In [1]:

```
import pandas as pd
```

we will using pandas for-

- Data import/export
- Data selection, filtering
- Statistical Analysis on data
- Data cleaning
- Data Aggregation

Pandas has 3 datatypes-

1. Pandas Series- 1D Data
2. Pandas Dataframe - 2D Data
3. Pandas Panel - 3D Data

In [5]:

```
x = {'name':  ["Ujjawal", "Gahan", "Shubham"],
     "Age":[18,19,20],
     'City':["Jaipur", "Delhi", "Mumbai"]}
x = pd.DataFrame(x)
x
```

Out[5]:

	name	Age	City
0	Ujjawal	18	Jaipur
1	Gahan	19	Delhi
2	Shubham	20	Mumbai

In [6]:

```
type(x)
```

Out[6]:

```
pandas.core.frame.DataFrame
```

In [7]:

```
pd.DataFrame?
```

In [9]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh.csv")
df.shape
```

Out[9]:

```
(20, 5)
```

In [10]:

```
df
```

Out[10]:

	Dates	Temperature	Humidity	Pressure	Air Quality
0	30-04-2018	218	182	4	2
1	01-05-2018	2592	182	3	2
2	02-05-2018	509	439	4	0
3	03-05-2018	2439	53	5	1
4	04-05-2018	824	444	5	0
5	05-05-2018	1744	443	5	1
6	06-05-2018	786	226	5	1
7	07-05-2018	1326	309	0	1
8	08-05-2018	1804	188	4	2
9	09-05-2018	109	420	0	1
10	10-05-2018	2524	433	1	0
11	11-05-2018	2945	149	1	0
12	12-05-2018	2168	531	1	1
13	13-05-2018	1318	360	3	2
14	14-05-2018	788	435	3	2
15	15-05-2018	988	259	4	0
16	16-05-2018	1454	125	5	1
17	17-05-2018	2200	325	3	2
18	18-05-2018	1722	523	0	2
19	19-05-2018	766	535	3	2

In [11]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh.csv",index_col="Dates")
df.shape
```

Out[11]:

```
(20, 4)
```

In [12]:

```
df
```

Out[12]:

Dates	Temperature	Humidity	Pressure	Air Quality
30-04-2018	218	182	4	2
01-05-2018	2592	182	3	2
02-05-2018	509	439	4	0
03-05-2018	2439	53	5	1
04-05-2018	824	444	5	0
05-05-2018	1744	443	5	1
06-05-2018	786	226	5	1
07-05-2018	1326	309	0	1
08-05-2018	1804	188	4	2
09-05-2018	109	420	0	1
10-05-2018	2524	433	1	0
11-05-2018	2945	149	1	0
12-05-2018	2168	531	1	1
13-05-2018	1318	360	3	2
14-05-2018	788	435	3	2
15-05-2018	988	259	4	0
16-05-2018	1454	125	5	1
17-05-2018	2200	325	3	2
18-05-2018	1722	523	0	2
19-05-2018	766	535	3	2

In [13]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datanh.csv")
df.shape
```

Out[13]:

```
(19, 4)
```

In [14]:

```
df
```

Out[14]:

	18	37	22	10
0	16	27	19	30
1	19	49	19	16
2	30	36	21	13
3	23	45	16	13
4	12	47	22	19
5	10	30	21	14
6	19	49	24	24
7	31	36	9	17
8	18	26	17	19
9	14	52	11	19
10	29	31	17	25
11	24	49	24	32
12	16	26	18	30
13	16	38	10	27
14	32	28	24	28
15	20	47	6	34
16	32	40	23	30
17	24	25	10	32
18	31	21	11	34

In [15]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datanh.csv",header=None)
df.shape
```

Out[15]:

```
(20, 4)
```

In [16]:

```
df
```

Out[16]:

	0	1	2	3
0	18	37	22	10
1	16	27	19	30
2	19	49	19	16
3	30	36	21	13
4	23	45	16	13
5	12	47	22	19
6	10	30	21	14
7	19	49	24	24
8	31	36	9	17
9	18	26	17	19
10	14	52	11	19
11	29	31	17	25
12	24	49	24	32
13	16	26	18	30
14	16	38	10	27
15	32	28	24	28
16	20	47	6	34
17	32	40	23	30
18	24	25	10	32
19	31	21	11	34

In [17]:

```
df.columns = ["temp", "hum", "press", "air"]
df
```

Out[17]:

	temp	hum	press	air
0	18	37	22	10
1	16	27	19	30
2	19	49	19	16
3	30	36	21	13
4	23	45	16	13
5	12	47	22	19
6	10	30	21	14
7	19	49	24	24
8	31	36	9	17
9	18	26	17	19
10	14	52	11	19
11	29	31	17	25
12	24	49	24	32
13	16	26	18	30
14	16	38	10	27
15	32	28	24	28
16	20	47	6	34
17	32	40	23	30
18	24	25	10	32
19	31	21	11	34

In [18]:

```
df = pd.read_excel(r"C:\Users\gorav\Desktop\data\data.xlsx")
df.shape
```

Out[18]:

```
(15, 5)
```

In [19]:

```
df
```

Out[19]:

	Unnamed: 0	temp	hum	press	air_q
0	2018-10-20	72	79	68	75
1	2018-10-21	59	60	59	59
2	2018-10-22	51	68	57	66
3	2018-10-23	64	55	76	50
4	2018-10-24	66	54	64	54
5	2018-10-25	59	69	77	52
6	2018-10-26	54	56	59	69
7	2018-10-27	69	68	67	71
8	2018-10-28	57	54	62	63
9	2018-10-29	61	61	53	78
10	2018-10-30	51	73	53	77
11	2018-10-31	64	77	79	51
12	2018-11-01	75	71	57	66
13	2018-11-02	64	76	59	56
14	2018-11-03	65	66	55	52

In [20]:

```
df_list = pd.read_html(r"https://coinmarketcap.com/currencies/bitcoin/historical-data/"
)
len(df_list)
```

Out[20]:

```
3
```

In [21]:

```
df1 = df_list[0]
df1
```

Out[21]:

Date	Open*	High	Low	Close**	Volume	Market Cap

In [22]:

```
df2 = df_list[1]
df2
```

Out[22]:

Date

In [23]:

```
df3 = df_list[2]
df3
```

Out[23]:

