In [1]:

```
import pandas as pd
import numpy as np
from datetime import date
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("marketing_data.csv")
data.drop("ID",axis=1,inplace = True)
```

In [2]:

```
data.columns
```

Out[2]:

```
Index(['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')
```

In [3]:

```
#Calculating age
def age(born):
    PDate=date.today()
    PYear= PDate.year
    return PYear - born
data['Age'] = data['Year_Birth'].apply(age)
data
```

Out[3]:

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Re
0	1970	Graduation	Divorced	$84,835.00	0	0	6/16/14	
1	1961	Graduation	Single	$57,091.00	0	0	6/15/14	
2	1958	Graduation	Married	$67,267.00	0	1	5/13/14	
3	1967	Graduation	Together	$32,474.00	1	1	5/11/14	
4	1989	Graduation	Single	$21,474.00	1	0	4/8/14	
...	...	...	...	...	...	...	...	
2235	1976	PhD	Divorced	$66,476.00	0	1	3/7/13	
2236	1977	2n Cycle	Married	$31,056.00	1	0	1/22/13	
2237	1976	Graduation	Divorced	$46,310.00	1	0	12/3/12	
2238	1978	Graduation	Married	$65,819.00	0	0	11/29/12	
2239	1969	PhD	Married	$94,871.00	0	2	9/1/12	

2240 rows × 28 columns

In [4]:

```
# Isolate the column titles into a list
column_titles = []
for i in data.columns:
    column_titles.append(i)
print(column_titles)
# Rename the 'Income' title
data = data.rename(columns={column_titles[3]:'Income'})
# Change the Income field data type to Float
data["Income"] = data["Income"].str.replace("$","").str.replace(",","")
data["Income"] = data["Income"].astype(float)
data
```

```
['Year_Birth', 'Education', 'Marital_Status', ' Income ', 'Kidhome', 'Teenho
me', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits', 'MntMeatProducts',
'MntFishProducts', 'MntSweetProducts', 'MntGoldProds', 'NumDealsPurchases',
'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisits
Month', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'Acc
eptedCmp2', 'Response', 'Complain', 'Country', 'Age']
```

Out[4]:

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recer
0	1970	Graduation	Divorced	84835.0	0	0	6/16/14	
1	1961	Graduation	Single	57091.0	0	0	6/15/14	
2	1958	Graduation	Married	67267.0	0	1	5/13/14	
3	1967	Graduation	Together	32474.0	1	1	5/11/14	
4	1989	Graduation	Single	21474.0	1	0	4/8/14	
...	...	...	...	...	...	...	...	
2235	1976	PhD	Divorced	66476.0	0	1	3/7/13	
2236	1977	2n Cycle	Married	31056.0	1	0	1/22/13	
2237	1976	Graduation	Divorced	46310.0	1	0	12/3/12	
2238	1978	Graduation	Married	65819.0	0	0	11/29/12	
2239	1969	PhD	Married	94871.0	0	2	9/1/12	

2240 rows × 28 columns

In [5]:

```
data.dtypes
#df.describe()
```

Out[5]:

```
Year_Birth               int64
Education               object
Marital_Status          object
Income                 float64
Kidhome                  int64
Teenhome                 int64
Dt_Customer             object
Recency                  int64
MntWines                 int64
MntFruits                int64
MntMeatProducts          int64
MntFishProducts          int64
MntSweetProducts         int64
MntGoldProds             int64
NumDealsPurchases        int64
NumWebPurchases          int64
NumCatalogPurchases      int64
NumStorePurchases        int64
```

In [6]:

```
# Count null values for each field
data.isnull().sum()
#df.describe()
```

Out[6]:

```
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
Response                0
Complain                0
Country                 0
Age                     0
dtype: int64
```

--------> There are 24 records with missing "Income" values

In [7]:

```
# Impute missing income values using the median income
data["Income"] = data["Income"].fillna(value=data["Income"].median())
```

In [8]:

```
# Make list of categorical variables
cat_var = ["Education", "Marital_Status", "Country"]
# Obtain all unique values for each categorical variable to identify errors
for i in cat_var:
    print(f"{i} Unique Values: {data[i].unique()}")
```

```
Education Unique Values: ['Graduation' 'PhD' '2n Cycle' 'Master' 'Basic']
Marital_Status Unique Values: ['Divorced' 'Single' 'Married' 'Together' 'Wid
ow' 'YOLO' 'Alone' 'Absurd']
Country Unique Values: ['SP' 'CA' 'US' 'AUS' 'GER' 'IND' 'SA' 'ME']
```

Categorical variables

```
1.The variables '2n cycle' and 'Master' have the same meaning. The '2n cycle' valu
es should be merged to equal 'Master'.
2.The values 'YOLO', 'Alone', and 'Absurd' all mean 'Single', so these values shou
ld be merged to equal 'Single'.
3.The 'Marital_Status' variable does not require changes
```

In [9]:

```
# Convert '2n Cycle' values to 'Master'
data["Education"] = data["Education"].replace(["2n Cycle"], value="Master")
# Convert 'YOLO', 'Alone', and 'Absurd' values to 'Single'
data["Marital_Status"] = data["Marital_Status"].replace(["YOLO", "Alone", "Absurd",'Divorce
data["Marital_Status"] = data["Marital_Status"].replace(['Together'], value="Married")
```

In [10]:

```
data
```

Out[10]:

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recer
0	1970	Graduation	Single	84835.0	0	0	6/16/14	
1	1961	Graduation	Single	57091.0	0	0	6/15/14	
2	1958	Graduation	Married	67267.0	0	1	5/13/14	
3	1967	Graduation	Married	32474.0	1	1	5/11/14	
4	1989	Graduation	Single	21474.0	1	0	4/8/14	
...	...	...	...	...	...	...	...	
2235	1976	PhD	Single	66476.0	0	1	3/7/13	
2236	1977	Master	Married	31056.0	1	0	1/22/13	
2237	1976	Graduation	Single	46310.0	1	0	12/3/12	
2238	1978	Graduation	Married	65819.0	0	0	11/29/12	
2239	1969	PhD	Married	94871.0	0	2	9/1/12	

2240 rows × 28 columns

In [11]:

```
# Group numerical variables into a new dataframe
num = ['Year_Birth','Income', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth']
data_num = data[num]
```

In [12]:

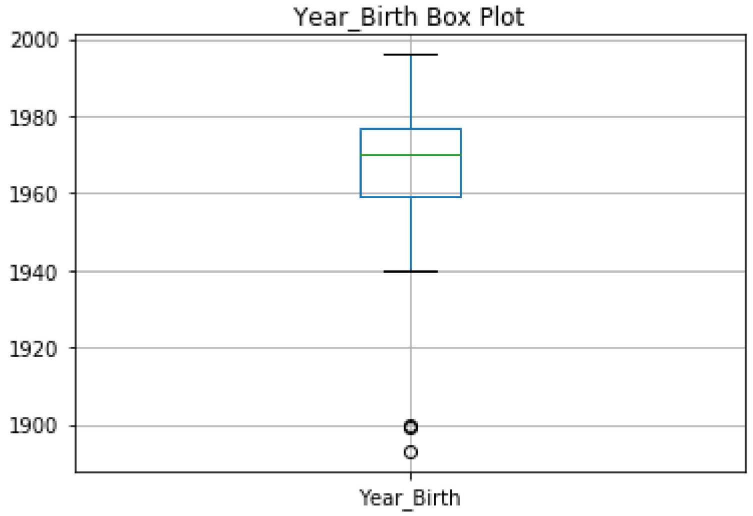
```
# View basic stats of the numerical variables
data_num.describe()
```

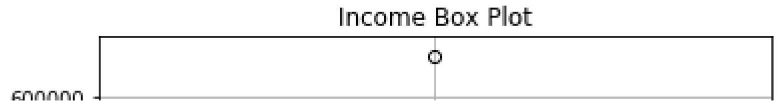
Out[12]:

	Year_Birth	Income	Recency	MntWines	MntFruits	MntMeatProducts	M
count	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	
mean	1968.805804	52237.975446	49.109375	303.935714	26.302232	166.950000	
std	11.984069	25037.955891	28.962453	336.597393	39.773434	225.715373	
min	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	
25%	1959.000000	35538.750000	24.000000	23.750000	1.000000	16.000000	
50%	1970.000000	51381.500000	49.000000	173.500000	8.000000	67.000000	
75%	1977.000000	68289.750000	74.000000	504.250000	33.000000	232.000000	
max	1996.000000	666666.000000	99.000000	1493.000000	199.000000	1725.000000	

In [13]:

```
# Obtaining outliers using boxplot
for col in data_num.columns:
    plt.figure()
    data_num.boxplot([col])
    plt.title(f'{col} Box Plot')
```





------------> # Removing Outliers

In [14]:

```
data.columns
```

Out[14]:

```
Index(['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', 'Age'],
      dtype='object')
```

In [16]:

```
#Removing birth year<1900
data = data[data['Year_Birth'] > 1900].reset_index(drop = True)
lst = ['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', 'Age']
df = data[lst]
df.to_csv("M_Data.csv")
df = pd.read_csv("M_Data.csv")
df.columns
```

Out[16]:

```
Index(['Unnamed: 0', 'Year_Birth', 'Education', 'Marital_Status', 'Income',
       'Kidhome', 'Teenhome', 'Dt_Customer', 'Recency', 'MntWines',
       'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProduct
s',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Response', 'Complain', 'Country', 'Age'],
      dtype='object')
```