Deshpande_Charudatta - PS1

January 10, 2018

Problem Set 1, due January 10th at 5:30pm
Student Name - Charudatta Deshpande
Collaborators - Ram Ganesan, Charles Hemstreet

0.0.1 Before You Start

Make sure to at least take a basic tutorial in the IPython notebook, otherwise you'll be totally lost. For this problem set, you should download INFX574-PS1.ipynb and the flights.zip dataset from Canvas. Create a local copy of the notebook and rename it LASTNAME_FIRSTNAME-PS1.ipynb. Then edit your renamed file directly in your browser by typing:

```
ipython notebook <name_of_downloaded_file>
```

You should also make sure the following libraries load correctly (click on the box below and hit Ctrl-Enter)

```
In [4]: # #IPython is what you are using now to run the notebook
        # import IPython
        # print "IPython version:      %6.6s (need at least 1.0)" % IPython.__version__

        # Numpy is a library for working with Arrays
        import numpy as np
        print( "Numpy version:        %6.6s (need at least 1.7.1)" )

        # SciPy implements many different numerical algorithms
        import scipy as sp
        print ("SciPy version:        %6.6s (need at least 0.12.0)" )

        # Pandas makes working with data tables easier
        import pandas as pd
        print( "Pandas version:       %6.6s (need at least 0.11.0)" )

        # Module for plotting
        import matplotlib.pyplot as mpl
        print( "Mapltolib version:    %6.6s (need at least 1.2.1)" )

        %matplotlib inline
```

```
# SciKit Learn implements several Machine Learning algorithms
import sklearn
print ("Scikit-Learn version: %6.6s (need at least 0.13.1)" )
```

```
Numpy version:        %6.6s (need at least 1.7.1)
SciPy version:        %6.6s (need at least 0.12.0)
Pandas version:       %6.6s (need at least 0.11.0)
Mapltolib version:    %6.6s (need at least 1.2.1)
Scikit-Learn version: %6.6s (need at least 0.13.1)
```

0.1 About the Problem Set:

This is the same problem set used by Emma Spiro in INFX573. The only difference is that instead of doing the problem set in R, you will use Python and the IPython notebook.

0.2 Instructions:

In this problem set you will perform a basic exploratory analysis on an example dataset, bringing to bear all of your new skills in data manipulation and visualization. You will be required to submit well commented python code, documenting all code used in this problem set, along with a write up answering all questions below. Use figures as appropriate to support your answers, and when required by the problem. This data set uses the NYCFlights13 dataset. You can download the dataset from canvas. Selected questions ask you to answer in multiple ways. Make sure to provide different functions or ways for answering the same question. This will help you see that most data questions can be answered in different ways even with the same software language.

```
In [5]: # Import required libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import os as os # Added by Charu
        os.chdir('C:\\Users\deshc\Desktop\INFX 574 Data Science 2\Problem Set 1')
```

```
In [6]: flights_df= pd.read_csv('flights.csv')
```

```
In [7]: print (flights_df.shape)
        print (flights_df.columns)
        print (flights_df.dtypes)
```

```
(336776, 17)
Index(['Unnamed: 0', 'year', 'month', 'day', 'dep_time', 'dep_delay',
       'arr_time', 'arr_delay', 'carrier', 'tailnum', 'flight', 'origin',
       'dest', 'air_time', 'distance', 'hour', 'minute'],
      dtype='object')
Unnamed: 0      int64
year            int64
month           int64
day             int64
```

```
dep_time      float64
dep_delay     float64
arr_time      float64
arr_delay     float64
carrier        object
tailnum        object
flight          int64
origin         object
dest           object
air_time      float64
distance        int64
hour          float64
minute        float64
dtype: object
```

```
In [8]: # This code
        a = flights_df.dest.unique()
        print(a)
        flights_df.head(10)
```

```
['IAH' 'MIA' 'BQN' 'ATL' 'ORD' 'FLL' 'IAD' 'MCO' 'PBI' 'TPA' 'LAX' 'SFO'
 'DFW' 'BOS' 'LAS' 'MSP' 'DTW' 'RSW' 'SJU' 'PHX' 'BWI' 'CLT' 'BUF' 'DEN'
 'SNA' 'MSY' 'SLC' 'XNA' 'MKE' 'SEA' 'ROC' 'SYR' 'SRQ' 'RDU' 'CMH' 'JAX'
 'CHS' 'MEM' 'PIT' 'SAN' 'DCA' 'CLE' 'STL' 'MYR' 'JAC' 'MDW' 'HNL' 'BNA'
 'AUS' 'BTV' 'PHL' 'STT' 'EGE' 'AVL' 'PWM' 'IND' 'SAV' 'CAK' 'HOU' 'LGB'
 'DAY' 'ALB' 'BDL' 'MHT' 'MSN' 'GSO' 'CVG' 'BUR' 'RIC' 'GSP' 'GRR' 'MCI'
 'ORF' 'SAT' 'SDF' 'PDX' 'SJC' 'OMA' 'CRW' 'OAK' 'SMF' 'TUL' 'TYS' 'OKC'
 'PVD' 'DSM' 'PSE' 'BHM' 'CAE' 'HDN' 'BZN' 'MTJ' 'EYW' 'PSP' 'ACK' 'BGR'
 'ABQ' 'ILM' 'MVY' 'SBN' 'LEX' 'CHO' 'TVC' 'ANC' 'LGA']
```

```
Out[8]:    Unnamed: 0  year  month  day  dep_time  dep_delay  arr_time  arr_delay  \
        0           1  2013      1    1     517.0        2.0     830.0       11.0
        1           2  2013      1    1     533.0        4.0     850.0       20.0
        2           3  2013      1    1     542.0        2.0     923.0       33.0
        3           4  2013      1    1     544.0       -1.0    1004.0      -18.0
        4           5  2013      1    1     554.0       -6.0     812.0      -25.0
        5           6  2013      1    1     554.0       -4.0     740.0       12.0
        6           7  2013      1    1     555.0       -5.0     913.0       19.0
        7           8  2013      1    1     557.0       -3.0     709.0      -14.0
        8           9  2013      1    1     557.0       -3.0     838.0       -8.0
        9          10  2013      1    1     558.0       -2.0     753.0        8.0

          carrier tailnum  flight origin dest  air_time  distance  hour  minute
        0      UA  N14228    1545    EWR  IAH     227.0      1400   5.0    17.0
        1      UA  N24211    1714    LGA  IAH     227.0      1416   5.0    33.0
        2      AA  N619AA    1141    JFK  MIA     160.0      1089   5.0    42.0
```

3	B6	N804JB	725	JFK	BQN	183.0	1576	5.0	44.0
4	DL	N668DN	461	LGA	ATL	116.0	762	5.0	54.0
5	UA	N39463	1696	EWR	ORD	150.0	719	5.0	54.0
6	B6	N516JB	507	EWR	FLL	158.0	1065	5.0	55.0
7	EV	N829AS	5708	LGA	IAD	53.0	229	5.0	57.0
8	B6	N593JB	79	JFK	MCO	140.0	944	5.0	57.0
9	AA	N3ALAA	301	LGA	ORD	138.0	733	5.0	58.0

0.3 Some Tips

- This assignment involves extensive Data frame splitting and aggregation. You should look into the details of the methods groupby, transform, sum, count, mean etc
- Many of the tasks in the assignment can be done either through the Pandas Data Frame or by converting the data frames to Series. Many of the methods in the numpy are applicable to Series only. When stuck, try to explore the type of object (Pandas Data Frame or Numpy Series) you are dealing with.

0.4 Question 1

Let's explore flights from NYC to Seattle. Use the flights dataset to answer the following questions.

(a) How many flights were there from NYC airports to Seattle in 2013?

```
In [9]: # Your code here
        flights_df1=flights_df[ flights_df.dest=='SEA']
        flights_df1.shape[0]
```

```
Out[9]: 3923
```

-- Write your answer in English here --
There were 3923 flights from NYC area airports to Seattle in 2013.

(b) How many airlines fly from NYC to Seattle?

```
In [10]: # Your code here
         # Identiy unique carriers and print their values
         carriersN_to_seattle = flights_df1.carrier.unique()
         a=carriersN_to_seattle.shape[0]
         print("Total airlines from NYC to SEA are : " , a)
         print("They are :", carriersN_to_seattle)
```

```
Total airlines from NYC to SEA are :  5
They are : ['AS' 'DL' 'UA' 'B6' 'AA']
```

-- Write your answer in English here --
Total airlines from NYC to SEA are : 5 They are : ['AS' 'DL' 'UA' 'B6' 'AA']

(c) How many unique air planes fly from NYC to Seattle?

```
In [11]: # Your code here
         # Find number of entries with unique tail number.
         # This will return the number of unique air planes.
         flights_df1['tailnum'].nunique()
```

```
Out[11]: 935
```

-- Write your answer in English here --
A total of 935 unique air planes flew from NYC area to Seattle in 2013.

(d) What is the average arrival delay for flights from NC to Seattle?

```
In [12]: # Your code here
         # Find mean arrival delay.
         flights_df1['arr_delay'].mean()
```

```
Out[12]: -1.0990990990990992
```

-- Write your answer in English here --
Mean arrival delay from NYC to Seattle flights is -1.099 minutes. This means flights to Seattle are actually early (negative delay).

(e) What proportion of flights to Seattle come from each NYC airport? Provide multiple ways of answering the question.

```
In [57]: # Your code here

         #total flights from NYC to SEA

         cnt_SEA = flights_df1.shape[0]

         # find unique origins

         nyc_origins = flights_df1['origin'].unique()

         # print(nyc_origins)

         print("method one:")
         for i in nyc_origins:
             x=(flights_df1[flights_df1['origin']==i].shape[0])
             print(i,": ",round(100*x/cnt_SEA,2),"%")

         #-------- Second method - use value counts #

         JFK = flights_df1['origin'].value_counts()[0]
         EWR = flights_df1['origin'].value_counts()[1]
         Total = JFK + EWR
         EWR_percent = EWR * 100/Total
         JFK_percent = JFK * 100/Total
```

```
print("method two:")
print("EWR percent: ", round(EWR_percent,2),"%")
print("JFK percent: ", round(JFK_percent,2),"%")
```

```
method one:
EWR :  46.67 %
JFK :  53.33 %
method two:
EWR percent:  46.67 %
JFK percent:  53.33 %
```

-- Write your answer in English here --
EWR percent: 46.67 % JFK percent: 53.33 %

0.5 Question 2

Flights are often delayed. Consider the following questions exploring delay patterns.

(a) Which date has the largest average departure delay? Which date has the largest average arrival delay?

```
In [65]: # Your code here
         avg_by_date = pd.DataFrame(flights_df.groupby(['year', 'month', 'day']).mean())
         y =avg_by_date.sort_values(['dep_delay'], ascending = False)['dep_delay'].head(1)
         x =avg_by_date.sort_values(['arr_delay'], ascending = False)['arr_delay'].head(1)
         print(y)
         print(x)
```

```
year  month  day
2013  3      8      83.536921
Name: dep_delay, dtype: float64
year  month  day
2013  3      8      85.862155
Name: arr_delay, dtype: float64
```

-- Write your answer in English here --
March 8, 2013 is the day with largest arrival and departure delays.

(b) What was the worst day to fly out of NYC in 2013 if you dislike delayed flights?

```
In [66]: # Your code here
         # Create a dataframe of delayed flights
         Delayed_Flights = flights_df.loc[:,["year","month","day","dep_delay"]][flights_df["de
         del_by_date = Delayed_Flights.groupby(['year', 'month', 'day']).count()
         # Calculate max departure delay
         max_del_cnt = max(del_by_date["dep_delay"])
         print(del_by_date[del_by_date["dep_delay"]==max_del_cnt])
```

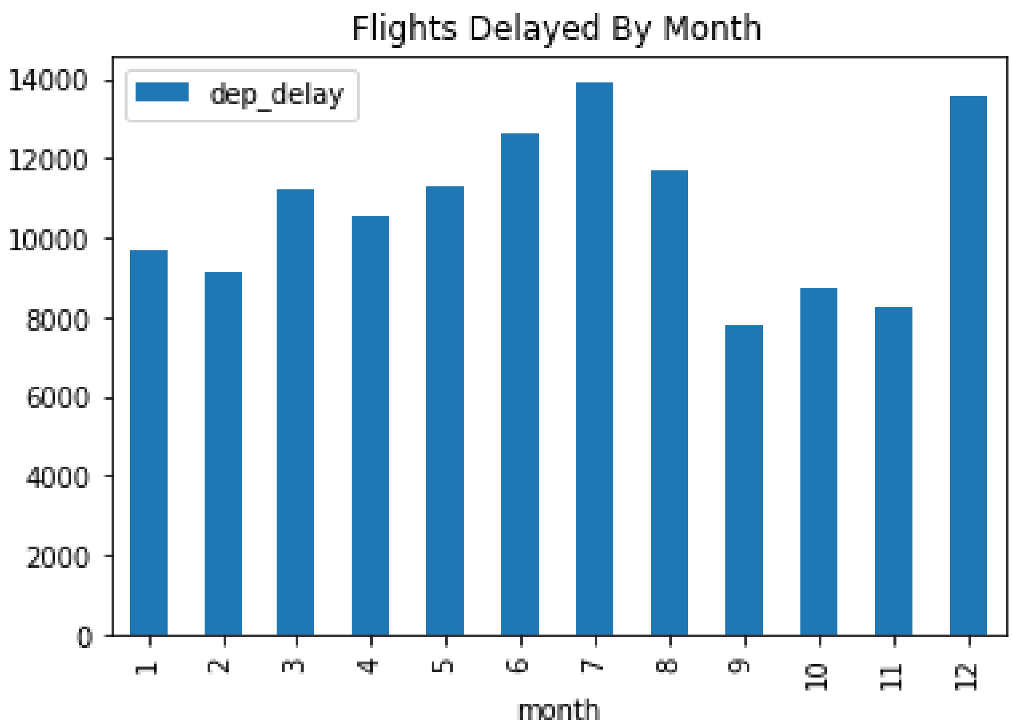
```
                 dep_delay
year month day
2013 12    23          674
```

-- Write your answer in English here --

December 23rd, 2013 was the worst day to fly out of NYC. This could be attributed to holiday time.

(c) Are there any seasonal patterns in departure delays for flights from NYC?

```
In [67]: # Your code here
         # Create a plot of delay by month
         df_by_month = flights_df.loc[:,["month","dep_delay"]][flights_df["dep_delay"]>0].grou
         p=df_by_month.plot(kind='bar', title="Flights Delayed By Month")
```



-- Write your answer in English here --

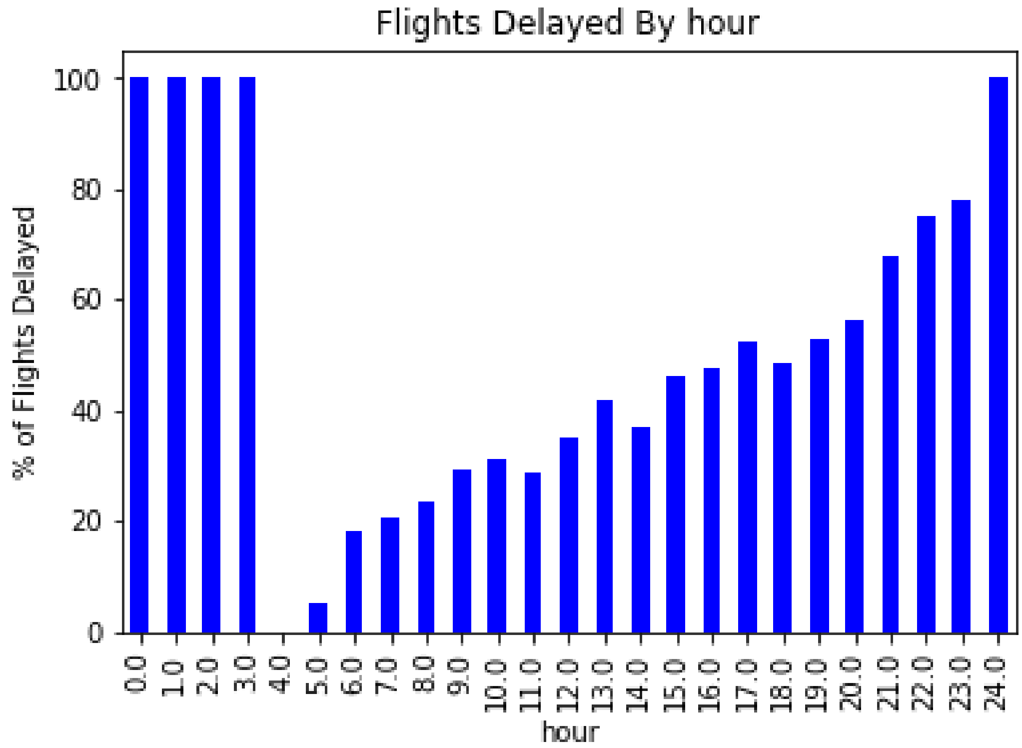
The delay seems to be more during holiday time (Dec, Jan) and Summer time (June thru August)

(d) On average, how do departure delays vary over the course of a day?

```
In [69]: # Your code here
         # Group the data by hour and calculate average delay
         df_by_hour = flights_df.loc[:,["hour","dep_delay"]].groupby('hour').count()
```

```
df_by_hour.rename(columns={'dep_delay': 'Total_Flights'}, inplace=True)
df_by_hour_del = flights_df.loc[:,["hour","dep_delay"]][flights_df["dep_delay"]>0].gro
df_by_hour_del.rename(columns={'dep_delay': 'Total_Delayed_Flights'}, inplace=True)
# Join two data frames (hour and delay) to create a single data frame and plot a bar
df_join=df_by_hour.join(df_by_hour_del,how='left')
df_join["Percent_Delayed"] = round(100*df_join["Total_Delayed_Flights"].div(df_join["
p=df_join.Percent_Delayed.plot(kind='bar', title="Flights Delayed By hour",color="blue
p.set_ylabel('% of Flights Delayed')
```

Out[69]: Text(0,0.5,'% of Flights Delayed')



-- Write your answer in English here --

We note that -

Percent of Flight delays increase as the day progresses. Percent of Flight delays are worst in late night hours. This could be possibly due to understaffing at late hours in the night.

0.6 Question 3

Which flight departing NYC in 2013 flew the fastest?

In [70]:
```
# Your code here
# Calculate speed based on distance and air time.
flights_df['speed'] = flights_df['distance'] / (flights_df['air_time']/60)
Fastest_flight = flights_df[(flights_df['speed'] == flights_df['speed'].max())]
print(Fastest_flight)
```

```
        Unnamed: 0  year  month  day  dep_time  dep_delay  arr_time  \
216447      216448  2013      5   25    1709.0        9.0    1923.0

        arr_delay carrier tailnum  flight origin dest  air_time  distance  \
216447      -14.0      DL  N666DN    1499    LGA  ATL      65.0       762

        hour  minute       speed
216447  17.0     9.0  703.384615
```

-- Write your answer in English here --
The flight from LGA to ATL on May 25, 2013 (flight number 1499) was the fastest.

0.7 Question 4

Which flights (i.e. carrier + flight + dest) happen every day? Where do they fly to?

```
In [75]: # Your code here

         # find carrier and flight combination that happens each day of the year
         flightdata = flights_df.loc[:,["carrier","flight", "dest", "year","month","day" ]]

         # Ignore duplicates, just keep first record
         flightdata = flightdata.drop_duplicates(subset=None, keep='first', inplace=False)
         flightdata = flightdata.groupby(['carrier','flight','dest']).count().reset_index()
         funiquelightdata = flightdata[(flightdata['year'] == 365)]

         # print the data in tabular format.
         funiquelightdata = funiquelightdata.loc[:,['carrier','flight','dest']]
         funiquelightdata
```

Out[75]:

	carrier	flight	dest
767	AA	59	SFO
775	AA	119	LAX
783	AA	181	LAX
904	AA	1357	SJU
914	AA	1611	MIA
1118	B6	219	CLT
1147	B6	359	BUR
1150	B6	371	FLL
1169	B6	431	SRQ
1243	B6	703	SJU
1379	B6	1783	MCO
2012	DL	2159	MCO
2081	DL	2391	TPA
4631	EV	5712	IAD
5116	UA	15	HNL
10607	VX	251	LAS

```
10609       VX     407   LAX
10613       VX     413   LAX
```

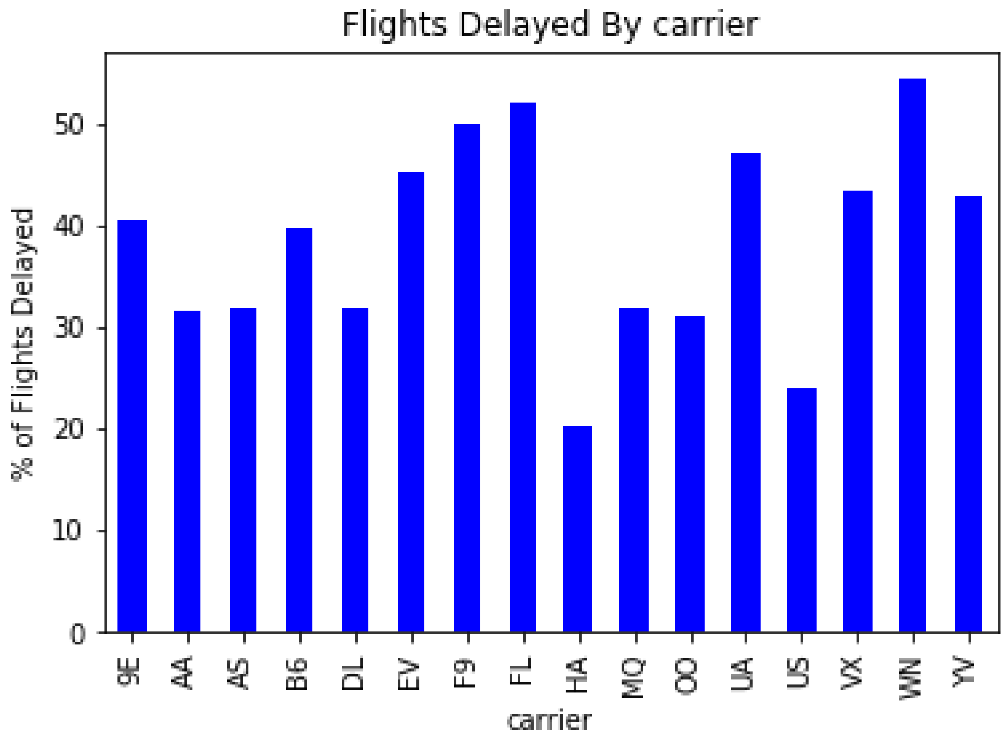
-- Write your answer in English here --
Above chart represents the 18 flights that fly every day. The destinations are printed.

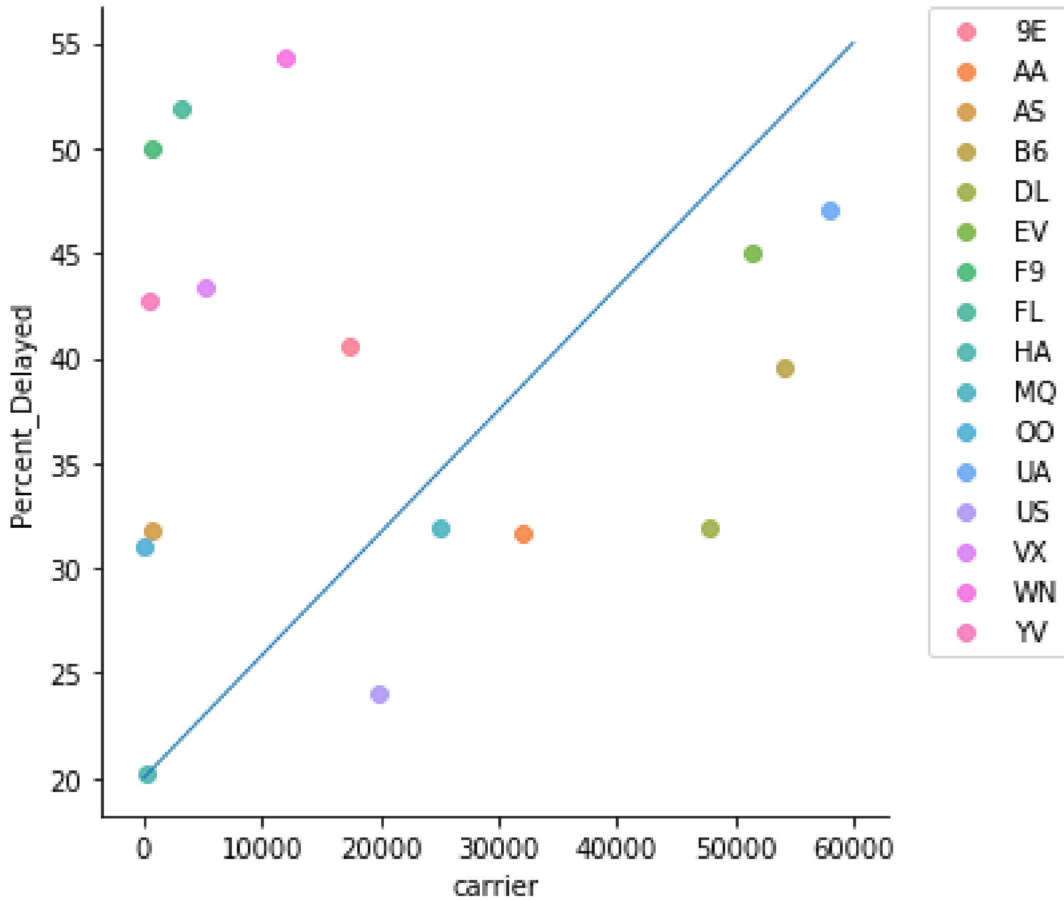
0.8 Question 5

Develop one research question you can address using the nycflights2013 dataset. Provide two visualizations to support your exploration of this question. Discuss what you find.

In [76]:
```
# Your code here

import seaborn as sns

# group by carrier, create new data frames for total flights and total delayed flight
df_by_carrier = flights_df.loc[:,["carrier","dep_delay"]].groupby('carrier').count()
df_by_carrier.rename(columns={'dep_delay': 'Total_Flights'}, inplace=True)
df_by_carrier_del = flights_df.loc[:,["carrier","dep_delay"]][flights_df["dep_delay"]
df_by_carrier_del.rename(columns={'dep_delay': 'Total_Delayed_Flights'}, inplace=True

# Join carrier dataframe and delay dataframes.
df_join=df_by_carrier.join(df_by_carrier_del,how='left')
df_join["Percent_Delayed"] = round(100*df_join["Total_Delayed_Flights"].div(df_join["
p1=df_join.Percent_Delayed.plot(kind='bar', title="Flights Delayed By carrier",color=
p1.set_ylabel('% of Flights Delayed')
df_join['carrier']=df_join.index

#Plot the results in bar plot and scatter plot
sns.lmplot(x='Total_Flights', y='Percent_Delayed', data=df_join, fit_reg=False, hue='
plt.xlabel("carrier")
plt.legend(bbox_to_anchor=(1.05,1), loc=2, borderaxespad=0.)
plt.plot([0,60000], [20,55], linewidth=1)
```

Out[76]: [<matplotlib.lines.Line2D at 0x1a065fbd8d0>]


Flights Delayed By carrier
% of Flights Delayed
0
10
20
30
40
50
9E
AA
AS
B6
DL
EV
F9
FL
HA
MQ
OO
UA
US
VX
WN
YV
carrier



-- Enter your discussion here --

Research question -

Which are the worst performing airlines operating out of NYC (airlines with highest percentage of flight delays)? Does number of flights operated by the airlines have a correlation with this delay?

Answer -

1. Referring to visualization 1 - bar chart -

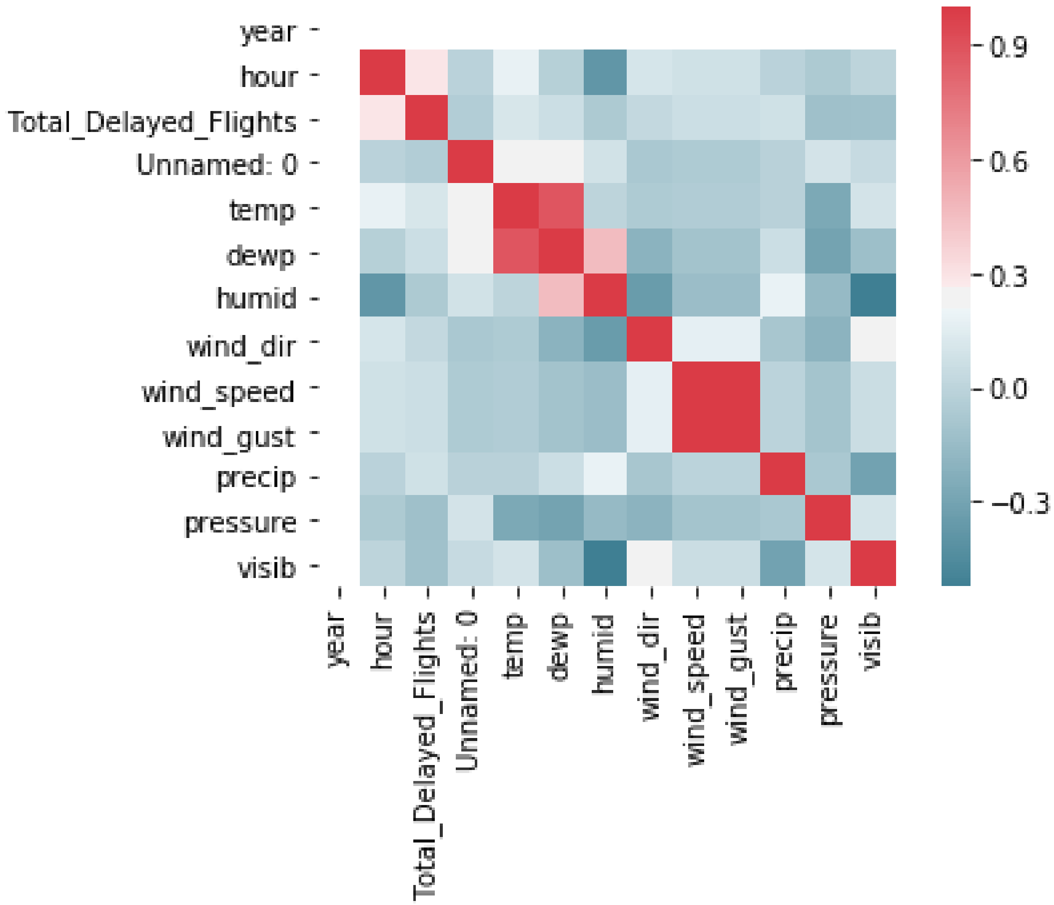
F9, FL and WN are the airlines with highest amount of delays (worst performing). This may not directly relate to airline efficiency as there are many factors that contribute to delays (priority, amount of fees that airlines pay etc).

2. The second visualization is a scatter plot that indicates relation between total flights that the airlines operate on x-axis (vs) percetage of flight delays on y-axis. The 45 degree line is provided for reference. The airlines that are to the left of the line indicate airlines that operate lower number of flights, yet have significant delays. The airlines on the right of the line operate much higher number of flights. Yet they experience smaller amounts of delays than the ones on left.

0.9 Question 6

What weather conditions are associated with flight delays leaving NYC? Use graphics to explore.

```
In [80]: # Your code here
         # create data frame for delayed flights
         df_delayed = flights_df[flights_df["dep_delay"]>0]
         df_delayed = df_delayed.loc[:,["origin","year","month","day","hour", "dep_delay"]].gro
         df_delayed.rename(columns={'dep_delay': 'Total_Delayed_Flights'}, inplace=True)

         # read weather file, create a dataframe.
         weather_df = pd.read_csv("weather.csv")
         df_join_wthr = pd.merge(df_delayed,weather_df)

         # Join two dataframes
         df_join_wthr
         df_join_temp = df_join_wthr.loc[:,["Total_Delayed_Flights","temp"]]
         corr = df_join_wthr.corr()

         # Plot a heat map
         sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool), cmap=sns.diverging_palette
                     square=True)

Out[80]: <matplotlib.axes._subplots.AxesSubplot at 0x1a065ffe128>
```



-- Enter your interpretation here --

Above heat map rtepresents effect of weather parameters on the flight delays.

The boxes corresponding to Flight Delays variable are the correlation with other variables. Darker RED shade represents a positive correlation while darker BLUE shade represents negative correlation.

It can be seen that Temperature, Windspeed and Windgust have a positive correlation with flight delays whereas visibility, pressure and humidity have a negative correlation with flight delays.