In [1]:
```
#Customer segentation using k mean clustring useding python
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# Loading the data set
mall_customer = pd.read_csv('Mall_customers.csv')
```

In [4]:
```
mall_customer
```

Out[4]:

	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
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 [5]:
```
# Data pre-processing
mall_customer.describe()
```

Out[5]:

	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 [6]:
```
mall_customer.shape
```

Out[6]:
```
(200, 5)
```

In [7]:
```
mall_customer.columns
```

Out[7]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [8]:
```
mall_customer.dtypes
```

Out[8]:
```
CustomerID                int64
Gender                   object
Age                       int64
Annual Income (k$)        int64
Spending Score (1-100)    int64
dtype: object
```

In [9]:
```
mall_customer.isnull().sum()
```

Out[9]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

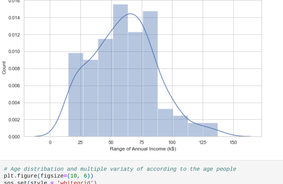
In [10]:
```
mall_customer.corr()
```

Out[10]:

	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

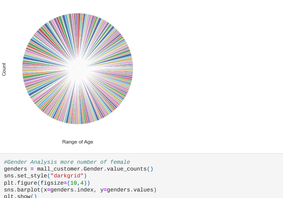
In [11]:
```
# ANNUAL INCOME DISTRIBUTION
# most of annual income 50 to 99 people
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(mall_customer['Annual Income (k$)'])
plt.title('Distribution of Annual Income (k$)', fontsize = 20)
plt.xlabel('Range of Annual Income (k$)')
plt.ylabel('Count')
```

Out[11]:
```
Text(0, 0.5, 'Count')
```



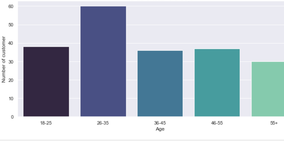
In [12]:
```
# Age distribution and multiple variaty of according to the age people
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
plt.pie(mall_customer['Age'])
plt.title('Distribution of Age', fontsize = 25)
plt.xlabel('Range of Age')
plt.ylabel('Count')
```

Out[12]:
```
Text(0, 0.5, 'Count')
```



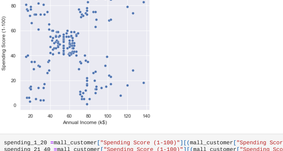
In [14]:
```
#Gender Analysis more number of female
genders = mall_customer.Gender.value_counts()
sns.set_style("darkgrid")
plt.figure(figsize=(10,4))
sns.barplot(x=genders.index, y=genders.values)
plt.show()
```

In [26]:
```
age_18_25 =mall_customer.Age[(mall_customer.Age >=18) & (mall_customer.Age <=25)]
age_26_35 =mall_customer.Age[(mall_customer.Age >=26) & (mall_customer.Age <=35)]
age_36_45 =mall_customer.Age[(mall_customer.Age >=36) & (mall_customer.Age <=45)]
age_46_55 =mall_customer.Age[(mall_customer.Age >=46) & (mall_customer.Age <=55)]
age_55above =mall_customer.Age[mall_customer.Age >=55]
agex =["18-25","26-35","36-45","46-55","55+"]
agey =[len(age_18_25.values),len(age_26_35.values),len(age_36_45.values),len(age_46_55.values),len(age_55above)]
plt.figure(figsize=(12,5))
sns.barplot(x=agex,y=agey, palette="rocket")
plt.title("Number of customer and age")
plt.xlabel("Age")
plt.ylabel("Number of customer")
plt.show()
```



In [28]:
```
sns.relplot(x ="Annual Income (k$)", y="Spending Score (1-100)", data=mall_customer)
```

Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x16d7d843160>
```



In [31]:
```
spending_1_20 =mall_customer["Spending Score (1-100)"][(mall_customer["Spending Score (1-100)"] >=1) & (mall_customer["Spending Score (1-100)"] <=20)]
spending_21_40 =mall_customer["Spending Score (1-100)"][(mall_customer["Spending Score (1-100)"] >=20) & (mall_customer["Spending Score (1-100)"] <=40)]
spending_41_60 =mall_customer["Spending Score (1-100)"][(mall_customer["Spending Score (1-100)"] >=41) & (mall_customer["Spending Score (1-100)"] <=60)]
spending_61_80 =mall_customer["Spending Score (1-100)"][(mall_customer["Spending Score (1-100)"] >=61) & (mall_customer["Spending Score (1-100)"] <=80)]
spending_81_100 =mall_customer["Spending Score (1-100)"][(mall_customer["Spending Score (1-100)"] >=81) & (mall_customer["Spending Score (1-100)"] <=100)]
ssx = ["1-20","21-40","41-60","61-80","81-100"]
ssy = [len(spending_1_20.values),len(spending_21_40.values),len(spending_41_60.values),len(spending_61_80.values),len(spending_81_100.values)]
plt.figure(figsize=(12,5))
sns.barplot(x=ssx,y=ssy, palette="rocket")
plt.title("Spending score")
plt.xlabel("Score")
plt.ylabel("Nmumber of customer having score")
plt.show()
```

In [51]:
```
#Relationship between age and Spending Score (1-100)
x1=df.loc[:,["Age","Spending Score (1-100)"]].values
try:
    wcss=[]
    for k in range(1,11):
        kmeans =KMeans(n_clusters=k, init="k-means++",random_state=0)
except:
    kmeans.fit(x1)
    wcss.append(kmeans.inertia_)
```

In [56]:
```
plt.figure(figsize=(12,5))
plt.grid()
plt.xlabel("k values")
plt.ylabel("wcss")
plt.show()
```

In [70]:
```
try:
    kmeans=KMeans(n_clusters=2)
except:
    label=kmeans.predict(x1)
    print(label)
```

In [85]:
```
centers=kmeans.cluster_centers_
print(centers)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-85-218465de3c35> in <module>
----> 1 centers=kmeans.cluster_centers_
      2 print(centers)

AttributeError: 'KMeans' object has no attribute 'cluster_centers_'
```

In [ ]:
```

```

In [ ]:
```

```