

Project 8 - Marketing Campaign

January 19, 2023

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

```
[2]: df = pd.read_excel('marketing_campaign.xlsx')
```

```
[3]: df.head()
```

```
[3]:      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
```

```
      Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  AcceptedCmp3  \
0  2012-09-04      58      635  ...      7      0
1  2014-03-08      38       11  ...      5      0
2  2013-08-21      26      426  ...      4      0
3  2014-02-10      26       11  ...      6      0
4  2014-01-19      94      173  ...      5      0
```

```
      AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0      0      0      0      0      0
1      0      0      0      0      0
2      0      0      0      0      0
3      0      0      0      0      0
4      0      0      0      0      0
```

```
      Z_CostContact  Z_Revenue  Response
0      3      11      1
1      3      11      0
2      3      11      0
3      3      11      0
4      3      11      0
```

[5 rows x 29 columns]

```
[4]: df.tail()
```

```
[4]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	\
2235	10870	1967	Graduation	Married	61223.0	0	
2236	4001	1946	PhD	Together	64014.0	2	
2237	7270	1981	Graduation	Divorced	56981.0	0	
2238	8235	1956	Master	Together	69245.0	0	
2239	9405	1954	PhD	Married	52869.0	1	

	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	\
2235	1	2013-06-13	46	709	...	5	
2236	1	2014-06-10	56	406	...	7	
2237	0	2014-01-25	91	908	...	6	
2238	1	2014-01-24	8	428	...	3	
2239	1	2012-10-15	40	84	...	7	

	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	\
2235	0	0	0	0	0	
2236	0	0	0	1	0	
2237	0	1	0	0	0	
2238	0	0	0	0	0	
2239	0	0	0	0	0	

	Complain	Z_CostContact	Z_Revenue	Response
2235	0	3	11	0
2236	0	3	11	0
2237	0	3	11	0
2238	0	3	11	0
2239	0	3	11	1

[5 rows x 29 columns]

```
[5]: df.shape
```

```
[5]: (2240, 29)
```

```
[6]: df.describe()
```

```
[6]:
```

	ID	Year_Birth	Income	Kidhome	Teenhome	\
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	
std	3246.662198	11.984069	25173.076661	0.538398	0.544538	
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	
25%	2828.250000	1959.000000	35303.000000	0.000000	0.000000	
50%	5458.500000	1970.000000	51381.500000	0.000000	0.000000	
75%	8427.750000	1977.000000	68522.000000	1.000000	1.000000	
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	

	Recency	MntWines	MntFruits	MntMeatProducts	\
count	2240.000000	2240.000000	2240.000000	2240.000000	
mean	49.109375	303.935714	26.302232	166.950000	
std	28.962453	336.597393	39.773434	225.715373	
min	0.000000	0.000000	0.000000	0.000000	
25%	24.000000	23.750000	1.000000	16.000000	
50%	49.000000	173.500000	8.000000	67.000000	
75%	74.000000	504.250000	33.000000	232.000000	
max	99.000000	1493.000000	199.000000	1725.000000	

	MntFishProducts	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	\
count	2240.000000	...	2240.000000	2240.000000	2240.000000	
mean	37.525446	...	5.316518	0.072768	0.074554	
std	54.628979	...	2.426645	0.259813	0.262728	
min	0.000000	...	0.000000	0.000000	0.000000	
25%	3.000000	...	3.000000	0.000000	0.000000	
50%	12.000000	...	6.000000	0.000000	0.000000	
75%	50.000000	...	7.000000	0.000000	0.000000	
max	259.000000	...	20.000000	1.000000	1.000000	

	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Complain	Z_CostContact	\
count	2240.000000	2240.000000	2240.000000	2240.000000	2240.0	
mean	0.072768	0.064286	0.013393	0.009375	3.0	
std	0.259813	0.245316	0.114976	0.096391	0.0	
min	0.000000	0.000000	0.000000	0.000000	3.0	
25%	0.000000	0.000000	0.000000	0.000000	3.0	
50%	0.000000	0.000000	0.000000	0.000000	3.0	
75%	0.000000	0.000000	0.000000	0.000000	3.0	
max	1.000000	1.000000	1.000000	1.000000	3.0	

	Z_Revenue	Response
count	2240.0	2240.000000
mean	11.0	0.149107
std	0.0	0.356274
min	11.0	0.000000
25%	11.0	0.000000
50%	11.0	0.000000
75%	11.0	0.000000
max	11.0	1.000000

[8 rows x 26 columns]

```
[7]: df.dtypes
```

```
[7]: ID                int64
      Year_Birth        int64
```

Education	object
Marital_Status	object
Income	float64
Kidhome	int64
Teenhome	int64
Dt_Customer	object
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
dtype:	object

```
[8]: df.columns
```

```
[8]: Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
        'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
        'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
        'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
        'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
        'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
        'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
        dtype='object')
```

```
[9]: df.rename(columns = {'Year_Birth': 'YearBirth',
        'Marital_Status': 'MaritalStatus',
        'Dt_Customer': 'DtCustomer',
        'Z_CostContact': 'ZCostContact',
        'Z_Revenue': 'ZRevenue',
        'Kidhome': 'KidHome',
        'Teenhome': 'TeenHome'}, inplace = True)
```

```
[10]: df.head()
```

```
[10]:      ID  YearBirth  Education MaritalStatus  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

      DtCustomer  Recency  MntWines  ...  NumWebVisitsMonth  AcceptedCmp3  \
0  2012-09-04      58      635  ...              7         0
1  2014-03-08      38       11  ...              5         0
2  2013-08-21      26      426  ...              4         0
3  2014-02-10      26       11  ...              6         0
4  2014-01-19      94      173  ...              5         0

      AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0              0              0              0              0         0
1              0              0              0              0         0
2              0              0              0              0         0
3              0              0              0              0         0
4              0              0              0              0         0

      ZCostContact  ZRevenue  Response
0              3          11         1
1              3          11         0
2              3          11         0
3              3          11         0
4              3          11         0
```

```
[5 rows x 29 columns]
```

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

```
[11]: ID              0
      YearBirth      0
      Education      0
      MaritalStatus  0
      Income        24
      KidHome       0
      TeenHome      0
      DtCustomer    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
ZCostContact          0
ZRevenue              0
Response              0
dtype: int64

```

```

[12]: unique_Education = pd.unique(df.Education)

temp_data = df.copy()

columns= ['Income']

for c in unique_Education:
    Education_filter = temp_data.Education == c
    filtered_data = temp_data[Education_filter]

    for s in columns:
        mean = np.round(np.mean(filtered_data[s]),2)
        if ~np.isnan(mean):
            filtered_data[s] = filtered_data[s].fillna(mean)
            print(f"Missing value in {s} column fill with {mean} when Education:
↪{c}")
        else:
            all_data_mean = np.round(np.mean(data[s]),2)
            filtered_data[s] = filtered_data[s].fillna(all_data_mean)
            print(f"Missing value in {s} column fill with {all_data_mean}")

    temp_data[Education_filter] = filtered_data

```

```

Missing value in Income column fill with 52720.37 when Education:Graduation
Missing value in Income column fill with 56145.31 when Education:PhD
Missing value in Income column fill with 52917.53 when Education:Master
Missing value in Income column fill with 20306.26 when Education:Basic
Missing value in Income column fill with 47633.19 when Education:2n Cycle

```

C:\Users\Vinosh\AppData\Local\Temp\ipykernel_18304\2230681947.py:14:

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/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
filtered_data[s] = filtered_data[s].fillna(mean)
```

```
[13]: df = temp_data.copy()
```

```
[14]: df.isnull().sum()
```

```
[14]: ID                0
      YearBirth         0
      Education         0
      MaritalStatus     0
      Income            0
      KidHome           0
      TeenHome          0
      DtCustomer        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
      ZCostContact      0
      ZRevenue          0
      Response          0
      dtype: int64
```

```
[15]: df.duplicated().sum()
```

```
[15]: 0
```

```
[16]: from datetime import datetime as dt
```

```
[17]: df['DtCustomer'] = pd.to_datetime(df['DtCustomer'])
```

```
[18]: df['Year'] = df['DtCustomer'].dt.year
```

```
[19]: df['Month'] = df['DtCustomer'].dt.month
```

```
[20]: df.drop(['DtCustomer'],axis=1,inplace=True)
```

```
[21]: df['YearBirth'].value_counts().sort_values(ascending=True)
```

```
[21]: 1940      1
      1899      1
      1900      1
      1893      1
      1941      1
      1996      2
      1994      3
      1995      5
      1993      5
      1944      7
      1943      7
      1945      8
      1992     13
      1991     15
      1946     16
      1947     16
      1990     18
      1948     21
      1987     27
      1988     29
      1950     29
      1949     30
      1989     30
      1985     32
      1953     35
      1961     36
      1984     38
      1981     39
      1980     39
      1964     42
      1983     42
      1986     42
      1951     43
      1957     43
      1962     44
      1967     44
      1963     45
```



```

1982    45
1960    49
1955    49
1954    50
1966    50
1959    51
1968    51
1952    52
1977    52
1979    53
1958    53
1956    55
1974    69
1969    71
1965    74
1973    74
1970    77
1978    77
1972    79
1975    83
1971    87
1976    89

```

Name: YearBirth, dtype: int64

```

[22]: age = []

for i in df['YearBirth']:
    if i <= 1959:
        age.append('Elderly')
    elif i>1959 and i<=1977:
        age.append('MiddleAge')
    else:
        age.append('Young')

df['Age'] = age

```

```

[23]: df.drop(['ID', 'ZCostContact', 'ZRevenue'],axis=1,inplace=True)

```

```

[24]: df.head()

```

```

[24]:   YearBirth  Education MaritalStatus  Income  KidHome  TeenHome  Recency  \
0      1957  Graduation         Single  58138.0         0          0        58
1      1954  Graduation         Single  46344.0         1          1        38
2      1965  Graduation      Together  71613.0         0          0        26
3      1984  Graduation      Together  26646.0         1          0        26
4      1981         PhD      Married  58293.0         1          0        94

```

	MntWines	MntFruits	MntMeatProducts	...	AcceptedCmp3	AcceptedCmp4	\
0	635	88	546	...	0	0	
1	11	1	6	...	0	0	
2	426	49	127	...	0	0	
3	11	4	20	...	0	0	
4	173	43	118	...	0	0	

	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Complain	Response	Year	Month	\
0	0	0	0	0	1	2012	9	
1	0	0	0	0	0	2014	3	
2	0	0	0	0	0	2013	8	
3	0	0	0	0	0	2014	2	
4	0	0	0	0	0	2014	1	

	Age
0	Elderly
1	Elderly
2	MiddleAge
3	Young
4	Young

[5 rows x 28 columns]

```
[25]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 28 columns):
#   Column                Non-Null Count  Dtype
---  -
0   YearBirth              2240 non-null   int64
1   Education              2240 non-null   object
2   MaritalStatus          2240 non-null   object
3   Income                 2240 non-null   float64
4   KidHome                2240 non-null   int64
5   TeenHome               2240 non-null   int64
6   Recency                2240 non-null   int64
7   MntWines               2240 non-null   int64
8   MntFruits              2240 non-null   int64
9   MntMeatProducts        2240 non-null   int64
10  MntFishProducts        2240 non-null   int64
11  MntSweetProducts       2240 non-null   int64
12  MntGoldProds           2240 non-null   int64
13  NumDealsPurchases      2240 non-null   int64
14  NumWebPurchases        2240 non-null   int64
15  NumCatalogPurchases    2240 non-null   int64
16  NumStorePurchases      2240 non-null   int64
17  NumWebVisitsMonth      2240 non-null   int64
```

```

18 AcceptedCmp3      2240 non-null  int64
19 AcceptedCmp4      2240 non-null  int64
20 AcceptedCmp5      2240 non-null  int64
21 AcceptedCmp1      2240 non-null  int64
22 AcceptedCmp2      2240 non-null  int64
23 Complain          2240 non-null  int64
24 Response           2240 non-null  int64
25 Year               2240 non-null  int64
26 Month              2240 non-null  int64
27 Age                2240 non-null  object
dtypes: float64(1), int64(24), object(3)
memory usage: 490.1+ KB

```

```
[26]: data = df.copy()
```

```

[27]: df.
      ↪drop(['AcceptedCmp1','AcceptedCmp2','AcceptedCmp3','AcceptedCmp4','AcceptedCmp5','Complain'
      ↪describe()

```

```

[27]:      YearBirth      Income      KidHome      TeenHome      Recency  \
count  2240.000000    2240.000000    2240.000000    2240.000000    2240.000000
mean   1968.805804    52253.592339      0.444196      0.506250     49.109375
std     11.984069     25039.085599      0.538398      0.544538     28.962453
min    1893.000000     1730.000000      0.000000      0.000000      0.000000
25%    1959.000000     35538.750000      0.000000      0.000000     24.000000
50%    1970.000000     51609.500000      0.000000      0.000000     49.000000
75%    1977.000000     68289.750000      1.000000      1.000000     74.000000
max    1996.000000    666666.000000      2.000000      2.000000     99.000000

```

```

      MntWines      MntFruits      MntMeatProducts      MntFishProducts  \
count  2240.000000    2240.000000      2240.000000      2240.000000
mean    303.935714      26.302232      166.950000      37.525446
std     336.597393      39.773434      225.715373      54.628979
min       0.000000      0.000000       0.000000      0.000000
25%      23.750000       1.000000      16.000000       3.000000
50%     173.500000       8.000000      67.000000      12.000000
75%     504.250000      33.000000     232.000000     50.000000
max    1493.000000     199.000000    1725.000000     259.000000

```

```

      MntSweetProducts      MntGoldProds      NumDealsPurchases      NumWebPurchases  \
count      2240.000000      2240.000000      2240.000000      2240.000000
mean         27.062946         44.021875         2.325000         4.084821
std          41.280498         52.167439         1.932238         2.778714
min           0.000000           0.000000           0.000000           0.000000
25%           1.000000           9.000000           1.000000           2.000000
50%           8.000000          24.000000           2.000000           4.000000
75%          33.000000          56.000000           3.000000           6.000000

```

max	263.000000	362.000000	15.000000	27.000000
-----	------------	------------	-----------	-----------

	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
count	2240.000000	2240.000000	2240.000000
mean	2.662054	5.790179	5.316518
std	2.923101	3.250958	2.426645
min	0.000000	0.000000	0.000000
25%	0.000000	3.000000	3.000000
50%	2.000000	5.000000	6.000000
75%	4.000000	8.000000	7.000000
max	28.000000	13.000000	20.000000

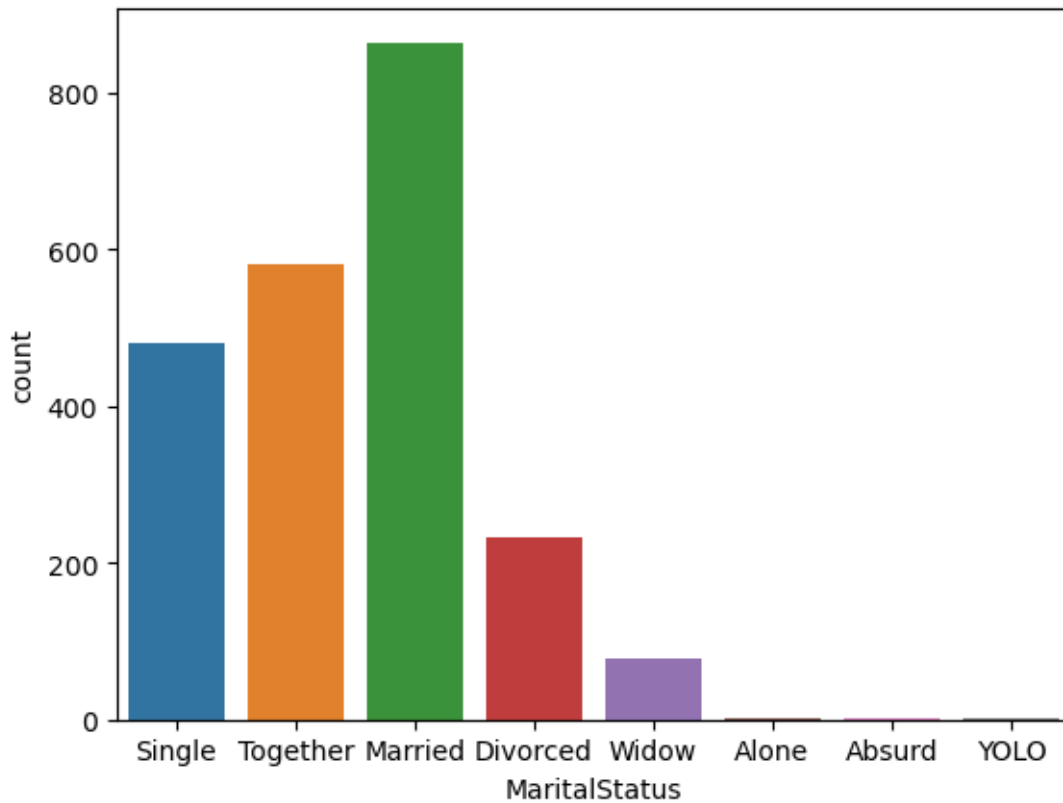
```
[28]: df[['Education', 'MaritalStatus']].describe()
```

```
[28]:
```

	Education	MaritalStatus
count	2240	2240
unique	5	8
top	Graduation	Married
freq	1127	864

```
[29]: sns.countplot(df['MaritalStatus'])
plt.show()
```

C:\Users\Vinosh\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

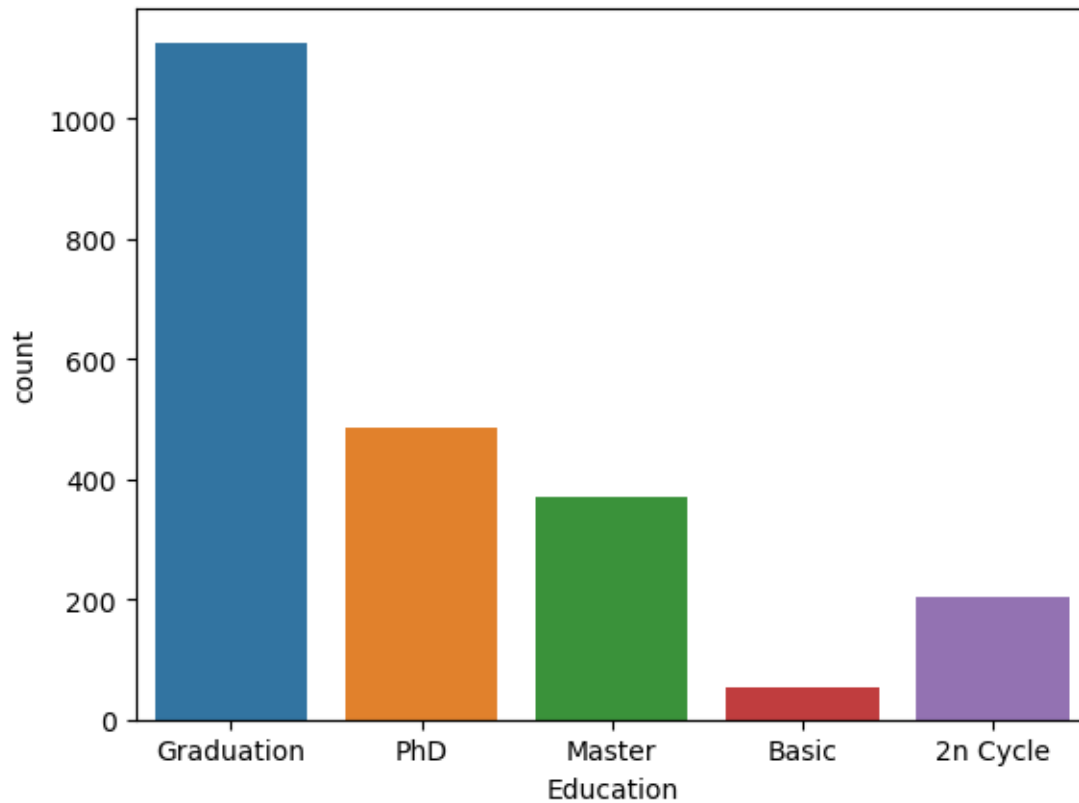


```
[30]: data['MaritalStatus'].value_counts()
```

```
[30]: Married      864
      Together    580
      Single      480
      Divorced    232
      Widow       77
      Alone        3
      Absurd       2
      YOLO         2
      Name: MaritalStatus, dtype: int64
```

```
[31]: sns.countplot(df['Education'])
      plt.show()
```

```
C:\Users\Vinosh\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



```
[32]: df['Education'].value_counts()
```

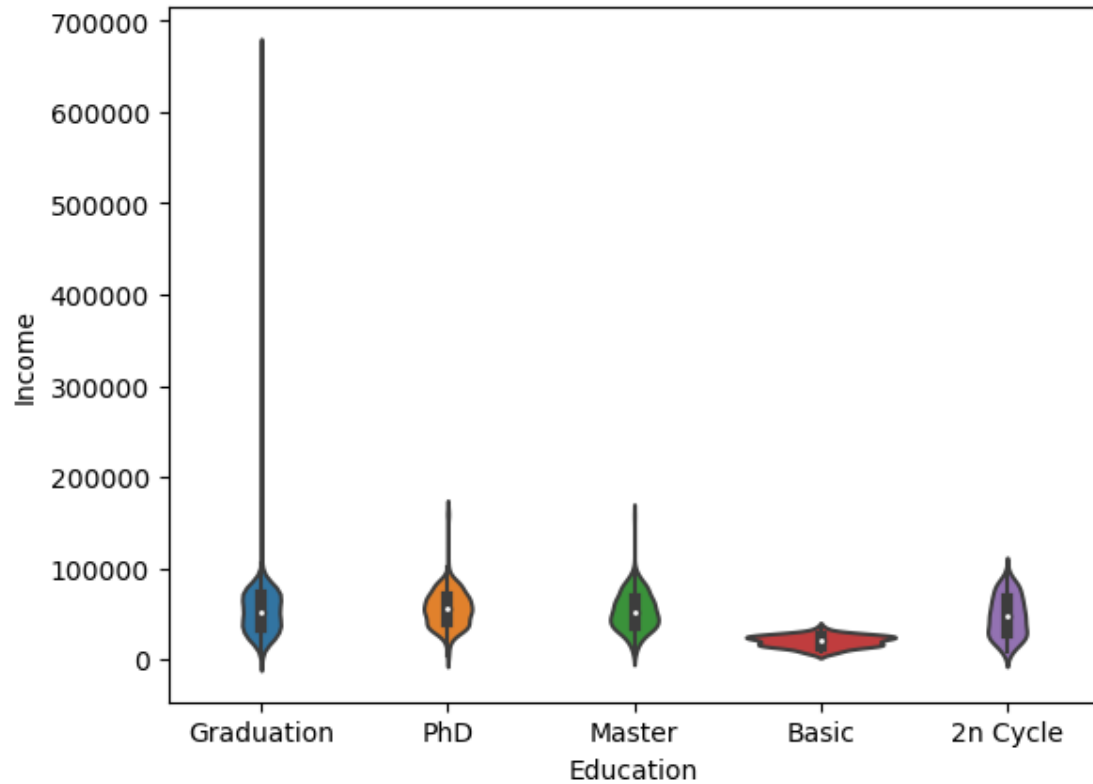
```
[32]: Graduation    1127
      PhD           486
      Master        370
      2n Cycle       203
      Basic          54
      Name: Education, dtype: int64
```

```
[33]: table = df[["Education", 'MaritalStatus']].value_counts().groupby(["Education",
    ↳ "MaritalStatus"]).sum().reset_index().
    ↳ sort_values(by=['Education', 'MaritalStatus'], ascending=False)
      table = (table.pivot(index='Education', columns='MaritalStatus', values=0))
      table
```

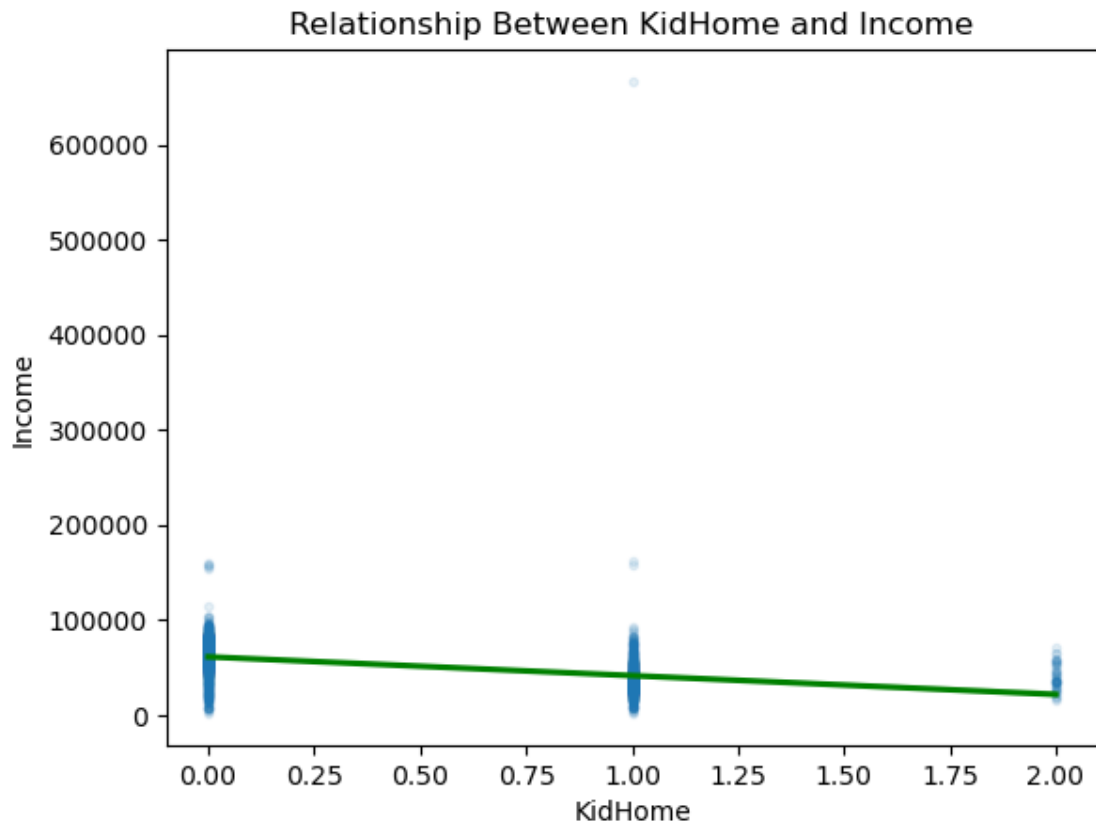
```
[33]: MaritalStatus  Absurd  Alone  Divorced  Married  Single  Together  Widow  YOLO
      Education
      2n Cycle      NaN    NaN      23.0     81.0     37.0      57.0      5.0    NaN
      Basic         NaN    NaN       1.0     20.0     18.0      14.0      1.0    NaN
      Graduation     1.0     1.0    119.0    433.0    252.0    286.0     35.0    NaN
      Master         1.0     1.0     37.0    138.0     75.0    106.0     12.0    NaN
```

PhD NaN 1.0 52.0 192.0 98.0 117.0 24.0 2.0

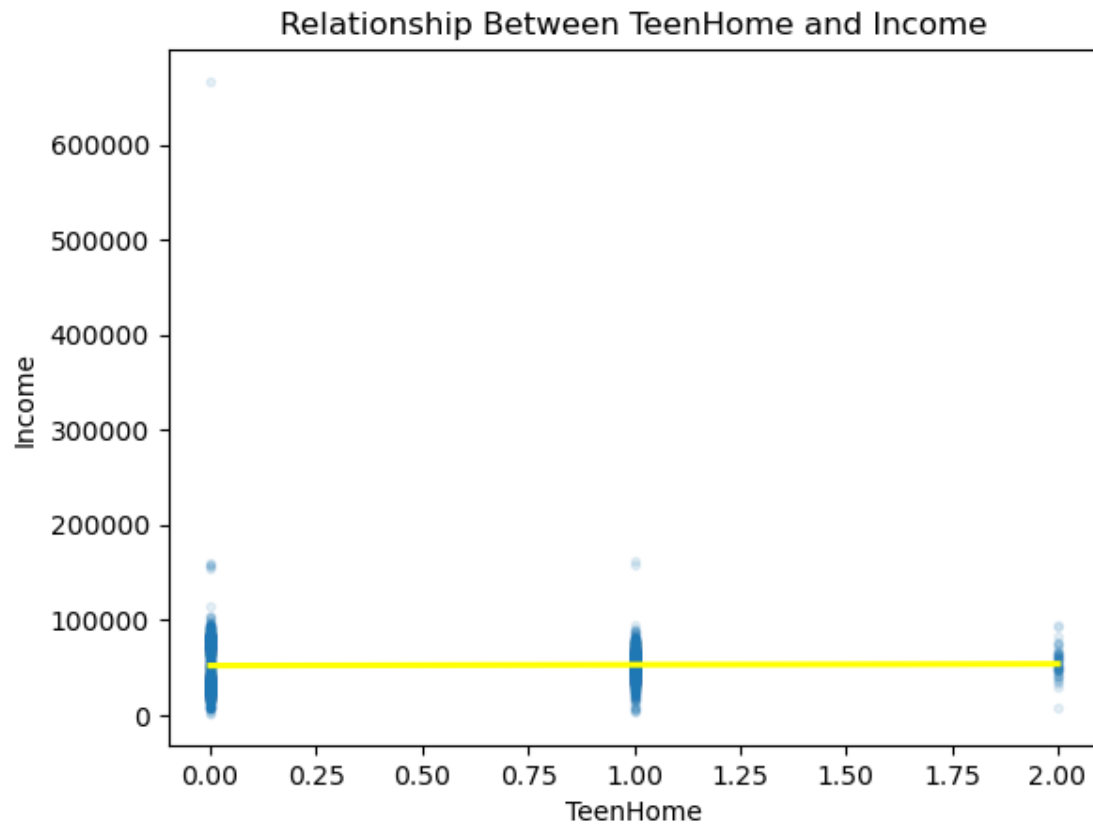
```
[34]: sns.violinplot(x=df['Education'],y=df['Income'])  
plt.show()
```



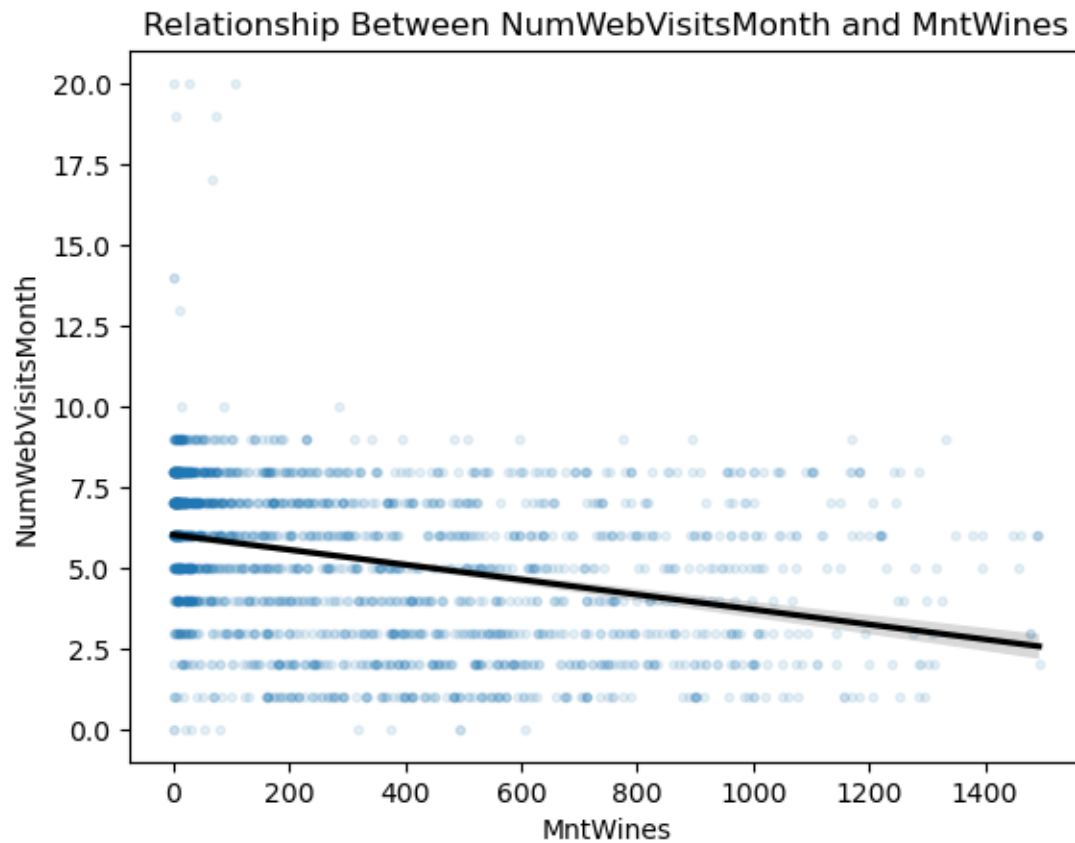
```
[35]: sns.regplot(x=df['KidHome'],y=df['Income'],marker='.',scatter_kws={'alpha':0.  
↵1},line_kws={'color':'green'})  
plt.title('Relationship Between KidHome and Income')  
plt.show()
```



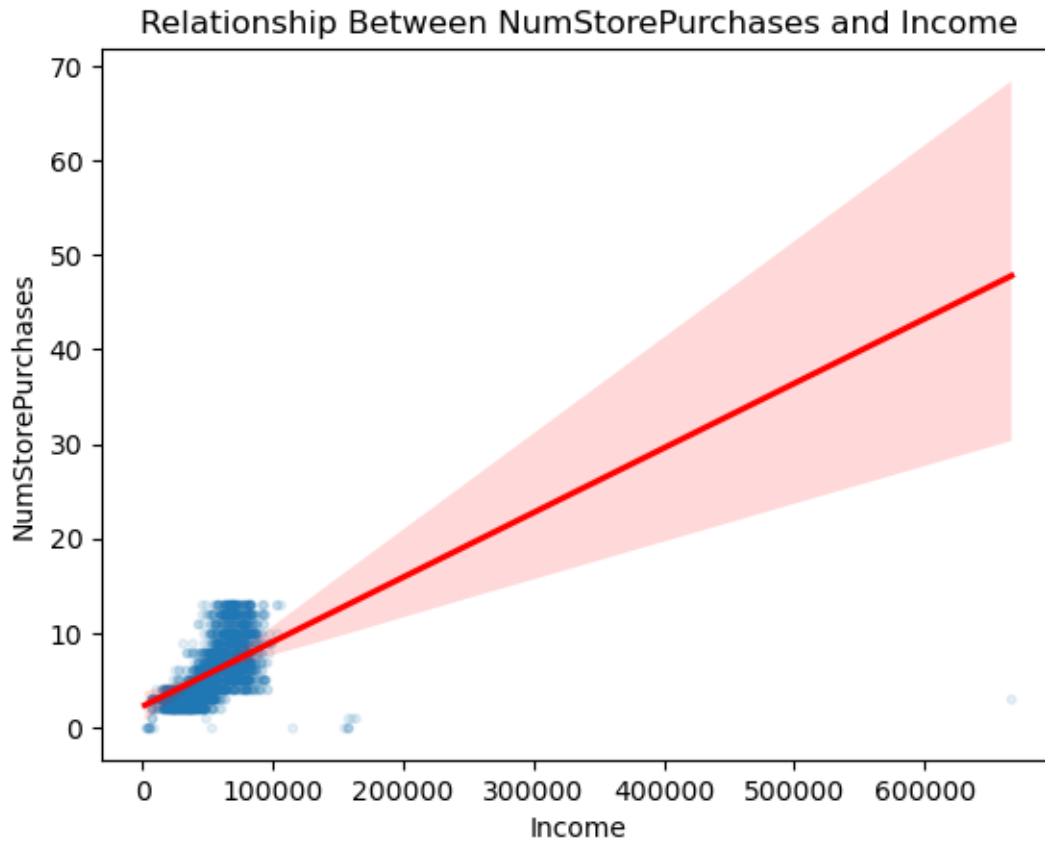
```
[36]: sns.regplot(x=df['TeenHome'],y=df['Income'],marker='.',scatter_kws={'alpha':0.  
↪1},line_kws={'color':'yellow'})  
plt.title('Relationship Between TeenHome and Income')  
plt.show()
```

```
[37]: sns.regplot(x=df['MntWines'],y=df['NumWebVisitsMonth'],marker='.\n↪',scatter_kws={'alpha':0.1},line_kws={'color':'black'})\nplt.title('Relationship Between NumWebVisitsMonth and MntWines')\nplt.show()
```



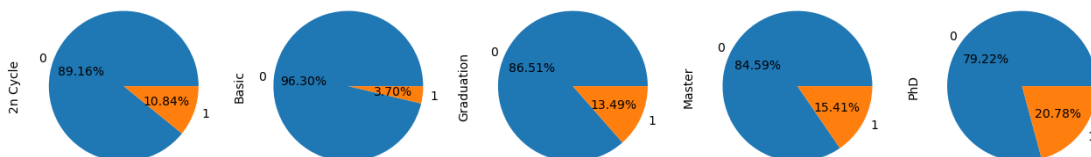
```
[38]: sns.regplot(x=df['Income'],y=df['NumStorePurchases'],marker='.',
↳,scatter_kws={'alpha':0.1},line_kws={'color':'red'})
plt.title('Relationship Between NumStorePurchases and Income')
plt.show()
```



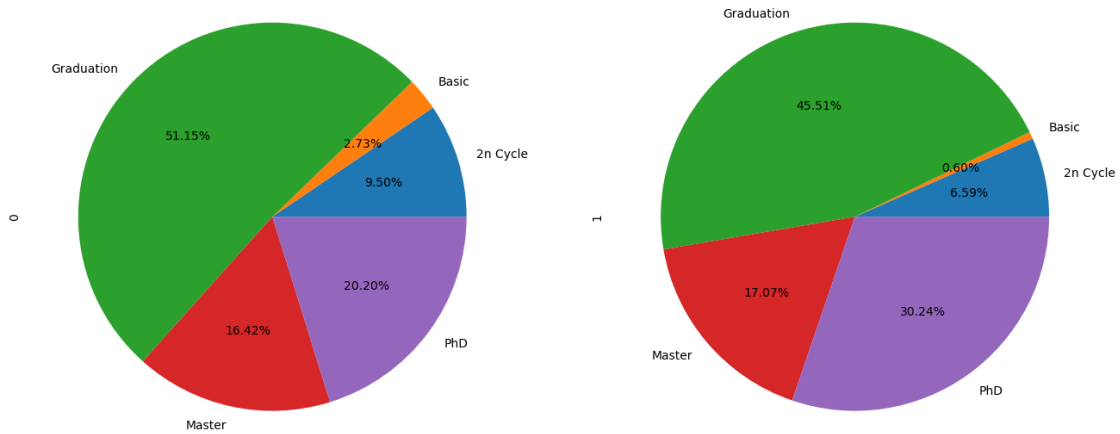
```
[39]: pd.crosstab(index=df['Education'], columns=df['Response'])
```

```
[39]: Response    0    1
Education
2n Cycle    181   22
Basic        52    2
Graduation  975  152
Master      313   57
PhD         385  101
```

```
[40]: pd.crosstab(index=df['Response'], columns=df['Education']).
      plot(kind='pie', figsize=(16,8), subplots=True, autopct='%1.2f%%', legend=False)
plt.show()
```



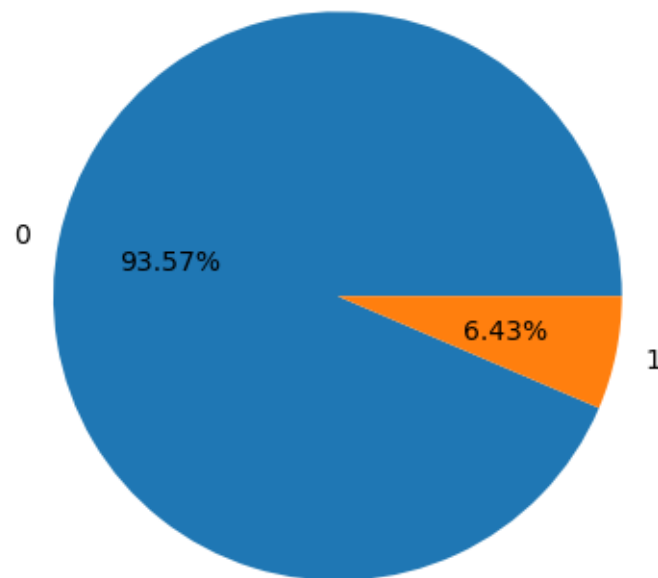
```
[41]: pd.crosstab(index=df['Education'], columns=df['Response']).
      ↪ plot(kind='pie', figsize=(16,8), subplots=True, autopct='%1.2f%%', legend=False)
      plt.show()
```



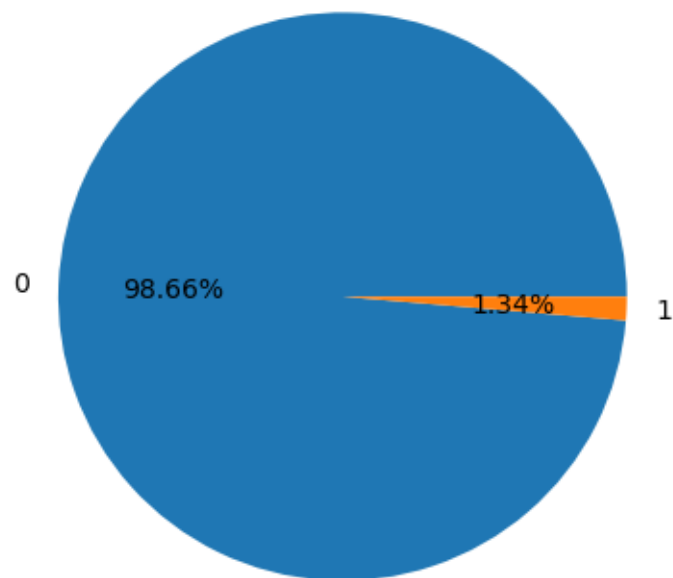
```
[42]: campaigns = _
      ↪ ['AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5']
```

```
[43]: for i in campaigns:
      accept_rate = (df.groupby(i).size() / df[i].count()) * 100
      plt.title(f"Accept rates for {i}")
      plt.pie(accept_rate, labels=df[i].unique(), autopct='%1.2f%%')
      plt.show()
```

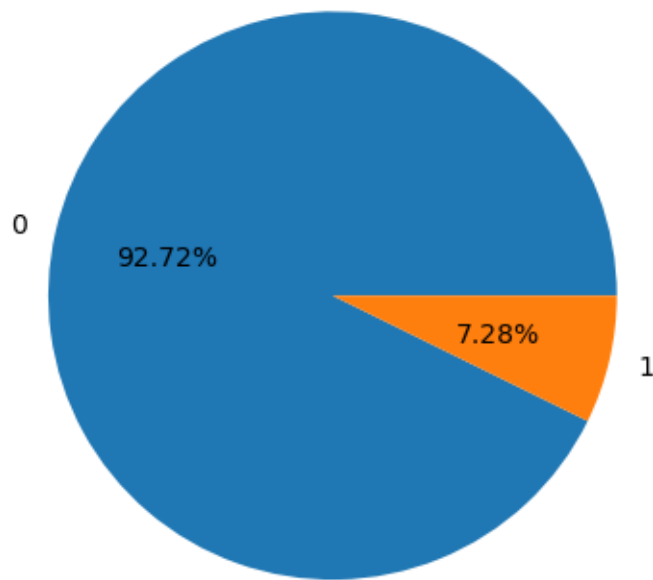
Accept rates for AcceptedCmp1



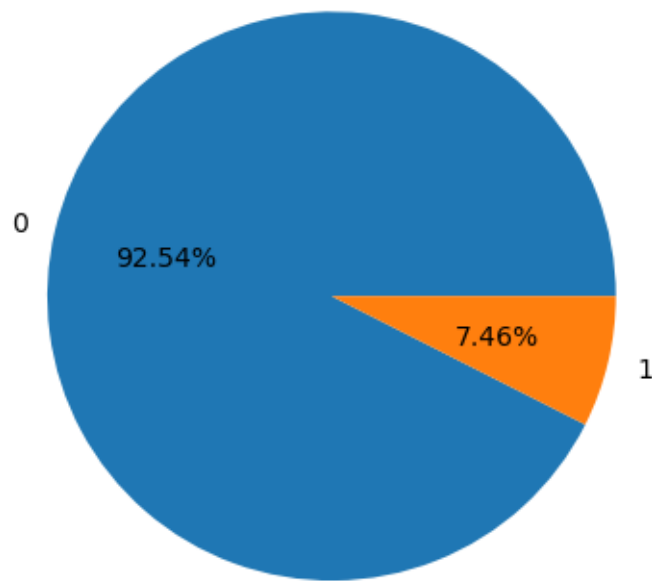
Accept rates for AcceptedCmp2



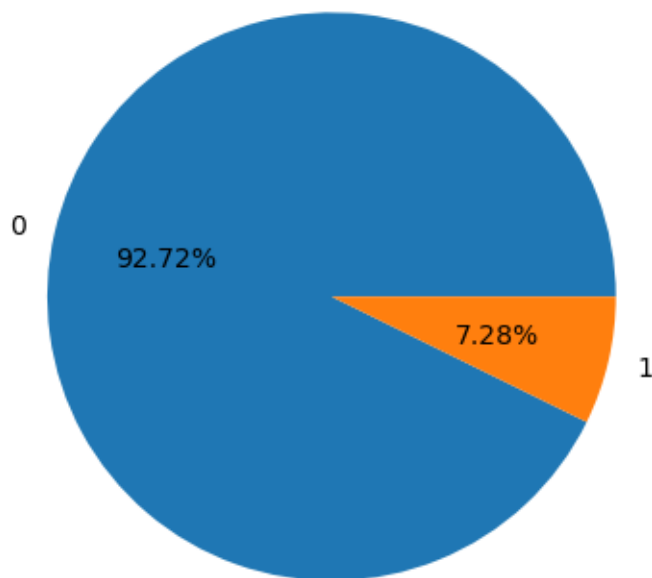
Accept rates for AcceptedCmp3



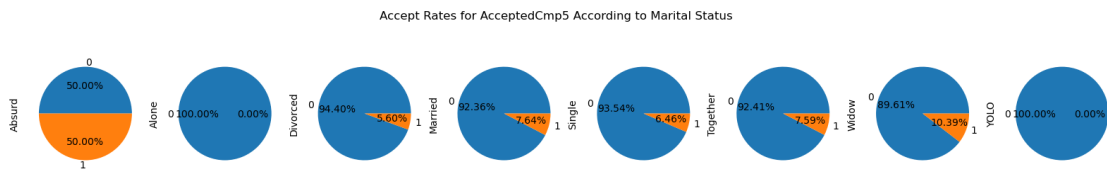
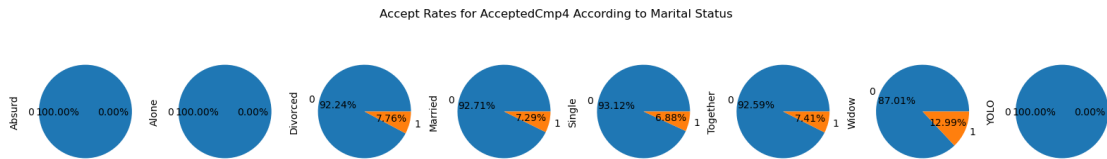
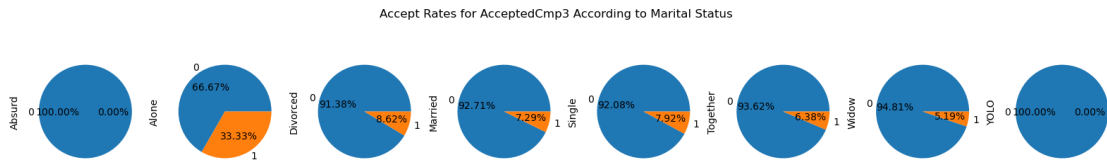
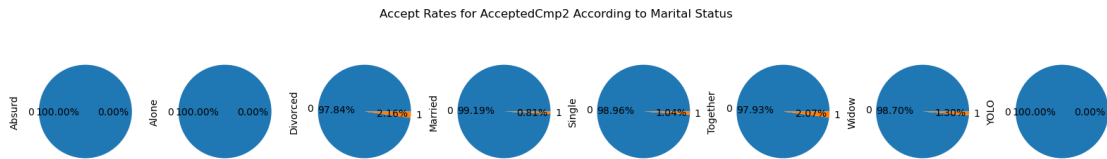
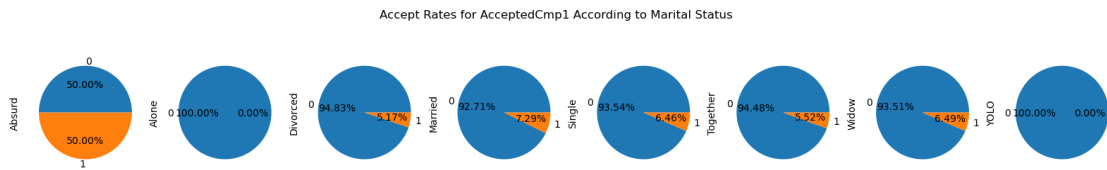
Accept rates for AcceptedCmp4



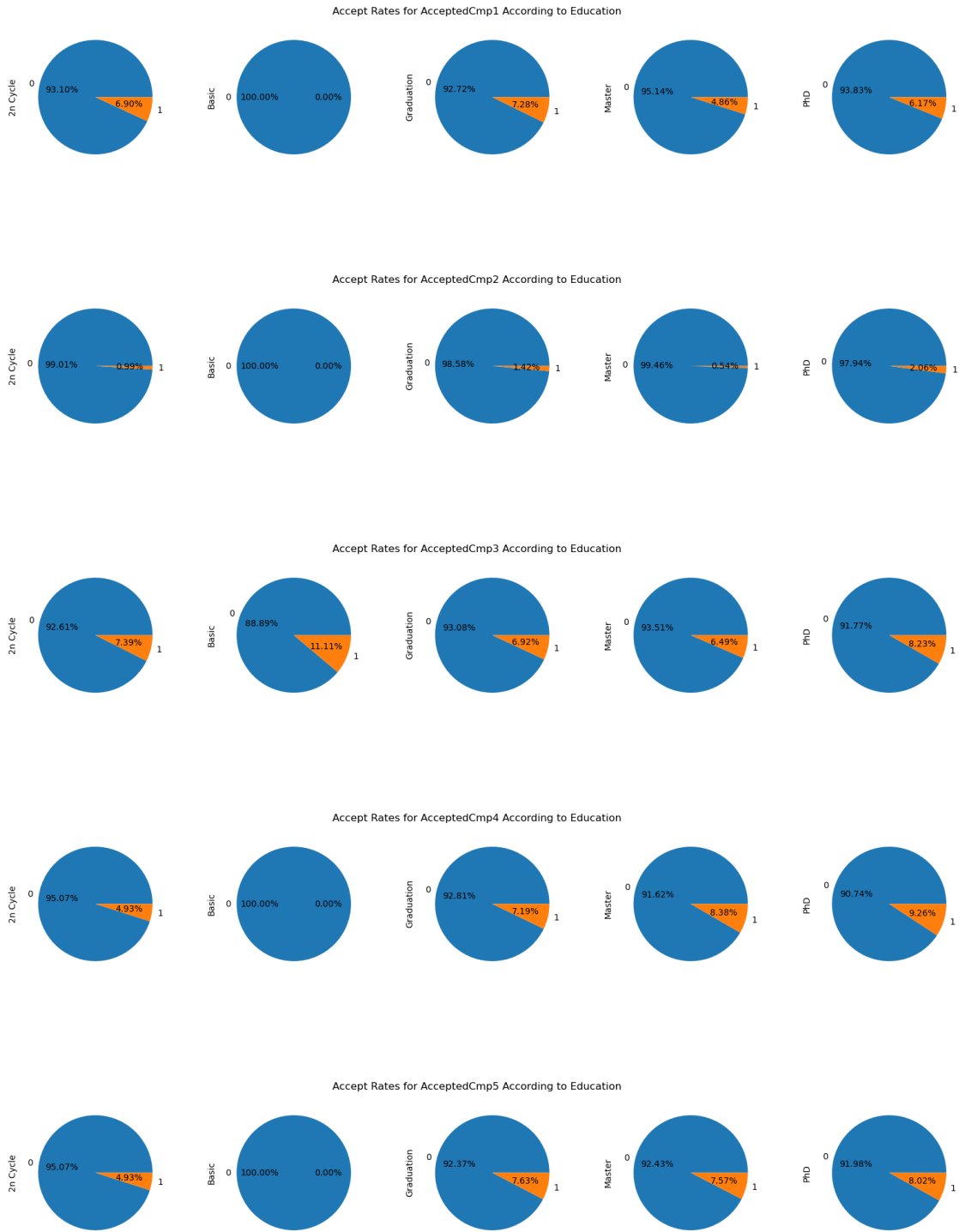
Accept rates for AcceptedCmp5



```
[44]: for i in campaigns:
    pd.crosstab(index=df[i],columns=df['MaritalStatus']).
    plot(kind='pie',figsize=(20,3),subplots=True,autopct='%1.
    2f%%',legend=False,title=f'Accept Rates for {i} According to Marital Status')
    plt.show()
```

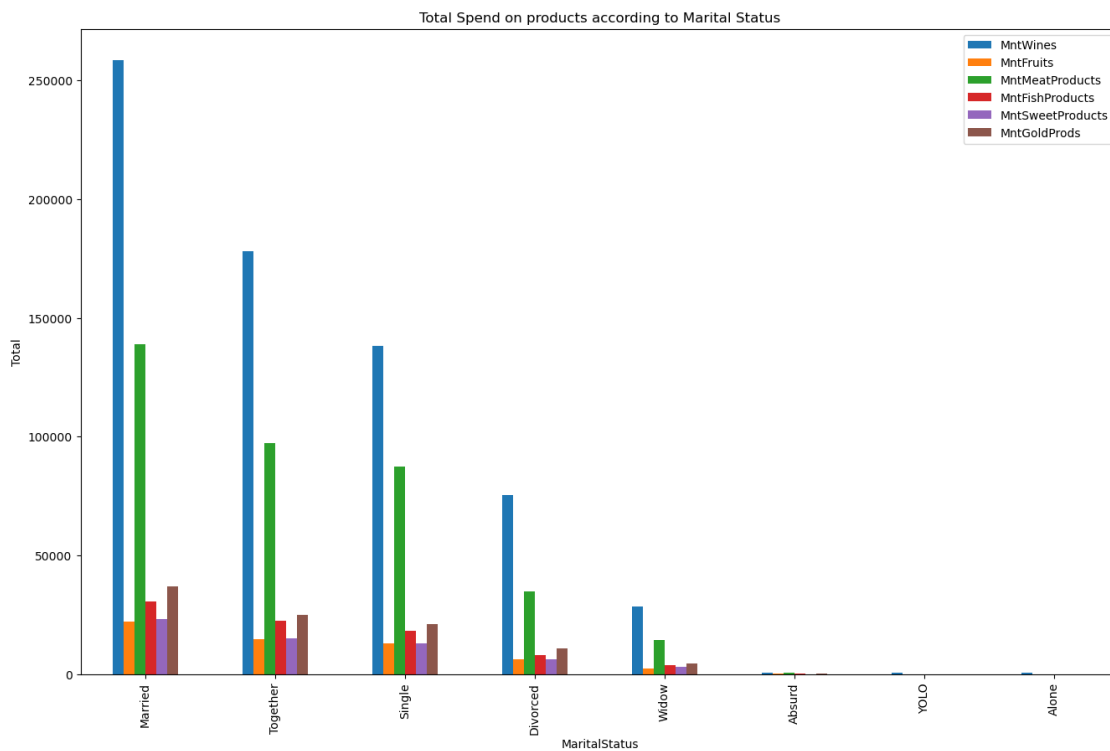



```
[45]: for i in campaigns:
        pd.crosstab(index=df[i], columns=df['Education']).
        plot(kind='pie', figsize=(20, 3), subplots=True, autopct='%1.
        2f%', legend=False, title=f'Accept Rates for {i} According to Education')
        plt.show()
```



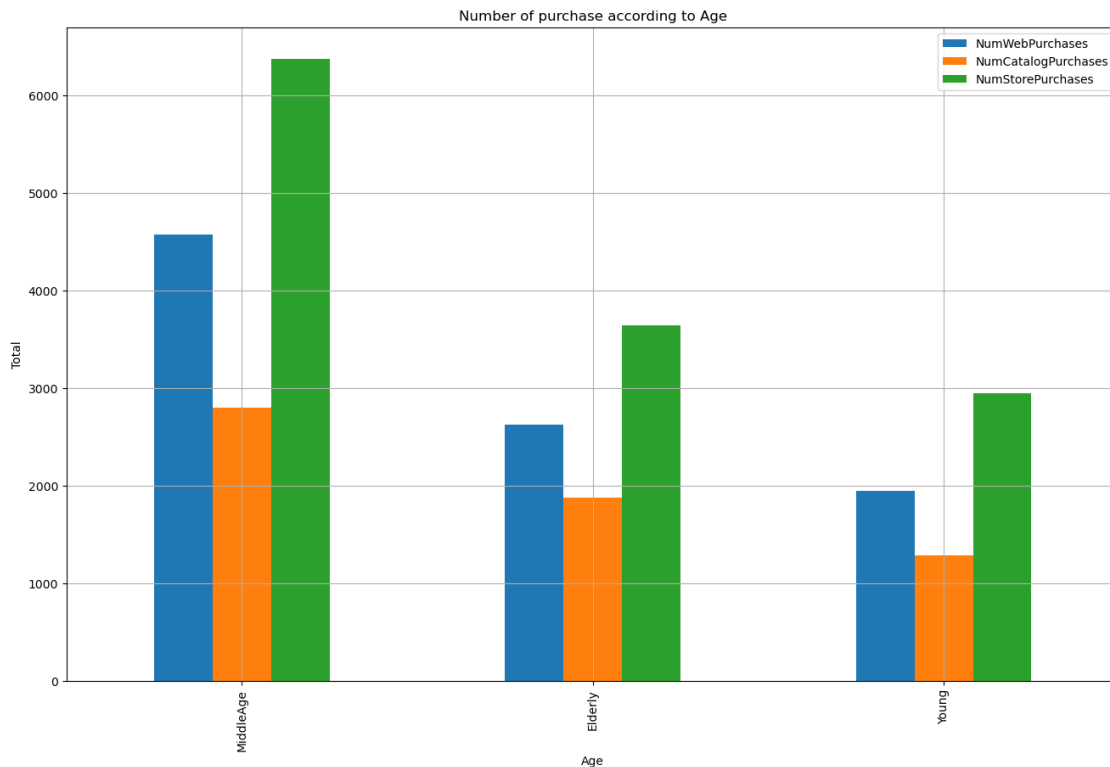
```
[46]: table1 =
↳ df[['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldP
↳ groupby(df['MaritalStatus']).sum().reset_index().
↳ sort_values(by=['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProduct
```

```
[47]: table1.
↳ plot(x='MaritalStatus', ylabel='Total', kind='bar', legend=True, figsize=(16,10), title='Total_
↳ Spend on products according to Marital Status')
plt.show()
```



```
[48]: table2 = df[['NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases']].
↳ groupby(df['Age']).sum().reset_index().
↳ sort_values(by=['NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases'], ascending=False
```

```
[49]: table2.
↳ plot(x='Age', ylabel='Total', kind='bar', legend=True, grid=True, figsize=(16,10), title='Number_
↳ of purchase according to Age')
plt.show()
```



```
[50]: table2
```

```
[50]:      Age  NumWebPurchases  NumCatalogPurchases  NumStorePurchases
1  MiddleAge             4580                2802             6382
0   Elderly             2623                1877             3642
2    Young             1947                1284             2946
```

```
[51]: target_row_index = [0,1,2]
```

```
[52]: plt.figure(figsize=plt.figaspect(1))
```

```
[52]: <Figure size 480x480 with 0 Axes>
```

```
<Figure size 480x480 with 0 Axes>
```

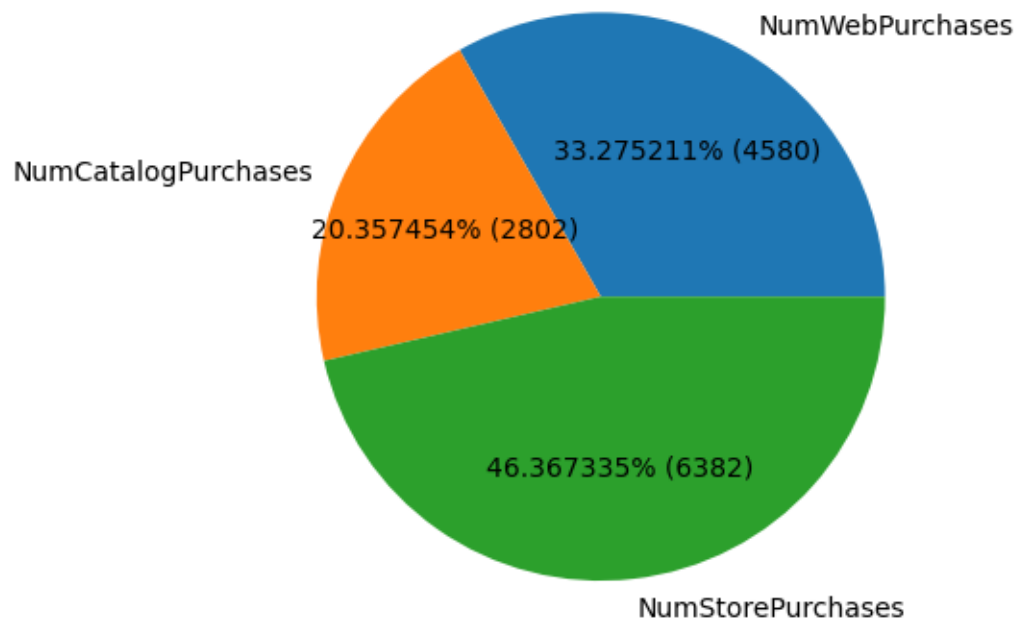
```
[53]: values = table2.iloc[target_row_index,1:]
labels = table2.columns[1:]
```

```
[54]: def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        val = int(round(pct*total/100.0))
        return f'{pct:2f}% ({val:d})'
```

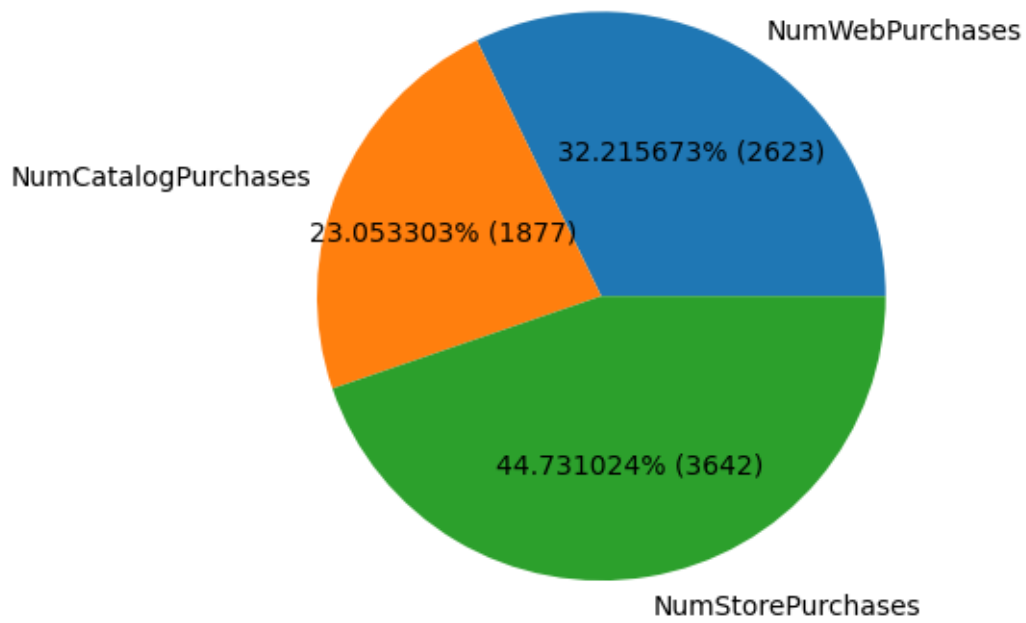
```
return my_autopct
```

```
[55]: for i in target_row_index:
      values = table2.iloc[i,1:]
      labels = table2.columns[1:]
      plt.title(f'Purchase Types Rates for {df.loc[i].Age}')
      plt.pie(values, labels = labels, autopct = make_autopct(values))
      plt.show()
```

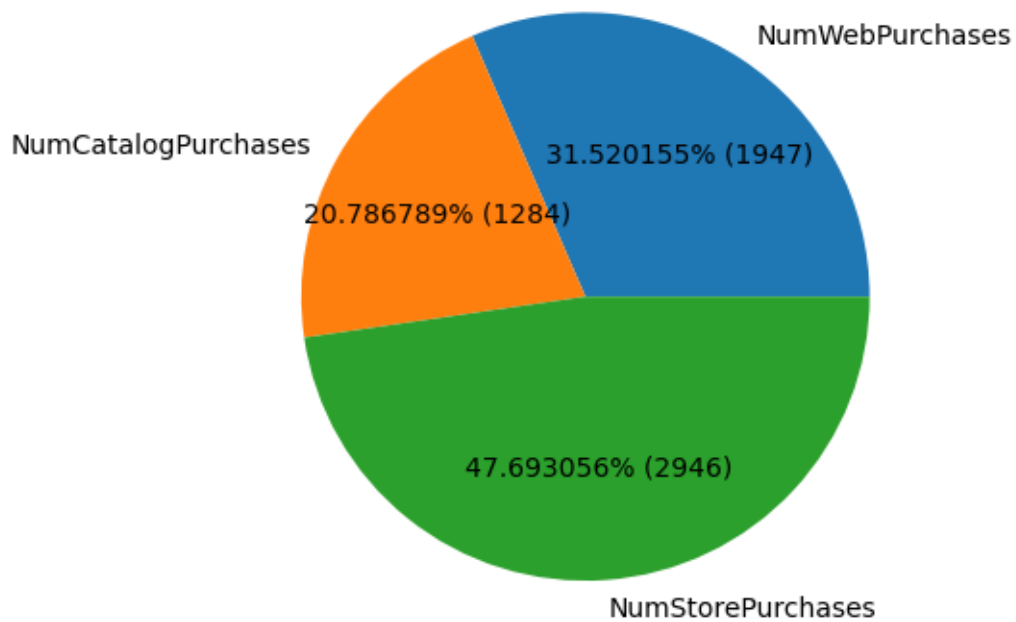
Purchase Types Rates for Elderly



Purchase Types Rates for Elderly

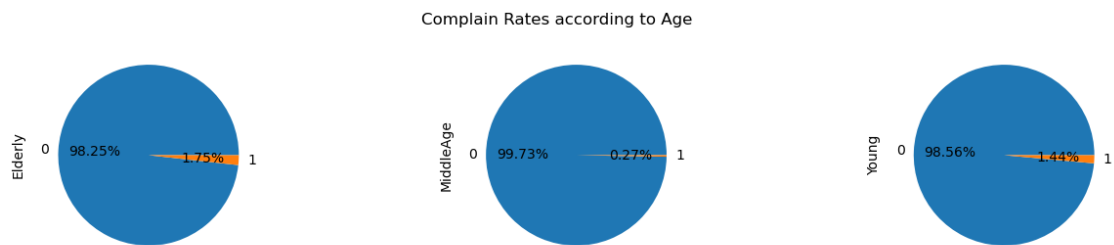


Purchase Types Rates for MiddleAge



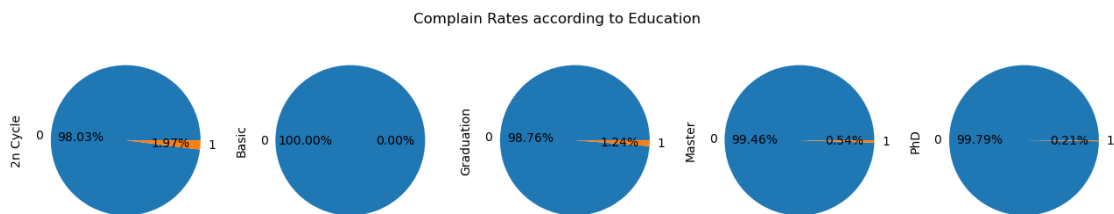
```
[56]: table3 = pd.crosstab(index=df['Complain'],columns=df['Age'])
```

```
[57]: table3.plot(x='Complain',kind='pie',figsize=(16,3),legend=False, subplots =_
↳ True, autopct = '%1.2f%%',title='Complain Rates according to Age')
plt.show()
```

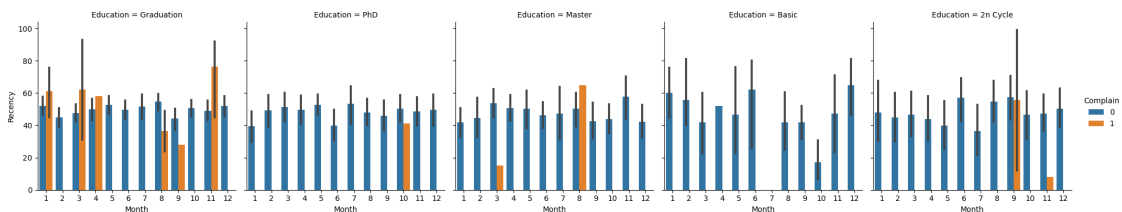


```
[58]: table4 = pd.crosstab(index=df['Complain'],columns=df['Education'])
```

```
[59]: table4.plot(x='Complain',kind='pie',figsize=(16,3),legend=False, subplots =_
↳ True, autopct = '%1.2f%%',title='Complain Rates according to Education')
plt.show()
```



```
[60]: sns.catplot(x='Month',y='Recency', hue='Complain',_
↳ col='Education',data=df,kind='bar',height=4)
plt.show()
```



1 Hypotheses Testing

```
[61]: data.dtypes
```

```
[61]: YearBirth          int64
      Education          object
      MaritalStatus      object
      Income             float64
      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
      Response           int64
      Year               int64
      Month              int64
      Age                object
      dtype: object
```

```
[62]: X =  
      ↪data[['AcceptedCmp1','AcceptedCmp2','AcceptedCmp3','AcceptedCmp4','AcceptedCmp5']]
```

```
[63]: y = data['Response']
```

```
[64]: from sklearn.model_selection import train_test_split
```

```
[65]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.  
      ↪30,random_state=50)
```

```
[66]: X_train.shape
```

```
[66]: (1568, 5)
```

```
[67]: y_train.shape
```

```
[67]: (1568,)
```

```
[68]: X_test.shape
```

```
[68]: (672, 5)
```

```
[69]: y_test.shape
```

```
[69]: (672,)
```

```
[70]: from sklearn.linear_model import LinearRegression
```

```
[71]: linreg = LinearRegression()
```

```
[72]: linreg.fit(X_train,y_train)
```

```
[72]: LinearRegression()
```

```
[73]: print(linreg.intercept_)
```

```
0.07592218362192815
```

```
[74]: coeff = pd.DataFrame(linreg.coef_,X.columns,columns=['Coefficient'])
```

```
[75]: coeff
```

```
[75]:
```

	Coefficient
AcceptedCmp1	0.247120
AcceptedCmp2	0.249020
AcceptedCmp3	0.356702
AcceptedCmp4	0.085506
AcceptedCmp5	0.275781

```
[77]: from sklearn.feature_selection import f_regression
```

```
[78]: fregression = f_regression(X_train,y_train)
```

```
[79]: fregression
```

```
[79]: (array([146.6547474 ,  42.84932877, 148.10584146,  34.72068905,  
          169.42997667]),  
      array([2.47587034e-32,  7.99056506e-11,  1.27000334e-32,  4.65051699e-09,  
            7.47052090e-37]))
```


2 Since the P-value is less than 0.05, we reject the null hypothesis and accept the alternative hypothesis

```
[81]: y_pred = linreg.predict(X_test)
```

```
[82]: y_pred
```

```
[82]: array([0.07592218, 0.07592218, 0.32304247, 0.43262404, 0.07592218,
            0.07592218, 0.07592218, 0.35170335, 0.07592218, 0.07592218,
            0.07592218, 0.40854834, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.16142806, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.43262404, 0.07592218, 0.07592218,
            0.07592218, 0.40854834, 0.07592218, 0.07592218, 0.32304247,
            0.07592218, 0.07592218, 0.4104482 , 0.07592218, 0.07592218,
            0.07592218, 0.68432951, 0.16142806, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.16142806,
            0.68432951, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.68432951, 0.07592218, 0.35170335, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.16142806,
            0.07592218, 0.07592218, 0.07592218, 0.35170335, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.35170335, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.4104482 , 0.07592218, 0.4104482 ,
            0.07592218, 0.07592218, 0.07592218, 0.40854834, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.43262404, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.43262404, 0.07592218, 0.07592218,
            0.93334965, 0.07592218, 0.07592218, 0.07592218, 0.43262404,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.07592218, 0.95552549, 0.07592218, 0.35170335,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.16142806, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.43262404, 0.07592218, 0.07592218, 0.07592218,
            0.07592218, 0.43262404, 0.07592218, 0.07592218, 0.43262404,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.59882363, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
            0.59882363, 0.07592218, 0.07592218, 0.07592218, 0.59882363,
            0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
```

0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.43262404,
0.07592218, 0.93334965, 0.68432951, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.68164418, 0.35170335, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.93334965, 0.07592218,
0.07592218, 0.07592218, 0.68432951, 0.07592218, 0.07592218,
0.43262404, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.35170335, 0.16142806, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.16142806, 0.07592218,
0.43262404, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.43262404, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.43262404, 0.59882363, 0.07592218, 0.07592218, 0.16142806,
0.07592218, 0.07592218, 0.07592218, 0.35170335, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.68164418, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.43262404, 0.07592218, 0.32304247,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.16142806, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.7084052 , 0.65756849, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.43262404, 0.4104482 , 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.35170335, 0.35170335, 0.16142806, 0.07592218, 0.43262404,
0.07592218, 0.43720922, 0.43262404, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.35170335, 0.07592218,
0.43262404, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.16142806,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.68622937,
0.07592218, 0.07592218, 0.67974432, 0.43262404, 0.07592218,
0.68622937, 0.95552549, 0.07592218, 0.43262404, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.43720922, 0.68432951, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.43262404, 0.68432951, 0.07592218,
0.35170335, 0.43262404, 0.67974432, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.16142806, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.35170335, 0.07592218,
0.67974432, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,

0.07592218, 0.16142806, 0.35170335, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.43262404, 0.43262404, 0.07592218,
0.07592218, 0.95742535, 0.95552549, 0.16142806, 0.07592218,
0.32304247, 0.16142806, 0.07592218, 0.43720922, 0.40854834,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.43262404, 0.07592218, 0.32304247, 0.07592218,
0.4104482 , 0.16142806, 0.68432951, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.43262404, 0.07592218, 0.07592218, 0.07592218, 0.43720922,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.43262404,
0.35170335, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.32304247, 0.40854834, 0.43262404, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.68622937,
0.07592218, 0.07592218, 0.07592218, 0.93334965, 0.07592218,
0.07592218, 1.20454563, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.43262404, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.32304247, 0.16142806, 0.43262404,
0.16142806, 0.07592218, 0.32304247, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.35170335, 0.07592218, 0.07592218, 0.07592218, 0.32304247,
0.07592218, 0.43262404, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.43720922, 0.07592218,
0.32304247, 0.43262404, 0.07592218, 0.07592218, 0.07592218,
0.43262404, 0.95552549, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.59882363, 0.07592218,
0.07592218, 0.7084052 , 0.07592218, 0.07592218, 0.16142806,
0.07592218, 0.16142806, 0.43720922, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.16142806,
0.07592218, 0.43262404, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.16142806, 0.07592218,
0.07592218, 0.07592218, 0.59882363, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.43262404,
0.07592218, 0.07592218, 0.07592218, 0.35170335, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.43262404, 0.07592218, 0.07592218, 0.07592218, 0.43262404,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.43720922, 0.07592218, 0.07592218, 0.07592218,
0.07592218, 0.07592218, 0.68432951, 0.07592218, 0.68432951,
0.07592218, 0.07592218, 0.07592218, 0.16142806, 0.16142806,
0.32304247, 0.07592218, 0.07592218, 0.07592218, 0.07592218,
0.35170335, 0.07592218, 0.07592218, 0.07592218, 0.35170335,

```
0.59882363, 0.07592218])
```

```
[83]: y_test
```

```
[83]: 126      1
      1393    0
      1278    0
      1227    1
      2235    0
      ..
      775     0
      382     0
      1298    1
      116     0
      227     0
      Name: Response, Length: 672, dtype: int64
```

```
[80]: from sklearn import metrics
```

```
[84]: print('RSME:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
RSME: 0.3381088854535124
```

```
[85]: print('R Squared:', metrics.r2_score(y_test, y_pred))
```

```
R Squared: 0.1587659915303209
```

2.0.1 The R squared value is close to 0, thus the data points doesn't fit properly

```
[ ]:
```