Predictive Analytics 3 - Dimension Reduction, Clustering, and Association Rules: Python 090823

Final

James Ward

Segmenting Consumers of Bath Soap BathSoapHousehold.csv is the dataset for this case study. Business Situation CRISA is an Asian market research agency that specializes in tracking consumer purchase behavior in consumer goods (both durable and nondurable). In one major research project, CRISA tracks numerous consumer product categories (e.g., “detergents”), and, within each category, perhaps dozens of brands. To track purchase behavior, CRISA constituted household panels in over 100 cities and towns in India, covering most of the Indian urban market. The households were carefully selected using stratified sampling to ensure a representative sample; a subset of 600 records is analyzed here. The strata were defined on the basis of socioeconomic status and the market (a collection of cities). CRISA has both transaction data (each row is a transaction) and household data (each row is a household), and for the household data it maintains the following information: Demographics of the households (updated annually) Possession of durable goods (car, washing machine, etc., updated annually; an “affluence index” is computed from this information) Purchase data of product categories and brands (updated monthly) CRISA has two categories of clients: (1) advertising agencies that subscribe to the database services, obtain updated data every month, and use the data to advise their clients on advertising and promotion strategies; (2) consumer goods manufacturers, which monitor their market share using the CRISA database.

Key Problems

CRISA has traditionally segmented markets on the basis of purchaser demographics. They would now like to segment the market based on two key sets of variables more directly related to the purchase process and to brand loyalty:

1. Purchase behavior (volume, frequency, susceptibility to discounts, and brand loyalty)
2. Basis of purchase (price, selling proposition)

Doing so would allow CRISA to gain information about what demographic attributes are associated with different purchase behaviors and degrees of brand loyalty, and thus deploy promotion budgets more effectively. More effective market segmentation would enable CRISA’s clients (in this case, a firm called IMRB) to design more cost-effective promotions targeted at appropriate segments. Thus, multiple promotions could be launched, each targeted at different market segments at different times of the year. This would result in a more cost effective allocation of the promotion budget to different market segments. It would also enable IMRB to design more effective customer reward systems and thereby increase brand loyalty.

Data The data in the below table profile each household, each row containing the data for one household.

Measuring Brand Loyalty Several variables in this case measure aspects of brand loyalty. The number of different brands purchased by the customer is one measure of loyalty. However, a consumer who purchases one or two brands in quick succession, then settles on a third for a long streak, is different from a consumer who constantly switches back and forth among three brands. Therefore, how often customers switch from one brand to another is another measure of loyalty. Yet a third perspective on the same issue is the proportion of purchases that go to different brands—a consumer who spends 90% of his or her purchase money on one brand is more loyal than a consumer who spends more equally among several brands. All three of these components can be measured with the data in the purchase summary worksheet.

In [1]:
```
from pathlib import Path

import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import matplotlib.pylab as plt
from sklearn import preprocessing
from sklearn.cluster import KMeans
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from dmba import classificationSummary
from sklearn import metrics
from sklearn.feature_selection import VarianceThreshold
from sklearn.preprocessing import StandardScaler
import numpy as np
import seaborn as sns
from sklearn.decomposition import PCA
from scipy.cluster import hierarchy
from scipy.spatial.distance import pdist
from scipy.cluster.hierarchy import fcluster

from surprise import Dataset
from surprise import Reader
from surprise import KNNBasic

import dmba

%matplotlib inline
```

In [2]:
```
# Load data for data frame
soap_df = pd.read_csv('C:/Users/Grant/Desktop/Desktop/School/Datasets/BathSoapHousehold.csv')
```

In [3]:
```
# Observe data
soap_df
```

Out[3]:

	Member id	SEC	FEH	MT	SEX	AGE	EDU	HS	CHILD	CS	...	PropCat 6	PropCat 7	PropCat 8	PropCat 9	PropCat 10	PropCat 11	PropCat 12	Pr
0	1010010	4	3	10	1	4	4	2	4	1	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.028037	0.0
1	1010020	3	2	10	2	2	4	4	2	1	...	0.347048	0.026834	0.016100	0.014311	0.000000	0.059034	0.000000	0.0
2	1014020	2	3	10	2	4	5	6	4	1	...	0.121212	0.033550	0.010823	0.008658	0.000000	0.000000	0.016234	0.0
3	1014030	4	0	0	0	4	0	0	5	0	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
4	1014190	4	1	10	2	3	4	4	3	1	...	0.000000	0.000000	0.048193	0.000000	0.000000	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
595	1167090	2	3	10	2	4	5	5	4	1	...	0.000000	0.000000	0.000000	0.000000	0.024931	0.897507	0.000000	0.0
596	1167230	3	3	10	2	3	5	4	4	1	...	0.430693	0.074257	0.148515	0.000000	0.039604	0.000000	0.000000	0.0
597	1167340	3	3	17	2	4	4	9	4	1	...	0.127148	0.000000	0.030928	0.000000	0.038660	0.000000	0.000000	0.0
598	1167350	2	1	4	2	4	5	2	4	2	...	0.145455	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
599	1167670	3	3	10	2	4	4	6	4	1	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

600 rows × 46 columns

In [4]:
```
# Examine column data types for analysis. No categorical data
column_data_types = soap_df.dtypes
print(column_data_types)
```

```
Member id                 int64
SEC                       int64
FEH                       int64
MT                        int64
SEX                       int64
AGE                       int64
EDU                       int64
HS                        int64
CHILD                     int64
CS                        int64
Affluence Index           int64
No. of Brands             int64
Brand Runs                int64
Total Volume              int64
No. of  Trans             int64
Value                   float64
Trans / Brand Runs      float64
Vol/Tran                float64
Avg. Price              float64
Pur Vol No Promo - %    float64
Pur Vol Promo 6 %       float64
Pur Vol Other Promo %   float64
Br. Cd. 57, 144         float64
Br. Cd. 55              float64
Br. Cd. 272             float64
Br. Cd. 286             float64
Br. Cd. 24              float64
Br. Cd. 481             float64
Br. Cd. 352             float64
Br. Cd. 5               float64
Others 999              float64
Pr Cat 1                float64
Pr Cat 2                float64
Pr Cat 3                float64
Pr Cat 4                float64
PropCat 5               float64
PropCat 6               float64
PropCat 7               float64
PropCat 8               float64
PropCat 9               float64
PropCat 10              float64
PropCat 11              float64
PropCat 12              float64
PropCat 13              float64
PropCat 14              float64
```

```
PropCat 15              float64
dtype: object
```

In [5]:
```python
# Check for missing or NaN data
missing_data = soap_df.isna().sum()
print(missing_data)
```

Member id	0
SEC	0
FEH	0
MT	0
SEX	0
AGE	0
EDU	0
HS	0
CHILD	0
CS	0
Affluence Index	0
No. of Brands	0
Brand Runs	0
Total Volume	0
No. of Trans	0
Value	0
Trans / Brand Runs	0
Vol/Tran	0
Avg. Price	0
Pur Vol No Promo - %	0
Pur Vol Promo 6 %	0
Pur Vol Other Promo %	0
Br. Cd. 57, 144	0
Br. Cd. 55	0
Br. Cd. 272	0
Br. Cd. 286	0
Br. Cd. 24	0
Br. Cd. 481	0
Br. Cd. 352	0
Br. Cd. 5	0
Others 999	0
Pr Cat 1	0
Pr Cat 2	0
Pr Cat 3	0
Pr Cat 4	0
PropCat 5	0
PropCat 6	0
PropCat 7	0
PropCat 8	0
PropCat 9	0
PropCat 10	0
PropCat 11	0
PropCat 12	0
PropCat 13	0
PropCat 14	0

```
PropCat 15              0
dtype: int64
```

In [6]:
```python
# Run a Variance Threshold to remove redundant low variance variables

# Set a threshold for variance (e.g., 0.01)
threshold = 0.01

# Initialize the VarianceThreshold selector
selector = VarianceThreshold(threshold)

# Fit the selector to the data
selector.fit(soap_df)

# Get the indices of non-constant columns
non_constant_columns = soap_df.columns[selector.get_support()]

# Select only non-constant columns from the DataFrame
filtered_df = soap_df[non_constant_columns]
```

In [7]:
```python
# Re-examine filtered_df
filtered_df
```

Out[7]:

	Member id	SEC	FEH	MT	SEX	AGE	EDU	HS	CHILD	CS	...	Others 999	Pr Cat 1	Pr Cat 2	Pr Cat 3	Pr Cat 4	PropCat 5	PropCat 6	Pr
0	1010010	4	3	10	1	4	4	2	4	1	...	0.492212	0.233645	0.560748	0.130841	0.074766	0.501558	0.000000	0.0
1	1010020	3	2	10	2	2	4	4	2	1	...	0.699463	0.293381	0.547406	0.094812	0.064401	0.456172	0.347048	0.0
2	1014020	2	3	10	2	4	5	6	4	1	...	0.378788	0.120130	0.318182	0.561688	0.000000	0.244589	0.121212	0.0
3	1014030	4	0	0	0	4	0	0	5	0	...	0.000000	0.000000	0.400000	0.600000	0.000000	0.400000	0.000000	0.0
4	1014190	4	1	10	2	3	4	4	3	1	...	0.807229	0.000000	0.048193	0.144578	0.807229	0.807229	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
595	1167090	2	3	10	2	4	5	5	4	1	...	0.058172	0.058172	0.941828	0.000000	0.000000	0.077562	0.000000	0.0
596	1167230	3	3	10	2	3	5	4	4	1	...	0.678218	0.534653	0.366337	0.000000	0.099010	0.168317	0.430693	0.0
597	1167340	3	3	17	2	4	4	9	4	1	...	0.557560	0.170103	0.649485	0.180412	0.000000	0.622852	0.127148	0.0
598	1167350	2	1	4	2	4	5	2	4	2	...	0.863636	0.136364	0.509091	0.209091	0.145455	0.690909	0.145455	0.0
599	1167670	3	3	10	2	4	4	6	4	1	...	0.186992	0.097561	0.902439	0.000000	0.000000	1.000000	0.000000	0.0

600 rows × 34 columns

In [8]:
```
# Get the data types of each column in the filtered DataFrame
column_data_types = filtered_df.dtypes

# Filter columns with non-continuous data types (e.g., object or category)
categorical_columns = column_data_types[column_data_types == 'object'].index.tolist()

# Display the list of categorical columns
print("Categorical Columns:")
print(categorical_columns)
```

```
Categorical Columns:
[]
```

In [9]:
```
# Drop 'Member id' from the data frame filtered_df
filtered_df = filtered_df.drop(columns=['Member id'])
```

In [10]:
```
# Re-examine filtered_df after droping 'Member id'
filtered_df
```

Out[10]:

	SEC	FEH	MT	SEX	AGE	EDU	HS	CHILD	CS	Affluence Index	...	Others 999	Pr Cat 1	Pr Cat 2	Pr Cat 3	Pr Cat 4	PropCat 5	PropCat 6	P
0	4	3	10	1	4	4	2	4	1	2	...	0.492212	0.233645	0.560748	0.130841	0.074766	0.501558	0.000000	0.(
1	3	2	10	2	2	4	4	2	1	19	...	0.699463	0.293381	0.547406	0.094812	0.064401	0.456172	0.347048	0.(
2	2	3	10	2	4	5	6	4	1	23	...	0.378788	0.120130	0.318182	0.561688	0.000000	0.244589	0.121212	0.(
3	4	0	0	0	4	0	0	5	0	0	...	0.000000	0.000000	0.400000	0.600000	0.000000	0.400000	0.000000	0.(
4	4	1	10	2	3	4	4	3	1	10	...	0.807229	0.000000	0.048193	0.144578	0.807229	0.807229	0.000000	0.(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
595	2	3	10	2	4	5	5	4	1	15	...	0.058172	0.058172	0.941828	0.000000	0.000000	0.077562	0.000000	0.(
596	3	3	10	2	3	5	4	4	1	29	...	0.678218	0.534653	0.366337	0.000000	0.099010	0.168317	0.430693	0.(
597	3	3	17	2	4	4	9	4	1	13	...	0.557560	0.170103	0.649485	0.180412	0.000000	0.622852	0.127148	0.(
598	2	1	4	2	4	5	2	4	2	20	...	0.863636	0.136364	0.509091	0.209091	0.145455	0.690909	0.145455	0.(
599	3	3	10	2	4	4	6	4	1	15	...	0.186992	0.097561	0.902439	0.000000	0.000000	1.000000	0.000000	0.(

600 rows × 33 columns

In [11]:
```
# Normalize/Standardize the data
# Create a StandardScaler object
scaler = StandardScaler()

# Fit the scaler to the data and transform it
standardized_data = scaler.fit_transform(filtered_df)

# Create a DataFrame with the standardized data
filtered_df = pd.DataFrame(standardized_data, columns=filtered_df.columns)
```

In [12]:
```
# Re-examine filtered_df after the data has been scaled
filtered_df
```

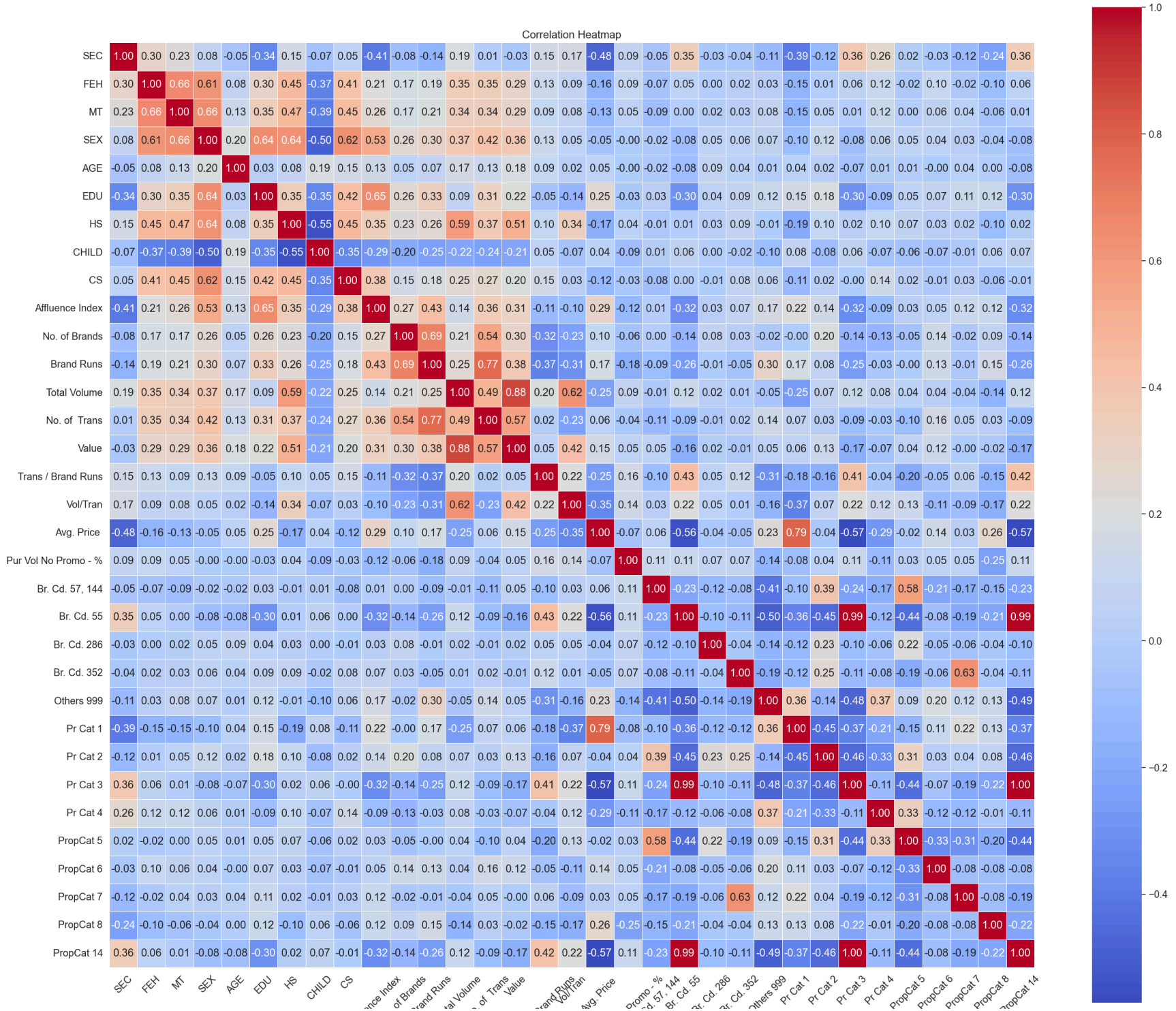
Out[12]:

	SEC	FEH	MT	SEX	AGE	EDU	HS	CHILD	CS	Affluence Index	...	Others 999	Pr Cat 1	Pr
0	1.341641	0.839199	0.424526	-1.139458	0.909686	-0.019803	-0.953656	0.630433	0.134793	-1.317478	...	-0.100244	-0.161736	0.2
1	0.447214	-0.042621	0.424526	0.403826	-1.403075	-0.019803	-0.083400	-1.014175	0.134793	0.173676	...	0.597393	0.051111	0.1
2	-0.447214	0.839199	0.424526	0.403826	0.909686	0.437198	0.786857	0.630433	0.134793	0.524535	...	-0.482045	-0.566204	-0.5
3	1.341641	-1.806261	-1.905900	-2.682741	0.909686	-1.847808	-1.823913	1.452736	-1.837792	-1.492908	...	-1.757097	-0.994241	-0.2
4	1.341641	-0.924441	0.424526	0.403826	-0.246695	-0.019803	-0.083400	-0.191871	0.134793	-0.615759	...	0.960147	-0.994241	-1.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
595	-0.447214	0.839199	0.424526	0.403826	0.909686	0.437198	0.351729	0.630433	0.134793	-0.177184	...	-1.561283	-0.786968	1.4
596	0.447214	0.839199	0.424526	0.403826	-0.246695	0.437198	-0.083400	0.630433	0.134793	1.050825	...	0.525878	0.910793	-0.4
597	0.447214	0.839199	2.055825	0.403826	0.909686	-0.019803	2.092242	0.630433	0.134793	-0.352614	...	0.119727	-0.388143	0.5
598	-0.447214	-0.924441	-0.973730	0.403826	0.909686	0.437198	-0.953656	0.630433	2.107379	0.261391	...	1.150022	-0.508361	0.0
599	0.447214	0.839199	0.424526	0.403826	0.909686	-0.019803	0.786857	0.630433	0.134793	-0.177184	...	-1.127657	-0.646620	1.3

600 rows × 33 columns

In [13]:

```
# Calculate the correlation matrix
correlation_matrix = filtered_df.corr()

# Create a larger heatmap with clear and larger annotations
plt.figure(figsize=(30, 26))
sns.set(font_scale=1.3)
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', linewidths=0.5, square=True)
plt.title('Correlation Heatmap')
plt.xticks(rotation=45)
plt.show()
```


Correlation Heatmap
SEC
FEH
MT
SEX
AGE
EDU
HS
CHILD
CS
Affluence Index
No. of Brands
Brand Runs
Total Volume
No. of Trans
Value
Trans / Brand Runs
Vol/Tran
Avg. Price
Pur Vol No Promo - %
Br. Cd. 57, 144
Br. Cd. 55
Br. Cd. 286
Br. Cd. 352
Others 999
Pr Cat 1
Pr Cat 2
Pr Cat 3
Pr Cat 4
PropCat 5
PropCat 6
PropCat 7
PropCat 8
PropCat 14
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Higer correlations tend to be clustered around demographic variables. Lower correlations around price and item cats.

In [14]:
```
# Calculate the correlation matrix
correlation_matrix = filtered_df.corr()

# Print the correlation matrix with labels
print("Correlation Matrix:")
print(correlation_matrix)
```

Correlation Matrix:

	SEC	FEH	MT	SEX	AGE
SEC	1.000000	0.299058	0.226678	0.083972	-0.048267
FEH	0.299058	1.000000	0.656519	0.609202	0.084659
MT	0.226678	0.656519	1.000000	0.655163	0.132575
SEX	0.083972	0.609202	0.655163	1.000000	0.197776
AGE	-0.048267	0.084659	0.132575	0.197776	1.000000
EDU	-0.340629	0.297371	0.347612	0.640400	0.032988
HS	0.152757	0.449217	0.468905	0.635814	0.081738
CHILD	-0.071097	-0.371948	-0.392515	-0.497819	0.185636
CS	0.045579	0.414519	0.445378	0.620378	0.147306
Affluence Index	-0.405610	0.211345	0.255750	0.528873	0.134979
No. of Brands	-0.082167	0.172760	0.174438	0.255776	0.053124
Brand Runs	-0.143578	0.191172	0.207314	0.304812	0.068423
Total Volume	0.188524	0.354604	0.341867	0.372670	0.165085
No. of Trans	0.008903	0.346693	0.340823	0.415536	0.126553
Value	-0.028740	0.292899	0.289864	0.362982	0.180891
Trans / Brand Runs	0.152281	0.131114	0.093842	0.129355	0.093231
Vol/Tran	0.170971	0.085769	0.080123	0.049334	0.022137
Avg. Price	-0.478328	-0.155945	-0.127836	-0.053394	0.047557
Pur Vol No Promo - %	0.086770	0.091833	0.049489	-0.001124	-0.003286
Br. Cd. 57, 144	-0.045165	-0.072900	-0.091909	-0.020061	-0.018693
Br. Cd. 55	0.347713	0.048044	0.001761	-0.084083	-0.082339
Br. Cd. 286	-0.028673	0.001448	0.015672	0.052214	0.088763
Br. Cd. 352	-0.042881	0.020008	0.032339	0.059987	0.042116
Others 999	-0.111187	0.027322	0.076231	0.065921	0.006408
Pr Cat 1	-0.394096	-0.149874	-0.146389	-0.100914	0.038473
Pr Cat 2	-0.120461	0.007977	0.052045	0.117682	0.021184
Pr Cat 3	0.364035	0.062999	0.009786	-0.077301	-0.073496
Pr Cat 4	0.264269	0.118559	0.116229	0.064671	0.011958
PropCat 5	0.015124	-0.024532	0.001579	0.053198	0.009438
PropCat 6	-0.031306	0.102199	0.061889	0.042404	-0.004071
PropCat 7	-0.117010	-0.024065	0.041144	0.034149	0.038957
PropCat 8	-0.242010	-0.102663	-0.060184	-0.042011	0.002299
PropCat 14	0.360679	0.058541	0.010079	-0.080152	-0.075245

	EDU	HS	CHILD	CS	Affluence Index
SEC	-0.340629	0.152757	-0.071097	0.045579	-0.405610
FEH	0.297371	0.449217	-0.371948	0.414519	0.211345
MT	0.347612	0.468905	-0.392515	0.445378	0.255750
SEX	0.640400	0.635814	-0.497819	0.620378	0.528873
AGE	0.032988	0.081738	0.185636	0.147306	0.134979
EDU	1.000000	0.348663	-0.354541	0.424860	0.649089
HS	0.348663	1.000000	-0.548539	0.447558	0.347430
CHILD	-0.354541	-0.548539	1.000000	-0.347212	-0.290653

```
CS                     0.424860  0.447558 -0.347212  1.000000         0.377720
Affluence Index        0.649089  0.347430 -0.290653  0.377720         1.000000
No. of Brands          0.255972  0.231930 -0.198086  0.150184         0.273357
Brand Runs             0.332408  0.263237 -0.246220  0.178443         0.433614
Total Volume           0.089795  0.588335 -0.217673  0.252444         0.144811
No. of  Trans          0.313106  0.366605 -0.235371  0.267405         0.356196
Value                  0.223011  0.508337 -0.211280  0.198460         0.313907
Trans / Brand Runs    -0.050773  0.096434  0.047906  0.153573        -0.110310
Vol/Tran              -0.136161  0.335586 -0.070434  0.033855        -0.102701
Avg. Price             0.248446 -0.174598  0.040834 -0.119323         0.288343
Pur Vol No Promo - % -0.025732  0.039405 -0.087193 -0.027659        -0.117793
Br. Cd. 57, 144        0.034292 -0.007750  0.013446 -0.079257         0.010817
Br. Cd. 55            -0.300700  0.011107  0.064976  0.000445        -0.316051
Br. Cd. 286            0.042753  0.030575  0.004756 -0.005508         0.031640
Br. Cd. 352            0.090543  0.087254 -0.015202  0.076157         0.072250
Others 999             0.116839 -0.005891 -0.100402  0.062329         0.170043
Pr Cat 1               0.150928 -0.193844  0.080584 -0.114837         0.218354
Pr Cat 2               0.179610  0.096912 -0.082623  0.019300         0.136339
Pr Cat 3              -0.301265  0.020954  0.063504 -0.002298        -0.320399
Pr Cat 4              -0.091859  0.097218 -0.072578  0.140117        -0.093571
PropCat 5              0.046768  0.066487 -0.059691  0.018409         0.031577
PropCat 6              0.066818  0.026647 -0.065404 -0.008242         0.050516
PropCat 7              0.111341  0.020185 -0.008247  0.025228         0.119004
PropCat 8              0.115574 -0.103354  0.060404 -0.063462         0.119975
PropCat 14            -0.303237  0.016090  0.065931 -0.006235        -0.321604

                       ...  Others 999  Pr Cat 1  Pr Cat 2  Pr Cat 3  Pr Cat 4  \
SEC                    ...   -0.111187 -0.394096 -0.120461  0.364035  0.264269
FEH                    ...    0.027322 -0.149874  0.007977  0.062999  0.118559
MT                     ...    0.076231 -0.146389  0.052045  0.009786  0.116229
SEX                    ...    0.065921 -0.100914  0.117682 -0.077301  0.064671
AGE                    ...    0.006408  0.038473  0.021184 -0.073496  0.011958
EDU                    ...    0.116839  0.150928  0.179610 -0.301265 -0.091859
HS                     ...   -0.005891 -0.193844  0.096912  0.020954  0.097218
CHILD                  ...   -0.100402  0.080584 -0.082623  0.063504 -0.072578
CS                     ...    0.062329 -0.114837  0.019300 -0.002298  0.140117
Affluence Index        ...    0.170043  0.218354  0.136339 -0.320399 -0.093571
No. of Brands          ...   -0.020629 -0.001552  0.199095 -0.136240 -0.130851
Brand Runs             ...    0.300369  0.172568  0.079927 -0.254291 -0.027219
Total Volume           ...   -0.048676 -0.253026  0.074945  0.120735  0.080123
No. of  Trans          ...    0.138972  0.069699  0.034534 -0.092840 -0.028449
Value                  ...    0.054444  0.059832  0.131485 -0.167519 -0.067163
Trans / Brand Runs     ...   -0.313319 -0.184578 -0.162885  0.410151 -0.038273
Vol/Tran               ...   -0.157371 -0.369190  0.067192  0.219698  0.124568
Avg. Price             ...    0.232646  0.786073 -0.036586 -0.571698 -0.292996
```

Pur Vol No Promo - %	...	-0.142841	-0.075085	0.040284	0.110385	-0.109806
Br. Cd. 57, 144	...	-0.412884	-0.096714	0.393025	-0.237260	-0.165382
Br. Cd. 55	...	-0.500151	-0.359844	-0.454932	0.988652	-0.115641
Br. Cd. 286	...	-0.143858	-0.122409	0.233023	-0.096683	-0.064216
Br. Cd. 352	...	-0.191198	-0.118342	0.248498	-0.109934	-0.076802
Others 999	...	1.000000	0.357685	-0.137877	-0.482266	0.374338
Pr Cat 1	...	0.357685	1.000000	-0.448498	-0.373099	-0.214611
Pr Cat 2	...	-0.137877	-0.448498	1.000000	-0.457383	-0.328738
Pr Cat 3	...	-0.482266	-0.373099	-0.457383	1.000000	-0.108100
Pr Cat 4	...	0.374338	-0.214611	-0.328738	-0.108100	1.000000
PropCat 5	...	0.089655	-0.148190	0.309856	-0.442524	0.332259
PropCat 6	...	0.195615	0.111493	0.032779	-0.069017	-0.120152
PropCat 7	...	0.120642	0.219124	0.043915	-0.192609	-0.123153
PropCat 8	...	0.125030	0.128941	0.076665	-0.217542	-0.009374
PropCat 14	...	-0.489744	-0.369065	-0.457042	0.997297	-0.110787

	PropCat 5	PropCat 6	PropCat 7	PropCat 8	PropCat 14
SEC	0.015124	-0.031306	-0.117010	-0.242010	0.360679
FEH	-0.024532	0.102199	-0.024065	-0.102663	0.058541
MT	0.001579	0.061889	0.041144	-0.060184	0.010079
SEX	0.053198	0.042404	0.034149	-0.042011	-0.080152
AGE	0.009438	-0.004071	0.038957	0.002299	-0.075245
EDU	0.046768	0.066818	0.111341	0.115574	-0.303237
HS	0.066487	0.026647	0.020185	-0.103354	0.016090
CHILD	-0.059691	-0.065404	-0.008247	0.060404	0.065931
CS	0.018409	-0.008242	0.025228	-0.063462	-0.006235
Affluence Index	0.031577	0.050516	0.119004	0.119975	-0.321604
No. of Brands	-0.051679	0.138059	-0.024206	0.085408	-0.137285
Brand Runs	-0.004715	0.134625	-0.014167	0.153481	-0.259063
Total Volume	0.040042	0.035995	-0.042979	-0.142895	0.117253
No. of Trans	-0.102087	0.156319	0.047981	0.034927	-0.094324
Value	0.042661	0.122436	-0.003889	-0.015935	-0.168652
Trans / Brand Runs	-0.196279	-0.052273	0.063011	-0.147289	0.416105
Vol/Tran	0.131647	-0.105346	-0.085344	-0.166199	0.216795
Avg. Price	-0.022752	0.142057	0.033161	0.264520	-0.567289
Pur Vol No Promo - %	0.027425	0.049619	0.047688	-0.252287	0.111063
Br. Cd. 57, 144	0.581049	-0.207545	-0.168446	-0.152274	-0.234338
Br. Cd. 55	-0.440953	-0.080348	-0.188765	-0.211693	0.992985
Br. Cd. 286	0.216867	-0.053074	-0.059868	-0.038940	-0.095048
Br. Cd. 352	-0.186873	-0.058707	0.626383	-0.036124	-0.111417
Others 999	0.089655	0.195615	0.120642	0.125030	-0.489744
Pr Cat 1	-0.148190	0.111493	0.219124	0.128941	-0.369065
Pr Cat 2	0.309856	0.032779	0.043915	0.076665	-0.457042
Pr Cat 3	-0.442524	-0.069017	-0.192609	-0.217542	0.997297
Pr Cat 4	0.332259	-0.120152	-0.123153	-0.009374	-0.110787

```
PropCat 5              1.000000  -0.330332  -0.312384  -0.200354   -0.442311
PropCat 6             -0.330332   1.000000  -0.075992  -0.080462   -0.078197
PropCat 7             -0.312384  -0.075992   1.000000  -0.079426   -0.191997
PropCat 8             -0.200354  -0.080462  -0.079426   1.000000   -0.216222
PropCat 14            -0.442311  -0.078197  -0.191997  -0.216222    1.000000

[33 rows x 33 columns]
```

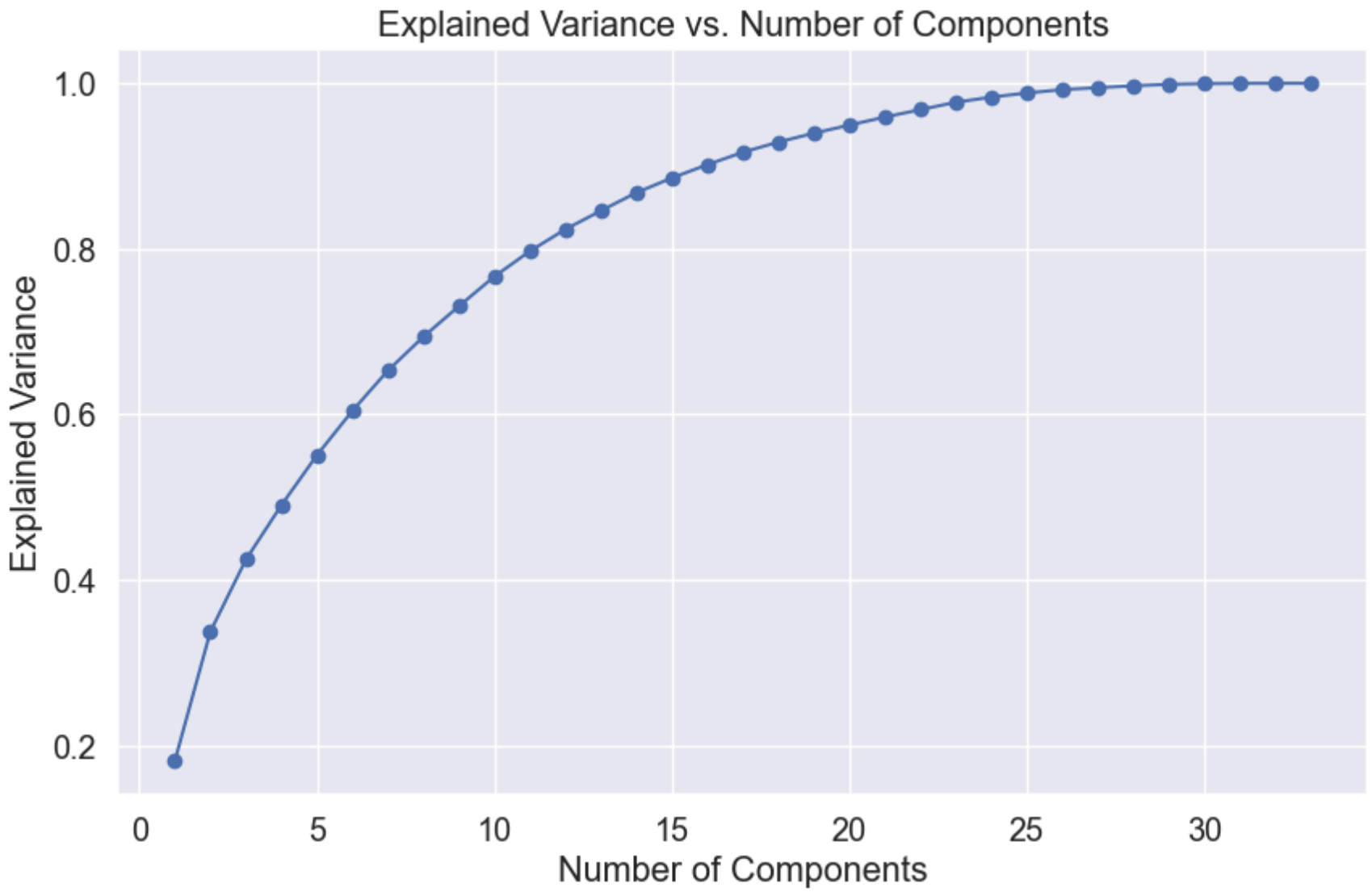
In [15]:

```python
# Examine the top correlations
# Create a DataFrame from the correlation matrix
df = pd.DataFrame(correlation_matrix)

# Get the upper triangle of the correlation matrix (excluding the diagonal)
upper_triangle = df.where(np.triu(np.ones(df.shape), k=1).astype(bool))

# Stack the upper triangle into a series
correlation_series = upper_triangle.stack()

# Sort the correlations in descending order (highest positive first)
sorted_correlations = correlation_series.sort_values(ascending=False)

# Get the top 20 positive correlations
top_positive_correlations = sorted_correlations.head(20)

# Get the top 20 negative correlations
top_negative_correlations = sorted_correlations.tail(20)

# Display the results
print("Top 20 Positive Correlations:")
print(top_positive_correlations)

print("\nTop 20 Negative Correlations:")
print(top_negative_correlations)
```

```
Top 20 Positive Correlations:
Pr Cat 3         PropCat 14         0.997297
Br. Cd. 55       PropCat 14         0.992985
                 Pr Cat 3           0.988652
Total Volume     Value              0.876424
Avg. Price       Pr Cat 1           0.786073
Brand Runs       No. of  Trans      0.774296
No. of Brands    Brand Runs         0.688973
FEH              MT                 0.656519
MT               SEX                0.655163
EDU              Affluence Index    0.649089
SEX              EDU                0.640400
                 HS                 0.635814
Br. Cd. 352      PropCat 7          0.626383
SEX              CS                 0.620378
Total Volume     Vol/Tran           0.620242
FEH              SEX                0.609202
HS               Total Volume       0.588335
Br. Cd. 57, 144  PropCat 5          0.581049
No. of  Trans    Value              0.571126
No. of Brands    No. of  Trans      0.543916
dtype: float64

Top 20 Negative Correlations:
MT               CHILD             -0.392515
SEC              Pr Cat 1          -0.394096
                 Affluence Index   -0.405610
Br. Cd. 57, 144  Others 999        -0.412884
Br. Cd. 55       PropCat 5         -0.440953
PropCat 5        PropCat 14        -0.442311
Pr Cat 3         PropCat 5         -0.442524
Pr Cat 1         Pr Cat 2          -0.448498
Br. Cd. 55       Pr Cat 2          -0.454932
Pr Cat 2         PropCat 14        -0.457042
                 Pr Cat 3          -0.457383
SEC              Avg. Price        -0.478328
Others 999       Pr Cat 3          -0.482266
                 PropCat 14        -0.489744
SEX              CHILD             -0.497819
Br. Cd. 55       Others 999        -0.500151
HS               CHILD             -0.548539
Avg. Price       Br. Cd. 55        -0.558534
                 PropCat 14        -0.567289
                 Pr Cat 3          -0.571698
dtype: float64
```

In [16]:
```
# Run Explained Variance to determine the best number of components for PCA
# Create an empty list to store explained variances
explained_variances = []

# Define a range of component numbers (e.g., 1 to n)
component_range = range(1, min(filtered_df.shape) + 1)

for n_components in component_range:
    pca = PCA(n_components=n_components)
    pca.fit(filtered_df)
    explained_variances.append(np.sum(pca.explained_variance_ratio_))
```

In [17]:
```
# Plot the EV vs the Number of Components
# Use the elbow method to determine the number of components
plt.figure(figsize=(10, 6))
plt.plot(component_range, explained_variances, marker='o', linestyle='-')
plt.title('Explained Variance vs. Number of Components')
plt.xlabel('Number of Components')
plt.ylabel('Explained Variance')
plt.grid(True)
plt.show()
```



In [18]:
```
# View the numerical Explained Variance vs Number of Components for PCA
# Define the range of component numbers to consider
component_range = range(1, min(filtered_df.shape) + 1)

for n_components in component_range:
    pca = PCA(n_components=n_components)
    pca.fit(filtered_df)

    # Calculate the explained variance ratio
```

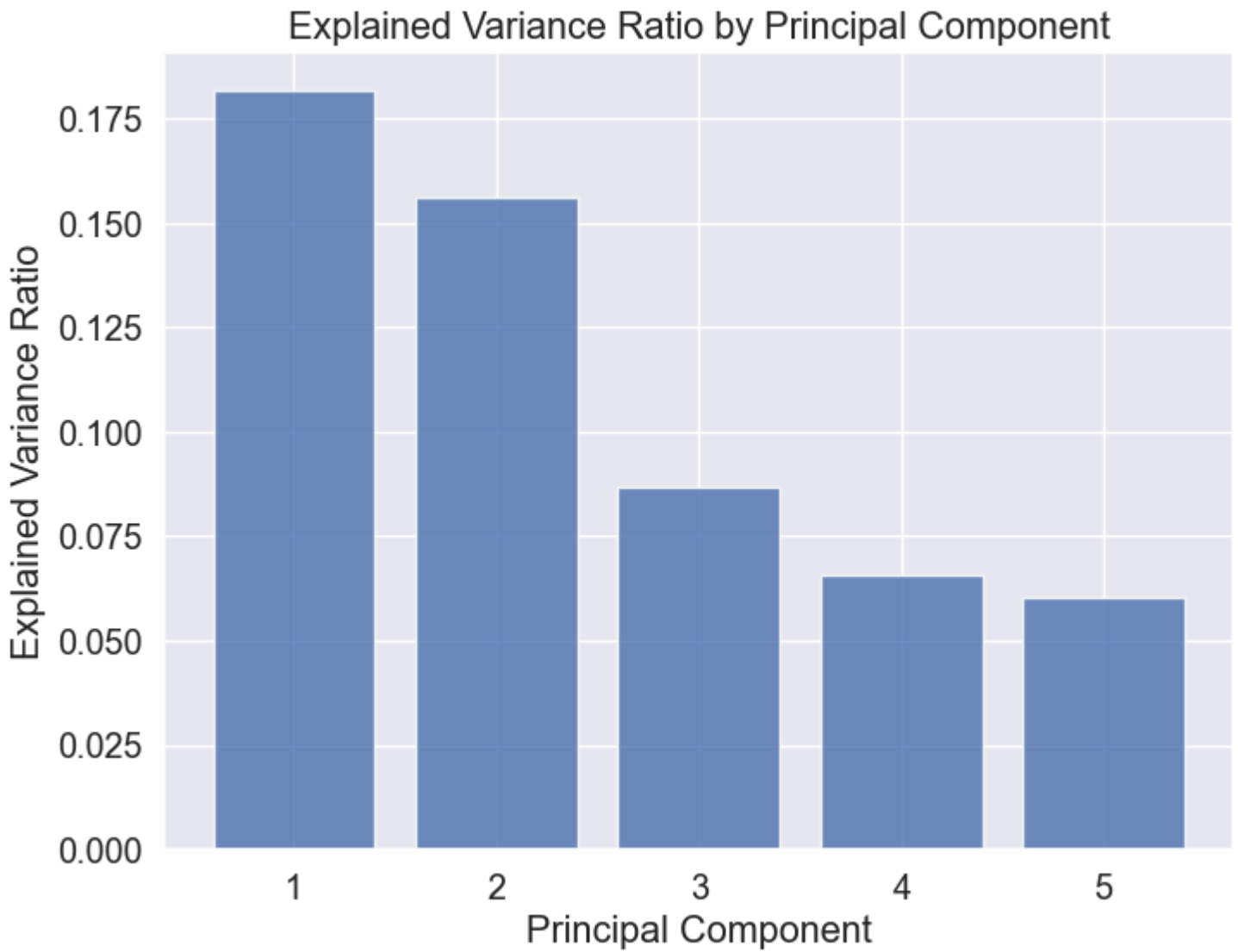
```
    explained_variance = np.sum(pca.explained_variance_ratio_)

    # Print the result
    print(f"Number of Components: {n_components}, Explained Variance: {explained_variance:.4f}")
```

```
Number of Components: 1, Explained Variance: 0.1816
Number of Components: 2, Explained Variance: 0.3377
Number of Components: 3, Explained Variance: 0.4246
Number of Components: 4, Explained Variance: 0.4904
Number of Components: 5, Explained Variance: 0.5507
Number of Components: 6, Explained Variance: 0.6044
Number of Components: 7, Explained Variance: 0.6530
Number of Components: 8, Explained Variance: 0.6938
Number of Components: 9, Explained Variance: 0.7307
Number of Components: 10, Explained Variance: 0.7666
Number of Components: 11, Explained Variance: 0.7972
Number of Components: 12, Explained Variance: 0.8236
Number of Components: 13, Explained Variance: 0.8460
Number of Components: 14, Explained Variance: 0.8679
Number of Components: 15, Explained Variance: 0.8857
Number of Components: 16, Explained Variance: 0.9017
Number of Components: 17, Explained Variance: 0.9167
Number of Components: 18, Explained Variance: 0.9290
Number of Components: 19, Explained Variance: 0.9397
Number of Components: 20, Explained Variance: 0.9495
Number of Components: 21, Explained Variance: 0.9592
Number of Components: 22, Explained Variance: 0.9682
Number of Components: 23, Explained Variance: 0.9771
Number of Components: 24, Explained Variance: 0.9833
Number of Components: 25, Explained Variance: 0.9883
Number of Components: 26, Explained Variance: 0.9924
Number of Components: 27, Explained Variance: 0.9947
Number of Components: 28, Explained Variance: 0.9970
Number of Components: 29, Explained Variance: 0.9987
Number of Components: 30, Explained Variance: 0.9996
Number of Components: 31, Explained Variance: 0.9999
Number of Components: 32, Explained Variance: 1.0000
Number of Components: 33, Explained Variance: 1.0000
```

The explained variance begins to level off after 5 components, so I will use 5 components.

In [19]:
```
# Run PCA with 5 components
n_components = 5  # You can adjust this value as needed
pca = PCA(n_components=n_components)
```

```python
# Fit PCA to the standardized data
pca.fit(filtered_df)

# Transform the data into principal components
pca_data = pca.transform(filtered_df)

# Create a DataFrame to store the principal components
pca_df = pd.DataFrame(data=pca_data, columns=[f'PC{i+1}' for i in range(n_components)])

# Variance explained by each principal component
explained_variance_ratio = pca.explained_variance_ratio_

# Plot the explained variance ratio
plt.figure(figsize=(8, 6))
plt.bar(range(1, n_components + 1), explained_variance_ratio, alpha=0.8, align='center')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.title('Explained Variance Ratio by Principal Component')
plt.show()

# Inspect the loadings (coefficients) of each original variable on the principal components
loadings_df = pd.DataFrame(pca.components_, columns=filtered_df.columns)

# Print the loadings for each principal component
for i, pc in enumerate(range(1, n_components + 1)):
    print(f"Principal Component {pc} Loadings:")
    print(loadings_df.iloc[i].sort_values(ascending=False))
    print()
```


Explained Variance Ratio by Principal Component
Explained Variance Ratio
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
1
2
3
4
5
Principal Component

```
Principal Component 1 Loadings:
PropCat 14              0.263395
Br. Cd. 55              0.262931
Pr Cat 3                0.261883
CHILD                   0.184551
Trans / Brand Runs      0.113947
SEC                     0.113249
Vol/Tran                0.056679
Pur Vol No Promo - %    0.041664
Pr Cat 4               -0.006326
Br. Cd. 57, 144        -0.020461
Br. Cd. 286            -0.025761
Br. Cd. 352            -0.034636
PropCat 7              -0.056042
PropCat 8              -0.063098
PropCat 6              -0.065186
AGE                    -0.070453
PropCat 5              -0.079493
Pr Cat 1               -0.100422
Pr Cat 2               -0.130859
Total Volume           -0.141215
Others 999             -0.157910
Avg. Price             -0.158951
FEH                    -0.165702
CS                     -0.180740
MT                     -0.189502
HS                     -0.208206
No. of Brands          -0.208296
Value                  -0.230740
No. of  Trans          -0.252418
SEX                    -0.267544
Brand Runs             -0.274952
EDU                    -0.278355
Affluence Index        -0.285266
Name: 0, dtype: float64

Principal Component 2 Loadings:
Total Volume            0.288037
Pr Cat 3                0.280732
PropCat 14              0.279513
Br. Cd. 55              0.277536
HS                      0.267787
FEH                     0.249508
MT                      0.234335
SEX                     0.229147
```

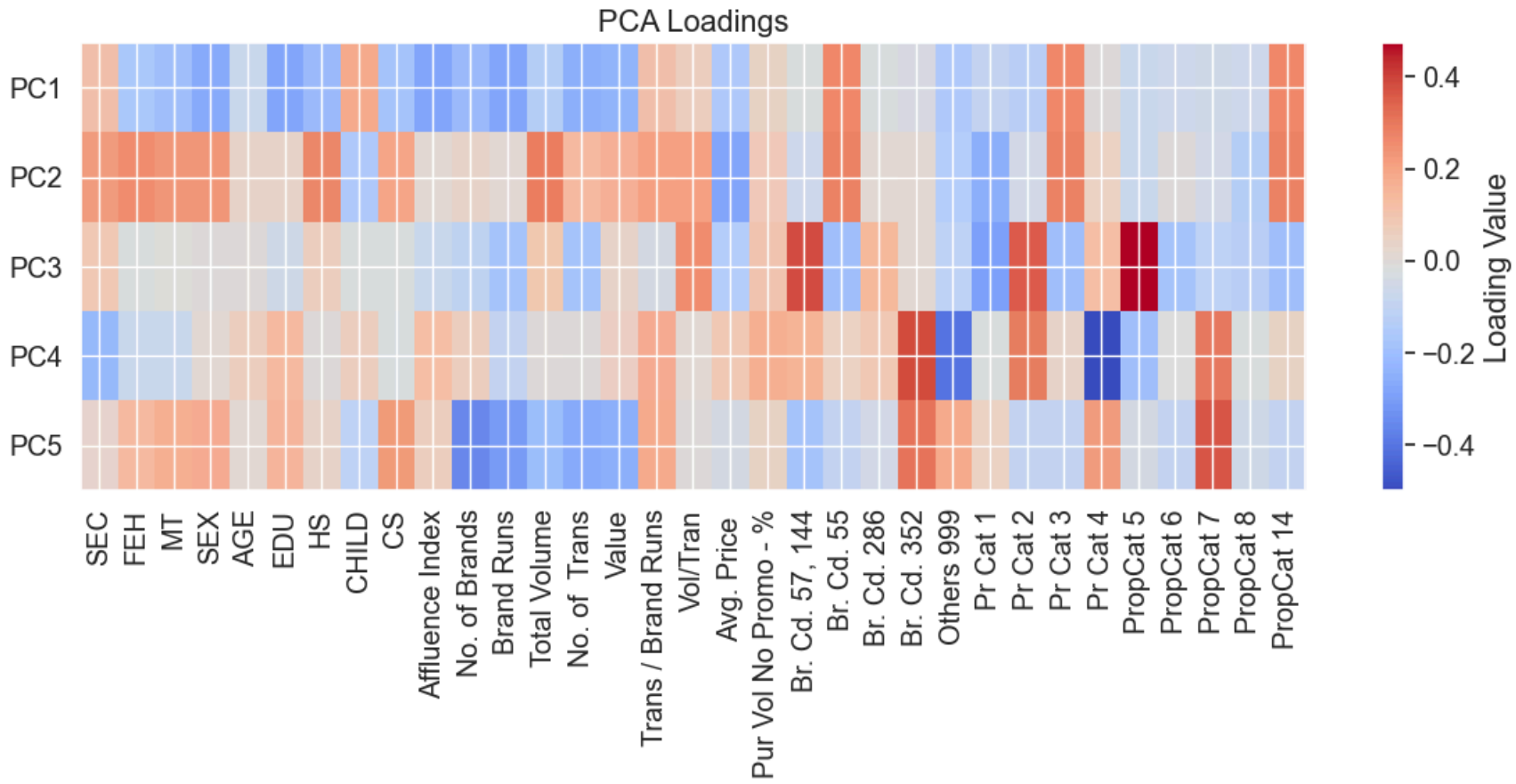
```
SEC                     0.221482
Vol/Tran                0.208330
Trans / Brand Runs      0.201185
CS                      0.193761
Value                   0.159659
No. of  Trans           0.132013
Pur Vol No Promo - %    0.077850
Pr Cat 4                0.048931
No. of Brands           0.039580
AGE                     0.033542
EDU                     0.029209
Br. Cd. 352             0.012747
Brand Runs              0.007155
Br. Cd. 286             0.005824
Affluence Index         0.004134
PropCat 6              -0.006612
PropCat 7              -0.049165
Pr Cat 2               -0.050553
Br. Cd. 57, 144        -0.062610
PropCat 5              -0.071825
PropCat 8              -0.132050
Others 999             -0.143048
CHILD                  -0.156373
Pr Cat 1               -0.245201
Avg. Price             -0.281881
Name: 1, dtype: float64

Principal Component 3 Loadings:
PropCat 5               0.473685
Br. Cd. 57, 144         0.379512
Pr Cat 2                0.352302
Vol/Tran                0.252164
Br. Cd. 286             0.146497
Pr Cat 4                0.116596
Pur Vol No Promo - %    0.094213
Total Volume            0.092011
SEC                     0.085343
HS                      0.066306
Value                   0.039794
Br. Cd. 352             0.010800
AGE                     0.000972
SEX                    -0.006527
MT                     -0.012776
CHILD                  -0.017752
FEH                    -0.019136
```

```
CS                     -0.024015
Trans / Brand Runs     -0.037456
EDU                    -0.054580
Affluence Index        -0.083063
No. of Brands          -0.106693
PropCat 7              -0.113785
Others 999             -0.114367
PropCat 8              -0.123972
Avg. Price             -0.143432
Brand Runs             -0.177554
No. of  Trans          -0.181458
PropCat 6              -0.186074
Br. Cd. 55             -0.191855
PropCat 14             -0.191898
Pr Cat 3               -0.192131
Pr Cat 1               -0.287036
Name: 2, dtype: float64

Principal Component 4 Loadings:
Br. Cd. 352             0.385652
PropCat 7               0.298714
Pr Cat 2                0.294379
Trans / Brand Runs      0.176888
Pur Vol No Promo - %    0.162255
Br. Cd. 57, 144         0.154551
EDU                     0.135505
Affluence Index         0.119719
Br. Cd. 286             0.085476
Avg. Price              0.080070
No. of Brands           0.064796
AGE                     0.059649
CHILD                   0.058884
Value                   0.052619
Br. Cd. 55              0.047814
PropCat 14              0.040536
Pr Cat 3                0.037101
SEX                     0.011659
Vol/Tran                0.005660
Total Volume           -0.000077
HS                     -0.002370
No. of  Trans          -0.005289
PropCat 6              -0.013595
Pr Cat 1               -0.020837
PropCat 8              -0.023976
CS                     -0.026109
```

```
MT                     -0.073165
FEH                    -0.085070
Brand Runs             -0.086263
PropCat 5              -0.202628
SEC                    -0.225475
Others 999             -0.405348
Pr Cat 4               -0.499858
Name: 3, dtype: float64

Principal Component 5 Loadings:
PropCat 7               0.370254
Br. Cd. 352             0.305763
CS                      0.225649
Pr Cat 4                0.222567
Trans / Brand Runs      0.185901
Others 999              0.178344
SEX                     0.174046
MT                      0.170338
EDU                     0.149192
FEH                     0.130097
Affluence Index         0.063876
Pr Cat 1                0.044918
Pur Vol No Promo - %    0.041540
HS                      0.039586
SEC                     0.025735
AGE                     0.010072
Vol/Tran               -0.004824
Avg. Price             -0.041874
PropCat 5              -0.042072
Br. Cd. 286            -0.050542
PropCat 8              -0.052943
PropCat 6              -0.088748
Pr Cat 2               -0.093455
Pr Cat 3               -0.097603
PropCat 14             -0.098075
Br. Cd. 55             -0.099760
CHILD                  -0.111120
Br. Cd. 57, 144        -0.179171
Total Volume           -0.206729
Value                  -0.251174
No. of  Trans          -0.256923
Brand Runs             -0.311128
No. of Brands          -0.355332
Name: 4, dtype: float64
```

In [20]:

```
# Select the relevant columns for PCA
data_for_pca = filtered_df.select_dtypes(include=['float64', 'int64'])

# Access PCA loadings
loadings = pca.components_

# Create a DataFrame to display the loadings
loadings_df = pd.DataFrame(loadings, columns=data_for_pca.columns, index=[f'PC{i+1}' for i in range(n_components)])

# Visualize the loadings
plt.figure(figsize=(12, 6))
plt.title("PCA Loadings")
plt.imshow(loadings_df, cmap='coolwarm', aspect='auto', interpolation='none')
plt.colorbar(label='Loading Value')
plt.xticks(range(len(data_for_pca.columns)), data_for_pca.columns, rotation=90)
plt.yticks(range(n_components), loadings_df.index)
plt.tight_layout()
plt.show()
```



In [21]:
```python
# Print the loadings for each principal component
for i, pc in enumerate(range(1, n_components + 1)):
    print(f"Principal Component {pc} Loadings:")
    print(loadings_df.iloc[i].sort_values(ascending=False))
    print()
```

```
Principal Component 1 Loadings:
PropCat 14              0.263395
Br. Cd. 55              0.262931
Pr Cat 3                0.261883
CHILD                   0.184551
Trans / Brand Runs      0.113947
SEC                     0.113249
Vol/Tran                0.056679
Pur Vol No Promo - %    0.041664
Pr Cat 4               -0.006326
Br. Cd. 57, 144        -0.020461
Br. Cd. 286            -0.025761
Br. Cd. 352            -0.034636
PropCat 7              -0.056042
PropCat 8              -0.063098
PropCat 6              -0.065186
AGE                    -0.070453
PropCat 5              -0.079493
Pr Cat 1               -0.100422
Pr Cat 2               -0.130859
Total Volume           -0.141215
Others 999             -0.157910
Avg. Price             -0.158951
FEH                    -0.165702
CS                     -0.180740
MT                     -0.189502
HS                     -0.208206
No. of Brands          -0.208296
Value                  -0.230740
No. of  Trans          -0.252418
SEX                    -0.267544
Brand Runs             -0.274952
EDU                    -0.278355
Affluence Index        -0.285266
Name: PC1, dtype: float64

Principal Component 2 Loadings:
Total Volume            0.288037
Pr Cat 3                0.280732
PropCat 14              0.279513
Br. Cd. 55              0.277536
HS                      0.267787
FEH                     0.249508
MT                      0.234335
SEX                     0.229147
```

```
SEC                     0.221482
Vol/Tran                0.208330
Trans / Brand Runs      0.201185
CS                      0.193761
Value                   0.159659
No. of  Trans           0.132013
Pur Vol No Promo - %    0.077850
Pr Cat 4                0.048931
No. of Brands           0.039580
AGE                     0.033542
EDU                     0.029209
Br. Cd. 352             0.012747
Brand Runs              0.007155
Br. Cd. 286             0.005824
Affluence Index         0.004134
PropCat 6              -0.006612
PropCat 7              -0.049165
Pr Cat 2               -0.050553
Br. Cd. 57, 144        -0.062610
PropCat 5              -0.071825
PropCat 8              -0.132050
Others 999             -0.143048
CHILD                  -0.156373
Pr Cat 1               -0.245201
Avg. Price             -0.281881
Name: PC2, dtype: float64

Principal Component 3 Loadings:
PropCat 5               0.473685
Br. Cd. 57, 144         0.379512
Pr Cat 2                0.352302
Vol/Tran                0.252164
Br. Cd. 286             0.146497
Pr Cat 4                0.116596
Pur Vol No Promo - %    0.094213
Total Volume            0.092011
SEC                     0.085343
HS                      0.066306
Value                   0.039794
Br. Cd. 352             0.010800
AGE                     0.000972
SEX                    -0.006527
MT                     -0.012776
CHILD                  -0.017752
FEH                    -0.019136
```

```
CS                     -0.024015
Trans / Brand Runs     -0.037456
EDU                    -0.054580
Affluence Index        -0.083063
No. of Brands          -0.106693
PropCat 7              -0.113785
Others 999             -0.114367
PropCat 8              -0.123972
Avg. Price             -0.143432
Brand Runs             -0.177554
No. of  Trans          -0.181458
PropCat 6              -0.186074
Br. Cd. 55             -0.191855
PropCat 14             -0.191898
Pr Cat 3               -0.192131
Pr Cat 1               -0.287036
Name: PC3, dtype: float64

Principal Component 4 Loadings:
Br. Cd. 352             0.385652
PropCat 7               0.298714
Pr Cat 2                0.294379
Trans / Brand Runs      0.176888
Pur Vol No Promo - %    0.162255
Br. Cd. 57, 144         0.154551
EDU                     0.135505
Affluence Index         0.119719
Br. Cd. 286             0.085476
Avg. Price              0.080070
No. of Brands           0.064796
AGE                     0.059649
CHILD                   0.058884
Value                   0.052619
Br. Cd. 55              0.047814
PropCat 14              0.040536
Pr Cat 3                0.037101
SEX                     0.011659
Vol/Tran                0.005660
Total Volume           -0.000077
HS                     -0.002370
No. of  Trans          -0.005289
PropCat 6              -0.013595
Pr Cat 1               -0.020837
PropCat 8              -0.023976
CS                     -0.026109
```

```
MT                     -0.073165
FEH                    -0.085070
Brand Runs             -0.086263
PropCat 5              -0.202628
SEC                    -0.225475
Others 999             -0.405348
Pr Cat 4               -0.499858
Name: PC4, dtype: float64

Principal Component 5 Loadings:
PropCat 7               0.370254
Br. Cd. 352             0.305763
CS                      0.225649
Pr Cat 4                0.222567
Trans / Brand Runs      0.185901
Others 999              0.178344
SEX                     0.174046
MT                      0.170338
EDU                     0.149192
FEH                     0.130097
Affluence Index         0.063876
Pr Cat 1                0.044918
Pur Vol No Promo - %    0.041540
HS                      0.039586
SEC                     0.025735
AGE                     0.010072
Vol/Tran               -0.004824
Avg. Price             -0.041874
PropCat 5              -0.042072
Br. Cd. 286            -0.050542
PropCat 8              -0.052943
PropCat 6              -0.088748
Pr Cat 2               -0.093455
Pr Cat 3               -0.097603
PropCat 14             -0.098075
Br. Cd. 55             -0.099760
CHILD                  -0.111120
Br. Cd. 57, 144        -0.179171
Total Volume           -0.206729
Value                  -0.251174
No. of  Trans          -0.256923
Brand Runs             -0.311128
No. of Brands          -0.355332
Name: PC5, dtype: float64
```

In [22]:
```
# Print the data frame columns to comform variable names to loadings_data
print("Columns in data_for_pca:")
print(data_for_pca.columns)
```

```
Columns in data_for_pca:
Index(['SEC', 'FEH', 'MT', 'SEX', 'AGE', 'EDU', 'HS', 'CHILD', 'CS',
       'Affluence Index', 'No. of Brands', 'Brand Runs', 'Total Volume',
       'No. of  Trans', 'Value', 'Trans / Brand Runs', 'Vol/Tran',
       'Avg. Price ', 'Pur Vol No Promo - %', 'Br. Cd. 57, 144', 'Br. Cd. 55',
       'Br. Cd. 286', 'Br. Cd. 352', 'Others 999', 'Pr Cat 1', 'Pr Cat 2',
       'Pr Cat 3', 'Pr Cat 4', 'PropCat 5', 'PropCat 6', 'PropCat 7',
       'PropCat 8', 'PropCat 14'],
      dtype='object')
```

In [23]:
```
# Select and print the appropriate individual variables per column to be included in the cluster analysis abs() >= .2

# Define the PC loadings data
loadings_data = {
    'Variable': ['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'CHILD', 'Trans / Brand Runs', 'SEC', 'Vol/Tran', 'Pur Vol No
    'PC1 Loadings': [0.263394, 0.262931, 0.261883, 0.184555, 0.113949, 0.113251, 0.056679, 0.041664, -0.006329, -0.0204
    'PC2 Loadings': [0.288039, 0.280732, 0.279513, 0.277536, 0.267782, 0.249505, 0.234331, 0.229146, 0.221480, 0.208332
    'PC3 Loadings': [0.473647, 0.379503, 0.352349, 0.252189, 0.146359, 0.116624, 0.094494, 0.092042, 0.085249, 0.066222
    'PC4 Loadings': [0.384784, 0.299136, 0.294861, 0.177062, 0.161533, 0.154180, 0.135952, 0.119114, 0.086330, 0.080234
    'PC5 Loadings': [0.370049, 0.306358, 0.224505, 0.223490, 0.186108, 0.177854, 0.173907, 0.169921, 0.148892, 0.129842
}

# Create a DataFrame from the loadings data
loadings_df = pd.DataFrame(loadings_data)

# Specify the number of principal components for clustering
selected_pcs = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5']

# Create a list to store variables with high loadings
selected_variables = []

# Iterate through selected principal components and identify variables with high loadings
for pc in selected_pcs:
    # Get variables with loadings above a threshold (e.g., 0.2 for high positive and -0.2 for high negative loadings)
    high_loadings = loadings_df[loadings_df[f'{pc} Loadings'].abs() >= 0.2]['Variable'].tolist()
    selected_variables.extend(high_loadings)

# Remove duplicates (in case a variable has high loadings in multiple selected PCs)
selected_variables = list(set(selected_variables))

# Create a subset with the selected variables for cluster analysis
```

```
subset_data = data_for_pca[['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'CHILD', 'Trans / Brand Runs', 'SEC', 'Vol/Tran', '

# Print the selected variables for cluster analysis
print("Selected Variables for Cluster Analysis:")
print(subset_data)
```

Selected Variables for Cluster Analysis:

	PropCat 14	Br. Cd. 55	Pr Cat 3	CHILD	Trans / Brand Runs	SEC	\
0	-0.021215	0.005756	-0.031227	0.630433	-0.464084	1.341641	
1	-0.210590	-0.208906	-0.165759	-1.014175	-0.391077	0.447214	
2	1.599587	1.603433	1.577546	0.630433	-0.352653	-0.447214	
3	1.743712	1.813619	1.720601	1.452736	-0.621624	1.341641	
4	0.030462	0.058691	0.020067	-0.191871	-0.172058	1.341641	
..	...	...	...	...	...	...	
595	-0.513426	-0.498429	-0.519785	0.630433	0.684807	-0.447214	
596	-0.513426	-0.498429	-0.519785	0.630433	-0.537090	0.447214	
597	0.165266	0.196775	0.153871	0.630433	-0.498666	0.447214	
598	-0.000440	-0.078056	0.260956	0.630433	-0.517878	-0.447214	
599	-0.513426	-0.498429	-0.519785	0.630433	-0.406447	0.447214	

	Vol/Tran	Pur Vol No Promo - %	Pr Cat 4	Br. Cd. 57, 144	...	CS	\
0	-0.324562	0.728635	-0.072320	0.817531	...	0.134793	
1	-0.264213	-0.215341	-0.126441	-0.687277	...	0.134793	
2	-0.194651	0.239133	-0.462691	-0.668198	...	0.134793	
3	-0.161137	0.728635	-0.462691	0.915118	...	-1.837792	
4	0.898835	-2.500632	3.752024	-0.574142	...	0.134793	
..	...	...	...	...	...	...	
595	-0.019397	-1.150725	-0.462691	-0.590530	...	0.134793	
596	-0.944227	0.106661	0.054261	-0.778150	...	0.134793	
597	3.013226	0.728635	-0.462691	0.265596	...	0.134793	
598	-0.505248	-1.289198	0.296758	-0.778150	...	2.107379	
599	1.864863	0.728635	-0.462691	0.219915	...	0.134793	

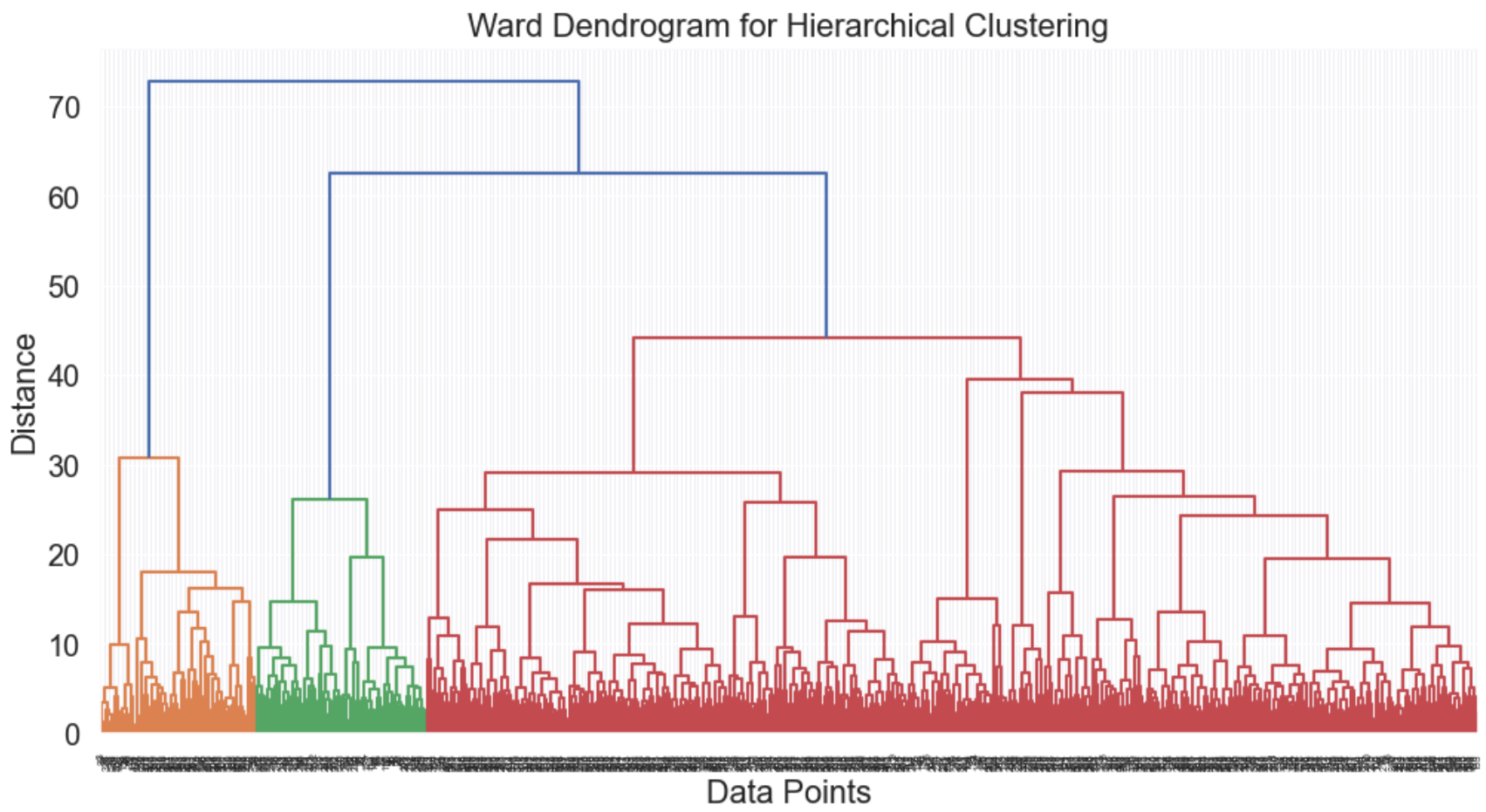
	MT	HS	No. of Brands	Value	No. of Trans	SEX	\
0	0.424526	-0.953656	-0.403364	-0.588594	-0.410811	-1.139458	
1	0.424526	-0.083400	0.863748	0.389966	0.508057	0.403826	
2	0.424526	0.786857	0.863748	0.694243	1.828930	0.403826	
3	-1.905900	-1.823913	-1.036920	-1.386401	-1.559396	-2.682741	
4	0.424526	-0.083400	-0.403364	-0.845841	-1.042532	0.403826	
..	...	...	...	...	...	...	
595	0.424526	0.351729	-0.403364	-0.135294	-0.525669	0.403826	
596	0.424526	-0.083400	0.863748	-0.551197	-0.181094	0.403826	
597	2.055825	2.092242	0.230192	2.227245	-0.353381	0.403826	
598	-0.973730	-0.953656	-0.403364	-0.994297	-0.697957	0.403826	
599	0.424526	0.786857	-0.403364	0.053958	-0.985103	0.403826	

	Brand Runs	EDU	Affluence Index
0	0.120173	-0.019803	-1.317478
1	0.890306	-0.019803	0.173676
2	2.045506	0.437198	0.524535
3	-1.131294	-1.847808	-1.492908

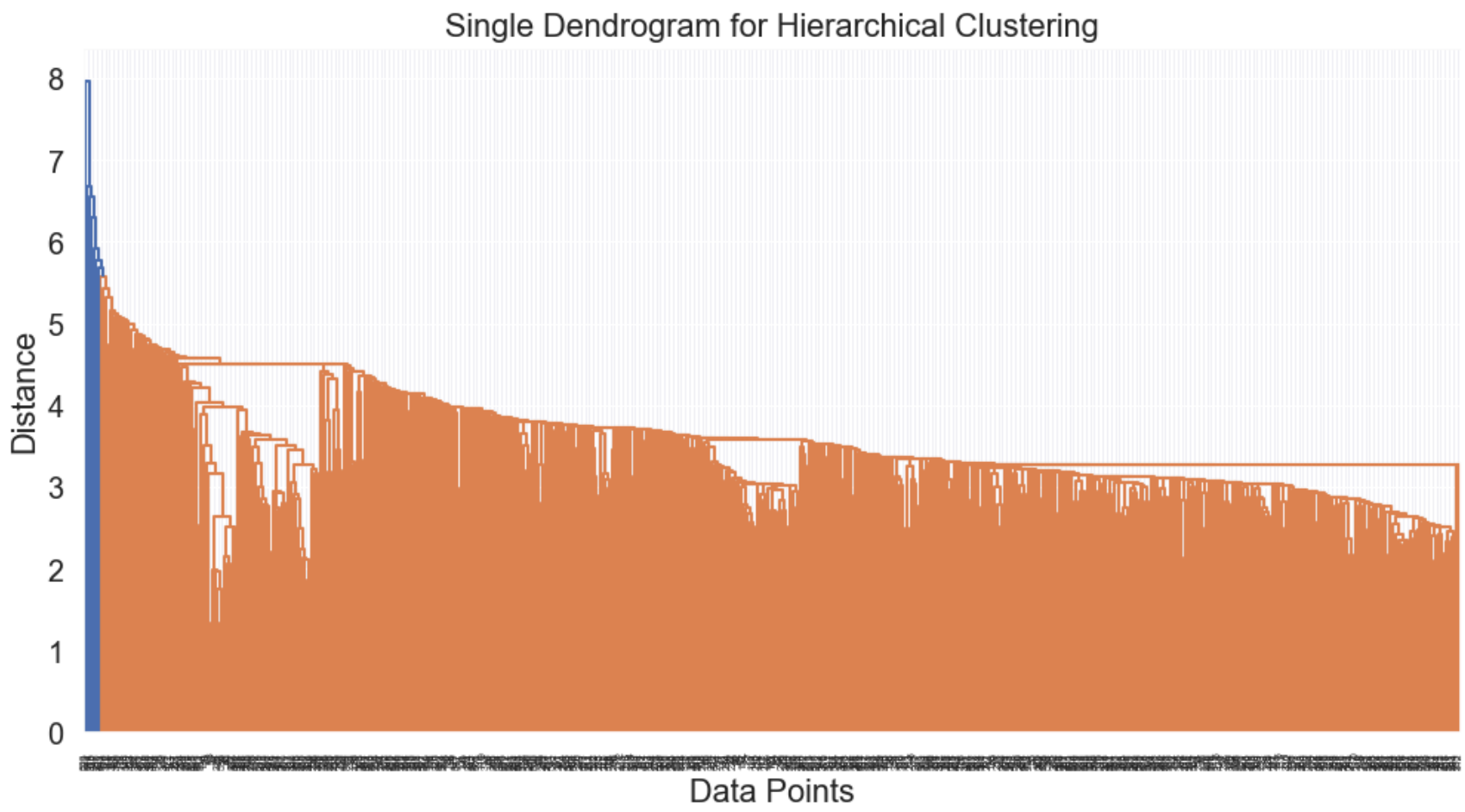
```
4     -0.938760 -0.019803        -0.615759
..          ...       ...              ...
595   -1.035027  0.437198        -0.177184
596    0.697773  0.437198         1.050825
597    0.312706 -0.019803        -0.352614
598   -0.072360  0.437198         0.261391
599   -0.649960 -0.019803        -0.177184

[600 rows x 33 columns]
```

In [24]:

```python
# Create a distance matrix
distance_matrix = pdist(subset_data)

# Perform hierarchical clustering
dendrogram = hierarchy.linkage(distance_matrix, method='ward')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
hierarchy.dendrogram(dendrogram, labels=subset_data.index, leaf_rotation=90)
plt.title('Ward Dendrogram for Hierarchical Clustering')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```

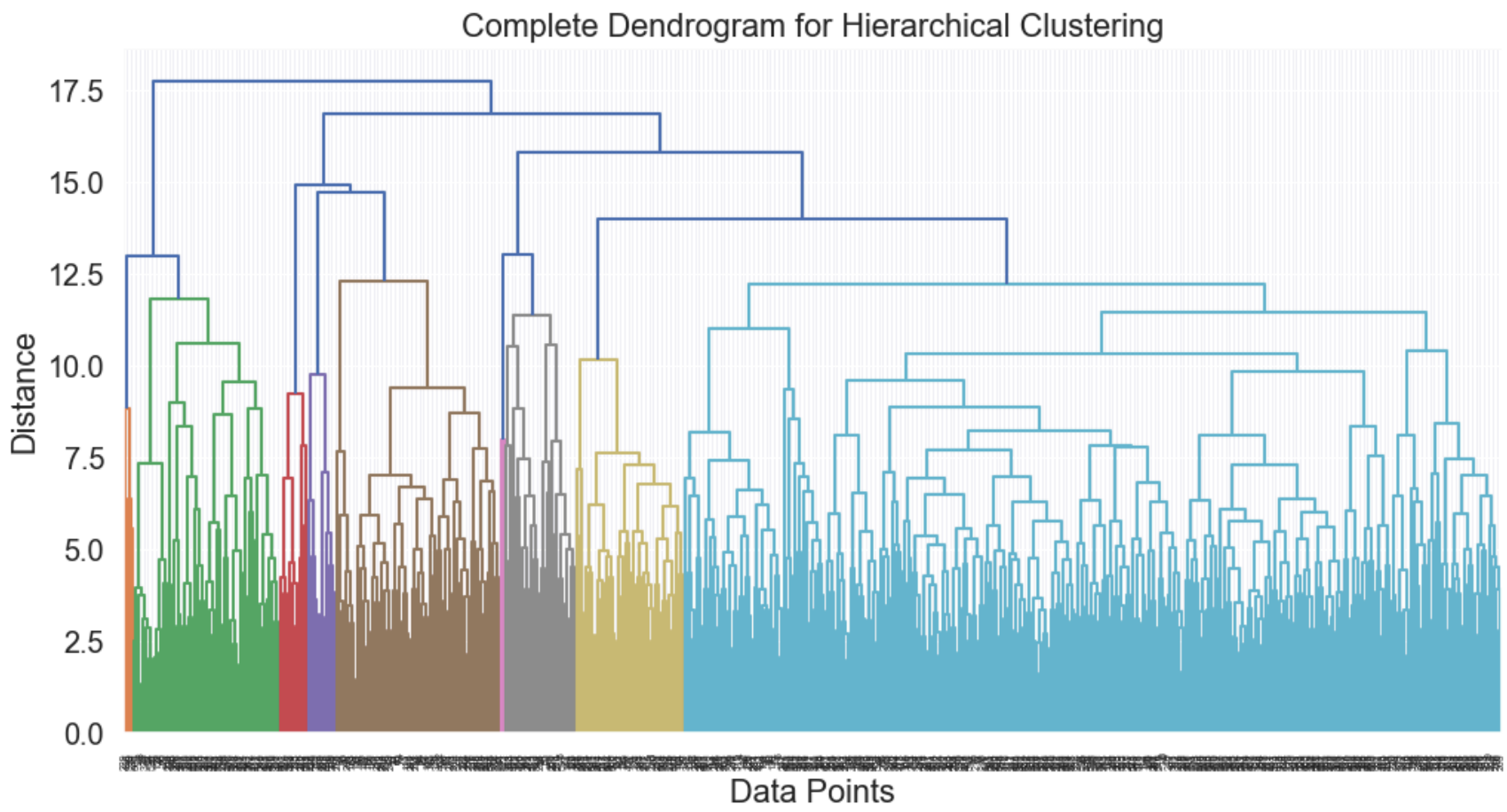


In [25]:
```
# Create a distance matrix
distance_matrix = pdist(subset_data)

# Perform hierarchical clustering
dendrogram = hierarchy.linkage(distance_matrix, method='single')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
hierarchy.dendrogram(dendrogram, labels=subset_data.index, leaf_rotation=90)
plt.title('Single Dendrogram for Hierarchical Clustering')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```

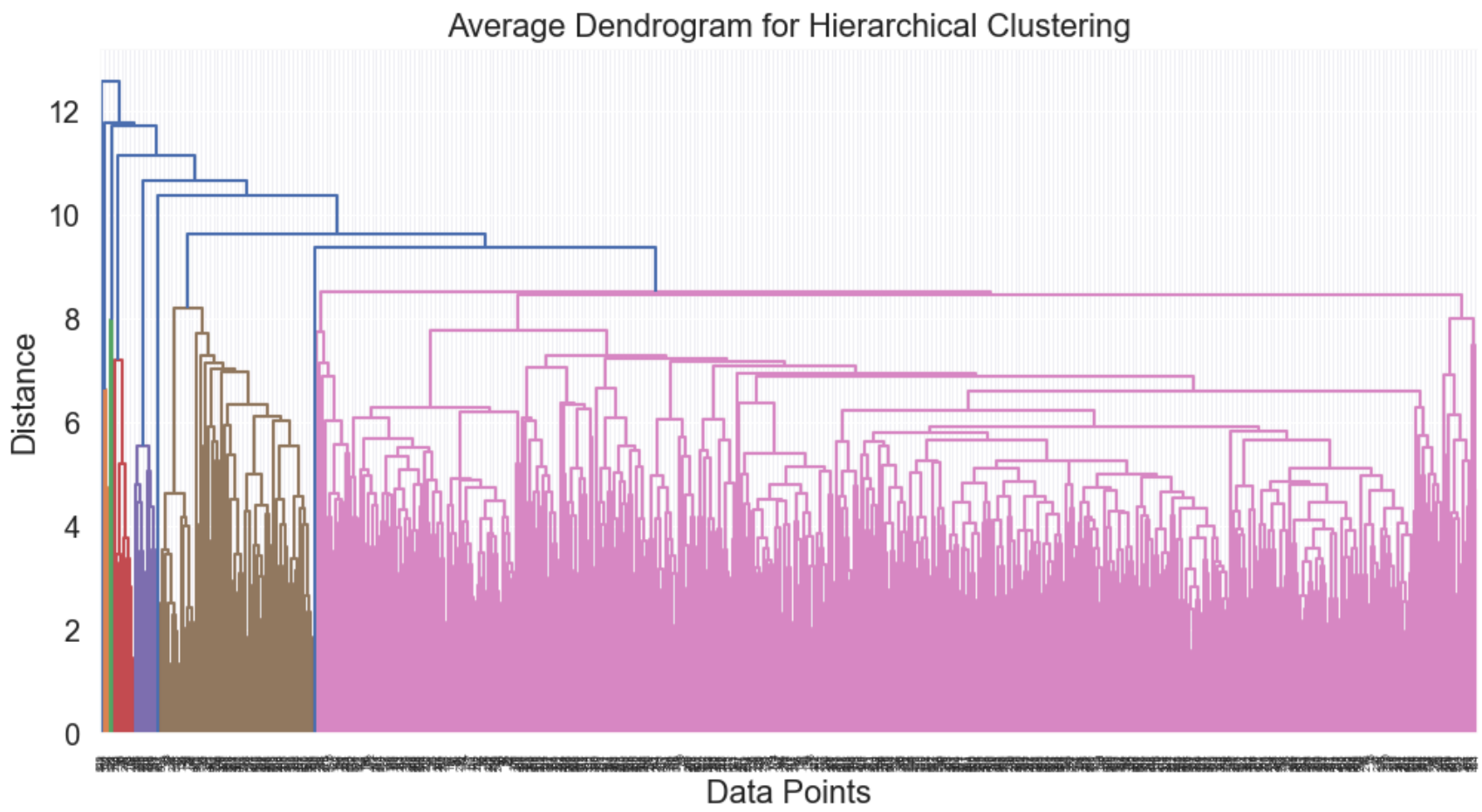


In [26]:

```
# Create a distance matrix
distance_matrix = pdist(subset_data)

# Perform hierarchical clustering
dendrogram = hierarchy.linkage(distance_matrix, method='complete')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
hierarchy.dendrogram(dendrogram, labels=subset_data.index, leaf_rotation=90)
plt.title('Complete Dendrogram for Hierarchical Clustering')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



In [27]:
```python
# Create a distance matrix
distance_matrix = pdist(subset_data)

# Perform hierarchical clustering
dendrogram = hierarchy.linkage(distance_matrix, method='average')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
hierarchy.dendrogram(dendrogram, labels=subset_data.index, leaf_rotation=90)
plt.title('Average Dendrogram for Hierarchical Clustering')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



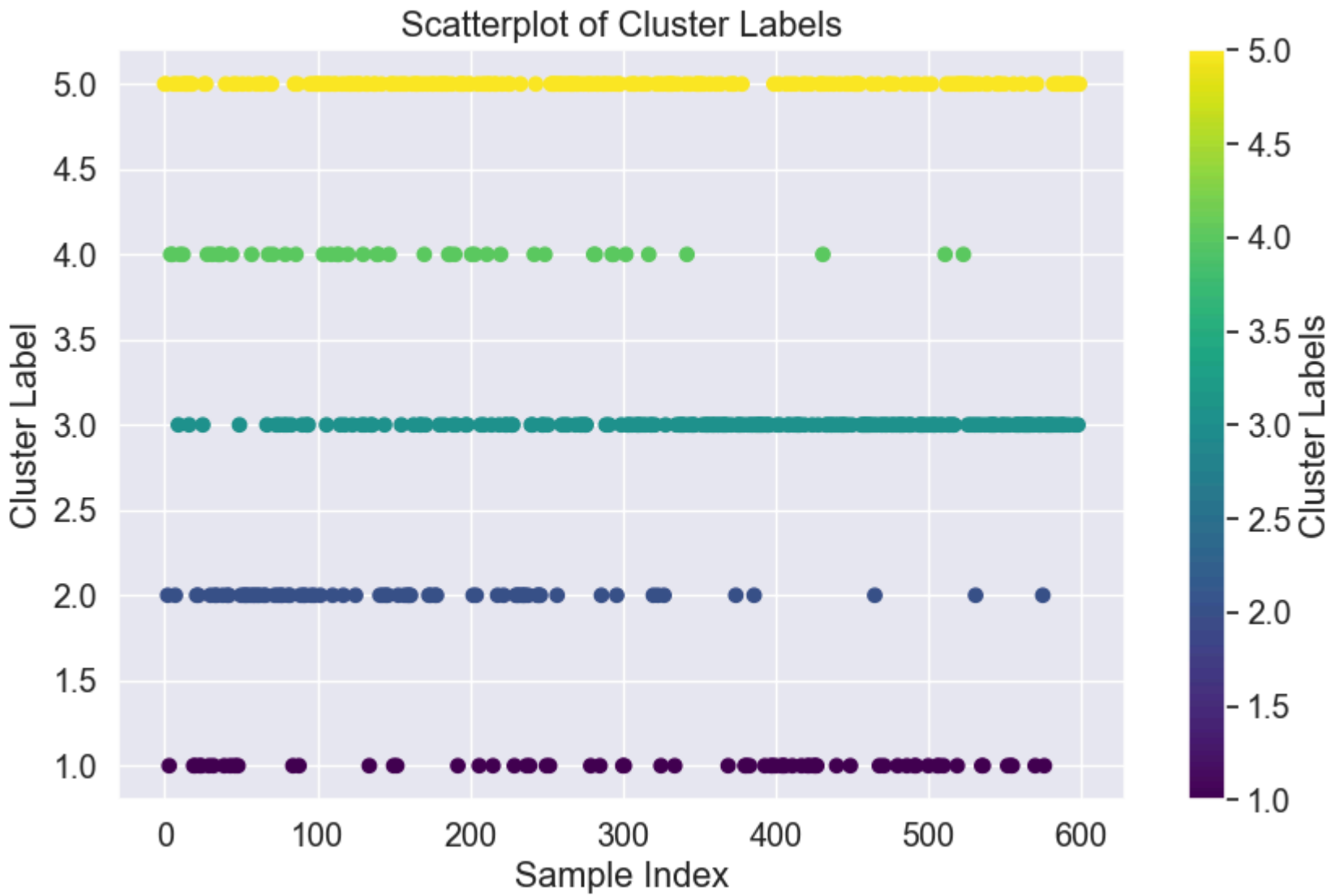
I will use the Ward method because it closely resembles the PCA.

In [28]:
```
# Create a hierarchical clustering linkage matrix
linkage_matrix = hierarchy.linkage(distance_matrix, method='ward')

# Determine the number of clusters (you can adjust the threshold as needed)
num_clusters = 5

# Perform clustering to get cluster labels
cluster_labels = fcluster(linkage_matrix, num_clusters, criterion='maxclust')

# Print the cluster labels
print("Cluster Labels:")
print(cluster_labels)
```

```
Cluster Labels:
[5 5 2 1 4 4 5 2 5 3 4 5 4 5 5 5 3 5 5 1 1 2 2 1 1 3 5 5 4 1 2 4 1 2 2 4 4
 4 2 1 5 2 2 1 4 5 1 5 1 3 2 5 2 2 2 5 2 4 2 2 5 2 2 5 5 2 2 3 4 5 5 4 2 3
 2 3 2 2 3 4 3 2 2 3 1 5 4 5 1 2 3 2 2 3 3 5 2 5 2 5 5 5 2 5 4 5 3 5 5 4 2
 5 5 4 4 3 5 2 3 5 4 5 5 3 5 2 5 5 5 3 4 3 5 5 1 3 3 5 5 4 4 2 5 2 3 2 2 4
 5 5 1 5 1 2 5 3 5 2 5 2 2 2 5 3 5 5 5 3 3 5 4 3 5 2 2 5 5 2 2 5 3 5 5 3 5
 5 4 4 5 3 4 3 1 5 5 5 5 3 3 5 5 4 2 4 2 5 1 3 5 3 5 4 5 5 3 1 5 5 2 3 4 5
 2 3 3 5 5 3 3 1 2 2 2 5 2 2 2 1 2 1 3 3 4 5 2 2 2 3 3 4 1 3 1 5 5 5 5 2 5
 5 3 5 3 5 3 5 5 5 5 3 5 5 3 3 5 3 3 5 5 1 5 4 4 5 5 1 2 5 5 3 5 3 5 4 4 5
 2 5 5 3 1 1 4 3 3 5 3 5 5 3 3 3 5 3 5 3 5 4 3 3 2 3 5 2 5 1 5 2 3 5 5 5 5
 5 1 3 5 3 3 3 3 5 4 3 5 3 3 3 5 5 5 5 3 5 3 3 3 5 3 3 3 5 3 3 5 3 3 3 3 1
 5 3 5 5 2 3 3 3 5 3 1 1 3 1 3 3 2 3 3 3 3 3 3 1 3 3 3 1 1 5 1 3 5 3 1 1 1
 5 3 3 5 1 3 5 3 3 3 1 5 3 5 1 1 5 3 1 1 1 3 5 5 4 5 3 3 5 3 3 5 3 1 3 5 3
 3 3 5 3 5 1 3 5 5 5 5 5 3 3 3 3 3 3 3 5 3 2 3 5 1 3 3 1 3 3 5 5 3 3 5 3 1
 3 3 3 3 5 1 3 3 3 5 1 1 5 3 3 3 3 5 3 1 3 5 3 3 3 1 1 3 3 1 4 5 3 3 5 3 3
 5 1 5 5 5 4 5 5 3 5 3 5 3 2 3 5 3 1 1 3 5 5 3 3 3 3 3 5 5 5 3 3 5 3 1 3 1
 1 5 3 3 3 3 5 3 3 3 3 3 3 5 3 1 5 3 3 3 2 1 3 3 3 3 3 5 3 5 3 5 3 3 3 5 5
 3 5 5 5 3 5 3 5]
```

In [29]:

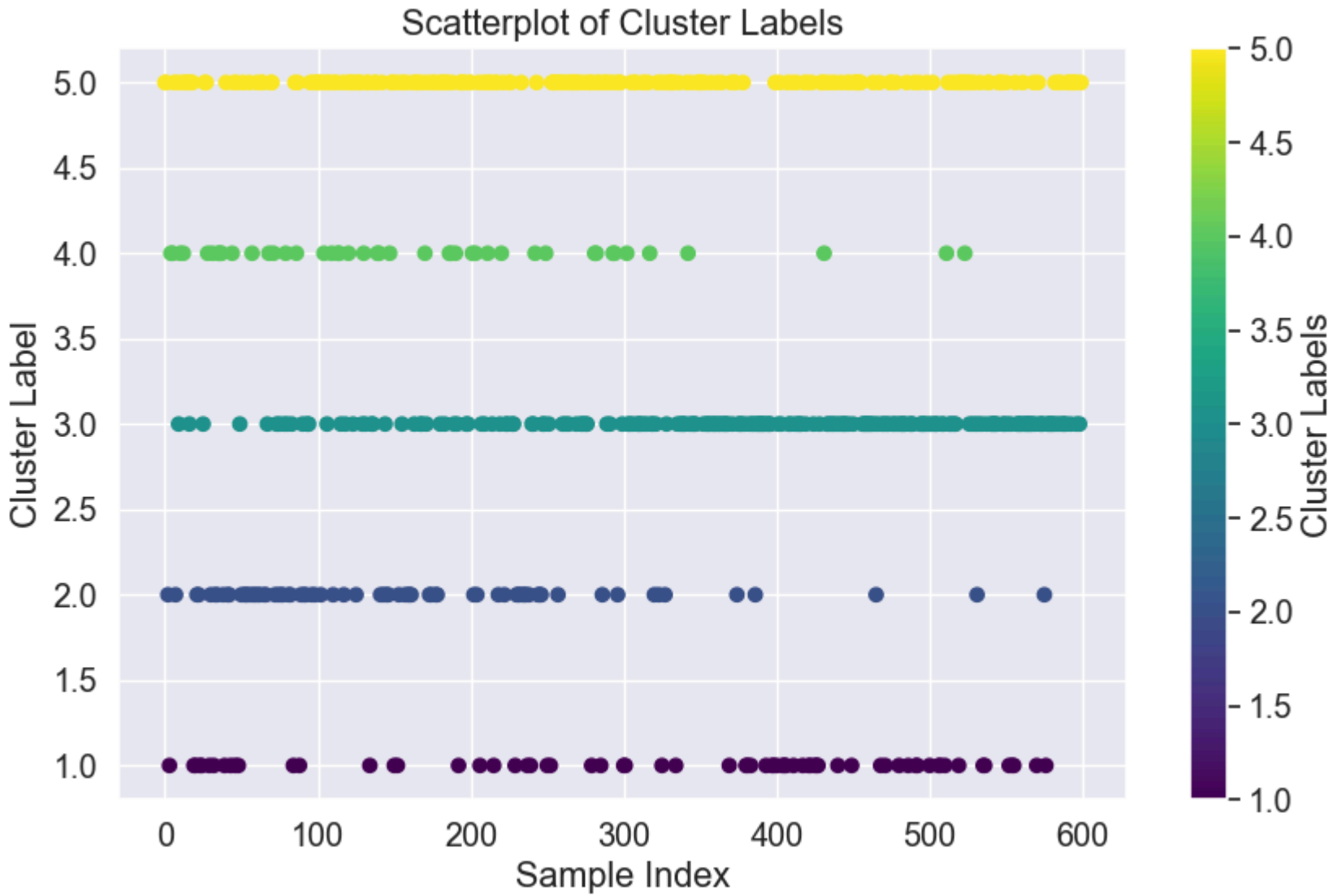
```
# Create a scatterplot of cluster labels
plt.figure(figsize=(10, 6))
plt.scatter(range(len(cluster_labels)), cluster_labels, c=cluster_labels, cmap='viridis')
plt.title('Scatterplot of Cluster Labels')
plt.xlabel('Sample Index')
plt.ylabel('Cluster Label')
plt.colorbar(label='Cluster Labels')
plt.show()
```



Visual displays a higher number of data points in Clusters 3 and 5.

In [30]:
```
# Create a scatterplot of cluster labels
plt.figure(figsize=(10, 6))
plt.scatter(range(len(cluster_labels)), cluster_labels, c=cluster_labels, cmap='viridis')
plt.title('Scatterplot of Cluster Labels')
plt.xlabel('Sample Index')
plt.ylabel('Cluster Label')
plt.colorbar(label='Cluster Labels')
```

```
plt.show()

# Print cluster labels in textual format
for i, label in enumerate(cluster_labels):
    print(f"Sample {i}: Cluster Label {label}")
```



```
Sample 0: Cluster Label 5
Sample 1: Cluster Label 5
Sample 2: Cluster Label 2
Sample 3: Cluster Label 1
Sample 4: Cluster Label 4
Sample 5: Cluster Label 4
Sample 6: Cluster Label 5
Sample 7: Cluster Label 2
Sample 8: Cluster Label 5
Sample 9: Cluster Label 3
Sample 10: Cluster Label 4
Sample 11: Cluster Label 5
Sample 12: Cluster Label 4
Sample 13: Cluster Label 5
Sample 14: Cluster Label 5
Sample 15: Cluster Label 5
Sample 16: Cluster Label 3
Sample 17: Cluster Label 5
Sample 18: Cluster Label 5
Sample 19: Cluster Label 1
Sample 20: Cluster Label 1
Sample 21: Cluster Label 2
Sample 22: Cluster Label 2
Sample 23: Cluster Label 1
Sample 24: Cluster Label 1
Sample 25: Cluster Label 3
Sample 26: Cluster Label 5
Sample 27: Cluster Label 5
Sample 28: Cluster Label 4
Sample 29: Cluster Label 1
Sample 30: Cluster Label 2
Sample 31: Cluster Label 4
Sample 32: Cluster Label 1
Sample 33: Cluster Label 2
Sample 34: Cluster Label 2
Sample 35: Cluster Label 4
Sample 36: Cluster Label 4
Sample 37: Cluster Label 4
Sample 38: Cluster Label 2
Sample 39: Cluster Label 1
Sample 40: Cluster Label 5
Sample 41: Cluster Label 2
Sample 42: Cluster Label 2
Sample 43: Cluster Label 1
Sample 44: Cluster Label 4
```

```
Sample 45: Cluster Label 5
Sample 46: Cluster Label 1
Sample 47: Cluster Label 5
Sample 48: Cluster Label 1
Sample 49: Cluster Label 3
Sample 50: Cluster Label 2
Sample 51: Cluster Label 5
Sample 52: Cluster Label 2
Sample 53: Cluster Label 2
Sample 54: Cluster Label 2
Sample 55: Cluster Label 5
Sample 56: Cluster Label 2
Sample 57: Cluster Label 4
Sample 58: Cluster Label 2
Sample 59: Cluster Label 2
Sample 60: Cluster Label 5
Sample 61: Cluster Label 2
Sample 62: Cluster Label 2
Sample 63: Cluster Label 5
Sample 64: Cluster Label 5
Sample 65: Cluster Label 2
Sample 66: Cluster Label 2
Sample 67: Cluster Label 3
Sample 68: Cluster Label 4
Sample 69: Cluster Label 5
Sample 70: Cluster Label 5
Sample 71: Cluster Label 4
Sample 72: Cluster Label 2
Sample 73: Cluster Label 3
Sample 74: Cluster Label 2
Sample 75: Cluster Label 3
Sample 76: Cluster Label 2
Sample 77: Cluster Label 2
Sample 78: Cluster Label 3
Sample 79: Cluster Label 4
Sample 80: Cluster Label 3
Sample 81: Cluster Label 2
Sample 82: Cluster Label 2
Sample 83: Cluster Label 3
Sample 84: Cluster Label 1
Sample 85: Cluster Label 5
Sample 86: Cluster Label 4
Sample 87: Cluster Label 5
Sample 88: Cluster Label 1
Sample 89: Cluster Label 2
```

```
Sample 90: Cluster Label 3
Sample 91: Cluster Label 2
Sample 92: Cluster Label 2
Sample 93: Cluster Label 3
Sample 94: Cluster Label 3
Sample 95: Cluster Label 5
Sample 96: Cluster Label 2
Sample 97: Cluster Label 5
Sample 98: Cluster Label 2
Sample 99: Cluster Label 5
Sample 100: Cluster Label 5
Sample 101: Cluster Label 5
Sample 102: Cluster Label 2
Sample 103: Cluster Label 5
Sample 104: Cluster Label 4
Sample 105: Cluster Label 5
Sample 106: Cluster Label 3
Sample 107: Cluster Label 5
Sample 108: Cluster Label 5
Sample 109: Cluster Label 4
Sample 110: Cluster Label 2
Sample 111: Cluster Label 5
Sample 112: Cluster Label 5
Sample 113: Cluster Label 4
Sample 114: Cluster Label 4
Sample 115: Cluster Label 3
Sample 116: Cluster Label 5
Sample 117: Cluster Label 2
Sample 118: Cluster Label 3
Sample 119: Cluster Label 5
Sample 120: Cluster Label 4
Sample 121: Cluster Label 5
Sample 122: Cluster Label 5
Sample 123: Cluster Label 3
Sample 124: Cluster Label 5
Sample 125: Cluster Label 2
Sample 126: Cluster Label 5
Sample 127: Cluster Label 5
Sample 128: Cluster Label 5
Sample 129: Cluster Label 3
Sample 130: Cluster Label 4
Sample 131: Cluster Label 3
Sample 132: Cluster Label 5
Sample 133: Cluster Label 5
Sample 134: Cluster Label 1
```

```
Sample 135: Cluster Label 3
Sample 136: Cluster Label 3
Sample 137: Cluster Label 5
Sample 138: Cluster Label 5
Sample 139: Cluster Label 4
Sample 140: Cluster Label 4
Sample 141: Cluster Label 2
Sample 142: Cluster Label 5
Sample 143: Cluster Label 2
Sample 144: Cluster Label 3
Sample 145: Cluster Label 2
Sample 146: Cluster Label 2
Sample 147: Cluster Label 4
Sample 148: Cluster Label 5
Sample 149: Cluster Label 5
Sample 150: Cluster Label 1
Sample 151: Cluster Label 5
Sample 152: Cluster Label 1
Sample 153: Cluster Label 2
Sample 154: Cluster Label 5
Sample 155: Cluster Label 3
Sample 156: Cluster Label 5
Sample 157: Cluster Label 2
Sample 158: Cluster Label 5
Sample 159: Cluster Label 2
Sample 160: Cluster Label 2
Sample 161: Cluster Label 2
Sample 162: Cluster Label 5
Sample 163: Cluster Label 3
Sample 164: Cluster Label 5
Sample 165: Cluster Label 5
Sample 166: Cluster Label 5
Sample 167: Cluster Label 3
Sample 168: Cluster Label 3
Sample 169: Cluster Label 5
Sample 170: Cluster Label 4
Sample 171: Cluster Label 3
Sample 172: Cluster Label 5
Sample 173: Cluster Label 2
Sample 174: Cluster Label 2
Sample 175: Cluster Label 5
Sample 176: Cluster Label 5
Sample 177: Cluster Label 2
Sample 178: Cluster Label 2
Sample 179: Cluster Label 5
```

```
Sample 180: Cluster Label 3
Sample 181: Cluster Label 5
Sample 182: Cluster Label 5
Sample 183: Cluster Label 3
Sample 184: Cluster Label 5
Sample 185: Cluster Label 5
Sample 186: Cluster Label 4
Sample 187: Cluster Label 4
Sample 188: Cluster Label 5
Sample 189: Cluster Label 3
Sample 190: Cluster Label 4
Sample 191: Cluster Label 3
Sample 192: Cluster Label 1
Sample 193: Cluster Label 5
Sample 194: Cluster Label 5
Sample 195: Cluster Label 5
Sample 196: Cluster Label 5
Sample 197: Cluster Label 3
Sample 198: Cluster Label 3
Sample 199: Cluster Label 5
Sample 200: Cluster Label 5
Sample 201: Cluster Label 4
Sample 202: Cluster Label 2
Sample 203: Cluster Label 4
Sample 204: Cluster Label 2
Sample 205: Cluster Label 5
Sample 206: Cluster Label 1
Sample 207: Cluster Label 3
Sample 208: Cluster Label 5
Sample 209: Cluster Label 3
Sample 210: Cluster Label 5
Sample 211: Cluster Label 4
Sample 212: Cluster Label 5
Sample 213: Cluster Label 5
Sample 214: Cluster Label 3
Sample 215: Cluster Label 1
Sample 216: Cluster Label 5
Sample 217: Cluster Label 5
Sample 218: Cluster Label 2
Sample 219: Cluster Label 3
Sample 220: Cluster Label 4
Sample 221: Cluster Label 5
Sample 222: Cluster Label 2
Sample 223: Cluster Label 3
Sample 224: Cluster Label 3
```

```
Sample 225: Cluster Label 5
Sample 226: Cluster Label 5
Sample 227: Cluster Label 3
Sample 228: Cluster Label 3
Sample 229: Cluster Label 1
Sample 230: Cluster Label 2
Sample 231: Cluster Label 2
Sample 232: Cluster Label 2
Sample 233: Cluster Label 5
Sample 234: Cluster Label 2
Sample 235: Cluster Label 2
Sample 236: Cluster Label 2
Sample 237: Cluster Label 1
Sample 238: Cluster Label 2
Sample 239: Cluster Label 1
Sample 240: Cluster Label 3
Sample 241: Cluster Label 3
Sample 242: Cluster Label 4
Sample 243: Cluster Label 5
Sample 244: Cluster Label 2
Sample 245: Cluster Label 2
Sample 246: Cluster Label 2
Sample 247: Cluster Label 3
Sample 248: Cluster Label 3
Sample 249: Cluster Label 4
Sample 250: Cluster Label 1
Sample 251: Cluster Label 3
Sample 252: Cluster Label 1
Sample 253: Cluster Label 5
Sample 254: Cluster Label 5
Sample 255: Cluster Label 5
Sample 256: Cluster Label 5
Sample 257: Cluster Label 2
Sample 258: Cluster Label 5
Sample 259: Cluster Label 5
Sample 260: Cluster Label 3
Sample 261: Cluster Label 5
Sample 262: Cluster Label 3
Sample 263: Cluster Label 5
Sample 264: Cluster Label 3
Sample 265: Cluster Label 5
Sample 266: Cluster Label 5
Sample 267: Cluster Label 5
Sample 268: Cluster Label 5
Sample 269: Cluster Label 3
```

```
Sample 270: Cluster Label 5
Sample 271: Cluster Label 5
Sample 272: Cluster Label 3
Sample 273: Cluster Label 3
Sample 274: Cluster Label 5
Sample 275: Cluster Label 3
Sample 276: Cluster Label 3
Sample 277: Cluster Label 5
Sample 278: Cluster Label 5
Sample 279: Cluster Label 1
Sample 280: Cluster Label 5
Sample 281: Cluster Label 4
Sample 282: Cluster Label 4
Sample 283: Cluster Label 5
Sample 284: Cluster Label 5
Sample 285: Cluster Label 1
Sample 286: Cluster Label 2
Sample 287: Cluster Label 5
Sample 288: Cluster Label 5
Sample 289: Cluster Label 3
Sample 290: Cluster Label 5
Sample 291: Cluster Label 3
Sample 292: Cluster Label 5
Sample 293: Cluster Label 4
Sample 294: Cluster Label 4
Sample 295: Cluster Label 5
Sample 296: Cluster Label 2
Sample 297: Cluster Label 5
Sample 298: Cluster Label 5
Sample 299: Cluster Label 3
Sample 300: Cluster Label 1
Sample 301: Cluster Label 1
Sample 302: Cluster Label 4
Sample 303: Cluster Label 3
Sample 304: Cluster Label 3
Sample 305: Cluster Label 5
Sample 306: Cluster Label 3
Sample 307: Cluster Label 5
Sample 308: Cluster Label 5
Sample 309: Cluster Label 3
Sample 310: Cluster Label 3
Sample 311: Cluster Label 3
Sample 312: Cluster Label 5
Sample 313: Cluster Label 3
Sample 314: Cluster Label 5
```

```
Sample 315: Cluster Label 3
Sample 316: Cluster Label 5
Sample 317: Cluster Label 4
Sample 318: Cluster Label 3
Sample 319: Cluster Label 3
Sample 320: Cluster Label 2
Sample 321: Cluster Label 3
Sample 322: Cluster Label 5
Sample 323: Cluster Label 2
Sample 324: Cluster Label 5
Sample 325: Cluster Label 1
Sample 326: Cluster Label 5
Sample 327: Cluster Label 2
Sample 328: Cluster Label 3
Sample 329: Cluster Label 5
Sample 330: Cluster Label 5
Sample 331: Cluster Label 5
Sample 332: Cluster Label 5
Sample 333: Cluster Label 5
Sample 334: Cluster Label 1
Sample 335: Cluster Label 3
Sample 336: Cluster Label 5
Sample 337: Cluster Label 3
Sample 338: Cluster Label 3
Sample 339: Cluster Label 3
Sample 340: Cluster Label 3
Sample 341: Cluster Label 5
Sample 342: Cluster Label 4
Sample 343: Cluster Label 3
Sample 344: Cluster Label 5
Sample 345: Cluster Label 3
Sample 346: Cluster Label 3
Sample 347: Cluster Label 3
Sample 348: Cluster Label 5
Sample 349: Cluster Label 5
Sample 350: Cluster Label 5
Sample 351: Cluster Label 5
Sample 352: Cluster Label 3
Sample 353: Cluster Label 5
Sample 354: Cluster Label 3
Sample 355: Cluster Label 3
Sample 356: Cluster Label 3
Sample 357: Cluster Label 5
Sample 358: Cluster Label 3
Sample 359: Cluster Label 3
```

```
Sample 360: Cluster Label 3
Sample 361: Cluster Label 5
Sample 362: Cluster Label 3
Sample 363: Cluster Label 3
Sample 364: Cluster Label 5
Sample 365: Cluster Label 3
Sample 366: Cluster Label 3
Sample 367: Cluster Label 3
Sample 368: Cluster Label 3
Sample 369: Cluster Label 1
Sample 370: Cluster Label 5
Sample 371: Cluster Label 3
Sample 372: Cluster Label 5
Sample 373: Cluster Label 5
Sample 374: Cluster Label 2
Sample 375: Cluster Label 3
Sample 376: Cluster Label 3
Sample 377: Cluster Label 3
Sample 378: Cluster Label 5
Sample 379: Cluster Label 3
Sample 380: Cluster Label 1
Sample 381: Cluster Label 1
Sample 382: Cluster Label 3
Sample 383: Cluster Label 1
Sample 384: Cluster Label 3
Sample 385: Cluster Label 3
Sample 386: Cluster Label 2
Sample 387: Cluster Label 3
Sample 388: Cluster Label 3
Sample 389: Cluster Label 3
Sample 390: Cluster Label 3
Sample 391: Cluster Label 3
Sample 392: Cluster Label 3
Sample 393: Cluster Label 1
Sample 394: Cluster Label 3
Sample 395: Cluster Label 3
Sample 396: Cluster Label 3
Sample 397: Cluster Label 1
Sample 398: Cluster Label 1
Sample 399: Cluster Label 5
Sample 400: Cluster Label 1
Sample 401: Cluster Label 3
Sample 402: Cluster Label 5
Sample 403: Cluster Label 3
Sample 404: Cluster Label 1
```

```
Sample 405: Cluster Label 1
Sample 406: Cluster Label 1
Sample 407: Cluster Label 5
Sample 408: Cluster Label 3
Sample 409: Cluster Label 3
Sample 410: Cluster Label 5
Sample 411: Cluster Label 1
Sample 412: Cluster Label 3
Sample 413: Cluster Label 5
Sample 414: Cluster Label 3
Sample 415: Cluster Label 3
Sample 416: Cluster Label 3
Sample 417: Cluster Label 1
Sample 418: Cluster Label 5
Sample 419: Cluster Label 3
Sample 420: Cluster Label 5
Sample 421: Cluster Label 1
Sample 422: Cluster Label 1
Sample 423: Cluster Label 5
Sample 424: Cluster Label 3
Sample 425: Cluster Label 1
Sample 426: Cluster Label 1
Sample 427: Cluster Label 1
Sample 428: Cluster Label 3
Sample 429: Cluster Label 5
Sample 430: Cluster Label 5
Sample 431: Cluster Label 4
Sample 432: Cluster Label 5
Sample 433: Cluster Label 3
Sample 434: Cluster Label 3
Sample 435: Cluster Label 5
Sample 436: Cluster Label 3
Sample 437: Cluster Label 3
Sample 438: Cluster Label 5
Sample 439: Cluster Label 3
Sample 440: Cluster Label 1
Sample 441: Cluster Label 3
Sample 442: Cluster Label 5
Sample 443: Cluster Label 3
Sample 444: Cluster Label 3
Sample 445: Cluster Label 3
Sample 446: Cluster Label 5
Sample 447: Cluster Label 3
Sample 448: Cluster Label 5
Sample 449: Cluster Label 1
```

```
Sample 450: Cluster Label 3
Sample 451: Cluster Label 5
Sample 452: Cluster Label 5
Sample 453: Cluster Label 5
Sample 454: Cluster Label 5
Sample 455: Cluster Label 5
Sample 456: Cluster Label 3
Sample 457: Cluster Label 3
Sample 458: Cluster Label 3
Sample 459: Cluster Label 3
Sample 460: Cluster Label 3
Sample 461: Cluster Label 3
Sample 462: Cluster Label 3
Sample 463: Cluster Label 5
Sample 464: Cluster Label 3
Sample 465: Cluster Label 2
Sample 466: Cluster Label 3
Sample 467: Cluster Label 5
Sample 468: Cluster Label 1
Sample 469: Cluster Label 3
Sample 470: Cluster Label 3
Sample 471: Cluster Label 1
Sample 472: Cluster Label 3
Sample 473: Cluster Label 3
Sample 474: Cluster Label 5
Sample 475: Cluster Label 5
Sample 476: Cluster Label 3
Sample 477: Cluster Label 3
Sample 478: Cluster Label 5
Sample 479: Cluster Label 3
Sample 480: Cluster Label 1
Sample 481: Cluster Label 3
Sample 482: Cluster Label 3
Sample 483: Cluster Label 3
Sample 484: Cluster Label 3
Sample 485: Cluster Label 5
Sample 486: Cluster Label 1
Sample 487: Cluster Label 3
Sample 488: Cluster Label 3
Sample 489: Cluster Label 3
Sample 490: Cluster Label 5
Sample 491: Cluster Label 1
Sample 492: Cluster Label 1
Sample 493: Cluster Label 5
Sample 494: Cluster Label 3
```

```
Sample 495: Cluster Label 3
Sample 496: Cluster Label 3
Sample 497: Cluster Label 3
Sample 498: Cluster Label 5
Sample 499: Cluster Label 3
Sample 500: Cluster Label 1
Sample 501: Cluster Label 3
Sample 502: Cluster Label 5
Sample 503: Cluster Label 3
Sample 504: Cluster Label 3
Sample 505: Cluster Label 3
Sample 506: Cluster Label 1
Sample 507: Cluster Label 1
Sample 508: Cluster Label 3
Sample 509: Cluster Label 3
Sample 510: Cluster Label 1
Sample 511: Cluster Label 4
Sample 512: Cluster Label 5
Sample 513: Cluster Label 3
Sample 514: Cluster Label 3
Sample 515: Cluster Label 5
Sample 516: Cluster Label 3
Sample 517: Cluster Label 3
Sample 518: Cluster Label 5
Sample 519: Cluster Label 1
Sample 520: Cluster Label 5
Sample 521: Cluster Label 5
Sample 522: Cluster Label 5
Sample 523: Cluster Label 4
Sample 524: Cluster Label 5
Sample 525: Cluster Label 5
Sample 526: Cluster Label 3
Sample 527: Cluster Label 5
Sample 528: Cluster Label 3
Sample 529: Cluster Label 5
Sample 530: Cluster Label 3
Sample 531: Cluster Label 2
Sample 532: Cluster Label 3
Sample 533: Cluster Label 5
Sample 534: Cluster Label 3
Sample 535: Cluster Label 1
Sample 536: Cluster Label 1
Sample 537: Cluster Label 3
Sample 538: Cluster Label 5
Sample 539: Cluster Label 5
```

