In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

Importing Data

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

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

Out[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Cust
0	1826	1970	Graduation	Divorced	$84,835.00	0	0	6/
1	1	1961	Graduation	Single	$57,091.00	0	0	6/
2	10476	1958	Graduation	Married	$67,267.00	0	1	5/
3	1386	1967	Graduation	Together	$32,474.00	1	1	5/
4	5371	1989	Graduation	Single	$21,474.00	1	0	4
...	...	...	...	...	...	...	...	
2235	10142	1976	PhD	Divorced	$66,476.00	0	1	3
2236	5263	1977	2n Cycle	Married	$31,056.00	1	0	1/
2237	22	1976	Graduation	Divorced	$46,310.00	1	0	12
2238	528	1978	Graduation	Married	$65,819.00	0	0	11/
2239	4070	1969	PhD	Married	$94,871.00	0	2	9

2240 rows × 28 columns

In [4]:
```
df.shape
```

Out[4]:
```
(2240, 28)
```

In [5]:
```
df_excel=pd.read_excel('1688639964_datadictionaryresponsetomarketingcampaign
```

In [6]:
```
df_excel
```

Out[6]:

	Variable	Description
0	ID	Customer's unique identifier
1	Year_Birth	Customer's birth year
2	Education	Customer's education level
3	Marital_Status	Customer's marital status
4	Income	Customer's yearly household income
5	Kidhome	number of small children in customer's househo...
6	Teenhome	no of teenagers in customer's house
7	Dt_Customer	Date of customer's enrollment with the company
8	Recency	number of days since the last purchase
9	MntWines	amount spent on wine in last 2 years
10	MntFruits	amount spent on fruits in last 2 years
11	MntMeatProducts	amount spent on meat products in last 2 years
12	MntFishProducts	amount spent on fish products in last 2 years
13	MntSweetProducts	amount spent on sweet products in last 2 years
14	MntGoldProds	amount spent on gold in last 2 years
15	NumDealsPurchases	no of purchases made with discount
16	NumWebPurchases	no of purchases made through company's website
17	NumCatalogPurchases	no of purchases made using catelogue
18	NumStorePurchases	no of purchases made directly in store
19	NumWebVisitsMonth	no of visits to company's website in the last ...
20	AcceptedCmp3	1 if the customer accepted the offer in the 3r...
21	AcceptedCmp4	1 if the customer accepted the offer in the 4t...
22	AcceptedCmp5	1 if the customer accepted the offer in the 5t...
23	AcceptedCmp1	1 if the customer accepted the offer in the fi...
24	AcceptedCmp2	1 if the customer accepted the offer in the 2n...
25	Response	1 if the customer accepted the offer in the la...
26	Complain	1 if customer complained in the last 2 years
27	Country	Customer's location

In [7]:
```
df.columns
```

Out[7]:
```
Index(['ID', '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'],
      dtype='object')
```

- **Since the column name 'Income' has spaces we need to remove those spaces.**

In [8]:
```
df.columns = ['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', 'Response', 'Complain', 'Country']
```

In [9]:
```
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   object
 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  Response             2240 non-null   int64 
 26  Complain             2240 non-null   int64 
 27  Country              2240 non-null   object
dtypes: int64(23), object(5)
memory usage: 490.1+ KB
```

- **The income dtype is non-numeric. We need to convert it into float by removing comma and dollar sign.**

In [10]:
```
df['Income']= df['Income'].str.replace('$','')
df['Income']= df['Income'].str.replace(',','')
df['Income']= df['Income'].astype(float)
```

In [11]:
```
df
```

Out[11]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Custom
0	1826	1970	Graduation	Divorced	84835.0	0	0	6/16/
1	1	1961	Graduation	Single	57091.0	0	0	6/15/
2	10476	1958	Graduation	Married	67267.0	0	1	5/13/
3	1386	1967	Graduation	Together	32474.0	1	1	5/11/
4	5371	1989	Graduation	Single	21474.0	1	0	4/8/
...	...	...	...	...	...	...	...	
2235	10142	1976	PhD	Divorced	66476.0	0	1	3/7/
2236	5263	1977	2n Cycle	Married	31056.0	1	0	1/22/
2237	22	1976	Graduation	Divorced	46310.0	1	0	12/3/
2238	528	1978	Graduation	Married	65819.0	0	0	11/29/
2239	4070	1969	PhD	Married	94871.0	0	2	9/1/

2240 rows × 28 columns

In [12]:
```
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   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  Response             2240 non-null   int64  
 26  Complain             2240 non-null   int64  
 27  Country              2240 non-null   object 
dtypes: float64(1), int64(23), object(4)
memory usage: 490.1+ KB
```

Checking and filling nulls

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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
```

- **There are 24 nulls in income column. We need to treat them with their similar education background and marital status. For that we need to find the average incomes on various education and marital status.**

In [14]:
```
df_income = df.groupby(['Education','Marital_Status'])["Income"].mean()
```

In [15]:
```
df_income
```

Out[15]:
```
Education   Marital_Status
2n Cycle    Divorced          49395.130435
            Married           46201.100000
            Single            53673.944444
            Together          44736.410714
            Widow             51392.200000
Basic       Divorced           9548.000000
            Married           21960.500000
            Single            18238.666667
            Together          21240.071429
            Widow             22123.000000
Graduation  Absurd            79244.000000
            Alone             34176.000000
            Divorced          54526.042017
            Married           50800.258741
            Single            51322.182927
            Together          55758.480702
            Widow             54976.657143
Master      Absurd            65487.000000
            Alone             61331.000000
            Divorced          50331.945946
            Married           53286.028986
            Single            53530.560000
            Together          52109.009804
            Widow             58401.545455
PhD         Alone             35860.000000
            Divorced          53096.615385
            Married           58138.031579
            Single            53314.614583
            Together          56041.422414
            Widow             60288.083333
            YOLO              48432.000000
Name: Income, dtype: float64
```

In [16]:
```
df_income = pd.DataFrame(df_income).reset_index()
```

In [17]:
```
df_income
```

Out[17]:

	Education	Marital_Status	Income
0	2n Cycle	Divorced	49395.130435
1	2n Cycle	Married	46201.100000
2	2n Cycle	Single	53673.944444
3	2n Cycle	Together	44736.410714
4	2n Cycle	Widow	51392.200000
5	Basic	Divorced	9548.000000
6	Basic	Married	21960.500000
7	Basic	Single	18238.666667
8	Basic	Together	21240.071429
9	Basic	Widow	22123.000000
10	Graduation	Absurd	79244.000000
11	Graduation	Alone	34176.000000
12	Graduation	Divorced	54526.042017
13	Graduation	Married	50800.258741
14	Graduation	Single	51322.182927
15	Graduation	Together	55758.480702
16	Graduation	Widow	54976.657143
17	Master	Absurd	65487.000000
18	Master	Alone	61331.000000
19	Master	Divorced	50331.945946
20	Master	Married	53286.028986
21	Master	Single	53530.560000
22	Master	Together	52109.009804
23	Master	Widow	58401.545455
24	PhD	Alone	35860.000000
25	PhD	Divorced	53096.615385
26	PhD	Married	58138.031579
27	PhD	Single	53314.614583
28	PhD	Together	56041.422414
29	PhD	Widow	60288.083333
30	PhD	YOLO	48432.000000

- **After finding the average incomes we have to merge the two dataframe and have to replace null by second income column.**

In [18]:
```
df2 = pd.merge(df,df_income, on=['Education','Marital_Status'] , how='left')
```

In [19]:
```
df2
```

Out[19]:

	ID	Year_Birth	Education	Marital_Status	Income_x	Kidhome	Teenhome	Dt_Custo
0	1826	1970	Graduation	Divorced	84835.0	0	0	6/1
1	1	1961	Graduation	Single	57091.0	0	0	6/1
2	10476	1958	Graduation	Married	67267.0	0	1	5/1
3	1386	1967	Graduation	Together	32474.0	1	1	5/1
4	5371	1989	Graduation	Single	21474.0	1	0	4/
...	...	...	...	...	...	...	...	
2235	10142	1976	PhD	Divorced	66476.0	0	1	3/
2236	5263	1977	2n Cycle	Married	31056.0	1	0	1/2
2237	22	1976	Graduation	Divorced	46310.0	1	0	12/
2238	528	1978	Graduation	Married	65819.0	0	0	11/2
2239	4070	1969	PhD	Married	94871.0	0	2	9/

2240 rows × 29 columns

In [20]:
```
df2['Income_x'] = np.where(df2['Income_x'].isnull(),df2['Income_y'], df2['Ir
```

In [21]:
```
df2.isnull().sum()
```

Out[21]:
```
ID                   0
Year_Birth           0
Education            0
Marital_Status       0
Income_x             0
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
Income_y             0
dtype: int64
```

- **All the 24 nulls have been treated with average incomes based on education and marital status.**

- **Since the original df has only one income column, we have to put the new dataframe back in original dataframe by dropping the second income column and renaming first income column.**

In [22]:
```
df = df2.drop('Income_y',axis=1)
df = df.rename(columns ={'Income_x' : 'Income'})
```

- **Now lets check whether the data has been correctly transferred or not.**

In [23]:
```
df.head(10)
```

Out[23]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
0	1826	1970	Graduation	Divorced	84835.0	0	0	6/16/14
1	1	1961	Graduation	Single	57091.0	0	0	6/15/14
2	10476	1958	Graduation	Married	67267.0	0	1	5/13/14
3	1386	1967	Graduation	Together	32474.0	1	1	5/11/14
4	5371	1989	Graduation	Single	21474.0	1	0	4/8/14
5	7348	1958	PhD	Single	71691.0	0	0	3/17/14
6	4073	1954	2n Cycle	Married	63564.0	0	0	1/29/14
7	1991	1967	Graduation	Together	44931.0	0	1	1/18/14
8	4047	1954	PhD	Married	65324.0	0	1	1/11/14
9	9477	1954	PhD	Married	65324.0	0	1	1/11/14

10 rows × 28 columns

Creating Variables

- **We need to create variables to find total number of children, total spending and total purchases.**

In [24]:
```
df['Total_Children'] = df['Kidhome'] + df['Teenhome']
df['Total_Spending'] = df['MntWines'] + df['MntFruits'] + df['MntMeatProduct
df['Total_Purchases'] = df['NumWebPurchases'] + df['NumCatalogPurchases'] +
```

- **Since the last transaction year was 2014, we assume that the data had been taken from that year and using it as current year to infer ages of customers.**

In [25]:
```
df['Age'] = 2014 - df['Year_Birth']
```

In [26]:
```
df.columns
```

Out[26]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidho
me',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Response', 'Complain', 'Country', 'Total_Childre
n',
       'Total_Spending', 'Total_Purchases', 'Age'],
      dtype='object')
```

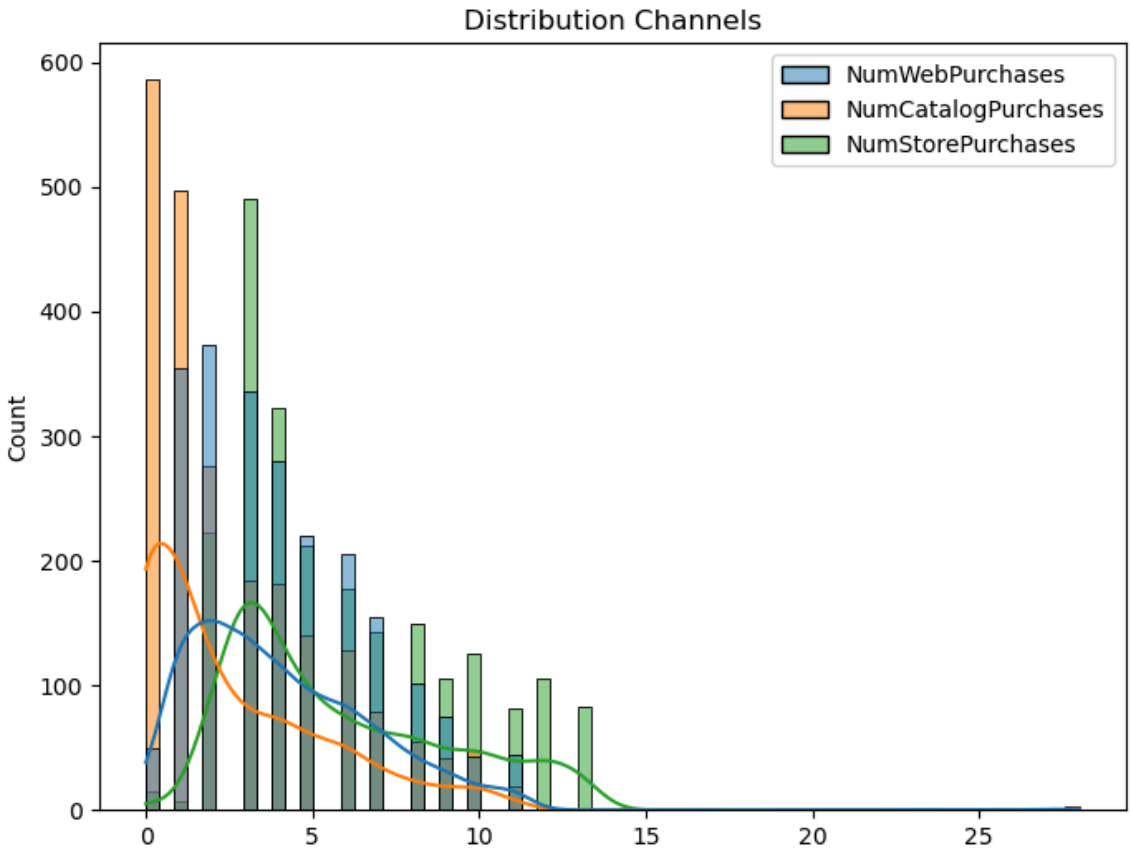
- **The variables created are being reflected as columns in data.**

Creating figures and performing outlier treatment

- **We need to create box plot and histograms for better understanding and analysis of distribution channels.**

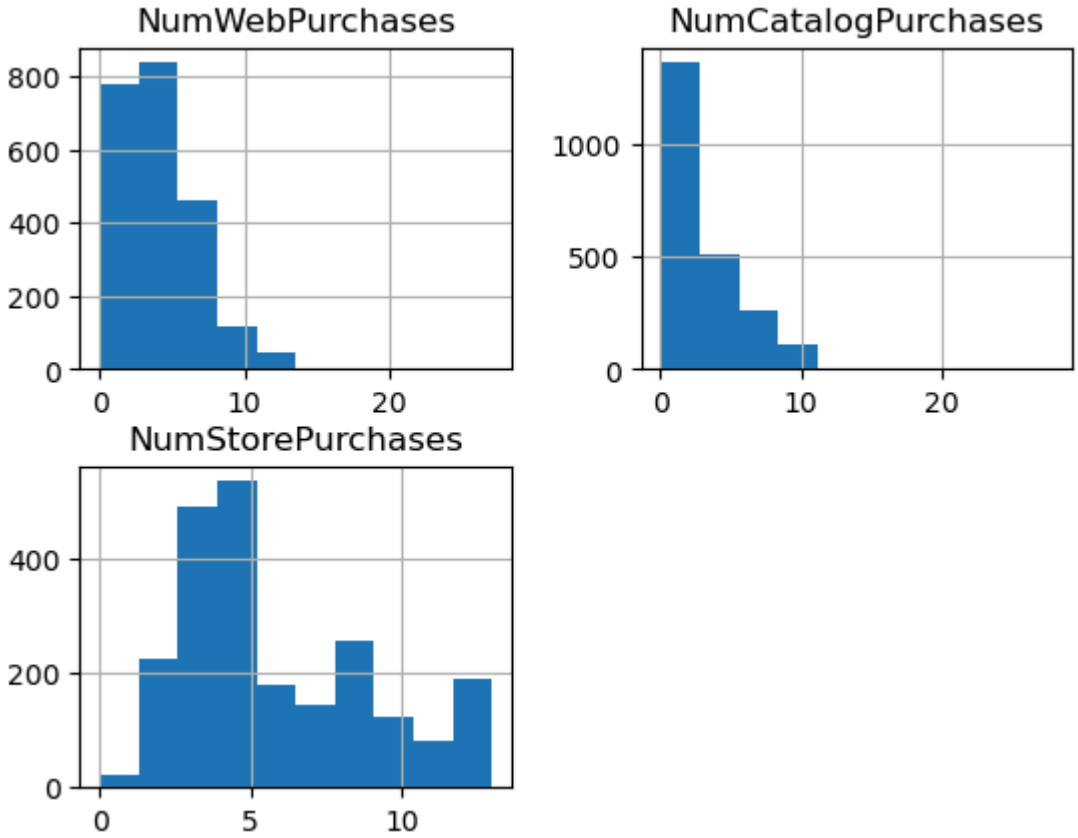
In [27]:
```
plt.figure(figsize=(8,6))
dis_cnl = df[['NumWebPurchases','NumCatalogPurchases','NumStorePurchases']]
sns.histplot(dis_cnl,kde=True)
plt.title('Distribution Channels')
```

