Project: Marketing Campaign Analysis

Context

Marketing Analytics broadly refers to the practice of using analytical methods and techniques to und driven decisions to optimize for ROI on conversion rates. It typically involves analyzing various metri and costs associated with various marketing channels. These can generate valuable insights that ca and achieve overall growth.

Problem Statement

Company 'All You Need' has hired you as a Data Scientist and you've been told by the Chief Marketing Officer that recent marketing campaigns have not been as effective as they were expected to be and the conversion rate is very low. Your task is to analyze the related data, understand the problem, and identify key insights and recommendations for the CMO to potentially implement.

The data set marketing_data.csv consists of 2,240 customers of All You Need company with data on:

- Campaign successes/failures
- Product preferences
- Channel performances
- Customer profiles based on the spending habits

Data Dictionary

- ID : Unique ID of each customer
- Year_Birth : Age of the customer
- Education : Customer's level of education
- Marital_Status : Customer's marital status
- Kidhome : Number of small children in customer's household
- Teenhome : Number of teenagers in customer's household
- Income : Customer's yearly household income
- Recency : Number of days since the last purchase
- MntFishProducts : The amount spent on fish products in the last 2 years
- MntMeatProducts : The amount spent on meat products in the last 2 years
- MntFruits : The amount spent on fruits products in the last 2 years

- MntSweetProducts : Amount spent on sweet products in the last 2 years
- MntWines : The amount spent on wine products in the last 2 years
- MntGoldProds : The amount spent on gold products in the last 2 years
- NumDealsPurchases : Number of purchases made with discount
- NumCatalogPurchases : Number of purchases made using catalog (buying goods to be shipped through the mail)
- NumStorePurchases : Number of purchases made directly in stores
- NumWebPurchases : Number of purchases made through the company's website
- NumWebVisitsMonth : Number of visits to company's website in the last month
- AcceptedCmp1 : 1 if customer accepted the offer in the first campaign, 0 otherwise
- AcceptedCmp2 : 1 if customer accepted the offer in the second campaign, 0 otherwise
- AcceptedCmp3 : 1 if customer accepted the offer in the third campaign, 0 otherwise
- AcceptedCmp4 : 1 if customer accepted the offer in the fourth campaign, 0 otherwise
- AcceptedCmp5 : 1 if customer accepted the offer in the fifth campaign, 0 otherwise
- AcceptedCmp6 : 1 if customer accepted the offer in the last campaign, 0 otherwise
- Complain : 1 If the customer complained in the last 2 years, 0 otherwise
- Country: Country customer belongs to

Importing libraries and overview of the dataset

In [1]:

```
# Library to supress warnings or deprecation notes
import warnings
warnings.filterwarnings('ignore')

# Libraries to help with reading and manipulating data
import numpy as np
import pandas as pd

# Libraries to help with data visualization
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [ ]:

```
# from google.colab import files
# uploaded = files.upload()
```

Load the dataset

In [3]:

```
# loading the datset

df = pd.read_csv('Marketing data.csv')
df.head()
```

Out[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	MntWin
0	1826	1970	Graduation	Divorced	84835.0	0	0	0	1
1	1	1961	Graduation	Single	57091.0	0	0	0	46

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	MntWin
2	10476	1958	Graduation	Married	67267.0	0	1	0	13
3	1386	1967	Graduation	Together	32474.0	1	1	0	
4	5371	1989	Graduation	Single	21474.0	1	0	0	

5 rows × 27 columns

Check info of the dataset

In [ ]:

```python
#Checking the info

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 27 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   Recency              2240 non-null   int64  
 8   MntWines             2240 non-null   int64  
 9   MntFruits            2240 non-null   int64  
 10  MntMeatProducts      2240 non-null   int64  
 11  MntFishProducts      2240 non-null   int64  
 12  MntSweetProducts     2240 non-null   int64  
 13  MntGoldProds         2240 non-null   int64  
 14  NumDealsPurchases    2240 non-null   int64  
 15  NumWebPurchases      2240 non-null   int64  
 16  NumCatalogPurchases  2240 non-null   int64  
 17  NumStorePurchases    2240 non-null   int64  
 18  NumWebVisitsMonth    2240 non-null   int64  
 19  AcceptedCmp1         2240 non-null   int64  
 20  AcceptedCmp2         2240 non-null   int64  
 21  AcceptedCmp3         2240 non-null   int64  
 22  AcceptedCmp4         2240 non-null   int64  
 23  AcceptedCmp5         2240 non-null   int64  
 24  AcceptedCmp6         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Country              2240 non-null   object 
dtypes: float64(1), int64(23), object(3)
memory usage: 472.6+ KB
```

Observations:

- There are a total of 27 columns and 2,240 observations in the dataset
- We can see that the Income column has less than 2,240 non-null values i.e. column has missing values. We'll explore this further

Let's check the percentage of missing values for the Income column.

In [ ]:

```
# % Null values in the Income column

(df.isnull().sum()/df.shape[0]*100)['Income']
```

Out[ ]: 1.0714285714285714

Observations:

- Income has ~1.07% missing values.

Let's create a list for numerical columns in the dataset and check the summary statistics

Question 1: Find the summary statistics for numerical columns and write your observations. (use describe function). - 4 Marks

In [9]:

```
# num_cols contain numerical varibales
num_cols=['Year_Birth','Income','Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth','Kidhome'
       'Teenhome']
```

In [11]:

```
# printing descriptive statistics of numerical columns

#Uncomment the following code and fill in the blanks
df[num_cols].describe().T
```

Out[11]:

	count	mean	std	min	25%	50%	75%
Year_Birth	2240.0	1968.805804	11.984069	1893.0	1959.00	1970.0	1977.00
Income	2216.0	52247.251354	25173.076661	1730.0	35303.00	51381.5	68522.00
Recency	2240.0	49.109375	28.962453	0.0	24.00	49.0	74.00
MntWines	2240.0	303.935714	336.597393	0.0	23.75	173.5	504.25
MntFruits	2240.0	26.302232	39.773434	0.0	1.00	8.0	33.00
MntMeatProducts	2240.0	166.950000	225.715373	0.0	16.00	67.0	232.00
MntFishProducts	2240.0	37.525446	54.628979	0.0	3.00	12.0	50.00
MntSweetProducts	2240.0	27.062946	41.280498	0.0	1.00	8.0	33.00
MntGoldProds	2240.0	44.021875	52.167439	0.0	9.00	24.0	56.00
NumDealsPurchases	2240.0	2.325000	1.932238	0.0	1.00	2.0	3.00
NumWebPurchases	2240.0	4.084821	2.778714	0.0	2.00	4.0	6.00
NumCatalogPurchases	2240.0	2.662054	2.923101	0.0	0.00	2.0	4.00
NumStorePurchases	2240.0	5.790179	3.250958	0.0	3.00	5.0	8.00
NumWebVisitsMonth	2240.0	5.316518	2.426645	0.0	3.00	6.0	7.00
Kidhome	2240.0	0.444196	0.538398	0.0	0.00	0.0	1.00

	count	mean	std	min	25%	50%	75%
Teenhome	2240.0	0.506250	0.544538	0.0	0.00	0.0	1.00

**Observations:*Income has missing values*. Customer's age and income range are wide. Minimum age is<1900s, which indicates there are errors. Customer spent most on meat compared to others in the last 2 years. Average number of purchases made from store is higher than from mail and web in the last 2 years. As of max number of purchases, catlog purchase did the best in the last 2 years. Number of small kids/tennagers in customer's household ranges from 0 to 2 in the last 2 years.

Let's create a list for categorical columns in the dataset and check the count of each category

In [ ]:

```
#cat_cols contain categorical variables
cat_cols=['Education', 'Marital_Status', 'AcceptedCmp3', 'AcceptedCmp4', 'Accept
       'AcceptedCmp2', 'AcceptedCmp6', 'Complain', 'Country']
```

In [ ]:

```
# Printing the count of each unique value in each column

for column in cat_cols:
    print(df[column].value_counts(normalize=True))
    print("-" * 40)
```

```
Graduation    0.503125
PhD           0.216964
Master        0.165179
2n Cycle      0.090625
Basic         0.024107
Name: Education, dtype: float64
----------------------------------------
Married     0.385714
Together    0.258929
Single      0.214286
Divorced    0.103571
Widow       0.034375
Alone       0.001339
Absurd      0.000893
YOLO        0.000893
Name: Marital_Status, dtype: float64
----------------------------------------
0    0.927232
1    0.072768
Name: AcceptedCmp3, dtype: float64
----------------------------------------
0    0.935714
1    0.064286
Name: AcceptedCmp4, dtype: float64
----------------------------------------
0    0.925446
1    0.074554
Name: AcceptedCmp5, dtype: float64
----------------------------------------
0    0.927232
1    0.072768
Name: AcceptedCmp1, dtype: float64
----------------------------------------
```

```
0    0.850893
1    0.149107
Name: AcceptedCmp2, dtype: float64
----------------------------------------
0    0.986607
1    0.013393
Name: AcceptedCmp6, dtype: float64
----------------------------------------
0    0.990625
1    0.009375
Name: Complain, dtype: float64
----------------------------------------
SP     0.488839
SA     0.150446
CA     0.119643
AUS    0.071429
IND    0.066071
GER    0.053571
US     0.048661
ME     0.001339
Name: Country, dtype: float64
----------------------------------------
```

Observations:

- In education, 2n cycle and Master means the same thing. We can combine these two categories.
- There are many categories in marital status. We can combine the category 'Alone' with 'Single'.
- It is not clear from the data that what do the terms 'Absurd', and 'YOLO' actually mean. We can combine these categories to make a new category - 'Others'.
- There are only 21 customers who complained in the last two years.
- The majority of the customers belong to Spain and least to Mexico.
- The most common educational status is Graduation
- The most common marital status is Married

Data Preprocessing and Exploratory Data Analysis

In this section, we will first prepare our dataset for analysis.

- Fixing the categories
- Creating new columns as the total amount spent, total purchase made, total kids at home, and total accepted campaigns
- Dealing with missing values and outliers
- Extract key insights from the data

Replacing the "2n Cycle" category with "Master" in Education and "YOLO", "Alone", and "Absurd" categories with "Single" in Marital_Status

In [44]:
```
# Replacing 2n Cycle with Master

df["Education"].replace("2n Cycle", "Master", inplace=True)
```

In [45]:
```
# Replacing YOLO, Alone, Absurd with Single

df["Marital_Status"].replace(["Alone",], "Single", inplace=True)
```

In [46]:
```
df['Marital_Status'].replace(["Absurd", "YOLO"], "Others", inplace=True)
```

We have fixed the categories in the Marital_Status. Now, let's see the distribution count in different categories for marital status.

In [ ]:
```
df.Marital_Status.value_counts()
```

Out[ ]:
```
Married     864
Together    580
Single      483
Divorced    232
Widow        77
Others        4
Name: Marital_Status, dtype: int64
```

Observation:

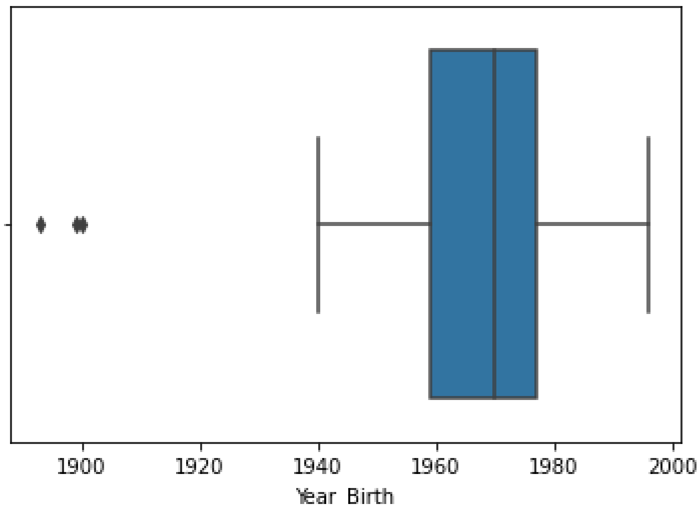
- The majority of customer belong to married category and the other category have only 4 observations.

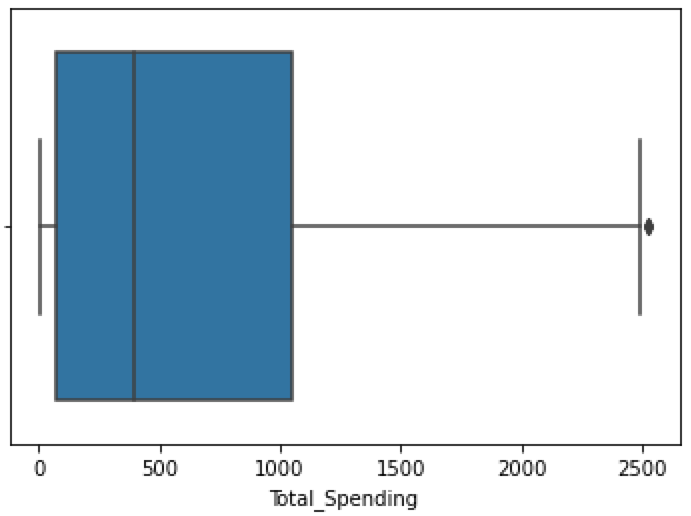
Creating new features from the existing features

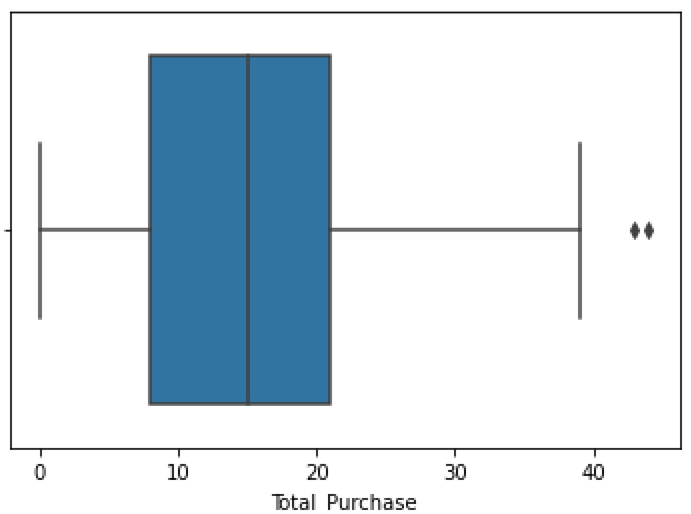
In [12]:
```
# creating new features to get overall picture of a customer, how much he/she ha
#how many children he/she has, total campaigns accepted, etc.


# total spending by a customer
spending_col = [col for col in df.columns if 'Mnt' in col]
df['Total_Spending'] = df[spending_col].sum(axis = 1)

#total purchases made by a customer
platform_col = [col for col in df.columns if 'Purchases' in col]
df['Total_Purchase'] = df[platform_col].sum(axis = 1)

#total no. of childern
df['NumberofChildren'] = df['Kidhome'] + df['Teenhome']

# Total no. of campaign accepted by a customer
campaigns_cols = [col for col in df.columns if 'Cmp' in col]
df['TotalCampaignsAcc'] = df[campaigns_cols].sum(axis=1)
```

Let's check outliers for new variables - Total_Spending, Total_Purchase. Also, let's analyze the Year_Birth column as we observed above that it had a minimum value of 1893.

In [ ]:
```
# Plotting boxplot for Year_Birth, Total_Spending, Total_Purchase

cols=['Year_Birth','Total_Spending','Total_Purchase']
for i in cols:
    sns.boxplot(x=df[i])
    plt.show()
```







Observations:

- The birth year is reported as <=1900 for some users, while the current year is 2021. it's very unlikely that the person is alive. it may be a reporting error.
- There are some outliers in total spending and total purchase.
- The observations marked as outliers are very closed to the upper whisker and some extreme points can be expected for variables like total spending. We can leave these outliers untreated.

Let's check the number of observations for which year birth is less than 1900.

In [ ]:

```
df[df['Year_Birth'] < 1900]
```

Out[ ]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	MntWi
513	11004	1893	Master	Single	60182.0	0	1	23	
827	1150	1899	PhD	Together	83532.0	0	0	36	

2 rows × 31 columns

Observation:

- There are only 2 observations for which birth year is less than 1900. We can drop these observations.

In [14]:

```
#keeping data for customers having birth year >1900

df = df[df['Year_Birth'] > 1900]
df
```

Out[14]:

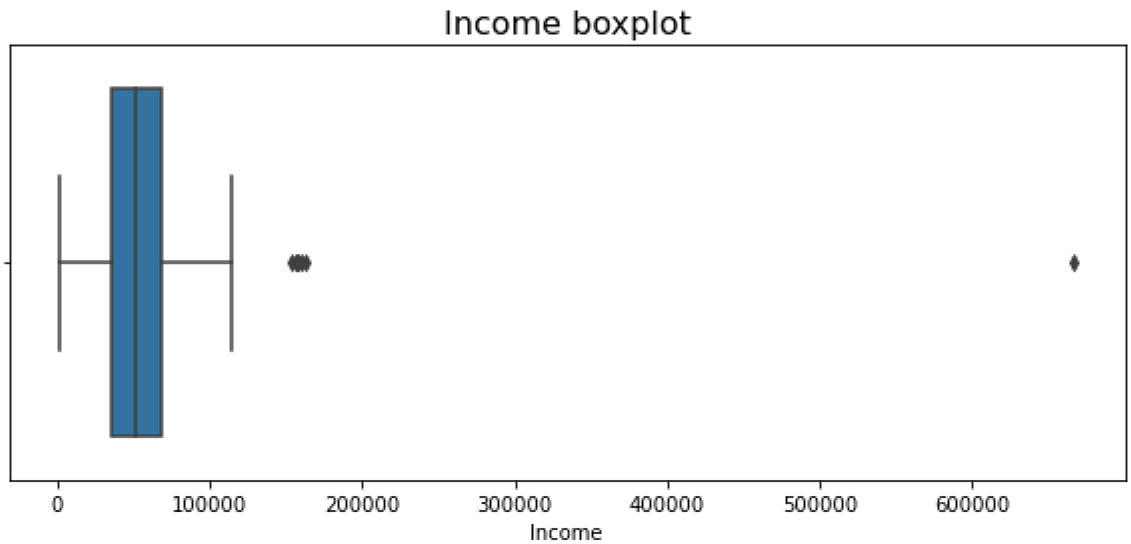
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	Mnt'
0	1826	1970	Graduation	Divorced	84835.0	0	0	0	
1	1	1961	Graduation	Single	57091.0	0	0	0	
2	10476	1958	Graduation	Married	67267.0	0	1	0	
3	1386	1967	Graduation	Together	32474.0	1	1	0	
4	5371	1989	Graduation	Single	21474.0	1	0	0	
...	...	...	...	...	...	...	...	...	
2235	10142	1976	PhD	Divorced	66476.0	0	1	99	
2236	5263	1977	2n Cycle	Married	31056.0	1	0	99	
2237	22	1976	Graduation	Divorced	46310.0	1	0	99	
2238	528	1978	Graduation	Married	65819.0	0	0	99	
2239	4070	1969	PhD	Married	94871.0	0	2	99	

2237 rows × 31 columns

Check the outliers and impute the missing values for the Income variable

In [ ]:

```
#plotting Boxplot for income

plt.figure(figsize=(10,4))
sns.boxplot(df['Income'])
plt.title('Income boxplot', size=16)
plt.show()
```



Observations:

- We can see from the boxplot that there are some outliers in the income variable.
- Let's find the value at upper whisker to check how many observations are marked as outliers.

In [ ]:

```
#Calculating the upper whisker for the Income variable

Q1 = df.quantile(q=0.25) #First quartile
Q3 = df.quantile(q=0.75) #Third quartile
IQR = Q3 - Q1            #Inter Quartile Range

upper_whisker = (Q3 + 1.5*IQR)['Income']   #Upper Whisker
print(upper_whisker)
```

```
118348.5
```

In [ ]:

```
#Checking the observations marked as outliers
df[df.Income>upper_whisker]
```

Out[ ]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	Mnt
325	4931	1977	Graduation	Together	157146.0	0	0	13	
497	1501	1982	PhD	Married	160803.0	0	0	21	
527	9432	1977	Graduation	Together	666666.0	1	0	23	

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Recency	Mnt
731	1503	1976	PhD	Together	162397.0	1	1	31	
853	5336	1971	Master	Together	157733.0	1	0	37	
1826	5555	1975	Graduation	Divorced	153924.0	0	0	81	
1925	11181	1949	PhD	Married	156924.0	0	0	85	
2204	8475	1973	PhD	Married	157243.0	0	1	98	

8 rows × 31 columns

Observations:

- We have only 8 observations with an income greater than the upper whisker.
- Only 3 observations (ID- 4931, 1501, 8475) out of 8 outliers have purchased more than 11 times in the last 2 years.
- Other 5 observations have very less amount of total spending.

Let's compare the summary statistics for these observations with observations on the other side of the upper whisker.

In [ ]:

```
#Checking the summary statistics for observations marked as outliers
df[df.Income>upper_whisker].describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%
ID	8.0	5989.250	3525.251308	1501.0	4074.00	5445.5	8714.2
Year_Birth	8.0	1972.500	10.028531	1949.0	1972.50	1975.5	1977.0
Income	8.0	221604.500	179850.404431	153924.0	157090.50	157488.0	161201.5
Kidhome	8.0	0.375	0.517549	0.0	0.00	0.0	1.0
Teenhome	8.0	0.250	0.462910	0.0	0.00	0.0	0.2
Recency	8.0	48.625	33.687376	13.0	22.50	34.0	82.0
MntWines	8.0	26.500	30.798887	1.0	1.75	14.5	43.0
MntFruits	8.0	4.500	6.524678	0.0	1.00	1.0	5.0
MntMeatProducts	8.0	621.875	846.511402	1.0	7.25	17.0	1592.0
MntFishProducts	8.0	4.250	5.650537	1.0	1.00	2.0	3.5
MntSweetProducts	8.0	1.250	0.886405	0.0	1.00	1.0	1.2
MntGoldProds	8.0	3.750	4.131759	1.0	1.00	1.5	5.0
NumDealsPurchases	8.0	4.250	6.777062	0.0	0.00	0.0	6.7
NumWebPurchases	8.0	0.500	1.069045	0.0	0.00	0.0	0.2
NumCatalogPurchases	8.0	9.875	13.484780	0.0	0.00	0.5	23.5
NumStorePurchases	8.0	0.750	1.035098	0.0	0.00	0.5	1.0
NumWebVisitsMonth	8.0	1.125	2.031010	0.0	0.00	0.5	1.0
AcceptedCmp1	8.0	0.000	0.000000	0.0	0.00	0.0	0.0

	count	mean	std	min	25%	50%	75%
AcceptedCmp2	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
AcceptedCmp3	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
AcceptedCmp4	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
AcceptedCmp5	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
AcceptedCmp6	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
Complain	8.0	0.000	0.000000	0.0	0.00	0.0	0.0
Total_Spending	8.0	662.125	848.380884	6.0	46.25	84.5	1635.2
Total_Purchase	8.0	15.375	18.220377	0.0	0.75	6.5	30.2
NumberofChildren	8.0	0.625	0.744024	0.0	0.00	0.5	1.0
TotalCampaignsAcc	8.0	0.000	0.000000	0.0	0.00	0.0	0.0

In [ ]:

```
#Checking the summary statistics for observations not marked as outliers
df[df.Income<upper_whisker].describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	
ID	2205.0	5585.439456	3247.546423	0.0	2815.0	5455.0	8418.0	1
Year_Birth	2205.0	1968.904308	11.705801	1940.0	1959.0	1970.0	1977.0	1
Income	2205.0	51622.094785	20713.063826	1730.0	35196.0	51287.0	68281.0	113
Kidhome	2205.0	0.442177	0.537132	0.0	0.0	0.0	1.0	
Teenhome	2205.0	0.506576	0.544380	0.0	0.0	0.0	1.0	
Recency	2205.0	49.009070	28.932111	0.0	24.0	49.0	74.0	
MntWines	2205.0	306.164626	337.493839	0.0	24.0	178.0	507.0	1
MntFruits	2205.0	26.403175	39.784484	0.0	2.0	8.0	33.0	
MntMeatProducts	2205.0	165.312018	217.784507	0.0	16.0	68.0	232.0	
MntFishProducts	2205.0	37.756463	54.824635	0.0	3.0	12.0	50.0	
MntSweetProducts	2205.0	27.128345	41.130468	0.0	1.0	8.0	34.0	
MntGoldProds	2205.0	44.057143	51.736211	0.0	9.0	25.0	56.0	
NumDealsPurchases	2205.0	2.318367	1.886107	0.0	1.0	2.0	3.0	
NumWebPurchases	2205.0	4.100680	2.737424	0.0	2.0	4.0	6.0	
NumCatalogPurchases	2205.0	2.645351	2.798647	0.0	0.0	2.0	4.0	
NumStorePurchases	2205.0	5.823583	3.241796	0.0	3.0	5.0	8.0	
NumWebVisitsMonth	2205.0	5.336961	2.413535	0.0	3.0	6.0	7.0	
AcceptedCmp1	2205.0	0.073923	0.261705	0.0	0.0	0.0	0.0	
AcceptedCmp2	2205.0	0.151020	0.358150	0.0	0.0	0.0	0.0	
AcceptedCmp3	2205.0	0.073016	0.260222	0.0	0.0	0.0	0.0	
AcceptedCmp4	2205.0	0.064399	0.245518	0.0	0.0	0.0	0.0	

	count	mean	std	min	25%	50%	75%
AcceptedCmp5	2205.0	0.074376	0.262442	0.0	0.0	0.0	0.0
AcceptedCmp6	2205.0	0.013605	0.115872	0.0	0.0	0.0	0.0
Complain	2205.0	0.009070	0.094827	0.0	0.0	0.0	0.0
Total_Spending	2205.0	606.821769	601.675284	5.0	69.0	397.0	1047.0
Total_Purchase	2205.0	14.887982	7.615277	0.0	8.0	15.0	21.0
NumberofChildren	2205.0	0.948753	0.749231	0.0	0.0	1.0	1.0
TotalCampaignsAcc	2205.0	0.450340	0.894075	0.0	0.0	0.0	1.0

Observations:

- None of the outliers have accepted any of the campaigns or have submitted any complaints in the last 2 years.
- We can see that customers who are outliers have lower mean expenditure per customer for all the products except meat products.
- The outliers have a higher number of catalog purchases on average and very low number of web purchases.
- We can drop the 5 observations at indices [527, 731, 853, 1826, 1925] as they would not add value to our analysis.

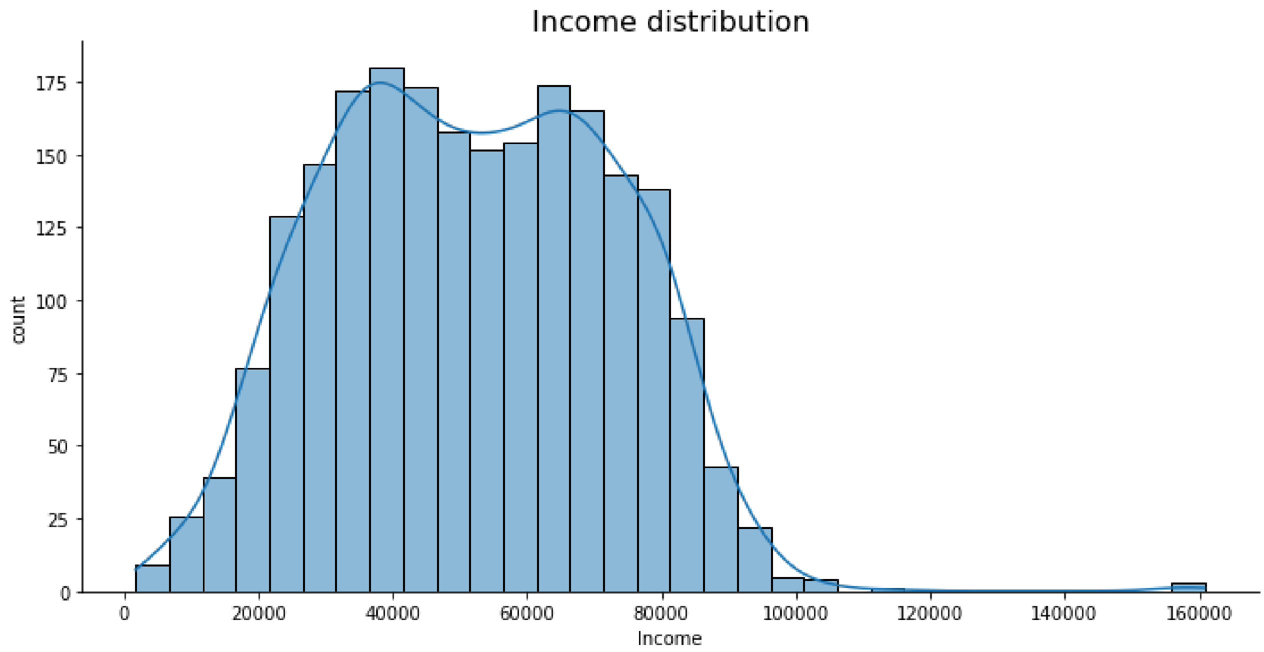
In [ ]:

```
#Dropping 5 observations at indices 527, 731, 853, 1826, 1925
df.drop(index=[527, 731, 853, 1826, 1925], inplace=True)
```

Check the distribution for Income

In [ ]:

```
#plotting displot for income

sns.displot(df['Income'], kde=True, height=5, aspect=2)
plt.title('Income distribution', size=16, )
plt.ylabel('count');
```



Observations:

- After treating outliers, the distribution for the income variable is close to normal distribution with very few extreme observations to the right.
- We will replace the missing values for the income variable with the median, and not mean, as the variable is slightly skewed to the right

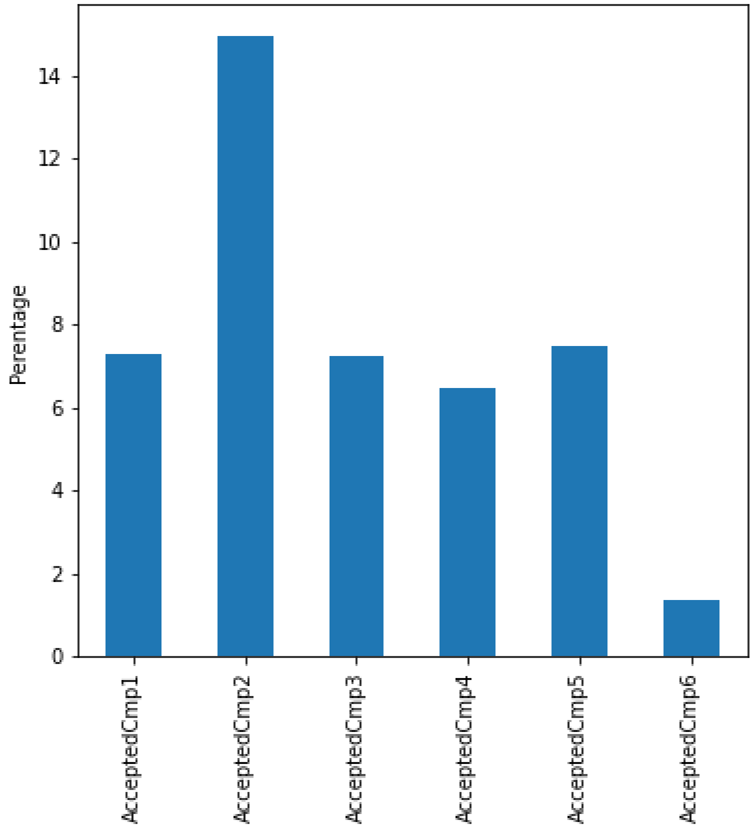
In [ ]:
```
#filling null values with median

df['Income'].fillna(df.Income.median(), inplace=True)
```

Analyzing all the campaigns

Question 2: Write your observations on acceptance rate for each campaign given in the below plot. - 4 Marks

Let's find out what is the acceptance rate for each campaign?

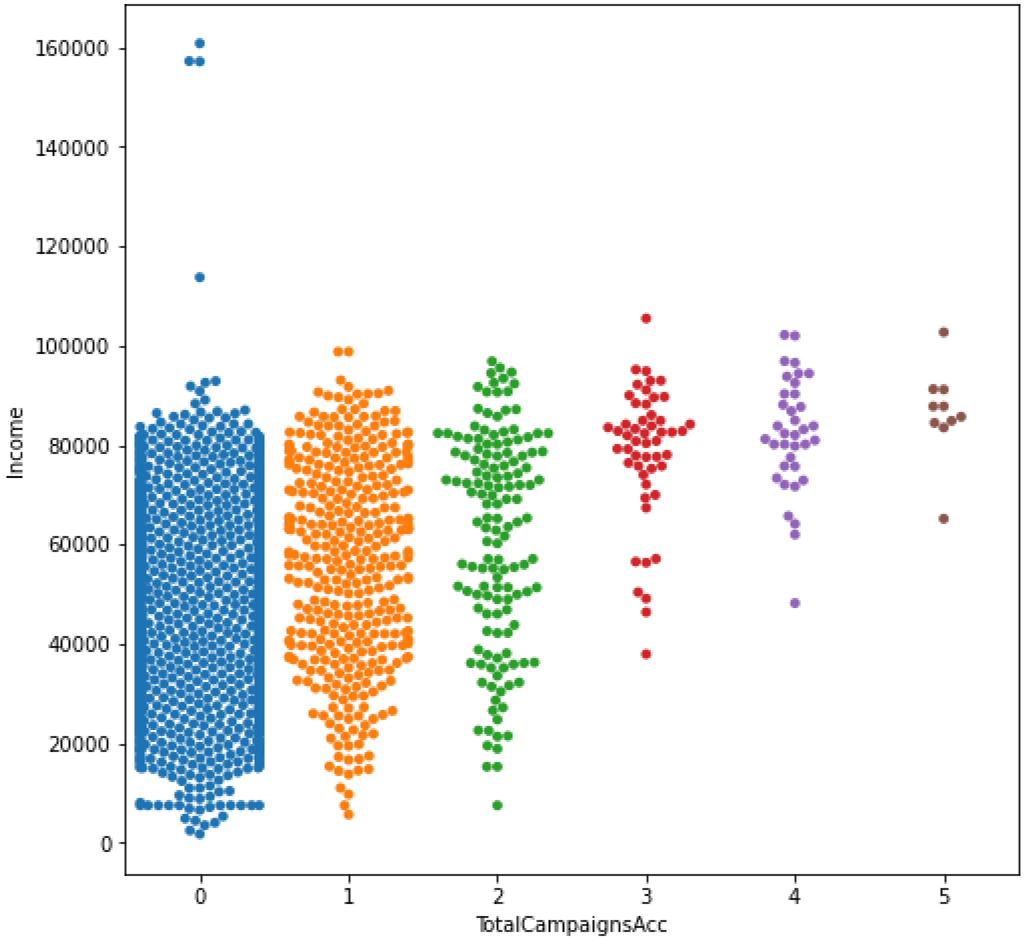
In [ ]:
```
# PLotting the % acceptance for every campaign

Camp_cols=['AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3', 'AcceptedCmp4', 'Acce

success_campaign=(df[Camp_cols].sum()/df.shape[0])*100

# plot
success_campaign.plot(kind='bar', figsize=(6,6))
plt.ylabel("Perentage")
plt.show()
```



Observations:Most customer accept the offer in the 2nd campaign. Percentage of accepting the offer in the 1st,3rd and 5th campagin does not has significant difference. Only less than 2% of customer accept the offer in the 6th campaign.

Let's analyze what kind of customer are accepting campaigns?

In [ ]:
```
plt.figure(figsize=(8,8))
sns.swarmplot(x='TotalCampaignsAcc', y='Income', data=df)
plt.show()
```



Observations:

- Higher the income higher the number of campaigns accepted.

In [ ]:
```
# Let's see the mean income of customers
df.Income.mean()
```

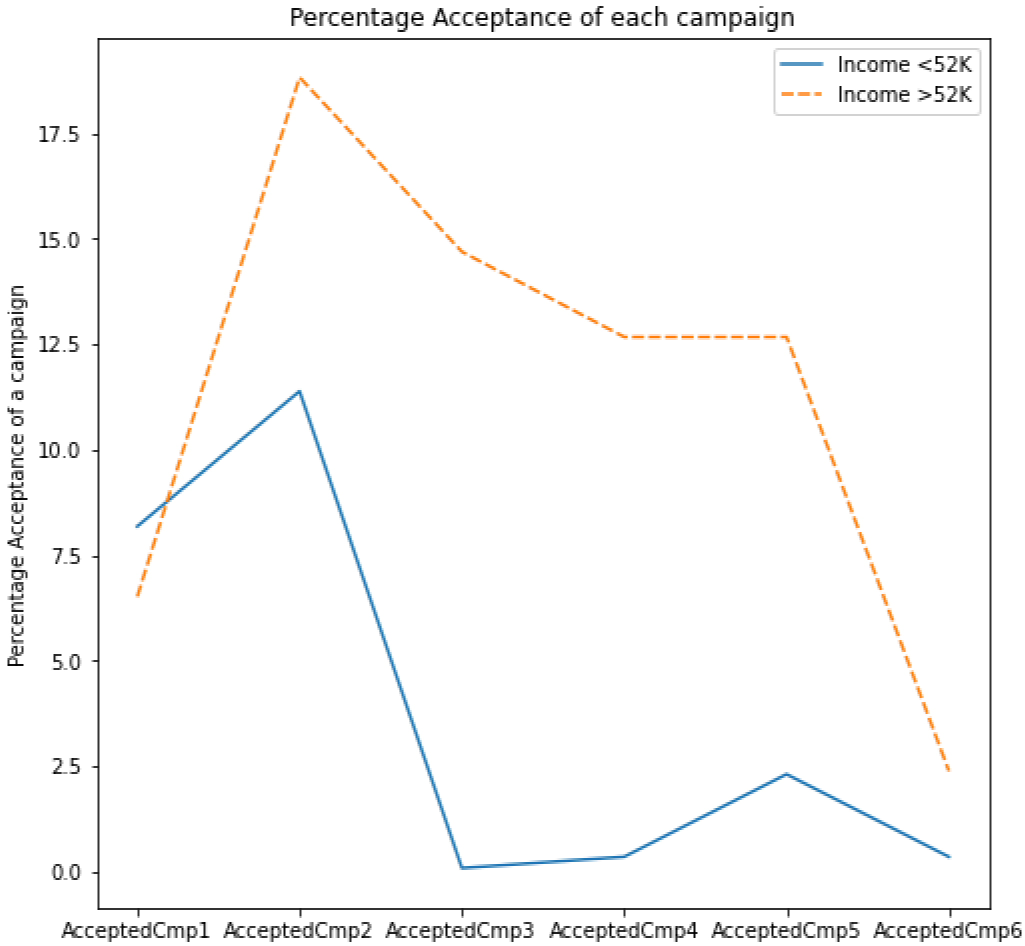
Out[ ]: 51762.59811827957

Question 3: Write your observations on acceptance rate for each campaign according to the income level. - 7 Marks

The mean income of customers is close to 52K. Let's divide the income into 2 segments of income>52k and income<52k and see the acceptance rate in each segment.

In [18]:
```
# making dataframes of customers having income <52k and >52K
df1=df[df.Income<52000]
df2=df[df.Income>52000]

Camp_cols=['AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3', 'AcceptedCmp4', 'Acce

#Calculating success rate of each campaign for both segments
success_campaign1=pd.DataFrame((df1[Camp_cols].sum()/df1.shape[0])*100, columns=

success_campaign2=pd.DataFrame((df2[Camp_cols].sum()/df2.shape[0])*100, columns=
```

```
new_df=pd.concat([success_campaign1, success_campaign2], axis=1)

# plot
plt.figure(figsize=(8,8))
sns.lineplot(data=new_df)
plt.title("Percentage Acceptance of each campaign")
plt.ylabel("Percentage Acceptance of a campaign")
plt.show()
```



Observations:_Income<52k and income>52 have the same trend on percentage acceptance of each campaign. Both income segments has the highest acceptance rate in the 2nd campaign. Income>52k has much higher acceptance rate than Income<52k of each campaign. As of income<52k,acceptance rate drops rapidly in the 3rd campaign and very low number of customer accept the offer in the 6th campaign. As of income>52k,acceptance rate drops rapidly in the 6th campaign. 5th campaign has better performance than 3rd and 4th campaign on both 2 income segments.

Let's find out who has accepted the last campaign and what could be the reason?

In [19]:
```
df[df['AcceptedCmp6']==1].shape
```

Out[19]:
```
(30, 31)
```

- There are only 30 customers who have accepted the last campaign.
- Let's check if these customers are new or they have accepted previous campaigns as well.

In [20]:
```
grouped2=df.groupby('AcceptedCmp6').mean()['TotalCampaignsAcc']
grouped2
```

Out[20]:
```
AcceptedCmp6
0    0.403715
1    3.633333
Name: TotalCampaignsAcc, dtype: float64
```

Observations:

- We know that the maximum number of campaigns any customer has accepted is 5.
- We can observe that the value for TotalCampaignsAcc is ~3.6 for customers who have accepted the last campaign.
- This implies that these 30 customers are those loyal customers who have been accepting most of the campaigns.

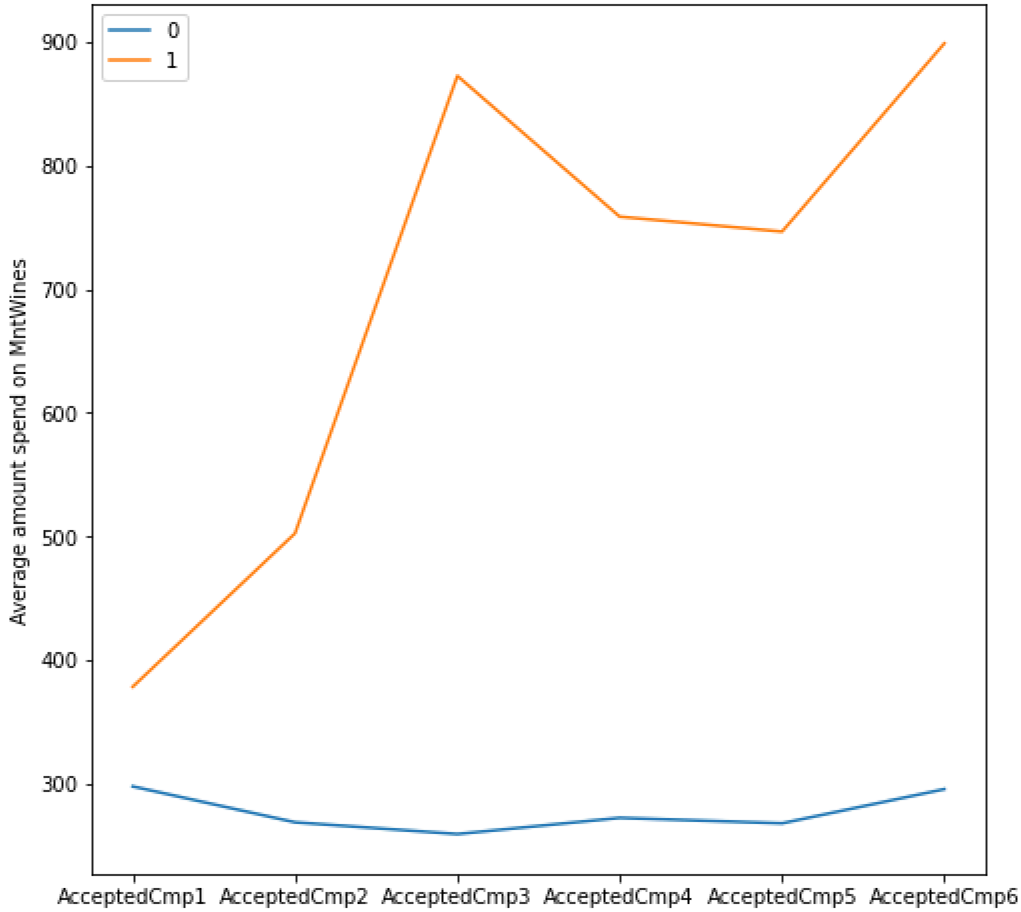
It could be that different campaigns are focussed on different set of products. Let's check if the product preference for those who accepted the campaigns is different from those who didn't - using amount spent and number of purchases

Let's define a function which will take the column name for the product as input and will generate the barplot for every campaign and average amount spent on a product

In [16]:
```
def amount_per_campaign(columns_name):
    p1=pd.DataFrame(df.groupby(['AcceptedCmp1']).mean()[columns_name]).T
    p2=pd.DataFrame(df.groupby(['AcceptedCmp2']).mean()[columns_name]).T
    p3=pd.DataFrame(df.groupby(['AcceptedCmp3']).mean()[columns_name]).T
    p4=pd.DataFrame(df.groupby(['AcceptedCmp4']).mean()[columns_name]).T
    p5=pd.DataFrame(df.groupby(['AcceptedCmp5']).mean()[columns_name]).T
    p6=pd.DataFrame(df.groupby(['AcceptedCmp6']).mean()[columns_name]).T
    pd.concat([p1,p2,p3,p4,p5,p6],axis=0).set_index([Camp_cols]).plot(kind='line
    plt.ylabel('Average amount spend on' + ' ' + columns_name)
    plt.show()
```

Use the function defined above to generate barplots for different purchasing Products

In [21]:
```
#here is an example showing how to use this function on the column MntWines
amount_per_campaign('MntWines')
```



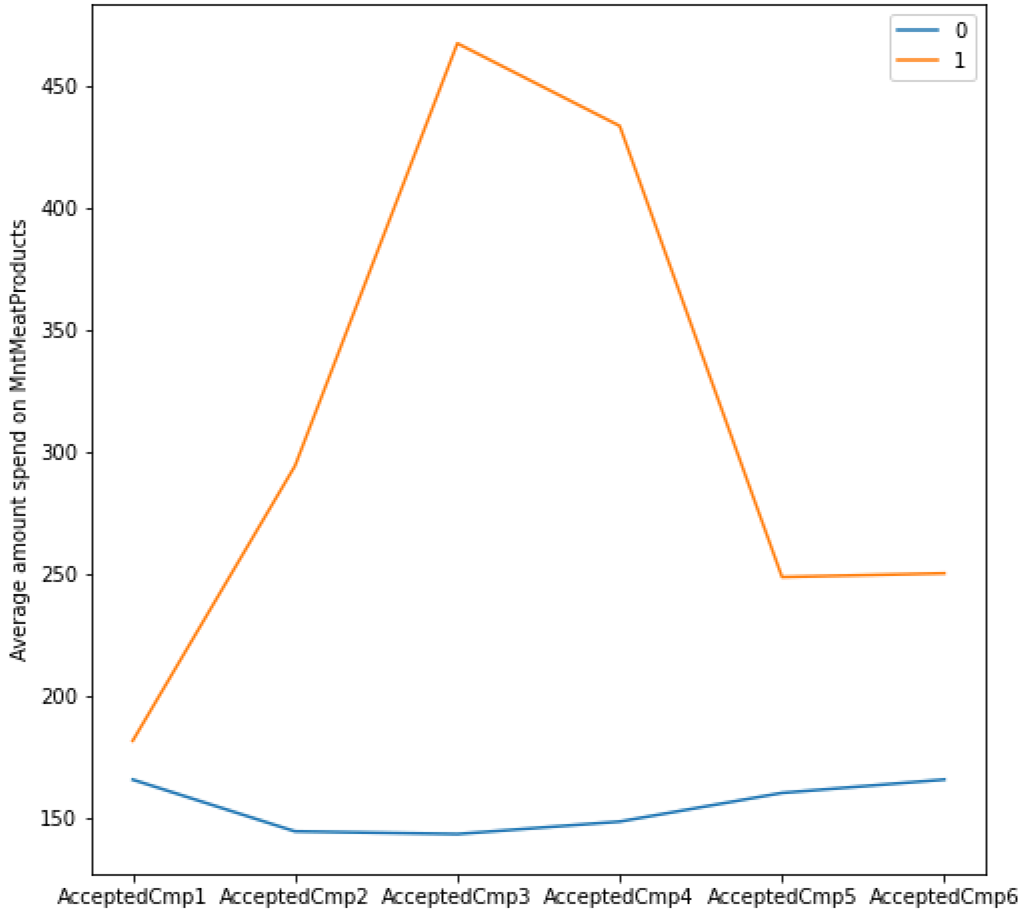
Observations:

- For the customers accepting campaign 3, 4, 5, and 6 the average amount spent on wine is quite high.

Question 4: Write the code and your observations on average amount spent on different products across all campaigns. - 7 Marks

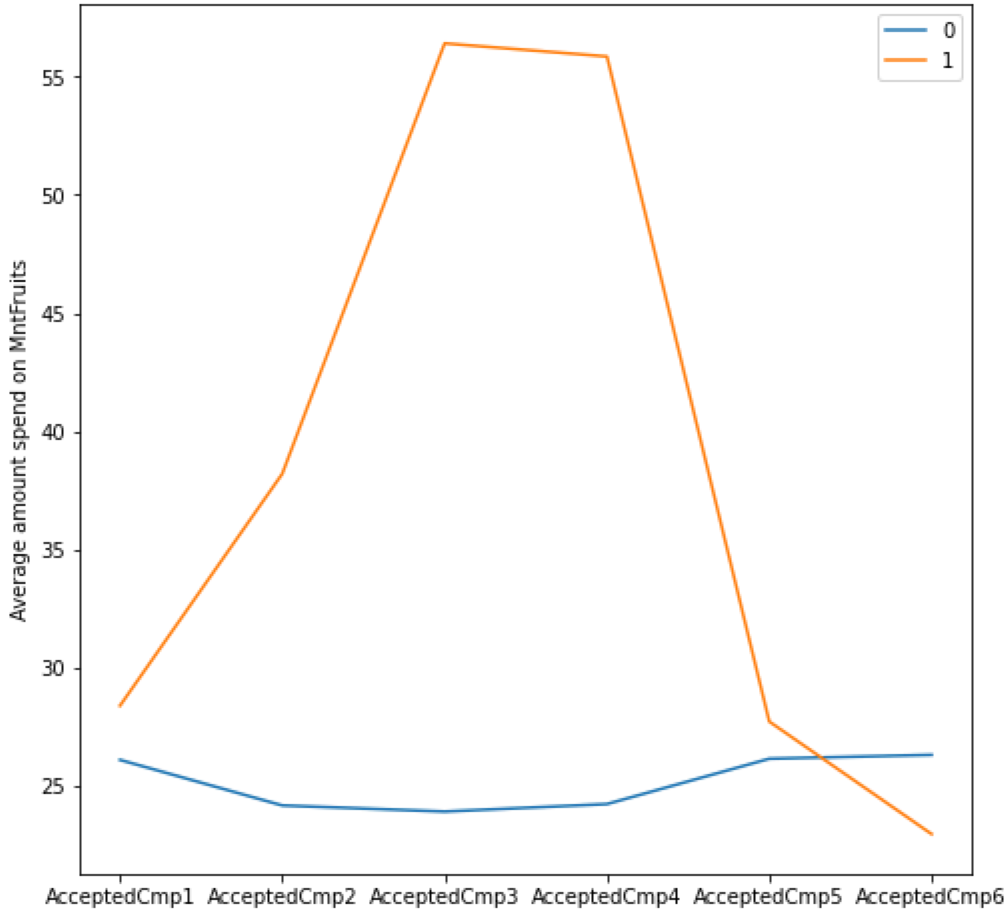
In [22]:

```
#meat products
amount_per_campaign('MntMeatProducts')
#call the function amount_per_campaign for MntMeatProducts
```



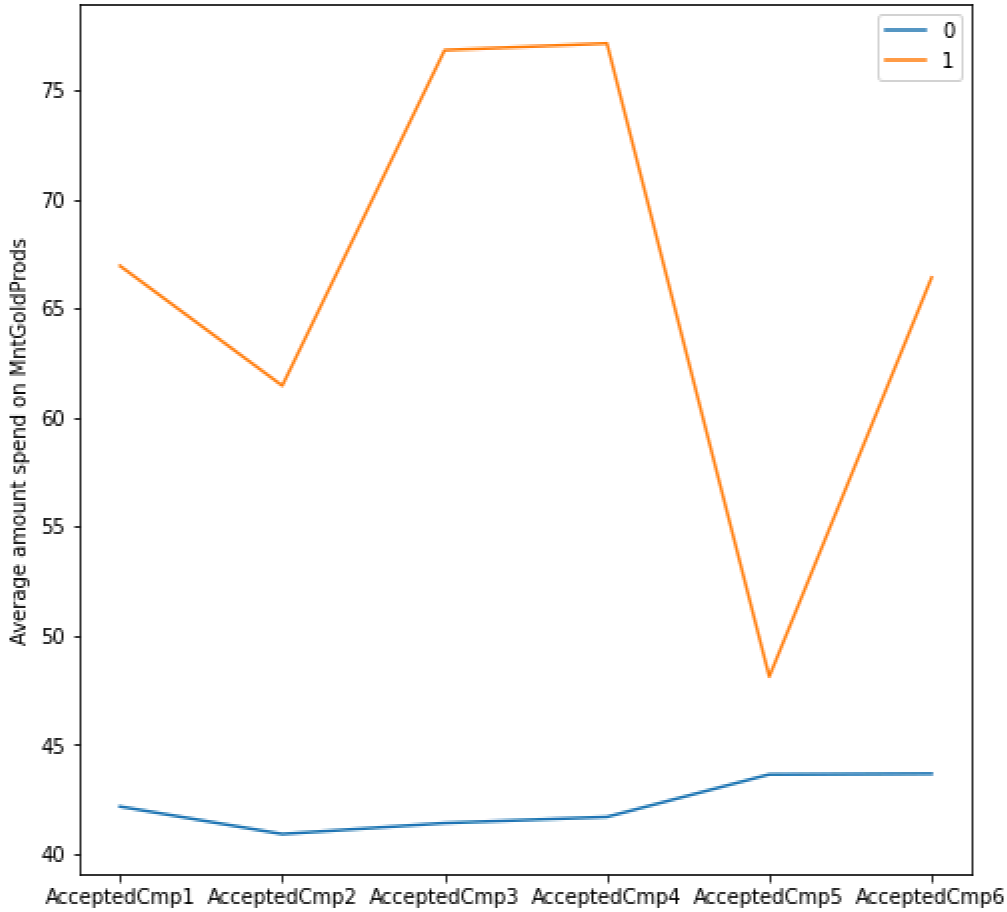
In [23]:

```
# Fruit products
amount_per_campaign('MntFruits')
#call the function amount_per_campaign for MntFruits
```



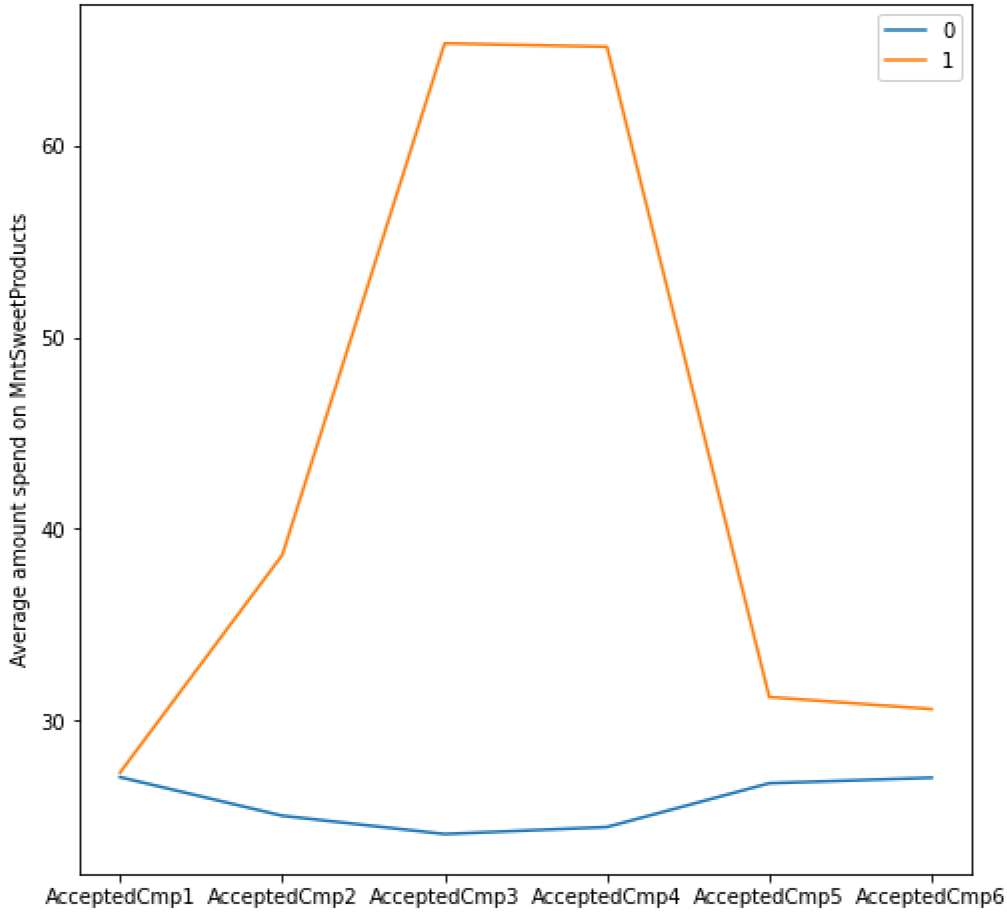
In [24]:

```
# gold products
amount_per_campaign('MntGoldProds')
#call the function amount_per_campaign for MntGoldProds
```



In [25]:

```
#sweet products
amount_per_campaign('MntSweetProducts')
#call the function amount_per_campaign for MntSweetProducts
```



Observations:_For the customers accepting campaign 3 and 4 the average amount spent on Meat is quite high For the customers accepting campaign 3 and 4 the average amount spent on Fruits is quite high. For the customers accepting campaign 1,3,4 and 6 the average amount spent on Gold is quite high. For the customers accepting campaign 3 and 4 the average amount spent on Sweet Products is quite high. It could be different campaigns are focussed on different set of products

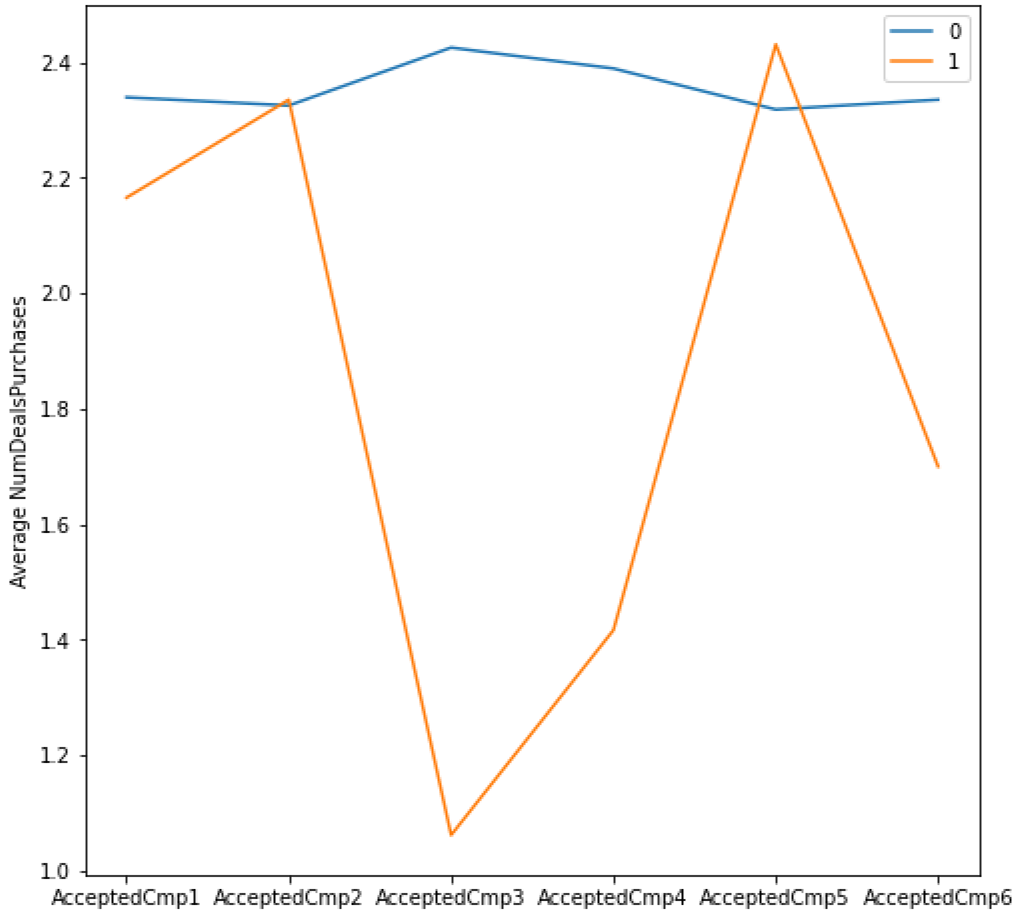
We have analyzed the relationship between campaigns and different products. Now, let's see the relationship of campaigns with different purchasing channels.

We have a defined a function which will take the column name of the channel name as input and will generate the barplot for every campaign and average purchase made through that channel if the campaign is accepted

In [26]:

```
def Purchases_per_campaign(columns_name):
    dp1=pd.DataFrame(df.groupby(['AcceptedCmp1']).mean()[columns_name]).T
    dp2=pd.DataFrame(df.groupby(['AcceptedCmp2']).mean()[columns_name]).T
    dp3=pd.DataFrame(df.groupby(['AcceptedCmp3']).mean()[columns_name]).T
    dp4=pd.DataFrame(df.groupby(['AcceptedCmp4']).mean()[columns_name]).T
    dp5=pd.DataFrame(df.groupby(['AcceptedCmp5']).mean()[columns_name]).T
    dp6=pd.DataFrame(df.groupby(['AcceptedCmp6']).mean()[columns_name]).T
    pd.concat([dp1,dp2,dp3,dp4,dp5,dp6],axis=0).set_index([Camp_cols]).plot(kinc
    plt.ylabel('Average' + ' ' + columns_name)
    plt.show()
```

In [27]:
```
#here is an example showing how to use this function on the column NumDealsPurch
Purchases_per_campaign('NumDealsPurchases')
```

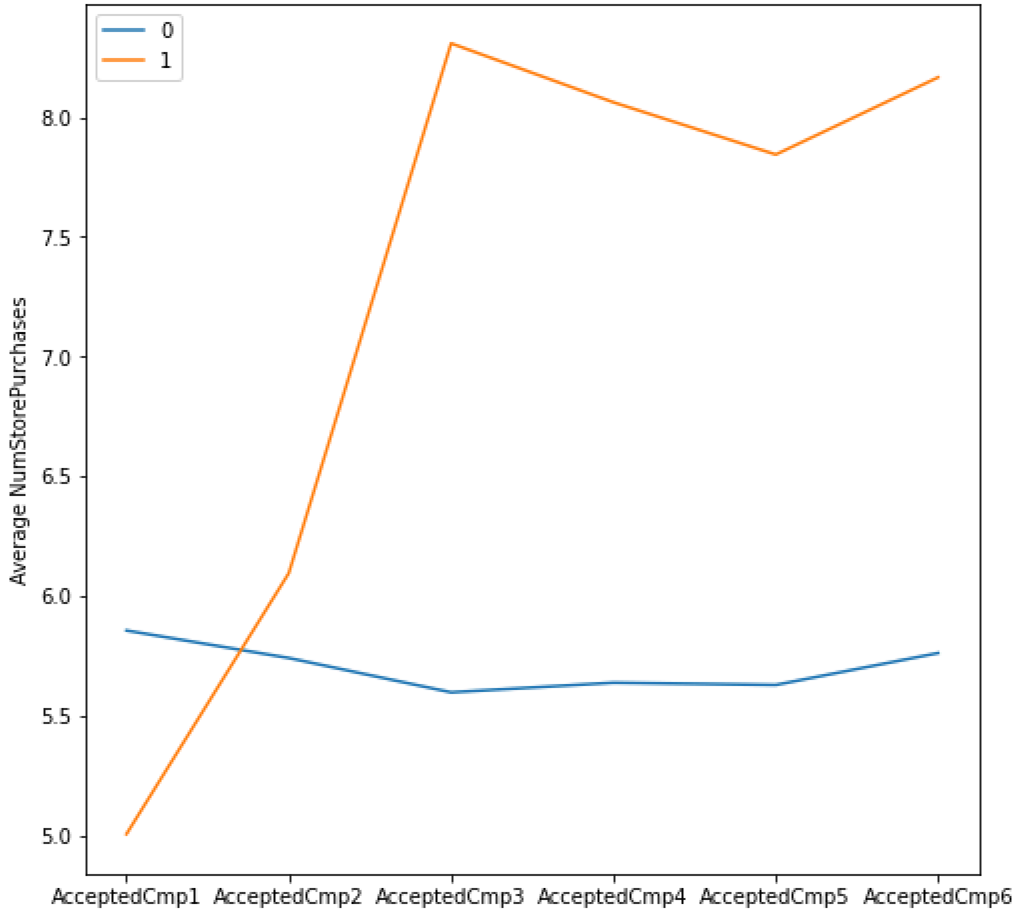


Observations:

- For the customers accepting campaign 3, 4, and 6 the average deals purchase is quite low.

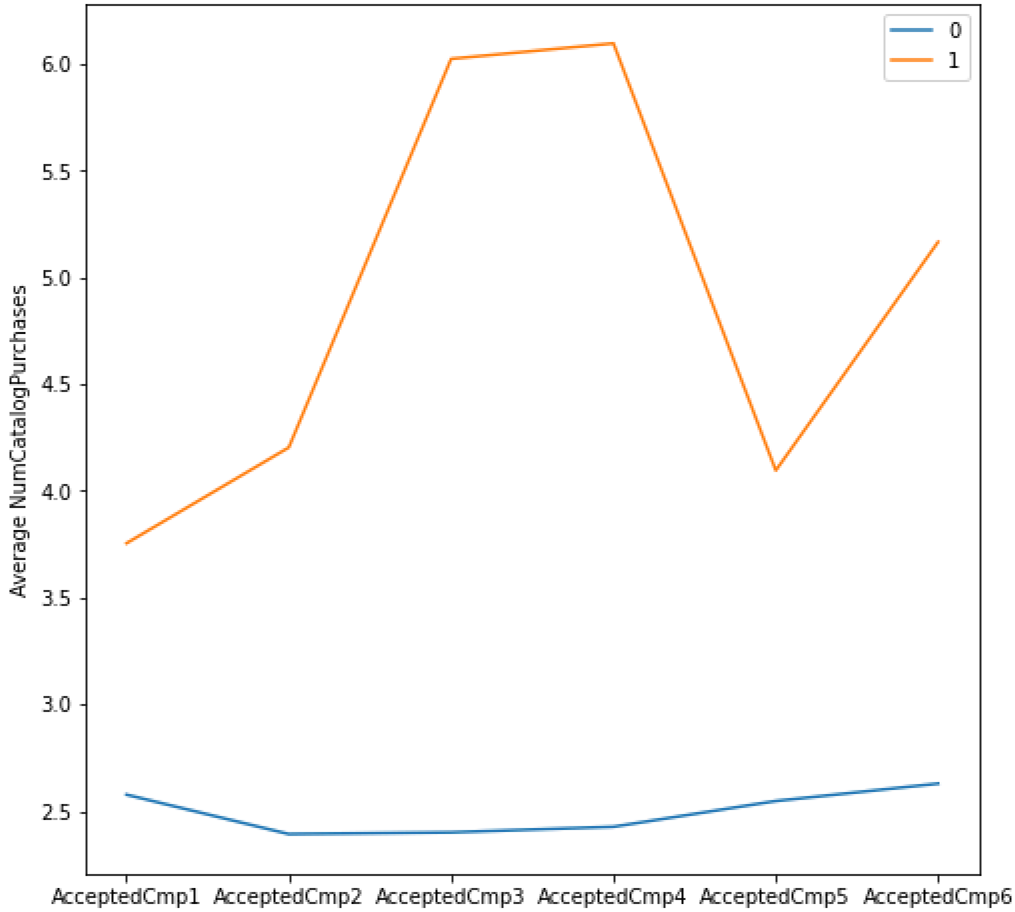
Question 5: Write the code and your observations on average number of purchases from different channels across all campaigns. - 7 Marks

In [28]:
```
# store purchase
Purchases_per_campaign('NumStorePurchases')
#call the function Purchases_per_campaign for NumStorePurchases
```

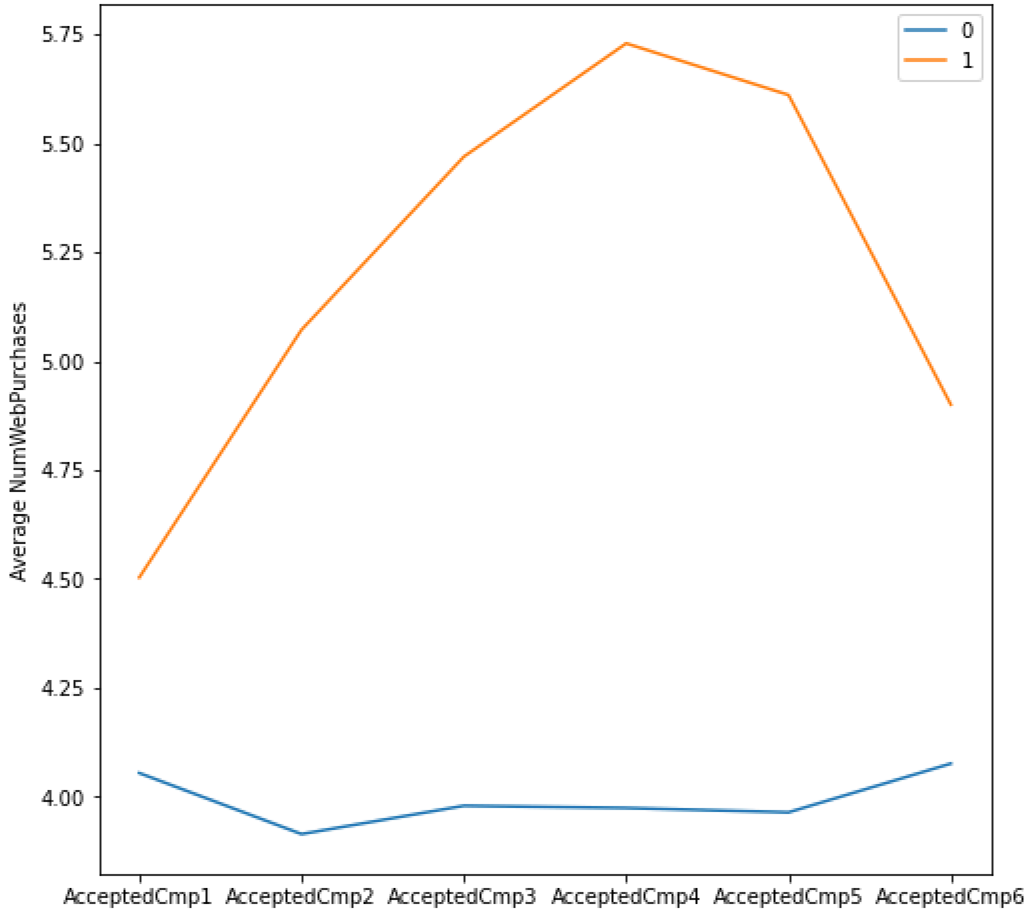


In [29]:

```
#Catalog purchase
Purchases_per_campaign('NumCatalogPurchases')
#call the function Purchases_per_campaign for NumCatalogPurchases
```

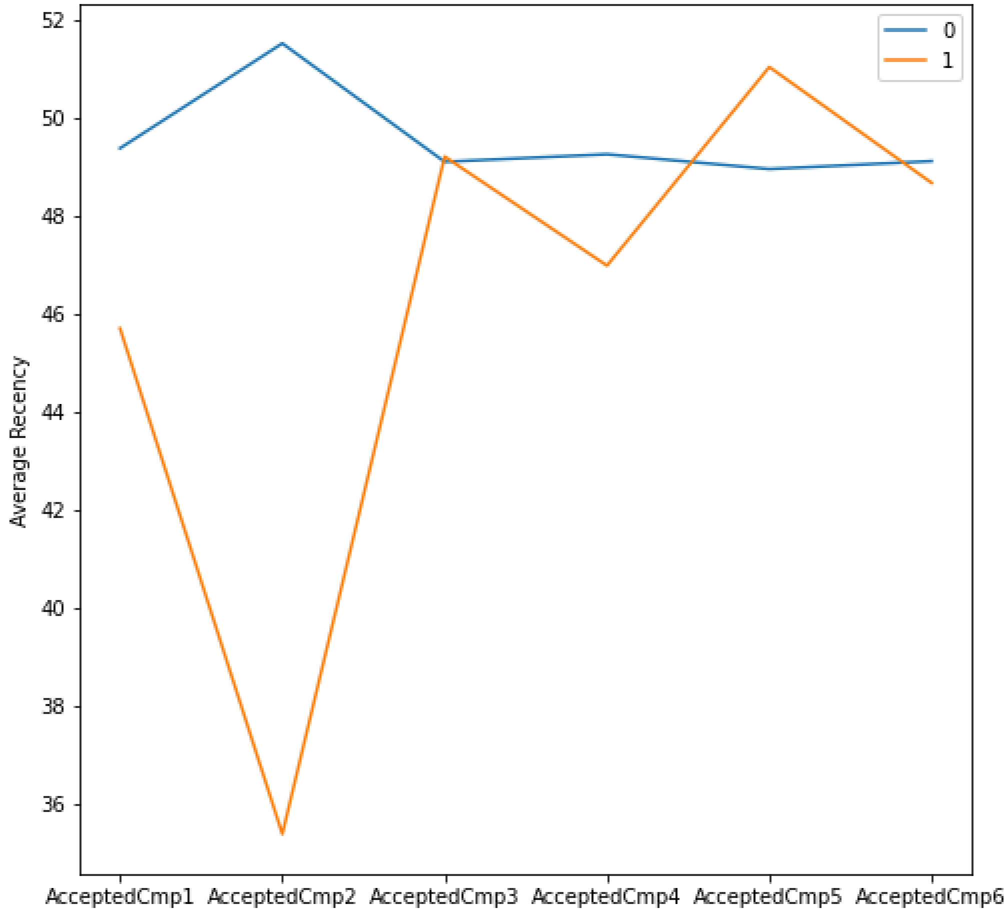


In [30]:
```
#Web purchases
Purchases_per_campaign('NumWebPurchases')
#call the function Purchases_per_campaign for NumWebPurchases
```



Observations:_For the customers accepting campaign 1 and 2 the average store purchase is quite low For the customers accepting campaign 1, 2 and 5 the average deals purchase is relavant lower. For the customers accepting campaign 1, and 6 the average web purchase is relavant lower.

In [31]:
```
#Recency

Purchases_per_campaign('Recency')
```



Observations:

- Average recency of the customers who accepted campaign 2 is quite low which implies that campaign 2 was accepted by the customers who recently purchased an item.

We have analyzed the relationship between campaigns and numerical variables. Let's see the relationship of campaigns with different categorical variables

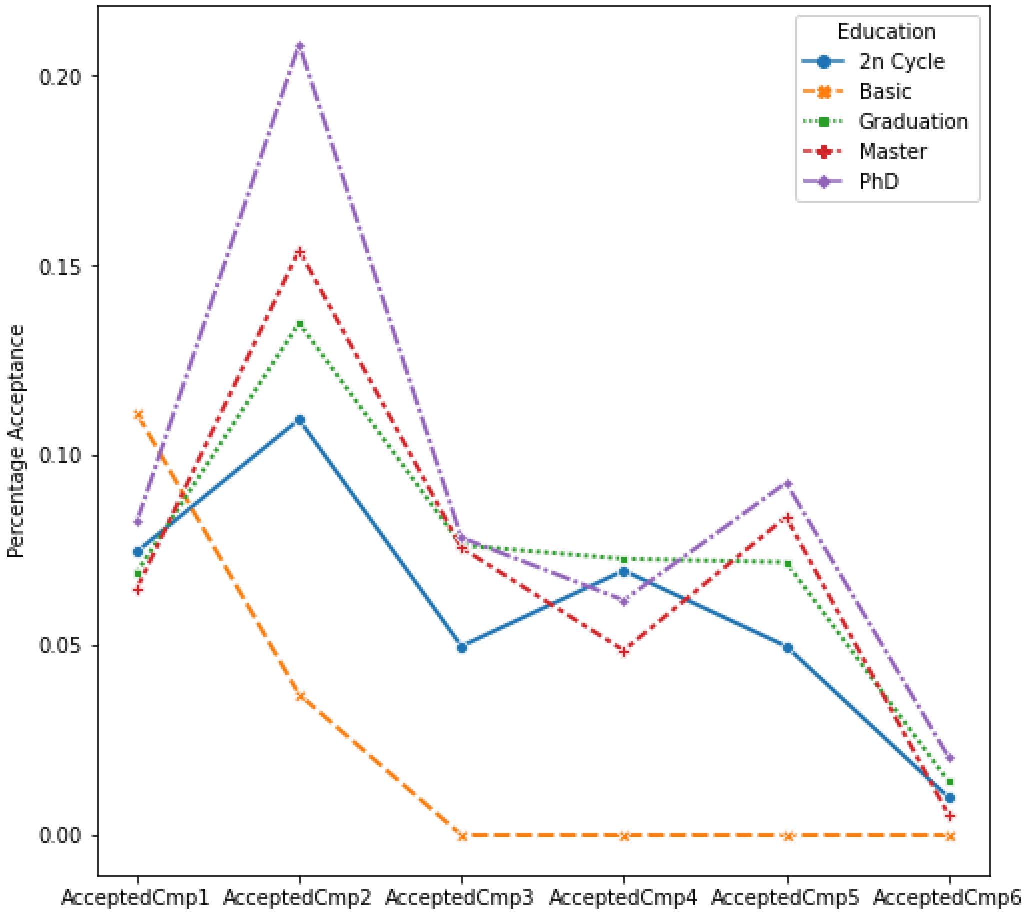
We will check the percentage acceptance of each campaign with respect to each category in the categorical variable. The percentage acceptance is calculated as number of customers who have accepted the campaign to the total number of customers.

In [32]:

```
def Cat_Campaign_Relation(df, column_name):
    e1=(df.groupby([column_name]).sum()['AcceptedCmp1']/df.groupby([column_name]
    e2=(df.groupby([column_name]).sum()['AcceptedCmp2']/df.groupby([column_name]
    e3=(df.groupby([column_name]).sum()['AcceptedCmp3']/df.groupby([column_name]
    e4=(df.groupby([column_name]).sum()['AcceptedCmp4']/df.groupby([column_name]
    e5=(df.groupby([column_name]).sum()['AcceptedCmp5']/df.groupby([column_name]
    e6=(df.groupby([column_name]).sum()['AcceptedCmp6']/df.groupby([column_name]
    df_new=pd.concat([e1,e2,e3,e4,e5,e6],axis=1).T
    plt.figure(figsize=(8,8))
    sns.lineplot(data=df_new, markers=True, linewidth=2)
    plt.ylabel('Percentage Acceptance')
    plt.show()
```

In [33]:

```
#here is an example showing how to use this function on the column Education
Cat_Campaign_Relation(df, 'Education')
```

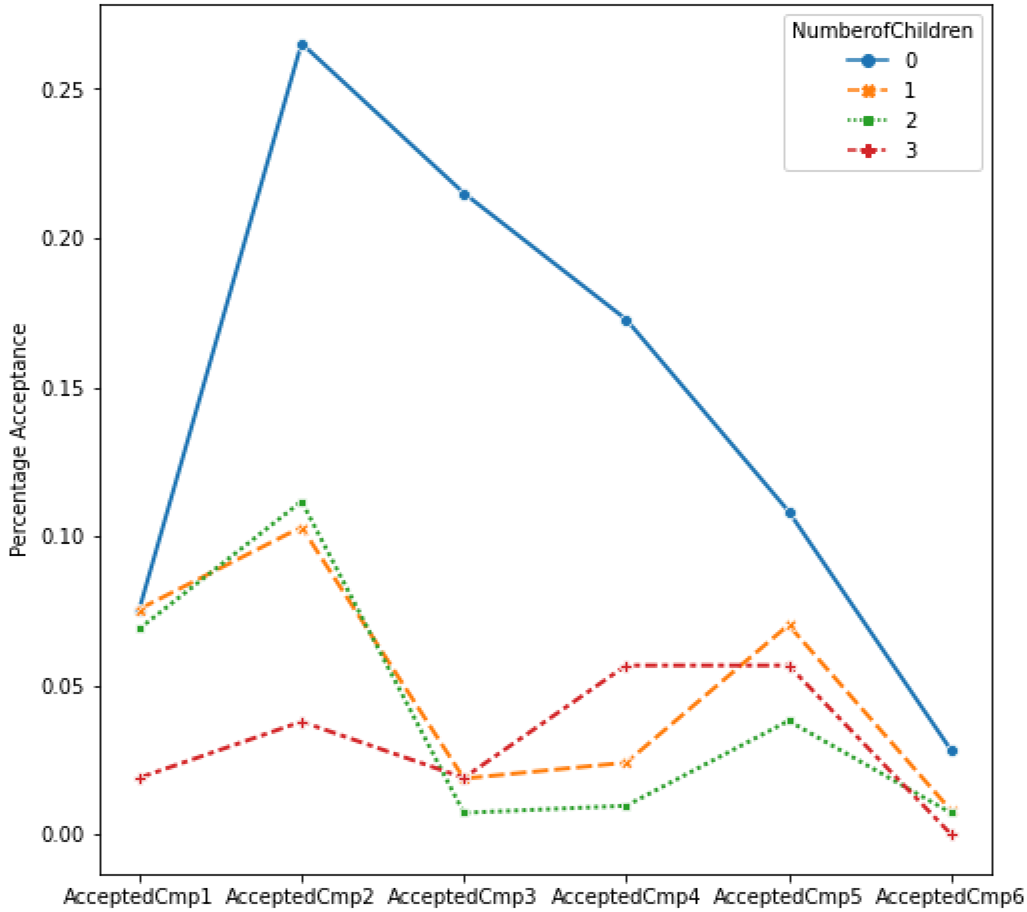


Observations:

- More than 20% of the customers with Ph.D have accepted campaign 2.
- Customers with basic education have only accepted campaign 1 and 2.
- Except customers with basic education level, all education levels follow the same trend.

Question 6: Write the code and your observations on percentage acceptance for different categorical variables across all campaigns. - 7 Marks

In [34]:
```
#NumberofChildren
Cat_Campaign_Relation(df, 'NumberofChildren')
#call the function Cat_Campaign_Relation for NumberofChildren
```



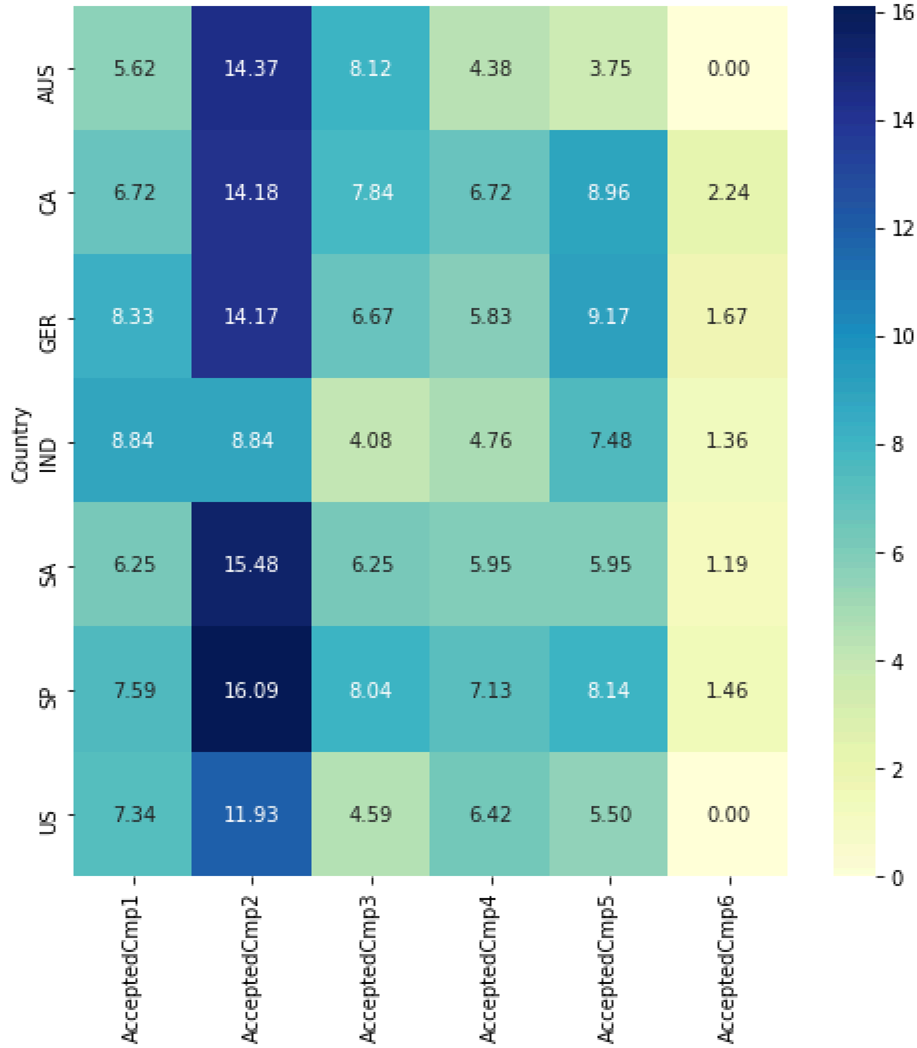
In [ ]:

```
#Let's filter the observations with 'Others' category as they are only 4 such ob
df_rest=df[df.Marital_Status!='Others']

#call the function Cat_Campaign_Relation for Marital_Status with dataframe df_re
```

In [35]:

```
#Let's filter the observations for 'ME' country as they are only 3 such observat
df_not_mexico=df[df.Country!='ME']

#Plot
plt.figure(figsize=(8,8))
sns.heatmap((df_not_mexico.groupby('Country').sum()[Camp_cols]/df_not_mexico.grc
```

Out[35]: <AxesSubplot:ylabel='Country'>

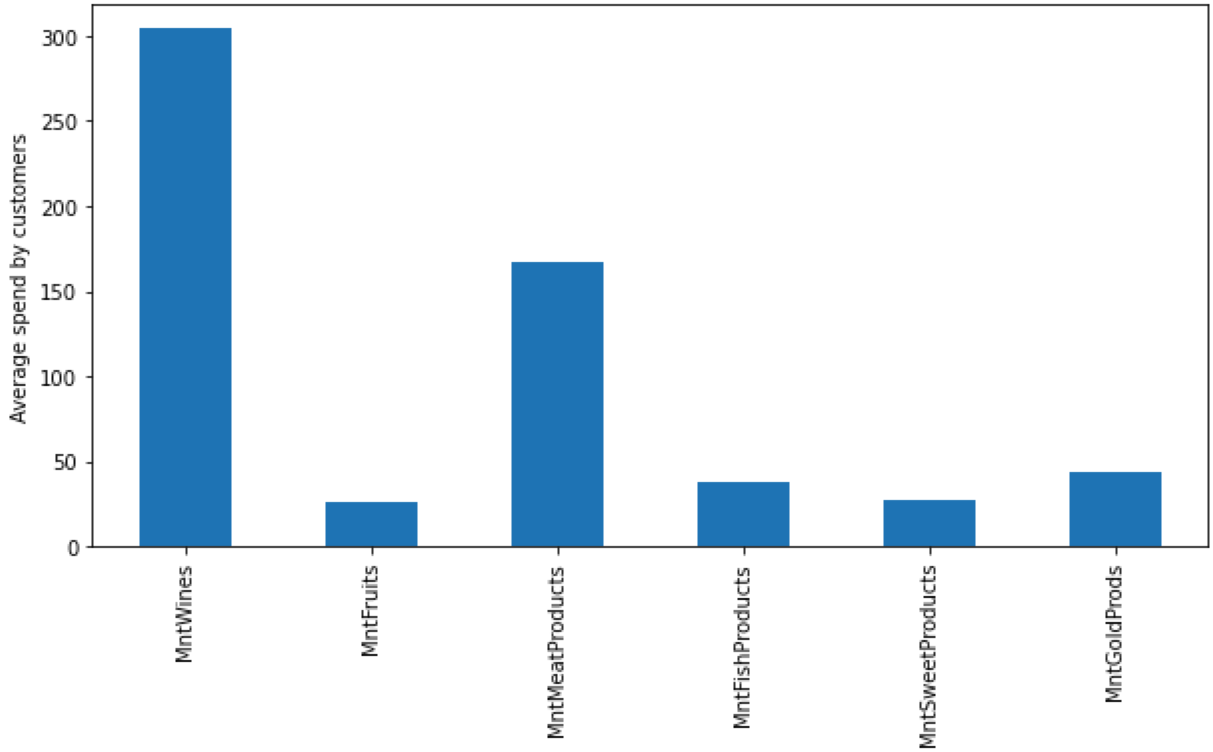


**Observation:US,SP,SA,IND,GER,CA and AUS all have highet acceptance rate in 2nd campaign and lowest acceptance rate in 6th campaign. SP has the highest acceptance rate in 2nd campaign while IND has the lowest acceptance rate. Acceptance of 1st campaign over these countries does not have signicant difference.

Check the product preferences by customers

In [38]:

```
#creating a list which contains name of all products

mnt_cols = [col for col in df.columns if 'Mnt' in col]

spending=df[mnt_cols].mean(axis=0)
spending.plot(kind='bar', figsize=(10,5))
plt.ylabel("Average spend by customers")
plt.show()
```



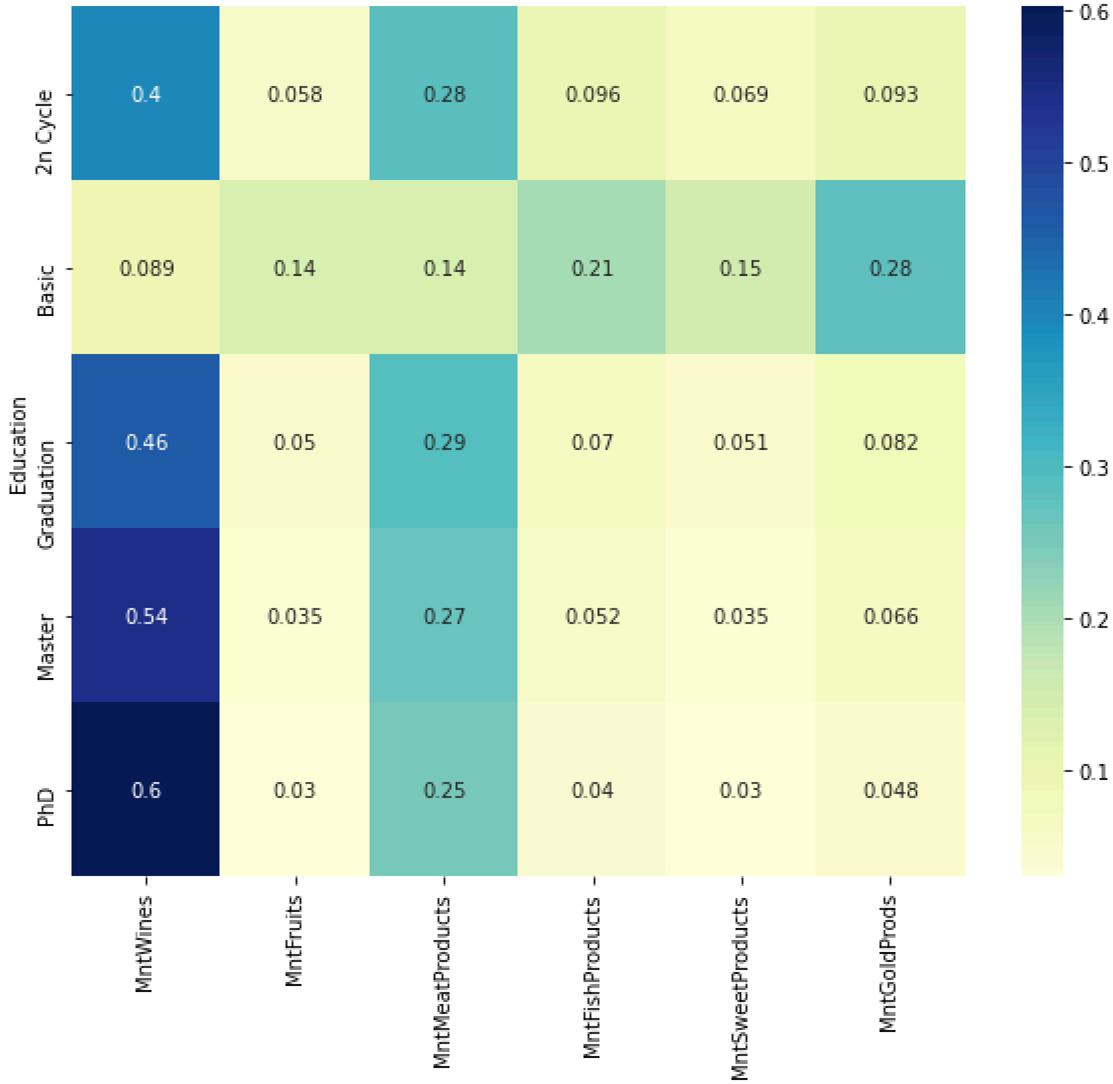
Observations:

- The mean amount spent by customers in the last 2 years is highest for wines followed by meat products.

Let's check if the product preferences are similar for different types of customers. We will calculate the percentage amount spent by customers on a product for each category with respect to the total spending by customers belonging to that category.

In [39]:
```
def amount_per_category(df, column_name):
    df_new1=((df.groupby([column_name]).sum()[mnt_cols].T)/df.groupby([column_na
    plt.figure(figsize=(10,8))
    sns.heatmap(df_new1.T, annot=True, cmap="YlGnBu")
    plt.show()
```

In [40]:
```
# plot showing the percentage of total spending of different products by a group

amount_per_category(df, 'Education')
```

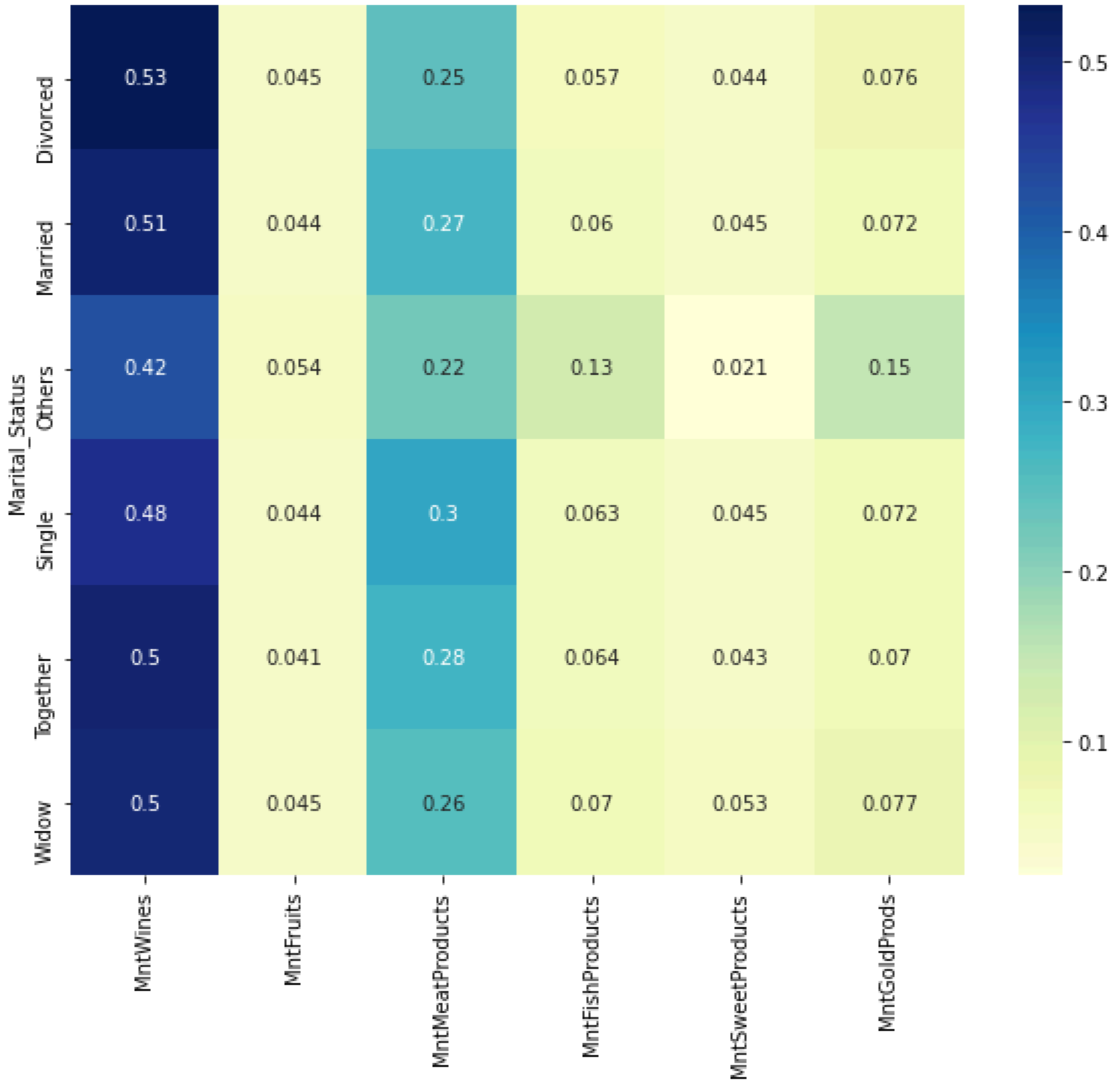


Observations:

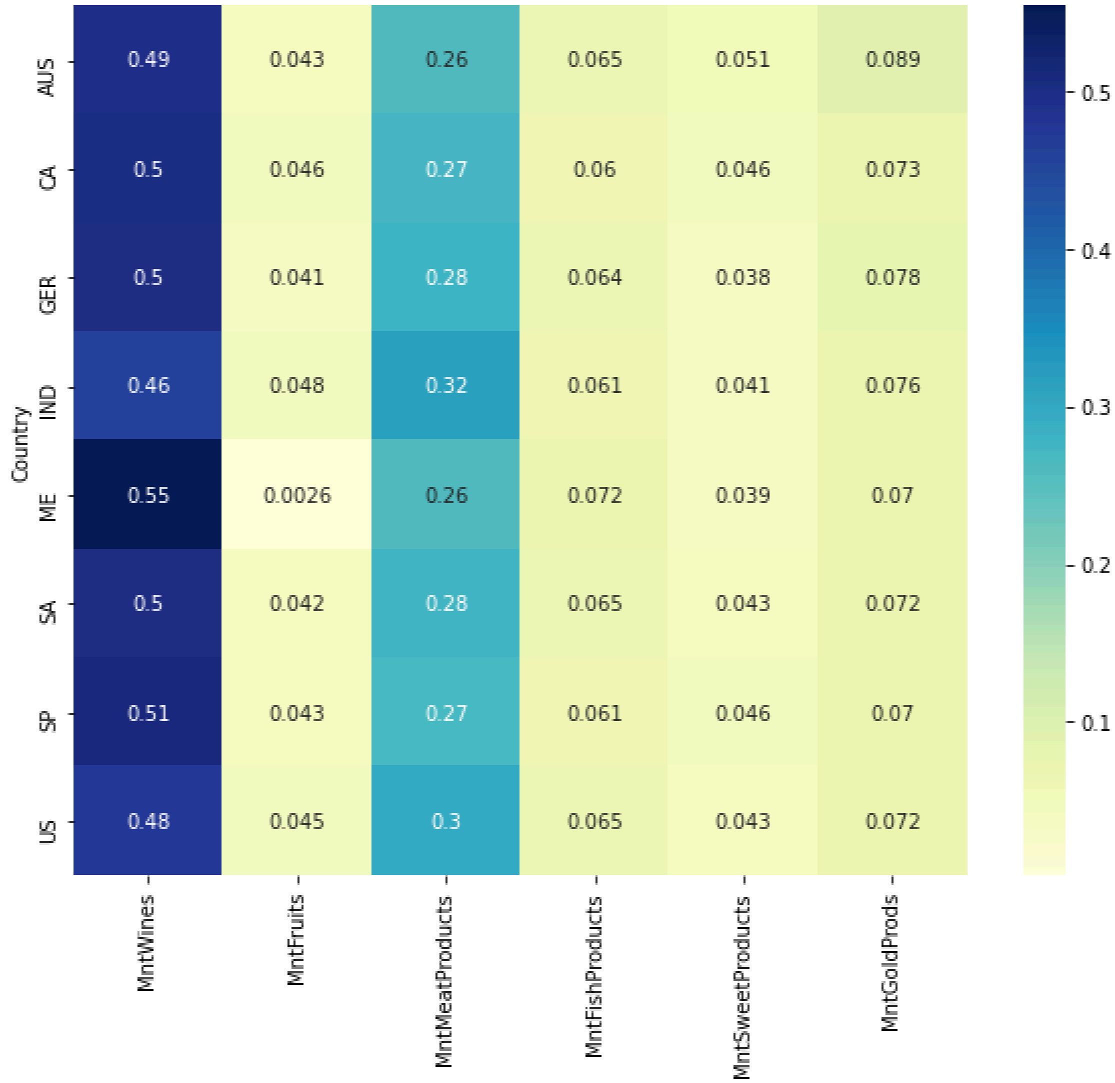
- Customers with PhD spend ~60% of their total spending on wines.
- Customers with Graduation and Master's spend ~45-50% of their total spending on wines.
- Customers with Graduation and Master's spend ~27-29% of their total spending on meat.
- Customers with PhD spend ~25% of their total spending on meat.
- Customers having education level Master or PhD spend ~80% on meat and wines.
- Customers with basic education spend more on Fruits, Fish, Sweet, and Gold products.

Question 7: Write the code and your observations on percentage amount spent on different products for each category of the mentioned categorical variables. - 7 Marks

In [47]:
```
#call the function amount_per_category for Marital_Status with dataframe df_rest
amount_per_category(df, 'Marital_Status')
```



In [48]:
```
#call the function amount_per_category for Country with dataframe df_not_mexico
amount_per_category(df, 'Country')
```

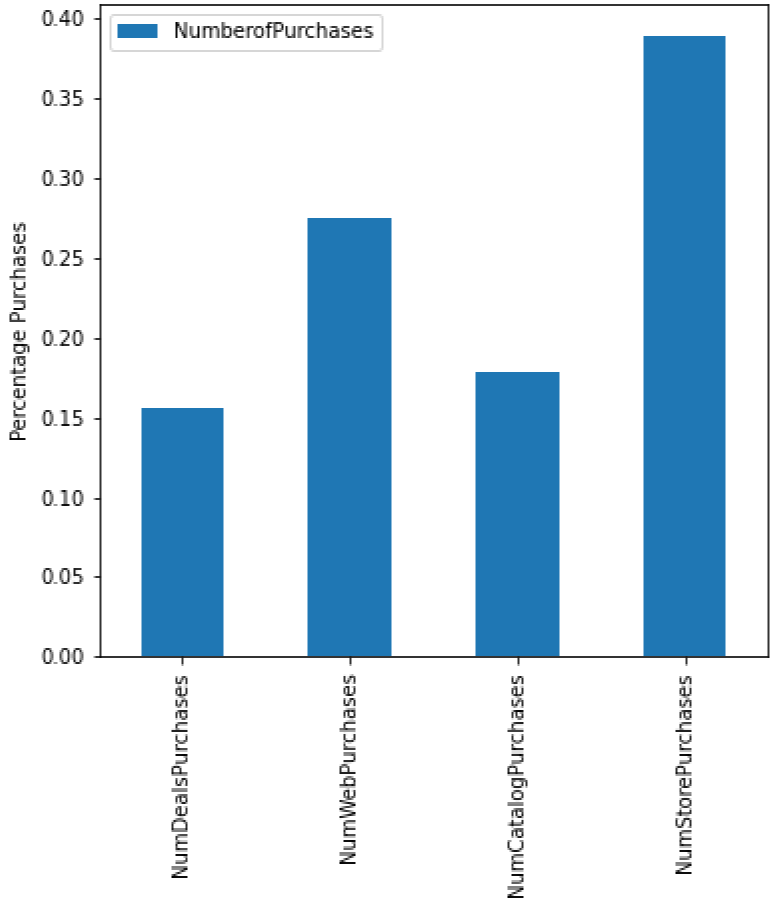


Observations:Customers spent ~50% of their total spendings on wines Customers who are divorced spent highest of their total spending on wines, and followed by married customers. Single customers spent more on meat products. Customer whose marrital status is other spent more on gold products. IND customer spent more on meat products. ME customer spent <0.3% of their total spendings on Fruits ___

Check different channel performances

Let's calculate the percentage of purchases for all the channels.

In [49]:
```
# list of cols for channels

channel_cols = [col for col in df.columns if 'Purchases' in col]

#making dataframe of columns having purchase and taking sum of them.
channels = pd.DataFrame(df[channel_cols].sum()/df.Total_Purchase.sum(), columns=

# plot
channels.plot(kind='bar', figsize=(6,6))
plt.ylabel("Percentage Purchases")
plt.show()
```



Observations:

- We can see that the most purchases are from the stores followed by web purchases.
- Number of deal purchases and catalog purchases are low.

Question 8: Write your observations on percentage purchases from different channels for different categories of the income_cat column. - 4 Marks

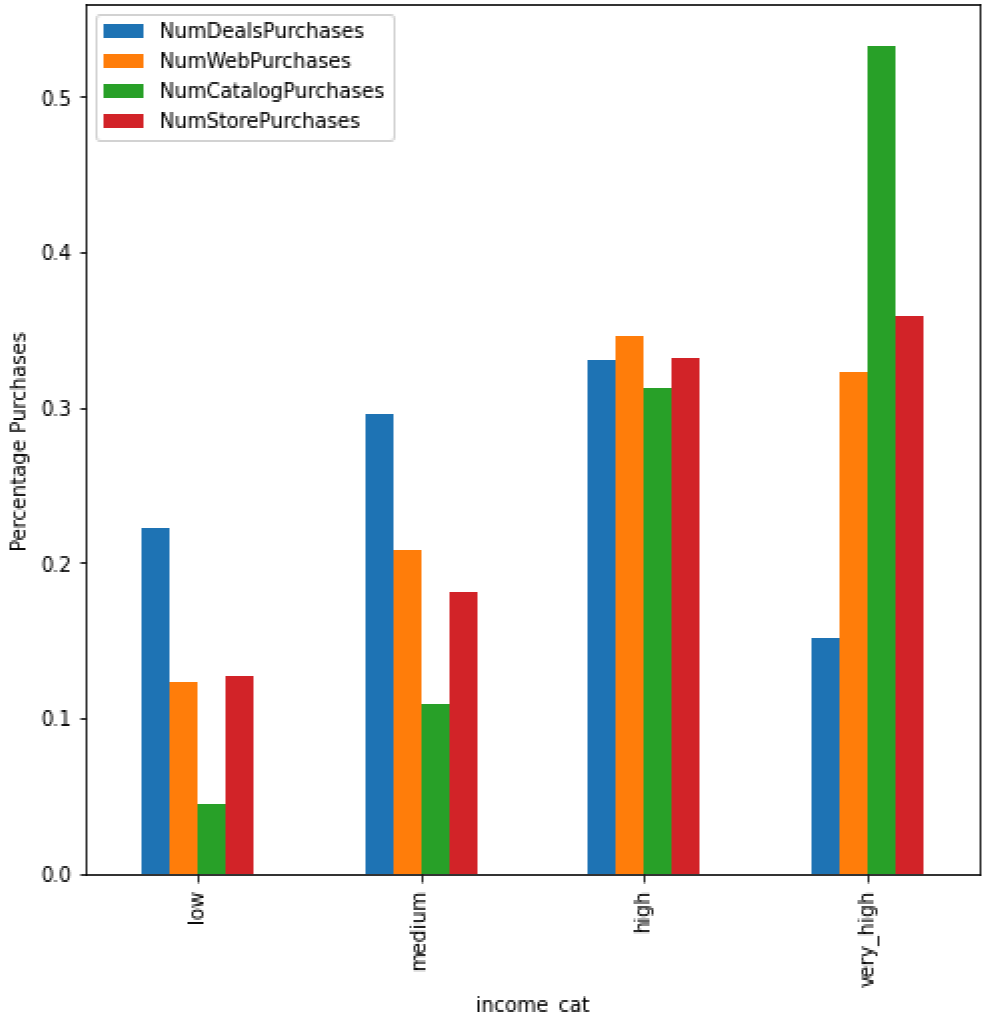
Let's check how number of purchases via different channels varies for different income bins.

In [50]:

```
#Binning the income column
df['income_cat']=pd.qcut(df.Income, q=[0, 0.25, 0.50, 0.75, 1], labels=['low', '
```

In [51]:

```
group=df.groupby('income_cat').sum()[channel_cols]
(group/group.sum()).plot(kind='bar', figsize=(8,8))
plt.ylabel("Percentage Purchases")
plt.show()
```

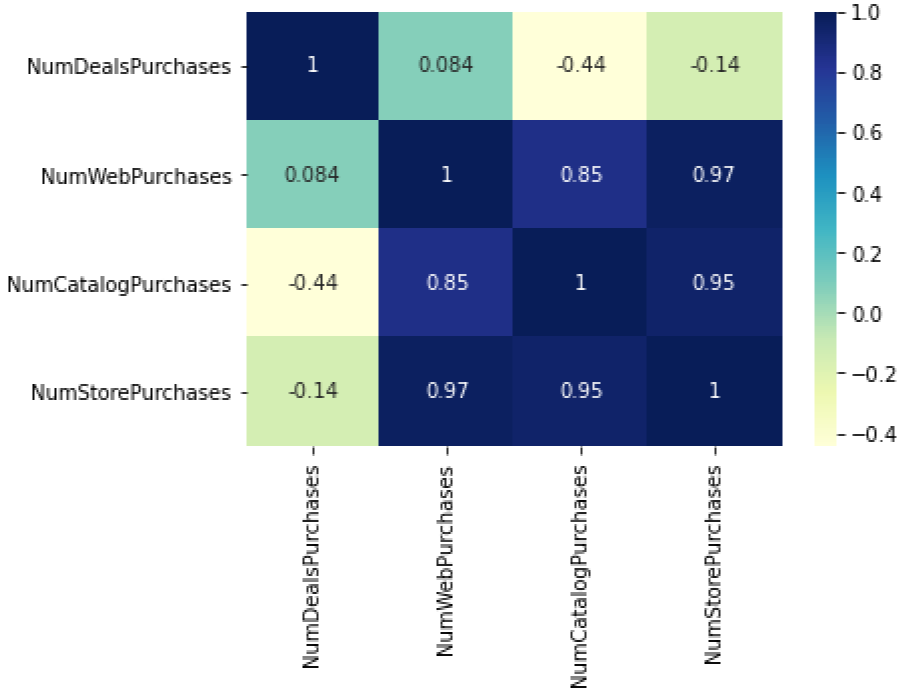


**Observations:_Low and Medium income customer make least purchase from Catlog and most purchase are from deals followed by web purchase. Different channels didn't show signicant difference of puchase amount on high income customer. Very high income customer make most purchase from Catalog fowllowed by store and make least purchase from deals. Customers in different income segments show different preference on purchase channgels.

We can also visualize the correlation by purchases from different channels and income of the customer.

Question 9: Find the correlation matrix for the columns mentioned below and visualize the same using heatmap. - 3 Marks

In [59]:
```
corr=df[['Income', 'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases'
dataplot = sns.heatmap(group.corr(),cmap="YlGnBu", annot=True)
#Write your code here
```



**Observations:_Customer who make deals purchases are high likely to make purchases from web but unlikely to make catalog purchases and store purchases Customer who make web purchases are high likely to make catalog puchase and store purchase but unlikely to purchase from web Customer who make catalog and store purchase unlikely make deal purchases.

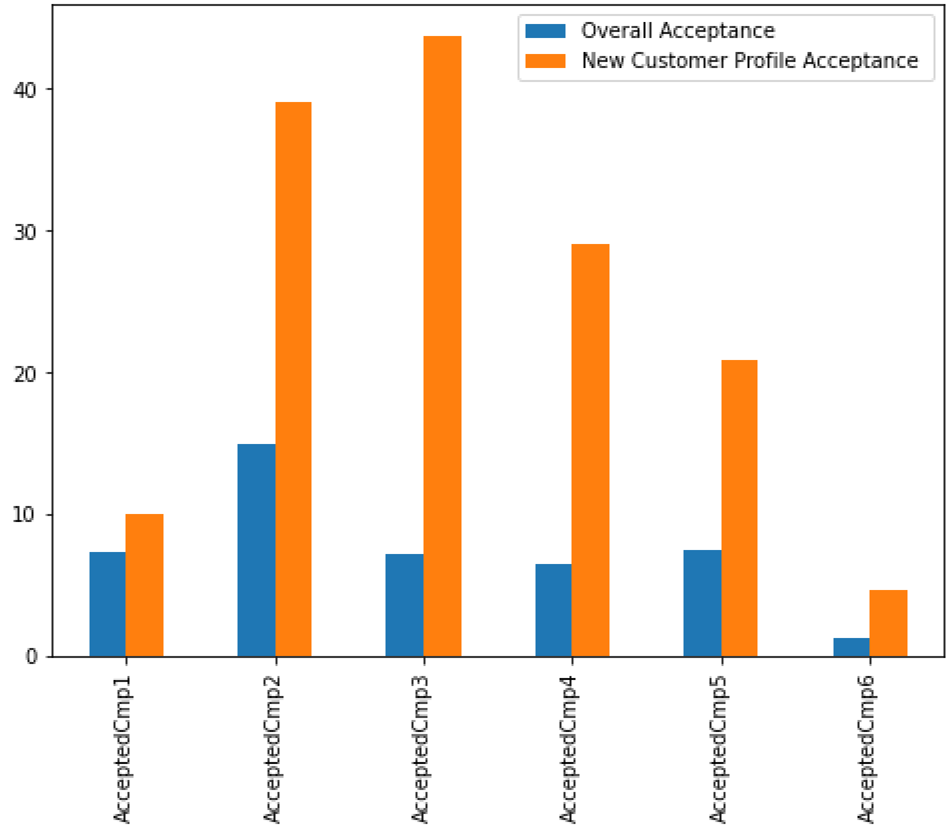
As we know from our analysis we have done so far that customers with income, number of children, and amount spending on wines are the important factors. Let's try to come up with new customer profile on the basis of these 3 attributes and check what would be the acceptance rate for that customer profile.

In [ ]:

```
df3=df[df.Income>52000]
df4=df3[df3.MntWines>df3.MntWines.mean()]
new_profile=df4[df4.NumberofChildren==0]
```

In [ ]:

```
#Calculating success rate of each campaign for both segments
success_campaign3=pd.DataFrame(success_campaign, columns=['Overall Acceptance'])

success_campaign4=pd.DataFrame((new_profile[Camp_cols].sum()/new_profile.shape[0

# plot
pd.concat([success_campaign3, success_campaign4], axis=1).plot(kind='bar', figsi
plt.title("")
plt.ylabel("")
plt.show()
```



Observations:

- Orange bars in the plot indicates that acceptance rate would have been high for new customer profile i.e. income greater than the mean income, no kid at home, amount spent of wines is greater than the mean amount spent on wines.

Question 10: Based on your analysis, write the conclusions and recommendations for the CMO to help make the next marketing campaign strategy. - 10 Marks

Conclusion and Recommendations

1.Different campaign has different accpetance rate. Most customer accept the offer in 2nd campaign. 2.The higher income the higher acceptance rate. 3.Customer's product preference and acceptance rate are relavant. 4.Customer with different income level has different preference on purchase channels, therefore influence the acceptance rate. 5.Customer spend different amount on each category products.

Recommendations: 1.Use user profile to better target potential customers.Let data drive the creative. 2.Provide personalized market campaign to different kind of customers. 3.Send 5th and 6th market campaign to royal customers instead of everyone.