

+ Code + Text

RAM
Disk

4s
from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

3s
[2] import numpy as np
import pandas as pd
from pandas import plotting
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os
warnings.filterwarnings("ignore")
py.offline.init_notebook_mode(connected = True)
plt.style.use('fivethirtyeight')
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
import plotly.offline as py
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

3s completed at 2:56 PM

```
[2] from plotly import tools
init_notebook_mode(connected = True)
import plotly.figure_factory as ff
import seaborn as sns
plt.style.use('ggplot')
```

```
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
import plotly.graph_objs as go
from plotly import tools
from plotly.subplots import make_subplots
import plotly.offline as py
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import cufflinks as cf
import warnings
warnings.filterwarnings("ignore")
from sklearn.cluster import KMeans
import missingno as msno
from sklearn.preprocessing import LabelEncoder
%matplotlib inline
from sklearn.cluster import KMeans
```

3s completed at 2:56 PM

+ Code + Text RAM Disk

```
[4] from sklearn.preprocessing import LabelEncoder
%matplotlib inline
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
```

```
[5] path="/content/drive/MyDrive/Mall_Customers.csv"
dataset=pd.read_csv(path)
```

```
dataset.head(3)
```

	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

```
[7] dataset.dtypes
```

CustomerID	int64
Gender	object
Age	int64
Annual Income (k\$)	int64
Spending Score (1-100)	int64

3s completed at 2:56 PM

+ Code + Text RAM Disk

```
[7] Annual Income (k$)      int64
     Spending Score (1-100)  int64
     dtype: object
```

```
dataset.rename(index=str, columns={'Annual Income (k$)': 'Income', 'Spending Score (1-100)': 'Score'}, inplace=True)
dataset.head(3)
```

	CustomerID	Gender	Age	Income	Score
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6

```
[9] dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   CustomerID  200 non-null   int64
1   Gender      200 non-null   object
2   Age         200 non-null   int64
3   Income      200 non-null   int64
```

3s completed at 2:56 PM

+ Code + Text RAM Disk

```
[9] 2 Age 200 non-null int64
     3 Income 200 non-null int64
     4 Score 200 non-null int64
dtypes: int64(4), object(1)
memory usage: 9.4+ KB
```

```
print('This dataset contains ',dataset.shape[0],'rows')
print('This dataset contains ',dataset.shape[1],'columns')
```

```
This dataset contains 200 rows
This dataset contains 5 columns
```

```
[11] dataset.iloc[:,1:].corr()
```

	Age	Income	Score
Age	1.000000	-0.012398	-0.327227
Income	-0.012398	1.000000	0.009903
Score	-0.327227	0.009903	1.000000

```
[12] dataset.drop('CustomerID',axis=1,inplace=True)
print("Mean of Annual Income (k$) of Female:",dataset['Income'].loc[dataset['Gender'] == 'Female'].mean())
print("Mean of Annual Income (k$) of Male:",dataset['Income'].loc[dataset['Gender'] == 'Male'].mean())
```

3s completed at 2:56 PM

+ Code + Text

RAM Disk

```
[12] print("Mean of Annual Income (k$) of Female:",dataset['Income'].loc[dataset['Gender'] == 'Female'].mean())  
      print("Mean of Annual Income (k$) of Male:",dataset['Income'].loc[dataset['Gender'] == 'Male'].mean())
```

Mean of Annual Income (k\$) of Female: 59.25
Mean of Annual Income (k\$) of Male: 62.22727272727273

```
[13] plt.figure(figsize=(10,5))  
      sns.heatmap(dataset.corr(),annot=True,cmap='hsv',fmt='.2f',linewidths=2)  
      plt.show()
```

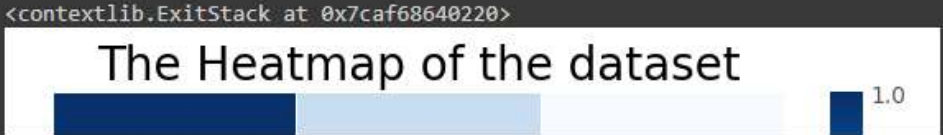


3s completed at 2:56 PM

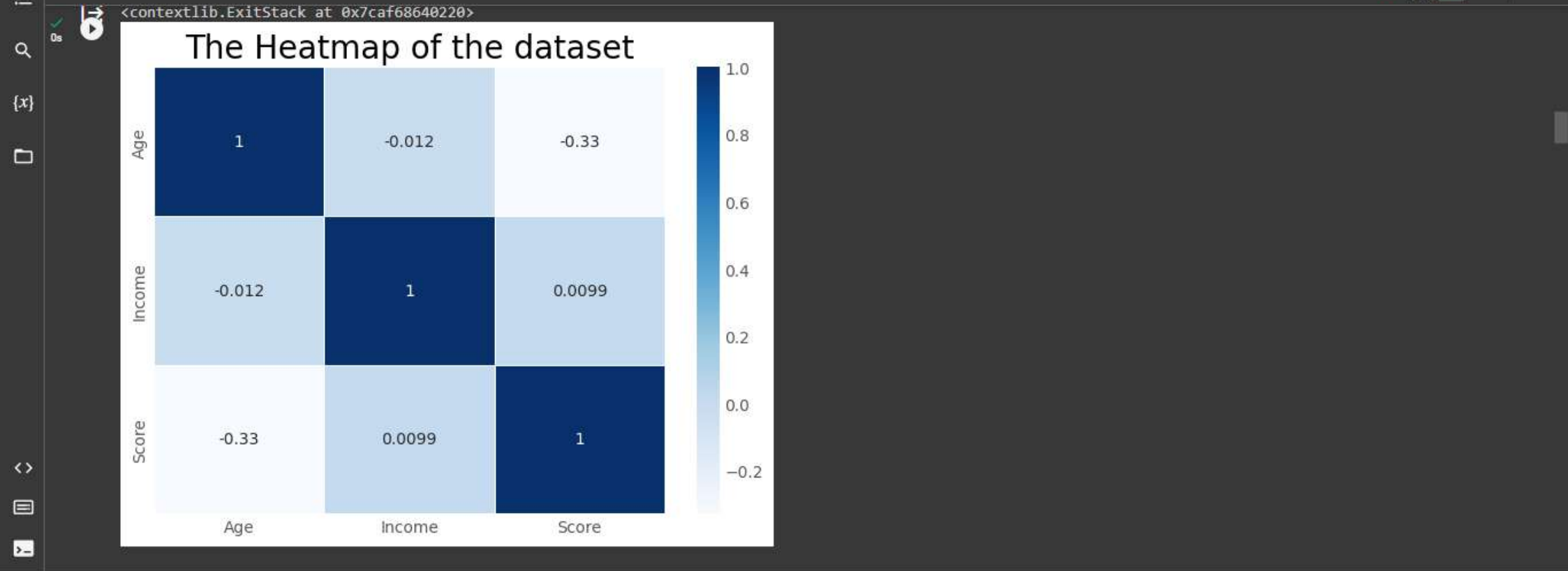
+ Code + Text



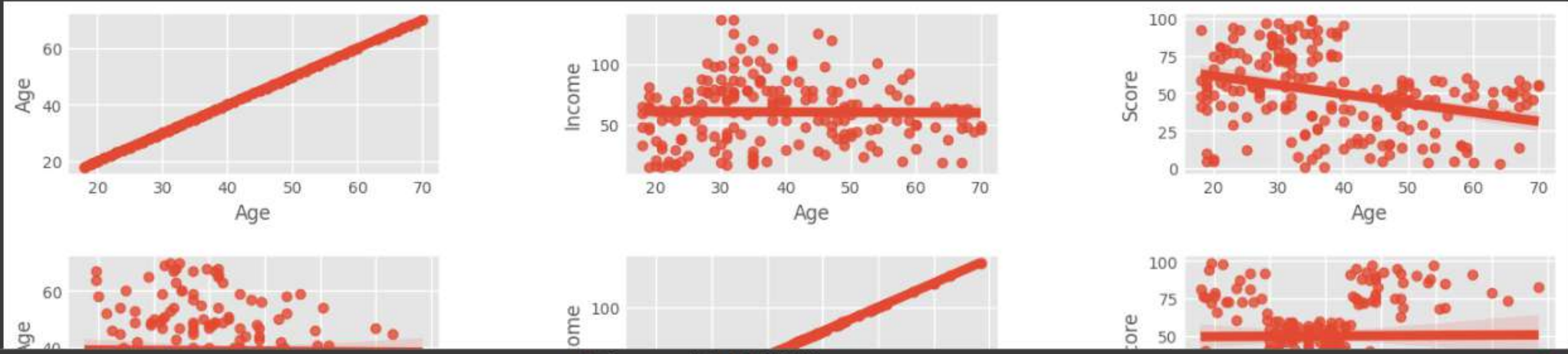
```
[14] hm=sns.heatmap(dataset.iloc[:,1:5].corr(), annot = True, linewidths=0.5, cmap='Blues')
hm.set_title(label='The Heatmap of the dataset', fontsize=20)
hm
plt.ioff()
```



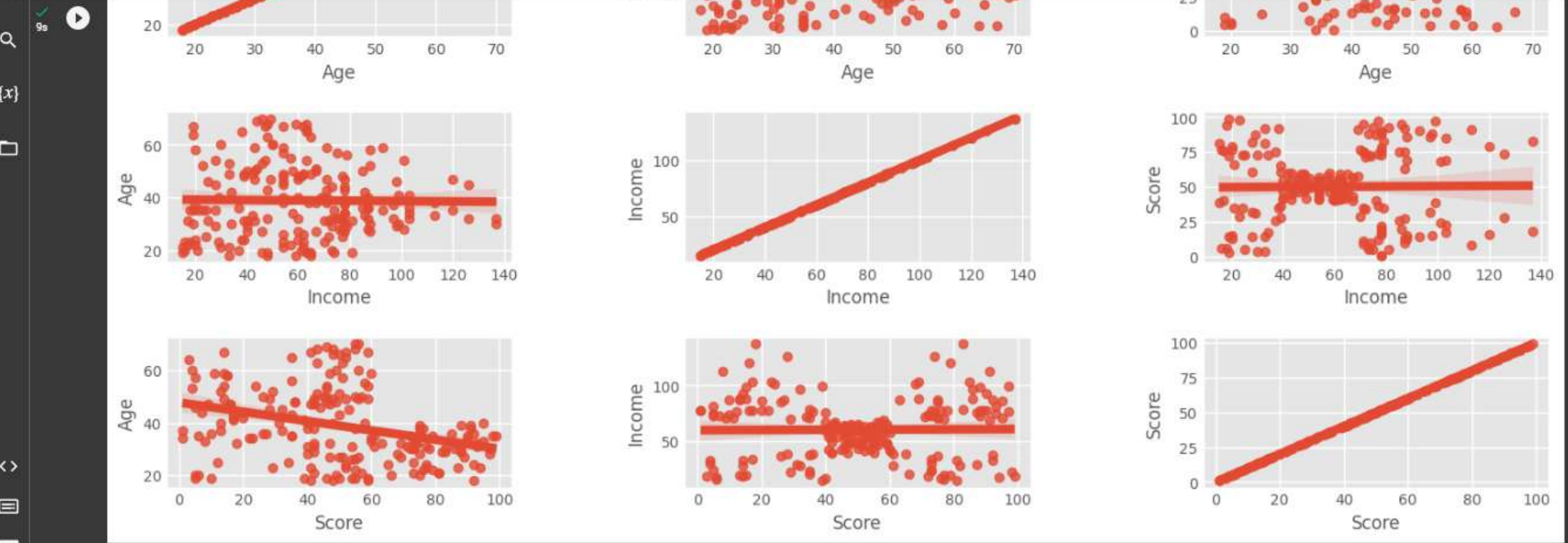
3s completed at 2:56 PM




```
plt.figure(1,figsize=(15,7))
n=0
for x in ['Age','Income','Score']:
    for y in ['Age','Income','Score']:
        n+=1
        plt.subplot(3,3,n)
        plt.subplots_adjust(hspace=0.5,wspace=0.5)
        sns.regplot(x=x,y=y,data=dataset)
        plt.ylabel(y.split()[0]+'+'y.split()[1] if len(y.split())>1 else y)
plt.show()
```



3s completed at 2:56 PM



+ Code + Text

RAM Disk

```
dataset.groupby('Gender').mean()
```

	Age	Income	Score
Gender			
Female	38.098214	59.250000	51.526786
Male	39.806818	62.227273	48.511364

```
[17] print(sum(dataset.duplicated()))  
dataset = dataset.drop_duplicates()
```

0

```
[18] dataset.Age.unique()
```

```
array([19, 21, 20, 23, 31, 22, 35, 64, 30, 67, 58, 24, 37, 52, 25, 46, 54,  
       29, 45, 40, 60, 53, 18, 49, 42, 36, 65, 48, 50, 27, 33, 59, 47, 51,  
       69, 70, 63, 43, 68, 32, 26, 57, 38, 55, 34, 66, 39, 44, 28, 56, 41])
```

```
[20] print(len(dataset.Age.unique()))
```

51

3s completed at 2:56 PM



+ Code + Text

51

dataset.Age.value_counts()[:10]



```
32    11
35     9
19     8
31     8
30     7
49     7
40     6
38     6
47     6
27     6
Name: Age, dtype: int64
```

[21] dataset['Age'].describe()

```
count    200.000000
mean     38.850000
std      13.969007
min      18.000000
25%      28.750000
50%      36.000000
75%      49.000000
max      70.000000
Name: Age, dtype: float64
```

3s completed at 2:56 PM

+ Code + Text

[21] dataset['Age'].describe()

count 200.000000
mean 38.850000
std 13.969007
min 18.000000
25% 28.750000
50% 36.000000
75% 49.000000
max 70.000000
Name: Age, dtype: float64

bins = [18, 22, 50, 70]
labels = ['Young', 'Adult', 'Senior']
dataset['Age Range'] = pd.cut(dataset['Age'], bins=bins, labels=labels, right=False)
dataset.head()

	Gender	Age	Income	Score	Age Range
0	Male	19	15	39	Young
1	Male	21	15	81	Young
2	Female	20	16	6	Young
3	Female	23	16	77	Adult

3s completed at 2:56 PM

+ Code + Text

```
bins = [18, 22, 50, 70]
labels = ['Young', 'Adult', 'Senior']
dataset['Age Range'] = pd.cut(dataset['Age'], bins=bins, labels=labels, right=False)
dataset.head()
```

	Gender	Age	Income	Score	Age Range
0	Male	19	15	39	Young
1	Male	21	15	81	Young
2	Female	20	16	6	Young
3	Female	23	16	77	Adult
4	Female	31	17	40	Adult

```
[26] dataset.Age.value_counts()[:3]
```

```
32    11
35     9
19     8
Name: Age, dtype: int64
```

```
[27] AnnualIncome_32=0
      AnnualIncome_35=0
```

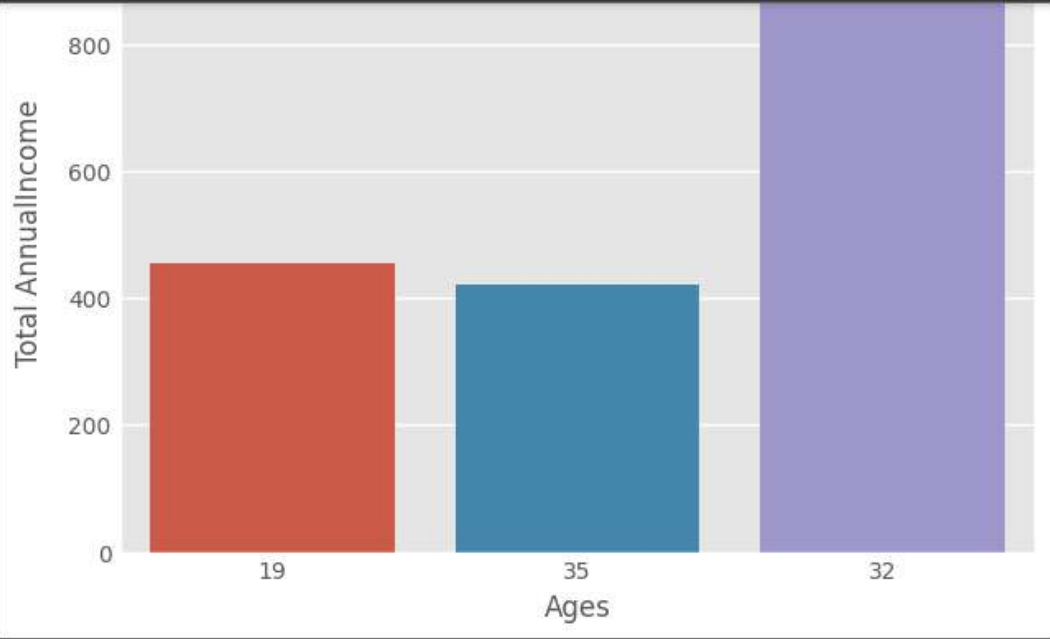
3s completed at 2:56 PM

```
+ Code + Text
```

```
AnnualIncome_32=0
AnnualIncome_35=0
AnnualIncome_19=0
AnnualIncome_32=sum(dataset[dataset['Age']==32].Income)
AnnualIncome_35=sum(dataset[dataset['Age']==35].Income)
AnnualIncome_19=sum(dataset[dataset['Age']==19].Income)
plt.title('First Popular Ages of the Customers')
sns.barplot(x=['19','35','32'],y=[AnnualIncome_19,AnnualIncome_35,AnnualIncome_32])
plt.xlabel('Ages')
plt.ylabel('Total AnnualIncome')
plt.show()
```



3s completed at 2:56 PM



```
[28] dataset.Gender.unique()  
  
array(['Male', 'Female'], dtype=object)
```

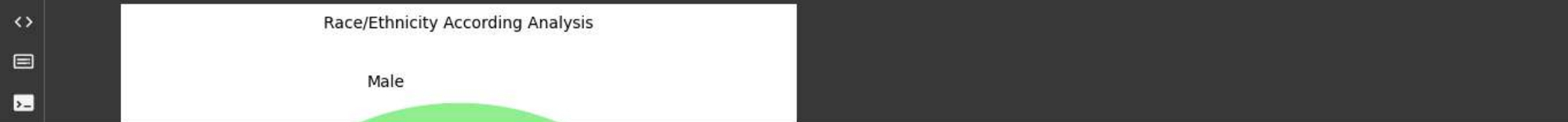
3s completed at 2:56 PM

+ Code + Text

```
[28] dataset.Gender.unique()  
  
array(['Male', 'Female'], dtype=object)
```

```
dataset.Gender.value_counts()  
  
Female    112  
Male      88  
Name: Gender, dtype: int64
```

```
[31] labels=dataset.Gender.unique()  
      colors=['lightgreen','red']  
      explode=[0,0.1]  
      values=dataset.Gender.value_counts().values  
      plt.figure(figsize=(7,7))  
      plt.pie(values,explode=explode,labels=labels,colors=colors,autopct='%1.1f%%')  
      plt.title('Race/Ethnicity According Analysis',color='black',fontsize=10)  
      plt.show()
```

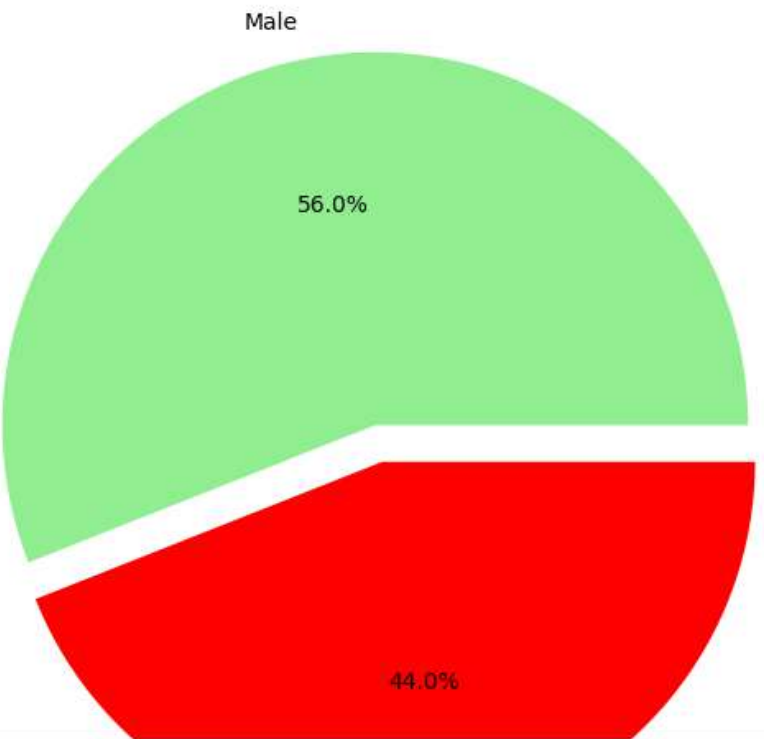


3s completed at 2:56 PM

+ Code + Text

✓ [31]

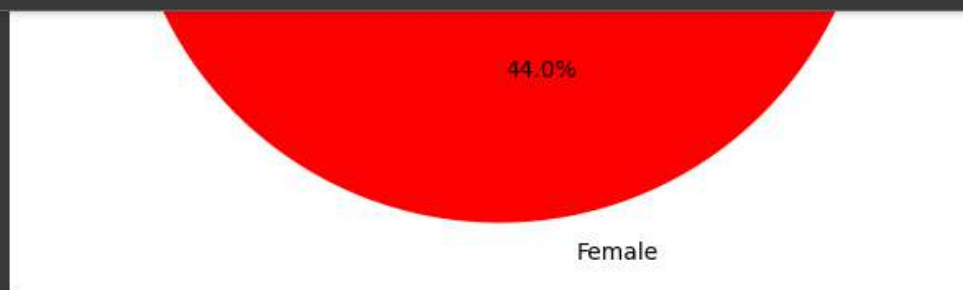
Race/Ethnicity According Analysis



✓ 3s completed at 2:56 PM

+ Code + Text

✓ [31]



✓ [32]

```
#Male Age
print('Max :',max(dataset[dataset['Gender']=='Male'].Age))
print('Min :',min(dataset[dataset['Gender']=='Male'].Age))
print('Mean :',np.mean(dataset[dataset['Gender']=='Male'].Age))
print('Std :',np.std(dataset[dataset['Gender']=='Male'].Age))
```

```
Max : 70
Min : 18
Mean : 39.80681818181818
Std : 15.426407372472944
```

✓ [33]

```
#Female Age
print('Max :',max(dataset[dataset['Gender']=='Female'].Age))
print('Min :',min(dataset[dataset['Gender']=='Female'].Age))
print('Mean :',np.mean(dataset[dataset['Gender']=='Female'].Age))
```

✓ 3s completed at 2:56 PM

+ Code + Text

RAM Disk

```
[32] Max : 70
      Min : 18
      Mean : 39.80681818181818
      Std : 15.426407372472944
```

```
#Female Age
print('Max :',max(dataset[dataset['Gender']=='Female'].Age))
print('Min :',min(dataset[dataset['Gender']=='Female'].Age))
print('Mean :',np.mean(dataset[dataset['Gender']=='Female'].Age))
print('Std :',np.std(dataset[dataset['Gender']=='Female'].Age))
```

```
Max : 68
Min : 18
Mean : 38.098214285714285
Std : 12.587522039581055
```

```
[34] print('Max :',max(dataset[dataset['Gender']=='Male'].Income))
      print('Min :',min(dataset[dataset['Gender']=='Male'].Income))
      print('Mean :',np.mean(dataset[dataset['Gender']=='Male'].Income))
      print('Std :',np.std(dataset[dataset['Gender']=='Male'].Income))
```

```
Max : 137
Min : 15
Mean : 62.22727272727273
Std : 26.486586344758496
```

3s completed at 2:56 PM

+ Code + Text

RAM
Disk

```
[34] print('Max :',max(dataset[dataset['Gender']=='Male'].Income))
      print('Min :',min(dataset[dataset['Gender']=='Male'].Income))
      print('Mean :',np.mean(dataset[dataset['Gender']=='Male'].Income))
      print('Std :',np.std(dataset[dataset['Gender']=='Male'].Income))
```

```
Max : 137
Min : 15
Mean : 62.22727272727273
Std : 26.486586344758496
```

```
[35] print('Max :',max(dataset[dataset['Gender']=='Female'].Income))
      print('Min :',min(dataset[dataset['Gender']=='Female'].Income))
      print('Mean :',np.mean(dataset[dataset['Gender']=='Female'].Income))
      print('Std :',np.std(dataset[dataset['Gender']=='Female'].Income))
```

```
Max : 126
Min : 16
Mean : 59.25
Std : 25.8955663607278
```

```
[36] #Male SpendingScore
      print('Max :',max(dataset[dataset['Gender']=='Male'].Score))
      print('Min :',min(dataset[dataset['Gender']=='Male'].Score))
      print('Mean :',np.mean(dataset[dataset['Gender']=='Male'].Score))
      print('Std :',np.std(dataset[dataset['Gender']=='Male'].Score))
```

3s completed at 2:56 PM

```
+ Code + Text

0s [35] Mean : 59.25
Std : 25.8955663607278

[36] #Male SpendingScore
print('Max :',max(dataset[dataset['Gender']=='Male'].Score))
print('Min :',min(dataset[dataset['Gender']=='Male'].Score))
print('Mean :',np.mean(dataset[dataset['Gender']=='Male'].Score))
print('Std :',np.std(dataset[dataset['Gender']=='Male'].Score))

Max : 97
Min : 1
Mean : 48.51136363636363
Std : 27.737812359739767

[37] #Female SpendingScore
print('Max :',max(dataset[dataset['Gender']=='Female'].Score))
print('Min :',min(dataset[dataset['Gender']=='Female'].Score))
print('Mean :',np.mean(dataset[dataset['Gender']=='Female'].Score))
print('Std :',np.std(dataset[dataset['Gender']=='Female'].Score))

Max : 99
Min : 5
Mean : 51.526785714285715
Std : 24.00705246880166
```

✓ 3s completed at 2:56 PM

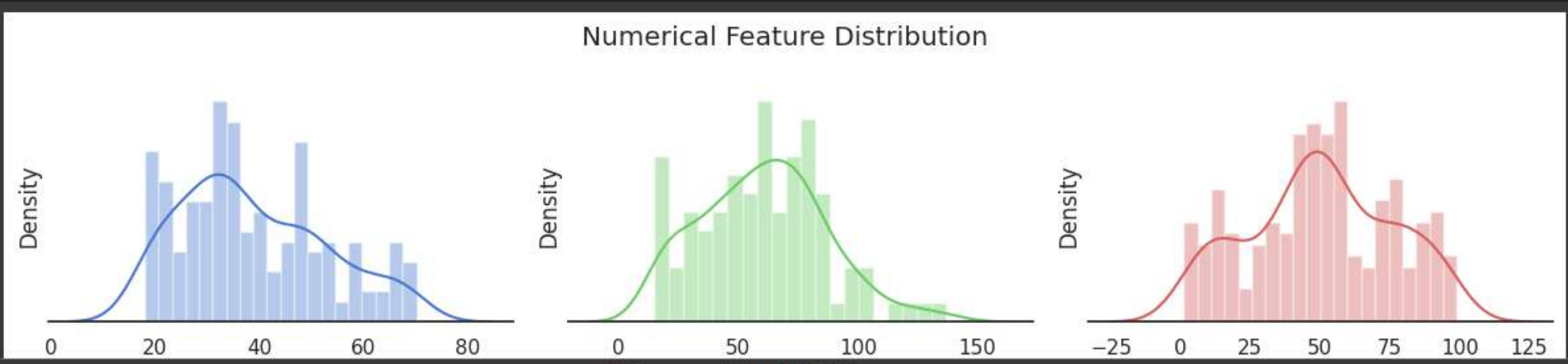
```
[38] from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(dataset['Gender'].values)
dataset['Gender'] = integer_encoded
dataset.head(7)
```

	Gender	Age	Income	Score	Age Range
0	1	19	15	39	Young
1	1	21	15	81	Young
2	0	20	16	6	Young
3	0	23	16	77	Adult
4	0	31	17	40	Adult
5	0	22	17	76	Adult
6	0	35	18	6	Adult

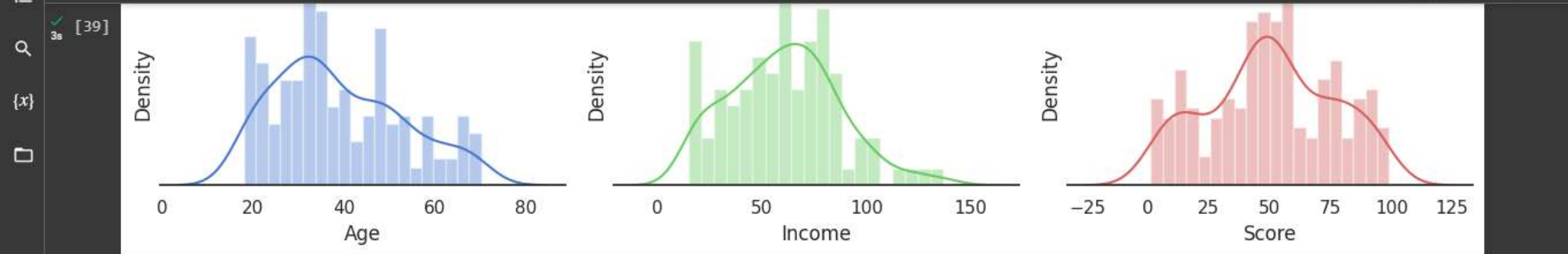
```
[39] sns.set(style='white', palette='muted', color_codes=True)
fig, axs = plt.subplots(1, 3, figsize=(12, 3))
sns.despine(left=True)
axs[0] = sns.distplot(dataset['Age'], bins=20, ax=axs[0])
axs[1] = sns.distplot(dataset['Income'], bins=20, ax=axs[1], color='g')
```

3s completed at 2:56 PM

```
[39] sns.set(style='white', palette='muted', color_codes=True)
fig, axs = plt.subplots(1, 3, figsize=(12, 3))
sns.despine(left=True)
axs[0] = sns.distplot(dataset['Age'], bins=20, ax=axs[0])
axs[1] = sns.distplot(dataset['Income'], bins=20, ax=axs[1], color='g')
axs[2] = sns.distplot(dataset['Score'], bins=20, ax=axs[2], color='r')
fig.suptitle('Numerical Feature Distribution')
plt.setp(axs, yticks=[])
plt.tight_layout()
plt.show()
```



3s completed at 2:56 PM



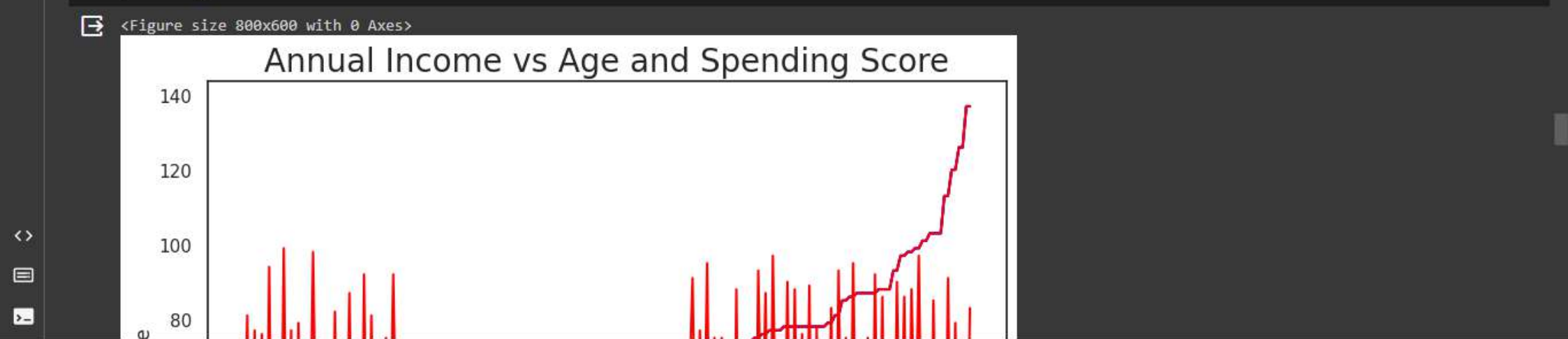
```
[42] plt.figure(figsize=(8,6))
sns.distplot(dataset['Income'],bins=5,color='brown')

<Axes: xlabel='Income', ylabel='Density'>
```

```
[45] x = dataset['Income']
y = dataset['Age']
z = dataset['Score']

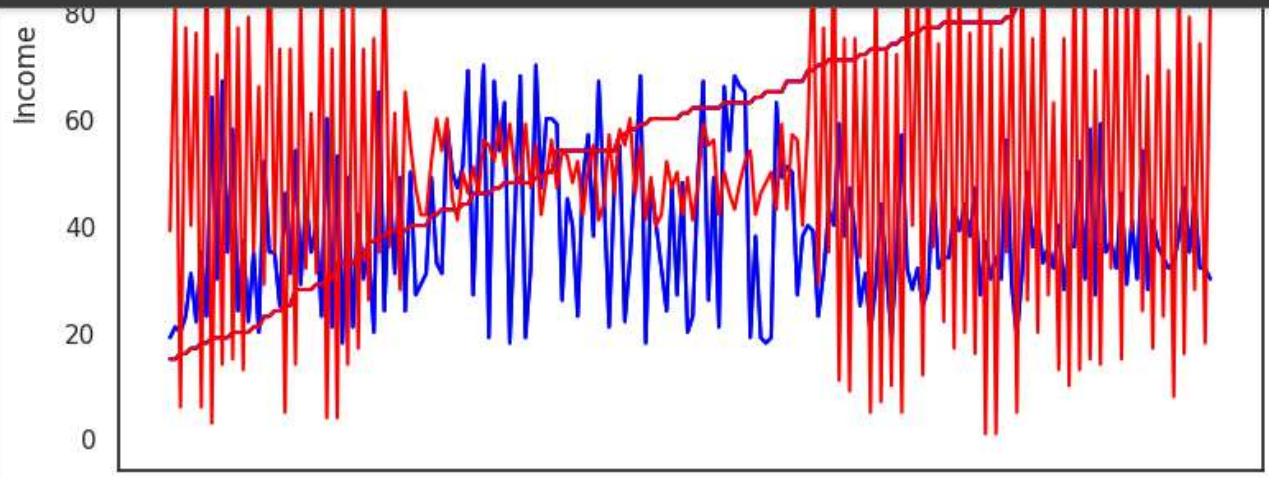
sns.lineplot(x, color = 'blue')
sns.lineplot(y,color='blue')
sns.lineplot(x, color = 'red')
```

```
x = dataset['Income']  
y = dataset['Age']  
z = dataset['Score']  
  
sns.lineplot(x, color = 'blue')  
sns.lineplot(y,color='blue')  
sns.lineplot(x, color = 'red')  
sns.lineplot(z,color='red')  
plt.title('Annual Income vs Age and Spending Score', fontsize = 20)  
plt.show()
```



+ Code + Text

2s [45]



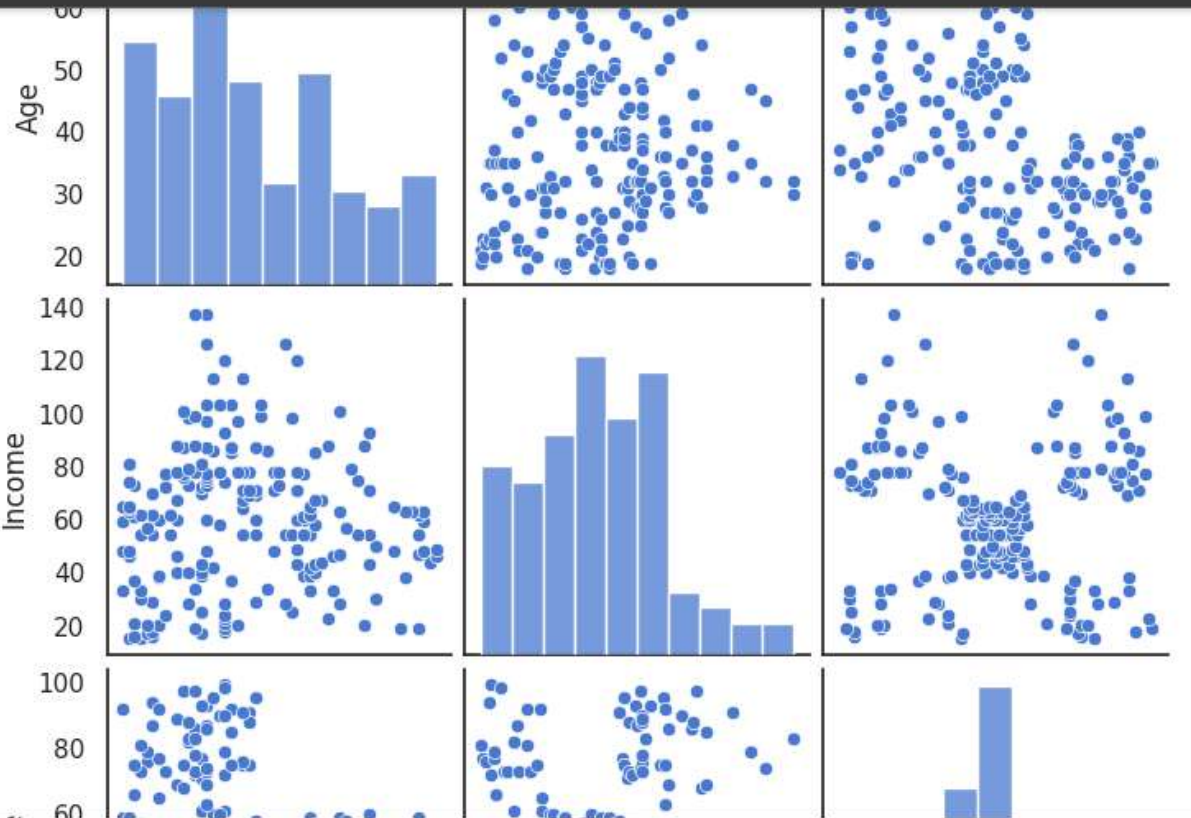
0.24641111111111112

4s

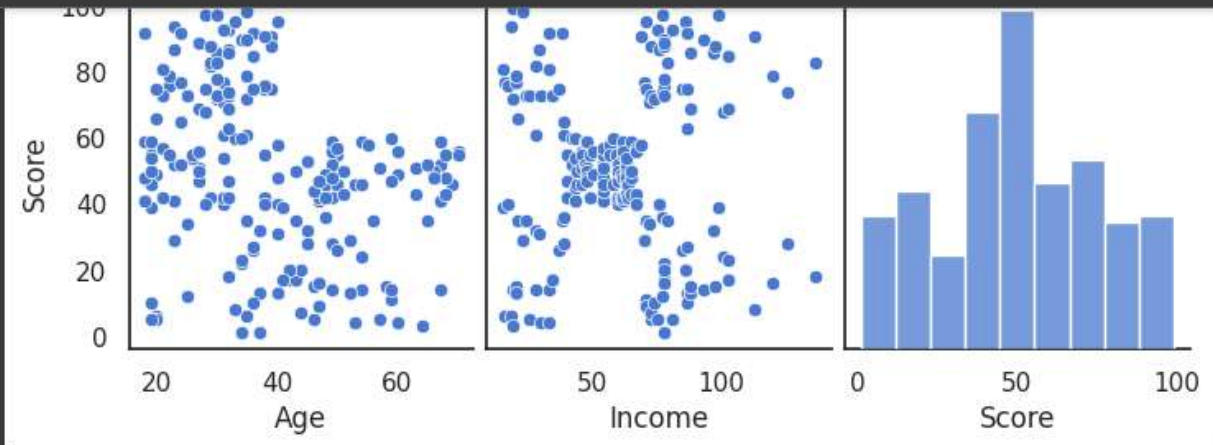
```
[46] sns.pairplot(dataset[['Age', 'Income', 'Score']])  
plt.show()
```



3s completed at 2:56 PM



3s completed at 2:56 PM



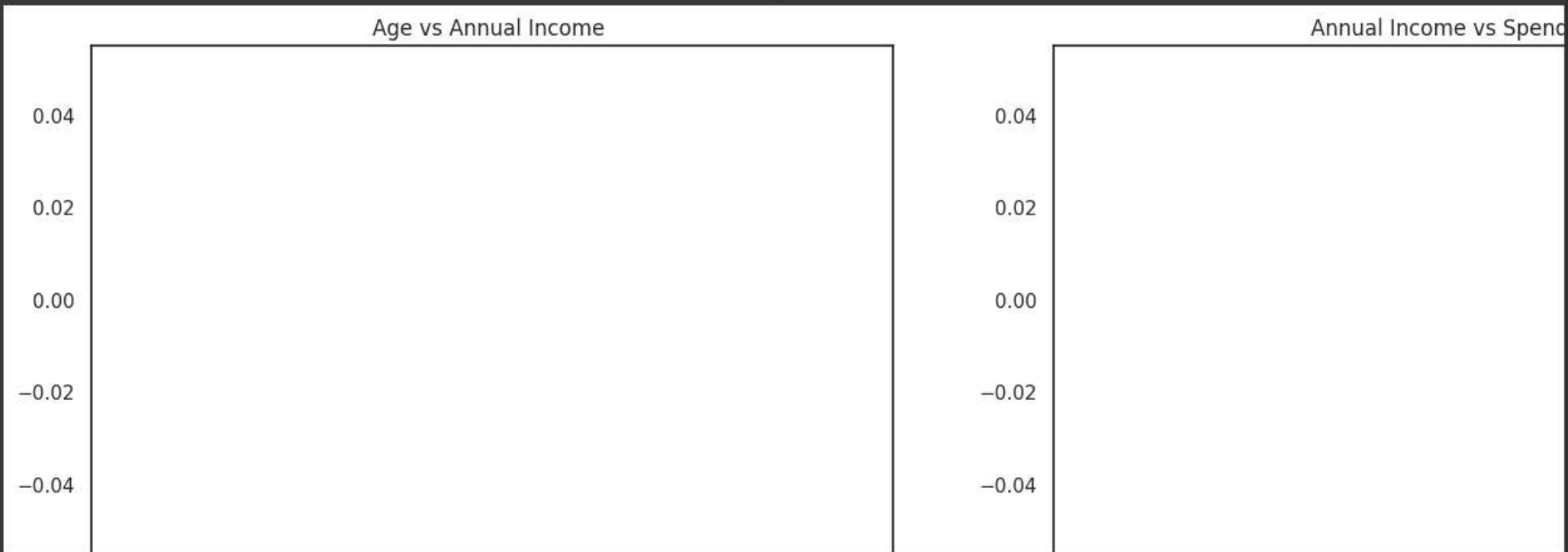
```
plt.figure(figsize=(25,5))
plt.subplot(1,3,1)
for gender in ['Male', 'Female']:
    plt.scatter(x = 'Age', y = 'Income', data = dataset[dataset['Gender'] == gender], label = gender, s=100)
    plt.title('Age vs Annual Income')
plt.subplot(1,3,2)
for gender in ['Male', 'Female']:
    plt.scatter(x = 'Income', y = 'Score', data = dataset[dataset['Gender'] == gender], s = 100, label = gender)
    plt.title('Annual Income vs Spending')
plt.legend()
```

3s completed at 2:56 PM

+ Code + Text

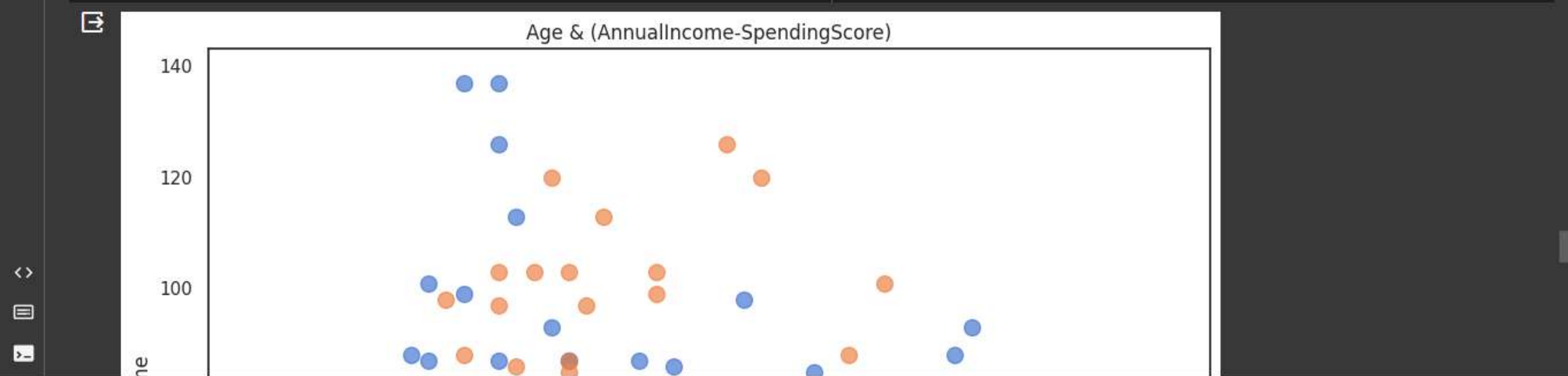


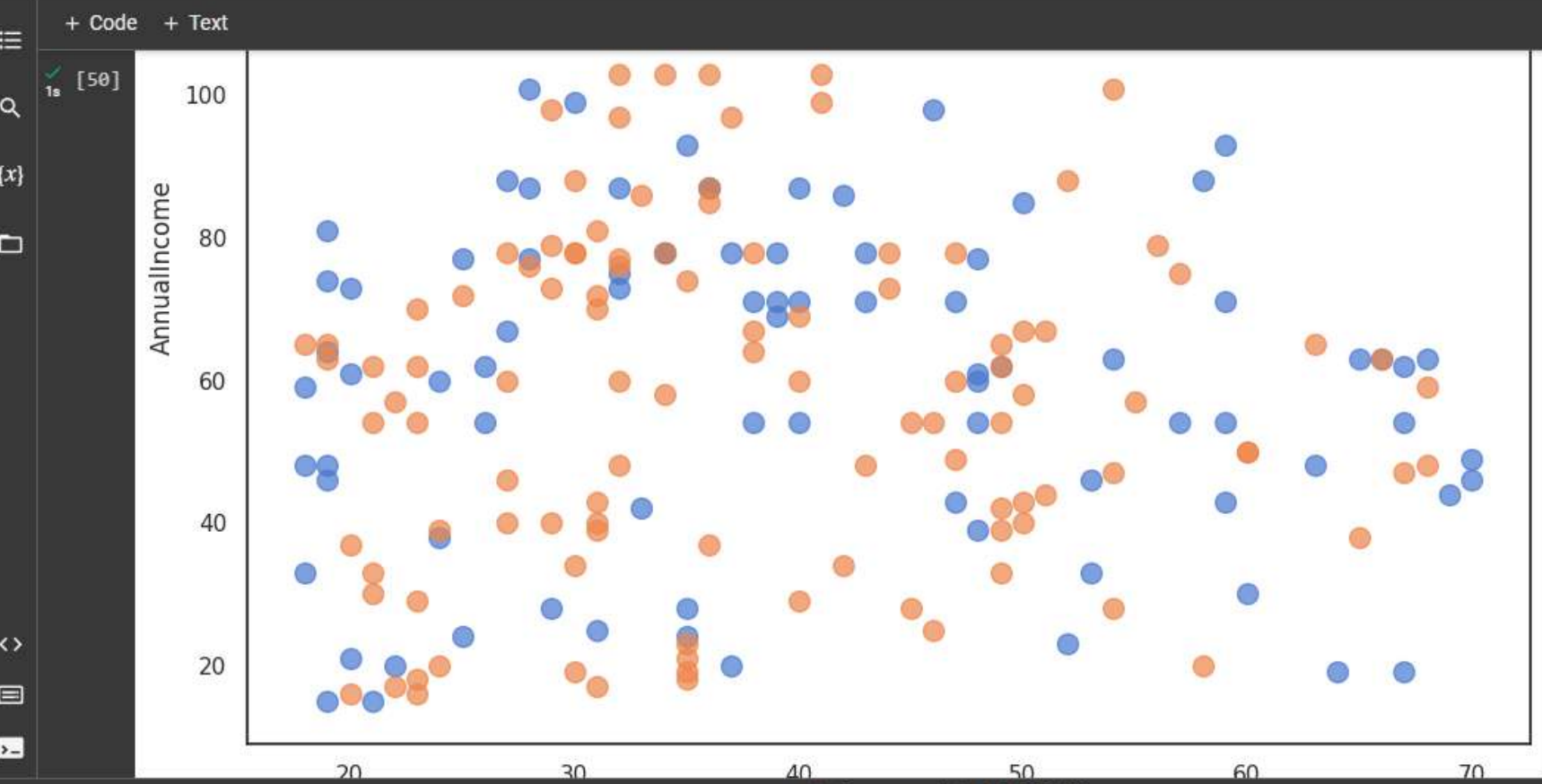
```
plt.legend()  
plt.show()
```



✓ 3s completed at 2:56 PM

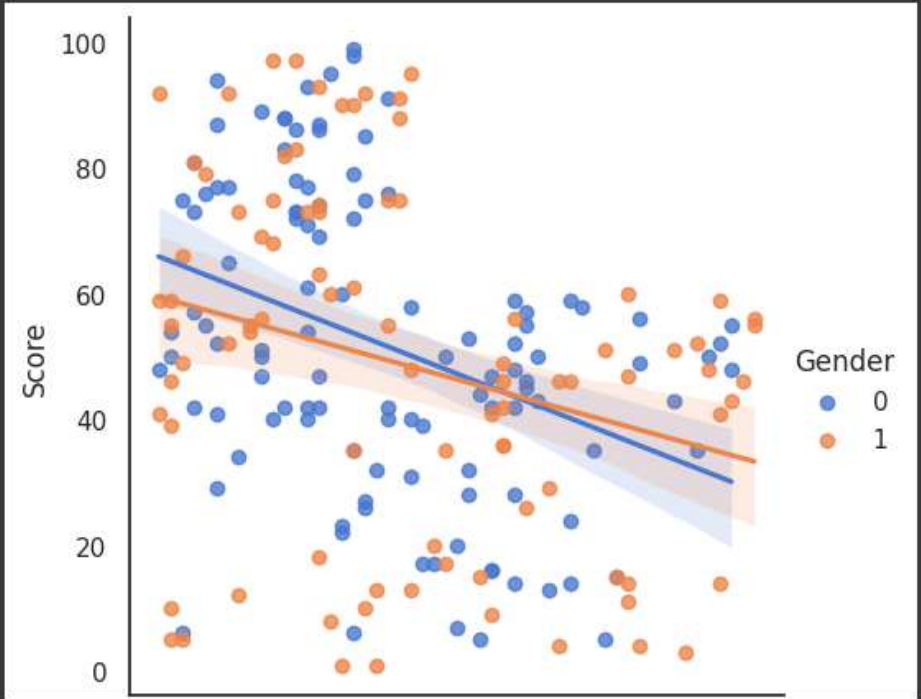
```
plt.figure(figsize=(10,8))
for gender in dataset.Gender.unique():
    plt.scatter(x='Age',y='Income',data=dataset[dataset['Gender']==gender],s=100,alpha=.7)
    plt.xlabel('Age')
    plt.ylabel('AnnualIncome')
    plt.title('Age & (AnnualIncome-SpendingScore)')
plt.show()
```





✓ 3s completed at 2:56 PM


```
sns.lmplot(x='Age', y='Score', data=dataset, fit_reg=True, hue='Gender')  
plt.show()
```



3s completed at 2:56 PM



Assignment-V.ipynb ☆

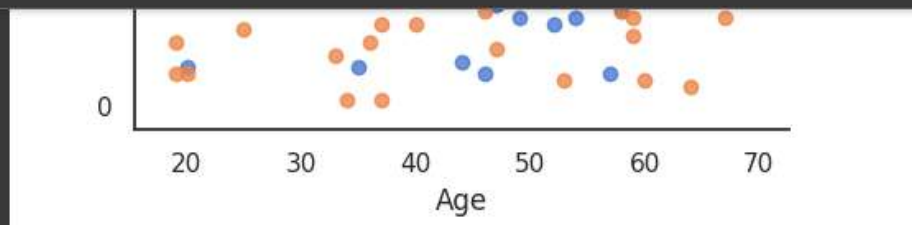
File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings

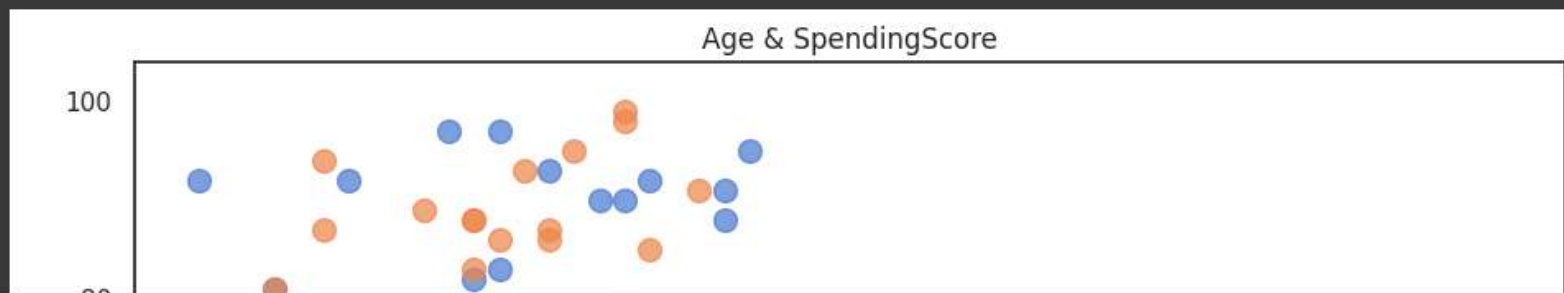
+ Code + Text

RAM Disk

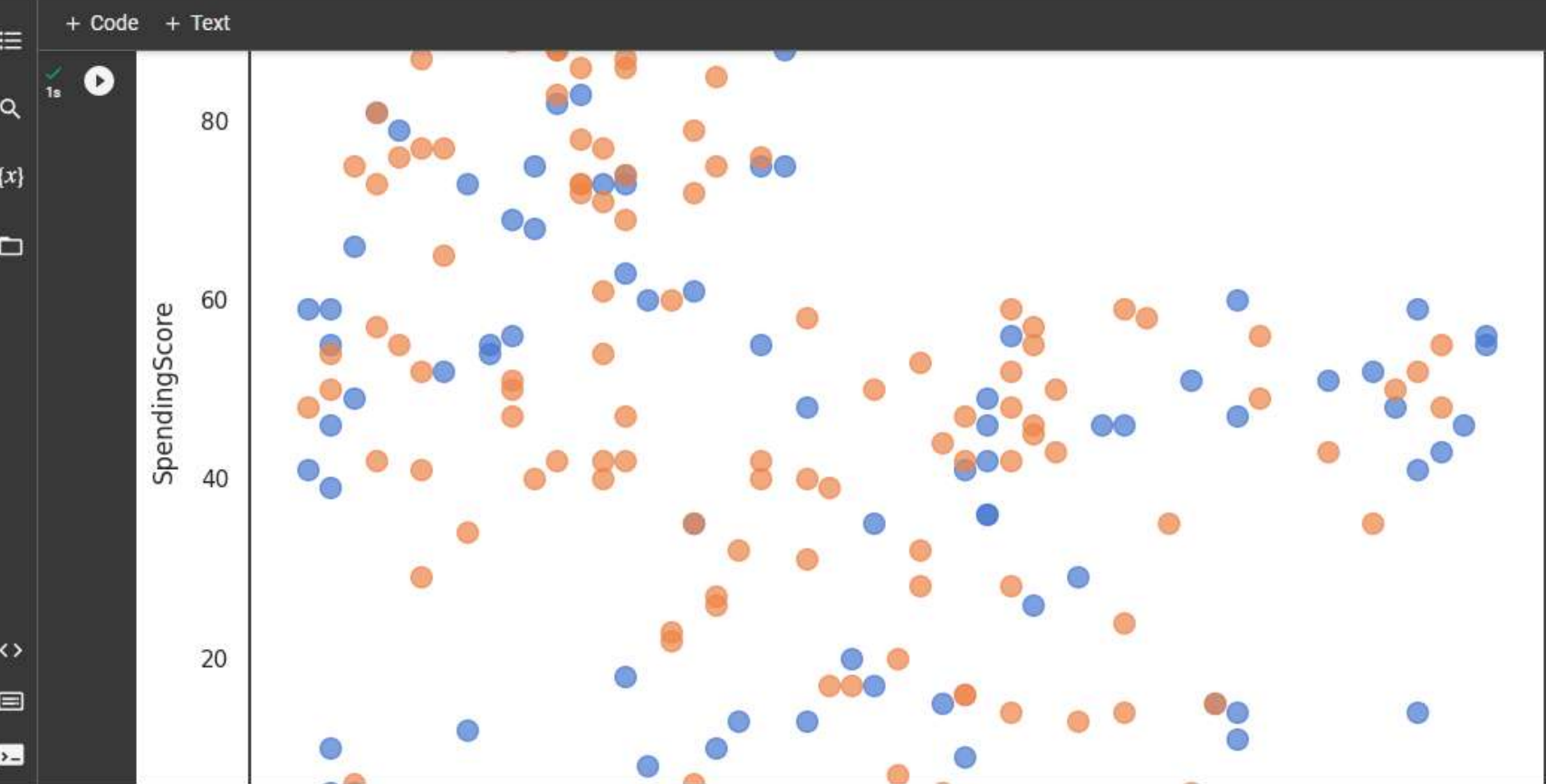
2s



```
[52] plt.figure(figsize=(10,8))
      for gender in dataset.Gender.unique():
        plt.scatter(x='Age',y='Score',data=dataset[dataset['Gender']==gender],s=100,alpha=.7)
        plt.xlabel('Age')
        plt.ylabel('SpendingScore')
        plt.title('Age & SpendingScore')
      plt.show()
```



3s completed at 2:56 PM



3s completed at 2:56 PM

+ Code + Text

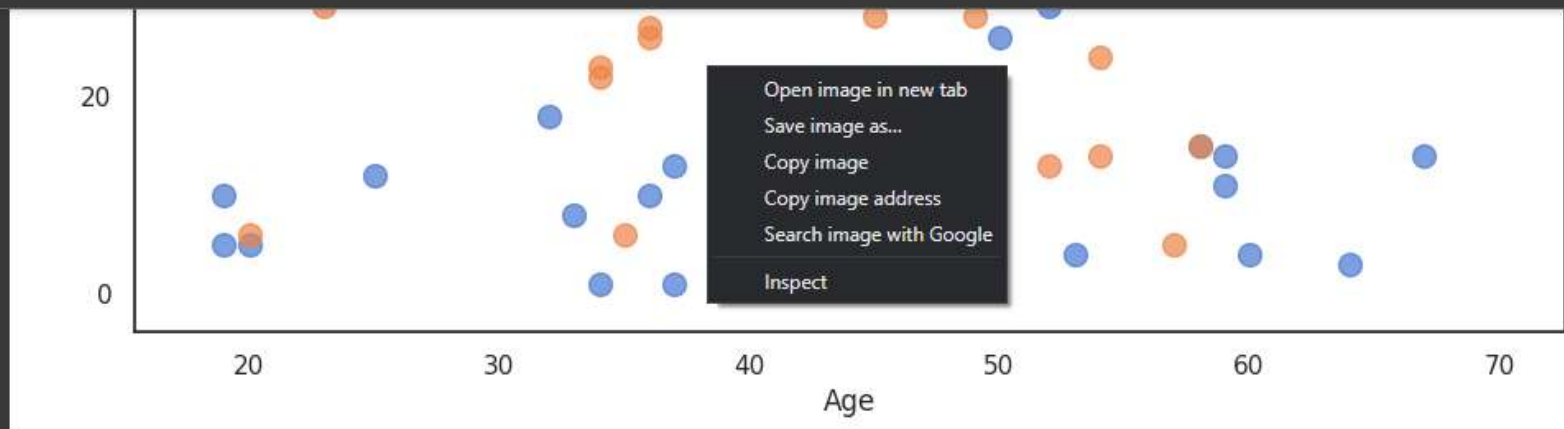


1s

{x}



<>



RAM
Disk



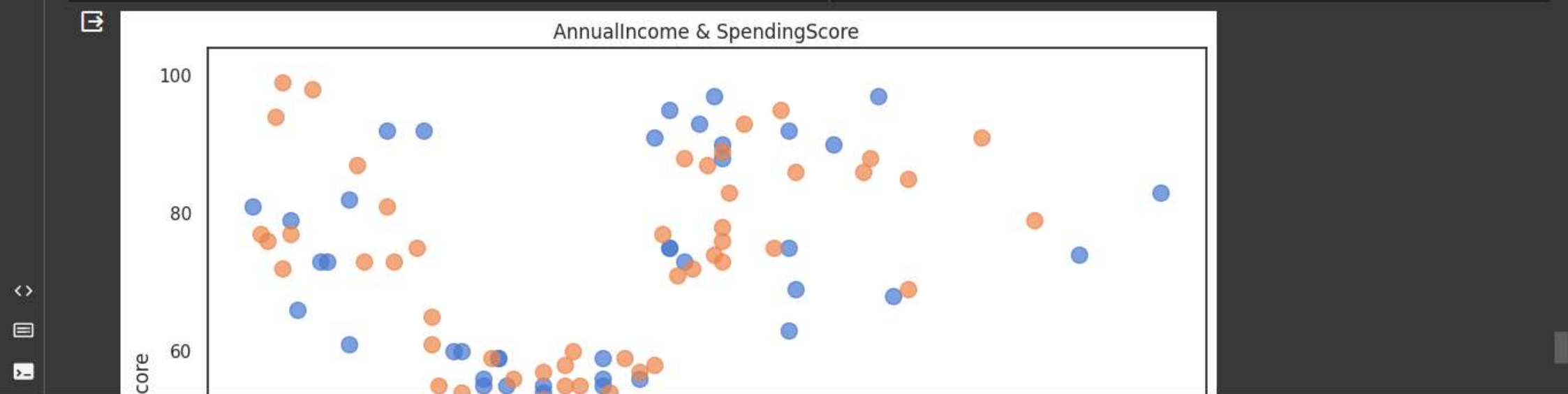
```
[53] plt.figure(figsize=(10,8))  
for gender in dataset.Gender.unique():  
    plt.scatter(x='Income',y='Score',data=dataset[dataset['Gender']==gender],s=100,alpha=.7)  
    plt.xlabel('AnnualIncome')  
    plt.ylabel('SpendingScore')  
    plt.title('AnnualIncome & SpendingScore')  
plt.show()
```

AnnualIncome & SpendingScore

3s completed at 2:56 PM


```
+ Code + Text
```

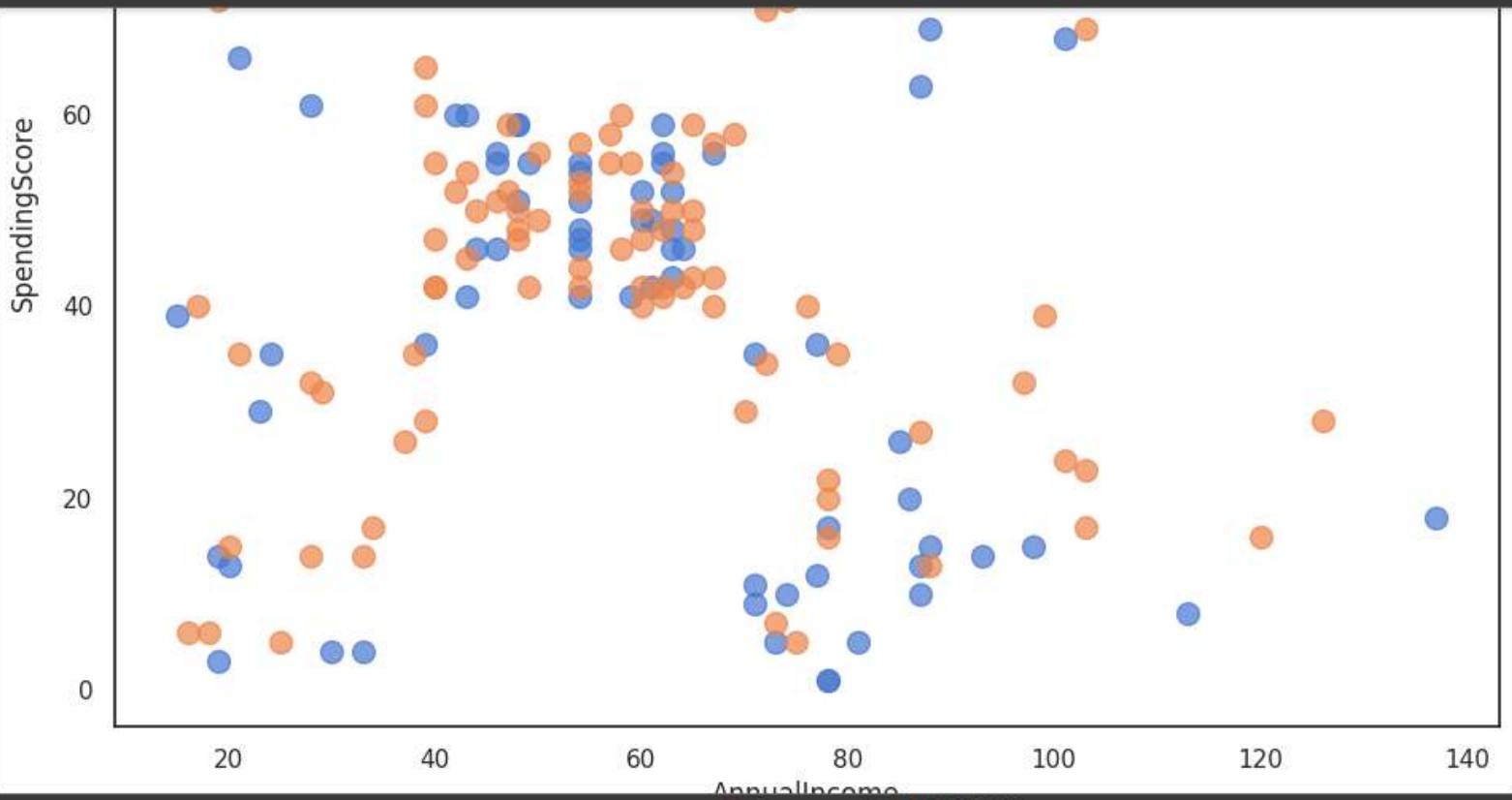
```
1s ▶ plt.figure(figsize=(10,8))
    for gender in dataset.Gender.unique():
        plt.scatter(x='Income',y='Score',data=dataset[dataset['Gender']==gender],s=100,alpha=.7)
        plt.xlabel('AnnualIncome')
        plt.ylabel('SpendingScore')
        plt.title('AnnualIncome & SpendingScore')
    plt.show()
```



3s completed at 2:56 PM

+ Code + Text

✓ 1s [53]



✓ 3s completed at 2:56 PM

