* **Data Summaries**

1. **Message From Lind, Eric, 2017-03-29**

The data represent messages for **A line (**[**https://www.metrotransit.org/Route/921**](https://www.metrotransit.org/Route/921)**) from 2016-10-01 to 2016-10-08**. This represents a sampling of normal service within a single version of the schedule. There are **twelve unique vehicles assigned to this route**.

The vehicle messages in our database are actually stored in two separate tables: (1) one for verbose messages about time point crossings and passenger loads (among others), and (2) a second table of “short messages” for simple vehicle location information. A challenge is that the short location messages contain the vehicle ID but not the route it’s working, time points it has crossed, etc.

I have combined these messages into a single file and filled in the data for the short messages where possible by joining on vehicle ID and timestamp (using a “last observation carried forward” method for the short messages). Thus in the attached zipfile is a CSV representing **126,883 messages across the specified week**, in the following types:

Message type N

Vehicle Location 99600

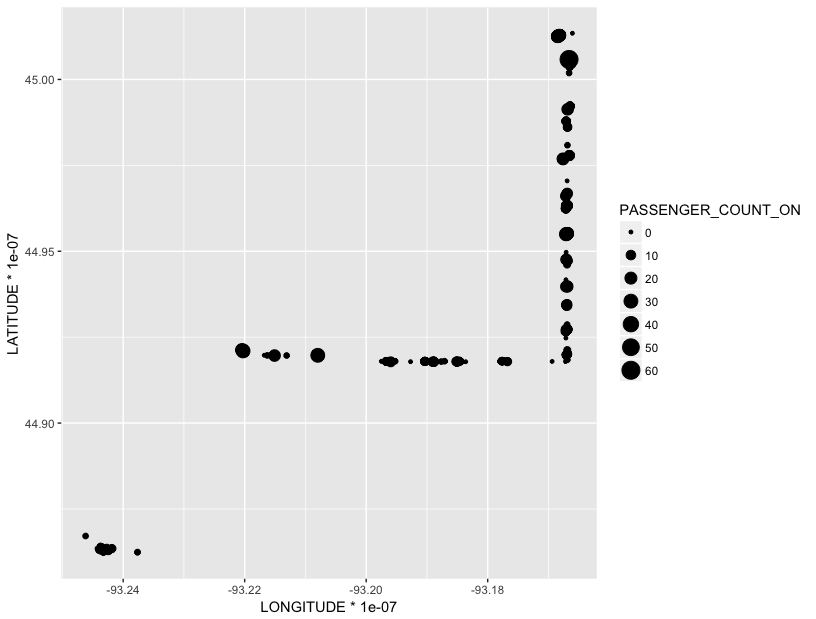
Timepoint Crossing 9804

Verbose Passenger Count 17479

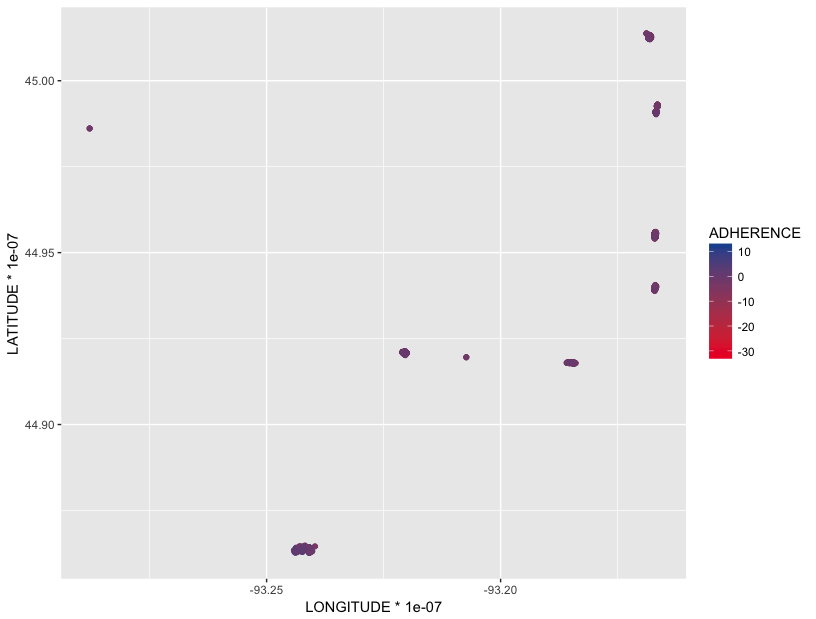
I have also included a metadata file in the zipfile, with the definitions of the fields/columns. In the system, different message types use some columns slightly differently, so there are additional columns in the metadata file to differentiate these fields. In this case **“APC\_Fields” refers to messages from the Automatic Passenger Counter** and **“TPC\_fields” refers to messages sent at Time Point Crossings.** Where different information is included in the field depending on the message type, I have put “OVERLOADED” in the field definition.

Finally, just some quick visuals to share, that might help you see what these data represent.

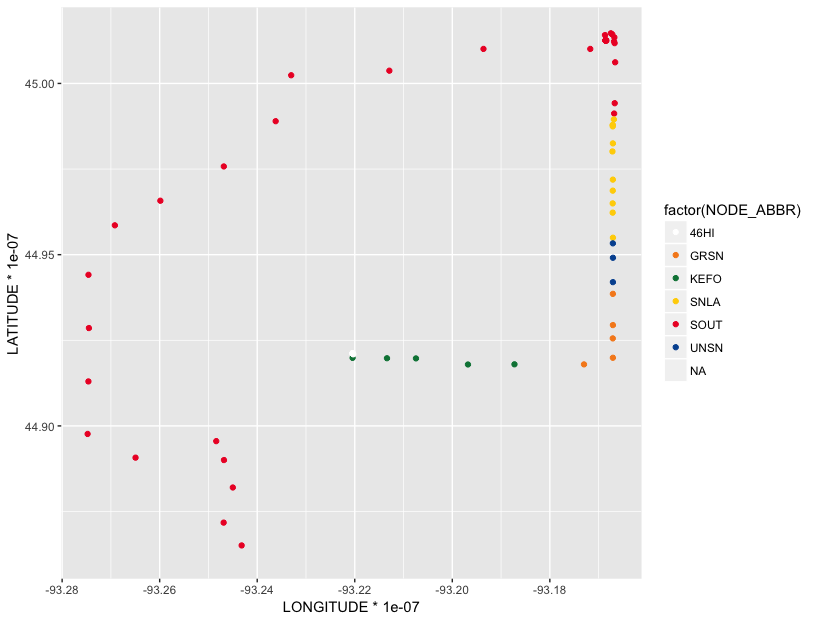
First: the records of boarding according to the Passenger Count messages - note that this includes South Garage where the Operators begin their work and trigger a boarding message.



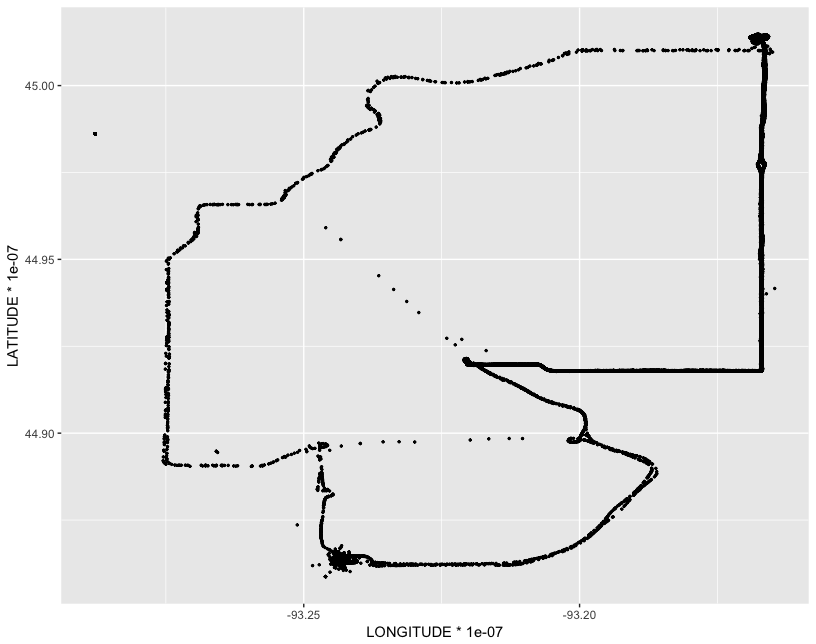
Second, a similar map but showing time point crossings - these are what the system uses to calculate adherence to the schedule. Again you can see there is a time point crossing at the garage(s) at the beginning of the trip.



Third, an example of some short location messages, grouped by the last time point the vehicle has passed:



Finally, a plot of \*all\* the location messages in this dataset - you can see that these locations include the deadheads travelled to begin service and return to the garage.



1. **Online Dataset**
2. **Shapefile from Minnesota Geospatial Commons**

* **Transit Routes Shapefile**

<https://gisdata.mn.gov/dataset/us-mn-state-metc-trans-transit-routes>

* **Transit Trip Count Headway by Route**

<https://gisdata.mn.gov/dataset/us-mn-state-metc-trans-transit-count-headway-rt>

* **Transit Stops Shapefile**

<https://gisdata.mn.gov/dataset/us-mn-state-metc-trans-transit-stops>

* **Transit Schedule csv**

<https://gisdata.mn.gov/dataset/us-mn-state-metc-trans-transit-schedule-google-fd>

<https://developers.google.com/transit/gtfs/reference/?csw=1>

1. **AADT GIS Shapefile**

<http://www.dot.state.mn.us/traffic/data/data-products.html#volume>

<https://svc.metrotransit.org/mtgtfs/archive/gtfs20201128.zip>

* **Processing Notes**

1. **Process data for the use as input for analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **COLUMN\_NAME** | **Description** | **IS\_NULLABLE** | **DATA\_TYPE** | **TPC\_Field** | **APC\_Field** |  |
| TRANSMITTED\_MESSAGE\_ID | The unique, system-assigned identifier | NO | bigint |  |  | 1 |
| **CALENDAR\_ID** | **The unique, system-assigned identifier for service day in format "1YYYYMMDD"** | **YES** | **int** |  |  | **2** |
| MESSAGE\_TYPE\_ID | The unique, system-assigned identifier for vehicle message type (MESSAGE\_TYPE\_TEXT) | NO | smallint |  |  | 0 |
| **LATITUDE** | **The MDT will cast the** [**PLGR**](https://www.ion.org/museum/files/PLGR%20brochure.pdf) **latitude and longitude to a long integer and multiply the values by 10,000,000. (i.e. Latitude = ( long ) ( ( lat ) \* 10,000,000 ).** | **YES** | **int** |  |  | **3** |
| **LONGITUDE** | **The MDT will cast the PLGR latitude and longitude to a long integer and multiply the values by 10,000,000. (i.e. Latitude = ( long ) ( ( lat ) \* 10,000,000 ). <WGS84>** | **YES** | **int** |  |  | **4** |
| ADHERENCE | Differential between scheduled and actual time, in minutes (negative values are behind schedule) | YES | smallint |  |  | 5 |
| ODOMETER | An unsigned integer that is multiplied by 100 at the MDT. Mileage is in hundredths of miles. It is set to zero on MDT startup or reboot, so the count restarts each day. Unit of measure is .01 miles. | YES | int |  |  | 6 |
| VALIDITY | multi-bit field used to determine GPS validity; see FOM, dGPS, ValidOdo, ValidAdh, ValidLoc | YES | smallint |  |  | 7 |
| SOURCE\_CLASS | V = vehicle | YES | varchar |  |  | 8 |
| **SOURCE\_HOST** | **identifier for the** **vehicle radio transmitter; unique to vehicle** | **YES** | **smallint** |  |  | **9** |
| DESTINATION\_CLASS | not used | YES | varchar |  |  | 10 |
| DESTINATION\_HOST | not used | YES | smallint |  |  | 11 |
| ROUTE\_VERSION | OVERLOADED | YES | int | log.m\_BlockOffsetID | ex.m\_First\_door\_open | 12 |
| MESSAGES\_VERSION | OVERLOADED | YES | int | log.m\_DailyBlockID | ex.m\_Last\_door\_closed | 13 |
| ROUTE\_OFFSET | system-assigned unique identifier for Route | YES | smallint |  |  | 14 |
| **DIRECTION** | **Cardinal direction of travel:** **1 = Southbound, 4 - Northbound** | **YES** | **tinyint** |  |  | **15** |
| TIME\_POINT\_OFFSET | system-assigned unique identifier for Time Point data | YES | smallint |  |  | 16 |
