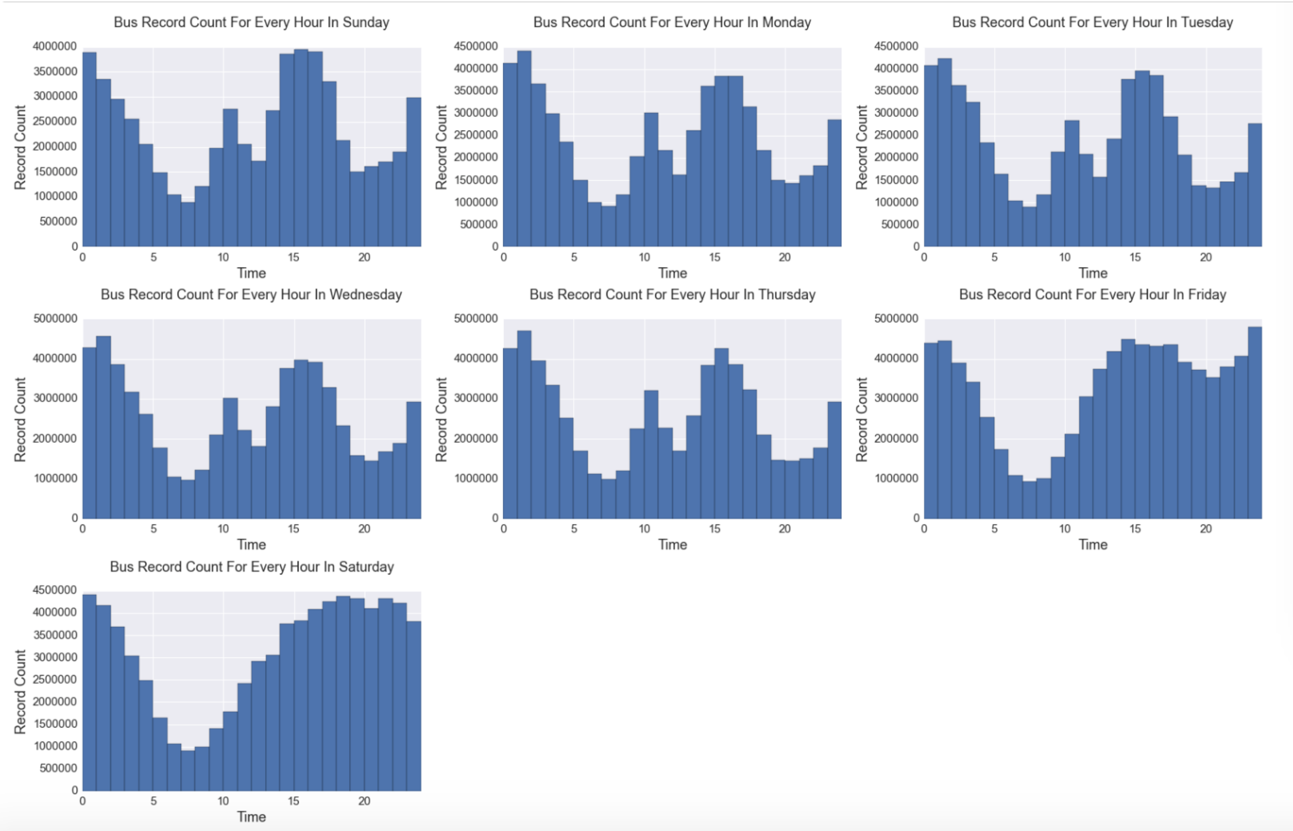


From the plot, some regularities are immediately apparent, including the seven-day cycle. However there are four obvious changes throughout the year. The first one is in February, second in May, third in June, and the last one in December. Further analysis is needed to find out these factors affecting the changes and could help with the bus schedule planning. Also, it can find that March 7th has an extremely high record but March 8th is a day without data which do not exist in other missing days. One can infer that data for March 8th were merged with March 7th.

Daily record counts by hour

From the plot, it can find that weekdays have the same trend and weekends have the same trend. On weekdays, the record count regularly decreases as the morning progresses before

Figure 3: Bus records count for day of the week in 2015.



peaking twice in the middle of the day. In fact, one would expect the opposite based on typical characteristics of urban mobility: one peak during the morning rush-hour and one during the evening rush-hour.

Data density with respect to level of scheduled activity

As the bus is transmitting every 30 seconds, the interval between Bus Time records is expected to also be 30 seconds so long as the vehicle is operating with the AVL equipment activated. Figure **??** shows the actual intervals from the dataset collected. In fact the typical interval turns out to be 60 seconds, with a significant portion of even longer intervals. We examine these long intervals by comparing a measurement of total vehicle activity between the Bus Time data (in terms of record count) and the schedule data (in terms of aggregate running time of all scheduled trips). The measured activity from the two sources turn out to be strongly anticorrelated in the months analyzed (in the table below), whenever materially non-zero. In other words, generally when scheduled activity increases, the density of data available in our dataset decreases.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
-0.83	-0.49	-0.63	-0.72	-0.60	0.08	-0.61	-0.97	-0.52	0.02

To further diagnose the issue, the comparison is made on a finer temporal scale (6-minute timesteps) for a few random dates. Figure **??** shows very different patterns when overlaying the level of scheduled activity onto the level of reported activity derived from the Bus Time data. On one of the two weekdays examined (the first and third), two large gaps are clearly visible and span the entirety of the two daily rush-hour peaks in the schedule. On the weekend date examined, a short gap is visible in the late evening.

The time series analysis showing counter-cyclical data density, the strong anticorrelation, and the tactical examination of a few dates all lead to the conclusion that the dataset collected by NYU CUSP cannot be used for performance analysis of an entire year without major risks to accuracy, in the form of both bias (as discussed in section 0.3.5: Arrival time estimation techniques) and variance (due to the most typical interval being twice as long as specified by MTA). To continue with the demonstration of applying the data for performance metrics, we identified a week with the best density and limited gaps during rush hours (2015-12-01 to 2015-12-07).

Figure 4: Distribution of intervals between SIRI response records.

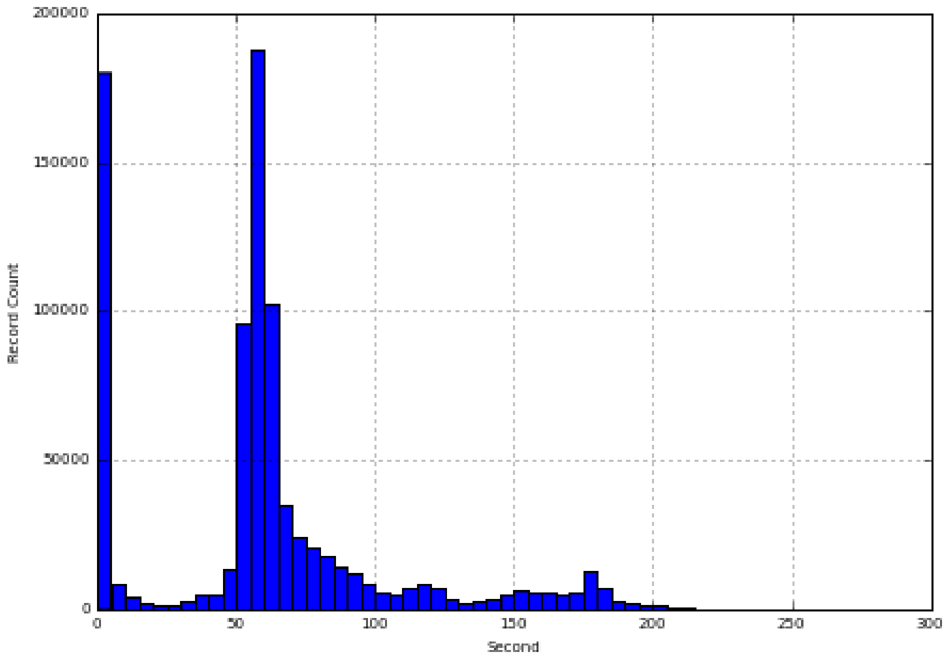
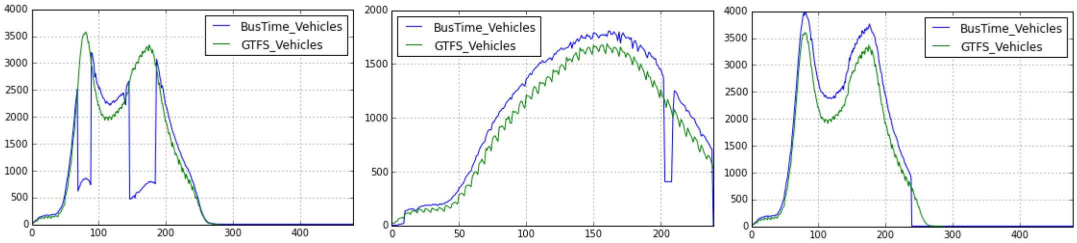


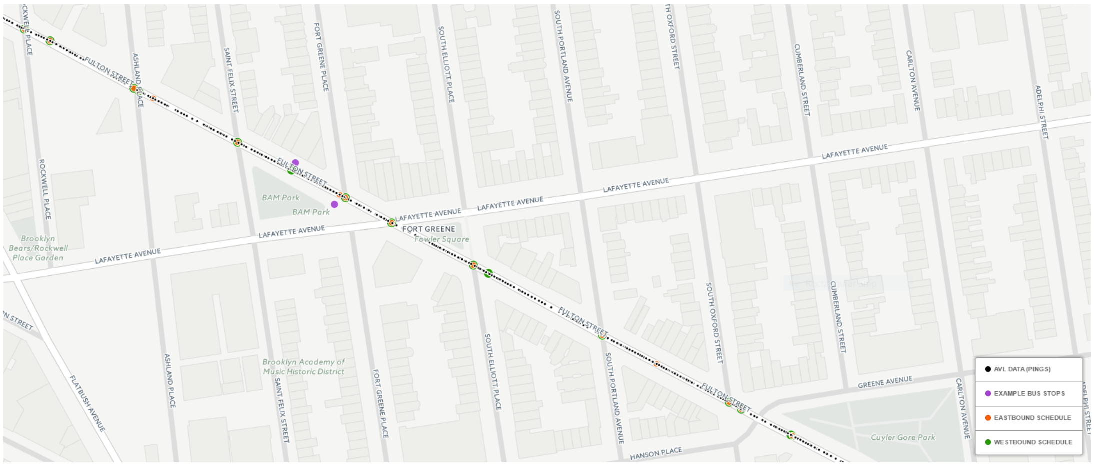
Figure 5: Comparison of active vehicle count for three example dates.



Tactical validation of individual elements

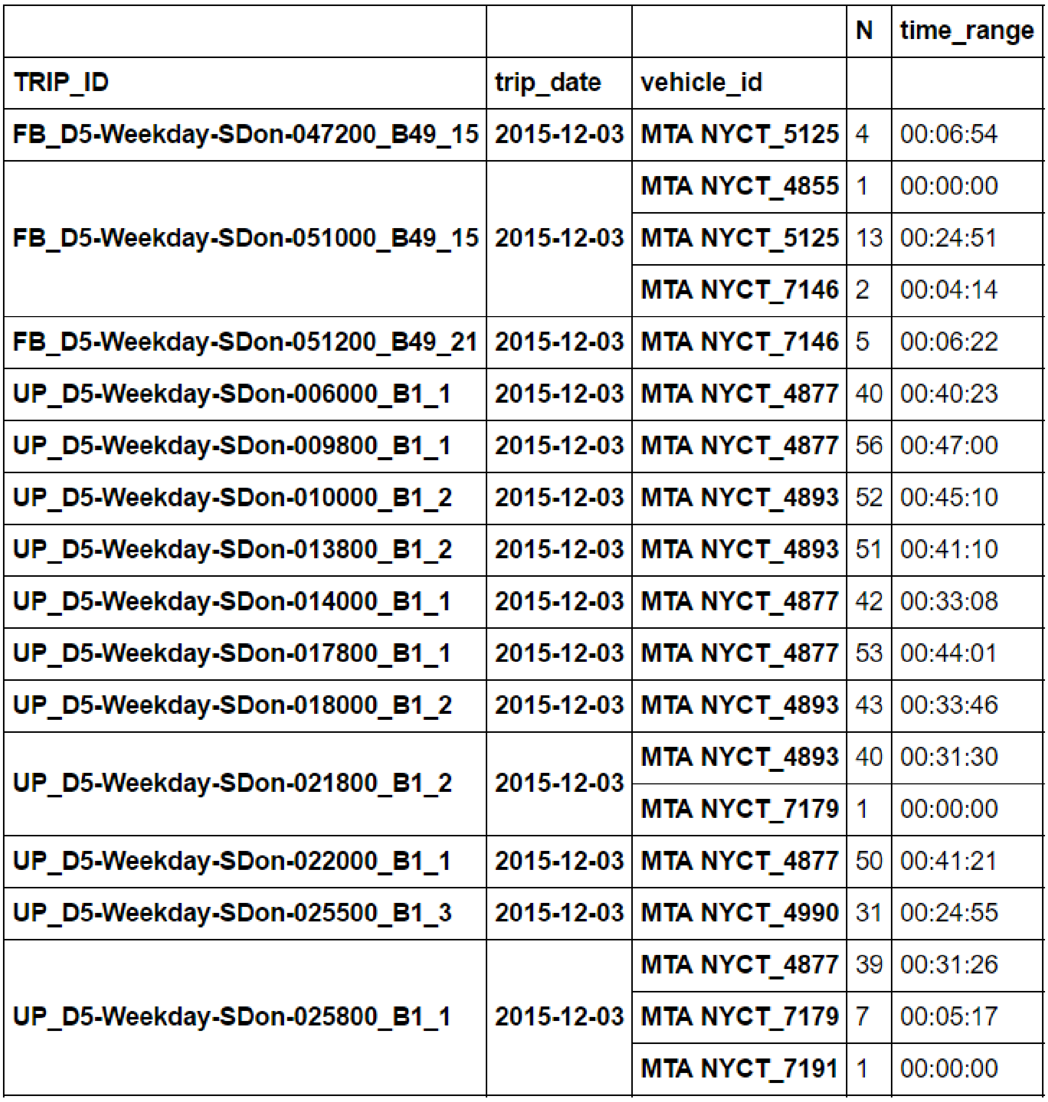
Coordinates estimated using the Global Position System include some level of noise due to factors such as atmospheric conditions and multi-path effects. However the location elements of the Bus Time data appear to be transformed to adhere to shapelines of the reported bus route, eliminating any variation laterally across the street. Figure **??** shows an overlay of points from both Bus Time records and GTFS shape file data, for a short stretch of the B26 route in Fort Greene. While the coordinates of stops are on the sides of the street, no lateral variation is observable in the Bus Time coordinates, even though the vehicles are known to travel along all four lanes of that specific roadway. Some error may remain along the orthogonal plane (the basis for distance calculations), but GPS-assistive equipment installed after the initial launch of Bus Time uses gyroscopes and speedometer readings to reduce this error to an immaterial level.

Figure 6: Sample of B25 route.



The only elements reported by the vehicle itself are the time, location and headsign (the route and destination text displayed above the front of the vehicle). All of the remaining elements are inferred by the Bus Time server. Because those inferences are made real-time, the software has no opportunity to retrospectively validate and correct errors. This analysis validates two key features by checking for two metadata properties: that reported trip IDs are unique to each vehicle and that stop distances are unique for each shape. The first validation failed while the second property was confirmed. Figure **??** is a sample summary of records records grouped by both trip and vehicle, for a sample date and route; it shows the duplication of trip ID across multiple vehicles, the number of records associate with each duplicate, and the timespan of those records.

Figure 7: Duplication of Trip ID across several vehicles



			N	time_range
TRIP_ID	trip_date	vehicle_id		
FB_D5-Weekday-SDon-047200_B49_15	2015-12-03	MTA NYCT_5125	4	00:06:54
FB_D5-Weekday-SDon-051000_B49_15	2015-12-03	MTA NYCT_4855	1	00:00:00
		MTA NYCT_5125	13	00:24:51
		MTA NYCT_7146	2	00:04:14
FB_D5-Weekday-SDon-051200_B49_21	2015-12-03	MTA NYCT_7146	5	00:06:22
UP_D5-Weekday-SDon-006000_B1_1	2015-12-03	MTA NYCT_4877	40	00:40:23
UP_D5-Weekday-SDon-009800_B1_1	2015-12-03	MTA NYCT_4877	56	00:47:00
UP_D5-Weekday-SDon-010000_B1_2	2015-12-03	MTA NYCT_4893	52	00:45:10
UP_D5-Weekday-SDon-013800_B1_2	2015-12-03	MTA NYCT_4893	51	00:41:10
UP_D5-Weekday-SDon-014000_B1_1	2015-12-03	MTA NYCT_4877	42	00:33:08
UP_D5-Weekday-SDon-017800_B1_1	2015-12-03	MTA NYCT_4877	53	00:44:01
UP_D5-Weekday-SDon-018000_B1_2	2015-12-03	MTA NYCT_4893	43	00:33:46
UP_D5-Weekday-SDon-021800_B1_2	2015-12-03	MTA NYCT_4893	40	00:31:30
		MTA NYCT_7179	1	00:00:00
UP_D5-Weekday-SDon-022000_B1_1	2015-12-03	MTA NYCT_4877	50	00:41:21
UP_D5-Weekday-SDon-025500_B1_3	2015-12-03	MTA NYCT_4990	31	00:24:55
UP_D5-Weekday-SDon-025800_B1_1	2015-12-03	MTA NYCT_4877	39	00:31:26
		MTA NYCT_7179	7	00:05:17
		MTA NYCT_7191	1	00:00:00

