Luis Santulli Uber Data Analysis 2

In [2]:
```
import pandas as luis # data processing
import numpy as np # linear algebra
import matplotlib.pyplot as plt # plotting
import seaborn as sns # Python stat graphics built on matplotlib
import datetime
%matplotlib inline
```

In [3]:
```
data = luis.read_csv(r"C:\Users\Luis Santulli\Desktop\uber-raw-data-aug14.csv")
data.head()
```

Out[3]:

	Date/Time	Lat	Lon	Base
0	8/1/2014 0:03:00	40.7366	-73.9906	B02512
1	8/1/2014 0:09:00	40.7260	-73.9918	B02512
2	8/1/2014 0:12:00	40.7209	-74.0507	B02512
3	8/1/2014 0:12:00	40.7387	-73.9856	B02512
4	8/1/2014 0:12:00	40.7323	-74.0077	B02512

In [4]:
```
data.info()
```

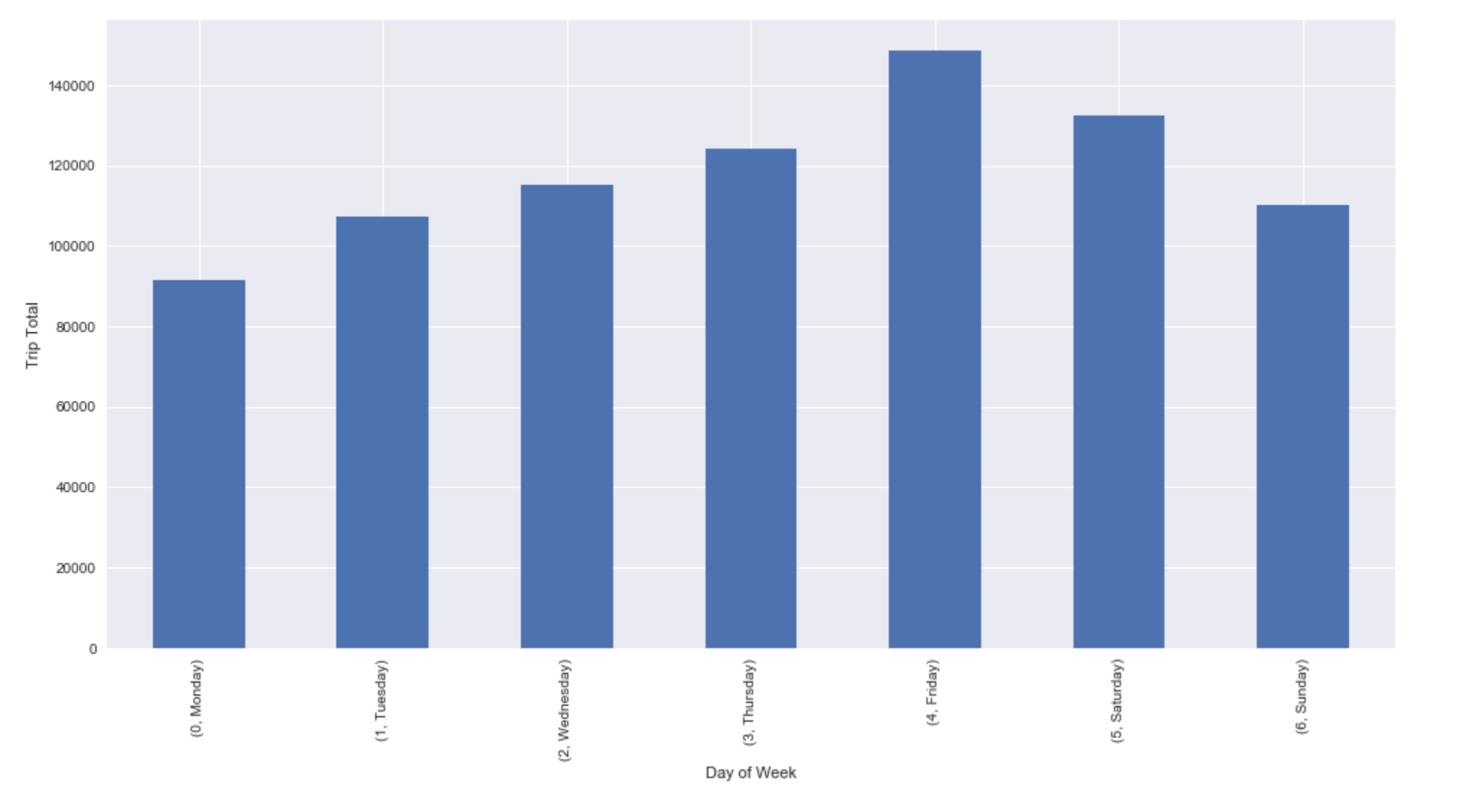
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 829275 entries, 0 to 829274
Data columns (total 4 columns):
Date/Time    829275 non-null object
Lat          829275 non-null float64
Lon          829275 non-null float64
Base         829275 non-null object
dtypes: float64(2), object(2)
memory usage: 25.3+ MB
```

Converting to Date/Time format for Python to understand

In [5]:
```
data['Date/Time'] = pd.to_datetime(data['Date/Time'], format="%m/%d/%Y %H:%M:%S") #conventional way of doing this
data['DayOfWeekNum'] = data['Date/Time'].dt.dayofweek #Creating DayofWeekNum by finding day of week from Date/Time column
data['DayOfWeek'] = data['Date/Time'].dt.weekday_name #Creating DayofWeek by finding day name from Date/Time column
data['MonthDayNum'] = data['Date/Time'].dt.day #Creating MonthDayNum by finding month day from Date/Time column
data['HourOfDay'] = data['Date/Time'].dt.hour #Creating HourofDay by finding hour from Date/Time column
```

In [6]:
```
#creating pivot table
augustDays = data.pivot_table(index=['DayOfWeekNum','DayOfWeek'])
#plotting
augustDays.plot(kind='bar', figsize=(16, 8))
plt.ylabel("Trip Total")
plt.xlabel("Day of Week")
```

Out[6]: `<matplotlib.text.Text at 0x1c55c950ef0>`



In [8]:
```
dfJanFeb =pd.read_csv(r"C:\Users\Luis Santulli\Desktop\Uber-Jan-Feb-FOIL.csv") # reads and creates DataFrame
```

In [9]:
```
# print "shape" of data frame

dfJanFeb.shape
```

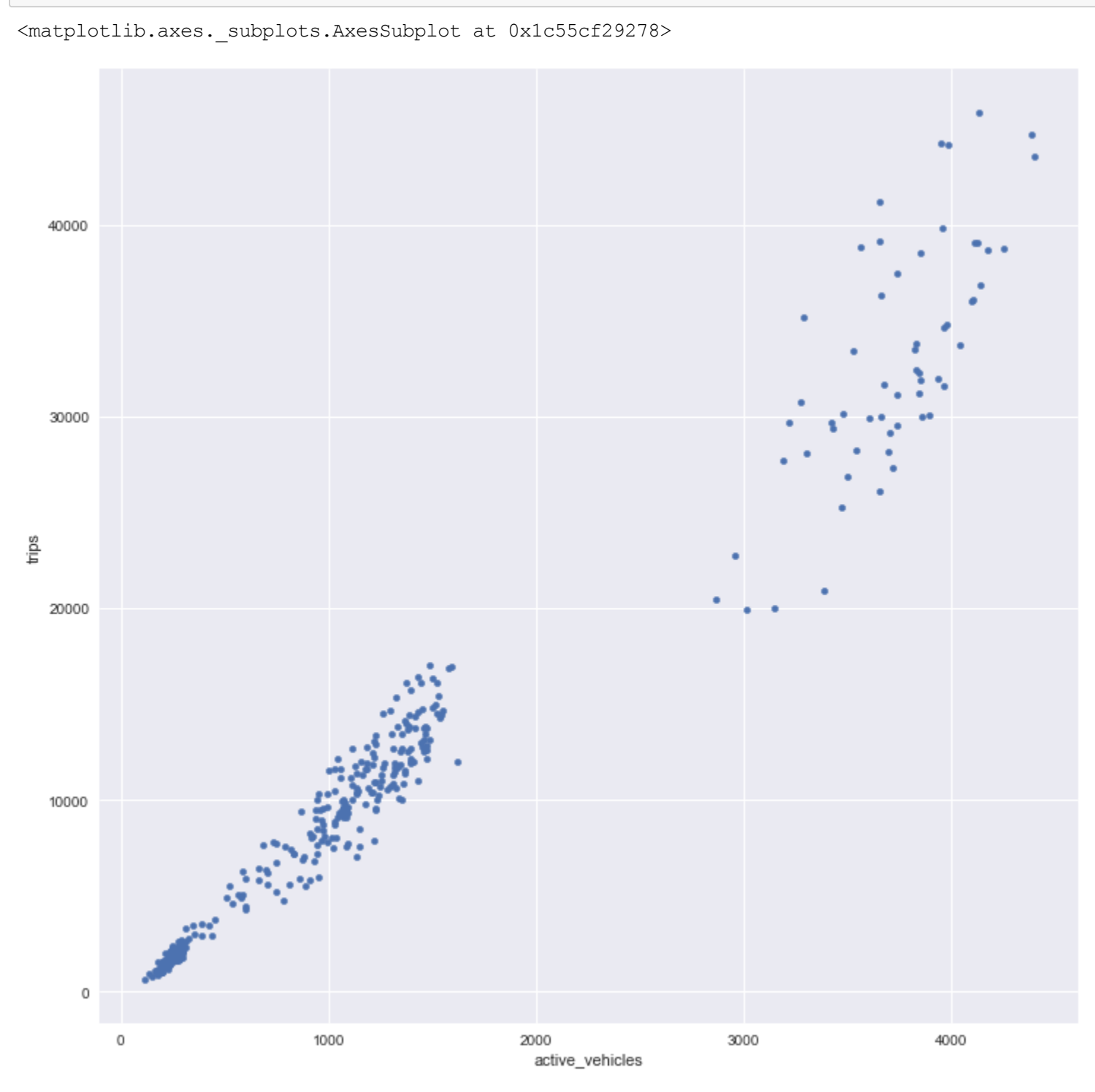
Out[9]: `(354, 4)`

354 observations and four columns.

Linear Regression

In [10]:
```
# check for a linear relationship between active_vehicles and trips
dfJanFeb.plot(kind='scatter', x='active_vehicles', y='trips', figsize=(12,12))
```

Out[10]: `<matplotlib.axes._subplots.AxesSubplot at 0x1c55cf29278>`



In [11]:
```
# linregress module calculates linear regression line
from scipy.stats import linregress
rv = dfJanFeb.as_matrix(columns=['active_vehicles', 'trips'])
a, b, r, p, stderr = linregress(rv)
print(a, b, r, p, stderr)
```

```
8.98104812156 -74.8205203389 0.980492539725 8.79590426331e-251 0.0959617963385
```

In [12]:
```
print(dfJanFeb)
```

```
    dispatching_base_number       date  active_vehicles  trips
0                    B02512   1/1/2015              190   1132
1                    B02765   1/1/2015              225   1765
2                    B02764   1/1/2015             3427  29421
3                    B02682   1/1/2015              945   7679
4                    B02617   1/1/2015             1228   9537
5                    B02598   1/1/2015              870   6903
6                    B02598   1/2/2015              785   4768
7                    B02617   1/2/2015             1137   7065
8                    B02512   1/2/2015              175    875
9                    B02682   1/2/2015              890   5506
10                   B02765   1/2/2015              196   1001
11                   B02764   1/2/2015             3147  19974
12                   B02765   1/3/2015              201   1526
13                   B02617   1/3/2015             1188  10664
14                   B02598   1/3/2015              818   7432
15                   B02682   1/3/2015              915   8010
16                   B02512   1/3/2015              173   1088
17                   B02764   1/3/2015             3215  29729
18                   B02512   1/4/2015              147    791
19                   B02682   1/4/2015              812   5621
20                   B02598   1/4/2015              746   5223
21                   B02765   1/4/2015              183    993
22                   B02617   1/4/2015             1088   7729
23                   B02764   1/4/2015             2862  20441
24                   B02512   1/5/2015              194    984
25                   B02682   1/5/2015              951   6012
26                   B02617   1/5/2015             1218   7899
27                   B02764   1/5/2015             3387  20926
28                   B02598   1/5/2015              907   5798
29                   B02765   1/5/2015              227   1133
..                      ...        ...              ...    ...
324                  B02764  2/24/2015             3965  34686
325                  B02512  2/24/2015              247   1869
326                  B02598  2/24/2015             1061   9954
327                  B02682  2/24/2015             1346  12497
328                  B02617  2/24/2015             1456  13719
329                  B02765  2/24/2015              698   6390
330                  B02512  2/25/2015              246   1647
331                  B02598  2/25/2015             1076   9405
332                  B02765  2/25/2015              706   6178
333                  B02682  2/25/2015             1395  12693
334                  B02617  2/25/2015             1473  12811
335                  B02764  2/25/2015             3934  31957
336                  B02598  2/26/2015             1134  10661
337                  B02617  2/26/2015             1539  14461
338                  B02682  2/26/2015             1465  13814
339                  B02512  2/26/2015              243   1797
340                  B02765  2/26/2015              745   6744
341                  B02764  2/26/2015             4101  36091
342                  B02765  2/27/2015              786   7563
343                  B02617  2/27/2015             1551  14677
344                  B02598  2/27/2015             1114  10755
345                  B02512  2/27/2015              272   2056
346                  B02764  2/27/2015             4253  38780
347                  B02682  2/27/2015             1510  14975
348                  B02598  2/28/2015              994  10319
349                  B02764  2/28/2015             3952  39812
350                  B02617  2/28/2015             1372  14022
351                  B02682  2/28/2015             1386  14472
352                  B02512  2/28/2015              230   1803
353                  B02765  2/28/2015              747   7753

[354 rows x 4 columns]
```

In [15]:
```
# Total Active Cars for this time period in NYC
dfJanFeb['active_vehicles'].sum()
```

Out[15]: `462832`

In [16]:
```
# Max Active Cars for this time period in NYC
dfJanFeb['active_vehicles'].max()
```

Out[16]: `4395`

In [17]:
```
# Mimimum
dfJanFeb['active_vehicles'].min()
```

Out[17]: `112`

In [19]:
```
# count number of dates
dfJanFeb['date'].count()
```

Out[19]: `354`

In [20]:
```
# creating a pivot table with groupby
dfJanFeb.groupby('date')['active_vehicles'].sum()
```

Out[20]:
```
date
1/1/2015     6885
1/10/2015    7346
1/11/2015    6571
1/12/2015    7364
1/13/2015    7559
1/14/2015    7849
1/15/2015    8080
1/16/2015    8273
1/17/2015    7527
1/18/2015    6863
1/19/2015    5945
1/2/2015     6330
1/20/2015    7592
1/21/2015    7948
1/22/2015    8267
1/23/2015    8490
1/24/2015    7643
1/25/2015    6787
1/26/2015    6533
1/27/2015    3496
1/28/2015    7815
1/29/2015    8376
1/3/2015     6510
1/30/2015    8693
1/31/2015    8223
1/4/2015     5838
1/5/2015     6884
1/6/2015     7216
1/7/2015     7444
1/8/2015     7999
1/9/2015     7989
2/1/2015     7752
2/10/2015    8029
2/11/2015    8515
2/12/2015    9123
2/13/2015    9604
2/14/2015    8973
2/15/2015    7939
2/16/2015    7551
2/17/2015    8403
2/18/2015    8442
2/19/2015    9030
2/2/2015     7080
2/20/2015    9649
2/21/2015    8765
2/22/2015    7620
2/23/2015    8197
2/24/2015    8773
2/25/2015    8830
2/26/2015    9227
2/27/2015    9486
2/28/2015    8681
2/3/2015     7840
2/4/2015     8185
2/5/2015     8833
2/6/2015     8937
2/7/2015     8119
2/8/2015     7226
2/9/2015     7688
Name: active_vehicles, dtype: int64
```

Pandas Functions for Statistics

In [21]:
```
dfJanFeb['active_vehicles'].mean()
```

Out[21]: `1307.4350282485875`

In [22]:
```
dfJanFeb['active_vehicles'].mode()
```

Out[22]:
```
0    238
dtype: int64
```

In [23]:
```
dfJanFeb['active_vehicles'].std()
```

Out[23]: `1162.5106256246545`

In [24]:
```
uberApril2014 = pd.read_csv(r"C:\Users\Luis Santulli\Desktop\uber-raw-data-apr14.csv", index_col = False) # Importing csv apr
il
uberApril2014.columns=['Date', 'Latitude', 'Longitude'] # Assigning columns
```

In [25]:
```
 uberJanJune15df = pd.read_csv(r"C:\Users\Luis Santulli\Desktop\uber-raw-data-janjune-15.csv") # Importing csv from jan-june
```

In [26]:
```
uberJanJune15df.keys()
```

Out[26]:
```
Index(['Dispatching_base_num', 'Pickup_date', 'Affiliated_base_num',
       'locationID'],
      dtype='object')
```

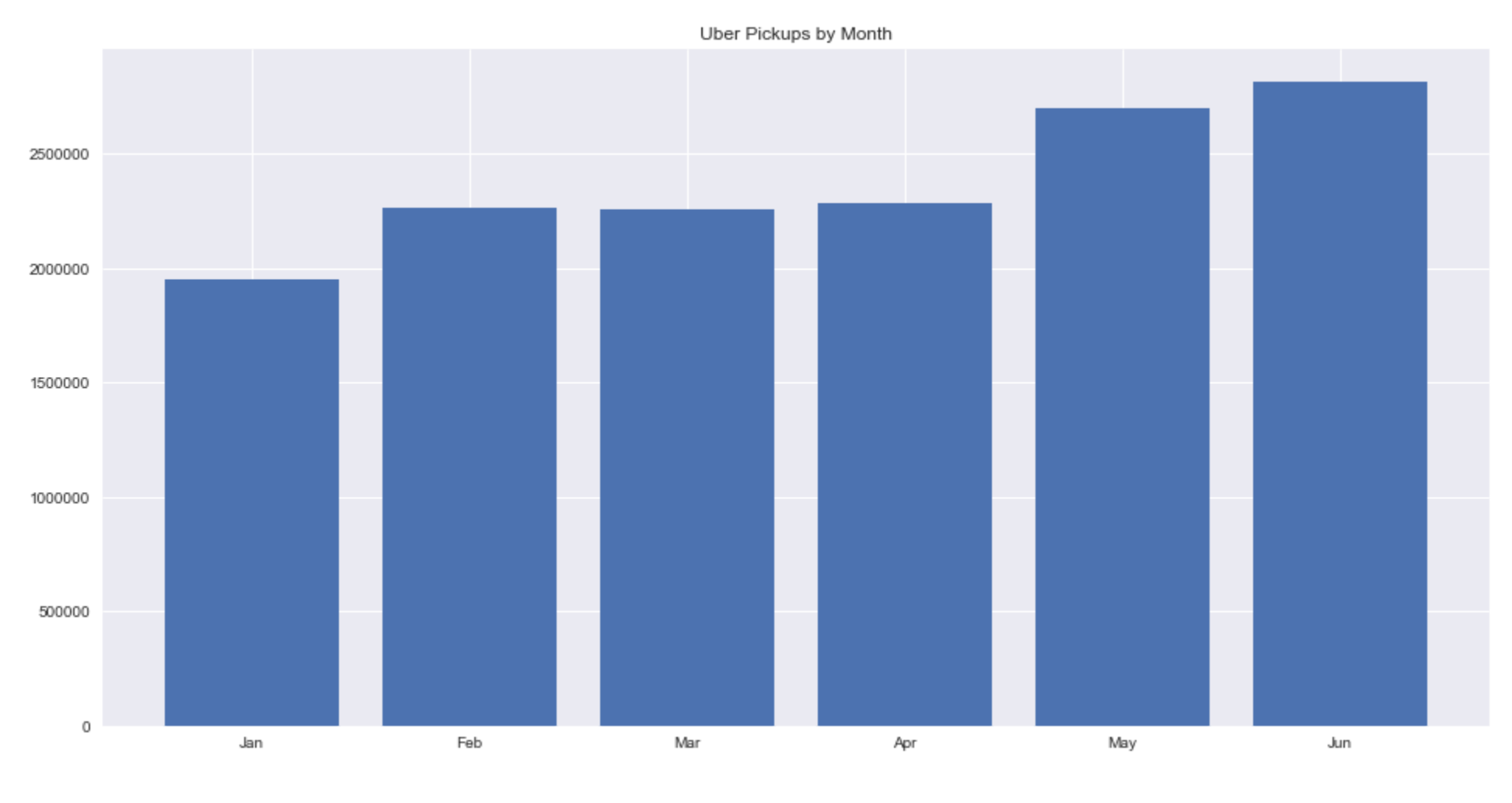
In [27]:
```
uberJanJune15df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14270479 entries, 0 to 14270478
Data columns (total 4 columns):
Dispatching_base_num    object
Pickup_date             object
Affiliated_base_num     object
locationID              int64
dtypes: int64(1), object(3)
memory usage: 435.5+ MB
```

In [29]:
```
uberJanJune15df['Hour'] = uberJanJune15df['Pickup_date'].apply(lambda x: x[11:13])
uberJanJune15df['Date'] = uberJanJune15df['Pickup_date'].apply(lambda x: x[0:10])
uberJanJune15df['Month'] = uberJanJune15df['Date'].apply(lambda x: x[5:7])
```

In [30]:
```
Month = ['Jan','Feb','Mar','Apr','May','Jun']
Index = [0,1,2,3,4,5]
Monthly_pickup = uberJanJune15df.groupby(['Month']).size()
plt.figure(1,figsize=(16,8))
plt.bar(Index,Monthly_pickup)
plt.xticks(Index,Month)
plt.title('Uber Pickups by Month')
```

Out[30]: `<matplotlib.text.Text at 0x1c5f5565e10>`



Uber Raw Data with extra information

In [31]:
```
# importing an enhanced version of the Jan-June Uber dataset
enhanced = pd.read_csv(r"C:\Users\Luis Santulli\Desktop\uber_nyc_enriched.csv")
```

In [32]:
```
# dataset from January through June of 2015 now includes extra keys including weather and location data
enhanced.keys()
```

Out[32]:
```
Index(['pickup_dt', 'borough', 'pickups', 'spd', 'vsb', 'temp', 'dewp', 'slp',
       'pcp01', 'pcp06', 'pcp24', 'sd', 'hday'],
      dtype='object')
```

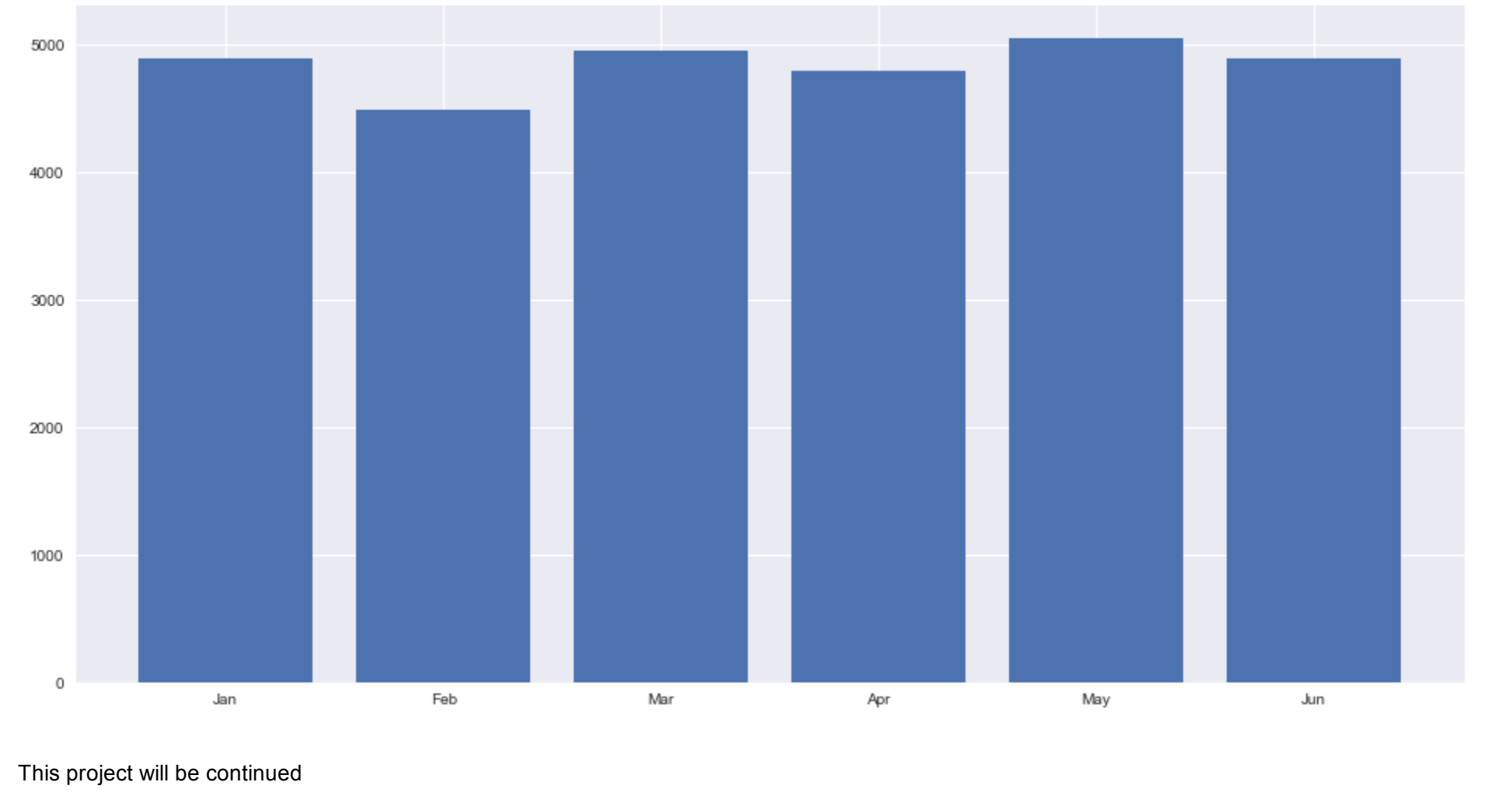
In [33]:
```
enhanced.shape
```

Out[33]: `(29101, 13)`

In [37]:
```
# plotting pick ups by month
enhanced['Hour'] = enhanced['pickup_dt'].apply(lambda x: x[11:13])
enhanced['Date'] = enhanced['pickup_dt'].apply(lambda x: x[0:10])
enhanced['Month'] = enhanced['pickup_dt'].apply(lambda x: x[5:7])
```

In [38]:
```
Month = ['Jan','Feb','Mar','Apr','May','Jun']
Index = [0,1,2,3,4,5]
Monthly_pickup = enhanced.groupby(['Month']).size()
plt.figure(1,figsize=(16,8))
plt.bar(Index,Monthly_pickup)
plt.xticks(Index,Month)
plt.title('Pick Ups by Month | Uber')
```

Out[38]: `<matplotlib.text.Text at 0x1c517ee3278>`



This project will be continued