In [2]:
```
import pandas as pd
df=pd.read_csv("F:\\New folder\\ML\\CSV files\\placement.csv")
```

In [3]:
```
df
```

Out[3]:
```
     cgpa  resume_score  placed
 0   8.14          6.52       1
 1   6.17          5.17       0
 2   8.27          8.86       1
 3   6.88          7.27       1
 4   7.52          7.30       1
...   ...           ...     ...
95   6.33          6.38       0
96   8.23          7.76       1
97   6.65          7.78       0
98   8.14          5.63       1
99   6.09          6.61       0

100 rows × 3 columns
```

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

Out[4]:
```
    cgpa  resume_score  placed
0   8.14          6.52       1
1   6.17          5.17       0
2   8.27          8.86       1
3   6.88          7.27       1
4   7.52          7.30       1
```

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

Out[5]:
```
     cgpa  resume_score  placed
95   6.33          6.38       0
96   8.23          7.76       1
97   6.65          7.78       0
98   8.14          5.63       1
99   6.09          6.61       0
```

In [6]:

```
df.describe()
```

Out[6]:

	cgpa	resume_score	placed
count	100.0000	100.000000	100.000000
mean	6.9422	6.930500	0.500000
std	1.1192	0.979608	0.502519
min	5.2700	4.950000	0.000000
25%	5.9800	6.190000	0.000000
50%	6.6200	7.055000	0.500000
75%	8.0450	7.640000	1.000000
max	9.4000	9.060000	1.000000

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   cgpa          100 non-null    float64
 1   resume_score  100 non-null    float64
 2   placed        100 non-null    int64  
dtypes: float64(2), int64(1)
memory usage: 2.5 KB
```

In [10]:

```
top_left_corner_df=df.iloc[:5,:3]
top_left_corner_df
```

Out[10]:

	cgpa	resume_score	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1

In [12]:

```
s=df.axes  # we can take the rows
s
```

Out[12]:

```
[RangeIndex(start=0, stop=100, step=1),
 Index(['cgpa', 'resume_score', 'placed'], dtype='object')]
```

In [13]:

```
df.dtypes
```

Out[13]:

```
cgpa            float64
resume_score    float64
placed            int64
dtype: object
```

In [17]:
```
df.empty # checks the empty values
```

Out[17]:
```
False
```

In [18]:
```
df.ndim # number of dimension
```

Out[18]:
```
2
```

In [19]:
```
df.shape
```

Out[19]:
```
(100, 3)
```

In [20]:
```
df.size
```

Out[20]:
```
300
```

In [22]:

```
df.values  # get the numpy are for dataframe
```

```
Out[22]: array([[8.14, 6.52, 1.  ],
                [6.17, 5.17, 0.  ],
                [8.27, 8.86, 1.  ],
                [6.88, 7.27, 1.  ],
                [7.52, 7.3 , 1.  ],
                [8.77, 6.19, 1.  ],
                [5.34, 7.09, 0.  ],
                [6.56, 6.29, 0.  ],
                [6.32, 6.71, 0.  ],
                [7.69, 7.12, 1.  ],
                [6.18, 6.35, 0.  ],
                [5.44, 6.54, 0.  ],
                [6.09, 7.01, 0.  ],
                [8.5 , 5.09, 1.  ],
                [7.51, 6.25, 1.  ],
                [8.88, 5.93, 1.  ],
                [8.04, 7.64, 1.  ],
                [7.81, 8.71, 1.  ],
                [5.94, 5.88, 0.  ],
                [6.75, 8.11, 1.  ],
                [5.8 , 8.06, 0.  ],
                [6.53, 7.64, 0.  ],
                [6.16, 5.77, 0.  ],
                [6.05, 7.13, 0.  ],
                [8.22, 6.18, 1.  ],
                [7.76, 5.68, 1.  ],
                [6.27, 6.47, 0.  ],
                [5.51, 6.15, 0.  ],
                [7.46, 7.67, 1.  ],
                [6.19, 7.3 , 0.  ],
                [7.36, 7.15, 1.  ],
                [5.92, 7.02, 0.  ],
                [5.87, 7.96, 0.  ],
                [8.43, 7.73, 1.  ],
                [8.87, 7.19, 1.  ],
                [8.07, 7.48, 1.  ],
                [8.16, 7.56, 1.  ],
                [9.05, 8.21, 1.  ],
                [6.  , 8.72, 0.  ],
                [7.5 , 6.19, 1.  ],
                [8.25, 5.32, 1.  ],
                [8.68, 5.15, 1.  ],
                [6.9 , 6.91, 1.  ],
                [8.21, 7.95, 1.  ],
                [5.47, 5.92, 0.  ],
                [8.1 , 5.44, 1.  ],
                [5.83, 5.21, 0.  ],
                [7.05, 8.14, 1.  ],
                [5.54, 6.57, 0.  ],
                [5.46, 6.73, 0.  ],
                [8.22, 6.74, 1.  ],
                [6.54, 7.39, 0.  ],
                [5.9 , 7.5 , 0.  ],
                [6.  , 7.16, 0.  ],
                [5.92, 7.18, 0.  ],
                [6.94, 6.87, 1.  ],
                [6.13, 6.43, 0.  ],
                [6.34, 7.21, 0.  ],
                [6.47, 7.37, 0.  ],
                [5.95, 7.57, 0.  ],
                [5.87, 6.64, 0.  ],
                [6.89, 7.96, 1.  ],
                [5.75, 8.43, 0.  ],
                [8.65, 7.58, 1.  ],
                [7.93, 8.09, 1.  ],
                [6.04, 8.75, 0.  ],
                [8.35, 8.02, 1.  ],
                [6.59, 6.81, 1.  ],
                [6.01, 7.49, 0.  ],
                [8.06, 9.06, 1.  ],
                [7.12, 7.41, 1.  ],
                [7.34, 8.22, 1.  ],
                [7.63, 7.98, 1.  ],
                [5.76, 6.48, 0.  ],
                [5.54, 7.36, 0.  ],
                [6.34, 7.94, 1.  ],
                [9.4 , 5.5 , 1.  ],
                [5.88, 6.92, 0.  ],
                [5.79, 5.66, 0.  ],
                [5.27, 7.28, 0.  ],
                [7.83, 7.7 , 1.  ],
                [6.12, 6.72, 0.  ],
                [7.92, 6.06, 1.  ],
                [7.6 , 8.08, 1.  ],
                [5.76, 6.49, 0.  ],
                [6.72, 5.46, 0.  ],
                [6.18, 5.76, 0.  ],
                [5.62, 5.05, 0.  ],
                [8.07, 6.07, 1.  ],
                [5.99, 7.49, 0.  ],
                [5.85, 5.56, 0.  ],
                [8.28, 6.3 , 1.  ],
                [5.43, 6.18, 0.  ],
                [9.31, 7.39, 1.  ],
                [8.01, 4.95, 1.  ],
                [6.33, 6.38, 0.  ],
                [8.23, 7.76, 1.  ],
                [6.65, 7.78, 0.  ],
                [8.14, 5.63, 1.  ],
                [6.09, 6.61, 0.  ]])
```

```
In [23]: a=df.copy()
```

In [24]:

```
df.sort_values(by='resume_score')
```

Out[24]:

	cgpa	resume_score	placed
94	8.01	4.95	1
87	5.62	5.05	0
13	8.50	5.09	1
41	8.68	5.15	1
1	6.17	5.17	0
...	...	...	...
17	7.81	8.71	1
38	6.00	8.72	0
65	6.04	8.75	0
2	8.27	8.86	1
69	8.06	9.06	1

100 rows × 3 columns

In [26]:

```
df.sort_index()    # to sort the data index wise
```

Out[26]:

	cgpa	resume_score	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1
...	...	...	...
95	6.33	6.38	0
96	8.23	7.76	1
97	6.65	7.78	0
98	8.14	5.63	1
99	6.09	6.61	0

100 rows × 3 columns

In [27]:

```
b=df.astype(int)  #tupe cov=nversion
# b=df["cgpa"].astype(int)
b
```

Out[27]:

	cgpa	resume_score	placed
0	8	6	1
1	6	5	0
2	8	8	1
3	6	7	1
4	7	7	1
...	...	...	...
95	6	6	0
96	8	7	1
97	6	7	0
98	8	5	1
99	6	6	0

100 rows × 3 columns

In [30]:

```
df.add(4)
#df['cgpa']=df['cgpa'].add(4)
```

Out[30]:

	cgpa	resume_score	placed
0	12.14	10.52	5
1	10.17	9.17	4
2	12.27	12.86	5
3	10.88	11.27	5
4	11.52	11.30	5
...	...	...	...
95	10.33	10.38	4
96	12.23	11.76	5
97	10.65	11.78	4
98	12.14	9.63	5
99	10.09	10.61	4

100 rows × 3 columns

In [31]:

```
df.abs()
```

Out[31]:

	cgpa	resume_score	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1
...	...	...	...
95	6.33	6.38	0
96	8.23	7.76	1
97	6.65	7.78	0
98	8.14	5.63	1
99	6.09	6.61	0

100 rows × 3 columns

In [32]:

```
df.count()
```

Out[32]:

```
cgpa            100
resume_score    100
placed          100
dtype: int64
```

In [33]:

```
df.max()
```

Out[33]:

```
cgpa            9.40
resume_score    9.06
placed          1.00
dtype: float64
```

In [34]:

```
df.min()
```

Out[34]:

```
cgpa            5.27
resume_score    4.95
placed          0.00
dtype: float64
```

In [35]:

```
df.median()
```

Out[35]:

```
cgpa            6.620
resume_score    7.055
placed          0.500
dtype: float64
```

In [36]:

```
df.mean()
```

Out[36]:

```
cgpa            6.9422
resume_score    6.9305
placed          0.5000
dtype: float64
```

In [37]:

```
df.sum()
```

Out[37]:

```
cgpa            694.22
resume_score    693.05
placed           50.00
dtype: float64
```

In [39]:

```
df.filter(items=['cgpa','placed'])
```

Out[39]:

	cgpa	placed
0	8.14	1
1	6.17	0
2	8.27	1
3	6.88	1
4	7.52	1
...	...	...
95	6.33	0
96	8.23	1
97	6.65	0
98	8.14	1
99	6.09	0

100 rows × 2 columns

In [41]:

```
df[['cgpa','placed']]
```

Out[41]:

	cgpa	placed
0	8.14	1
1	6.17	0
2	8.27	1
3	6.88	1
4	7.52	1
...	...	...
95	6.33	0
96	8.23	1
97	6.65	0
98	8.14	1
99	6.09	0

100 rows × 2 columns

In [42]:

```
df.filter(items=[5,6], axis=0) # for row=0 and column=1
```

Out[42]:

	cgpa	resume_score	placed
5	8.77	6.19	1
6	5.34	7.09	0

In [43]:

```
df.filter(like='5',axis=0) # return the data with index 5
```

Out[43]:

	cgpa	resume_score	placed
5	8.77	6.19	1
15	8.88	5.93	1
25	7.76	5.68	1
35	8.07	7.48	1
45	8.10	5.44	1
50	8.22	6.74	1
51	6.54	7.39	0
52	5.90	7.50	0
53	6.00	7.16	0
54	5.92	7.18	0
55	6.94	6.87	1
56	6.13	6.43	0
57	6.34	7.21	0
58	6.47	7.37	0
59	5.95	7.57	0
65	6.04	8.75	0
75	6.34	7.94	1
85	6.72	5.46	0
95	6.33	6.38	0

In [44]:

```
df.to_dict() # to save in dictionary
```

Out[44]:

```
{'cgpa': {0: 8.14,
  1: 6.17,
  2: 8.27,
  3: 6.88,
  4: 7.52,
  5: 8.77,
  6: 5.34,
  7: 6.56,
  8: 6.32,
  9: 7.69,
  10: 6.18,
  11: 5.44,
  12: 6.09,
  13: 8.5,
  14: 7.51,
  15: 8.88,
  16: 8.04,
  17: 7.81,
  18: 5.94,
```

In [45]:
```
df.to_string() # to save in string
```

Out[45]:
```
'    cgpa  resume_score  placed\n0    8.14          6.52      1\n1    6.17
5.17       0\n2   8.27          8.86       1\n3   6.88          7.27
1\n4   7.52          7.30       1\n5   8.77          6.19       1\n6   5.3
4          7.09       0\n7   6.56          6.29       0\n8   6.32
6.71       0\n9   7.69          7.12       1\n10  6.18          6.35
0\n11  5.44          6.54       0\n12  6.09          7.01       0\n13  8.5
0          5.09       1\n14  7.51          6.25       1\n15  8.88
5.93       1\n16  8.04          7.64       1\n17  7.81          8.71
1\n18  5.94          5.88       0\n19  6.75          8.11       1\n20  5.8
0          8.06       0\n21  6.53          7.64       0\n22  6.16
5.77       0\n23  6.05          7.13       0\n24  8.22          6.18
1\n25  7.76          5.68       1\n26  6.27          6.47       0\n27  5.5
1          6.15       0\n28  7.46          7.67       1\n29  6.19
7.30       0\n30  7.36          7.15       1\n31  5.92          7.02
0\n32  5.87          7.96       0\n33  8.43          7.73       1\n34  8.8
7          7.19       1\n35  8.07          7.48       1\n36  8.16
7.56       1\n37  9.05          8.21       1\n38  6.00          8.72
0\n39  7.50          6.19       1\n40  8.25          5.32       1\n41  8.6
8          5.15       1\n42  6.90          6.91       1\n43  8.21
7.95       1\n44  5.47          5.92       0\n45  8.10          5.44
1\n46  5.83          5.21       0\n47  7.05          8.14       1\n48  5.5
4          6.57       0\n49  5.46          6.73       0\n50  8.22
6.74       1\n51  6.54          7.39       0\n52  5.90          7.50
0\n53  6.00          7.16       0\n54  5.92          7.18       0\n55  6.9
4          6.87       1\n56  6.13          6.43       0\n57  6.34
7.21       0\n58  6.47          7.37       0\n59  5.95          7.57
0\n60  5.87          6.64       0\n61  6.89          7.96       1\n62  5.7
5          8.43       0\n63  8.65          7.58       1\n64  7.93
8.09       1\n65  6.04          8.75       0\n66  8.35          8.02
1\n67  6.59          6.81       1\n68  6.01          7.49       0\n69  8.0
6          9.06       1\n70  7.12          7.41       1\n71  7.34
8.22       1\n72  7.63          7.98       1\n73  5.76          6.48
0\n74  5.54          7.36       0\n75  6.34          7.94       1\n76  9.4
0          5.50       1\n77  5.88          6.92       0\n78  5.79
5.66       0\n79  5.27          7.28       0\n80  7.83          7.70
1\n81  6.12          6.72       0\n82  7.92          6.06       1\n83  7.6
0          8.08       1\n84  5.76          6.49       0\n85  6.72
5.46       0\n86  6.18          5.76       0\n87  5.62          5.05
0\n88  8.07          6.07       1\n89  5.99          7.49       0\n90  5.8
5          5.56       0\n91  8.28          6.30       1\n92  5.43
6.18       0\n93  9.31          7.39       1\n94  8.01          4.95
1\n95  6.33          6.38       0\n96  8.23          7.76       1\n97  6.6
5          7.78       0\n98  8.14          5.63       1\n99  6.09
6.61       0'
```

In [46]:
```
idx=df.columns
idx
```

Out[46]:
```
Index(['cgpa', 'resume_score', 'placed'], dtype='object')
```

In [47]:
```
df.columns[0]
```

Out[47]:
```
'cgpa'
```

In [49]:
```
df.columns.tolist() # list of column labels
```

Out[49]:
```
['cgpa', 'resume_score', 'placed']
```

In [53]:
```
df.columns.values  #3 array of colums labels
```

Out[53]:
```
array(['cgpa', 'resume_score', 'placed'], dtype=object)
```

In [54]:
```
df.rename(columns={'cgpa':'half_yearly_marks','resume_score':'semester_marks
```

Out[54]:

	half_yearly_marks	semester_marks	placed
0	8.14	6.52	1
1	6.17	5.17	0
2	8.27	8.86	1
3	6.88	7.27	1
4	7.52	7.30	1
...	...	...	...
95	6.33	6.38	0
96	8.23	7.76	1
97	6.65	7.78	0
98	8.14	5.63	1
99	6.09	6.61	0

100 rows × 3 columns

In [58]:
```
df['half']=df['cgpa'].where(df['cgpa']>5.00, other=0)
df.head(10)
```

Out[58]:

	cgpa	resume_score	placed	half
0	8.14	6.52	1	8.14
1	6.17	5.17	0	6.17
2	8.27	8.86	1	8.27
3	6.88	7.27	1	6.88
4	7.52	7.30	1	7.52
5	8.77	6.19	1	8.77
6	5.34	7.09	0	5.34
7	6.56	6.29	0	6.56
8	6.32	6.71	0	6.32
9	7.69	7.12	1	7.69

In [ ]:
```

```