The conclusions are that the Bus Time server sometimes makes errors in inference of the vehicle's trip ID, the inferred trip ID is not persistent for a vehicle (that is, it "flip flops"), but the reported stop distances for the inferred shape are persistent. The variation in inferred trip ID is an important consideration in subsequent performance measurement because so much of the analysis requires grouping or sorting data (both input and output) by trip.

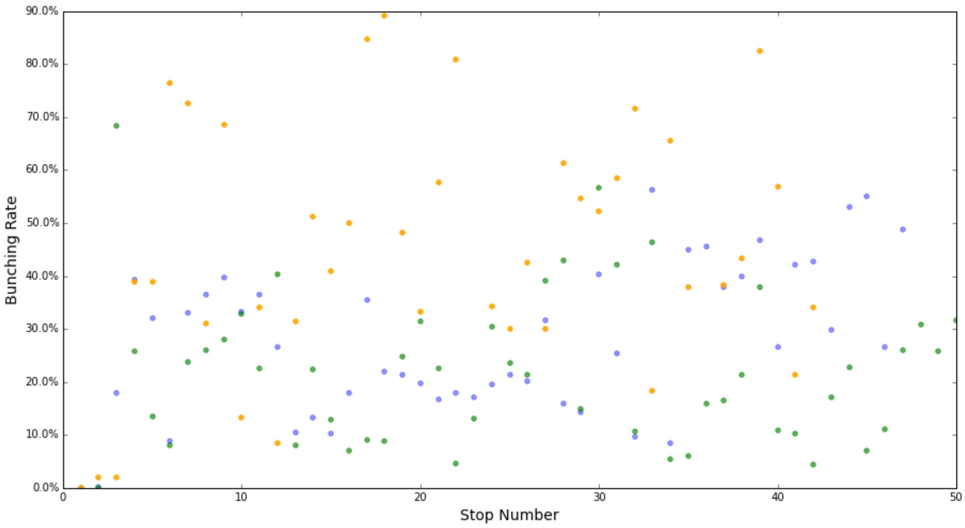
0.4.3 Demonstrations of the methods applied to the dataset

The practices of transportation planning and analysis rely heavily on vehicle arrival-time data. The first phase of this report explained the retrieval of real-time Automated Vehicle Location data and application of methods for estimating those arrival times in actual operations. The following list demonstrates some typical analytic techniques and discusses their validity when applying these data, given the known limitations to its density and accuracy. Included in the input data are arrival times from the schedule, published in the widely-adopted General Transit Feed Specification, but discussion of its data generation process is not in scope. Generally, many high-level performance metrics are simple ratios expressing the proportion of events (such as a vehicle arrival, or completed trip) that meet some criteria (such as arriving within 5 minutes of its scheduled time). The problem with these binary measurements is that the methods ignore the shape (referred to in mathematics as higher moments) of a distribution - for example, a "long tail," or if the distribution is multi-modal.

- **Distribution of headway**: In higher-frequency transit routes, arrivals are considered reliable when the headway is more consistent and closer to customers' expectation. (The ideal distribution is 100% density at the expected value, that is, no deviation. Our definition of Headway Adherence is binary and allows for some deviation). The distribution of headway for a less-than-reliable service is not always a normal bell-curve. This is because of the tendency for delayed buses to become even more delayed, as the larger number of waiting passengers increases dwell times, which in turn reduces overall travel speed. When bunching occurs, the headways of the "bunched" buses (i.e., those that closely follow a preceding delayed bus) approach zero, while there is no theoretical upper limit for the headway of the delayed bus.

- **Bunching rate**: Variations in dwell time and variations between-stop travel time related to traffic and operator behavior are the major causes of vehicle bunching along a route (Gellei 2010). Measurement of dwell time and travel time will be discussed later in this section. The resulting bunching condition can be measured. For this analysis, we consider the bunching condition to occur when headway is less than one minute. The bunching ratio is the percentage, at a certain stop, of arrivals under bunching condition compared to total vehicle arrivals. This is very similar to Headway Adherence except that it measures the tail of the distribution, not the middle. Bunching ratio can be calculated and compared for a variety of stops and routes, as shown in 8.

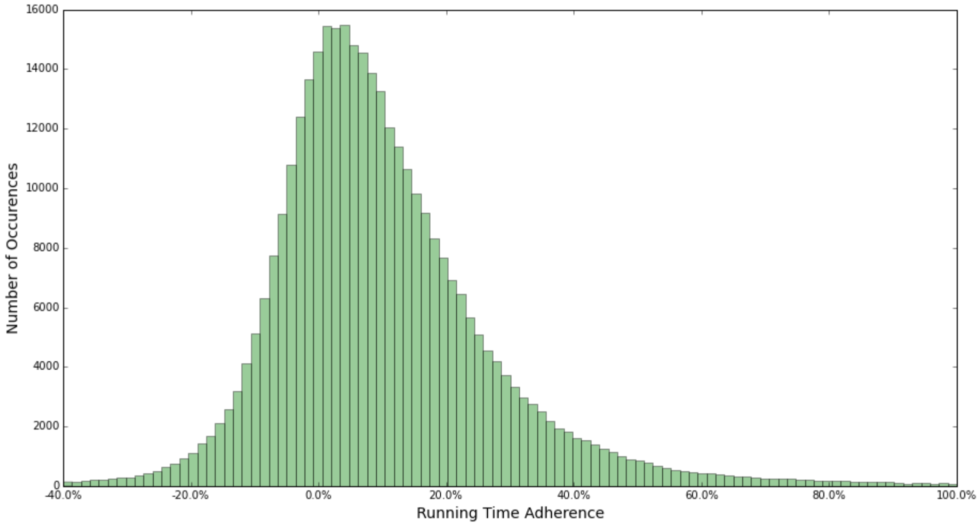
Figure 8: Headway Bunching Rate at Stops for Three Brooklyn Bus Routes.



- **Distribution of running time adherence:** Best practice in schedule planning is to forecast running time using historical data, but exclude outliers. An outlier often represents an occurrence of some enroute incident (such as a police action or a parade) and should not influence planning a typical-day bus schedule. Outliers skew upward the distribution of actual running times, since there is a physical upper limit to the vehicle's speed, but no limit to the number and severity of enroute incidents. Because the schedules are created based on historical distributions excluding outliers, the resulting error, defined as running time variance, will tend to be positive. The error can be mitigated by artificially increasing the planned running time (for example, by including the outliers, or adding some arbitrary value), but it is generally not cost-effective to do so, or impacts the reliability of subsequent trips operated by the same vehicle or operator.

  One example descriptive analysis is to plot the distribution of normalized running time variances. Visible in this example, Figure 9, is both the central tendency – slightly positive – and the wide spread of the distribution, indicating very inconsistent running time adherence. Strategies to reduce the inconsistency are not in the scope of this project.

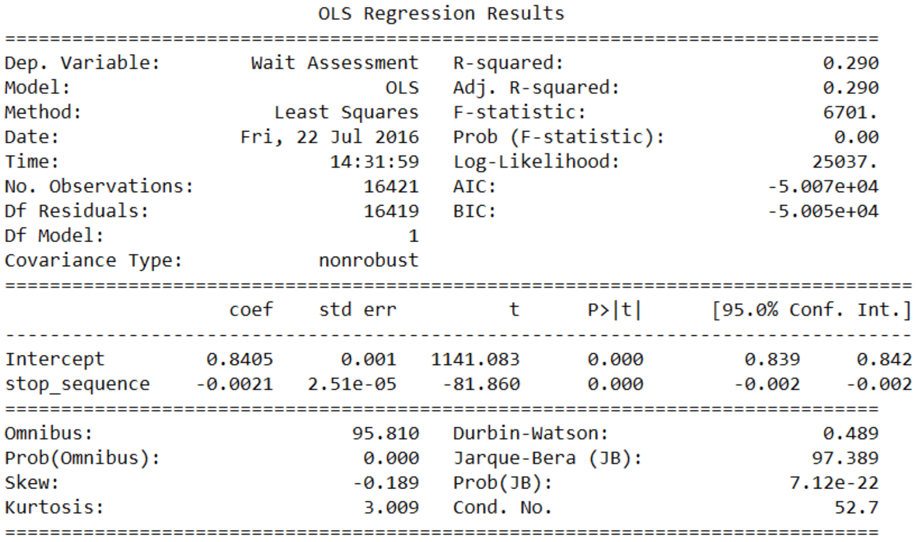
Figure 9: Running time adherence.



- **Performance with respect to vehicle distance along route:** It is generally accepted in the theory of urban public transportation that longer routes have worse reliability, as defined by the metrics discussed so far. Another analytic technique is to plot one of the metrics for a given route (or, more specifically - one shape variation of a route). Figure 10 is the summary of an ordinary least squares (OLS) regression, taking Wait Assessment as the dependent variable and the vehicle's progression along the shape (in terms of number of stops made) as the independent variable. The resulting parameter value has strong statistical significance, rejecting the null hypothesis that route length has no relationship to performance. The example in Figure 11 both supports that conclusion and suggests which segments along the route contribute most to the decline.

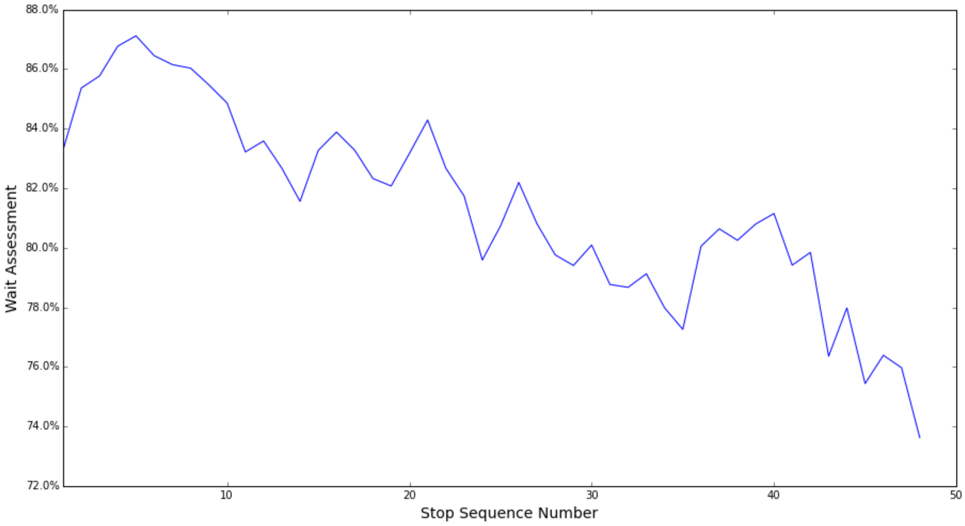
- **Spatial distribution of travel speeds:** Because Bus Time records contain discrete time and location, speed calculations are difference-based averages, not instantaneous (or quasi-instantaneous) samples. The other challenge in descriptive statistics about speed at fixed location(s) is that the sequential observation points of multiple vehicles are not aligned; for example it is extremely unlikely that multiple vehicles record the "ping" from exactly 100 meters and again at exactly 200 meters along a route-shape. However re-sampling is possible if mean speeds are treated as continuous curves with respect to distance. The new distribution at a point is defined as the collection of mean speeds of all vehicles passing that point. 12 demonstrates the changing moments of

Figure 10: Correlation of Wait Assessment and Stop Sequence Number.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Wait Assessment   R-squared:                       0.290
Model:                            OLS   Adj. R-squared:                  0.290
Method:                 Least Squares   F-statistic:                     6701.
Date:                Fri, 22 Jul 2016   Prob (F-statistic):               0.00
Time:                        14:31:59   Log-Likelihood:                 25037.
No. Observations:               16421   AIC:                        -5.007e+04
Df Residuals:                   16419   BIC:                        -5.005e+04
Df Model:                           1
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [95.0% Conf. Int.]
---------------------------------------------------------------------------------
Intercept         0.8405      0.001   1141.083      0.000         0.839     0.842
stop_sequence    -0.0021   2.51e-05    -81.860      0.000        -0.002    -0.002
==============================================================================
Omnibus:                       95.810   Durbin-Watson:                   0.489
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               97.389
Skew:                          -0.189   Prob(JB):                     7.12e-22
Kurtosis:                       3.009   Cond. No.                         52.7
==============================================================================
```

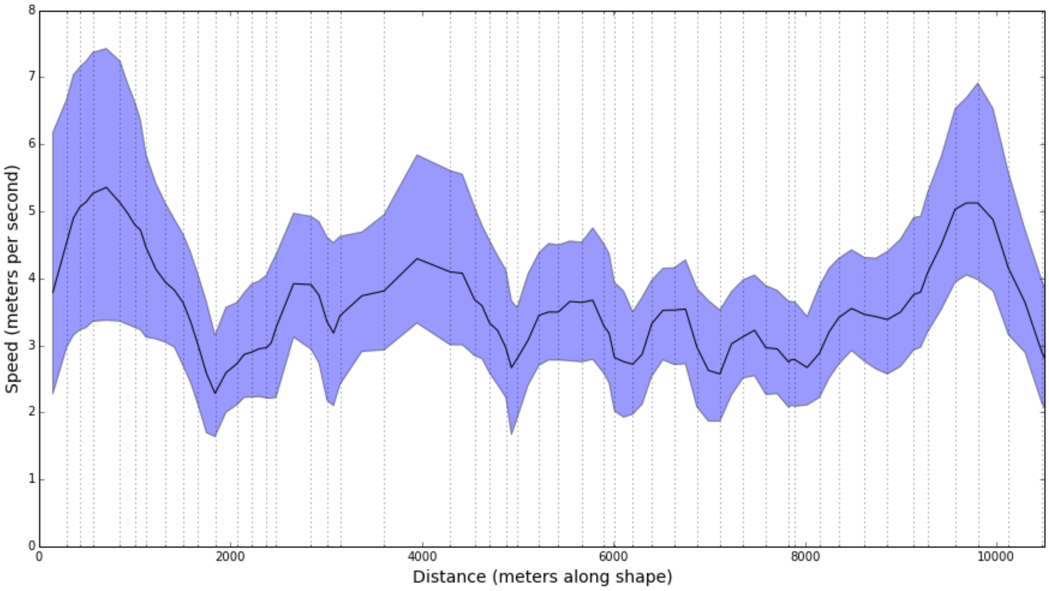
Figure 11: Average Wait Assessment by Stop Sequence Number, Route B41.



the distribution over the length of a route-shape, along with grid lines indicating the stop locations.

- **Spatial distribution of slow/stopped condition** - A list of slow/stopped events can be created by identifying the beginning and end location and times for each occurrence of

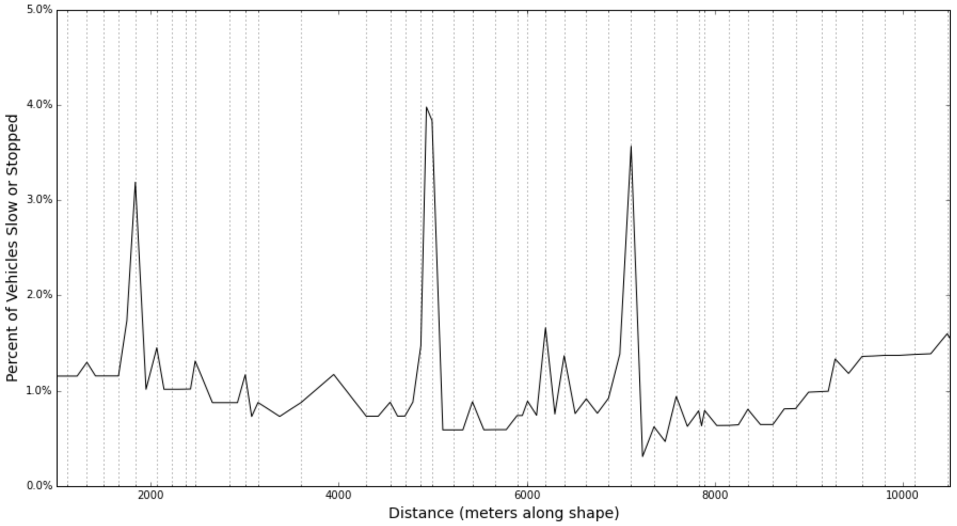
Figure 12: Moving speed distribution.



the condition. Naturally, many of these events will be at stop locations. That subset of the slow/stopped events may theoretically support analysis of dwell time, however even when using data having the maximum density possible according the data generation process (every 30 seconds), analysis of dwell time at individual stops is not possible; previous research suggests that the majority of stops have dwell time of less than 30 seconds (Pangilinan 2011). The remaining subset are slow/stopped events not occurring at stop locations. High spatial density of these events indicates a recurring problem with traffic flow, which in turn increases travel time and contributes to occurrence of bunching condition. (In order to exclude the normal interaction with traffic signals, the events can be filtered to only include those above some minimum, related to traffic signal cycle lengths).

Figure 13 shows the estimated percentage of vehicles in slow/stopped condition and many points along a route-shape. While some extreme values are apparent, they are generally at stop locations and the variance of the remaining points is minimal, even at other stop locations. This suggests that the density of the data may be insufficient to identify interruptions without a high rate of false negatives.

Figure 13: Stopped or Slow Vehicle Percentage.



0.4.4 More discussion on the approach

The main risk identified by our approach is reproducibility in terms of data fetching and of framework dependency. Our data feed depends on a public API that is always subject to interruptions, and the Big Data processing requires the HDFS management system and Spark. It may take time and computational and technical resources to handle them, and deprecation is always a hazard. Despite of this, those potential sources of failure have been proven to be increasing in robustness and trustworthiness during the last years.

The main advantages of our approach rely on the fact that it is based on open source software (such as Python, HDFS and Spark) with a wide and increasing support community around them, but also on the fact that our code is simple to share and reproduce. Since Big Data software is relatively new, we kept most of the data manipulation in simple SQL or Python scripts, which is specially convenient for the client.

The biggest challenge for our approach to be implemented in practice by cities is that of processing large amounts of raw data into information because it requires big data techniques for anything beyond small samples (i.e. one line, one day, etc.). Even processed datasets being structured can demand more than what can be offered by single/dual core applications. Data supporting higher-level analysis techniques can be managed with any off-the-shelf database system, including sqLite (open source SQL) or even a series of CSVs.

It is worth noting that, while the MTA does not publish archived Bus Time data (except for a sample in 2014), we are providing the code for anybody to collect it, and this is an important step that must be kept into account when discussing about privacy. Besides this, the Vehicle ID field is not anonymized, so the contributions of our project are subject to have unlikely yet feasible implications regarding tracking of individuals, to the extent they can be identified in a relatable data source.

0.4.5 Conclusions, recommendations and future work

Conclusions

This project studies the possibility of calculating performance metrics for bus performance in New York City based on the GPS data offered by MTA. The system collects the location of each bus every 30 seconds. After estimating the departure and arrival time for each bus at each stop, we use measurement like headways and wait assessment to assess the bus performance and reliability. Taking one-year data as our object of study, the data is extremely huge so that big data techniques (mainly Spark) are widely used in this project. In order to make our work easy to share and reproduce, we choose to use the SQL API to decide on the parsing of the data, which enables the DOT to easily make changes by editing the SQL script.

Positive aspects of our approach include the fact that we focused this analysis on the viewpoint of the DOT; the fact that we were able to successfully process large volumes of data; the fact that we identified pitfalls and challenges of using the MTA Bus Time data for the purposes of the agency and the fact that we enable a flexible implementation of different methods to estimate times and locations, as well as to measure schedule-reliability performance.

Potential pitfalls and areas of improvement are the fact that we used formal metrics that were not tailored for the DOT nor New York City, and the abundance of assumptions that inevitably scale up in the form of measurement errors.

The influence of our approach relies on the fact that similar methodologies and concepts could be used in other cities and traffic models. The big data technology applied in not only traffic analysis but also city wide urban issue offers more reliable results compared with traditional sample studies.

Recommendations to DOT

Use of the public API for real-time archiving of Bus Time data mimics an ongoing process already performed internally at MTA, and can result in an incomplete view of the same data. Moreover, the static file currently used by DOT, presumably provided by MTA, contains both greater temporal density and some additional, potentially useful elements that are not available through the Bus Time API. The results of this project suggest that combining the advantages of both data acquisition processes would yield the ideal dataset needed for these analyses. The DOT may request that MTA include the additional, useful vehicle monitoring elements in the file provided directly to DOT and therefore eliminate temporal gaps that arise due to problems with the public API.

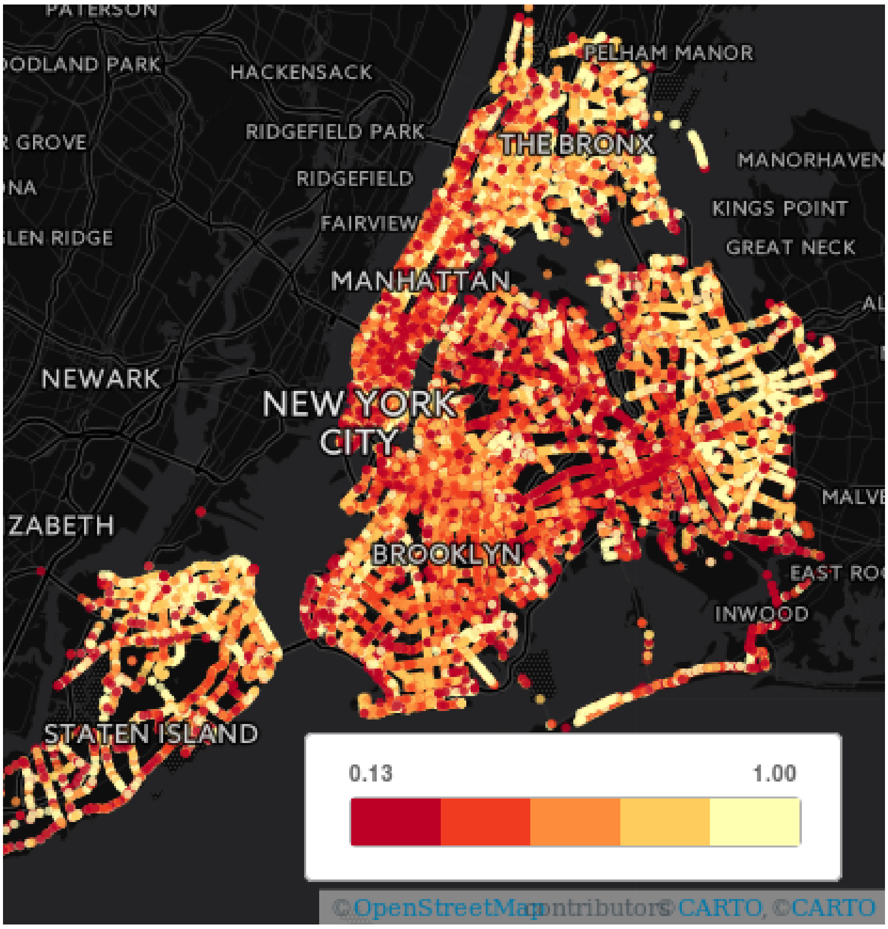
The subsequent interpolation and performance analysis steps do not require the user to implement a so-called big data platform if Bus Time data is processed in smaller batches (for example, one full day) as it accumulates. Once headway, running time, and vehicle speed calculations are performed and output stored with proper indexing, the performance metrics can be dynamically analyzed using a typical personal computer.

Suggestions for future work

- Inference study of the effect of several traffic or design conditions on reliability.
- Autocorrelation to identify recurring patterns in reliability with respect to time (for example, every Monday morning).
- Integrate and improve the current visualization tool developed by NYU CUSP, and adapt it to be responsive and interactive to different queries.
- Further optimize the algorithms hereby used for the Big Data portion of the work.
- Generalization of code for further applications.
  - Other transit system data feeds may use a different interface standard and as a result a different data structure
  - AVL systems in some cities collect stop departure and arrival time directly, instead of location at fixed time intervals.
  - Other transit modes, for example ubway or bike-share, may contain different features specific to that mode.

SAMPLE VISUALIZATION

Figure 14: On time performance aggregated by stop over 2015.



Bibliography

[1] W. H. Lin and J. Zeng, "An experimental study of real-time bus arrival time prediction with GPS data," Transp. Res. Rec., no. 1666, pp. 101- 109, Jan. 1999.

[2] Rajat Rajbhandari, Steven I. Chien, and Janice R. Daniel, ”Estimation of bus dwell times with APC information”, Transportation Research Record 1841 Paper No. 03- 2675, 2003.

[3] Min-Tang Li, Fang Zhao, Lee-Fang Chow, Haitao Zhang, and Shi-Chiang Li, ”Simulation Model for Estimating Bus Dwell Time by Simultaneously Considering Numbers of Disembarking and Boarding Passengers”, Transportation Research Record 1971, 2006.

[4] LI Fazhi, Yang Dongyuan, and Ma Kai, ”Bus Rapid Transit (BRT) Bunching Analysis With Massive GPS Data,” National Science and Technology Support Program of China (NO. 2009BAG17B01), 2013.

[5] Brian Levine, Alex Lu, and Alla Reddy, ”Measuring Subway Service Performance at New York City Transit: A Case Study Using Automated Train Supervision (ATS) Track-Occupancy Data,” TRB 2013 Annual Meeting, 2013.

[6] Dan Wan, Camille Kamga, Jun Liu, Aaron Sugiura, and Eric B. Beaton, ”Rider perception of a âĂŸlight’ Bus Rapid Transit system - The New York City Select Bus Service,” Transport Policy 49 (2016) 41âĂŞ55, Apr. 2016.

[7] Yiming Bie, Xiaolin Gong, and Zhiyuan Liu, ”Time of day intervals partition for bus schedule using GPS data,” Transportation Research Part C 60, pp. 443âĂŞ456, Sep. 2015.

[8] Jeremy S. Safran, Eric B. Beaton, and Robert Thompson, ”Factors Contributing to Bus Lane Obstruction and Usage in NYC: Does Design Matter?” TRB 2014 Annual Meeting, 2014

[9] Christopher Pangilinan, Nigel Wilson, and Angela Moore, ”Bus Supervision Deployment Strategies and Use of Real-Time Automatic Vehicle Location for Improved Bus Service

Reliability,” Transportation Research Record: Journal of the Transportation Research Board, No. 2063, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 28âĂŞ33. DOI: 10.3141/2063-04.

[10] Yingxiang Yang, David Gerstle, Peter Widhalm, and Dietmar Bauer, ”The Potential of Low-Frequency AVL Data for the Monitoring and Control of Bus Performance,” TRB 2013 Annual Meeting, 2013.

[11] Shi An, Xinming Zhang and Jian Wang, ”Finding Causes of Irregular Headways Integrating Data Mining and AHP,” ISPRS Int. J. Geo-Inf. 2015, 4, 2604-2618; DOI:10.3390/ijgi4042604, Nov. 2015.

[12] Jinil Chang, Mohamad Tala, and Satya Muthuswamy, ”A Simple Methodology To Estimate Queue Lengths at Signalized Intersections Using Detector Data,” TRB 2013 Annual Meeting, 2013.

[13] Christopher Pangilinan and Kristen Carnarius, ”Traffic Signal Timing for Optimal Transit Progression in Downtown San Francisco,” San Francisco Municipal Transportation Agency, San Francisco, CA, 2011.

[14] Simon Reed, ”Transport for London - Using Tools, Analytics and Data to Inform Passengers,” JOURNEYS, September 2013. https://www.lta.gov.sg/ltaacademy/doc/13Sep096-Reed_TfL-InformPassengers.pdf,accessed20Jul2016.