Data Exploration of NYC Subway Dataset

Ivailo Kassamakov

October 2015

Abstract

The present project is an effort towards fulfilling the requirements of the Data Analyst Nanodegree at Udacity.

The goal is to perform exploratory data analysis on a dataset collected from readings of weather sensors and turnstile counters situated at the stations of the NYC subway.

All calculations and graphics have been done within an IPython notebook (an indispensable part of this project), with the `pandas`, `statmodels` and `matplotlib` packages.

Ideas, explanations and guidelines for using Pandas and (I)Python have been taken from [2], [4], and [6].

Some useful tools used during the work on this project were: www.sharelatex.com, www.tablesgenerator.com, www.ottobib.com, and www.maptiler.org.

Contents

List of Figures

List of Tables

1 Introduction

The input dataset comes as a CSV file containing 42649 records. Each record consist of 27 columns, and their meaning is given in table 1.

No.	Column name	Column Description
1	UNIT	Remote unit that collects turnstile information. Can collect from multiple banks of turnstiles. Large subway stations can have more than one unit.
2	DATEn	Date in "yyyy-mm-dd" (2011-05-21) format.
3	TIMEn	Time in "hh:mm:ss" (08:05:02) format.
4	ENTRIESn	Raw reading of cumulative turnstile entries from the remote unit. Occasionally resets to 0.
5	EXITSn	Raw reading of cumulative turnstile exits from the remote unit. Occasionally resets to 0.
6	ENTRIESn_hourly	Difference in ENTRIES from the previous REGULAR reading.
7	EXITSn_hourly	Difference in EXITS from the previous REGULAR reading.
8	datetime	Date and time in "yyyy-mm-dd hh:mm:ss" format (2011-05-01 00:00:00). Can be parsed into a Pandas `datetime` object without modifications.
9	hour	Hour of the timestamp from TIMEn. Truncated rather than rounded.
10	day_week	Integer (0–6 Mon–Sun) corresponding to the day of the week.
11	weekday	Indicator (0 or 1) if the date is a weekday (Mon–Fri).
12	station	Subway station corresponding to the remote unit.
13	latitude	Latitude of the subway station corresponding to the remote unit.
14	longitude	Longitude of the subway station corresponding to the remote unit.
15	conds	Categorical variable of the weather conditions (Clear, Cloudy etc.) for the time and location.
16	fog	Indicator (0 or 1) if there was fog at the time and location.
17	precipi	Precipitation in inches at the time and location.
18	pressurei	Barometric pressure in inches Hg at the time and location.
19	rain	Indicator (0 or 1) if rain occurred within the calendar day at the location.
20	tempi	Temperature in °F at the time and location.
21	wspdi	Wind speed in mph at the time and location.
22	meanprecipi	Daily average of `precipi` for the location.
23	meanpressurei	Daily average of `pressurei` for the location.
24	meantempi	Daily average of `tempi` for the location.
25	meanwspdi	Daily average of `wspdi` for the location.

(Continues on next page)

(Table Cont'd)

No.	Column name	Column Description
26	weather_lat	Latitude of the weather station the weather data is from.
27	weather_lon	Longitude of the weather station the weather data is from.

Table 1: Meaning of the columns in the input dataset

2 Some information about the NYC subway

Looking at the input dataset we can glean the following information about the NYC subway:

- There are 207 subway stations.
- There are 240 UNITs, indicating that some stations have more than one unit (*multi-unit* stations).
- There are 28 multi-unit stations, as given in table 2
- The dataset has been collected during the period 5/1/11–5/31/11, at times 00:00:00, 04:00:00, 08:00:00, 12:00:00, 16:00:00, and 20:00:00. For most, but not all, UNITs data has been collected for each of the 31 days of the period. For the days when data has been collected, this has been done for all 6 time points. The UNITs which miss data for some days of the period are: R228, R253, R260, R273, R295, R356, R453, and R459.

Station	Number of UNITs
111 ST	2
125 ST	2
145 ST	3
167 ST	2
18 AVE	2
23 ST-6 AVE	2
25 ST	2
34 ST-HERALD SQ	2
34 ST-PENN STA	3
42 ST-TIMES SQ	2
50 ST	3
86 ST	3
ATLANTIC AVE	2
BOWLING GREEN	2
CHAMBERS ST	2
CHURCH AVE	2
DEKALB AVE	2
FORDHAM ROAD	2
GRAND ST	2
HOYT ST	2
JAY ST-METROTEC	2
LEXINGTON AVE	3
LEXINGTON-53 ST	2
NOSTRAND AVE	2
PROSPECT AVE	2
RECTOR ST	2
SPRING ST	2
WALL ST	2

Table 2: Multi-unit subway stations

3 Input dataset correctness

To judge the quality of the input dataset, we will check it for missing values in the columns, as well as evaluate the range of data in each column.

A quick check with `df.isnull().any().any()` for missing/NaN values shows that all columns of the input dataset contain **valid** values.

The `describe()` function of the input DataFrame returns the data ranges of the numeric columns, as shown in Table 3. A simple visual check confirms that the data are within their expected ranges, and there are **no outliers**.

Description	Count	min	max
ENTRIESn	42649	0	23577460
EXITSn	42649	0	14937820
ENTRIESn_hourly	42649	0	32814
EXITSn_hourly	42649	0	34828
hour	42649	0	20
day_week	42649	0	6
weekday	42649	0	1
latitude	42649	40.576152	40.88918
longitude	42649	-74.073622	-73.75538
fog	42649	0	1
precipi	42649	0	0.3
pressurei	42649	29.55	30.32
rain	42649	0	1
tempi	42649	46.90	86.00
wspdi	42649	0	23.00
meanprecipi	42649	0	0.1575
meanpressurei	42649	29.59	30.29333
meantempi	42649	49.40	79.80
meanwspdi	42649	0	17.08333
weather_lat	42649	40.600204	40.86206
weather_lon	42649	-74.014870	-73.69418

Table 3: Description of the input dataset

4 Visualizing the subway stations

Figure 1 shows a map of the subway UNITs, according to the geographical coordinates contained in the input set. The map was prepared using the information in [11] and [1]. All the details of the map production can be seen in the accompanying IPython notebook.

5 Meteorology

5.1 Meteorologic stations and their regions of responsibility

The weather at the the subway UNITs is measured by meteo stations, located in the UNITs' vicinity. There are 37 meteorologic stations. A given meteo station can service one or more UNITs, and each UNIT is serviced by exactly one meteo station.

Although not required for this project, an effort was made to visualize also the location of the meteo stations. This was done together with calculating a theoretical **region of responsibility** for each station. The idea of this region is to determine which subway UNITs should be serviced by which meteo station. This calculation was done by performing a Voronoi tessellation of the mesh formed by the meteo stations. Figure 2 shows these regions in a color-coded way, as per [8]. Additionally, it shows again the subway UNITs, this time color-coding them with the color of their servicing meteo station (as reported by the dataset). We see that most UNITs (not all!) have the same color as the Voronoi region in which they are located—this means that the Voronoi tessellation was a really good way to determine the extents of the area that is best served by a given meteo station.

5.2 Meteorologic conditions

Using the Haversine geodesic formula (details in the IPython notebook),

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_2)\cos(\phi_1)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

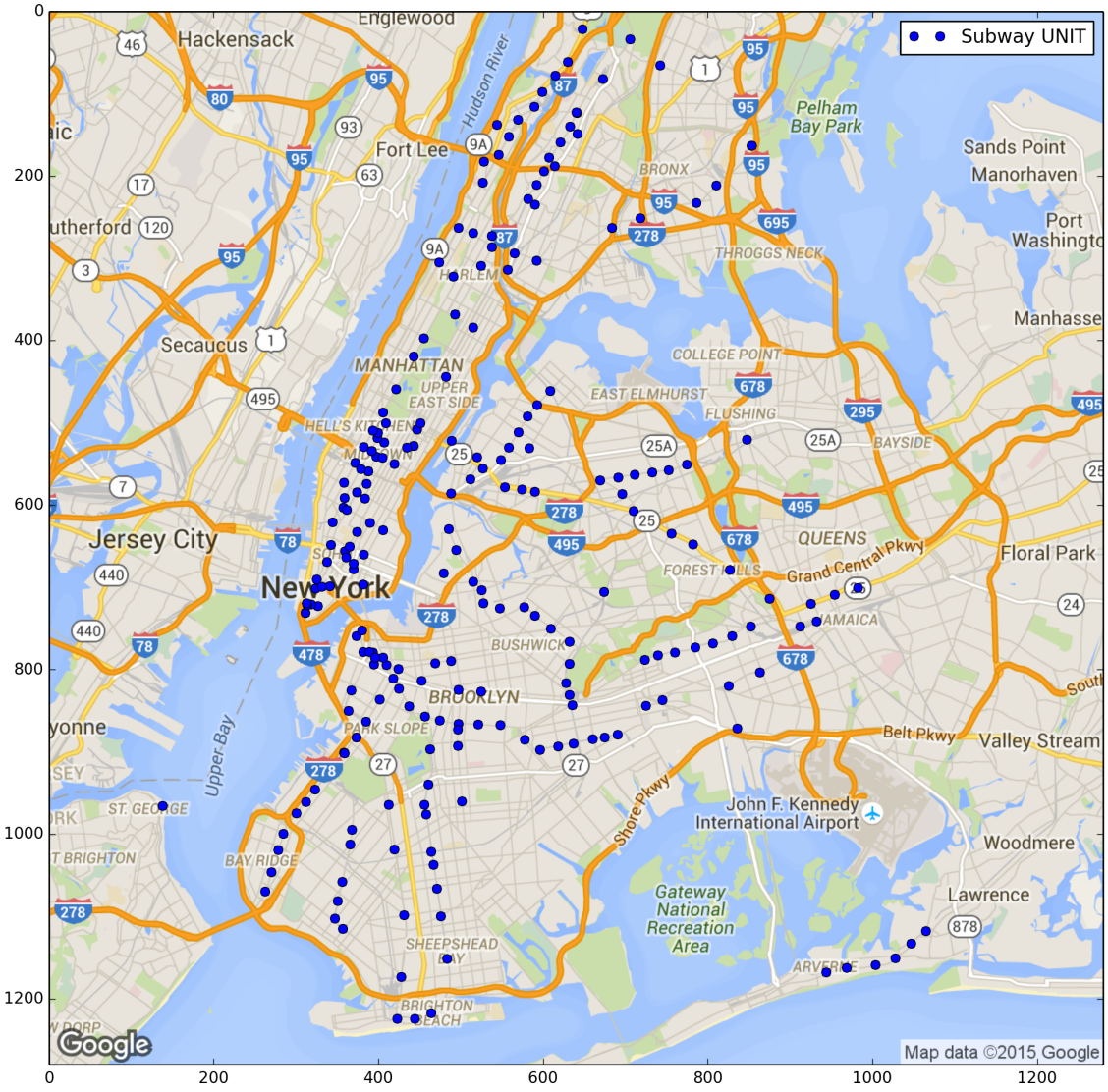

Figure 1: Map of NYC subway stations

we find the **extents of the NYC subway area** to be approximately 35×35 km. In this pretty big area we can expect to find different meteorologic conditions at different places at the same time.

Indeed, after performing some exploration with `pandas` of the initial dataset, we discover the following:

- It is possible for different subway stations to have different weather on the same day—so we cannot always talk about completely rainy or completely dry days for the whole subway.
- Of the 31 days in the dataset, there are 9 days when we have had some subway stations receiving rain, while others not.
- Of the remaining 22 days, during 21 of them the weather has been 100% dry for all stations, while on 1 day (2011-05-17) we have had rain over 100% of the subway.
- Of the 9 "mixed weather" days, on '2011-05-19' we have had the most diversified rain conditions, i.e. almost half of the subway stations have been reporting rain, while the rest have been "dry". A map showing this can be seen in Figure 3.

6 Predicting ridership

6.1 Brainstorming factors that can affect ridership

Predicting the volume of ridership is not an easy task—there are simply too many known and unknown variables that can affect it. By intuition, we would expect the hourly number of entries into the subway to depend on factors like:

- **Time of the day**—Usually the morning hours will see more entries in stations located in residential regions; in the evening hours there should be more entries in business areas stations.

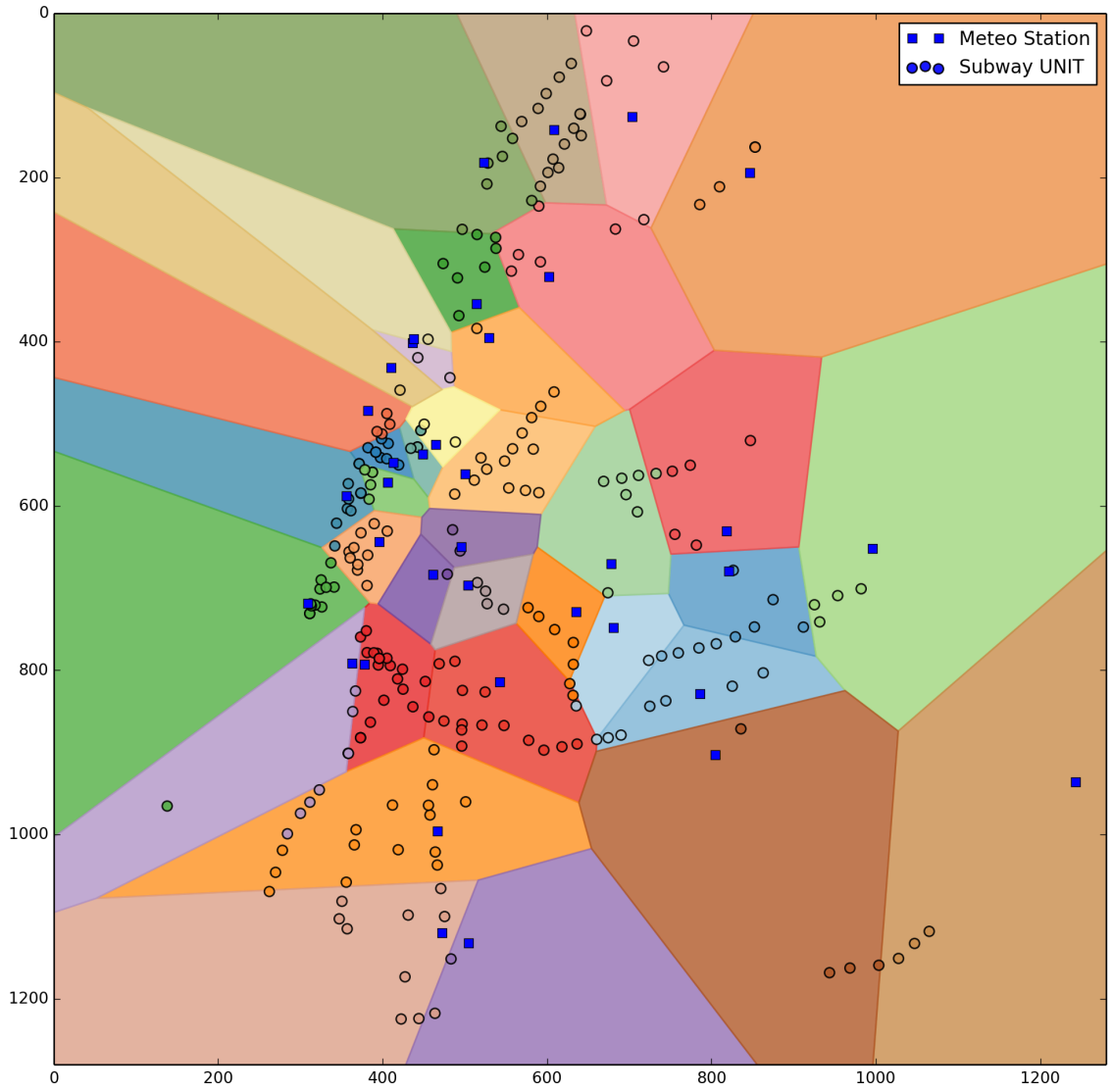

Figure 2: Map of NYC meteo stations, their Voronoi-calculated regions of responsibility and subway UNITs

- **Day of the week**—Probably during the weekend there will be less subway entries.
- **Location**—Stations located at transportation hubs, for example, probably will have more entries.
- **Meteorologic conditions**—Do people prefer to take more the subway when it rains or is cold?
- **State of the subway network**—Accidents, bottlenecks, jams or maintenance work somewhere on the subway network can reduce (or eliminate) the entries in some places and increase them in others.
- **State of the other transportation networks (e.g. roads, buses and trains)**—All networks exhibit some kind of interdependence. Therefore problems on other networks (e.g. strikes of bus drivers, closed streets, etc.) can have profound effect on the subway usage.
- **Significant public events**—Demonstrations, U.N. assemblies, big sport events, ticker-tape parades—all this can also significantly increase (or, on the contrary, block) the subway entries.

Figure 4 shows barplots of the average entries as a function of some of the above factors; namely: time-of-day, day-of-week, weekend-or-weekday, and rain conditions. We see a really high dependence of the ridership on the time of the day and the day of the week, but much less pronounced dependence on the presence/absence of rain. We can draw the following conclusions:

- The weekends manifest significantly lower subway entries than the weekdays. This is an indication that the subway is heavily used for work commuting.
- The highest number of subway entries are registered from the morning until after 20:00h—this corresponds to the period of highest economic and entertainment activity.

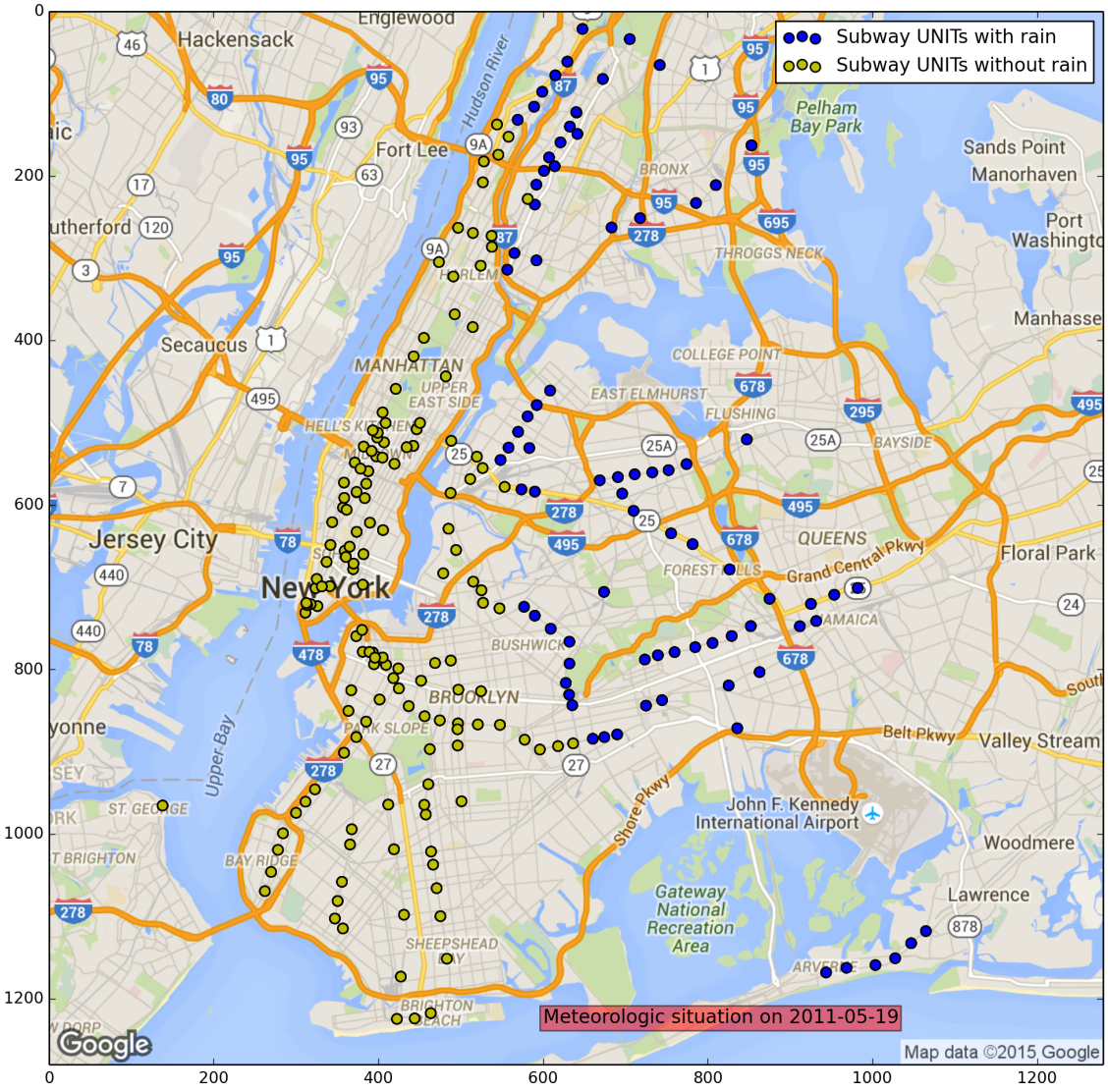

Figure 3: Map of NYC meteo stations, at the moment with most diversified rain conditions

- Still, a little strange looks the relatively feeble entry activity at 8:00h, a time when we would expect a rush hour for the commuters. Probably this is due to the fact that NYC has the **latest median time** of arrival at work from all metro areas in the U.S., as reported in [7].

To illustrate the ridership dependence on the location of the subway station, Figure 5 shows a map of the subway UNIT, where the size of the circles correspond to the average number of entries for the respective UNITs. We see especially numerous subway entries (bigger and more reddish circles) in the following places:

- In Manhattan
  - Grand Central Station, Times Square and 59-Columbus stations—important transportation hubs
  - Lower Manhattan—a distinguished business & cultural center
- In Brooklyn
  - Jamaica Center station—an important connection and hub for the passenger traffic from JFK airport.
  - Roosevelt Av. station—the same, but for La Guardia airport

In fact, table 4 lists the top ten busiest stations according to our dataset. For comparison, the last column of the table shows the top-10 busiest stations according to the 2014 annual ranking given in [5].

6.2 Linear regression

Next we will try to find a linear regression model for predicting the numbers of entries (ENTRIESn_hourly) based on several exogenous variables. We can choose between *Gradient Descent* and *Ordinary Least Square (OLS)* methods for performing the linear regression. Gradient Descent is quicker but less precise (as it can get caught in local minima),

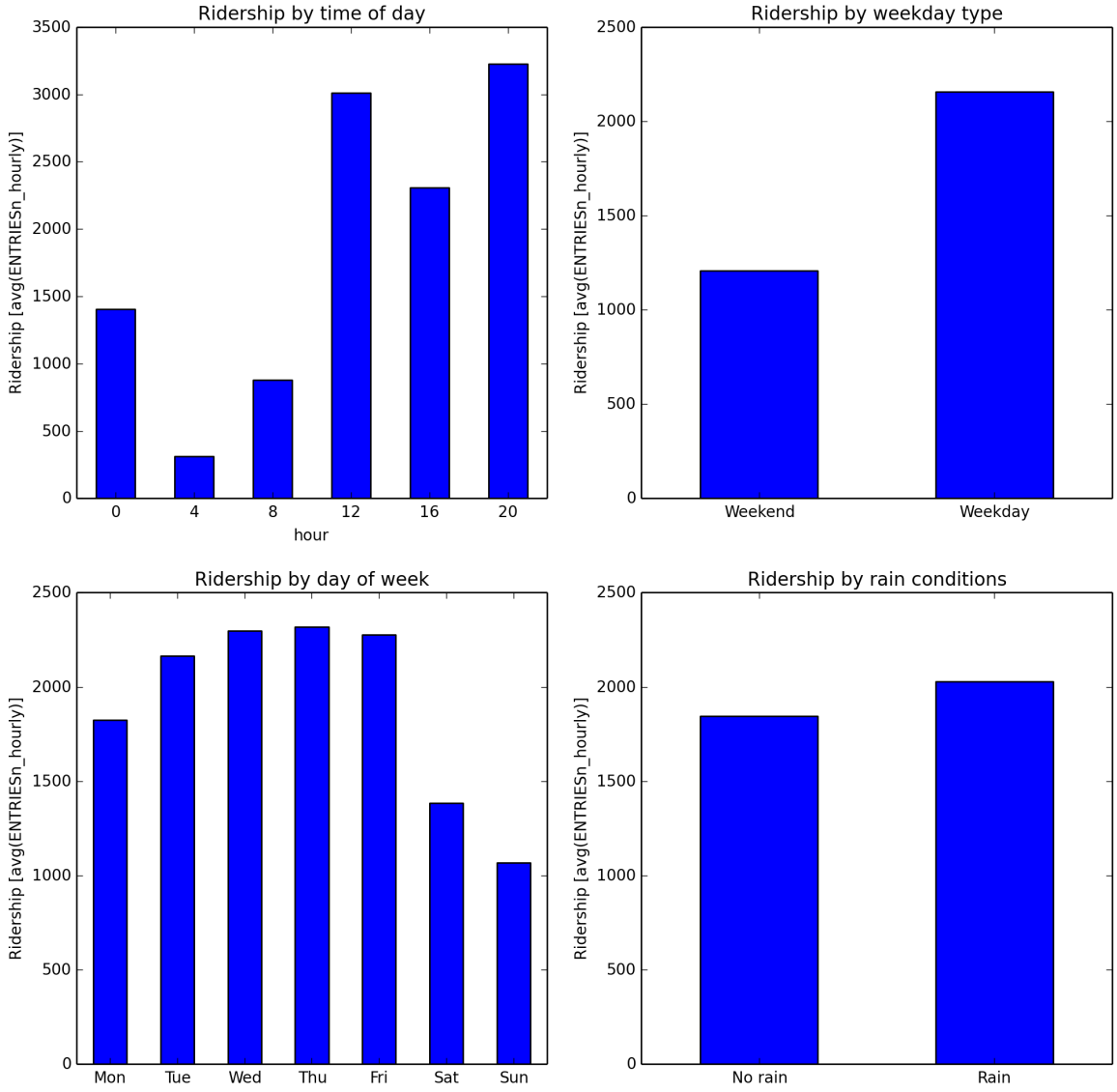

Figure 4: Ridership as a function of several factors

Rank	Station	Average hourly entries	Rank acc. to [5]
1	59 ST-COLUMBUS	10046	7
2	42 ST-GRD CNTRL	8360	2
3	MAIN ST	8359	n/a
4	34 ST-HERALD SQ	7851	3
5	ROOSEVELT AVE	7808	n/a
6	42 ST-PA BUS TE	7366	n/a
7	WORLD TRADE CTR	7245	n/a
8	47-50 ST-ROCK	6389	n/a
9	42 ST-TIMES SQ	6363	1
10	JAMAICA CENTER	5364	n/a

Table 4: The top ten stations with highest hourly entries

while the OLS method is computationally more intensive, but also more precise (it is guaranteed to find the global minimum). Due to the small size of the input dataset, we **choose to use OLS**.

As we saw in the analysis above, the following variables have clear effect on the volume of ridership, therefore they make good candidates for becoming exogenous variables: hour, day_week, UNIT (actually UNIT is the best proxy for the location of the subway station). We will be also interested in exploring the effect of the weather factors ('rain', 'fog', etc.) on the hourly entries.

The linear regression was performed with `statsmodel.api.OLS()` (see the accompanying IPython notebook) as shown in the following code:

```
import statsmodels.api as sm
```

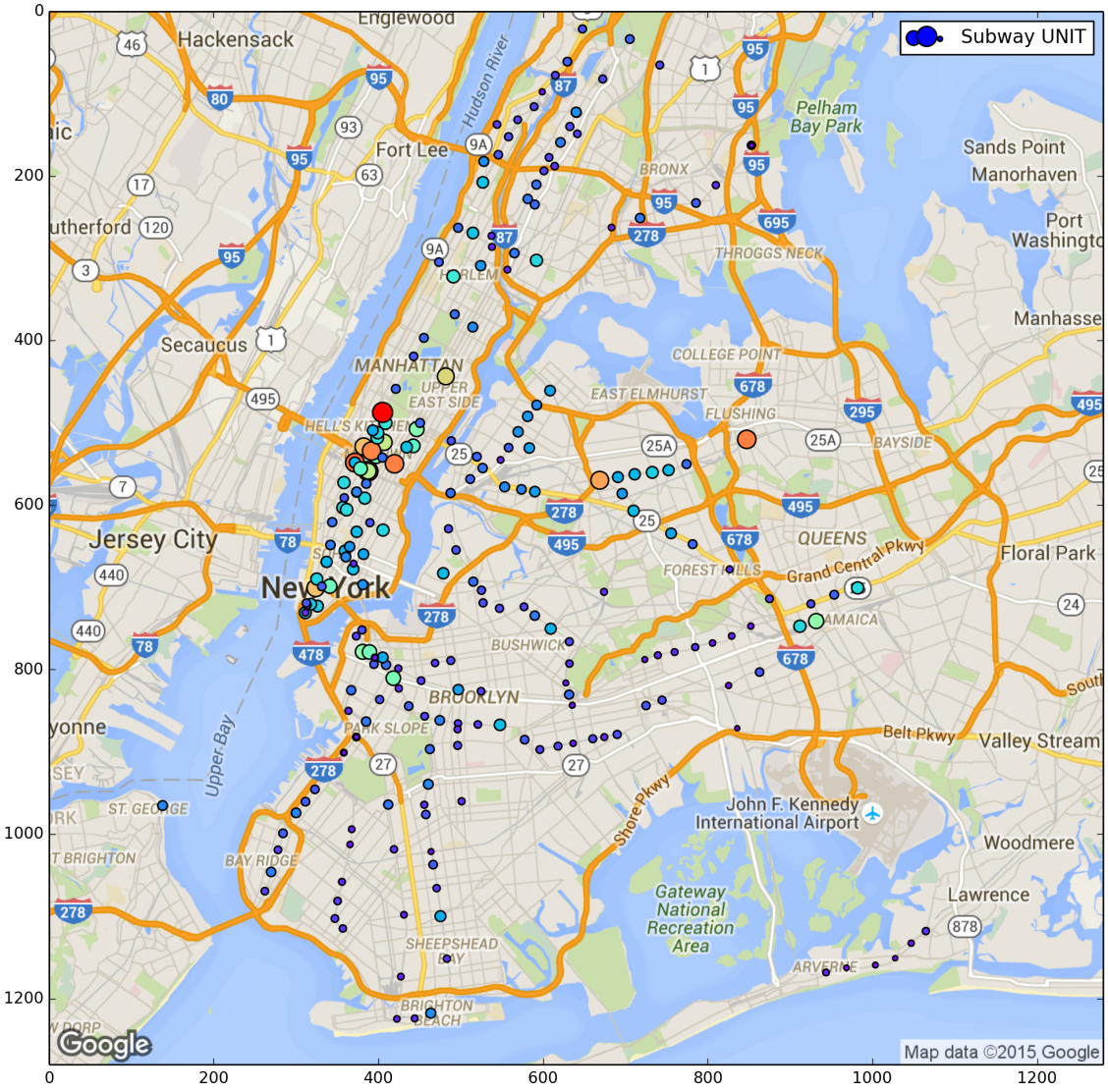

Figure 5: Ridership as a function of subway UNIT location

```
# Convert categorical variables to dummies
dummies_u = pd.get_dummies(df['UNIT'])
dummies_u.drop("R003", axis=1, inplace=True) # to avoid multicollinearity problems

# The exogenous variables
df_for_lr = df[["rain", "day_week", "hour"]]

# Add the intercept constant
df_for_lr["intercept"] = 1.0

# Add the dummies
df_for_lr = df_for_lr.join(dummies_u)

result = sm.OLS(df["ENTRIESn_hourly"], df_for_lr).fit()

print(result.summary())
```

Several experiments were performed, adding and removing exogenous variables and dummies, and recording the resulting **coefficient of determination** R^2 to assess the prediction quality of the model. The results are summarized in table 5.

We clearly see that the only variables that contribute to the predictive power of the model are: **day_week, weekday, hour, UNIT**. The weather-related factors, on the other hand, had no significant influence on the precision.

Table 6 shows the non-dummy coefficient weights from one of the OLS experiments.

We notice that the 95% CI for the 'rain' factor is pretty wide. This indicates a rather uncertain, non-robust factor;

Exogenous variables	Dummies	R^2	Notes
day_week, hour	UNIT	0.469	
day_week, weekday, hour	UNIT	0.483	'weekday' helps
day_week, weekday	UNIT, hour	0.542	Treating 'hour' as dummy helps
day_week, weekday, rain	UNIT, hour	0.542	'rain' doesn't help
day_week, weekday, rain, precipi	UNIT, hour	0.542	'precipi' doesn't help
day_week, weekday, rain	UNIT, hour, conds	0.543	'conds doesn't help
day_week, weekday, rain, tempi	UNIT, hour	0.543	'tempi' doesn't help
day_week, weekday, meanprecipi	UNIT, hour	0.542	'meanprecipi' doesn't help
day_week, weekday, meantempi	UNIT, hour	0.542	'meantempi' doesn't help
day_week, weekday, fog	UNIT, hour	0.542	'fog' doesn't help
day_week, weekday, pressurei	UNIT, hour	0.542	'pressurei' doesn't help

Table 5: Experimenting with different exogenous variables for the OLS Linear Regression

Factor	Weight θ	[95% CI]	
intercept	-1421.89	-1741.07	-1102.71
day_week	72.22	56.75	87.69
weekday	1272.51	1200.94	1344.09
rain	32.19	-13.85	78.22

Table 6: Resulting weights from one of the OLS Linear Regression experiments

a fact corroborating our finding that it adds virtually nothing to the predictive power of the model. In comparison, the other factors have narrower CIs, and as a consequence clearer effect on the predictions.

We can plot the residuals after applying the above model by:

```
(df.ENTRIESn_hourly - result.fittedvalues).hist(bins=40)
```

The residues histogram is show in figure 6. We see that around 20% of the residues have significant amplitude reaching 50% of the range of the dependent value. At the same time the residues are not very well behaved (see [3]), since they exhibit a positive skew—probably suggesting the existence of undiscovered higher-order factors.

This (as well the relatively low value of $R^2 = 0.542$) is an indication that the Linear Regression is **not an adequate data prediction model** for this dataset. Other methods for machine learning, like *Random Forests* could be more appropriate.

7 Statistical test of ridership on rainy and dry days

7.1 Selecting an appropriate statistical test

Before selecting an appropriate statistical test we have to explore the distribution of the data. Figure 7 shows the two superimposed hourly entries' histograms for rainy and dry days. We see that the two samples have **identical** distributions, which however are **not normal**. Therefore we cannot use tests like Welch's T-test, that require normal distribution of the observations. We shall instead use a non-parametric test, like the rank-sum based **Mann-Whitney U test.**

7.2 Applying the Mann-Whitney U test

7.2.1 Assumptions

According to [9], the input dataset need to fulfill the following requirements in order to be acceptable for a Mann-Whitney U test:

- Have one **dependent continuous or ordinal** variable—this is fulfilled as our dependent variable is the hourly count of entries into the subway.
- Have one **independent categorical dichotomous** variable, consisting of two independent groups—this is fulfilled as our independent variable has two independent levels: "rain" and "no rain".
- Have **independence of observations**, i.e. there shall be no relationship between the observations in each group of the independent variable or between the groups themselves. To ensure this we must have different participants in each group with no participant being in more than one group. This requirement in fact is something that needs to be guaranteed by the test design, and is not something that can be tested for. In our case this is also fulfilled, since each observation represents an **unique combination of** `UNIT` and `datetime` . If this assumptions were not fulfilled, we could have used the **Wilcoxon signed-rank test** instead.

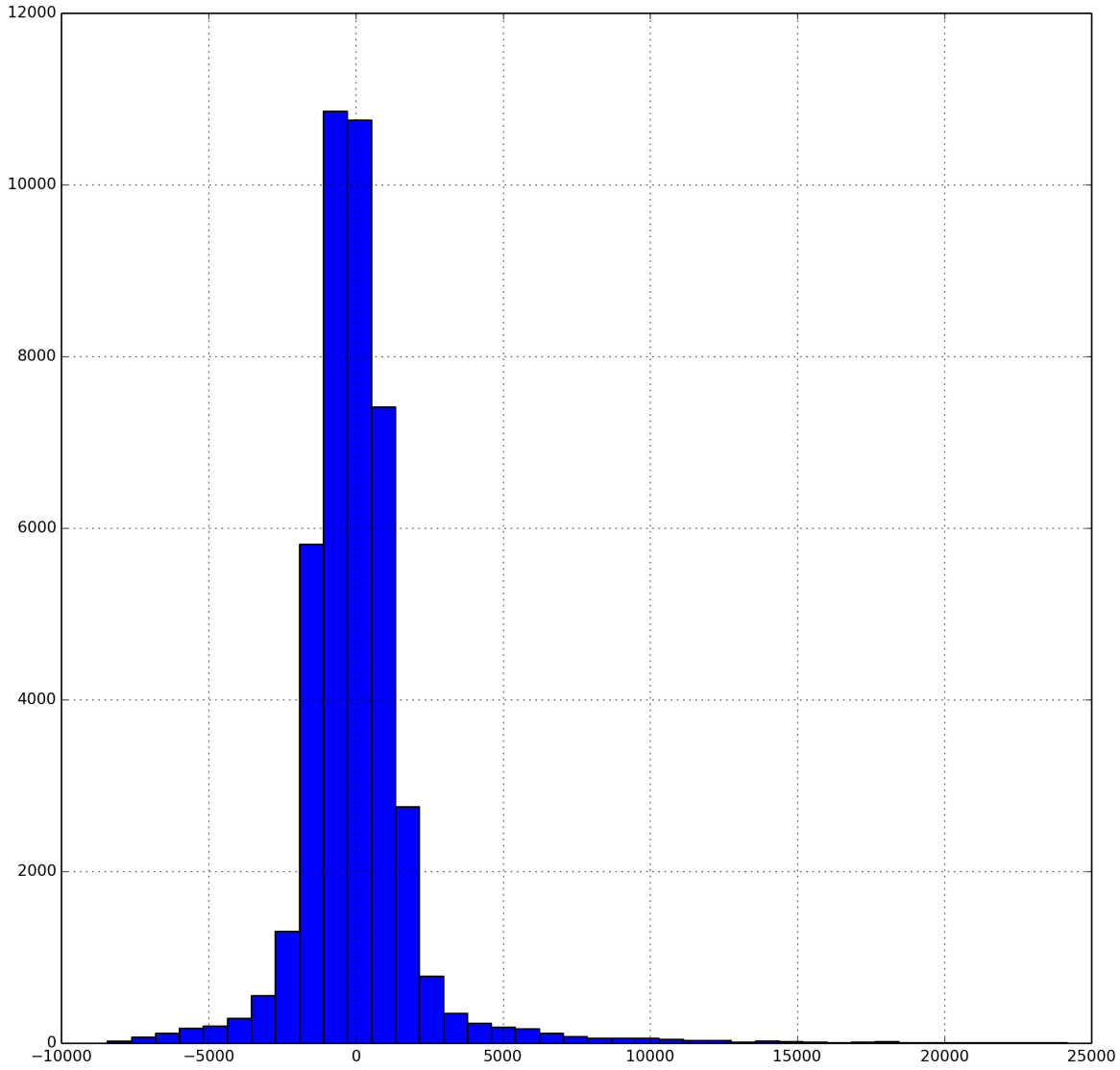

Figure 6: Histogram of residues after applying the OLS model on the initial dataset

7.2.2 Hypothesis

Furthermore, if the distributions of the two samples have the **same shape**, we can use the Mann-Whitney U test to compare the much more intuitive **median** statistics of the samples, rather than their **mean ranks**.

In our case, we have same-shaped distributions, therefore we can postulate the following hypotheses:

- **Null-Hypothesis,** H_0: the distributions of the two groups are equal (identical)
- **Alternative Hypothesis,** H_a: the median of the two sample groups are statistically different.

This formulation of the hypotheses calls for a **two-tailed** statistical test.

7.2.3 Mann-Whitney U test results and their representation

The statistical test was performed via `scipy.stats.mannwhitneyu()` (see the accompanying IPython notebook):

```
U, p = scipy.stats.mannwhitneyu(df[df.rain==0].ENTRIESn_hourly,
                                df[df.rain==1].ENTRIESn_hourly)
```

The returned result was:

$$U = 153635120.5, \quad p = 2.74x10^{-6}$$

Since the scipy implementation of the test returns a one-tailed p-value, we have to multiply it by 2 to obtain the two-tailed value:

$$p = 5.48x10^{-6} \text{ (two} - \text{tailed)}$$

Since the critical p-value ($\alpha - level$) is conventionally set at 0.05, we have enough statistical evidence ($p << p_{crit}$) to **reject the null-hypothesis**, i.e. we prove a significant difference between the two samples.

The median values of hourly entries between the rainy and dry observations calculated via:

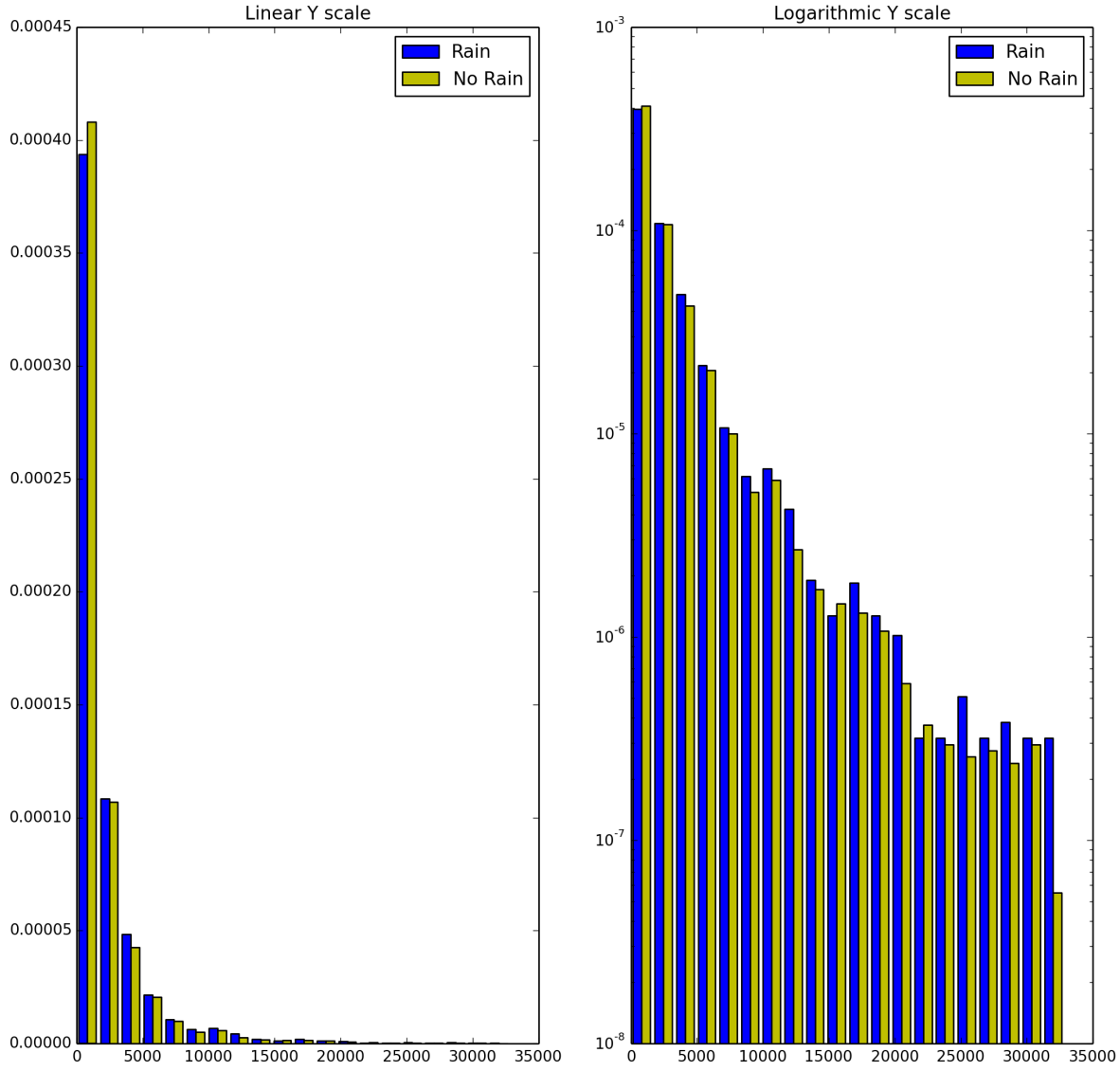

Figure 7: Histogram of ridership on rainy and dry days

```
df[df.rain==0].ENTRIESn_hourly.median()
df[df.rain==1].ENTRIESn_hourly.median()
```

are pretty close:

$$\widetilde{entries}_{rain} = 939 \quad \widetilde{entries}_{dry} = 893$$

This also gives an effect size (samples median difference) of $939 - 893 = 46$ in favor of the rainy days.

Following [10] we can report our findings as follows:

Median number of hourly subway entries for groups "rain" and "no rain" are 939 and 893 respectively; the two group distributions were found to be statistically different (Mann–Whitney $U = 153635120.5$, $n_{rain} = 9585$, $n_{dry} = 33064$, $P < 0.05$ two $-$ tailed*). We can therefore conclude that the analyzed dataset indicates a significant difference between subway entries on rainy and non-rainy days, with an effect size of 46.*

8 Conclusion

The performed non-parametric Mann-Whitney U test did indicate a significant difference between the volumes of ridership on rainy and non-rainy days; so, people really ride more the subway on rainy days.

At the same time we saw that 'rain' is not a significant predictor in a Linear Regression model trained on the given dataset, i.e. it does not help us predict reliably ridership.

9 Reflection

There are some shortcomings related to the input dataset:

- The records span only a very short period of time—one calendar month—thus we cannot estimate any effects related to the seasonal changes.
- To improve the quality of the prediction models, collecting data about additional factors could be useful: e.g. maintenance work on the subway, closed down lines, black-outs, presence of big public events, disturbances on alternative transportation networks (e.g. blocked roads).

References

[1] Alastair Aitchison. *The Google Maps / Bing Maps Spherical Mercator Projection*. Jan. 23, 2011. URL: https://alastaira.wordpress.com/2011/01/23/the-google-maps-bing-maps-spherical-mercator-projection.

[2] Femi Anthony. *Mastering pandas : master the features and capabilities of pandas, a data analysis toolkit for Python*. Birmingham, UK: Packt Publishing, 2015. ISBN: 9781783981960.

[3] Engineering Statistics Handbook. *Are the model residuals well-behaved?* Oct. 17, 2015. URL: http://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm.

[4] Wes McKinney. *Python for data analysis*. Sebastopol, Calif: O'Reilly, 2013. ISBN: 9781449319793.

[5] MTA. *The Ten Busiest Subway Stations 2014*. Oct. 17, 2015. URL: http://web.mta.info/nyct/facts/ffsubway.htm.

[6] Cyrille Rossant. *Learning IPython for interactive computing and data visualization*. Birmingham, UK: Packt Publishing, 2013. ISBN: 9781782169932.

[7] Nate Silver. *Which Cities Sleep In, And Which Get To Work Early*. Apr. 22, 2014. URL: http://fivethirtyeight.com/datalab/which-cities-sleep-in-and-which-get-to-work-early/.

[8] Stackoverflow. *Colorize Voronoi Diagram*. Dec. 11, 2013. URL: http://stackoverflow.com/questions/20515554/colorize-voronoi-diagram.

[9] Laerd Statistics. *Mann-Whitney U test in SPSS*. Oct. 17, 2015. URL: https://statistics.laerd.com/premium-sample/mwut/mann-whitney-test-in-spss-2.php.

[10] Wikipedia. *Mann–Whitney U test*. Oct. 17, 2015. URL: https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test.

[11] Wikipedia. *Mercator projection*. Oct. 17, 2015. URL: https://en.wikipedia.org/wiki/Mercator_projection.