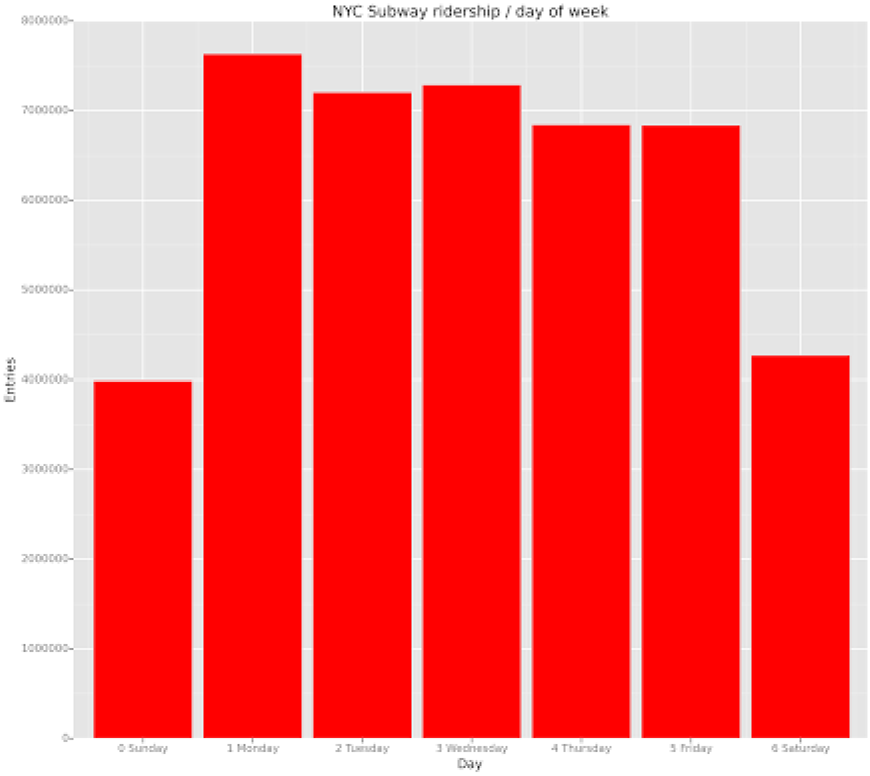
Problem Set 4: Visualizing Subway Data

Exercise - Visualization 1:

```
from pandas import *
from ggplot import *

def plot_weather_data(turnstile_weather):
'''
   You are passed in a dataframe called turnstile_weather.
   Use turnstile_weather along with ggplot to make a data visualization
   focused on the MTA and weather data we used in assignment #3.
   You should feel free to implement something that we discussed in class
   (e.g., scatterplots, line plots, or histograms) or attempt to implement
   something more advanced if you'd like.

   Here are some suggestions for things to investigate and illustrate:
    * Ridership by time of day or day of week
    * How ridership varies based on Subway station
    * Which stations have more exits or entries at different times of day
      (You can use UNIT as a proxy for subway station.)

   If you'd like to learn more about ggplot and its capabilities, take
   a look at the documentation at:
   https://pypi.python.org/pypi/ggplot/

   You can check out:
   https://www.dropbox.com/s/meyki2wl9xfa7yk/turnstile_data_master_with_weather.csv

   To see all the columns and data points included in the turnstile_weather
   dataframe.

   However, due to the limitation of our Amazon EC2 server, we are giving you a random
   subset, about 1/3 of the actual data in the turnstile_weather dataframe.
   '''

   # your code here
pandas.options.mode.chained_assignment = None
dataTW = turnstile_weather
entries_DayOfMonth = dataTW[['DATEn', 'ENTRIESn_hourly']].groupby('DATEn',
as_index=False).sum()
entries_DayOfMonth['Day'] = [datetime.strptime(x, '%Y-%m-%d').strftime('%w %A')
for x in entries_DayOfMonth['DATEn']]
```

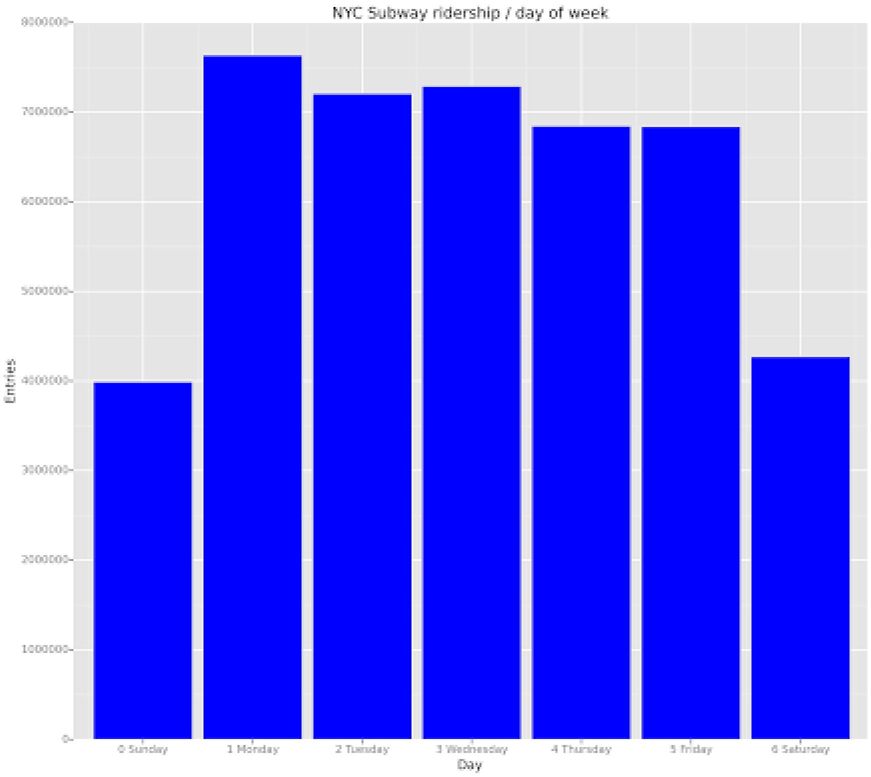
```
entries_Day = entries_DayOfMonth[['Day', 'ENTRIESn_hourly']].groupby('Day', as_index=False).sum()
plot = ggplot(entries_Day, aes(x='Day', y='ENTRIESn_hourly')) +
geom_bar(aes(weight='ENTRIESn_hourly'), stat = 'bar', fill='red') \
        + ggtitle('NYC Subway ridership / day of week') + xlab('Day') + ylab('Entries')

return plot
```



2 - Make Another Visualization:

```
from pandas import *
from ggplot import *

def plot_weather_data(turnstile_weather):
'''
   plot_weather_data is passed a dataframe called turnstile_weather.
   Use turnstile_weather along with ggplot to make another data visualization
   focused on the MTA and weather data we used in Project 3.

   Make a type of visualization different than what you did in the previous exercise.
   Try to use the data in a different way (e.g., if you made a lineplot concerning
   ridership and time of day in exercise #1, maybe look at weather and try to make a
```

```
    histogram in this exercise). Or try to use multiple encodings in your graph if
    you didn't in the previous exercise.

    You should feel free to implement something that we discussed in class
    (e.g., scatterplots, line plots, or histograms) or attempt to implement
    something more advanced if you'd like.

    Here are some suggestions for things to investigate and illustrate:
     * Ridership by time-of-day or day-of-week
     * How ridership varies by subway station
     * Which stations have more exits or entries at different times of day
       (You can use UNIT as a proxy for subway station.)

    If you'd like to learn more about ggplot and its capabilities, take
    a look at the documentation at:
    https://pypi.python.org/pypi/ggplot/

    You can check out the link
    https://www.dropbox.com/s/meyki2wl9xfa7yk/turnstile_data_master_with_weather.csv
    to see all the columns and data points included in the turnstile_weather
    dataframe.

   However, due to the limitation of our Amazon EC2 server, we are giving you a random
    subset, about 1/3 of the actual data in the turnstile_weather dataframe.
    '''

    # your code here
pandas.options.mode.chained_assignment = None
dataTW = turnstile_weather
entries_DayOfMonth = dataTW[['DATEn', 'ENTRIESn_hourly']].groupby('DATEn',
as_index=False).sum()
entries_DayOfMonth['Day'] = [datetime.strptime(x, '%Y-%m-%d').strftime('%w %A')
for x in entries_DayOfMonth['DATEn']]
entries_Day = entries_DayOfMonth[['Day', 'ENTRIESn_hourly']].groupby('Day', as_index=False).sum()
plot = ggplot(entries_Day, aes(x='Day', y='ENTRIESn_hourly',)) +
geom_bar(aes(weight='ENTRIESn_hourly'),  stat='bar', fill='blue') + ggtitle('NYC Subway ridership /
day of week') + xlab('Day') + ylab('Entries')
return plot
```


NYC Subway ridership / day of week
Entries
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
0 Sunday
1 Monday
2 Tuesday
3 Wednesday
4 Thursday
5 Friday
6 Saturday
Day