PROJECT INTRODUCTION

Customer Personality Analysis is a detailed analysis of a company's ideal customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviors and concerns of different types of customers.

Customer personality analysis helps a business to modify its product based on its target customers from different types of customer segments. For example, instead of spending money to market a new product to every customer in the company's database, a company can analyze which customer segment is most likely to buy the product and then market the product only on that particular segment.

ANALYSIS OBJECTIVES

Here are the list of question whose analysis which be given below.

1. Data Analysis using statistical methods.
2. Data Analysis using conditional filtering when Marital_Status is Single.
3. Data Analysis using conditional filtering when Education is PhD.
4. Data Analysis by grouping the data on the basis of Education.
5. Data Analysis using sorting the data in ascending/descending order.
6. Data Analysis after sorting the data on the basis of ID, filtering ony PhD students and group by on their marital status.
7. Data Analysis after sorting the data on the basis of ID, filtering ony Single and group by on their Education status.
8. Visualize data of expenditure on the basis of Education using a chart with proper headings and legends.
9. Visualize data of wine expenditure on the basis of Education using a chart with proper headings and legends.
10. Visualize data of Fruits expenditure on the basis of Education using a chart with proper headings and legends.

DATA ACQUISITION AND CLEANING

Code to read the data from Excel / CSV / HTML.

To read the dataset in xlsx format, we will load it into Pandas data frame but first let's import the pandas library and set an alias by typing **"import pandas as pd"**. After importing the library with the alias **"pd"**, let us load the .csv file using the following line of code:

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import numpy as np
        import warnings
        warnings.simplefilter(action='ignore', category=FutureWarning)
        import plotly.graph_objects as go

In [2]: data = pd.read_csv("marketing_campaign.csv", sep='\t')

In [3]: data.head()

Out[3]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	Acc
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	5	0	

5 rows × 29 columns

```
In [4]: data.info()
```

Here we have import our csv files and read through pandas library.

Here the xlsx file can be read through (Pandas library) and store in **data Dataframe**. The Dataframe can be shown through **.head()**. The number of rows we want to show, that number we have to pass in head parentheses as an argument.

Now if we want to describe our dataframe for our better understanding to know the stats. and other parameter that our dataset should follow

Run Code

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64
```

In [5]:
```
data.head()
```

Out[5]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	Acc
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	5	0	

5 rows × 29 columns

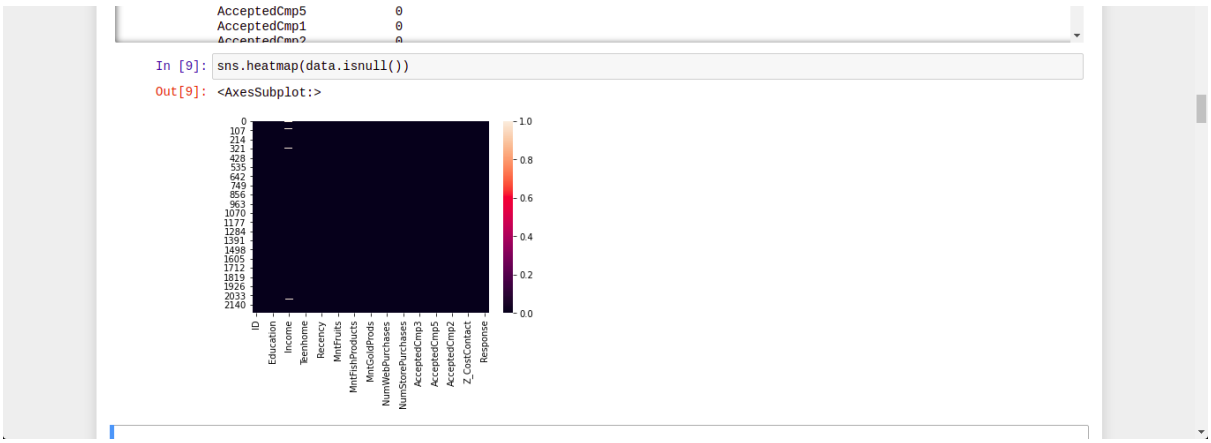
In [6]:
```
print(data.columns)
print(len(data.columns))
```

```
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', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
      dtype='object')
29
```

Clean the unnecessary data, by removing, replace the missing data and renaming the columns.

Dataset generally contains some null value, which is generally caused by misplacing some values. So its necessary to clean this mess from our dataset for better visualization

```
In [8]: data.isnull().sum()
Kidhome              0
Teenhome             0
Dt_Customer          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

In [9]: sns.heatmap(data.isnull())
Out[9]: <AxesSubplot:>
```



In the above diagram we see there is no columns of this dataset containing null value.

DATA AND EXPLORATORY ANALYSIS

Code and its output with Explanation

1. **Data Analysis using statistical methods.**

This data gives the estimated amount to various statistical methods that can be applied.

Analysis part begins from here

```
In [7]: #  1. Data Analysis using statistical methods.
```

```
In [10]: data.describe()
```

Out[10]:

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	...	Nu
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	...	
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	49.109375	303.935714	26.302232	166.950000	37.525446	...	
std	3246.662198	11.984069	25173.076661	0.538398	0.544538	28.962453	336.597393	39.773434	225.715373	54.628979	...	
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
25%	2828.250000	1959.000000	35303.000000	0.000000	0.000000	24.000000	23.750000	1.000000	16.000000	3.000000	...	
50%	5458.500000	1970.000000	51381.500000	0.000000	0.000000	49.000000	173.500000	8.000000	67.000000	12.000000	...	
75%	8427.750000	1977.000000	68522.000000	1.000000	1.000000	74.000000	504.250000	33.000000	232.000000	50.000000	...	
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	1725.000000	259.000000	...	

8 rows × 26 columns

```
In [11]: # 2,3. Data Analysis using conditional filtering with more than one condition.
```

2. **Data Analysis using conditional filtering when Marital_Status is Single.**

```
In [11]: # 2,3. Data Analysis using conditional filtering with more than one condition.
```

```
In [12]: data['Marital_Status'].unique()
```

```
Out[12]: array(['Single', 'Together', 'Married', 'Divorced', 'Widow', 'Alone',
                'Absurd', 'YOLO'], dtype=object)
```

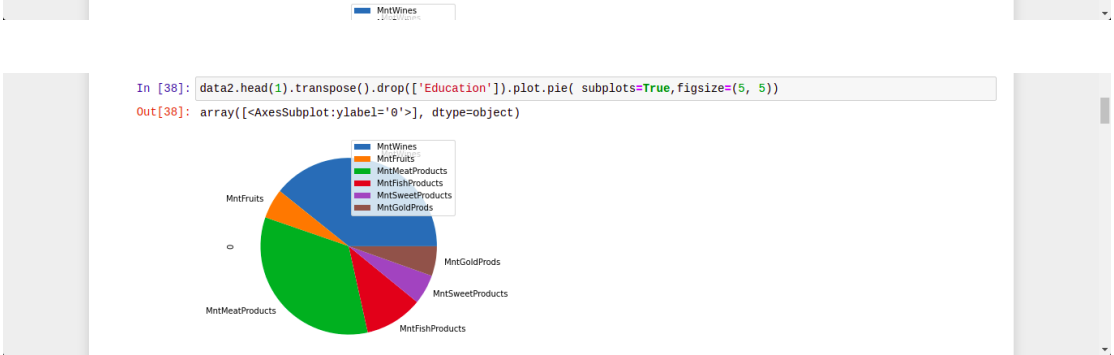
```
In [36]: data2 = data[data['Marital_Status']=='Single'][['Education', 'MntWines', 'MntFruits',
                'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
                'MntGoldProds']].reset_index(drop = True)
         data2.head()
```

Out[36]:

	Education	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	Graduation	635	88	546	172	88	88
1	Graduation	11	1	6	2	1	6
2	PhD	1006	22	115	59	68	45
3	2n Cycle	4	17	19	30	24	39
4	Graduation	6	4	25	15	12	13

```
In [38]: data2.head(1).transpose().drop(['Education']).plot.pie( subplots=True,figsize=(5, 5))
```

```
Out[38]: array([<AxesSubplot:ylabel='0'>], dtype=object)
```



3. **Data Analysis using conditional filtering when Education is PhD.**

In [39]:
```
data['Education'].unique()
```

Out[39]:
```
array(['Graduation', 'PhD', 'Master', 'Basic', '2n Cycle'], dtype=object)
```

In [40]:
```
data3 = data[data['Education']=='PhD'][['ID', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds']].reset_index(drop = True)
data3.head()
```

Out[40]:

	ID	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	5324	173	43	118	46	27	15
1	6177	76	10	56	3	1	23
2	4855	14	0	24	3	3	2
3	5899	28	0	6	1	1	13
4	2114	1006	22	115	59	68	45

4. **Data Analysis by grouping the data on the basis of Education.**

In [18]:
```
# 4. Data Analysis using group the data by column(s)
```

In [19]:
```
data_groupby = data.groupby('Education')['MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds'].sum().reset_index()
```

In [20]:
```
data_groupby
```

Out[20]:

	Education	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	2n Cycle	40231	5878	28675	9639	6953	9419
1	Basic	391	600	618	921	654	1233
2	Graduation	320371	34683	202284	48630	35351	57307
3	Master	123238	8012	60450	11877	7835	14947
4	PhD	196585	9744	81941	12990	9828	15703

5. **Data Analysis using sorting the data in ascending/descending order.**

In [21]:
```
# 5: Data Analysis using sorting the data in ascending/descending order.
```

In [22]:
```
data_5 = data.sort_values(by=['ID'], ascending=True).reset_index(drop = True)
data_5
```

Out[22]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3
0	0	1985	Graduation	Married	70951.0	0	0	04-05-2013	66	239	...	1	0
1	1	1961	Graduation	Single	57091.0	0	0	15-06-2014	0	464	...	5	0
2	9	1975	Master	Single	46098.0	1	1	18-08-2012	86	57	...	8	0
3	13	1947	PhD	Widow	25358.0	0	1	22-07-2013	57	19	...	6	0
4	17	1971	PhD	Married	60491.0	0	1	06-09-2013	81	637	...	5	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2235	11178	1972	Master	Single	42394.0	1	0	23-03-2014	69	15	...	7	0
2236	11181	1949	PhD	Married	156924.0	0	0	29-08-2013	85	2	...	0	0
2237	11187	1978	Basic	Single	26487.0	1	0	20-05-2013	23	2	...	5	0
2238	11188	1957	Graduation	Together	26091.0	1	1	25-02-2014	84	15	...	5	0
2239	11191	1986	Graduation	Divorced	41411.0	0	0	07-12-2013	11	37	...	6	0

2240 rows × 29 columns

6. **Data Analysis after sorting the data on the basis of ID, filtering ony PhD students and group by on their marital status.**

In [23]:
```
# 6,7: Data Analysis using combination of sorting, condition filter and/or grouping.
```

In [24]:
```
data_6 = data.sort_values(by=['ID'], ascending=True).reset_index(drop = True)
data_6 = data_6[data_6['Education']=='PhD']
data_6_groupby = data_6.groupby('Marital_Status')['MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds'].sum().reset_index()
data_6_groupby.head()
```

Out[24]:

	Marital_Status	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	Alone	15	0	8	4	2	20
1	Divorced	20366	1117	6911	1794	715	2006
2	Married	82663	4194	35720	5187	3829	5846
3	Single	35469	1777	15301	2417	2585	2854
4	Together	45764	1891	19920	2713	2050	3661

7. **Data Analysis after sorting the data on the basis of ID, filtering ony Single and group by on their Education status.**

In [25]:
```
data_7 = data.sort_values(by=['ID'], ascending=True).reset_index(drop = True)
data_7 = data_7[data_7['Marital_Status']=='Single']
data_7_groupby = data_7.groupby('Education')['MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds'].sum().reset_index()
data_7_groupby.head()
```

Out[25]:

	Education	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	2n Cycle	7768	1152	6564	1463	1191	1809
1	Basic	56	165	137	176	153	352
2	Graduation	66342	8244	47530	11363	7404	12887
3	Master	28764	1543	17880	2925	1753	3088
4	PhD	35469	1777	15301	2417	2585	2854

DATA ANALYSIS – VISUALIZATION

Code and its output with vizualization

1. **Visualize data of expenditure on the basis of Education using a chart with proper headings and legends.**

Code:-

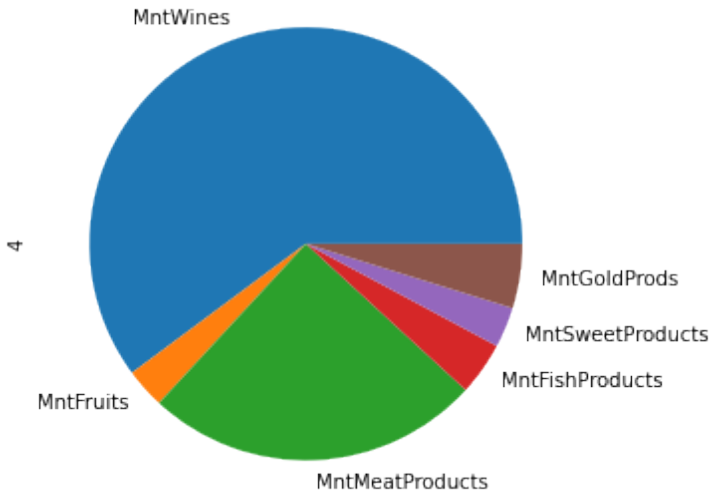
```
In [26]: # 8,9,10: Visualize data using a chart with proper headings and legends

In [27]: # Percentage share of expenditure of PhD grads

In [28]: data_8_groupby = data.groupby('Education')['MntWines', 'MntFruits',
             'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
             'MntGoldProds'].sum().reset_index()
         data_8_groupby
```

Out[28]:

	Education	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	2n Cycle	40231	5878	28675	9639	6953	9419
1	Basic	391	600	618	921	654	1233
2	Graduation	320371	34683	202284	48630	35351	57307
3	Master	123238	8012	60450	11877	7835	14947
4	PhD	196585	9744	81941	12990	9828	15703

```
In [29]: data_8_groupby.iloc[4].transpose().drop(['Education']).plot.pie( subplots=True,figsize=(5, 5))
Out[29]: array([<AxesSubplot:ylabel='4'>], dtype=object)
```

Output:-



2. Visualize data of wine expenditure on the basis of Education using a chart with proper headings and legends.

Code:-

```
In [30]: # Percentage share of expenditure on wine

In [31]: data_9_groupby = data.groupby('Education')['MntWines'].sum().reset_index()
         data_9_groupby
```

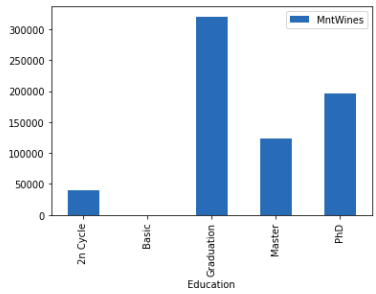
Out[31]:

	Education	MntWines
0	2n Cycle	40231
1	Basic	391
2	Graduation	320371
3	Master	123238
4	PhD	196585

```
In [32]: data_9_groupby.plot.bar(x='Education',y = 'MntWines')
Out[32]: <AxesSubplot:xlabel='Education'>
```

Output:-

```
In [32]: data_9_groupby.plot.bar(x='Education',y = 'MntWines')
Out[32]: <AxesSubplot:xlabel='Education'>
```




```
In [33]: # Percentage share of expenditure on fruit

In [34]: data_10_groupby = data.groupby('Education')['MntFruits'].sum().reset_index()
         data_10_groupby
```

3. **Visualize data of Fruits expenditure on the basis of Education using a chart with proper headings and legends.**

Code:-

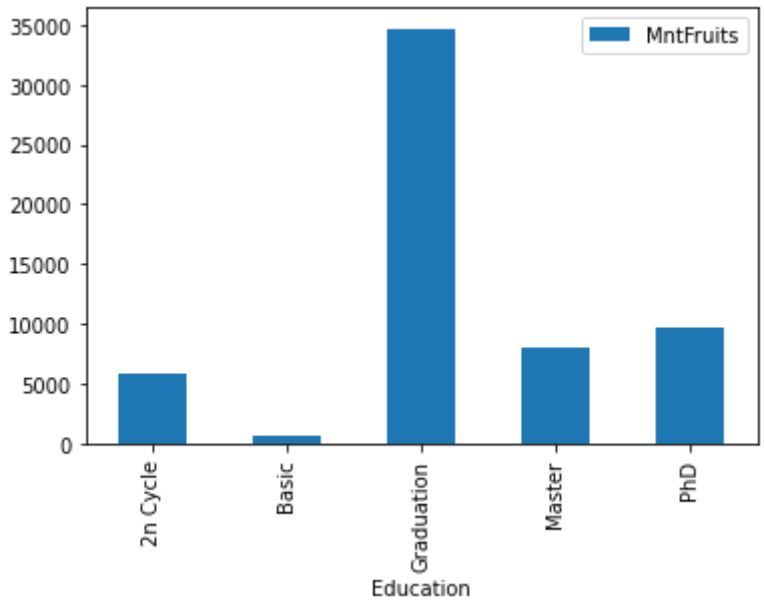
```
In [33]: # Percentage share of expenditure on fruit

In [34]: data_10_groupby = data.groupby('Education')['MntFruits'].sum().reset_index()
         data_10_groupby
```

Out[34]:

	Education	MntFruits
0	2n Cycle	5878
1	Basic	600
2	Graduation	34683
3	Master	8012
4	PhD	9744

```
In [35]: data_10_groupby.plot.bar(x='Education',y = 'MntFruits')
Out[35]: <AxesSubplot:xlabel='Education'>
```

Output:-



EXECUTIVE SUMMARY

CONCLUSION

From the above observations, we concluded that the biggest customers of wines are:

1. Customers with an average income of around $69,500.
2. Customers with an average total spend of approximately $1,252.
3. Customers registered with the company for approximately 21 months.
4. Customers with a graduate degree.
5. And customers who are also heavy consumers of meat products.

REFERENCES

- https://pandas.pydata.org/docs/
- https://www.kaggle.com/imakash3011/customer-personality-analysis
- https://www.w3schools.com/python/pandas/pandas_plotting.asp