Title: Final Report

Author: Dennis

Date: Jan 2022

Import Your Data

In [1]:
```
import numpy as np
from numpy import count_nonzero
from numpy import median
from numpy import mean
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import random

import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols

import datetime
from datetime import datetime, timedelta

import scipy.stats
from collections import Counter

%matplotlib inline
#sets the default autosave frequency in seconds
%autosave 60
sns.set_style('dark')
sns.set(font_scale=1.2)

plt.rc('axes', titlesize=9)
plt.rc('axes', labelsize=14)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)

import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)
pd.set_option('display.width', 1000)
pd.set_option('display.float_format','{:.2f}'.format)

random.seed(0)
np.random.seed(0)
np.set_printoptions(suppress=True)
```

```
Autosaving every 60 seconds
```

Exploratory Data Analysis

In [2]:
```
df = pd.read_csv("marketing_data.csv",parse_dates=['Dt_Customer'])
```

In [3]:
```
df
```

Out[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits
0	1826	1970	Graduation	Divorced	84835.00	0	0	2014-06-16	0	189	104
1	1	1961	Graduation	Single	57091.00	0	0	2014-06-15	0	464	5
2	10476	1958	Graduation	Married	67267.00	0	1	2014-05-13	0	134	11
3	1386	1967	Graduation	Together	32474.00	1	1	2014-11-05	0	10	0
4	5371	1989	Graduation	Single	21474.00	1	0	2014-08-04	0	6	16
...	...	...	...	...	...	...	...	...	...	...	...
2235	10142	1976	PhD	Divorced	66476.00	0	1	2013-07-03	99	372	18
2236	5263	1977	2n Cycle	Married	31056.00	1	0	2013-01-22	99	5	10
2237	22	1976	Graduation	Divorced	46310.00	1	0	2012-03-12	99	185	2
2238	528	1978	Graduation	Married	65819.00	0	0	2012-11-29	99	267	38
2239	4070	1969	PhD	Married	94871.00	0	2	2012-01-09	99	169	24

2240 rows × 28 columns

In [4]:
```
df.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   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   datetime64[ns]
 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: datetime64[ns](1), float64(1), int64(23), object(3)
memory usage: 490.1+ KB
```

In [5]:
```
df.describe(include='all')
```

Out[5]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits
count	2240.00	2240.00	2240	2240	2216.00	2240.00	2240.00	2240	2240.00	2240.00	2240.00
unique	NaN	NaN	5	8	NaN	NaN	NaN	663	NaN	NaN	NaN
top	NaN	NaN	Graduation	Married	NaN	NaN	NaN	2012-08-31 00:00:00	NaN	NaN	NaN
freq	NaN	NaN	1127	864	NaN	NaN	NaN	12	NaN	NaN	NaN
first	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2012-01-08 00:00:00	NaN	NaN	NaN
last	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014-12-06 00:00:00	NaN	NaN	NaN
mean	5592.16	1968.81	NaN	NaN	52247.25	0.44	0.51	NaN	49.11	303.94	26.30
std	3246.66	11.98	NaN	NaN	25173.08	0.54	0.54	NaN	28.96	336.60	39.77
min	0.00	1893.00	NaN	NaN	1730.00	0.00	0.00	NaN	0.00	0.00	0.00
25%	2828.25	1959.00	NaN	NaN	35303.00	0.00	0.00	NaN	24.00	23.75	1.00
50%	5458.50	1970.00	NaN	NaN	51381.50	0.00	0.00	NaN	49.00	173.50	8.00
75%	8427.75	1977.00	NaN	NaN	68522.00	1.00	1.00	NaN	74.00	504.25	33.00
max	11191.00	1996.00	NaN	NaN	666666.00	2.00	2.00	NaN	99.00	1493.00	199.00

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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
Response                0
Complain                0
Country                 0
dtype: int64
```

In [7]:
```
df.dropna(inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits
0	1826	1970	Graduation	Divorced	84835.00	0	0	2014-06-16	0	189	104
1	1	1961	Graduation	Single	57091.00	0	0	2014-06-15	0	464	5
2	10476	1958	Graduation	Married	67267.00	0	1	2014-05-13	0	134	11
3	1386	1967	Graduation	Together	32474.00	1	1	2014-11-05	0	10	0
4	5371	1989	Graduation	Single	21474.00	1	0	2014-08-04	0	6	16

In [9]:
```
df["Age"] = 2022 - df["Year_Birth"]
```

In [10]:
```
df.head()
```

Out[10]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits
0	1826	1970	Graduation	Divorced	84835.00	0	0	2014-06-16	0	189	104
1	1	1961	Graduation	Single	57091.00	0	0	2014-06-15	0	464	5
2	10476	1958	Graduation	Married	67267.00	0	1	2014-05-13	0	134	11
3	1386	1967	Graduation	Together	32474.00	1	1	2014-11-05	0	10	0
4	5371	1989	Graduation	Single	21474.00	1	0	2014-08-04	0	6	16

In [ ]:

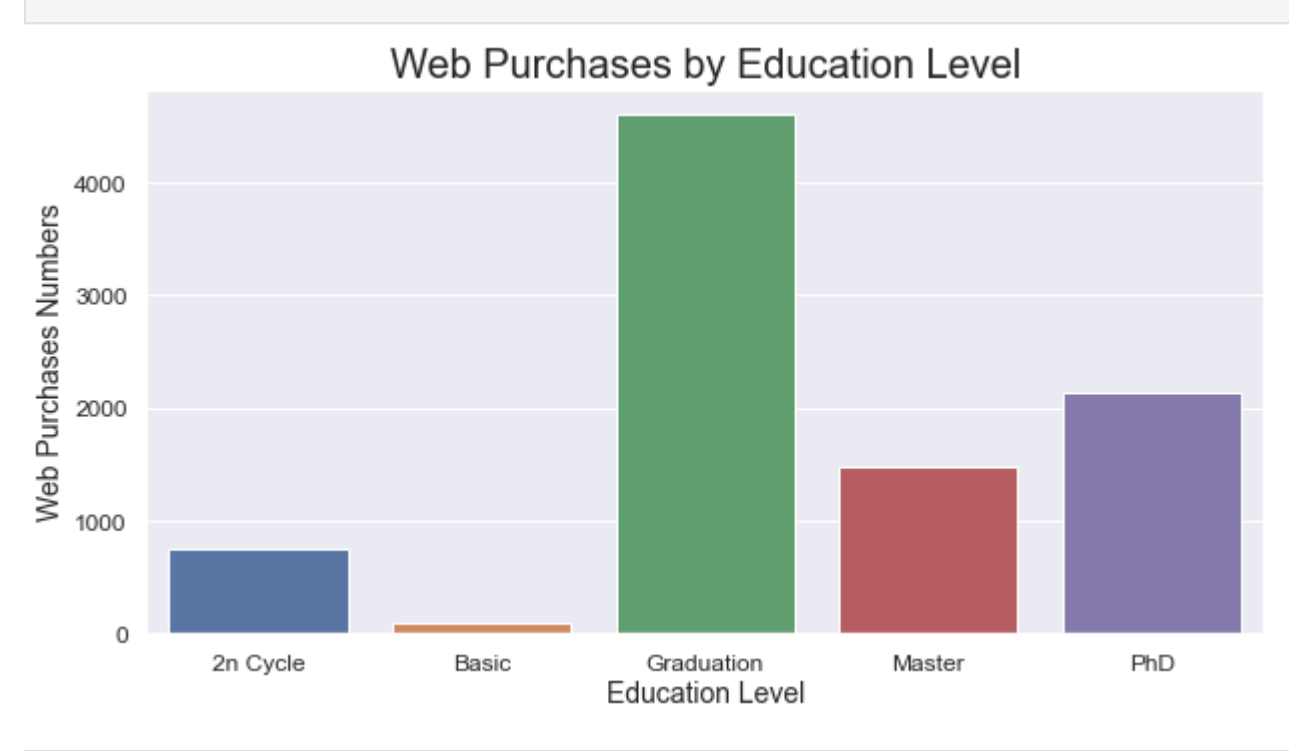
8 interesting graphs

In [11]:
```
barplot = pd.DataFrame(df.groupby(["Education"])["NumWebPurchases"].sum())
barplot
```

Out[11]:

Education	NumWebPurchases
2n Cycle	753
Basic	102
Graduation	4593
Master	1473
PhD	2132

In [12]:
```
plt.figure(figsize=(10,5))
sns.barplot(x=barplot.index, y="NumWebPurchases", data=barplot)
plt.title("Web Purchases by Education Level", size=20)
plt.xlabel("Education Level")
plt.ylabel("Web Purchases Numbers")
plt.show()
```



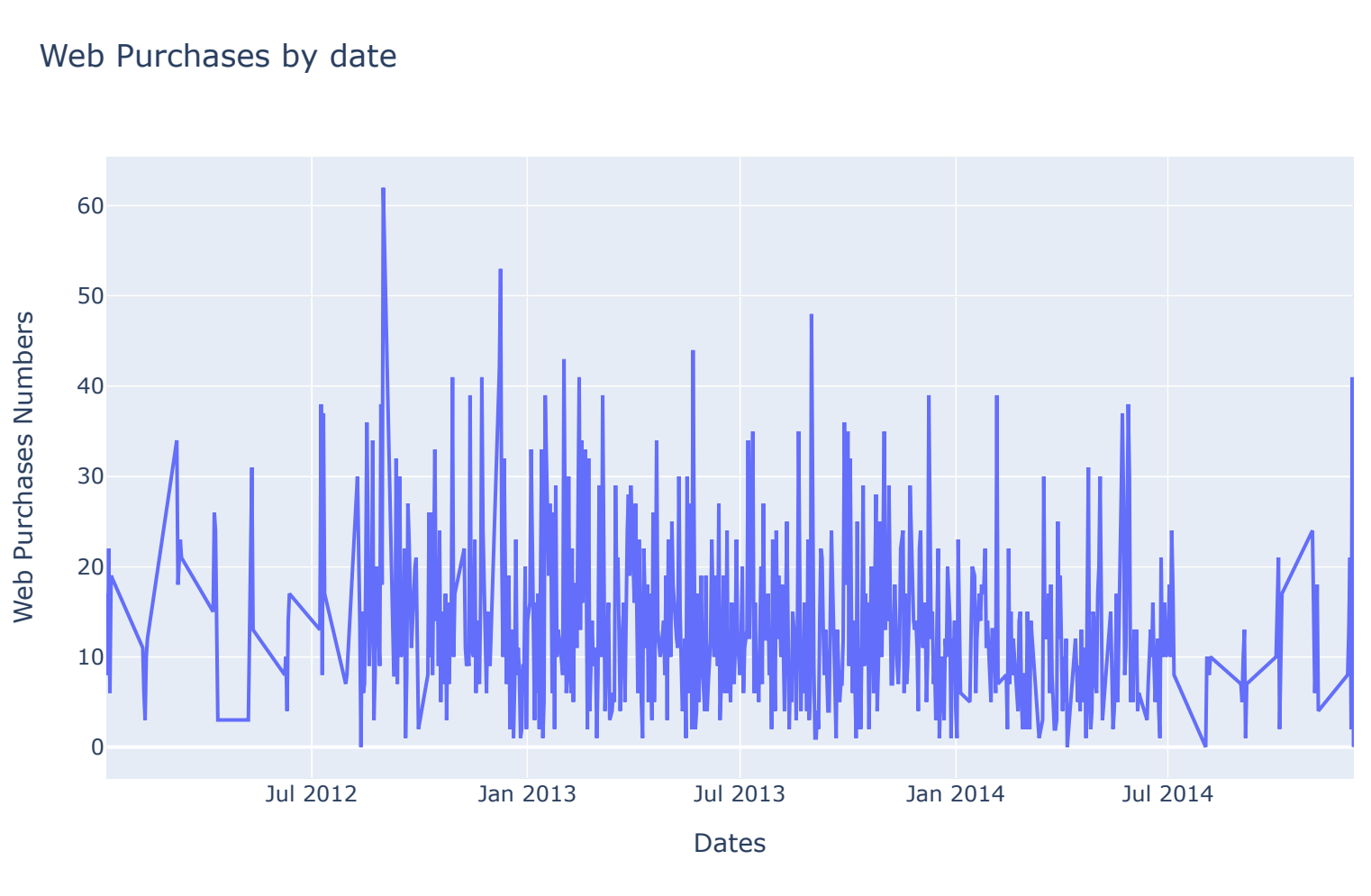
In [13]:
```
lineplot = pd.DataFrame(df.groupby(["Dt_Customer"])["NumWebPurchases"].sum())
lineplot
```

Out[13]:

Dt_Customer	NumWebPurchases
2012-01-08	17
2012-01-09	8
2012-01-10	22
2012-01-11	6
2012-01-12	19
...	...
2014-12-02	13
2014-12-03	21
2014-12-04	2
2014-12-05	41
2014-12-06	0

662 rows × 1 columns

In [14]:
```
fig = px.line(data_frame=lineplot, x=lineplot.index, y='NumWebPurchases',
              labels={
                     "Dt_Customer": "Dates",
                     "NumWebPurchases": "Web Purchases Numbers"

                 }, title='Web Purchases by date')
fig.show()
```



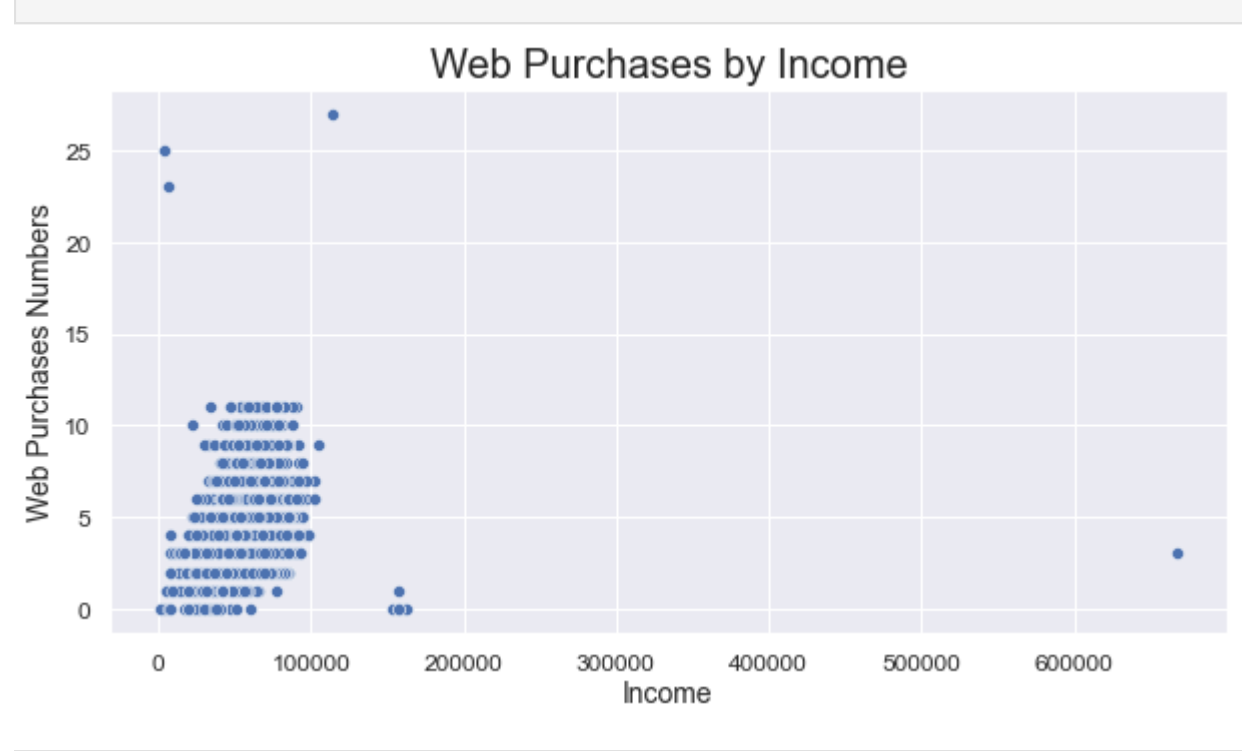
In [15]:
```
scatter = df[["Income","NumWebPurchases"]]
scatter
```

Out[15]:

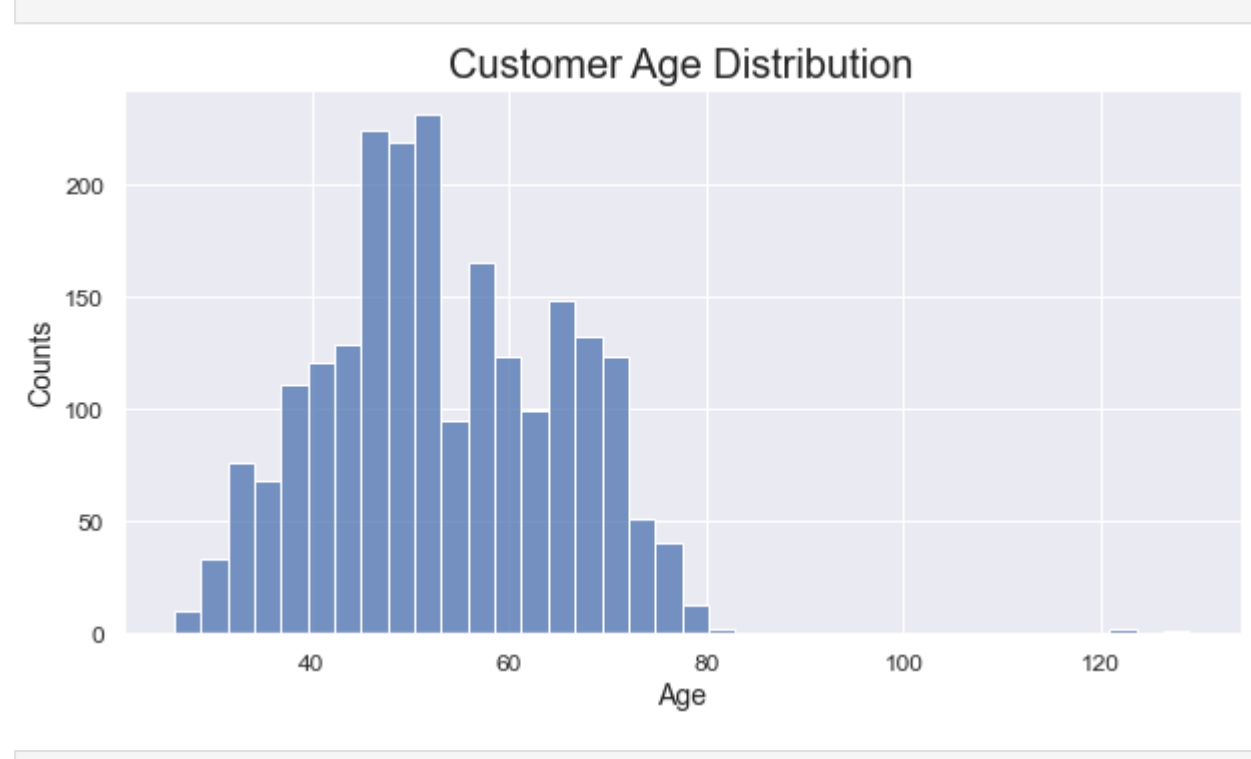
	Income	NumWebPurchases
0	84835.00	4
1	57091.00	7
2	67267.00	3
3	32474.00	1
4	21474.00	3
...	...	...
2235	66476.00	5
2236	31056.00	1
2237	46310.00	6
2238	65819.00	5
2239	94871.00	8

2216 rows × 2 columns

In [16]:
```
plt.figure(figsize=(10,5))
sns.scatterplot(x=scatter.Income, y=scatter.NumWebPurchases, data=scatter)
plt.title("Web Purchases by Income", size=20)
plt.xlabel("Income")
plt.ylabel("Web Purchases Numbers")
plt.show()
```



In [17]:
```
plt.figure(figsize=(10,5))
sns.histplot(x="Age", data=df)
plt.title("Customer Age Distribution", size=20)
plt.xlabel("Age")
plt.ylabel("Counts")
plt.show()
```

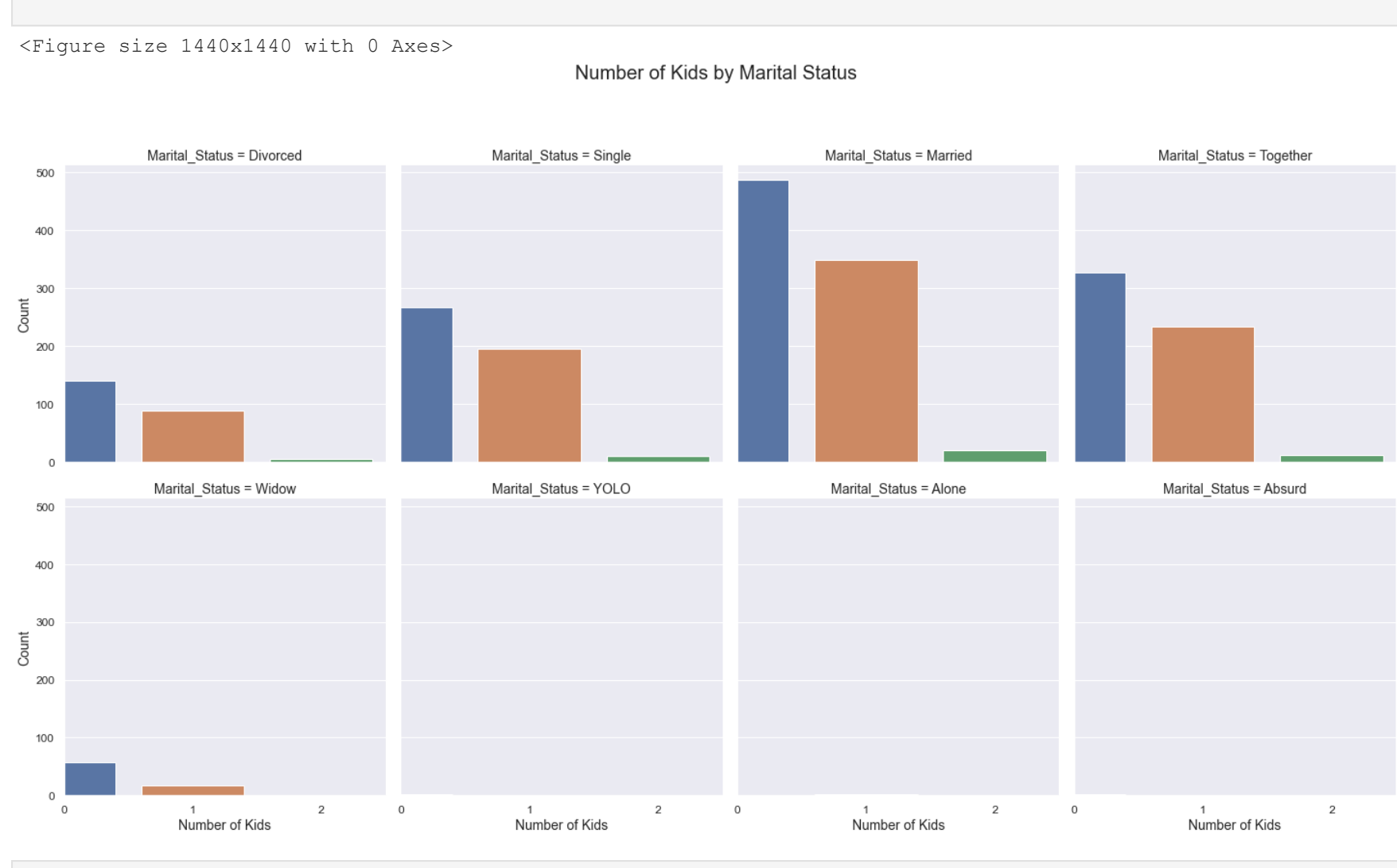


In [18]:
```
df.Marital_Status.value_counts()
```

Out[18]:
```
Married     857
Together    573
Single      471
Divorced    232
Widow        76
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

In [19]:
```
plt.figure(figsize=(20,20))


g = sns.catplot(x='Kidhome', col = 'Marital_Status', col_wrap=4,
            kind='count', data=df,
            height = 5, aspect = 1)

g.set_xlabels("Number of Kids")
g.set_ylabels("Count")
#g = (g.set_axis_labels("Tip","Total bill(USD)").set(xlim=(0,10),ylim=(0,100)


g.set(xlim=(0,None))
g.set_xticklabels(rotation=0)

plt.suptitle('Number of Kids by Marital Status', x=0.5, y=1.1, ha='center', fontsize=20)

plt.show()
```

```
<Figure size 1440x1440 with 0 Axes>
```



In [20]:
```
df.NumWebVisitsMonth.unique()
```

Out[20]:
```
array([ 1,  5,  2,  7,  6,  4,  8,  3,  9,  0, 17, 13, 10, 14, 19, 20],
      dtype=int64)
```

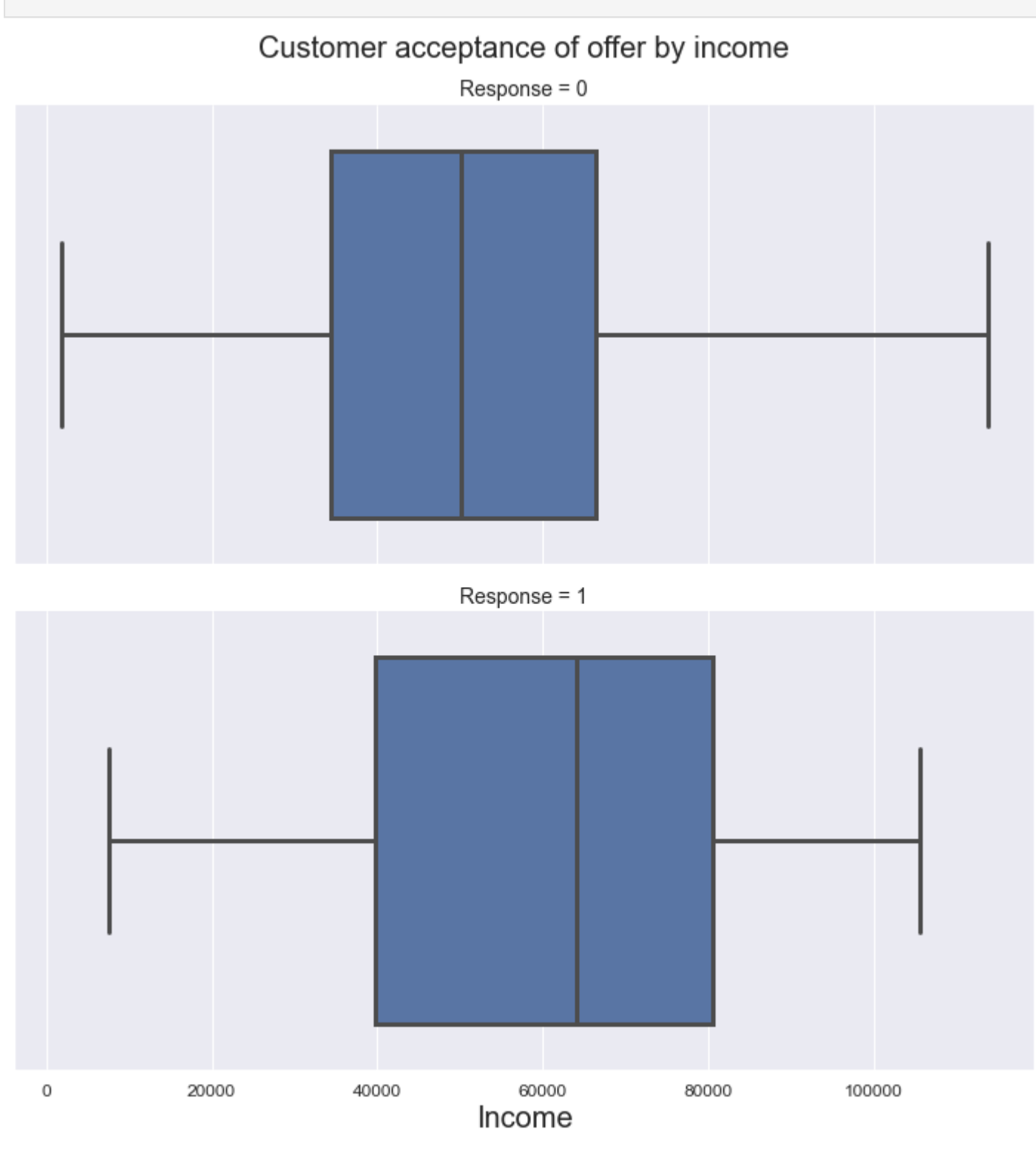
In [21]:
```
plt.figure(figsize=(20,20))


g = sns.catplot(x='NumWebVisitsMonth', y='Income', col = 'Country', col_wrap=4,
            kind='bar', data=df,
            height = 5, aspect = 1)

g.set_xlabels("Web Visits")
g.set_ylabels("Count")
#g = (g.set_axis_labels("Tip","Total bill(USD)").set(xlim=(0,10),ylim=(0,100)


g.set(xlim=(0,None))
g.set_xticklabels(rotation=0)

plt.suptitle('Number of Web Visits Last Month by Country', x=0.5, y=1.1, ha='center', fontsize=20)

plt.show()
```

```
<Figure size 1440x1440 with 0 Axes>
```

In [22]:
```
df.Response.unique()
```

Out[22]:
```
array([1, 0], dtype=int64)
```

In [23]:
```
sns.catplot(x="Income", row="Response", ci=None,

                data=df, color=None, linewidth=3, showfliers = False,

                orient="h", height=5, aspect=2, palette=None,

                kind="box", dodge=True)

plt.xlabel("Income", size=20)
plt.ylabel("", size=20)
plt.suptitle('Customer acceptance of offer by income', x=0.5, y=1.02, ha='center', fontsize=20)
plt.show()
```



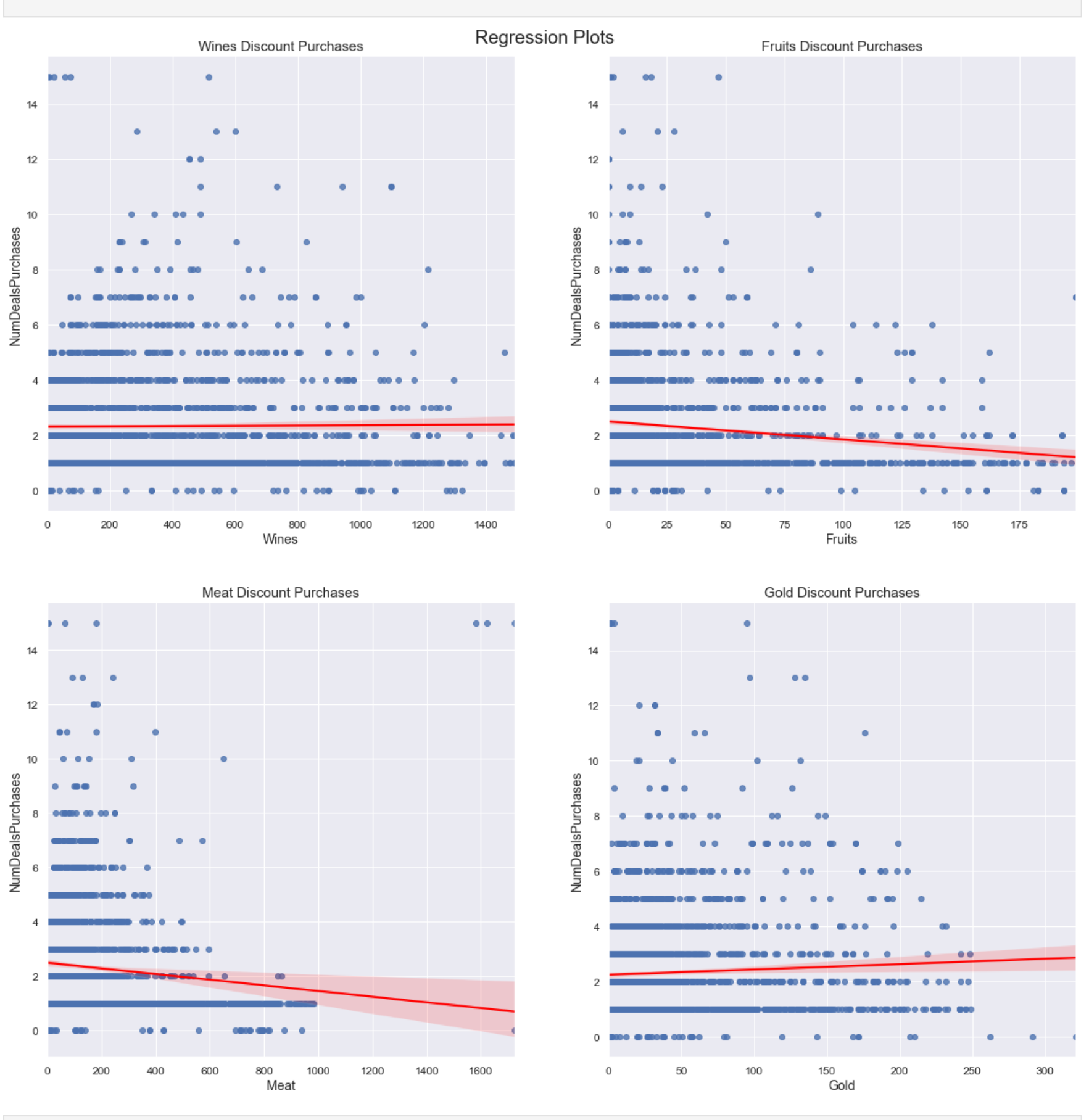
In [24]:
```
df.NumDealsPurchases.unique()
```

Out[24]:
```
array([ 1,  2,  3,  0,  4, 12,  7,  5,  6, 11,  9,  8, 10, 15, 13],
      dtype=int64)
```

In [25]:
```
df.columns
```

Out[25]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome', 'Teenhome', 'Dt_Customer', 'Rece
ncy', 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds', 'NumDe
alsPurchases', 'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp
3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'AcceptedCmp2', 'Response', 'Complain', 'Country', 'Age'],
dtype='object')
```

In [26]:
```
line_color = {'color': 'red'}
fig , ax = plt.subplots(2,2, figsize=(20,20))

#Feature

ax1 = sns.regplot(x=df.MntWines, y=df.NumDealsPurchases, line_kws=line_color, ax=ax[0,0])
ax1.set_xlabel("Wines")
ax1.set_ylabel("NumDealsPurchases")
ax1.set_title("Wines Discount Purchases", size=15)

#Feature

ax2 = sns.regplot(x=df.MntFruits, y=df.NumDealsPurchases, line_kws=line_color, ax=ax[0,1])
ax2.set_xlabel("Fruits")
ax2.set_ylabel("NumDealsPurchases")
ax2.set_title("Fruits Discount Purchases", size=15)

#Feature

ax3 = sns.regplot(x=df.MntMeatProducts, y=df.NumDealsPurchases, line_kws=line_color, ax=ax[1,0])
ax3.set_xlabel("Meat")
ax3.set_ylabel("NumDealsPurchases")
ax3.set_title("Meat Discount Purchases", size=15)

#Feature

ax4 = sns.regplot(x=df.MntGoldProds, y=df.NumDealsPurchases, line_kws=line_color, ax=ax[1,1])
ax4.set_xlabel("Gold")
ax4.set_ylabel("NumDealsPurchases")
ax4.set_title("Gold Discount Purchases", size=15)
plt.suptitle('Regression Plots', x=0.5, y=0.9, ha='center', fontsize=20)
plt.show()
```



In [ ]: