Different ways of creating dataframes

In [12]:
```
#Method1: Using List
import pandas as pd
data = [['srishti', 1, 'A'],['Goutham',2, 'B'],['Sanskaar',3,'C']]
print(type(data))
print(data)
df=pd.DataFrame(data)
df
```

```
<class 'list'>
[['srishti', 1, 'A'], ['Goutham', 2, 'B'], ['Sanskaar', 3, 'C']]
```

Out[12]:

	0	1	2
0	srishti	1	A
1	Goutham	2	B
2	Sanskaar	3	C

In [13]:
```
#Method2 Dictionary
import pandas as pd
data={'Name':['Srishti', 'Goutham', 'sanskar'], 'Regnum':[1,2,3]}
print(type(data))
df=pd.DataFrame(data)
df
```

```
<class 'dict'>
```

Out[13]:

	Name	Regnum
0	Srishti	1
1	Goutham	2
2	sanskar	3

In [34]:
```
#Method3 Using Numpy array
import numpy as np
import pandas as pd
data=np.random.rand(2,2)
print(type(data))
df=pd.DataFrame(data)
df=pd.DataFrame(data, columns=['c1', 'c2'])
print(df)
###Instead of typing 5 lines of code you can just write a single line code as given below
pd.DataFrame(np.random.rand(2,2),columns=['c1','c2'], index=['r1','r2'])
```

```
<class 'numpy.ndarray'>
         c1        c2
0  0.057217  0.279724
1  0.895036  0.189275
```

Out[34]:

	c1	c2
r1	0.758522	0.154609
r2	0.611082	0.370033

In [15]:
```
#Method4 List of dictionaries
data =[{'Regnum':1, 'Rank':1}, {'Regnum':2, 'Rank':2}]
print(type(data))
df=pd.DataFrame(data)
df
```

```
<class 'list'>
```

Out[15]:

	Rank	Regnum
0	1	1
1	2	2

In [23]:
```
#Method5 Zipped Tuples
Name = ['srishti', 'Hrithik']
Marks = [89,99]
x=(Name,Marks)
print("Original datatype of variable x before zipping ", type(x))
x = (zip(Name, Marks)) #Zipped data
print("After zipping", type(x))
data = list(x)
df=pd.DataFrame(data)
df
```

```
Original datatype of variable x before zipping  <class 'tuple'>
After zipping <class 'zip'>
```

Out[23]:

	0	1
0	srishti	89
1	Hrithik	99

In [24]:
```
#Method 6 using data series
data ={'col1': pd.Series([10,100,1000], index=['r1','r2','r3']), 'col2': pd.Series([1,2,3], index=['r1','r2','r3'])}
df=pd.DataFrame(data)
df
```

Out[24]:

	col1	col2
r1	10	1
r2	100	2
r3	1000	3

In [25]:
```
#Method7 Using files
data = pd.read_csv("WBCDdata.csv")
df=pd.DataFrame(data)
df
```

Out[25]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	co
0	842302	M	17.990	10.38	122.80	1001.0	0.11840	0.27760	
1	842517	M	20.570	17.77	132.90	1326.0	0.08474	0.07864	
2	84300903	M	19.690	21.25	130.00	1203.0	0.10960	0.15990	
3	84348301	M	11.420	20.38	77.58	386.1	0.14250	0.28390	
4	84358402	M	20.290	14.34	135.10	1297.0	0.10030	0.13280	
5	843786	M	12.450	15.70	82.57	477.1	0.12780	0.17000	
6	844359	M	18.250	19.98	119.60	1040.0	0.09463	0.10900	
7	84458202	M	13.710	20.83	90.20	577.9	0.11890	0.16450	
8	844981	M	13.000	21.82	87.50	519.8	0.12730	0.19320	
9	84501001	M	12.460	24.04	83.97	475.9	0.11860	0.23960	
10	845636	M	16.020	23.24	102.70	797.8	0.08206	0.06669	
11	84610002	M	15.780	17.89	103.60	781.0	0.09710	0.12920	
12	846226	M	19.170	24.80	132.40	1123.0	0.09740	0.24580	
13	846381	M	15.850	23.95	103.70	782.7	0.08401	0.10020	
14	84667401	M	13.730	22.61	93.60	578.3	0.11310	0.22930	
15	84799002	M	14.540	27.54	96.73	658.8	0.11390	0.15950	
16	848406	M	14.680	20.13	94.74	684.5	0.09867	0.07200	
17	84862001	M	16.130	20.68	108.10	798.8	0.11700	0.20220	
18	849014	M	19.810	22.15	130.00	1260.0	0.09831	0.10270	

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	co
19	8510426	B	13.540	14.36	87.46	566.3	0.09779	0.08129	
20	8510653	B	13.080	15.71	85.63	520.0	0.10750	0.12700	
21	8510824	B	9.504	12.44	60.34	273.9	0.10240	0.06492	
22	8511133	M	15.340	14.26	102.50	704.4	0.10730	0.21350	
23	851509	M	21.160	23.04	137.20	1404.0	0.09428	0.10220	
24	852552	M	16.650	21.38	110.00	904.6	0.11210	0.14570	
25	852631	M	17.140	16.40	116.00	912.7	0.11860	0.22760	
26	852763	M	14.580	21.53	97.41	644.8	0.10540	0.18680	
27	852781	M	18.610	20.25	122.10	1094.0	0.09440	0.10660	
28	852973	M	15.300	25.27	102.40	732.4	0.10820	0.16970	
29	853201	M	17.570	15.05	115.00	955.1	0.09847	0.11570	
...	...	...	...	...	...	...	...	...	
539	921362	B	7.691	25.44	48.34	170.4	0.08668	0.11990	
540	921385	B	11.540	14.44	74.65	402.9	0.09984	0.11200	
541	921386	B	14.470	24.99	95.81	656.4	0.08837	0.12300	
542	921644	B	14.740	25.42	94.70	668.6	0.08275	0.07214	
543	922296	B	13.210	28.06	84.88	538.4	0.08671	0.06877	
544	922297	B	13.870	20.70	89.77	584.8	0.09578	0.10180	
545	922576	B	13.620	23.23	87.19	573.2	0.09246	0.06747	
546	922577	B	10.320	16.35	65.31	324.9	0.09434	0.04994	
547	922840	B	10.260	16.58	65.85	320.8	0.08877	0.08066	
548	923169	B	9.683	19.34	61.05	285.7	0.08491	0.05030	
549	923465	B	10.820	24.21	68.89	361.6	0.08192	0.06602	

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	co
550	923748	B	10.860	21.48	68.51	360.5	0.07431	0.04227	
551	923780	B	11.130	22.44	71.49	378.4	0.09566	0.08194	
552	924084	B	12.770	29.43	81.35	507.9	0.08276	0.04234	
553	924342	B	9.333	21.94	59.01	264.0	0.09240	0.05605	
554	924632	B	12.880	28.92	82.50	514.3	0.08123	0.05824	
555	924934	B	10.290	27.61	65.67	321.4	0.09030	0.07658	
556	924964	B	10.160	19.59	64.73	311.7	0.10030	0.07504	
557	925236	B	9.423	27.88	59.26	271.3	0.08123	0.04971	
558	925277	B	14.590	22.68	96.39	657.1	0.08473	0.13300	
559	925291	B	11.510	23.93	74.52	403.5	0.09261	0.10210	
560	925292	B	14.050	27.15	91.38	600.4	0.09929	0.11260	
561	925311	B	11.200	29.37	70.67	386.0	0.07449	0.03558	
562	925622	M	15.220	30.62	103.40	716.9	0.10480	0.20870	
563	926125	M	20.920	25.09	143.00	1347.0	0.10990	0.22360	
564	926424	M	21.560	22.39	142.00	1479.0	0.11100	0.11590	
565	926682	M	20.130	28.25	131.20	1261.0	0.09780	0.10340	
566	926954	M	16.600	28.08	108.30	858.1	0.08455	0.10230	
567	927241	M	20.600	29.33	140.10	1265.0	0.11780	0.27700	
568	92751	B	7.760	24.54	47.92	181.0	0.05263	0.04362	

569 rows × 33 columns

In [26]:

```
#### IMPORTANT

'''Perform the basic statistical analysis  on the given dataframe to identify the data patterns.
 Also provide a brief inference to each of your observation.'''

#Analyzing Sales of local '99Variety Dosa stall@Roadside' and 'pizzahut@Mall'
#Note:Day starts from Monday
import pandas as pd
import numpy as np
XYZ_mall= {'Days1':[1,2,3,4,5,6,7], "Sales1":[1000, 1500 ,3000,1000,4000,8000, 9000]}
XYZ_road= {'Days2':[1,2,3,4,5,6,7], "Sales2":[10000, 4500 ,6000,5500,4000,1000, 1000]}
df1=pd.DataFrame(XYZ_mall)
df2=pd.DataFrame(XYZ_road)
df1
df2
print("Pizzahut \n", df1)
print("99VarietyDosa \n", df2)
```

```
Pizzahut
    Days1  Sales1
0      1    1000
1      2    1500
2      3    3000
3      4    1000
4      5    4000
5      6    8000
6      7    9000
99VarietyDosa
    Days2  Sales2
0      1   10000
1      2    4500
2      3    6000
3      4    5500
4      5    4000
5      6    1000
6      7    1000
```