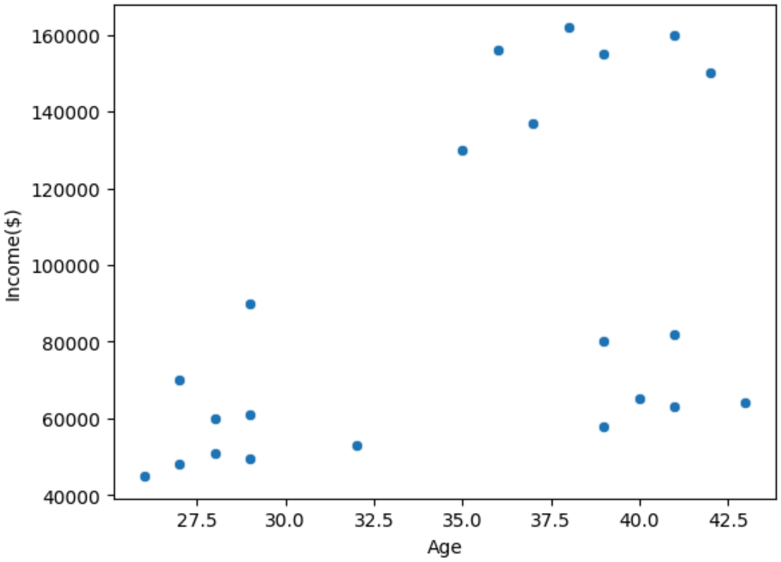
```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

```
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: Generate code with df View recommended plots New interactive sheet

```
sns.scatterplot(x='Age', y='Income($)', data=df)
plt.show()
```



KMEANS CLUSTERING

```
from sklearn.cluster import KMeans
model = KMeans(n_clusters=3)
model.fit(df[['Age', 'Income($)']])
```

```
KMeans
KMeans(n_clusters=3)
```

```
print("Sum of squared errirs with K:3  = {}".format(model.inertia_))
print("Clusters for each instance  = {}".format(model.labels_))
```

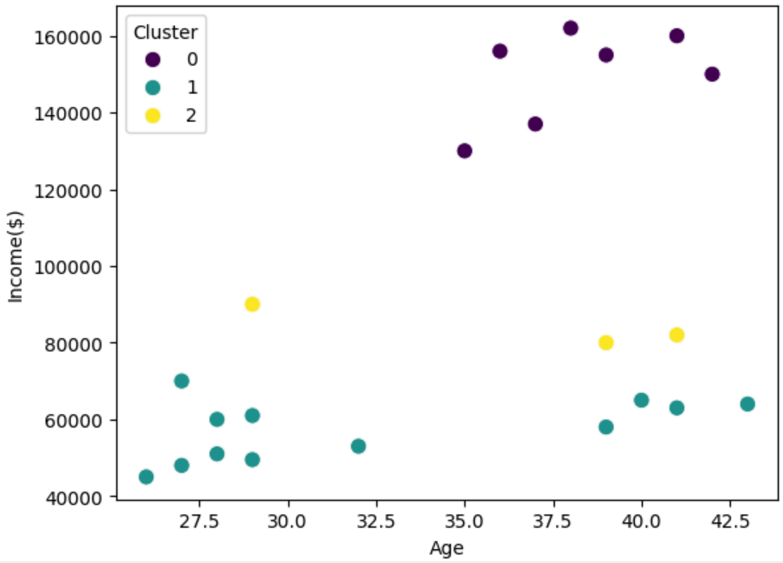
```
Sum of squared errirs with K:3  = 1606229737.6785712
Clusters for each instance  = [1 2 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 2 2 1]
```

```
df['Cluster'] = model.labels_
df.head()
```

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

Next steps: Generate code with df   View recommended plots   New interactive sheet

```
sns.scatterplot(x='Age', y='Income($)', data=df, hue='Cluster' , palette="viridis", s=80)
plt.show()
```



```
model.cluster_centers_
```

```
array([[3.82857143e+01, 1.50000000e+05],
       [3.24166667e+01, 5.72916667e+04],
       [3.63333333e+01, 8.40000000e+04]])
```

```
df_centers = pd.DataFrame(model.cluster_centers_)
```

Min Max Scaling

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df_scaled = pd.DataFrame(columns=['Age', 'Income($)'])

df_scaled[["Age", "Income($)"]] = scaler.fit_transform(df[["Age", "Income($)"]])
df_scaled.head()
```

	Age	Income($)
0	0.058824	0.213675
1	0.176471	0.384615
2	0.176471	0.136752
3	0.117647	0.128205
4	0.941176	0.897436

Next steps: Generate code with df_scaled   View recommended plots   New interactive sheet

```
km = KMeans(n_clusters=3)
km.fit(df_scaled[["Age", "Income($)"]])
df_scaled['Cluster'] = km.labels_
df_scaled.head()
```

	Age	Income($)	Cluster
0	0.058824	0.213675	2
1	0.176471	0.384615	2
2	0.176471	0.136752	2
3	0.117647	0.128205	2
4	0.941176	0.897436	1

Next steps: Generate code with df_scaled View recommended plots New interactive sheet

```
km.inertia_
```

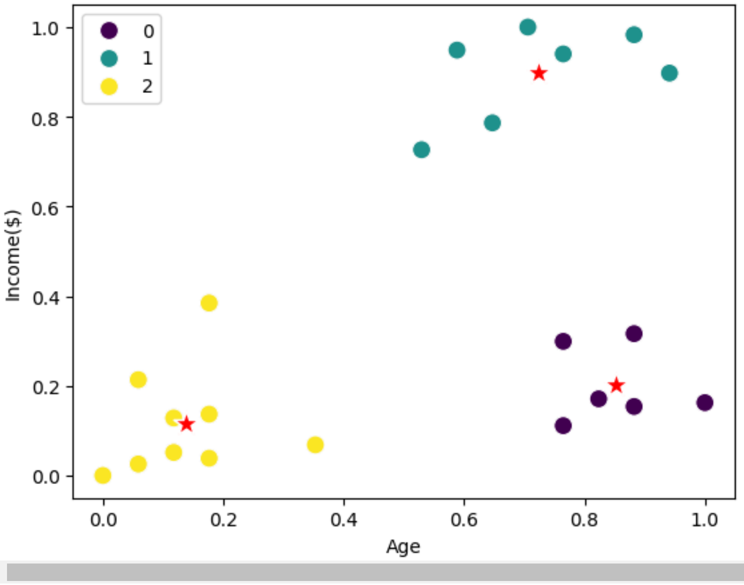
```
0.4750783498553097
```

```
centroids = km.cluster_centers_
centroids
```

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

```
sns.scatterplot(df_scaled, x='Age', y='Income($)',  palette='viridis', s=100, hue="Cluster")
sns.scatterplot(x=centroids[:,0], y=centroids[:,1],  s=200, color="red", marker='*')
```

```
<Axes: xlabel='Age', ylabel='Income($)'>
```



Elbow method to determine optimal number of clusters

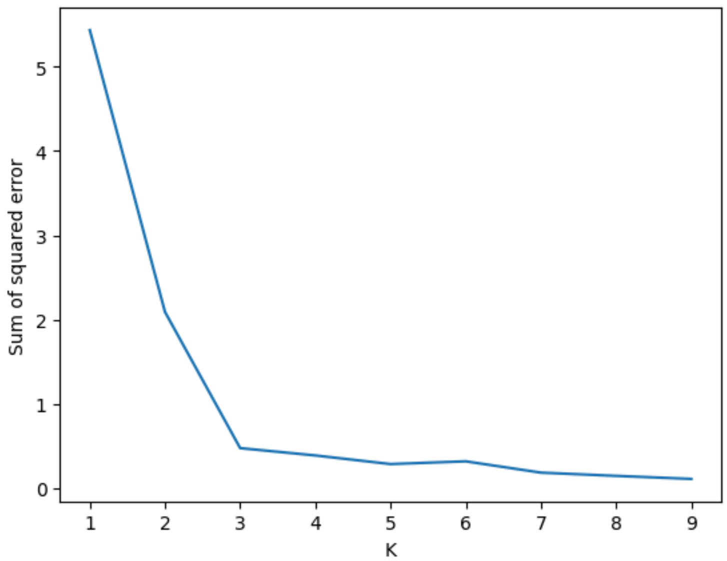
```
sse = []

k_rng = range(1,10)

for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df_scaled[["Age", "Income($)"]])
    sse.append(km.inertia_)
sse
```

```
[5.434011511988179,
 2.091136388699078,
 0.4750783498553097,
 0.38815291664787444,
 0.2860717106689441,
 0.3184894248142853,
 0.18427868455224797,
 0.14580719346046292,
 0.10995816883086573]
```

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


Sum of squared error
K