# Project 2 - Analyzing the NYC Subway Dataset

October 19, 2015

Author: Lee Clemmer

# 1 Project 2 - Analyzing the NYC Subway Dataset

This project is part of the Udacity Data Analyst Nanodegree.
*Project Specifications*

## 1.1 Questions

### 1.1.1 Statistical Test

**Selection, P value, Hypotheses** *Which statistical test did you use to analyze the NYC subway data? Did you use a one-tail or a two-tail P value? What is the null hypothesis? What is your p-critical value?*

In analyzing whether or not rain had a significant effect on ridership, I used the Mann-Whitney U-test statistical test. One of the assumptions underlying Welch's t-test is that the data are normally distributed, which the ridership data are not.

I used two-tailed test because I was not making any assumptions in regards to the directionality of the effect; in other words, whether ridership would be more or less on rainy days, just that it would be different.

The null hypothesis in this case was that the ridership means, as measured by turnstile entries, would be the same on rainy and non-rainy days. The p-critical value I used was $\alpha < 0.05$.

$H_0$: $\mu_{rainy} = \mu_{non-rainy}$
$H_A$: $\mu_{rainy} \neq \mu_{non-rainy}$

**Applicability** *Why is this statistical test applicable to the dataset? In particular, consider the assumptions that the test is making about the distribution of ridership in the two samples.*

One of the assumptions that the Welch's t-test makes is that the data are normally distributed. Because the data were extremely positively skewed, I chose to use the nonparametric Mann-Whitney U-test to determine whether there was a difference in means.

**Results** *What results did you get from this statistical test? These should include the following numerical values: p-values, as well as the means for each of the two samples under test.*

Because I used a two-tailed hypothesis, I doubled the p-value returned by the `scipy.stats.mannwhitneyu` function as it is one-tailed by default. The resultant p value was 0.0499998, just under the alpha value. Therefore, we reject the null hypothesis. The means were as follows:

- Mean with rain: $\bar{x}_{rainy} = 1105.45$
- Mean without rain: $\bar{x}_{non-rainy} = 1090.28$

**Significance** *What is the significance and interpretation of these results?*

We can say we have significant statistical evidence that ridership is slightly higher on rainy days as opposed to non-rainy days.

### 1.1.2 Linear Regression

**Approach** *What approach did you use to compute the coefficients theta and produce prediction for ENTRIESn_hourly in your regression model:*

- *OLS using Statsmodels or Scikit Learn*
- *Gradient descent using Scikit Learn*
- *Or something different?*

I applied both the Ordinarly Least Squares approach via the statsmodels python package as well as the Gradient Descent approach using Scikit Learn's `SGDRegressor`.

**Features** *What features (input variables) did you use in your model? Did you use any dummy variables as part of your features?*

For the OLS regression, I used `Hour`, `maxpressurei`, `meantempi`, `maxdewpti`, `meanwindspdi` as features in addition the UNIT dummy variables.

**Feature Selection** *Why did you select these features in your model? We are looking for specific reasons that lead you to believe that the selected features will contribute to the predictive power of your model.*

- *Your reasons might be based on intuition. For example, response for fog might be: "I decided to use fog because I thought that when it is very foggy outside people might decide to use the subway more often."*
- *Your reasons might also be based on data exploration and experimentation, for example: "I used feature X because as soon as I included it in my model, it drastically improved my $R^2$ value."*

My final list of features was based on exploration and experimentation rather than on intuition. At first, I discovered that simply adding all features would yield the highest $R^2$ result. But after reading up on the bias-variance tradeoff I decided to limit my features to limit variance. I then tested the model with each (non-dummy) feature individually to see which had the biggest impact on the $R^2$ results. I then chose the features I that had the most impact on $R^2$ and would therefore have the best predictive power, while also avoiding features of the same type (e.g. I did not include two features that both measured temperature, like `mintempi` and `meantempi`). Somewhat suprising to me was that neither the `rain` nor the `precipi` features had much predictive powers, although we found out it does have effect ridership. In the end, I left these out.

**Parameters** *What are the parameters (also known as "coefficients" or "weights") of the non-dummy features in your linear regression model?*

- `constant`: 9332.1106
- `Hour`: 65.4174
- `maxpressurei`: -267.6333
- `meantempi`: -12.1288
- `maxdewpti`: 3.5295
- `meanwindspdi`: 25.7264

**Coefficients of Determination** *What is your model's $R^2$ (coefficients of determination) value?*

0.48

**Fit** *What does this $R^2$ value mean for the goodness of fit for your regression model? Do you think this linear model to predict ridership is appropriate for this dataset, given this $R^2$ value?*

The $R^2$ means I am explaining 48% of the linear variation of 465 subway turnstile entries over the span of 30 days in May 2011. Given this particular dataset and the constraint of a linear model, I do think this particular model is appropriate, striking a deliberate balance in the bias-variance tradeoff.

### 1.1.3 Visualization

*Please include two visualizations that show the relationships between two or more variables in the NYC subway data.*

*Remember to add appropriate titles and axes labels to your plots. Also, please add a short description below each figure commenting on the key insights depicted in the figure.*

**Histograms** *One visualization should contain two histograms: one of ENTRIESn_hourly for rainy days and one of ENTRIESn_hourly for non-rainy days.*

- *You can combine the two histograms in a single plot or you can use two separate plots.*
- *If you decide to use to two separate plots for the two histograms, please ensure that the x-axis limits for both of the plots are identical. It is much easier to compare the two in that case.*
- *For the histograms, you should have intervals representing the volume of ridership (value of ENTRIESn_hourly) on the x-axis and the frequency of occurrence on the y-axis. For example, each interval (along the x-axis), the height of the bar for this interval will represent the number of records (rows in our data) that have ENTRIESn_hourly that falls in this interval.*
- *Remember to increase the number of bins in the histogram (by having larger number of bars). The default bin width is not sufficient to capture the variability in the two samples.*

In [1]:
```
import pandas as pd
%matplotlib inline
from pylab import *
from ggplot import *
```

In [2]:
```
df = pd.read_csv('turnstile_data_master_with_weather.csv')
plt.figure()
plt.ylabel('Frequency')
plt.xlabel('Value of ENTRIESn_hourly')
plt.title('Fig. 1: Histogram of ENTRIESn_hourly')
hist_norain = df[(df.rain == 0) & (df.ENTRIESn_hourly < 10000)]['ENTRIESn_hourly'].hist(bins=50
hist_rain = df[(df.rain == 1) & (df.ENTRIESn_hourly < 10000)]['ENTRIESn_hourly'].hist(bins=50, 
plt.legend()
```

Out[2]:
```
<matplotlib.legend.Legend at 0x10e1f3f90>
```

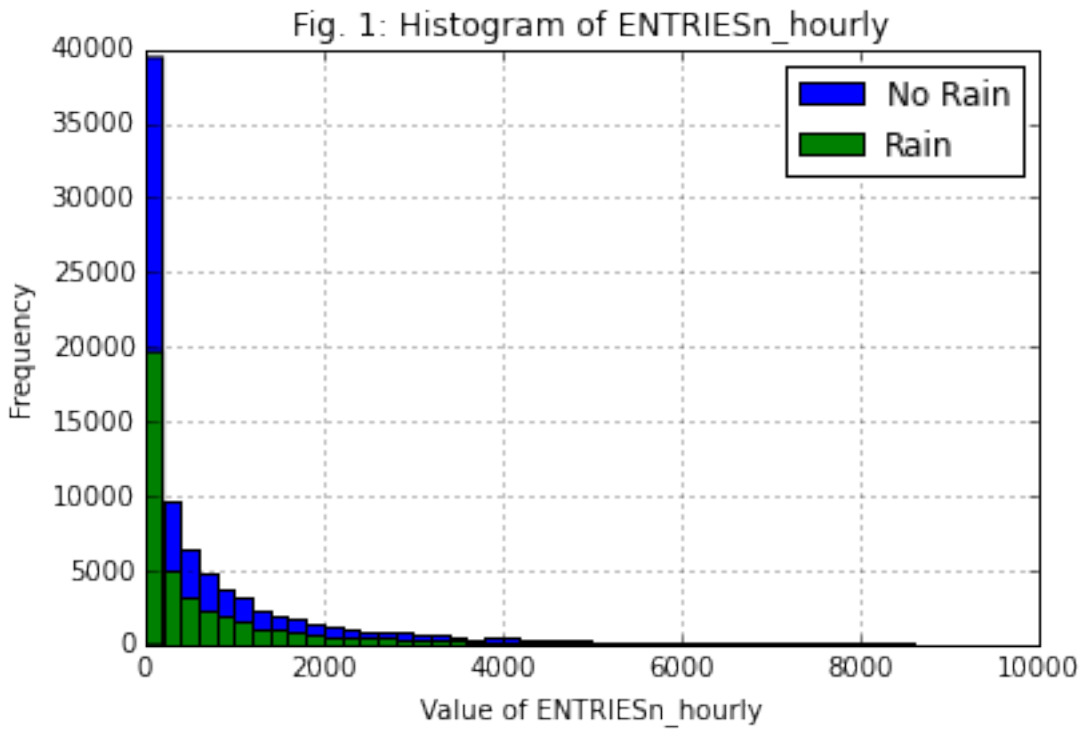



*Figure 1* is a histogram of hourly entries for both rainy and non-rainy days. Both distributions are extremely positively skewed. The x-axis is capped at 10000 to filter out outliers for sake of clarity.

**Additional Visualization** *One visualization can be more freeform. You should feel free to implement something that we discussed in class (e.g., scatter plots, line plots) or attempt to implement something more advanced if you'd like. Some suggestions are:*

- *Ridership by time-of-day*
- *Ridership by day-of-week*

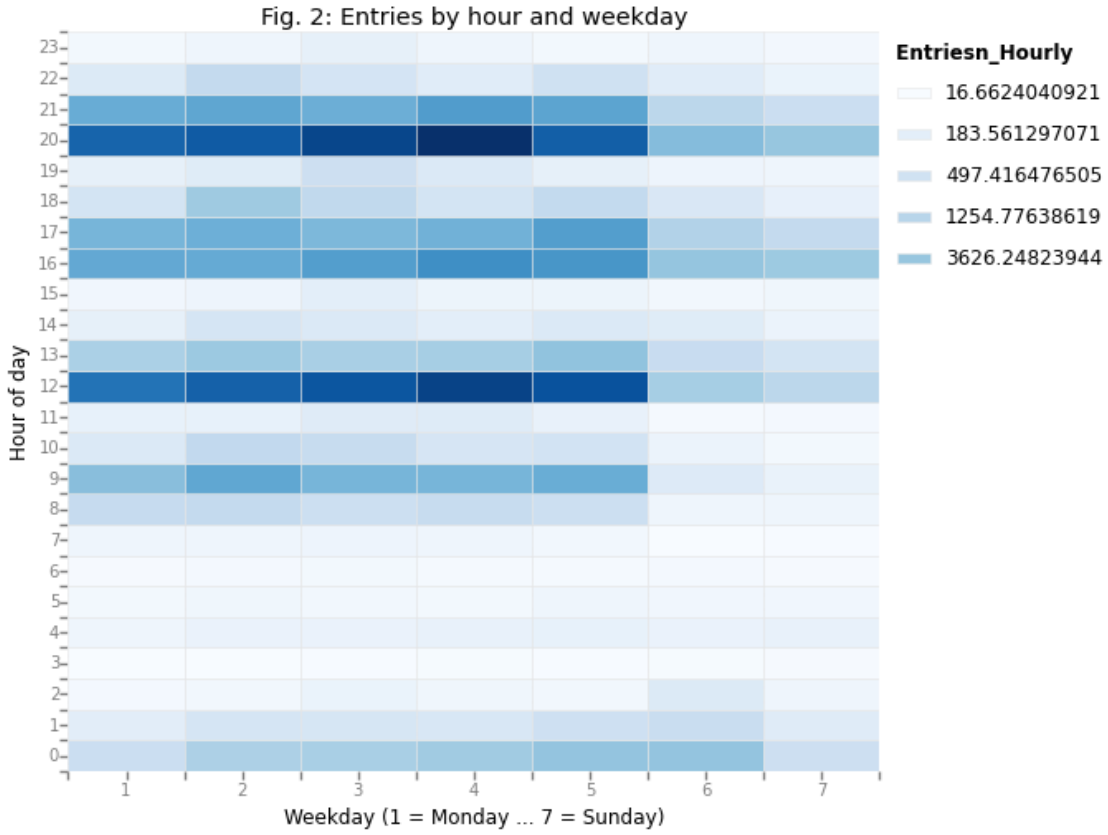
In [4]:
```
pd.options.mode.chained_assignment = None

# Add week day numbers to data
df['Datetime'] = pd.to_datetime(df['DATEn'])
df['Weekday'] = df['Datetime'].apply(lambda x: x.isoweekday())

# Group by Weekday and Hour
group = df[['Weekday','Hour','ENTRIESn_hourly']].groupby(['Weekday','Hour']).mean()
group.reset_index(inplace=True)

# Plot
plot = ggplot(group, aes('Weekday','Hour',fill='ENTRIESn_hourly')) + \
       geom_tile() + ggtitle("Fig. 2: Entries by hour and weekday") + \
       xlab("Weekday (1 = Monday ... 7 = Sunday)") + ylab("Hour of day")

print plot
```




```
<ggplot: (292393053)>
```

*Figure 2* is a heatmap of the average *ENTRIESn_hourly* by weekday and hour of the day. We discover that the busiest times of the week are on Thursday at noon and 8pm. Highest ridership times for all days are around 9am, noon, 4 to 5pm, and again at 8-9 pm. Ridership appears lowest, regardless of weekday, during 3am. Saturday and Sunday are noticeably less busy than work days. The busiest days for the midnight hour are Friday and Saturday, possibly explained by people going out for evening entertainment.

## 1.2 References


- statsmodels.regression.linear_model.OLS Documentation
- sklearn.linear_model.SGDRegressor Documentation
- Linear Regression Example
- Udacity discussion: Linear Regression
- Udacity discussion: Feature Selection: why not use all?
- Wikipedia: Bias–variance tradeoff
- Wikipedia: Dummy variable (statistics)
- Wikipedia: Multicollinearity
- Are the model residuals well-behaved?

- Endogenous variable
- scipy.stats.linregress Documenation
- matplotlib: PyPlot Tutorial
- Python Datetime Documentation
- ggplot Documenation
- ggplot2: Quick Heatmap Plotting