

Deena

20104016

libraries

```
In [1]: import numpy as np
import pandas as pd
```

importing data set

```
In [26]: df=pd.read_csv("6_Salesworkload1.csv")
df
```

Out[26]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursI
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	
...	
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	

mean median mode()

```
In [3]: df.mean()
```

```
Out[3]: Time index      5.000000e+00
StoreID      6.199522e+04
Dept_ID      9.470588e+00
HoursLease    2.203608e+01
Sales units   1.076471e+06
Turnover      3.721393e+06
Customer      NaN
dtype: float64
```

```
In [4]: df.median()
```

```
Out[4]: Time index      5.0  
StoreID      75400.5  
Dept_ID       9.0  
HoursLease     0.0  
Sales units  293230.0  
Turnover     931957.5  
Customer      NaN  
dtype: float64
```

In [5]: `df.mode()`

Out[5]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLeas
0	01.2017	1.0	France	12227.0	Aalborg (I)	1.0	Admin	47.205	0
1	02.2017	2.0	Germany	15552.0	Aalborg (II)	2.0	Checkout	NaN	NaN
2	03.2017	3.0	United Kingdom	16927.0	Amsterdam	3.0	Clothing	NaN	NaN
3	04.2017	4.0	NaN	17647.0	Antwerp	4.0	Customer Services	NaN	NaN
4	05.2017	5.0	NaN	18808.0	Barcelona (I)	5.0	Delivery	NaN	NaN
5	06.2017	6.0	NaN	19000.0	Barcelona (II)	6.0	Dry	NaN	NaN
6	10.2016	7.0	NaN	19340.0	Berlin (I)	7.0	Fish	NaN	NaN
7	11.2016	8.0	NaN	19769.0	Berlin (II)	8.0	Food	NaN	NaN
8	12.2016	9.0	NaN	20166.0	Bilbao	9.0	Frozen	NaN	NaN
9	NaN	NaN	NaN	20891.0	Birmingham	11.0	Fruits & Vegetables	NaN	NaN
10	NaN	NaN	NaN	22117.0	Bologna	12.0	Hardware	NaN	NaN
11	NaN	NaN	NaN	23623.0	Bordeaux	13.0	Household	NaN	NaN
12	NaN	NaN	NaN	29650.0	Brno	14.0	Meat	NaN	NaN
13	NaN	NaN	NaN	32949.0	Brussels (I)	15.0	Non Food	NaN	NaN
14	NaN	NaN	NaN	34378.0	Brussels (II)	16.0	all	NaN	NaN
15	NaN	NaN	NaN	38560.0	Cologne	17.0	other	NaN	NaN
16	NaN	NaN	NaN	38976.0	Copenhagen (I)	18.0	others	NaN	NaN
17	NaN	NaN	NaN	42367.0	Copenhagen (II)	NaN	NaN	NaN	NaN
18	NaN	NaN	NaN	45583.0	Den Haag	NaN	NaN	NaN	NaN
19	NaN	NaN	NaN	63354.0	Frankfurt	NaN	NaN	NaN	NaN
20	NaN	NaN	NaN	64983.0	Göteborg	NaN	NaN	NaN	NaN
21	NaN	NaN	NaN	71991.0	Groningen	NaN	NaN	NaN	NaN
22	NaN	NaN	NaN	73422.0	Hamburg	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	73762.0	Krakow	NaN	NaN	NaN	NaN
24	NaN	NaN	NaN	73949.0	Leicester	NaN	NaN	NaN	NaN
25	NaN	NaN	NaN	76852.0	Liverpool	NaN	NaN	NaN	NaN
26	NaN	NaN	NaN	77348.0	London (I)	NaN	NaN	NaN	NaN
27	NaN	NaN	NaN	78325.0	London (II)	NaN	NaN	NaN	NaN
28	NaN	NaN	NaN	78450.0	Lyon	NaN	NaN	NaN	NaN
29	NaN	NaN	NaN	79785.0	Madrid (I)	NaN	NaN	NaN	NaN
30	NaN	NaN	NaN	81473.0	Madrid (II)	NaN	NaN	NaN	NaN
31	NaN	NaN	NaN	83160.0	Malmö	NaN	NaN	NaN	NaN

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLeas
32	NaN	NaN	NaN	85124.0	Manchester	NaN	NaN	NaN	NaN
33	NaN	NaN	NaN	85321.0	Marseille	NaN	NaN	NaN	NaN
34	NaN	NaN	NaN	85696.0	Milano	NaN	NaN	NaN	NaN
35	NaN	NaN	NaN	86089.0	Munich	NaN	NaN	NaN	NaN
36	NaN	NaN	NaN	86208.0	Nantes	NaN	NaN	NaN	NaN
37	NaN	NaN	NaN	87703.0	Napoli	NaN	NaN	NaN	NaN
38	NaN	NaN	NaN	88253.0	Ostrava	NaN	NaN	NaN	NaN
39	NaN	NaN	NaN	88750.0	Paris (I)	NaN	NaN	NaN	NaN
40	NaN	NaN	NaN	88994.0	Paris (II)	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	90992.0	Poznan	NaN	NaN	NaN	NaN
42	NaN	NaN	NaN	91973.0	Prague (I)	NaN	NaN	NaN	NaN
43	NaN	NaN	NaN	93033.0	Prague (II)	NaN	NaN	NaN	NaN
44	NaN	NaN	NaN	94153.0	Rome (I)	NaN	NaN	NaN	NaN
45	NaN	NaN	NaN	94882.0	Rome (II)	NaN	NaN	NaN	NaN
46	NaN	NaN	NaN	95434.0	Rotterdam	NaN	NaN	NaN	NaN
47	NaN	NaN	NaN	96493.0	Stockholm	NaN	NaN	NaN	NaN
48	NaN	NaN	NaN	96857.0	Warsaw (I)	NaN	NaN	NaN	NaN
49	NaN	NaN	NaN	98422.0	Warsaw (II)	NaN	NaN	NaN	NaN

describe ()

In [6]: df.describe()

Out[6]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Custom
count	7650.000000	7650.000000	7650.000000	7650.000000	7.650000e+03	7.650000e+03	(
mean	5.000000	61995.220000	9.470588	22.036078	1.076471e+06	3.721393e+06	N
std	2.582158	29924.581631	5.337429	133.299513	1.728113e+06	6.003380e+06	N
min	1.000000	12227.000000	1.000000	0.000000	0.000000e+00	0.000000e+00	N
25%	3.000000	29650.000000	5.000000	0.000000	5.457125e+04	2.726798e+05	N
50%	5.000000	75400.500000	9.000000	0.000000	2.932300e+05	9.319575e+05	N
75%	7.000000	87703.000000	14.000000	0.000000	9.175075e+05	3.264432e+06	N
max	9.000000	98422.000000	18.000000	3984.000000	1.124296e+07	4.271739e+07	N

sum()

```
In [7]: df.sum()
```

```
Out[7]: MonthYear      10.201610.201610.201610.201610.201610.201610.2...
Time index              38250.0
StoreID                 474263433.0
Dept_ID                 72450.0
HoursLease              168576.0
Sales units             8235000965.0
Turnover                28468656015.0
Customer                0.0
dtype: object
```

cumsum ()

```
In [23]: a=df.head(50)
a.cumsum()
```

14	10.201610.201610.201610.201610.201610.2...	15.0	United	1323795.0	Loi ▲
			KingdomUnited		(I)Loi
			KingdomUnited		(I)Loi
			KingdomUnit...		(I)Loi
15	10.201610.201610.201610.201610.201610.2...	16.0	United	1412048.0	(I)Lond
			KingdomUnited		(I)Loi
			KingdomUnited		(I)Loi
			KingdomUnit...		(I)Loi
16	10.201610.201610.201610.201610.201610.2...	17.0	United	1500301.0	(I)Lond ▼
			KingdomUnited		(I)Loi
			KingdomUnited		(I)Loi
			KingdomUnit...		(I)Loi

min() and min()

```
In [14]: df.min()
```

```
Out[14]: MonthYear      - - - -  
Time index           1.0  
StoreID             12227.0  
Dept_ID              1.0  
HoursLease            0.0  
Sales units           0.0  
Turnover              0.0  
Customer             NaN  
dtype: object
```

```
In [15]: df.max()
```

```
Out[15]: MonthYear           12.2016  
Time index                9.0  
StoreID                 98422.0  
Dept_ID                 18.0  
HoursLease              3984.0  
Sales units          11242955.0  
Turnover             42717390.0  
Customer                NaN  
dtype: object
```

count()

```
In [16]: df.count()
```

```
Out[16]: MonthYear           7658  
Time index           7650  
Country              7650  
StoreID              7650  
City                 7650  
Dept_ID              7650  
Dept. Name           7650  
HoursOwn             7650  
HoursLease           7650  
Sales units          7650  
Turnover             7650  
Customer              0  
Area (m2)            7650  
Opening hours        7650  
dtype: int64
```

Covariance

```
In [17]: from numpy import cov
```

```
In [18]: cov(df['StoreID'],df['Sales units'])
```

```
Out[18]: array([[nan, nan],  
               [nan, nan]])
```

pearsonr and spearmanr

```
In [19]: from scipy.stats import pearsonr  
         from scipy.stats import spearmanr
```

```
In [25]: pearsonr(a['StoreID'],a['Sales units'])
```

```
Out[25]: (-0.10382256510563574, 0.47305915125322995)
```

```
In [22]: spearmanr(df['StoreID'],df['Sales units'])
```

```
Out[22]: SpearmanrResult(correlation=nan, pvalue=nan)
```