Analyzing the NYC Subway Dataset

Questions

Overview

This project consists of two parts. In Part 1 of the project, you should 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. Please use this document as a template and answer the following questions to explain your reasoning and conclusion behind your work in the problem sets. You will attach a document with your answers to these questions as part of your final project submission.

Section 0.References

Please include a list of references you have used for this project. Please be specific - for example, instead of including a general website such as stackoverflow.com, try to include a specific topic from Stackoverflow that you have found useful.

- The following are references that were very useful
- http://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/
- http://stats.stackexchange.com/questions/116315/problem-with-mann-whitney-u-test-in-scipy
- several wiki pages

Section 1.Statistical Test

1.1 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?
1.2 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.
1.3 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.
1.4 What is the significance and interpretation of these results?

1.1 First I wanted to test the data to check if it's a normal distribution, so I plotted the histogram and found the data is not a normal distribution. A Welch's T test will not give the desired results. I used Mann-Whitney U test which is a non-parametric test to analyze the data. A two tail test is more suitable for this data, because the data can be on either side of the tails in the dataset. Null hypothesis states that there is no significant difference in ridership during rainy and non-rainy days. (in other words the distributions for ENTRIESn_hourly rainy and non-rainy are same) for P-critical value is set at 0.05. To reject the null hypothesis the probability should be less than 0.05. For a two tailed test the probability is split between the two tails with 0.025 on each side. Mann Whitney U test is suitable for non-normal distributions. This distribution is a non-normal distribution since the histogram that was plotted was not a normal distribution.

1.2 Results from the Mann Whitney U test are as follows.
Mean Rain 2028.1960354720918, Mean No Rain 1845.5394386644084, U value is 153635120.5
p-value is very small it was nan when I used the scipy function scipy.stats.mannwhitneyu. Later I found that value to be 5.4826938714192724e-06 – reference.

(http://stats.stackexchange.com/questions/116315/problem-with-mann-whitney-u-test-in-scipy)

1.3 Since the p value is less than p-critical value of 0.05, the null hypothesis rejected. Now we can state that there is a significant increase in ridership on rainy days when compared to non-rainy days.

Section 2.Linear Regression

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

1. Gradient descent (as implemented in exercise 3.5)
2. OLS using Statsmodels
3. Or something different?

2.2 What features (input variables) did you use in your model? Did you use any dummy variables as part of your features?
2.3 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."

2.4 What are the coefficients (or weights) of the non-dummy features in your linear regression model?
2.5 What is your model's R^2 (coefficients of determination) value?
2.6 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?

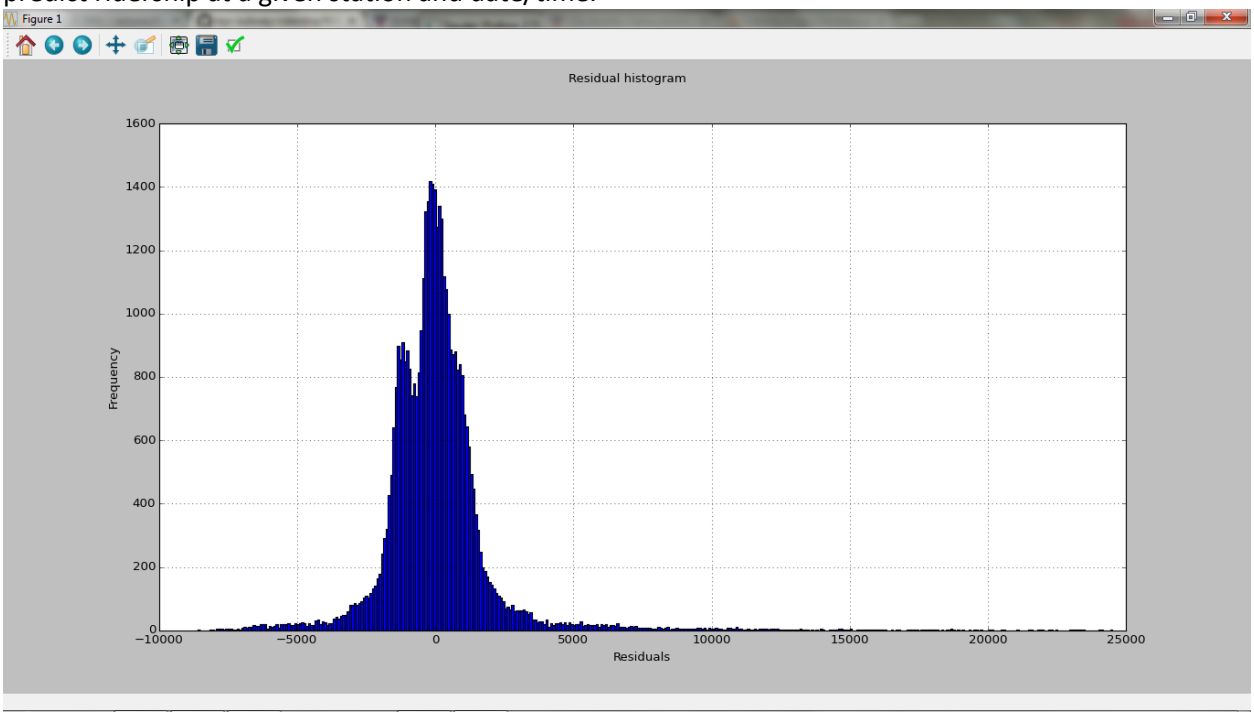
2.1 I used gradient descent.
2.2 The following features were used meanprecipi, meantempi, meanpressurei, meanwspdi. I used UNIT and hour as dummy variables. The UN IT and hour are chosen as dummy variables because ridership and hour are not linear functions. I added UNIT as a dummy variable for this same reason.

2.3 I selected these features that are all weather related, because our analysis is to find the effect of the weather on ridership and all of these are a linear function of ridership. On days that are rainy and foggy the ridership on the subway increases due to hazardous driving conditions, people will prefer taking the subway over driving to their destination, this will reject our null hypothesis.
2.4 The coefficients of non-dummy features

('meanprecipi', 46.166890526031501)
('meantempi', -77.356805744654849)
('meanpressurei', -12.500573076110726)
('meanwspdi', 43.761820795363676)

2.5 The model's R squared was calculated as 0.5184
2.6 In general -the value of R squared should be closer to 1, to predict accuracy of our statistical model. This model is not appropriate because the dataset includes data on multiple stations in varying hours, which can create unnecessary confusion in our analysis and skew the potential correlation between the weather and hourly entries. A value of 0.5184 for our model implies that the data can be explained 51.84% using our model.

The plot of residuals is shown below. The histogram looks like a bell curve with approximate center zero. In terms of probability the model has long tails, on both sides, this is not accurate for frequently occurring residuals. Also the variation in the form of random error follows a Gaussian distribution with mean 0 and some variance. This bell curve does not confirm that a simple linear model prediction can predict with a high degree of accuracy. The model is sufficient only to a certain degree to explain and predict ridership at a given station and date/time.



Section 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.
3.1 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.

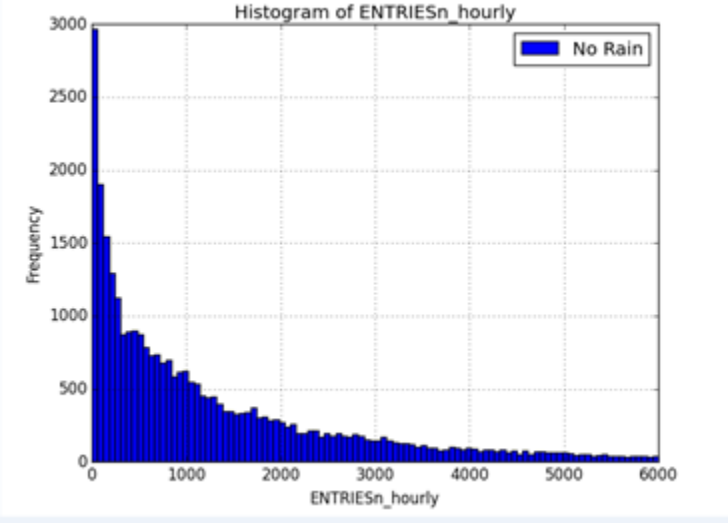
- Remember to increase the number of bins in the histogram (by having larger number of bars). The default bin width is not sufficient to capture the variability in the two samples.

3.2 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:

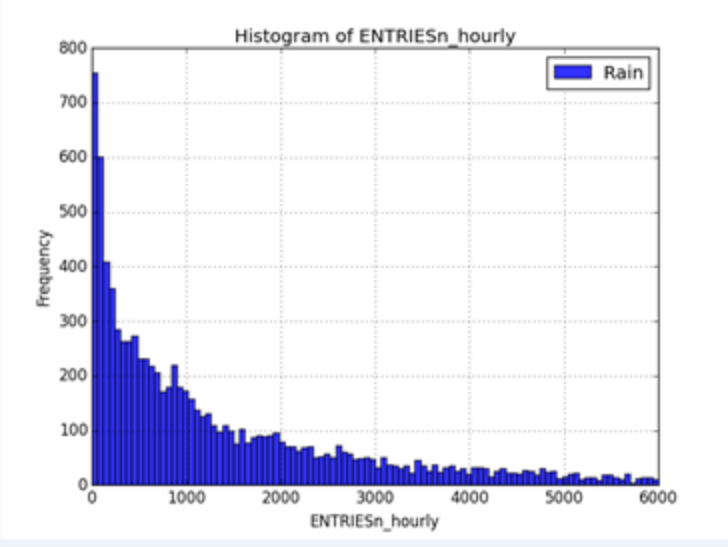
- Ridership by time-of-day
- Ridership by day-of-week

3.1Both histograms, one with rain and another without rain were completed separately. Further, a box plot after narrowing the y axis to (0.3000) entries clearly the ridership with rain is greater when compared to the ridership on non-rainy days.
3.2 I chose to plot the ridership by weekday or weekend. The box plot showed considerable increase in ridership on weekdays than weekends.
Another visualization I chose was to plot the day week ridership. The box plot showed a higher number of ridership in the following order Wed, Thursday, Fri, Tue, Mon, Sat and then lowest on Sunday.

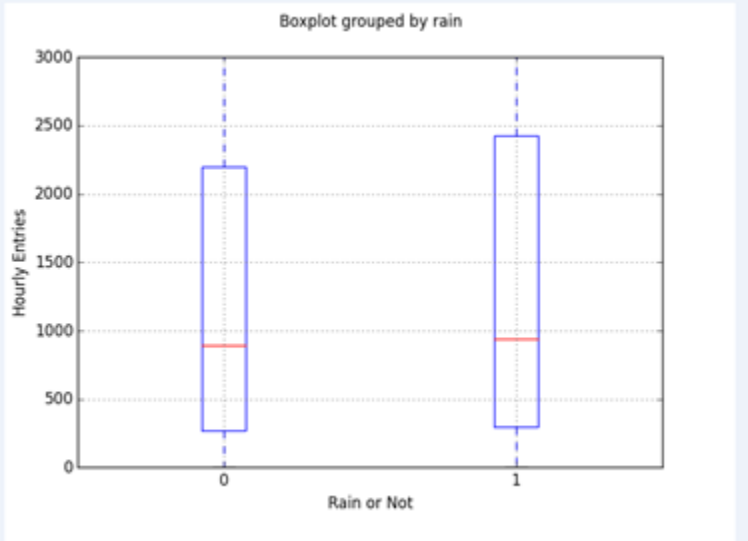
The below figure is for a Histogram with no rain.



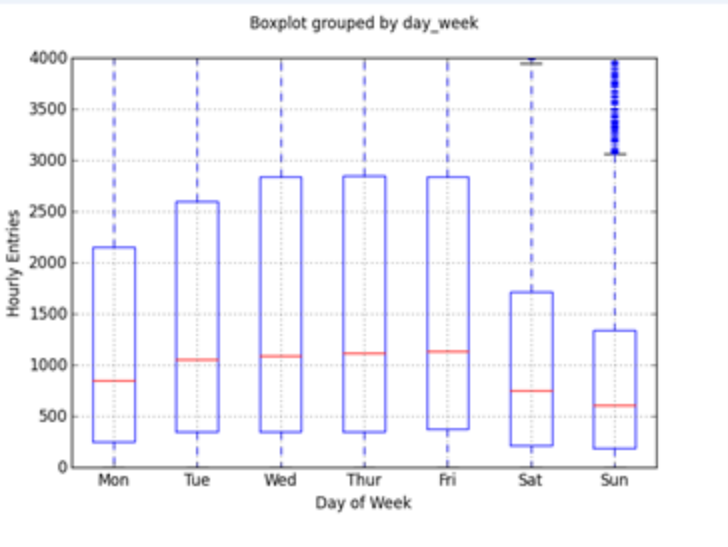
The below figure is for a Histogram with rain. When compare to the figure above, the frequency for rain entries is more, the y axis is enlarged for y entries.



The below figure is a box plot that shows hourly entries with rain (1) and without rain(0), even this figure shows the hourly entries is higher than the no rain entries.



The below figure shows a box plot that shows the hourly entries for the whole week. In this figure the hourly entries for weekdays is higher compare to the weekends. This could be due to riders going to work use the subway considerably.



Section 4.Conclusion

Please address the following questions in detail. Your answers should be 1-2 paragraphs long.
4.1 From your analysis and interpretation of the data, do more people ride the NYC subway when it is raining or when it is not raining?
4.2 What analyses lead you to this conclusion? You should use results from both your statistical tests and your linear regression to support your analysis.

4.1 From the regression results it is unclear if rain and ridership have a linear relationship, the model that I have created with a determination of 50% is insufficient to prove that ONLY weather has any correlation with ridership.
4.2 Analysis from the Mann Whitney U test the two tailed p value is less than alpha of 0.05, which indicates that the hypothesis that the probability of distributions of ENTRIESnhourly rainy and ENTRIESn_hourly non-rainy are the same is rejected.
I did not rely on the results from the linear regression using gradient descent. The coefficient of determination is at 50% which is not sufficient to prove any correlation with ridership and rain, there could be other factors that can affect ridership. Mann-Whitney U test on the distributions is more reliable, and we can conclude that during rainy hours there is a higher chance of entries.

Section 5.Reflection

Please address the following questions in detail. Your answers should be 1-2 paragraphs long.
5.1 Please discuss potential shortcomings of the methods of your analysis, including:

1. Dataset,
2. Analysis, such as the linear regression model or statistical test.

5.2 (Optional) Do you have any other insight about the dataset that you would like to share with us?

5.1.

1) Data Set – the dataset does not provide all the information that is needed for a thorough analysis. I realized that the hour column was incremented by 4 for consistency between different times and different stations, the timeslots that begin at midnight may not accurately associate with the weather data. Also there is not enough information as to where the weather data was collected, at the subway station or the nearest weather station.

2) Analysis, such as the linear regression model or statistical test - I did not rely on the results from the linear regression using gradient descent. The reason for that is the fact that there is no clear correlation between the features selected on the ridership.

So to conclude I used the Mann – Whitney U test for analysis decisions.

Programs in Python that were used for the exercises.

Section 1. Statistical Test

1.

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 22 05:48:15 2015

@author:
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy
import scipy.stats

def entries_histogram(turnstile_weather):
  norain_entries = turnstile_weather[turnstile_weather['rain'] == 0]
  rain_entries = turnstile_weather[turnstile_weather['rain'] == 1]

  norain_entries['ENTRIESn_hourly'].hist(bins=24, range=(0,6000))
  rain_entries['ENTRIESn_hourly'].hist(bins=20, range=(0,6000), alpha=0.8)
  plt.title('Histogram of ENTRIESn_hourly')
  plt.xlabel('ENTRIESn_hourly')
  plt.ylabel('Frequency')
  plt.legend(["No Rain", "Rain"])
  plt.show()
  return plt

if __name__== '__main__':
   turnstile_weather = pd.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
   result = entries_histogram(turnstile_weather)
```

2.

```
# -*- coding: utf-8 -*-
"""
Created on Sun May 24 04:43:02 2015

@author:
"""

import matplotlib.pyplot as plt
def entries_histogram(turnstile_weather):
df = turnstile_weather
plt.figure()
rainy = df['ENTRIESn_hourly'][df['rain'] == 1]
dry = df['ENTRIESn_hourly'][df['rain'] == 0]
dry.plot(kind='hist', alpha=0.5)
rainy.plot(kind='hist', alpha=0.5)
import scipy
```

```
import scipy.stats
def mann_whitney_plus_means(turnstile_weather):
rainy = turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain'] == 1]
dry = turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain'] == 0]
with_rain_mean = np.mean(rainy)
without_rain_mean = np.mean(dry)
U,p = scipy.stats.mannwhitneyu(rainy,dry)
x = turnstile_weather[turnstile_weather['rain'] == 0]['ENTRIESn_hourly']
y = turnstile_weather[turnstile_weather['rain'] == 1]['ENTRIESn_hourly']
m_u = len(x)*len(y)/2
sigma_u = np.sqrt(len(x)*len(y)*(len(x)+len(y)+1)/12)
z = (U - m_u)/sigma_u
p_val = 2*scipy.stats.norm.cdf(z)
return with_rain_mean, without_rain_mean, U, p_val # leave this line for the grader
if __name__== '__main__':
#turnstile_weather = pandas.read_csv('C:\NanoDegree\\turnstile_data_master_with_weather.csv')
turnstile_weather = pandas.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
w_r,wo_r,Uval,pval = mann_whitney_plus_means(turnstile_weather)

print w_r,wo_r,Uval,pval
```

Section 2. Linear Regression

```
# -*- coding: utf-8 -*-
"""
Created on Sun May 24 04:27:57 2015

@author:
"""
import matplotlib.pyplot as plt
import numpy as np
import pandas
from ggplot import *
def normalize_features(array):
array_normalized = (array-array.mean())/array.std()
mu = array.mean()
sigma = array.std()
return array_normalized, mu, sigma
def compute_cost(features, values, theta):
m = len(values)
sum_of_square_errors = np.square(np.dot(features, theta) - values).sum()
cost = sum_of_square_errors / (2*m)
return cost
def gradient_descent(features, values, theta, alpha, num_iterations):
m = len(values)
cost_history = []
for i in range(num_iterations):
predictions = np.dot(features,theta)
cost = compute_cost(features,values,theta)
cost_history.append(cost)
theta = theta + alpha / m * np.dot(values-predictions,features)
return theta, pandas.Series(cost_history)
def predictions(dataframe):
features = dataframe[['meanprecipi', 'meantempi', 'meanpressurei','meanwspdi']]
dummy_units1 = pandas.get_dummies(dataframe['UNIT'], prefix='unit')
dummy_units2 = pandas.get_dummies(dataframe['hour'], prefix='hour')
features = features.join(dummy_units1).join(dummy_units2)
```

```
values = dataframe[['ENTRIESn_hourly']]
m = len(values)
features, mu, sigma = normalize_features(features)
features['ones'] = np.ones(m) # Add a column of 1s (y intercept)
features_array = np.array(features)
values_array = np.array(values).flatten()
alpha = 0.01 # please feel free to change this value
num_iterations = 400 # please feel free to change this value
theta_gradient_descent = np.zeros(len(features.columns))
theta_gradient_descent, cost_history = gradient_descent(features_array,
values_array,
theta_gradient_descent,
alpha,
num_iterations)
plot = plot_cost_history(alpha, cost_history)
predictions = np.dot(features_array, theta_gradient_descent)
r_squared = compute_r_squared(values_array,predictions)
print "Coefficients for non-dummy variables: "
total_non_dummy_features = len(features)
weights = zip(features.columns, theta)
for pair in weights[:(total_non_dummy_features + 4)]:
print pair
return r_squared, theta_gradient_descent, predictions
def compute_r_squared(data, predictions):
mean = np.mean(data)
pred_diff = np.subtract(data, predictions) ** 2
mean_diff = np.sum([data, - mean]) ** 2
r_squared = 1 - pred_diff.sum() / mean_diff.sum()
return "{0}%".format(r_squared * 100)
def plot_cost_history(alpha, cost_history):
cost_df = pandas.DataFrame({
'Cost_History': cost_history,
'Iteration': range(len(cost_history))
})
return ggplot(cost_df, aes('Iteration', 'Cost_History')) + \
geom_point() + ggtitle('Cost History for alpha = %.3f' % alpha )
import scipy
def plot_residuals(turnstile_weather, predictions):
'''
makes a histogram of the residuals, the difference between the original
hourly entry data and the predicted values)
'''

plt.figure()
(turnstile_weather['ENTRIESn_hourly'] - predictions).hist(bins=400)
plt.suptitle('Residual histogram')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
return plt
if __name__== '__main__':
turnstile_weather = pandas.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
r_sq,theta,predic = predictions(turnstile_weather)
print r_sq
print "Plotting residuals:"
plot_residuals(turnstile_weather, predic)
```

Section 3. Visualization

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 22 09:15:20 2015

@author:
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy
import scipy.stats

def norainentries_histogram(turnstile_weather):
norain_entries = turnstile_weather[turnstile_weather['rain'] == 0]
#rain_entries = turnstile_weather[turnstile_weather['rain'] == 1]
norain_entries['ENTRIESn_hourly'].hist(bins=100, range=(0,6000))
#rain_entries['ENTRIESn_hourly'].hist(bins=20, range=(0,6000), alpha=0.8)
plt.title('Histogram of ENTRIESn_hourly')
plt.xlabel('ENTRIESn_hourly')
plt.ylabel('Frequency')
plt.legend(["No Rain"])
plt.show()
return plt
def rainentries_histogram(turnstile_weather):
#norain_entries = turnstile_weather[turnstile_weather['rain'] == 0]
rain_entries = turnstile_weather[turnstile_weather['rain'] == 1]
#norain_entries['ENTRIESn_hourly'].hist(bins=17, range=(0,6000))
rain_entries['ENTRIESn_hourly'].hist(bins=100, range=(0,6000), alpha=0.8)
plt.title('Histogram of ENTRIESn_hourly')
plt.xlabel('ENTRIESn_hourly')
plt.ylabel('Frequency')
plt.legend(["Rain"])
plt.show()
return plt

if __name__== '__main__':
turnstile_weather = pd.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
result1 = norainentries_histogram(turnstile_weather)

#result2 = rainentries_histogram(turnstile_weather)

# -*- coding: utf-8 -*-
"""
Created on Fri May 22 13:37:34 2015

@author:
"""
import pandas
import matplotlib.pyplot as plt

if __name__== '__main__':
turnstile_weather = pandas.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
p1 = turnstile_weather.boxplot('ENTRIESn_hourly','rain')
```

```
q11 = p1.set_xlabel('Rain or Not')
q12 = p1.set_ylabel('Hourly Entries')
q13 = p1.set_title(' ')
q14 = p1.set_ylim(0,3000)


# -*- coding: utf-8 -*-
"""
Created on Fri May 22 14:42:34 2015

@author:
"""

import pandas as pd

if __name__== '__main__':
turnstile_weather = pd.read_csv('C:\NanoDegree\improved-dataset\improved-dataset\\turnstile_weather_v2.csv')
dwr = pd.melt(turnstile_weather, id_vars=['day_week'], value_vars=['ENTRIESn_hourly'])
p1 = turnstile_weather.boxplot('ENTRIESn_hourly','day_week')
q11 = p1.set_xlabel('Day of Week')
q12 = p1.set_ylabel('Hourly Entries')
q13 = p1.set_title(' ')
q14 = p1.set_ylim(0,4000)
q15 = p1.set_xticklabels(['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun'])
```