[Jupyter Notebook](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/tree)

EDA\_question(unsaved changes)

Python 3

Trusted

* [File](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [Edit](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [View](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [Insert](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [Cell](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [Kernel](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)
* [Help](https://serversm43d9ju-dev-machine-server-8000.ind.hackerrank.com/notebooks/EDA_question.ipynb)

Run



**Directions**

- The data required for this task has been provided in the file 'data.csv'

- Read the questions provided for each cell and assign your answers to respective variables provided in the following cell.

- If answers are floating point numbers round of updo two floating point after the decimal

- for example 10.546 should be read as 10.55, 10.544 as 10.54 and 10.1 as 10.10

- pandas and numpy packages are preinstalled for this task which should be sufficient to complete this task.

- If you need any other additional package run !pip3 install <package\_name> --user in a new cell.

- You can either try out the solution in the same notebook or free to create additional notebook, but make sure you come back to this notebook to answer the questions.

- Please dont change variable name meant to assign your answers.

- Dont leave any of the answers blank for you test cases run smoothly

In [153]:



**import** pandas **as** pd

**import** numpy **as** np

In [155]:



*### Read the data (this will not be graded)*

data **=** pd.read\_csv('data.csv')

data.head()

Out[155]:

|  | **Day** | **Average temperature (°F)** | **Average humidity (%)** | **Average dewpoint (°F)** | **Average barometer (in)** | **Average windspeed (mph)** | **Average gustspeed (mph)** | **Average direction (°deg)** | **Rainfall for month (in)** | **Rainfall for year (in)** | **Maximum rain per minute** | **Maximum temperature (°F)** | **Minimum temperature (°F)** | **Maximum humidity (%)** | **Minimum humidity (%)** | **Maximum pressure** | **Minimum pressure** | **Maximum windspeed (mph)** | **Maximum gust speed (mph)** | **Maximum heat index (°F)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1/01/2009 | 37.8 | 35 | 12.7 | 29.7 | 26.4 | 36.8 | 274 | 0.0 | 0.0 | 0.0 | 40.1 | 34.5 | 44 | 27 | 29.762 | 29.596 | 41.4 | 59.0 | 40.1 |
| **1** | 2/01/2009 | 43.2 | 32 | 14.7 | 29.5 | 12.8 | 18.0 | 240 | 0.0 | 0.0 | 0.0 | 52.8 | 37.5 | 43 | 16 | 29.669 | 29.268 | 35.7 | 51.0 | 52.8 |
| **2** | 3/01/2009 | 25.7 | 60 | 12.7 | 29.7 | 8.3 | 12.2 | 290 | 0.0 | 0.0 | 0.0 | 41.2 | 6.7 | 89 | 35 | 30.232 | 29.260 | 25.3 | 38.0 | 41.2 |
| **3** | 4/01/2009 | 9.3 | 67 | 0.1 | 30.4 | 2.9 | 4.5 | 47 | 0.0 | 0.0 | 0.0 | 19.4 | -0.0 | 79 | 35 | 30.566 | 30.227 | 12.7 | 20.0 | 32.0 |
| **4** | 5/01/2009 | 23.5 | 30 | -5.3 | 29.9 | 16.7 | 23.1 | 265 | 0.0 | 0.0 | 0.0 | 30.3 | 15.1 | 56 | 13 | 30.233 | 29.568 | 38.0 | 53.0 | 32.0 |

**What is the standard deviation of maximum windspeed across all the days**

In [156]:



ws\_std **=** round(np.std(data['Maximum windspeed (mph)']),2)

print(ws\_std)

13.06

**What is the difference between 50th percentile and 75th percentile of average temperature**

In [157]:



p\_range **=** format((np.percentile(data['Average temperature (°F)'],75)**-**np.percentile(data['Average temperature (°F)'],50)),'.2f')

print(p\_range)

12.20

In [12]:



**!**pip3 install scipy **--**user

Collecting scipy

Downloading <https://files.pythonhosted.org/packages/7a/0e/3781e028d62a8422244582abd8f084e6314297026760587c85607f687bf3/scipy-1.3.1-cp35-cp35m-manylinux1_x86_64.whl> (25.1MB)

|████████████████████████████████| 25.1MB 639kB/s eta 0:00:01

Requirement already satisfied: numpy>=1.13.3 in /home/user/.local/lib/python3.5/site-packages (from scipy) (1.14.1)

Installing collected packages: scipy

Successfully installed scipy-1.3.1

In [158]:



**from** scipy.stats **import** pearsonr

**What is the pearson correlation between average dew point and average temperature**

In [159]:



corr, \_ **=** pearsonr(data['Average dewpoint (°F)'],data['Average temperature (°F)'])

corr **=** format(corr,'.2f')

**Out of all the available records which month has the lowest average humidity.**

- Assign your answer as month index, for example if its July index is 7

In [160]:



dew\_month **=** 1

**Which month has the highest median for maximum\_gust\_speed out of all the available records. Also find the repective value**

- hint: group by month

In [161]:



*#days=[d.split('/')[0] for d in data.Day]*

*#data['Days'] = days*

Month**=**[d.split('/')[1] **for** d **in** data.Day]

data['Month'] **=** Month

grp\_mnth **=** data.groupby('Month') [['Maximum gust speed (mph)']].median()

​

max\_gust\_value **=** grp\_mnth[['Maximum gust speed (mph)']].idxmax()

max\_gust\_month **=** grp\_mnth[['Maximum gust speed (mph)']].max()

In [162]:



days**=**[d.split('/')[0] **for** d **in** data.Day]

data['Days'] **=** days

Year**=**[d.split('/')[2] **for** d **in** data.Day]

data['Year'] **=**Year

In [163]:



**import** datetime

​

data['Day'] **=**data['Day'].apply(**lambda** x: datetime.datetime.strptime(x, '%d/%m/%Y'))

**Determine the average temperature between the months of March 2010 to May 2012 (including both the months)**

In [170]:



dateRange **=** (data['Day'] **>=** '2010-03-01') **&** (data['Day'] **<=** '2012-05-31')

df1 **=**data.loc[dateRange]

​

avg\_temp **=** df1['Average temperature (°F)'].mean()

**Find the range of averange temperature on Dec 2010**

In [180]:



dateRange **=** (data['Day'] **>=** '2010-12-01') **&** (data['Day'] **<=** '2010-12-31')

df2 **=**data.loc[dateRange]

temp\_range **=** format( np.ptp(df2['Average temperature (°F)']),'.2f')

print(temp\_range)

44.80

**Out of all available records which day has the highest difference between maximum\_pressure and minimum\_pressure**

- assign the date in string format as 'yyyy-mm-dd'. Make sure you enclose it with single quote

In [203]:



data['maxDiffPrser'] **=** data['Maximum pressure '] **-** data['Minimum pressure ']

*#data['maxDiffPrser'].max()*

​

df4**=**data.loc[data['maxDiffPrser'].idxmax()]

​

​

max\_p\_range\_day **=** df4['Day'].strftime("%Y-%m-%d")

**How many days falls under median of barrometer reading.**

In [218]:



*#data['Average barometer (in)'].median()#1518 2052*

median\_b\_days **=** data[(data['Average barometer (in)']**<** data['Average barometer (in)'].median())].shape[0]

print(median\_b\_days)

​

1518

**Out of all the available records how many days are within one standard deviation of average temperaturem**

In [233]:



stdDev **=** data['Average temperature (°F)'].std()

num\_days\_std **=** data[(data['Average temperature (°F)']**<** stdDev)].shape[0]

**Once you are done with your solution make sure you have saved the notebook (ctrl + s)**