PROJECT NAME:- CUSTOMER SEGMENTATION

SUBMITTED TO:- TECH-A-INTERN

SUBMITTED BY:- SHIVESH PANDEY

LEVEL-1 TASK-2 DATA SCIENTIST

```
import os
os.environ['KAGGLE_CONFIG_DIR']='/content'
```

```
! kaggle datasets download -d imakash3011/customer-personality-analysis
```

```
Downloading customer-personality-analysis.zip to /content
  0% 0.00/62.0k [00:00<?, ?B/s]
100% 62.0k/62.0k [00:00<00:00, 66.7MB/s]
```

```
! chmod 600 /content/kaggle.json
```

```
!unzip \*.zip && rm *.zip
```

```
Archive:  customer-personality-analysis.zip
replace marketing_campaign.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

```
import pandas as pd
df=pd.read_csv('marketing_campaign.csv' ,sep="\t")
```

```
df.head()
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
0	5524	1957	Graduation	Single	58138.0	0	0
1	2174	1954	Graduation	Single	46344.0	1	1
2	4141	1965	Graduation	Together	71613.0	0	0
3	6182	1984	Graduation	Together	26646.0	1	0
4	5324	1981	PhD	Married	58293.0	1	0

5 rows × 29 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
```

```
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   MntWines             2240 non-null   int64
 10  MntFruits            2240 non-null   int64
 11  MntMeatProducts      2240 non-null   int64
 12  MntFishProducts      2240 non-null   int64
 13  MntSweetProducts     2240 non-null   int64
 14  MntGoldProds         2240 non-null   int64
 15  NumDealsPurchases    2240 non-null   int64
 16  NumWebPurchases      2240 non-null   int64
 17  NumCatalogPurchases  2240 non-null   int64
 18  NumStorePurchases    2240 non-null   int64
 19  NumWebVisitsMonth    2240 non-null   int64
 20  AcceptedCmp3         2240 non-null   int64
 21  AcceptedCmp4         2240 non-null   int64
 22  AcceptedCmp5         2240 non-null   int64
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Complain             2240 non-null   int64
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

```
df.shape
```

```
(2240, 29)
```

```
df.dropna(inplace=True)
```

```
df.shape
```

```
(2216, 29)
```

```
df.drop_duplicates(inplace=True)
```

```
df.shape
```

```
(2216, 29)
```

```
from datetime import date,datetime

now=datetime.now()
year=now.strftime("%Y")
year
```

```
'2024'
```

```
df['age']=int(year)-df.Year_Birth
```

```
df['spend']=df.MntFishProducts+df.MntFruits+df.MntGoldProds+df.MntMeatProducts+
```

```
today=date.today()
print(today)
```

```
2024-02-23
```

```
df['seniority']=pd.to_datetime(df.Dt_Customer,dayfirst=True,format='%d-%m-%Y')
df.seniority
```

```
0      2012-09-04
1      2014-03-08
2      2013-08-21
3      2014-02-10
4      2014-01-19
          ...
2235   2013-06-13
2236   2014-06-10
2237   2014-01-25
2238   2014-01-24
2239   2012-10-15
Name: seniority, Length: 2216, dtype: datetime64[ns]
```

```
df.seniority=pd.to_numeric(df.seniority.dt.date.apply(lambda x: (today-x)).dt.d
df.seniority
```

```
0       139.633333
1       121.300000
2       127.933333
3       122.166667
4       122.900000
           ...
2235    130.233333
2236    118.166667
2237    122.700000
2238    122.733333
2239    138.266667
Name: seniority, Length: 2216, dtype: float64
```

```
df.head()
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
0	5524	1957	Graduation	Single	58138.0	0	0
1	2174	1954	Graduation	Single	46344.0	1	1
2	4141	1965	Graduation	Together	71613.0	0	0

3	6182	1984	Graduation	Together	26646.0	1	0
4	5324	1981	PhD	Married	58293.0	1	0

5 rows × 32 columns

```
import numpy as np


df=df.rename(columns={'NumWebPurchases': "Web",'NumCatalogPurchases':'Catalog',
df['Marital_Status']=df.Marital_Status.replace({'Divorced':'Alone','Single':'Al
df['Education']=df.Education.replace({'Basic':'Undergraduate','2n Cycle':'Under

df['children']=df.Kidhome+df.Teenhome
df['has_child'] = np.where(df.children> 0, 'Has child', 'No child')
df['children'].replace({3: "3 children",2:'2 children',1:'1 child',0:"No child'
df=df.rename(columns={'MntWines': "Wines",'MntFruits':'Fruits','MntMeatProducts
df.head()
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
0	5524	1957	Postgraduate	Alone	58138.0	0	0
1	2174	1954	Postgraduate	Alone	46344.0	1	1
2	4141	1965	Postgraduate	In couple	71613.0	0	0
3	6182	1984	Postgraduate	In couple	26646.0	1	0
4	5324	1981	Postgraduate	In couple	58293.0	1	0

5 rows × 34 columns

```
new_df=df[['age','Education','Marital_Status','Income','spend','seniority','has
new_df.head()
```

	age	Education	Marital_Status	Income	spend	seniority	has_child	c
0	67	Postgraduate	Alone	58138.0	1617	139.633333	No child	
1	70	Postgraduate	Alone	46344.0	27	121.300000	Has child	
2	59	Postgraduate	In couple	71613.0	776	127.933333	No child	
3	40	Postgraduate	In couple	26646.0	53	122.166667	Has child	
4	43	Postgraduate	In couple	58293.0	422	122.900000	Has child	

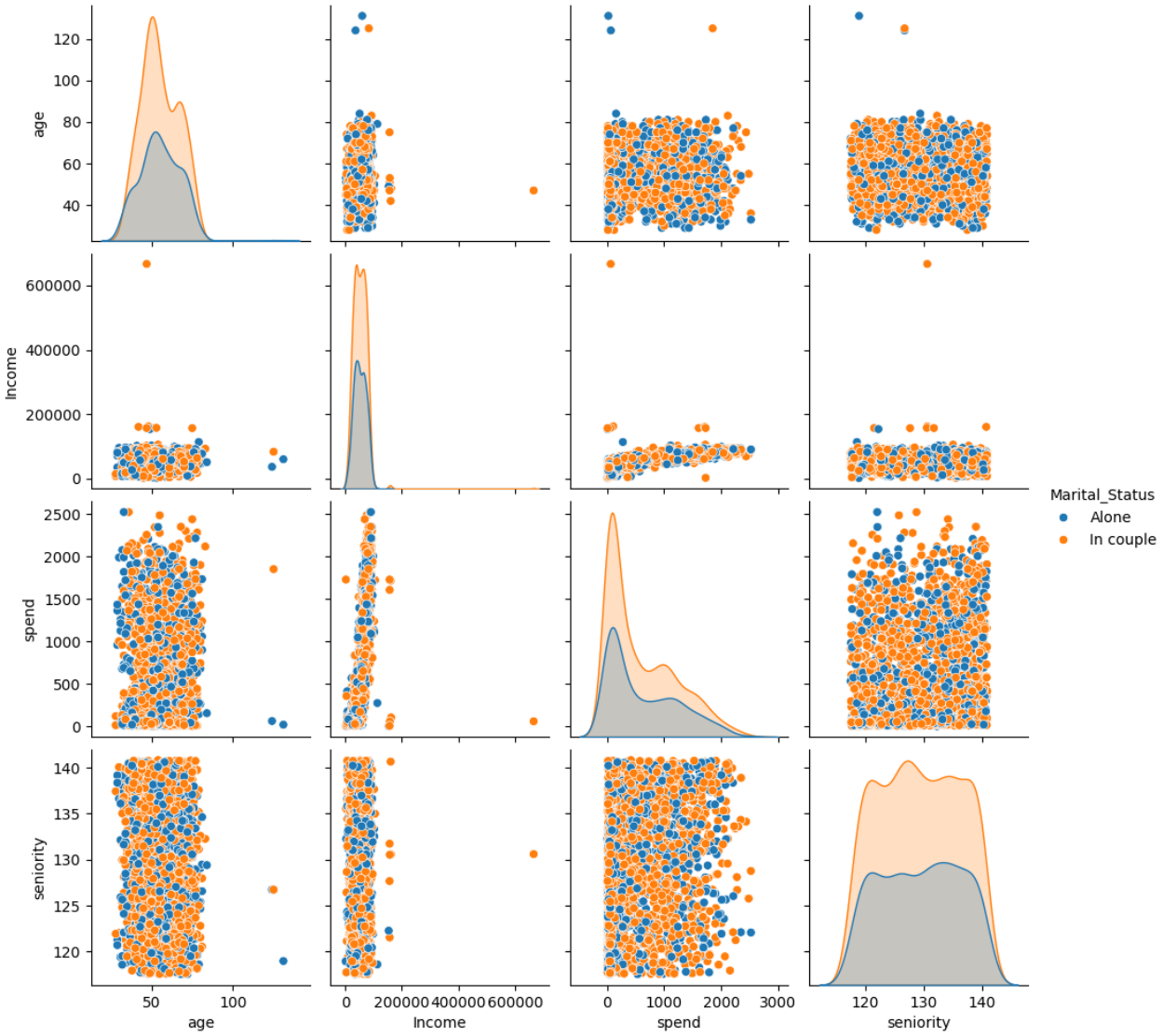
```
new_df.shape
```

```
(2216, 14)
```

```
import seaborn as sns
```

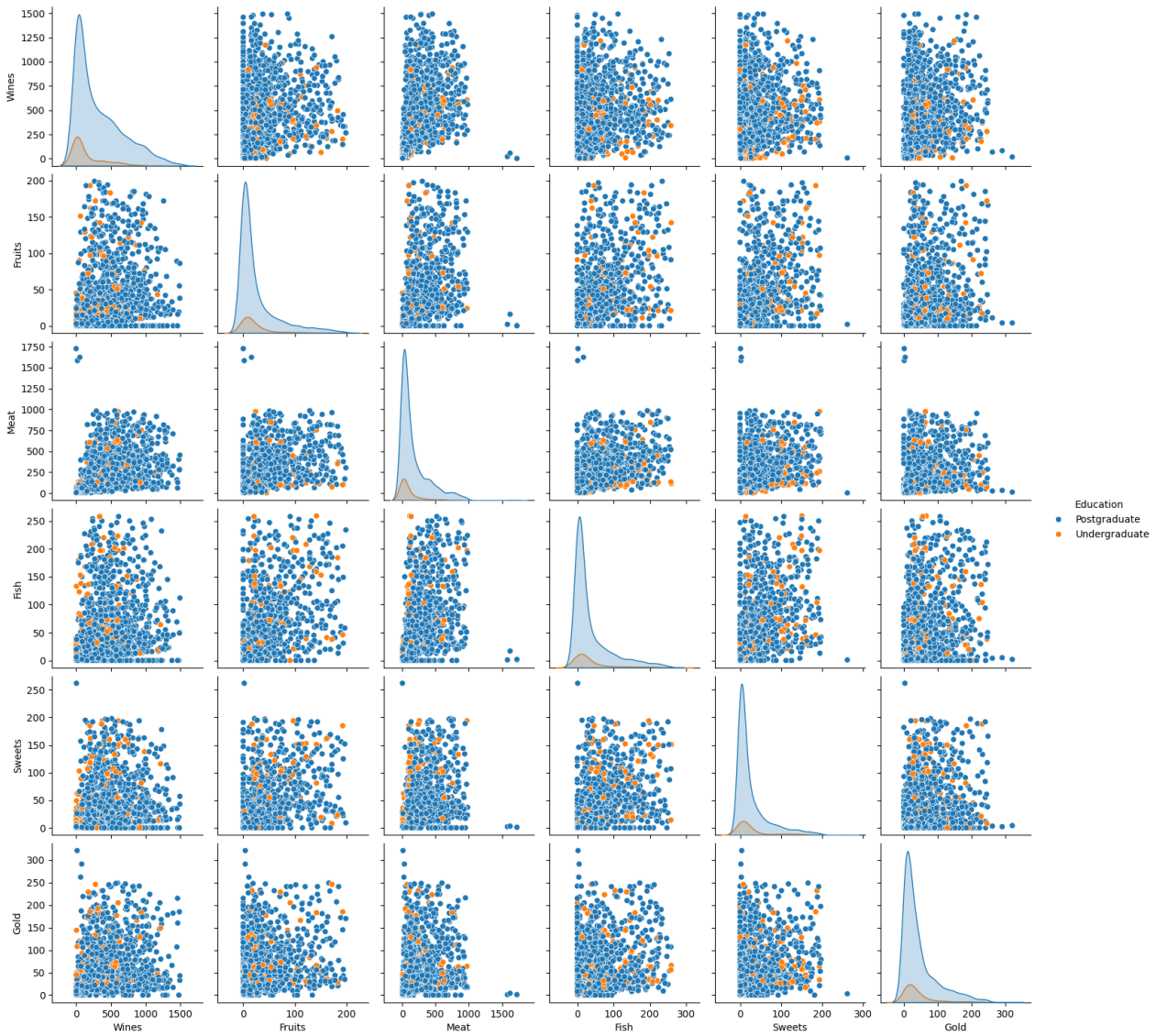
```
import seaborn as sns
import matplotlib.pyplot as plt


sns.pairplot(new_df[['age','Income','spend','seniority','Marital_Status']], hue
plt.show()
```



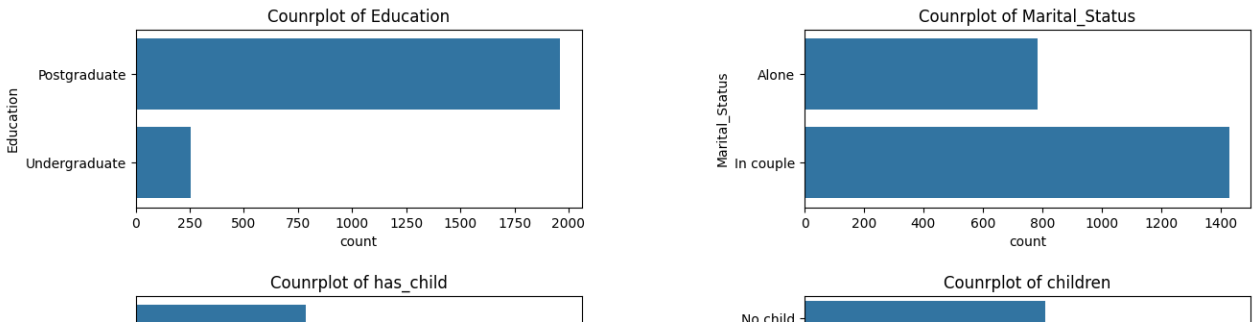
```
plt.figure(figsize=(15,5))
sns.pairplot(new_df[['Wines','Fruits','Meat','Fish','Sweets','Gold','Education'
plt.show()
```

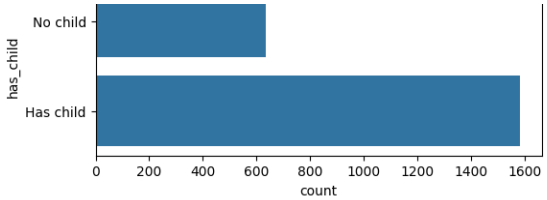
```
<Figure size 1500x500 with 0 Axes>
```

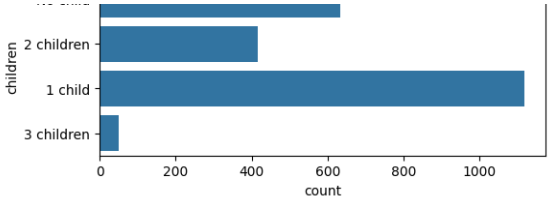


```
plt.figure(figsize=(15,6))
n=0
for x in new_df[['Education','Marital_Status','has_child','children']]:
  n+=1
  plt.subplot(2,2,n)
  plt.subplots_adjust(hspace=0.5,wspace=0.5)
  sns.countplot(y=x,data=new_df)
  plt.title('Counrplot of {}'.format(x))

plt.show()
```

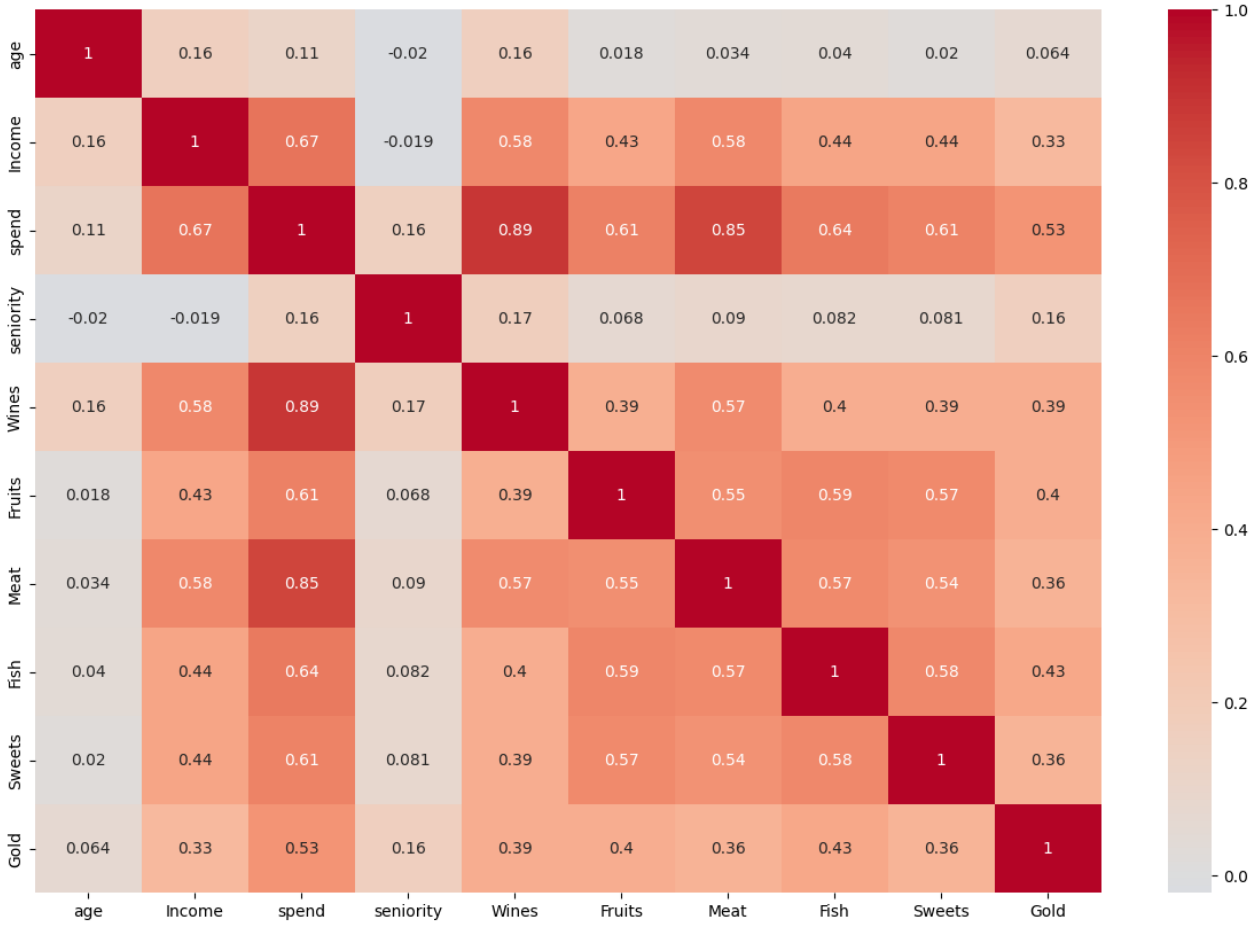




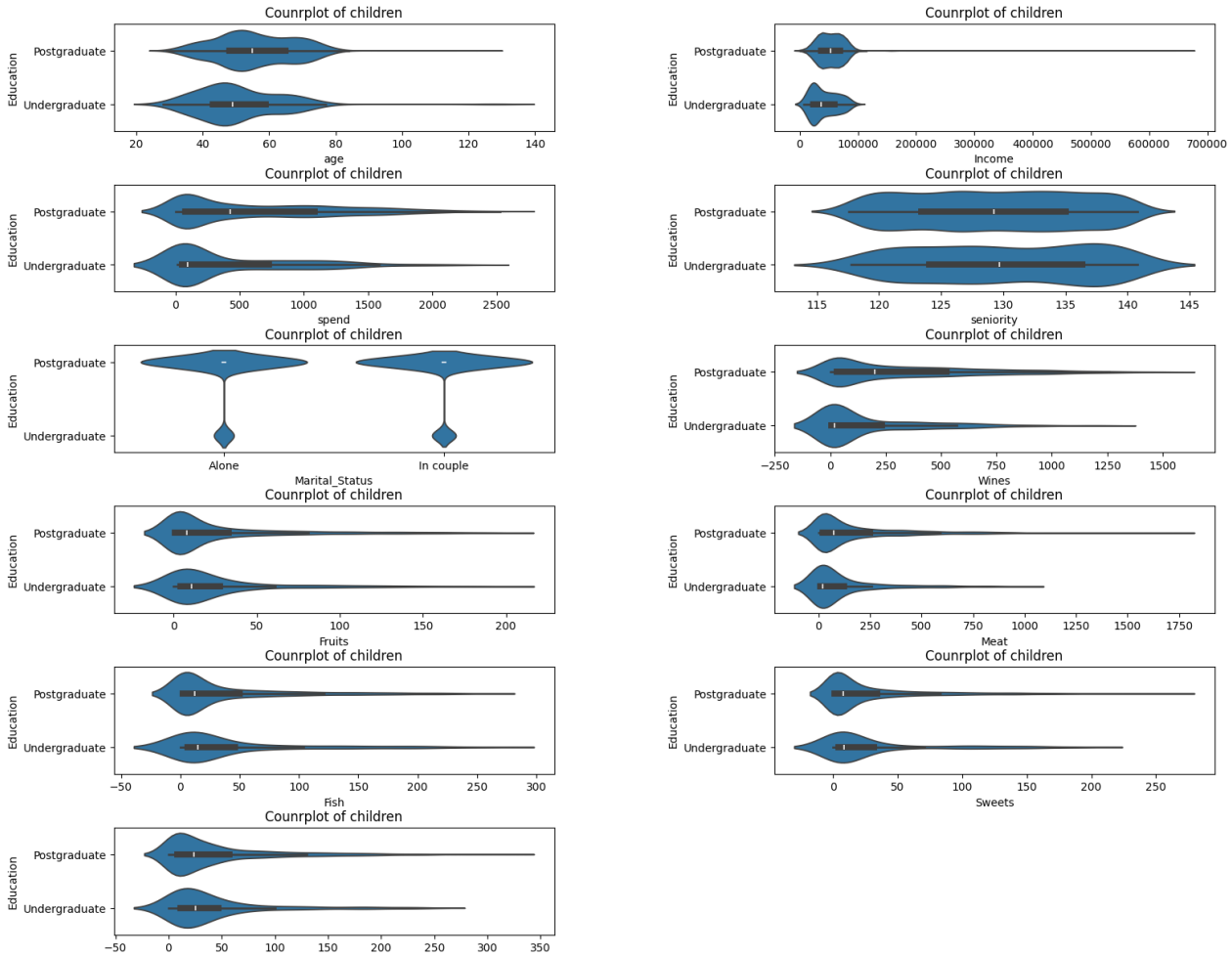



```
corrmat= new_df.corr()
plt.figure(figsize=(15,10))
sns.heatmap(corrmat,annot=True, cmap="coolwarm", center=0)
```

```
<ipython-input-75-ec67a493fb79>:1: FutureWarning: The default value of nume
  corrmat= new_df.corr()
<Axes: >
```



```
plt.figure(figsize=(18,15))
n=0
for i in new_df[['age','Income','spend','seniority','Marital_Status','Wines','F
  n+=1
  plt.subplot(6,2,n)
  plt.subplots_adjust(hspace=0.5,wspace=0.5)
  sns.violinplot(x=i,y='Education',data=new_df)
  plt.title('Counrplot of {}'.format(x))
```



Gold

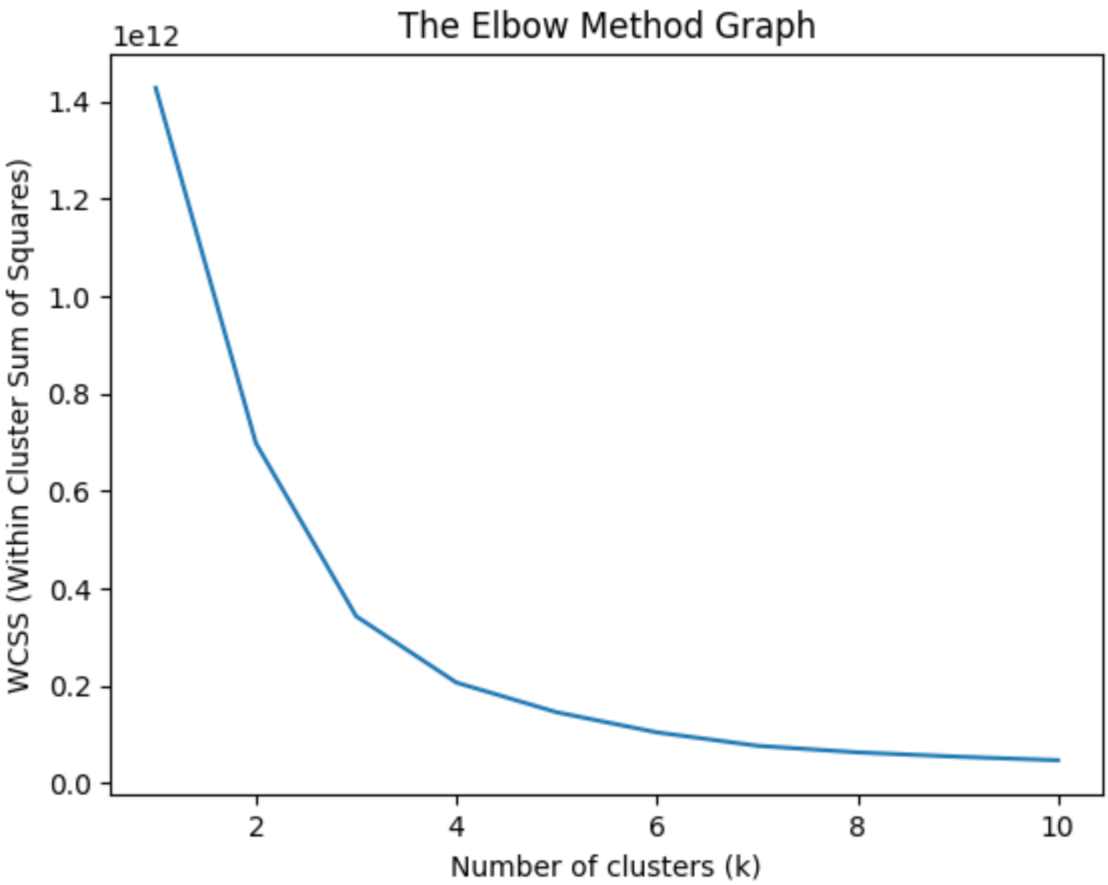
```
from sklearn.cluster import KMeans
wcss_list= []

numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
x = df[numeric_columns].values
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_)
plt.plot(range(1, 11), wcss_list)
plt.title('The Elbow Method Graph')
plt.xlabel('Number of clusters (k)')
```

```
plt.xlabel('Number of clusters (k)')
plt.ylabel('WCSS (Within Cluster Sum of Squares)')
plt.show()
```

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




```
num_clusters = 4
kmeans = KMeans(n_clusters=num_clusters, init='k-means++', random_state=42)
kmeans.fit(X)
new_df['Cluster'] = kmeans.labels_

plt.figure(figsize=(10, 6))
```

```
x_axis = 'Income'
y_axis = 'spend'

for cluster_label in range(num_clusters):
  cluster_data = new_df[new_df['Cluster'] == cluster_label]
  plt.scatter(cluster_data[x_axis], cluster_data[y_axis], label=f'Cluster {clust

plt.title('KMeans Clustering')
plt.xlabel(x_axis)
plt.ylabel(y_axis)
plt.legend()
plt.grid(True)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: Fut
  warnings.warn(
<ipython-input-80-8db975bd60f8>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs
  new_df['Cluster'] = kmeans.labels_
```