	Date	Open*	High	Low	Close**	Volume	Market Cap
0	Mar 03, 2020	8865.39	8901.60	8704.99	8787.79	42386715821	160383579416
1	Mar 02, 2020	8563.26	8921.31	8532.63	8869.67	42857674409	161861167745
2	Mar 01, 2020	8599.76	8726.80	8471.21	8562.45	35349164300	156238987740
3	Feb 29, 2020	8671.21	8775.63	8599.51	8599.51	35792392544	156895988084
4	Feb 28, 2020	8788.73	8890.46	8492.93	8672.46	44605450443	158211707019
5	Feb 27, 2020	8825.09	8932.89	8577.20	8784.49	45470195695	160238496932
6	Feb 26, 2020	9338.29	9354.78	8704.43	8820.52	50420050762	160879489024
7	Feb 25, 2020	9651.31	9652.74	9305.02	9341.71	42515259129	170369581558
8	Feb 24, 2020	9921.58	9951.75	9537.04	9650.17	45080496648	175977808526
9	Feb 23, 2020	9663.32	9937.40	9657.79	9924.52	41185185761	180963233540
10	Feb 22, 2020	9687.71	9698.23	9600.73	9663.18	35838025154	176180696548
11	Feb 21, 2020	9611.78	9723.01	9589.74	9686.44	40930547513	176587087363
12	Feb 20, 2020	9629.33	9643.22	9507.90	9608.48	44925260237	175147142158
13	Feb 19, 2020	10143.80	10191.68	9611.22	9633.39	46992019710	175585931679
14	Feb 18, 2020	9691.23	10161.94	9632.38	10142.00	47271023953	184838512656
15	Feb 17, 2020	9936.56	9938.82	9507.64	9690.14	45998298413	176585280987
16	Feb 16, 2020	9889.18	10053.97	9722.39	9934.43	43374780305	181017665264
17	Feb 15, 2020	10313.86	10341.56	9874.43	9889.42	43865054831	180179996219
18	Feb 14, 2020	10211.55	10322.00	10125.53	10312.12	43338264162	187862645449
19	Feb 13, 2020	10323.96	10457.63	10116.16	10214.38	49356071373	186065003526
20	Feb 12, 2020	10202.39	10393.61	10202.39	10326.05	43444303830	188081204386
21	Feb 11, 2020	9855.89	10210.05	9729.33	10208.24	37648059389	185917114989
22	Feb 10, 2020	10115.56	10165.77	9784.56	9856.61	39386548075	179494809266
23	Feb 09, 2020	9863.89	10129.44	9850.39	10116.67	35807884663	184214765394
24	Feb 08, 2020	9793.07	9876.75	9678.91	9865.12	35172043762	179615828322
25	Feb 07, 2020	9726.00	9834.72	9726.00	9795.94	34522718159	178339437206
26	Feb 06, 2020	9617.82	9824.62	9539.82	9729.80	37628823716	177118274394
27	Feb 05, 2020	9183.42	9701.30	9163.70	9613.42	35222060874	174983423933
28	Feb 04, 2020	9292.84	9331.27	9112.81	9180.96	29893183716	167093636162

In [34]:

```
# exporting data
df3.to_excel(r"D:\New folder.xlsx")
```

data selection and filtering

In [29]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh.csv",index_col="Dates")
df.shape
```

Out[29]:

```
(20, 4)
```

In [30]:

```
df.head()
```

Out[30]:

Dates	Temperature	Humidity	Pressure	Air Quality
30-04-2018	218	182	4	2
01-05-2018	2592	182	3	2
02-05-2018	509	439	4	0
03-05-2018	2439	53	5	1
04-05-2018	824	444	5	0

In [31]:

```
df.tail()
```

Out[31]:

Dates	Temperature	Humidity	Pressure	Air Quality
15-05-2018	988	259	4	0
16-05-2018	1454	125	5	1
17-05-2018	2200	325	3	2
18-05-2018	1722	523	0	2
19-05-2018	766	535	3	2

In [32]:

```
df.head(3)
```

Out[32]:

Dates	Temperature	Humidity	Pressure	Air Quality
30-04-2018	218	182	4	2
01-05-2018	2592	182	3	2
02-05-2018	509	439	4	0

In [33]:

```
df.tail(4)
```

Out[33]:

Dates	Temperature	Humidity	Pressure	Air Quality
16-05-2018	1454	125	5	1
17-05-2018	2200	325	3	2
18-05-2018	1722	523	0	2
19-05-2018	766	535	3	2

In [35]:

```
df['Temperature'] # recommended
```

Out[35]:

```
Dates
30-04-2018     218
01-05-2018    2592
02-05-2018     509
03-05-2018    2439
04-05-2018     824
05-05-2018    1744
06-05-2018     786
07-05-2018    1326
08-05-2018    1804
09-05-2018     109
10-05-2018    2524
11-05-2018    2945
12-05-2018    2168
13-05-2018    1318
14-05-2018     788
15-05-2018     988
16-05-2018    1454
17-05-2018    2200
18-05-2018    1722
19-05-2018     766
Name: Temperature, dtype: int64
```

In [36]:

```
df.Temperature # will not work if column name has space or dot
```

Out[36]:

```
Dates
30-04-2018     218
01-05-2018    2592
02-05-2018     509
03-05-2018    2439
04-05-2018     824
05-05-2018    1744
06-05-2018     786
07-05-2018    1326
08-05-2018    1804
09-05-2018     109
10-05-2018    2524
11-05-2018    2945
12-05-2018    2168
13-05-2018    1318
14-05-2018     788
15-05-2018     988
16-05-2018    1454
17-05-2018    2200
18-05-2018    1722
19-05-2018     766
Name: Temperature, dtype: int64
```

In [37]:

```
df.Temperature.drop
```

Out[37]:

```
<bound method Series.drop of Dates
30-04-2018     218
01-05-2018    2592
02-05-2018     509
03-05-2018    2439
04-05-2018     824
05-05-2018    1744
06-05-2018     786
07-05-2018    1326
08-05-2018    1804
09-05-2018     109
10-05-2018    2524
11-05-2018    2945
12-05-2018    2168
13-05-2018    1318
14-05-2018     788
15-05-2018     988
16-05-2018    1454
17-05-2018    2200
18-05-2018    1722
19-05-2018     766
Name: Temperature, dtype: int64>
```

In [38]:

```
import numpy as np
np.array(df.Temperature)
```

Out[38]:

```
array([ 218, 2592,  509, 2439,  824, 1744,  786, 1326, 1804,  109, 2524,
       2945, 2168, 1318,  788,  988, 1454, 2200, 1722,  766], dtype=int64)
```

In [40]:

```
df.Temperature.reset_index(drop=True)
```

Out[40]:

```
0      218
1     2592
2      509
3     2439
4      824
5     1744
6      786
7     1326
8     1804
9      109
10    2524
11    2945
12    2168
13    1318
14     788
15     988
16    1454
17    2200
18    1722
19     766
Name: Temperature, dtype: int64
```

In [48]:

```
df[["Temperature", "Pressure"]].reset_index(drop=True)
```

Out[48]:

	Temperature	Pressure
0	218	4
1	2592	3
2	509	4
3	2439	5
4	824	5
5	1744	5
6	786	5
7	1326	0
8	1804	4
9	109	0
10	2524	1
11	2945	1
12	2168	1
13	1318	3
14	788	3
15	988	4
16	1454	5
17	2200	3
18	1722	0
19	766	3

In [54]:

```
df.reset_index(drop=True, inplace=True)
```

In [55]:

```
df
```

Out[55]:

	Temperature	Humidity	Pressure	Air Quality
0	218	182	4	2
1	2592	182	3	2
2	509	439	4	0
3	2439	53	5	1
4	824	444	5	0
5	1744	443	5	1
6	786	226	5	1
7	1326	309	0	1
8	1804	188	4	2
9	109	420	0	1
10	2524	433	1	0
11	2945	149	1	0
12	2168	531	1	1
13	1318	360	3	2
14	788	435	3	2
15	988	259	4	0
16	1454	125	5	1
17	2200	325	3	2
18	1722	523	0	2
19	766	535	3	2

In [56]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh.csv",index_col="Dates")
df.shape
```

Out[56]:

```
(20, 4)
```

In [57]:

```
df["Temperature"]
```

Out[57]:

```
Dates
30-04-2018     218
01-05-2018    2592
02-05-2018     509
03-05-2018    2439
04-05-2018     824
05-05-2018    1744
06-05-2018     786
07-05-2018    1326
08-05-2018    1804
09-05-2018     109
10-05-2018    2524
11-05-2018    2945
12-05-2018    2168
13-05-2018    1318
14-05-2018     788
15-05-2018     988
16-05-2018    1454
17-05-2018    2200
18-05-2018    1722
19-05-2018     766
Name: Temperature, dtype: int64
```

In [58]:

```
df[["Temperature", "Pressure"]]
```

Out[58]:

Dates	Temperature	Pressure
30-04-2018	218	4
01-05-2018	2592	3
02-05-2018	509	4
03-05-2018	2439	5
04-05-2018	824	5
05-05-2018	1744	5
06-05-2018	786	5
07-05-2018	1326	0
08-05-2018	1804	4
09-05-2018	109	0
10-05-2018	2524	1
11-05-2018	2945	1
12-05-2018	2168	1
13-05-2018	1318	3
14-05-2018	788	3
15-05-2018	988	4
16-05-2018	1454	5
17-05-2018	2200	3
18-05-2018	1722	0
19-05-2018	766	3

In [59]:

```
type(df["Temperature"])
```

Out[59]:

```
pandas.core.series.Series
```

In [60]:

```
type(df[["Temperature", "Pressure"]])
```

Out[60]:

```
pandas.core.frame.DataFrame
```

In [61]:

```
df
```

Out[61]:

Dates	Temperature	Humidity	Pressure	Air Quality
30-04-2018	218	182	4	2
01-05-2018	2592	182	3	2
02-05-2018	509	439	4	0
03-05-2018	2439	53	5	1
04-05-2018	824	444	5	0
05-05-2018	1744	443	5	1
06-05-2018	786	226	5	1
07-05-2018	1326	309	0	1
08-05-2018	1804	188	4	2
09-05-2018	109	420	0	1
10-05-2018	2524	433	1	0
11-05-2018	2945	149	1	0
12-05-2018	2168	531	1	1
13-05-2018	1318	360	3	2
14-05-2018	788	435	3	2
15-05-2018	988	259	4	0
16-05-2018	1454	125	5	1
17-05-2018	2200	325	3	2
18-05-2018	1722	523	0	2
19-05-2018	766	535	3	2

In [63]:

```
df["05-05-2018" : "10-05-2018"]
```

Out[63]:

Dates	Temperature	Humidity	Pressure	Air Quality
05-05-2018	1744	443	5	1
06-05-2018	786	226	5	1
07-05-2018	1326	309	0	1
08-05-2018	1804	188	4	2
09-05-2018	109	420	0	1
10-05-2018	2524	433	1	0

In [64]:

```
df["05-05-2018" : "10-05-2018"][["Temperature", "Pressure"]]
```

Out[64]:

Dates	Temperature	Pressure
05-05-2018	1744	5
06-05-2018	786	5
07-05-2018	1326	0
08-05-2018	1804	4
09-05-2018	109	0
10-05-2018	2524	1

In [67]:

```
df.iloc[5:10, 1:3]
```

Out[67]:

Dates	Humidity	Pressure
05-05-2018	443	5
06-05-2018	226	5
07-05-2018	309	0
08-05-2018	188	4
09-05-2018	420	0

Filtering

In [68]:

```
df[df.Temperature>2500]
```

Out[68]:

Dates	Temperature	Humidity	Pressure	Air Quality
01-05-2018	2592	182	3	2
10-05-2018	2524	433	1	0
11-05-2018	2945	149	1	0

In [69]:

```
df[df.Temperature>2500][df.Pressure>2]
```

```
C:\Users\gorav\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

Out[69]:

Dates	Temperature	Humidity	Pressure	Air Quality
01-05-2018	2592	182	3	2

In [70]:

```
df[(df.Temperature>2500) & (df.Pressure>2)]
```

Out[70]:

Dates	Temperature	Humidity	Pressure	Air Quality
01-05-2018	2592	182	3	2

In [71]:

```
df[(df.Temperature>2500) | (df.Pressure>2)]
```

Out[71]:

Dates	Temperature	Humidity	Pressure	Air Quality
30-04-2018	218	182	4	2
01-05-2018	2592	182	3	2
02-05-2018	509	439	4	0
03-05-2018	2439	53	5	1
04-05-2018	824	444	5	0
05-05-2018	1744	443	5	1
06-05-2018	786	226	5	1
08-05-2018	1804	188	4	2
10-05-2018	2524	433	1	0
11-05-2018	2945	149	1	0
13-05-2018	1318	360	3	2
14-05-2018	788	435	3	2
15-05-2018	988	259	4	0
16-05-2018	1454	125	5	1
17-05-2018	2200	325	3	2
19-05-2018	766	535	3	2

In [72]:

```
df[(df.Temperature>2500) | (df.Pressure>2)][["Temperature", "Pressure"]]
```

Out[72]:

Dates	Temperature	Pressure
30-04-2018	218	4
01-05-2018	2592	3
02-05-2018	509	4
03-05-2018	2439	5
04-05-2018	824	5
05-05-2018	1744	5
06-05-2018	786	5
08-05-2018	1804	4
10-05-2018	2524	1
11-05-2018	2945	1
13-05-2018	1318	3
14-05-2018	788	3
15-05-2018	988	4
16-05-2018	1454	5
17-05-2018	2200	3
19-05-2018	766	3

statistical analysis

In [73]:

```
df.describe()
```

Out[73]:

	Temperature	Humidity	Pressure	Air Quality
count	20.000000	20.000000	20.000000	20.000000
mean	1461.200000	328.050000	2.950000	1.150000
std	834.100688	148.623323	1.820208	0.812728
min	109.000000	53.000000	0.000000	0.000000
25%	787.500000	186.500000	1.000000	0.750000
50%	1390.000000	342.500000	3.000000	1.000000
75%	2176.000000	440.000000	4.250000	2.000000
max	2945.000000	535.000000	5.000000	2.000000

In [74]:

```
df.mean()
```

Out[74]:

```
Temperature    1461.20
Humidity        328.05
Pressure          2.95
Air Quality       1.15
dtype: float64
```

In [77]:

```
df.Temperature.mean()
```

Out[77]:

```
1461.2
```

In [78]:

```
df.Temperature.median()
```

Out[78]:

```
1390.0
```

In [79]:

```
df.Temperature.mode()
```

Out[79]:

```
0      109
1      218
2      509
3      766
4      786
5      788
6      824
7      988
8     1318
9     1326
10    1454
11    1722
12    1744
13    1804
14    2168
15    2200
16    2439
17    2524
18    2592
19    2945
dtype: int64
```

In [80]:

```
df.Temperature.var() #variance
```

Out[80]:

```
695723.9578947368
```

In [81]:

```
df.Temperature.std()
```

Out[81]:

```
834.1006881035028
```

In [82]:

```
df.Temperature.skew()
```

Out[82]:

```
0.11175266975855103
```

In [83]:

```
df.Temperature.kurt()
```

Out[83]:

```
-1.0525996865252307
```

Data Cleaning

In [84]:

```
# 1. check for unrealistic values - df.describe()
# 2. check for duplicates - df.duplicated().sum()
# 3. check for missing values - df.isnull().sum()
```

In [104]:

```
#importing data with pandas
df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh_missing.csv")
df.shape
```

Out[104]:

```
(20, 5)
```

In [105]:

```
# check for duplicates
df.duplicated().sum()
```

Out[105]:

```
0
```

In [106]:

```
# to drop duplicated rows
df.drop_duplicates(inplace=True)
df
```

Out[106]:

	Dates	Temperature	Humidity	Pressure	Air Quality
0	30-04-2018	218	182	4	2
1	01-05-2018	?	182	3	2
2	02-05-2018	.	439	NaN	0
3	03-05-2018	2439	53	5	1
4	04-05-2018	824	444	5	NaN
5	05-05-2018	1744	.	5	1
6	06-05-2018	786	.	5	1
7	07-05-2018	1326	309	.	1
8	08-05-2018	1804	188	.	2
9	09-05-2018	?	420	0	1
10	10-05-2018	2524	433	1	0
11	11-05-2018	2945	149	.	0
12	12-05-2018	.	531	1	1
13	13-05-2018	1318	360	3	2
14	14-05-2018	.	435	NaN	2
15	15-05-2018	.	259	4	0
16	16-05-2018	.	.	.	.
17	17-05-2018	2200	325	3	2
18	18-05-2018	1722	523	.	2
19	19-05-2018	766	535	3	2

In [107]:

```
# check for missing values
df.isnull().sum()
```

Out[107]:

```
Dates          0
Temperature    0
Humidity       0
Pressure       2
Air Quality    1
dtype: int64
```

In [108]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20 entries, 0 to 19
Data columns (total 5 columns):
Dates          20 non-null object
Temperature    20 non-null object
Humidity       20 non-null object
Pressure       18 non-null object
Air Quality    19 non-null object
dtypes: object(5)
memory usage: 960.0+ bytes
```

In [109]:

```
#importing data with pandas

df = pd.read_csv(r"C:\Users\gorav\Desktop\data\datawh_missing.csv",na_values=[".", "?"
])
df.shape
```

Out[109]:

```
(20, 5)
```

In [110]:

```
df.isnull().sum()
```

Out[110]:

```
Dates          0
Temperature    7
Humidity       3
Pressure       7
Air Quality    2
dtype: int64
```

In [111]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
Dates          20 non-null object
Temperature    13 non-null float64
Humidity       17 non-null float64
Pressure       13 non-null float64
Air Quality    18 non-null float64
dtypes: float64(4), object(1)
memory usage: 928.0+ bytes
```

In [112]:

```
df.dropna(thresh=3, inplace=True)
df
```

Out[112]:

	Dates	Temperature	Humidity	Pressure	Air Quality
0	30-04-2018	218.0	182.0	4.0	2.0
1	01-05-2018	NaN	182.0	3.0	2.0
2	02-05-2018	NaN	439.0	NaN	0.0
3	03-05-2018	2439.0	53.0	5.0	1.0
4	04-05-2018	824.0	444.0	5.0	NaN
5	05-05-2018	1744.0	NaN	5.0	1.0
6	06-05-2018	786.0	NaN	5.0	1.0
7	07-05-2018	1326.0	309.0	NaN	1.0
8	08-05-2018	1804.0	188.0	NaN	2.0
9	09-05-2018	NaN	420.0	0.0	1.0
10	10-05-2018	2524.0	433.0	1.0	0.0
11	11-05-2018	2945.0	149.0	NaN	0.0
12	12-05-2018	NaN	531.0	1.0	1.0
13	13-05-2018	1318.0	360.0	3.0	2.0
14	14-05-2018	NaN	435.0	NaN	2.0
15	15-05-2018	NaN	259.0	4.0	0.0
17	17-05-2018	2200.0	325.0	3.0	2.0
18	18-05-2018	1722.0	523.0	NaN	2.0
19	19-05-2018	766.0	535.0	3.0	2.0

In [113]:

```
df.skew()
```

Out[113]:

```
Temperature    0.019636
Humidity      -0.367387
Pressure      -0.670499
Air Quality   -0.451480
dtype: float64
```

In [101]:

```
df
```

Out[101]:

	Dates	Temperature	Humidity	Pressure	Air Quality
0	30-04-2018	218	182	4	2
1	01-05-2018	NaN	182	3	2
2	02-05-2018	.	439	NaN	0
3	03-05-2018	2439	53	5	1
4	04-05-2018	824	444	5	NaN
5	05-05-2018	1744	.	5	1
6	06-05-2018	786	.	5	1
7	07-05-2018	1326	309	.	1
8	08-05-2018	1804	188	.	2
9	09-05-2018	NaN	420	0	1
10	10-05-2018	2524	433	1	0
11	11-05-2018	2945	149	.	0
12	12-05-2018	.	531	1	1
13	13-05-2018	1318	360	3	2
14	14-05-2018	.	435	NaN	2
15	15-05-2018	.	259	4	0
16	16-05-2018	.	.	.	.
17	17-05-2018	2200	325	3	2
18	18-05-2018	1722	523	.	2
19	19-05-2018	766	535	3	2

In [114]:

```
df.Temperature.fillna(df.Temperature.mean(), inplace=True)
df.isnull().sum()
```

Out[114]:

```
Dates          0
Temperature    0
Humidity       2
Pressure       6
Air Quality    1
dtype: int64
```

In [115]:

```
df.fillna(df.median(), inplace=True)
df.isnull().sum()
```

Out[115]:

```
Dates          0
Temperature    0
Humidity       0
Pressure       0
Air Quality    0
dtype: int64
```

Data Aggergation

In [3]:

```
# Load the file regiment.csv
import pandas as pd
df = pd.read_csv(r"C:\Users\gorav\Desktop\data\regiment.csv")
df.shape
```

Out[3]:

```
(12, 6)
```

In [4]:

```
df
```

Out[4]:

	index	regiment	company	name	preTestScore	postTestScore
0	0	Nighthawks	1st	Miller	4	25
1	1	Nighthawks	1st	Jacobson	24	94
2	2	Nighthawks	2nd	Ali	31	57
3	3	Nighthawks	2nd	Milner	2	62
4	4	Dragoons	1st	Cooze	3	70
5	5	Dragoons	1st	Jacon	4	25
6	6	Dragoons	2nd	Ryaner	24	94
7	7	Dragoons	2nd	Sone	31	57
8	8	Scouts	1st	Sloan	2	62
9	9	Scouts	1st	Piger	3	70
10	10	Scouts	2nd	Riani	2	62
11	11	Scouts	2nd	Ali	3	70

In [5]:

```
# 1. which is best regiment after training(regiment, postTestScore)
# 2. which company of which regiment is best after training
# 3. create a column which shows improvement made by every soldier during training
# 4. which company of which regiment has made maximum improvement during training
```

In [6]:

```
df.head()
```

Out[6]:

	index	regiment	company	name	preTestScore	postTestScore
0	0	Nighthawks	1st	Miller	4	25
1	1	Nighthawks	1st	Jacobson	24	94
2	2	Nighthawks	2nd	Ali	31	57
3	3	Nighthawks	2nd	Milner	2	62
4	4	Dragoons	1st	Cooze	3	70

In [8]:

```
df.postTestScore.mean()
```

Out[8]:

```
62.333333333333336
```

In [9]:

```
df.groupby(['regiment']).postTestScore.mean()
```

Out[9]:

```
regiment
Dragoons      61.5
Nighthawks    59.5
Scouts        66.0
Name: postTestScore, dtype: float64
```

In [11]:

```
df.groupby(['regiment', 'company']).postTestScore.mean()
```

Out[11]:

```
regiment    company
Dragoons    1st        47.5
            2nd        75.5
Nighthawks  1st        59.5
            2nd        59.5
Scouts      1st        66.0
            2nd        66.0
Name: postTestScore, dtype: float64
```

In [12]:

```
df['improvement'] = df.postTestScore - df.preTestScore
df
```

Out[12]:

	index	regiment	company	name	preTestScore	postTestScore	improvement
0	0	Nighthawks	1st	Miller	4	25	21
1	1	Nighthawks	1st	Jacobson	24	94	70
2	2	Nighthawks	2nd	Ali	31	57	26
3	3	Nighthawks	2nd	Milner	2	62	60
4	4	Dragoons	1st	Cooze	3	70	67
5	5	Dragoons	1st	Jacon	4	25	21
6	6	Dragoons	2nd	Ryaner	24	94	70
7	7	Dragoons	2nd	Sone	31	57	26
8	8	Scouts	1st	Sloan	2	62	60
9	9	Scouts	1st	Piger	3	70	67
10	10	Scouts	2nd	Riani	2	62	60
11	11	Scouts	2nd	Ali	3	70	67

In [13]:

```
df.groupby(["regiment", "company"]).improvement.mean()
```

Out[13]:

```
regiment    company
Dragoons    1st        44.0
            2nd        48.0
Nighthawks  1st        45.5
            2nd        43.0
Scouts      1st        63.5
            2nd        63.5
Name: improvement, dtype: float64
```

In [ ]:

```

```