DATA SCIENCE

DATASET: MARKETING ANALYSIS

CONTENT

- INTRODUCTION
- EDA(EXPLORATORY DATA ANALYSIS)
- FEATURE ENGINEERING
- DATA FEATURING
- VISUALIZATION
- CONCLUSION

INTRODUCTION

The Importance of Marketing Analytics

Market analysis is one of the vital components to help business with all the essential information and making wise business decisions.

What is EDA?

EDA stands for Exploratory Data Analysis. It is a process that shows you or rather helps you to explore your data using visualization and transformation methodologies in a systematic way.

EDA

DATA ANALYSIS

- Imported Libraries:

```
#Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
sns.set()
import matplotlib.pyplot as plt
%matplotlib inline
import datetime as dt
```

- Total number of rows and columns:

```
[ ]  data.columns

     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', 'Response', 'Complain', 'Country'],
           dtype='object')
```

```
data.shape

(2240, 28)
```

- Missing data:

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

ID                 0
Year_Birth         0
Education          0
Marital_Status     0
 Income           24
```

```
[ ]  #Remove null values
     data['Income'] = data['Income'].fillna(data['Income'].mean())
```

UNIQUE VALUES AND THEIR INFORMATION

```
data.nunique()
```

```
ID                     2240
Year_Birth               59
Education                 5
Marital_Status            8
 Income                1974
Kidhome                   3
Teenhome                  3
Dt_Customer             663
Recency                 100
MntWines                776
MntFruits               158
MntMeatProducts         558
MntFishProducts         182
MntSweetProducts        177
MntGoldProds            213
NumDealsPurchases        15
NumWebPurchases          15
NumCatalogPurchases      14
NumStorePurchases        14
NumWebVisitsMonth        16
AcceptedCmp3              2
AcceptedCmp4              2
AcceptedCmp5              2
AcceptedCmp1              2
AcceptedCmp2              2
Response                  2
Complain                  2
Country                   8
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 28 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   object
 5   Kidhome              2240 non-null   int64 
 6   Teenhome             2240 non-null   int64 
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64 
 9   MntWines             2240 non-null   int64 
 10  MntFruits            2240 non-null   int64 
 11  MntMeatProducts      2240 non-null   int64 
 12  MntFishProducts      2240 non-null   int64 
 13  MntSweetProducts     2240 non-null   int64 
 14  MntGoldProds         2240 non-null   int64 
 15  NumDealsPurchases    2240 non-null   int64 
 16  NumWebPurchases      2240 non-null   int64 
 17  NumCatalogPurchases  2240 non-null   int64 
 18  NumStorePurchases    2240 non-null   int64 
 19  NumWebVisitsMonth    2240 non-null   int64 
 20  AcceptedCmp3         2240 non-null   int64 
 21  AcceptedCmp4         2240 non-null   int64 
 22  AcceptedCmp5         2240 non-null   int64 
 23  AcceptedCmp1         2240 non-null   int64 
 24  AcceptedCmp2         2240 non-null   int64 
 25  Response             2240 non-null   int64 
 26  Complain             2240 non-null   int64 
 27  Country              2240 non-null   object
dtypes: int64(23), object(5)
memory usage: 490.1+ KB
```

DATA CLEANING

Removal of special symbol and converting the data type:

```
[6]  data.rename({' Income ' : 'Income'}, axis = 1, inplace = True)

     data['Income'] = data['Income'].str.replace('$', '')
     data['Income'] = data['Income'].str.replace(',', '').astype(float)
```

Formatting Dt_Customer into datetime:

```
[10]  data['Dt_Customer'] = pd.to_datetime(data['Dt_Customer'])
```

Deleting Unwanted Columns:

```
[21]  data.drop(columns=['ID', 'Dt_Customer','Teenhome','Kidhome'], inplace=True)
```

People With Age more than 80 are segregated:

```
[6]  data['Customer_Age'] = data['Dt_Customer'].dt.year - data['Year_Birth']
```

```
data[data['Customer_Age'] >=80]
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	Mnt
513	11004	1893	2n Cycle	Single	$60,182.00	0	1	2014-05-17	23	8	0	5	
827	1150	1899	PhD	Together	$83,532.00	0	0	2013-09-26	36	755	144	562	
2233	7829	1900	2n Cycle	Divorced	$36,640.00	1	0	2013-09-26	99	15	6	8	

FEATURE ENGINEERING

Adding different columns and creating new column:

```
[17] data['Customer_Age'] = data['Dt_Customer'].dt.year - data['Year_Birth']
```

```
[ ]  data['Total Spent'] = (data['MntFishProducts'] + data['MntWines'] + data['MntSweetProducts']
                          + data['MntFruits'] + data['MntMeatProducts'] + data['MntGoldProds'] )
```

```
[ ]  data['Total Purchases'] = (data['NumDealsPurchases'] + data['NumWebPurchases'] + data['NumStorePurchases'] +
                              data['NumCatalogPurchases'])
```

```
[19] data['Children'] = data['Kidhome'] + data['Teenhome']
```

One-hot encoding:

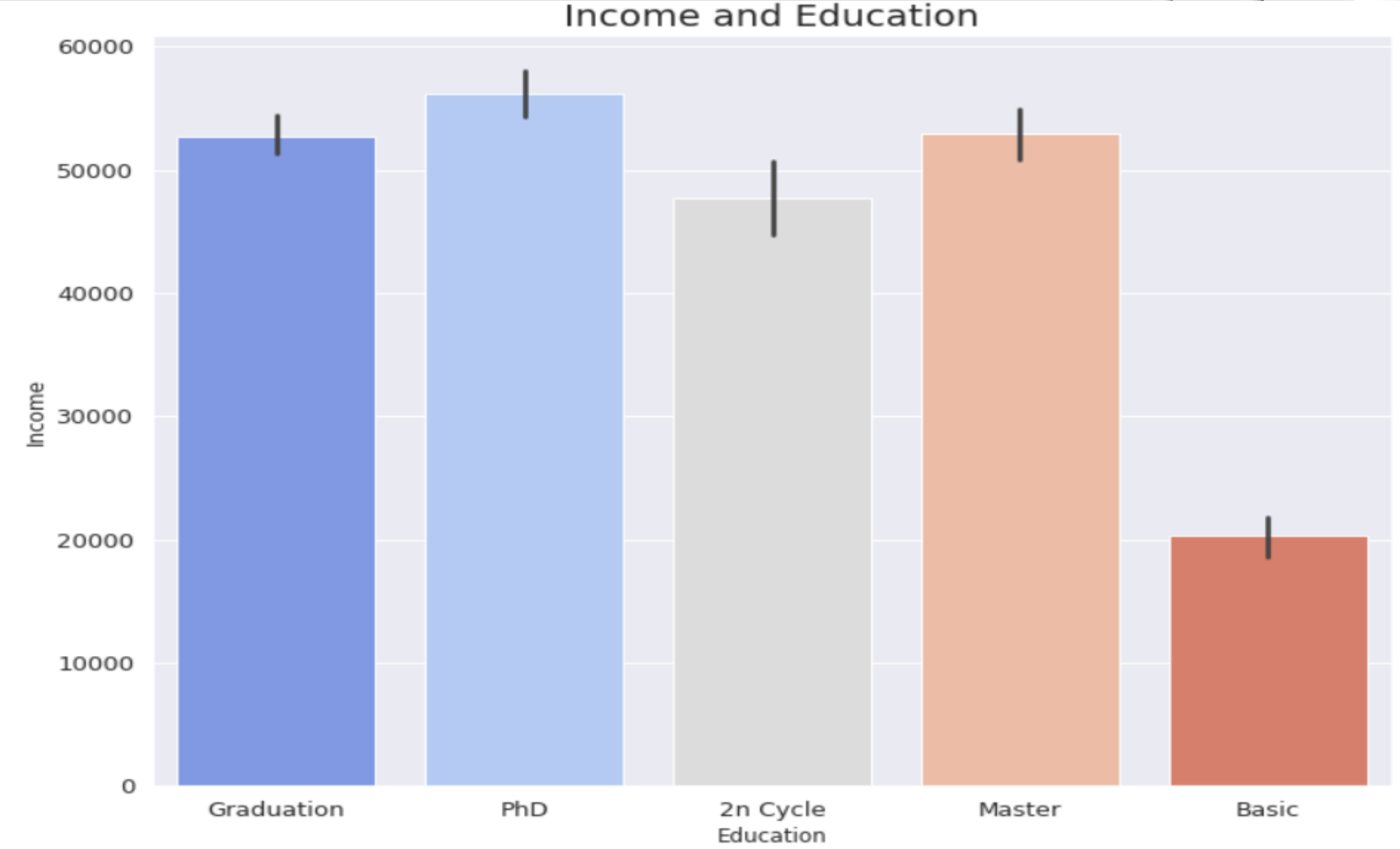
```
# one-hot encoding of categorical features
from sklearn.preprocessing import OneHotEncoder
# get categorical features and review number of unique values
cat = data.select_dtypes(exclude=np.number)
print("Number of unique values per categorical feature:\n", cat.nunique())
# use one hot encoder
enc = OneHotEncoder(sparse=False).fit(cat)
cat_encoded = pd.DataFrame(enc.transform(cat))
cat_encoded.columns = enc.get_feature_names(cat.columns)
# merge with numeric data
num = data.drop(columns=cat.columns)
data = pd.concat([cat_encoded, num], axis=1)
data.head()
```

```
Number of unique values per categorical feature:
 Education         5
Marital_Status    8
Country           8
dtype: int64
```

VISUALIZATION

Average Income W.R.T Education:

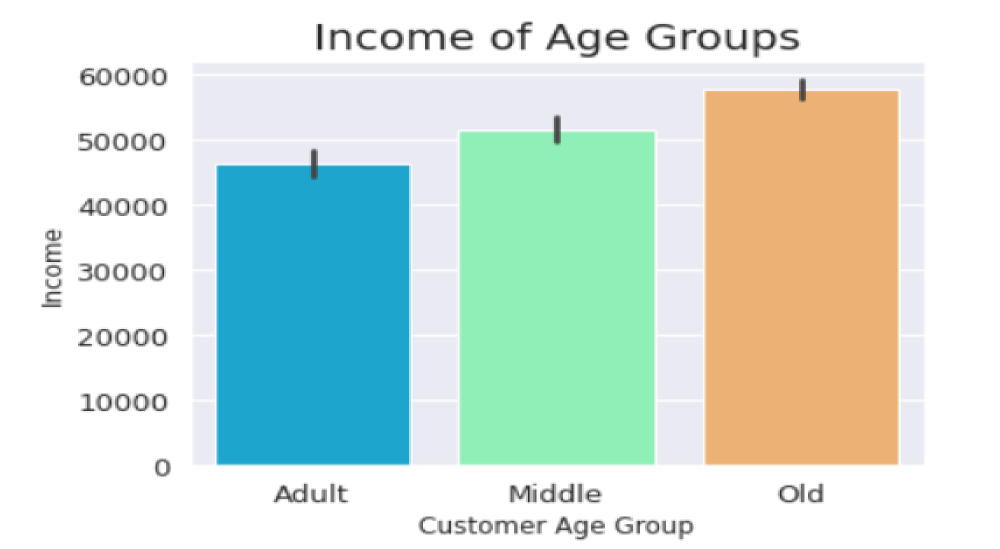
```
plt.figure(figsize=(12,9))

sns.barplot(x='Education', y='Income', data=data, estimator=np.mean, palette='coolwarm')
plt.title('Income and Education', fontsize=20)
plt.xlabel('Education',fontsize=12)
plt.ylabel('Income',fontsize=12);
```



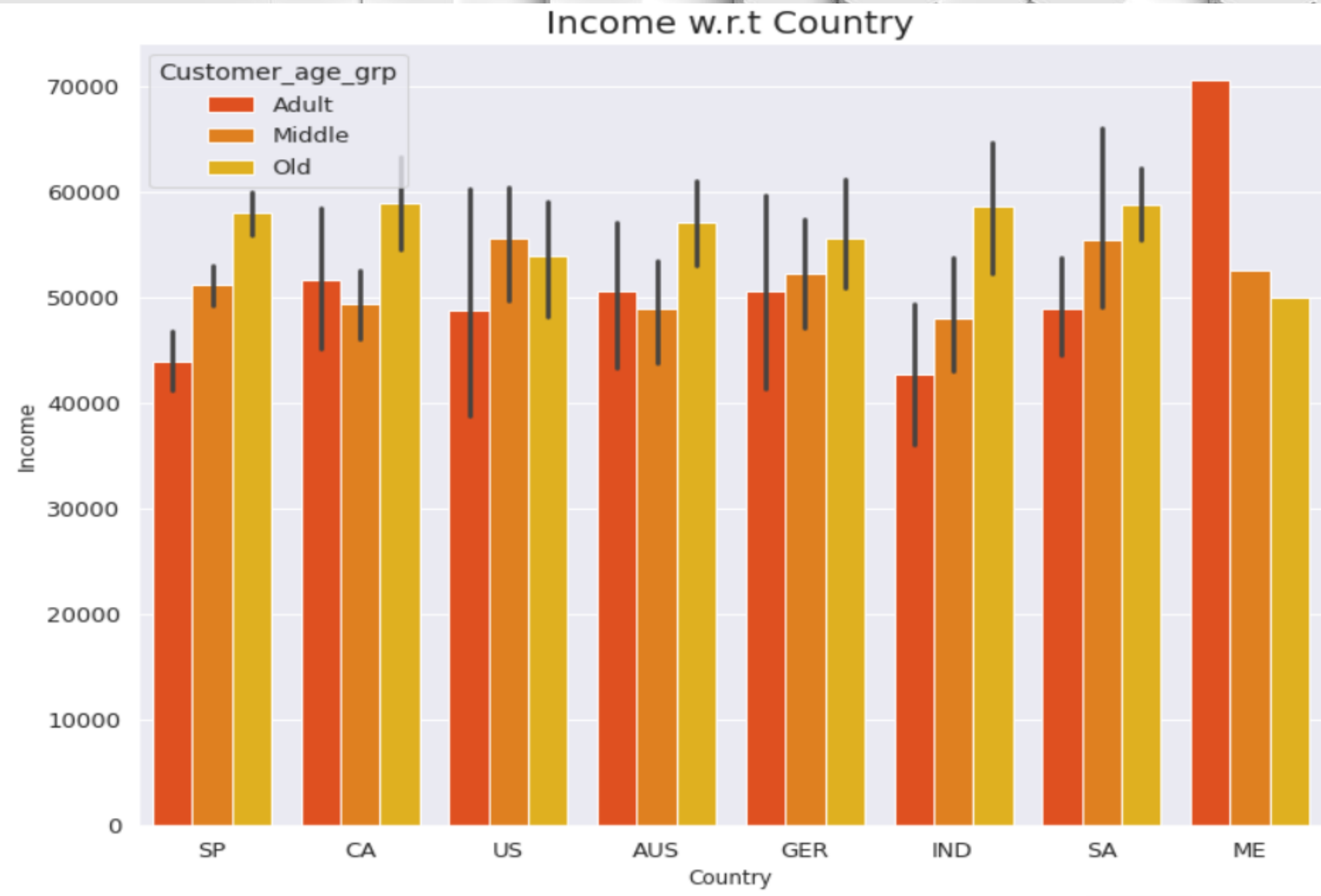
- People who have done graduation or higher have **better Income** than those who did not.

CATEGORIZATION OF INCOME BY DIFFERENT AGE GROUP AND COUNTRY

```
[ ]  data['Customer_age_grp'] = pd.cut(data.Customer_Age, bins=[18,35,50,80], labels=['Adult','Middle','Old'])
```

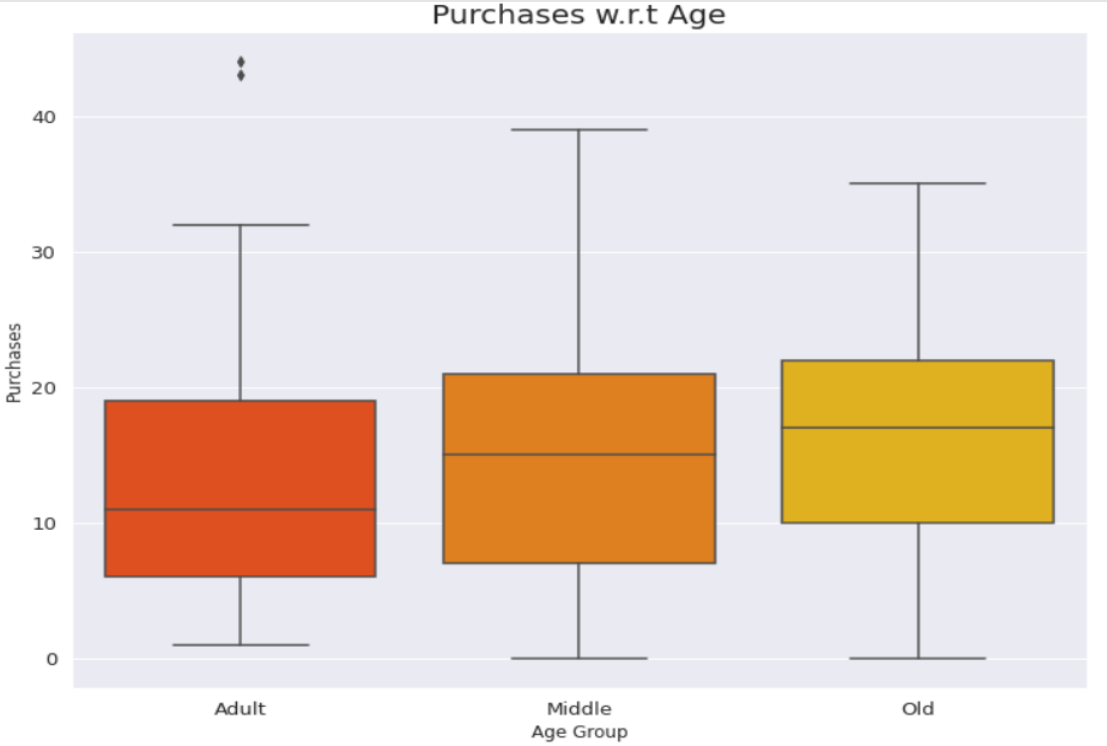


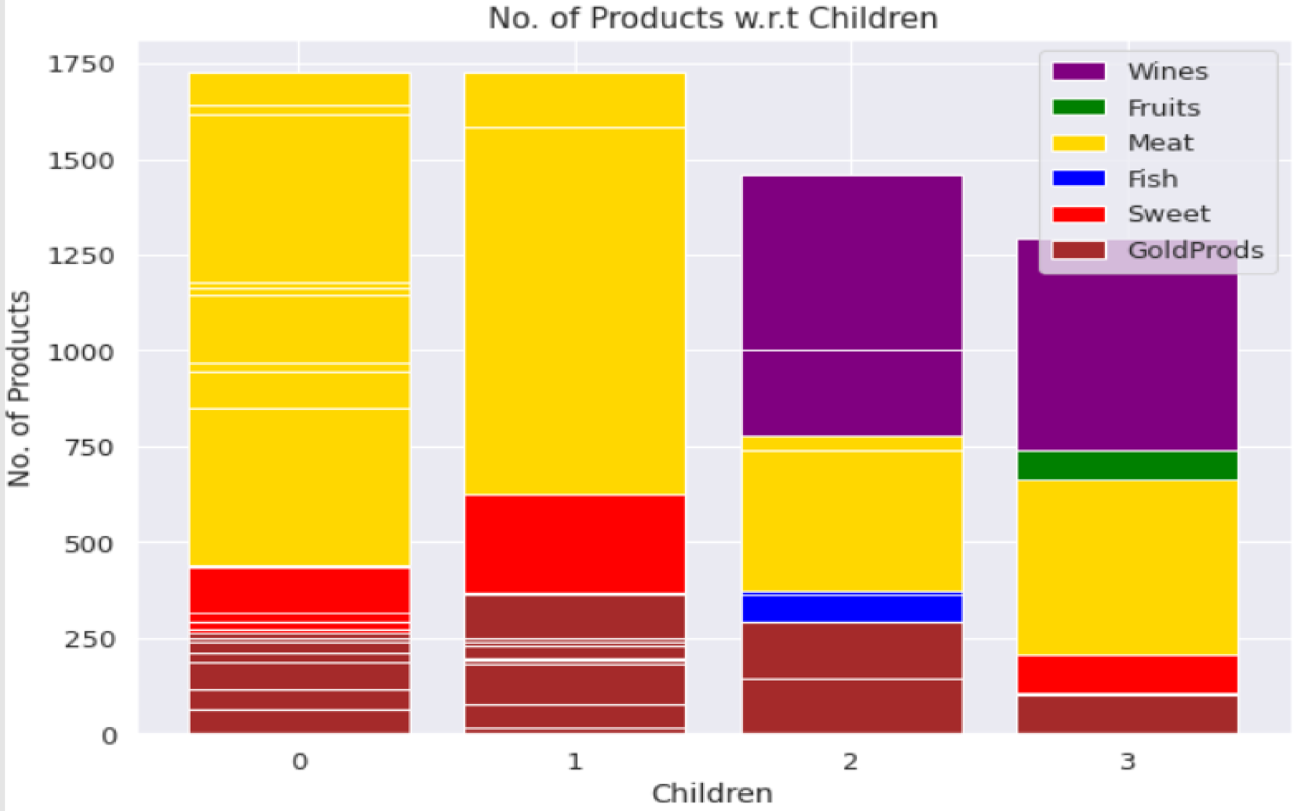
- ADULT: 18-35
- MIDDLE: 35- 50
- OLD: 50- 80



- Graph one and two illustrates that **old age group**, especially in countries such as Spain, Canada, Australia, Germany, India and south Africa have the **highest income**.
- According to graph two, **Mexico** have the **highest** where as **India** has the **lowest** income among **adults**.

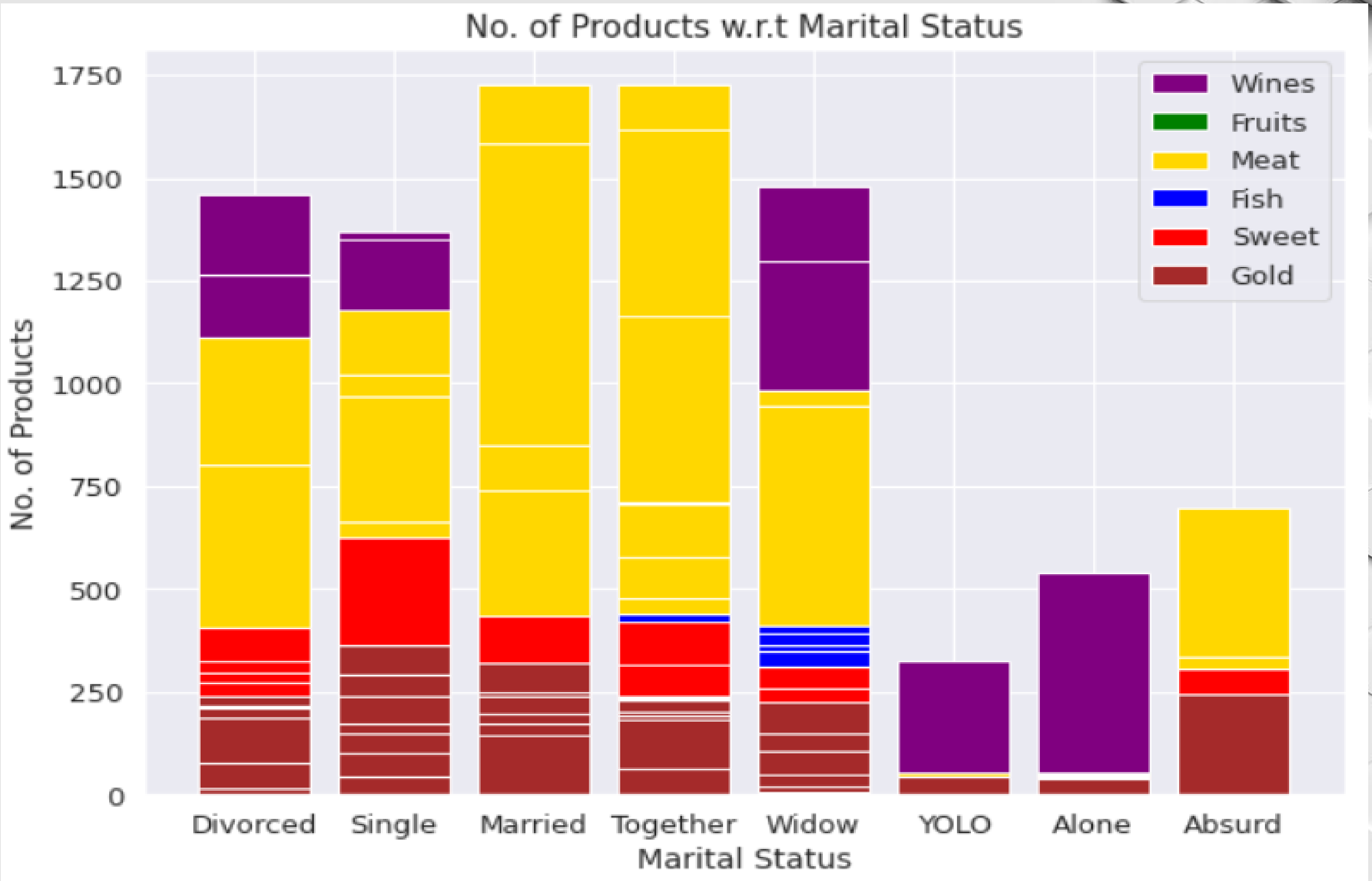
PURCHASE W.R.T AGE GROUP AND CHILDREN:





- People with 0 or 1 child tends to buy more **Meat** where as people with 2 to 3 kids buy more **Wine**.

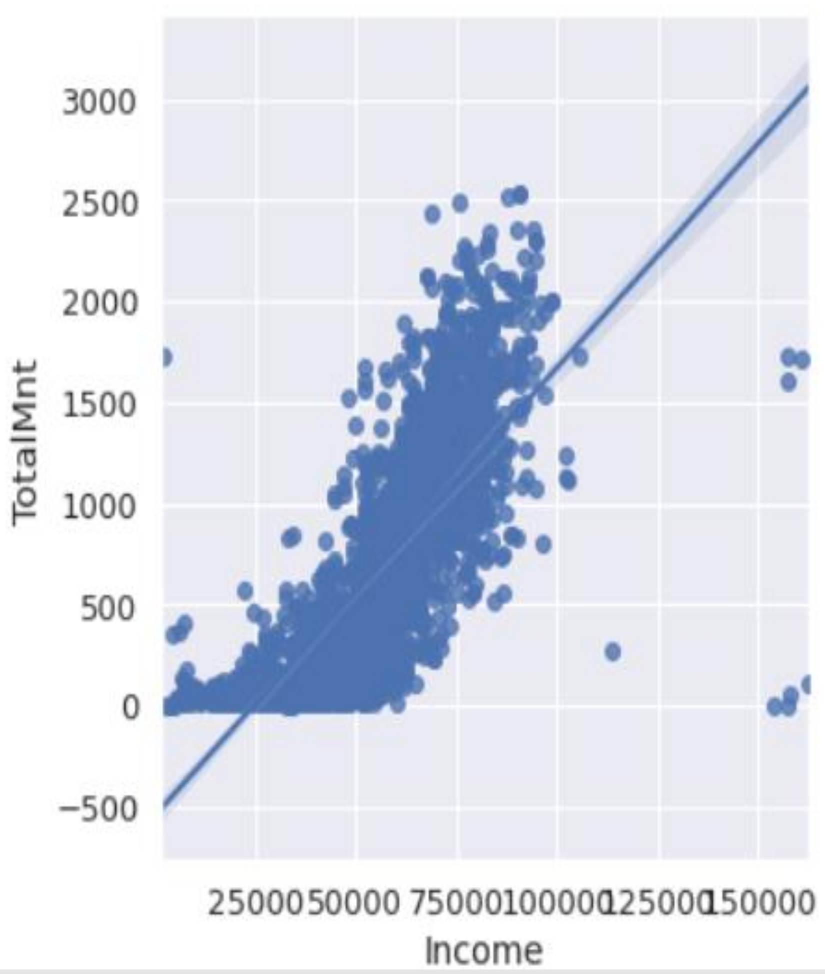
PRODUCT PURCHASED W.R.T MARITAL STATUS:



- Married people or people living together tends to purchase most amount of **Meat.**
- **Wine** is the second most purchased product.

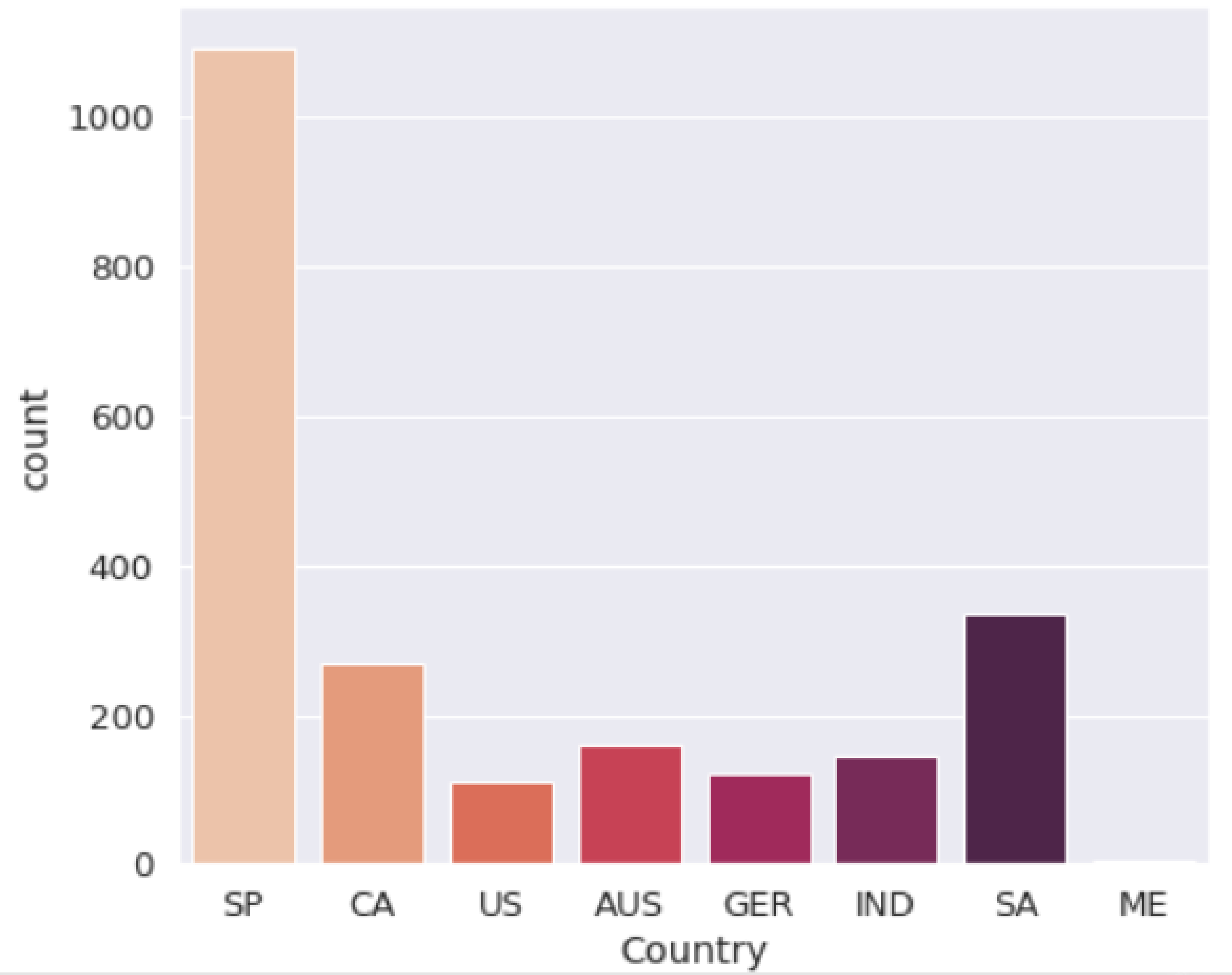
TOTAL EXPENDITURE OF CUSTOMERS:

```
sns.lmplot(x='Income', y='TotalMnt', data=data[data['Income'] < 200000]);
```



- The line-graph represents the total amount **spent** by customers according to their income.
- People with **high** income often spend **more** with some exceptions.
- To **remove outliers** we have limiting income to less than 200000.

USERS IN PARTICULAR COUNTRY:




```
[ ] data["Country"].value_counts().to_frame()
```

	Country
SP	1094
SA	336
CA	268
AUS	160
IND	147
GER	120
US	109
ME	3

- Spain has the **most** amount of Customers.
- And Mexico has the **least** amount of Customers.

WHICH COUNTRY SPENDS THE MOST?

→People in **Spain** purchase the most among all other countries.

```
[ ] data['Total Purchases'].sum()
```

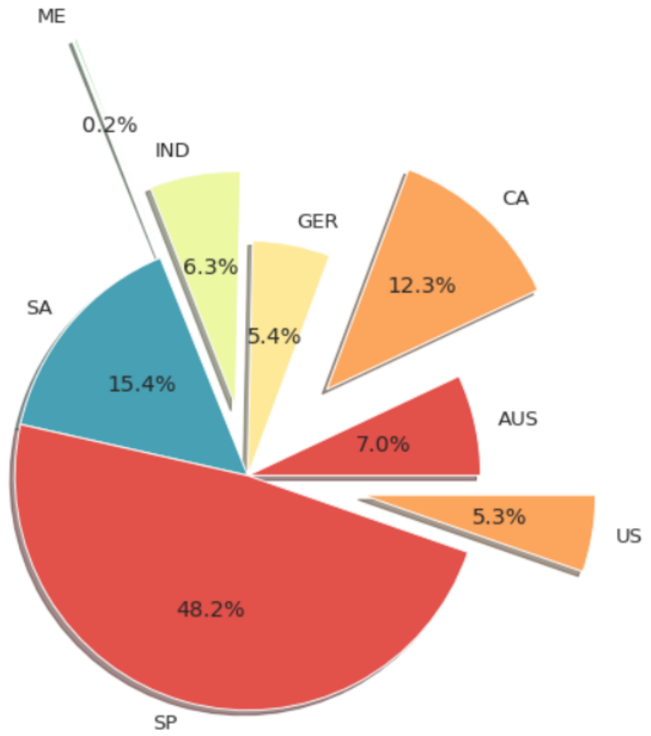
33266

```
[ ] data['Total Spent'].sum()
```

1355048

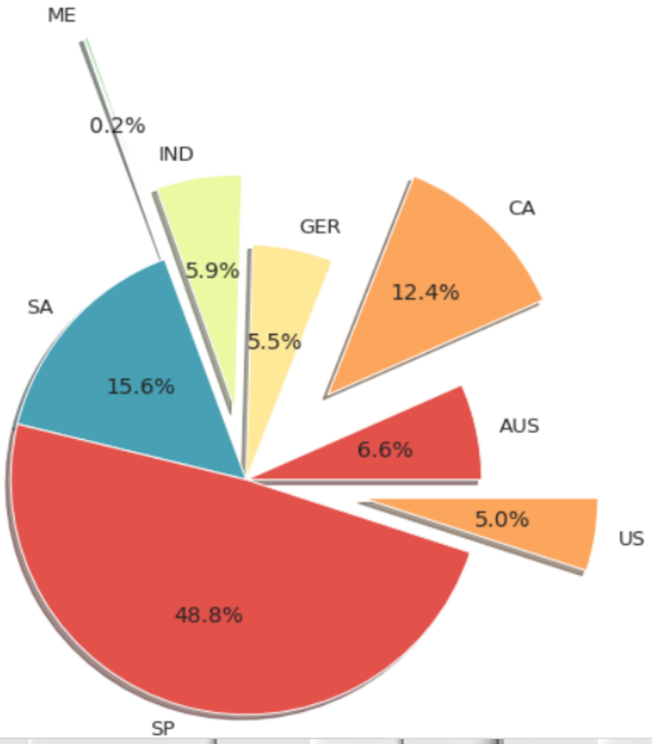
Total Purchase W.R.T Country:

```
[ ] pm = data[['Total Purchases', 'Country']].groupby(['Country']).agg([sum])

    sns.set_palette('Spectral')
    plt.figure(figsize = (7, 7))
    plt.pie(pm['Total Purchases']['sum'], labels = pm.index, explode = (0, 0.5, 0, 0.3, 1, 0, 0, 0.5),
            shadow = True, autopct = '%1.1f%%')
    plt.show()
```



Total Spent W.R.T Country:

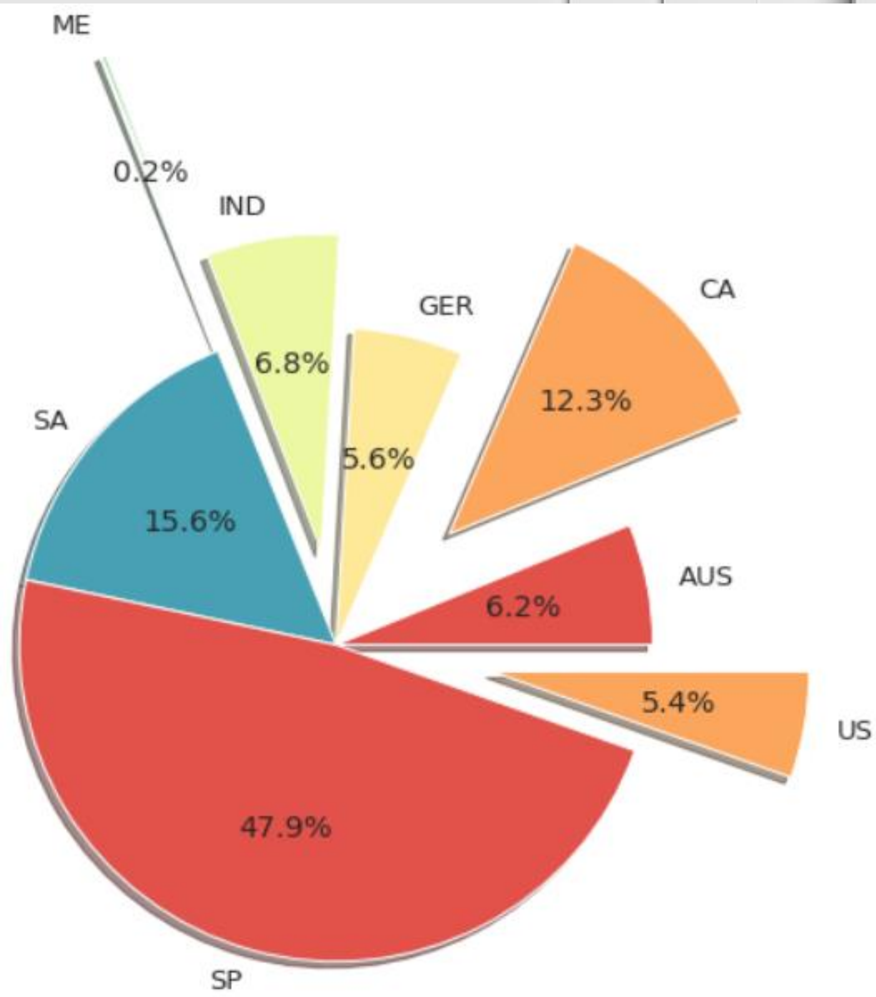
```
[ ] pm = data[['Total Spent', 'Country']].groupby(['Country']).agg([sum])

    sns.set_palette('Spectral')
    plt.figure(figsize = (7, 7))
    plt.pie(pm['Total Spent']['sum'], labels = pm.index, explode = (0, 0.5, 0, 0.3, 1, 0, 0, 0.5),
            shadow = True, autopct = '%1.1f%%')
    plt.show()
```



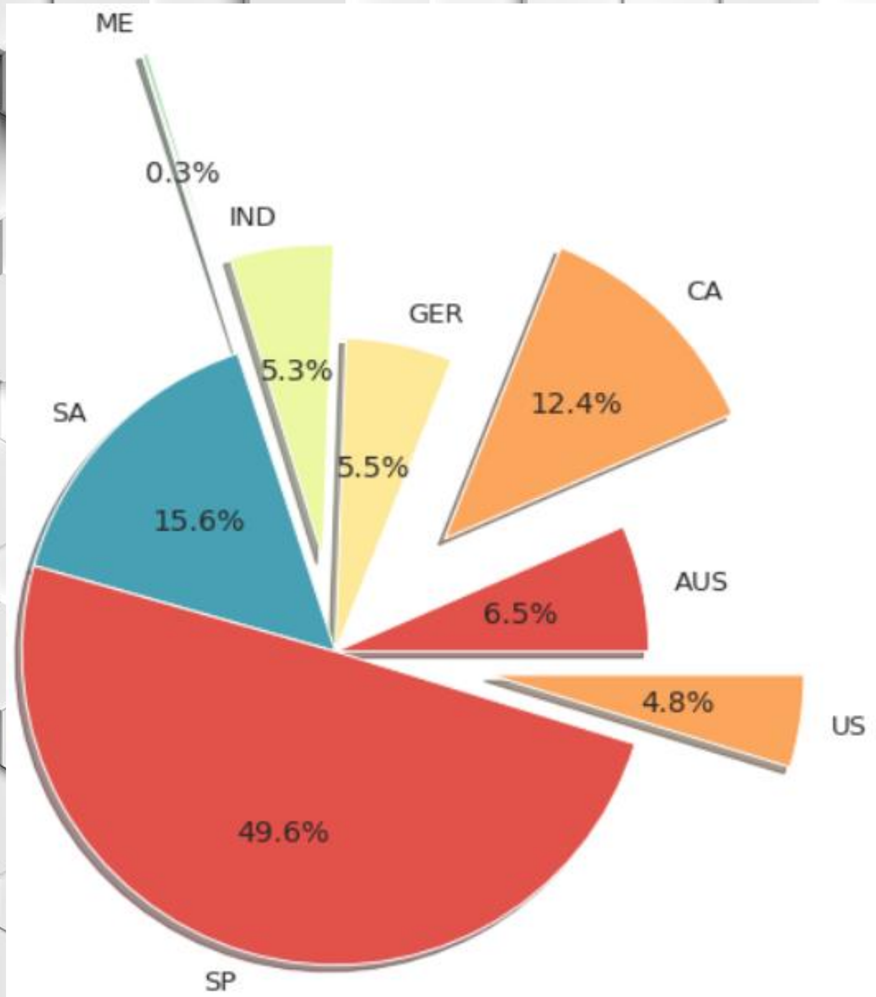
WHICH MARKETING CAMPAIGN IS MOST SUCCESSFUL?

→**Wine** and **Meat** seems to have done the best.

```
[ ] data["MntMeatProducts"].sum()

373968

[ ] data["MntGoldProds"].sum()

98609

[ ] data['MntSweetProducts'].sum()

60621

[ ] data["MntFishProducts"].sum()

84057

[ ] data["MntFruits"].sum()

58917

[ ] data["MntWines"].sum()

680816
```



• MEAT PRODUCT



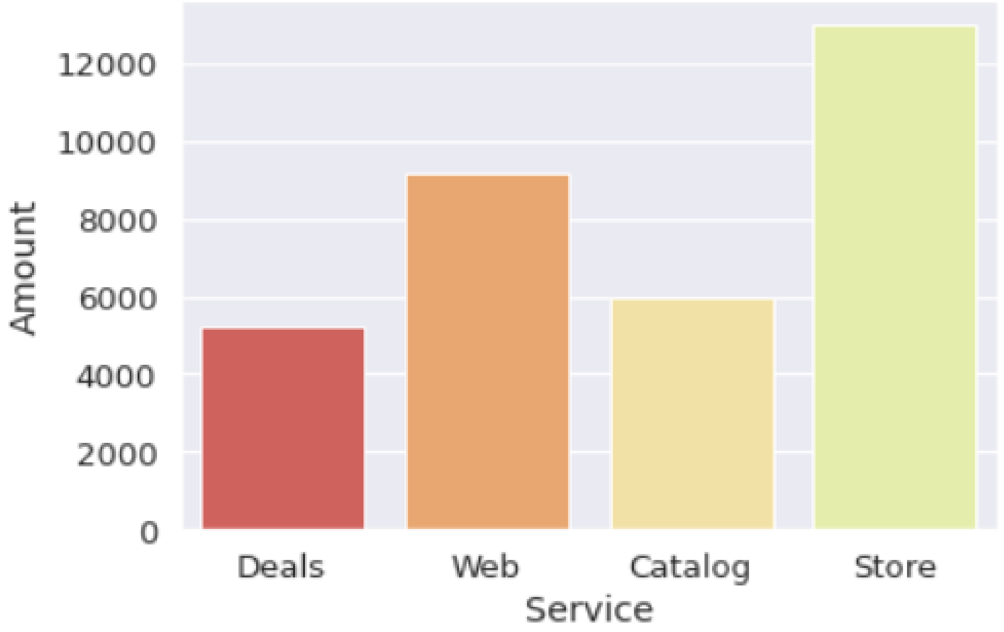
• WINE PRODUCT

HOW CUSTOMERS PREFER TO PURCHASE PRODUCTS?

→People prefer to purchase products from **Stores.**

```
[ ] prod = data[['NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases']].agg([sum]).T

    sns.barplot(x = prod.index, y = prod['sum'])

    plt.gca().set_xticklabels(['Deals', 'Web', 'Catalog', 'Store'])

    plt.xlabel('Service')

    plt.ylabel('Amount')
```

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



SUMMARY

- We can conclude that the most successful products are **wine** and **meat**.
- Advertising campaign acceptance is positively correlated with **income** and negatively correlated with having **children**.
- The new campaign was the most successful advertising campaign and was especially successful in **Spain**.
- The best performing channels are **web** and **store** purchases.
- The underperforming channels are deals and catalog purchases.
- Despite **Mexico** having the highest youth income, they spend the least because company has least users from Mexico.

THANK YOU!