In [1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.read_csv(r"C:\Users\Owner\Documents\techtern datas\Marketing_Data.csv")
```

In [3]:
```
df
```

Out[3]:

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	...	marital_Together	marital_Widow	education_2n Cycle	education_Basic
0	58138	0	0	58	635	88	546	172	88	88	...	0	0	0	0
1	46344	1	1	38	11	1	6	2	1	6	...	0	0	0	0
2	71613	0	0	26	426	49	127	111	21	42	...	1	0	0	0
3	26646	1	0	26	11	4	20	10	3	5	...	1	0	0	0
4	58293	1	0	94	173	43	118	46	27	15	...	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2200	61223	0	1	46	709	43	182	42	118	247	...	0	0	0	0
2201	64014	2	1	56	406	0	30	0	0	8	...	1	0	0	0
2202	56981	0	0	91	908	48	217	32	12	24	...	0	0	0	0
2203	69245	0	1	8	428	30	214	80	30	61	...	1	0	0	0
2204	52869	1	1	40	84	3	61	2	1	21	...	0	0	0	0

2205 rows × 39 columns

In [4]:
```
df.shape
```

Out[4]:
```
(2205, 39)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2205 entries, 0 to 2204
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Income                2205 non-null   int64
 1   Kidhome               2205 non-null   int64
 2   Teenhome              2205 non-null   int64
 3   Recency               2205 non-null   int64
 4   MntWines              2205 non-null   int64
 5   MntFruits             2205 non-null   int64
 6   MntMeatProducts       2205 non-null   int64
 7   MntFishProducts       2205 non-null   int64
 8   MntSweetProducts      2205 non-null   int64
 9   MntGoldProds          2205 non-null   int64
 10  NumDealsPurchases     2205 non-null   int64
 11  NumWebPurchases       2205 non-null   int64
 12  NumCatalogPurchases   2205 non-null   int64
 13  NumStorePurchases     2205 non-null   int64
 14  NumWebVisitsMonth     2205 non-null   int64
 15  AcceptedCmp3          2205 non-null   int64
 16  AcceptedCmp4          2205 non-null   int64
 17  AcceptedCmp5          2205 non-null   int64
 18  AcceptedCmp1          2205 non-null   int64
 19  AcceptedCmp2          2205 non-null   int64
 20  Complain              2205 non-null   int64
 21  Z_CostContact         2205 non-null   int64
 22  Z_Revenue             2205 non-null   int64
 23  Response              2205 non-null   int64
 24  Age                   2205 non-null   int64
 25  Customer_Days         2205 non-null   int64
 26  marital_Divorced      2205 non-null   int64
 27  marital_Married       2205 non-null   int64
 28  marital_Single        2205 non-null   int64
 29  marital_Together      2205 non-null   int64
 30  marital_Widow         2205 non-null   int64
 31  education_2n Cycle    2205 non-null   int64
 32  education_Basic       2205 non-null   int64
 33  education_Graduation  2205 non-null   int64
 34  education_Master      2205 non-null   int64
 35  education_PhD         2205 non-null   int64
 36  MntTotal              2205 non-null   int64
 37  MntRegularProds       2205 non-null   int64
 38  AcceptedCmpOverall    2205 non-null   int64
dtypes: int64(39)
memory usage: 672.0 KB
```

In [6]:
```
list(df.columns)
```

Out[6]:

```
['Income',
 'Kidhome',
 'Teenhome',
 'Recency',
 'MntWines',
 'MntFruits',
 'MntMeatProducts',
 'MntFishProducts',
 'MntSweetProducts',
 'MntGoldProds',
 'NumDealsPurchases',
 'NumWebPurchases',
 'NumCatalogPurchases',
 'NumStorePurchases',
 'NumWebVisitsMonth',
 'AcceptedCmp3',
 'AcceptedCmp4',
 'AcceptedCmp5',
 'AcceptedCmp1',
 'AcceptedCmp2',
 'Complain',
 'Z_CostContact',
 'Z_Revenue',
 'Response',
 'Age',
 'Customer_Days',
 'marital_Divorced',
 'marital_Married',
 'marital_Single',
 'marital_Together',
 'marital_Widow',
 'education_2n Cycle',
 'education_Basic',
 'education_Graduation',
 'education_Master',
 'education_PhD',
 'MntTotal',
 'MntRegularProds',
 'AcceptedCmpOverall']
```

In [7]:

```
df.values
```

Out[7]:

```
array([[58138,     0,     0, ...,  1529,  1441,     0],
       [46344,     1,     1, ...,    21,    15,     0],
       [71613,     0,     0, ...,   734,   692,     0],
       ...,
       [56981,     0,     0, ...,  1217,  1193,     1],
       [69245,     0,     1, ...,   782,   721,     0],
       [52869,     1,     1, ...,   151,   130,     0]], dtype=int64)
```

In [8]:

```
df.dtypes
```

Out[8]:

```
Income                  int64
Kidhome                 int64
Teenhome                int64
Recency                 int64
MntWines                int64
MntFruits               int64
MntMeatProducts         int64
MntFishProducts         int64
MntSweetProducts        int64
MntGoldProds            int64
NumDealsPurchases       int64
NumWebPurchases         int64
NumCatalogPurchases     int64
NumStorePurchases       int64
NumWebVisitsMonth       int64
AcceptedCmp3            int64
AcceptedCmp4            int64
AcceptedCmp5            int64
AcceptedCmp1            int64
AcceptedCmp2            int64
Complain                int64
Z_CostContact           int64
Z_Revenue               int64
Response                int64
Age                     int64
Customer_Days           int64
marital_Divorced        int64
marital_Married         int64
marital_Single          int64
marital_Together        int64
marital_Widow           int64
education_2n Cycle      int64
education_Basic         int64
education_Graduation    int64
education_Master        int64
education_PhD           int64
MntTotal                int64
MntRegularProds         int64
AcceptedCmpOverall      int64
dtype: object
```

In [9]:

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

Out[9]:

```
184
```

In [10]:

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

Out[10]:

```
Income                  0
Kidhome                 0
Teenhome                0
Recency                 0
MntWines                0
MntFruits               0
MntMeatProducts         0
MntFishProducts         0
MntSweetProducts        0
MntGoldProds            0
NumDealsPurchases       0
NumWebPurchases         0
NumCatalogPurchases     0
NumStorePurchases       0
NumWebVisitsMonth       0
AcceptedCmp3            0
AcceptedCmp4            0
AcceptedCmp5            0
AcceptedCmp1            0
AcceptedCmp2            0
Complain                0
Z_CostContact           0
Z_Revenue               0
Response                0
Age                     0
Customer_Days           0
marital_Divorced        0
marital_Married         0
marital_Single          0
marital_Together        0
marital_Widow           0
education_2n Cycle      0
education_Basic         0
education_Graduation    0
education_Master        0
education_PhD           0
MntTotal                0
MntRegularProds         0
AcceptedCmpOverall      0
dtype: int64
```

In [11]:

```
df.skew()
```

Out[11]:

```
Income                   0.013164
Kidhome                  0.635495
Teenhome                 0.404623
Recency                 -0.001874
MntWines                 1.166917
MntFruits                2.099281
MntMeatProducts          1.818916
MntFishProducts          1.912028
MntSweetProducts         2.098355
MntGoldProds             1.834468
NumDealsPurchases        2.312369
NumWebPurchases          1.201376
NumCatalogPurchases      1.368122
NumStorePurchases        0.706960
NumWebVisitsMonth        0.229994
AcceptedCmp3             3.259123
AcceptedCmp4             3.246508
AcceptedCmp5             3.284676
AcceptedCmp1             3.551642
AcceptedCmp2             8.402967
Complain                10.363651
Z_CostContact            0.000000
Z_Revenue                0.000000
Response                 1.950559
Age                      0.089941
Customer_Days           -0.019176
marital_Divorced         2.590858
marital_Married          0.463015
marital_Single           1.378865
marital_Together         1.109366
marital_Widow            5.107283
education_2n Cycle       2.871626
education_Basic          6.157110
education_Graduation    -0.019061
education_Master         1.805504
education_PhD            1.382120
MntTotal                 0.915811
MntRegularProds          0.984218
AcceptedCmpOverall       2.719448
dtype: float64
```

In [ ]:

```

```

In [12]:

```
#Data Cleaning

df.drop_duplicates(inplace=True)
```

In [13]:

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

Out[13]:

```
0
```

In [14]:

```
refCol = ['education_2n Cycle', 'education_Basic', 'education_Graduation',
       'education_Master', 'education_PhD'] # hEduc


df['highEd'] = ' '
df['lowEd'] = " "


hList = []
lList = []

for sample in df.values:

    if sample[33] ==1.0:
        hList.append(1.0)
        lList.append(0.0)

    elif sample[34] ==1.0:
        hList.append(1.0)
        lList.append(0.0)
    elif sample[35] ==1.0:
        hList.append(1.0)
        lList.append(0.0)
    else:
        hList.append(0.0)
        lList.append(1.0)
```

In [15]:

```
df['highEd'] = hList
df['lowEd'] = lList
```

In [16]:

```
df['highEd'].value_counts()
```

Out[16]:

```
1.0    1789
0.0     232
Name: highEd, dtype: int64
```

In [17]:

```
df['lowEd'].value_counts()
```

Out[17]:

```
0.0    1789
1.0     232
Name: lowEd, dtype: int64
```

In [18]:

```
# segmenting based on family size

# add all children

df['numKids'] = df['Kidhome'] + df['Teenhome']
```

In [19]:

```
df[['Kidhome', 'Teenhome','numKids' ]]
```

Out[19]:

	Kidhome	Teenhome	numKids
0	0	0	0
1	1	1	2
2	0	0	0
3	1	0	1
4	1	0	1
...	...	...	...
2198	0	0	0
2200	0	1	1
2202	0	0	0
2203	0	1	1
2204	1	1	2

2021 rows × 3 columns

In [20]:

```
#Calculating total number of family

col2 = ['marital_Divorced', 'marital_Married','marital_Single', 'marital_Together', 'marital_Widow']
tFSize = []
for sample in df.values:
    numKids = sample[-1]

    if sample[26] == 1 or sample[28]==1 or sample[30]==1:
        tFSize.append(numKids+1)
    else:
        tFSize.append(numKids+2)
```

In [21]:

```
df['fSize'] = tFSize
```

In [22]:

```
#Segmenting the families

fSize = list(df.fSize)
fSeg = []
for val in fSize:
    if val == 1:
        fSeg.append('mono')
    elif val == 2:
        fSeg.append('duo')
    elif val == 3:
        fSeg.append('Triad')
    elif val == 4:
        fSeg.append('Tetra')
    else:
        fSeg.append('mega')

df['f_Seg'] = fSeg
```

In [23]:

```
#income brackets:

#0 - 20000 : Low income
#20000 - 40000 : Medium Low
#40000 - 75000: Medium High
#75000 - 100000 High
#over 100000 : Wealthy


IncomeBra = []

for sample in df.values:
    if sample[0] >= 0 and sample[0] <= 20000:
        IncomeBra.append('Low income')
    elif sample[0] >= 20000 and sample[0] <= 40000:
        IncomeBra.append('Medium Low')
    elif sample[0] >= 40000 and sample[0] <= 75000:
        IncomeBra.append('Medium High')
    elif sample[0] >= 75000 and sample[0] <= 100000:
        IncomeBra.append('High income')
    else:
        IncomeBra.append('Wealthy')

df['IncomeBrac'] = IncomeBra
```

In [24]:

```
df.Age.min()
```

Out[24]:

```
24
```

In [25]:

```
df.Age.max()
```

Out[25]:

```
80
```

In [26]:

```
#Binning of Age to be able to us to gain insight

df['Age_Group'] = pd.cut(df.Age, bins=[5,12,17,65,85], labels=['Child','Teenager','Adult','Elderly'])
```

In [ ]:

```

```

In [27]:

```
df
```

Out[27]:

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	...	MntTotal	MntRegularProds	AcceptedCmpOverall	highEd	lowE
0	58138	0	0	58	635	88	546	172	88	88	...	1529	1441	0	1.0	0
1	46344	1	1	38	11	1	6	2	1	6	...	21	15	0	1.0	0
2	71613	0	0	26	426	49	127	111	21	42	...	734	692	0	1.0	0
3	26646	1	0	26	11	4	20	10	3	5	...	48	43	0	1.0	0
4	58293	1	0	94	173	43	118	46	27	15	...	407	392	0	1.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2198	26816	0	0	50	5	1	6	3	4	3	...	19	16	0	1.0	0
2200	61223	0	1	46	709	43	182	42	118	247	...	1094	847	0	1.0	0
2202	56981	0	0	91	908	48	217	32	12	24	...	1217	1193	1	1.0	0
2203	69245	0	1	8	428	30	214	80	30	61	...	782	721	0	1.0	0
2204	52869	1	1	40	84	3	61	2	1	21	...	151	130	0	1.0	0

2021 rows × 46 columns

In [28]:

```
df1 = df
df1.to_csv('Cleaned Marketing.csv')
```

In [ ]:

In [29]:

```
df.T
```

Out[29]:

	0	1	2	3	4	5	6	7	8	9	...	2193	2194	2195	2196	2197	2198	2200	2202	2203	2204
Income	58138	46344	71613	26646	58293	62513	55635	33454	30351	5648	...	63777	57967	24434	11012	44802	26816	61223	56981	69245	52869
Kidhome	0	1	0	1	1	0	0	1	1	1	...	1	0	2	1	0	0	0	0	0	1
Teenhome	0	1	0	0	0	1	1	0	0	1	...	1	1	0	0	0	0	1	0	1	1
Recency	58	38	26	26	94	16	34	32	19	68	...	87	39	9	82	71	50	46	91	8	40
MntWines	635	11	426	11	173	520	235	76	14	28	...	457	229	3	24	853	5	709	908	428	84
MntFruits	88	1	49	4	43	42	65	10	0	0	...	5	7	2	3	10	1	43	48	30	3
MntMeatProducts	546	6	127	20	118	98	164	56	24	6	...	106	137	8	26	143	6	182	217	214	61
MntFishProducts	172	2	111	10	46	0	50	3	3	1	...	15	4	20	7	13	3	42	32	80	2
MntSweetProducts	88	1	21	3	27	42	49	1	3	1	...	17	0	0	1	10	4	118	12	30	1
MntGoldProds	88	6	42	5	15	14	27	23	2	13	...	53	91	17	23	20	3	247	24	61	21
NumDealsPurchases	3	2	1	2	5	2	4	2	1	1	...	8	5	2	3	2	1	2	1	2	3
NumWebPurchases	8	1	8	2	5	6	7	4	3	1	...	11	4	2	3	9	0	9	2	6	3
NumCatalogPurchases	10	1	2	0	3	4	3	0	0	0	...	1	2	1	1	4	0	3	3	5	1
NumStorePurchases	4	2	10	4	6	10	7	4	2	0	...	6	8	2	2	12	3	4	13	10	4
NumWebVisitsMonth	7	5	4	6	5	6	6	8	9	20	...	8	5	7	9	8	4	5	6	3	7
AcceptedCmp3	0	0	0	0	0	0	0	0	0	1	...	0	0	0	1	0	0	0	0	0	0
AcceptedCmp4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
AcceptedCmp5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
AcceptedCmp1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
AcceptedCmp2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Complain	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Z_CostContact	3	3	3	3	3	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	3
Z_Revenue	11	11	11	11	11	11	11	11	11	11	...	11	11	11	11	11	11	11	11	11	11
Response	1	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	1
Age	63	66	55	36	39	53	49	35	46	70	...	41	58	48	36	50	34	53	39	64	66
Customer_Days	2822	2272	2471	2298	2320	2452	2752	2576	2547	2267	...	2621	2642	2201	2629	2836	2840	2540	2314	2315	2781
marital_Divorced	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	1	0	0
marital_Married	0	0	0	0	1	0	0	1	0	0	...	1	0	1	0	0	0	1	0	0	1
marital_Single	1	1	0	0	0	0	0	0	0	0	...	0	1	0	1	1	1	0	0	0	0
marital_Together	0	0	1	1	0	1	0	0	1	1	...	0	0	0	0	0	0	0	0	1	0
marital_Widow	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
education_2n Cycle	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
education_Basic	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
education_Graduation	1	1	1	1	0	0	1	0	0	0	...	1	1	1	1	0	1	1	1	0	0
education_Master	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	1	0	0	0	1	0
education_PhD	0	0	0	0	1	0	0	1	1	1	...	0	0	0	0	0	0	0	0	0	1
MntTotal	1529	21	734	48	407	702	563	146	44	36	...	600	377	33	61	1029	19	1094	1217	782	151
MntRegularProds	1441	15	692	43	392	688	536	123	42	23	...	547	286	16	38	1009	16	847	1193	721	130
AcceptedCmpOverall	0	0	0	0	0	0	0	0	0	1	...	0	0	0	1	0	0	0	1	0	0
highEd	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
lowEd	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
numKids	0	2	0	1	1	1	1	1	1	2	...	2	1	2	1	0	0	1	0	1	2
fSize	1.0	3.0	2.0	3.0	3.0	3.0	2.0	3.0	3.0	4.0	...	4.0	2.0	4.0	2.0	1.0	1.0	3.0	1.0	3.0	4.0
f_Seg	mono	Triad	duo	Triad	Triad	Triad	duo	Triad	Triad	Tetra	...	Tetra	duo	Tetra	duo	mono	mono	Triad	mono	Triad	Tetra
IncomeBrac	Medium High	Medium High	Medium High	Medium Low	Medium High	Medium High	Medium High	Medium Low	Medium Low	Low income	...	Medium High	Medium High	Medium Low	Low income	Medium High	Medium Low	Medium High	Medium High	Medium High	Medium High
Age_Group	Adult	Elderly	Adult	Adult	Adult	Adult	Adult	Adult	Adult	Elderly	...	Adult	Adult	Adult	Adult	Adult	Adult	Adult	Adult	Adult	Elderly

46 rows × 2021 columns

In [30]:
```
df.f_Seg.value_counts()
```

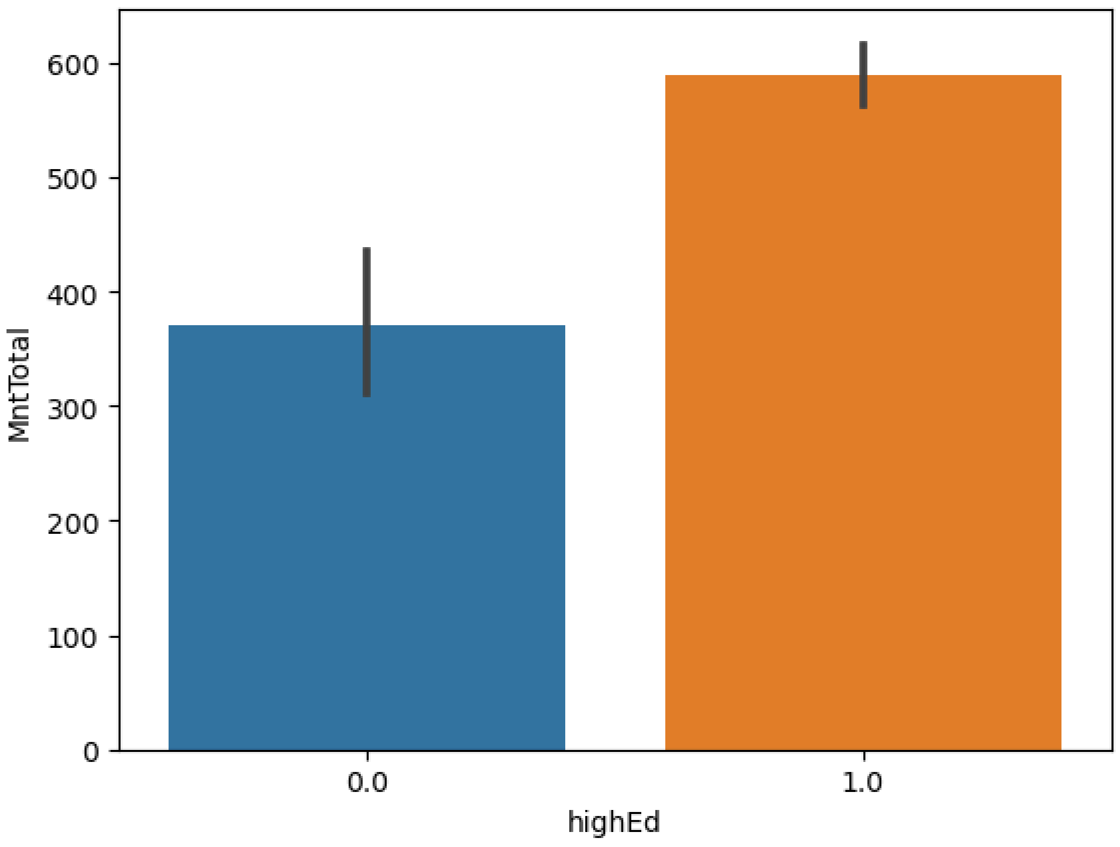
Out[30]:
```
Triad    806
duo      684
Tetra    272
mono     232
mega      27
Name: f_Seg, dtype: int64
```

In [31]:
```
df.f_Seg.unique()
```

Out[31]:
```
array(['mono', 'Triad', 'duo', 'Tetra', 'mega'], dtype=object)
```

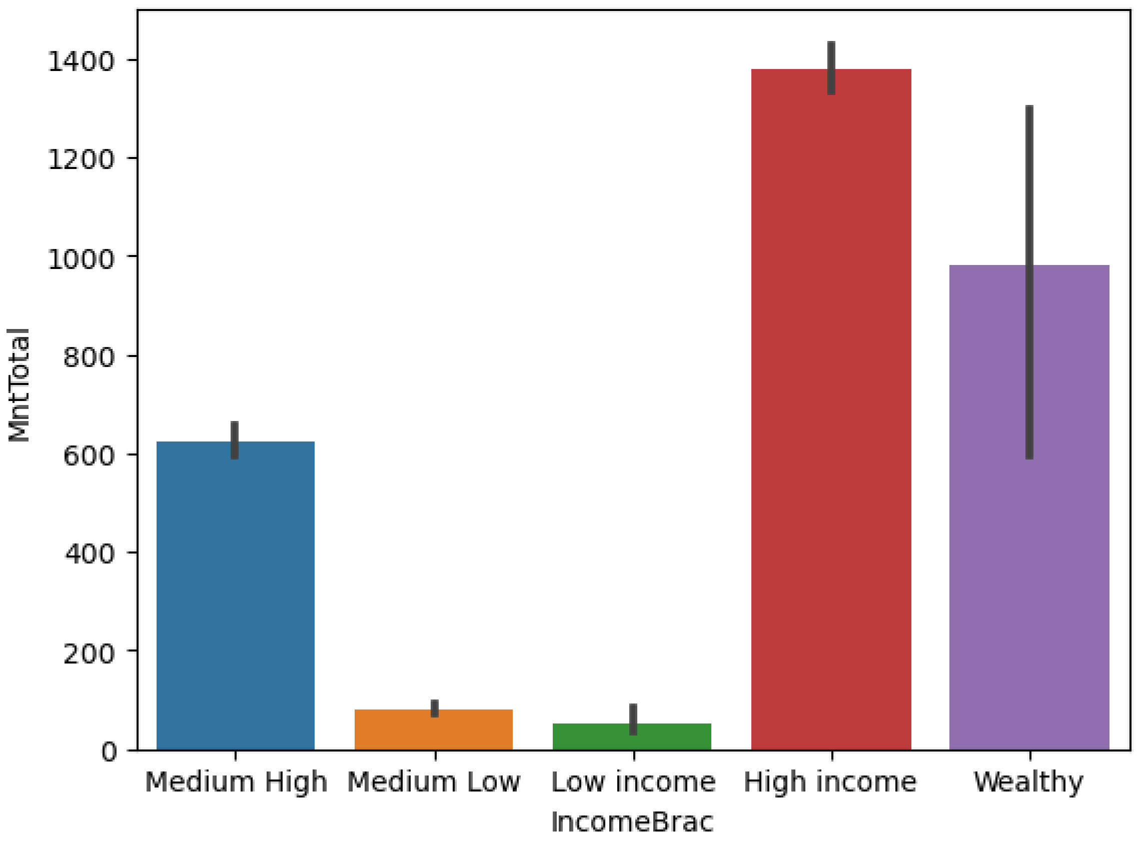
In [32]:
```
sns.barplot(data=df, x="highEd", y="MntTotal")
```

Out[32]:
```
<AxesSubplot:xlabel='highEd', ylabel='MntTotal'>
```



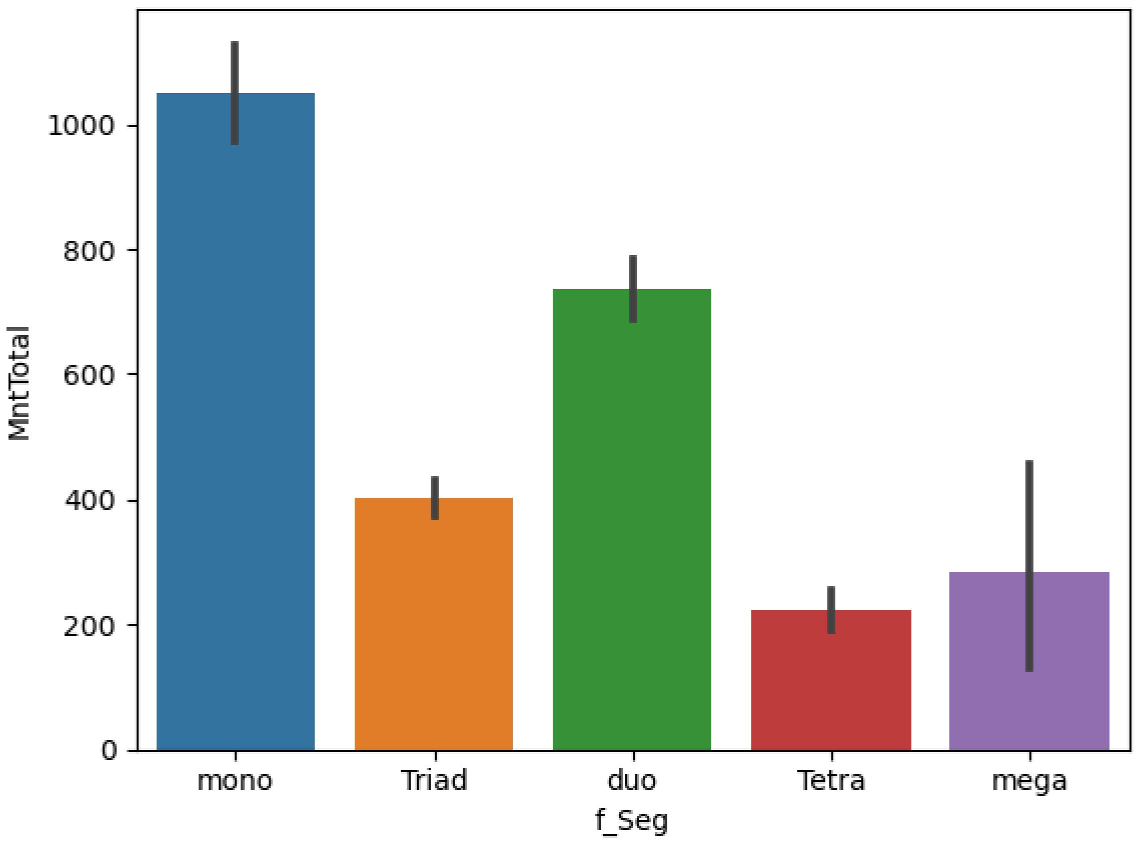
```
In [33]: sns.barplot(data=df, x="IncomeBrac", y="MntTotal")
```

```
Out[33]: <AxesSubplot:xlabel='IncomeBrac', ylabel='MntTotal'>
```



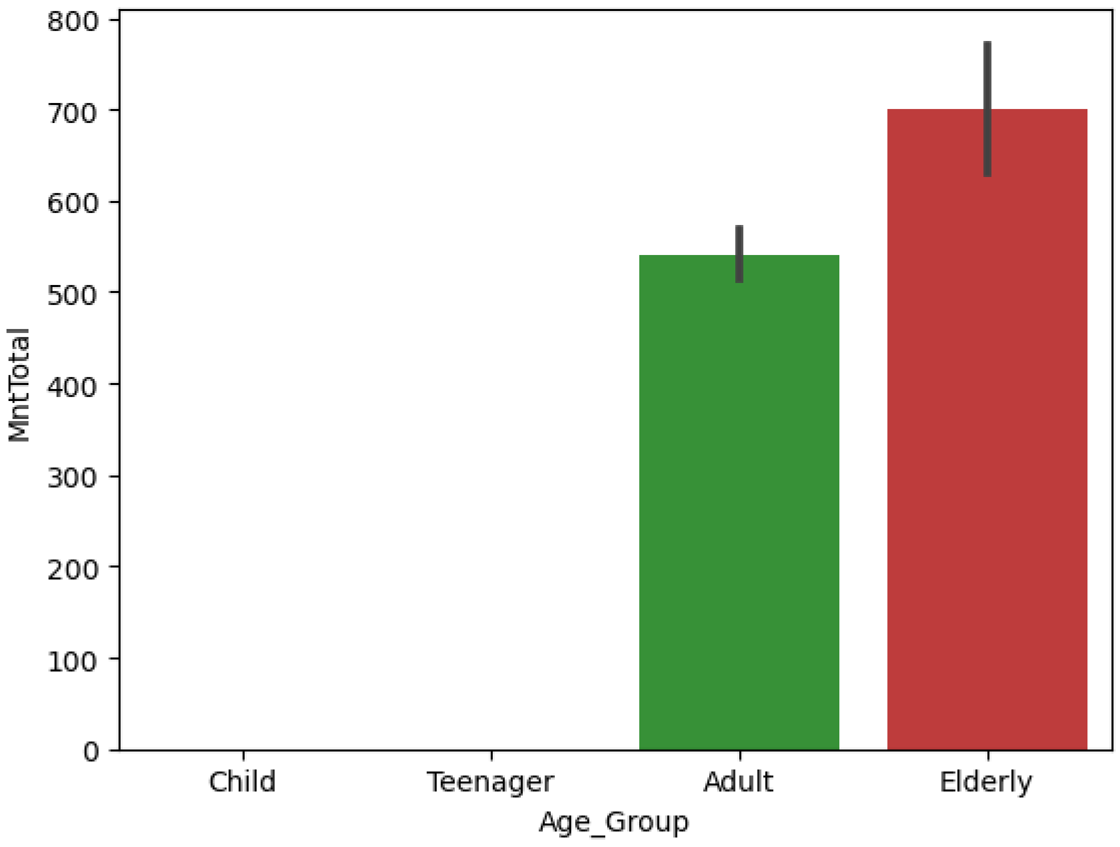
```
In [34]: sns.barplot(data=df, x="f_Seg", y="MntTotal")
```

```
Out[34]: <AxesSubplot:xlabel='f_Seg', ylabel='MntTotal'>
```



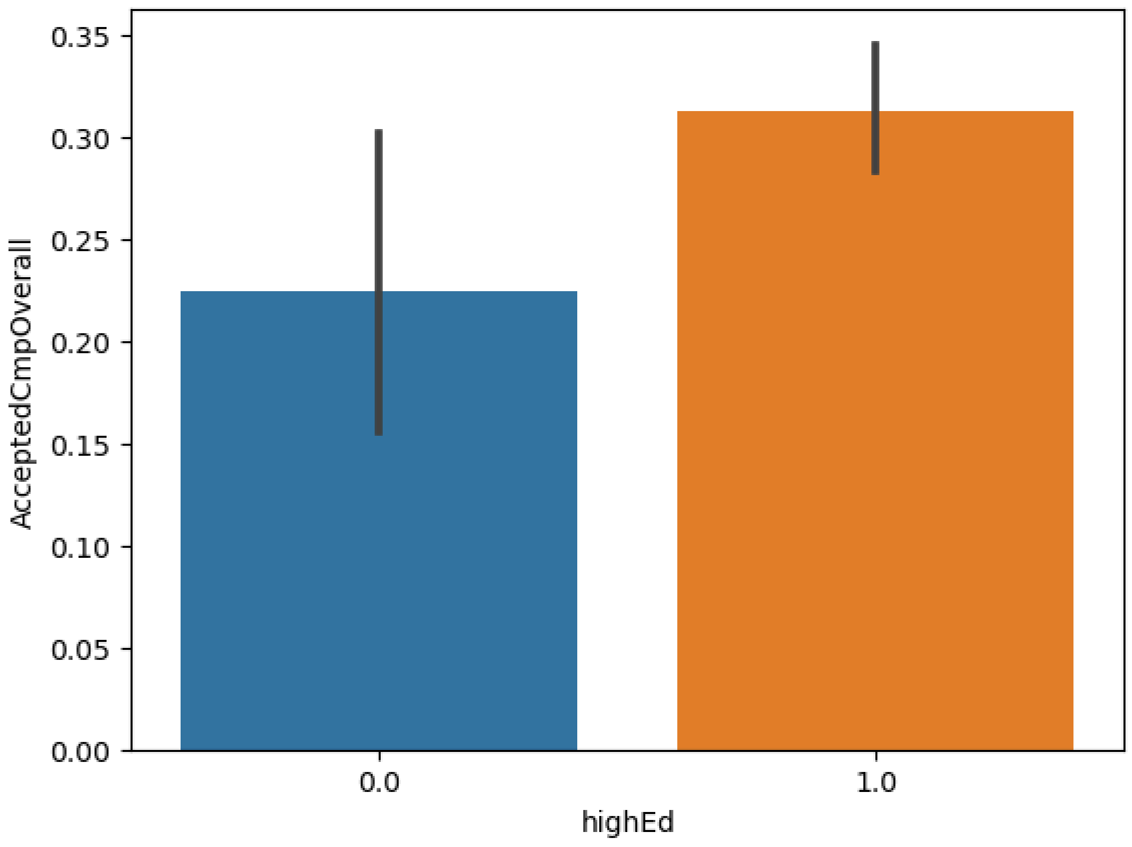
```
In [35]: sns.barplot(data=df, x="Age_Group", y="MntTotal")
```

```
Out[35]: <AxesSubplot:xlabel='Age_Group', ylabel='MntTotal'>
```



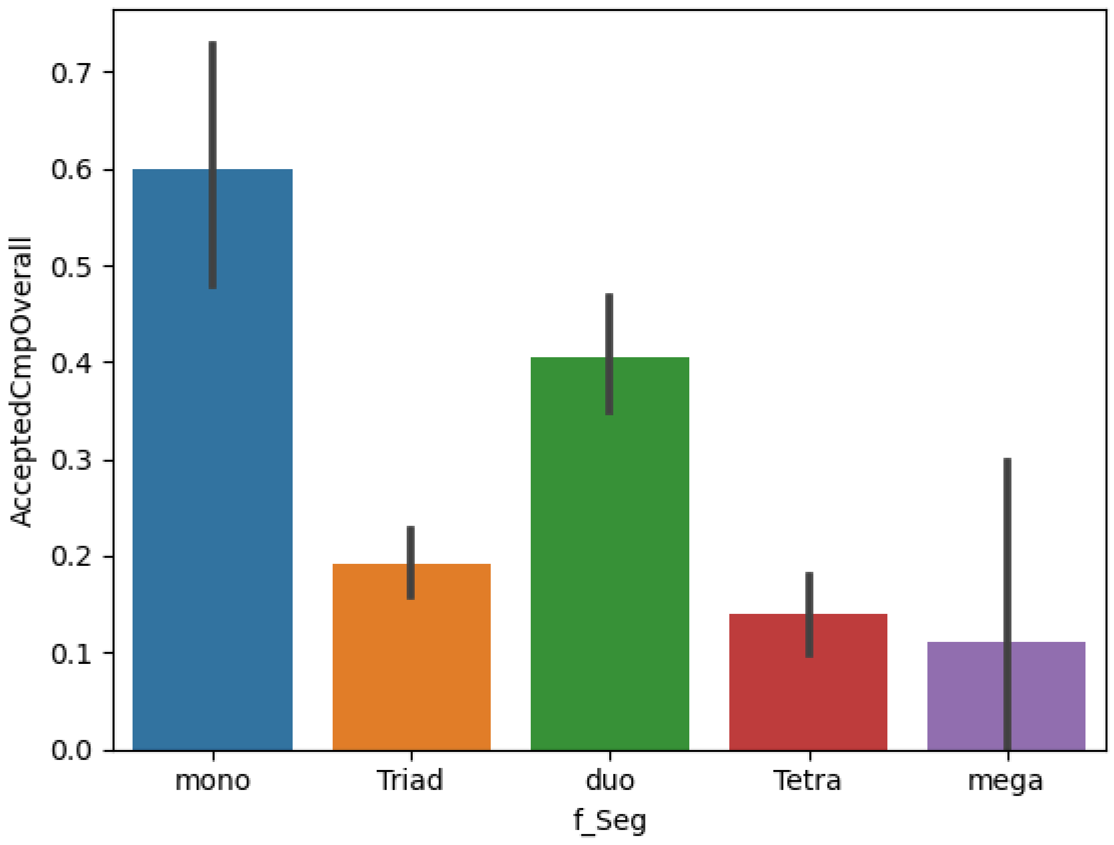
```
In [36]: sns.barplot(data=df, x="highEd", y="AcceptedCmpOverall")
```

```
Out[36]: <AxesSubplot:xlabel='highEd', ylabel='AcceptedCmpOverall'>
```



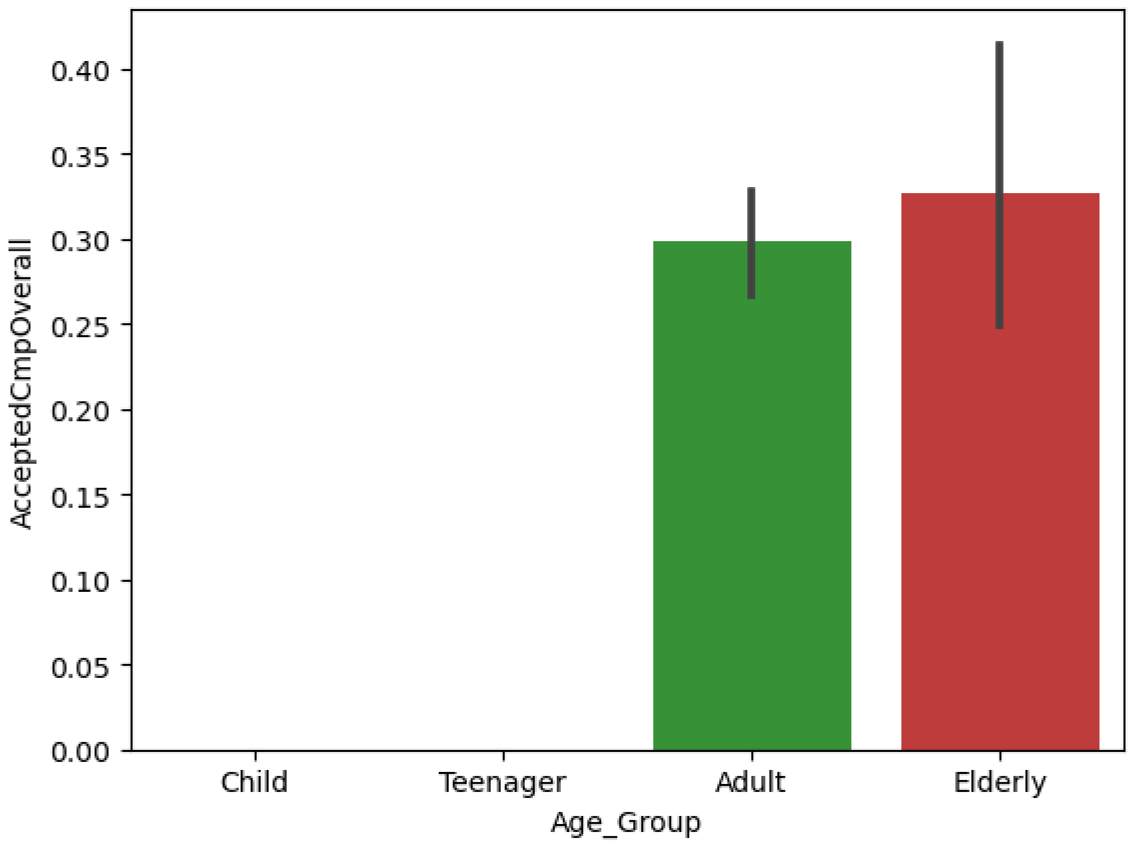
```
In [37]: sns.barplot(data=df, x="f_Seg", y="AcceptedCmpOverall")
```

```
Out[37]: <AxesSubplot:xlabel='f_Seg', ylabel='AcceptedCmpOverall'>
```



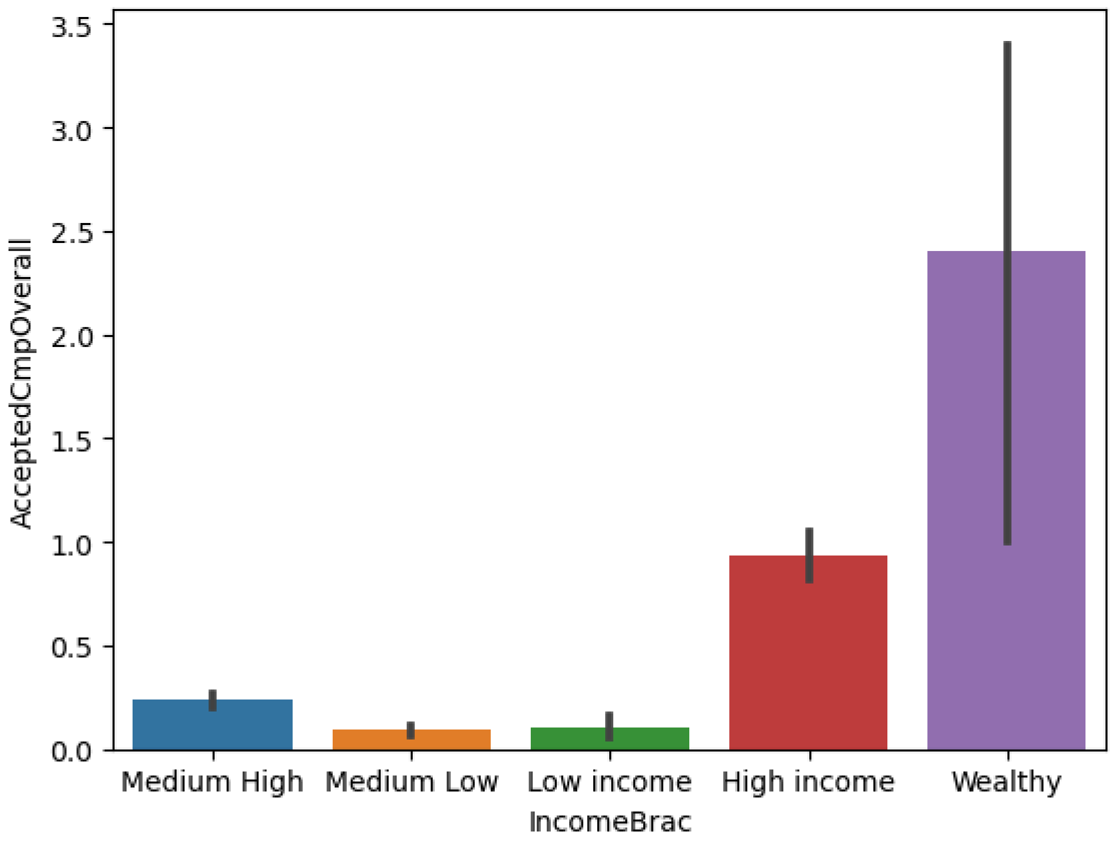
```
In [38]: sns.barplot(data=df, x="Age_Group", y="AcceptedCmpOverall")
```

```
Out[38]: <AxesSubplot:xlabel='Age_Group', ylabel='AcceptedCmpOverall'>
```



```
In [39]: sns.barplot(data=df, x="IncomeBrac", y="AcceptedCmpOverall")
```

```
Out[39]: <AxesSubplot:xlabel='IncomeBrac', ylabel='AcceptedCmpOverall'>
```



```
In [ ]:
```

```
In [40]: eduList = ['education_2n Cycle', 'education_Basic', 'education_Graduation',
                'education_Master', 'education_PhD']
```

```
In [41]: eduList2 = []

         for val in eduList:
```

```
    edu = '_'.join(x for x in  val.split(' '))
    eduList2.append(edu)
```

In [42]:
```
eduList2
```

Out[42]:
```
['education_2n_Cycle',
 'education_Basic',
 'education_Graduation',
 'education_Master',
 'education_PhD']
```

In [43]:
```
def colChanger(data):
    eduList = []
    cols = list(data.columns)
    for val in cols:
        edu = '_'.join(x for x in  val.split(' '))
        eduList.append(edu)
    return eduList
```

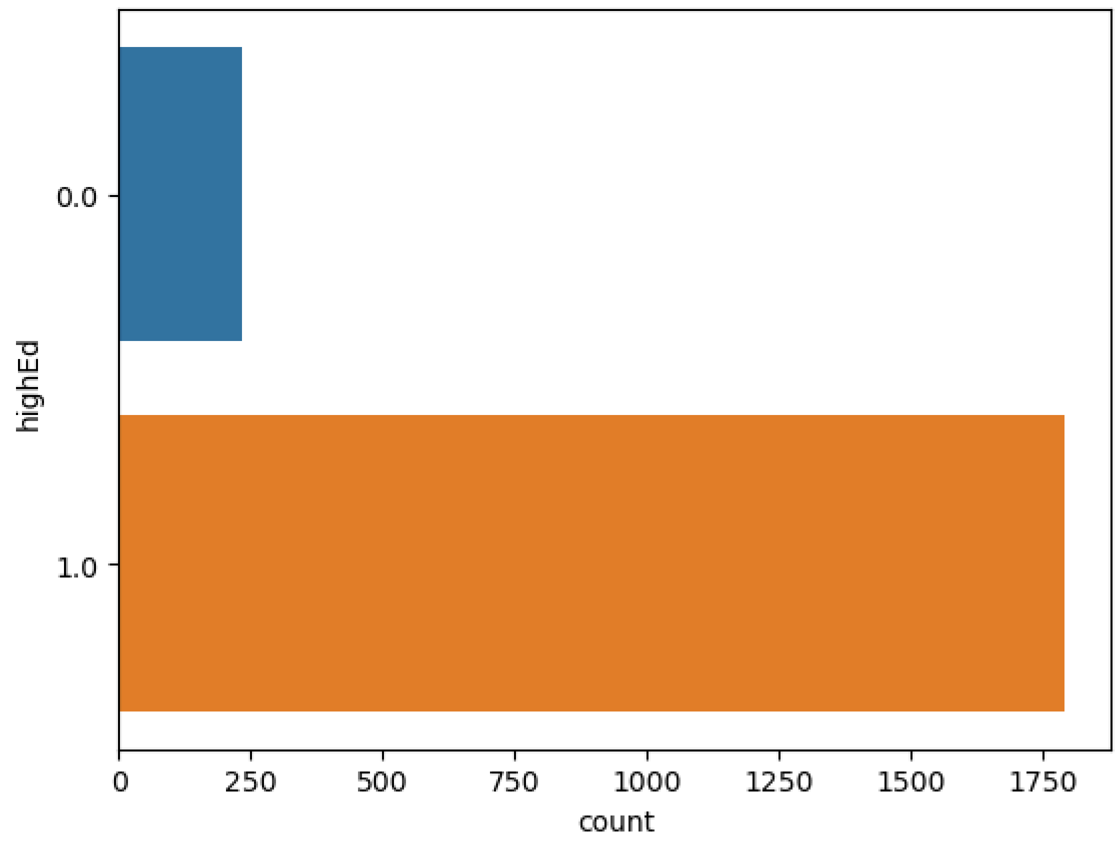
In [44]:
```
df.columns = colChanger(df)
```

In [45]:
```
df.columns
```

Out[45]:
```
Index(['Income', 'Kidhome', 'Teenhome', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response',
       'Age', 'Customer_Days', 'marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow',
       'education_2n_Cycle', 'education_Basic', 'education_Graduation',
       'education_Master', 'education_PhD', 'MntTotal', 'MntRegularProds',
       'AcceptedCmpOverall', 'highEd', 'lowEd', 'numKids', 'fSize', 'f_Seg',
       'IncomeBrac', 'Age_Group'],
      dtype='object')
```

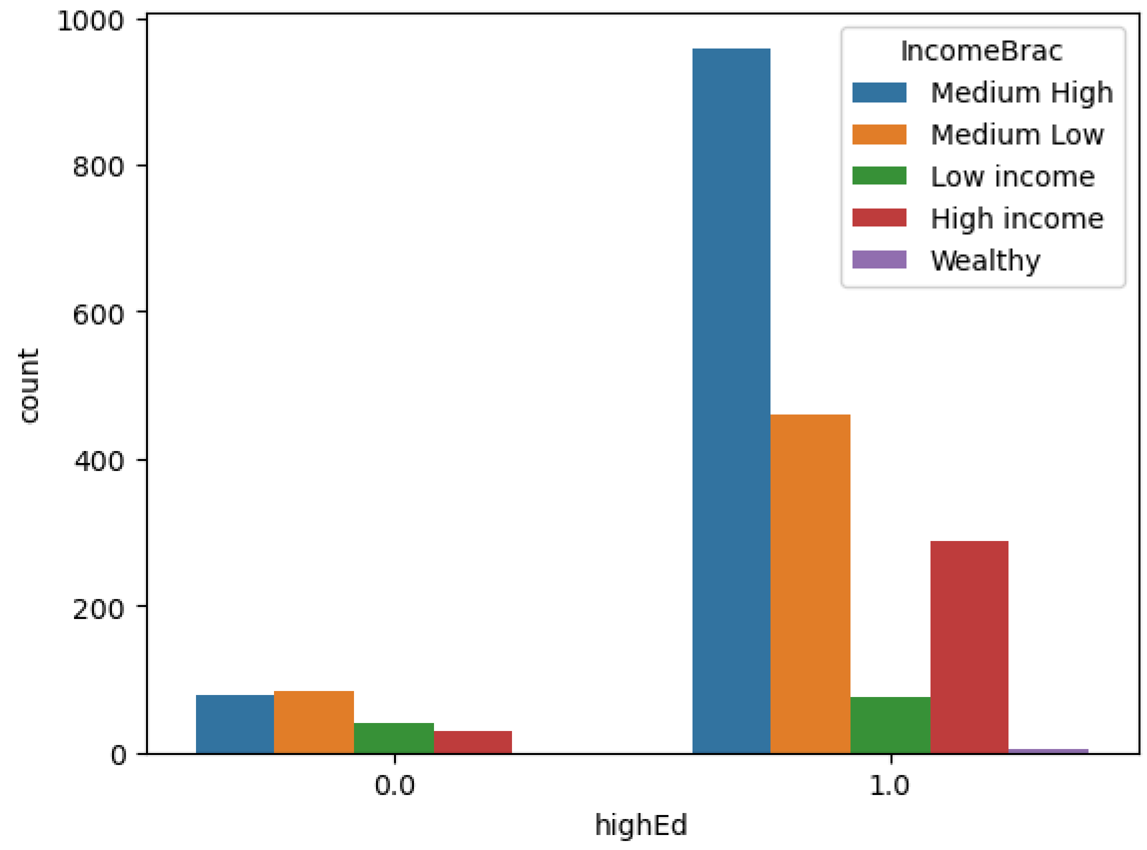
In [46]:
```
sns.countplot(y='highEd', data=df)
```

Out[46]:
```
<AxesSubplot:xlabel='count', ylabel='highEd'>
```

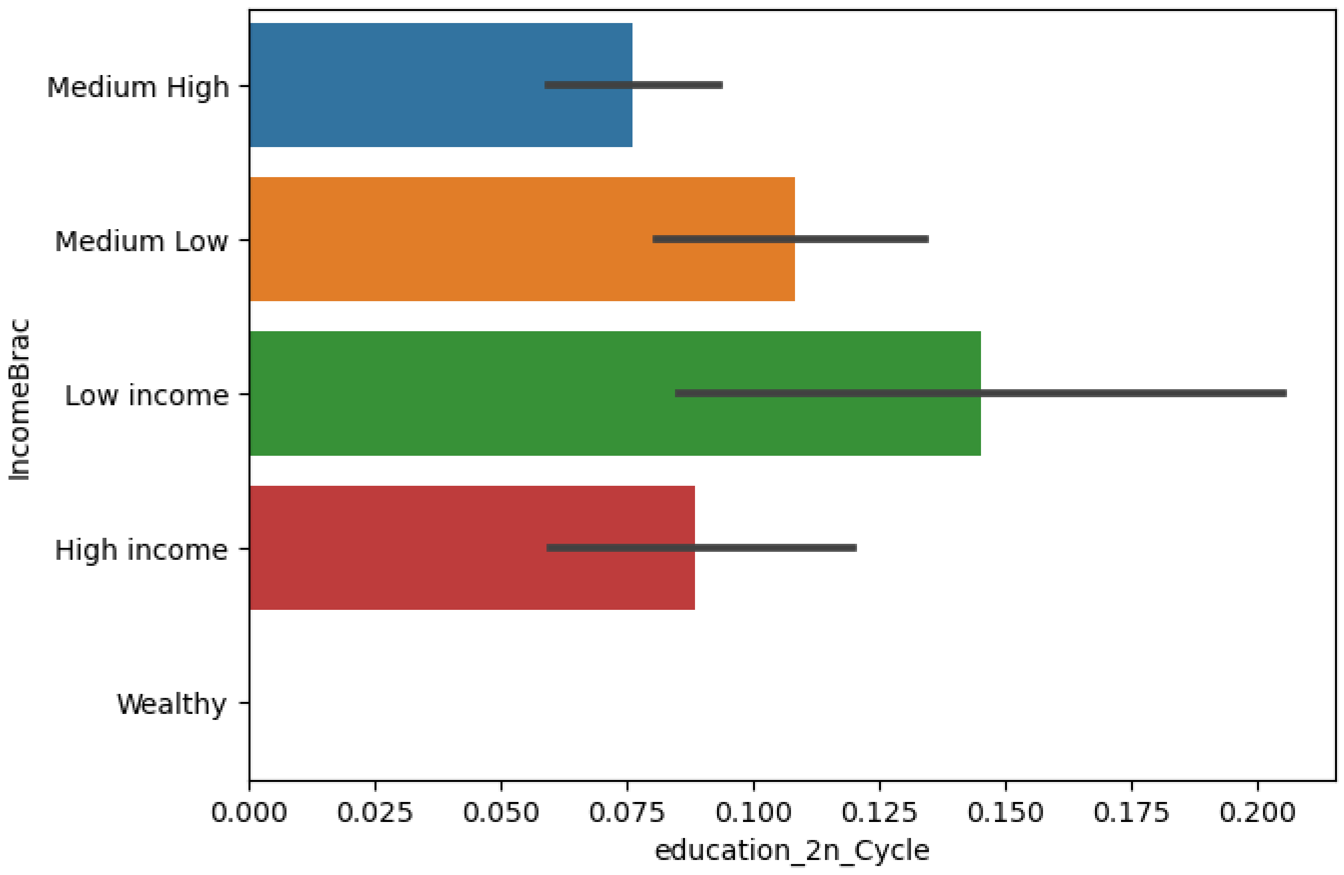


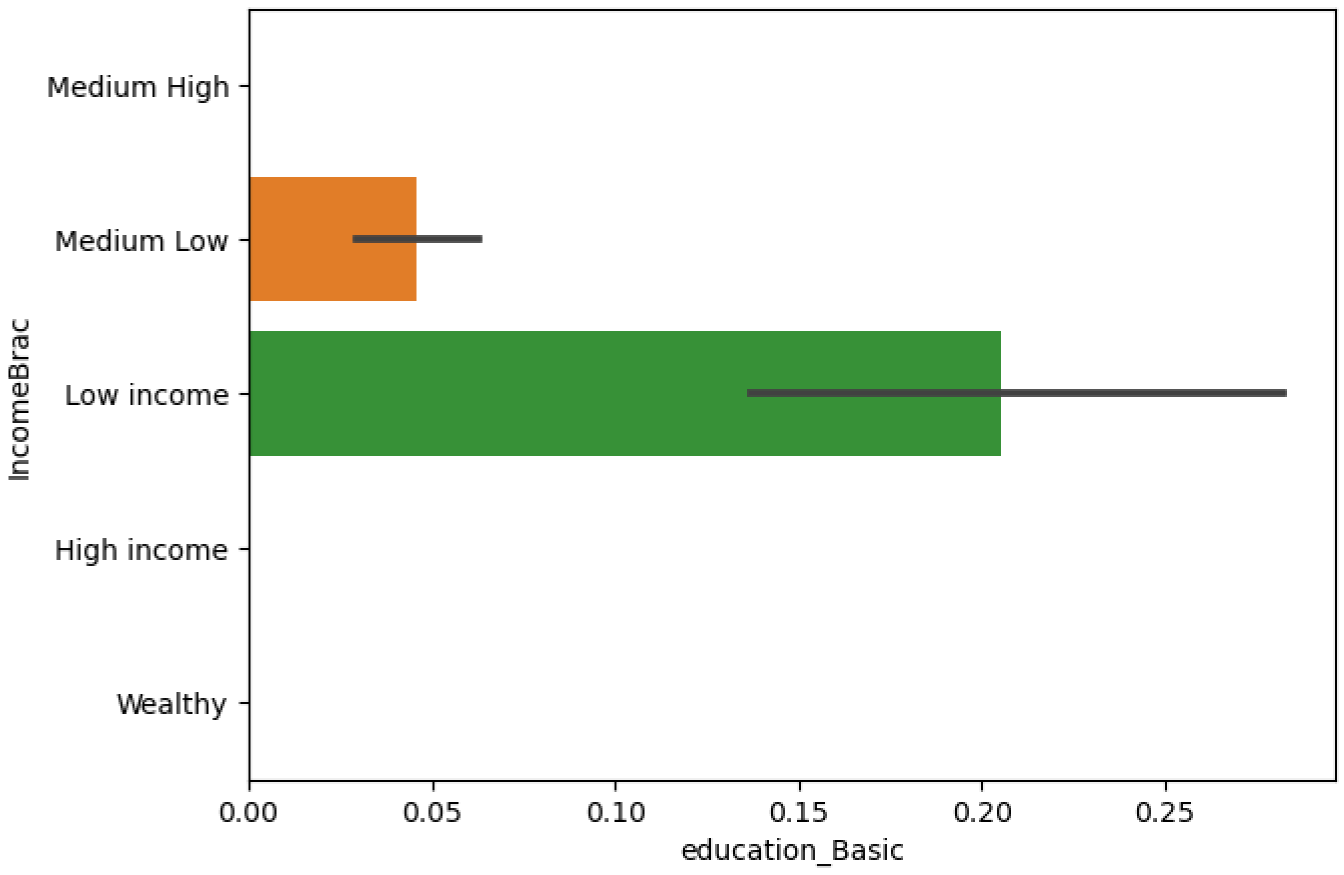
In [47]:
```
sns.countplot(data=df, x="highEd", hue="IncomeBrac")
```

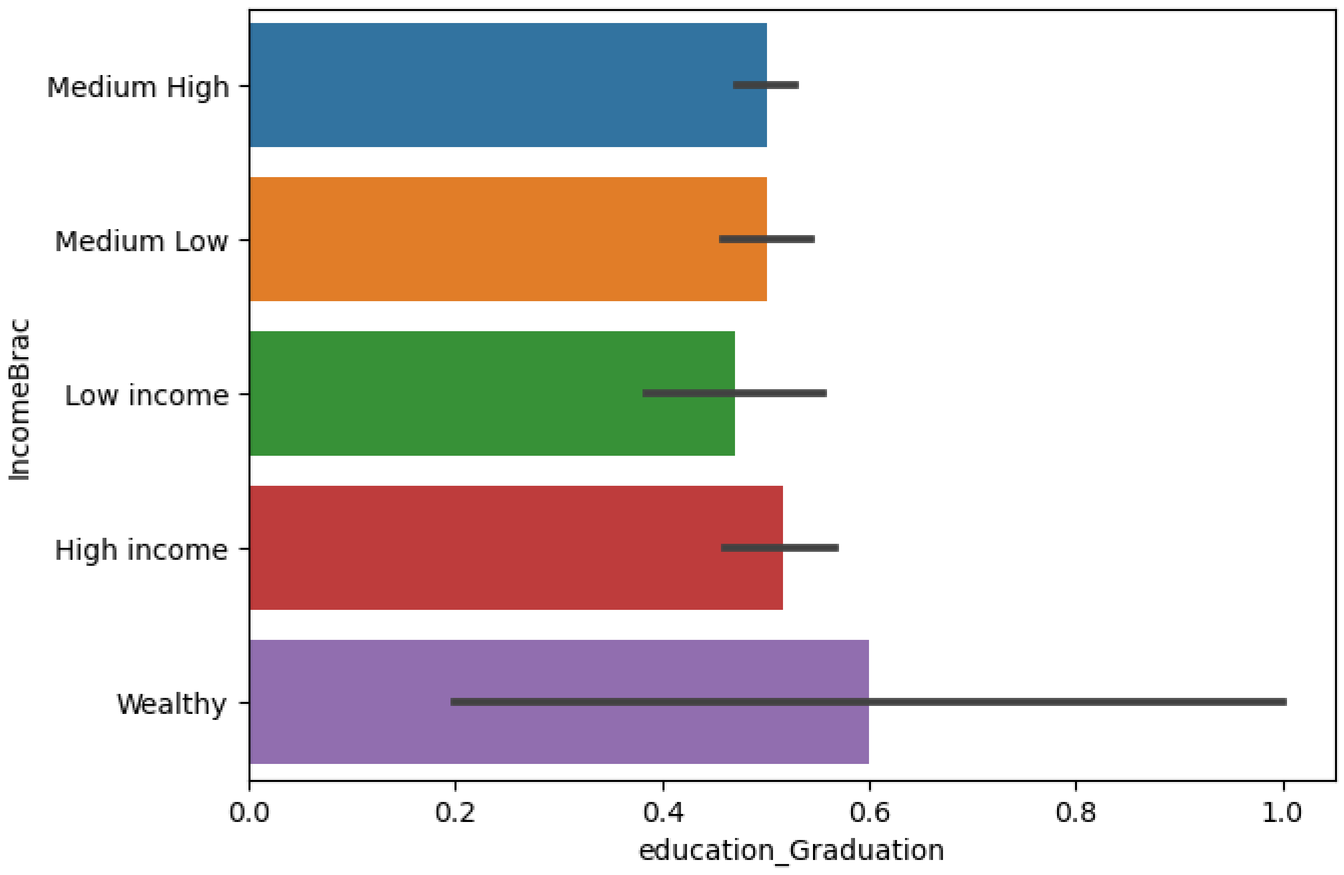
Out[47]:
```
<AxesSubplot:xlabel='highEd', ylabel='count'>
```

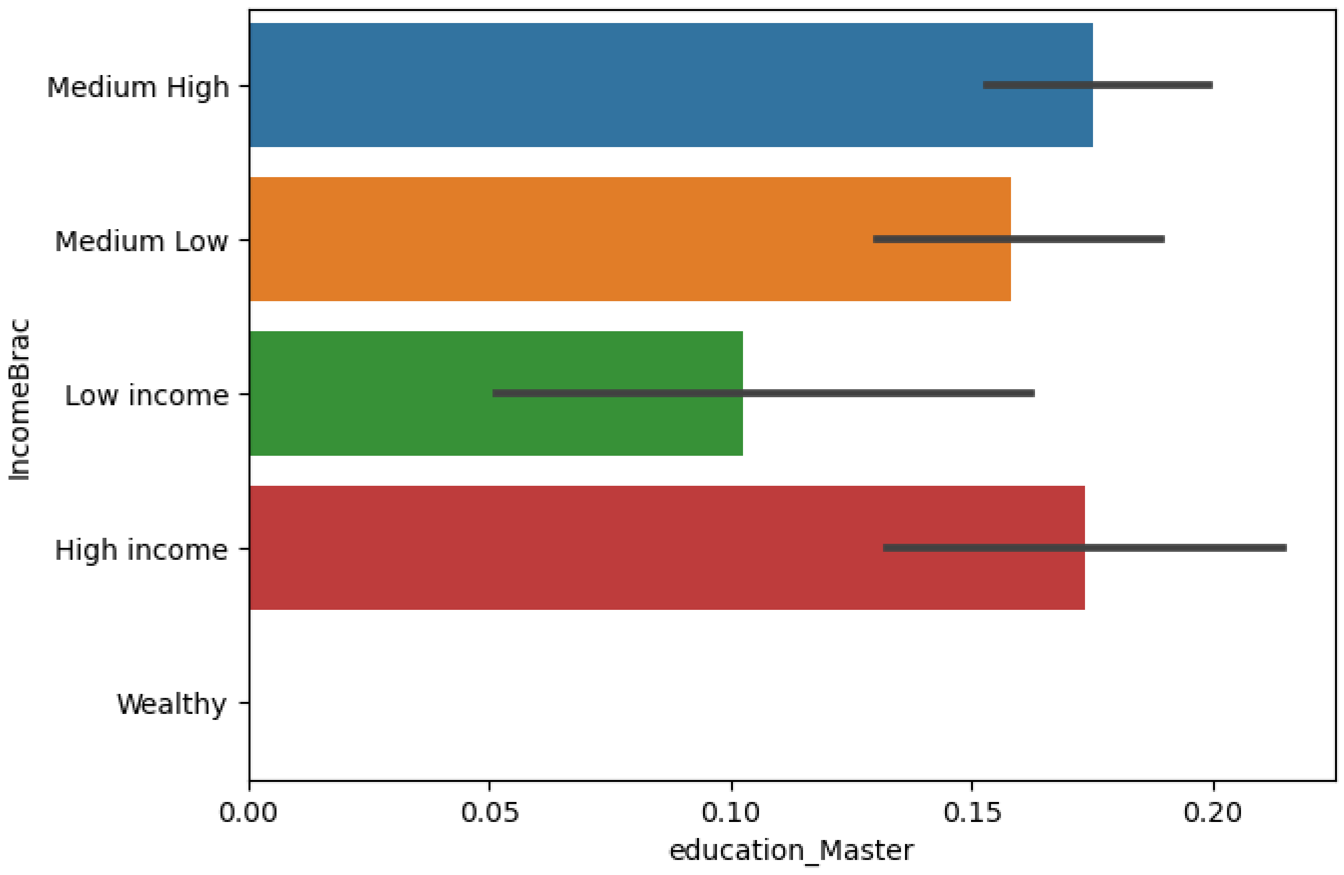


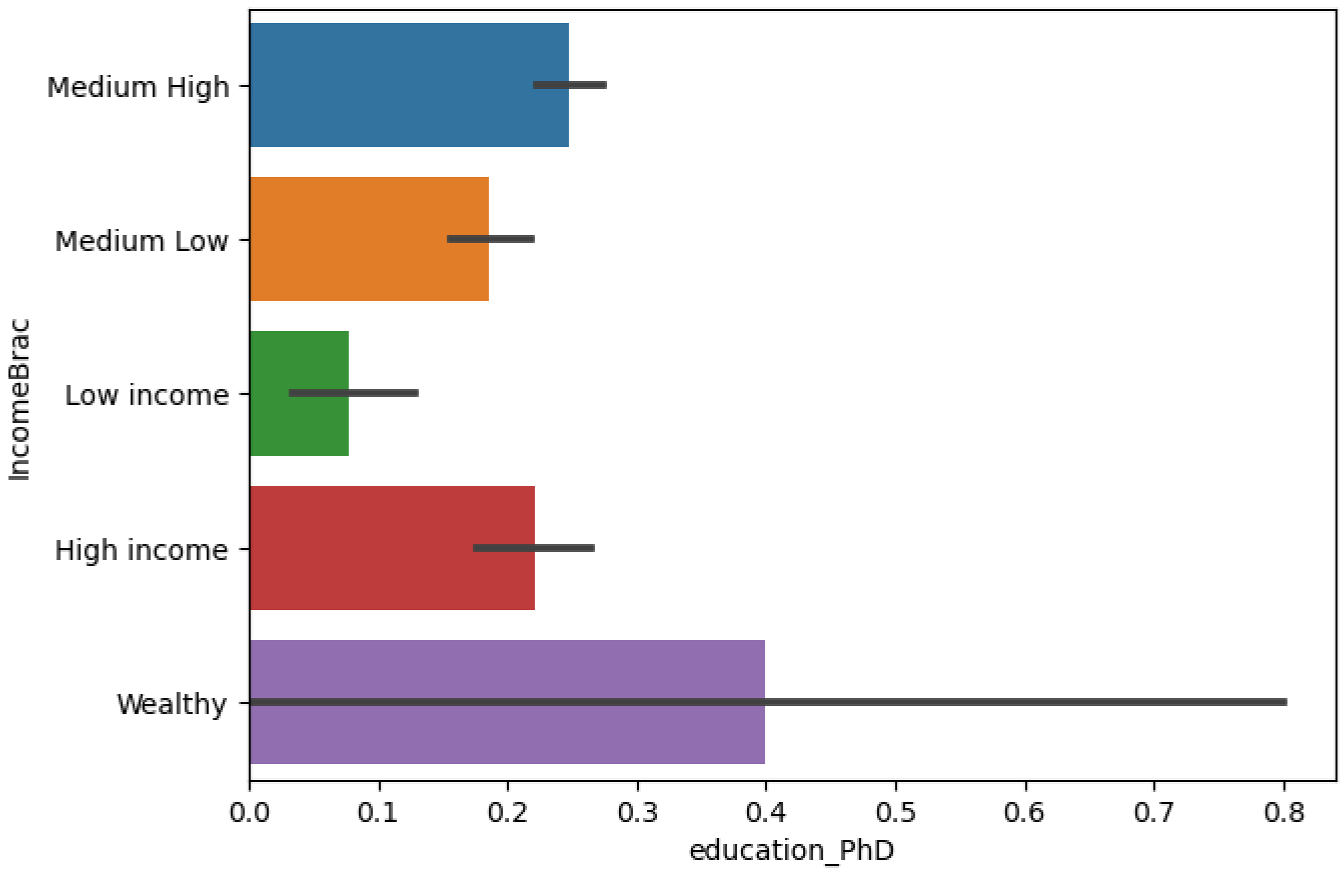
In [48]:
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


eduList = ['education_2n_Cycle', 'education_Basic', 'education_Graduation',
       'education_Master', 'education_PhD']

for edu in eduList:
    plotter(edu, 'IncomeBrac', df)
```


Medium High
Medium Low
Low income
High income
Wealthy
IncomeBrac
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
education_2n_Cycle

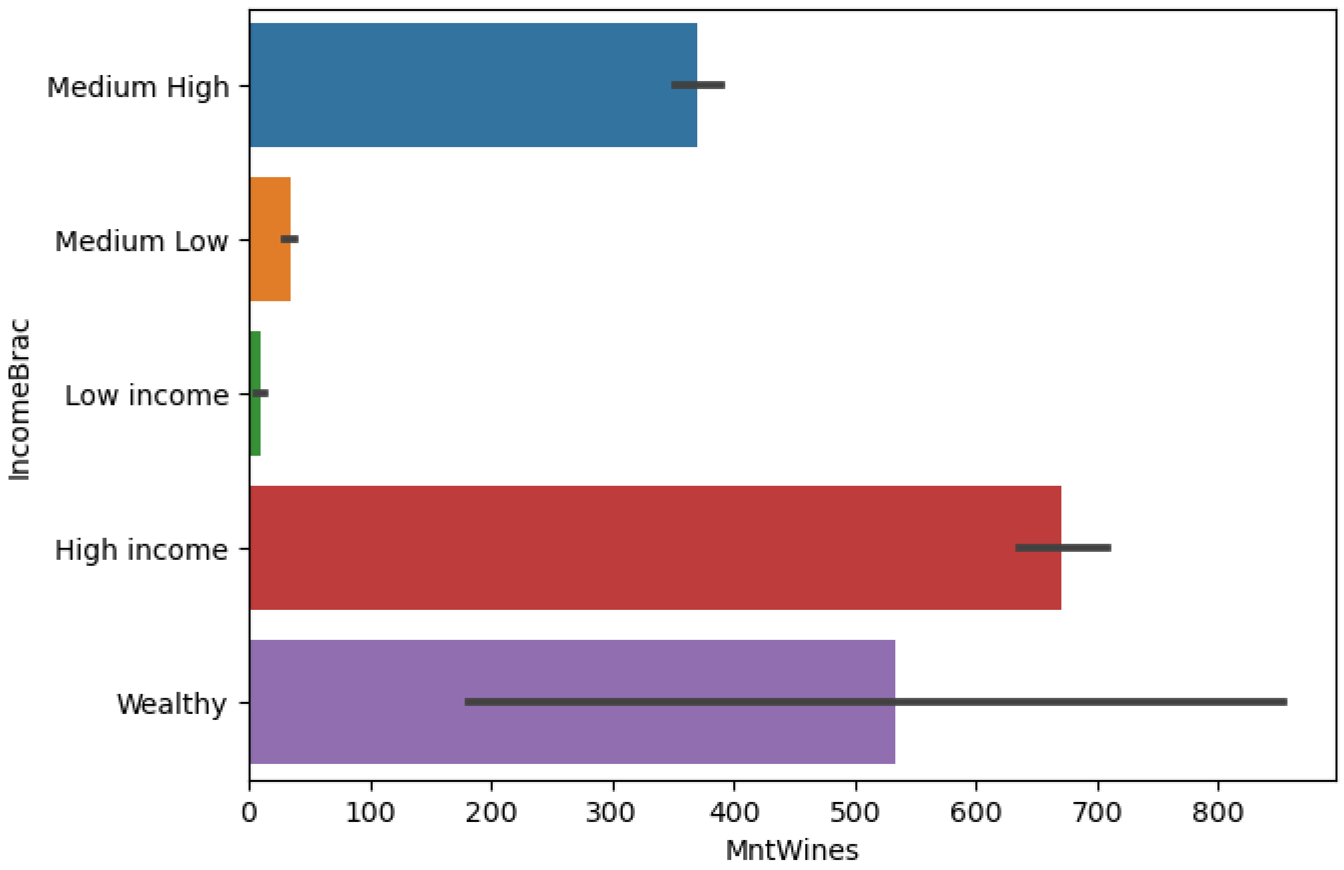

Medium High
Medium Low
Low income
High income
Wealthy
IncomeBrac
0.00
0.05
0.10
0.15
0.20
0.25
education_Basic

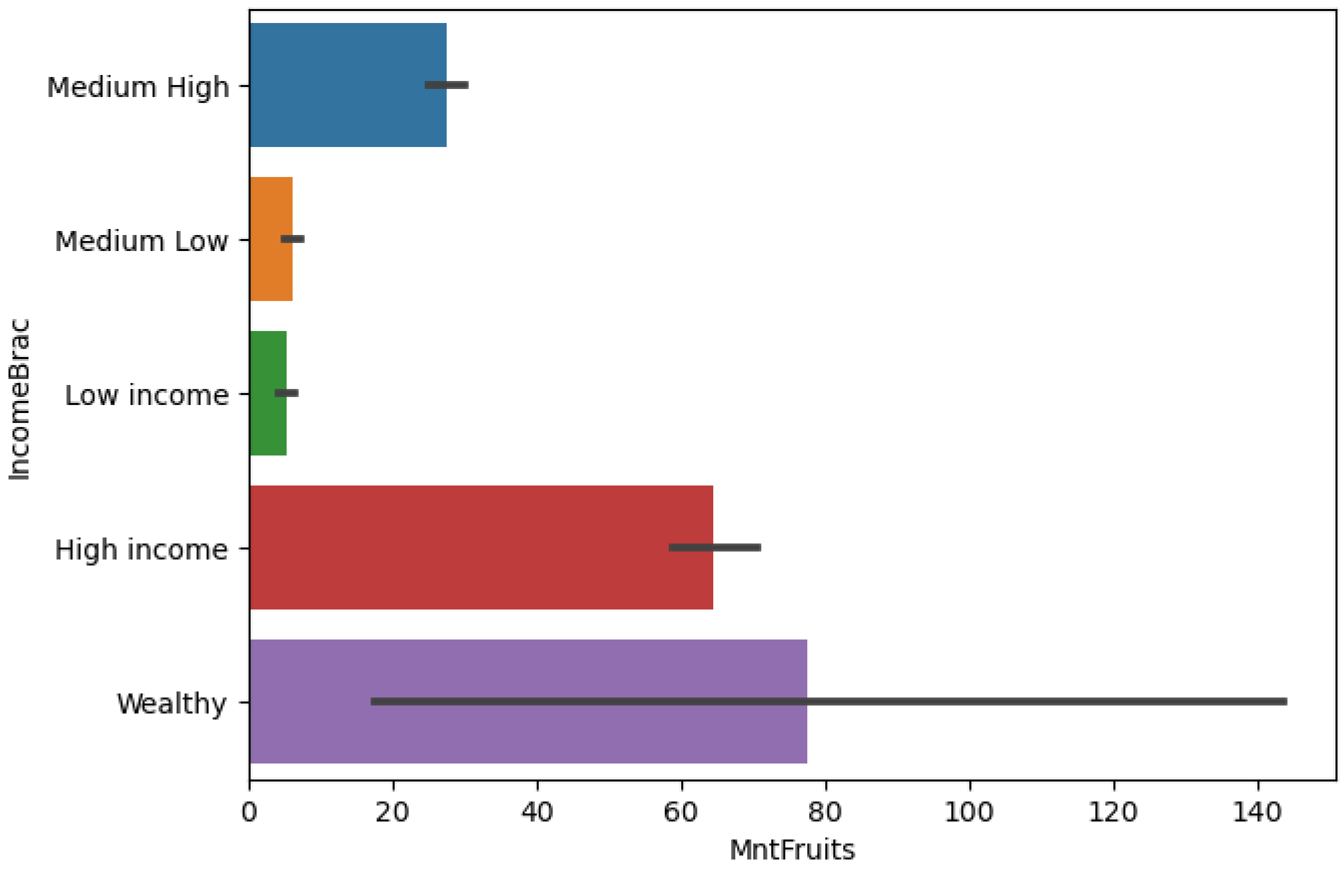

Medium High
Medium Low
Low income
High income
Wealthy
IncomeBrac
0.0
0.2
0.4
0.6
0.8
1.0
education_Graduation

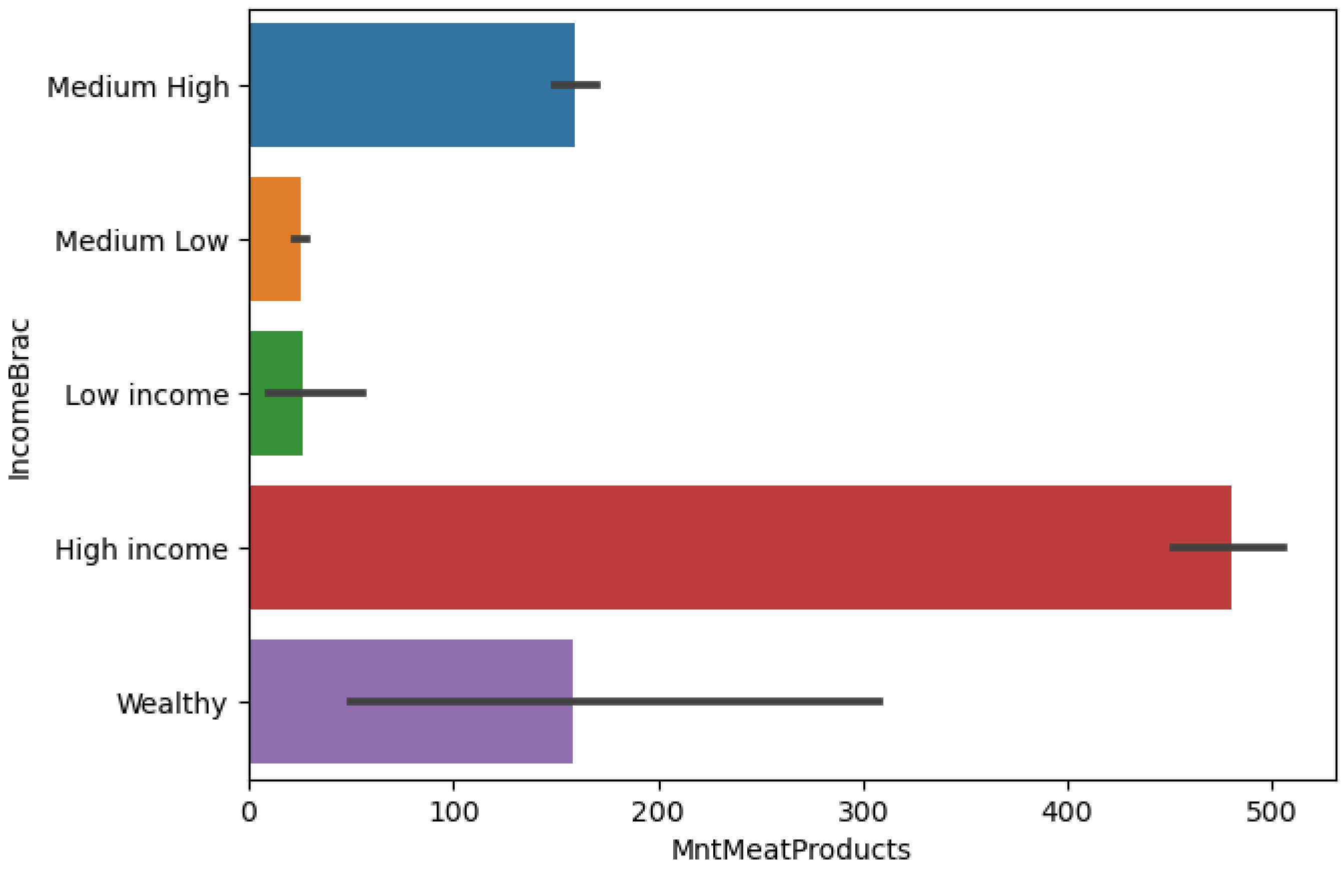

Medium High
Medium Low
Low income
High income
Wealthy
IncomeBrac
0.00
0.05
0.10
0.15
0.20
education_Master

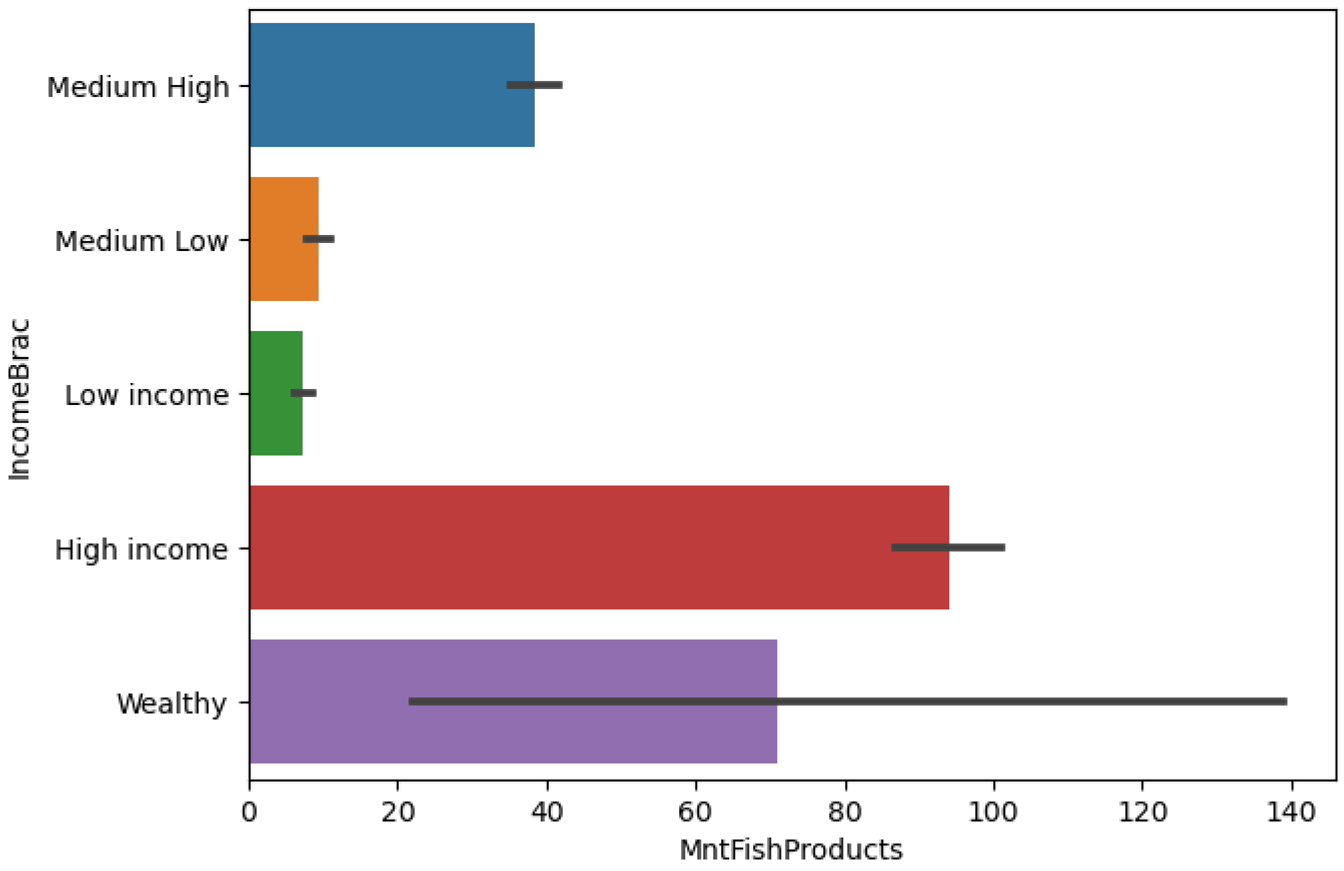


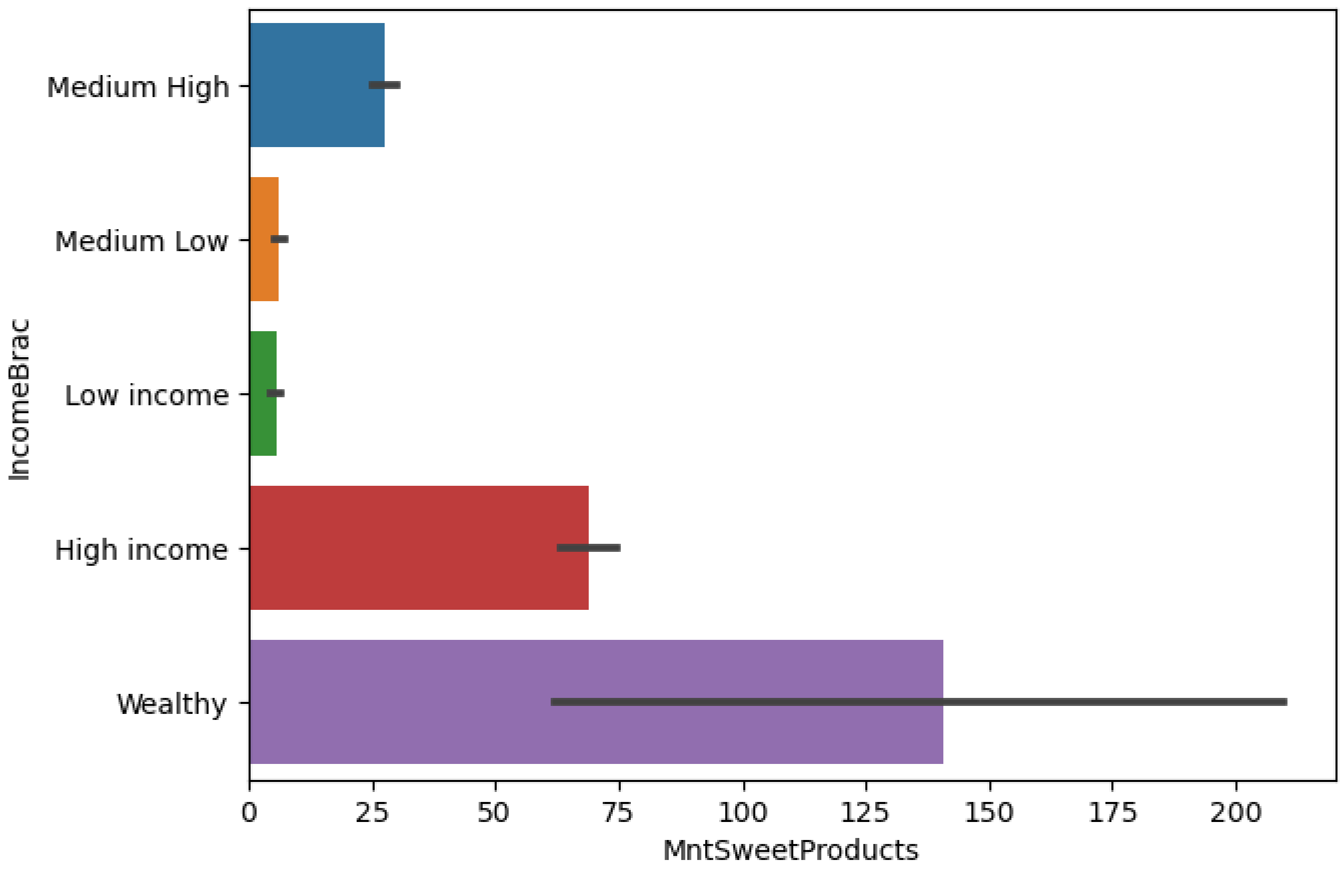
In [49]:
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


Products = ['MntWines', 'MntFruits','MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
 'MntGoldProds']

for Prod in Products:
    plotter(Prod, 'IncomeBrac', df)
```

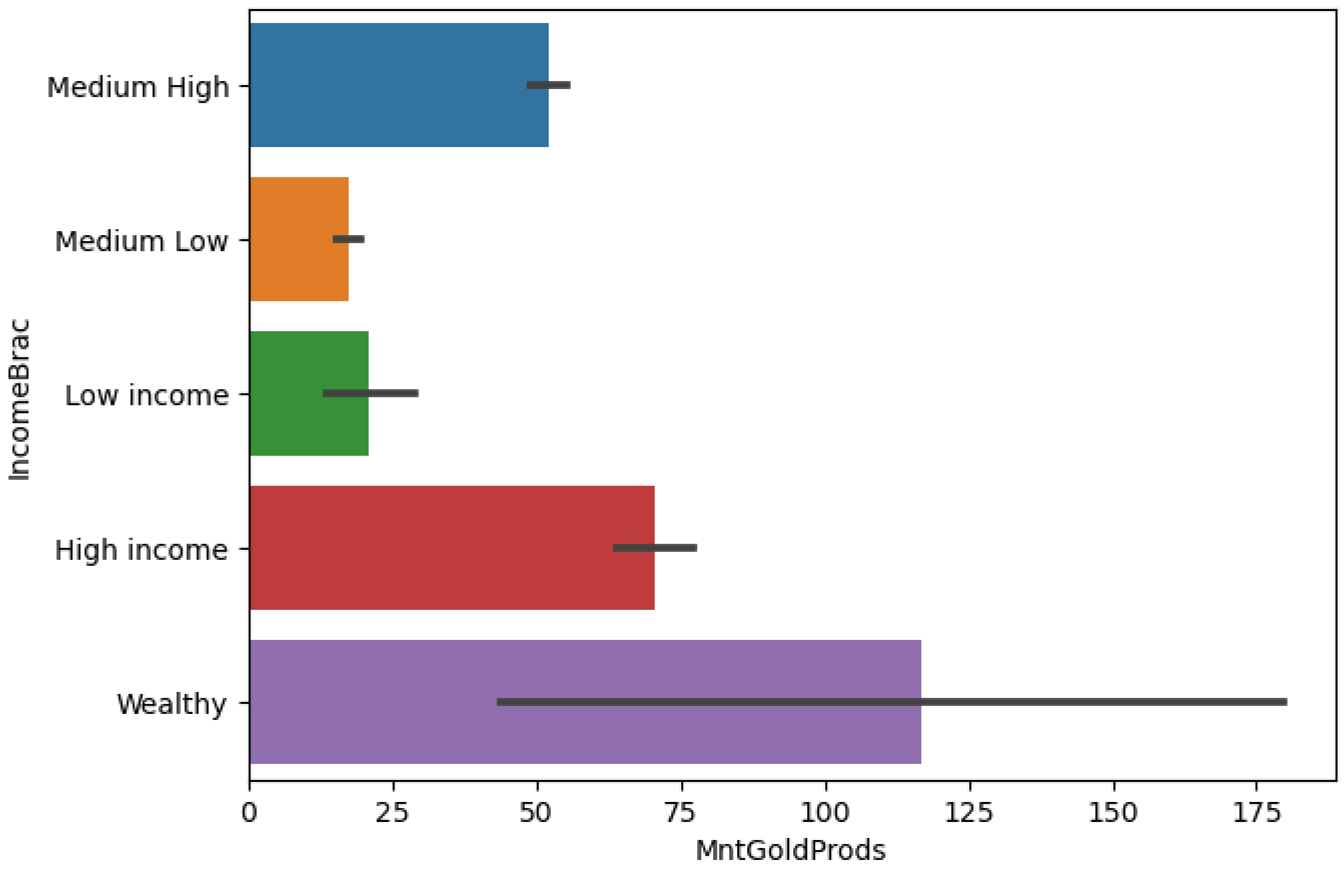






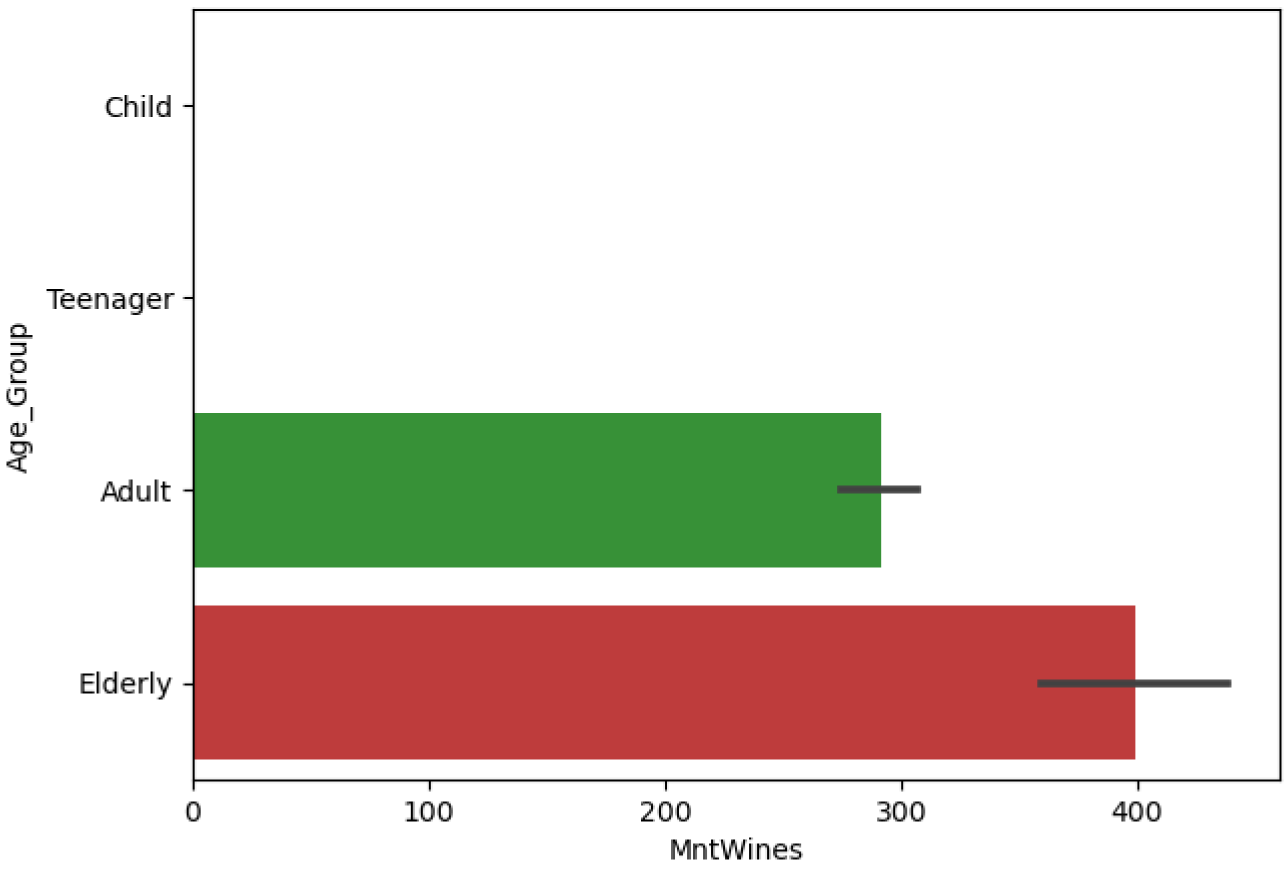


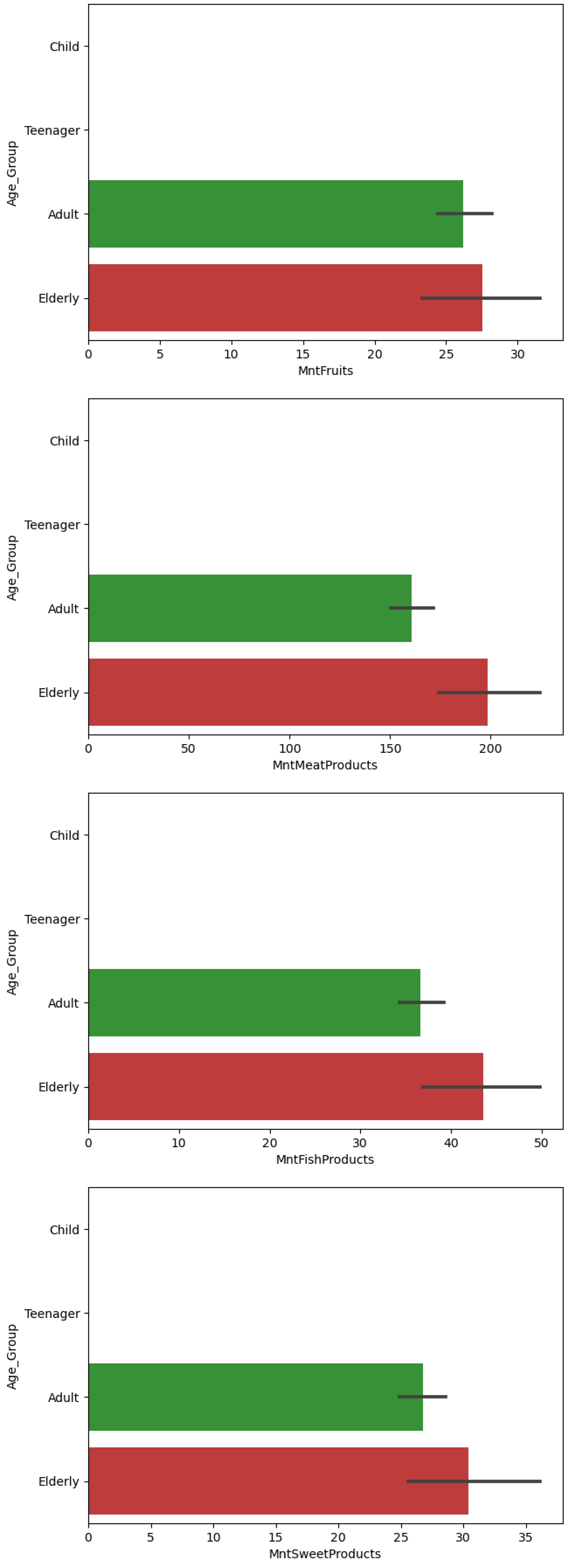


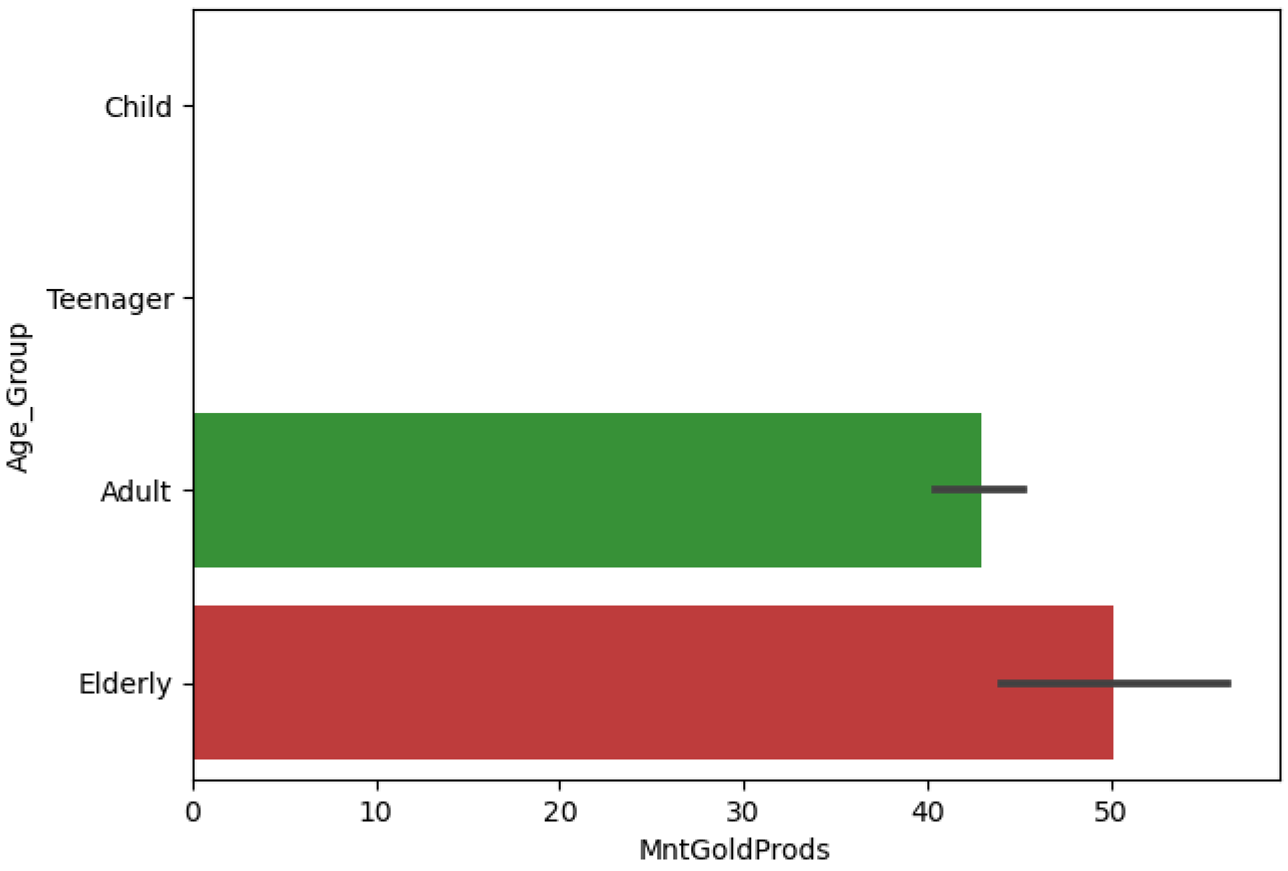


In [50]:

```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


Products = ['MntWines', 'MntFruits','MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
 'MntGoldProds']

for Prod in Products:
    plotter(Prod, 'Age_Group', df)
```

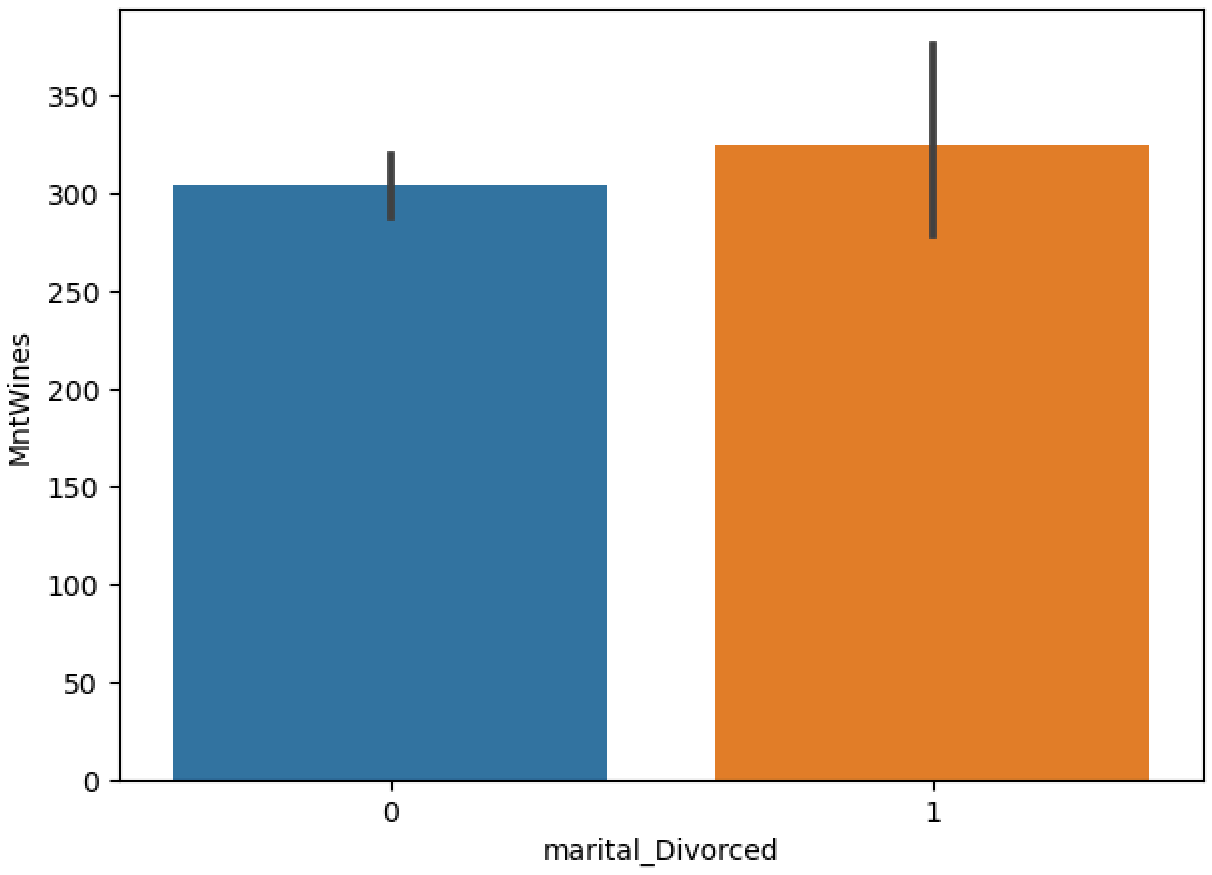


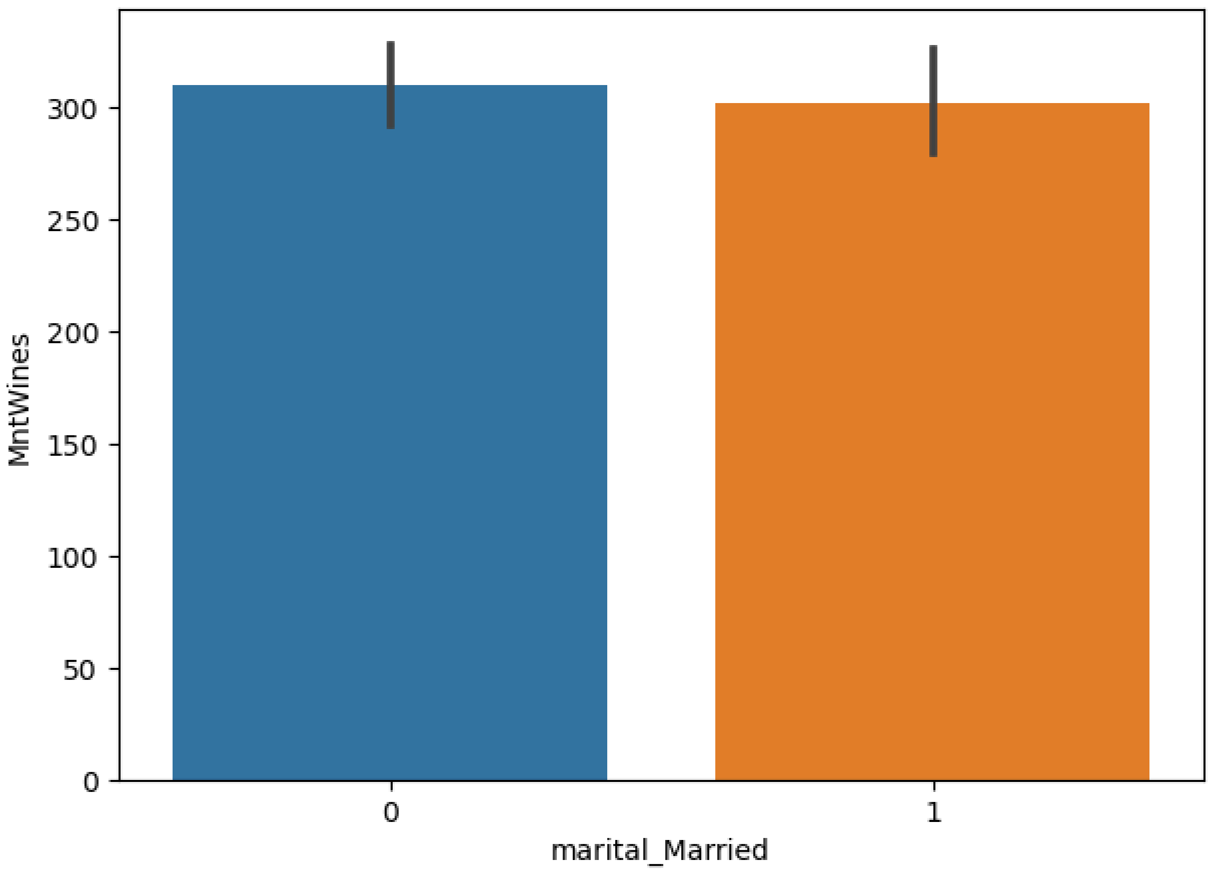

Age_Group
Child
Teenager
Adult
Elderly
0
5
10
15
20
25
30
MntFruits
Age_Group
Child
Teenager
Adult
Elderly
0
50
100
150
200
MntMeatProducts
Age_Group
Child
Teenager
Adult
Elderly
0
10
20
30
40
50
MntFishProducts
Age_Group
Child
Teenager
Adult
Elderly
0
5
10
15
20
25
30
35
MntSweetProducts

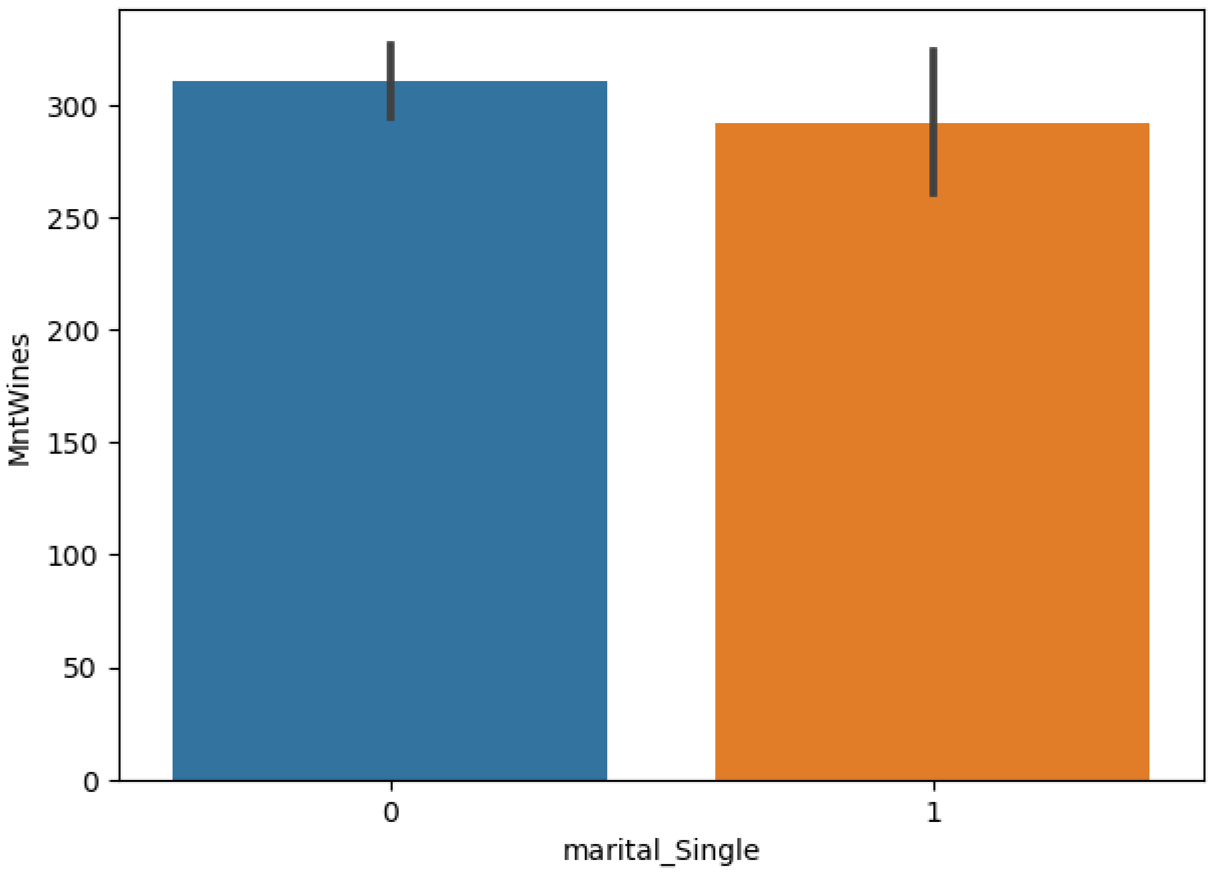


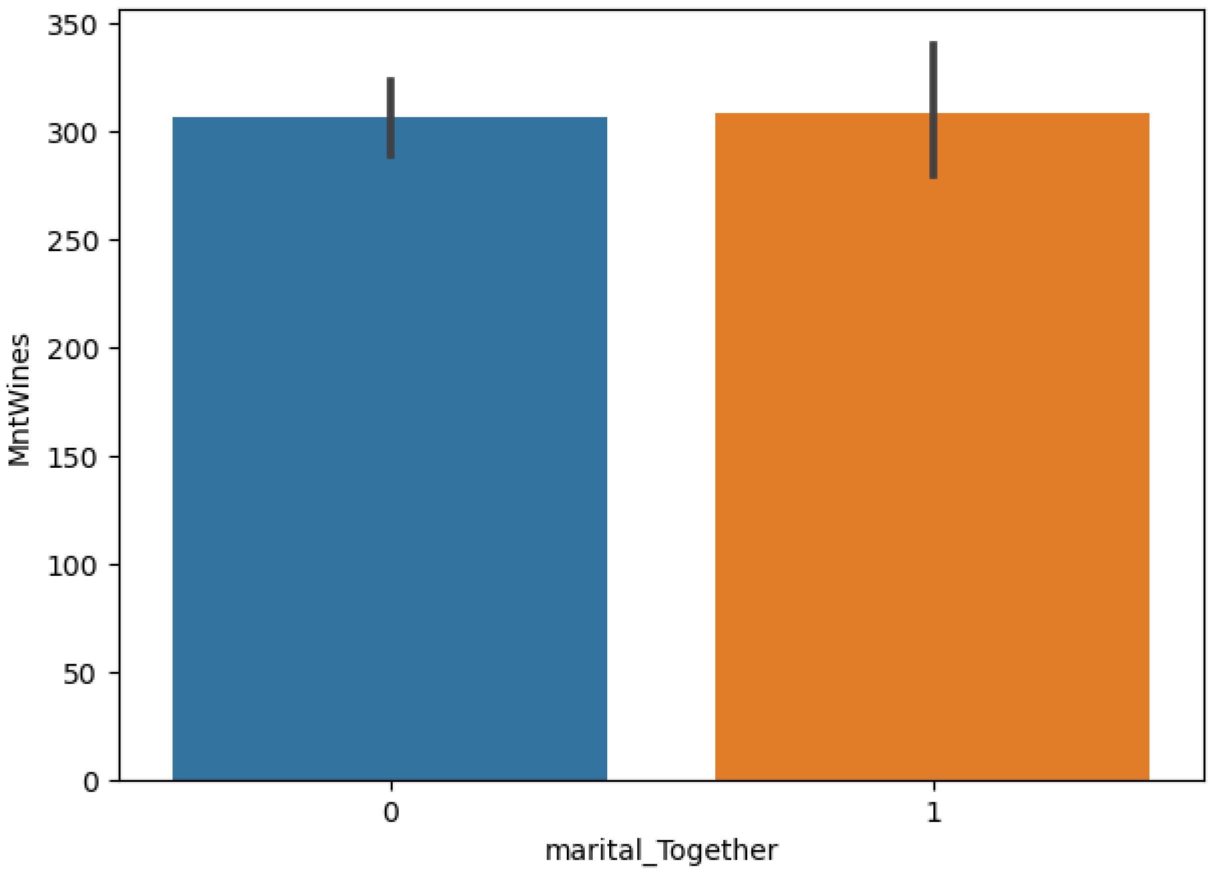
In [51]:

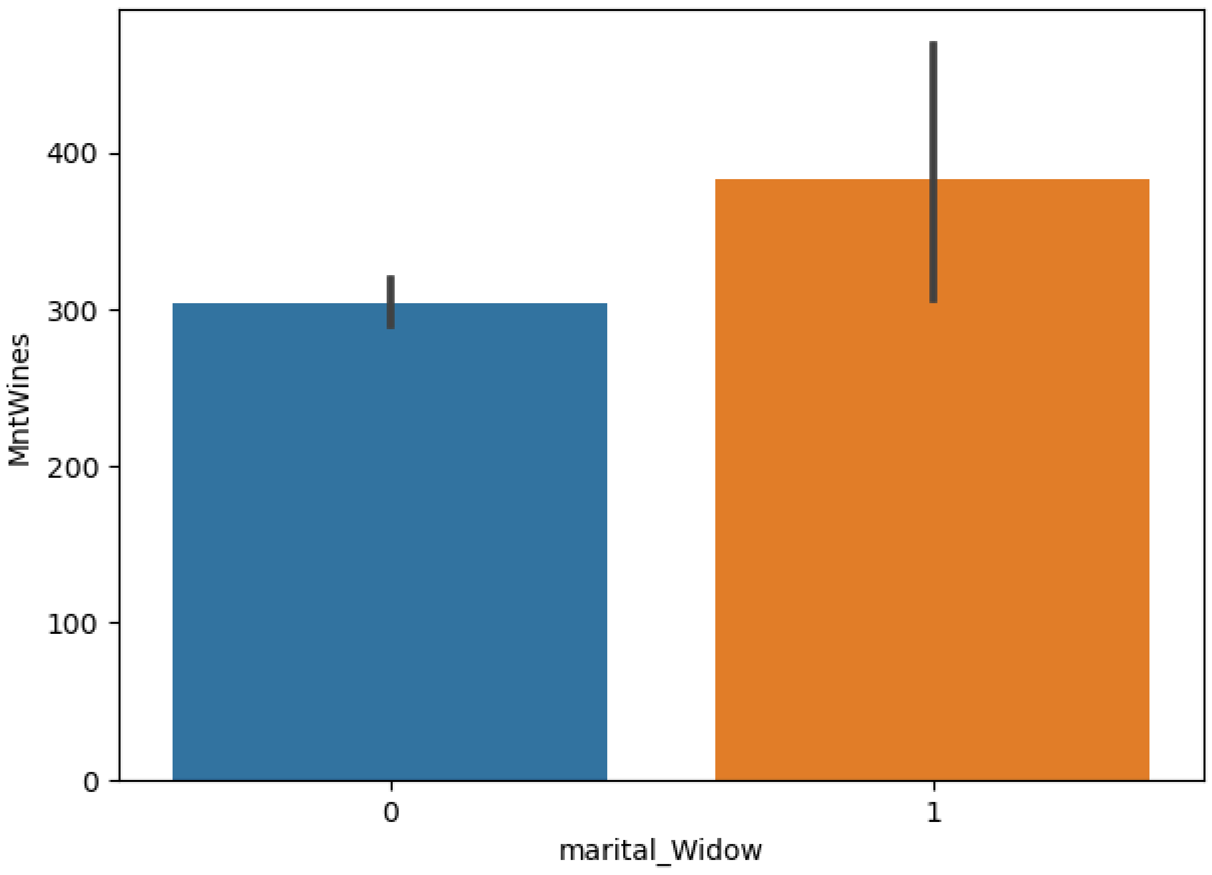
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'MntWines', df)
```





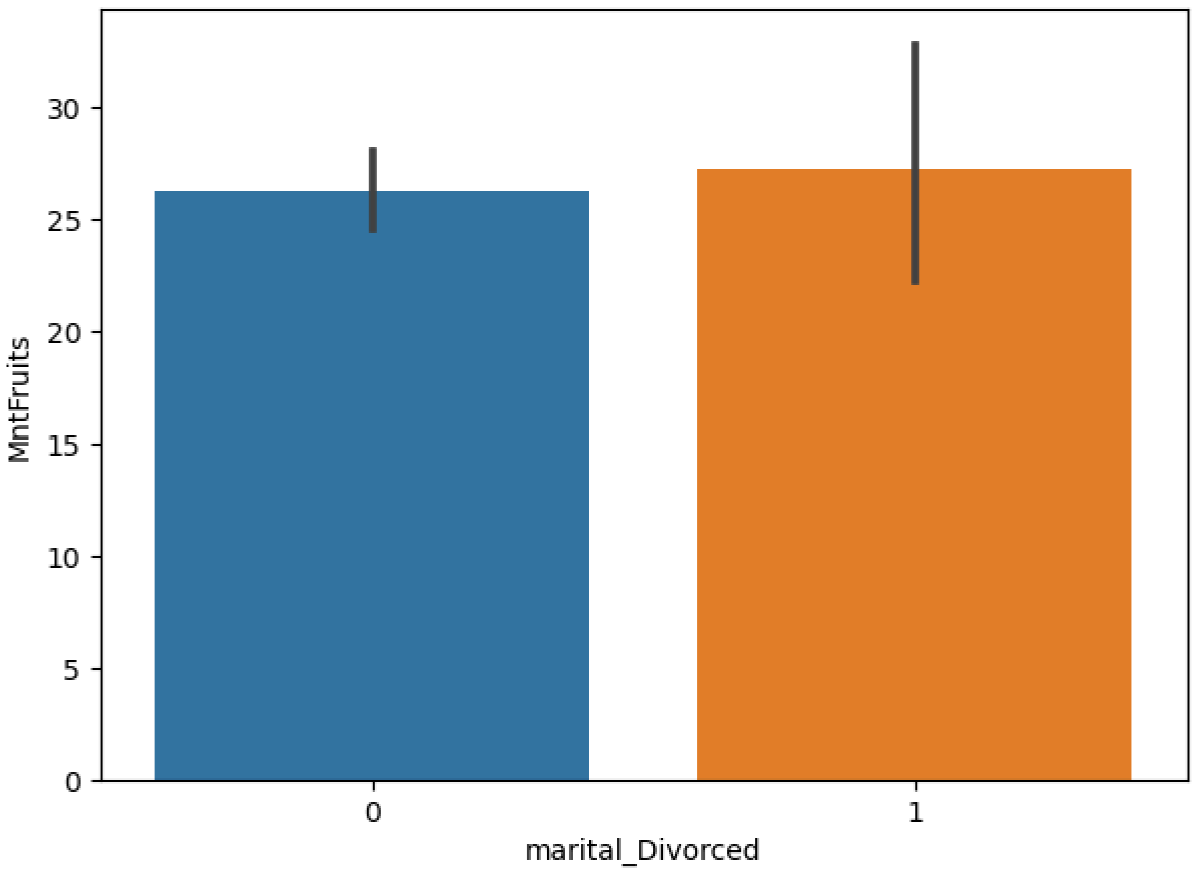


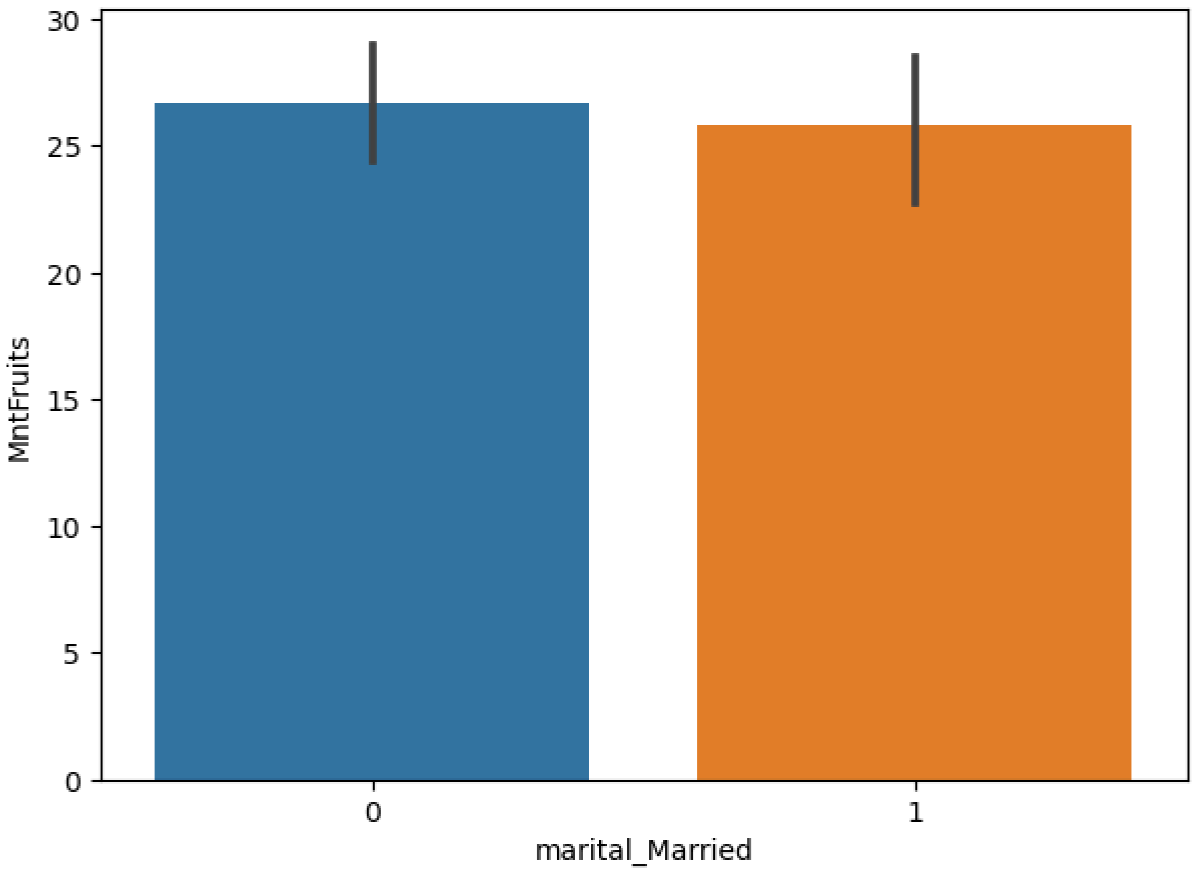


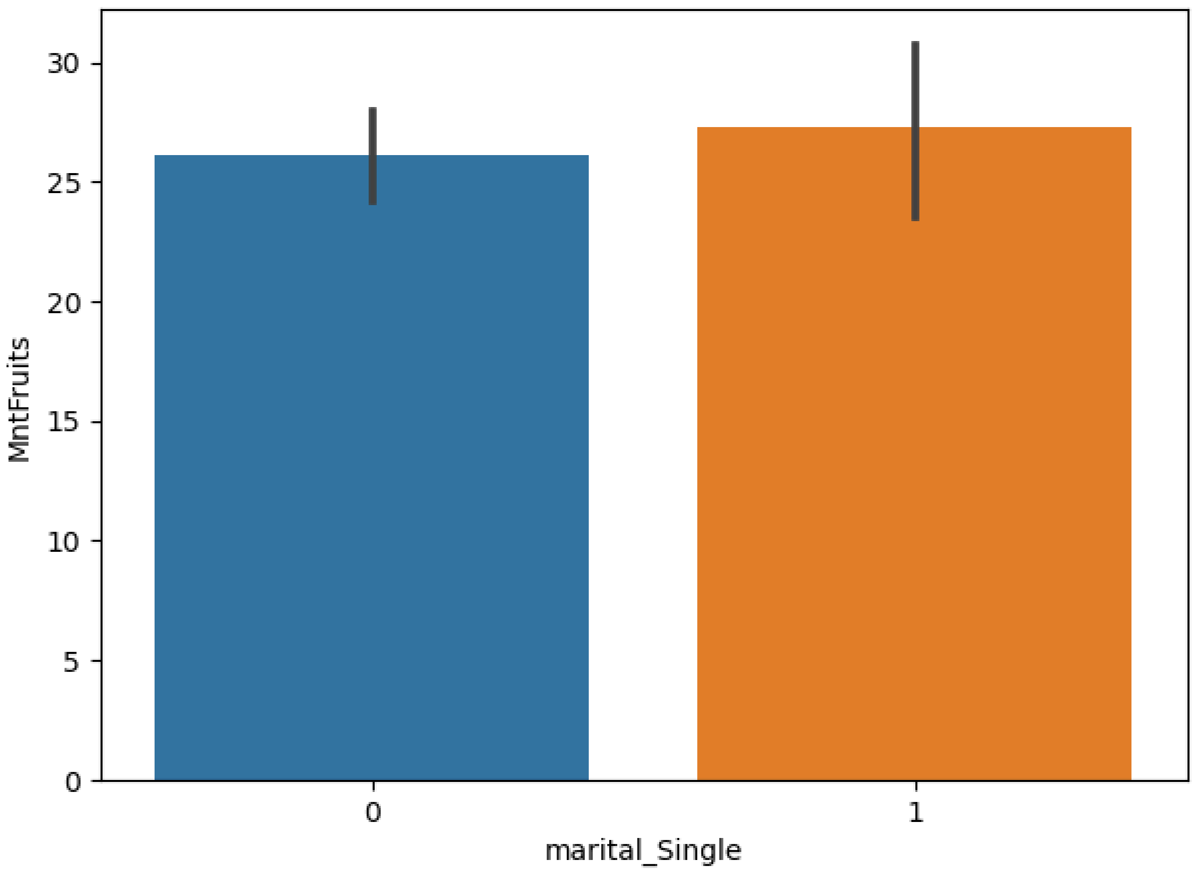


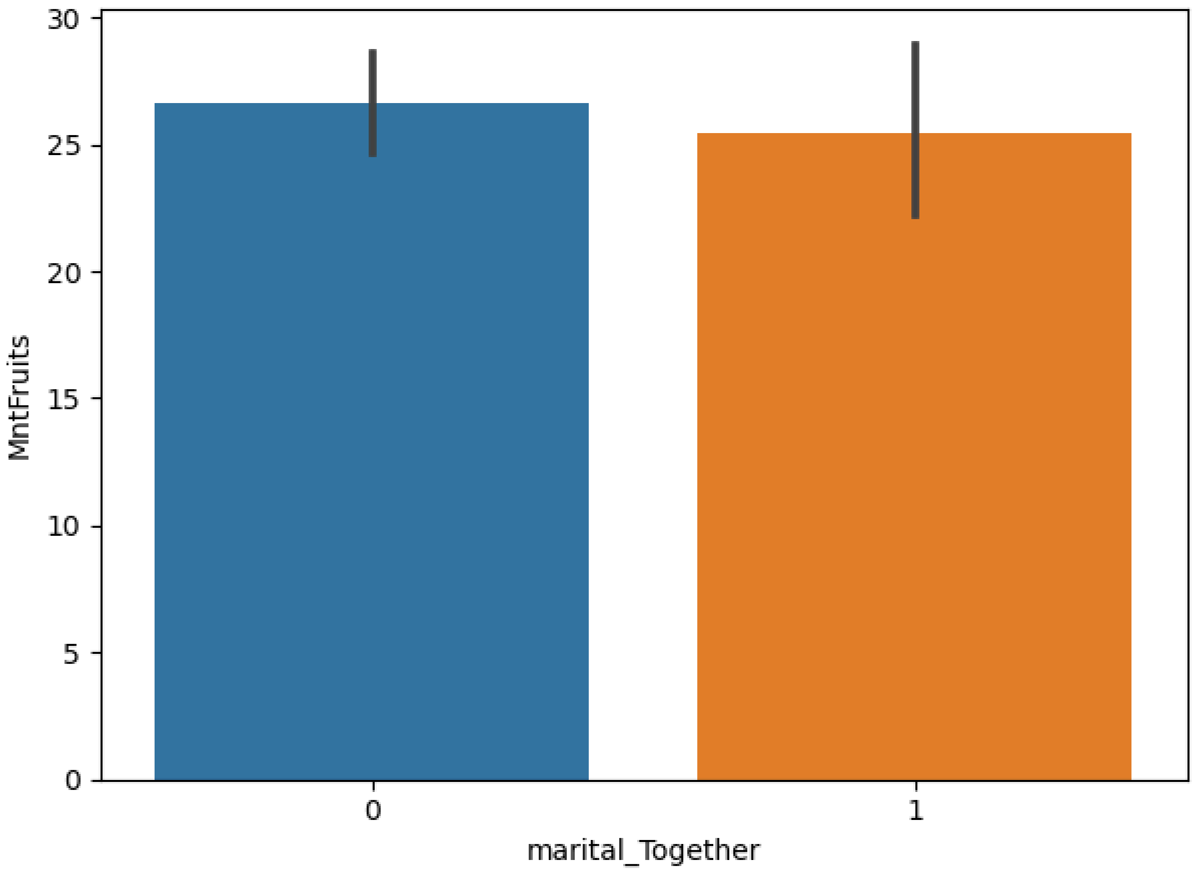
In [52]:

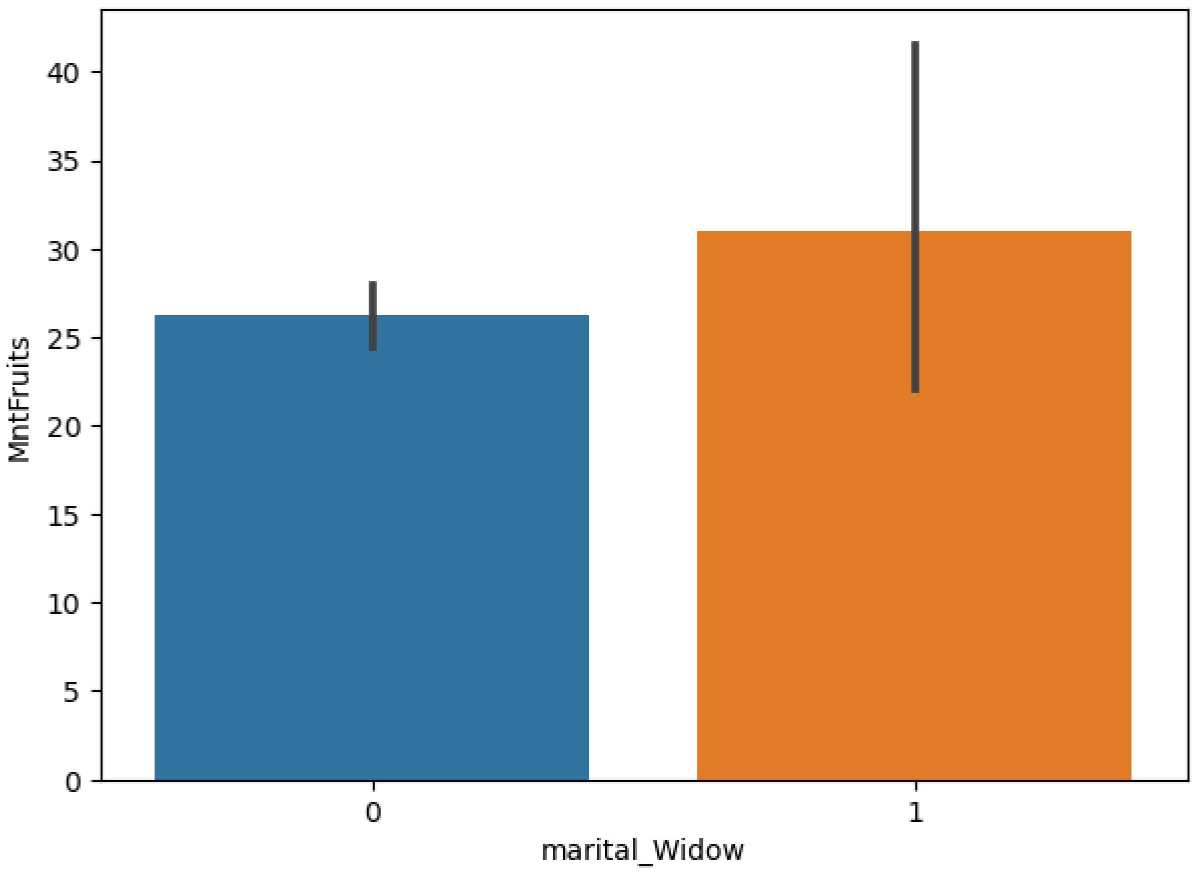
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'MntFruits', df)
```







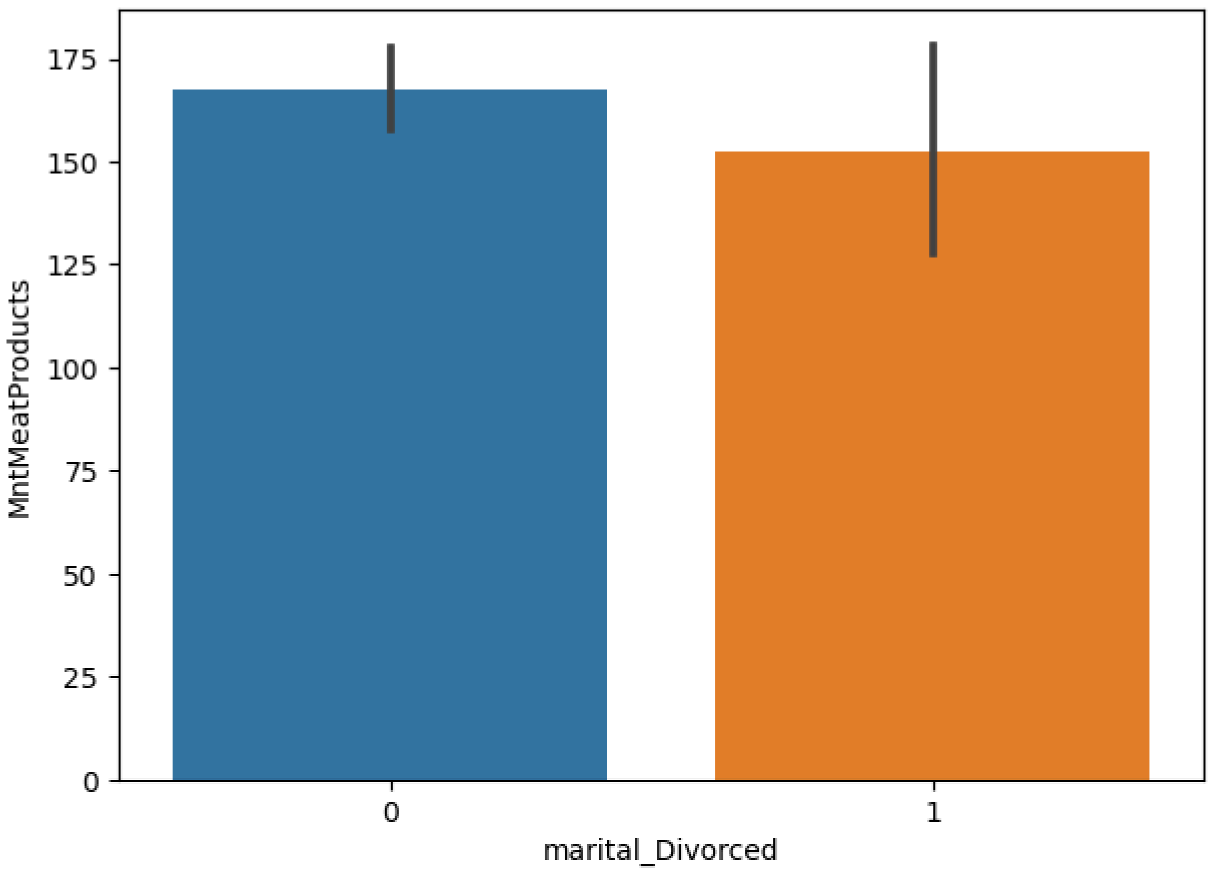


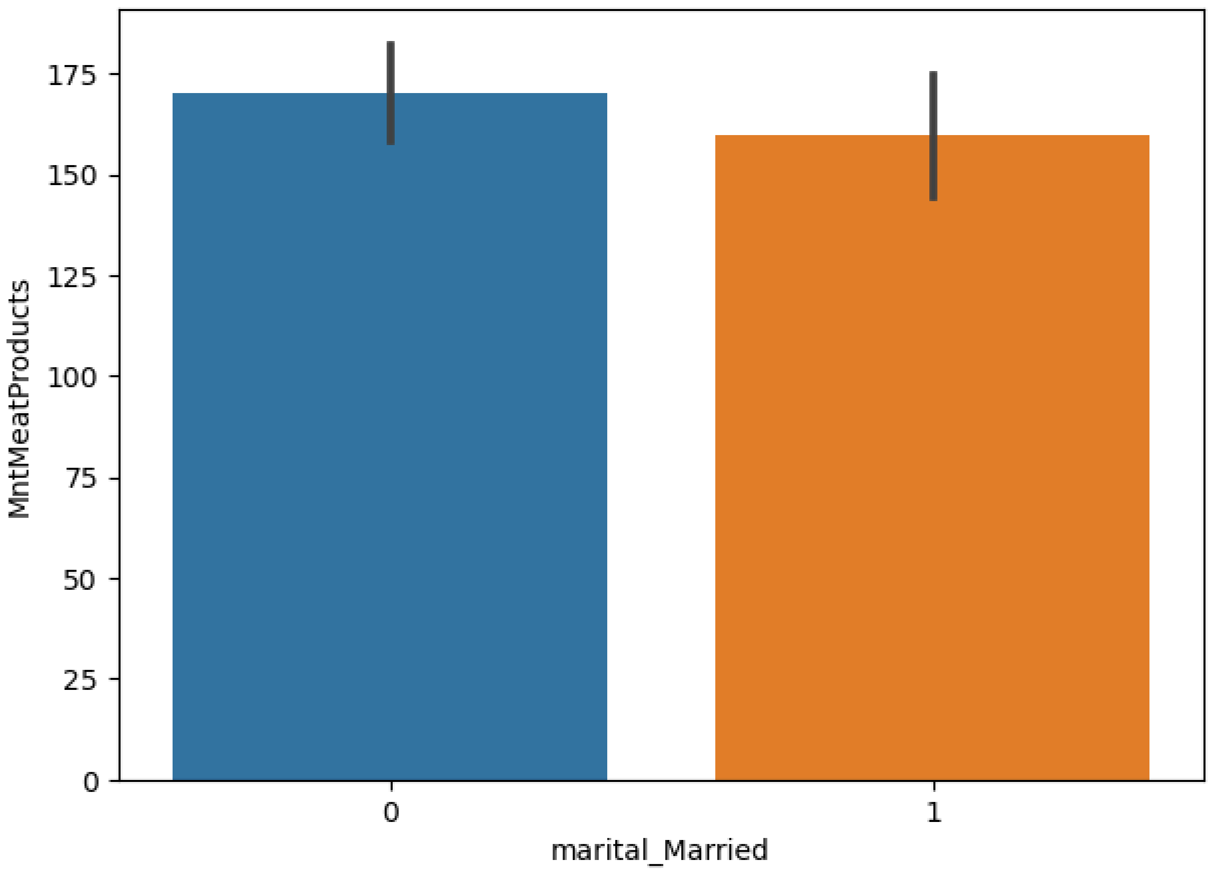


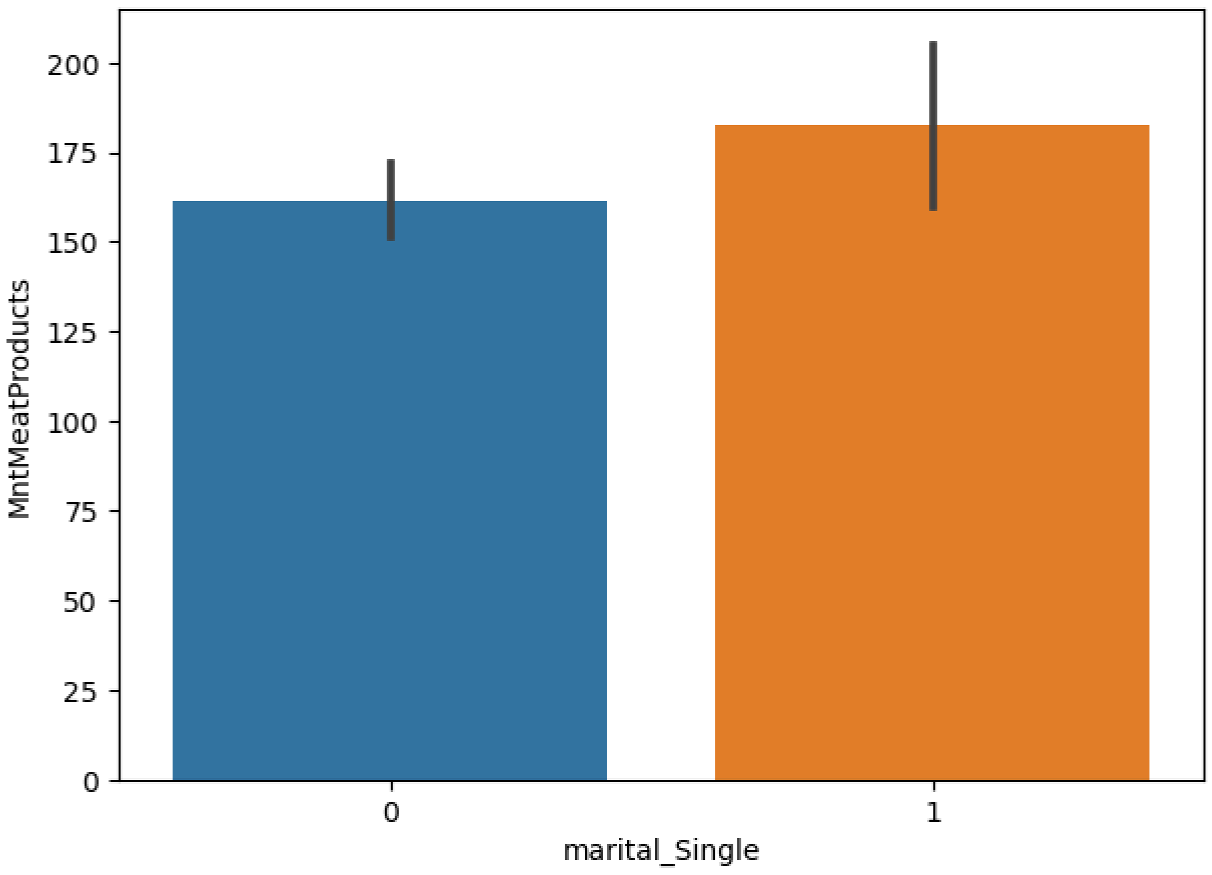
In [ ]:

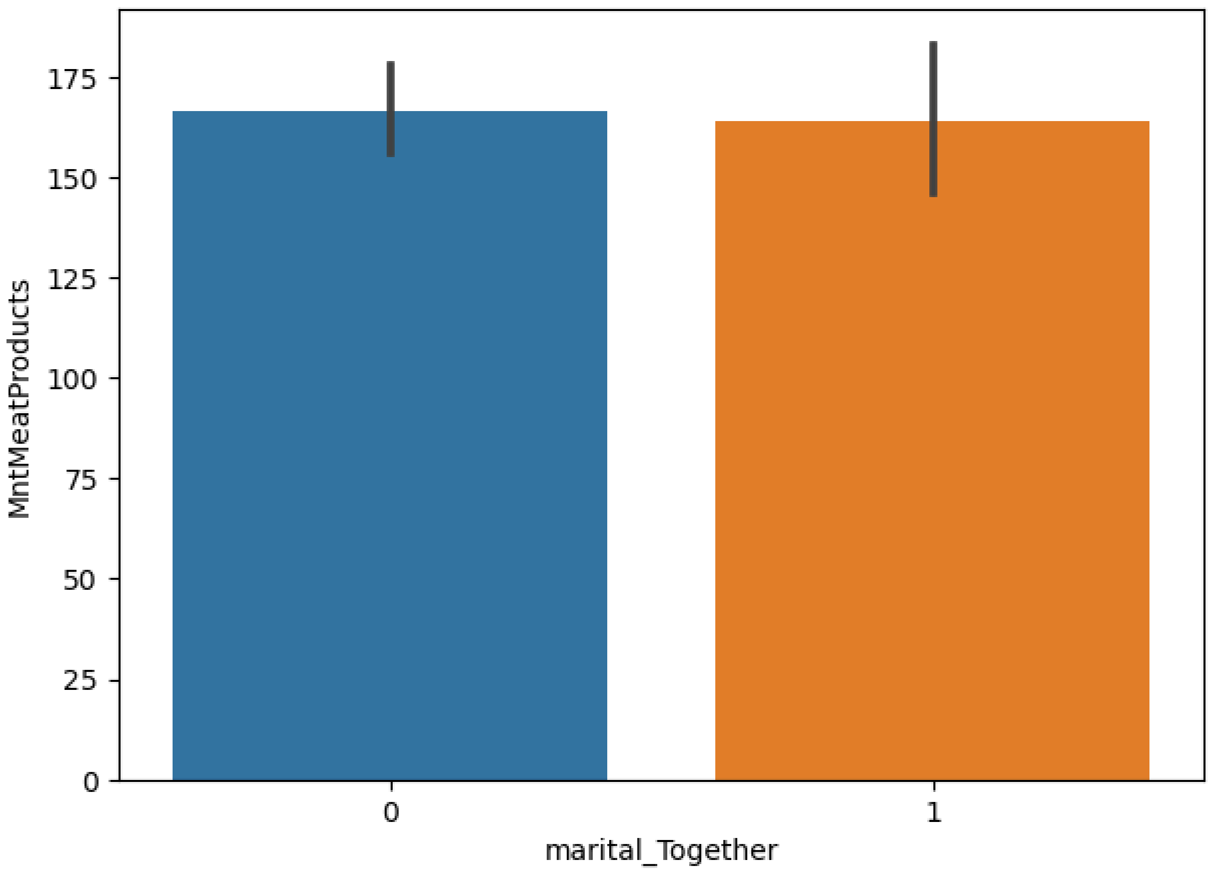
In [53]:

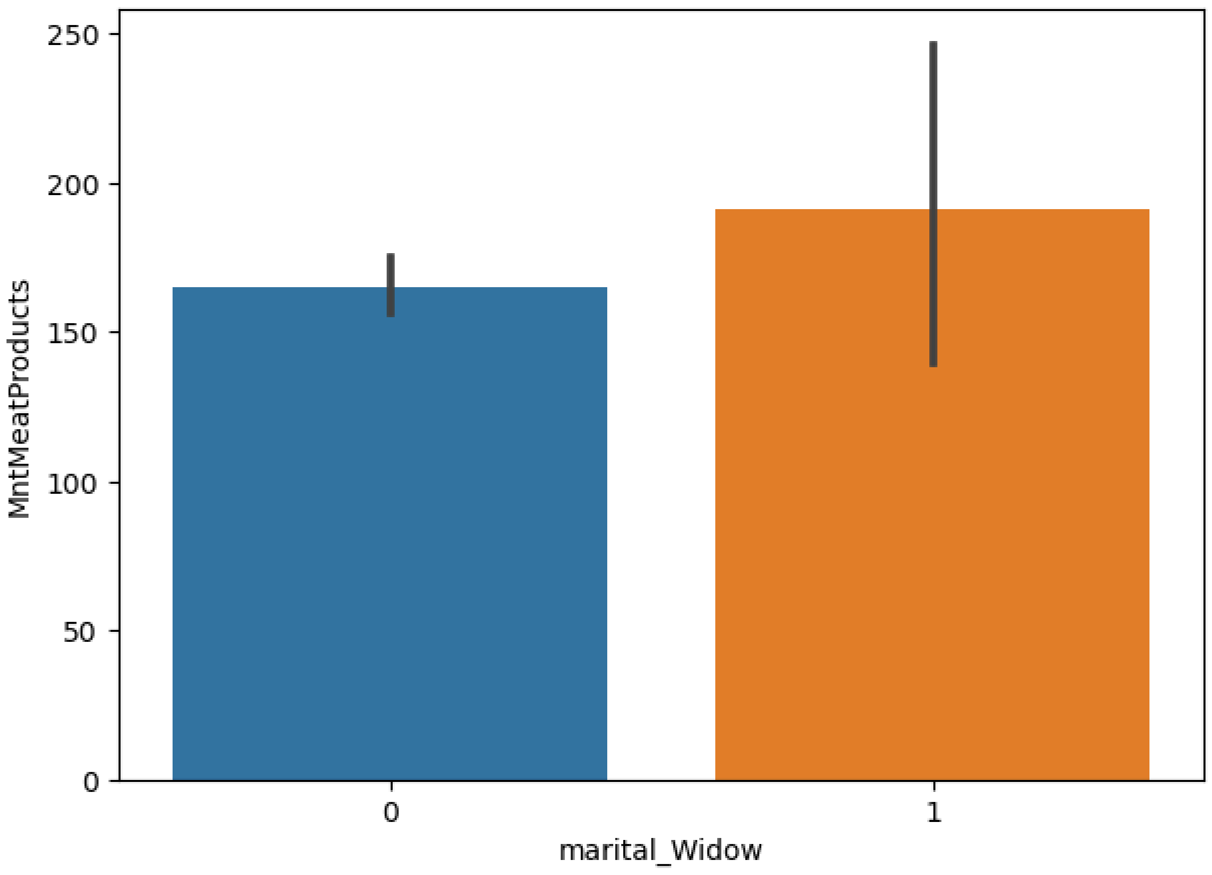
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'MntMeatProducts', df)
```





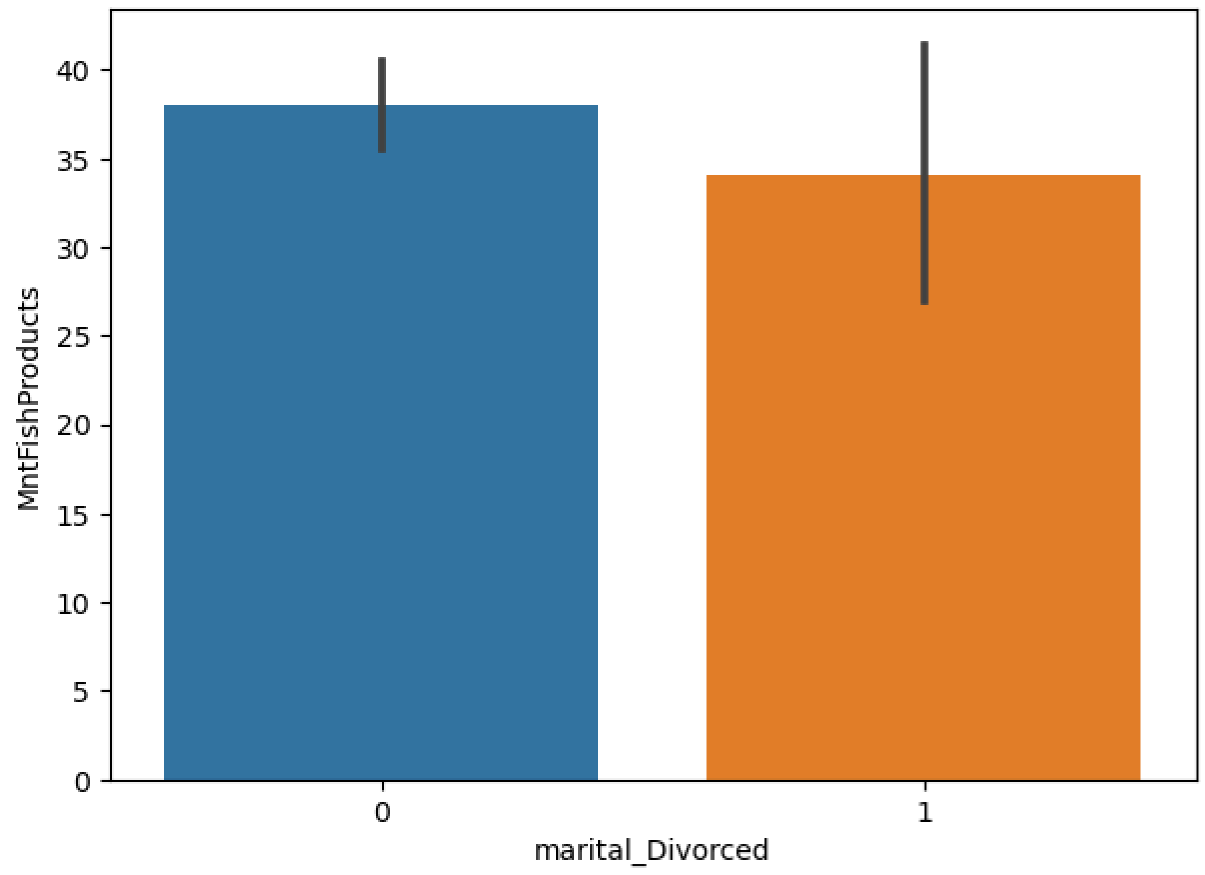




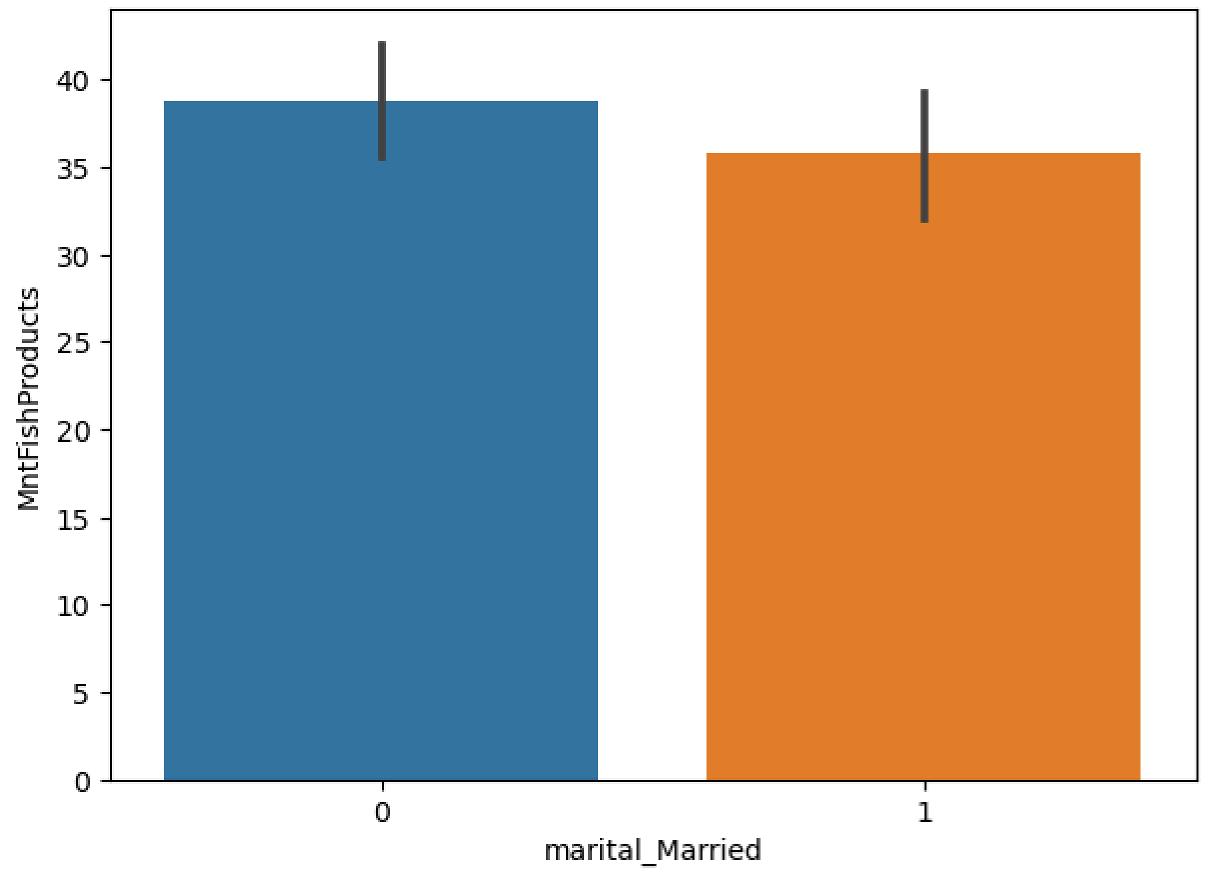


In [54]:

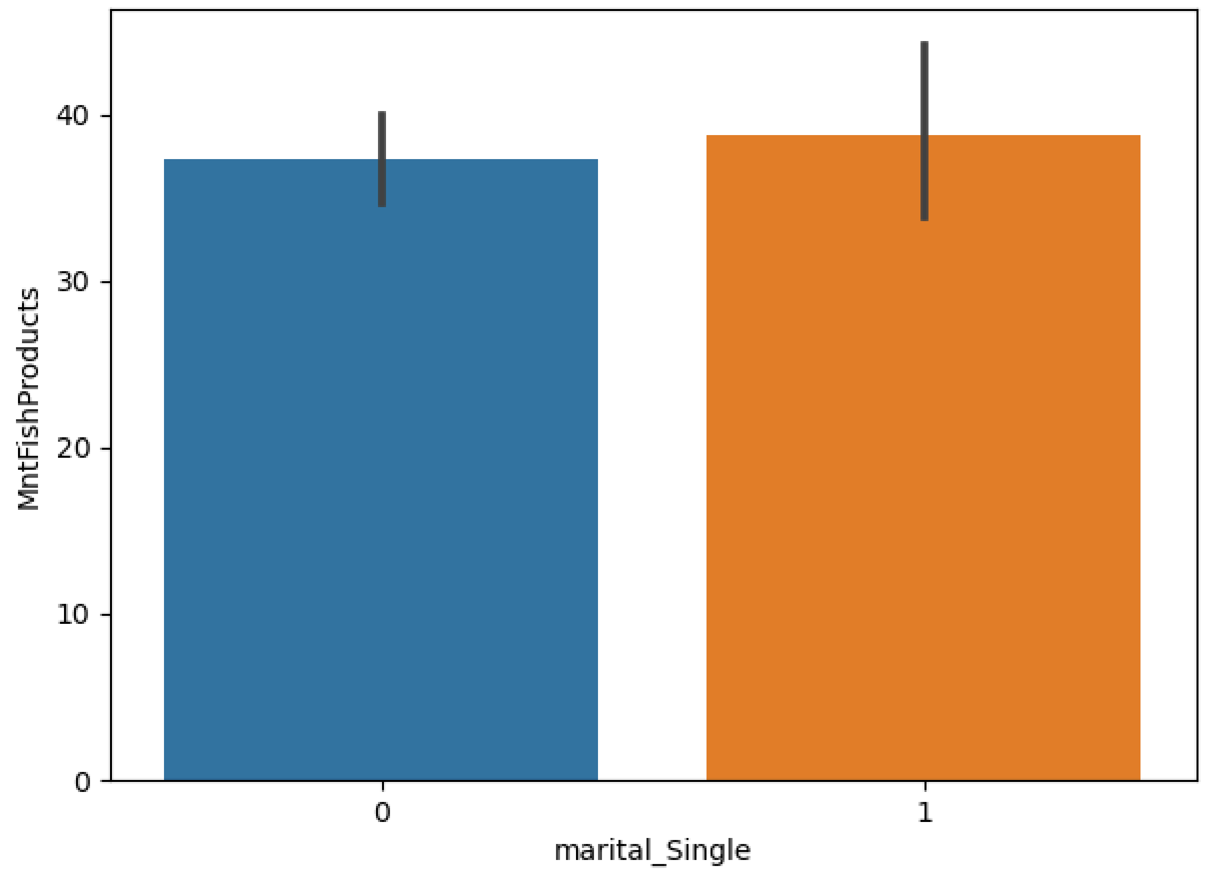
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'MntFishProducts', df)
```



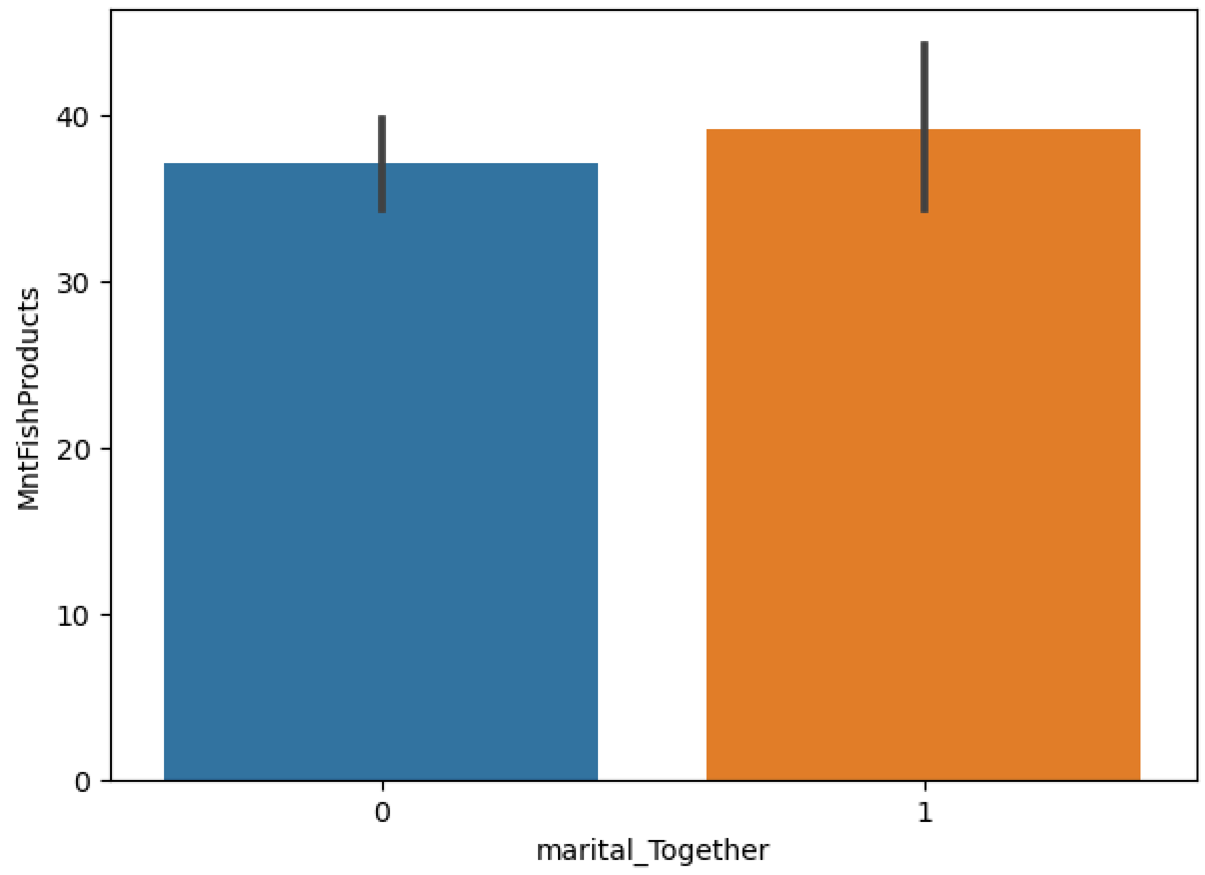
MntFishProducts
0
5
10
15
20
25
30
35
40
0
1
marital_Divorced



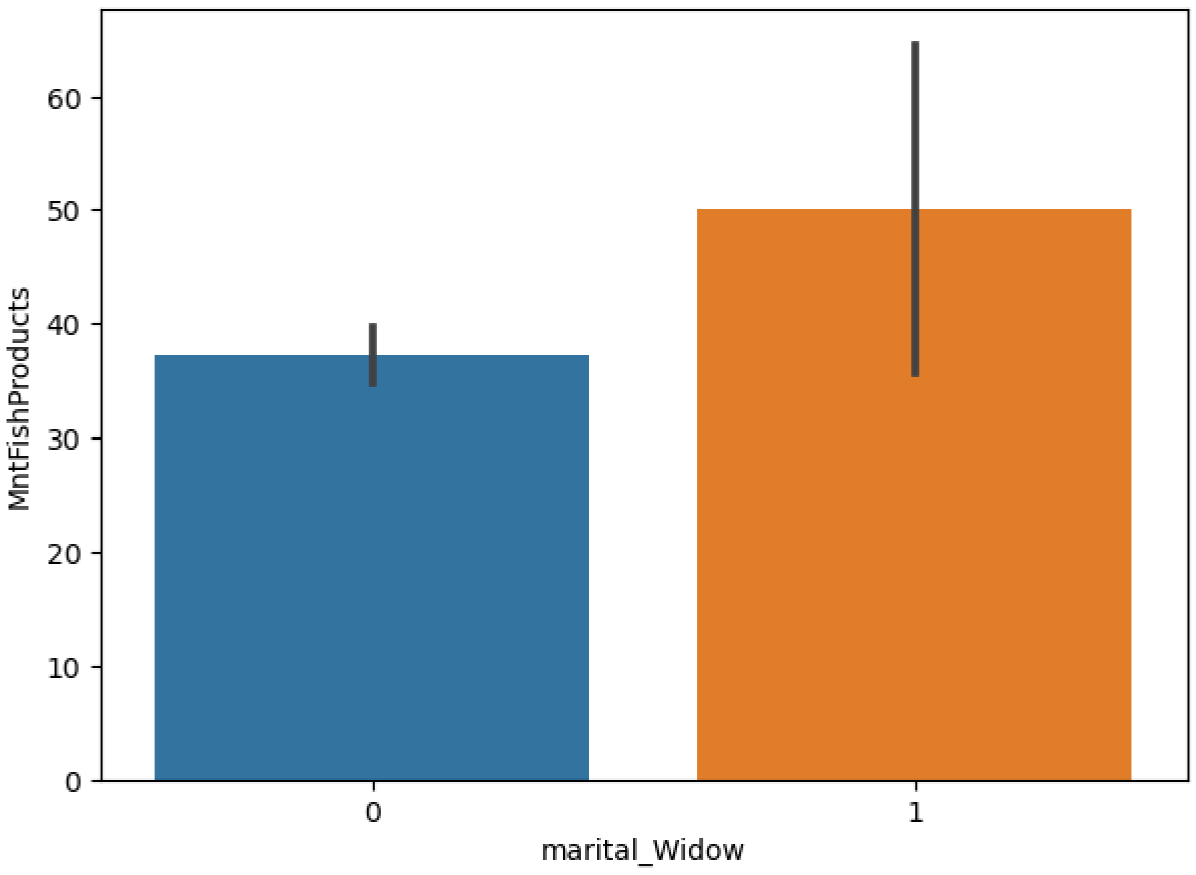
MntFishProducts
0
5
10
15
20
25
30
35
40
0
1
marital_Married



MntFishProducts
0
10
20
30
40
0
1
marital_Single

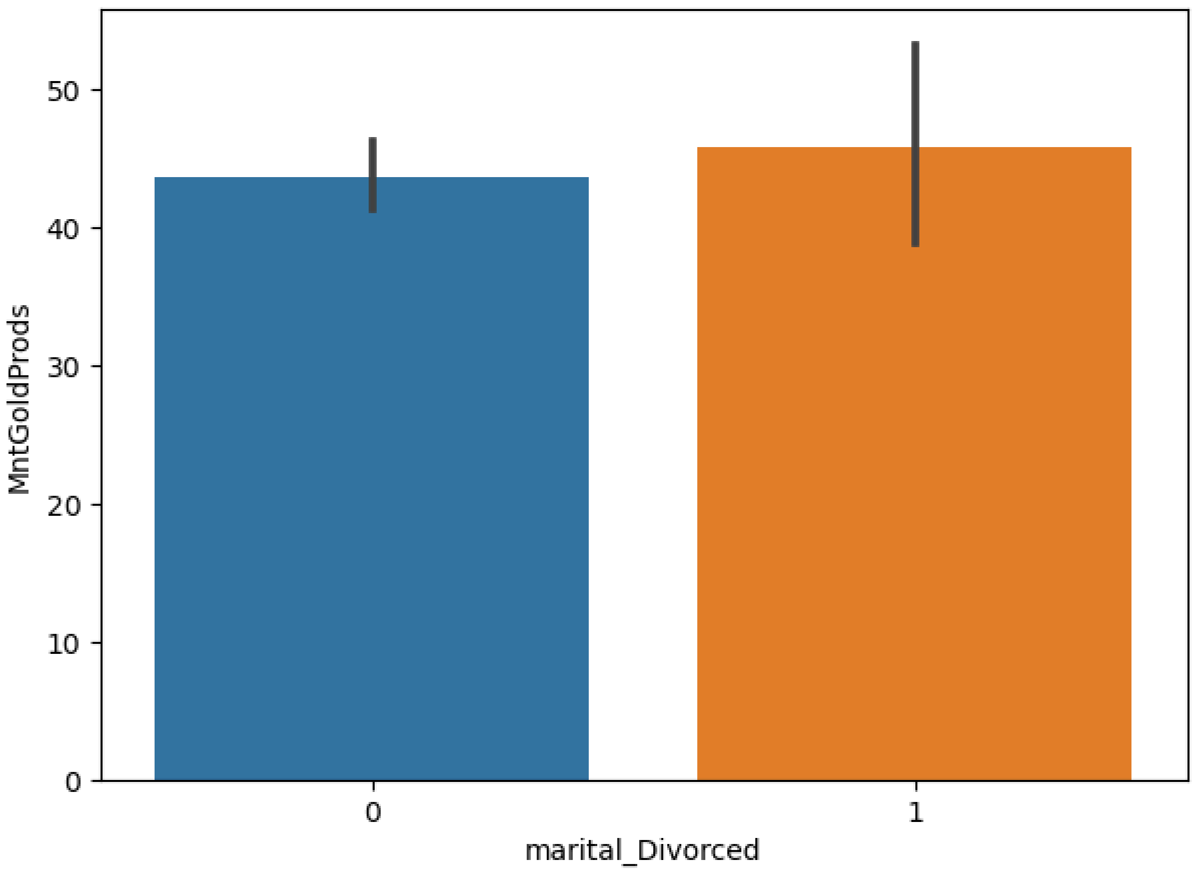


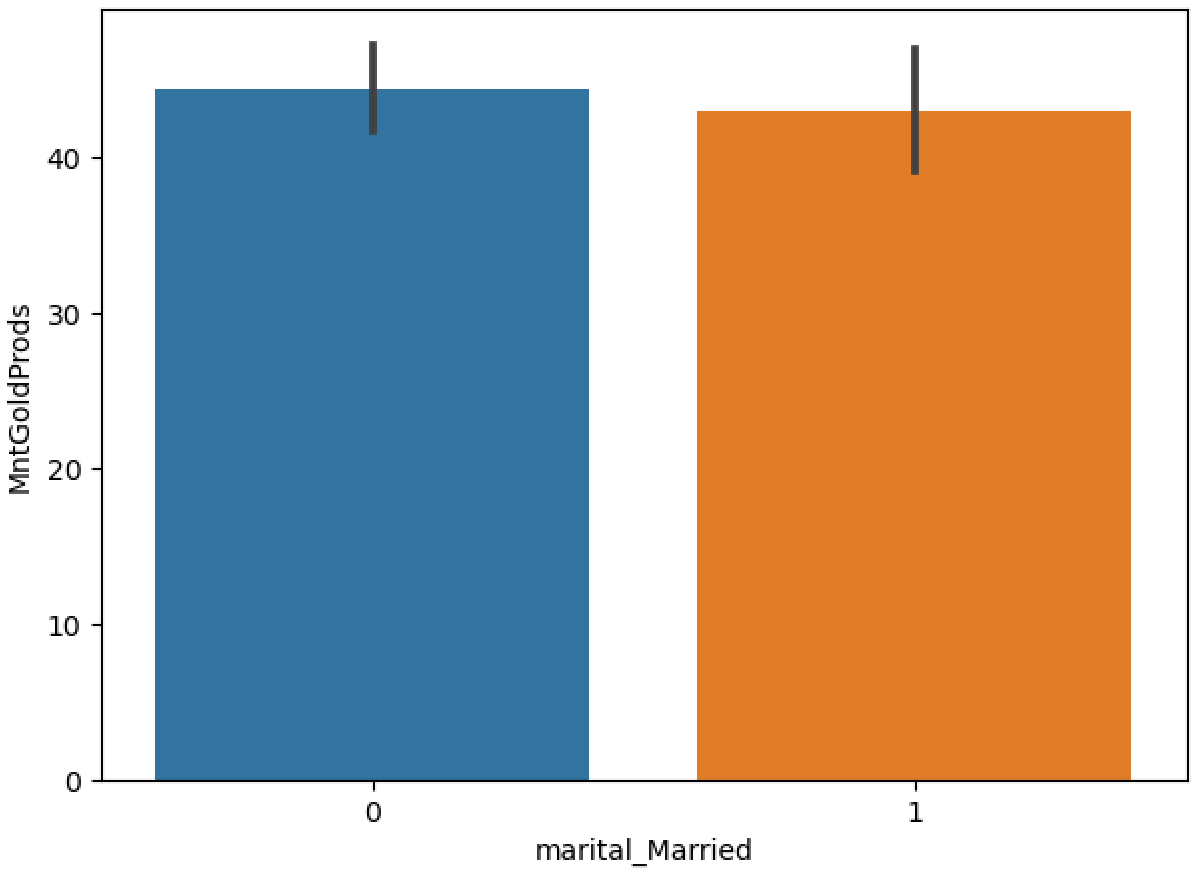
MntFishProducts
0
10
20
30
40
0
1
marital_Together

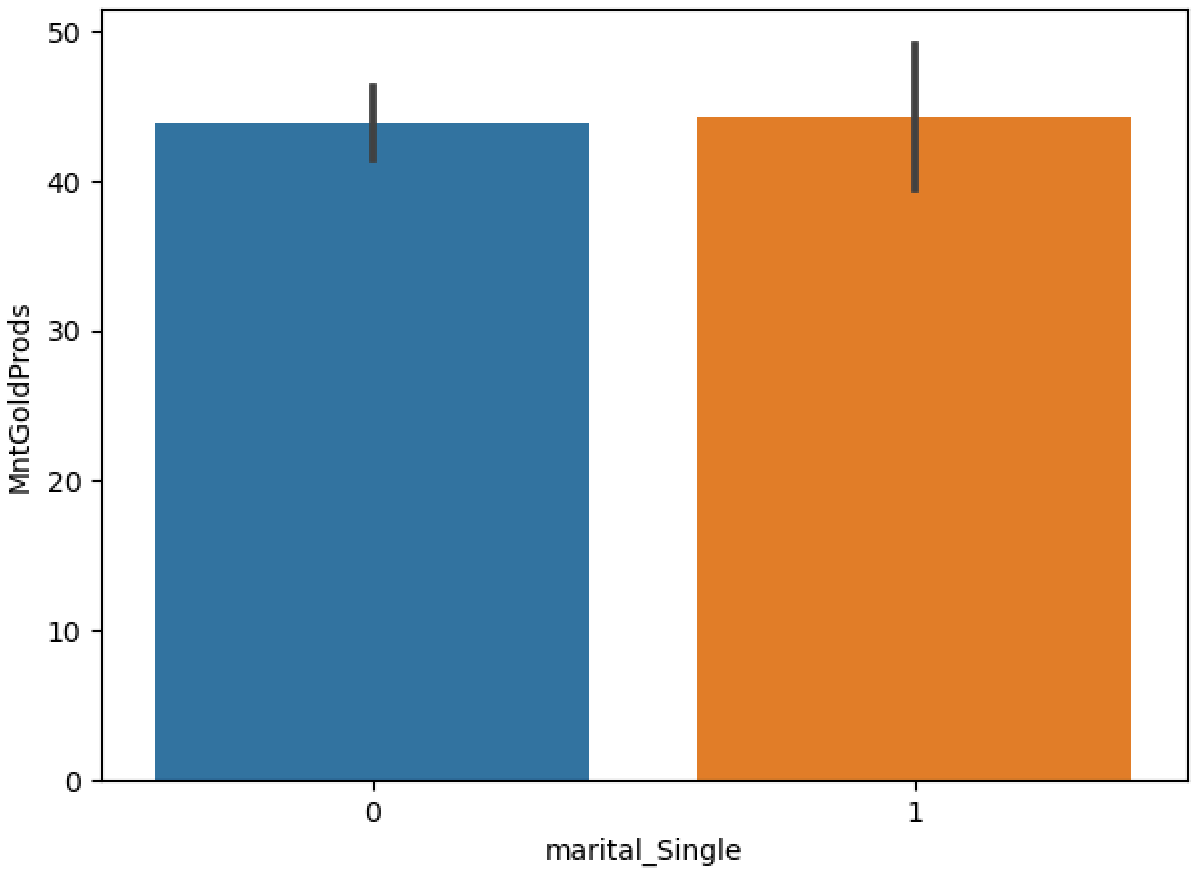


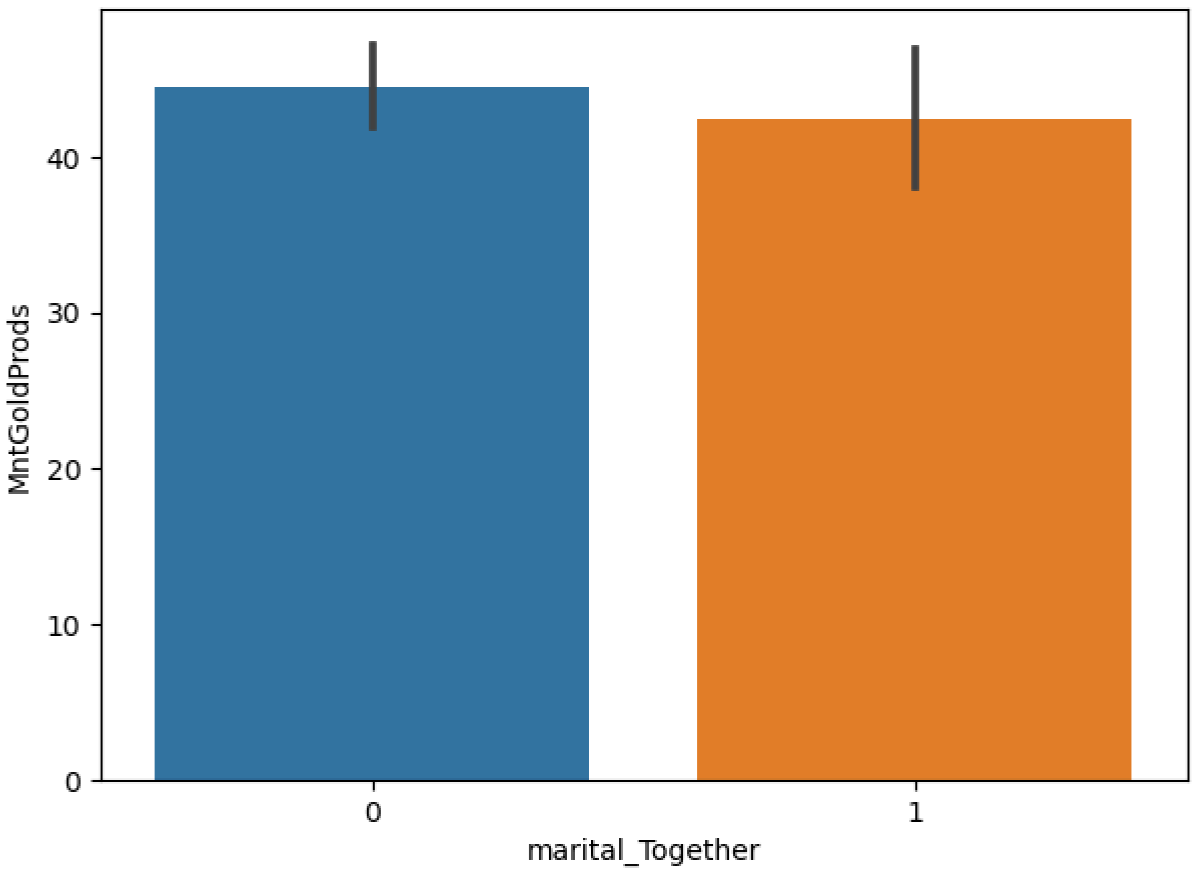
In [55]:

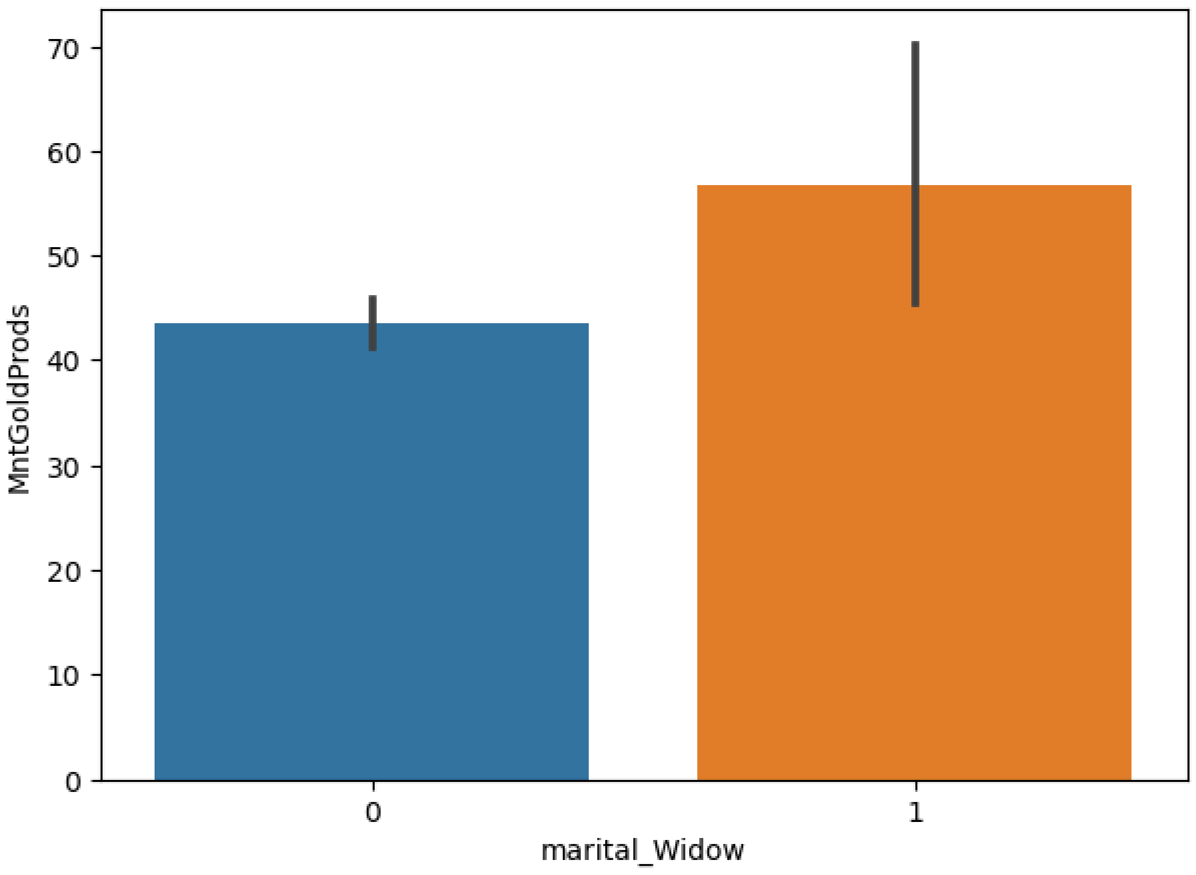
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'MntGoldProds', df)
```





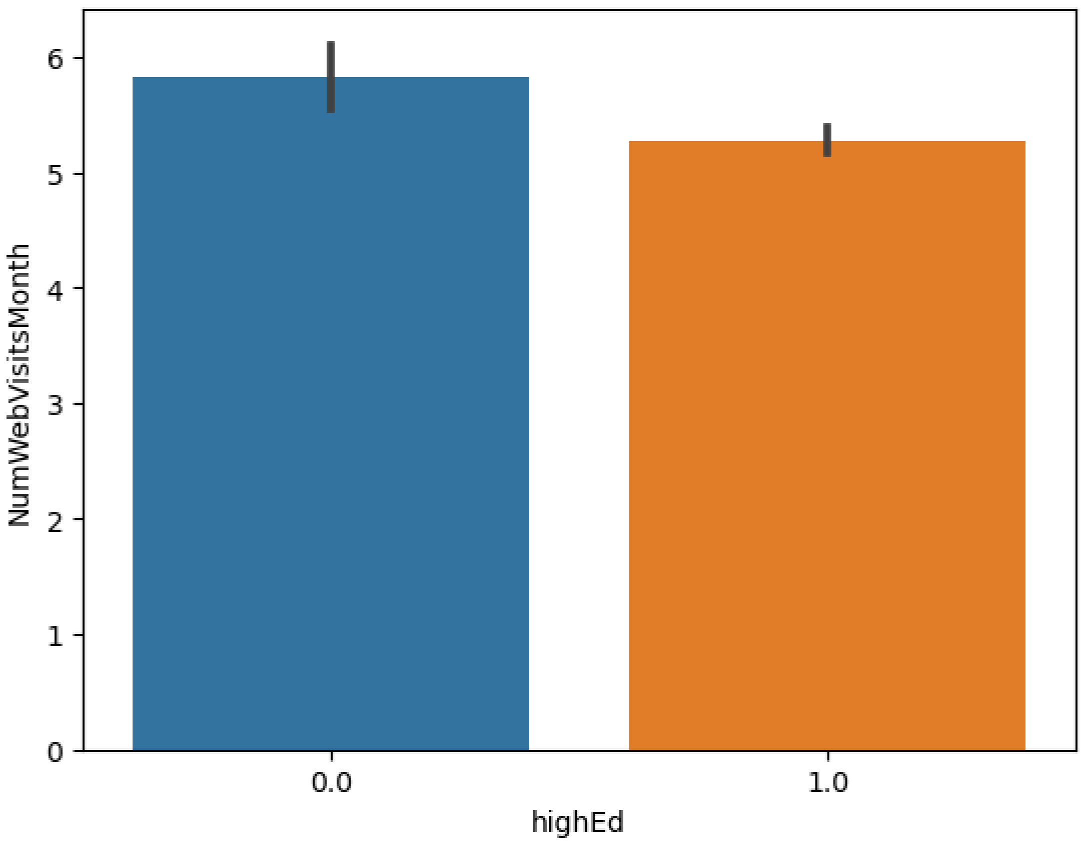






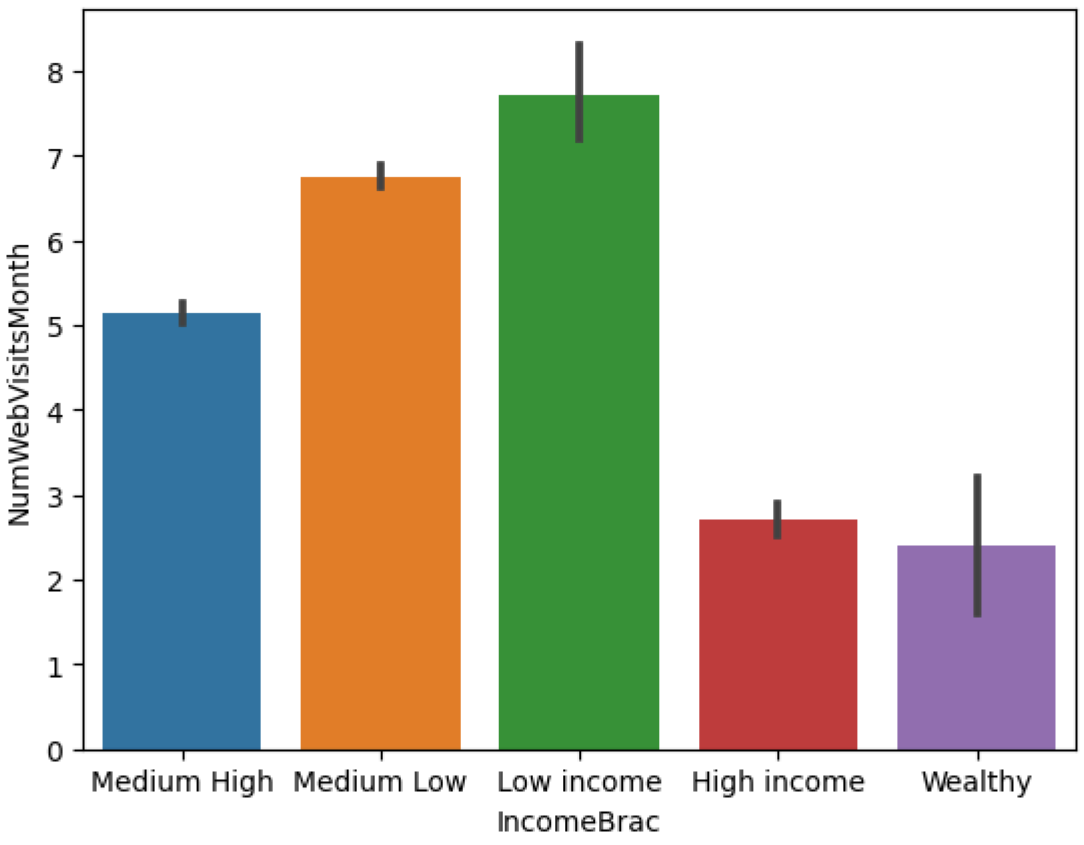
```
In [56]: sns.barplot(x='highEd', y='NumWebVisitsMonth', data = df)
```

```
Out[56]: <AxesSubplot:xlabel='highEd', ylabel='NumWebVisitsMonth'>
```



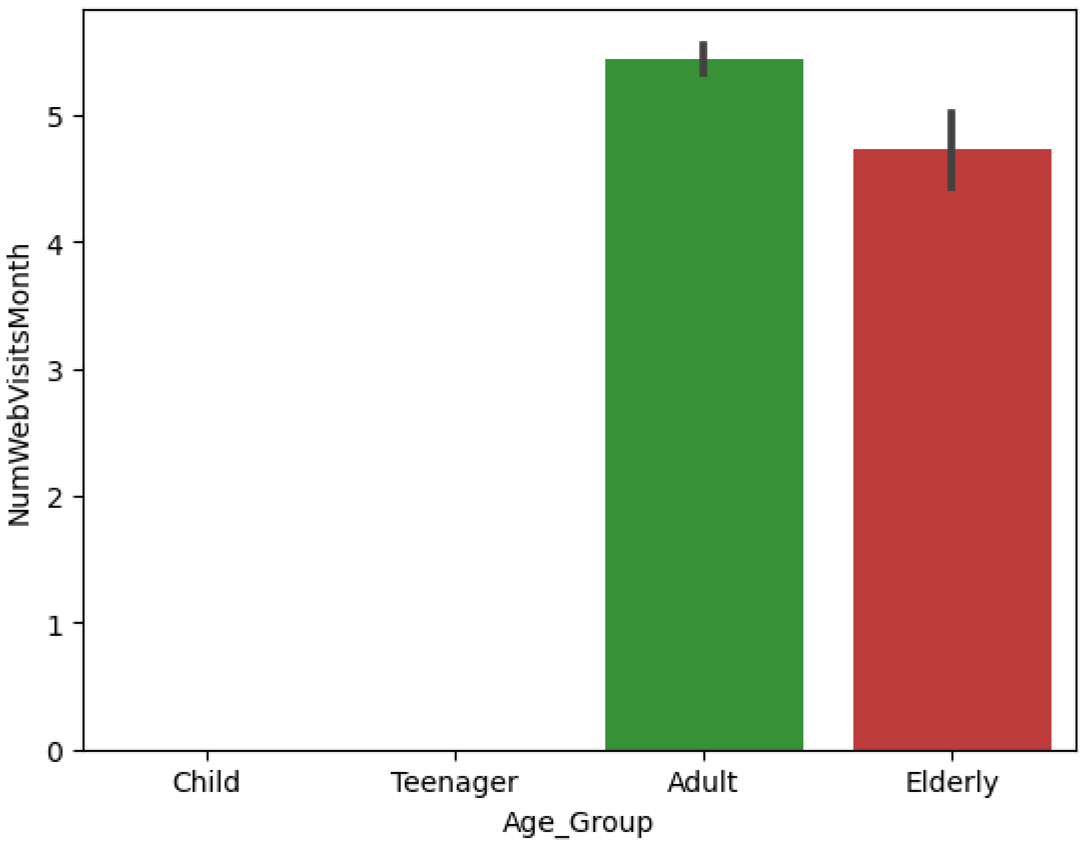
```
In [57]: sns.barplot(x='IncomeBrac', y='NumWebVisitsMonth', data = df)
```

```
Out[57]: <AxesSubplot:xlabel='IncomeBrac', ylabel='NumWebVisitsMonth'>
```

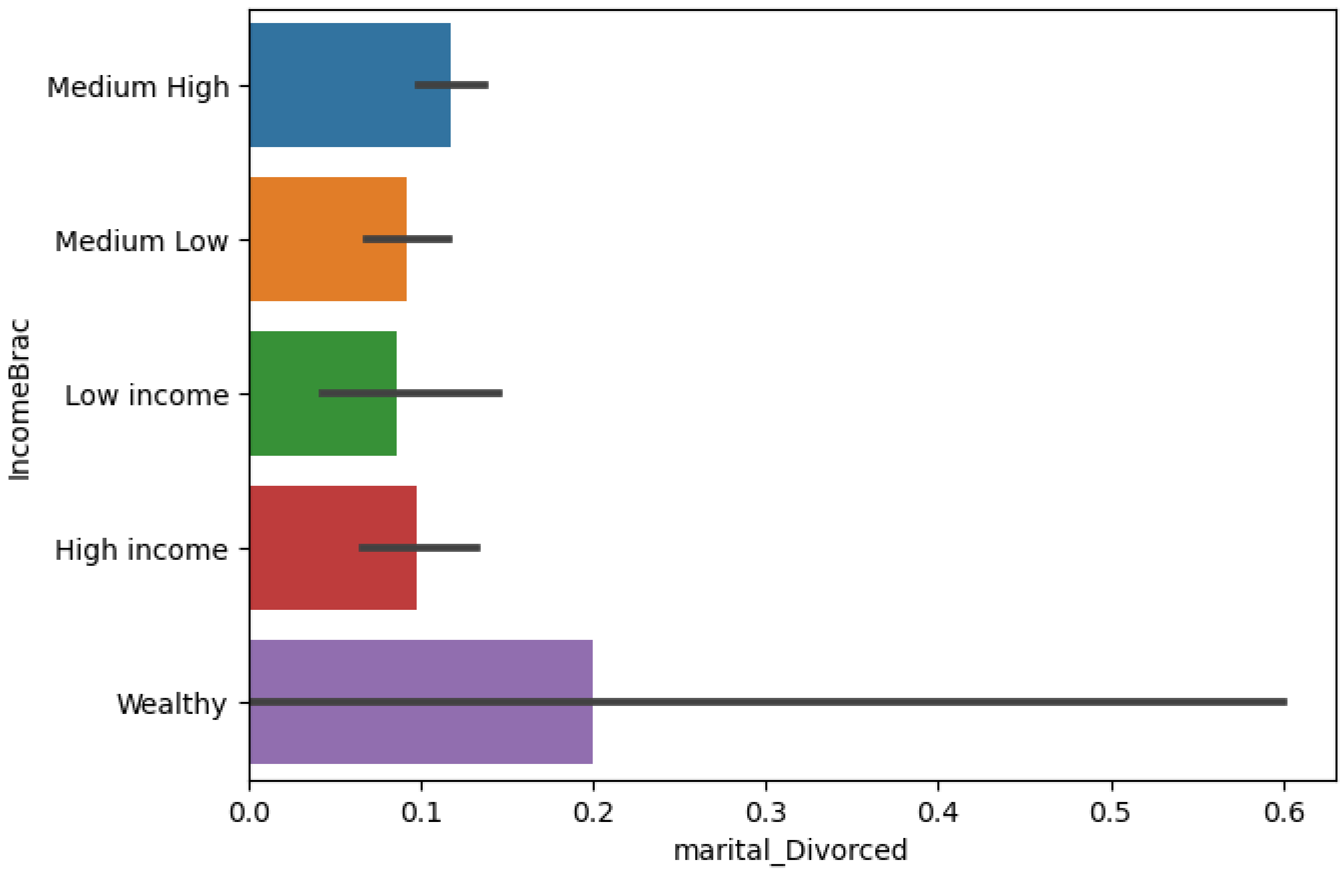


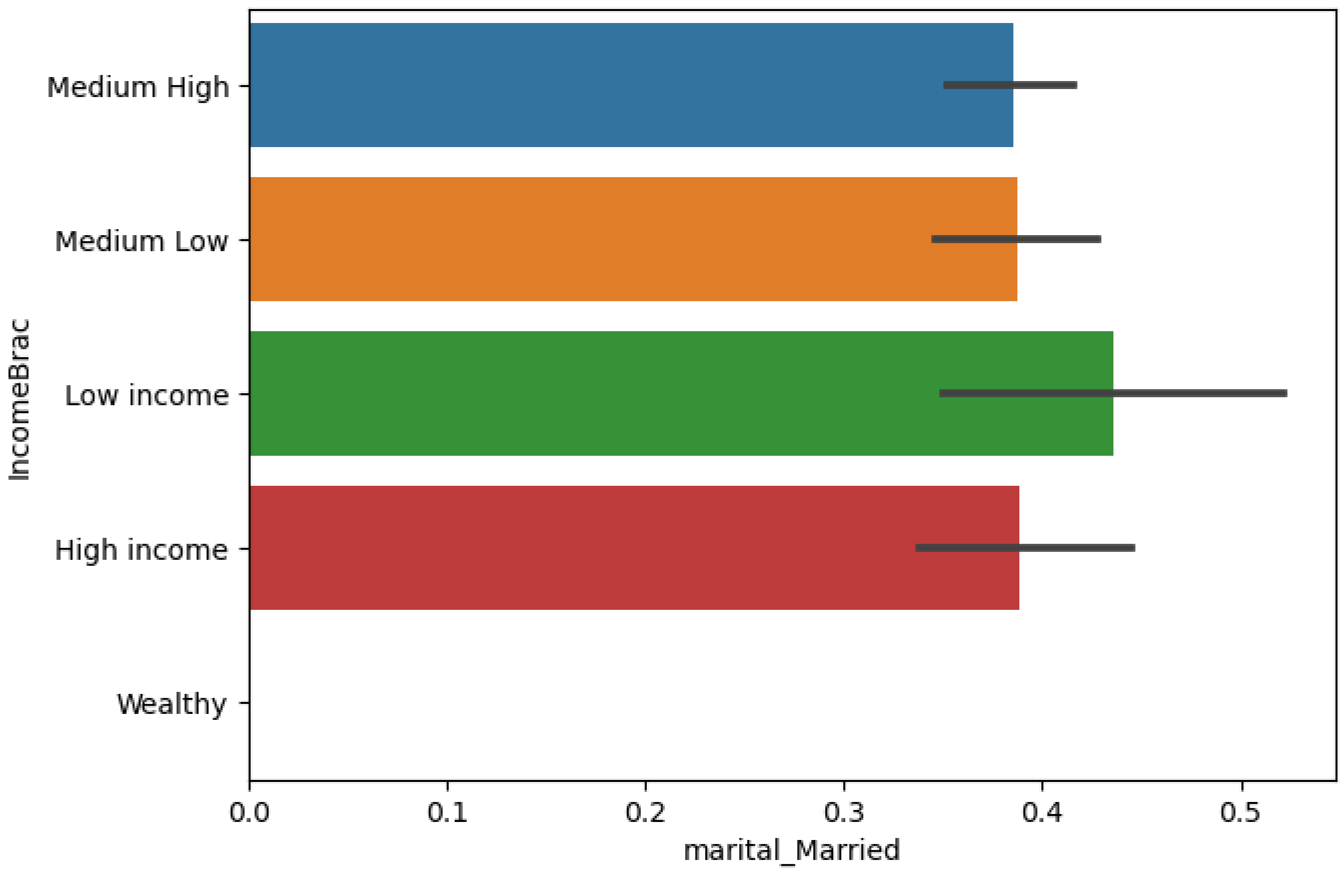
```
In [58]: sns.barplot(x='Age_Group', y='NumWebVisitsMonth', data = df)
```

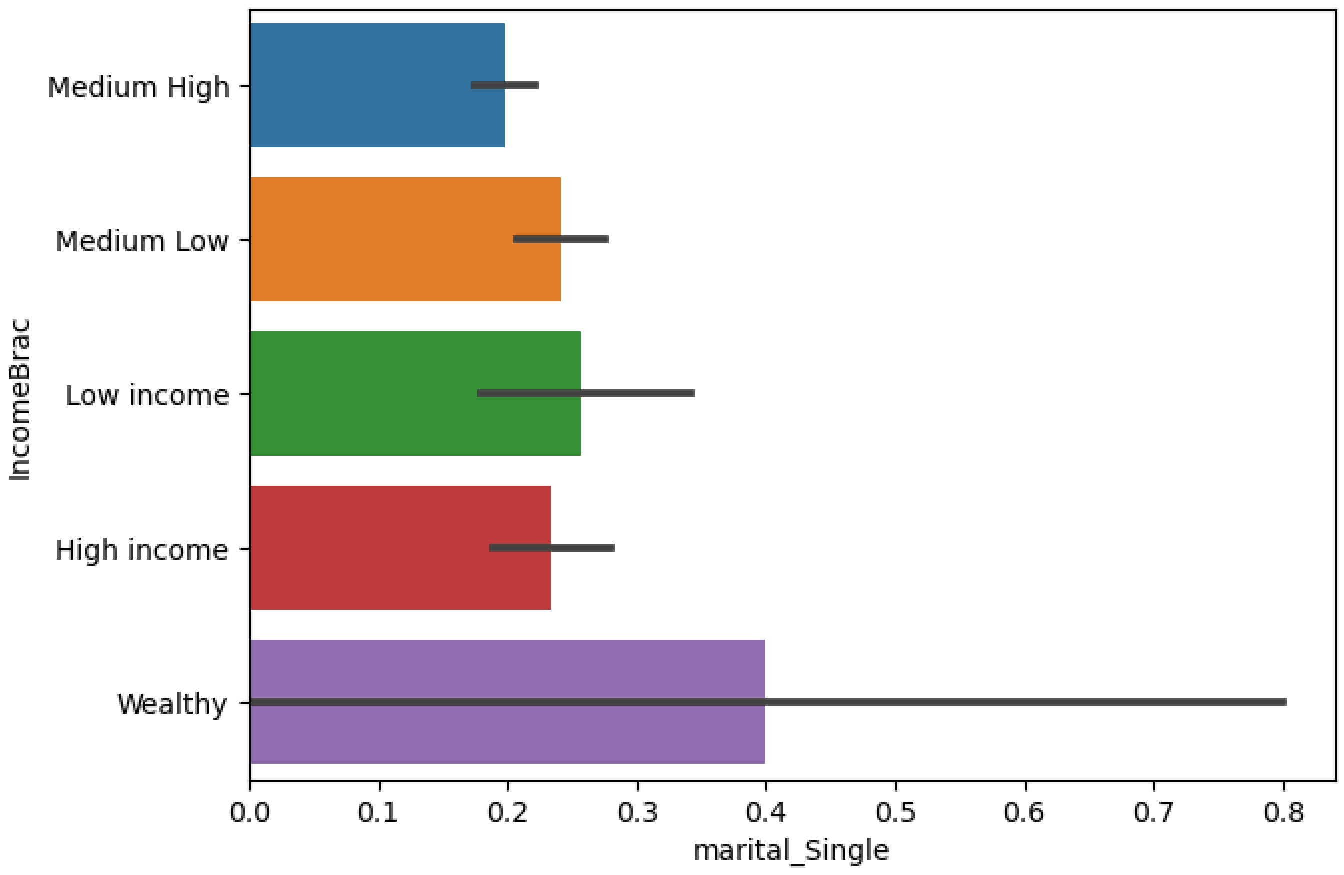
```
Out[58]: <AxesSubplot:xlabel='Age_Group', ylabel='NumWebVisitsMonth'>
```

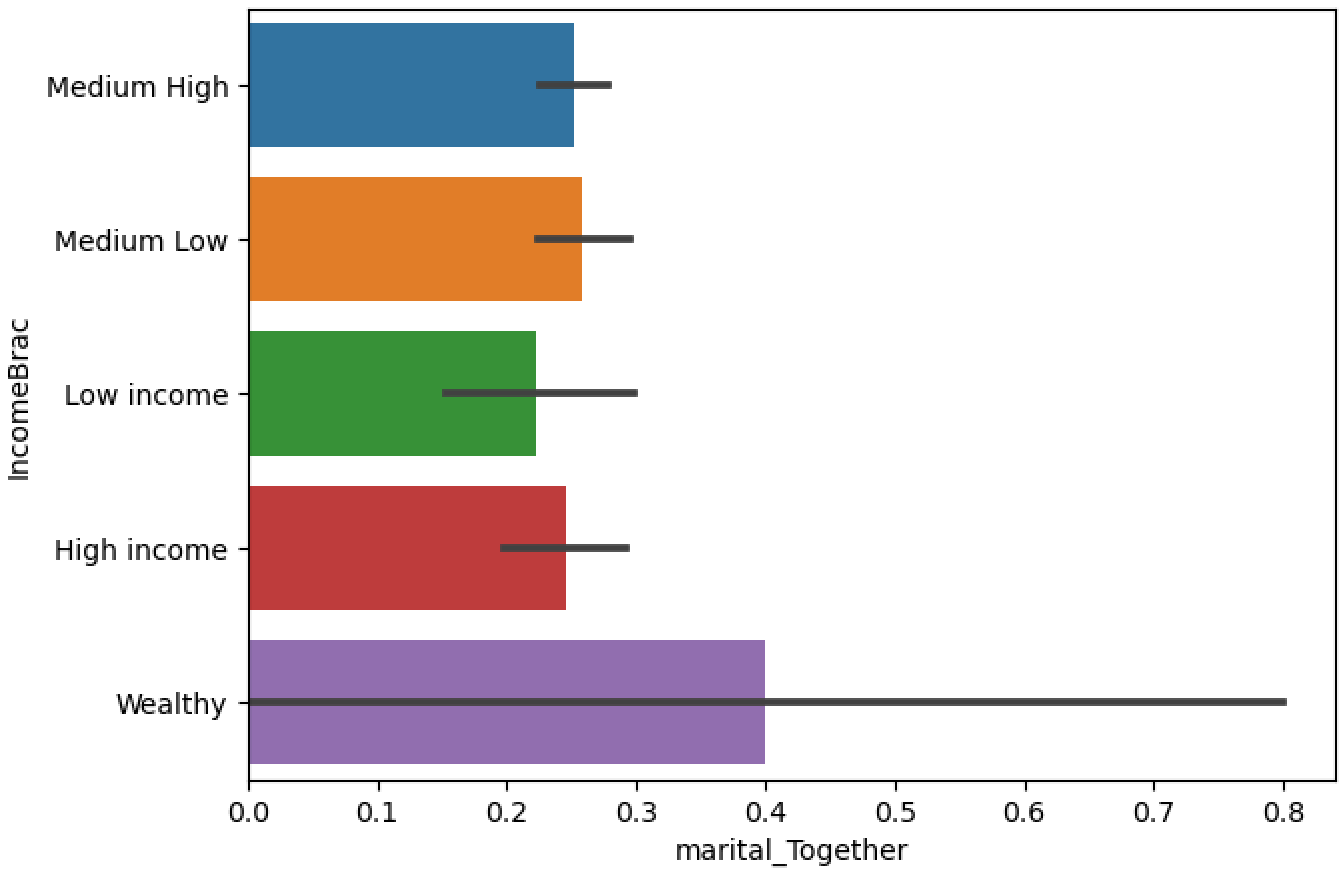


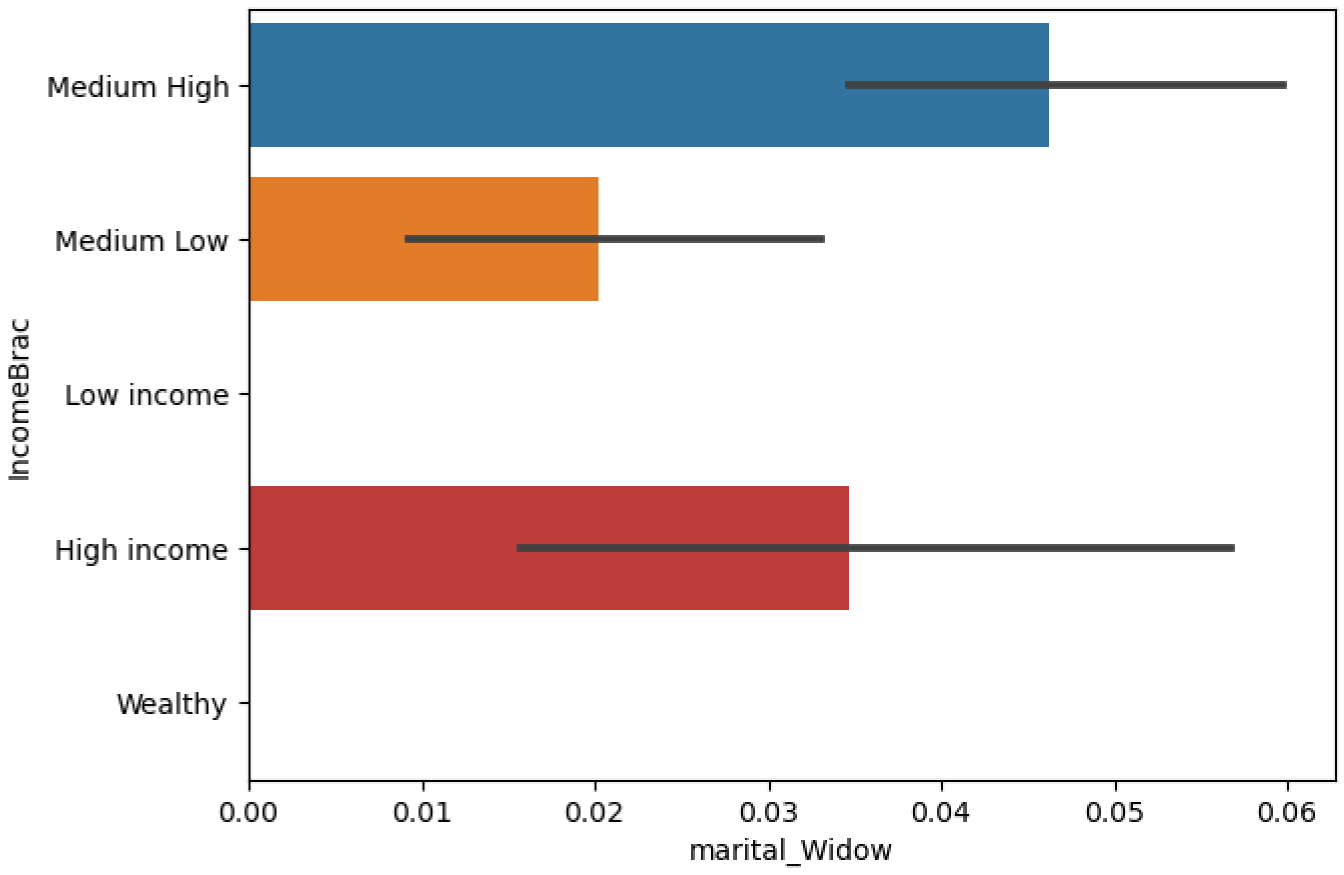
```
In [59]: def plotter( X, Y, df):
             plt.figure(figsize=(7,5))


             sns.barplot(x=X, y=Y, data = df)
             plt.show()


         maritalStat = ['marital_Divorced', 'marital_Married',
                'marital_Single', 'marital_Together', 'marital_Widow']
         for marit in maritalStat:
             plotter(marit, 'IncomeBrac', df)
```





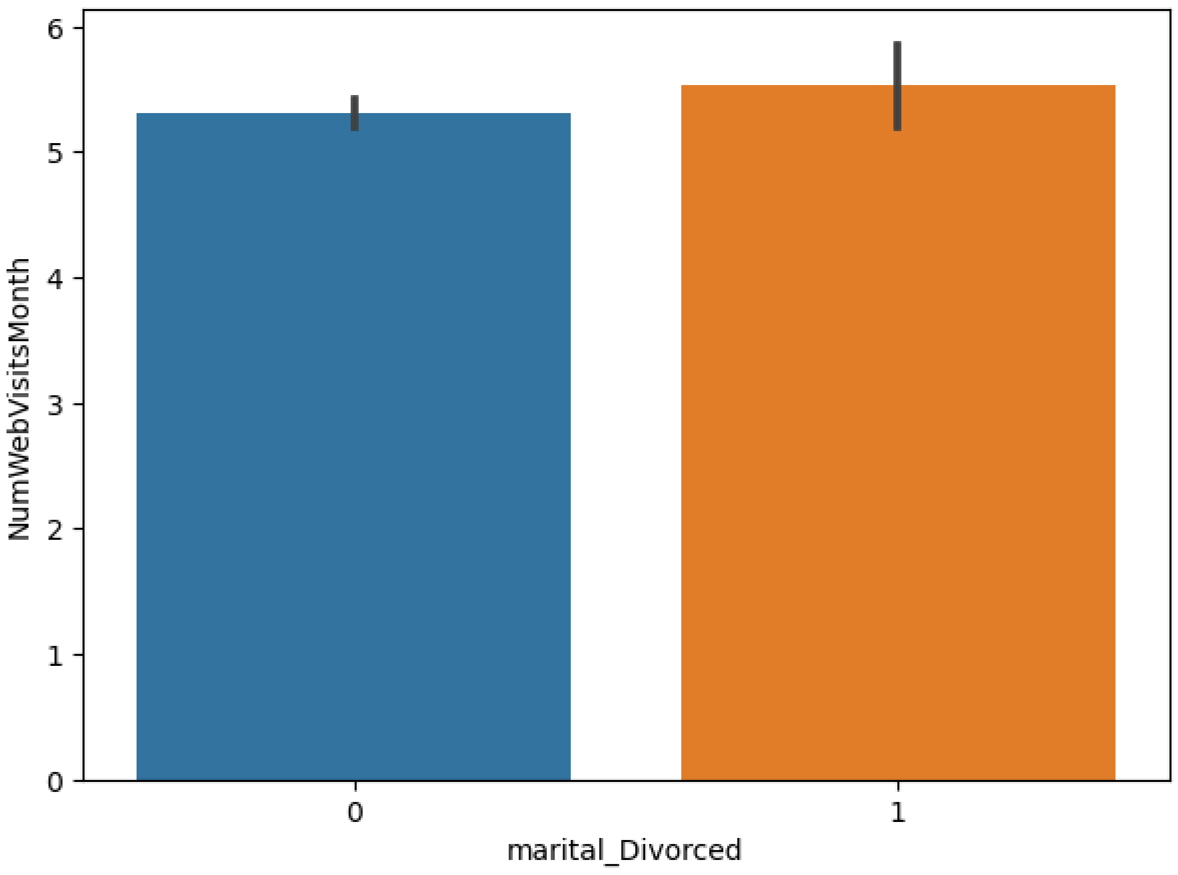


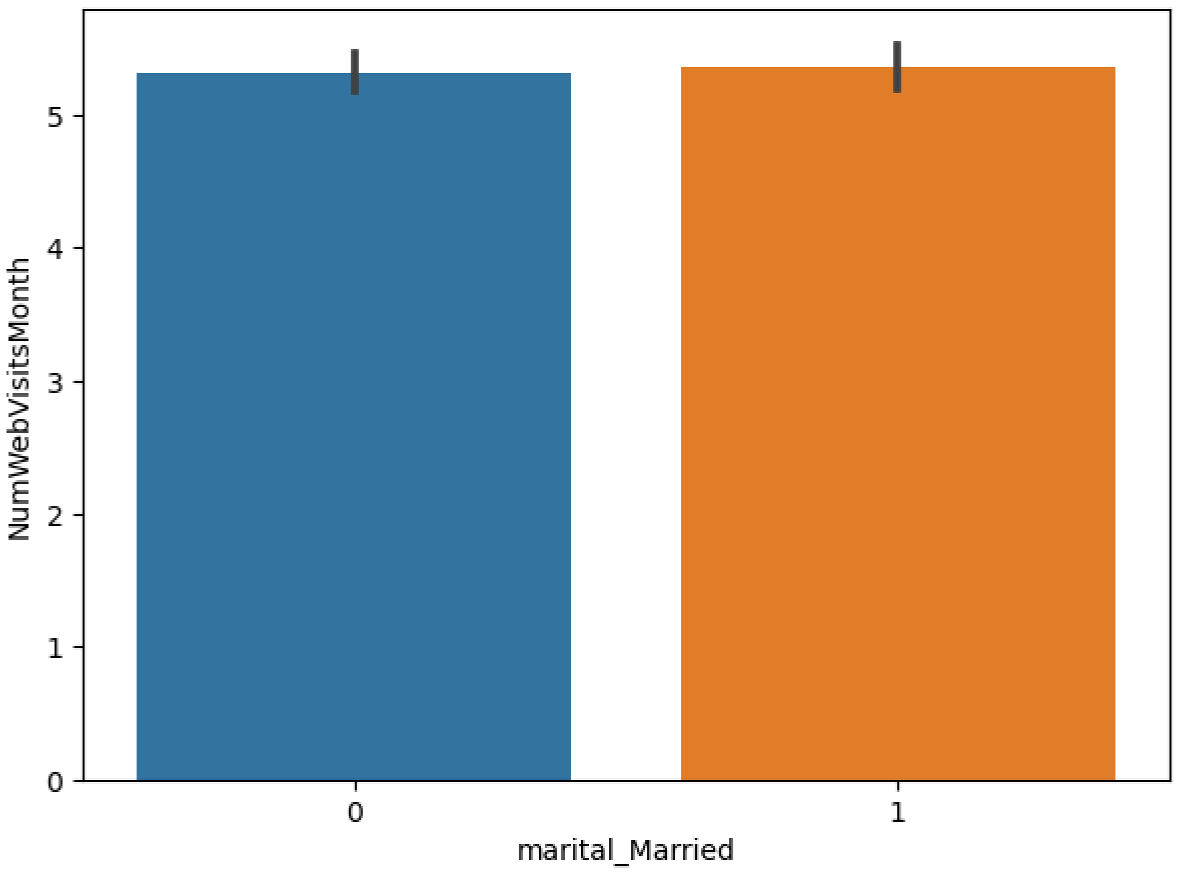


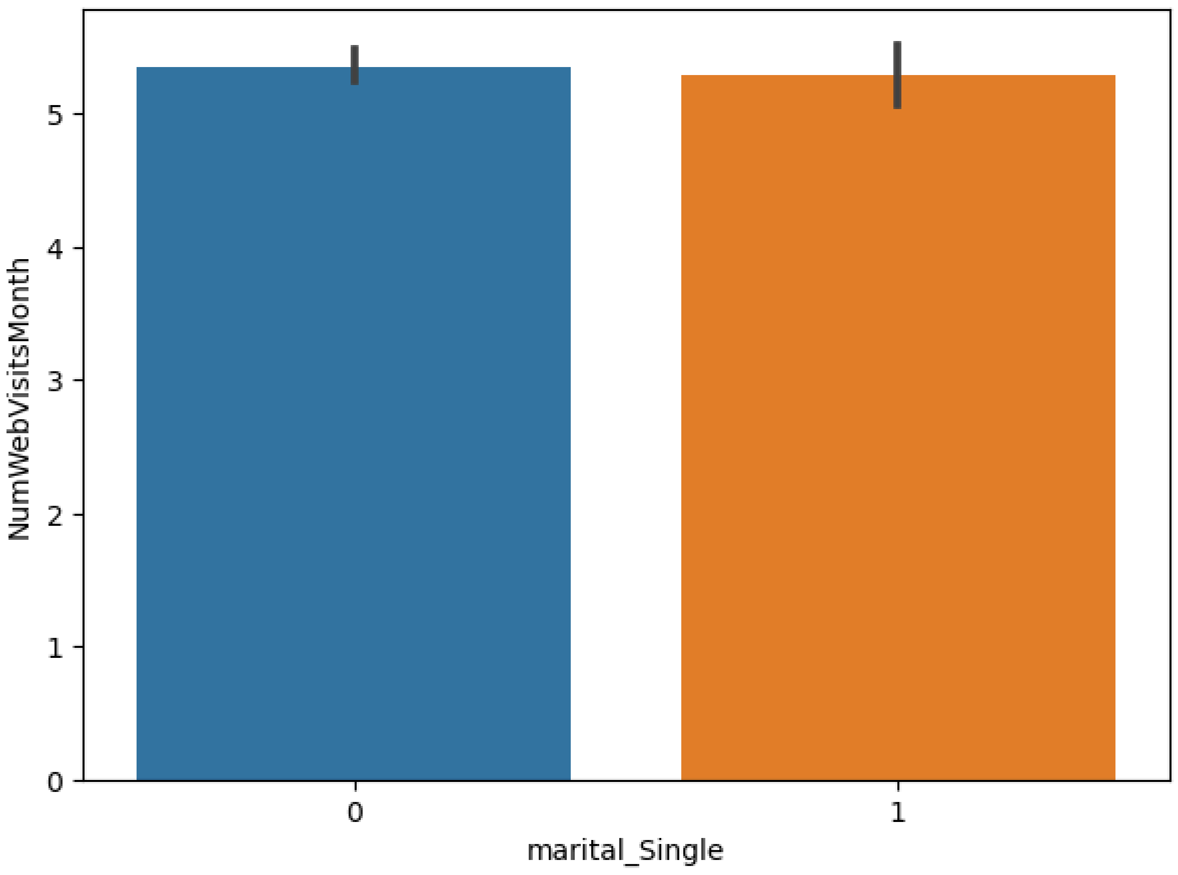


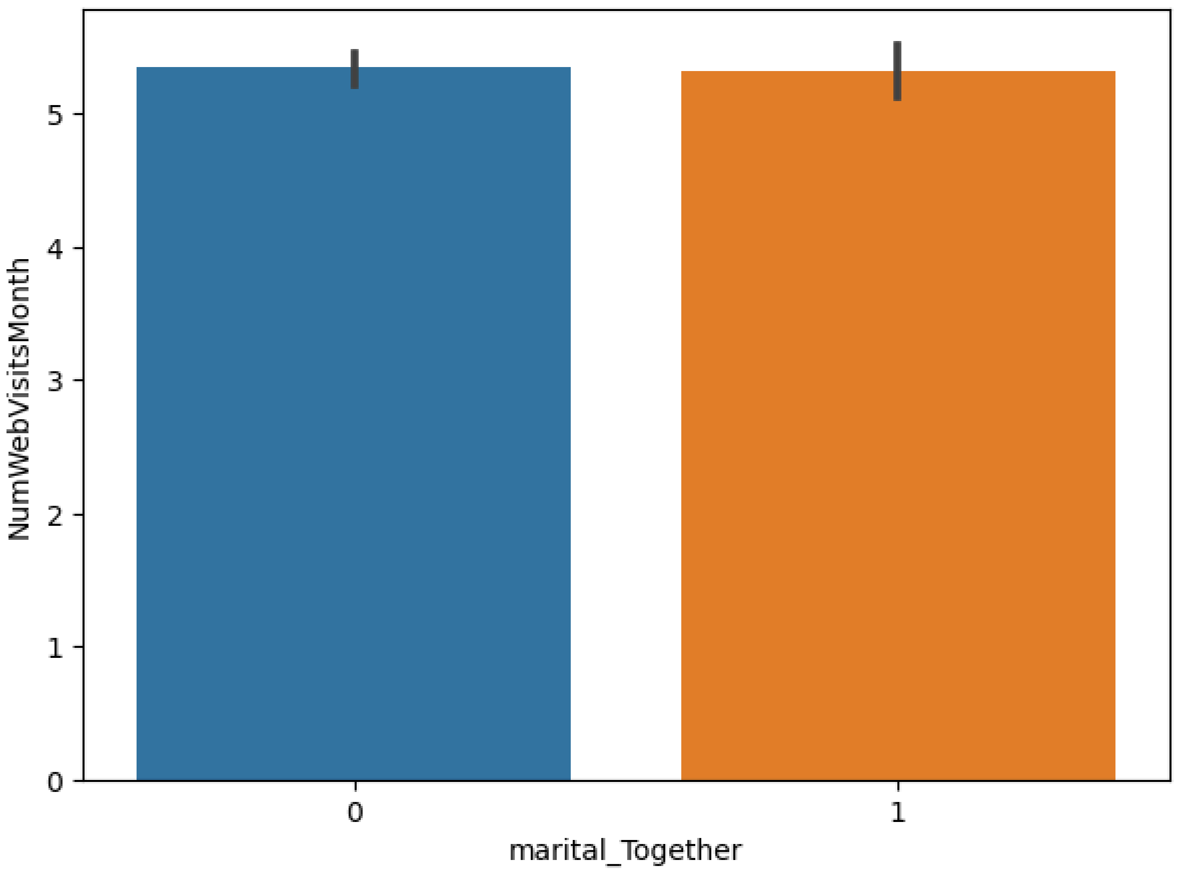
In [60]:

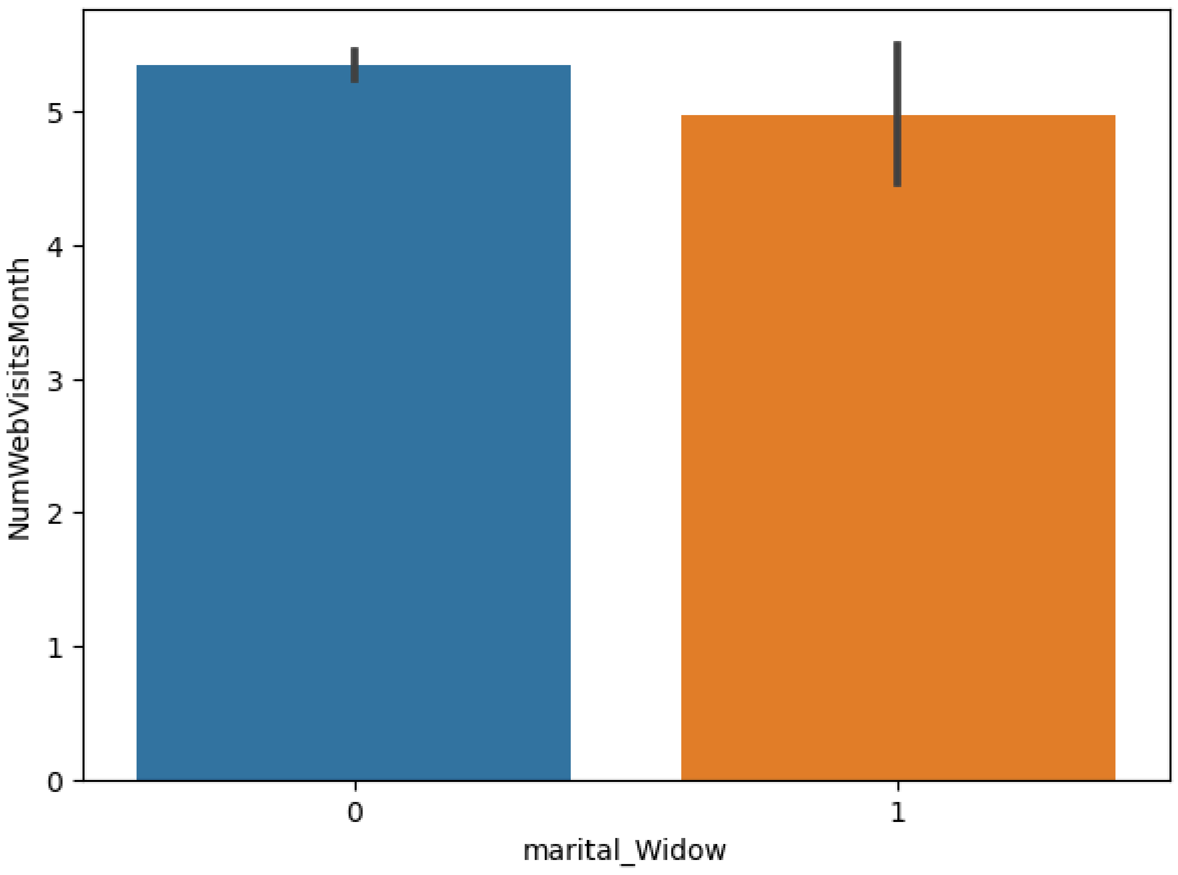
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'NumWebVisitsMonth', df)
```





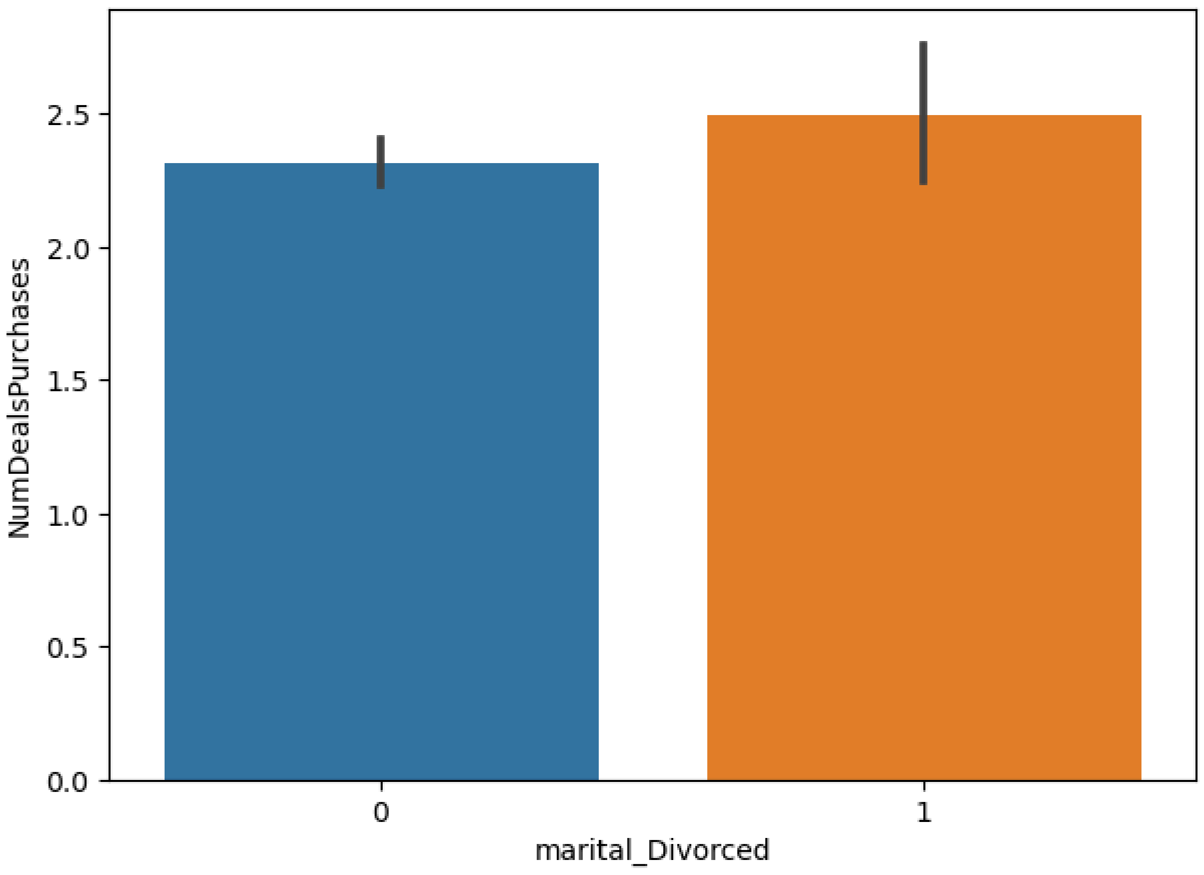


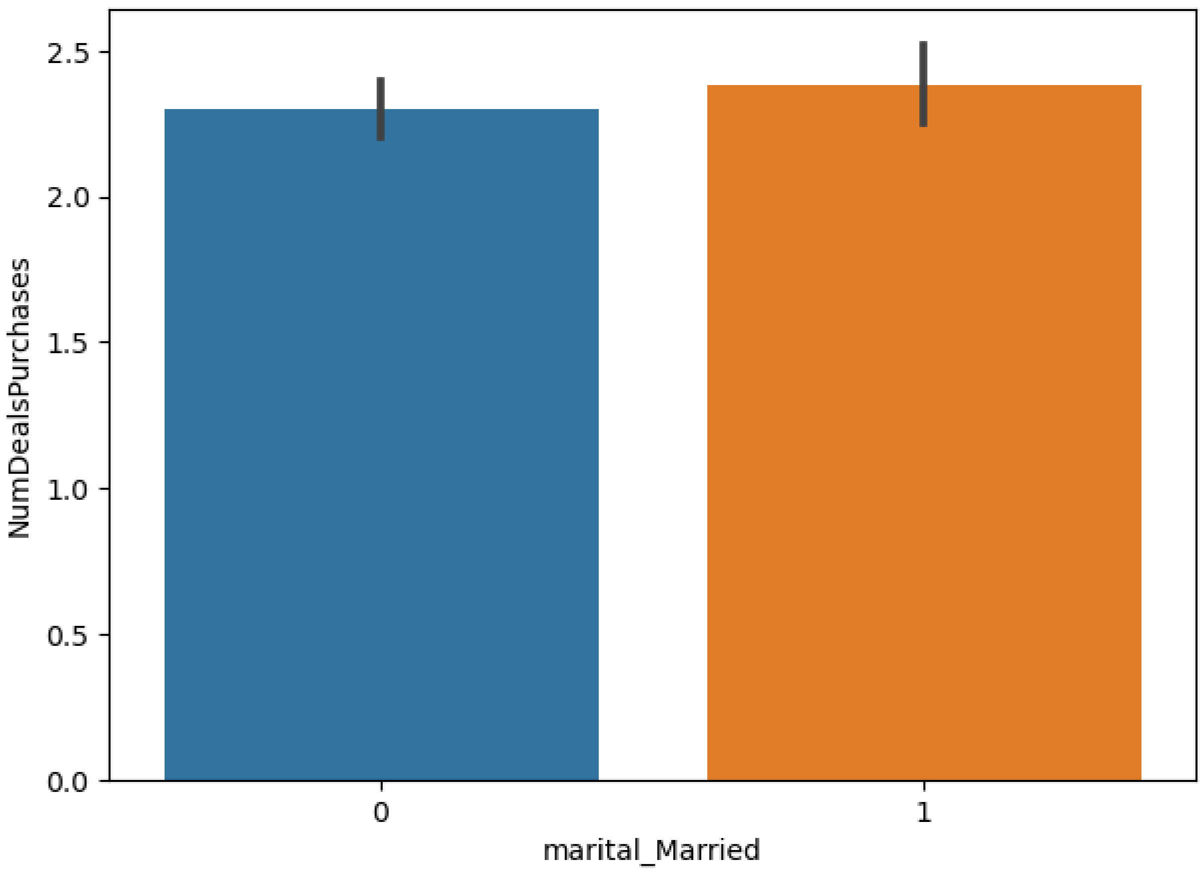


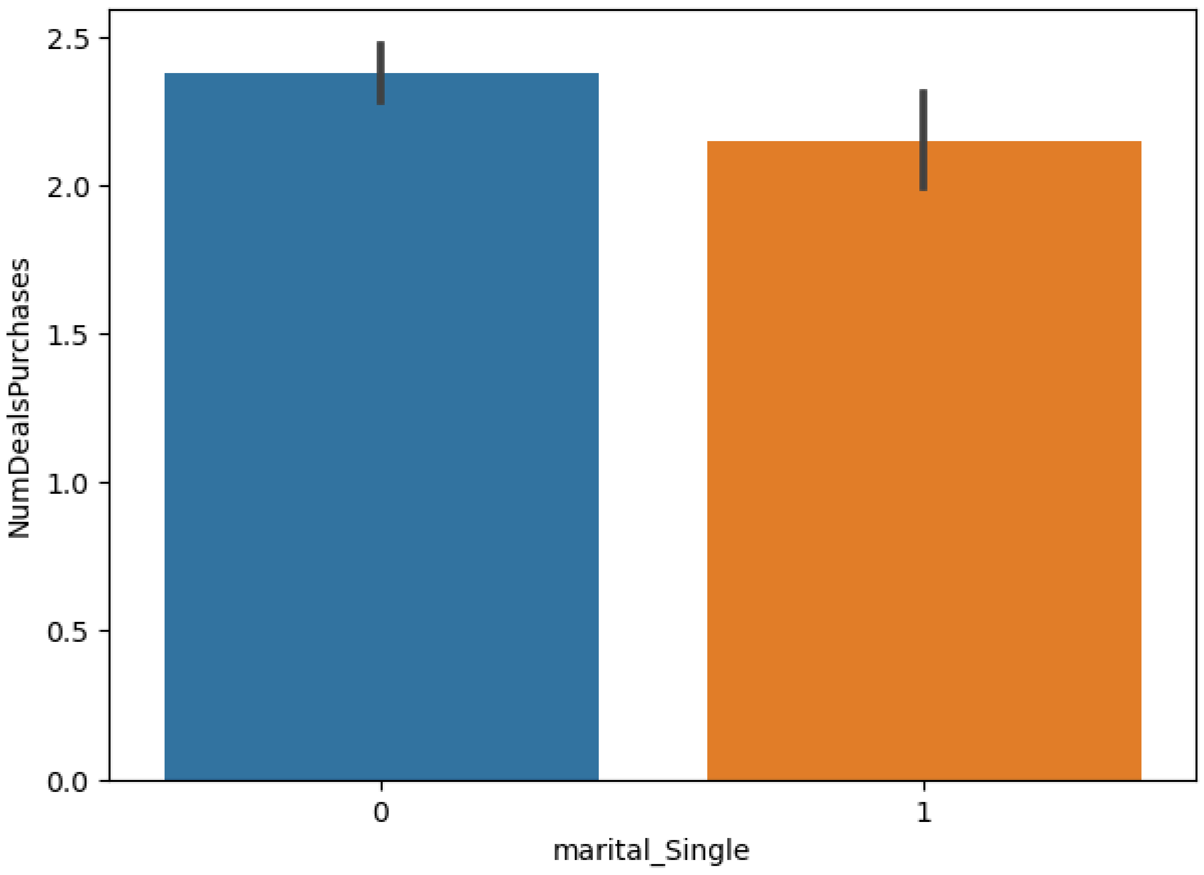


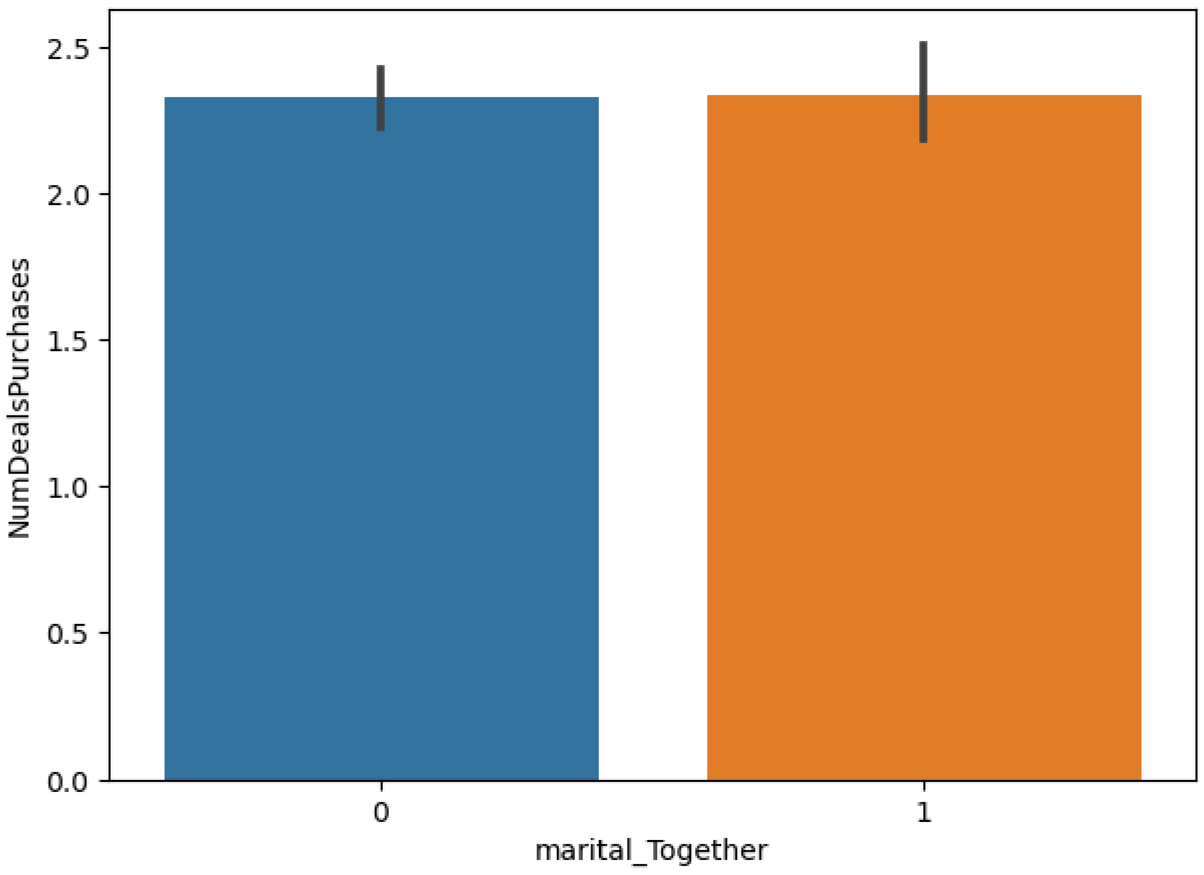
In [61]:

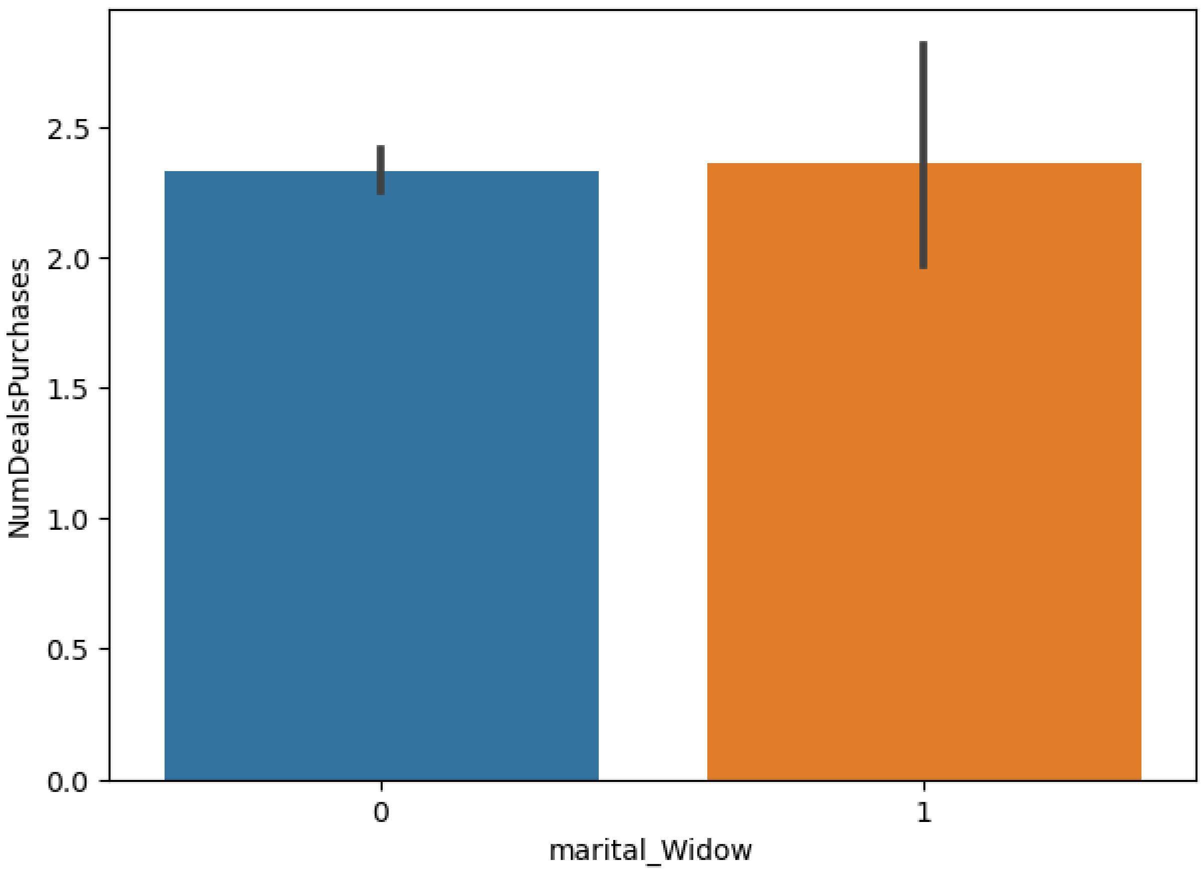
```
def plotter( X, Y, df):
    plt.figure(figsize=(7,5))


    sns.barplot(x=X, y=Y, data = df)
    plt.show()


maritalStat = ['marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow']
for marit in maritalStat:
    plotter(marit, 'NumDealsPurchases', df)
```











```
In [62]: df.MntMeatProducts.sum()
```

```
Out[62]: 335607
```

```
In [63]: df.MntMeatProducts.mean()
```

```
Out[63]: 166.05987135081642
```

```
In [64]: df.MntWines.sum()
```

```
Out[64]: 619421
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

In [ ]:

In [ ]: