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

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

	Age	City	Gender	Education Level	Occupation	Family Members	Annual Income	Charging Hours	Electric Cars are economical	Charging stations	Convert your car to all electric	Hyb B
0	29	Kolkata	Male	Postgraduate	Working Professional	5	150000	1	Yes	15	Yes	Hy
1	29	Mumbai	Male	Graduate	Working Professional	4	1000000	1	Yes	8	Yes	Ele
2	25	Mumbai	Male	Graduate	Working Professional	4	5000	5	Yes	15	Didn't think about it	Hy
3	25	Bangalore	Male	Graduate	Working Professional	4	750000	1	Don't know	8	Didn't think about it	Hy

```
df.shape
```

```
(299, 13)
```

```
df.describe()
```

	Age	Family Members	Annual Income	Charging Hours	Charging stations	Next car will be electric car?
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000
mean	33.066890	3.816054	603812.709030	3.685619	13.026756	4.986622
std	10.390474	0.949890	387815.052751	2.697531	6.631678	2.965091
min	17.000000	2.000000	5000.000000	1.000000	8.000000	1.000000
25%	23.000000	3.000000	150000.000000	2.000000	8.000000	3.000000
50%	31.000000	4.000000	750000.000000	2.000000	8.000000	5.000000
75%	42.000000	4.000000	1000000.000000	5.000000	15.000000	5.000000
max	57.000000	7.000000	1000000.000000	12.000000	30.000000	10.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Age                             299 non-null    int64
 1   City                            299 non-null    object
 2   Gender                          299 non-null    object
 3   Education Level                 299 non-null    object
 4   Occupation                      299 non-null    object
 5   Family Members                  299 non-null    int64
 6   Annual Income                   299 non-null    int64
 7   Charging Hours                  299 non-null    int64
 8   Electric Cars are economical    299 non-null    object
 9   Charging stations               299 non-null    int64
 10  Convert your car to all electric  299 non-null    object
 11  Hybrid or Both                  299 non-null    object
 12  Next car will be electric car?    299 non-null    int64
dtypes: int64(6), object(7)
memory usage: 30.5+ KB
```

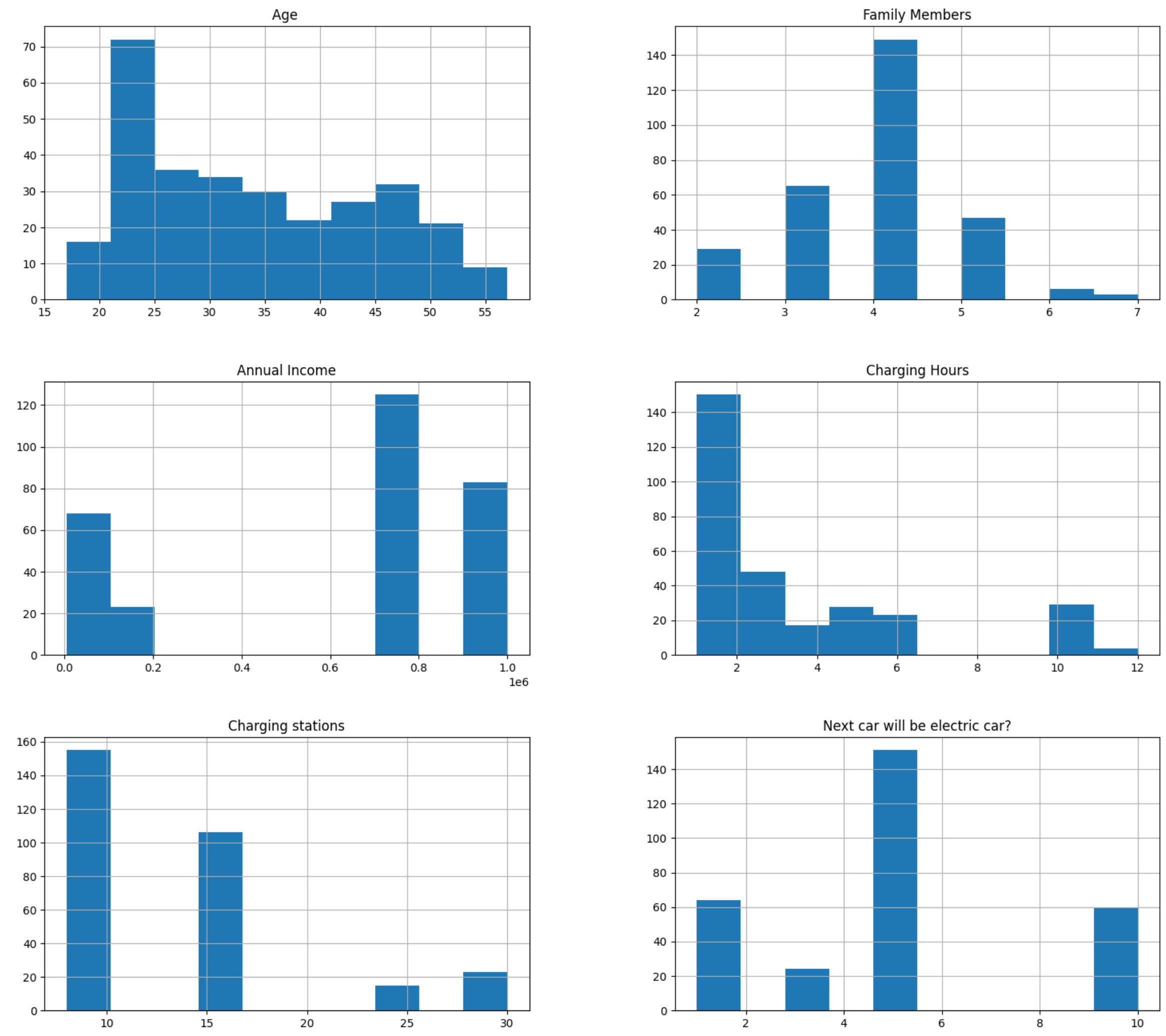
```
df.isnull().sum()
```

```
Age                             0
City                            0
Gender                          0
Education Level                 0
Occupation                      0
Family Members                  0
Annual Income                   0
Charging Hours                  0
Electric Cars are economical    0
```

```
    Charging stations                0
    Convert your car to all electric    0
    Hybrid or Both                   0
    Next car will be electric car?      0
    dtype: int64
```

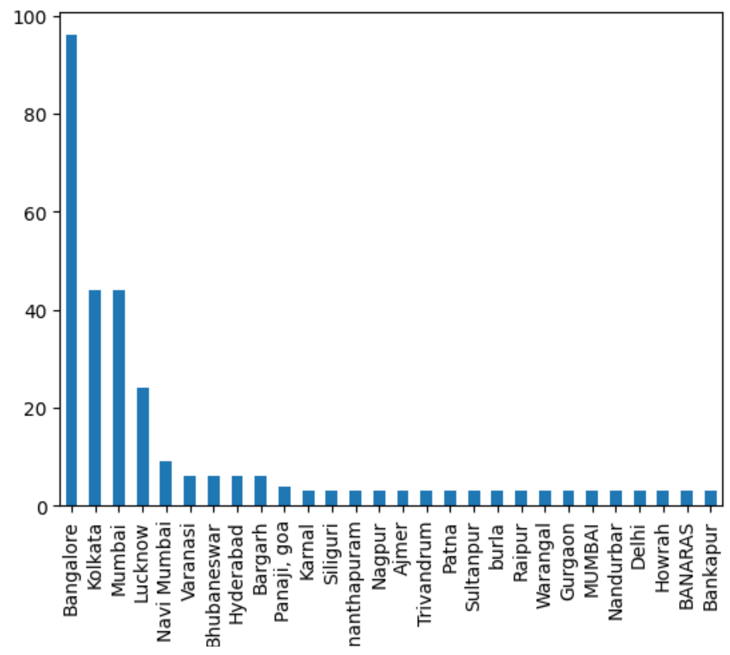
```
df.hist(figsize=(18, 16))
```

```
    array([[<Axes: title={'center': 'Age '}>,
            <Axes: title={'center': 'Family Members'}>],
           [<Axes: title={'center': 'Annual Income'}>,
            <Axes: title={'center': 'Charging Hours'}>],
           [<Axes: title={'center': 'Charging stations'}>,
            <Axes: title={'center': 'Next car will be electric car?'}>]],
          dtype=object)
```



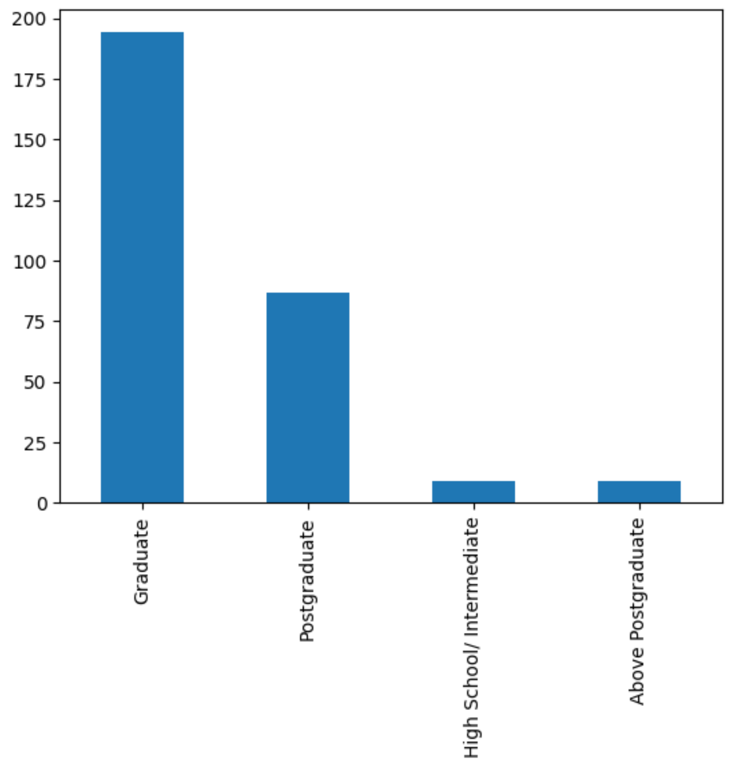
```
df['City'].value_counts().plot.bar()
```

```
<Axes: >
```



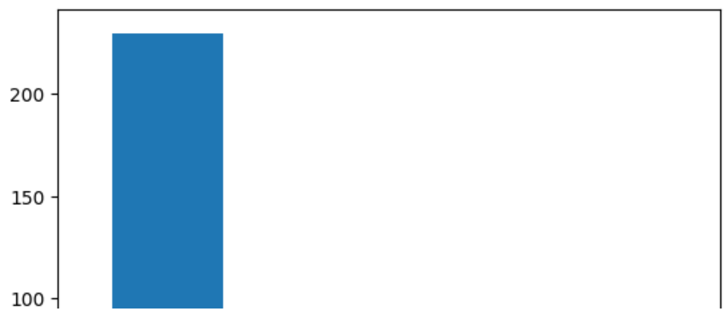
```
df['Education Level'].value_counts().plot.bar()
```

```
<Axes: >
```



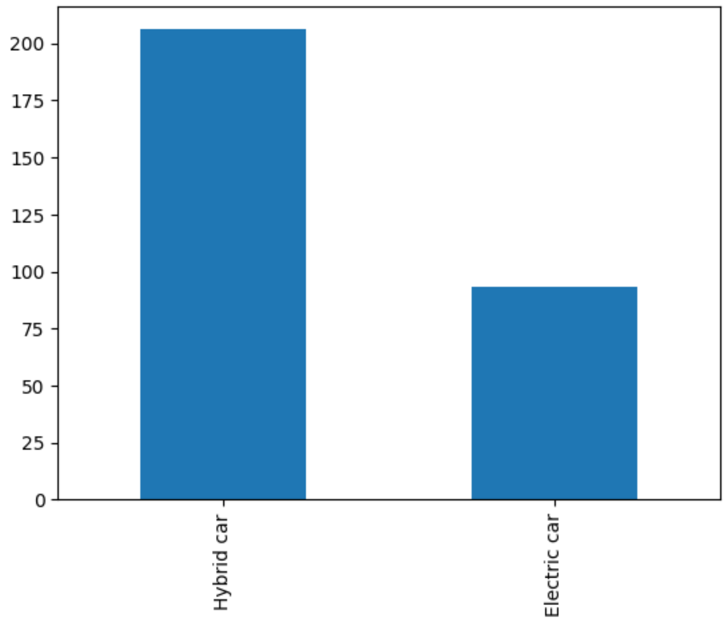
```
df['Occupation'].value_counts().plot.bar()
```

```
<Axes: >
```



```
df['Hybrid or Both'].value_counts().plot.bar()
```

```
<Axes: >
```



```
sns.distplot(df['Annual Income'])
```

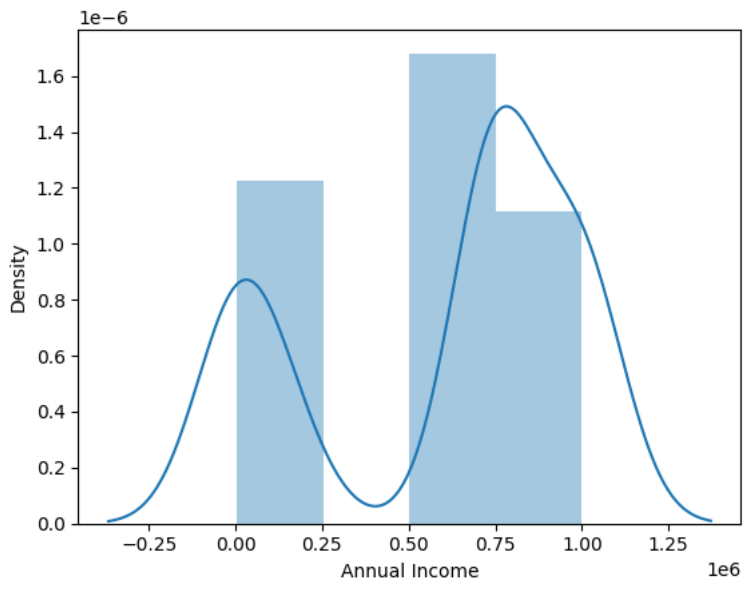
```
<ipython-input-18-522a83451b3b>:1: UserWarning:

`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['Annual Income'])
<Axes: xlabel='Annual Income', ylabel='Density'>
```



```
df["Annual Income"].describe()
```

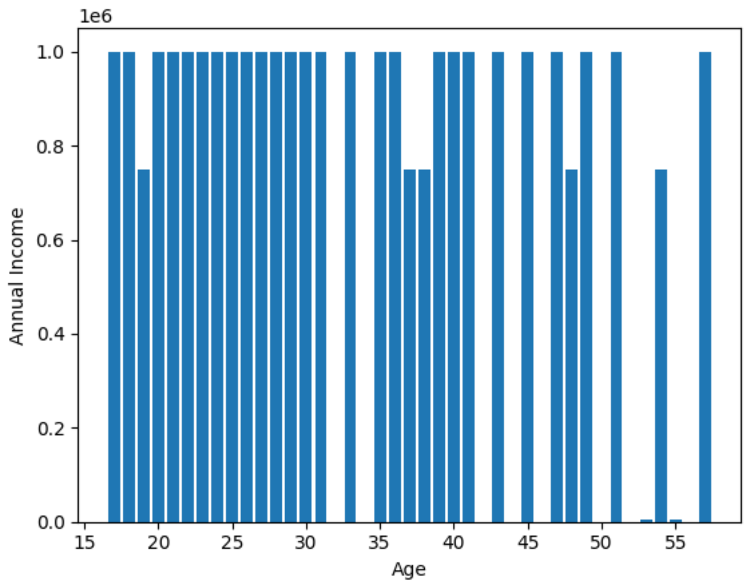
```
count        299.000000
mean      603812.709030
std       387815.052751
min         5000.000000
25%       150000.000000
50%       750000.000000
75%      1000000.000000
max      1000000.000000
Name: Annual Income, dtype: float64
```

```
df["Annual Income"].describe()
```

```
count        299.000000
mean      603812.709030
std       387815.052751
min         5000.000000
25%       150000.000000
50%       750000.000000
75%      1000000.000000
max      1000000.000000
Name: Annual Income, dtype: float64
```

```
plt.bar(df['Age '], df['Annual Income'])
plt.xlabel('Age')
plt.ylabel('Annual Income')
```

```
Text(0, 0.5, 'Annual Income')
```



```
df['Hybrid or Both'].unique()
```

```
array(['Hybrid car', 'Electric car'], dtype=object)
```

```
df.duplicated().sum()
```

```
0
```

PRE_PROCESSING

```
from sklearn import preprocessing
```

```
df['Occupation']=preprocessing.LabelEncoder().fit_transform(df['Occupation'].values.reshape(-1,1))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vector y was passed wh
  y = column_or_1d(y, warn=True)
```

```
df
```

	Age	City	Gender	Education Level	Occupation	Family Members	Annual Income	Charging Hours	Electric Cars are economical	Charging stations	Convert your car to all electric	H
0	29	Kolkata	Male	Postgraduate	2	5	150000	1	Yes	15	Yes	
1	29	Mumbai	Male	Graduate	2	4	1000000	1	Yes	8	Yes	E
2	25	Mumbai	Male	Graduate	2	4	5000	5	Yes	15	Didn't think about it	
3	25	Bangalore	Male	Graduate	2	4	750000	1	Don't know	8	Didn't think about it	
4	21	Mumbai	Male	Postgraduate	1	4	5000	12	Don't know	15	Didn't think about it	
...	...	...	...	...	...	...	...	...	...	...	...	
294	19	Mumbai	Male	Graduate	2	4	750000	1	Yes	8	Yes	

```
df['Occupation'].unique()

     array([2, 1, 0])
```

```
df['Education Level']=preprocessing.LabelEncoder().fit_transform(df['Education Level'].values.reshape(-1,1))

     /usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vector y was passed wh
       y = column_or_1d(y, warn=True)
```

```
df
```

	Age	City	Gender	Education Level	Occupation	Family Members	Annual Income	Charging Hours	Electric Cars are economical	Charging stations	Convert your car to all electric	Hyb E
0	29	Kolkata	Male	3	2	5	150000	1	Yes	15	Yes	Hy
1	29	Mumbai	Male	1	2	4	1000000	1	Yes	8	Yes	Ele
2	25	Mumbai	Male	1	2	4	5000	5	Yes	15	Didn't think about it	Hy
3	25	Bangalore	Male	1	2	4	750000	1	Don't know	8	Didn't think about it	Hy
4	21	Mumbai	Male	3	1	4	5000	12	Don't know	15	Didn't think about it	Hy
...	...	...	...	...	...	...	...	...	...	...	...	
294	19	Mumbai	Male	1	2	4	750000	1	Yes	8	Yes	Hy
295	21	Bangalore	Male	1	2	4	750000	1	Yes	30	Yes	Hy

```
df['Education Level'].unique()

     array([3, 1, 2, 0])
```

```
bins=[0,1000000,2000000,3000000,4000000]
group=['Low','Average','High','Very high']
df['Income_bin']=pd.cut(df['Annual Income'],bins,labels=group)
df['Income_bin']

     0       Low
     1       Low
     2       Low
     3       Low
     4       Low
            ...
```

```
294    Low
295    Low
296    Low
297    Low
298    Low
Name: Income_bin, Length: 299, dtype: category
Categories (4, object): ['Low' < 'Average' < 'High' < 'Very high']
```

```
df
```

	Age	City	Gender	Education Level	Occupation	Family Members	Annual Income	Charging Hours	Electric Cars are economical	Charging stations	Convert your car to all electric	Hyt E
0	29	Kolkata	Male	3	2	5	150000	1	Yes	15	Yes	Hy
1	29	Mumbai	Male	1	2	4	1000000	1	Yes	8	Yes	Ele
2	25	Mumbai	Male	1	2	4	5000	5	Yes	15	Didn't think about it	Hy
3	25	Bangalore	Male	1	2	4	750000	1	Don't know	8	Didn't think about it	Hy
4	21	Mumbai	Male	3	1	4	5000	12	Don't know	15	Didn't think about it	Hy
...	...	...	...	...	...	...	...	...	...	...	...	
294	19	Mumbai	Male	1	2	4	750000	1	Yes	8	Yes	Hy
295	21	Bangalore	Male	1	2	4	750000	1	Yes	30	Yes	Hy
296	23	Panaji, goa	Male	1	2	3	750000	1	Yes	8	Didn't think about it	Ele
297	25	Mumbai	Male	1	2	2	750000	1	Yes	15	Didn't think	Hy

```
df['Income_bin1']=df['Income_bin'].cat.codes
```

```
df['Income_bin1'].unique()
```

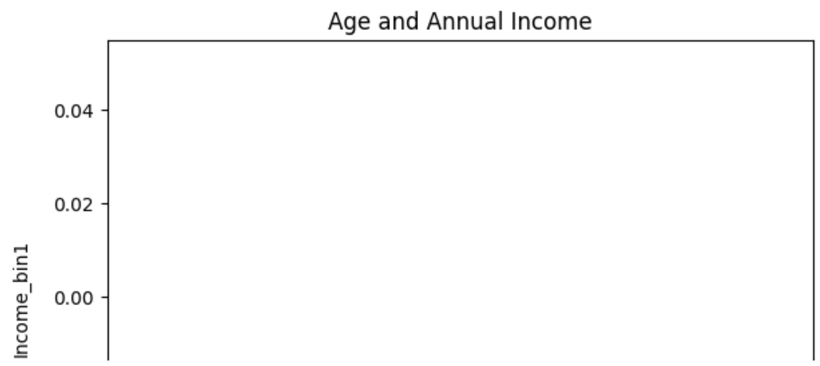
```
array([0], dtype=int8)
```

```
df['Annual Income'].describe()
```

```
count        299.000000
mean      603812.709030
std       387815.052751
min         5000.000000
25%       150000.000000
50%       750000.000000
75%      1000000.000000
max      1000000.000000
Name: Annual Income, dtype: float64
```

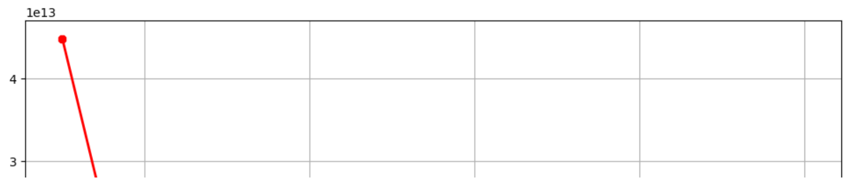
```
plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
plt.title('Age and Annual Income')
pic=sns.barplot(x='Age ', y='Income_bin1', data=df,palette='ocean')
pic.set_xticklabels(pic.get_xticklabels(),rotation=90,ha='right')
```

```
[Text(0, 0, '17'),
 Text(1, 0, '18'),
 Text(2, 0, '19'),
 Text(3, 0, '20'),
 Text(4, 0, '21'),
 Text(5, 0, '22'),
 Text(6, 0, '23'),
 Text(7, 0, '24'),
 Text(8, 0, '25'),
 Text(9, 0, '26'),
 Text(10, 0, '27'),
 Text(11, 0, '28'),
 Text(12, 0, '29'),
 Text(13, 0, '30'),
 Text(14, 0, '31'),
 Text(15, 0, '33'),
 Text(16, 0, '35'),
 Text(17, 0, '36'),
 Text(18, 0, '37'),
 Text(19, 0, '38'),
 Text(20, 0, '39'),
 Text(21, 0, '40'),
 Text(22, 0, '41'),
 Text(23, 0, '43'),
 Text(24, 0, '45'),
 Text(25, 0, '47'),
 Text(26, 0, '48'),
 Text(27, 0, '49'),
 Text(28, 0, '51'),
 Text(29, 0, '53'),
 Text(30, 0, '54'),
 Text(31, 0, '55'),
 Text(32, 0, '57')]
```



```
x1 = df.loc[:,['Age ', 'Annual Income']].values
from sklearn.cluster import KMeans
wcss= []
for k in range(1,11):
  kmeans= KMeans(n_clusters=k, init='k-means++')
  kmeans.fit(x1)
  wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
```



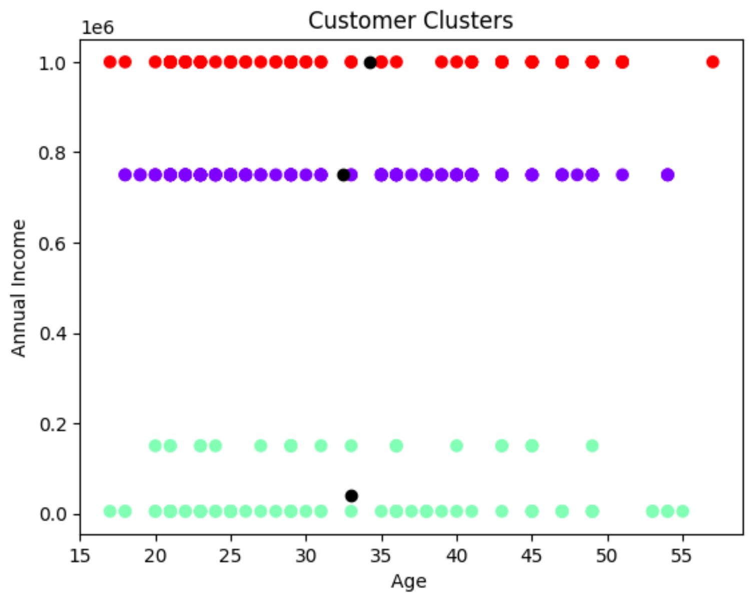
```
kmeans = KMeans(n_clusters=3)
label = kmeans.fit_predict(x1)
print(label)
```

```
[1 2 1 0 1 1 0 0 0 0 2 1 2 0 0 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0 0 2 2 2 0 0 0
 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1 1 2 2 2 0 0 2
 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0 0 0 2 1 2 0 0
 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0 0 2 2 2 0 0 0 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1
 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1 1 2 2 2 0 0 2 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1
 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0 0 0 2 1 2 0 0 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0
 0 2 2 2 0 0 0 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1
 1 2 2 2 0 0 2 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0
 0 0 2]
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```

```
print(kmeans.cluster_centers_)
```

```
[[3.24080000e+01 7.50000000e+05]
 [3.29450549e+01 4.16483516e+04]
 [3.41927711e+01 1.00000000e+06]]
```

```
plt.scatter(x1[:,0],x1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Customer Clusters')
plt.xlabel('Age ')
plt.ylabel('Annual Income')
plt.show()
```



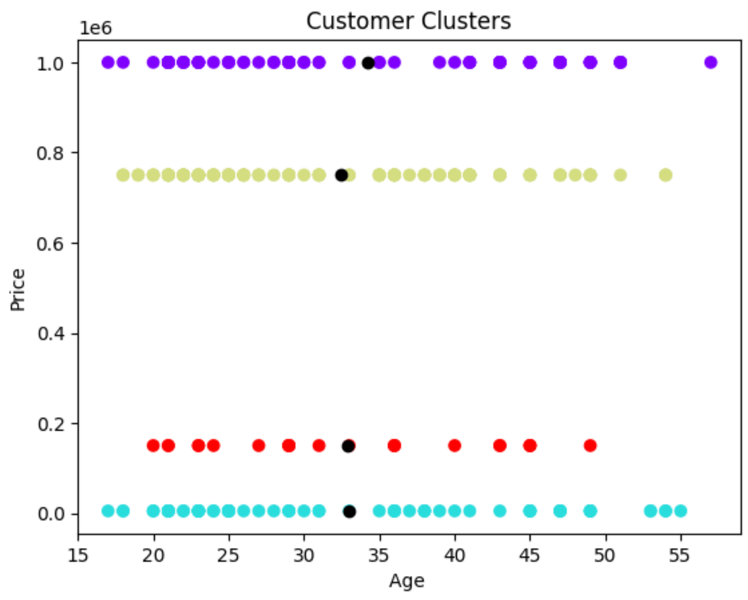
```
kmeans = KMeans(n_clusters=4)
label = kmeans.fit_predict(x1)
print(label)
```

```
    [3 0 1 2 1 3 2 2 2 2 0 1 0 2 2 1 2 3 1 0 1 0 2 0 1 2 1 1 2 2 2 0 0 0 2 2 2
     2 0 0 2 2 2 2 0 2 2 2 0 0 1 1 0 2 2 0 0 2 2 2 3 1 2 2 0 0 3 3 0 0 0 2 2 0
     0 1 0 1 2 2 2 2 1 1 3 1 0 1 3 0 1 1 1 1 2 2 2 0 1 2 1 3 2 2 2 2 0 1 0 2 2
     1 2 3 1 0 1 0 2 0 1 2 1 1 2 2 2 0 0 0 2 2 2 2 0 0 2 2 2 2 0 2 2 2 0 0 1 1
     0 2 2 0 0 2 2 2 3 1 2 2 0 0 3 3 0 0 0 2 2 0 0 1 0 1 2 2 2 2 1 1 3 1 0 1 3
     0 1 1 1 1 2 2 2 0 1 2 1 3 2 2 2 2 0 1 0 2 2 1 2 3 1 0 1 0 2 0 1 2 1 1 2 2
     2 0 0 0 2 2 2 2 0 0 2 2 2 2 0 2 2 2 0 0 1 1 0 2 2 0 0 2 2 2 3 1 2 2 0 0 3
     3 0 0 0 2 2 0 0 1 0 1 2 2 2 2 1 1 3 1 0 1 3 0 1 1 1 1 2 2 2 0 1 2 1 3 2 2
     2 2 0]
    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
      warnings.warn(
```

```
print(kmeans.cluster_centers_)
```

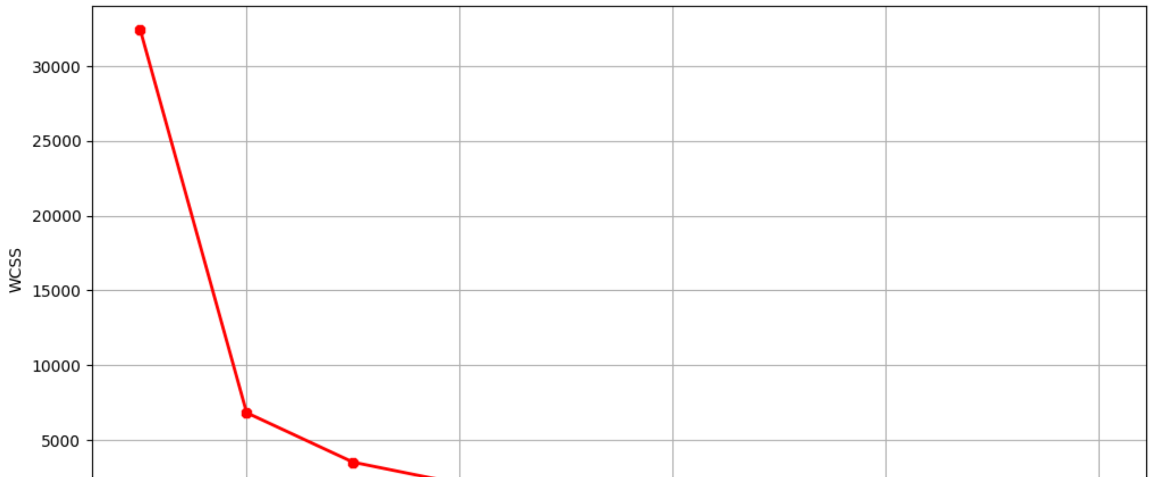
```
    [[3.41927711e+01 1.00000000e+06]
     [3.29558824e+01 5.00000000e+03]
     [3.24080000e+01 7.50000000e+05]
     [3.29130435e+01 1.50000000e+05]]
```

```
plt.scatter(x1[:,0],x1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Customer Clusters')
plt.xlabel('Age ')
plt.ylabel('Price')
plt.show()
```



```
x2 = df.loc[:,['Age ', 'Education Level']].values
from sklearn.cluster import KMeans
wcss= []
for k in range(1,11):
  kmeans= KMeans(n_clusters=k, init='k-means++')
  kmeans.fit(x2)
  wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
```



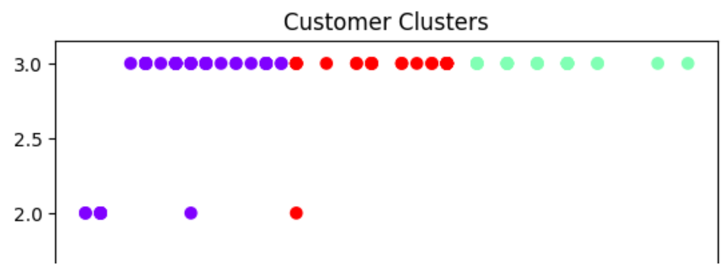
```
kmeans = KMeans(n_clusters=3)
label = kmeans.fit_predict(x2)
print(label)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 0 1 1 1 1 1 1 1 1 2 1 1 1 1 0 0 0 0
 0 2 0 2 2 2 2 2 1 1 0 1 1 1 0 0 0 0 0 0 0 2 0 2 0 2 0 2 2 1 1 1 1 0 0 0 0
 0 0 0 0 2 2 2 2 2 2 2 2 2 2 1 1 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 1 1 0 0
 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 1 1 1 1 0 0 0 0 0 0 0 0 1 2 1 1 1 1 0 0 0 0
 0 2 0 2 2 1 2 1 1 1 1 0 0 0 0 0 2 0 2 2 2 1 1 1 1 0 0 0 0 0 0 0 0 1 2 1 1
 1 1 0 0 0 0 0 2 0 2 2 0 2 2 0 0 0 1 1 1 0 0 2 2 2 2 1 1 1 1 0 0 0 0 0 0 0
 0 0 1 0 0 2 2 2 2 0 2 2 2 2 1 1 1 1 0 0 0 0 0 0 0 0 2 2 2 2 1 1 1 1 0 0 0
 0 0 0 0 2 2 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 2 2 2 2 1 1 1 1 0 0 0 0 0
 0 0 0]
```

```
print(kmeans.cluster_centers_)
```

```
[[23.90540541  1.57432432]
 [47.42666667  1.56      ]
 [36.73684211  1.61842105]]
```

```
plt.scatter(x2[:,0],x2[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Customer Clusters')
plt.xlabel('Annual Income')
plt.ylabel('Price')
plt.show()
```



```
df.head(5)
```

	Age	City	Gender	Education Level	Occupation	Family Members	Annual Income	Charging Hours	Electric Cars are economical	Charging stations	Convert your car to all electric	Hybri c Bot
0	29	Kolkata	Male	3	2	5	150000	1	Yes	15	Yes	Hybr ca
1	29	Mumbai	Male	1	2	4	1000000	1	Yes	8	Yes	Electr ca
2	25	Mumbai	Male	1	2	4	5000	5	Yes	15	Didn't think about it	Hybr ca
3	25	Bangalore	Male	1	2	4	750000	1	Don't know	8	Didn't think about it	Hybr ca
											Didn't	

```
df1 = df.drop(columns=['Income_bin', 'Charging Hours' ,'Charging stations','Income_bin1','Hybrid or Both','Occupation','Gender', 'Educati
```

```
df1.head()
```

	Age	Annual Income
0	29	150000
1	29	1000000
2	25	5000
3	25	750000
4	21	5000

```
x5 = df1.iloc[:,0:]
wcss= []
for k in range(1,11):
  kmeans= KMeans(n_clusters=k, init='k-means++')
  kmeans.fit(x5)
  wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_in
  warnings.warn(
```

```
kmeans = KMeans(n_clusters=3)
label = kmeans.fit_predict(x5)
print(label)
```

```
[1 2 1 0 1 1 0 0 0 0 2 1 2 0 0 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0 0 2 2 2 0 0 0
 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1 1 2 2 2 0 0 2
 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0 0 0 2 1 2 0 0
 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0 0 2 2 2 0 0 0 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1
 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1 1 2 2 2 0 0 2 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1
 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0 0 0 2 1 2 0 0 1 0 1 1 2 1 2 0 2 1 0 1 1 0 0
 0 2 2 2 0 0 0 0 2 2 0 0 0 0 2 0 0 0 2 2 1 1 2 0 0 2 2 0 0 0 1 1 0 0 2 2 1
 1 2 2 2 0 0 2 2 1 2 1 0 0 0 0 1 1 1 1 2 1 1 2 1 1 1 1 0 0 0 2 1 0 1 1 0 0
 0 0 2]
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```

```
print(kmeans.cluster_centers_)
```

```
[[3.24080000e+01 7.50000000e+05]
 [3.29450549e+01 4.16483516e+04]
 [3.41927711e+01 1.00000000e+06]]
```