Out[27]: Text(0.5, 1.0, 'Distribution Channels')



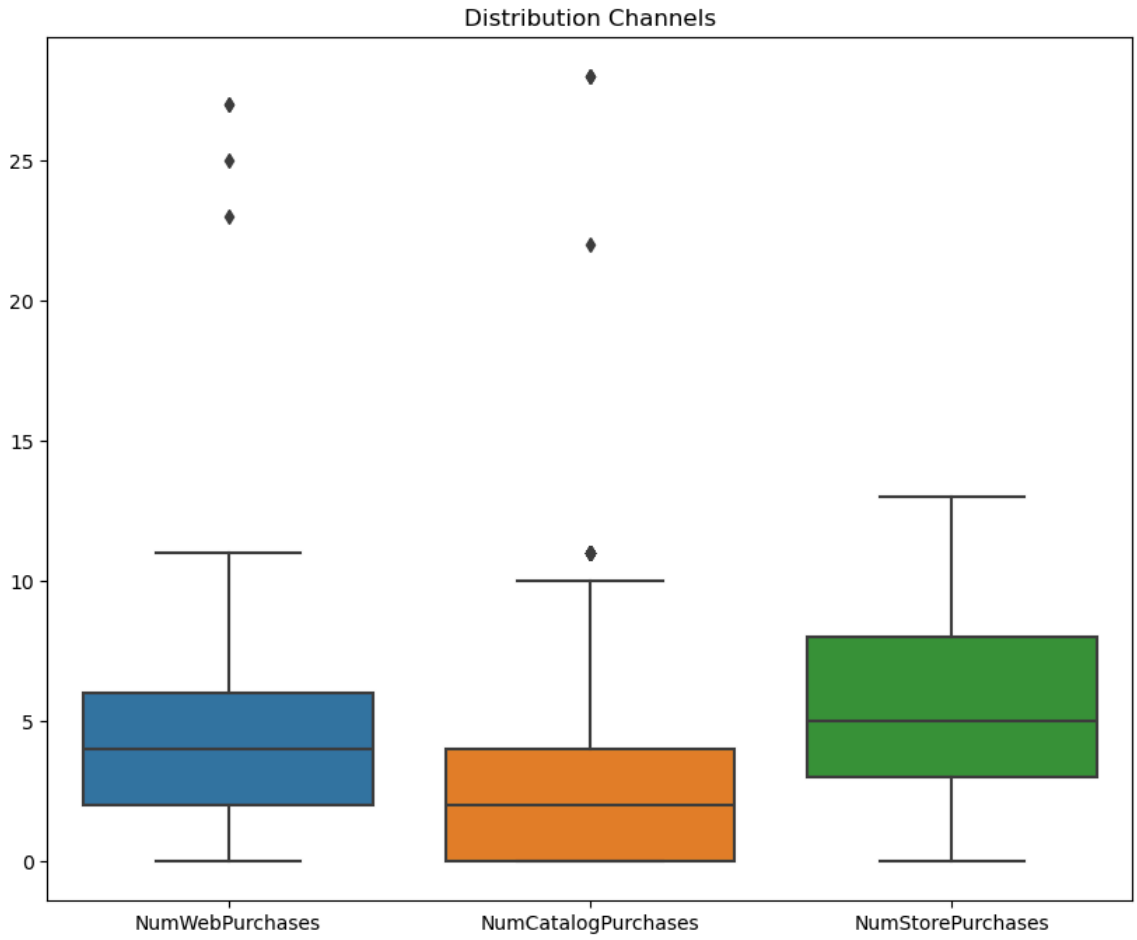
In [28]:
```
dis_cnl.hist()
```

Out[28]:
```
array([[<Axes: title={'center': 'NumWebPurchases'}>,
        <Axes: title={'center': 'NumCatalogPurchases'}>],
       [<Axes: title={'center': 'NumStorePurchases'}>, <Axes: >]],
      dtype=object)
```



In [29]:
```
plt.figure(figsize=(10,8))
sns.boxplot(dis_cnl)
plt.title('Distribution Channels')
```

Out[29]:
```
Text(0.5, 1.0, 'Distribution Channels')
```



- **The dots in the above boxplot are outliers. We have to treat them to make sure that we won't get problem in our statistical analysis later.**

In [30]:
```
df['NumWebPurchases'].describe()
```

Out[30]:
```
count    2240.000000
mean        4.084821
std         2.778714
min         0.000000
25%         2.000000
50%         4.000000
75%         6.000000
max        27.000000
Name: NumWebPurchases, dtype: float64
```

In [31]:
```
web_q1 = df['NumWebPurchases'].quantile(0.25)
web_q2 = df['NumWebPurchases'].quantile(0.75)
```

In [32]:
```
web_iqr = web_q2 - web_q1
```

In [33]:
```
web_ul = web_q2 + 1.5 * web_iqr
web_ll = web_q1 + 1.5 * web_iqr
```

- **For treating outliers, first we need to find them.**

In [34]:
```
df[df['NumWebPurchases'] > web_ul]
```

Out[34]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_C
14	10311	1969	Graduation	Married	4428.000000	0	1	
210	4619	1945	PhD	Single	113734.000000	0	0	
449	5255	1986	Graduation	Single	51322.182927	1	0	
2063	6237	1966	PhD	Single	7144.000000	0	2	

4 rows × 32 columns

- **Now let's trim the outliers.**

In [35]:
```
df3 = df[df['NumWebPurchases'] < web_ul]
```

In [36]:
```
df3.shape
```

Out[36]:
```
(2236, 32)
```

- **Comparing the dataframes.**

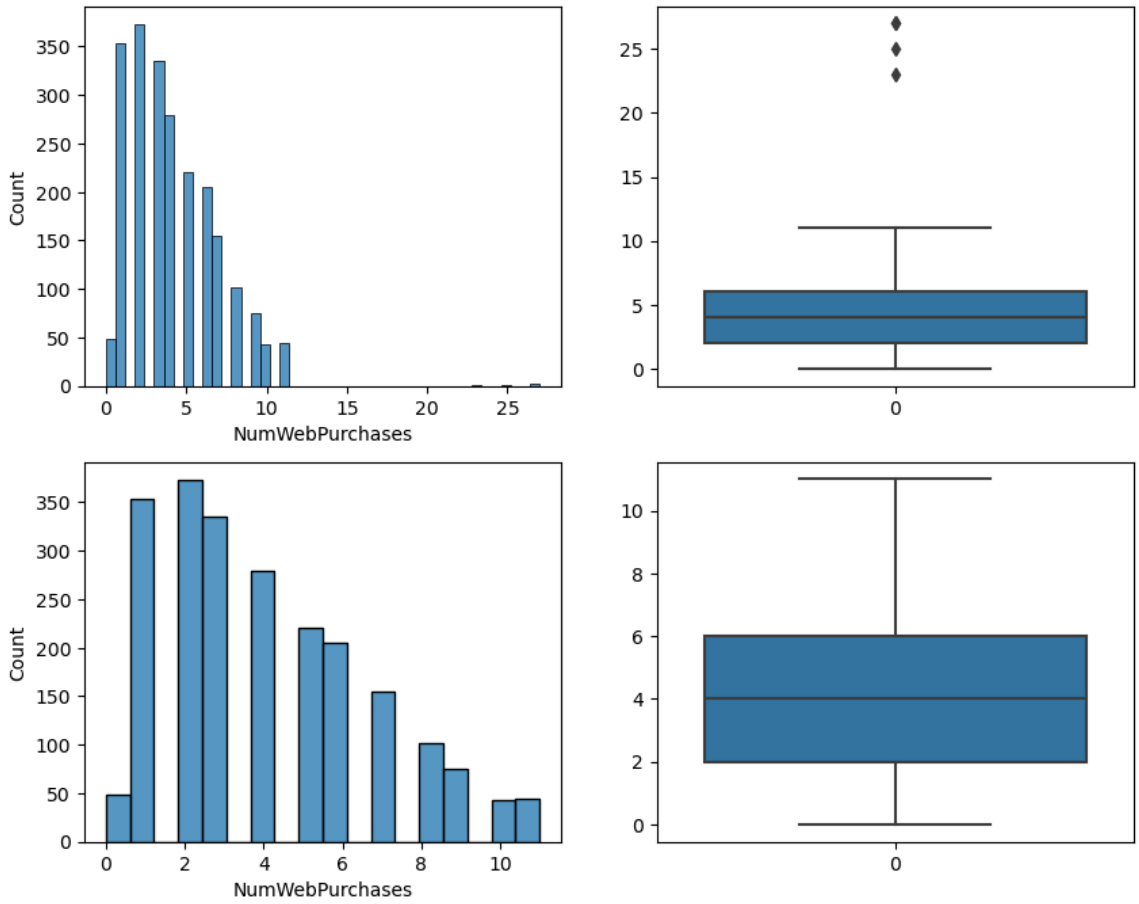
In [37]:
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
sns.histplot(df['NumWebPurchases'])

plt.subplot(2,2,2)
sns.boxplot(df['NumWebPurchases'])

plt.subplot(2,2,3)
sns.histplot(df3['NumWebPurchases'])

plt.subplot(2,2,4)
sns.boxplot(df3['NumWebPurchases'])
```

Out[37]: <Axes: >



- **The outliers are trimmed from the data.**

- **Similarly for the catalog purchases.**

In [38]:
```
df['NumCatalogPurchases'].describe()
```

Out[38]:
```
count    2240.000000
mean        2.662054
std         2.923101
min         0.000000
25%         0.000000
50%         2.000000
75%         4.000000
max        28.000000
Name: NumCatalogPurchases, dtype: float64
```

In [39]:
```
cat_q1 = df['NumCatalogPurchases'].quantile(0.25)
cat_q2 = df['NumCatalogPurchases'].quantile(0.75)
```

In [40]:
```
cat_iqr = cat_q2 - cat_q1
```

In [41]:
```
cat_ul = cat_q2 + 1.5 * cat_iqr
cat_ll = cat_q1 + 1.5 * cat_iqr
```

In [42]:
```
df[df['NumCatalogPurchases'] > cat_ul]
```

Out[42]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Custor
292	2324	1972	Graduation	Together	77044.0	0	1	10/27
325	4931	1977	Graduation	Together	157146.0	0	0	4/29
399	5718	1950	Graduation	Married	80763.0	0	0	8/15
434	10102	1966	Graduation	Widow	79946.0	0	0	5/12
497	1501	1982	PhD	Married	160803.0	0	0	8/4
588	7627	1975	Master	Married	92163.0	0	0	12/12
661	4299	1960	Graduation	Together	70971.0	0	1	9/21
678	10524	1963	Master	Divorced	49476.0	0	1	6/20
803	8908	1959	Graduation	Married	87195.0	0	0	5/8
911	6246	1953	Graduation	Single	73892.0	0	0	11/13
961	5376	1979	Graduation	Married	2447.0	1	0	1/6
1275	1139	1984	PhD	Married	73356.0	0	0	2/6
1368	2109	1990	Graduation	Single	96843.0	0	0	4/23
1404	1763	1988	Graduation	Together	87679.0	0	0	7/27
1489	9058	1955	Graduation	Widow	79800.0	0	0	9/23
1628	6945	1952	Graduation	Single	84574.0	0	0	6/4
1687	9292	1952	Graduation	Married	81795.0	0	0	10/26
1834	17	1971	PhD	Married	60491.0	0	1	9/6
2026	4964	1958	PhD	Together	74250.0	0	0	1/26
2060	4508	1952	Graduation	Single	75127.0	0	0	5/22
2061	4843	1952	Graduation	Single	75127.0	0	0	5/22
2074	4687	1958	Master	Married	80739.0	0	0	5/23
2204	8475	1973	PhD	Married	157243.0	0	1	3/1

23 rows × 32 columns

In [43]:
```
df4 = df[df['NumCatalogPurchases'] < cat_ul]
```

In [44]:
```
df4.shape
```

Out[44]:
```
(2169, 32)
```

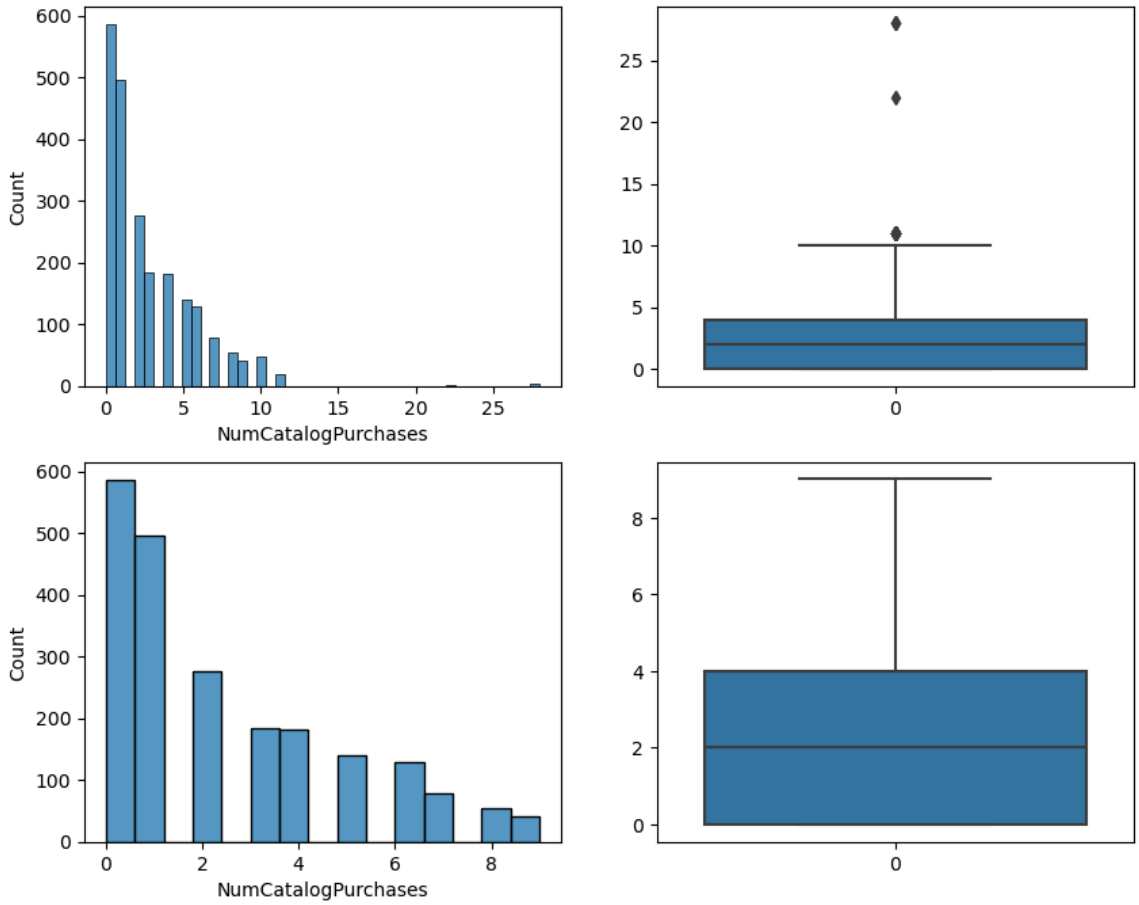
In [45]:
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
sns.histplot(df['NumCatalogPurchases'])

plt.subplot(2,2,2)
sns.boxplot(df['NumCatalogPurchases'])

plt.subplot(2,2,3)
sns.histplot(df4['NumCatalogPurchases'])

plt.subplot(2,2,4)
sns.boxplot(df4['NumCatalogPurchases'])
```

Out[45]: <Axes: >



Encoding

In [46]:
```
df5 = pd.get_dummies(df[['Education','Marital_Status','Country']],drop_first
df = pd.concat([df,df5], axis=1)
df
```

Out[46]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Custom
0	1826	1970	Graduation	Divorced	84835.0	0	0	6/16/
1	1	1961	Graduation	Single	57091.0	0	0	6/15/
2	10476	1958	Graduation	Married	67267.0	0	1	5/13/
3	1386	1967	Graduation	Together	32474.0	1	1	5/11/
4	5371	1989	Graduation	Single	21474.0	1	0	4/8/
...	...	...	...	...	...	...	...	
2235	10142	1976	PhD	Divorced	66476.0	0	1	3/7/
2236	5263	1977	2n Cycle	Married	31056.0	1	0	1/22/
2237	22	1976	Graduation	Divorced	46310.0	1	0	12/3/
2238	528	1978	Graduation	Married	65819.0	0	0	11/29/
2239	4070	1969	PhD	Married	94871.0	0	2	9/1/

2240 rows × 50 columns

Creating heatmap to show correlation

- **We will create a heatmap to show correlation between the variables.**

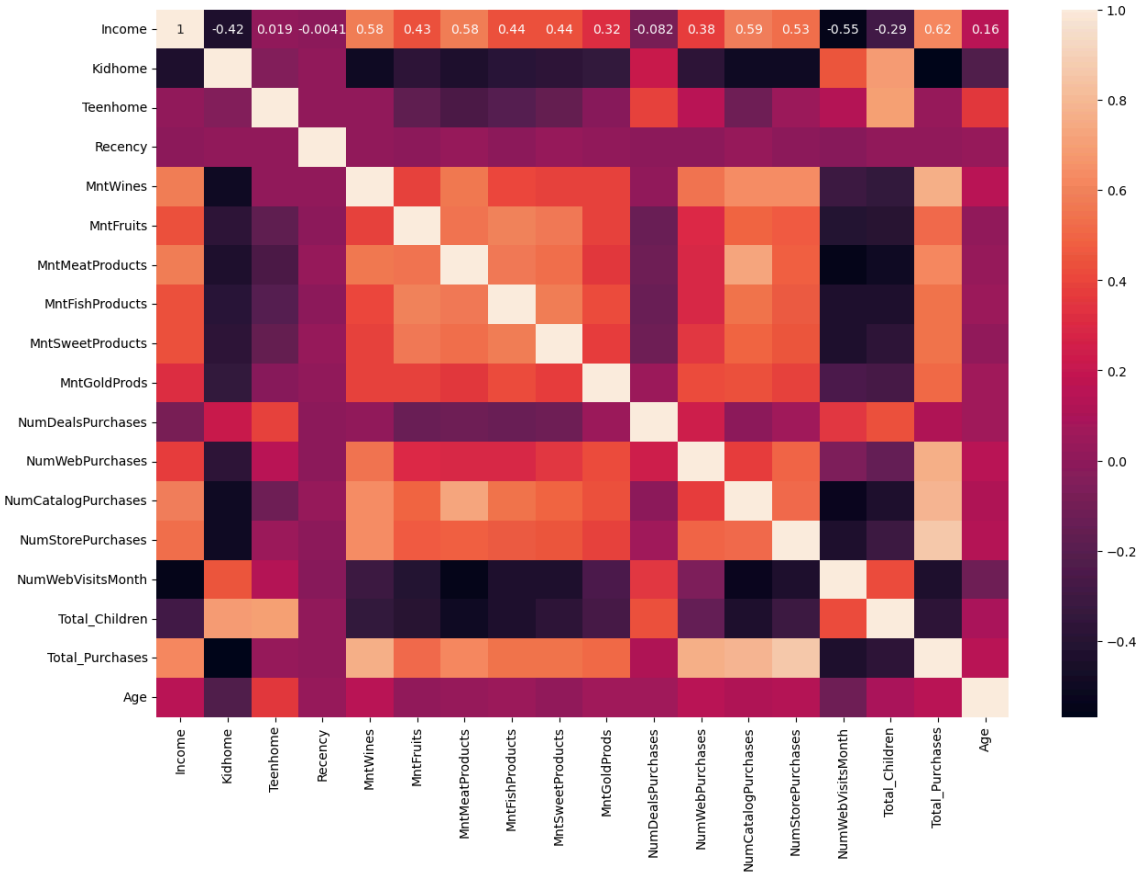
In [47]:
```
df_correlation = df[['Income', 'Kidhome', 'Teenhome',
       'Recency', 'MntWines', 'MntFruits', 'MntMeatProducts',
       'MntFishProducts', 'MntSweetProducts', 'MntGoldProds',
       'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
       'NumStorePurchases', 'NumWebVisitsMonth','Total_Children', 'Total_Pur
       ]]
df_correlation.corr()
```

Out[47]:

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntM
Income	1.000000	-0.424958	0.019249	-0.004094	0.576883	0.428975	
Kidhome	-0.424958	1.000000	-0.036133	0.008827	-0.496297	-0.372581	
Teenhome	0.019249	-0.036133	1.000000	0.016198	0.004846	-0.176764	
Recency	-0.004094	0.008827	0.016198	1.000000	0.016064	-0.004306	
MntWines	0.576883	-0.496297	0.004846	0.016064	1.000000	0.389637	
MntFruits	0.428975	-0.372581	-0.176764	-0.004306	0.389637	1.000000	
MntMeatProducts	0.576984	-0.437129	-0.261160	0.023056	0.562667	0.543105	
MntFishProducts	0.437722	-0.387644	-0.204187	0.001079	0.399753	0.594804	
MntSweetProducts	0.436292	-0.370673	-0.162475	0.022670	0.386581	0.567164	
MntGoldProds	0.321964	-0.349595	-0.021725	0.016693	0.387516	0.392995	
NumDealsPurchases	-0.081776	0.221798	0.387741	-0.001098	0.010940	-0.132114	
NumWebPurchases	0.380696	-0.361647	0.155500	-0.010726	0.542265	0.296735	
NumCatalogPurchases	0.586794	-0.502237	-0.110769	0.025110	0.635226	0.487917	
NumStorePurchases	0.526900	-0.499683	0.050695	0.000799	0.642100	0.461758	
NumWebVisitsMonth	-0.549588	0.447846	0.134884	-0.021445	-0.320653	-0.418383	
Total_Children	-0.290389	0.689971	0.698433	0.018053	-0.351909	-0.394853	
Total_Purchases	0.622564	-0.568637	0.037902	0.006410	0.756490	0.520686	
Age	0.161570	-0.230176	0.352111	0.019871	0.157773	0.017917	

In [48]:
```
plt.figure(figsize=(15,10))
sns.heatmap(df_correlation.corr(),annot=True)
```

Out[48]: <Axes: >



- **From the heatmap it is clearly visible that Total Purchases and Total Children, Total Purchases and Number of Visits are highly negative correlated, while Wines and Total Purchase are highly positive correlated.**

Hypothesis Testing

Older vs Younger Generation

- **For testing generationwise, first we need to put bins in age and put them in groups.**

In [49]:
```
bins = [0,18,32,60,140]
```

In [50]:
```
df['age_bucket'] = pd.cut(df['Age'],bins)
```

In [51]:
```
df[['Age','age_bucket']]
```

Out[51]:

	Age	age_bucket
0	44	(32, 60]
1	53	(32, 60]
2	56	(32, 60]
3	47	(32, 60]
4	25	(18, 32]
...	...	...
2235	38	(32, 60]
2236	37	(32, 60]
2237	38	(32, 60]
2238	36	(32, 60]
2239	45	(32, 60]

2240 rows × 2 columns

In [52]:
```
df['age_group'] = pd.cut(df['Age'],bins)
df['Web_Purchase'] = df['NumWebPurchases']*100/df['Total_Purchases']
df['Store_Purchase'] = df['NumStorePurchases']*100/df['Total_Purchases']
df['Catalogue_Purchase'] = df['NumCatalogPurchases']*100/df['Total_Purchases
```

In [53]:
```
df.groupby("age_group")[["Web_Purchase","Store_Purchase", "Catalogue_Purchas
```

Out[53]:

age_group	Web_Purchase	Store_Purchase	Catalogue_Purchase
(0, 18]	38.095238	54.761905	7.142857
(18, 32]	31.548070	52.835757	15.616173
(32, 60]	33.473508	50.398124	16.128368
(60, 140]	31.858576	48.349981	19.791444

In [54]:
```
df.groupby('age_bucket')[['Web_Purchase','Store_Purchase', 'Catalogue_Purcha
```

Out[54]:

age_bucket	Web_Purchase	Store_Purchase	Catalogue_Purchase
(0, 18]	38.10	54.76	7.14
(18, 32]	31.55	52.84	15.62
(32, 60]	33.47	50.40	16.13
(60, 140]	31.86	48.35	19.79

- **From the data there is no concrete evidence that older people are lesser tech savvy and are purchaing more from the store.**

Kids vs No Kids

In [55]:
```
df.groupby('Kidhome')[['Web_Purchase','Store_Purchase', 'Catalogue_Purchase'
```

Out[55]:

Kidhome	Web_Purchase	Store_Purchase	Catalogue_Purchase
0	30.95	46.98	22.07
1	35.73	55.33	8.94
2	36.51	56.11	7.38

- **It is clearly visible that the people with kids are making more web purchases. So, the hypothesis made seems to be true.**

Cannibalization

In [56]:
```
df.groupby("age_group")[["Web_Purchase","Store_Purchase", "Catalogue_Purchas
```

Out[56]:

age_group	Web_Purchase	Store_Purchase	Catalogue_Purchase
(0, 18]	38.095238	54.761905	7.142857
(18, 32]	31.548070	52.835757	15.616173
(32, 60]	33.473508	50.398124	16.128368
(60, 140]	31.858576	48.349981	19.791444

In [57]:
```
df.groupby('Kidhome')[['Web_Purchase','Store_Purchase', 'Catalogue_Purchase'
```

Out[57]:

Kidhome	Web_Purchase	Store_Purchase	Catalogue_Purchase
0	30.946758	46.982887	22.070354
1	35.731500	55.332906	8.935594
2	36.507922	56.108791	7.383288

- **From the above data, we can conclude that cannibalization is happening since 50% or more than 50% sales are happening from non-store channels.**

USA vs rest of the world

In [58]:
```
df.groupby('Country_US')['Total_Purchases'].mean()
```

Out[58]:
```
Country_US
False    12.487095
True     13.513761
Name: Total_Purchases, dtype: float64
```

- **The purchase of USA is more than rest of the world.**

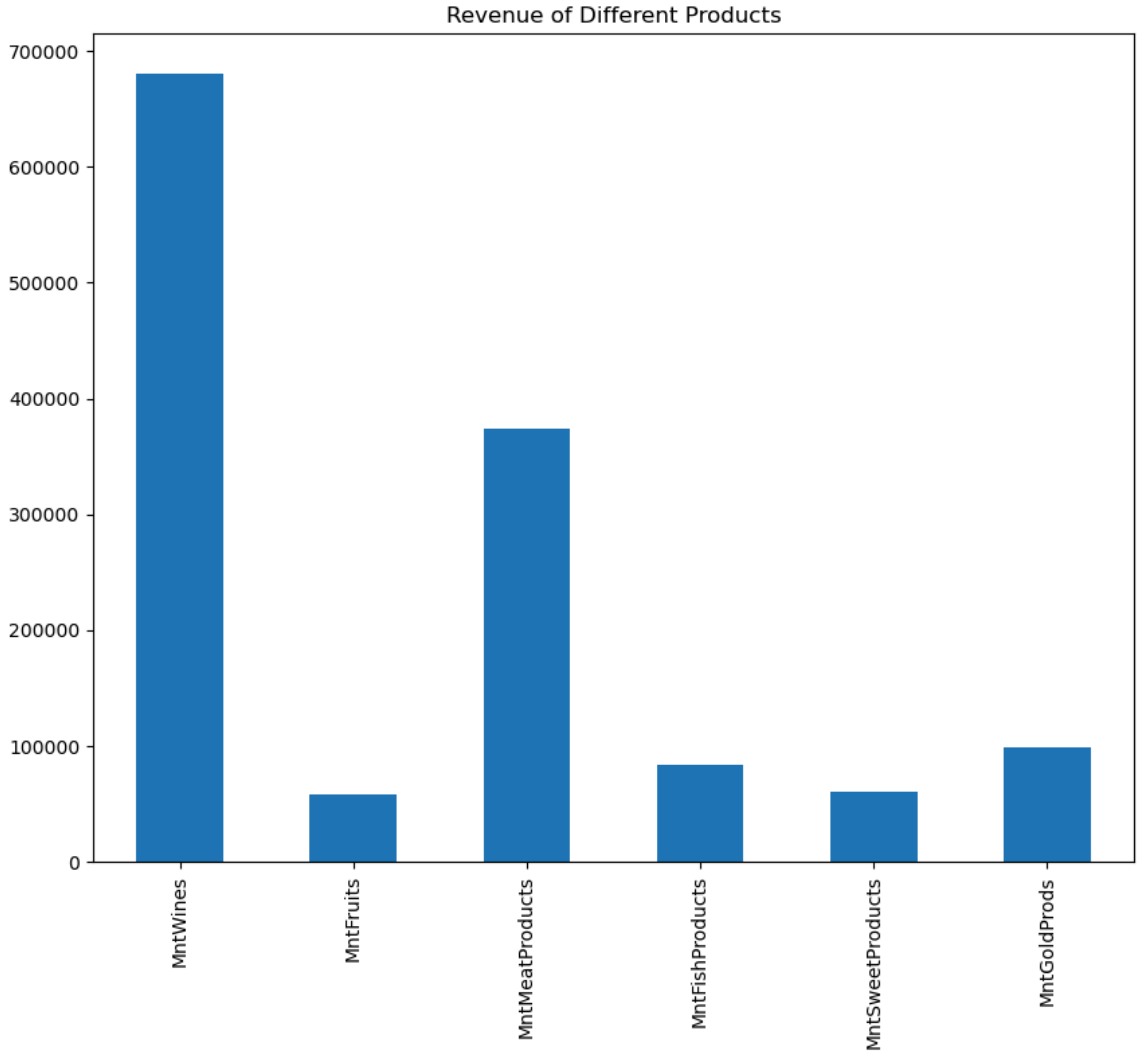
Visualisation

Product sales

In [59]:
```
plt.figure(figsize=(10,8))

revenue =  df[['MntWines','MntFruits','MntMeatProducts','MntFishProducts','M
revenue.plot(kind='bar')
plt.title('Revenue of Different Products')
```

Out[59]:
```
Text(0.5, 1.0, 'Revenue of Different Products')
```



In [60]:
```
revenue
```

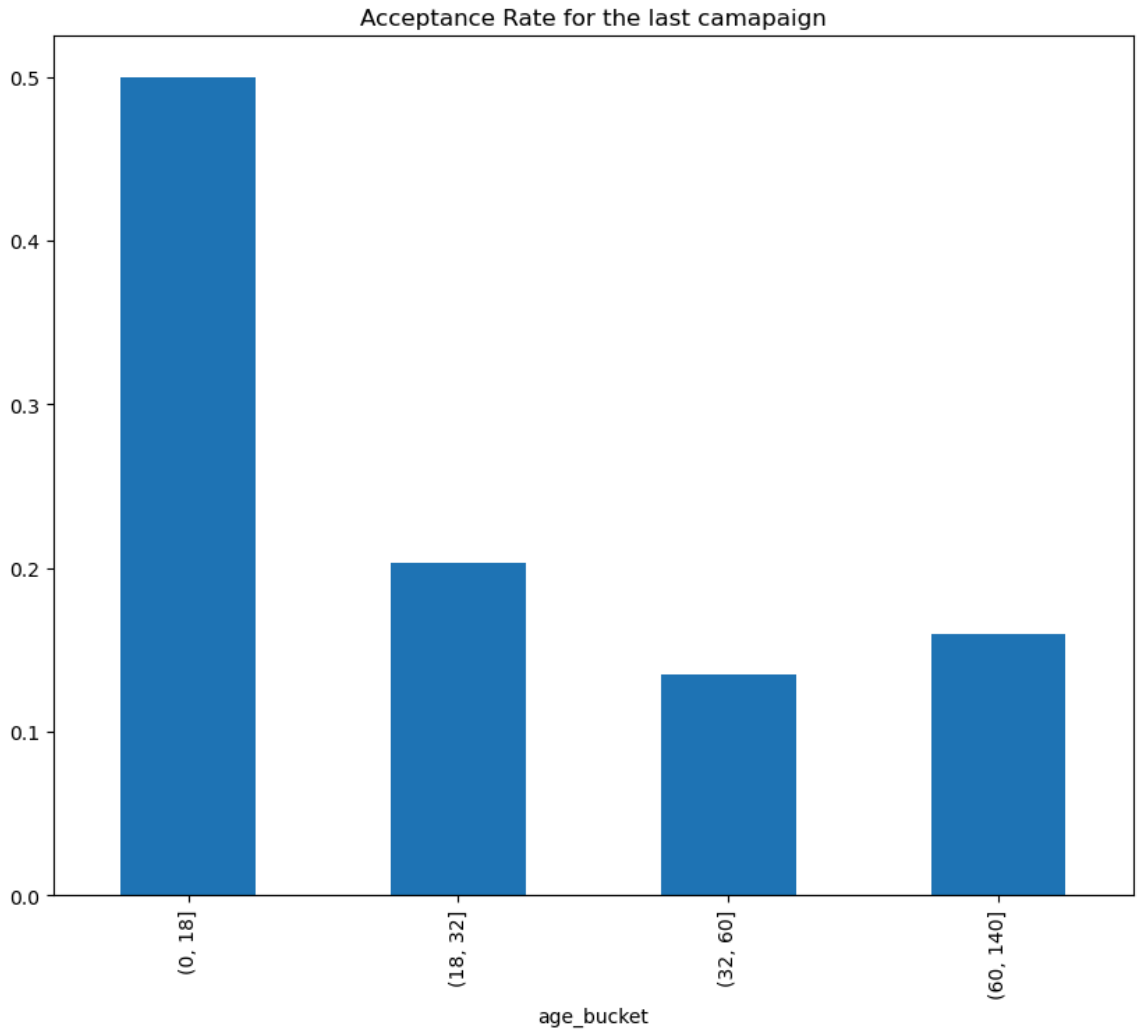
Out[60]:
```
MntWines            680816
MntFruits            58917
MntMeatProducts     373968
MntFishProducts      84057
MntSweetProducts     60621
MntGoldProds         98609
dtype: int64
```

- **Wines are making the most revenue whereas fruits are making the least revenue.**

Response Rate

In [61]:
```
plt.figure(figsize=(10,8))
res_rate = df.groupby(['age_bucket'])['Response'].mean()
res_rate.plot(kind='bar')
plt.title('Acceptance Rate for the last camapaign')
```

Out[61]:
```
Text(0.5, 1.0, 'Acceptance Rate for the last camapaign')
```



In [62]:
```
res_rate
```

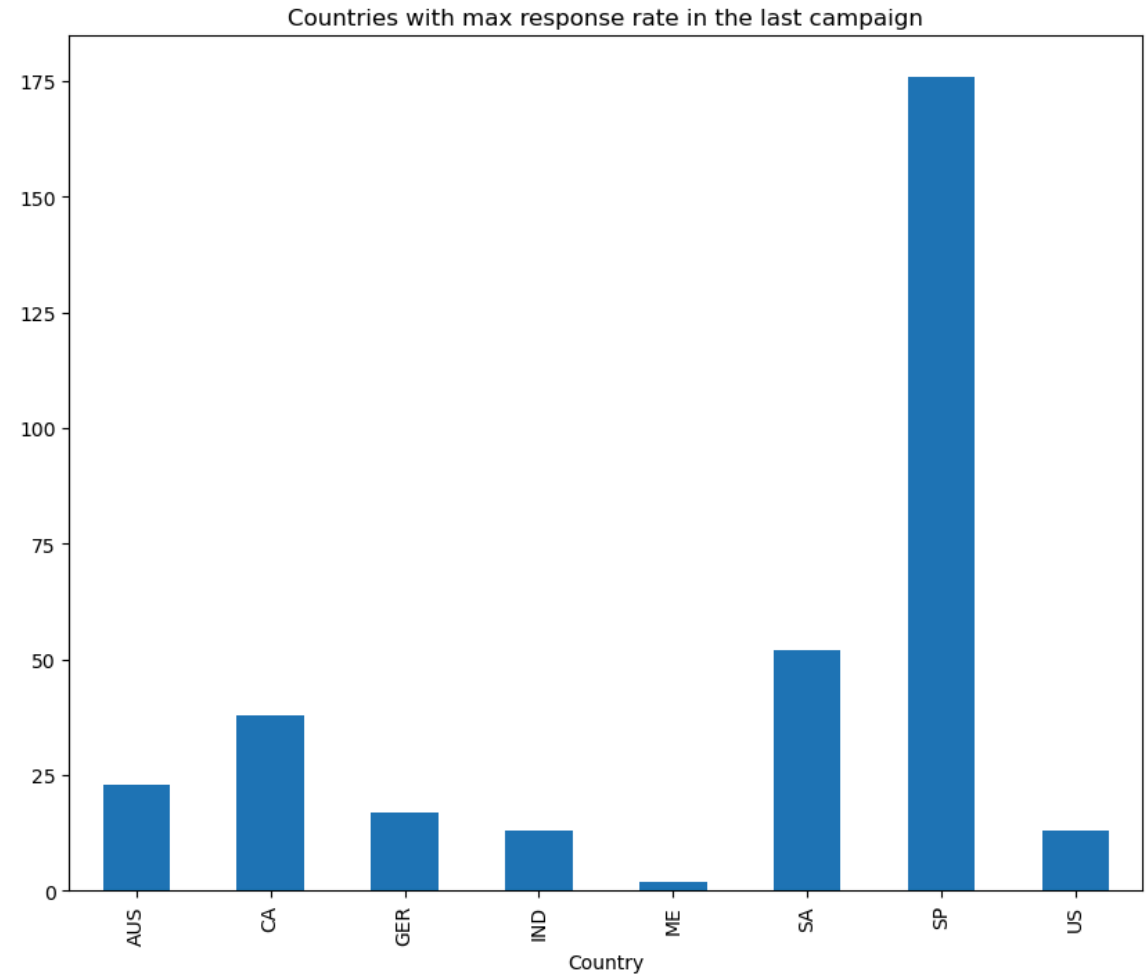
Out[62]:
```
age_bucket
(0, 18]      0.500000
(18, 32]     0.203488
(32, 60]     0.135385
(60, 140]    0.159851
Name: Response, dtype: float64
```

- **The younger generation has better response rates.The chart is following a decreasing trend except for the last bar column.**

Countries with max response rate

In [63]:
```
plt.figure(figsize=(10,8))
ctr_res_rate = df.groupby('Country')['Response'].sum()
ctr_res_rate.plot(kind='bar')
plt.title('Countries with max response rate in the last campaign')
```

Out[63]:
```
Text(0.5, 1.0, 'Countries with max response rate in the last campaign')
```



In [64]:
```
ctr_res_rate
```

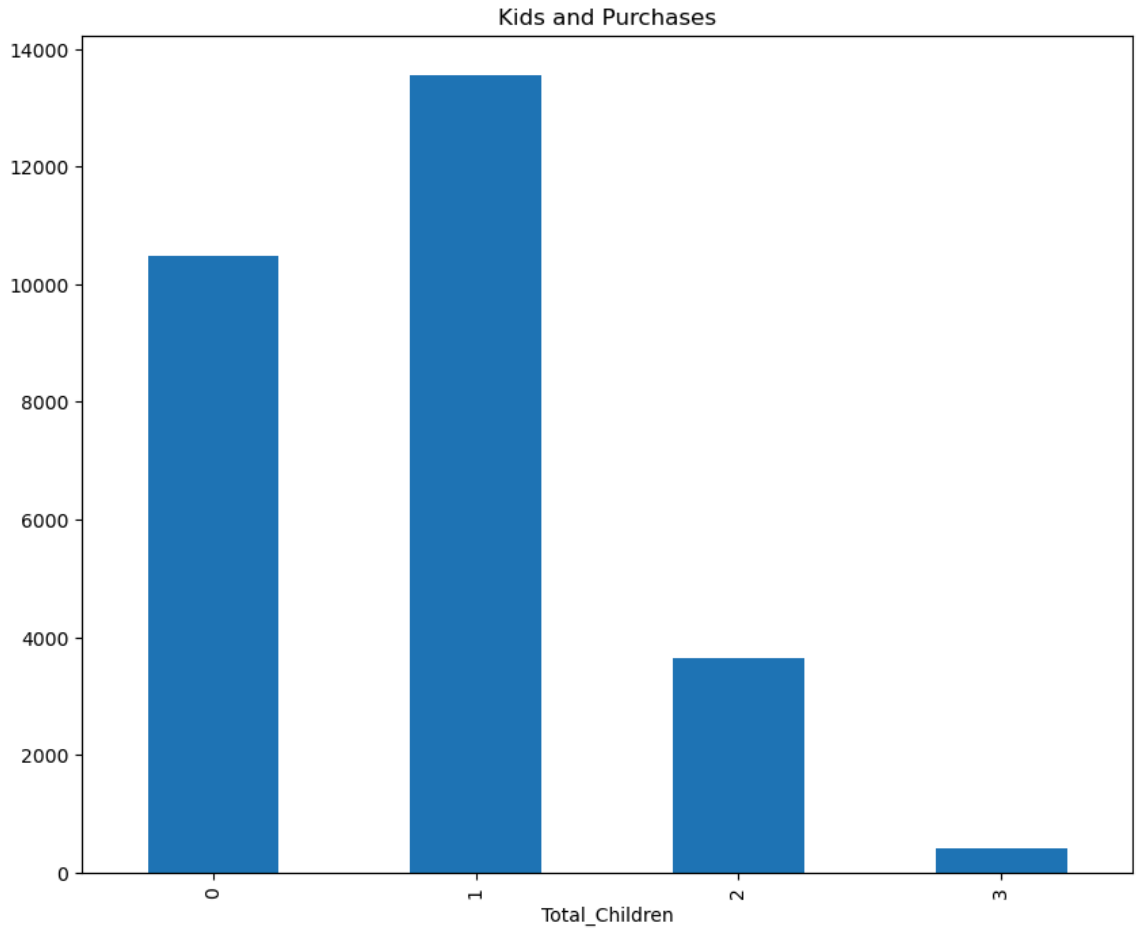
Out[64]:
```
Country
AUS      23
CA       38
GER      17
IND      13
ME        2
SA       52
SP      176
US       13
Name: Response, dtype: int64
```

- **SP is the country with most acceptance rate in last campaign.**

Kids and Purchases

In [65]:
```
plt.figure(figsize=(10,8))
kids_pchs = df.groupby('Total_Children')['Total_Purchases'].sum()
kids_pchs.plot(kind='bar')
plt.title('Kids and Purchases')
```

Out[65]:
```
Text(0.5, 1.0, 'Kids and Purchases')
```



In [66]:
```
kids_pchs
```

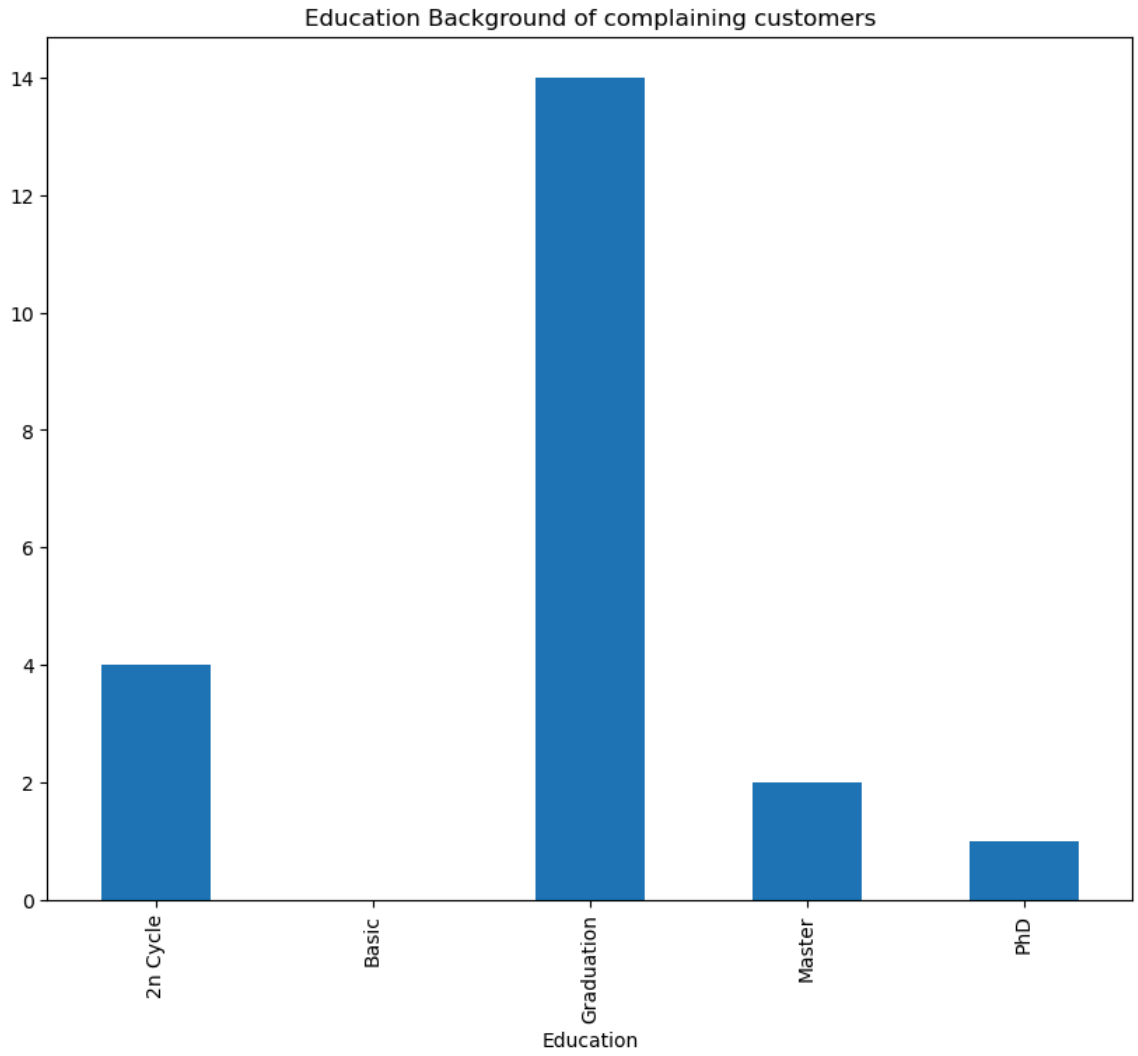
Out[66]:
```
Total_Children
0    10474
1    13551
2     3640
3      418
Name: Total_Purchases, dtype: int64
```

- **With the increasing number of children the purchasing power is decreasing.**

Education Background

In [67]:
```
plt.figure(figsize=(10,8))
edu_bkgd = df.groupby('Education')['Complain'].sum()
edu_bkgd.plot(kind='bar')
plt.title('Education Background of complaining customers')
```

Out[67]:
```
Text(0.5, 1.0, 'Education Background of complaining customers')
```



In [68]:
```
edu_bkgd
```

Out[68]:
```
Education
2n Cycle       4
Basic          0
Graduation    14
Master         2
PhD            1
Name: Complain, dtype: int64
```

- **The majority of complains have come from Graduates followed by 2nd Cycle. These two groups themselves are making more than 85% of the complains.**