| STOP\_OFFSET | system-assigned unique identifier for Stop data | YES | smallint |  |  | 17 |
| FLAG32 | OVERLOADED | YES | int | ex.m\_ArrivalTime |  | 18 |
| EFFECTIVE\_SERVICE | OVERLOADED | YES | smallint | msg.logonService | log.m\_Service | 19 |
| MDT\_TIMESTAMP | OVERLOADED - Time in seconds past midnight | YES | int | ex.m\_DepartureTime | ex.m\_Arrival\_time | 20 |
| **MESSAGE\_TIMESTAMP** | **Time of message in UTC (Note it thus does not always align with CalendarID)** | **YES** | **datetime** |  |  | **21** |
| **PASSENGER\_COUNT\_ON** | **Count of passengers boarding** | **YES** | **tinyint** |  | **ex.m\_TotalOn** | **22** |
| **PASSENGER\_COUNT\_OFF** | **Count of passengers alighting** | **YES** | **tinyint** |  | **ex.m\_TotalOff** | **23** |
| ST\_MDT\_VERSION | OVERLOADED | YES | smallint | msg.dwell |  | 24 |
| SYSPARAM\_FLAG | not used | YES | int |  |  | 25 |
| MSG\_GROUP | OVERLOADED | YES | int | msg.blockOffsetID | log.m\_bFromStoredFile | 26 |
| CAT\_1 | OVERLOADED | YES | tinyint | msg.exceptionComboL | msg.category1 | 27 |
| CAT\_2 | OVERLOADED | YES | tinyint | msg.exceptionComboH | msg.category2 | 28 |
| CAT\_3 | OVERLOADED | YES | tinyint | msg.isArrival | msg.category3 | 29 |
| CAT\_4 | OVERLOADED | YES | tinyint |  | msg.category4 | 30 |
| CAT\_5 | OVERLOADED | YES | tinyint |  | msg.category5 | 31 |
| CAT\_6 | OVERLOADED | YES | tinyint |  | msg.category6 | 32 |
| CAT\_7 | OVERLOADED | YES | tinyint | log.m\_bFromStoredFile | msg.category7 | 33 |
| CAT\_8 | OVERLOADED | YES | tinyint |  | msg.category8 | 34 |
| CAT\_9 | OVERLOADED | YES | tinyint |  | msg.category9 | 35 |
| CAT\_10 | OVERLOADED | YES | tinyint |  | msg.category10 | 36 |
| LOWER32 | OVERLOADED | YES | int |  | log.m\_DailyWorkPieceID | 37 |
| UPPER32 | OVERLOADED | YES | int | msg.blockOffsetID | ex.m\_Departure\_time | 38 |
| CURRENT\_DRIVER |  | YES | int |  |  | 39 |
| **FREE\_TEXT\_MSG** | **For APC messages, boardings and alightings** | **YES** | **varchar** |  | **ex.m\_OnOffsByDoor** | **40** |
| LOCAL\_TIMESTAMP | Timestamp of message in local time | YES | datetime |  |  | 41 |
| MDT\_BLOCK\_ID | system-assigned unique Identifier for Schedule Block | YES | int |  |  | 42 |
| SIGNAL\_STRENGTH | Radio signal strength value | YES | tinyint |  |  | 43 |
| BLOCK\_ABBR | Lookup number of schedule block | YES | varchar |  |  | 44 |
| ROUTE\_ABBR | Human-readable Route ID | YES | varchar |  |  | 45 |
| PROPERTY\_TAG | unique identification printed on vehicle | YES | varchar |  |  | 46 |
| SERVICE\_ABBR | Type of scheduled service: WK = Weekday; SAT = Saturday' SUN = Sunday | YES | varchar |  |  | 47 |
| NODE\_ABBR | Human-readable Bus stop code | YES | varchar |  |  | 48 |
| **FOM** | **Figure of Merit of GPS reception. Range 2 - 14 with 2 meaning highest GPS accuracy.** | **YES** | **tinyint** |  |  | **49** |
| **dGPS** | **Vehicle using differential navigation** | **YES** | **bit** |  |  | **50** |
| **ValidOdo** | **Vehicle providing valid odometer reading** | **YES** | **bit** |  |  | **51** |
| **ValidAdh** | **Vehicle providing valid Adherence values** | **YES** | **bit** |  |  | **52** |
| **ValidLoc** | **Vehicle providing valid Location information** | **YES** | **bit** |  |  | **53** |
| MESSAGE\_TYPE\_TEXT | Human-readable type of message. One of Vehicle Location, Timepoint Crossing (TPC) or Verbose Pass Count (APC) | YES | varchar |  |  | 54 |

1. **Python Classes & Packages**

* **Double linked lists <Modules/DoubleLinkedList.py>**
* Used for stop\_times and stops in GTFS
* <stop\_id> could be the node\_key)
* **Pytz <package for different time zones>**

Link: <https://pypi.python.org/pypi/pytz>

* **Pygtfs <package for handling GTFS data>**

Link: <http://pygtfs.readthedocs.io/en/latest/>

<https://github.com/graphhopper/jsprit>

* **Geopy <package for spatial points>**

Link: <https://github.com/geopy/geopy>

<http://geopy.readthedocs.io/en/latest/index.html?highlight=WGS84>

1. **ArcGIS Map Matching**
2. Select GTFS data for Aline (S00\_SelectGTFSforRoutes.py)

* **Output**
* Routes, stop-times, stops, trips

1. Separate data into files by vehicle radio transmitter (S01\_SeperateAVLByRadioID.py)

* **Total number of records**: 126,884
* **Selected attributes**

# select columns for processing: latitude, longitude, vehicle radio transmitter ID,

# service day <1YYYMMDD>, timestamp UTC <datetime>

# trip direction 1 = Southbound, 4 = Northbound,

# count of passenger boarding, count of passenger alighting

* **Output**

12 csv files

* **Note**
* The service day is not consistent with the timestamp UTC
* Some of the date in timestamp UTC is out of range (09-21-2016).

They are excluded from analysis <from the sorted set starting with S01\_RadioID>

['2016-09-21 15:36:31.417000-05:00', '44.9273019', '-93.167032' …]

['2016-09-21 15:36:31.323000-05:00', '44.9549581', '-93.1670761'…]

1. Get moving distance, average speed, and time difference between two consecutive GPS points

* Some of the date and times repeat

We take the second one for calculation, and use -9999.99990 as the default for speed and time differences (see S03\_LogFile for detailed information)

Note: These are just approximate numbers based on raw data. After linear referencing, these numbers will be updated.

1. Linear Reference AVL GPS Points and Transit Stops along Routes

* **Note**
* The time difference cannot be the way to separate the data, this may be due to the fact that radio is not required to be turned off after service, and may be turned on “arbitrarily”.
* **The direction can be used in combination with the time difference (600 seconds = 10 minutes) to get the initial separation of the trips.**
* When vehicle went to the garage, they may turn on and off the radio at irregular pattern
* In the next step, first remove points when staying at the garage, and select the one starting and ending at the first TP of each direction.

**Create trip id by RadioID\_Day\_Trip, and add sequence number to each point.**

**<These are the results before map-matching the points>**

* **The distance and GPS coordinates will be used to determine dwelling locations or stops.**
* The threshold distance is 20 meters for linear referencing GPS points
* Also, linear reference the map-matched the stops
* These two linear reference files are used to separate GPS points into segments.

1. Get speed and distance between two LR GPS points

* **Type of buses** used by metro transit: <https://www.metrotransit.org/our-vehicles>
* 40-feet standard diesel buses (12 meters)
* 20 meters around the bus stops will be considered as stopping
* **Speed limits**
* 40 meter per second, around 90 mph for local road should be sufficient enough to filter traces

[Updated 20170817]

Weather data: http://www.weather.gov/mpx/mspclimate

Jul 9-16, 2016 (Summer, Sunny days, after national day)

Oct 1-8, 2016 (Autumn, Sunny days, regular week)

Nov. 5-12, 2016 (Winter, Sunny days, before holiday)

Jan. 7-14, 2017 (Winter, Snow days, after holiday)

Apr. 1-8 2017 (Spring, Sunny days, regular week)

[Updated 20180415]

[0] TRANSMITTED\_MESSAGE\_ID

[2] LOGGED\_MESSAGE\_SHORT\_ID

[3] CALENDAR\_ID

[4] LM\_CALENDAR\_ID\_ORIG

[5] MESSAGE\_TYPE\_ID

**[6] LATITUDE**

**[7] LONGITUDE**

[8] ADHERENCE

[9] ODOMETER

[10] VALIDITY

[11] MESSAGE\_TIMESTAMP

[12] LOCAL\_TIMESTAMP

[13] SOURCE\_CLASS

[14] SOURCE\_HOST

[15] DESTINATION\_CLASS

[16] DESTINATION\_HOST

[17] ROUTE\_VERSION

[18] SIGNAL\_STRENGTH

[19] ETL\_VERSION

[20] LM\_BLOCK\_ABBR

[21] LM\_ROUTE\_ABBR

[22] LM\_ROUTE\_DIRECTION\_ABBR

[23] LM\_PROPERTY\_TAG

[24] SCH\_TRIP\_SERIAL\_NO

[25] SCH\_PATTERN\_NO

[26] SCH\_PREV\_TIME\_POINT\_ABBR

[27] SCH\_NEXT\_TIME\_POINT\_ABBR

[28] MESSAGES\_VERSION

[29] ROUTE\_OFFSET

[30] DIRECTION

[31] TIME\_POINT\_OFFSET

[32] STOP\_OFFSET

[33] FLAG32 EFFECTIVE\_SERVICE

[34] MDT\_TIMESTAMP

[35] PASSENGER\_COUNT\_ON

[36] PASSENGER\_COUNT\_OFF

[37] ST\_MDT\_VERSION

[38] SYSPARAM\_FLAG

[39] MSG\_GROUP

[40] LOWER32

[41] UPPER32

[42] CURRENT\_DRIVER

[43] FREE\_TEXT\_MSG

[44] MDT\_BLOCK\_ID

[45] TIME\_TABLE\_VERSION\_ID

[46] SERVICE\_TYPE\_ID

[47] CAT\_1

[48] CAT\_2

[49] CAT\_3

[50] CAT\_4

[51] CAT\_5

[52] CAT\_6

[53] CAT\_7

[54] CAT\_8

[55] CAT\_9

[56] CAT\_10

[57] REVISED\_ROUTE\_ABBR

[58] REVISED\_ROUTE\_DIRECTION\_ABBR

[59] REVISED\_PROPERTY\_TAG

[60] SCH\_TRIP\_START\_TIME

[61] SCH\_TRIP\_END\_TIME

[62] LM\_TIMEPOINT\_ABBR

[63] PROCESSED\_TIMESTAMP

[64] REVISED\_LOCAL\_TIMESTAMP

[65] REVISED\_MESSAGE\_TIMESTAMP

[66] PROCESSED\_FLAG

Message Data – not linked to any trip?