

```
In [37]: import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
In [38]: df = pd.read_csv("C:/Users/SSOS09/Music/Book1.csv")
```

```
In [39]: df
```

```
Out[39]:
```

	customerID	gender	age	annual income(k\$)	spending score(1-100)
0	1	female	41	97	88
1	2	female	57	86	98
2	3	male	43	62	100
3	4	male	56	74	100
4	5	male	30	28	92
...	...	...	...	...	...
195	196	female	23	23	95
196	197	male	51	56	19
197	198	female	35	72	47
198	199	female	37	25	82
199	200	male	49	64	91

200 rows × 5 columns

```
In [40]: df.mean()
```

```
Out[40]: customerID      100.500
age                39.945
annual income(k$)   60.020
spending score(1-100)  80.955
dtype: float64
```

```
In [41]: df.loc[:, 'age'].mean()
```

```
Out[41]: 39.945
```

```
In [42]: df.mean(axis=1)[0:4]
```

```
Out[42]: 0    56.75
1    60.75
2    52.00
3    58.50
dtype: float64
```

```
In [43]: df.median()
```

```
Out[43]: customerID      100.5  
age          40.5  
annual income(k$)      61.0  
spending score(1-100)   92.5  
dtype: float64
```

```
In [44]: df.loc[:, 'age'].median()
```

```
Out[44]: 40.5
```

```
In [45]: df.median(axis=1)[0:4]
```

```
Out[45]: 0    64.5  
1    71.5  
2    52.5  
3    65.0  
dtype: float64
```

```
In [46]: df.mode()
```

```
Out[46]:
```

	customerID	gender	age	annual income(k\$)	spending score(1-100)
0	1	female	41.0	23.0	100.0
1	2	NaN	NaN	NaN	NaN
2	3	NaN	NaN	NaN	NaN
3	4	NaN	NaN	NaN	NaN
4	5	NaN	NaN	NaN	NaN
...	...	...	...	...	...
195	196	NaN	NaN	NaN	NaN
196	197	NaN	NaN	NaN	NaN
197	198	NaN	NaN	NaN	NaN
198	199	NaN	NaN	NaN	NaN
199	200	NaN	NaN	NaN	NaN

200 rows × 5 columns

```
In [47]: df.loc[:, 'age'].mode()
```

```
Out[47]: 0    41  
dtype: int64
```

```
In [48]: df.min()
```

```
Out[48]: customerID      1  
gender          female  
age            20  
annual income(k$)      20  
spending score(1-100)   1  
dtype: object
```

```
In [49]: df.loc[:, 'age'].min(skipna = False)
```

```
Out[49]: 20
```

```
In [50]: df.max()
```

```
Out[50]: customerID      200  
gender      male  
age         60  
annual income(k$)      100  
spending score(1-100)  100  
dtype: object
```

```
In [51]: df.loc[:, 'age'].max(skipna = False)
```

```
Out[51]: 60
```

```
In [52]: df.std()
```

```
Out[52]: customerID      57.879185  
age      12.044389  
annual income(k$)      23.691476  
spending score(1-100)  25.164244  
dtype: float64
```

```
In [53]: df.loc[:, 'age'].std()
```

```
Out[53]: 12.044389214808232
```

```
In [54]: df.std(axis=1)[0:4]
```

```
Out[54]: 0    44.544921  
1    42.781421  
2    40.356743  
3    40.575033  
dtype: float64
```

```
In [55]: df.groupby(['gender'])['age'].mean()
```

```
Out[55]: gender  
female    39.063063  
male      41.044944  
Name: age, dtype: float64
```

```
In [56]: df_u= df.rename(columns= {'annual income(k$)': 'income'}, inplace=False)
```

```
In [57]: df_u.groupby(['gender']).income.mean()
```

```
Out[57]: gender  
female    59.783784  
male      60.314607  
Name: income, dtype: float64
```

```
In [58]: from sklearn import preprocessing
enc = preprocessing.OneHotEncoder()
enc_df = pd.DataFrame(enc.fit_transform(df[['gender']]).toarray())
enc_df
```

Out[58]:

	0	1
0	1.0	0.0
1	1.0	0.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0
...	...	...
195	1.0	0.0
196	0.0	1.0
197	1.0	0.0
198	1.0	0.0
199	0.0	1.0

200 rows × 2 columns

```
In [59]: df_encode = df_u.join(enc_df)
df_encode
```

Out[59]:

	customerID	gender	age	income	spending score(1-100)	0	1
0	1	female	41	97	88	1.0	0.0
1	2	female	57	86	98	1.0	0.0
2	3	male	43	62	100	0.0	1.0
3	4	male	56	74	100	0.0	1.0
4	5	male	30	28	92	0.0	1.0
...	...	...	...	...	...	...	...
195	196	female	23	23	95	1.0	0.0
196	197	male	51	56	19	0.0	1.0
197	198	female	35	72	47	1.0	0.0
198	199	female	37	25	82	1.0	0.0
199	200	male	49	64	91	0.0	1.0

200 rows × 7 columns

```
In [60]: csv_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
```

```
In [61]: col_names= ['sepal_length', 'sepal_width', 'petal_length', 'petal-width', 'species']
```

```
In [62]: iris = pd.read_csv(csv_url, names = col_names)
```

```
In [63]: irisSet = (iris['species'] == 'Iris-setosa')
```

```
In [64]: print('Iris-setosa')
print(iris[irisSet].describe())
```

```
Iris-setosa
      sepal_length  sepal_width  petal_length  petal-width
count      50.00000      50.00000      50.00000      50.00000
mean         5.00600         3.41800         1.46400         0.24400
std          0.35249         0.38102         0.17351         0.10721
min          4.30000         2.30000         1.00000         0.10000
25%          4.80000         3.12500         1.40000         0.20000
50%          5.00000         3.40000         1.50000         0.20000
75%          5.20000         3.67500         1.57500         0.30000
max          5.80000         4.40000         1.90000         0.60000
```

```
In [65]: irisVer = (iris['species']== 'Iris-versicolor')
```

```
In [66]: print('Iris-versicolor')
print(iris[irisVer].describe())
```

```
Iris-versicolor
      sepal_length  sepal_width  petal_length  petal-width
count      50.00000      50.00000      50.00000      50.00000
mean         5.93600         2.77000         4.26000         1.32600
std          0.51617         0.31379         0.46991         0.19775
min          4.90000         2.00000         3.00000         1.00000
25%          5.60000         2.52500         4.00000         1.20000
50%          5.90000         2.80000         4.35000         1.30000
75%          6.30000         3.00000         4.60000         1.50000
max          7.00000         3.40000         5.10000         1.80000
```

```
In [67]: irisVir = (iris["species"]== 'Iris-virginica')
```

```
In [68]: print('Iris-virginica')
print(iris[irisVir].describe())
```

```
Iris-virginica
      sepal_length  sepal_width  petal_length  petal-width
count      50.00000      50.00000      50.00000      50.00000
mean         6.58800         2.97400         5.55200         2.02600
std          0.63588         0.32249         0.55189         0.27465
min          4.90000         2.20000         4.50000         1.40000
25%          6.22500         2.80000         5.10000         1.80000
50%          6.50000         3.00000         5.55000         2.00000
75%          6.90000         3.17500         5.87500         2.30000
max          7.90000         3.80000         6.90000         2.50000
```

In [ ]:

In [ ]: