In [62]:
```
#Data Wrapping and testing of data
import pandas as pd
```

In [7]:
```
data_source = pd.read_csv('Mall_Customers.csv')
data_source.head()
```

In [9]:
```
type(data_source)
```

Out[9]:
```
pandas.core.frame.DataFrame
```

In [10]:
```
len(data_source)
```

Out[10]:
```
200
```

In [11]:
```
df = data_source
```

In [12]:
```
df.isnull().any
```

Out[12]:
```
<bound method NDFrame._add_numeric_operations.<locals>.any of      CustomerID  Gende
r    Age  Annual Income (k$)  Spending Score (1-100)
0          False   False  False               False                   False
1          False   False  False               False                   False
2          False   False  False               False                   False
3          False   False  False               False                   False
4          False   False  False               False                   False
..           ...     ...    ...                 ...                     ...
195        False   False  False               False                   False
196        False   False  False               False                   False
197        False   False  False               False                   False
198        False   False  False               False                   False
199        False   False  False               False                   False

[200 rows x 5 columns]>
```

In [13]:
```
df.isna().any()
```

Out[13]:
```
CustomerID                False
Gender                    False
Age                       False
Annual Income (k$)        False
Spending Score (1-100)    False
dtype: bool
```

In [14]:
```
df.head(10)
```

Out[14]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
9	10	Female	30	19	72

In [15]:
```
df.head(20)
```

Out[15]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99
12	13	Female	58	20	15
13	14	Female	24	20	77
14	15	Male	37	20	13
15	16	Male	22	20	79
16	17	Female	35	21	35
17	18	Male	20	21	66
18	19	Male	52	23	29
19	20	Female	35	23	98

In [16]:
```
df.head(40)
```

Out[16]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99
12	13	Female	58	20	15
13	14	Female	24	20	77
14	15	Male	37	20	13
15	16	Male	22	20	79
16	17	Female	35	21	35
17	18	Male	20	21	66
18	19	Male	52	23	29
19	20	Female	35	23	98
20	21	Male	35	24	35
21	22	Male	25	24	73
22	23	Female	46	25	5
23	24	Male	31	25	73
24	25	Female	54	28	14
25	26	Male	29	28	82
26	27	Female	45	28	32
27	28	Male	35	28	61
28	29	Female	40	29	31
29	30	Female	23	29	87
30	31	Male	60	30	4
31	32	Female	21	30	73
32	33	Male	53	33	4
33	34	Male	18	33	92
34	35	Female	49	33	14
35	36	Female	21	33	81
36	37	Female	42	34	17
37	38	Female	30	34	73
38	39	Female	36	37	26
39	40	Female	20	37	75

In [17]:
```
df.head(80)
```

Out[17]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
75	76	Male	26	54	54
76	77	Female	45	54	53
77	78	Male	40	54	48
78	79	Female	23	54	52
79	80	Female	49	54	42

80 rows × 5 columns

In [18]:
```
df.head(150)
```

Out[18]:

	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
...	...	...	...	...	...
145	146	Male	28	77	97
146	147	Male	48	77	36
147	148	Female	32	77	74
148	149	Female	34	78	22
149	150	Male	34	78	90

150 rows × 5 columns

In [19]:
```
df.head(200)
```

Out[19]:

	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

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
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 [20]:
```
df.tail()
```

Out[20]:

	CustomerID	Gender	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 [21]:
```
df.head(120)
```

Out[21]:

	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
...	...	...	...	...	...
115	116	Female	19	65	50
116	117	Female	63	65	43
117	118	Female	49	65	59
118	119	Female	51	67	43
119	120	Female	50	67	57

120 rows × 5 columns

In [22]:
```
df.describe()
```

Out[22]:

	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

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
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 [23]:
```
df.info()
```

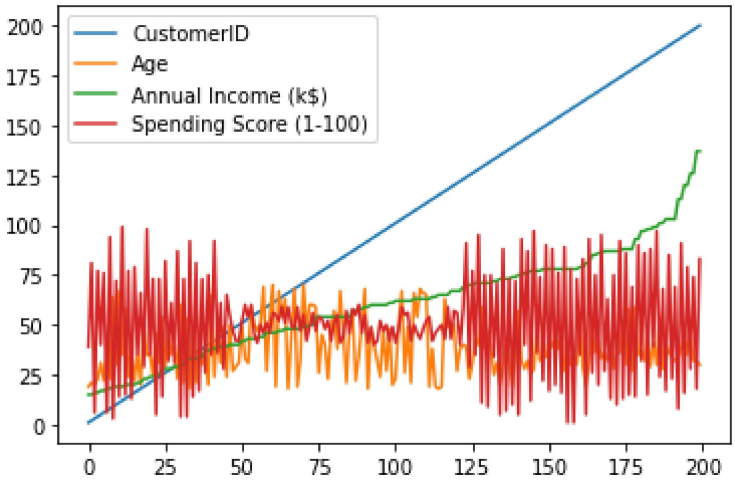
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null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

In [24]:
```
df.dtypes
```

Out[24]:
```
CustomerID                int64
Gender                   object
Age                       int64
Annual Income (k$)        int64
Spending Score (1-100)    int64
dtype: object
```

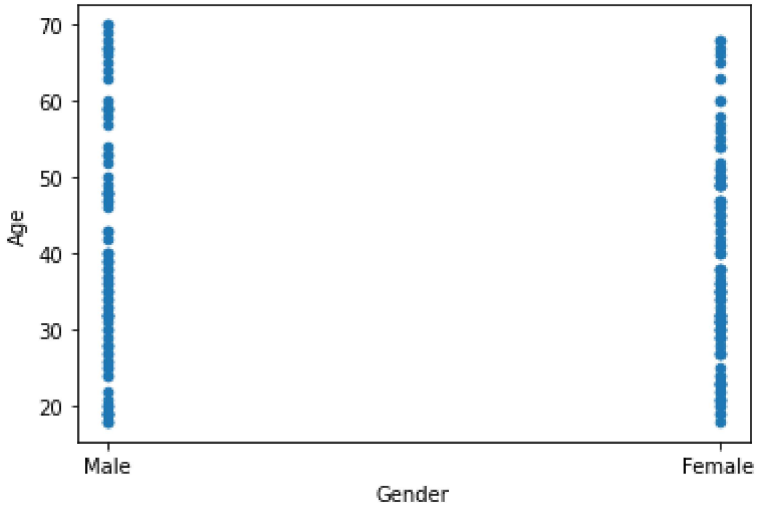
In [25]:
```
df.plot()
```

Out[25]:
```
<AxesSubplot:>
```



In [35]:
```
df.plot(x = 'Gender', y = 'Age', kind = 'scatter')
```

Out[35]:
```
<AxesSubplot:xlabel='Gender', ylabel='Age'>
```



In [29]:
```
df[['Gender','Age']].head()
```

Out[29]:

	Gender	Age
0	Male	19
1	Male	21
2	Female	20
3	Female	23
4	Female	31

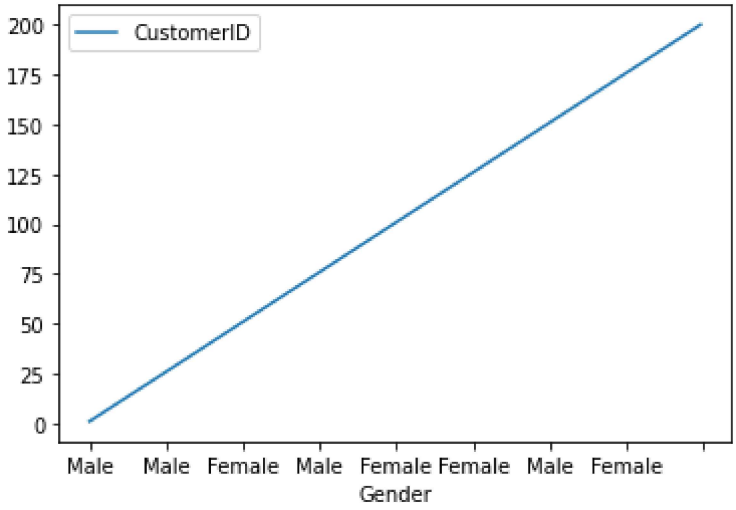
In [33]:
```
df[['CustomerID', 'Age']].head()
```

Out[33]:

	CustomerID	Age
0	1	19
1	2	21
2	3	20
3	4	23
4	5	31

In [38]:
```
df.plot(x='Gender', y = 'CustomerID', kind='line')
```

Out[38]:
```
<AxesSubplot:xlabel='Gender'>
```



In [9]:
```
#K means Clustering for Customer Segmentation
```

In [3]:
```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

In [4]:
```
#Importing My Data Set
```

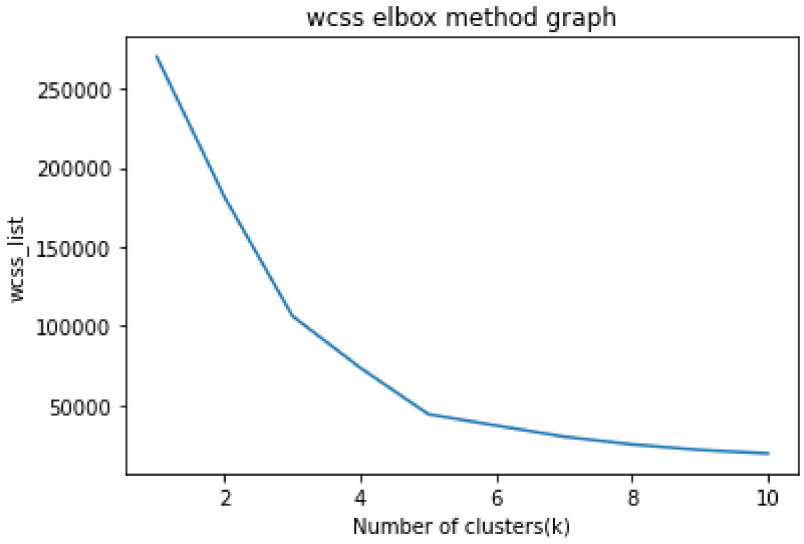
In [22]:
```
dataset = pd.read_csv('Mall_Customers.csv')
```

In [23]:
```
#Extracting Independent variables since it is for clustering
```

In [24]:
```
x = dataset.iloc[:,[3,4]].values
```

In [25]:
```
#Finding Optimal numbers of Clusters using Elbox Method
```

In [26]:
```
from sklearn.cluster import KMeans
wcss_list = [] #Initialising the values for wcss

for i in range(1,11):
    kmeans = KMeans(n_clusters=i, init='k-means++',random_state = 42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11), wcss_list)
mtp.title('wcss elbox method graph')
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()
```



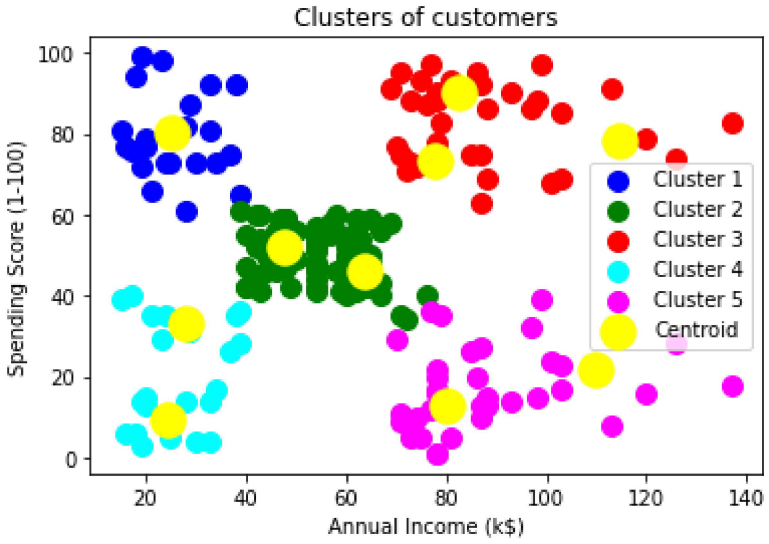
In [18]:
```
#Now training the K Model on my data set and I am using 5 as there are 5 clusters th
```

In [21]:
```
kmeans =  KMeans(n_clusters=5, init='k-means++',random_state=42)
y_predict = kmeans.fit_predict(x)
```

In [27]:
```
#I have created y_predict which is called as dependent variable to train my model in
```

In [28]:
```
#Visualizing the five clusters and using mtp.scatter() function to visualize the clu
```

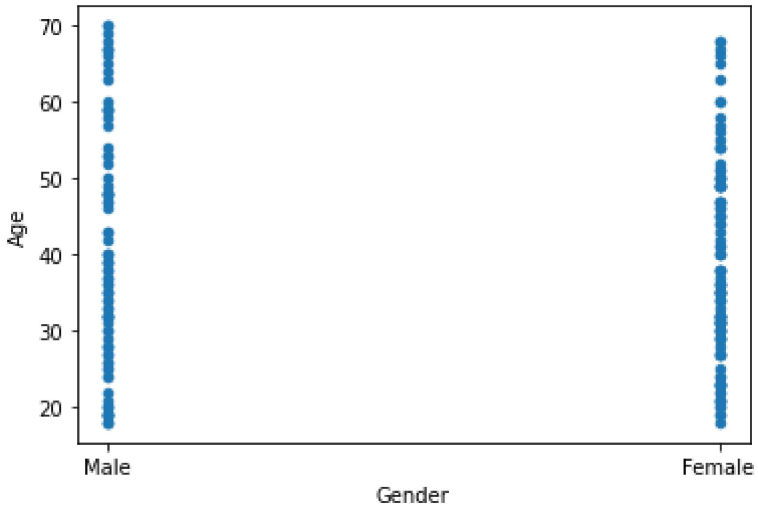
In [48]:
```
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label =

mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label

mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label = '

mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c = 'cyan', label =

mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c = 'magenta', labe

mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c

mtp.title('Clusters of customers')

mtp.xlabel('Annual Income (k$)')

mtp.ylabel('Spending Score (1-100)')

mtp.legend()

mtp.show()
```



In [49]:
```
#Now Visualizing the Gender and Age Distributions from my dataset using different pl
```

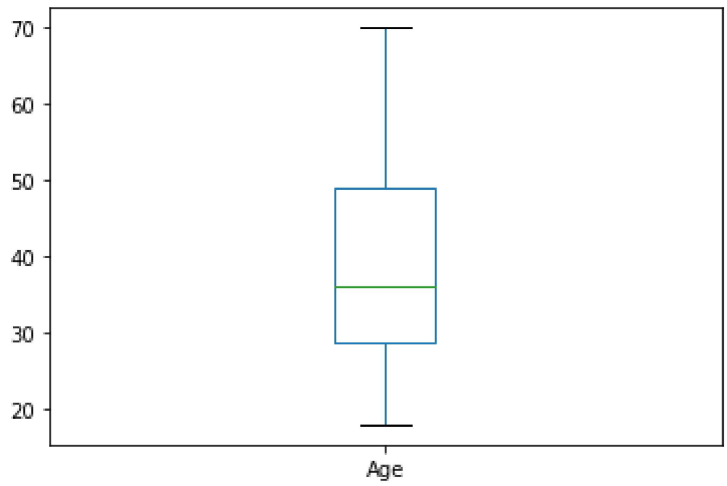
In [50]:
```
dataset.plot(x='Gender',y='Age', kind='scatter')
```

Out[50]: <AxesSubplot:xlabel='Gender', ylabel='Age'>



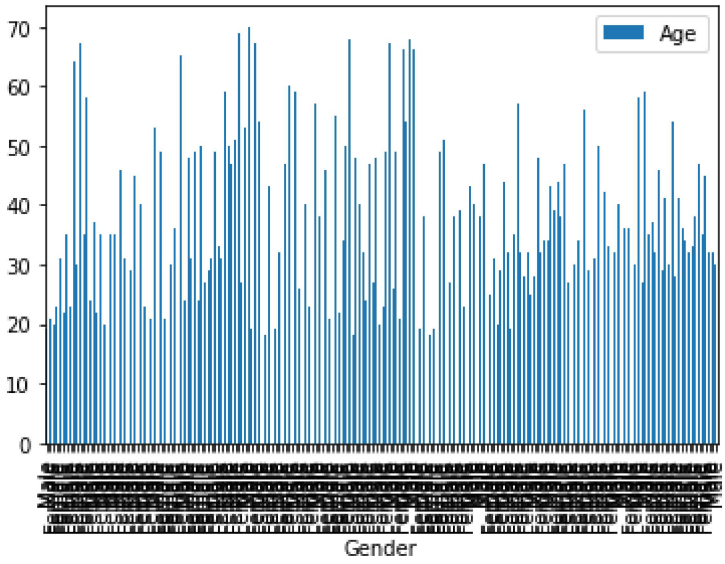
In [51]:
```
dataset.plot(x='Gender', y='Age', kind='box')
```

Out[51]: <AxesSubplot:>



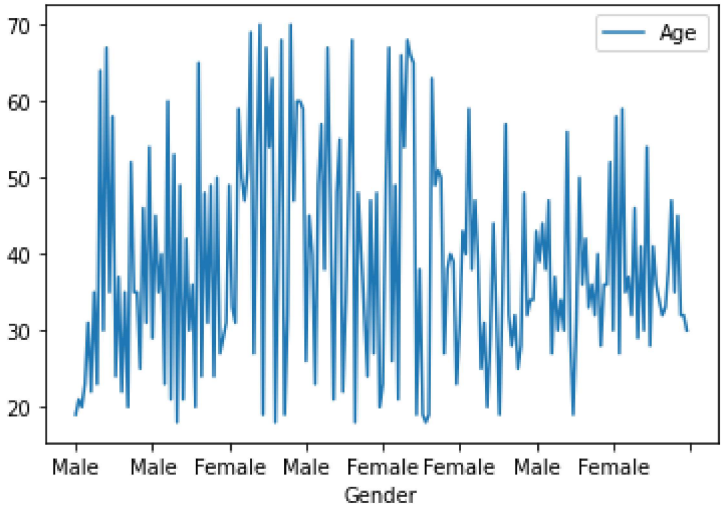
In [53]:
```
dataset.plot(x='Gender',y='Age', kind='bar')
```

Out[53]: <AxesSubplot:xlabel='Gender'>



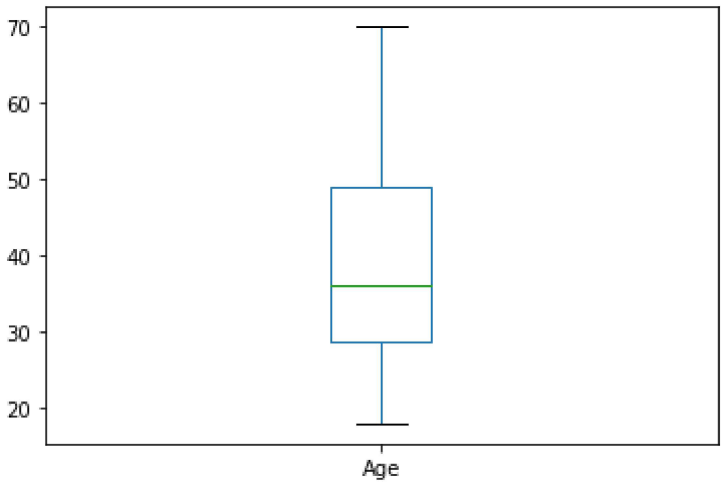
In [54]:
```
dataset.plot(x='Gender',y='Age', kind='line')
```

Out[54]: <AxesSubplot:xlabel='Gender'>



In [63]:
```
dataset.plot(x='Gender',y='Age', kind='box')
```

Out[63]: <AxesSubplot:>



In [ ]: