In [1]:
```
# Library untuk dataframe dan visualisasi
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt # Import library untuk Clustering
import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

In [2]:
```
# Load dataset OnlineRetail
df = pd.read_csv('D:/OnlineRetail.csv',header=0, encoding =
'unicode_escape')
df.head()
```

Out[2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

In [5]:
```
df_null = round(100*(df.isnull().sum())/len(df),2)
df_null
```

Out[5]:
```
InvoiceNo       0.00
StockCode       0.00
Description     0.27
Quantity        0.00
InvoiceDate     0.00
UnitPrice       0.00
CustomerID     24.93
Country         0.00
dtype: float64
```

In [6]:
```
df.dropna()
df.shape
```

Out[6]:
```
(541909, 8)
```

In [7]:
```
df['CustomerID']=df.CustomerID.astype(str)
```

In [50]:
```
# Membuat atribut baru : Monetary
df['Monetary'] = df['Quantity']*df['UnitPrice']
rfm_m = df.groupby('CustomerID')['Monetary'].sum()
rfm_m = rfm_m.reset_index()
rfm_m.head()
```

Out[50]:

	CustomerID	Monetary
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

In [51]:
```
# Membuat atribut baru : Frequency
rfm_f = df.groupby('CustomerID')['InvoiceNo'].count()
rfm_f = rfm_f.reset_index()
rfm_f.columns = ['CustomerID', 'Frequency']
rfm_f.head()
```

Out[51]:

	CustomerID	Frequency
0	12346.0	2
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

In [52]:
```
# Menggabungkan (merging) dua dataframe
rfm = pd.merge(rfm_m, rfm_f, on='CustomerID', how='inner')
rfm.head()
```

Out[52]:

	CustomerID	Monetary	Frequency
0	12346.0	0.00	2
1	12347.0	4310.00	182
2	12348.0	1797.24	31
3	12349.0	1757.55	73
4	12350.0	334.40	17

In [54]:
```
# Membuat atribut baru : Recency
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'],format='%d-%m-%Y %H:%M')
```

In [55]:
```
max_date = max(df['InvoiceDate'])
max_date
```

Out[55]:
```
Timestamp('2011-12-09 12:50:00')
```

In [14]:
```
# Menghitung selisih antara max_date dengan InvoiceDate
df['Diff'] = max_date - df['InvoiceDate']
df.head()
```

Out[14]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Monetary	Diff
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30	373 days 04:24:00
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00	373 days 04:24:00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00

In [56]:
```
# Menghitung the last transaction date untuk atribut Recency
rfm_p = df.groupby('CustomerID')['Diff'].min()
rfm_p = rfm_p.reset_index()
rfm_p.head()
```

Out[56]:

	CustomerID	Diff
0	12346.0	325 days 02:33:00
1	12347.0	1 days 20:58:00
2	12348.0	74 days 23:37:00
3	12349.0	18 days 02:59:00
4	12350.0	309 days 20:49:00

In [57]:
```
# Extract jumlah hari
rfm_p['Diff'] = rfm_p['Diff'].dt.days
rfm_p.head()
```

Out[57]:

	CustomerID	Diff
0	12346.0	325
1	12347.0	1
2	12348.0	74
3	12349.0	18
4	12350.0	309

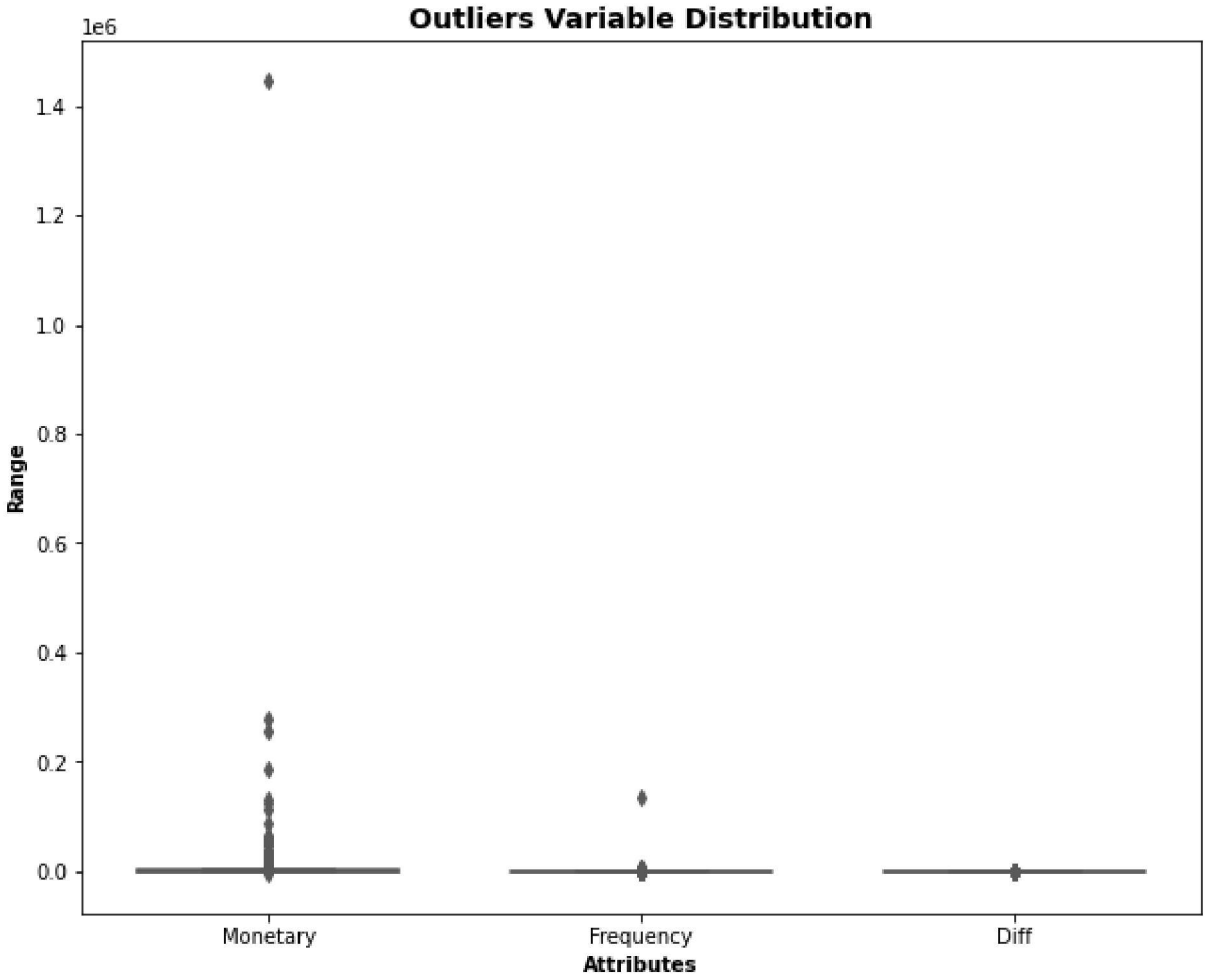
In [65]:
```
# Menggabungkan dataframe
rfm = pd.merge(rfm, rfm_p, on='CustomerID', how='inner')
#rfm.columns = ['CustomerID', 'Monetary', 'Frequency','Recency']
rfm.head()
```

Out[65]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y	Diff
0	12346.0	0.00	2	325	325	325
1	12347.0	4310.00	182	1	1	1
2	12348.0	1797.24	31	74	74	74
3	12349.0	1757.55	73	18	18	18
4	12350.0	334.40	17	309	309	309

In [67]:
```
# Outlier Analysis of Monetary, Frequency and Recency
attributes = ['Monetary','Frequency','Diff']
plt.rcParams['figure.figsize'] = [10,8]
sns.boxplot(data = rfm[attributes], orient='v', palette='Set2', whis=1.5,saturation=1, width=0.7)
plt.title("Outliers Variable Distribution", fontsize=14,fontweight = 'bold')
plt.ylabel("Range", fontweight='bold')
plt.xlabel("Attributes", fontweight='bold')
```

Out[67]:
```
Text(0.5, 0, 'Attributes')
```



In [34]:
```
rfm.head()
```

Out[34]:

	CustomerID	Monetary	Frequency	Recency	Diff_x	Diff_y
0	12346.0	0.00	2	325	325	325
1	12347.0	4310.00	182	1	1	1
2	12348.0	1797.24	31	74	74	74
3	12349.0	1757.55	73	18	18	18
4	12350.0	334.40	17	309	309	309

In [32]:
```
rfm.drop(columns='Recency')
```

Out[32]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y
0	12346.0	0.00	2	325	325
1	12347.0	4310.00	182	1	1
2	12348.0	1797.24	31	74	74
3	12349.0	1757.55	73	18	18
4	12350.0	334.40	17	309	309
...	...	...	...	...	...
4368	18281.0	80.82	7	180	180
4369	18282.0	176.60	13	7	7
4370	18283.0	2094.88	756	3	3
4371	18287.0	1837.28	70	42	42
4372	nan	1447682.12	135080	0	0

4373 rows × 5 columns

In [63]:
```
rfm.head()
```

Out[63]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y
0	12346.0	0.00	2	325	325
1	12347.0	4310.00	182	1	1
2	12348.0	1797.24	31	74	74
3	12349.0	1757.55	73	18	18
4	12350.0	334.40	17	309	309

In [36]:
```
del rfm['Diff_x']
```

In [37]:
```
del rfm['Diff_y']
```

In [38]:
```
del rfm['Recency']
```

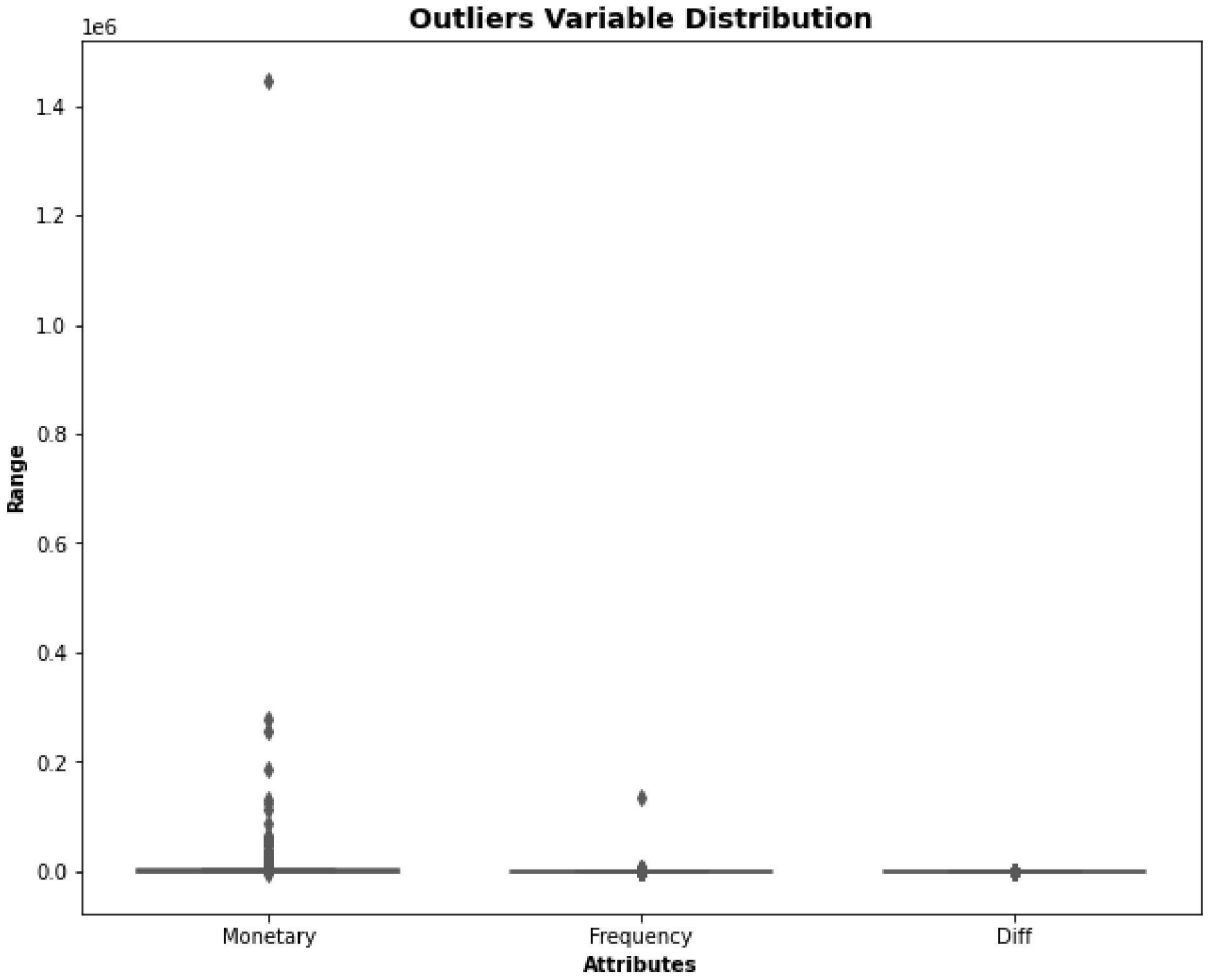
In [79]:
```
rfm.head()
```

Out[79]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y	Diff
0	12346.0	0.00	2	325	325	325
1	12347.0	4310.00	182	1	1	1
2	12348.0	1797.24	31	74	74	74
3	12349.0	1757.55	73	18	18	18
4	12350.0	334.40	17	309	309	309

In [68]:
```
attributes = ['Monetary','Frequency','Diff']
plt.rcParams['figure.figsize'] = [10,8]
sns.boxplot(data = rfm[attributes], orient="v", palette="Set2",
whis=1.5,saturation=1, width=0.7)
plt.title("Outliers Variable Distribution", fontsize=14,
fontweight = 'bold')
plt.ylabel("Range", fontweight='bold')
plt.xlabel("Attributes", fontweight='bold')
```

Out[68]:
```
Text(0.5, 0, 'Attributes')
```



In [46]:
```
rfm = rfm.reindex(columns=['Fequency'])
```

In [75]:
```
rfm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4277 entries, 0 to 4371
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   CustomerID  4277 non-null   object 
 1   Monetary    4277 non-null   float64
 2   Frequency   4277 non-null   int64  
 3   Diff_x      4277 non-null   int64  
 4   Diff_y      4277 non-null   int64  
 5   Diff        4277 non-null   int64  
dtypes: float64(1), int64(4), object(1)
memory usage: 233.9+ KB
```

In [80]:
```
rfm.head()
```

Out[80]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y	Diff
0	12346.0	0.00	2	325	325	325
1	12347.0	4310.00	182	1	1	1
2	12348.0	1797.24	31	74	74	74
3	12349.0	1757.55	73	18	18	18
4	12350.0	334.40	17	309	309	309

In [83]:
```
# Removing (statistical) outliers for Monetary
Q1 = rfm.Monetary.quantile(0.05)
Q3 = rfm.Monetary.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Monetary >= Q1 - 1.5*IQR) & (rfm.Monetary <= Q3 +
1.5*IQR)]# Removing (statistical) outliers for Recency
Q1 = rfm.Recency.quantile(0.05)
Q3 = rfm.Recency.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Recency >= Q1 - 1.5*IQR) & (rfm.Recency <= Q3 +
1.5*IQR)]# Removing (statistical) outliers for Frequency
Q1 = rfm.Frequency.quantile(0.05)
Q3 = rfm.Frequency.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Frequency >= Q1 - 1.5*IQR) & (rfm.Frequency <= Q3
+ 1.5*IQR)]
```

In [74]:
```
rfm_df = rfm[['Monetary', 'Frequency', 'Diff']]# Instantiate
scaler = StandardScaler()# fit_transform
rfm_df_scaled = scaler.fit_transform(rfm_df)
rfm_df_scaled.shape
```

Out[74]:
```
(4277, 3)
```

In [82]:
```
rfm.columns = ['CustomerID', 'Monetary', 'Frequency','Diff_x','Diff_y','Recency']
```

In [81]:
```
rfm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4277 entries, 0 to 4371
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   CustomerID  4277 non-null   object
 1   Monetary    4277 non-null   float64
 2   Frequency   4277 non-null   int64
 3   Diff_x      4277 non-null   int64
 4   Diff_y      4277 non-null   int64
 5   Diff        4277 non-null   int64
dtypes: float64(1), int64(4), object(1)
memory usage: 233.9+ KB
```

In [84]:
```
# Rescaling Atribute
rfm_df = rfm[['Monetary', 'Frequency', 'Recency']]# Instantiate
scaler = StandardScaler()# fit_transform
rfm_df_scaled = scaler.fit_transform(rfm_df)
rfm_df_scaled.shape
```

Out[84]:
```
(4266, 3)
```

In [85]:
```
rfm_df_scaled = pd.DataFrame(rfm_df_scaled)
rfm_df_scaled.columns = ['Amount', 'Frequency', 'Recency']
rfm_df_scaled.head()
```

Out[85]:

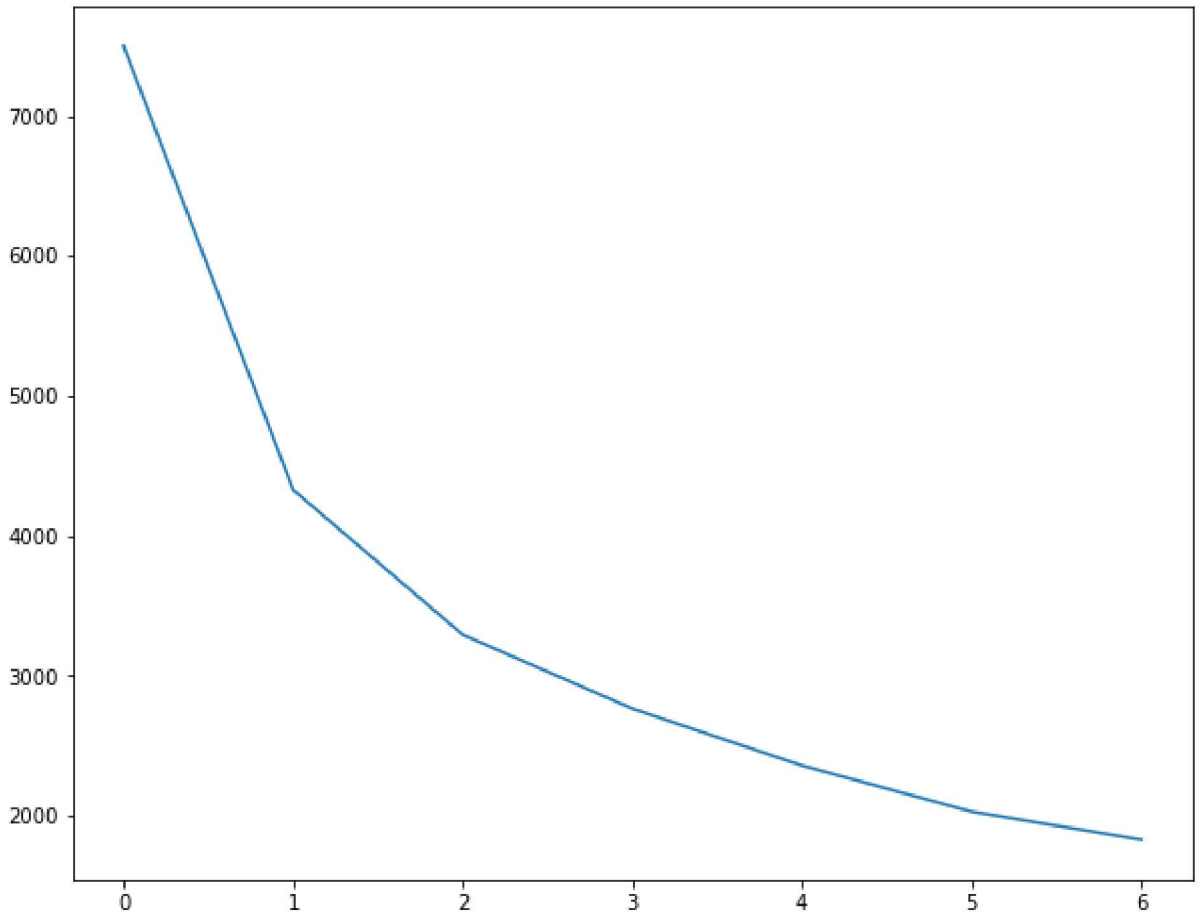
	Amount	Frequency	Recency
0	-0.769297	-0.773358	2.294182
1	1.966597	1.126142	-0.910568
2	0.371552	-0.467328	-0.188510
3	0.346358	-0.024111	-0.742418
4	-0.557027	-0.615067	2.135923

In [87]:
```
kmeans = KMeans(n_clusters=4, max_iter=50)
kmeans.fit(rfm_df_scaled)
KMeans(algorithm='auto', copy_x=True, init='k-means++',
max_iter=50, n_clusters=4, n_init=10, random_state=None,
tol=0.0001, verbose=0)
kmeans.labels_
```

Out[87]:
```
array([0, 3, 1, ..., 0, 1, 1])
```

In [89]:
```
# Elbow-curve/SSD
ssd = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(rfm_df_scaled)
    ssd.append(kmeans.inertia_)
# plot the SSDs for each n_clusters
plt.plot(ssd)
```

Out[89]:
```
[<matplotlib.lines.Line2D at 0x1a8779ed6a0>]
```



In [90]:
```
# Silhouette Analysis
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    # Initialise kmeans
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(rfm_df_scaled)
    cluster_labels = kmeans.labels_
# Silhouette Score
silhouette_avg = silhouette_score(rfm_df_scaled,cluster_labels)
print("For n_clusters={0}, the silhouette score is{1}".format(num_clusters, silhouette_avg))
```

```
For n_clusters=8, the silhouette score is0.37276927267649357
```

In [91]:
```
# Final model with k=2
kmeans = KMeans(n_clusters=2, max_iter=50)
kmeans.fit(rfm_df_scaled)
KMeans(algorithm='auto', copy_x=True, init='k-means++',
max_iter=50, n_clusters=2, n_init=10, random_state=None,
tol=0.0001, verbose=0)
kmeans.labels_
```

Out[91]:
```
array([0, 1, 0, ..., 0, 0, 0])
```

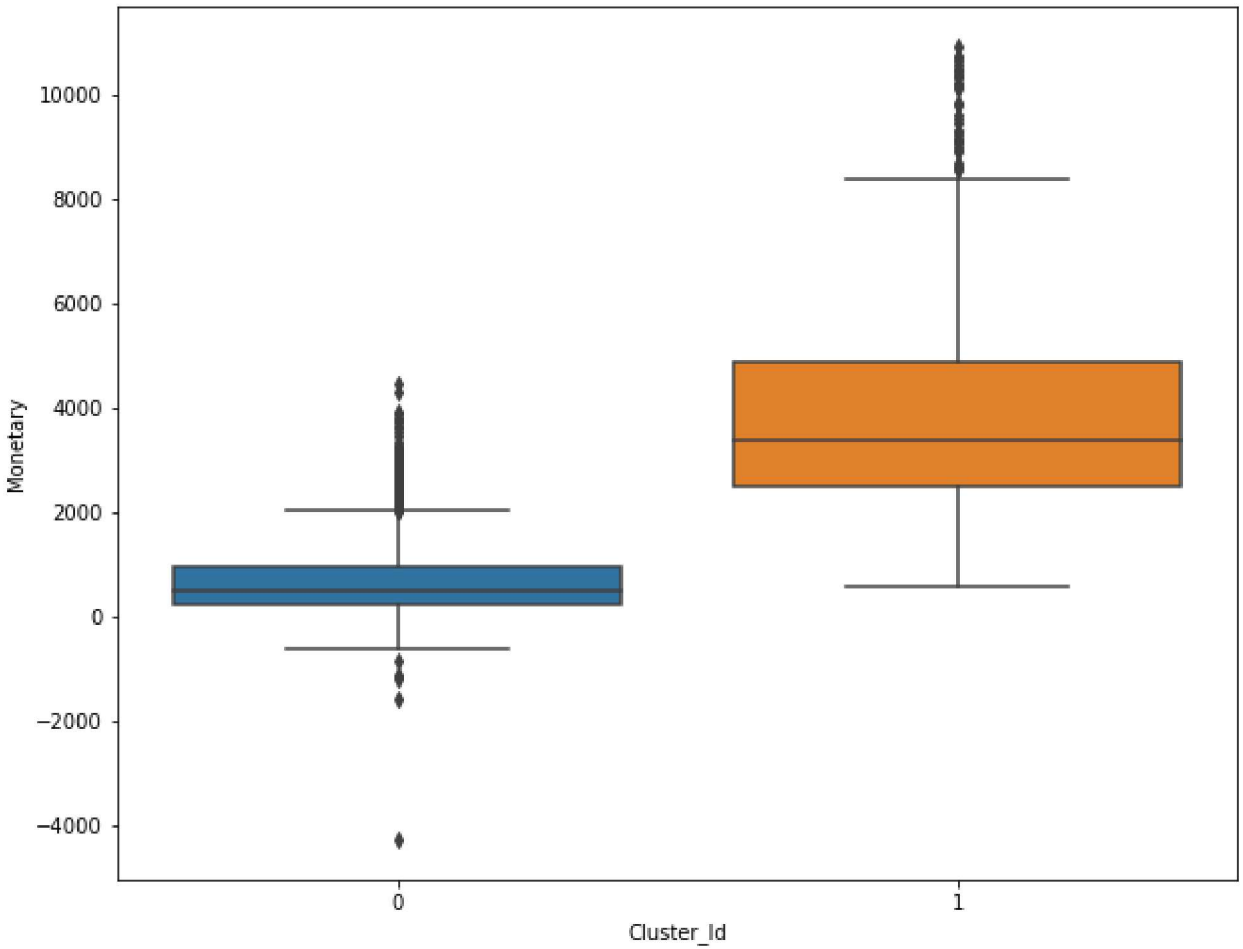
In [92]:
```
# Assign the label
rfm['Cluster_Id'] = kmeans.labels_
rfm.head()
```

Out[92]:

	CustomerID	Monetary	Frequency	Diff_x	Diff_y	Recency	Cluster_Id
0	12346.0	0.00	2	325	325	325	0
1	12347.0	4310.00	182	1	1	1	1
2	12348.0	1797.24	31	74	74	74	0
3	12349.0	1757.55	73	18	18	18	0
4	12350.0	334.40	17	309	309	309	0

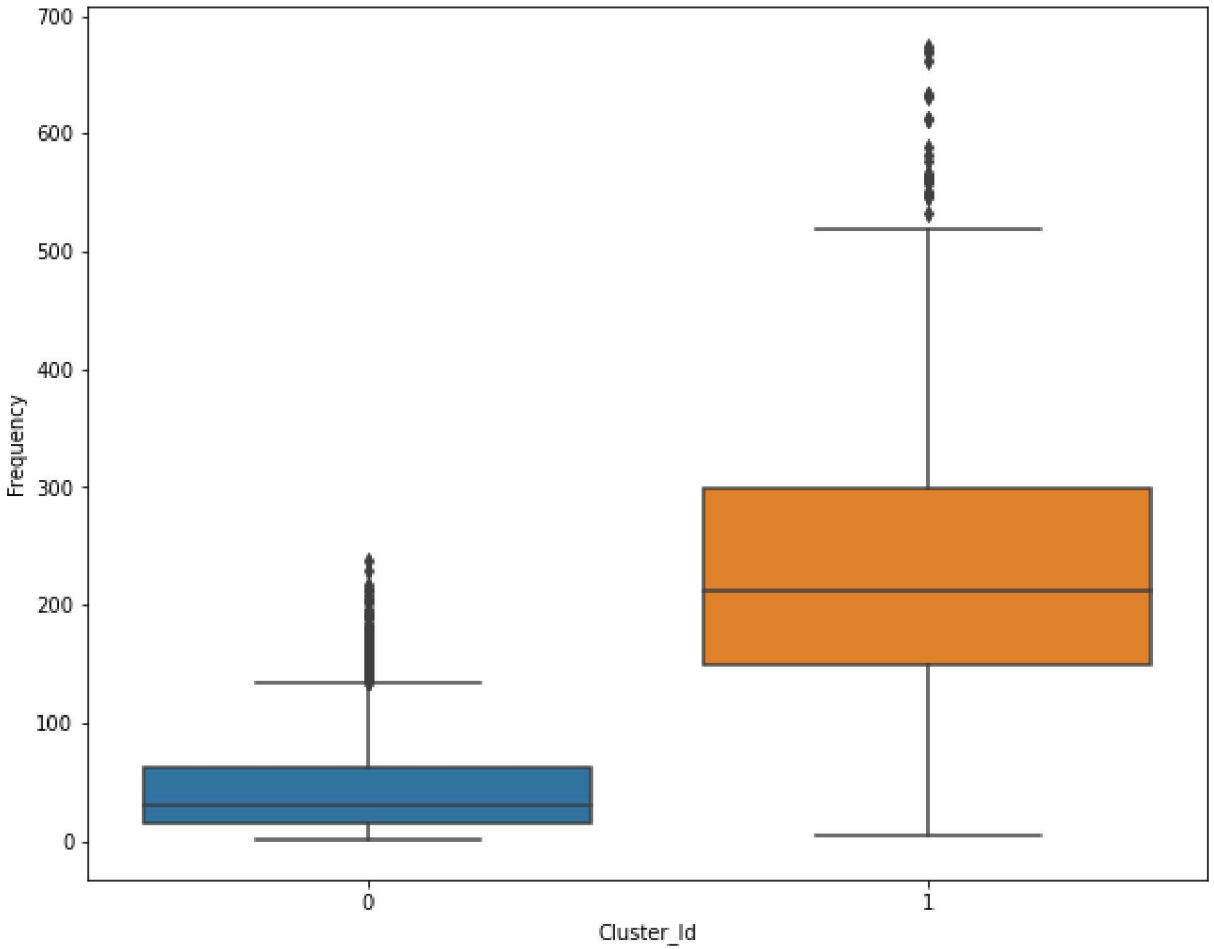
In [93]:
```
# Boxplot untuk memvisualisasikan Cluster Id dan Monetary
sns.boxplot(x='Cluster_Id', y='Monetary', data=rfm)
```

Out[93]:
```
<AxesSubplot:xlabel='Cluster_Id', ylabel='Monetary'>
```



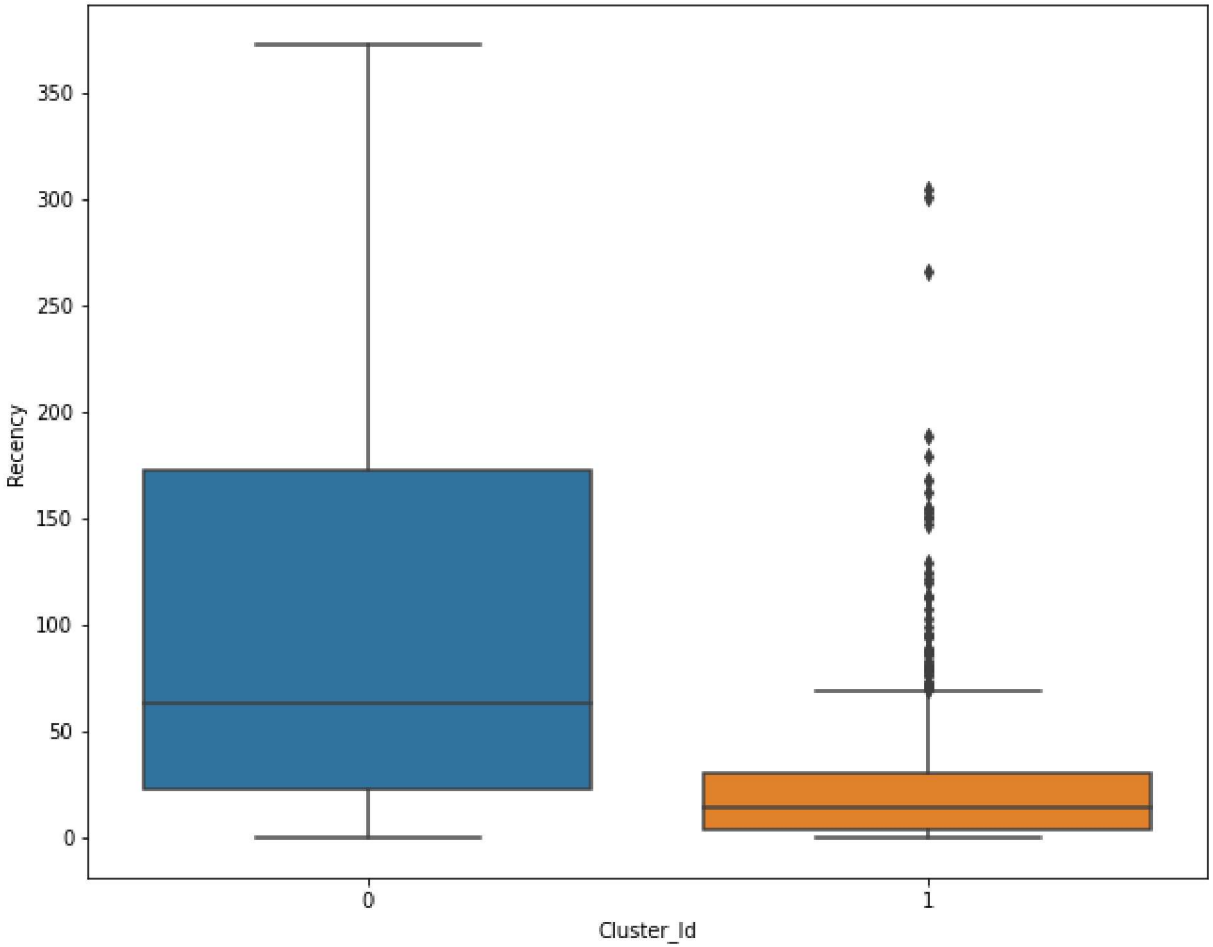
In [94]:
```
# Boxplot untuk memvisualisasikan Cluster Id vs Frequency
sns.boxplot(x='Cluster_Id', y='Frequency', data=rfm)
```

Out[94]:
```
<AxesSubplot:xlabel='Cluster_Id', ylabel='Frequency'>
```



In [95]:
```
# Boxplot untuk memvisualisasikan Cluster Id vs Recency
sns.boxplot(x='Cluster_Id', y='Recency', data=rfm)
```

Out[95]:
```
<AxesSubplot:xlabel='Cluster_Id', ylabel='Recency'>
```



In [ ]: