In [3]:

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

In [4]:

```
data = pd.read_csv('Mall_Customers.csv')
data.drop(['CustomerID'],axis=1,inplace=True)
data.columns = ['Gender','Age','Income','Spending']
```

In [5]:

```
data.head()
```

Out[5]:

	Gender	Age	Income	Spending
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

In [6]:

```
data.tail()
```

Out[6]:

	Gender	Age	Income	Spending
195	Female	35	120	79
196	Female	45	126	28
197	Male	32	126	74
198	Male	32	137	18
199	Male	30	137	83

In [7]:

```
data.sample(3)
```

Out[7]:

	Gender	Age	Income	Spending
186	Female	54	101	24
87	Female	22	57	55
188	Female	41	103	17

In [8]:

```
data.describe()
```

Out[8]:

	Age	Income	Spending
count	200.000000	200.000000	200.000000
mean	38.850000	60.560000	50.200000
std	13.969007	26.264721	25.823522
min	18.000000	15.000000	1.000000
25%	28.750000	41.500000	34.750000
50%	36.000000	61.500000	50.000000
75%	49.000000	78.000000	73.000000
max	70.000000	137.000000	99.000000

In [9]:

```
data.info()
```

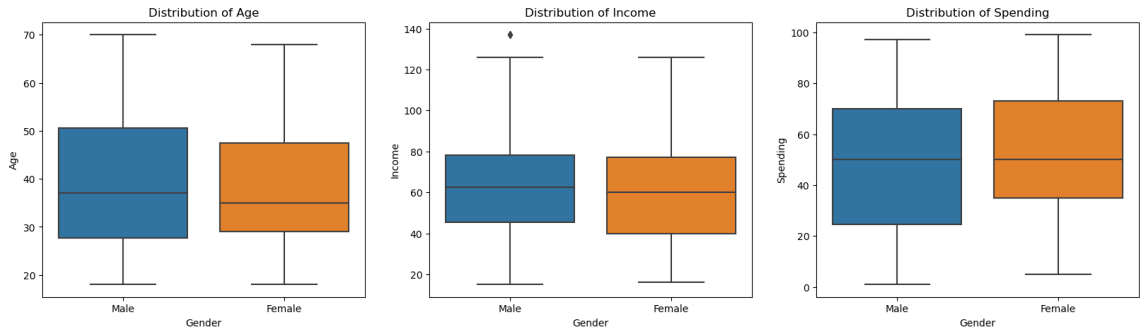
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Gender    200 non-null    object
 1   Age       200 non-null    int64 
 2   Income    200 non-null    int64 
 3   Spending  200 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 6.4+ KB
```

In [10]:

```
#visualization of data
plt.figure(figsize=(20,5))
plt.subplot(1,3,1)
sns.boxplot(x=data.Gender, y=data.Age)
plt.title('Distribution of Age')

plt.subplot(1,3,2)
sns.boxplot(x=data.Gender, y=data.Income)
plt.title('Distribution of Income')

plt.subplot(1,3,3)
sns.boxplot(x=data.Gender, y=data.Spending)
plt.title('Distribution of Spending')
```

Out[10]:

```
Text(0.5, 1.0, 'Distribution of Spending')
```

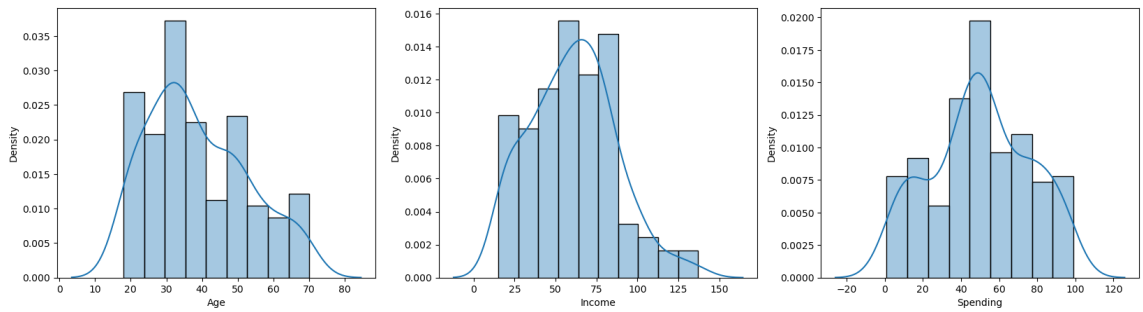


In [11]:

```
plt.figure(figsize=(20,5))

# Density of age
plt.subplot(1,3,1)
sns.histplot(data['Age'],kde=True,stat="density", kde_kws=dict(cut=3), alpha=.4)

# Density of income
plt.subplot(1,3,2)
sns.histplot(data['Income'],kde=True,stat="density", kde_kws=dict(cut=3), alpha=.4)

# Density of spending
plt.subplot(1,3,3)
sns.histplot(data['Spending'],kde=True,stat="density", kde_kws=dict(cut=3), alpha=.4)
```

Out[11]:

```
<AxesSubplot:xlabel='Spending', ylabel='Density'>
```

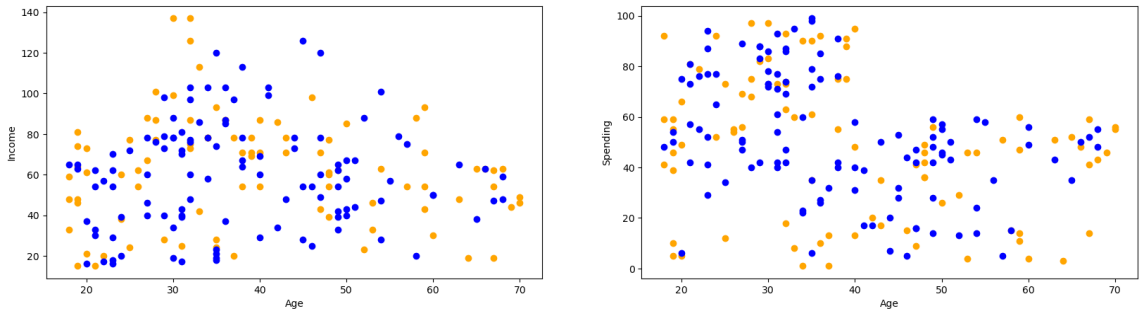


In [12]:

```
female = data[data.Gender == 'Female']
male = data[data.Gender == 'Male']

# Plot
plt.figure(figsize=(20,5))
plt.subplot(1,2,1)
plt.scatter(male.Age,male.Income,color='orange')
plt.scatter(female.Age,female.Income,color='blue')
plt.xlabel('Age')
plt.ylabel('Income')

plt.subplot(1,2,2)
plt.scatter(male.Age,male.Spending,color='orange')
plt.scatter(female.Age,female.Spending,color='blue')
plt.xlabel('Age')
plt.ylabel('Spending')
```

Out[12]:

```
Text(0, 0.5, 'Spending')
```



In [13]:

```
#preparing the data
data['Gender'] = [1 if each == "Female" else 0 for each in  data.loc[:,'Gender']]
data.head()
```

Out[13]:

	Gender	Age	Income	Spending
0	0	19	15	39
1	0	21	15	81
2	1	20	16	6
3	1	23	16	77
4	1	31	17	40

In [14]:

```
data.isnull().sum()
```

Out[14]:

```
Gender      0
Age         0
Income      0
Spending    0
dtype: int64
```

In [15]:

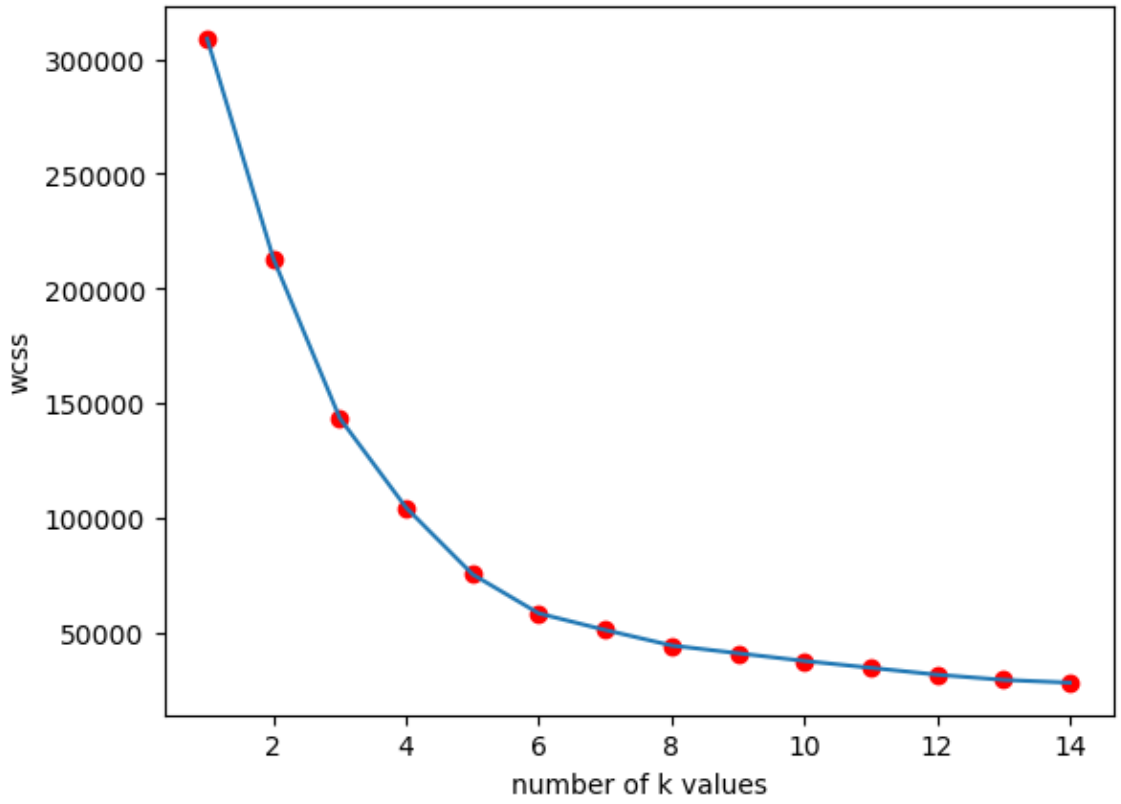
```
data.isna().sum()
```

Out[15]:

```
Gender      0
Age         0
Income      0
Spending    0
dtype: int64
```

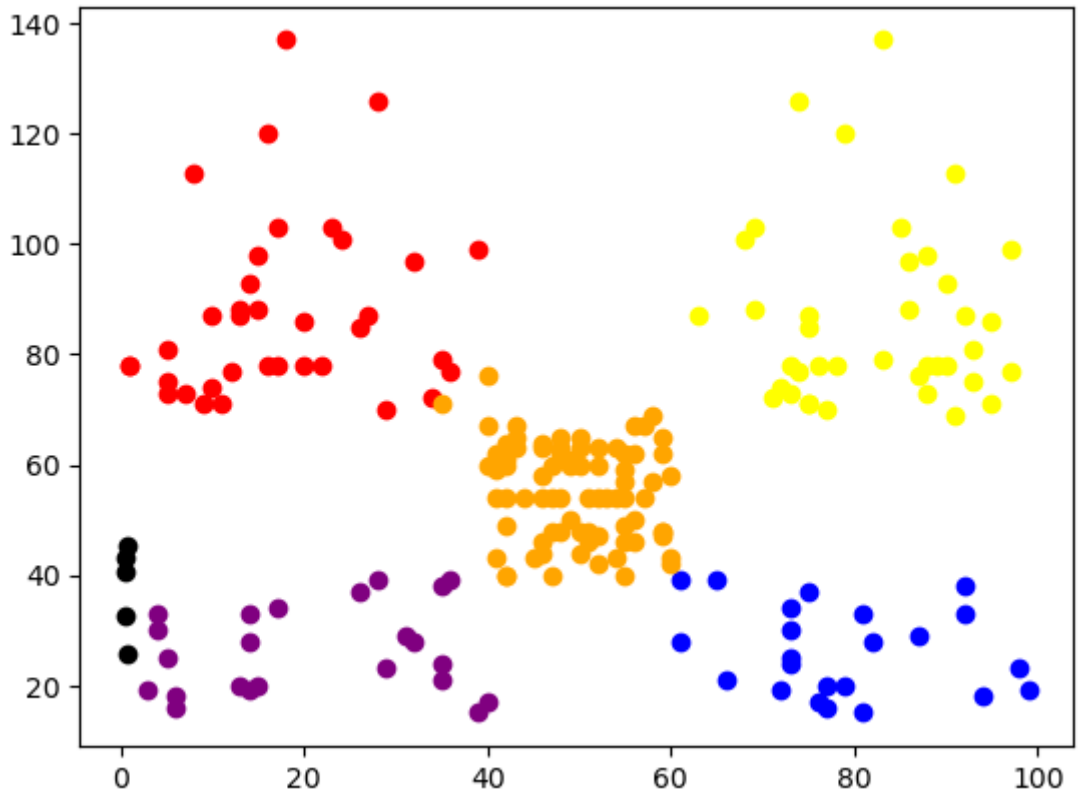
In [16]:

```
#clustering using k-means algorithm
from sklearn.cluster import KMeans
wcss = []

for each in range(1,15):
    kmeans = KMeans(n_clusters=each,init="k-means++",random_state=0)
    kmeans.fit(data)
    wcss.append(kmeans.inertia_)

# Plot
plt.plot(range(1,15), wcss)
plt.scatter(range(1,15),wcss,c='r')
plt.xlabel('number of k values')
plt.ylabel('wcss')
plt.show()
```



In [17]:

```
kmeans_2 = KMeans(n_clusters=5, init="k-means++")

clusters = kmeans_2.fit_predict(data)
data['label'] = clusters

centroid_1 = data[data.label == 0]
centroid_2 = data[data.label == 1]
centroid_3 = data[data.label == 2]
centroid_4 = data[data.label == 3]
centroid_5 = data[data.label == 4]


plt.scatter(centroid_1.Spending,centroid_1.Income,color='red')
plt.scatter(centroid_2.Spending,centroid_2.Income,color='blue')
plt.scatter(centroid_3.Spending,centroid_3.Income,color='orange')
plt.scatter(centroid_4.Spending,centroid_4.Income,color='yellow')
plt.scatter(centroid_5.Spending,centroid_5.Income,color='purple')
plt.scatter(kmeans_2.cluster_centers_[:,0],kmeans_2.cluster_centers_[:,1], color = 'blac
plt.show()
```



In [ ]: