In [1]: `import numpy as np`
`import pandas as pd`
`import matplotlib as mpl`
`import matplotlib.pyplot as plt`
`import seaborn as sns`

`data = pd.read_csv("content/Mall_Customers (1) (1).csv")`
`data`

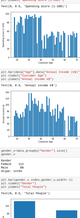
Out[1]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| 0 | 1 | Male | 19 | 15 | 39 |
| 1 | 2 | Male | 21 | 15 | 81 |
| 2 | 3 | Female | 20 | 16 | 6 |
| 3 | 4 | Female | 23 | 16 | 77 |
| 4 | 5 | Female | 31 | 17 | 40 |
| ... | ... | ... | ... | ... | ... |
| 195 | 196 | Female | 35 | 120 | 79 |
| 196 | 197 | Female | 45 | 126 | 28 |
| 197 | 198 | Male | 32 | 126 | 74 |
| 198 | 199 | Male | 32 | 137 | 18 |
| 199 | 200 | Male | 30 | 137 | 83 |

200 rows × 5 columns

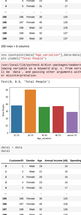
In [2]: `plt.hist(data["Age"],data["Spending Score (1-100)"])`
`plt.xlabel("Annual Income (k$)")`
`plt.ylabel("Spending Score (1-100)")`

Out[2]: Text(0, 0.5, 'Spending Score (1-100)')




In [3]: `plt.hist(data["Age"],data["Annual Income (k$)"])`
`plt.xlabel("Annual Income (k$)")`
`plt.ylabel("Spending Score (1-100)")`

Out[3]: Text(0, 0.5, 'Spending Score (1-100)')



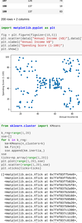

In [4]: `gender.replace(to_replace="Female", value=1)`
`gender`

Out[4]:

| | |
|---|---|
| Female | 112 |
| Male | 88 |
| Name: count, dtype: int64 | |

In [5]: `plt.hist(gender,rwidth=0.5,color=['r','b'])`
`plt.xlabel("Gender")`
`plt.ylabel("Total People")`

Out[5]: Text(0, 0.5, 'Total People')




In [11]: `age.variance("CG.CRT","CG.SRT","CB.CRT","CB.SRT","Labone SRT")`
`Output[11,20,35,40,50]`
`data["Age_cat.variance"]=pd.cut(data["Age"],bins=bins,labels=age_variance,right=False)`
`data`

Out[11]:

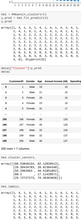
| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) | Age_variance |
|---|---|---|---|---|---|---|
| 0 | 1 | Male | 19 | 15 | 39 | 11.00 |
| 1 | 2 | Male | 21 | 15 | 81 | 11.75 |
| 2 | 3 | Female | 20 | 16 | 6 | 11.25 |
| 3 | 4 | Female | 23 | 16 | 77 | 11.50 |
| 4 | 5 | Female | 31 | 17 | 40 | 15.00 |
| ... | ... | ... | ... | ... | ... | ... |
| 195 | 196 | Female | 35 | 120 | 79 | 34.00 |
| 196 | 197 | Female | 45 | 126 | 28 | 35.00 |
| 197 | 198 | Male | 32 | 126 | 74 | 31.00 |
| 198 | 199 | Male | 32 | 137 | 18 | 31.00 |
| 199 | 200 | Male | 30 | 137 | 83 | 29.00 |

200 rows × 6 columns

In [12]: `sns.countplot(data="Age_variance",data=data)`
`plt.xlabel("Total People")`

```
/usr/local/lib/python3.9/dist-packages/seaborn/_oldcoreutils.py:83: FutureWarning: Pass the fol
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.9/dist-packages/matplotlib/cbook/__init__.py:1335: FutureWarning: Using
  in 3.6. Use keyword only arguments without an explicit keyword will result in an error
  deprecation_message(msg, *args, **kwd)
```

Out[12]: Text(0.5, 0.0, 'Total People')




In [20]: `data1 = data`
`data1`

Out[20]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) | Age_variance |
|---|---|---|---|---|---|---|
| 0 | 1 | Male | 19 | 15 | 39 | 11.00 |
| 1 | 2 | Male | 21 | 15 | 81 | 11.75 |
| 2 | 3 | Female | 20 | 16 | 6 | 11.25 |
| 3 | 4 | Female | 23 | 16 | 77 | 11.50 |
| 4 | 5 | Female | 31 | 17 | 40 | 15.00 |
| ... | ... | ... | ... | ... | ... | ... |
| 195 | 196 | Female | 35 | 120 | 79 | 34.00 |
| 196 | 197 | Female | 45 | 126 | 28 | 35.00 |
| 197 | 198 | Male | 32 | 126 | 74 | 31.00 |
| 198 | 199 | Male | 32 | 137 | 18 | 31.00 |
| 199 | 200 | Male | 30 | 137 | 83 | 29.00 |

200 rows × 6 columns

In [30]: `x = data1.iloc[:,3:5]`
`x`

Out[30]:

| | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|
| 0 | 15 | 39 |
| 1 | 15 | 81 |
| 2 | 16 | 6 |
| 3 | 16 | 77 |
| 4 | 17 | 40 |
| ... | ... | ... |
| 195 | 120 | 79 |
| 196 | 126 | 28 |
| 197 | 126 | 74 |
| 198 | 137 | 18 |
| 199 | 137 | 83 |

200 rows × 2 columns

In [31]: `import matplotlib.pyplot as plt`

`fig = plt.figure(figsize=(8,6))`
`plt.scatter(x["Annual Income (k$)"],x["Spending Score (1-100)"])`
`plt.xlabel("Annual Income (k$)")`
`plt.ylabel("Spending Score (1-100)")`
`plt.show()`

Out[31]:

In [34]: `from sklearn.cluster import KMeans`

`k_rng=range(1,11)`
`sse=[]`
`for k in k_rng:`
`  km=KMeans(n_clusters=k)`
`  km.fit(x)`
`  sse.append(km.inertia_)`
`print(sse)`
`clusters.append(km.fit_predict(x))`
`df["cluster"]=clusters`
`df.head(200)`
`print(df.shape)`
`print(df.columns)`

Out[34]:

```
[244.07489114658925, 212.8092686690502, 157.70400161533203, 109.77368421052632, 73.679786286731
[array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       
```