In [1]:
```
#loading all the libraries

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
pd.set_option('display.max_columns', None)
%matplotlib inline

sns.set_context('notebook')
sns.set_style('whitegrid')
sns.set_palette('Blues_r')
```

In [4]:
```
# turning off all the warnings for the final notebook

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

In [5]:
```
# loading the dataset

df = pd.read_csv('marketing_data.csv')
```

In [7]:
```
df.head()
```

Out[7]:

	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

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

Out[8]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_C
2235	10142	1976	PhD	Divorced	$66,476.00	0	1	
2236	5263	1977	2n Cycle	Married	$31,056.00	1	0	
2237	22	1976	Graduation	Divorced	$46,310.00	1	0	
2238	528	1978	Graduation	Married	$65,819.00	0	0	
2239	4070	1969	PhD	Married	$94,871.00	0	2	

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
```

In [12]:
```
#clean up the coloumns name that contain white space

df.columns = df.columns.str.replace(' ', '')

#lets transform the income coloumn to the numerical

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

In [13]:
```
#lets check out the clean dataset

df.head()
```

Out[13]:

	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/

In [16]:
```
#identify the null values

df.isnull().sum().sort_values(ascending = True)
```

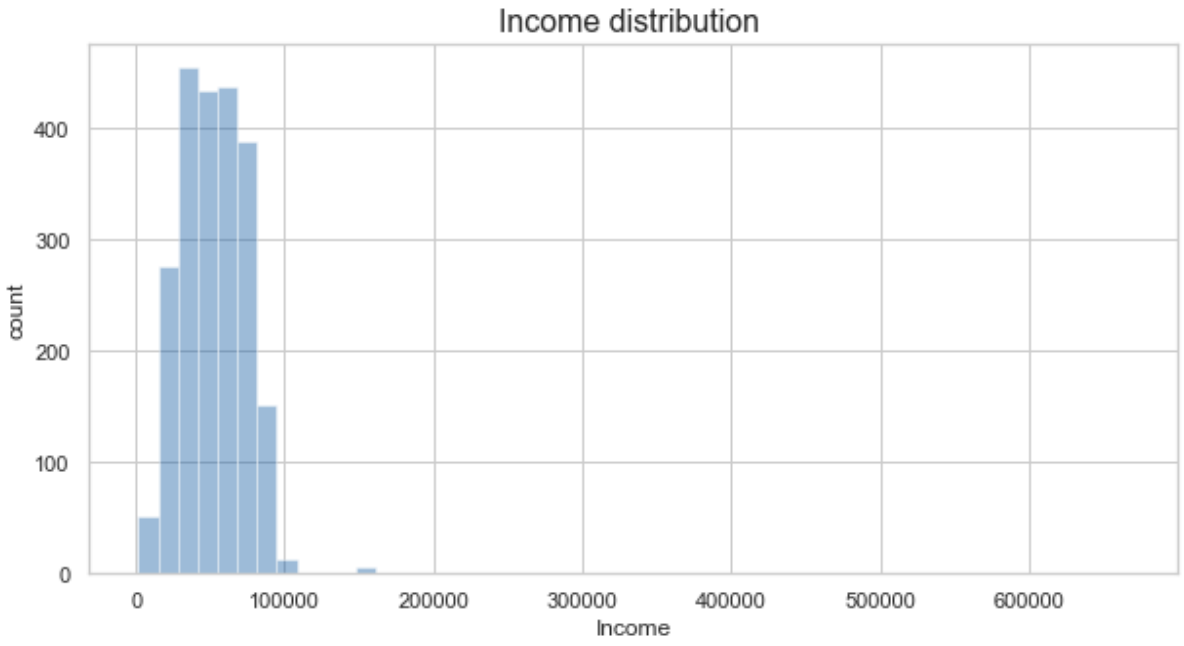
```
#we have the null values in the income coloumn
```

Out[16]:

```
ID                     0
Response               0
AcceptedCmp2           0
AcceptedCmp1           0
AcceptedCmp5           0
AcceptedCmp4           0
AcceptedCmp3           0
NumWebVisitsMonth      0
NumStorePurchases      0
NumCatalogPurchases    0
NumWebPurchases        0
NumDealsPurchases      0
MntGoldProds           0
MntSweetProducts       0
MntFishProducts        0
MntMeatProducts        0
MntFruits              0
MntWines               0
Recency                0
Dt_Customer            0
Teenhome               0
Kidhome                0
Marital_Status         0
Education              0
Year_Birth             0
Complain               0
Country                0
Income                24
dtype: int64
```

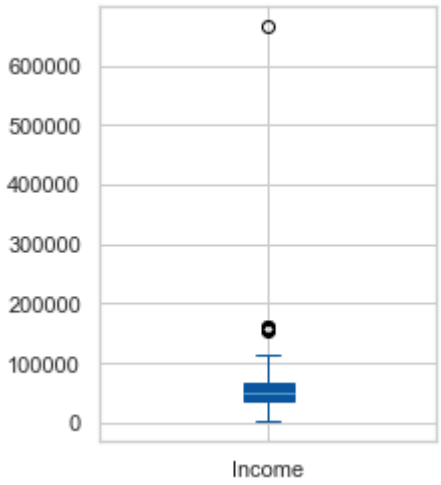
In [18]:

```
#visualizing the income coloumn

plt.figure(figsize=(10,5))
sns.distplot(df['Income'], kde = False, hist = True)
plt.title('Income distribution ', size = 16)
plt.ylabel('count');


#looking the distribution we can see that its right skewed and some outliers
```



In [19]:
```
df['Income'].plot(kind='box', figsize=(3,4), patch_artist=True)
```

Out[19]:
```
<AxesSubplot:>
```

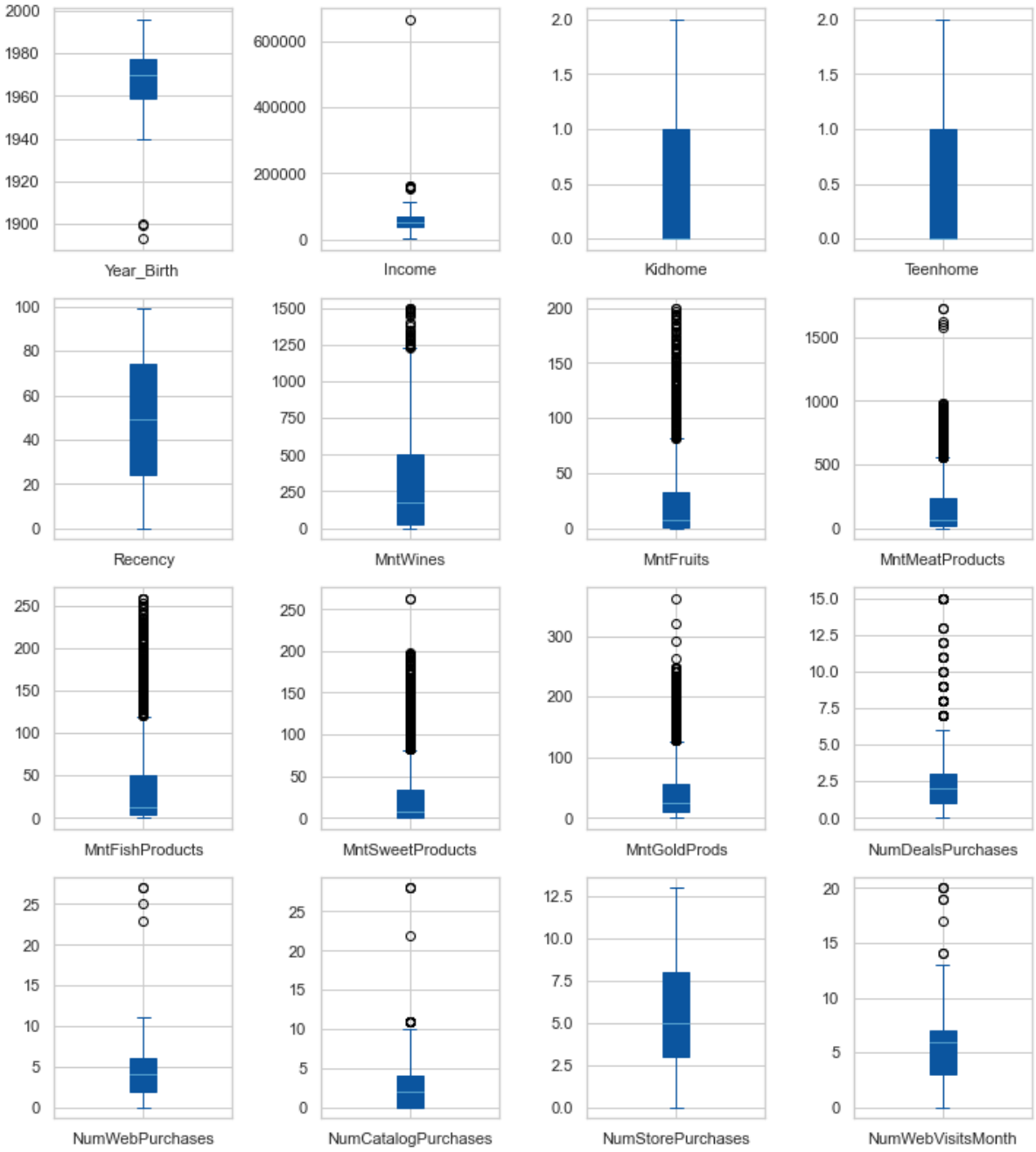


In [20]:
```
#so we have the missing values in the income coloumn we are filling it with

df['Income'] = df['Income'].fillna(df['Income'].median())
```

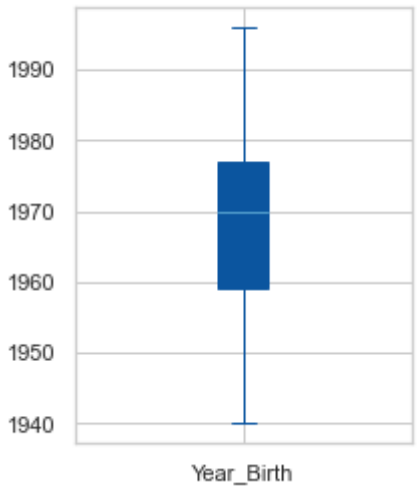
In [26]:
```
df.isnull().sum().sort_values(ascending = True)
```

Out[26]:
```
ID                   0
Response             0
AcceptedCmp2         0
AcceptedCmp1         0
AcceptedCmp5         0
AcceptedCmp4         0
AcceptedCmp3         0
NumWebVisitsMonth    0
NumStorePurchases    0
NumCatalogPurchases  0
NumWebPurchases      0
NumDealsPurchases    0
MntGoldProds         0
MntSweetProducts     0
MntFishProducts      0
MntMeatProducts      0
MntFruits            0
MntWines             0
Recency              0
Dt_Customer          0
Teenhome             0
Kidhome              0
Income               0
Marital_Status       0
Education            0
Year_Birth           0
Complain             0
Country              0
dtype: int64
```

In [27]:
```
# select columns to plot
df_to_plot = df.drop(columns=['ID', 'AcceptedCmp1', 'AcceptedCmp2', 'Accepte

# subplots
df_to_plot.plot(subplots=True, layout=(4,4), kind='box', figsize=(12,14), pa
plt.subplots_adjust(wspace=0.5);
```



In [28]:
```
df = df[df['Year_Birth'] > 1900].reset_index(drop=True)

plt.figure(figsize=(3,4))
df['Year_Birth'].plot(kind='box', patch_artist=True);
```




```
df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'])
```

In [29]:

In [30]:
```
list(df.columns)
```

Out[30]:
```
['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']
```

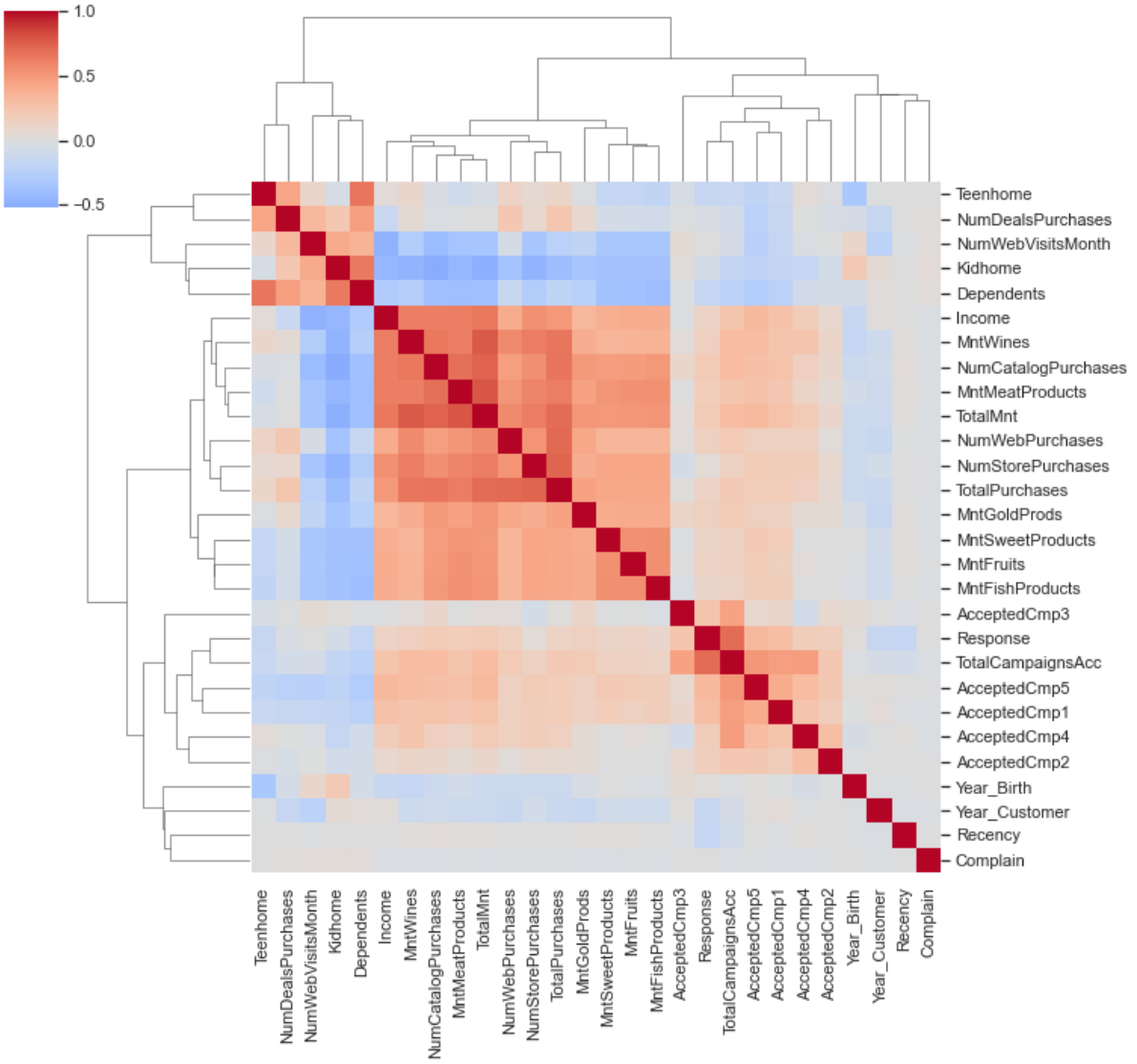
In [32]:
```
# FEATURE ENG.


#Dependents
df['Dependents'] = df['Kidhome'] + df['Teenhome']

# Year becoming a Customer
df['Year_Customer'] = pd.DatetimeIndex(df['Dt_Customer']).year

# Total Amount Spent
mnt_cols = [col for col in df.columns if 'Mnt' in col]
df['TotalMnt'] = df[mnt_cols].sum(axis=1)

# Total Purchases
purchases_cols = [col for col in df.columns if 'Purchases' in col]
df['TotalPurchases'] = df[purchases_cols].sum(axis=1)

# Total Campaigns Accepted
campaigns_cols = [col for col in df.columns if 'Cmp' in col] + ['Response']
df['TotalCampaignsAcc'] = df[campaigns_cols].sum(axis=1)

# view new features, by customer ID
df[['ID', 'Dependents', 'Year_Customer', 'TotalMnt', 'TotalPurchases', 'Tota
```

Out[32]:

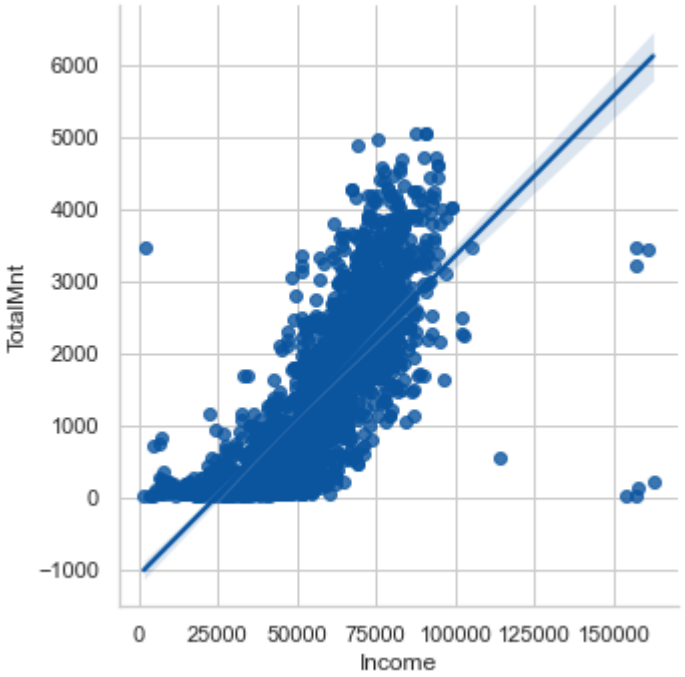
	ID	Dependents	Year_Customer	TotalMnt	TotalPurchases	TotalCampaignsAcc
0	1826	0	2014	2380	30	1
1	1	0	2014	1154	36	2
2	10476	1	2014	502	22	0
3	1386	2	2014	22	8	0
4	5371	1	2014	182	16	2
5	7348	0	2014	2384	34	1
6	4073	0	2014	2430	56	2
7	1991	1	2014	192	14	0
8	4047	1	2014	1088	40	0
9	9477	1	2014	1088	40	0

In [33]:

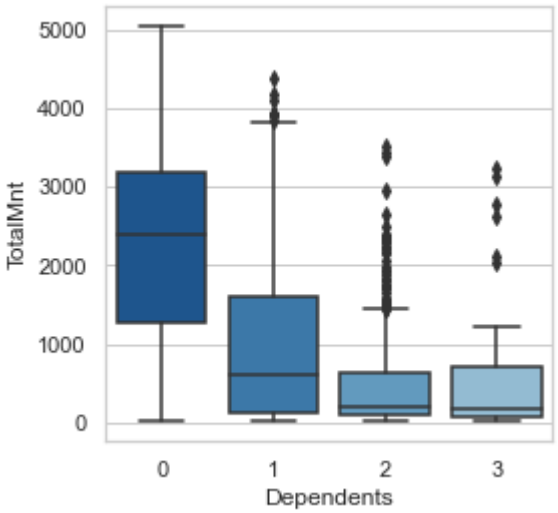
```
# calculate correlation matrix
## using non-parametric test of correlation (kendall), since some features a

corrs = df.drop(columns='ID').select_dtypes(include=np.number).corr(method =

# plot clustered heatmap of correlations
    sns.clustermap(corrs, cbar_pos=(-0.05, 0.8, 0.05, 0.18), cmap='coolwarm'
```



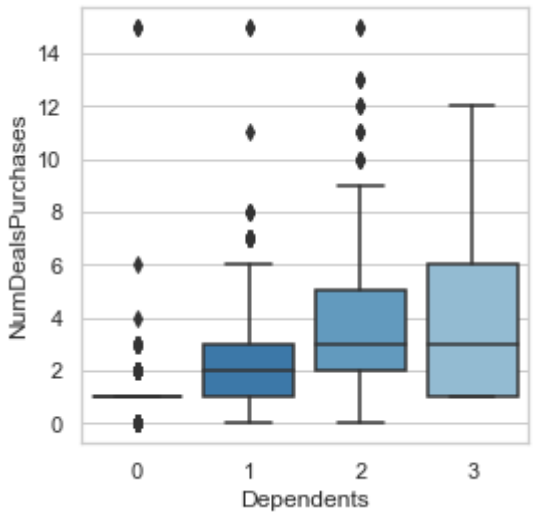
In [34]:
```
sns.lmplot(x='Income', y='TotalMnt', data=df[df['Income'] < 200000]);
```



In [35]:
```
plt.figure(figsize=(4,4))
sns.boxplot(x='Dependents', y='TotalMnt', data=df);
```



In [36]:
```
plt.figure(figsize=(4,4))
sns.boxplot(x='Dependents', y='NumDealsPurchases', data=df);
```



In [ ]: