```
# importing common libraries for data analysis
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# to see all the columns
pd.set_option("display.max_columns", None)
```

```
# importind market campaign data
df = pd.read_csv("/content/marketing_campaign_22-24.csv")
df.head()
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	NumD
0	5524	1967	Graduation	Single	58138.0	0	0	2022-09-04	58	635	88	546	172	88	88	
1	2174	1964	Graduation	Single	46344.0	1	1	2024-03-08	38	11	1	6	2	1	6	
2	4141	1975	Graduation	Together	71613.0	0	0	2023-08-21	26	426	49	127	111	21	42	
3	6182	1994	Graduation	Together	26646.0	1	0	2024-02-10	26	11	4	20	10	3	5	
4	5324	1991	PhD	Married	58293.0	1	0	2024-01-19	94	173	43	118	46	27	15	

1. Basic Exploration & fixing

```
# Shapeof our dataset (rows , columns)
df.shape
```

```
(2240, 29)
```

```
# Here we checking data types are in correct format or not.
df.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
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Complain             2240 non-null   int64
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

```
# Here in Dt_Customer dtype from obj -> Dateformat
df["Dt_Customer"] = pd.to_datetime(df["Dt_Customer"])
```

```
# Dtype change successfully
df[["Dt_Customer"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 1 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Dt_Customer  2240 non-null   datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 17.6 KB
```

```
# There are no duplicate values
df.duplicated().sum()
```

```
np.int64(0)
```

```
# Checking distribution or any mis spell values in object columns
for i in df.select_dtypes(include = "object"):
  print(i, df[i].nunique())
  print(df[i].value_counts())
  print()
```

```
Education 5
Education
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: count, dtype: int64

Marital_Status 8
Marital_Status
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: count, dtype: int64
```

```
# Drop rows with 'Alone', 'YOLO', or 'Absurd' in 'Marital_Status'
df = df[~df['Marital_Status'].isin(['Alone', 'YOLO', 'Absurd'])]
```

```
print(df['Marital_Status'].value_counts())
```

```
Marital_Status
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Name: count, dtype: int64
```

```
df.describe(percentiles=[0.01,0.99 ]).T
```

	count	mean	min	1%	50%	99%	max	std
ID	2233.0	5595.398567	0.0	130.76	5462.0	11073.04	11191.0	3243.198115
Year_Birth	2233.0	1978.790864	1903.0	1955.0	1980.0	2002.0	2006.0	11.979248
Income	2209.0	52243.977818	1730.0	7542.24	51390.0	94464.96	666666.0	25198.475858
Kidhome	2233.0	0.444245	0.0	0.0	0.0	2.0	2.0	0.538528
Teenhome	2233.0	0.506046	0.0	0.0	0.0	2.0	2.0	0.544675
Dt_Customer	2233	2023-07-10 14:46:41.970443264	2022-07-30 00:00:00	2022-08-07 00:00:00	2023-07-09 00:00:00	2024-06-22 00:00:00	2024-06-29 00:00:00	NaN
Recency	2233.0	49.172414	0.0	0.0	50.0	98.0	99.0	28.962114
MntWines	2233.0	304.033587	0.0	1.0	173.0	1285.0	1493.0	336.953019
MntFruits	2233.0	26.30094	0.0	0.0	8.0	172.0	199.0	39.779532
MntMeatProducts	2233.0	167.1133	0.0	2.0	67.0	915.0	1725.0	225.870588
MntFishProducts	2233.0	37.445141	0.0	0.0	12.0	226.36	259.0	54.461569
MntSweetProducts	2233.0	27.108374	0.0	0.0	8.0	177.36	263.0	41.3217
MntGoldProds	2233.0	43.903269	0.0	0.0	24.0	227.0	362.0	52.009602
NumDealsPurchases	2233.0	2.321093	0.0	0.0	2.0	10.0	15.0	1.932181
NumWebPurchases	2233.0	4.081505	0.0	0.0	4.0	11.0	27.0	2.776886
NumCatalogPurchases	2233.0	2.66189	0.0	0.0	2.0	10.68	28.0	2.921749
NumStorePurchases	2233.0	5.79176	0.0	2.0	5.0	13.0	13.0	3.25473
NumWebVisitsMonth	2233.0	5.316167	0.0	1.0	6.0	9.0	20.0	2.425668
AcceptedCmp3	2233.0	0.072548	0.0	0.0	0.0	1.0	1.0	0.259451
AcceptedCmp4	2233.0	0.074787	0.0	0.0	0.0	1.0	1.0	0.263107
AcceptedCmp5	2233.0	0.072548	0.0	0.0	0.0	1.0	1.0	0.259451
AcceptedCmp1	2233.0	0.064039	0.0	0.0	0.0	1.0	1.0	0.244878
AcceptedCmp2	2233.0	0.013435	0.0	0.0	0.0	1.0	1.0	0.115153
Complain	2233.0	0.009404	0.0	0.0	0.0	0.0	1.0	0.096541
Z_CostContact	2233.0	3.0	3.0	3.0	3.0	3.0	3.0	0.0
Z_Revenue	2233.0	11.0	11.0	11.0	11.0	11.0	11.0	0.0
Response	2233.0	0.148231	0.0	0.0	0.0	1.0	1.0	0.355408

```
# Checking null values in columns
df.isna().sum()
```

```
# Inocome colums have 24 null values
```

	0
ID	0
Year_Birth	0
Education	0
Marital_Status	0
Income	24
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
Complain	0
Z_CostContact	0
Z_Revenue	0
Response	0

dtype: int64

```
# displaying rows where income have null values
df[df['Income'].isnull()].head()
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	Num
10	1994	1993	Graduation	Married	NaN	1	0	2023-11-15	11	5	5	6	0	2	1	
27	5255	1996	Graduation	Single	NaN	1	0	2023-02-20	19	5	1	3	3	263	362	
43	7281	1969	PhD	Single	NaN	0	0	2023-11-05	80	81	11	50	3	2	39	
48	7244	1961	Graduation	Single	NaN	2	1	2024-01-01	96	48	5	48	6	10	7	
58	8557	1992	Graduation	Single	NaN	1	0	2023-06-17	57	11	3	22	2	2	6	

```
# Fill null values in 'Income' with the mean income grouped by 'Marital_Status', and 'Education'.
df["Income"] = df.groupby([ 'Marital_Status', 'Education'])["Income"].transform(lambda x : x.fillna(x.median()))
```

2. Feature Engineering

```
# Extracting year, month with name, month quarter,days and phase of month from "Dt_customer" (joining data of the customer)
df["Cust_enroll_year"] = df["Dt_Customer"].dt.year
df["Cust_enroll_month"] = df["Dt_Customer"].dt.month_name()
df["Cust_enroll_Quarter"] = df["Dt_Customer"].dt.quarter.map({1:"Q1", 2:"Q2", 3:"Q3", 4:"Q4"})
df["Cust_enroll_day"] = df["Dt_Customer"].dt.day_name()
df["Cust_enroll_month_Phase"] = df["Dt_Customer"].apply(lambda x : "Start" if 1 <= x.day <= 10 else ( "Middle" if 11 <= x.day <= 20 else "End"))


# Extracting age and joining age from df
from datetime import datetime
df["Age"] = datetime.today().year - df["Year_Birth"]
df["Joining_Age"] = df["Cust_enroll_year"] - df["Year_Birth"]
```

```
Start coding or generate with AI.
```