

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
```

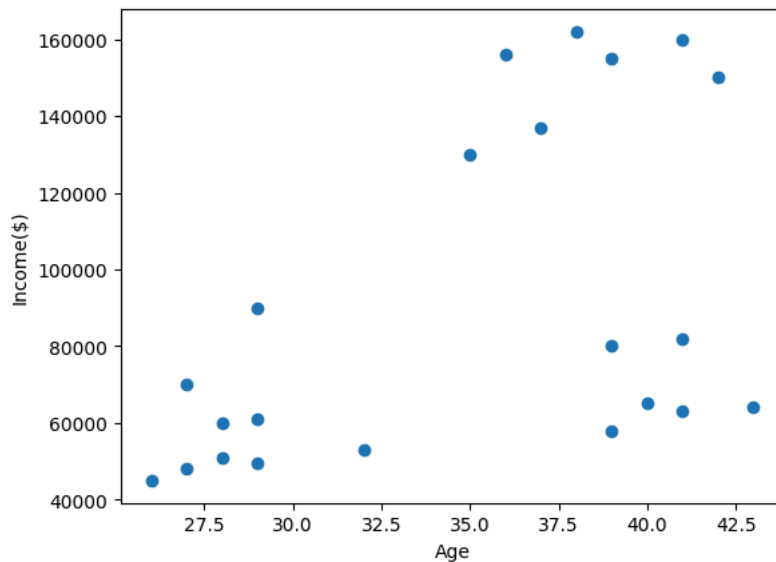
```
df = pd.read_csv("/content/income.csv")
df.head()
```

	Name	Age	Income(\$)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

Next steps: [View recommended plots](#)

```
plt.scatter(df.Age, df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

```
Text(0, 0.5, 'Income($)')
```



```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age', 'Income($)']])
y_predicted
```

```

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 1 in the future.
warnings.warn(
array([2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 1],
      dtype=int32)
```

```
df['cluster']=y_predicted
df.head()
```

	Name	Age	Income(\$)	cluster
0	Rob	27	70000	2
1	Michael	29	90000	2
2	Mohan	29	61000	1
3	Ismail	28	60000	1
4	Kory	42	150000	0

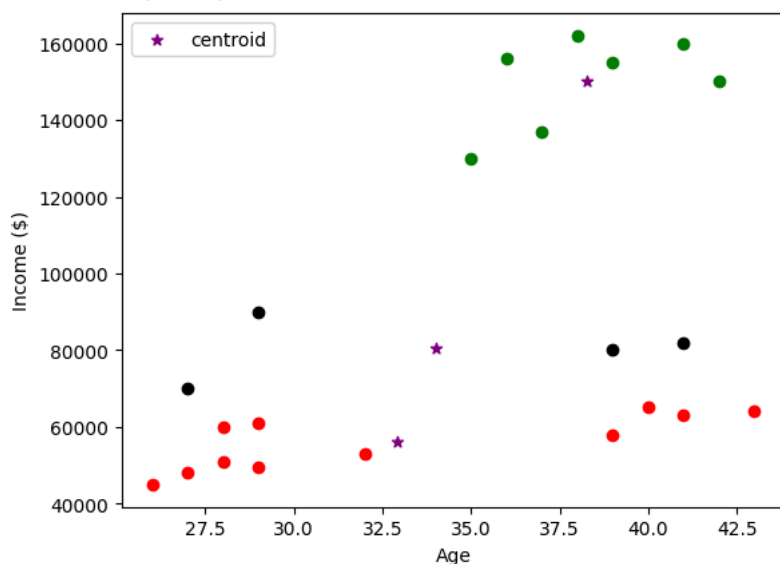
Next steps: [View recommended plots](#)

```
km.cluster_centers_
```

```
array([[3.82857143e+01, 1.50000000e+05],
       [3.29090909e+01, 5.61363636e+04],
       [3.40000000e+01, 8.05000000e+04]])
```

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)',color='green')
plt.scatter(df2.Age,df2['Income($)',color='red')
plt.scatter(df3.Age,df3['Income($)',color='black')
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color='purple',marker='*',label='centroid')
plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7c5f3f4094b0>
```



```
scaler = MinMaxScaler()

scaler.fit(df[['Income($)']])
df['Income($)' = scaler.transform(df[['Income($)']])

scaler.fit(df[['Age']])
df['Age' = scaler.transform(df[['Age']])
```

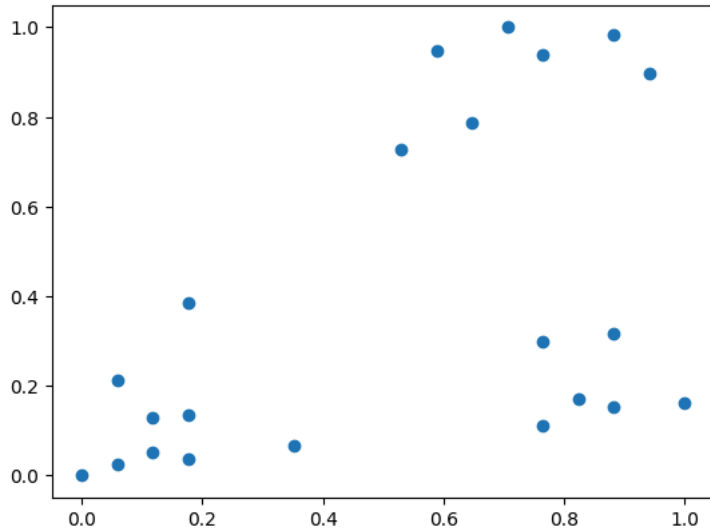
```
df.head()
```

	Name	Age	Income(\$)	cluster
0	Rob	0.058824	0.213675	2
1	Michael	0.176471	0.384615	2
2	Mohan	0.176471	0.136752	1
3	Ismail	0.117647	0.128205	1
4	Kory	0.941176	0.897436	0


Next steps: [View recommended plots](#)

```
plt.scatter(df.Age,df['Income($)'])
```

 <matplotlib.collections.PathCollection at 0x7c5f3f4a2710>



```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age', 'Income($)']])
y_predicted
```

 /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 1 in the future. This will affect the results of the fit method. To avoid this warning, you should explicitly define `n_init` when calling the function. Please refer to the documentation for more details.

warnings.warn(
array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0],
dtype=int32)


```
df['cluster']=y_predicted
df.head()
```



	Name	Age	Income(\$)	cluster	
0	Rob	0.058824	0.213675	1	
1	Michael	0.176471	0.384615	1	
2	Mohan	0.176471	0.136752	1	
3	Ismail	0.117647	0.128205	1	
4	Kory	0.941176	0.897436	2	

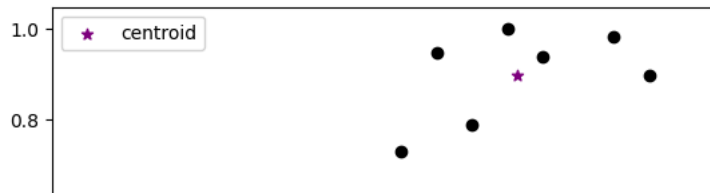
Next steps:  [View recommended plots](#)

```
km.cluster_centers_
```

 array([[0.85294118, 0.2022792],
[0.1372549 , 0.11633428],
[0.72268908, 0.8974359]])

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color='purple',marker='*',label='centroid')
plt.legend()
```

 <matplotlib.legend.Legend at 0x7c5f3f1d72e0>



▼ Elbow Plot

```

sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['Age', 'Income($)']])
    sse.append(km.inertia_)


```

 [Show hidden output](#)

```

plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)

```

 [<matplotlib.lines.Line2D at 0x7c5f3f247be0>]

