In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

In [2]:

```
df = pd.read_csv('26_problem1.csv')
df
```

Out[2]:

	ID	Year_Birth	Marital_Status	Income	Kidhome	Teenhome	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
0	5524	1957	Single	58138.0	0	0	635	88	546	172	88	3	8	10	4	7
1	2174	1954	Single	46344.0	1	1	11	1	6	2	1	2	1	1	2	5
2	4141	1965	Together	71613.0	0	0	426	49	127	111	21	1	8	2	10	4
3	6182	1984	Together	26646.0	1	0	11	4	20	10	3	2	2	0	4	6
4	5324	1981	Married	58293.0	1	0	173	43	118	46	27	5	5	3	6	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2235	10870	1967	Married	61223.0	0	1	709	43	182	42	118	2	9	3	4	5
2236	4001	1946	Together	64014.0	2	1	406	0	30	0	0	7	8	2	5	7
2237	7270	1981	Divorced	56981.0	0	0	908	48	217	32	12	1	2	3	13	6
2238	8235	1956	Together	69245.0	0	1	428	30	214	80	30	2	6	5	10	3
2239	9405	1954	Married	52869.0	1	1	84	3	61	2	1	3	3	1	4	7

2240 rows × 16 columns

In [3]:

```
df.isna().sum()
```

Out[3]:

```
ID                      0
Year_Birth              0
Marital_Status          0
Income                 24
Kidhome                 0
Teenhome                0
MntWines                0
MntFruits               0
MntMeatProducts         0
MntFishProducts         0
MntSweetProducts        0
NumDealsPurchases       0
NumWebPurchases         0
NumCatalogPurchases     0
NumStorePurchases       0
NumWebVisitsMonth       0
dtype: int64
```

In [4]:

```
df.columns
```

Out[4]:

```
Index(['ID', 'Year_Birth', 'Marital_Status', 'Income', 'Kidhome', 'Teenhome',
       'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',
       'MntSweetProducts', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth'],
      dtype='object')
```

In [5]:

```
df[df['Income'].isna()]
```

Out[5]:

	ID	Year_Birth	Marital_Status	Income	Kidhome	Teenhome	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
10	1994	1983	Married	NaN	1	0	5	5	6	0	2	1	1	0	2	7
27	5255	1986	Single	NaN	1	0	5	1	3	3	263	0	27	0	0	1
43	7281	1959	Single	NaN	0	0	81	11	50	3	2	1	1	3	4	2
48	7244	1951	Single	NaN	2	1	48	5	48	6	10	3	2	1	4	6
58	8557	1982	Single	NaN	1	0	11	3	22	2	2	2	2	0	3	6
71	10629	1973	Married	NaN	1	0	25	3	43	17	4	3	3	0	3	8
90	8996	1957	Married	NaN	2	1	230	42	192	49	37	12	7	2	8	9
91	9235	1957	Single	NaN	1	1	7	0	8	2	0	1	1	0	2	7
92	5798	1973	Together	NaN	0	0	445	37	359	98	28	1	2	4	8	1
128	8268	1961	Married	NaN	0	1	352	0	27	10	0	3	6	1	7	6
133	1295	1963	Married	NaN	0	1	231	65	196	38	71	1	6	5	7	4
312	2437	1989	Married	NaN	0	0	861	138	461	60	30	1	6	5	12	3
319	2863	1970	Single	NaN	1	2	738	20	172	52	50	6	2	3	10	7
1379	10475	1970	Together	NaN	0	1	187	5	65	26	20	2	4	2	6	5
1382	2902	1958	Together	NaN	1	1	19	4	12	2	2	1	1	0	3	5
1383	4345	1964	Single	NaN	1	1	5	1	9	2	0	1	1	0	2	7
1386	3769	1972	Together	NaN	1	0	25	1	13	0	0	1	1	0	3	7
2059	7187	1969	Together	NaN	1	1	375	42	48	94	66	7	4	10	4	3
2061	1612	1981	Single	NaN	1	0	23	0	15	0	2	2	3	0	3	6
2078	5079	1971	Married	NaN	1	1	71	1	16	0	0	4	2	1	3	8
2079	10339	1954	Together	NaN	0	1	161	0	22	0	0	2	4	1	4	6
2081	3117	1955	Single	NaN	0	1	264	0	21	12	6	3	6	1	5	7
2084	5250	1943	Widow	NaN	0	0	532	126	490	164	126	1	5	5	11	1
2228	8720	1978	Together	NaN	0	0	32	2	1607	12	4	0	0	0	1	0

In [6]:

```
df.shape
```

Out[6]:

```
(2240, 16)
```

In [7]:

```
df = df.dropna()
```

In [8]:

```
df.shape
```

Out[8]:

```
(2216, 16)
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2216 entries, 0 to 2239
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2216 non-null   int64  
 1   Year_Birth           2216 non-null   int64  
 2   Marital_Status       2216 non-null   object 
 3   Income               2216 non-null   float64
 4   Kidhome              2216 non-null   int64  
 5   Teenhome             2216 non-null   int64  
 6   MntWines             2216 non-null   int64  
 7   MntFruits            2216 non-null   int64  
 8   MntMeatProducts      2216 non-null   int64  
 9   MntFishProducts      2216 non-null   int64  
 10  MntSweetProducts     2216 non-null   int64  
 11  NumDealsPurchases    2216 non-null   int64  
 12  NumWebPurchases      2216 non-null   int64  
 13  NumCatalogPurchases  2216 non-null   int64  
 14  NumStorePurchases    2216 non-null   int64  
 15  NumWebVisitsMonth    2216 non-null   int64  
dtypes: float64(1), int64(14), object(1)
memory usage: 294.3+ KB
```

In [10]:

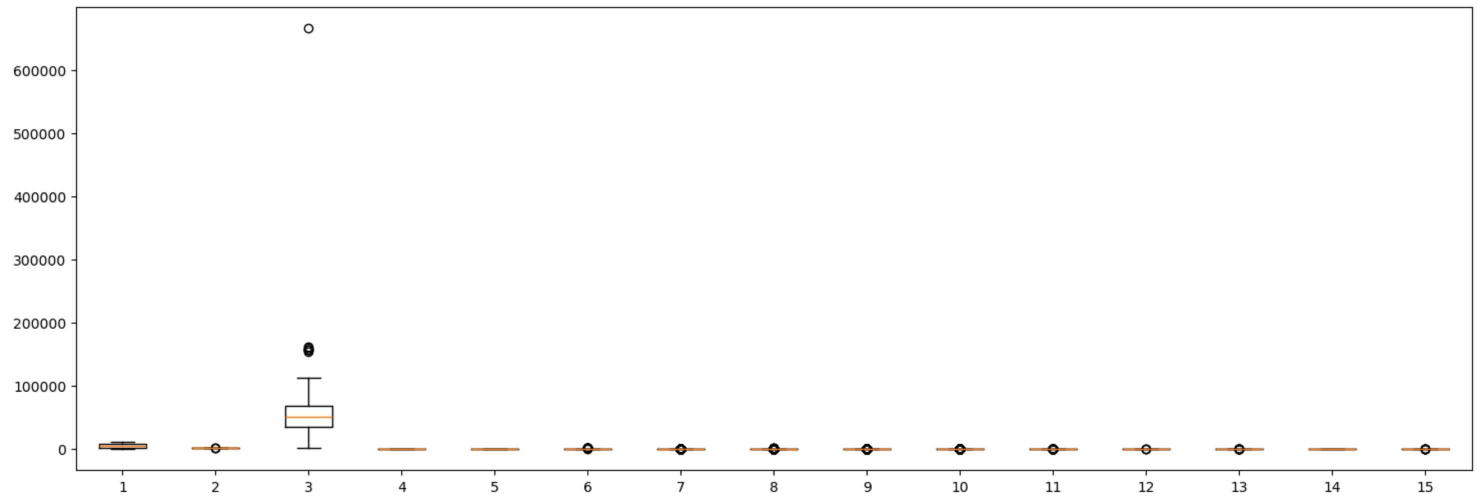
```
df.isna().sum()
```

Out[10]:

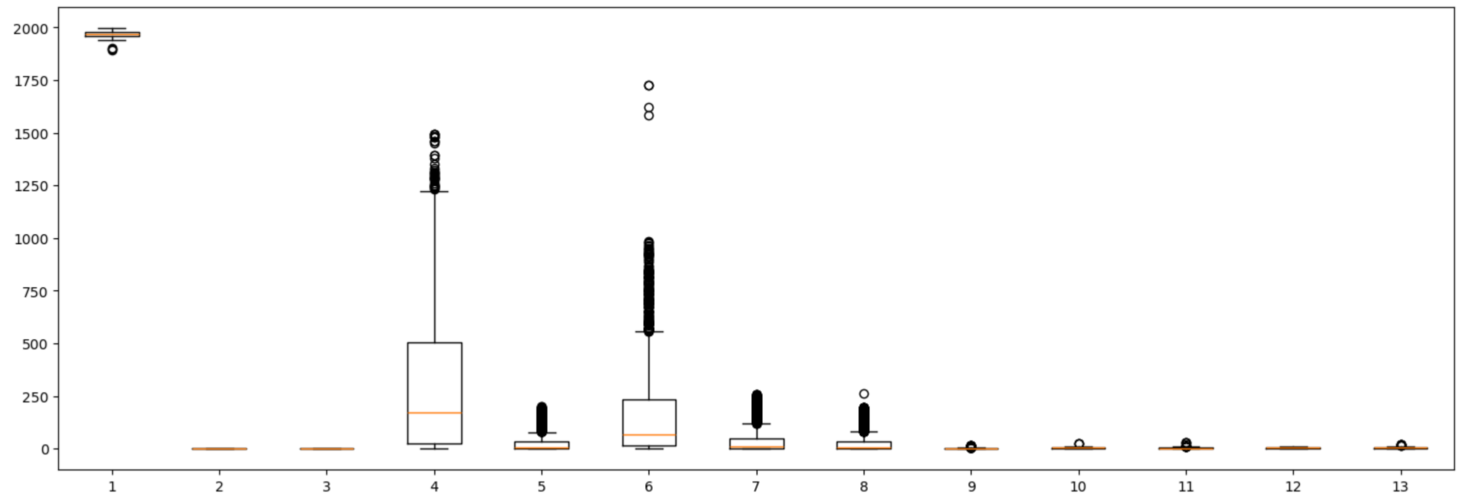
```
ID                     0
Year_Birth             0
Marital_Status         0
Income                 0
Kidhome                0
Teenhome               0
MntWines               0
MntFruits              0
MntMeatProducts        0
MntFishProducts        0
MntSweetProducts       0
NumDealsPurchases      0
NumWebPurchases        0
NumCatalogPurchases    0
NumStorePurchases      0
NumWebVisitsMonth      0
dtype: int64
```

In [21]:

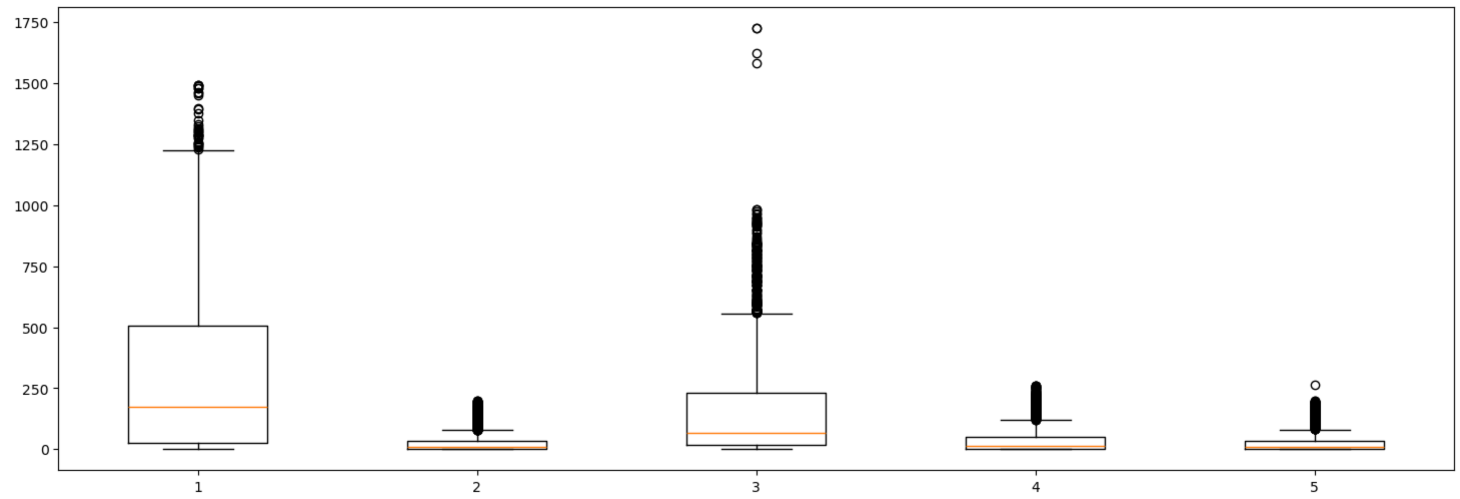
```
fig =plt.figure(figsize=(18,6))
ax= fig.add_subplot(111)
ax.boxplot(df.drop(columns='Marital_Status'))
plt.show()
```



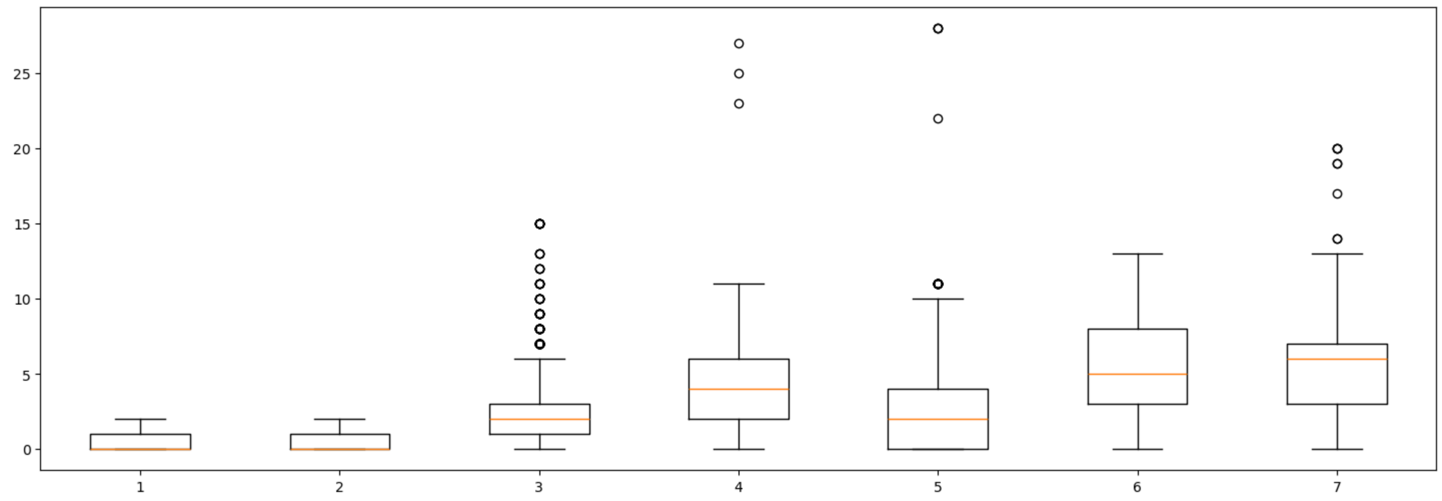
In [22]:
```
fig =plt.figure(figsize=(18,6))
ax= fig.add_subplot(111)
ax.boxplot(df.drop(columns=['ID','Marital_Status','Income']))
plt.show()
```



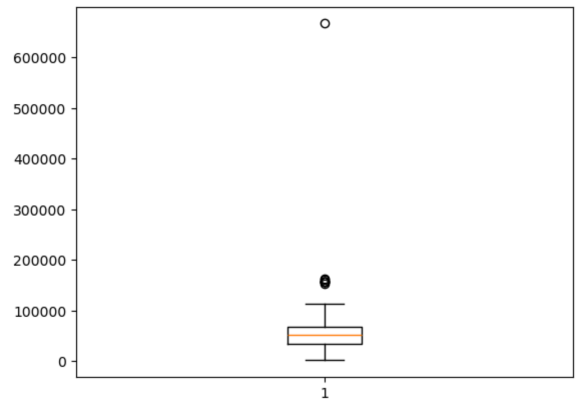
In [27]:
```
fig =plt.figure(figsize=(18,6))
ax= fig.add_subplot(111)
ax.boxplot(df[['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts','MntSweetProducts']])
plt.show()
```



In [29]:
```
fig =plt.figure(figsize=(18,6))
ax= fig.add_subplot(111)
ax.boxplot(df[['Kidhome', 'Teenhome', 'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth']])
plt.show()
```



In [31]:
```
#Index(['ID', 'Year_Birth', 'Marital_Status', 'Income', 'Kidhome', 'Teenhome','MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',\
#'MntSweetProducts', 'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth']
plt.boxplot(df['Income'])
plt.show()
```



In [33]:
```
def outliers_iqr(dt,col):
    quantile_1, quantile_3 = np.percentile(df[col], [25, 75])
    iqr = quantile_3 - quantile_1
    lower_whis = quantile_1 - (iqr * 1.5)
    upper_whis = quantile_3 + (iqr * 1.5)
    outliers = df[(df[col] > upper_whis) | (df[col] < lower_whis)]
    return outliers[[col]]
```

In [35]:
```
outliers_Income = outliers_iqr(df, 'Income')
outliers_Income
```

Out[35]:

	Income
164	157243.0
617	162397.0
655	153924.0
687	160803.0
1300	157733.0
1653	157146.0
2132	156924.0
2233	666666.0

In [37]:
```
df.loc[outliers_Income.index, 'Income'] = np.NaN

df['Income']
```

Out[37]:
```
0       58138.0
1       46344.0
2       71613.0
3       26646.0
4       58293.0
         ...
2235    61223.0
2236    64014.0
2237    56981.0
2238    69245.0
2239    52869.0
Name: Income, Length: 2216, dtype: float64
```

In [39]:
```
df['Income'] = df['Income'].fillna(df['Income'].mean())
```

```
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\1783921682.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Income'] = df['Income'].fillna(df['Income'].mean())
```

In [41]:
```
df.loc[outliers_Income.index, 'Income']
```

Out[41]:
```
164     51633.638134
617     51633.638134
655     51633.638134
687     51633.638134
1300    51633.638134
1653    51633.638134
2132    51633.638134
2233    51633.638134
Name: Income, dtype: float64
```

In [43]:
```
df['Income']
```

Out[43]:
```
0       58138.0
1       46344.0
2       71613.0
3       26646.0
4       58293.0
         ...
2235    61223.0
2236    64014.0
2237    56981.0
2238    69245.0
2239    52869.0
Name: Income, Length: 2216, dtype: float64
```

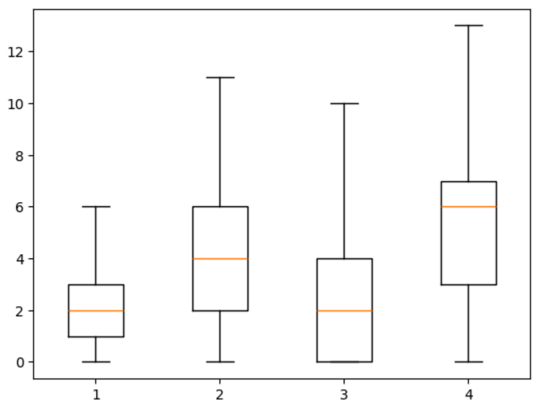
In [45]:
```
list = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts','MntSweetProducts', 'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases','NumWebVisitsMonth']
for x in list:
    outliers_x = outliers_iqr(df, x)
    print(outliers_x)
    df.loc[outliers_x.index, x] = np.NaN
    df[x] = df[x].fillna(df[x].mean())
    print(df.loc[outliers_x.index, x])
```

```
      MntWines
111       1332
161       1349
295       1241
424       1285
430       1248
466       1239
497       1396
515       1288
523       1379
543       1478
559       1492
824       1492
826       1279
870       1308
917       1478
937       1253
987       1394
990       1296
1001      1285
1010      1230
1052      1315
1191      1298
1458      1302
1488      1449
1492      1259
1577      1252
1641      1459
1749      1493
1922      1324
1953      1285
1961      1462
1992      1276
2067      1245
2098      1486
2127      1311
111     288.45713
161     288.45713
295     288.45713
424     288.45713
430     288.45713
466     288.45713
497     288.45713
515     288.45713
523     288.45713
543     288.45713
559     288.45713
824     288.45713
826     288.45713
870     288.45713
917     288.45713
937     288.45713
987     288.45713
990     288.45713
1001    288.45713
1010    288.45713
1052    288.45713
1191    288.45713
1458    288.45713
1488    288.45713
1492    288.45713
1577    288.45713
1641    288.45713
1749    288.45713
1922    288.45713
1953    288.45713
1961    288.45713
1992    288.45713
2067    288.45713
2098    288.45713
2127    288.45713
Name: MntWines, dtype: float64
      MntFruits
0            88
18           80
29          100
45          164
53          120
...         ...
2185        142
2194         80
2203        124
2206        129
2217        194

[246 rows x 1 columns]
0       14.335025
18      14.335025
29      14.335025
45      14.335025
53      14.335025
          ...
2185    14.335025
2194    14.335025
2203    14.335025
2206    14.335025
2217    14.335025
Name: MntFruits, Length: 246, dtype: float64
      MntMeatProducts
21               1725
29                801
51                780
76                925
77                779
...               ...
2187              749
2190              655
2193              845
2211              860
2213              631

[174 rows x 1 columns]
21      116.727718
29      116.727718
51      116.727718
76      116.727718
77      116.727718
           ...
2187    116.727718
2190    116.727718
2193    116.727718
2211    116.727718
2213    116.727718
Name: MntMeatProducts, Length: 174, dtype: float64
      MntFishProducts
0                 172
12                225
17                150
39                160
45                227
...               ...
2188              199
2190              145
2193              202
2206              182
2217              149

[222 rows x 1 columns]
0       22.357071
12      22.357071
17      22.357071
39      22.357071
45      22.357071
          ...
2188    22.357071
2190    22.357071
2193    22.357071
2206    22.357071
2217    22.357071
Name: MntFishProducts, Length: 222, dtype: float64
      MntSweetProducts
0                   88
12                 112
40                 178
51                 167
55                 120
...                ...
2175                92
2190               111
2193               133
2217               125
2235               118

[246 rows x 1 columns]
0       14.577157
12      14.577157
40      14.577157
51      14.577157
55      14.577157
          ...
2175    14.577157
2190    14.577157
2193    14.577157
2217    14.577157
2235    14.577157
```

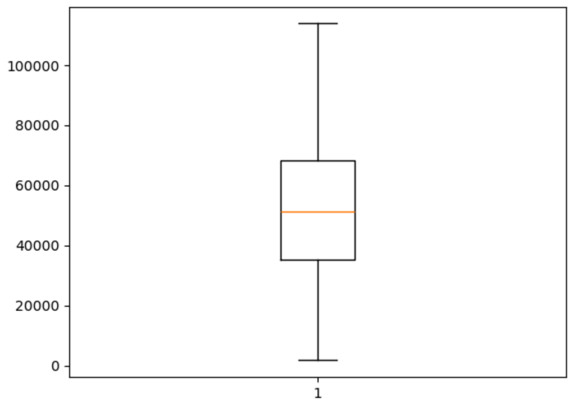
```
Name: MntSweetProducts, Length: 246, dtype: float64
      NumDealsPurchases
21                   15
24                    7
49                    9
54                    7
69                    7
...                 ...
2090                  7
2144                  7
2198                  7
2226                  8
2236                  7

[84 rows x 1 columns]
21      2.067073
24      2.067073
49      2.067073
54      2.067073
69      2.067073
          ...   
2090    2.067073
2144    2.067073
2198    2.067073
2226    2.067073
2236    2.067073
Name: NumDealsPurchases, Length: 84, dtype: float64
      NumWebPurchases
1806               23
1898               27
1975               25
1806    4.056936
1898    4.056936
1975    4.056936
Name: NumWebPurchases, dtype: float64
      NumCatalogPurchases
21                     28
104                    11
164                    22
288                    11
586                    11
591                    11
627                    11
636                    11
687                    28
764                    11
777                    11
934                    11
984                    11
1212                   11
1452                   11
1465                   11
1492                   11
1653                   28
1745                   11
1828                   11
1906                   11
1940                   11
1958                   11
21      2.555404
104     2.555404
164     2.555404
288     2.555404
586     2.555404
591     2.555404
627     2.555404
636     2.555404
687     2.555404
764     2.555404
777     2.555404
934     2.555404
984     2.555404
1212    2.555404
1452    2.555404
1465    2.555404
1492    2.555404
1653    2.555404
1745    2.555404
1828    2.555404
1906    2.555404
1940    2.555404
1958    2.555404
Name: NumCatalogPurchases, dtype: float64
      NumWebVisitsMonth
9                    20
774                  20
981                  14
1042                 19
1245                 20
1328                 17
1524                 14
1846                 19
9       5.273551
774     5.273551
981     5.273551
1042    5.273551
1245    5.273551
1328    5.273551
1524    5.273551
1846    5.273551
Name: NumWebVisitsMonth, dtype: float64
```

```
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3641009375.py:6: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[x] = df[x].fillna(df[x].mean())
```

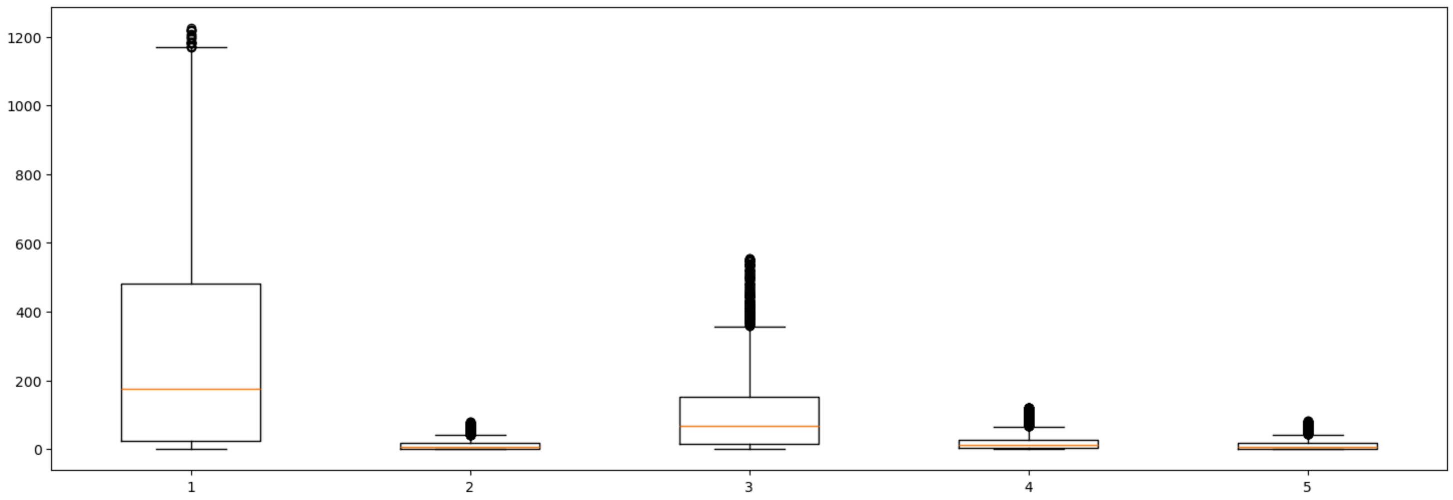
```
In [49]: plt.boxplot(df[['NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases','NumWebVisitsMonth']])
         plt.show()
```



In [51]:
```
plt.boxplot(df['Income'])
plt.show()
```



In [53]:
```
fig =plt.figure(figsize=(18,6))
ax= fig.add_subplot(111)
ax.boxplot(df[['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts','MntSweetProducts']])
plt.show()
```



군집생성하기

In [56]:
```
print(set(df['Marital_Status']))
```

```
{'YOLO', 'Together', 'Widow', 'Single', 'Absurd', 'Alone', 'Divorced', 'Married'}
```

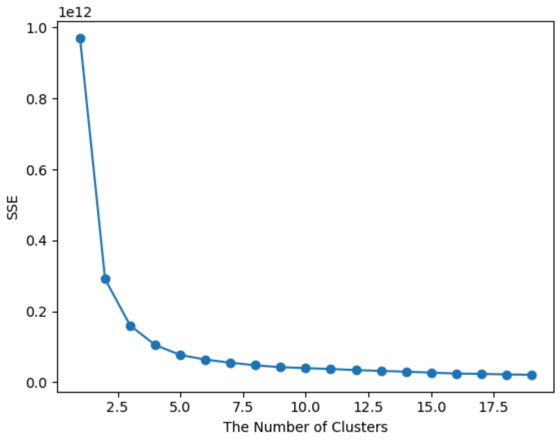
In [58]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()

df['Marital_Status'] = label_encoder.fit_transform(df['Marital_Status'])
```

```
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\112776462.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Marital_Status'] = label_encoder.fit_transform(df['Marital_Status'])
```

In [60]:
```
print(set(df['Marital_Status']))
```

```
{0, 1, 2, 3, 4, 5, 6, 7}
```

In [66]:
```
from sklearn.cluster import KMeans
X = df

def elbow(X):
    sse = []
    for i in range(1,20):
        km = KMeans(n_clusters=i, random_state=1)
        km.fit(X)
        sse.append(km.inertia_)

    plt.plot(range(1,20), sse, marker='o')
    plt.xlabel('The Number of Clusters')
    plt.ylabel('SSE')
    plt.show()
    print(sse)

elbow(X)
```



```
[970543986610.9198, 290651856687.75946, 159707280555.48395, 104911347502.17007, 76793066532.67589, 63735128665.97969, 54990097856.68637, 47604617396.77124, 42219603144.42308, 39721810542.89242, 37145173367.265724, 34161051283.82306, 31745528146.482, 29689328099.4404, 26
894870557.744766, 24427068149.95391, 23366512070.575924, 21893748930.570484, 20953414977.9661]
```

In [68]:
```
km = KMeans(n_clusters=5, random_state=1)
km.fit(X)

new_labels = km.labels_
df['clusters'] = new_labels

df.groupby(['clusters']).mean()
```

```
C:\Users\minje\AppData\Local\Temp\ipykernel_41864\3352184059.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['clusters'] = new_labels
```

Out[68]:

	ID	Year_Birth	Marital_Status	Income	Kidhome	Teenhome	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
clusters																
0	5728.060241	1966.343373	3.658635	66570.568273	0.122490	0.622490	514.103557	22.812285	213.705038	35.761318	23.387289	2.067489	5.720884	4.154171	8.379518	4.182731
1	5758.841499	1974.858790	3.700288	20946.409222	0.763689	0.172911	12.605187	5.971182	16.889705	7.931865	5.984372	1.940254	1.838945	0.413704	2.824207	6.847805
2	5459.341651	1971.314779	3.715931	36637.109405	0.809981	0.485605	62.403071	5.476670	33.882917	8.459692	5.809465	2.199499	2.600768	0.700576	3.491363	6.623800
3	5429.724846	1965.521561	3.788501	51506.109045	0.418891	0.850103	263.425990	10.525032	78.975176	17.204457	10.632057	2.701470	4.587269	2.086697	5.804928	5.745380
4	5631.696970	1967.292011	3.774105	82171.311295	0.074380	0.228650	600.614245	28.525793	248.674588	44.616270	28.581241	1.146560	5.272884	5.700331	8.418733	2.694215

In [70]:

```
df[df['ID']==10870]
```

Out[70]:

	ID	Year_Birth	Marital_Status	Income	Kidhome	Teenhome	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	clusters
2235	10870	1967	3	61223.0	0	1	709.0	43.0	182.0	42.0	14.577157	2.0	9.0	3.0	4	5.0	0

In [ ]:

In [ ]: