In [0]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.g
oogle.com/o/oauth2/auth?client_id=947318989803-
6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleuse
rcontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoau
th%3A2.0%3Aoob&scope=email%20https%3A%2F%2Fwww.
googleapis.com%2Fauth%2Fdocs.test%20https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fdrive%20https%3
A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photo
s.readonly%20https%3A%2F%2Fwww.googleapis.com%2
Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
··········
Mounted at /content/drive
```

In [0]:
```
!ls 'drive/My Drive/Super/clustering/Mall_Custc
```

```
'drive/My Drive/Super/clustering/Mall_Customer
s.csv'
```

Installing the Libraries

In [0]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

Importing the Dataset

In [0]:
```
%time data = pd.read_csv('drive/My Drive/Super/

print(data.shape)
```

```
CPU times: user 10.1 ms, sys: 4.09 ms, total: 1
4.2 ms
Wall time: 761 ms
(200, 5)
```

In [0]:
```
data.head()
```

Out[0]:

	CustomerID	Genre	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 [0]:
```
data.tail()
```

Out[0]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
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

In [0]:
```
data.info()
```

```
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
CustomerID                200 non-null int64
Genre                     200 non-null object
Age                       200 non-null int64
Annual Income (k$)        200 non-null int64
Spending Score (1-100)    200 non-null int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [0]:
```
data.describe()
```

Out[0]:

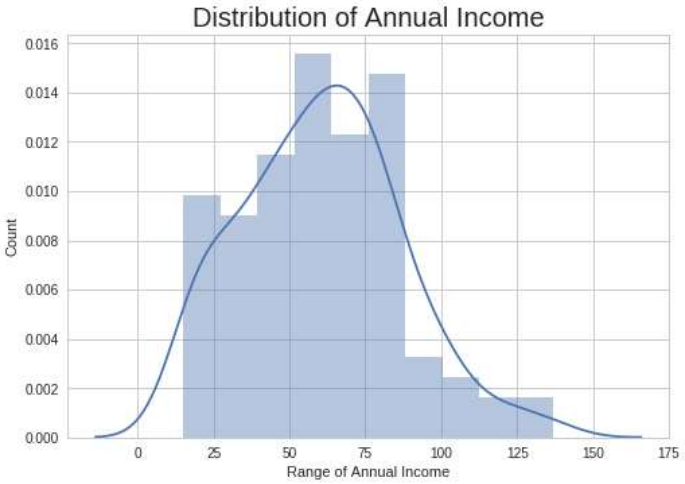
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000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

In [0]:
```
data.isnull().any()
```

Out[0]:
```
CustomerID                False
Genre                     False
Age                       False
Annual Income (k$)        False
Spending Score (1-100)    False
dtype: bool
```

In [0]:
```
sns.set(style = 'whitegrid')
sns.distplot(data['Annual Income (k$)'])
plt.title('Distribution of Annual Income', font
plt.xlabel('Range of Annual Income')
plt.ylabel('Count')
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/matplotl
ib/axes/_axes.py:6521: MatplotlibDeprecationWar
ning:
The 'normed' kwarg was deprecated in Matplotlib
2.1 and will be removed in 3.1. Use 'density' i
nstead.
  alternative="'density'", removal="3.1")
```



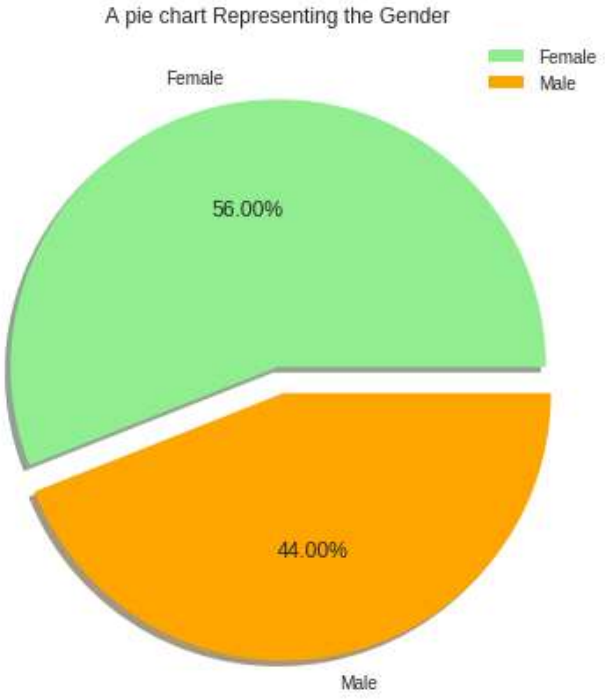
In [0]:
```
sns.set(style = 'dark')
sns.distplot(data['Age'])
plt.title('Distribution of Age', fontsize = 20)
plt.xlabel('Range of Age')
plt.ylabel('Count')
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/matplotl
ib/axes/_axes.py:6521: MatplotlibDeprecationWar
ning:
The 'normed' kwarg was deprecated in Matplotlib
2.1 and will be removed in 3.1. Use 'density' i
nstead.
  alternative="'density'", removal="3.1")
```

In [0]:
```
data['Genre'].value_counts()
```

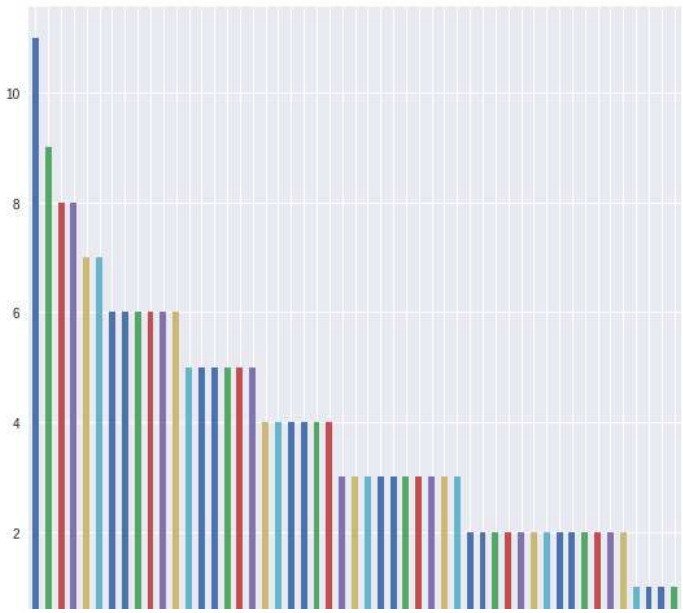
Out[0]:
```
Female    112
Male       88
Name: Genre, dtype: int64
```

In [0]:
```
labels = ['Female', 'Male']
size = [112, 88]
colors = ['lightgreen', 'orange']
explode = [0, 0.1]

plt.rcParams['figure.figsize'] = (7, 7)
plt.pie(size, colors = colors, explode = explod
plt.title('A pie chart Representing the Gender'
plt.axis('off')
plt.legend()
plt.show()
```



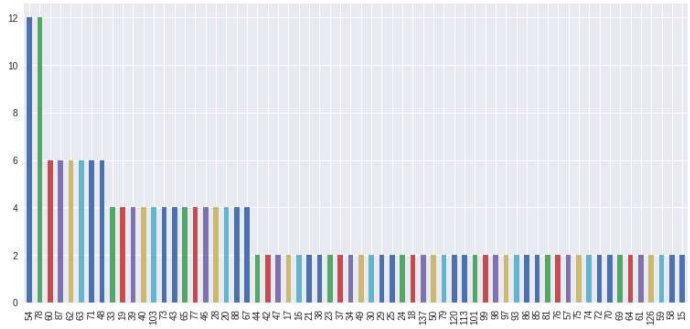
In [0]:
```
data['Age'].value_counts().plot.bar(figsize = (
```

Out[0]:



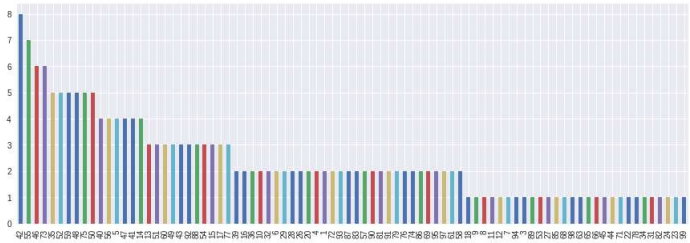
In [0]:
```
data['Annual Income (k$)'].value_counts().plot.
```

Out[0]:



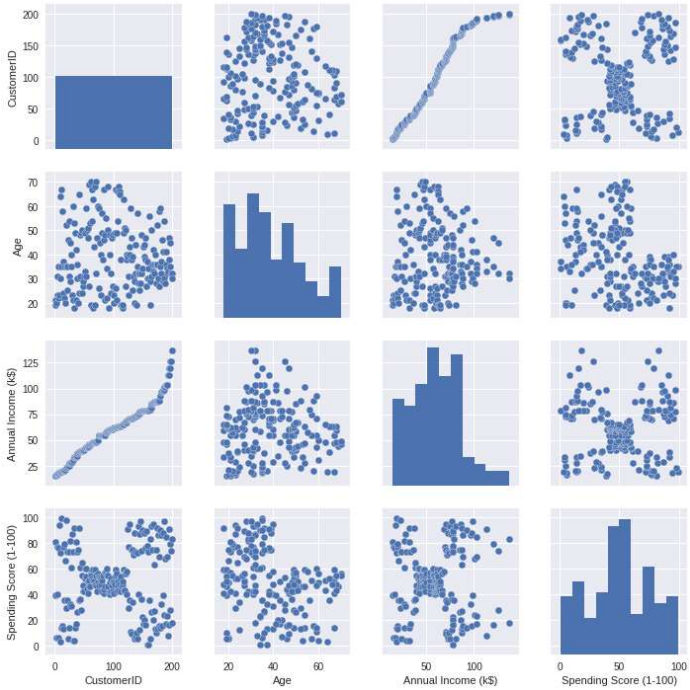
In [0]:
```
data['Spending Score (1-100)'].value_counts().p
```

Out[0]:



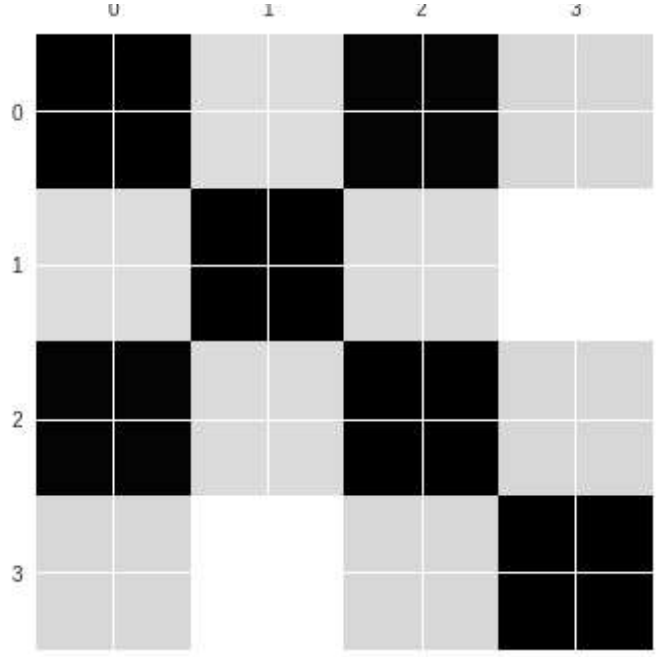
In [0]:
```
sns.pairplot(data)
```

Out[0]:



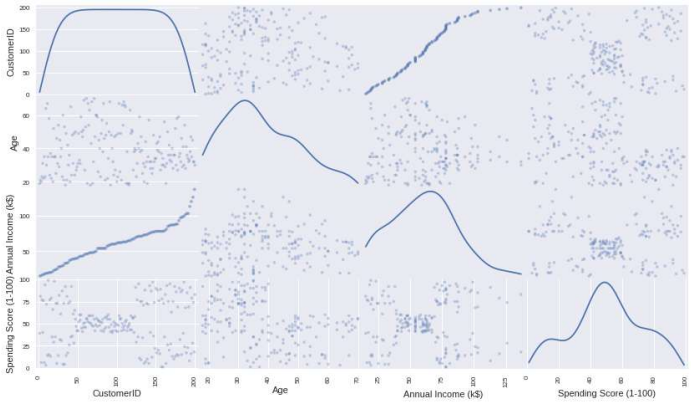
In [0]:
```
plt.matshow(data.corr())
```

Out[0]:



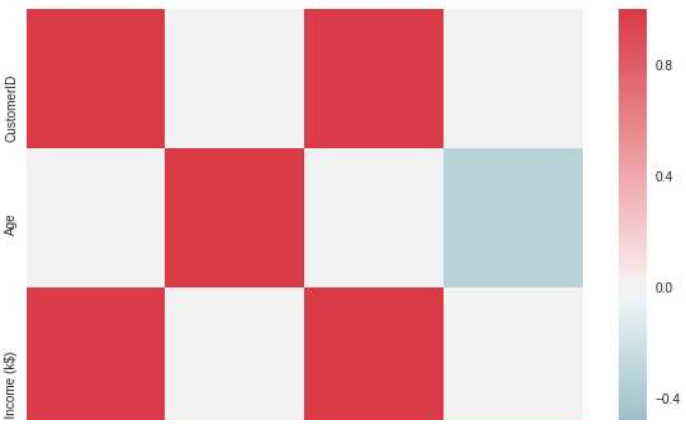
In [0]:
```
pd.scatter_matrix(data, alpha = 0.3, figsize =
```

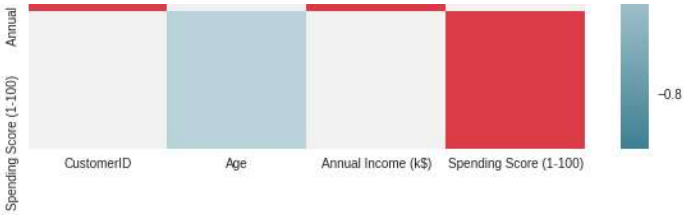
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning: pandas.scatter_matrix is deprecated. Use pandas.plotting.scatter_matrix instead
  """Entry point for launching an IPython kernel.
```



In [0]:
```
fig, axis = plt.subplots(figsize=(10, 8))
corr = data.corr()
sns.heatmap(corr, mask = np.zeros_like(corr, dt
            square = True, ax = axis)
```

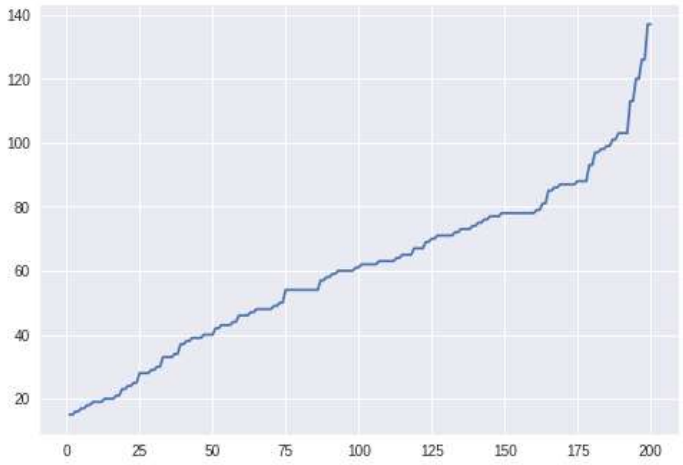
Out[0]:





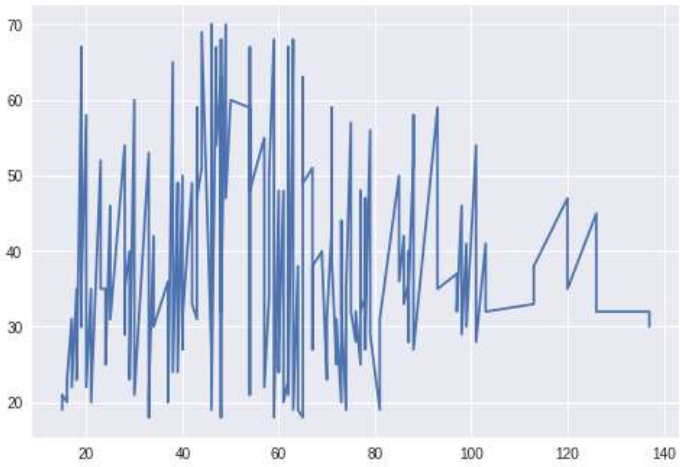
In [0]:
```
x = data['CustomerID']
y = data['Annual Income (k$)']

plt.plot(x, y)
```

Out[0]: []



In [0]:
```
x = data['Annual Income (k$)']
y = data['Age']

plt.plot(x, y)
```

Out[0]: []

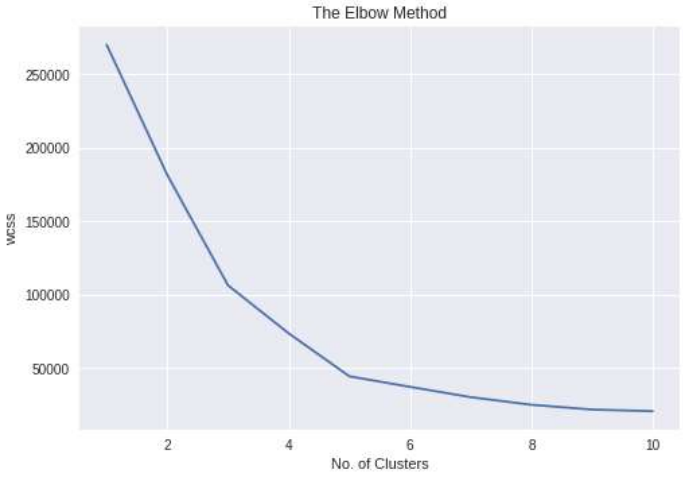


In [0]:
```
x = data.iloc[:, [3, 4]].values

print(x.shape)
```

```
(200, 2)
```

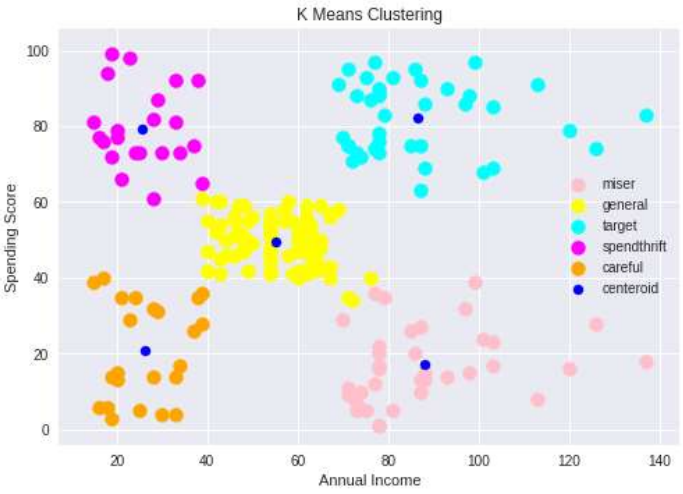
In [0]:
```
from sklearn.cluster import KMeans

wcss = []
```

```
for i in range(1, 11):
  km = KMeans(n_clusters = i, init = 'k-means++
  km.fit(x)
  wcss.append(km.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('No. of Clusters')
plt.ylabel('wcss')
plt.show()
```



In [0]:

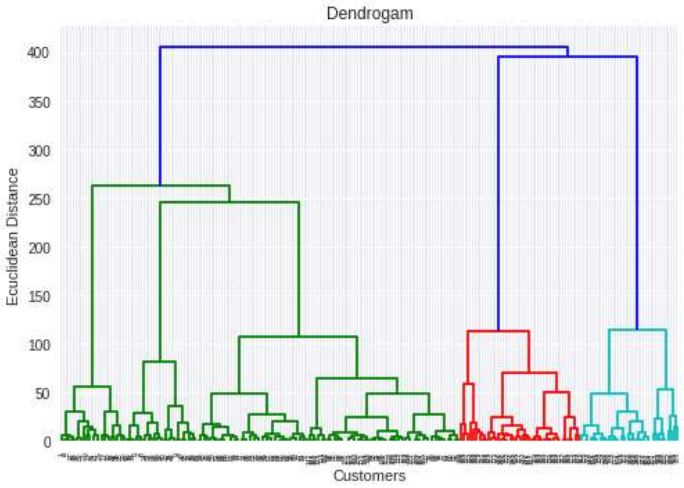
```
km = KMeans(n_clusters = 5, init = 'k-means++',
y_means = km.fit_predict(x)

plt.scatter(x[y_means == 0, 0], x[y_means == 0,
plt.scatter(x[y_means == 1, 0], x[y_means == 1,
plt.scatter(x[y_means == 2, 0], x[y_means == 2,
plt.scatter(x[y_means == 3, 0], x[y_means == 3,
plt.scatter(x[y_means == 4, 0], x[y_means == 4,
plt.scatter(km.cluster_centers_[:,0], km.cluste

plt.title('K Means Clustering')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```



In [0]:

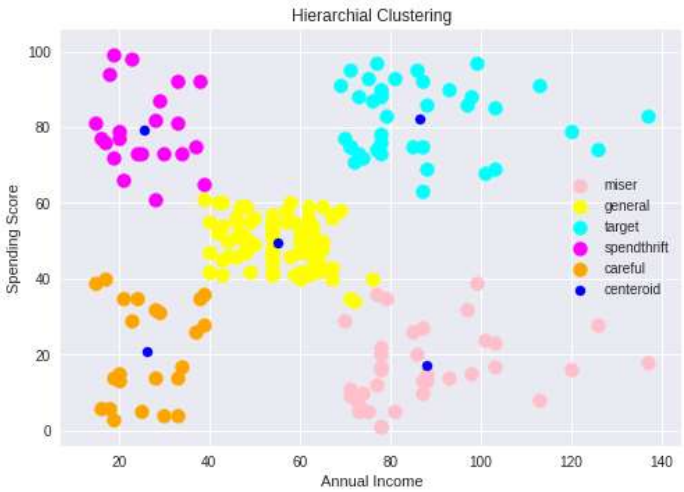
```
import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage(x, meth
```

```
plt.title('Dendrogam')
plt.xlabel('Customers')
plt.ylabel('Ecuclidean Distance')
plt.show()
```



In [0]:

```
from sklearn.cluster import AgglomerativeCluste

hc = AgglomerativeClustering(n_clusters = 5, af
y_hc = hc.fit_predict(x)

plt.scatter(x[y_means == 0, 0], x[y_means == 0,
plt.scatter(x[y_means == 1, 0], x[y_means == 1,
plt.scatter(x[y_means == 2, 0], x[y_means == 2,
plt.scatter(x[y_means == 3, 0], x[y_means == 3,
plt.scatter(x[y_means == 4, 0], x[y_means == 4,
plt.scatter(km.cluster_centers_[:,0], km.cluste

plt.title('Hierarchial Clustering')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```



Clusters of Customers Based on their Ages

In [0]:

```
data.columns
```

Out[0]:

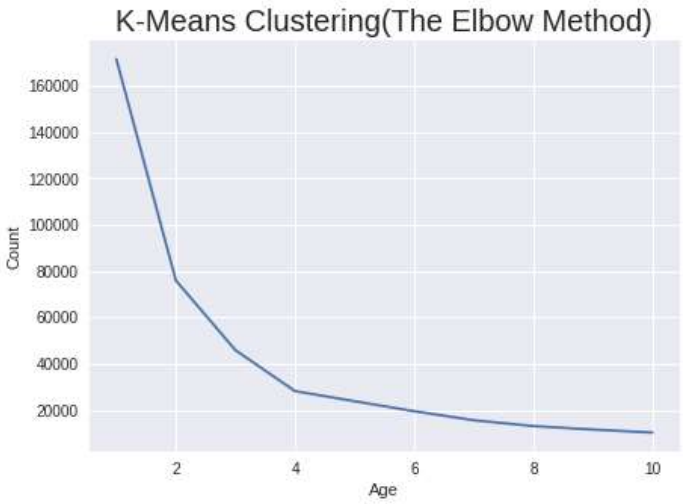
```
Index(['CustomerID', 'Genre', 'Age', 'Annual In
come (k$)',
```

```
      Spending Score (1-100) ],
     dtype='object')
```

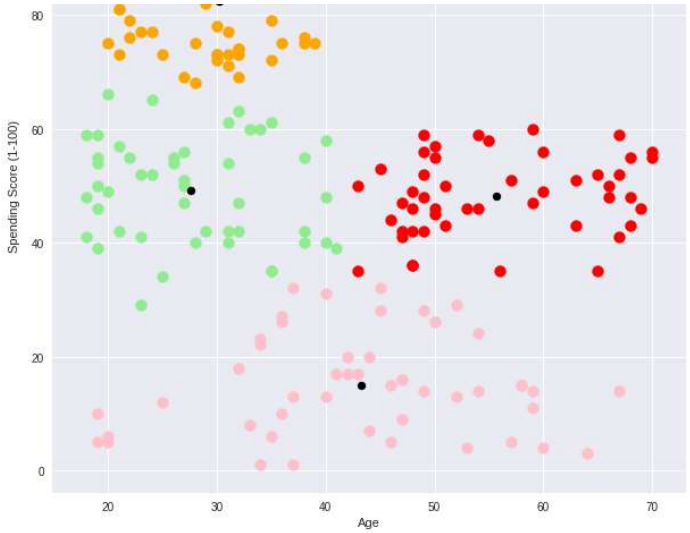
In [0]:
```
x = data.iloc[:, [2, 4]].values
x.shape
```

Out[0]: (200, 2)

In [0]:
```
from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters = i, init = 'k-mea
  kmeans.fit(x)
  wcss.append(kmeans.inertia_)

plt.rcParams['figure.figsize'] = (7, 5)
plt.plot(range(1, 11), wcss)
plt.title('K-Means Clustering(The Elbow Method)
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



In [0]:
```
kmeans = KMeans(n_clusters = 4, init = 'k-means
ymeans = kmeans.fit_predict(x)

plt.rcParams['figure.figsize'] = (10, 10)
plt.title('Cluster of Ages', fontsize = 30)

plt.scatter(x[ymeans == 0, 0], x[ymeans == 0, 1
plt.scatter(x[ymeans == 1, 0], x[ymeans == 1, 1
plt.scatter(x[ymeans == 2, 0], x[ymeans == 2, 1
plt.scatter(x[ymeans == 3, 0], x[ymeans == 3, 1
plt.scatter(kmeans.cluster_centers_[:, 0], kmea

plt.xlabel('Age')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



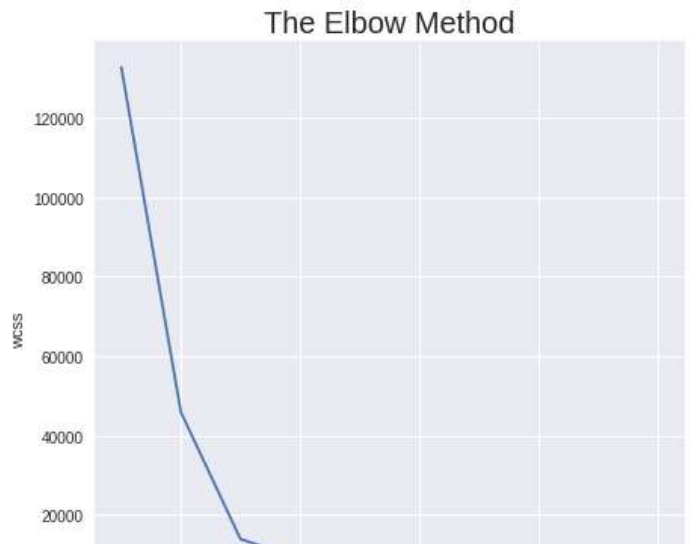
In [0]:
```
data['Genre'].replace(['Male', 'Female'], [0, 1
data['Genre'].value_counts()
```

Out[0]:
```
1    112
0     88
Name: Genre, dtype: int64
```

In [0]:
```
x = data.iloc[:, [1, 4]].values
x.shape
```

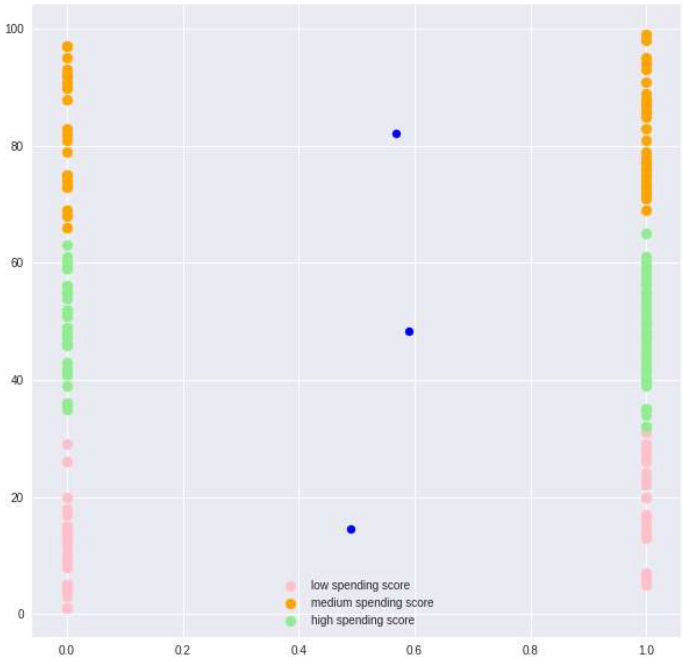
Out[0]:
```
(200, 2)
```

In [0]:
```
from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters = i, init = 'k-mea
  kmeans.fit(x)
  wcss.append(kmeans.inertia_)

plt.rcParams['figure.figsize'] = (7, 7)
plt.title('The Elbow Method', fontsize = 20)
plt.plot(range(1, 11), wcss)
plt.xlabel('No. of Clusters', fontsize = 10)
plt.ylabel('wcss')
plt.show()
```



In [0]:

```
kmeans = KMeans(n_clusters = 3, max_iter = 300,
ymeans = kmeans.fit_predict(x)

plt.rcParams['figure.figsize'] = (10, 10)
plt.scatter(x[ymeans == 0, 0], x[ymeans == 0, 1
plt.scatter(x[ymeans == 1, 0], x[ymeans == 1, 1
plt.scatter(x[ymeans == 2, 0], x[ymeans == 2, 1
plt.scatter(kmeans.cluster_centers_[:,0], kmear
plt.legend()
plt.show()
```



From Above cluster plot we can clearly see that males and females are in all the catgory that is high low and medium spending score category

In [0]: