Greg Foss
Udacity NanoDegree
July 18, 2015

NY Subway Turnstile Data

An analysis of Entries Based On Rain

New York subway turnstile data can be obtained from MTA Developers. Udacity has paired this data with weather data for Project One to give students the ability to test "if more people ride the subway when it is raining versus when it is not raining." In this paper, I will answer the questions the project poses and explain my methods and thoughts on the process.

1: Statistical Test

I chose the Mann-Whitney U test to determine if there was a relationship between my datasets. The two datasets are "Subway ridership when it rains" and conversely "Subway ridership when it is not raining." I chose a non-parametric test because there are no assumptions on the distribution of these sets of data. A cursory thought about these data sets tells us they are unlikely to have a normal distribution. The Mann-Whitney U test has a null hypotheses that the two samples are from the same population. That is, the values will not be greatly different in each one. If one thinks of comparing values from one set with the other to see which is greater, the null hypothesis says that the 'wins' of each should be close (the number of values in each test, multiplied by one another and then divided by 2). A two tail test was chosen because no assumptions are made as to whether there were more riders when it rained or less. One could guess the answer, but it is best to not make that assumption. We are going to assume a 95% confidence interval, or a p-value of .05 for our test (.025 in each tail).

The following was the results found accompanied with the code used to produce it. With a p-value so small we can safely reject the null hypothesis, which states that the two samples are from the same population. Because this is a two-tailed test we multiply the p-value by 2. This is below our threshold of 5% confidence. Notice also that the U statistic is close to the maximum, therefore, we can also reject the null.

```
import numpy as np
import pandas as pd
import scipy.stats as sp
from ggplot import *

tsd_df = pd.read_csv('~/Downloads/tsd.csv')

#Overall Mean regardless of hour
tsd_norain = np.mean(tsd_df[tsd_df.rain==0]['ENTRIESn_hourly'])
tsd_rain = np.mean(tsd_df[tsd_df.rain==1]['ENTRIESn_hourly'])

print "Mean Entries Hourly with Rain: {0}".format(tsd_rain)
print "Mean Entries Hourly without Rain: {0}".format(tsd_norain)

#Mann-Whitney U critical
rain_df = tsd_df.rain==1                #Get boolean of rain or not
total_count = rain_df.count()           #Count 'em
rain_count = sum(rain_df)               #Only Trues add up
non_rain_count = total_count - rain_count
Umax = (rain_count * non_rain_count) / 2
Umin = Umax/2
[U, p] = sp.mannwhitneyu(tsd_df[tsd_df.rain==0]['ENTRIESn_hourly'],
tsd_df[tsd_df.rain==1]['ENTRIESn_hourly'])

print "Mann-Whitney U Max: {0}".format(Umax)
print "Mann-Whitney U Min: {0}".format(Umin)
print "Mann-Whitney U Test Statistic: {0}".format(U)
print "Mann-Whitney U p-value one-Tail: {0}".format(p)
print "Mann-Whitney U p-value two-Tail: {0}".format(p*2)
```

```
Mean Entries Hourly with Rain: 1105.44637675
Mean Entries Hourly without Rain: 1090.27878015
Mann-Whitney U Max: 1937202044
Mann-Whitney U Min: 968601022
Mann-Whitney U Test Statistic: 1924409167.0
Mann-Whitney U p-value one-Tail: 0.0249999127935
Mann-Whitney U p-value two-Tail: 0.049999825587
```

2: Linear Regression

The approach used to compute the coefficients theta and product predictions for ENTRIESn_hourly in the regression model was Ordinary Least Squares (OLS). The best description of this regression model I found here.

"Ordinary Least Squares is the simplest and most common estimator in which the two βs (data sets) are chosen to minimize the square of the distance between the predicted values and the actual values. Even though this model is quite rigid and often does not reflect the true relationship, this still remains a popular approach for several reasons. For one, it is computationally cheap to calculate the coefficients. It is also easier to interpret than more sophisticated models, and in situations where the goal is understanding a simple model in detail, rather than estimating the response well, they can provide insight into what the model captures. Finally, in situations where there is a lot of noise, it may be hard to find the true functional form, so a constrained model can perform quite well compared to a complex model which is more affected by noise."

The features, or input variables used were: 'rain', 'precipi', 'meanwindspdi', 'meantempi' and 'hour'. These values were chosen based on both intuition and experimentation. For instance, if it was really windy outside perhaps more people would opt to use the subway. I plotted mean wind speed on a line chart to see if this was the case, and although there was not a direct linear relationship between subway ridership and windspeed it did show that overall windier days had higher entries per hour. The same was the case with meantempi; the colder the weather the more subway riders. *This is included in the visualization section.* Below is the python used for the OLS and to calculate the R^2 value. The value of R^2 ranges from zero to 1 with zero being no relationship and 1 being a perfect relationship between coefficients. The results I returned indicate this is a satisfactory model for the task at hand. However, while a correlation is seen, it is not significant enough to warrant health or welfare policy changes. A correlation this small should not be used if it is used to make decisions that impact health or welfare.

Below is the simple python code used. I used the dummy variable (or categorical variable) of 'UNIT'. The reason I found in Wikipedia, "In regression analysis, the dependent variables may be influenced not only by quantitative variables(income, output, prices, etc), but also by qualitative variables(gender, religion, geographic region etc.)" It is important to note that after

truly understanding what the dummy variable was doing, I decided to also add 'day' and 'hour'. I also think it valuable to add those summary results separately to illustrate the effect day of the week had.

OLS Regression Results

Dep. Variable:	ENTRIESn_hourly	R-squared:	0.458
Model:	OLS	Adj. R-squared:	0.456
Method:	Least Squares	F-statistic:	237.3
Date:	Tue, 21 Jul 2015	Prob (F-statistic):	0.00
Time:	16:35:20	Log-Likelihood:	-1.1703e+06
No. Observations:	131951	AIC:	2.341e+06
Df Residuals:	131481	BIC:	2.346e+06
Df Model:	469		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	1034.2542	58.200	17.771	0.000	920.183	1148.325
rain	-8.9874	12.293	-0.731	0.465	-33.082	15.108
precipi	20.0518	13.773	1.456	0.145	-6.943	47.047
meanwindspdi	23.4752	2.619	8.965	0.000	18.343	28.608
meantempi	-4.8493	0.784	-6.185	0.000	-6.386	-3.313
Hour	67.4023	0.691	97.571	0.000	66.048	68.756

Omnibus:	121619.247	Durbin-Watson:	1.591
Prob(Omnibus):	0.000	Jarque-Bera (JB):	15268994.352
Skew:	4.028	Prob(JB):	0.00
Kurtosis:	55.080	Cond. No.	2.25e+15

Results with only 'UNIT' as a dummy variable.

OLS Regression Results

Dep. Variable:	ENTRIESn_hourly	R-squared:	0.514
Model:	OLS	Adj. R-squared:	0.512
Method:	Least Squares	F-statistic:	280.0
Date:	Tue, 21 Jul 2015	Prob (F-statistic):	0.00
Time:	17:49:03	Log-Likelihood:	-1.1631e+06
No. Observations:	131951	AIC:	2.327e+06
Df Residuals:	131453	BIC:	2.332e+06
Df Model:	497		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	1634.1857	51.428	31.776	0.000	1533.388	1734.983
rain	-48.6777	12.936	-3.763	0.000	-74.032	-23.324
precipi	-8.0923	15.618	-0.518	0.604	-38.702	22.518
meanwindspdi	-3.6552	3.180	-1.149	0.250	-9.889	2.578
meantempi	-7.4800	0.781	-9.578	0.000	-9.011	-5.949

Omnibus:	126073.675	Durbin-Watson:	1.563
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18723273.237
Skew:	4.229	Prob(JB):	0.00
Kurtosis:	60.741	Cond. No.	2.77e+16

Results with 'day' added. Notice the increased success of the model.

Credit for my interpretation of the summary values of my results was obtained from the ArcGIS Resource Center.

Could I make this model better? Surely, but I do have to finish this paper someday. Now, let's interpret this army of results. I will only interpret the values for the results that include 'unit' and 'day' as a dummy variable. I will also, in full disclosure, describe only those results which I understand.

The first valuable result is our R^2 value. This is a measure of the model's performance and ranges from 0 to 1. 0 showing absolutely no correlation between the features and the values and 1 showing a perfect relationship between the two (rare). The Adjusted R^2 value is more accurate as it reflects the model's multiple variables. Our results show that approximately 51.4% of the variation in the values can be explained by the features.

The next valuable result is the coefficient, which is the strength of the relationship between the features and values. The coefficient represents a unit to unit change between the features and the values. As we can see, our result for rain is the greatest coefficient with a weighting of 48.68%.

The t statistic is used to determine if the feature (or explanatory variable) is statistically significant. Along with the p statistic this tells us if we can be confident the results are not due chance. The null hypothesis tells us that the weather features selected do not effect subway ridership. Our results for rain and meantempi have high t variables and very low (essentially zero) p values. Therefore, for both of these we can safely reject the null hypothesis in favor of the alternative, which is that these features affect subway ridership.

The F-statistic from the ANOVA test is not really necessary since we've already got the t statistic. However, we'll look at it anyway for fun. The F-statistic and the probability of the F-statistic is another determinator of whether the features (explanatory variables) in the model consistently relate to the values (dependent variables). This is another convincing value that shows a statistically significant model. However, this statistic should not be used if the Koenker

(BP) Statistic is also statistically significant. Considering I do not have this statistic, I have to ignore the F-statistic.

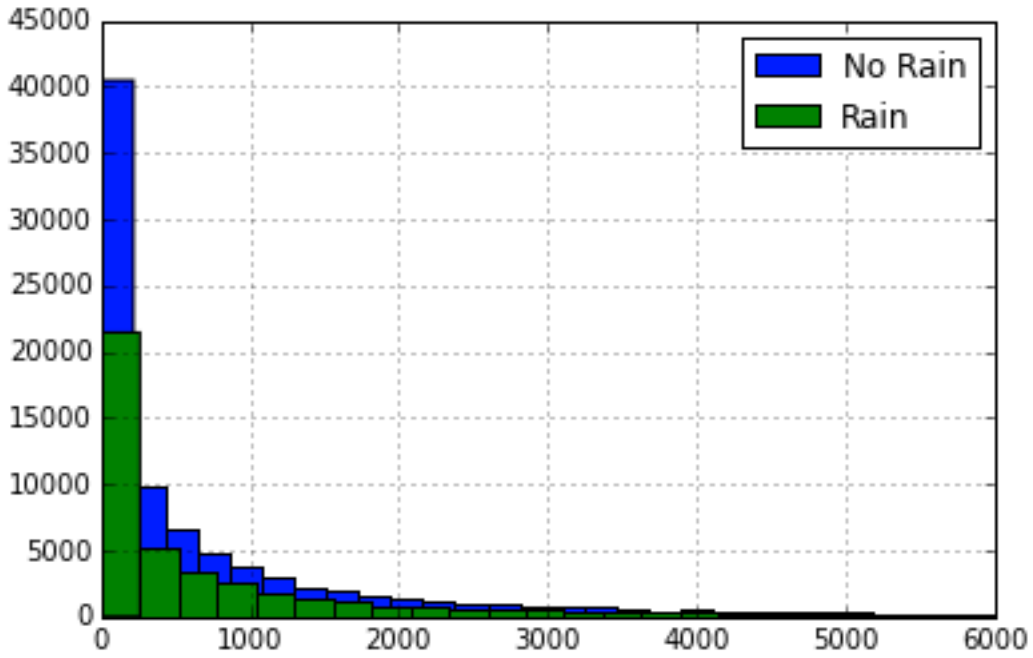
Now we are having fun. Let's look at the Jarque-Bera (JB) statistic. This indicates whether "the residuals (the observed/known dependent variable values minus the predicted/estimated values) are normally distributed." The H_0 is that they are normally distributed. The H_A is that there is not a normal distribution, which we can guess because we are dealing with weather data. It is interesting to see that this is true by the probability of JB being 0.00.

The summary of the results gives us a wealth of information to determine the accuracy, weighting and consistency of our model. I look forward to learning more powerful and detailed regression techniques in the upcoming projects.

3: Visualization

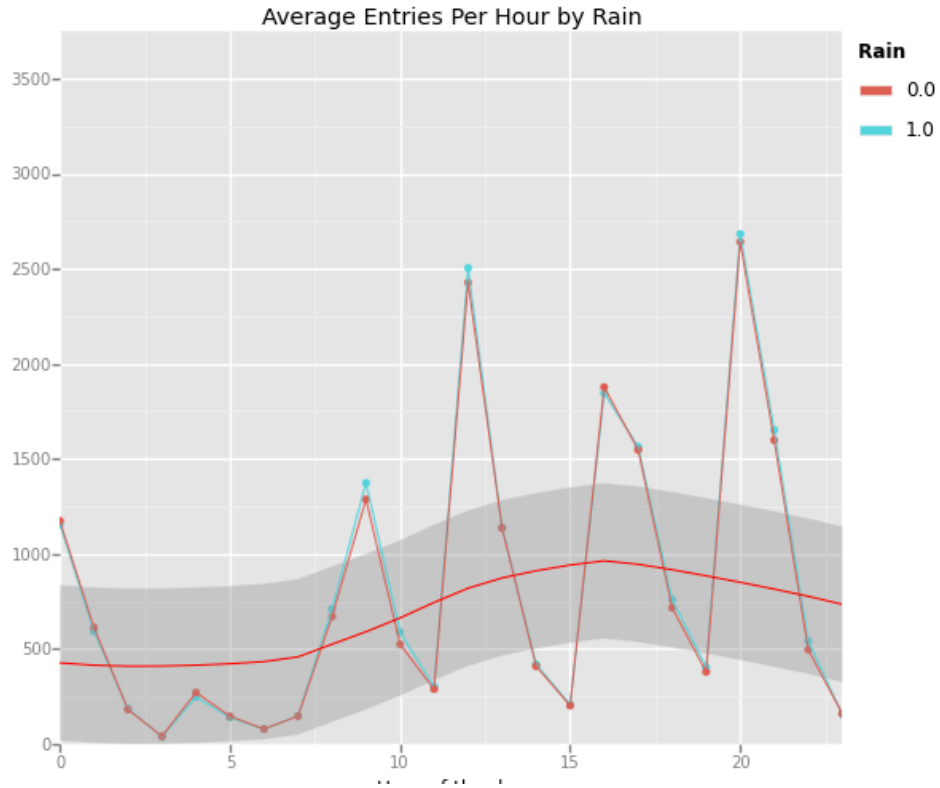
Data visualization gives us an opportunity to communicate findings in the data. The first data visualization uses matplotlib.pyplot and is a simple histogram for rainy and non-rainy days.

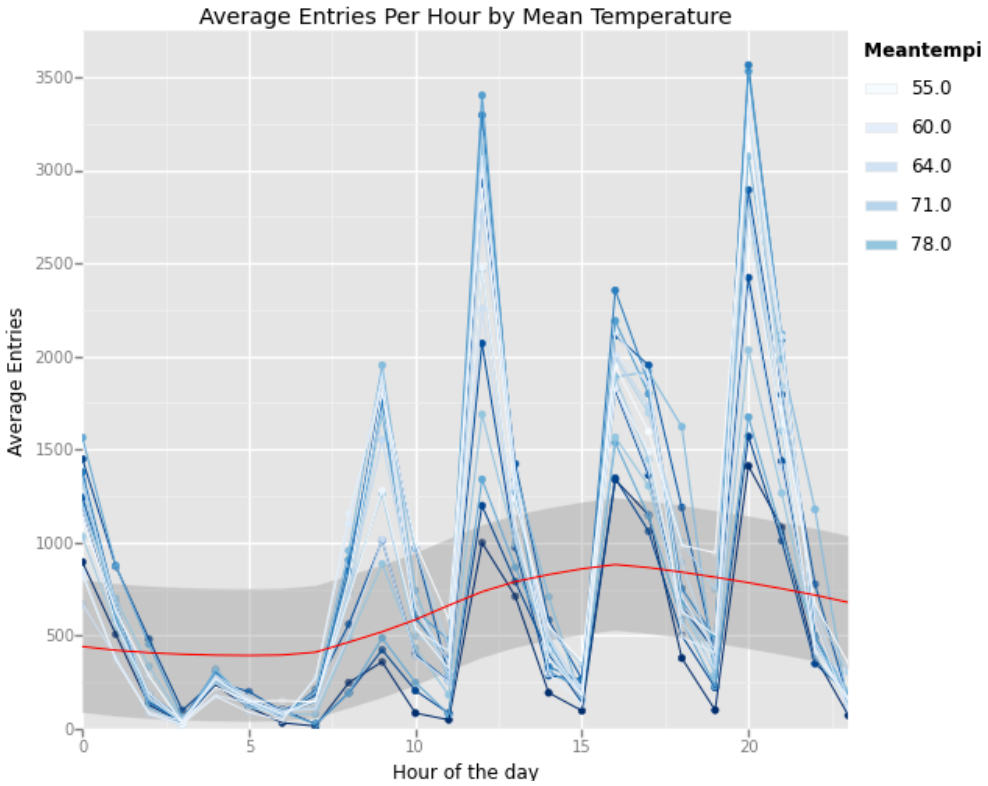
```
import matplotlib.pyplot as plt

plt.figure()
tsd_df['ENTRIESn_hourly'][tsd_df['rain']==0].hist(color='blue', bins=200, label='No Rain')
tsd_df['ENTRIESn_hourly'][tsd_df['rain']==1].hist(color='green', bins=200, label='Rain')
plt.xlim([0,6000])
plt.legend()
```

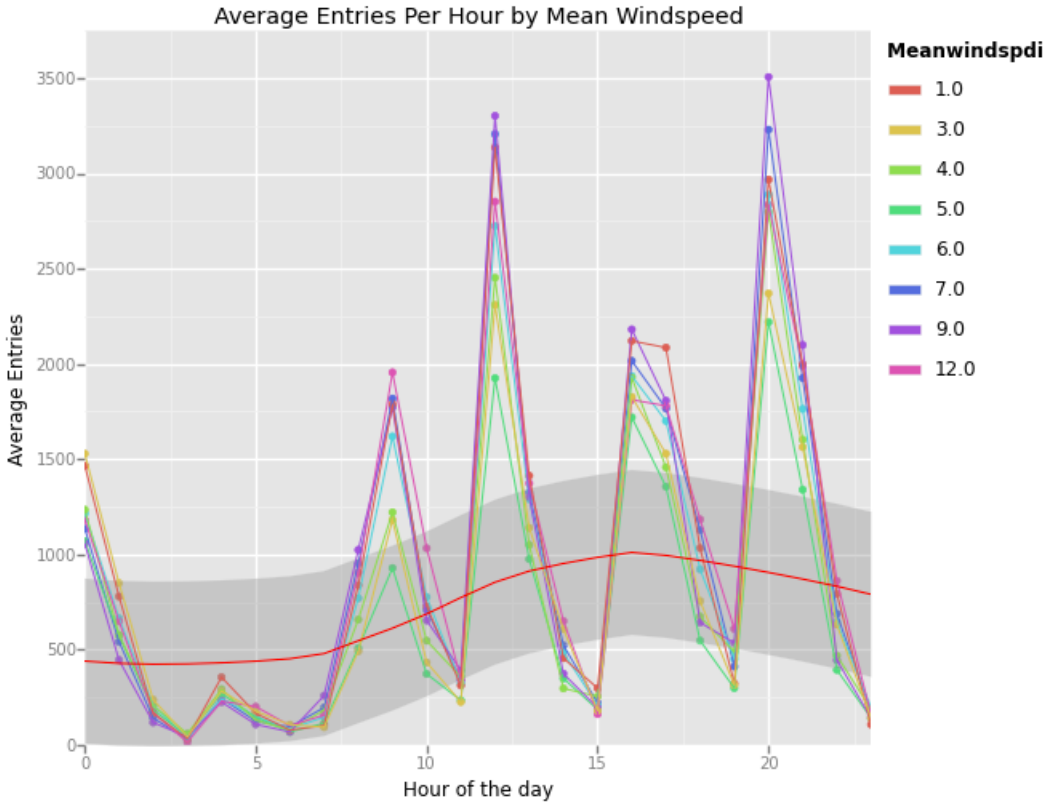


The next three charts show are all created by just changing the variable in the python below. These charts use ggplot instead of matplotlib.

```
##only three fields needed
tsd_limited = tsd_df[['Hour', 'ENTRIESn_hourly', 'rain']]
#get the average and group them by hour and rain
tsd_grp = tsd_limited.groupby(['Hour', 'rain'], as_index=False).mean()
#print it
print ggplot(tsd_grp, aes('Hour', 'ENTRIESn_hourly', color='rain')) +\
geom_point() + geom_line() + stat_smooth(color='red') + \
ggtitle('Average Entries Per Hour by Rain') +\
xlab('Hour of the day') + \
ylab('Average Entries') + \
xlim(0,23) +\
ylim(0,3750)
```




Average Entries Per Hour by Mean Temperature
Meantempi
55.0
60.0
64.0
71.0
78.0
Average Entries
0
500
1000
1500
2000
2500
3000
3500
0
5
10
15
20
Hour of the day


Average Entries Per Hour by Mean Windspeed
Meanwindspdi
1.0
3.0
4.0
5.0
6.0
7.0
9.0
12.0
Average Entries
0
500
1000
1500
2000
2500
3000
3500
0
5
10
15
20
Hour of the day

4: Conclusion

I think it is clear that there is more subway riders when it rains, when it gets cold and when it is windy. My Ordinary Least Squares regression showed indeed that there was a relationship between the features and the values; the Mann Whitney U test clearly showed that the datasets of 'Subway Riders when it Rains' and 'Subway Riders when it Doesn't Rain' are unlikely to be from the same population. With a 95% degree of confidence we can say that something happened to increase subway riders. The visualizations show that weather is clearly a factor. I can also conclude this from my R^2 value. I look forward to learning more regression tests to find one that may fit better.

5: Reflection

The real purpose of this exercise was to give students an opportunity to open the Data Scientists toolbox and start to understand what is in there. The dataset we used had a number of problems. First, it is only comprised on one month of data, a mild weather month at that. I did not like that I do not know how this data was composed. I can find MTA data online, as well as weather data so I would have felt better putting those sets together myself. There are numerous factors that influence subway ridership that are hidden contributors to this data, such as holiday's and special events. I am unhappy with my regression test and would like to use one more applicable, or at least try a number of them to get proper weighting on my variables, but to be honest…I'd not have finished this project on time, OLS is the only current regression test I know. I have to remember that I'm a beginner and I will learn more tools of the trade as this program continues over the next 12 months.

I must also reflect on this work and say that I'm not really happy with my results. Nearly everyday I run into a new data scientist tool that I find so useful I'm not sure how I did without it. For instance, today I watched the webcast on "Data Science from the Command Line" and I must say it seems to me not only invaluable, but surely a future requirement of this nanodegree. I do not however, have the time to rewrite this paper with my new command line skills…as much as I'd like to. I have to stick a flag in the ground and say "done", even though I'm tempted to do the whole thing over again in a different way just to see what those results look like. I've become a bit obsessed with getting my R^2 value higher.

5: Return to the data

Upon further consideration, I can make this better and have learned perhaps the most valuable lesson so far in this process. Step one is to sit back, relax and think about the data. Don't grab immediately for the hammer and start hitting. What variable do I want on my x axis, clearly for this lesson the number of turnstile entries is on the y…what is on the x? Also, could there be anything throwing off these numbers, any anomalies or outliers? Can they be identified?

Let's consider the outliers. What would throw off the number of subway riders? When I'm driving to or from work in my city of Minneapolis, when there is a sporting or political event then the traffic grinds to a halt. It doesn't matter what the weather is doing. Is it possible to see the events for May 2011 in NYC and thereby identify the dates that should be excluded? A google search turned up this Wikipedia page that says on May 2nd, "Thousands of people gather at Ground Zero of the September 11 attacks in New York to celebrate the news that Osama bin Laden has been killed." Surely this would affect subway ridership and therefore May 2nd should be excluded. Another search turned up this website that points out "…Mets and Yankees even face each other May 20 to 22 at Yankee Stadium—it's likely to be one of the hottest tickets of the season." Therefore, May 20th and May 22nd should also be excluded. There were other events, but these were the largest.

Now that we've identified the outliers and are determined to exclude them, let's next try to create accurate sets of data. We want to know if subway riders increases when it rains. Is subway ridership the same at 6 am as it is at 2 pm? Certainly not, we are all familiar with rush hour. So hour of day is part of x. Next, is subway ridership different on Monday and on Friday? Well, from experience I know that driving to work is different on Monday and Friday, therefore day of week is on x. In the end, the best is a combination of the two. Let's create a variable of day and hour, so 0-1 for Monday at 1 am, and 6-2 for Sunday at 2 am…etc. This will be our x.

How do we want to represent our y? Do we want to know the sum, mean, median, mode, min or max of turnstile entries per hour? Sum is going to be misleading, because there are more non-rainy days than rainy ones and therefore it will appear as if there are more riders when it is not raining. Median is interesting, but we already threw out the outliers we could identify so not necessary. Min and max do not make sense so that leaves us with mean.

Now that there is a better understanding of the data. Let's make set our hypotheses and then make a plan.

Our hypotheses is that when it rains more people take the subway.

H_0 = Subway ridership is unaffected by rain. *rain ridership = no rain ridership.*

H_A = More people take the subway when it rains. *rain ridership > no rain ridership.*

1. Remove May 2, May 20, May 21 and May 22 from the dataset. *Note: I was unable to safely remove these days from the data frame. Had to skip this step, but will return to it after more experience.*
2. Create a 'dayofweek' variable and fill it with the day of the week calculated off the date.
3. Create a 'dayhour' variable and concatenate 'dayofweek' '-' 'hour' into it.
4. Split the data frame into two, one data frame with rain and one without. Name them tsd_rain_df and tsd_no_rain_df.
5. Compare tsd_rain_df and tsd_no_rain_df and remove any non-paired entries. We want to know if ridership increased for a particular day and hour by comparing that day and hour when it doesn't rain to the same day and hour when it does. Non-matching pairs are not really doing us any good. Word of caution, the count for each of these tables will likely not be the same because of other variables, for instance Unit.
6. Run the Mann-Whitney U and report the p for a one-tail test. (Refer to hypothesis above). Prove these data frames are from two different populations.
7. Run the regression OLS and get the R2 value. What percentage of the dependent variable can be explained by the explanatory variable?
8. Create a simple density plot colored by rain and non-rain.

For the purposes of brevity, I will just show the results.

Mann-Whitney U Max: 1838475240

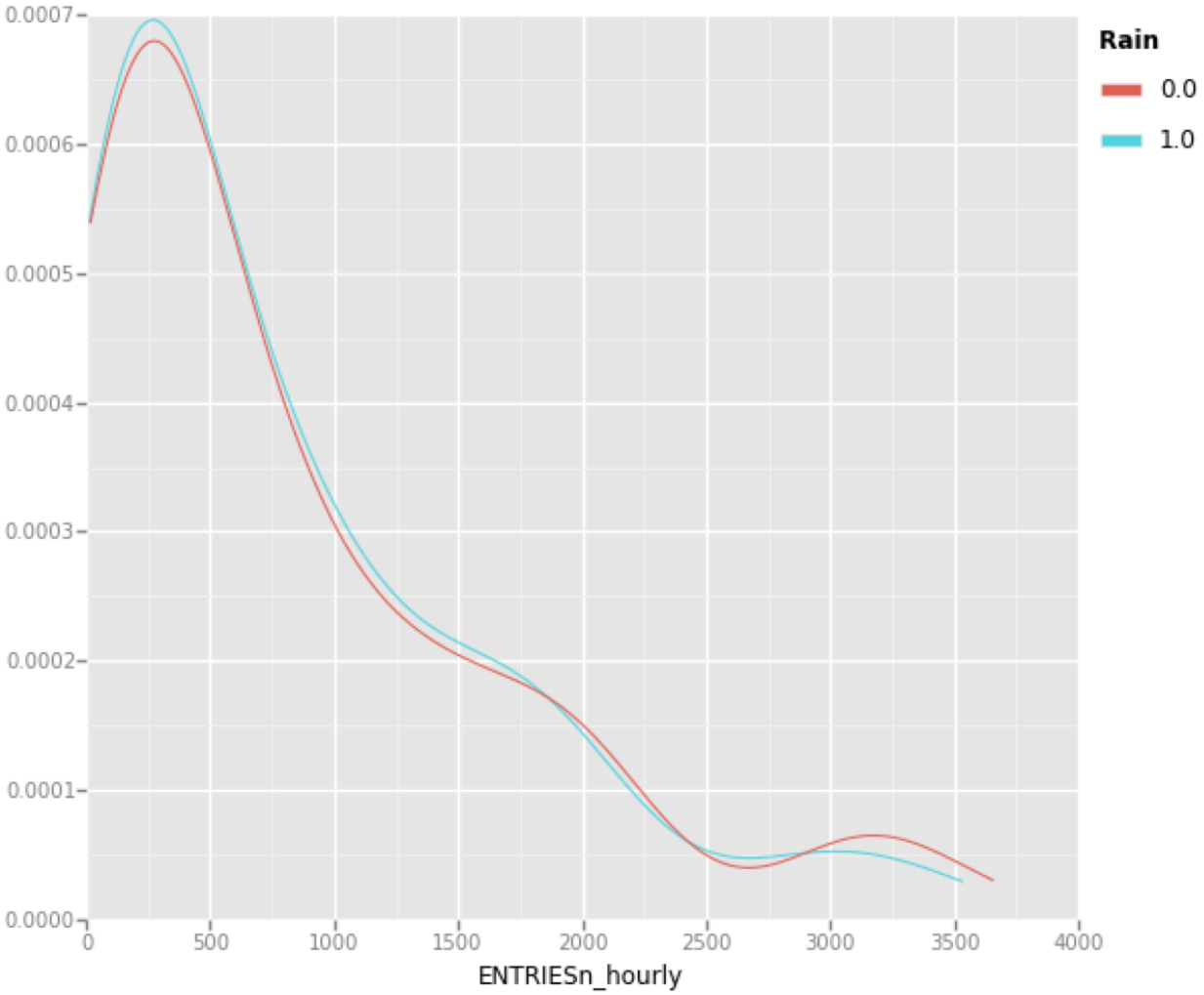
Mann-Whitney U Min: 919237620

Mann-Whitney U Test Statistic: 1823661869.5

Mann-Whitney U p-value one-Tail: 0.00888882337788

Mann-Whitney U p-value two-Tail: 0.0177776467558

R-squared: 0.531 *Note: Sadly only incrementally better than before.*


0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
500
1000
1500
2000
2500
3000
3500
4000
ENTRIESn_hourly
Rain
0.0
1.0