K-Means clustering (Unsupervised)

import required package

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

read data from source and describing

In [2]:
```
df = pd.read_csv('mall_customers.csv')
df.head()
```

Out[2]:

	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

In [3]:
```
print(df.columns)
```

```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [4]:
```
print(df.describe())
```

```
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```

In [5]:
```
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
```

check the relation between variables

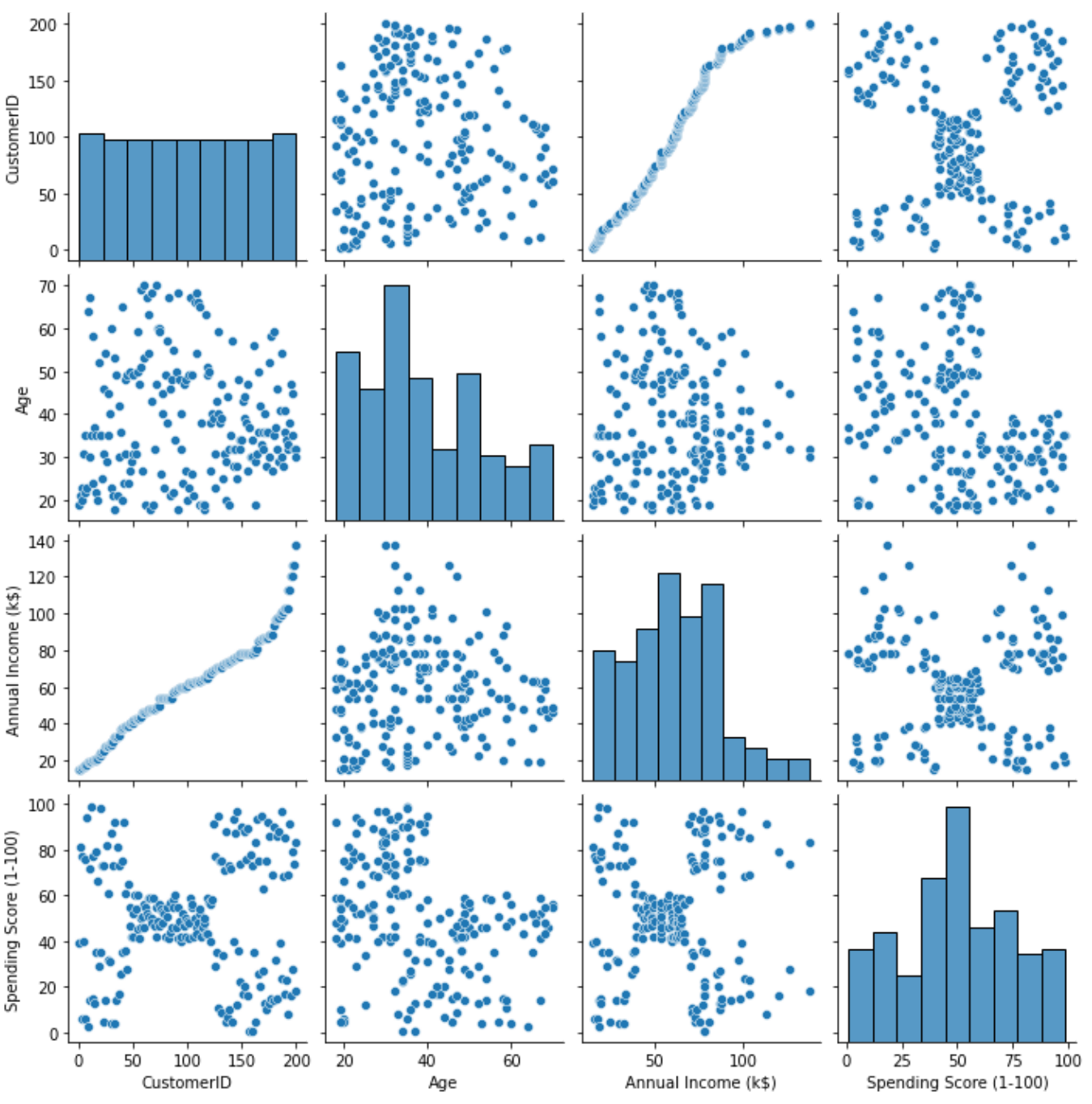
In [6]:
```
## state value is excluded as it is categorical
corr = df.corr()
corr.style.background_gradient(cmap = 'Greens')
```

Out[6]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	-0.026763	0.977548	0.013835
Age	-0.026763	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.977548	-0.012398	1.000000	0.009903
Spending Score (1-100)	0.013835	-0.327227	0.009903	1.000000

In [7]:
```
sns.pairplot(df)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x2ac8c453760>
```



select input and op variable

In [8]:
```
### since there is no op/predicted variable y is not required
x = df.drop(['CustomerID', 'Age', 'Gender'], axis=1)
```

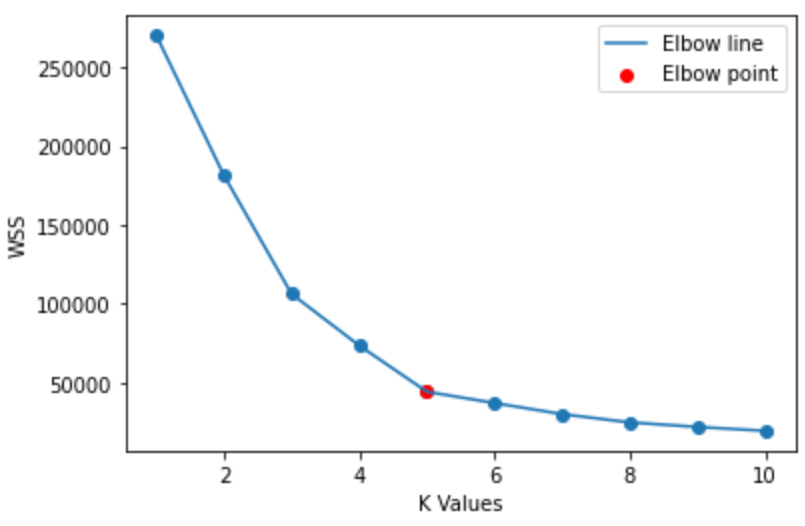
Selecting optimal value of k using elbow method

In [9]:
```
from sklearn.cluster import KMeans

wss = []

for ele in range(1, 11):
    model = KMeans(n_clusters= ele)
    model.fit(x)
    wss.append(model.inertia_)

print(wss)
```

```
[269981.28000000014, 181363.59595959607, 106348.37306211119, 73679.78903948837, 44448.45544793369, 37265.86520484345, 30241.34361793659, 24999.3682586117, 22143.222100767438, 19676.612585602812]
```

In [10]:
```
values = np.arange(1, 11)
plt.plot(values, wss, label = 'Elbow line')
plt.scatter(values,wss)
plt.ylabel('WSS')
plt.xlabel('K Values')
plt.scatter([values[4]], [wss[4]], c= 'red', label = 'Elbow point')
plt.legend()
```

Out[10]:
```
<matplotlib.legend.Legend at 0x2ac908fe6d0>
```



creating a model

In [11]:
```
from sklearn.cluster import KMeans
model = KMeans(n_clusters=5)
```

In [12]:
```
## fit the data
model.fit(x)
```

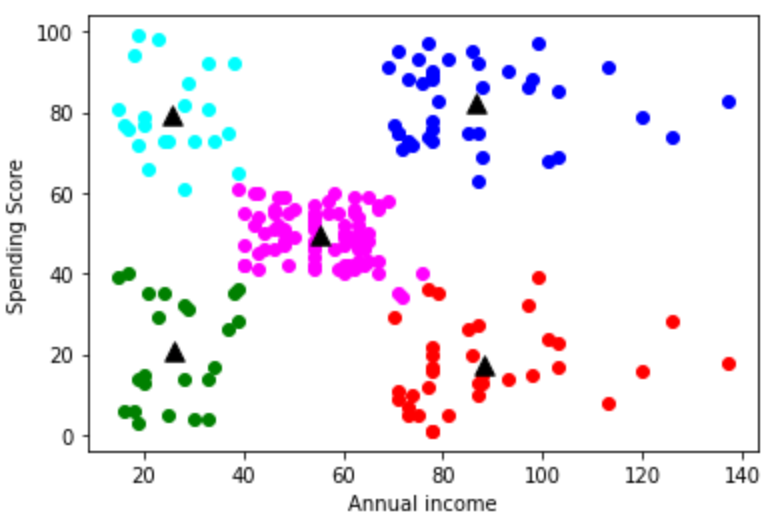
Out[12]:
```
KMeans(n_clusters=5)
```

In [13]:
```
print(model.labels_)
```

```
[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 4 1 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 2 0 2 4 2 0 2 0 2 4 2 0 2 0 2 0 2 0 2 4 2 0 2 0 2
 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0
 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]
```

In [14]:
```
plt.scatter(df['Annual Income (k$)'][model.labels_ == 0], df['Spending Score (1-100)'][
plt.scatter(df['Annual Income (k$)'][model.labels_ == 1], df['Spending Score (1-100)'][
plt.scatter(df['Annual Income (k$)'][model.labels_ == 2], df['Spending Score (1-100)'][
plt.scatter(df['Annual Income (k$)'][model.labels_ == 3], df['Spending Score (1-100)'][
plt.scatter(df['Annual Income (k$)'][model.labels_ == 4], df['Spending Score (1-100)'][
plt.scatter(model.cluster_centers_[:,0], model.cluster_centers_[:,1], c = 'black', s =
plt.xlabel('Annual income')
plt.ylabel('Spending Score')
```

Out[14]:
```
Text(0, 0.5, 'Spending Score')
```



In [ ]: