```
# Sukanth K
# 21BRS1617

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import silhouette_score
from sklearn import cluster

# reading the data

df = pd.read_csv('/content/Mall_Customers.csv')
df.head()
```

	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

```
df.describe()
print(df.info())
```

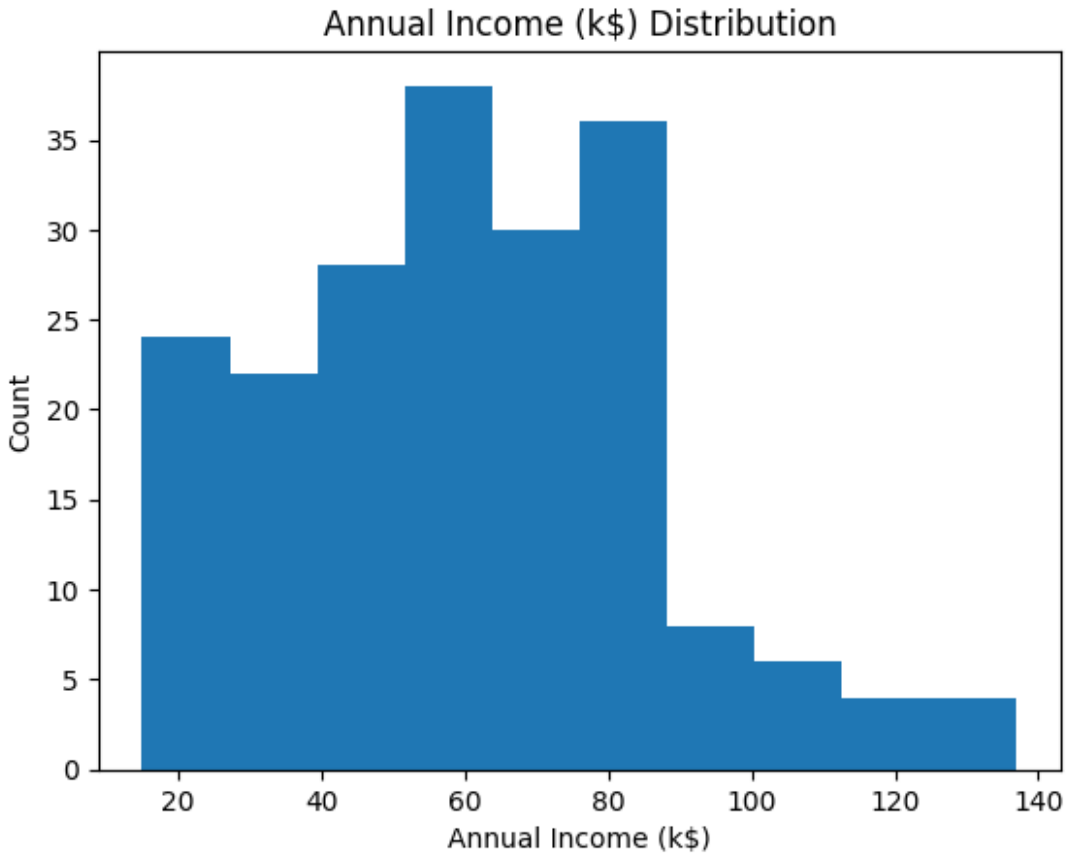
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None
```

```
#genre instead of gender

df['Gender'].value_counts()
```

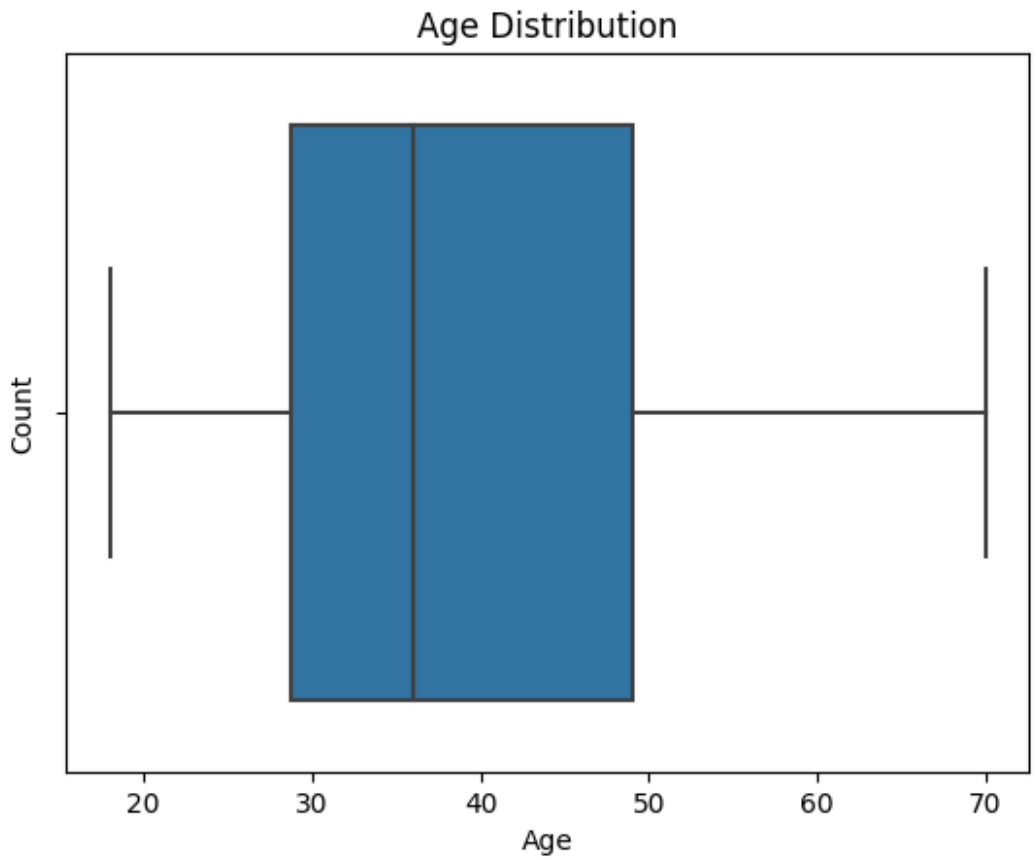
```
Female    112
Male       88
Name: Gender, dtype: int64
```

```
# remove unique columns

df.drop(columns=['CustomerID'], axis=1, inplace=True)
```

```
# plotting the data
#univariate visualisation
# for Annual Income (k$)

plt.hist(df['Annual Income (k$)'], bins=10)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Count')
plt.title('Annual Income (k$) Distribution')
plt.show()
```

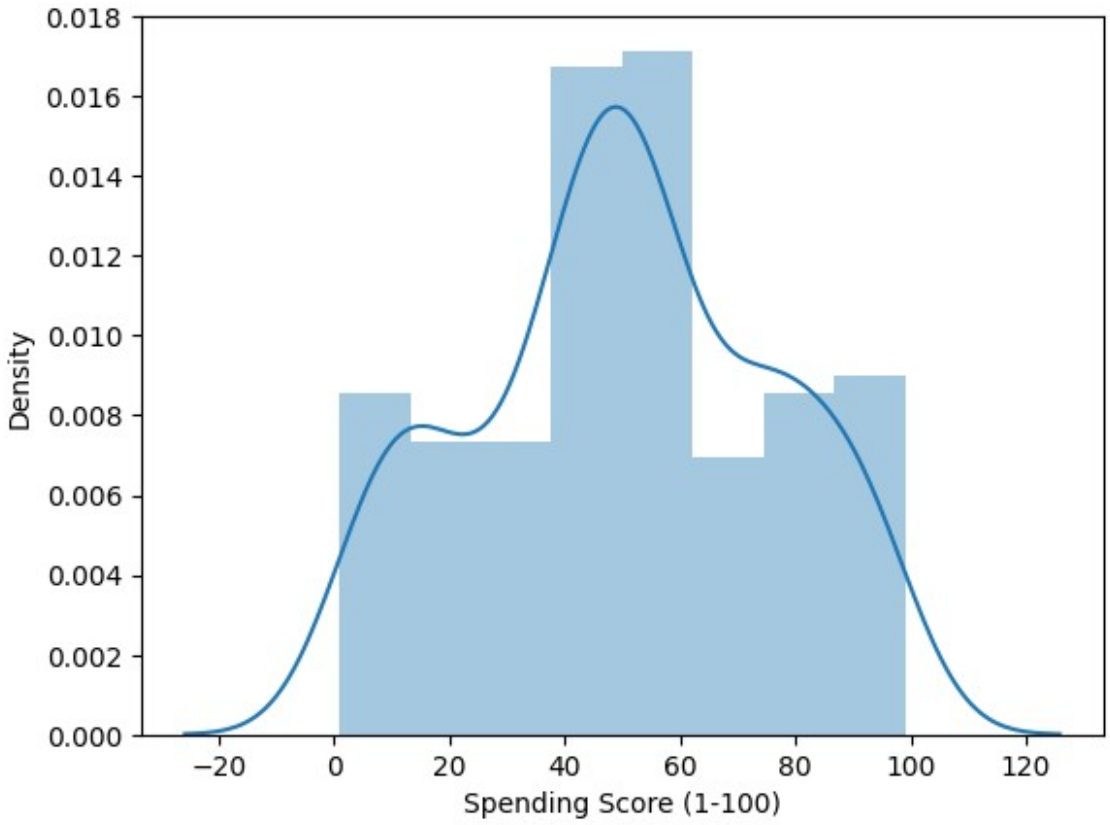



```
# plotting a box plot
sns.boxplot(x = df['Age'])
plt.ylabel('Count')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.show()
```

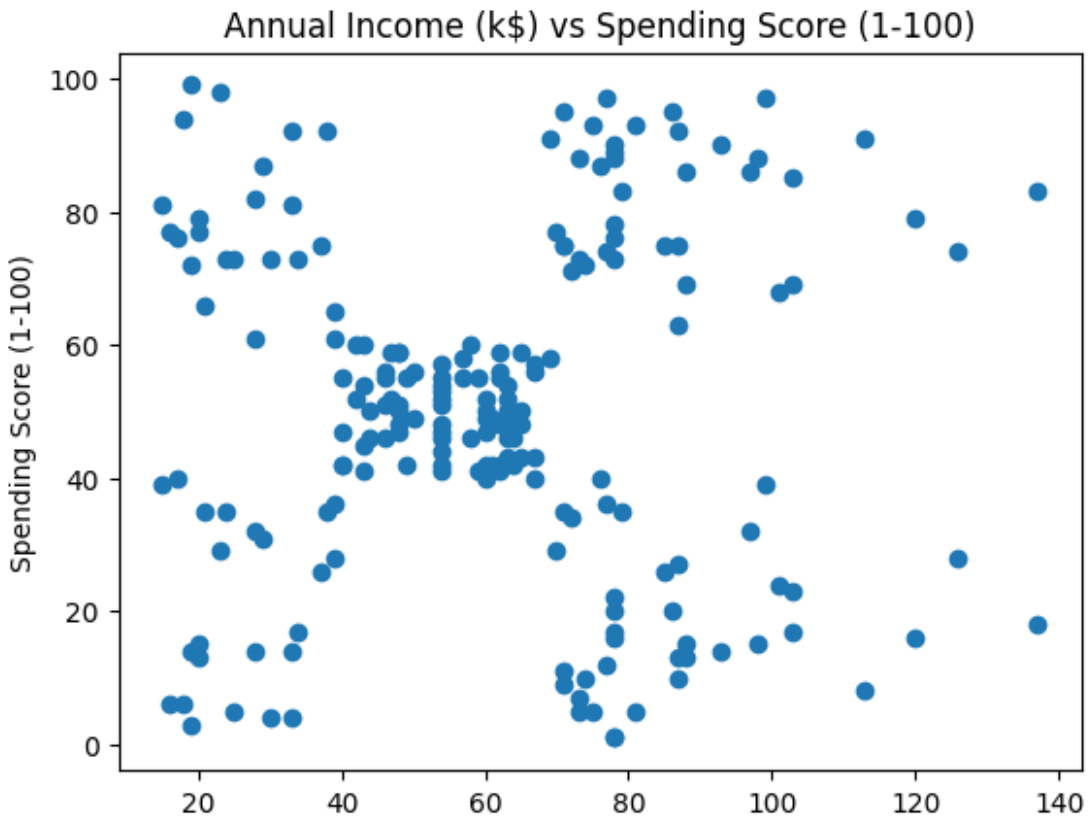



```
sns.distplot(df['Spending Score (1-100)'])
```

```
<ipython-input-19-beed7b40d5ab>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Spending Score (1-100)'])

<Axes: xlabel='Spending Score (1-100)', ylabel='Density'>
```

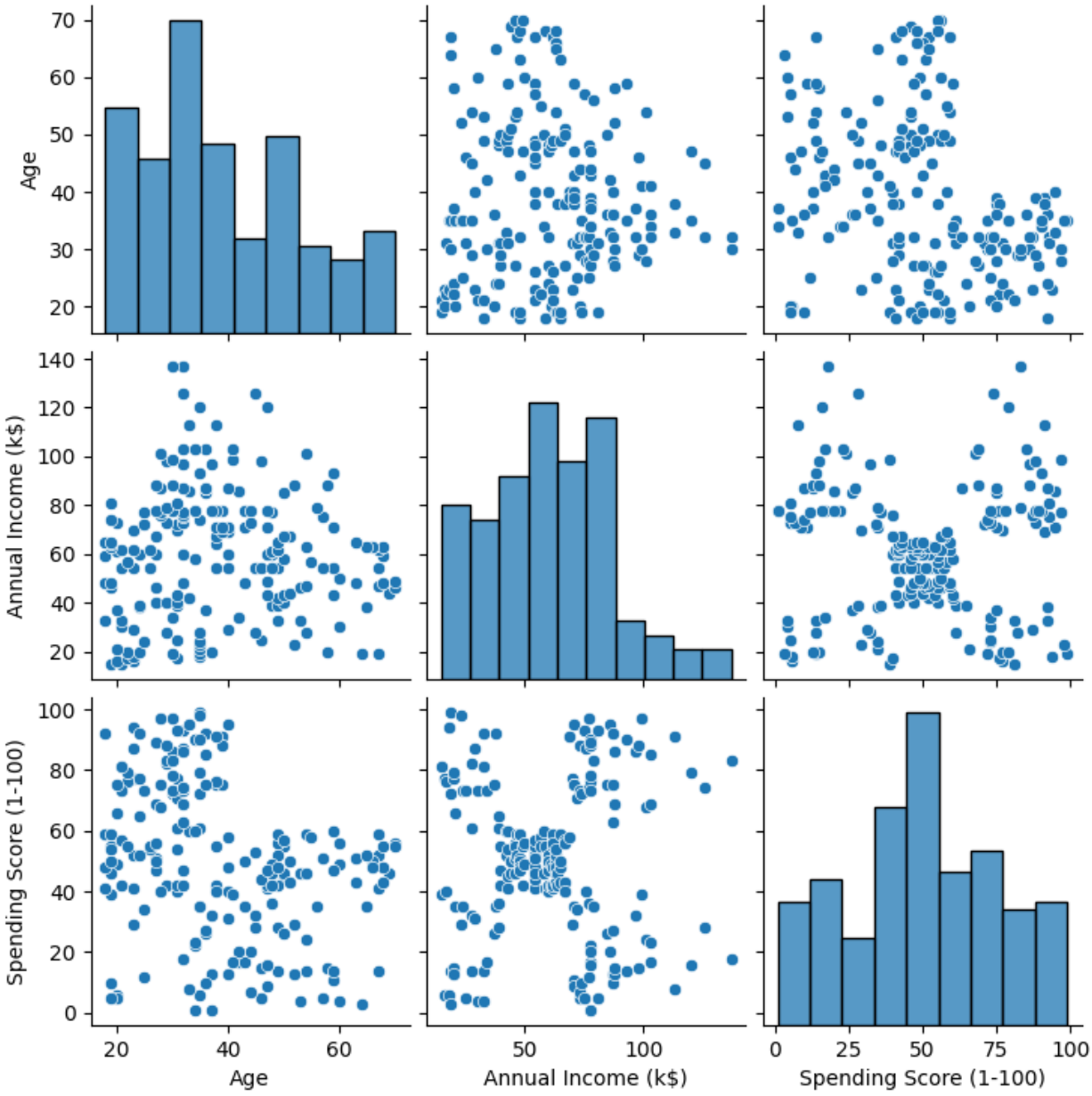



```
#bivariate visualization
#scatterplot of annual income and spending score
plt.scatter(data['Annual Income (k$)'],data['Spending Score (1-100)'])
plt.xlabel('Annual Income (k$)')
plt.scatter(df['Annual Income (k$)'],df['Spending Score (1-100)'])
plt.ylabel('Spending Score (1-100)')
plt.title('Annual Income (k$) vs Spending Score (1-100)')
plt.show()
```



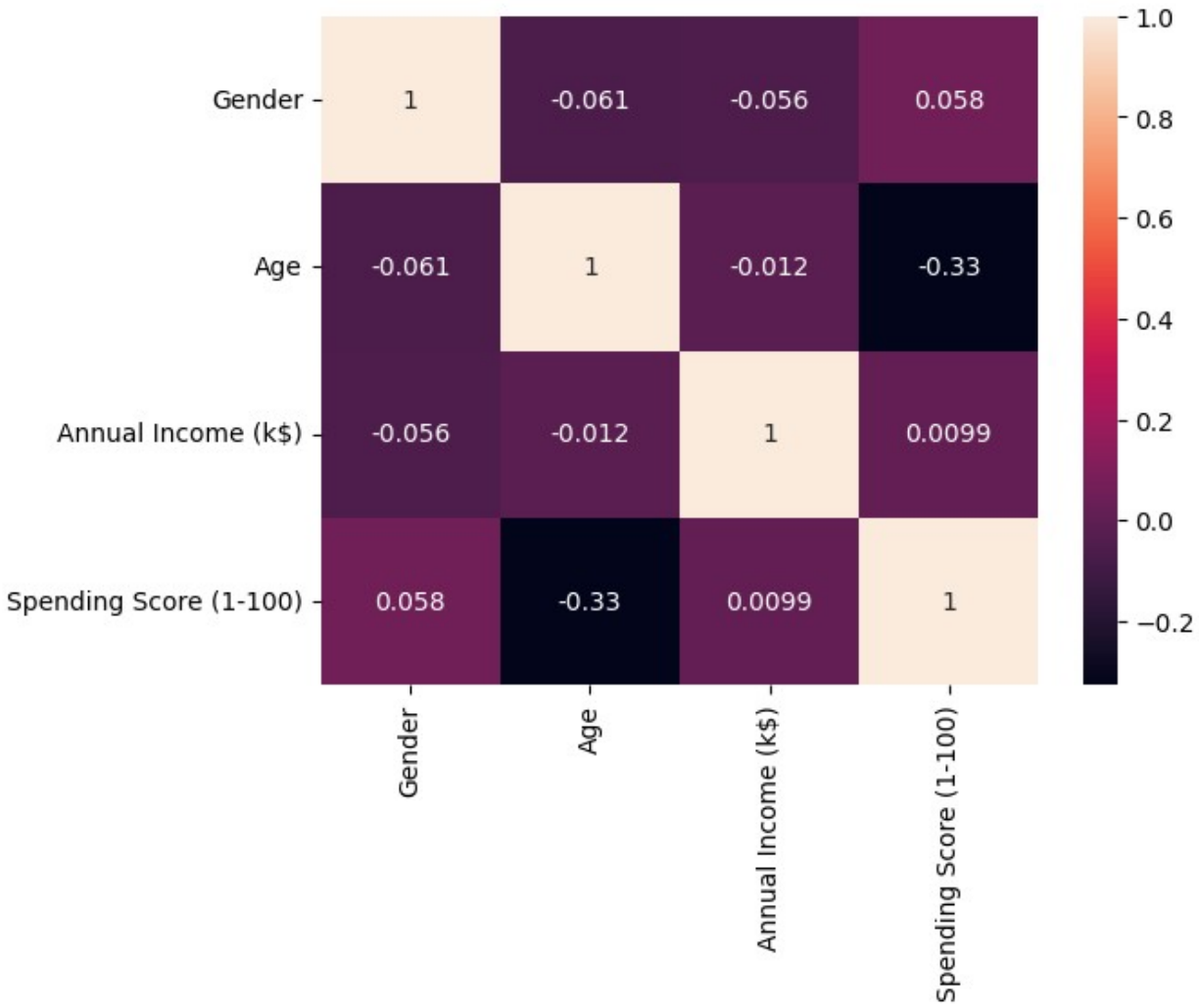

```
#multivariate visualization
#pairplot of all variables
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7ce7f0528310>
```




```
# convert Gender column to numerical representation
df['Gender'] = df['Gender'].replace({'Male': 0, 'Female': 1})

# correlation matrix
sns.heatmap(df.corr(), annot=True)
```

```
<Axes: >
```




```
#perform data preprocessing
#check for missing values
df.isnull().sum()
```

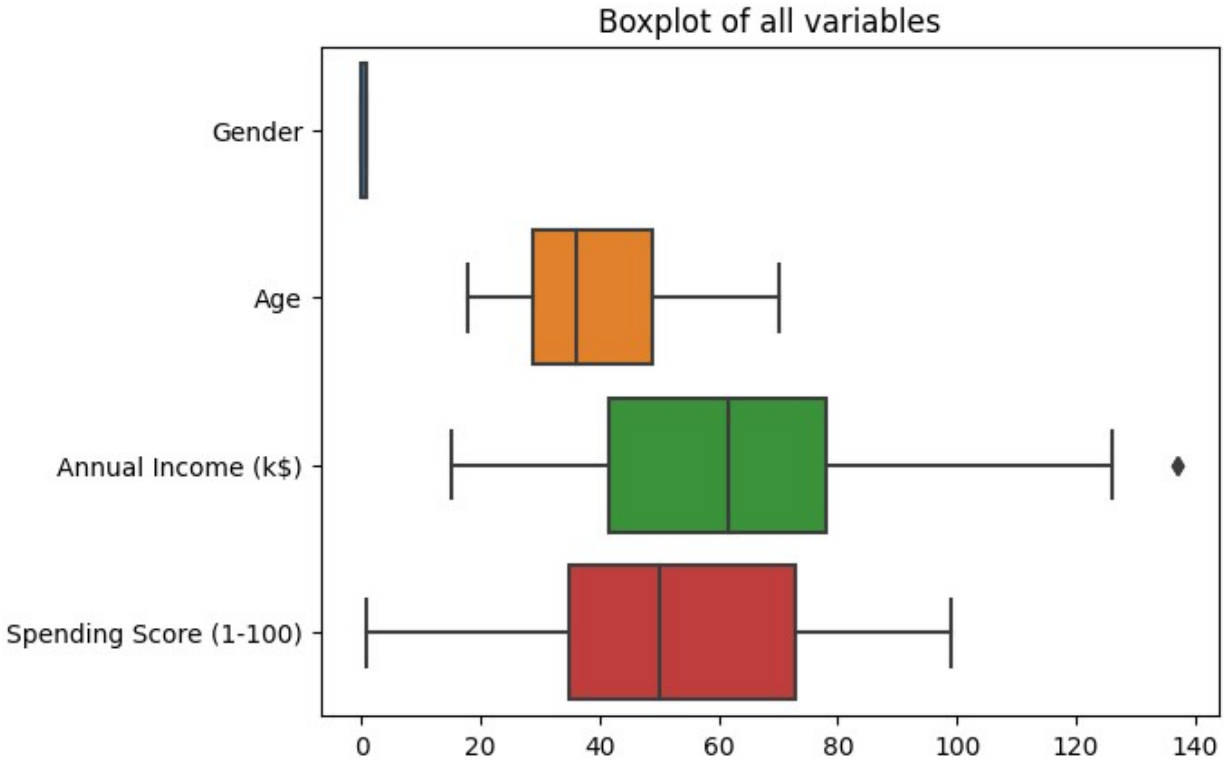
```
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
#no missing values
#check for shape
print(df.shape)
```

```
(200, 4)
```

```
sns.boxplot(data = df, orient = 'h')
#this is the boxplot of all the variables
```

```
plt.title('Boxplot of all variables')
plt.show()
```

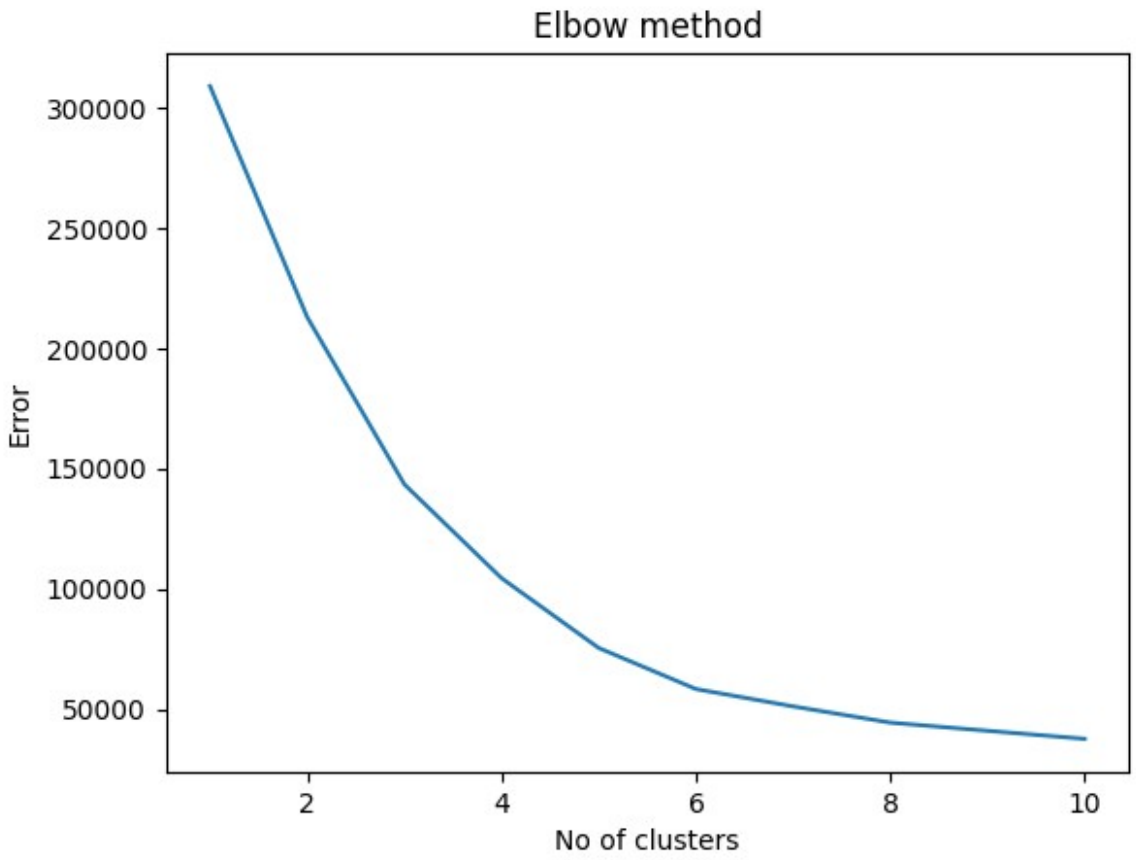



```
#perform clustering
error = []
for i in range(1, 11):
    kmeans = cluster.KMeans(n_clusters=i, init='k-means++',
random_state=0)
    kmeans.fit(df)
    error.append(kmeans.inertia_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/
_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
```

```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
```

```python
plt.plot(range(1, 11), error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
```



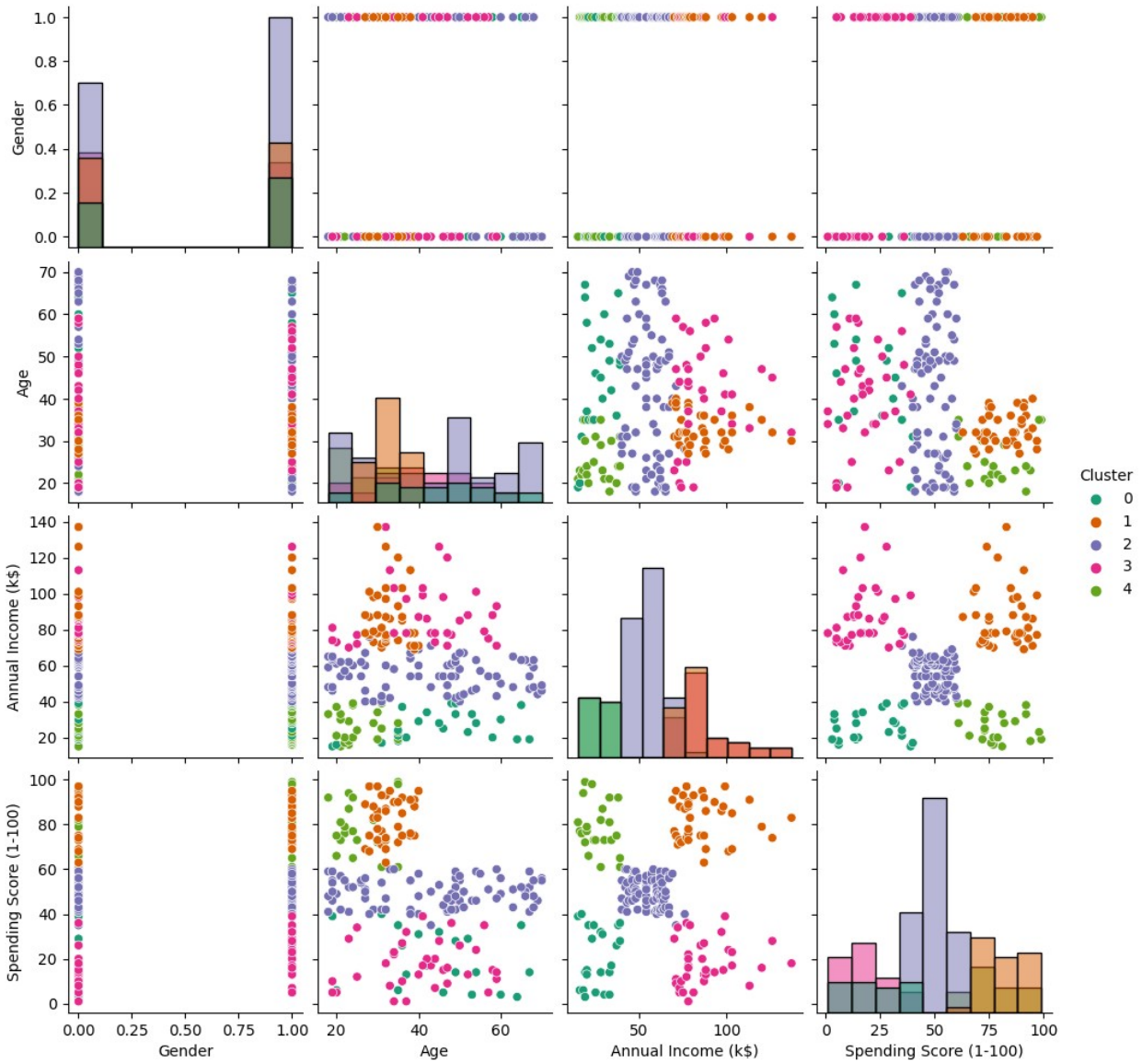

```
#taking 5 clusters
kmeans = cluster.KMeans(n_clusters=5, init='k-means++',
random_state=0)
kmeans.fit(df)
pred = kmeans.predict(df)
silhouette_score(df, pred)
silhouette_avg = silhouette_score(df, pred)
#print silhouette score
print("For n_clusters =", 5,"The average silhouette_score is :",
silhouette_avg)
pred
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/
_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(

For n_clusters = 5 The average silhouette_score is :
0.4440669204743008

array([0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
4,
```

```
       0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
4,
       0, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 3, 1, 2, 1, 3, 1, 3,
1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
1,
       3, 1], dtype=int32)
```

```
data_visualization = df.copy()
data_visualization['Cluster'] = pred
data_visualization.head()

columns_to_plot = ['Gender', 'Age', 'Annual Income (k$)', 'Spending
Score (1-100)', 'Cluster']

sns.pairplot(data_visualization[columns_to_plot], hue='Cluster',
palette='Dark2', diag_kind='hist')
```

```
<seaborn.axisgrid.PairGrid at 0x7ce7ec6b6770>
```




```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Perform K-means clustering on 'Annual Income' and 'Spending Score'
with 5 clusters
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=0)
kmeans.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
pred = kmeans.predict(df[['Annual Income (k$)','Spending Score (1-
100)']])

# Calculate silhouette score
silhouette_avg = silhouette_score(df[['Annual Income (k$)','Spending
Score (1-100)']], pred)

# Print silhouette score
```

```
print("For n_clusters =", 5, "The average silhouette_score is:",
silhouette_avg)
```

```
For n_clusters = 5 The average silhouette_score is: 0.553931997444648
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/
_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
```

```
#making a scatterplot of annual income and spending score
plt.scatter(df['Annual Income (k$)'],df['Spending Score (1-
100)'],c=pred)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Annual Income (k$) vs Spending Score (1-100)')
plt.show()
```