
Analyzing the NYC Subway Dataset

Release

Ignacio Toledo

January 09, 2015

CONTENTS

1	Overview	1
1.1	Supporting Material	2
1.2	Some remarks about the datasets used	2
2	Statistical Test	3
2.1	Statistical Test Used	4
2.2	Justify the Statistical Test	4
2.3	Interpretation and discussion	5
3	Linear Regression	11
3.1	Linear regression algorithm(s)	11
3.2	Models and features used	12
3.3	Results: coefficients and R Squared	15
3.4	Interpretation and limits	16
4	Visualization	25
4.1	Ridership distribution with weather	25
4.2	Supporting visualizations	25
5	Conclusions and Reflections	27
5.1	Conclusions	27
5.2	Shortcomings and limitations	27
5.3	Insights	27
	Bibliography	29

OVERVIEW

This project consists of two parts. In Part 1 of the project, we have completed the questions in Problem Sets 2, 3, 4, and 5 in the Introduction to Data Science course.

This document addresses part 2 of the project, where we answer a set questions to explain our reasoning and conclusions behind our work in the problem sets.

The main purpose of the project is to analyze the ridership behavior for the New York City subway. The dataset used contains a sample taken from the month of May 2011, using the publicly available turnstile data from [MTA](#). The turnstiles in different stations of the system report the absolute number of entries and exits at certain hours for a given time interval. The improved dataset that we use reports the number of entries for time intervals of 4 hours, so it present us with 6 daily reports by turnstile.

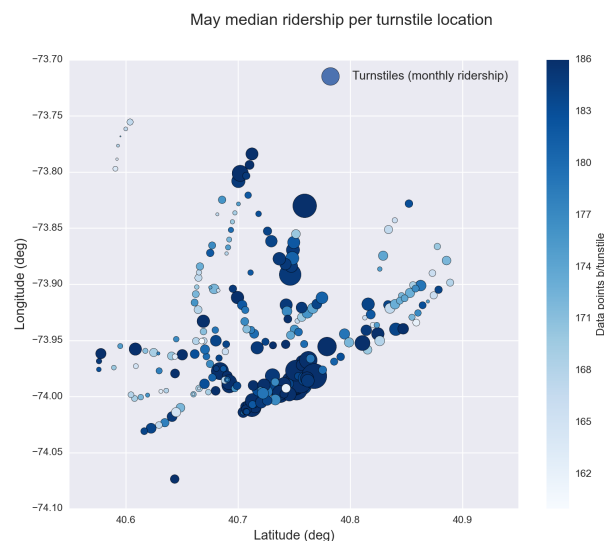


Figure 1.1: Turnstiles' locations within NYC, from the improved dataset.

Besides the information provided by the NYC subway, the dataset also includes weather information taken from several weather stations within the NYC area: each turnstile, depending on the location in NYC, is merged with the weather information of the closest weather station, thus providing temperatures, wind speed, pressure, conditions, precipitations, etc.

The project focuses one main question: Does the weather conditions, specifically precipitations, affect the NYC subway ridership? To answer this question we will use exploratory tools,

statistical tests and visualizations. Also, we will try to fit a model to the data by choosing certain predicting features; will the use of the precipitation variable improve the fit?

1.1 Supporting Material

Within the project github repository you will also find an ipython notebook, where most of the work done was recorded for reference.

1.2 Some remarks about the datasets used

For this project we use the data set provided at Data Analyst Nanodegree's portal for Project 1. The description of the variables can be found on...

However, after the exploratory and data analysis, we created another dataset by further munging the improved dataset. The basic idea was to smooth out features that might be caused by individual turnstiles or measurements. To do this, I grouped the data by time stamp and aggregated the entries by hour by adding all the entries. Also, the precipitation information for each time stamp was included by means of two columns:

- `rain_hour`: indicator (0 or 1) for precipitations for the particular date and time. It is 1 if for any of the stations the conditions were Rain, Light Rain, Hard Rain or Light Drizzle at that moment.
- `rain_day`: indicator (0 or 1) for precipitations for the particular day of the report. If at any station of our turnstiles the conditions reported precipitations during the day the value is set to 1.

STATISTICAL TEST

In lecture 3 and its problem set, the following question was given *Do rainy days affect the ridership of the NYC subway?* To answer this problem we began by creating two samples from our data:

- Sample A (*No rain*) is a subgroup containing the entries where no rain was reported, using the information of the `rain` variable ($rain = 0$)
- Sample B (*Rain*) is a subgroup with the entries where some precipitation was reported by means of the `rain` variable ($rain = 1$)

By studying the distributions, using histograms, we were able to characterize both data samples. We found out that both samples have a similar shape, clearly not normal, and positively skewed ([figure 2.1](#)).

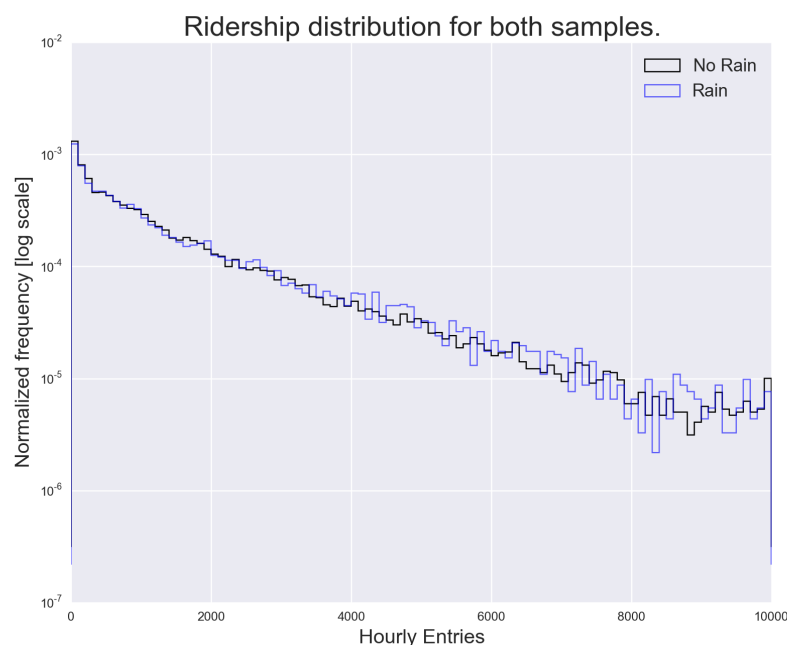


Figure 2.1: Ridership distribution comparison between rainy and dry days.
Please note the logarithmic scale on axis Y. It was used to allow us to study the visualization with more detail.

Because of the non-normal distribution we decided to use the median as measure of average for the samples:

- Sample A, days without precipitation, show a **median ridership of 901 passengers per hour**.
- Sample B, rainy days, report a **median of 945 passengers per hour**.

To assess the significance of this result, that rain seems to increase ridership in the NYC system by a small amount, we will use a non-parametric test.

2.1 Statistical Test Used

The Mann Whitney U test [\[wikiMann\]](#) is chosen to assess the statistical significance of this result. The null hypothesis in our case is that both populations are equal, or that there is no significant deviation on both populations medians (two-tailed hypothesis).

2.2 Justify the Statistical Test

The Mann Whitney U test, or Wilcoxon rank-sum test, is chosen because of characteristics of our samples: we can't use a parametric test because the distributions do not seem to follow any particular and well known probability distribution which we could use to make inferences that could directly report the significance of any difference between both populations.

The U test is particularly powerful to assess the significance of the difference between the median of two samples that have similar distributions. The assumptions that our data samples must comply with are basically:

- All observations of both groups are independent
- The responses are ordinal (so we can use the ranking algorithm of the U test).

Results =====p

We used the scipy implementation of the Mann Whitney U test (`scipy.stats.mannwhitneyu`). The results from the test are:

- $U = 150678745.0$
- $p = 1.91 \cdot 10^{-6}$

But the user should be aware that scipy reports a p-value for a one-tailed hypothesis, so we multiply by 2 to get the significance for our hypothesis:

- $p = 3.82 \cdot 10^{-6}$

2.3 Interpretation and discussion

The interpretation, given the result from the U test, is that the the ridership is not the same for rainy days than non-rainy days, with a significance higher than 95% ($p < 0.05$). Furthermore, from the descriptive statistics of our samples we would conclude that the ridership tends to be higher in rainy days.

However we have limited ourselves here to follow the procedure suggested by the lectures, assuming that observations of both groups are independent and there no other factors that might wrongly induce this result. Even when the data sample we use for the project has been through a more complete wrangling, there are still some issues that might affect the results:

- There is missing data for several turnstiles. From the original sample of 240 turnstiles, only 52 have complete data for May; also, as discussed on the forums, some precipitation data is missing from some weather stations.
- We are using the variable `rain` to create our samples: this variable indicates if the conditions at anytime of the day at a particular turnstile were rainy. Is it the appropriate variable to use to build the subgroups?
- There is one day which was a holiday (Monday 30th), should the data from this day be discarded?

Let's look with more detail at some these problems.

2.3.1 Missing data and precipitation distribution

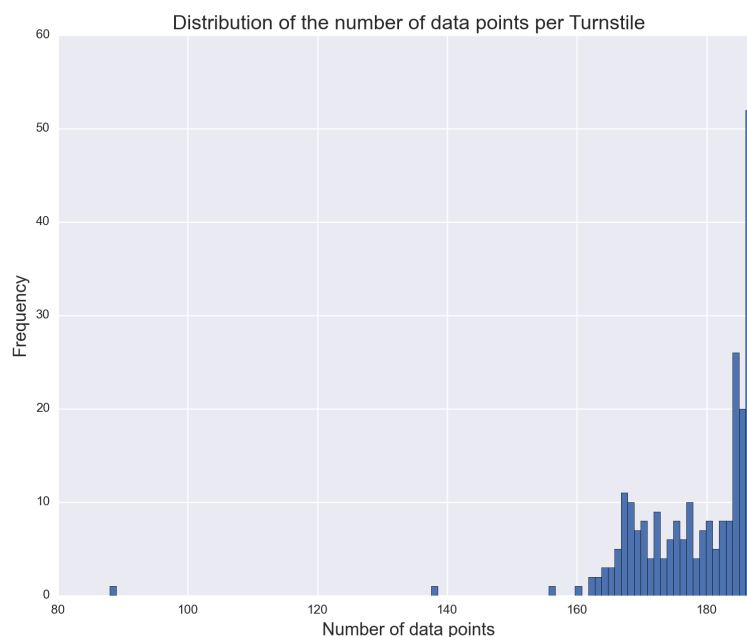


Figure 2.2: Number of data points (measurements) by turnstile on project's improved dataset.

Figure 2.2 shows some turnstiles have missing data for the month of May; with 31 days and 6 daily reports it is expected that a complete monitored turnstile should have 186 measurements. This is the case for 52 turnstiles, but 185 turnstiles have a number of measurements between 160 and 185. 3 turnstiles had less than 160 entries, and after inspections they have been removed because of the huge amount of missing data or time stamps reporting 0 entries. Of the 185 turnstiles with incomplete data, there was one case where at all time stamps the number of entries was 0, which was also removed as it does not add any information to our analysis (even when in other cases it could give further information).

The problem with the missing data is that, for some not clear explanation we could provide, affects more the suburb stations turnstiles than the ones in downtown areas. And suburb stations tend also to show lower number of hourly entries, i.e, a lower ridership, than downtown turnstiles. This effect can be seen in Figure 2.3.

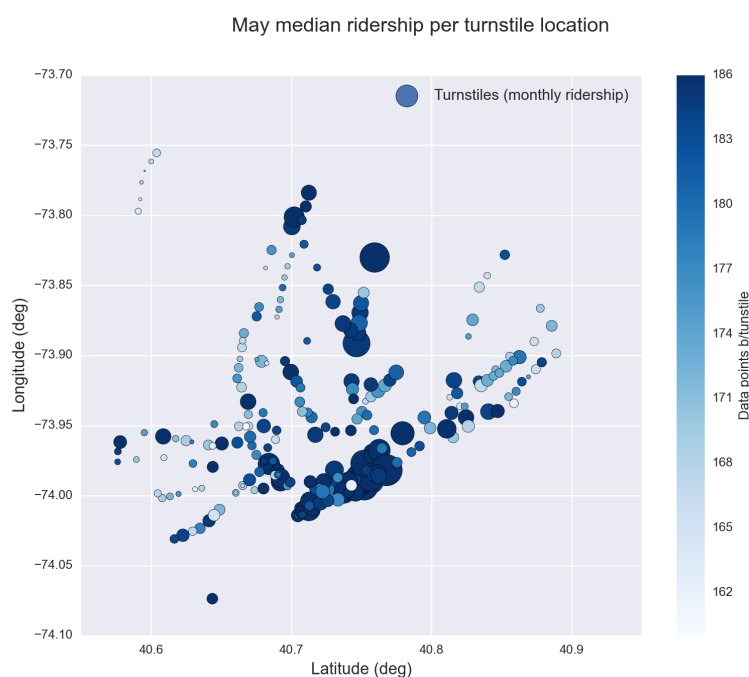


Figure 2.3: Turnstiles monthly median ridership, location and number of data points
The figure shows the distribution of the turnstiles within NYC which are in our dataset. The size is proportional to the monthly median ridership (entries by hour) while the color indicates the data completeness of each turnstile: whiter colors indicate locations with more missing data.

We wonder, as the reader also may, if this missing data could affect in anyway our previous study. We are not completely sure, but we think that given the way we performed our analysis it could happen that the results were affected: the downtown station data, which also correspond to the group of stations with higher ridership, is contributing to increase the median “entries by hour” that we calculated, as they are located in the higher values side of the ridership distribution. What happens if the stations in this locations are also the ones that tend to have more rainy days? We didn’t believe this was the case, but just to be sure we created the plot shown in Figure 2.4.

The figure shows that the precipitation is higher in the northern NYC, which is also the location of the most busy turnstiles: the median ridership of stations with higher precipitation (> 0.004

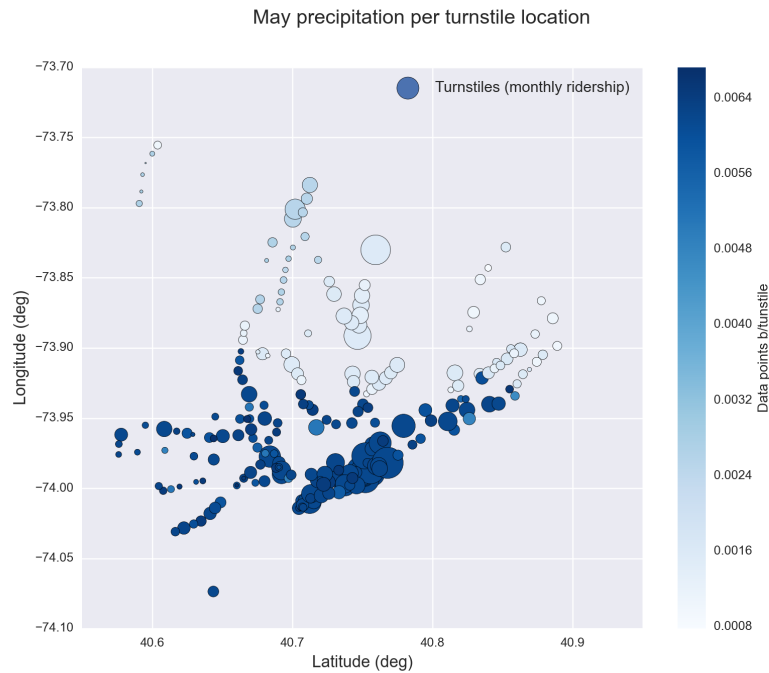


Figure 2.4: Turnstiles monthly median ridership, location and mean precipitation.

The figure shows the geographical distribution of the NYC turnstiles in the project's improved dataset. Size is proportional to the monthly median ridership and color represent the month's mean precipitation per turnstile. The figure shows that precipitations are higher in southern (and downtown) NYC.

inches) is 1116 entries by hour, while the stations with lower precipitation (≤ 0.004 inches) is 832 entries by hour. Also the stations with higher precipitation report on average 7 rainy days while the lower precipitation turnstiles only report 6 rainy days.

2.3.2 The use of the *rain* variable

The `rain` indicator in the improved data set reports if whether any precipitation happened at the turnstile location during the day. Because some of the precipitation data was missing in the weather tables, the conditions reported in the `conds` variable were used to create the `rain` column (as mentioned in the forums): if at anytime during a day the condition reported at a turnstile location was one of the following the `rain` indicator was set to one: 'Rain', 'Light Rain', 'Heavy Rain' or 'Light Drizzle'. This explains why for 94 entries reporting `rain` equal to 1, the `meanprecipi` variable (mean precipitation for the day at the location) was 0. Also, as shown before, this indicator is different for each turnstile depending on the closest weather station report. Thus, we find out that 216 turnstiles report 7 days of rain, 19 turnstiles report 6 rainy days, and 2 report 5 rainy days. Adding this analysis with the one in the previous subsection, we have to be aware that the samples might not be completely independent as previously thought.

Also, there is another important problem derived from the use of `rain` variable that we hope to make clear with the plot shown on [Figure 2.5](#).

The problem we see on using the `rain` variable as and indicator of rainy conditions for a

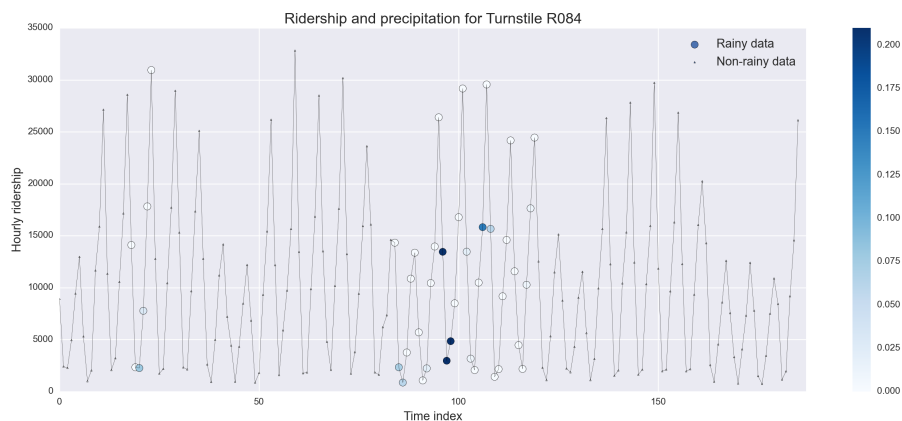


Figure 2.5: Ridership, precipitation and rain indicator for turnstile 084.

The figure shows the ridership evolution in May, in terms of entries per hour, for turnstile 084, which is one of the most busy stations in NYC subway. There is one point every four hours for the month of May, and the symbols indicate whether the day was rainy (big circles) or not rainy (small triangles). Also, the precipitation amount in inches for the rainy days is shown by means of the color bar in the right, with darker blue colors indicating more precipitation.

turnstile is that a whole day is tagged as rainy even when only rained at one time during the day. Furthermore, it can happen, as it can clearly be seen on the figure, that the rain happened in one of the less busy hours of the day, but still the whole day data will be tagged as rainy: this will clearly affect the results of our previous analysis.

2.3.3 Smoothing the data and answering the question again

In order to smooth out the previously mentioned effects we created a new data set from where two samples will be created later. For this dataset we grouped all individual turnstiles by time stamp, aggregating the ridership (`ENTRIESn_hourly`) using the `sum` function. In this way we have a set that represents the behavior of the whole NYC subway as one system, instead of individual turnstiles, reporting the total ridership at each time stamp. For each time stamp a variable called `rain_day` was created, which is 1 if in any turnstile during a day within the whole NYC subway network reported some precipitation, or 0 otherwise. Also, the data from May 30th is removed, since it changes the statistic for the Mondays. We will now redo the analysis using this dataset, and in this way try to answer the original question: *Does the NYC subway ridership changes with the precipitation conditions?*

- Sample A is the subgroup of all the data coming from non rainy days (`rain == 0`).
- Sample B is the subgroup of the data in rainy days (`rain == 1`).

The ridership distribution of both samples are again similar in shape, but they are not longer continuous, as shown in [Figure 2.6](#). We will use again the median to report the average of each sample, and the Mann Whitney U test to assess the significance of any difference we might find.

The ridership in non-rainy days has a **median of 370535 entries per hour**, while for rainy days the **median is 363124**. However the results from the U test are now different:

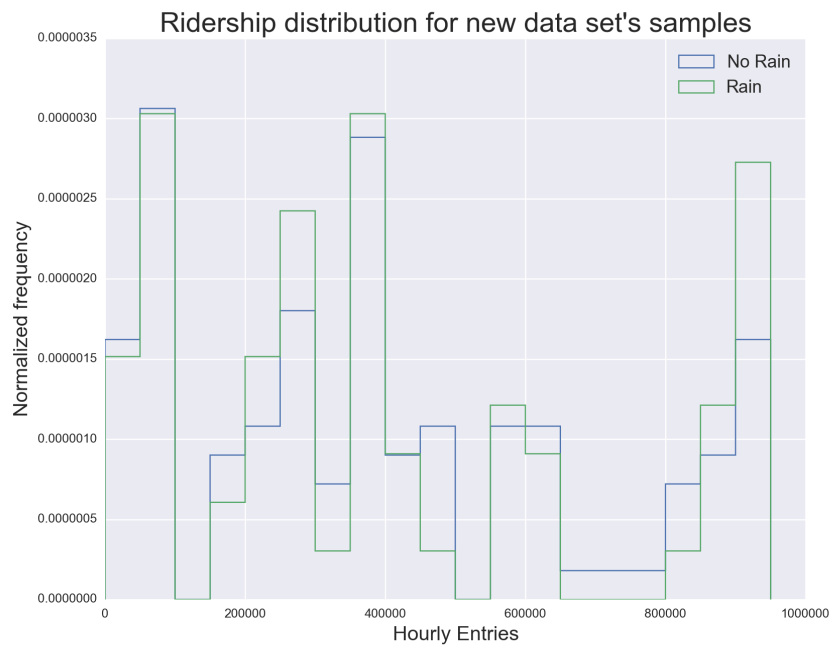


Figure 2.6: Ridership distribution comparison between rainy and dry days for the new samples taken from the aggregated data.

- $U\text{statistic} = 3477.0$
- $p\text{-value} = 0.71$ (Two-tailed hypothesis)

So the difference in the medians are not significant now, and we can't conclude that there is any meaningful difference in the ridership that could be explained by the precipitation conditions.

LINEAR REGRESSION

The second part of this work deals with the use of tools related to machine learning: can we use the data to create models that will allow us to predict the ridership based on some predicting features?

Problem Set 3 of the class has as one of the main goals the use of a linear regression model that could help us to predict the ridership in the NYC subway. We were asked to implement one of the algorithms that calculates the coefficients of a multiple linear regression model: gradient descent. The selection of the features, and thus the number of coefficients to fit, was left as an exercise for the student.

After implementing the linear regression model, and study its strengths and shortcomings, we used another algorithm to find the coefficients of the linear regression model: OLS, or ordinary least squares.

Finally, a third method was used, also based on a linear regression algorithm, but this time the model used higher order polynomials to learn from the data, in an effort to deal with the nonlinearity of it.

3.1 Linear regression algorithm(s)

3.1.1 Gradient descent

The code used to implement the gradient descent algorithm to find the linear regression coefficient can be checked on the python file available at the github repository associated to this work (`projectone.py`), and on the submissions to the *Introduction to Data Science* class (problem set 3).

3.1.2 Ordinary Least Squares (with statsmodels)

Selecting the same features as in the Gradient Descent exercise, we calculated the coefficients of the linear model by using the OLS implementation of the statsmodels python library [[statsmodels](#)].

3.1.3 Polynomial features with Ridge linear regression

After analysing the results from the previous regressions, and for reasons that will become clear after the description of their results, we went a little further and we used a polynomial transformation of the selected features, and another linear regression algorithm, the Ridge regression, was used to find the coefficients and predict ridership. The model used and results will be shown in the interpretation section.

3.2 Models and features used

The selection of the features to use and how to use them in the model was not a linear process, but an iterative work based on exploratory statistics, residuals analysis and study of the models obtained to predict the ridership. In fact, for the problem set 3 in the class we used a different set of variables as predictors (specifically `day_week` instead of `weekday`).

Also, it was because of these iterative analysis that we decided to give a try to a third method to model the data which included the use of polynomial features.

The multiple regression model used for the first two methods can be written as:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k \quad (3.1)$$

where \hat{y} is the predicted variable, x_i are the predictors (features) and θ_i are the coefficients or parameters that we are looking for using either the gradient descent or the OLS algorithms.

For our work we decided to use the following predictors:

- **UNIT**: turnstile unique identification. The use of the identification of each turnstile starts from the realization that the turnstiles have different ridership volumes for the same time periods, as it can be readily seen by comparing ridership averages. However this is a non-numerical categorical variable, so it was required to transform this variable to a numerical format by using dummy variables. This step adds at once n extra features or predictors, one for each turnstile in our data, which adds a lot of computing work to the algorithm.
- **hour**: numerical variable that indicates the hour of the day when the ridership is reported for each turnstile. This variable can take values that are continuous between 0 and 24, even when in the improved dataset the observed values are reported every four hours; it adds one coefficient to the calculations. [Figure 3.1](#) shows the relation of the ridership values with hour the day.
- **weekday**: numerical (and categorical) variable indicating if the day when the ridership measurement was done was either a weekday (1) or weekend day (0). [Figure 3.2](#) shows the relation of the ridership with kind of day.
- **rain**: daily precipitation condition for a turnstile location (0 for a clear day, 1 for rainy). Even when is a categorical variable, it is also numerical, and it is used as the final predictor feature for our linear model. [Figure 3.3](#) shows the relation of the ridership values with precipitation conditions.

The features were selected based partially on intuition and partially by exploratory analysis. First, it was clear that the behavior for each individual turnstile was mainly a function of the

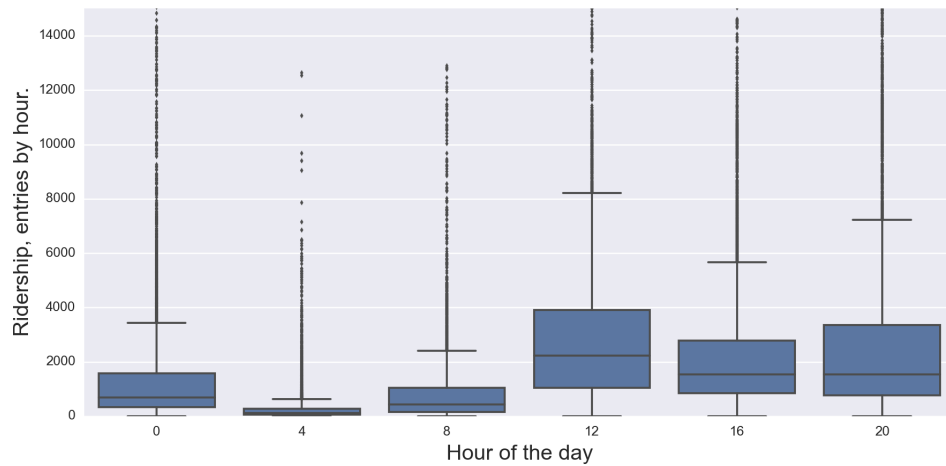


Figure 3.1: Ridership vs hour of the day.

Instead of just constructing a scatter plot, we decided to use another descriptive statistic method to study the relation, if any, between ridership and hour of the day. We used boxplots to visualize the distribution of ridership for each hour of the day. In this way we can see that the medians do not follow a linear relation with the hour of the day, and that there is a huge spread of possible ridership values for each hour.

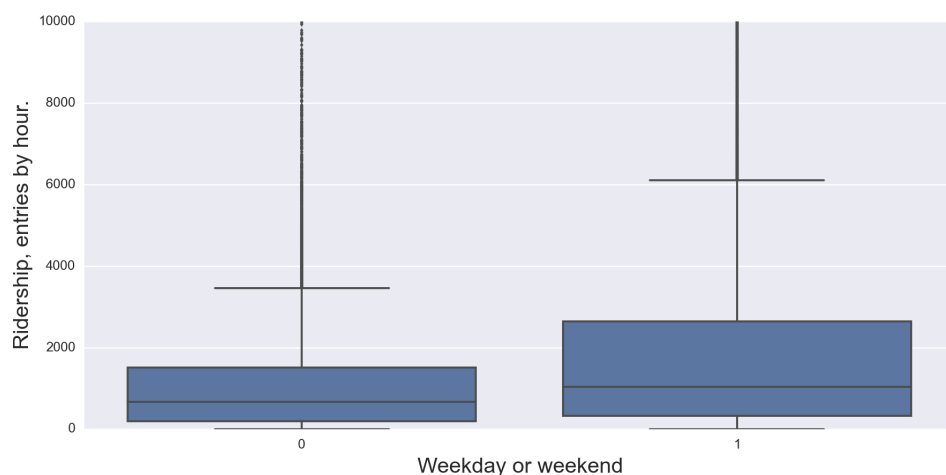


Figure 3.2: Ridership vs weekday/weekend-day.

This plot show the ridership distribution as boxplots for work days (Monday to Friday) and weekend days (Saturdays and Sundays). It is clear that even when the spread in entries per hour is still big a linear correlation can be used.

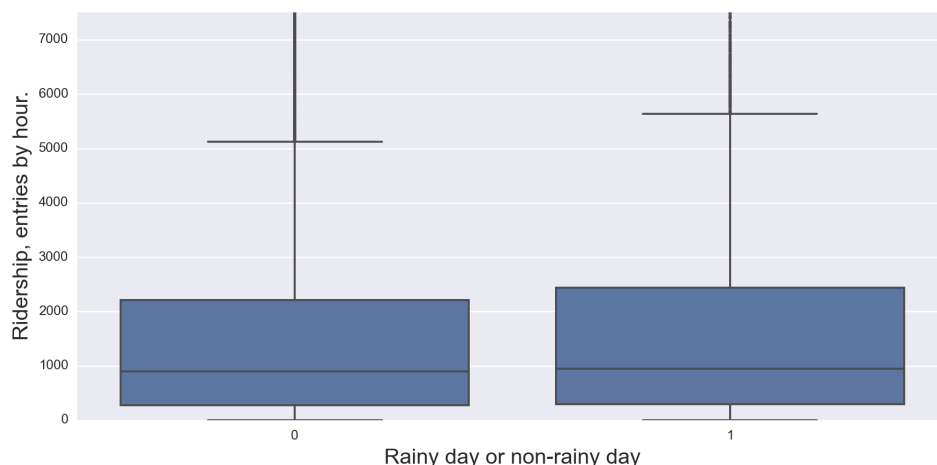


Figure 3.3: Ridership vs rainy conditions.

With the use of boxplots again, we can see in this figure that a really mild linear correlation exist between daily precipitation conditions and ridership. (Which as was shown in the previous section is not significant).

hour of the day and the day of the week, as is shown in *Figure 3.4*: there is a clear periodicity in the ridership behavior for each day, depending on the time of the day, and also a dependence on the day of the week. However the relation is clearly non-linear. We kept the `hour` as a predictor because is an important predictor, and in a very rough approximation one can see that ridership is lower in the beginning of the day while reaching a peak on the evenings.

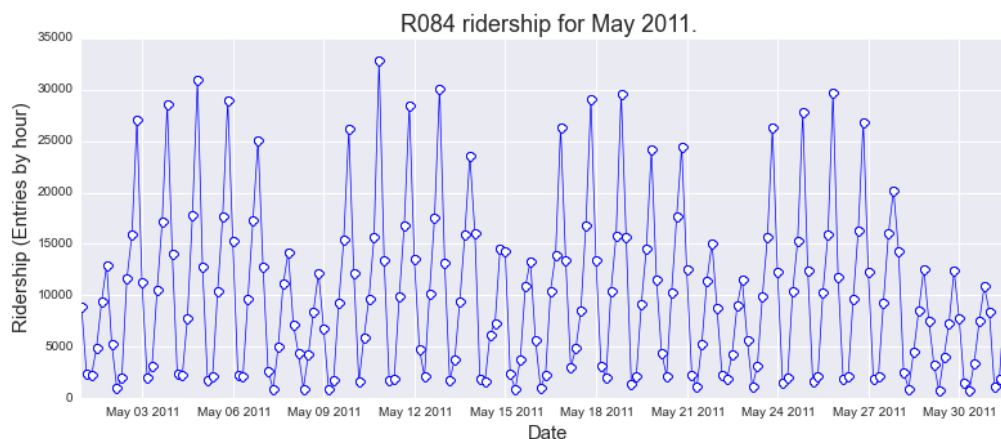


Figure 3.4: Ridership vs date for turnstile R084.

The figure clearly shows a periodic behavior for the ridership behavior for a particular turnstile, which is a function mainly of the hour of the day and day of the week. Ridership peaks are usually seen at 20 hours, while weekends and holidays (May 30th) being less busy than weekdays.

However, we decided to use `weekday` instead of `day_week` (the second being the day of the week, i.e, a number between 0 and 6, where 0 is Monday and 6 Sunday), because the major change on ridership behavior is seen between work days and off days (weekends), and `weekday` can be better modeled by a linear model than `day_week` (as it can be checked on *Figure 3.5*)

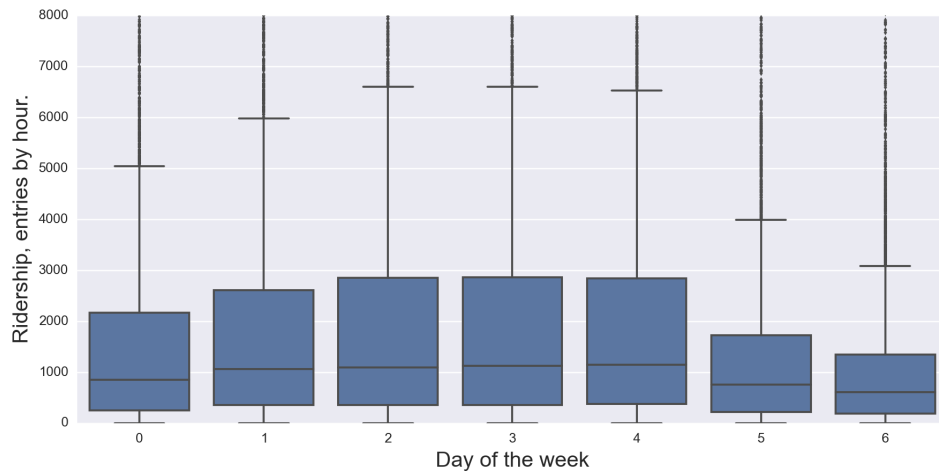


Figure 3.5: Ridership vs day of the week.

This plot show the ridership distribution as boxplots for the 7 days of the week (0 is Monday, 6 is Sunday). We can see that even when a relation exist between day of the week and ridership, this relation doesn't look linear, and thus we decided to use `weekday` instead.

Even when `UNIT` was not a numerical variable, we decided to use it given the different ridership patterns for each turnstile location. When using it as a dummy variable what we will be doing is adding or subtracting a constant offset which is particular for each location. Will this be enough to model the behaviors of different stations?

No further experiments where done to try other weather variables, since we were mainly interested in the behavior of the system as a function of precipitations; also, no other linear relationships were apparent from these variables, or there was not enough data to sample the ridership under some conditions (e.g, only 1 or 2 foggy days, no snow, etc.)

Finally, out from intuition we left out the variable `EXITSn`: besides having a highly linearly correlated relation with the ridership variable, it is clear that this variable is not completely independent from the number of entries to subway. Furthermore, it won't be a nice predicting feature, since its value will depend on the number of entries, and it should be treated as a observable or predictable variable on itself.

3.3 Results: coefficients and R Squared

The coefficients found with the gradient descent and OLS algorithms were the same in both cases, which was expected for a successful execution of the gradient descent algorithm. The selected features were enough to obtain a $R^2 = 0.481$. More in depth details of the result can be seen in [Table 3.1](#). Also, thanks to the statsmodels OLS implementation we can report some of the coefficients obtained from the linear model fit, using the predictor variables `hour`, `weekday`, `rain` and dummies from `UNIT` ([Eq. 3.1](#)), and their statistical significances ([Table 3.2](#)).

Table 3.1: OLS Regression Results

OLS Regression Results	
Dep. Variable: ENTRIESn_hourly	R-squared: 0.481
Model: OLS	Adj. R-squared: 0.478
Method: Least Squares	F-statistic: 163.1
Date: Wed, 07 Jan 2015	Prob (F-statistic): 0.00
Time: 14:12:52	Log-Likelihood: -3.8397e+05
No. Observations: 42267	AIC: 7.684e+05
Df Residuals: 42027	BIC: 7.705e+05
Df Model: 239	
Covariance Type: nonrobust	

Table 3.2: Linear regression coefficients

Predictor	coef	std err	t	P> t	[95% Conf. Int.]
Intercept	-1750.5171	166.661	-10.503	0.000	-2077.175 -1423.859
C(UNIT)[T.R004]	334.1581	231.108	1.446	0.148	-118.819 787.135
C(UNIT)[T.R005]	335.0522	232.095	1.444	0.149	-119.859 789.963
C(UNIT)[T.R006]	451.3319	229.532	1.966	0.049	1.445 901.218
C(UNIT)[T.R007]	164.5844	232.767	0.707	0.480	-291.644 620.812
...
hour	124.0989	1.500	82.741	0.000	121.159 127.039
weekday	980.9091	23.243	42.203	0.000	935.353 1026.465
rain	36.3145	25.167	1.443	0.149	-13.013 85.642

3.4 Interpretation and limits

Even when a relatively high R^2 was achieved by the use of a multiple linear regression model, a successful model should also comply with several assumptions, which can be checked by analysing the residuals [Diez2012].

1. **Are the residuals for the model nearly normal?:** *figure 3.6 top rows*, shows that the residuals obtained do not seem to follow a normal distribution. Even when the peak of the residuals tend to be zero, the wings do not follow a Gaussian distribution, as is more easily seen on the top left plot. Most probably, we have a big number of outliers.
2. **Is the variability of the residuals nearly constant?:** the variance of the residuals can be checked on the bottom left plot of *figure 3.6*, where the residuals vs predicted values are plotted. The figure doesn't show a constant variance along the x axis, with a lot of features that might be related to a poorly fit.
3. **Are the residuals independent?:** a plot of the residuals in the order of the data collected in the original data frame should show no relation between close neighbours. Our data frame mix data from several turnstiles, but it is ordered in such way that all data from the turnstiles can be found on sequenced blocks, where the data is again ordered by date and time. From the bottom right plot on *figure 3.6* it seems that the residuals do not look independent between different turnstiles.

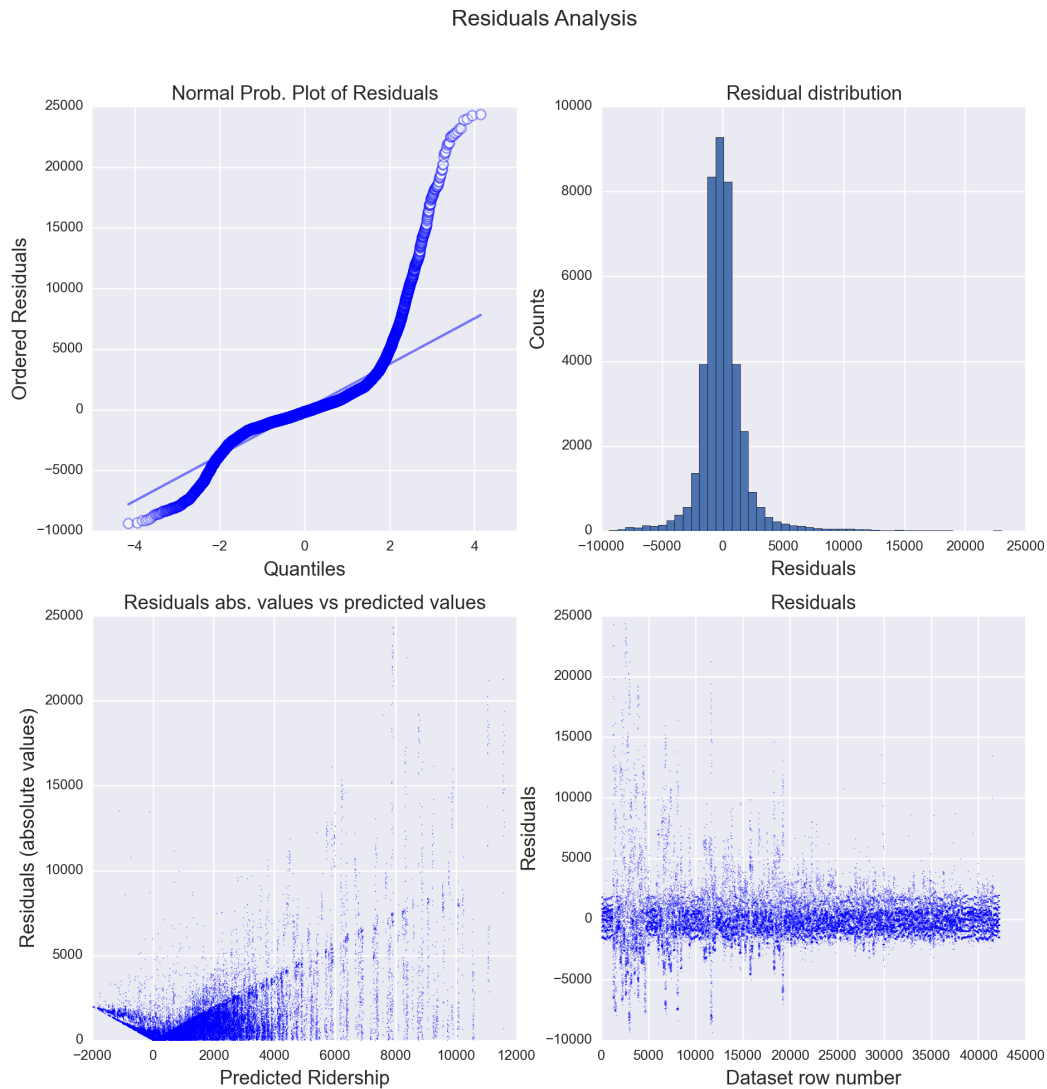


Figure 3.6: Residuals analysis plots for the linear regression model (improved dataset). *Top left*: normal probability plot of the residuals and *top right*: residuals distribution. It is clear that residuals do not adjust well to a simple normal probability distribution. *Bottom left* shows the residuals versus the predicted ridership, and *bottom right* just the residuals following the order on which the observed values were found on the improved dataset.

4. **Is each variable linearly related to the outcome?:** we can check the linearity from the figures presented in section 3.2; also the reader can check some other figures withing the ipython notebook associated to this project. It has been already established that there is a linear relation between ridership and the variables `weekday` and `rain`; however there is a poor relation with the `hour` variable ([figure 3.7](#)). However, there are some issues raised given the way the `UNIT` variable was included in the model, and that can be seen in the plots shown in [figure 3.8](#) and [figure 3.9](#).

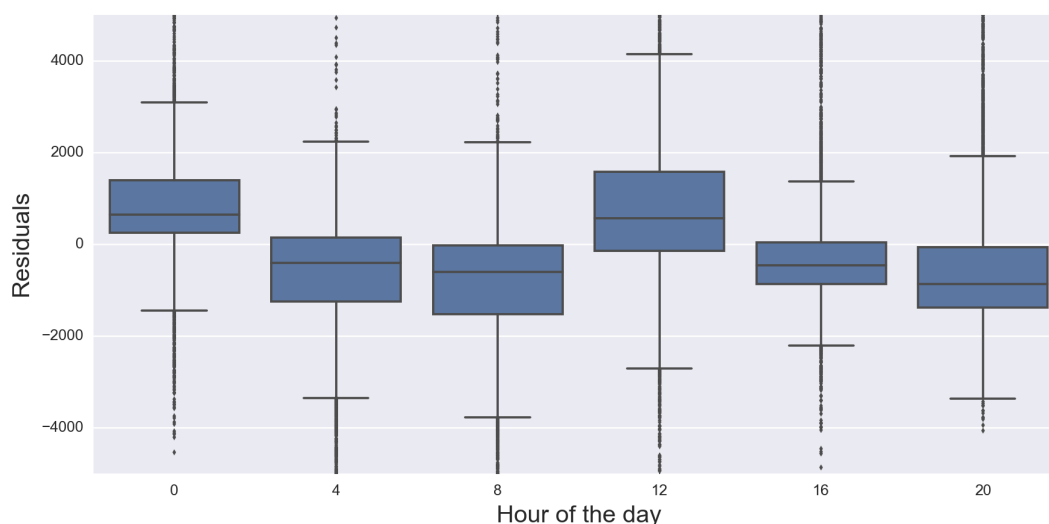


Figure 3.7: Residuals (as boxplots) vs hour of the day.

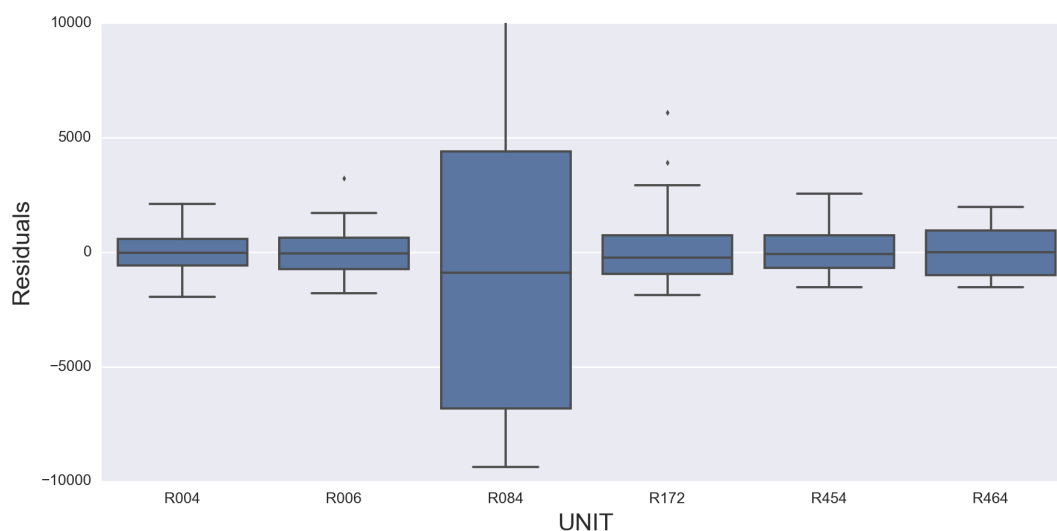


Figure 3.8: Residuals (as boxplots) for different turnstiles.

Besides the mild coefficient of determination it seems that many of the assumptions are not met by our data to successfully apply a multiple regression model to it. The residuals analysis are very good indicators of the behaviors of the ridership that the model can't explain, mainly because it was a very rough assumption to use `hour` as it is clearly not well modeled by the linear regression ([figure 3.7](#)). [Figure 3.9](#) is also a nice diagnosis tool to show that using the turnstiles names as dummy variables can help to improve the fit to the model, but is not

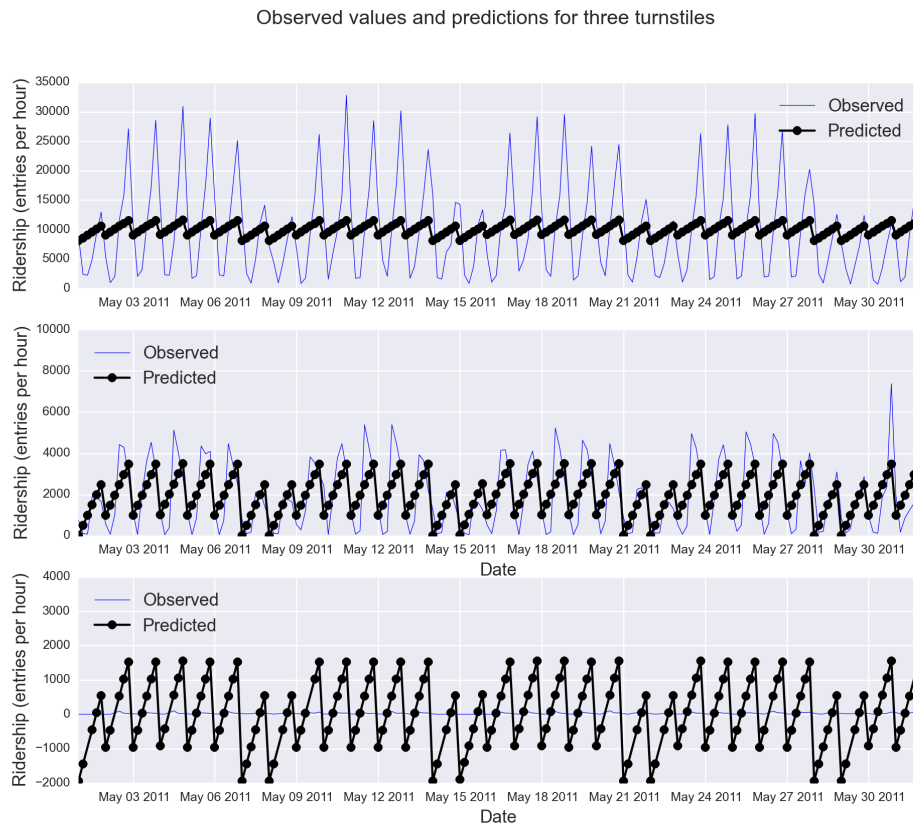


Figure 3.9: Observed and predicted ridership values for three different turnstiles.

The turnstiles used were R084, R172 and R338, one at downtown and the other two at the periphery.

The predicted values come from the linear regression model applied in previous section. Note how besides the middle panel, the model predictions do not follow well the observed ridership for stations with too much traffic or low traffic. Also, because of the way the `UNIT` dummy variables are used, we can see that just adding a constant is not enough to scale the ridership for individual locations.

enough. From the figure we can see again that the ridership varies from location to location, with peaks and valley hours happening at different times of the day for different turnstiles. Our model only corrects each turnstile by adding or subtracting a constant to each turnstile, which is not enough to model the ridership of the different locations. We also found that given the negative value of the intercept coefficient and small values for some turnstile coefficients we have several ridership predictions that are negative: this is meaningless for our problem, since it doesn't make sense a negative ridership.

Finally, it is interesting to independently check that even when the `rain` variable can be fit by a linear model, its significance is very low as can be seen by the low p-value of the coefficient: 0.15. In fact, removing `rain` as predictor feature only reduces the R^2 by less than 0.0001, and the reported coefficient of determination is still 0.481.

3.4.1 Aggregated dataset and polynomial features

We will now take advantage of the extra wrangling done with the improved data set in the previous chapter, and we will use the smoothed dataset that we created: **`nycsubway_weather`**. This data set was created by aggregating the ridership for each time stamp by adding all the ridership of the individual turnstiles, so we have a dataset that reports the ridership of the NYC subway as a whole. The columns of this dataset are:

- `ENTRIESn_hourly`: the total ridership as entries per hour for the whole NYC subway system
- `dateTime`: `datetime` variable, is the date and time for each observed value.
- `hour`: integer value, is the hour of the day for each reported value. It has a 24 hour format.
- `day_week`: integer value, is the day of the week for the observation (0 for Monday, 6 for Sunday)
- `weekday`: indicator variable, 1 for a weekday, 0 for a weekend day.
- `holiday`: categorical variable, 1 for days that are holidays.
- `rain_hour`: indicator variable, it reports whether at any location within the NYC Subway system was raining at the particular time
- `rain_day`: indicator variable, it reports whether at any location in the NYC Subway system there was any precipitation (rain) for the particular day of the reported value.

After some tries with multiple regression models, using the OLS `statsmodels` implementation, we were able to raise the R^2 value to 0.563 using this new dataset and three predicting variables: `hour`, `weekday` and `holiday`. Neither `rain_day` nor `rain_hour` improved the coefficient of determination noticeably, with p-values higher than 0.61. Even with the smoothing achieved by the removal of individual turnstiles we were able to see the same kinds of problems as described previously, being the most important factor the nonlinearity of the relation between the hour of the day and ridership, plus the difference in this relation for different days of the week: having a constant added (or subtracted) given the type of day (weekday or day off) is not enough to account for the variations seen between days. *Figure 3.10* shows a plot with the observed and predicted values, which further explains the shortcoming of using

a linear model with our data. The reader can also check the ipython notebook associated with this project to look for the residuals analysis.

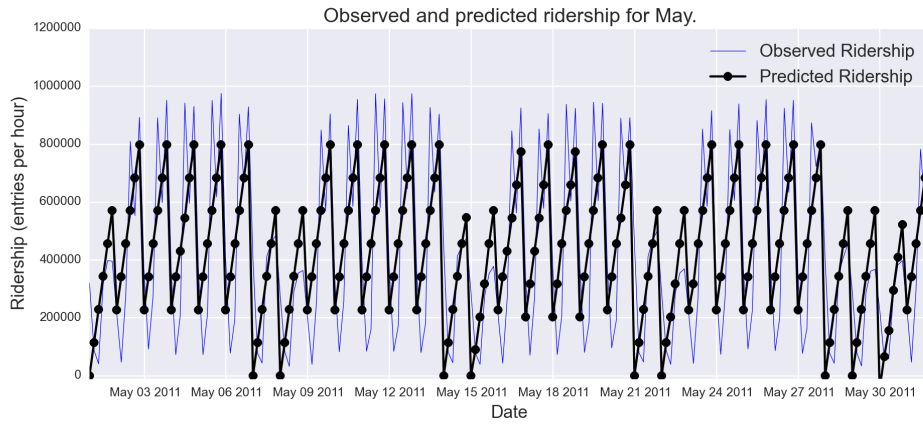


Figure 3.10: Observed and predicted ridership values for the NYC Subway, month of May. Even when we have eliminated the complexities by taking the whole NYC Subway as a whole and increased the percentage of the ridership behavior for May 2011 that is explained by the used linear model, it is still apparent the problems produced by the lack of linearity of `hour` vs ridership, and the changes in ridership behavior for different days of the week.

Because of these reasons we decide to try a different method. This method is still an algorithm that uses the linear regression tools, but the predictors are now converted into *polynomial features* [glmnet]. The problem can be resolved with a linear regression by taking advantage of the linearity of the coefficients in the system of equations needed to solve the problem of finding these coefficients. The function used to convert our selected predictors, that for this optional exercise will be `hour`, `day_week`, `holiday` and `rain_hour`, was the library `PolynomialFeatures` from the `scikit-learn` libraries for machine learning with python. We won't enter into the details of this method, since it goes beyond the goal of this project, but it suffice to say here that the new model is now a polynomial of degree 5 (that was our selection), where the predictors also interact with each other. So, if $x_1 = \text{hour}$, $x_2 = \text{day_week}$, $x_3 = \text{holiday}$ and $x_4 = \text{rain_hour}$, the model we are trying to use to explain our data is going to be of the form:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \dots + \theta_5 x_1^5 + \theta_6 x_2 + \dots + \theta_n x_1 x_2 + \theta_{n+1} x_1 x_3 + \dots$$

Also, we mentioned that we used a Ridge regression algorithm, that was suggested by the `scikit-learn` documentation as a more robust method to find the coefficients in a model like the one we are trying to use.

The main idea was to try to overcome the limitation given for the non-linearity of the hourly and daily ridership in our NYC subways system set. The improvement was amazing, by reaching a $R^2 = 0.968$, and the reader can check the residual analysis plots in [Figure 3.11](#).

Even when (a) the residual distribution is now closer to a normal distribution, (b) the variance seems to be more constant and (c) the residuals seem independent, we must draw the attention to the reader to the fact that this model, while an improvement, still have shortcomings, that can be seen in [Figure 3.12](#). While the hourly and daily ridership are now modeled with higher precision we have to be aware of the overfitting our model is suffering of, explained by the

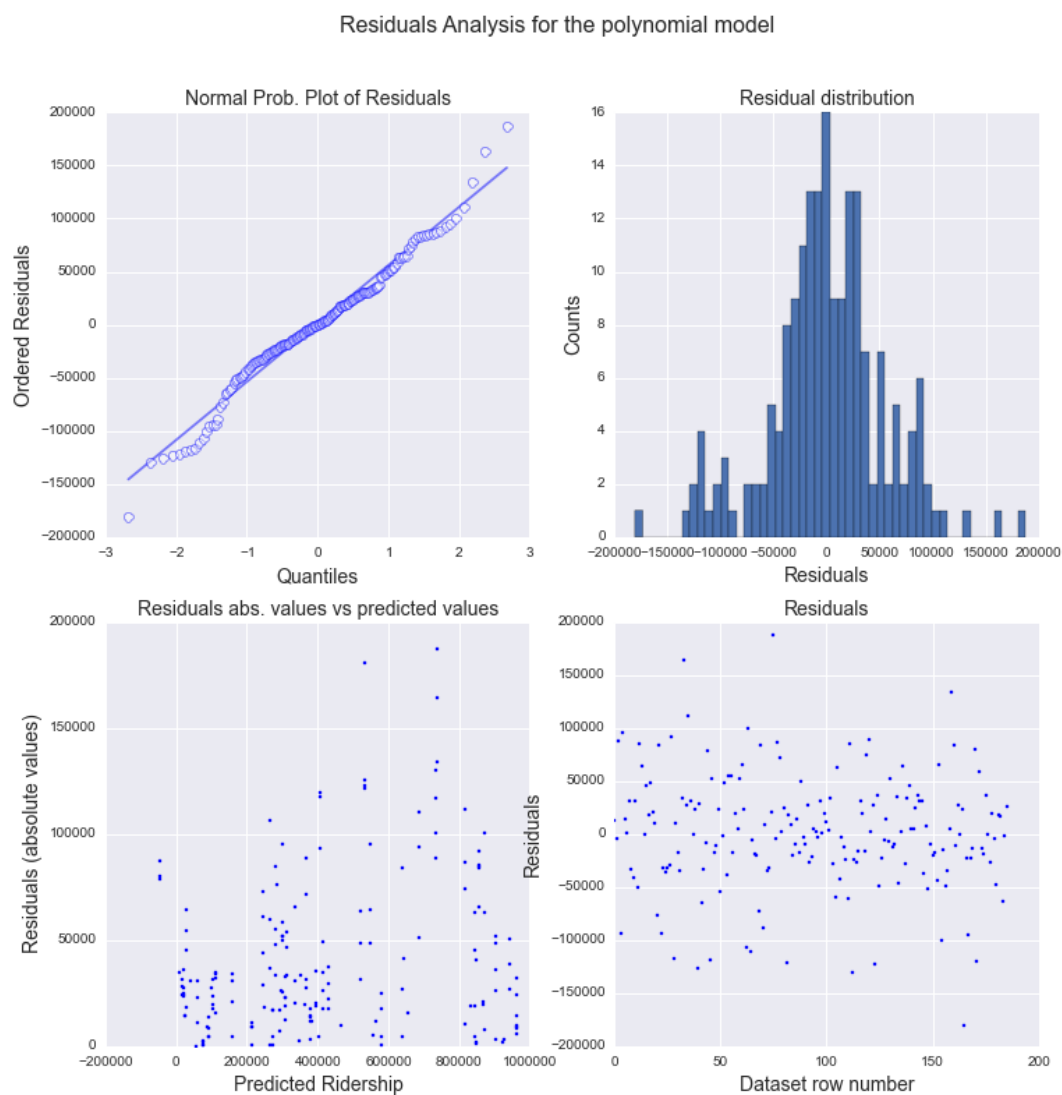


Figure 3.11: Residuals analysis plots for the polynomial model (nycsubway_weather dataset). *Top left:* normal probability plot of the residuals and *top right:* residuals distribution. The residuals are distributed now following more closely the shape of a Gaussian, and less outliers are visible; *Bottom left* shows the residuals versus the predicted ridership, and *bottom right* just the residuals following the order on which the observed values are reported.

large number of coefficients to be found (126 coefficients). However, it is clear that a much better work can be done with more complex machine learning algorithms, and the idea was just to show that with the data we have we should be able to predict ridership with much more accuracy than the linear regression is capable of.

Daily ridership behavior and polynomial regression model.

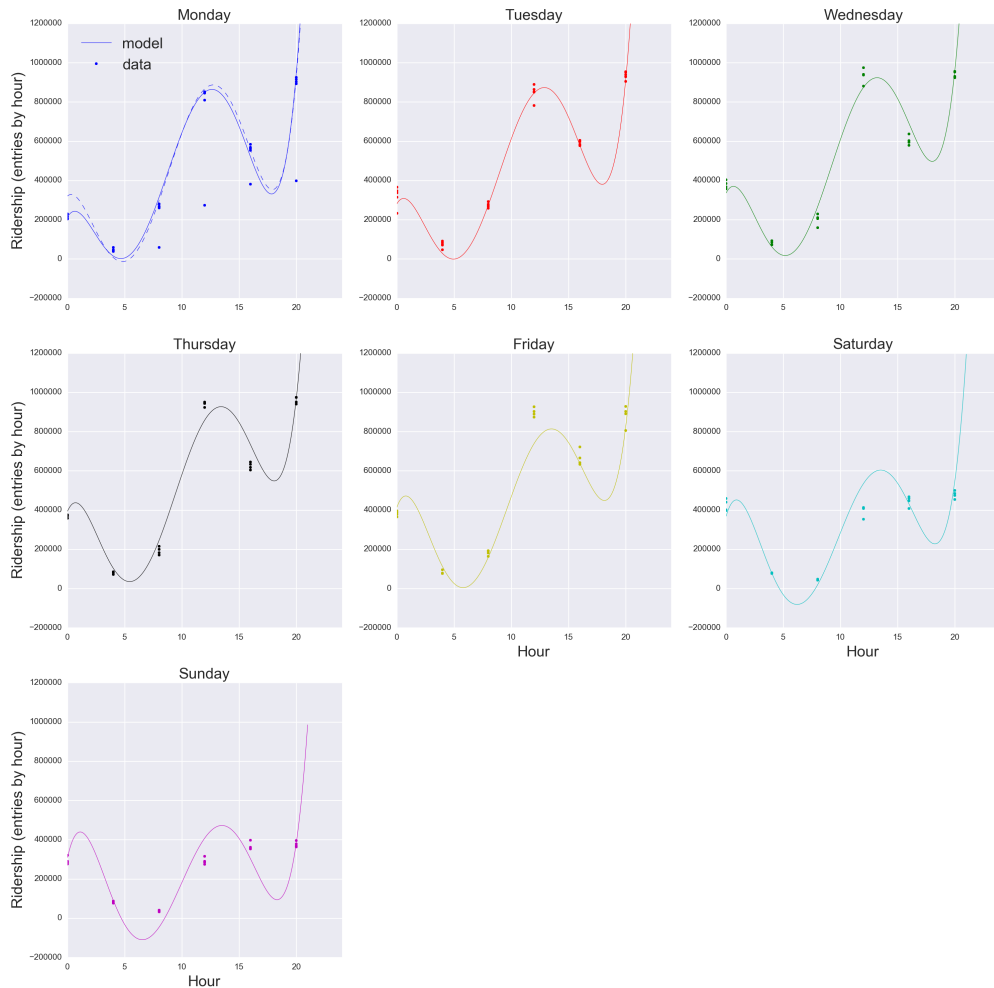


Figure 3.12: Observed and predicted ridership values for days of the week as a function of hour.

VISUALIZATION

4.1 Ridership distribution with weather

4.2 Supporting visualizations

CONCLUSIONS AND REFLECTIONS

5.1 Conclusions

5.2 Shortcomings and limitations

5.3 Insights

BIBLIOGRAPHY

- [Diez2014] Diez, Barr, Cetinkaya-Rundel. Introductory Statistics with Randomizations and Simulation. 1st Edition. openintro.org
- [Diez2012] Diez, Barr, Cetinkaya-Rundel. Open Intro Statistics. 2nd Edition. openintro.org
- [McKinney2013] McKinney. Python for Data Analysis. 1st Edition. O'Reilly Media. 2012.
- [Peck2012] Peck, Olsen, Devore. Introduction to Statistics and Data Analysis. 4th Edition. Brooks/Cole. 2012.
- [Lyman2010] Lyman Ott, Longnecker. An Introduction to Statistical Methods and Data analysis. Sixth Edition. Brooks/Cole. 2010.
- [Rossant2014] Rossant. IPython Interactive Computing and Visualization Cookbook. 1st Edition. Packt Publishing. 2014.
- [glmshkit] http://scikit-learn.org/stable/modules/linear_model.html
- [statsmodels] <http://statsmodels.sourceforge.net/stable/>
- [wikiMann] http://en.wikipedia.org/wiki/Mann%E2%80%93U_test