In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
df=pd.read_csv('Mall_Customers.csv')
df.head()
```

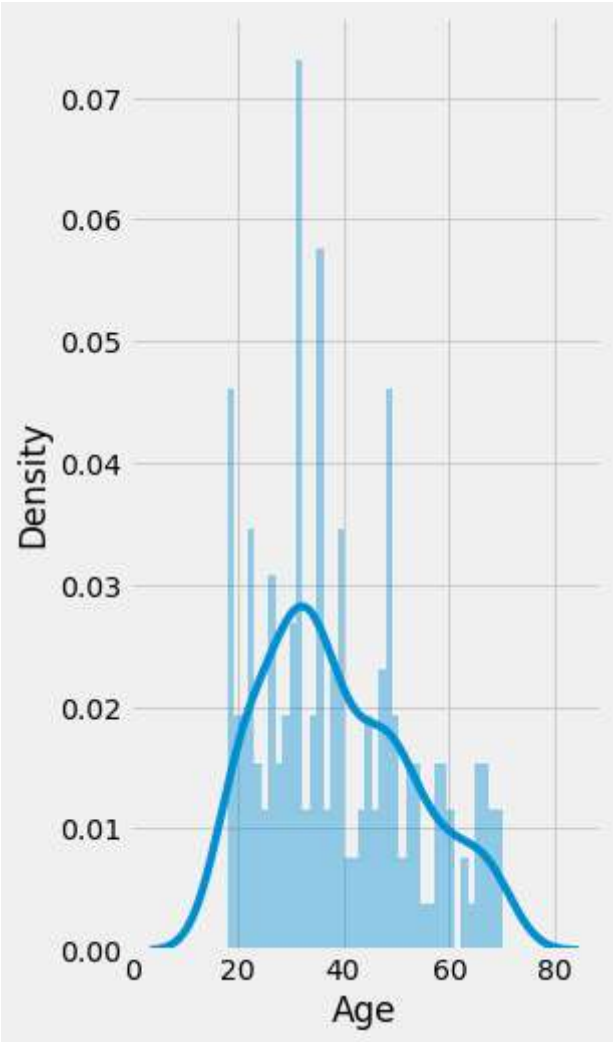
Out[6]:

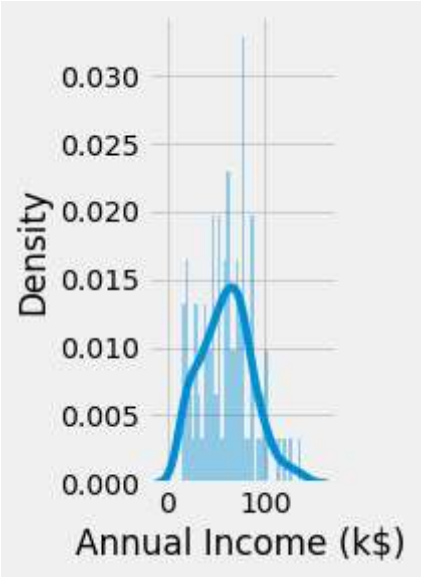
	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

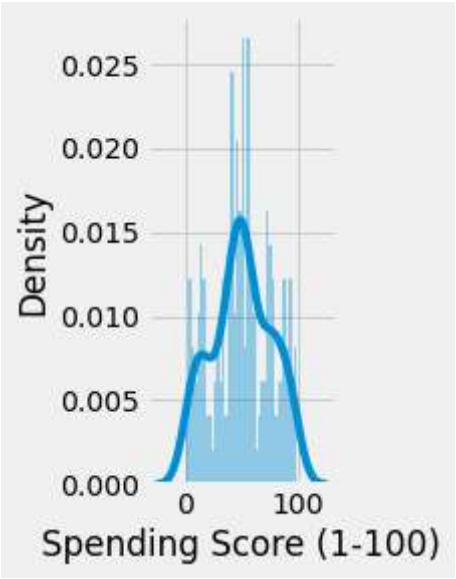
In [7]:
```
df.info()
```

```
<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   Genre                   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
```

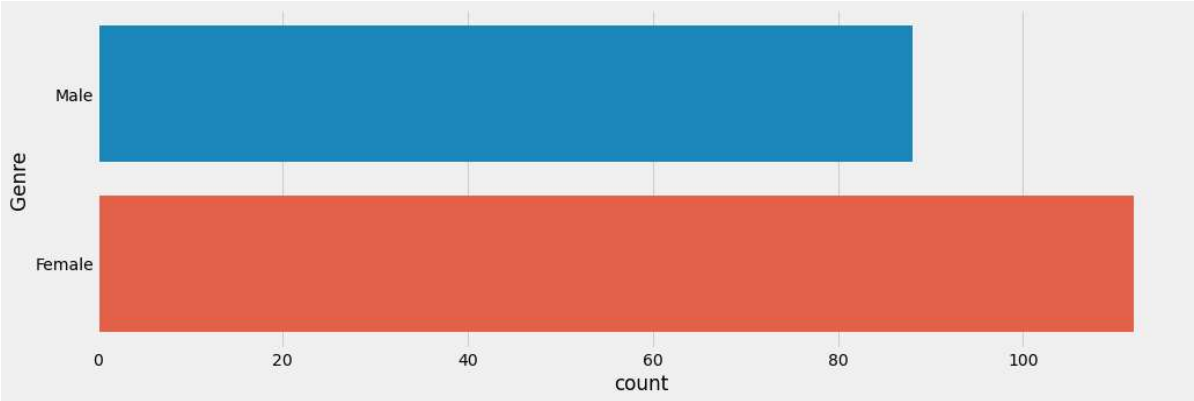
In [8]:
```
plt.style.use("fivethirtyeight")
plt.figure(1,(15,8))
n=0
for x in ['Age','Annual Income (k$)', 'Spending Score (1-100)']:
    n+=1
    plt.subplot(1,3,n)
    plt.subplots_adjust (hspace=0.5,wspace=0.5)
    sns.distplot(df[x], bins=40)
    plt.show()
```



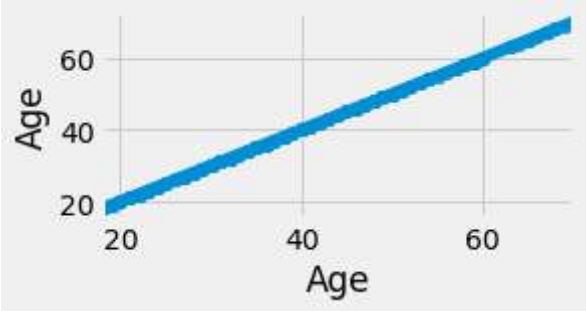


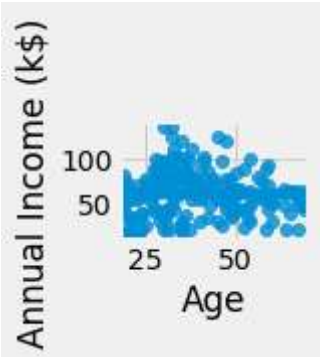


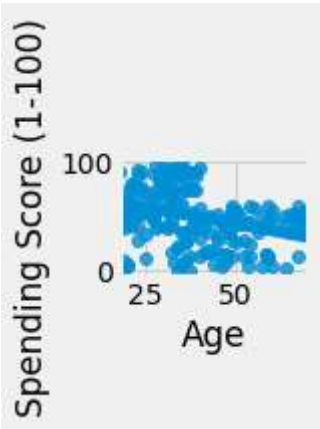
In [9]:
```
plt.figure(1,(15,5))
sns.countplot(y="Genre",data=df)
plt.show()
```

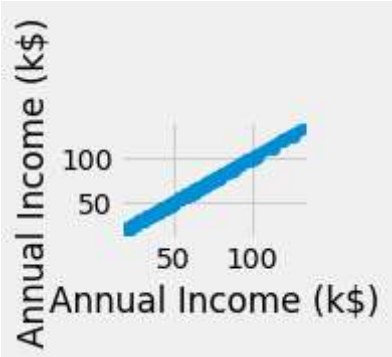


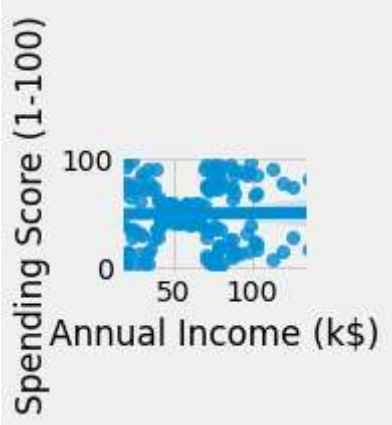
In [10]:
```
plt.style.use("fivethirtyeight")
plt.figure(1,(15,7))
n=0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    for y in  ['Age','Annual Income (k$)','Spending Score (1-100)']:
        n+=1
        plt.subplot(3,3,n)
        plt.subplots_adjust(hspace=0.5,wspace=0.5)
        sns.regplot(x=x,y=y,data=df)
        plt.show()
```



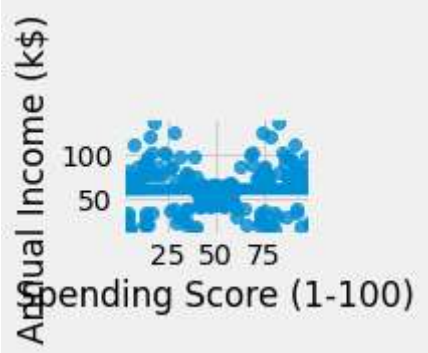


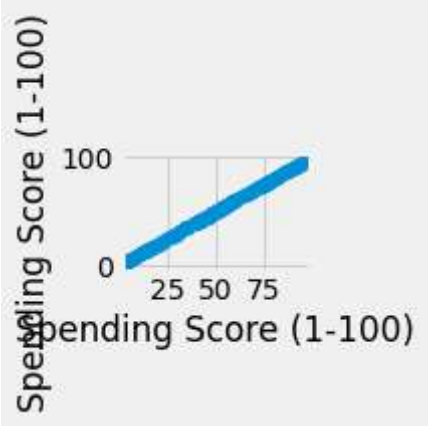



Annual Income (k$)
100
50
50
100
Annual Income (k$)


Spending Score (1-100)
100
0
50
100
Annual Income (k$)


Age
50
25
25 50 75
Spending Score (1-100)


Annual Income (k$)
100
50
25 50 75
Spending Score (1-100)


Spending Score (1-100)
100
0
25 50 75
Spending Score (1-100)

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [12]:
```
df.keys()
```

Out[12]:
```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

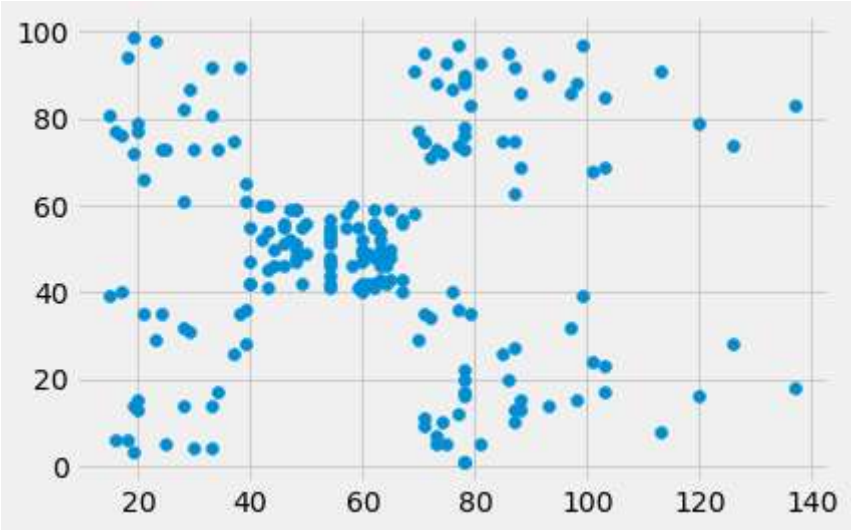
In [13]:
```
df.drop_duplicates(inplace=True)
```

In [14]:
```
df.keys()
```

Out[14]:
```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [16]:
```
plt.scatter(df ['Annual Income (k$)'], df ['Spending Score (1-100)'])
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x1ffb02e1c10>
```



In [31]:
```
X=df.iloc[:,[3,4]].values
```

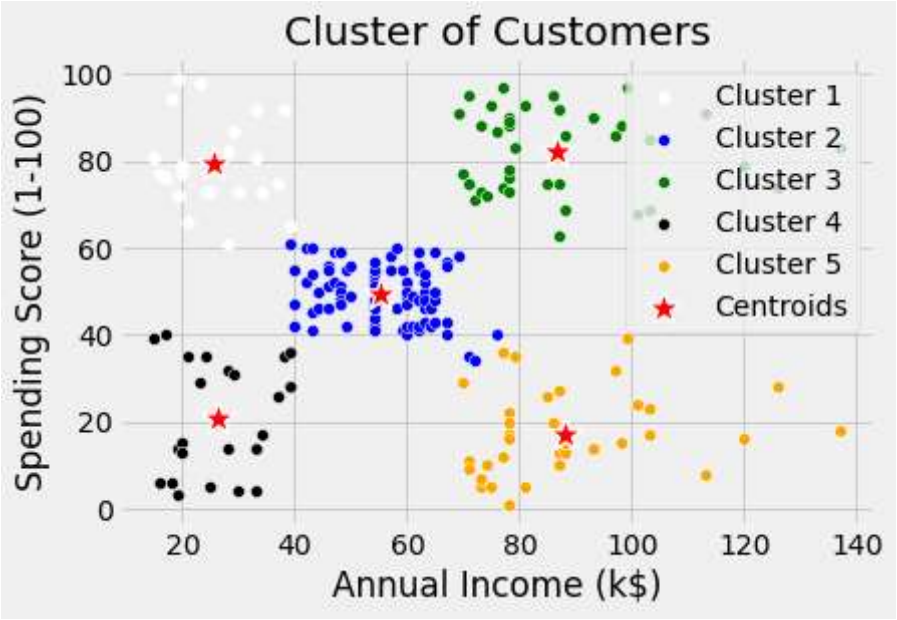
In [25]:
```
from sklearn.cluster import KMeans
```

In [32]:
```
kmeans=KMeans(n_clusters = 5 , init='k-means++',random_state = 42)
y_kmeans=kmeans.fit_predict(X)
```

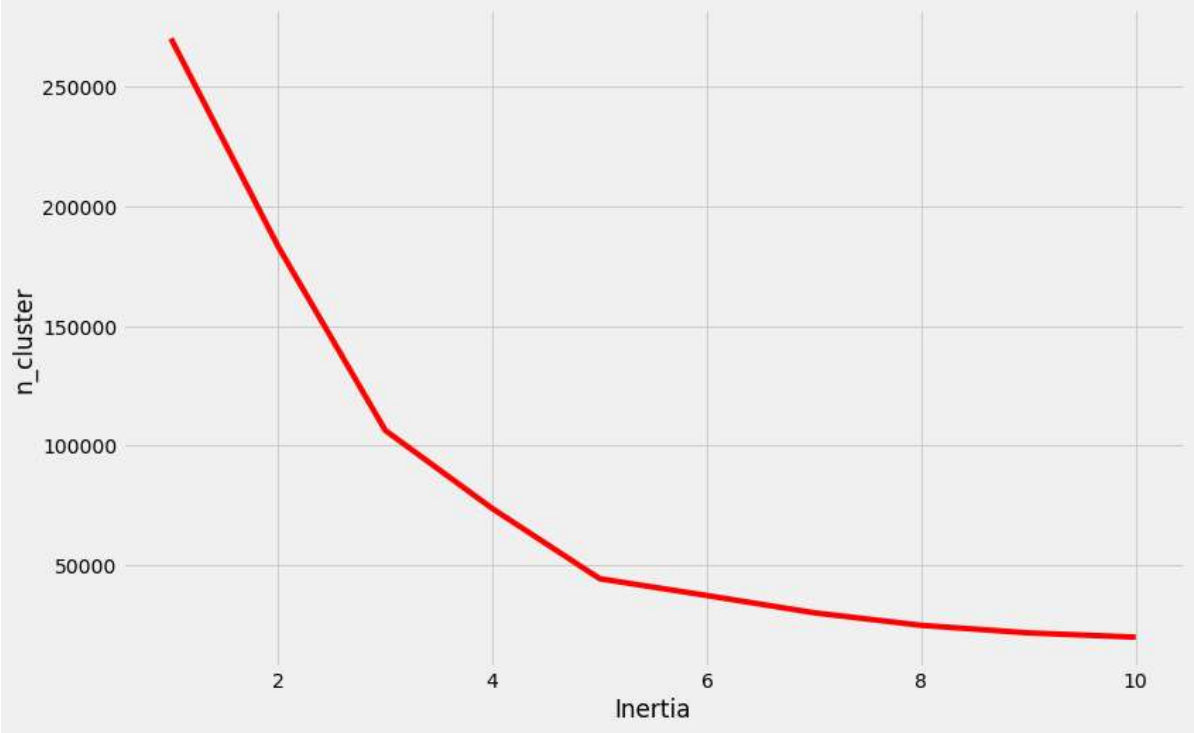
In [33]:
```
sns.scatterplot(X[y_kmeans == 0, 0], X[y_kmeans == 0,1],color ='white', label=
sns.scatterplot(X[y_kmeans == 1, 0], X[y_kmeans == 1,1],color ='blue', label='
sns.scatterplot(X[y_kmeans == 2, 0], X[y_kmeans == 2,1],color ='green', label=
sns.scatterplot(X[y_kmeans == 3, 0], X[y_kmeans == 3,1],color ='black', label=
sns.scatterplot(X[y_kmeans == 4, 0], X[y_kmeans == 4,1],color ='orange', label
sns.scatterplot(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], col

plt.title('Cluster of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
```

Out[33]:
```
<matplotlib.legend.Legend at 0x1ffb14d0850>
```



In [34]:
```
cluster = []
for k in range (1,11):
    kmean = KMeans(n_clusters=k).fit(X)
    cluster.append(kmean.inertia_)
plt.figure(figsize=(12,8))
plt.plot(range(1,11),cluster,'r-')
plt.xlabel('Inertia')
plt.ylabel('n_cluster')
plt.show()
```



In [ ]: