Assg 2

In [2]:
```
import pandas as pd
```

In [3]:
```
df_dataset = pd.read_csv('~/Desktop/Polaris/Ecommerce_marketing_campaign.csv')
```

In [4]:
```
df_dataset.dtypes
```

Out[4]:
```
ID                     int64
Year_Birth             int64
Education             object
Marital_Status        object
Income               float64
Kidhome                int64
Teenhome               int64
Dt_Customer           object
Recency                int64
MntA                   int64
MntB                   int64
MntC                   int64
MntD                   int64
MntE                   int64
MntF                   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
```

In [5]:
```
# 1. Display the first 20 rows of the dataframe
df_dataset.head(20)
```

Out[5]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntA	...	NumWebVisits
0	5524	1957	Graduation	Single	58138.0	0	0	9/4/2012	58	635	...	
1	2174	1954	Graduation	Single	46344.0	1	1	3/8/2014	38	11	...	
2	4141	1965	Graduation	Together	71613.0	0	0	8/21/2013	26	426	...	
3	6182	1984	Graduation	Together	26646.0	1	0	2/10/2014	26	11	...	
4	5324	1981	PhD	Married	58293.0	1	0	1/19/2014	94	173	...	
5	7446	1967	Master	Together	62513.0	0	1	9/9/2013	16	520	...	
6	965	1971	Graduation	Divorced	55635.0	0	1	11/13/2012	34	235	...	
7	6177	1985	PhD	Married	33454.0	1	0	5/8/2013	32	76	...	
8	4855	1974	PhD	Together	30351.0	1	0	6/6/2013	19	14	...	
9	5899	1950	PhD	Together	5648.0	1	1	3/13/2014	68	28	...	
10	1994	1983	Graduation	Married	NaN	1	0	11/15/2013	11	5	...	
11	387	1976	Basic	Married	7500.0	0	0	11/13/2012	59	6	...	
12	2125	1959	Graduation	Divorced	63033.0	0	0	11/15/2013	82	194	...	
13	8180	1952	Master	Divorced	59354.0	1	1	11/15/2013	53	233	...	
14	2569	1987	Graduation	Married	17323.0	0	0	10/10/2012	38	3	...	
15	2114	1946	PhD	Single	82800.0	0	0	11/24/2012	23	1006	...	
16	9736	1980	Graduation	Married	41850.0	1	1	12/24/2012	51	53	...	
17	4939	1946	Graduation	Together	37760.0	0	0	8/31/2012	20	84	...	
18	6565	1949	Master	Married	76995.0	0	1	3/28/2013	91	1012	...	
19	2278	1985	2n Cycle	Single	33812.0	1	0	11/3/2012	86	4	...	

20 rows × 29 columns

In [27]:
```
#2. Sort the dataframe by Year_birth on ascending order
df_sorted = df_dataset.sort_values(by='Year_Birth', ascending=True)
df_sorted
```

Out[27]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntA	...	NumWebVi
239	11004	1893	2n Cycle	Single	60182.0	0	1	5/17/2014	23	8	...	
339	1150	1899	PhD	Together	83532.0	0	0	9/26/2013	36	755	...	
192	7829	1900	2n Cycle	Divorced	36640.0	1	0	9/26/2013	99	15	...	
1950	6663	1940	PhD	Single	51141.0	0	0	7/8/2013	96	144	...	
424	6932	1941	PhD	Married	93027.0	0	0	4/13/2013	77	1285	...	
...	...	...	...	...	...	...	...	...	...	...	...	
747	10548	1995	Graduation	Single	71163.0	0	0	3/9/2014	30	283	...	
1850	4427	1995	2n Cycle	Single	83257.0	0	0	9/18/2012	56	536	...	
696	8315	1995	Graduation	Single	34824.0	0	0	3/26/2014	65	4	...	
1170	193	1996	Basic	Married	14421.0	0	0	2/17/2014	81	0	...	
46	9909	1996	2n Cycle	Married	7500.0	0	0	11/9/2012	24	3	...	

2240 rows × 29 columns

In [6]:
```
#3. Filter the dataframe where customers were born after 1985
df_sub = df_dataset[df_dataset['Year_Birth']>1985]
df_sub
```

Out[6]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntA	...	NumWebVis
14	2569	1987	Graduation	Married	17323.0	0	0	10/10/2012	38	3	...	
27	5255	1986	Graduation	Single	NaN	1	0	2/20/2013	19	5	...	
28	9422	1989	Graduation	Married	38360.0	1	0	5/31/2013	26	36	...	
30	6864	1989	Master	Divorced	10979.0	0	0	5/22/2014	34	8	...	
46	9909	1996	2n Cycle	Married	7500.0	0	0	11/9/2012	24	3	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2167	3520	1990	Master	Single	91172.0	0	0	3/27/2013	94	162	...	
2200	7620	1990	Basic	Single	16185.0	1	0	8/5/2013	71	5	...	
2204	1876	1990	Graduation	Married	18929.0	0	0	2/16/2013	15	32	...	
2213	3661	1995	2n Cycle	Single	80617.0	0	0	10/12/2012	42	594	...	
2232	8080	1986	Graduation	Single	26816.0	0	0	8/17/2012	50	5	...	

189 rows × 29 columns

In [30]:
```
#4. Count the number of customers was born after 1985
df_sub['ID'].count()
```

Out[30]:
```
189
```

In [7]:
```
#5. What is the average income for customers with different education level?
df_avg = df_dataset.groupby('Education')['Income'].mean()
df_avg
```

Out[7]:
```
Education
2n Cycle      47633.190000
Basic         20306.259259
Graduation    52720.373656
Master        52917.534247
PhD           56145.313929
Name: Income, dtype: float64
```

In [ ]:
```
# ANS of question 5:
# For customers who are educated in 2n Cycle, their average yearly household income is $47633.19
# For customers who have Basic education, their average yearly household income is $20306.26
# For customers who are Graduated, their average yearly household income is $52720.37
# For customers who are Master, their average yearly household income is $52917.53
# For customers who are PhD, their average yearly household income is $56145.31
```

In [8]:
```
#6. For those accept / reject the offer (reponse), what are their average spending in category A, B, C,
df_sub_1 = df_dataset.groupby('Response')['MntA','MntB','MntC','MntD','MntE','MntF'].mean()
df_sub_1
```

```
<ipython-input-8-67a12139e06a>:2: FutureWarning: Indexing with multiple keys (implicitly converted to a
tuple of keys) will be deprecated, use a list instead.
  df_sub_1 = df_dataset.groupby('Response')['MntA','MntB','MntC','MntD','MntE','MntF'].mean()
```

Out[8]:

Response	MntA	MntB	MntC	MntD	MntE	MntF
0	269.104407	24.216684	144.624344	34.980063	25.035152	40.968520
1	502.703593	38.203593	294.353293	52.050898	38.634731	61.446108

In [ ]:
```
# ANS of question 6:
# For those accept the offer (reponese=1), the required average spending on A, B, C, D, E and F products
# in the last 2 years are $502.70, $38.20, $294.35, $52.05, $38.63 and $61.45 respectively.
# For those reject the offer (reponese=0), the required average spending on A, B, C, D, E and F products
# in the last 2 years are $269.10, $24.22, $144.62, $34.98, $25.04 and $40.97 respectively.
```