Analyzing NYC Data Set

Ivan Lee 5/29/2015 Course: Intro Data Science

Section 0: Resources Used

1. http://stackoverflow.com/questions/18299523/basic-example-for-pca-with-matplotlib
2. http://stackoverflow.com/questions/7941207/is-there-a-function-to-make-scatterplot-matrices-in-matplotlib
3. http://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test
4. http://stackoverflow.com/questions/3949226/calculating-pearson-correlation-and-significance-in-python
5. http://matplotlib.org/users/legend_guide.html

Section 1: Statistical Reasoning

1.1-1.4 Statistical tests used and reason for usage:

1. T-Test: Once the matrix scatter plot (see below) was analyzed for pairs of correlated independent variables, the T-test was used to test the null hypotheses that that there was no statistical difference between the aforementioned pair. A two tailed test was used at a 95% confidence interval with the p-critical value set at 5%. The null hypothesis held for all scrutinized pairs. This test assumes that samples have a Normal distribution and dissimilar variances. This is a good starting assumption since most natural phenomenon have a normal distribution. Results are in Table 1.
2. Pearson's correlation coefficient: Once independent variables were grouped by correlation, the next step was to select the best in each individual group to represent that phenomenon in the model. The variables were chosen on how highly they correlated with the dependent variable, thus the ones with the highest correlation coefficient per group was chosen. Results are in Table 2.
3. U-test: For the remaining independent variables it was determined if they could produce a dichotomy of samples when the population of responses (ENTRIESn_hourly) was coerced into binary bins according the values of the independent variable. Each independent variable was split into one side of its mean or the other. The samples of the dependent variable that corresponded to each side was used for the U-test. The U-test from all splits were able to reject the null hypotheses that the samples came from the same population i.e. the distributions of the split samples are statistically similar. This test assumes all responses are oridinal, which they are and values from

both groups are independent. The cut off value for rejecting the null hypothesis $P(X > Y) + 0.5\ P(X = Y) > 0.5$, where X and Y are the respective populations. Results are in Table 3.

4. Data normalization: This was done to reduce round off error and to "center" values as to obtain better results from the regression model.
5. R-Square: this was used to assess the total variance explained by our model. A score closer to 1 would indicate a better fitting model i.e a more accurate prediction using in hand data.

Section 2: Model and Results

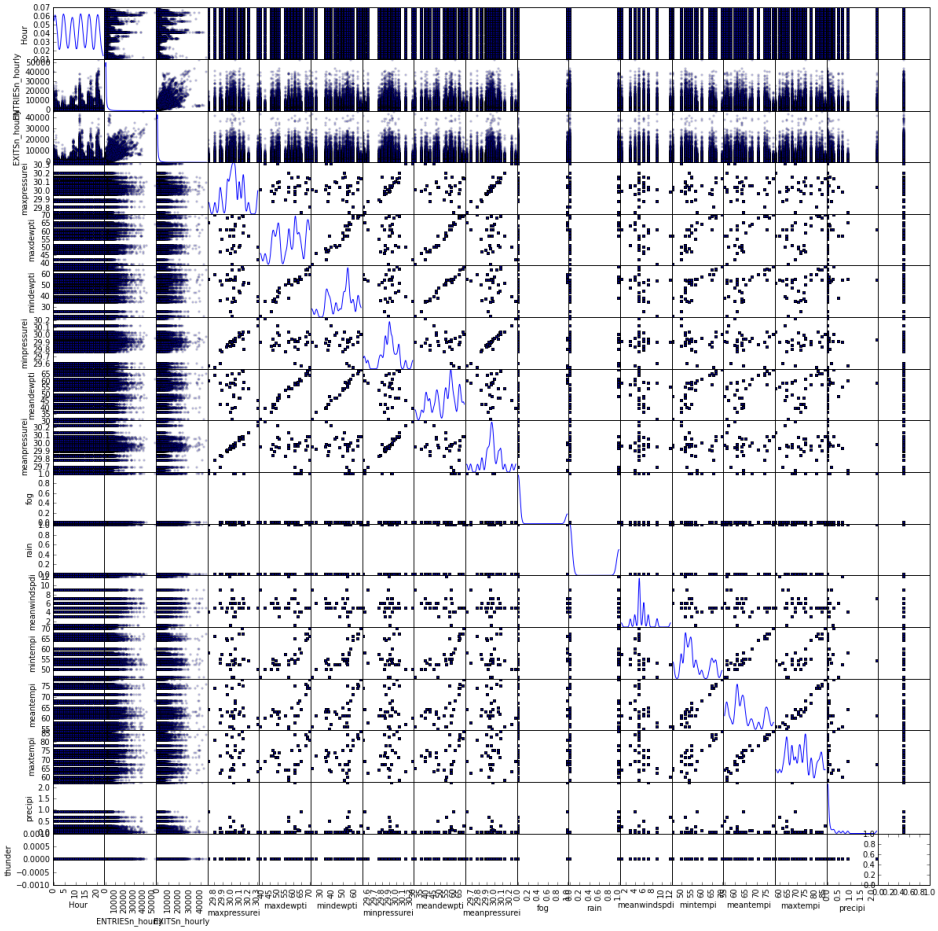
- OLS was used for this model. The Beta hat vector was found using the normal equation, which was programmed in long hand. Since the response was quantitative and the assumption was that is was distributed normally, this seemed a good starting point. The hope would be that the response would also distribute itself normally and with homoskedasticity across the regression line, which would serve as the mean.

- The variables used in the model and their respective coefficients (no dummy variables were used):
    - Constant: -5.57346676e-13
    - Hour: 5.83743404e-02
    - EXITSn_hourly: 7.34415571e-01
    - mindewpti: 7.03726655e-03
    - minpressurei: -6.22627983e-03
    - fog: 6.67611844e-03
    - rain: -5.06251653e-03
    - meanwindspdi: 7.02787826e-03
    - mintempi: -1.63916160e-02
    - precipi: -2.15648418e-03
- The R-square from this model is .56. This is an improvement over the model used in lesson 3 which yield a R-square of .43.

- The unused variables were culled due to possible over representation of the variables used, which they were highly correlated with. The model variables were chosen based on the correlation they showed with the response variable. See section 1 for full explanation. No intuitive reasoning was used as to reduce model bias.

Section 3: Visualization

The first graph is a matrix scatterplot. All variables except for 'Unnamed: 0', 'UNIT', 'DATEn', 'TIMEn', 'DESCn' were plotted against each other in a Cartesian fashion. This visually displays the relationship the variables independent and dependent have with each other. The selection of candidate variables for stage 1 elimination was based on this graph.



The second graph shows the distribution of ENTRIESn_hourly when it was condition on rain or no rain. This is an example of the how the response can be divided into two subpopulations conditioned on a prior dependent variable.

Chart 1 shows the entire distribution. Unfortunately the large bars at the left create too large of a scale for lower frequency bars to be seen.

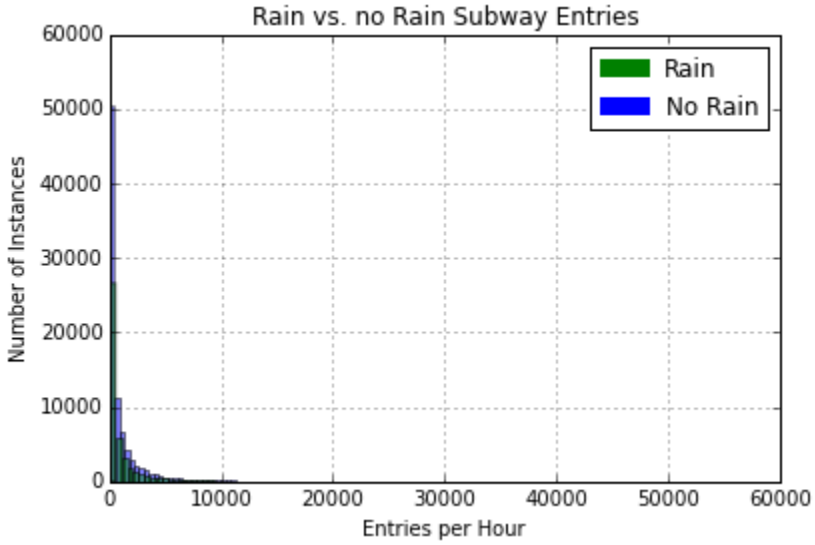


Chart 2 shows finer detail for 0-1000 entries per hour.

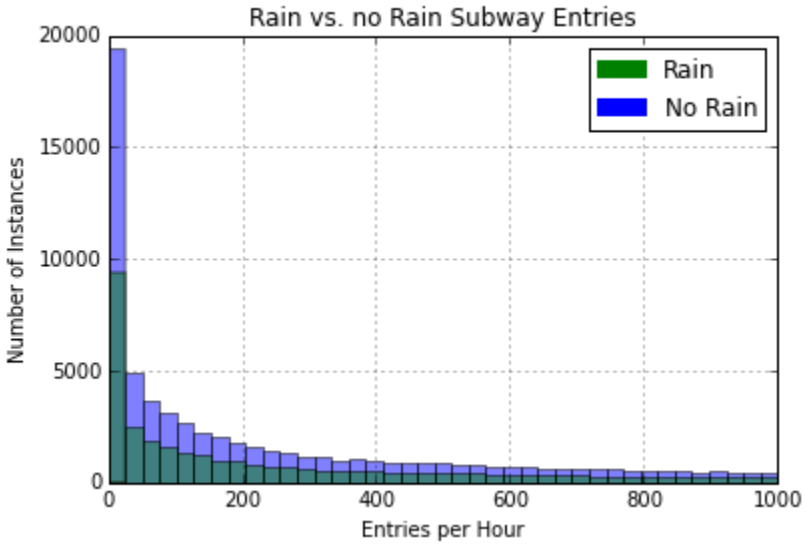


Chart 3 shows 2000-16000 entries per hour.

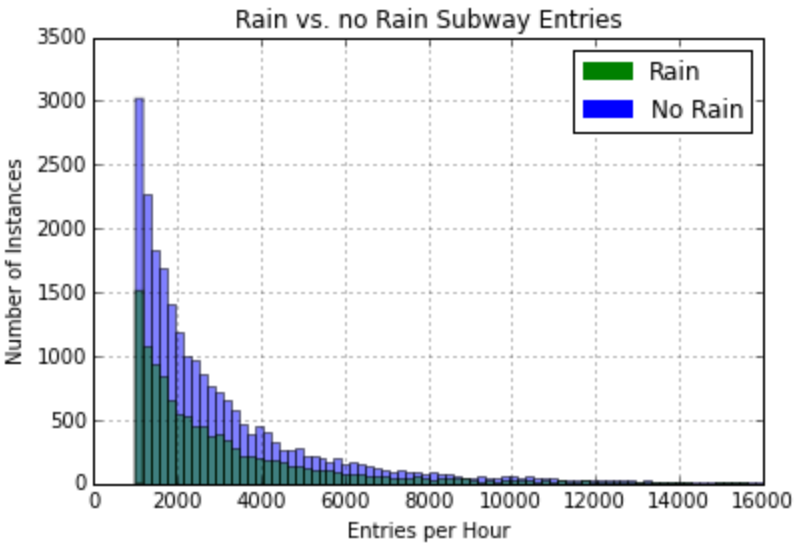
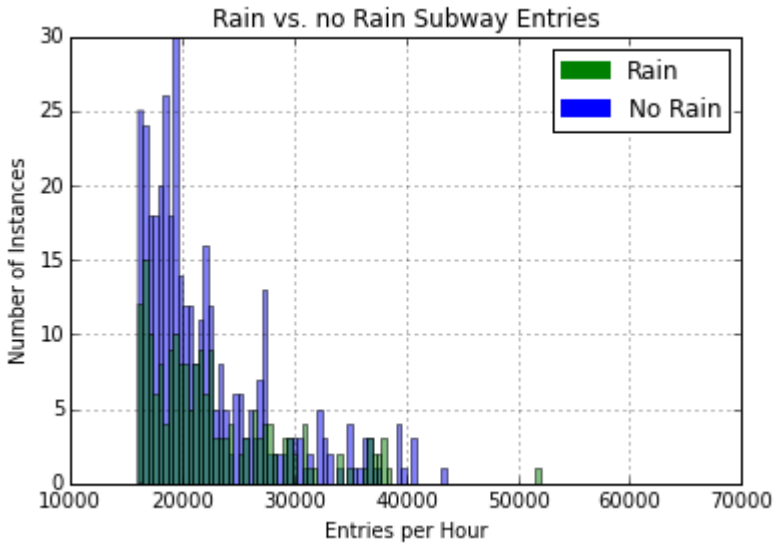


Chart 4 shows 16000 to the max entries per hour.



Section 4: Conclusions

Based purely on the histogram and frequencies, the ridership is greater for no rain than when there is rain. Summing all the no rain entries we get 95,777,720.0 and rain entries sum at 48,754,607.0. The t-test score for rain vs no rain is -1.1042068992726768 with a p-value of 0.26950646351201701. While the p-value is below the .05 p-critical value there is room for no rain being an influence on positive ridership.

This conclusion is supported by the OLS coefficient which is negative for when there is rain. In addition, the U-test score for no rain/rain implies that these two subpopulation have

statistically different distributions and thus the increased ridership during no rain should not be considered a random occurrence.

Section 5: Reflections

The shortcoming in the dataset was that it contained too many correlated variables. This was overcome by eliminating them. However, the results could have been much better if other factors besides weather were included, such as traffic or occurrence of sporting events.

The shortcomings in the analysis is that in hindsight including weekday or weekend might have improved results. OLS might not have been the best model if the relationship was non-linear. Also the t-tests assumed that all variables had normal distributions, which they might not have had. However, with the scatter matrix confirming the t-test results, I believe this to be a moot point.

Appendix

Table 1: t-test scores

Variables compared	Null-Hyp	t-score	p-value
'maxpressurei', 'minpressurei',	True,	3.41e-08,	0.99
'maxpressurei', 'meanpressurei'	True,	-1.9-08,	0.99
'maxdewpti', 'mindewpti'	True,	2.6e-14,	0.99
'maxdewpti', 'meandewpti'	True,	4.4e-14,	0.99
'mindewpti', 'meandewpti'	True,	1.8e-14,	0.99
'minpressurei', 'meanpressurei'	True,	-5.3e-08,	0.99
'mintempi', 'meantempi'	True,	-2.8e-13,	0.99
'mintempi', 'maxtempi'	True,	-2.4e-13,	0.99
'meantempi', 'maxtempi'	True,	4.3e-14	0.99

Table 2: Pearson's correlation coefficient as correlated to Response Variable

Variable	Correlation Coefficient	p-value
'maxpressurei'	-0.017084364969893878,	5.4246586702236083e-10
'minpressurei',	-0.020517187364530752,	9.0830838233646152e-14
'meanpressurei',	-0.016128237261487311,	4.6589425056375821e-09
'mindewpti',	-0.020135113918292414,	2.5783326666311019e-13
'maxdewpti',	-0.0098932776645542456,	0.00032589882963662126
'meandewpti',	-0.016197867524249149,	3.9994673114641798e-09
'mintempi',	-0.029034361121668188,	5.1410057362955601e-26

'maxtempi', -0.014303225741266237, 2.0376253318773601e-07
'meantempi', -0.022796041448386178, 1.2145630079843014e-16

Table 3: U-test Response population split according to mean of dependent variable

Varaible	U-Score	p-value
'Hour',	1487799274.0,	0.0
'EXITSn_hourly'	273736727.0,	0.0)),
'maxpressurei'	2017808915.0,	1.7556390589564093e-78
'maxdewpti'	2153701400.0,	0.00027055879235769959
'mindewpti'	2076443841.0,	0.012437502068012219
'minpressurei'	2100822885.5,	7.8199752743736322e-09
'meandewpti'	2076443841.0,	0.012437502068012219
'meanpressurei'	2096825277.5,	8.5893067839946357e-28
'fog'	1189034717.5,	1.9570617095483498e-06
'rain'	1924409167.0,	0.019309634413792565
'meanwindspdi'	1940299198.0,	1.6916325608871966e-124
'mintempi'	1990512284.0,	1.2960150501683112e-05
'meantempi'	1928197952.5,	0.24653832859158603
'maxtempi'	2165921182.0,	0.055616817673773154
'precipi'	1386215493.5,	0.32741731751535574
'thunder'	0.0,	0.0

Code used for Analysis:

```
    # -*- coding: utf-8 -*-
"""
Created on Fri May 29 11:53:04 2015



@author: Ivan
"""
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#from matplotlib.mlab import PCA
```

```
from pandas.tools.plotting import scatter_matrix
import scipy.stats
import statsmodels.api as sm
from scipy.stats.stats import pearsonr
import matplotlib.patches as mpatches


file = "C:/Users/Ivan/Desktop/UdacityDAnano/turnstile_data_master_with_weather.csv"
weatherdata=pd.read_csv(file)
df=weatherdata
df.is_copy=False




## describe data
'''
df.head()
df.describe()
list(df.columns.values)
df.info()
'''


#drop extraneous data
df.drop(['Unnamed: 0', 'UNIT', 'DATEn', 'TIMEn', 'DESCn'], axis=1, inplace=True)


#Df=df.drop([df.columns[[0, 1, 2,3,5]]], axis=1)
```

```
##normalize data
def normalize(df):
'center the data using the mean and sigma from dataset'
return (df - df.mean())/df.std()
#U_test
def utest(x1,x2):
[U,p]= scipy.stats.mannwhitneyu(x1, x2)
return U,p


'"Welch's t_test"'
def tcompare(x,y):
t=scipy.stats.ttest_ind(x,y,equal_var=False)
if t[1] >=.5:
return True, t
else:
return False, t


def corrtest(x,y):
p=pearsonr
if p[0] >=.5:
return i,j, True, p[0]
else:
return i,j, False,p[0]


def compute_r_squared(data, predictions):
```

```
r_squared=1-np.sum((data-predictions)**2)/np.sum((data-np.mean(data))**2)
# your code here
return r_squared
#normalize data
dfnorm=normalize(df)


#visuablize scatterplot
scatter_matrix(df, alpha=0.2, figsize=(15, 15), diagonal='kde')


#use ttest to test correlation for visualized relationships
corlist=[('maxpressurei','minpressurei'),('maxpressurei','meanpressurei'),('maxdewpti','mindewpti'),('maxdewpti','meandewpti'), \
('mindewpti','meandewpti'),('minpressurei','meanpressurei'),('mintempi','meantempi'),('mintempi','maxtempi'),('meantempi','maxtempi')]


plist=[]
for i,j in corlist:
plist.append((i,j,tcompare(df[i],df[j])))


#pick best of correlated variables
pcorlist=[]
for i,j in corlist:
pcorlist.append((i,pearsonr(dfnorm['ENTRIESn_hourly'],dfnorm[i])))
```

```
pcorlist.append((j,pearsonr(dfnorm['ENTRIESn_hourly'],dfnorm[j])))
```

```
#Use MAnn Whitney to decide
Ulist=[]
for i in df.columns.values:
mu=dfnorm[i].mean()
x=dfnorm[dfnorm[i]>mu]['ENTRIESn_hourly']
y=dfnorm[dfnorm[i]<=mu]['ENTRIESn_hourly']
Ulist.append((i,utest(x,y)))
```

```
#drop the worst of the correlated varaibles
df.drop(['maxpressurei','meanpressurei','maxdewpti','meandewpti','meantempi','maxtempi'],
axis=1, inplace=True)
```

```
#plot no rin vs rain
plt.figure()
space=np.linspace(16000,60000,100)
df[df['rain']==0]['ENTRIESn_hourly'].hist(bins=space,alpha=.5) # your code here to plot a
historgram for hourly entries when it is raining
df[df['rain']==1]['ENTRIESn_hourly'].hist(bins=space,alpha=.5) # your code here to plot a
historgram for hourly entries when it is not raining
blue = mpatches.Patch(color='blue', label='No Rain')
#plt.legend(handles=[blue])
green=mpatches.Patch(color='green',label='Rain')
plt.legend(handles=[green,blue])
```

```
plt.ylabel('Number of Instances')
plt.xlabel('Entries per Hour')
plt.title('Rain vs. no Rain Subway Entries')
plt.show()
```

```
#run regression analysis
Y=np.matrix(dfnorm['ENTRIESn_hourly'])
X=np.matrix(dfnorm.drop(['ENTRIESn_hourly','thunder'], axis=1,inplace=False))
X = sm.add_constant(X)

hatm=np.dot(np.linalg.inv(np.dot(np.transpose(X),X)),np.dot(np.transpose(X),np.transpose(Y)
))
pred=np.dot(X,hatm)
predictions=np.ravel(pred)
print(compute_r_squared(dfnorm['ENTRIESn_hourly'],predictions))
```