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, and 4 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

https://en.wikipedia.org/wiki/Nonparametric_statistics (https://en.wikipedia.org/wiki/Nonparametric_statistics) https://en.wikipedia.org/wiki/Mann–Whitney_U_test (https://en.wikipedia.org/wiki/Mann–Whitney_U_test) http://dss.princeton.edu/online_help/analysis/interpreting_regression.htm (http://dss.princeton.edu/online_help/analysis/interpreting_regression.htm) https://en.wikipedia.org/wiki/Polynomial_regression (https://en.wikipedia.org/wiki/Polynomial_regression)

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?

ANS :

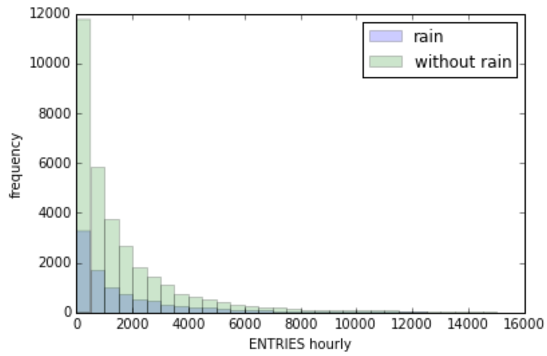
```
We choose Mann—Whitney U test two-tail P value.

H0 is  μ(rain) = μ(notrain)

We define α = 0.05
```

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.

In [1]:
```
import pandas as pd
import os
import scipy
import scipy.stats
import numpy as np
import statsmodels.api as sm
alpha = 0.05
filename = r'~/WorkSpace/Udemy-Datascience/IntroductionToDataScience/P2-Analyzing-NYC/turnstile_weather_v2.csv'
turnstile_weather = pd.read_csv(filename)
```

In [2]:
```
%pylab inline
import matplotlib.pyplot as plt

rain_df = turnstile_weather[turnstile_weather.rain == 1]['ENTRIESn_hourly']
norain_df = turnstile_weather[turnstile_weather.rain == 0]['ENTRIESn_hourly']
df_compare = pd.DataFrame({'rain': rain_df,
                'without rain': norain_df})
df_compare.plot(kind='hist', alpha=0.2,bins = range(0, 15000 + 500, 500) ) ##,bins = range(0, 37 + 3, 3)

plt.xlabel("ENTRIES hourly")
plt.ylabel("frequency")
plt.show()
```

```
Populating the interactive namespace from numpy and matplotlib
```



In [3]:
```
print "The number of rain data is %s ." % len(rain_df)
print "The number of without rain data is %s ." % len(norain_df)
```

```
The number of rain data is 9585 .
The number of without rain data is 33064 .
```

ANS : Because we do not know the data is droawn from any particular underlying probability distribution, two samples come from the same population against an alternative hypothesis, especially that a particular population tends to have larger values than the other .

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.

ANS:

In [4]:
```
def test_has_effect(df1,df2 , name_df1, name_df2 ):
    U,p = scipy.stats.mannwhitneyu(df1,df2)
    df1_mean = np.mean(df1)
    df2_mean = np.mean(df2)
    if p < alpha :
        print " We reject the null hypothesis (μ(%s) ≠ μ(%s))" % (name_df1,name_df2)
    else :
        print " We accept the null hypothesis (μ(%s) = μ(%s))" % (name_df1,name_df2)
    print " p-values is %.90f"%p
    print " %s mean is %.10f"% (name_df1,df1_mean )
    print " %s mean is %.10f"% (name_df2,df2_mean )

df1 = turnstile_weather[turnstile_weather.rain == 1]['ENTRIESn_hourly']
df2 = turnstile_weather[turnstile_weather.rain == 0]['ENTRIESn_hourly']
test_has_effect(df1,df2 , 'with rain', 'without rain' )
```

```
 We reject the null hypothesis (μ(with rain) ≠ μ(without rain))
 p-values is 0.000002741069571243749558478279559903256767938728444278240203857421875000000000000000000000
 with rain mean is 2028.1960354721
 without rain mean is 1845.5394386644
```

1.4 What is the significance and interpretation of these results?

ANS: The "ENTRIESn_hourly" mean of 2 samples statistically significance defferent so the rain has an effect on "ENTRIESn_hourly".

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:

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

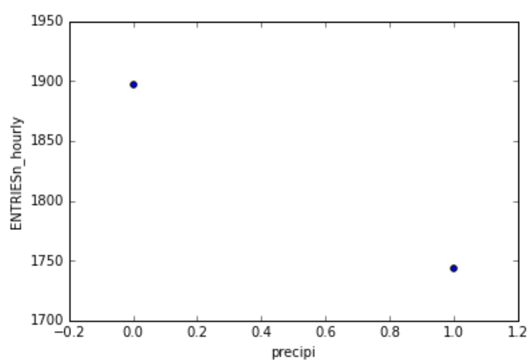
ANS: We decide to go for OLS.

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

In [5]:
```
df1 = turnstile_weather[turnstile_weather.fog == 1]['ENTRIESn_hourly']
df2 = turnstile_weather[turnstile_weather.fog == 0]['ENTRIESn_hourly']
test_has_effect(df1,df2 , 'with fog', 'without fog' )
```

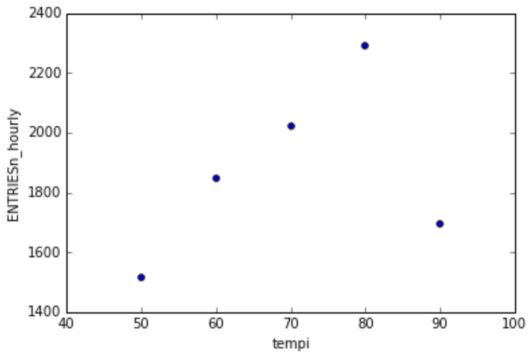
```
 We reject the null hypothesis (μ(with fog) ≠ μ(without fog))
 p-values is 0.00334419106247818738453236875329821486957371234893798828125000000000000000000000000000000000
 with fog mean is 1631.9809069212
 without fog mean is 1889.1161496566
```

In [6]:
```
resolution = 0.2
turnstile_weather['is_precipi'] =  turnstile_weather['precipi'].apply(lambda x: 0 if x == 0 else 1 )
agg = turnstile_weather.groupby(['is_precipi'], as_index=False).mean()


plt.scatter(x=agg['is_precipi'], y=agg['ENTRIESn_hourly'])

plt.xlabel("precipi")
plt.ylabel("ENTRIESn_hourly")
plt.show()

df1 = turnstile_weather[turnstile_weather.is_precipi == 0]['ENTRIESn_hourly']
df2 = turnstile_weather[turnstile_weather.is_precipi == 1]['ENTRIESn_hourly']
print len(df1)
print len(df2)
test_has_effect(df1,df2 , 'with precipi is 0', 'with precipi is not 0' )
```




```
39827
2822
 We reject the null hypothesis (μ(with precipi is 0) ≠ μ(with precipi is not 0))
 p-values is 0.00009560589055571569569620671291687585835461504757404327392578125000000000000000000000000000
 with precipi is 0 mean is 1896.7423104929
 with precipi is not 0 mean is 1743.3093550673
```

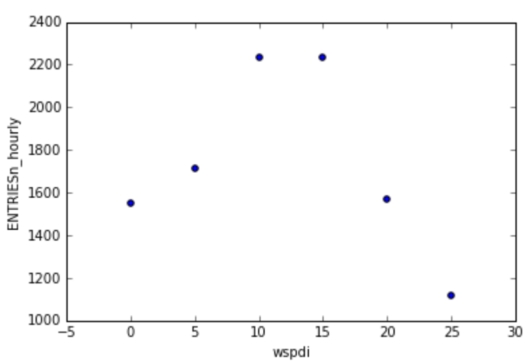
In [7]:
```
resolution = 10
turnstile_weather['tempi_group'] =  turnstile_weather['tempi'].apply(lambda x: np.round(x/resolution)*resolution )
agg = turnstile_weather.groupby(['tempi_group'], as_index=False).mean()


plt.scatter(x=agg['tempi_group'], y=agg['ENTRIESn_hourly'])

plt.xlabel("tempi")
plt.ylabel("ENTRIESn_hourly")
plt.show()
```



In [8]:
```
df1 = turnstile_weather[turnstile_weather.tempi_group == 70]['ENTRIESn_hourly']
df2 = turnstile_weather[turnstile_weather.tempi_group == 90]['ENTRIESn_hourly']

test_has_effect(df1,df2 , 'with low temp', 'with high temp' )
```

```
 We reject the null hypothesis (μ(with low temp) ≠ μ(with high temp))
 p-values is 0.011277793160554062598666114070056210039183497428894042968750000000000000000000000000000000
 with low temp mean is 2021.6490019960
 with high temp mean is 1695.4392523364
```

In [9]:
```
resolution = 5
turnstile_weather['wspdi_group'] =  turnstile_weather['wspdi'].apply(lambda x: np.round(x/resolution)*resolution )
agg = turnstile_weather.groupby(['wspdi_group'], as_index=False).mean()


plt.scatter(x=agg['wspdi_group'], y=agg['ENTRIESn_hourly'])

plt.xlabel("wspdi")
plt.ylabel("ENTRIESn_hourly")
plt.show()

df1 = turnstile_weather[turnstile_weather.wspdi_group == 0]['ENTRIESn_hourly']
df2 = turnstile_weather[turnstile_weather.wspdi_group == 5]['ENTRIESn_hourly']
print len(df1)
print len(df2)
test_has_effect(df1,df2 , 'with low wspdi', 'with high wspdi' )
```




```
5835
19682
 We reject the null hypothesis (μ(with low wspdi) ≠ μ(with high wspdi))
 p-values is 0.000000000000530516934301853578265511129698338247816409574131313320322078652679920196533203
 with low wspdi mean is 1549.0320479863
 with high wspdi mean is 1715.1301188904
```

ANS: We use rain,fog as features in our model and add UNIT, hour, tempi, wspdi and day_week to features using dummy varible.

In [10]:
```
features = turnstile_weather[['rain','fog']]
dummy_units = pd.get_dummies(turnstile_weather['UNIT'], prefix='unit')
features = features.join(dummy_units)
dummy_hour = pd.get_dummies(turnstile_weather['hour'], prefix='hour')
features = features.join(dummy_hour)
dummy_day_week = pd.get_dummies(turnstile_weather['day_week'], prefix='day_week')
features = features.join(dummy_day_week)
dummy_tempi = pd.get_dummies(turnstile_weather['tempi_group'], prefix='tempi_group')
features = features.join(dummy_tempi)
dummy_wspdi = pd.get_dummies(turnstile_weather['wspdi_group'], prefix='wspdi_group')
features = features.join(dummy_wspdi)
print len(features.columns  )
```

```
266
```

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 R2 value.”

ANS:

UNIT,Hour,day of week has obviously effect the ENTRIESn_hourly.

We decide to do not use station because it corresponding to the remote unit.

"rain,fog,tempi and precipi" has an effect by the t-test conclusion.

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

In [11]:
```
X = features
y = np.array(turnstile_weather['ENTRIESn_hourly'])
X = sm.add_constant(X)
olsmod = sm.OLS(y, X)
olsres = olsmod.fit()
ynewpred = olsres.predict(X)
intercept , params = olsres.params[0],olsres.params[1:]
print(olsres.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.546
Model:                            OLS   Adj. R-squared:                  0.544
Method:                 Least Squares   F-statistic:                     195.5
Date:                Sat, 31 Oct 2015   Prob (F-statistic):               0.00
Time:                        17:35:24   Log-Likelihood:            -3.8444e+05
No. Observations:               42649   AIC:                         7.694e+05
Df Residuals:                   42387   BIC:                         7.717e+05
Df Model:                         261
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------------
const             1.756e+14   2.74e+14      0.640      0.522     -3.62e+14  7.13e+14
rain               -54.5781     26.934     -2.026      0.043      -107.369    -1.787
fog               -215.4450    101.790     -2.117      0.034      -414.955   -15.935
unit_R003          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R004          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R005          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R006          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R007          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R008          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R009          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R011          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
unit_R012          1.11e+14   4.21e+14      0.264      0.792     -7.14e+14  9.36e+14
```

unit_R013	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R016	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R017	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R018	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R019	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R020	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R021	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R022	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R023	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R024	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R025	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R027	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R029	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R030	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R031	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R032	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R033	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R034	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R035	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R036	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R037	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R038	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R039	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R040	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R041	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R042	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R043	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R044	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R046	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R049	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R050	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R051	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R052	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R053	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R054	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R055	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R056	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R057	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R058	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R059	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R060	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R061	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R062	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R063	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R064	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R065	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R066	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R067	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R068	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R069	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R070	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R080	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R081	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R082	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R083	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R084	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R085	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R086	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R087	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R089	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R090	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R091	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R092	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R093	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R094	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R095	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R096	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R097	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R098	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R099	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R100	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R101	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R102	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R103	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R104	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R105	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R106	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R107	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R108	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R111	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14

unit_R112	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R114	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R115	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R116	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R117	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R119	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R120	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R121	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R122	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R123	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R124	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R126	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R127	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R137	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R139	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R163	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R172	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R179	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R181	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R183	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R184	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R186	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R188	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R189	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R194	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R196	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R198	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R199	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R200	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R202	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R203	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R204	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R205	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R207	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R208	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R209	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R210	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R211	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R212	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R213	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R214	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R215	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R216	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R217	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R218	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R219	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R220	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R221	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R223	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R224	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R225	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R226	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R227	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R228	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R229	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R230	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R231	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R232	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R233	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R234	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R235	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R236	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R237	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R238	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R239	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R240	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R242	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R243	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R244	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R246	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R247	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R248	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R249	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R250	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R251	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R252	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R253	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R254	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R255	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R256	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14

unit_R257	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R258	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R259	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R260	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R261	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R262	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R263	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R264	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R265	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R266	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R269	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R270	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R271	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R273	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R274	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R275	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R276	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R277	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R278	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R279	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R280	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R281	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R282	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R284	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R285	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R287	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R291	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R294	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R295	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R300	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R303	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R304	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R307	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R308	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R309	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R310	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R311	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R312	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R313	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R318	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R319	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R321	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R322	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R323	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R325	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R330	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R335	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R336	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R337	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R338	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R341	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R344	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R345	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R346	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R348	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R354	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R356	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R358	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R370	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R371	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R372	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R373	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R382	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R424	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R429	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R453	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R454	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R455	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R456	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R459	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
unit_R464	1.11e+14	4.21e+14	0.264	0.792	-7.14e+14	9.36e+14
hour_0	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
hour_4	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
hour_8	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
hour_12	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
hour_16	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
hour_20	3.394e+14	9.75e+14	0.348	0.728	-1.57e+15	2.25e+15
day_week_0	-4.988e+14	1.75e+15	-0.285	0.776	-3.93e+15	2.93e+15
day_week_1	-4.988e+14	1.75e+15	-0.285	0.776	-3.93e+15	2.93e+15
day_week_2	-4.988e+14	1.75e+15	-0.285	0.776	-3.93e+15	2.93e+15

```
day_week_3       -4.988e+14   1.75e+15     -0.285      0.776     -3.93e+15  2.93e+15
day_week_4       -4.988e+14   1.75e+15     -0.285      0.776     -3.93e+15  2.93e+15
day_week_5       -4.988e+14   1.75e+15     -0.285      0.776     -3.93e+15  2.93e+15
day_week_6       -4.988e+14   1.75e+15     -0.285      0.776     -3.93e+15  2.93e+15
tempi_group_50.0  7.354e+14   2.36e+15      0.312      0.755     -3.88e+15  5.35e+15
tempi_group_60.0  7.354e+14   2.36e+15      0.312      0.755     -3.88e+15  5.35e+15
tempi_group_70.0  7.354e+14   2.36e+15      0.312      0.755     -3.88e+15  5.35e+15
tempi_group_80.0  7.354e+14   2.36e+15      0.312      0.755     -3.88e+15  5.35e+15
tempi_group_90.0  7.354e+14   2.36e+15      0.312      0.755     -3.88e+15  5.35e+15
wspdi_group_0.0  -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
wspdi_group_5.0  -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
wspdi_group_10.0 -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
wspdi_group_15.0 -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
wspdi_group_20.0 -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
wspdi_group_25.0 -8.626e+14   1.89e+15     -0.456      0.648     -4.57e+15  2.84e+15
==============================================================================
Omnibus:                    30229.221   Durbin-Watson:                   1.530
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1106543.906
Skew:                           2.975   Prob(JB):                         0.00
Kurtosis:                      27.234   Cond. No.                     9.87e+14
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.07e-26. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

ANS:

In [12]:

```
print "rain coefficients is %.10f"%params[0]
print "fog coefficients is %.10f"%params[1]
```

```
rain coefficients is -54.5780521669
fog coefficients is -215.4449661619
```

2.5 What is your model’s R2 (coefficients of determination) value?

ANS:

In [13]:

```
print olsres.rsquared
```

```
0.54629540704
```

2.6 What does this R2 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 R2 value?

ANS :

```
R2 value higher mean that the model is better (the prediction is closer ).

As you can see this R2(0.546) is greater than Problem Set 3-5 minimum requrement (0.4) so this linear model is appropriate.
```

Section 3. Visualization

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

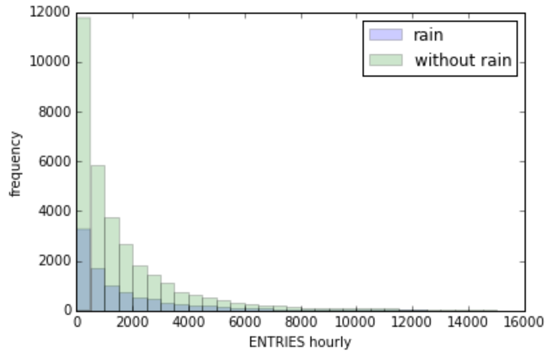
You can combine the two histograms in a single plot or you can use two separate plots.

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.

In [14]:
```
rain_df = turnstile_weather[turnstile_weather.rain == 1]['ENTRIESn_hourly']
norain_df = turnstile_weather[turnstile_weather.rain == 0]['ENTRIESn_hourly']
df_compare = pd.DataFrame({'rain': rain_df,
                'without rain': norain_df})
df_compare.plot(kind='hist', alpha=0.2,bins = range(0, 15000 + 500, 500) ) ##,bins = range(0, 37 + 3, 3)

plt.xlabel("ENTRIES hourly")
plt.ylabel("frequency")
plt.show()
```

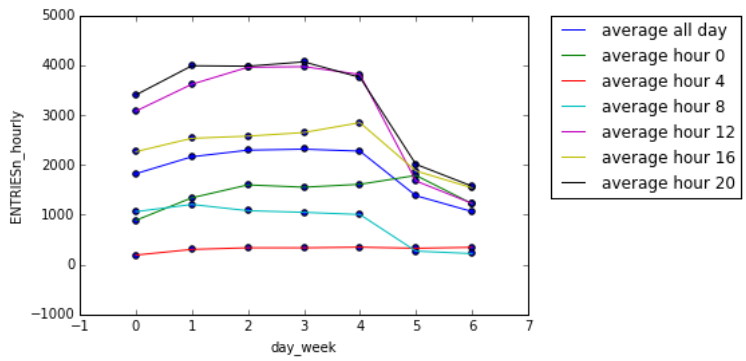


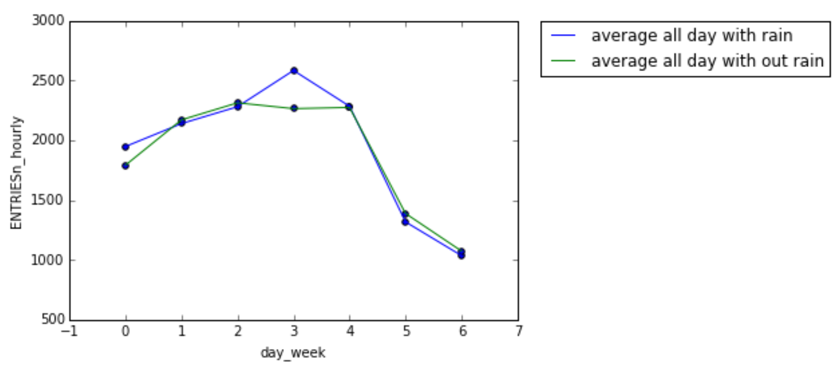
3.2 One visualization can be more freeform. You should feel free to implement something that we discussed in class (e.g., scatter plots, line plots) or attempt to implement something more advanced if you'd like. Some suggestions are:

Ridership by time-of-day

Ridership by day-of-week

In [15]:
```
agg = turnstile_weather.groupby(['day_week'], as_index=False).mean()

x_data  = agg['day_week']
y_data = agg['ENTRIESn_hourly']
plt.plot(x_data, y_data,label="average all day")
plt.scatter(x=agg['day_week'], y=agg['ENTRIESn_hourly'])

for hour in range(0,25):
    agg = turnstile_weather[turnstile_weather.hour == hour].groupby(['day_week'], as_index=False).mean()
    x_data  = agg['day_week']
    y_data = agg['ENTRIESn_hourly']
    if np.mean(y_data) > 0  :
        plt.plot(x_data, y_data, label="average hour %s"%hour)
        plt.scatter(x=agg['day_week'], y=agg['ENTRIESn_hourly'])

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel("day_week")
plt.ylabel("ENTRIESn_hourly")
plt.show()

agg = turnstile_weather[turnstile_weather.rain == 1].groupby(['day_week'], as_index=False).mean()
x_data  = agg['day_week']
y_data = agg['ENTRIESn_hourly']
plt.plot(x_data, y_data,label="average all day with rain")
plt.scatter(x=agg['day_week'], y=agg['ENTRIESn_hourly'])

agg = turnstile_weather[turnstile_weather.rain == 0].groupby(['day_week'], as_index=False).mean()
x_data  = agg['day_week']
y_data = agg['ENTRIESn_hourly']
plt.plot(x_data, y_data,label="average all day with out rain")
plt.scatter(x=agg['day_week'], y=agg['ENTRIESn_hourly'])
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

plt.xlabel("day_week")
plt.ylabel("ENTRIESn_hourly")

plt.show()
```





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?

From the null hypothesis testing rain has significant effect on Houly_entries.

From rain's coeffiecent in linear regression we can concluse that "The people do less ride when it is 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.

From the null hypothesis testing results with 2 samples rain and with out rain return p value = it mean that the possibility of different has occured by chance is about 0.0002741% so we assume that the rain has significantly effect on Houly_entries.

From the rain coeffiecent of linear regression is about -71.02 so we assume that the rain has effect on Houly_entries in negative way.

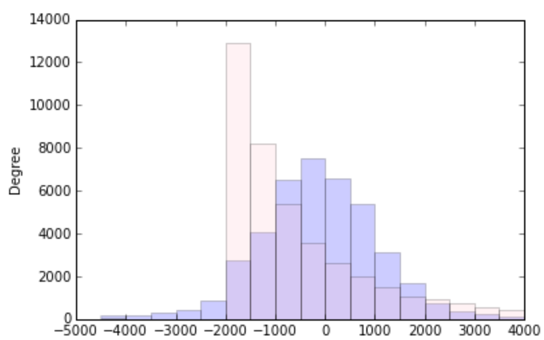
Section 5. Reflection

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:

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

In [20]:
```
def plot_residuals(turnstile_weather, predictions):
    plt.figure()
    df = (turnstile_weather['ENTRIESn_hourly'] - predictions)
    # df.hist()

    df.plot(kind='hist',alpha=0.2,  y='ENTRIESn_hourly',bins = range(-4500, 4500, 500))

    df = (turnstile_weather['ENTRIESn_hourly'] - turnstile_weather['ENTRIESn_hourly'].mean())
    df.plot(kind='hist',alpha=0.2,  y='ENTRIESn_hourly',bins = range(-4500, 4500, 500) , color="pink")
    return plt

predictions = ynewpred
plot_residuals(turnstile_weather, predictions)
plt.show()
print "r^2 is %.6f" % olsres.rsquared
```




```
r^2 is 0.546295
```

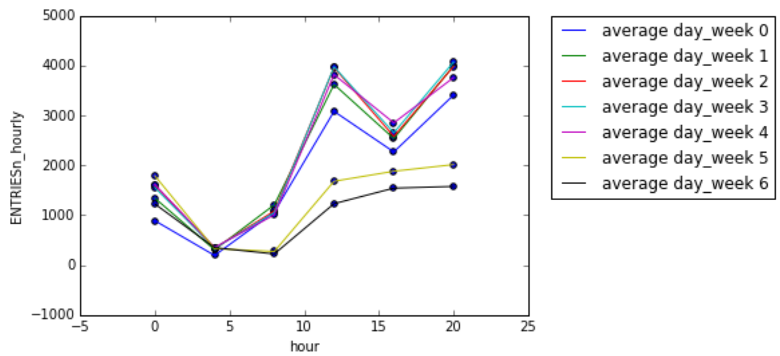
By the dataset if we have more parameters that effect "Hourly_entries" we possibly predict the result more accurate such as :

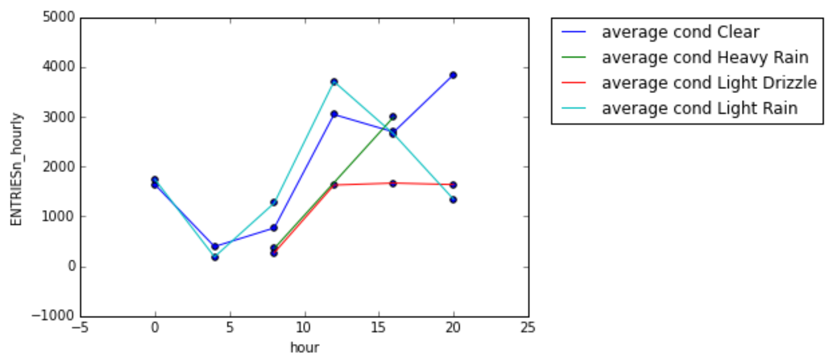
"Does it has nearby special events on the location at that time?" . "Is it holiday?" .

It is possible that the other models such as "Polynomial regression" has a better result.

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

In [28]:
```
for day_week in range(0,7):
    agg = turnstile_weather[turnstile_weather.day_week == day_week].groupby(['hour'], as_index=False).mean()
    x_data  = agg['hour']
    y_data = agg['ENTRIESn_hourly']
    if np.mean(y_data) > 0  :
        plt.plot(x_data, y_data, label="average day_week %s"%day_week)
        plt.scatter(x=agg['hour'], y=agg['ENTRIESn_hourly'])
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel("hour")
plt.ylabel("ENTRIESn_hourly")
plt.show()


agg2 = turnstile_weather.groupby(['conds'])
for key, cond  in agg2['conds']:
#     print key
    if key in ['Clear','Light Drizzle','Heavy Rain','Light Rain']:
        agg = turnstile_weather[turnstile_weather.conds == key].groupby(['hour'], as_index=False).mean()
        x_data  = agg['hour']
        y_data = agg['ENTRIESn_hourly']
#         print np.mean(y_data)
        if np.mean(y_data) > 0  :
            plt.plot(x_data, y_data, label="average cond %s"%key)
            plt.scatter(x=agg['hour'], y=agg['ENTRIESn_hourly'])

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel("hour")
plt.ylabel("ENTRIESn_hourly")
plt.show()
```

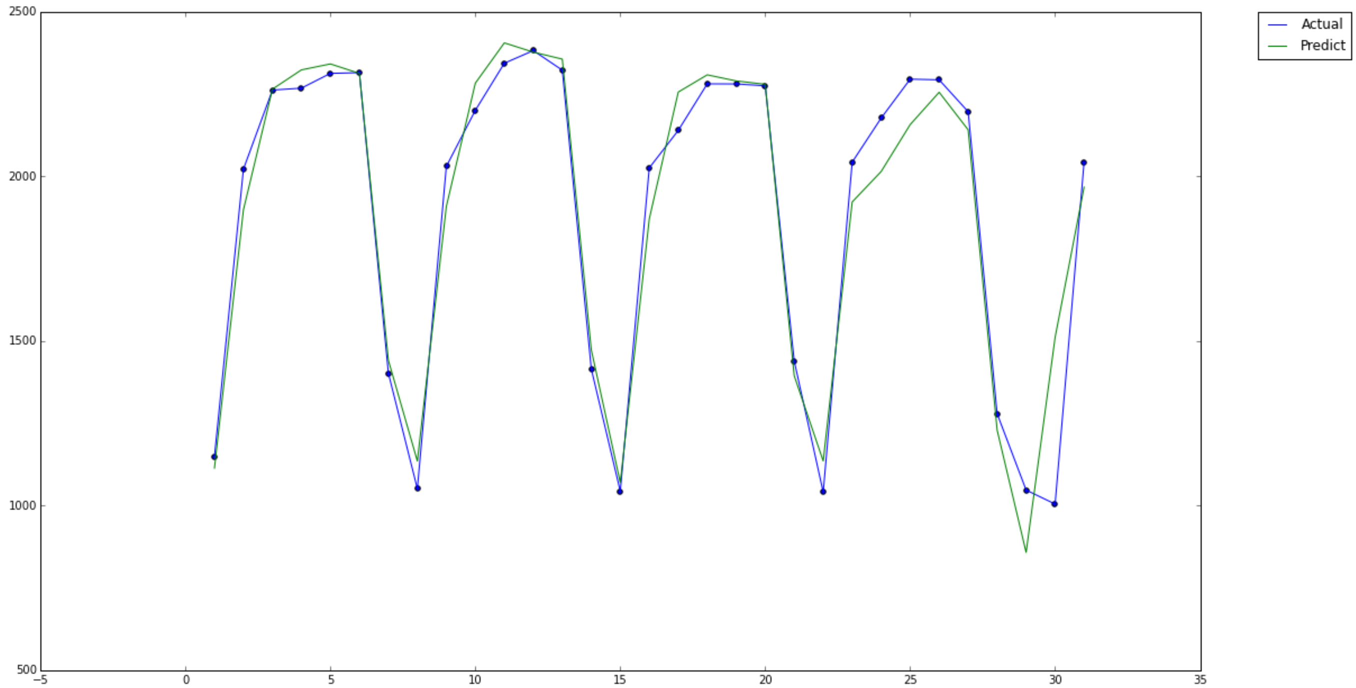




There are 2 patterns of Hourly_entries by hour first is day_week 0-4 second is dayweek 5-6. Peek Hour for Hourly_entries are about 12:00 and 20:00.

In [91]:

```
fig = matplotlib.pyplot.gcf()
fig.set_size_inches(18.5, 10.5)

turnstile_weather['predict_ENTRIESn_hourly'] = predictions
agg = turnstile_weather.groupby(['DATEn'], as_index=False).mean()
# agg['datetime_obj']  = pd.to_datetime(agg['DATEn'])
# agg['datetime_obj']  = pd.DatetimeIndex (agg['DATEn'])
# agg['datetime_obj']  = agg['DATEn'].astype('datetime64[ns]')
from time import strptime
# agg = agg[:30]
agg['datetime_obj']  = pd.to_datetime(agg['DATEn'])
# print agg['datetime_obj'][0].day
agg['datetime_num']  =  agg['datetime_obj'].apply(lambda x: x.day )

agg = agg.sort(['datetime_num'])
# print len(agg)

import matplotlib.dates as mdates
x_data = agg['datetime_num']
y_data = agg['ENTRIESn_hourly']
y2_data = agg['predict_ENTRIESn_hourly']

plt.scatter(x=agg['datetime_num'], y=agg['ENTRIESn_hourly'])
plt.plot(agg['datetime_num'], agg['ENTRIESn_hourly'],label="Actual")

plt.plot(agg['datetime_num'], agg['predict_ENTRIESn_hourly'],label="Predict")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



In [ ]: