FinalProject_WeatherData_2 |Untitled |Untitled |Untitled |Untitled |Untitled |Untitled |Untitled |U

Zeppelin

FinalProject_Weathe...

default

FINISHED

```
//time taken:- 30 sec
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
```

```
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
```

Took 30 sec. Last updated by anonymous at March 25 2017, 3:34:13 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 13 sec
import pandas as pd
import numpy as np
```

Took 13 sec. Last updated by anonymous at March 25 2017, 3:34:31 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 1 sec
inputPath =  "D:/Aakash_Documents/MS_Collections/AcceptanceFromSaintPeters/ClassStuff/DS_670_Capstone/

dewpoint_feb_jun1 = pd.read_csv(inputPath+"/dewpoint/dewptm_Feb_Jun.csv")
dewpoint_aug_sep1 = pd.read_csv(inputPath+"/dewpoint/dewptm_Aug_Sep.csv")

humidity_feb_jun1 = pd.read_csv(inputPath+"/humidity/hum_feb_jun.csv")
humidity_aug_sep1 = pd.read_csv(inputPath+"/humidity/hum_aug_sep.csv")

pressure_feb_jun1 = pd.read_csv(inputPath+"/pressure/pressurem_feb_jun.csv")
pressure_aug_sep1 = pd.read_csv(inputPath+"/pressure/pressurem_aug_sept.csv")

temp_feb_jun1 = pd.read_csv(inputPath+"/temperature/tempm_feb_jun.csv")
temp_aug_sep1 = pd.read_csv(inputPath+"/temperature/tempm_aug_sept.csv")

winddirection_feb_jun1 = pd.read_csv(inputPath+"/winddirection/wdird_feb_jun.csv")
winddirection_aug_sep1 = pd.read_csv(inputPath+"/winddirection/wdird_aug_sept.csv")

windspeed_feb_jun1 = pd.read_csv(inputPath+"/windspeed/wspdm_feb_jun.csv")
windspeed_aug_sep1 = pd.read_csv(inputPath+"/windspeed/wspdm_aug_sept.csv")
```

Took 1 sec. Last updated by anonymous at March 25 2017, 3:34:51 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
winddirection_feb_jun1.ix[:,1]
```

```
8205    170
```

FinalProject_WeatherData_2

```
8207    230
8208    190
8209    210
8210    160
8211    290
8212    100
8213    130
8214    130
8215    160
8216      0
8217    170
8218    170

8219    160
8220    120
8221    180
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:34:56 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
##concatinate the data first row wise.
dewpoint = pd.concat([dewpoint_feb_jun1,dewpoint_aug_sep1])
humidity = pd.concat([humidity_feb_jun1,humidity_aug_sep1])
pressure = pd.concat([pressure_feb_jun1,pressure_aug_sep1])
temperature = pd.concat([temp_feb_jun1,temp_aug_sep1])
winddirection = pd.concat([winddirection_feb_jun1,winddirection_aug_sep1])
windspeed = pd.concat([windspeed_feb_jun1,windspeed_aug_sep1])
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:04 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
##let us verify the length of dataframes, it must be same.
print(len(dewpoint))
print(len(humidity))
print(len(pressure))
print(len(temperature))
print(len(winddirection))
print(len(windspeed))
```

```
12579
12579
12579
12579
12579
12579
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:11 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
##concatenate the data frames column wise now
weather_dataset = pd.concat([dewpoint,humidity.ix[:,1],pressure.ix[:,1],temperature.ix[:,1],winddirect
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:18 PM. (outdated)

```
%pyspark
#time taken:- 1 second
weather_dataset.ix[:,1:].sum()
```

```
DewPoint            79816.0
Humidity           889275.0
Pressure         12730143.0
Temperature        142277.0
WindDirection     2285930.0
WindSpeed          152979.3
dtype: float64
```

Took 1 sec. Last updated by anonymous at March 25 2017, 3:35:23 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
weather_dataset.ix[:,1:].sum(axis=1)
```

```
4340    1225.3
4341    1230.3
4342    1246.5
4343    1235.1
4344    1186.1
4345    1232.7
4346    1227.1
4347    1255.5
4348    1303.0
4349    1239.0
4350    1235.5
4351    1228.4
4352    1227.7
4353    1237.6
4354    1251.8
4355    1226.3
4356    1220.0
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:38 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
weather_dataset.ix[:,1:].mean(axis=1,skipna=False)
```

```
18      211.950000
19      209.066667
20      212.016667
21      210.666667
22      207.783333
23      206.066667
24      213.016667
25      210.400000
26      210.966667
27      210.833333
28      212.733333
29      210.533333
           ...
4327    205.183333
4328    204.716667
4329    205.550000
```

```
     204.066667
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:43 PM. (outdated)

```
%pyspark
#time taken:- less than second
weather_dataset.ix[:,1:].idxmax()
```

```
DewPoint          150
Humidity          392
Pressure         1820
Temperature        80
WindDirection    1836
WindSpeed        2165
dtype: int64
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:35:49 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 3 second
weather_dataset.describe()
```

```
           DewPoint      Humidity      Pressure   Temperature  WindDirection  \
count  12563.000000  12563.000000  12558.000000  12563.000000   12463.000000
mean       6.353260     70.785083   1013.707836     11.325082     183.417315
std        4.648668     15.669026      8.981195      5.420558      88.411577
min       -9.000000     12.000000    986.000000     -3.000000       0.000000
25%        3.000000     61.000000   1008.000000      7.000000     110.000000
50%        7.000000     74.000000   1014.000000     11.000000     190.000000
75%       10.000000     82.000000   1020.000000     15.000000     260.000000
max       19.000000    100.000000   1038.000000     27.000000     360.000000
          WindSpeed
count  12510.000000
mean      12.228561
std       89.876584
min    -9999.000000
25%        7.400000
50%       11.100000
75%       16.700000
max       64.800000
```

Took 3 sec. Last updated by anonymous at March 25 2017, 3:36:02 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
#check the null values in dataframe if any
weather_dataset.isnull().any()
```

```
DateTime         False
DewPoint          True
Humidity          True
Pressure          True
Temperature       True
WindDirection     True
WindSpeed         True
dtype: bool
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:36:10 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
#check count of null values in all the columns
weather_dataset.isnull().sum()
```

```
DateTime           0
DewPoint          16
Humidity          16
Pressure          21
Temperature       16
WindDirection    116
WindSpeed         69
dtype: int64
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:36:35 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

#In this step, we will try to update null values.
#filling null values could be complicated.As we seen in previous data exploration steps
#that 116 was the maximum null values and total datasize is 12563. Since, maximum percent of null valu
#So, null values will be replaced by mean of the particular parameter.
def updatenullvalues(dataset):
    for col in dataset.ix[:,1:]:
        if dataset[col].isnull().any:
            mean = dataset[col].mean()
            dataset[col].fillna(mean,inplace=True)
    return dataset
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:37:10 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

#Let's update null values in our dataset.
weather_dataset = updatenullvalues(weather_dataset)
#verify is there still any null value left in the dataset
weather_dataset.isnull().sum()
```

```
DateTime         0
DewPoint         0
Humidity         0
Pressure         0
Temperature      0
WindDirection    0
WindSpeed        0
dtype: int64
```

Took 0 sec. Last updated by anonymous at March 25 2017, 3:37:28 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 2 second

import matplotlib.pyplot as plt

#now we are in good state as our null values are vanished.
#in this code step, we will check distribution of our data.
```

```
data = [weather_dataset.ix[:,1], weather_dataset.ix[:,2], weather_dataset.ix[:,3], weather_dataset.ix|

parameter_names = ['Dewpoint', 'Humidity', 'Pressure', 'Temperature', 'WindDirection', 'WindSpeed']

fig, axis = plt.subplots()
axis.set_title("Distribution of weather parameters")
axis.set_xlabel('Weather Parameters')
axis.set_ylabel('Values')
day_plot = plt.boxplot(data, sym='o', vert=1, whis=1.5)
plt.setp(day_plot['boxes'], color = 'black')
plt.setp(day_plot['whiskers'], color = 'black')
plt.setp(day_plot['fliers'], color = 'black', marker = 'o')
axis.set_xticklabels(parameter_names)
plt.show()
```

Took 2 sec. Last updated by anonymous at March 25 2017, 5:01:15 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
#now, let's perform data aggregation to know the hidden facts of dataset

# column-wise and Multiple Function Application
grouped_dewpoint = weather_dataset.groupby(['DewPoint'])

# get an idea of average windspeed at different levels of dewpoint
grouped_dewpoint['WindSpeed'].agg('mean')
```

```
DewPoint
-9.00000     11.733333
-8.00000     10.650000
-7.00000     11.320000
-6.00000     15.371429
-5.00000     14.823529
-4.00000     10.038412
-3.00000     14.086650
-2.00000     13.588805
-1.00000     12.904869
 0.00000     -6.784877
 1.00000     14.524253
 2.00000     14.985485
 3.00000     14.046857
 4.00000     13.896208
 5.00000     13.926867
 6.00000     13.605390
 6.35326     11.695526
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:11:33 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second
# column-wise and Multiple Function Application
grouped_pressure = weather_dataset.groupby(['Pressure'])

# get an idea of average windspeed at different levels of pressure
grouped_pressure['WindSpeed'].agg('mean')
```

```
1022.000000    12.183312
1023.000000    10.889481
```

```
1025.000000    12.255557
1026.000000    10.248894
1027.000000    10.849681
1028.000000    11.704578
1029.000000    14.470707
1030.000000    12.865714
1031.000000     9.023529
1032.000000    10.830357
1033.000000     9.664912
1034.000000     8.708333
1035.000000     9.270000
1036.000000     8.042857
1037.000000     6.344000
1038.000000     4.100000
Name: WindSpeed, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:15:37 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

# column-wise and Multiple Function Application
grouped_humidity = weather_dataset.groupby(['Humidity'])

# get an idea of average windspeed at different levels of humidity
grouped_humidity['WindSpeed'].agg('mean')
```

```
Humidity
12.000000    13.000000
13.000000    11.100000
15.000000    11.733333
16.000000    10.200000
17.000000     9.275000
18.000000    16.066667
19.000000    12.714286
20.000000    13.388889
21.000000    14.988094
22.000000    11.883333
23.000000    13.000000
24.000000     8.900000
25.000000     9.900000
26.000000    14.480952
27.000000    11.310000
28.000000    14.920000
29.000000    14.995238
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:16:30 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

# column-wise and Multiple Function Application
grouped_temp = weather_dataset.groupby(['Temperature'])

# get an idea of average windspeed at different levels of temperature
grouped_temp['WindSpeed'].agg('mean')
```

```
11.000000    12.010995
11.325082    11.695526
12.000000    11.869129
```

```
     00000    12.984498
  .000000    12.972555
16.000000    12.575000
17.000000    13.187631
18.000000    13.004934
19.000000    13.038578
20.000000    12.463614
21.000000    13.054048
22.000000    12.155828
23.000000    13.350562
24.000000    14.826786
25.000000     9.288462
26.000000     8.125000
27.000000    10.328571
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:17:54 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

# column-wise and Multiple Function Application
grouped_wdir = weather_dataset.groupby(['WindDirection'])

# get an idea of average windspeed at different levels of winddirection
grouped_wdir['WindSpeed'].agg('mean')
```

```
WindDirection
0.000000     -18.288482
10.000000     14.925926
20.000000     14.280909
30.000000     13.513475
40.000000     12.493506
50.000000     12.476923
60.000000     13.759336
70.000000     13.572848
80.000000     13.593025
90.000000     13.279126
100.000000    12.466133
110.000000    13.037070
120.000000    12.345444
130.000000    12.114730
140.000000    11.291940
150.000000    11.051722
160.000000    11.433779
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:19:20 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

def peak_to_peak(arr): return arr.max() - arr.min()
print(grouped_dewpoint.agg(['mean','std',peak_to_peak]))
```

```
 2.00000    85.495792         360  14.985485    8.481662     53.700000
 3.00000    80.720733         360  14.046857    9.205188     51.900000
 4.00000    87.810800         360  13.896208    7.986467     50.000000
 5.00000    93.620947         360  13.926867    8.957176     51.900000
 6.00000    86.238151         360  13.605390    9.340400     48.200000
 6.35526    0.000000            0  11.695526    3.132140      8.528561
```

```
7.00000   86.588146          360  11.567522    6.499150     35.300000
8.00000   7[illegible]        360  10.753933    6.212857     37.000000
9.00000   80.700559          360  11.686622    6.610552     33.300000
10.00000  78.459902          360  12.319274    6.928568     38.900000
11.00000  87.[illegible]     360  13.250440    6.84[illegible] 35.200000
12.00000  86.567135          360  13.117497    6.593684     35.200000
13.00000  78.570522          360  10.926104    6.305346     35.200000
14.00000  63.094542          360  10.814185    6.071400     31.500000
15.00000  68.309134          330   9.492754    5.028222     22.200000
16.00000  81.356363          250  11.604545    6.358917     22.200000
17.00000  64.097319          310  13.052083    7.138359     27.700000
18.00000  51.404516          250  16.726471    7.527175     29.600000
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:26:02 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

print(grouped_pressure.agg(['mean','std',peak_to_peak]))
```

```
1019.000000  103.265407          360  10.920965    6.539162          29.6
1020.000000  105.661905          360  12.251688    7.876031          33.3
1021.000000   96.699188          360  10.975955    7.541806          31.5
1022.000000  101.040447          360  12.183312    7.030416          27.8
1023.000000   98.523266          360  10.889481    6.853375          27.8
1024.000000   99.649108          360  12.170378    7.501655          33.3
1025.000000   90.336706          360  12.235657    7.100778          33.3
1026.000000   94.830730          360  10.248894    6.391423          33.3
1027.000000   91.123808          360  10.849681    5.510552          31.5
1028.000000   99.131413          340  11.704578    6.538266          33.3
1029.000000  100.437125          340  14.470707    8.203698          33.3
1030.000000   95.539265          340  12.865714    7.150451          31.4
1031.000000   75.319450          330   9.023529    2.586240          11.1
1032.000000   32.066793          150  10.830357    3.473480          13.0
1033.000000   70.645963          360   9.664912    4.087459          18.5
1034.000000   51.444993          290   8.708333    5.185219          18.5
1035.000000   21.832697           70   9.270000    6.101375          14.8
1036.000000   20.470653           80   8.042857    4.156510          13.0
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:27:03 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

print(grouped_humidity.agg(['mean','std',peak_to_peak]))
```

```
           DewPoint                          Pressure              \
               mean       std peak_to_peak        mean       std
Humidity
12.000000 -9.000000       NaN            0 1024.000000       NaN
13.000000 -8.000000       NaN            0 1024.000000       NaN
15.000000 -5.000000  3.464102            6 1022.333333  8.020806
16.000000 -0.500000  3.109126            7 1017.500000  5.000000
17.000000 -4.500000  3.696846            8 1023.000000  6.976150
18.000000 -2.666667  4.163332            8 1023.666667  8.082904
19.000000  0.428571  2.507133            7 1015.428571  5.711309
20.000000 -1.222222  3.113590            7 1024.777778  8.700255
21.000000 -0.666667  3.326660            7 1022.000000  9.736529
```

```
22.000000  -0.833333  3.149287            18  1019.416667   6.8689
23.000000  -1.500000  0.707107             1  1021.000000   8.485281
24.000000  -0.800000  6.196773            18  1020.800000   7.757434
25.000000  -0.222222  4.711098            15  1022.555556  10.465552
26.000000  -0.666667  5.15105             16  1016.619048   8.564322
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:28:57 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

print(grouped_temp.agg(['mean','std',peak_to_peak]))
```

```
             DewPoint                          Humidity             \
                 mean       std peak_to_peak       mean        std   
Temperature                                                          
-3.000000   -4.000000  0.000000            0  90.800000   3.675746   
-2.000000   -3.333333  0.480384            1  89.370370   3.454631   
-1.000000   -2.609756  0.737497            2  86.926829   5.569516   
 0.000000   -2.200000  0.935188            3  82.700000   8.097499   
 1.000000   -0.989796  1.188388            4  85.581633   7.294856   
 2.000000   -0.288889  1.376057            5  83.185185   8.997666   
 3.000000    0.701587  1.241467            6  83.346032   8.509134   
 4.000000    1.533762  1.367197            9  82.038585   8.936263   
 5.000000    2.210733  1.422102           10  80.643979   8.895429   
 6.000000    2.358116  1.626206            9  75.550186  10.057879   
 7.000000    2.898592  2.124044           10  73.514085  12.125423   
 8.000000    3.499369  2.416863           11  71.397226  13.459811   
 9.000000    3.918167  3.184982           14  69.530278  16.217385   
 10.000000   5.278912  3.157928           16  71.288435  16.163961   
 11.000000   6.559254  3.382920           18  73.673768  16.343687   
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:29:19 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

print(grouped_wdir.agg(['mean','std',peak_to_peak]))
```

```
               DewPoint                          Humidity             \
                   mean       std peak_to_peak       mean        std   
WindDirection                                                          
0.000000       5.198953  4.758438           25  72.829843  16.181057   
10.000000      5.768519  4.711502           18  62.157407  15.058864   
20.000000      6.554545  5.349417           22  63.436364  17.230508   
30.000000      5.375887  5.046270           24  63.822695  16.205063   
40.000000      5.435065  5.068067           24  64.363636  16.860372   
50.000000      6.664835  4.808231           23  62.939560  16.096109   
60.000000      6.964912  4.786968           26  64.337719  16.366533   
70.000000      6.900641  4.372495           21  62.849359  15.314487   
80.000000      7.167647  4.517238           23  64.597059  15.766904   
90.000000      7.265882  4.597209           19  64.131765  14.982901   
100.000000     6.840000  4.676257           22  69.890667  14.237718   
110.000000     6.648188  5.663451           25  70.786780  14.799708   
120.000000     7.031496  5.786043           24  71.986220  14.204404   
130.000000     7.497608  5.053787           21  73.122010  14.489290   
140.000000     7.662687  4.894970           21  74.537313  13.925947   
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:29:40 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

#let's check correlation between windspeed and other variables
from scipy.stats.stats import pearsonr
help(pearsonr)
```

```
    Parameters
    ----------
    x : (N,) array_like
        Input
    y : (N,) array_like
        Input

    Returns
    -------
    r : float
        Pearson's correlation coefficient
    p-value : float
        2-tailed p-value

    References
    ----------
    http://www.statsoft.com/textbook/glosp.html#Pearson%20Correlation
```

Took 0 sec. Last updated by anonymous at March 25 2017, 6:09:57 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

pearsonr(weather_dataset['DewPoint'], weather_dataset['WindSpeed'])
print("Pearson's correlation coefficient, between dewpoint & windspeed",pearsonr(weather_dataset['DewP
print("P-Value is",pearsonr(weather_dataset['DewPoint'], weather_dataset['WindSpeed'])[1])
```

```
("Pearson's correlation coefficient, between dewpoint & windspeed", 0.0032535097934414709)
('P-Value is', 0.71521157962571547)
```

Took 0 sec. Last updated by anonymous at March 25 2017, 6:20:10 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 1 second

pearsonr(weather_dataset['Humidity'], weather_dataset['WindSpeed'])
print("Pearson's correlation coefficient, between humidity & windspeed",pearsonr(weather_dataset['Humi
print("P-Value is",pearsonr(weather_dataset['Humidity'], weather_dataset['WindSpeed'])[1])
```

```
("Pearson's correlation coefficient, between humidity & windspeed", -0.011101209855137389)
('P-Value is', 0.21313782000295253)
```

Took 1 sec. Last updated by anonymous at March 25 2017, 6:20:52 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

pearsonr(weather_dataset['Pressure'], weather_dataset['WindSpeed'])
print("Pearson's correlation coefficient, between pressure & windspeed",pearsonr(weather_dataset['Pres
print("P-Value is",pearsonr(weather_dataset['Pressure'], weather_dataset['WindSpeed'])[1])
```

```
("Pearson's correlation coefficient, between pressure & windspeed", -0.0235369955238600085)
('P-Value is', 0.00977[illegible]98392302792)
```

Took 0 sec. Last updated by anonymous at March 25 2017, 6:21:45 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

pearsonr(weather_dataset['Temperature'], weather_dataset['WindSpeed'])
print("Pearson's correlation coefficient, between temperature & windspeed",pearsonr(weather_dataset['T
print("P-Value is",pearsonr(weather_dataset['Temperature'], weather_dataset['WindSpeed'])[1])
```

```
("Pearson's correlation coefficient, between temperature & windspeed", 0.008056395105400762)
('P-Value is', 0.36626109035149479)
```

Took 0 sec. Last updated by anonymous at March 25 2017, 6:22:19 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- less than second

pearsonr(weather_dataset['WindDirection'], weather_dataset['WindSpeed'])
print("Pearson's correlation coefficient, between winddirection & windspeed",pearsonr(weather_dataset|
print("P-Value is",pearsonr(weather_dataset['WindDirection'], weather_dataset['WindSpeed'])[1])
```

```
("Pearson's correlation coefficient, between winddirection & windspeed", 0.029772717129442031)
('P-Value is', 0.00083898687878755358)
```

Took 0 sec. Last updated by anonymous at March 25 2017, 6:22:51 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 2 second

import matplotlib.pyplot as plt

fig, axis = plt.subplots()
axis.set_title("Relation Temperature & DewPoint")
axis.set_xlabel('Temperature')
axis.set_ylabel('DewPoint')

plt.plot(weather_dataset['Temperature'], weather_dataset['DewPoint'])
plt.show()
```

Took 2 sec. Last updated by anonymous at March 25 2017, 6:55:58 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 2 second

fig, axis = plt.subplots()
axis.set_title("Relation Temperature & Pressure")
axis.set_xlabel('Temperature')
axis.set_ylabel('Pressure')

plt.plot(weather_dataset['Pressure'], weather_dataset['Temperature'])
plt.show()
```

Took 2 sec. Last updated by anonymous at March 25 2017, 6:56:10 PM. (outdated)

FINISHED

```
%pyspark
#time taken:- 2 second
```

Zeppelin

FinalProject_Weathe... default

```
fig, axis = plt.subplots()
axis.set_title("Relation Pressure & WindSpeed")
axis.set_xlabel('Temperature')
axis.set_ylabel('Pressure')

plt.plot(weather_dataset['Pressure'], weather_dataset['WindSpeed'])
plt.show()
```

Took 2 sec. Last updated by anonymous at March 25 2017, 6:56:31 PM.

READY