DATA ANALYSIS

TAXI

HELEN WIND

BACKGROUND INFORMATION

- Data acquisition: kaggle.com
- Scenario: Preparing data for business analyst
- Data limitations:
  - Data for training purposes: obvious flaws/contradictions in data
  - Data ‘gathered’ in 1 day
  - Missing data

DATA CLEANING

	TRIP_ID	CALL_TYPE	ORIGIN_CALL	ORIGIN_STAND	TAXI_ID	TIMESTAMP	DAY_TYPE	MISSING_DATA	POLYLINE
0	1372636858620000589	C	NaN	NaN	20000589	1372636858	A	False	[[-8.618643,41.141412], [-8.618499,41.141376],[...
1	1372637303620000596	B	NaN	7.0	20000596	1372637303	A	False	[[-8.639847,41.159826], [-8.640351,41.159871],[...
2	1372636951620000320	C	NaN	NaN	20000320	1372636951	A	False	[[-8.612964,41.140359], [-8.613378,41.14035],[-...

RAW DATA

DATA ENRICHMENT + AGGREGATION

- Splitting + removing columns:
  - Splitting and removing ‘timestamp’: adding ‘date’ and ‘start_time’
- Adding new columns:
  - ‘starting_point’ and ‘ending_point’ based on first and last coordinate of ‘polyline’
  - ‘distance’ : Pythagoras theorem with ‘starting_point’ and ‘ending_point’
  - ‘duration’ : number of coordinates * 15
  - ‘speed’ : distance / time (‘duration)
  - ‘end_time’ : ‘start_time’ + ‘duration’
- Reorder column order

DATA CLEANING + VALIDATION

- Removed 3 duplicate rows
- Edit missing data:
- Calculate speed and distance with data from other columns
- Ensure data is valid:
  - e.g. ‘call_type’ are either ‘A’, ‘B’, or ‘C’
- Edit data types:
  - Convert ‘trip_id’ and ‘taxi_id’ into string

```
Total missing value by column:

trip_id                  0
call_type                0
origin_call        1345900
origin_stand        904091
taxi_id                  0
day_type                 0
start_time               0
end_time                 0
date                     0
duration                 0
speed                36510
polyline                 0
starting_point           0
ending_point             0
distance             36510
missing_data             0
dtype: int64
```

DATA CLEANING

	trip_id	call_type	origin_call	origin_stand	taxi_id	day_type	start_time	end_time	date	duration	speed	polyline	starting_point	ending_point	distance	missing_data
0	1372636858620000589	C	NaN	NaN	20000589	A	00:00:01.372636	00:11:16.372636	1970-01-01	675	0.000026	[[-8.618643,41.141412], [-8.618499,41.141376], [...	[-8.618643,41.141412]	[-8.630838,41.154489]	0.017881	False
1	1372637303620000596	B	NaN	7.0	20000596	A	00:00:01.372637	00:09:16.372637	1970-01-01	555	0.000051	[[-8.639847,41.159826], [-8.640351,41.159871], [...	[-8.639847,41.159826]	[-8.66574,41.170671]	0.028072	False
2	1372636951620000320	C	NaN	NaN	20000320	A	00:00:01.372636	00:32:16.372636	1970-01-01	1935	0.000002	[[-8.612964,41.140359], [-8.613378,41.14035], [-...	[-8.612964,41.140359]	[-8.61597,41.14053]	0.003011	False

CLEANED DATA

DATA CLEANING

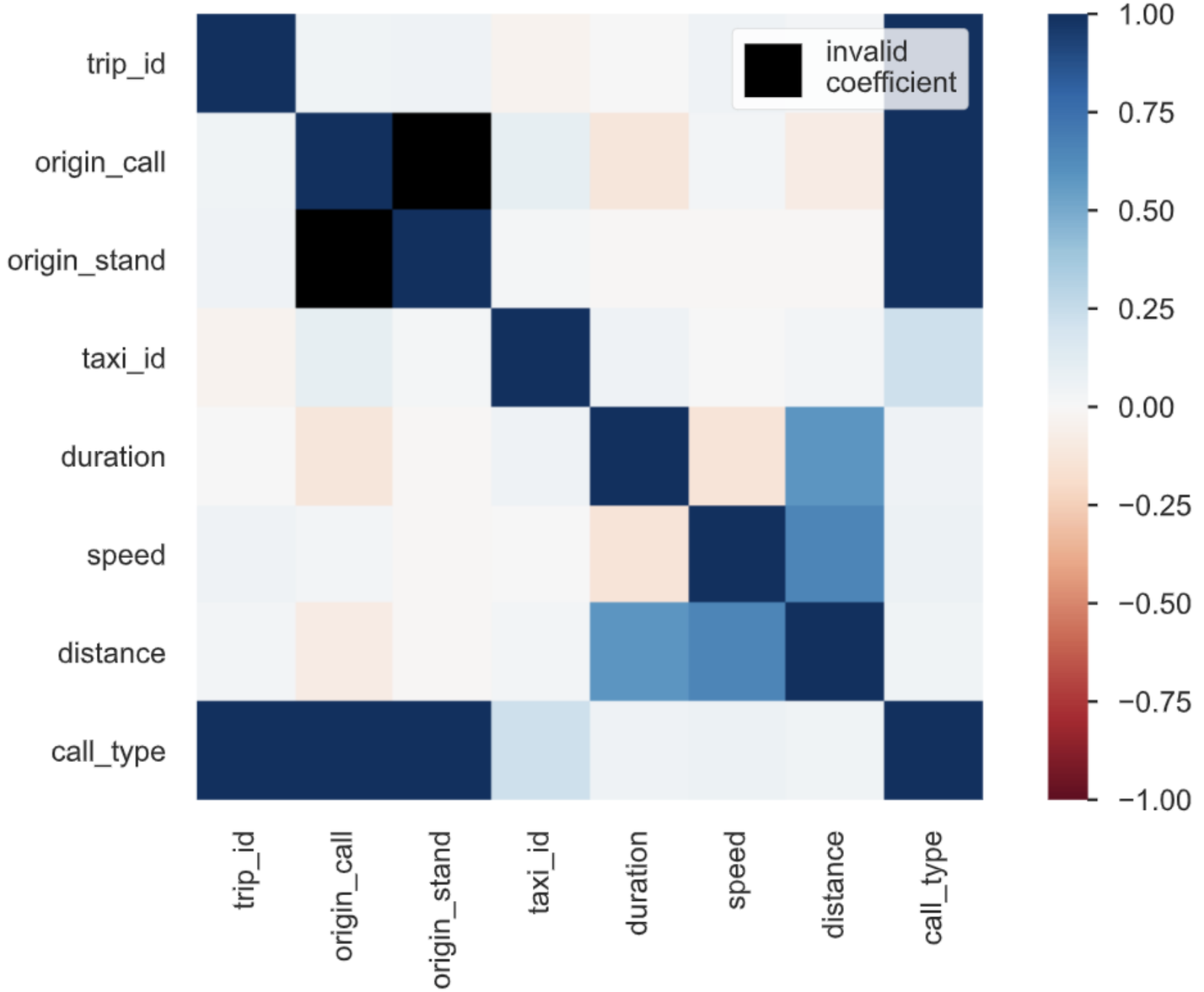
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1710670 entries, 0 to 1710669
Data columns (total 9 columns):
 #   Column        Dtype
---  ------        -----
 0   TRIP_ID       int64
 1   CALL_TYPE     object
 2   ORIGIN_CALL   float64
 3   ORIGIN_STAND  float64
 4   TAXI_ID       int64
 5   TIMESTAMP     int64
 6   DAY_TYPE      object
 7   MISSING_DATA  bool
 8   POLYLINE      object
dtypes: bool(1), float64(2), int64(3), object(3)
memory usage: 106.0+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1710670 entries, 0 to 1710669
Data columns (total 16 columns):
 #   Column          Dtype
---  ------          -----
 0   trip_id         object
 1   call_type       object
 2   origin_call     float64
 3   origin_stand    float64
 4   taxi_id         object
 5   day_type        object
 6   start_time      object
 7   end_time        object
 8   date            object
 9   duration        int64
 10  speed           float64
 11  polyline        object
 12  starting_point  object
 13  ending_point    object
 14  distance        float64
 15  missing_data    bool
dtypes: bool(1), float64(4), int64(1), object(10)
memory usage: 197.4+ MB
```

RAW VS CLEAN

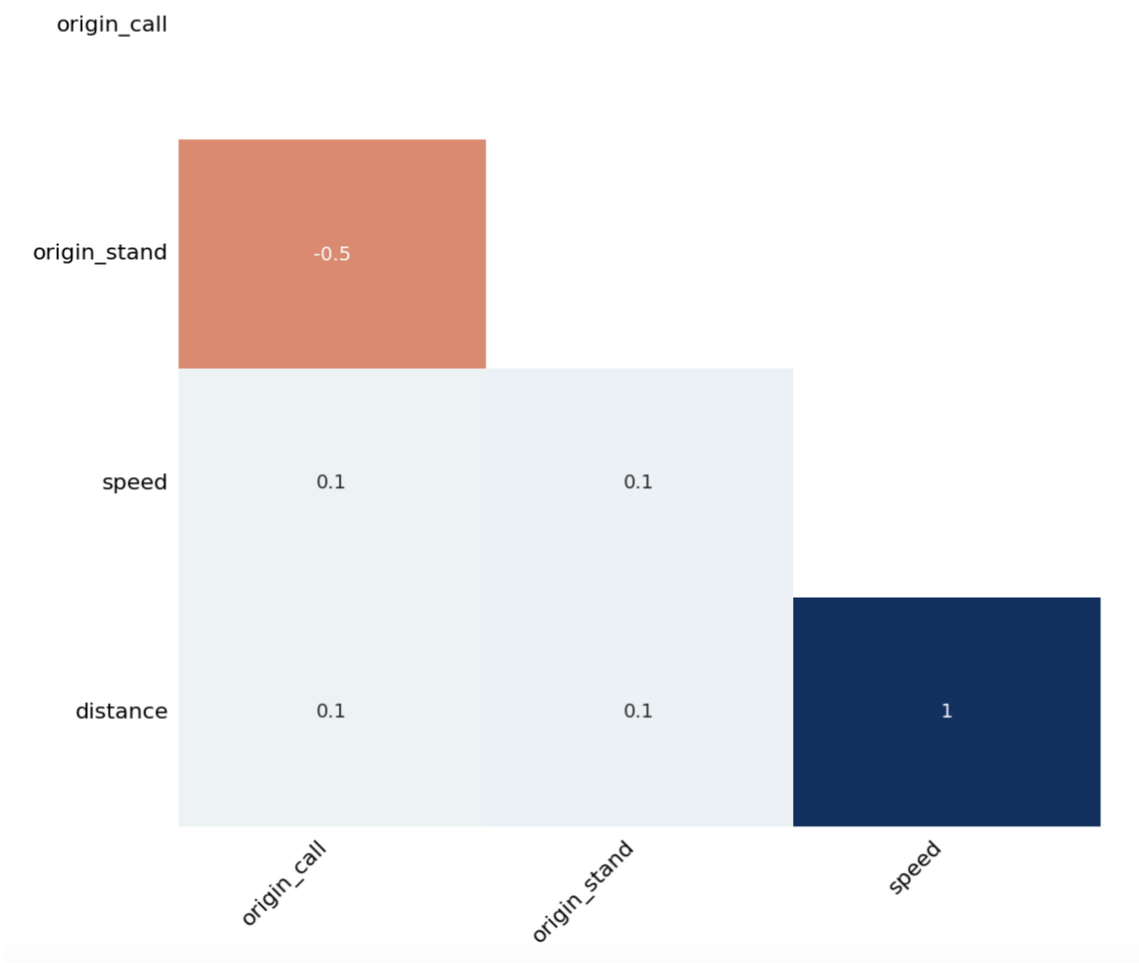
PROFILE REPORT

- Look at full report on : taxi_profile_report.html
- Use 1000 random rows
- Strong correlation between distance, duration, speed



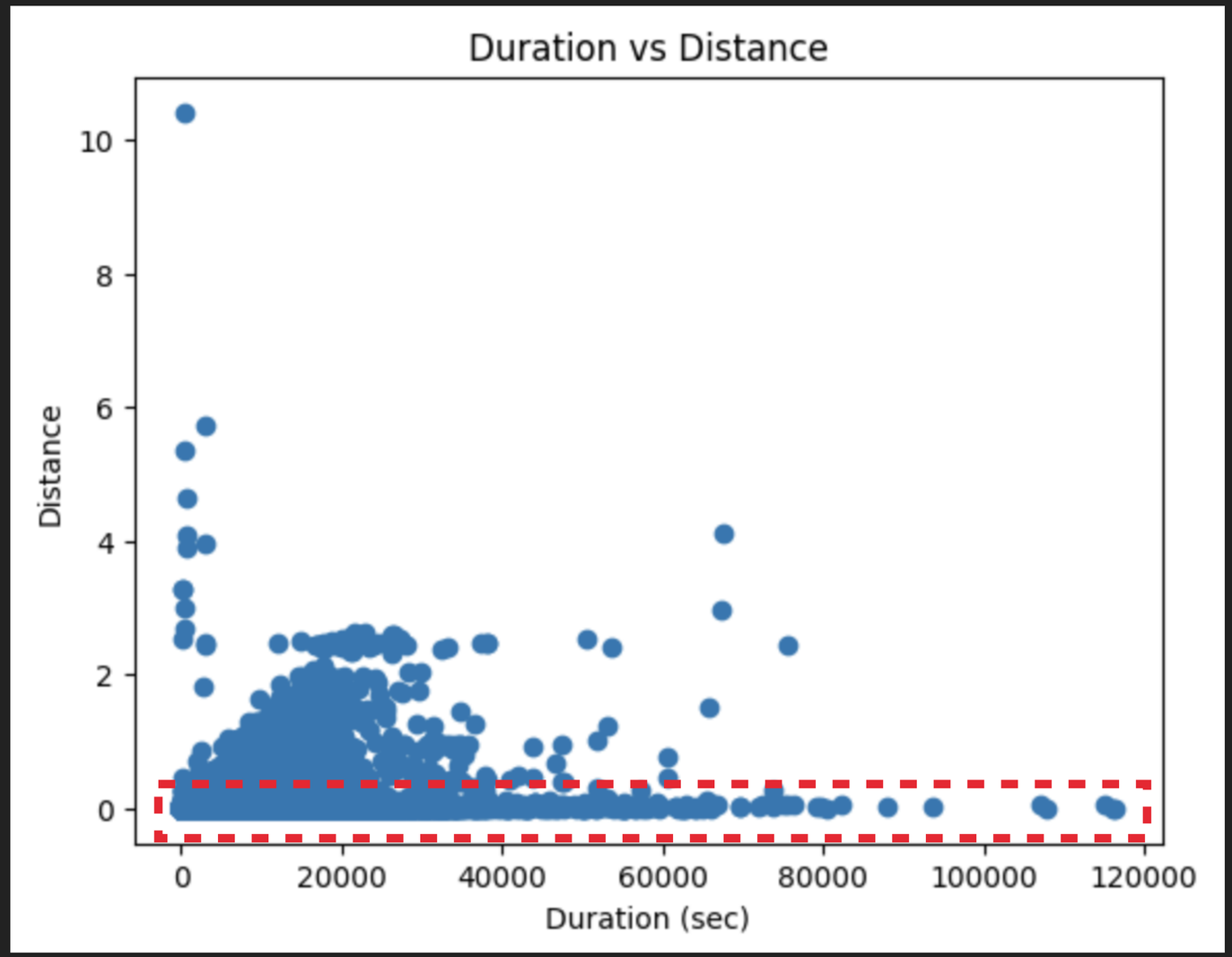
HEAT MAP

- Shows correlation between two columns
- String positive correlation with speed and distance
  - Expected outcome as speed is calculated using distance
- Negative correlation with origin_stand and origin_call
  - origin_stand is Null if origin_call is ‘B’



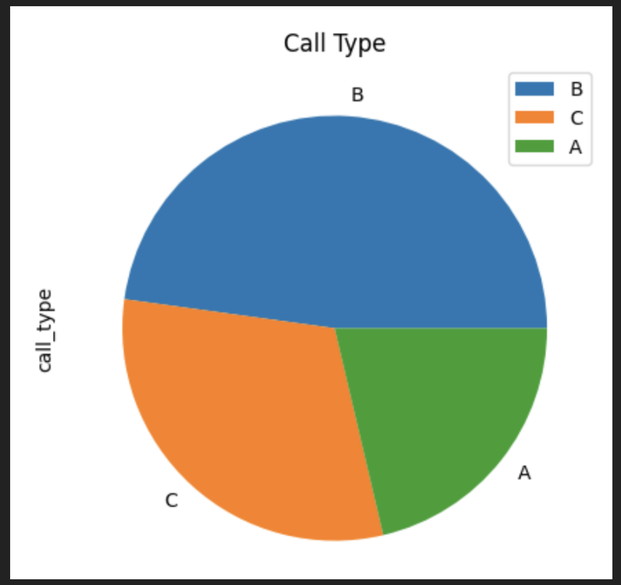
DISTANCE VS DURATION

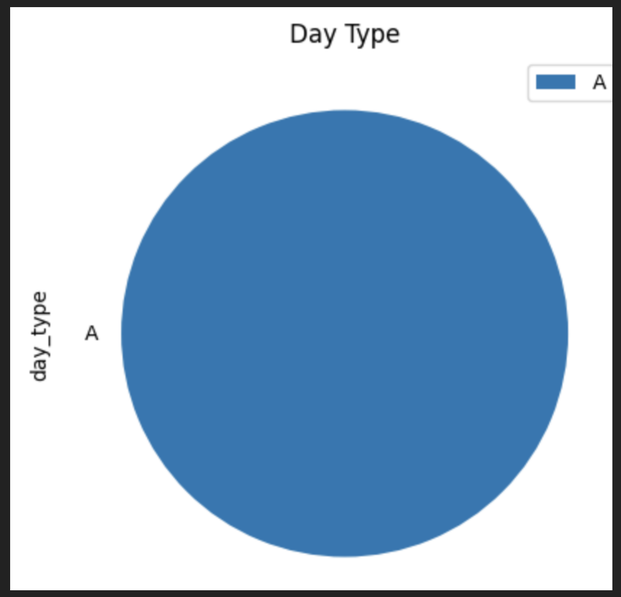
- Distance and duration should have a strong correlation, it takes more time the longer you travel
- Flawed data: high duration while distance travelled is 0. And vice versa though this appears less often



CALL TYPE + DAY TYPE

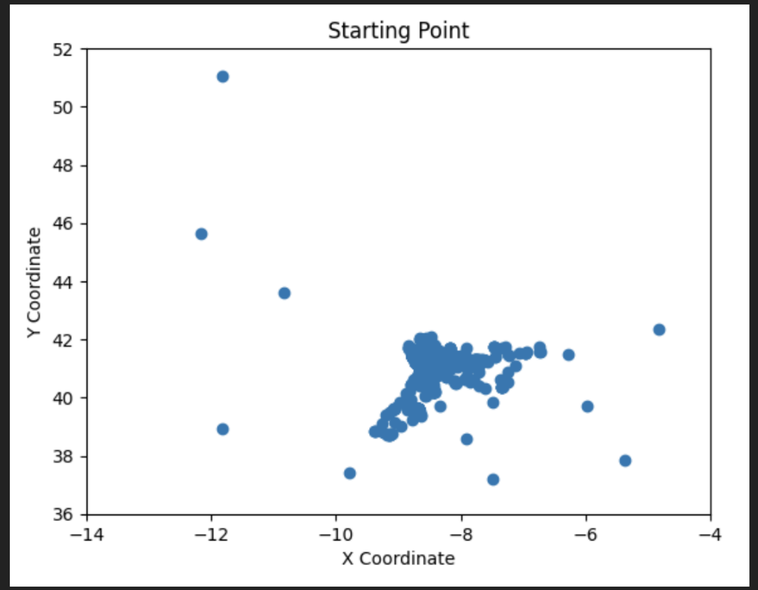
- Call type:
  - Type ‘B’ is the most popular with almost half of occurrences being ‘B’.
  - Type ‘A’ is the least popular
- Day type:
  - Since data occurred on the same day, 100% of day type is A

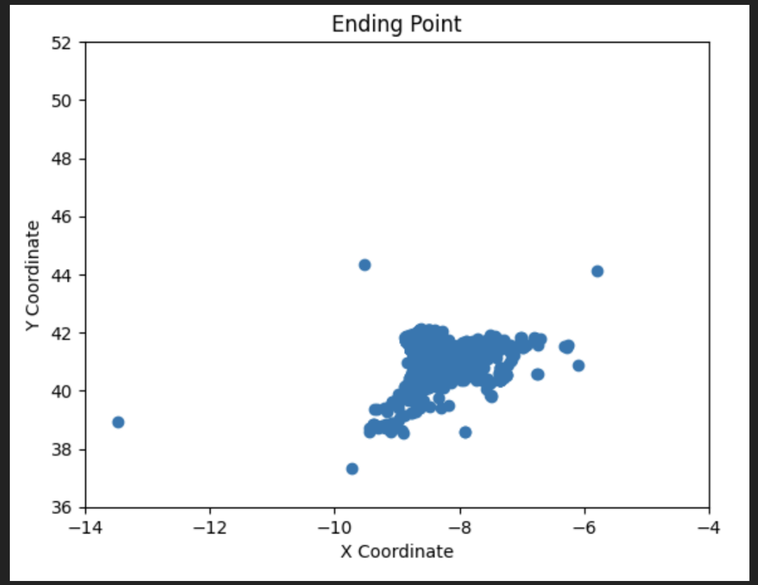




STARTING AND ENDING POINTS

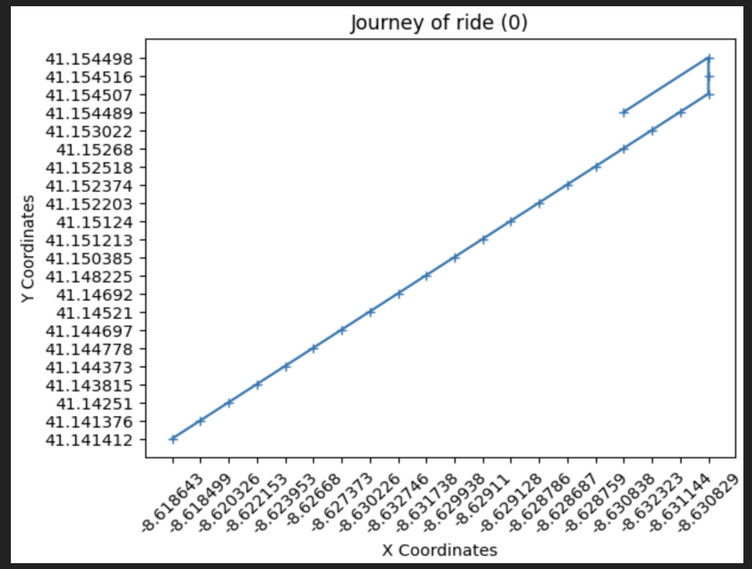
- Data limitations:
  - Coordinated aren’t accurate (located in the ocean)
  - Can’t apply to real-world situation
- Starting and ending points are located in a similar area with a couple of outliers.
- Difficult to extrapolate further analysis due to limited information and data source

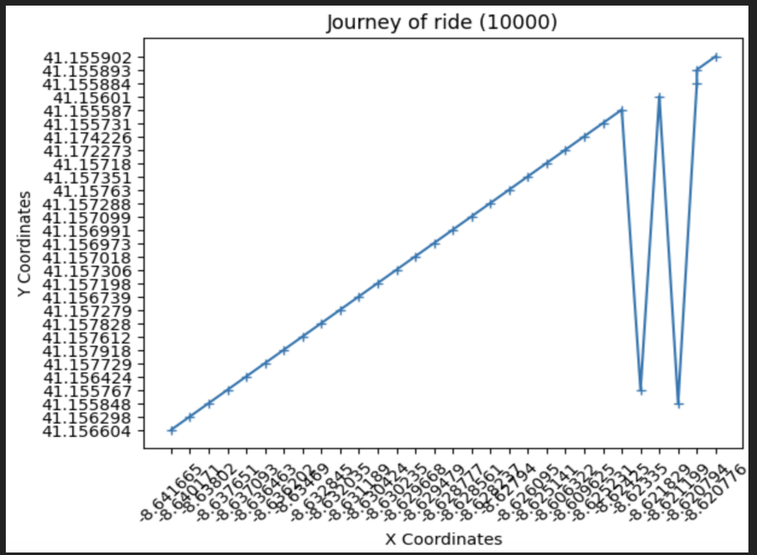




TRIP JOURNEY

- Data limitations:
  - Journeys are nonsensical
  - Can’t apply to real-world situation
- Picture right: 2 ‘journey’ from randomly selected rows
- Strong correlation with X and Y coordinate with erratic movements
- Further analysis of journey route will not result in relevant information





CONCLUSION

- Extrapolated addiontal information based on raw data:
  - Distance, duration, speed, starting point, ending point, etc.
- Initial data exploration conducted:
  - Profile report, plotting graphs
- Future considerations:
  - Further analysis on ‘polyline’ column: distance and duration
  - Create additional data frames for taxi and customer information”
    - Merging / linking dataframes
    - Track taxi and customer history, day_type on different days, etc