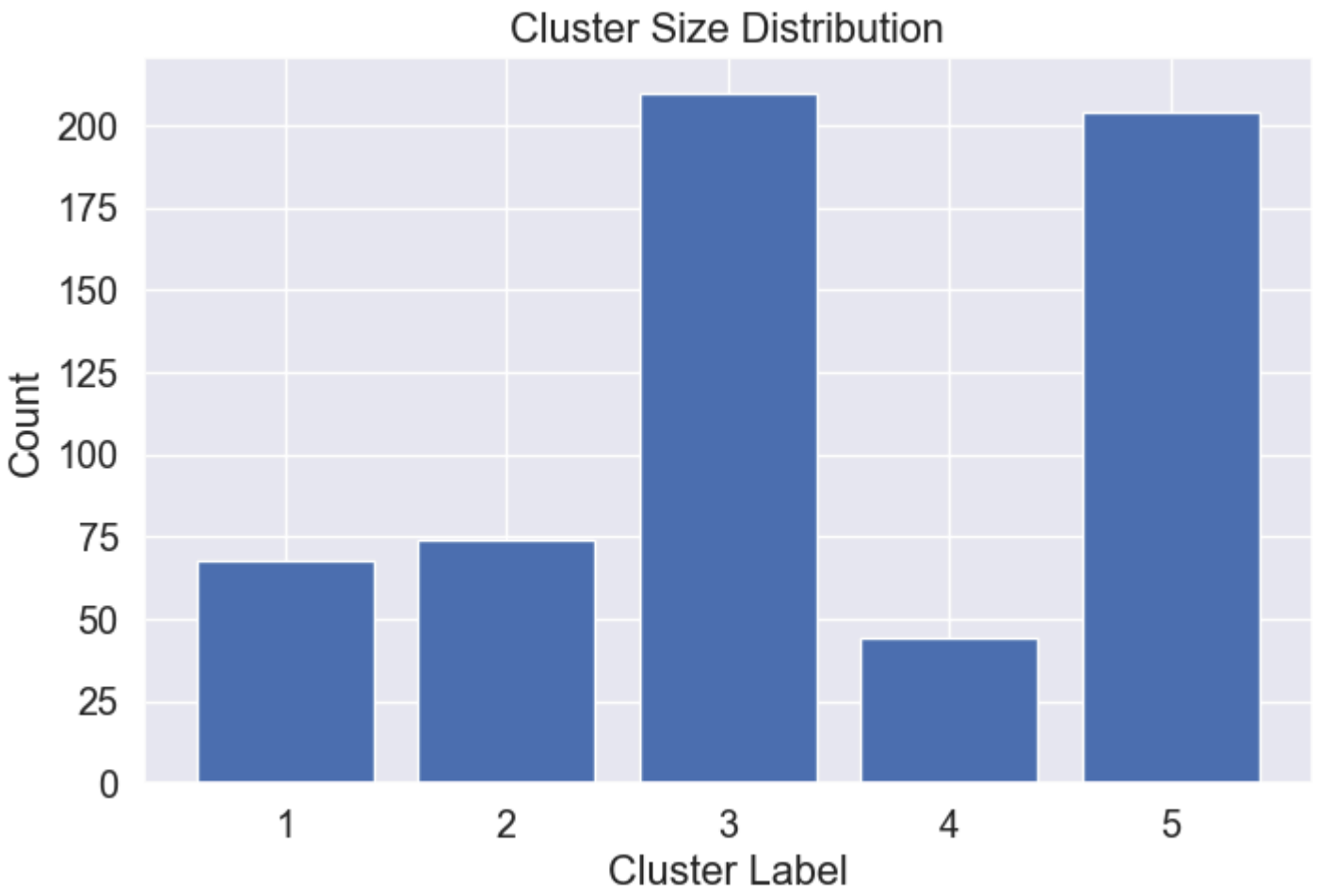
```
Sample 540: Cluster Label 3
Sample 541: Cluster Label 3
Sample 542: Cluster Label 3
Sample 543: Cluster Label 3
Sample 544: Cluster Label 3
Sample 545: Cluster Label 5
Sample 546: Cluster Label 5
Sample 547: Cluster Label 5
Sample 548: Cluster Label 3
Sample 549: Cluster Label 3
Sample 550: Cluster Label 5
Sample 551: Cluster Label 3
Sample 552: Cluster Label 1
Sample 553: Cluster Label 3
Sample 554: Cluster Label 1
Sample 555: Cluster Label 1
Sample 556: Cluster Label 5
Sample 557: Cluster Label 3
Sample 558: Cluster Label 3
Sample 559: Cluster Label 3
Sample 560: Cluster Label 3
Sample 561: Cluster Label 5
Sample 562: Cluster Label 3
Sample 563: Cluster Label 3
Sample 564: Cluster Label 3
Sample 565: Cluster Label 3
Sample 566: Cluster Label 3
Sample 567: Cluster Label 3
Sample 568: Cluster Label 5
Sample 569: Cluster Label 3
Sample 570: Cluster Label 1
Sample 571: Cluster Label 5
Sample 572: Cluster Label 3
Sample 573: Cluster Label 3
Sample 574: Cluster Label 3
Sample 575: Cluster Label 2
Sample 576: Cluster Label 1
Sample 577: Cluster Label 3
Sample 578: Cluster Label 3
Sample 579: Cluster Label 3
Sample 580: Cluster Label 3
Sample 581: Cluster Label 3
Sample 582: Cluster Label 5
Sample 583: Cluster Label 3
Sample 584: Cluster Label 5
```

```
Sample 585: Cluster Label 3
Sample 586: Cluster Label 5
Sample 587: Cluster Label 3
Sample 588: Cluster Label 3
Sample 589: Cluster Label 3
Sample 590: Cluster Label 5
Sample 591: Cluster Label 5
Sample 592: Cluster Label 3
Sample 593: Cluster Label 5
Sample 594: Cluster Label 5
Sample 595: Cluster Label 5
Sample 596: Cluster Label 3
Sample 597: Cluster Label 5
Sample 598: Cluster Label 3
Sample 599: Cluster Label 5
```

In [31]:

```python
# Create a DataFrame with sample indices and cluster labels
data = {'Sample Index': range(len(cluster_labels)), 'Cluster Label': cluster_labels}
df = pd.DataFrame(data)

# Calculate summary statistics for each cluster
cluster_stats = df.groupby('Cluster Label').agg({'Sample Index': 'count'}).reset_index()
cluster_stats.rename(columns={'Sample Index': 'Count'}, inplace=True)

# Display summary statistics
print(cluster_stats)

# Create a visualization of cluster statistics
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
plt.bar(cluster_stats['Cluster Label'], cluster_stats['Count'])
plt.xlabel('Cluster Label')
plt.ylabel('Count')
plt.title('Cluster Size Distribution')
plt.show()
```

```
   Cluster Label  Count
0              1     68
1              2     74
2              3    210
3              4     44
4              5    204
```



In [32]:
```python
# Create df DataFrame for the use of summary statistics
df = pd.DataFrame(data)

# Group data by cluster label (use the correct column name)
cluster_groups = df.groupby('Cluster Label')

# Iterate through clusters and analyze variables
for cluster_label, cluster_data in cluster_groups:
    print(f"Cluster {cluster_label}:")

    # Get the column names within this cluster
    cluster_columns = cluster_data.columns

    # Print the column names
```

```
    print("Columns within this cluster:")
    for column in cluster_columns:
        print(column)

    # Calculate summary statistics for each variable within the cluster
    summary_stats = cluster_data.describe()
    print(summary_stats)
```

```
Cluster 1:
Columns within this cluster:
Sample Index
Cluster Label
       Sample Index  Cluster Label
count     68.000000           68.0
mean     324.117647            1.0
std      176.455781            0.0
min        3.000000            1.0
25%      202.500000            1.0
50%      388.000000            1.0
75%      468.750000            1.0
max      576.000000            1.0
Cluster 2:
Columns within this cluster:
Sample Index
Cluster Label
       Sample Index  Cluster Label
count     74.000000           74.0
mean     158.270270            2.0
std      120.650213            0.0
min        2.000000            2.0
25%       62.750000            2.0
50%      142.000000            2.0
75%      231.750000            2.0
max      575.000000            2.0
Cluster 3:
Columns within this cluster:
Sample Index
Cluster Label
       Sample Index  Cluster Label
count    210.000000          210.0
mean     378.304762            3.0
std      150.467700            0.0
min        9.000000            3.0
25%      275.250000            3.0
50%      393.000000            3.0
75%      498.500000            3.0
max      598.000000            3.0
Cluster 4:
Columns within this cluster:
Sample Index
Cluster Label
       Sample Index  Cluster Label
count     44.000000           44.0
```

```
mean     167.136364            4.0
std      130.747154            0.0
min        4.000000            4.0
25%       65.250000            4.0
50%      139.500000            4.0
75%      243.750000            4.0
max      523.000000            4.0
Cluster 5:
Columns within this cluster:
Sample Index
Cluster Label
       Sample Index  Cluster Label
count    204.000000          204.0
mean     289.950980            5.0
std      168.642735            0.0
min        0.000000            5.0
25%      153.250000            5.0
50%      277.500000            5.0
75%      432.750000            5.0
max      599.000000            5.0
```

In [33]:
```
df
```

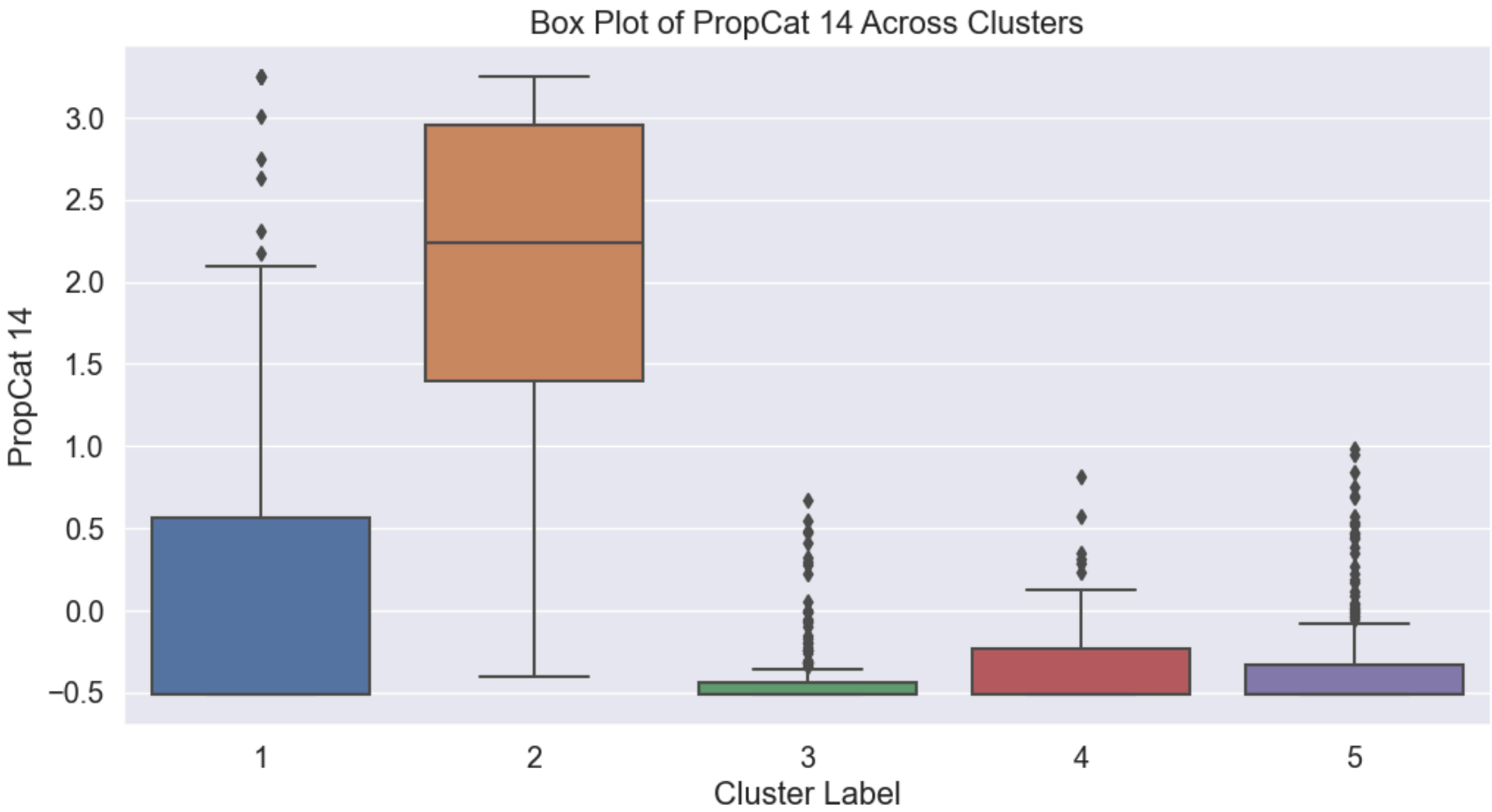
Out[33]:

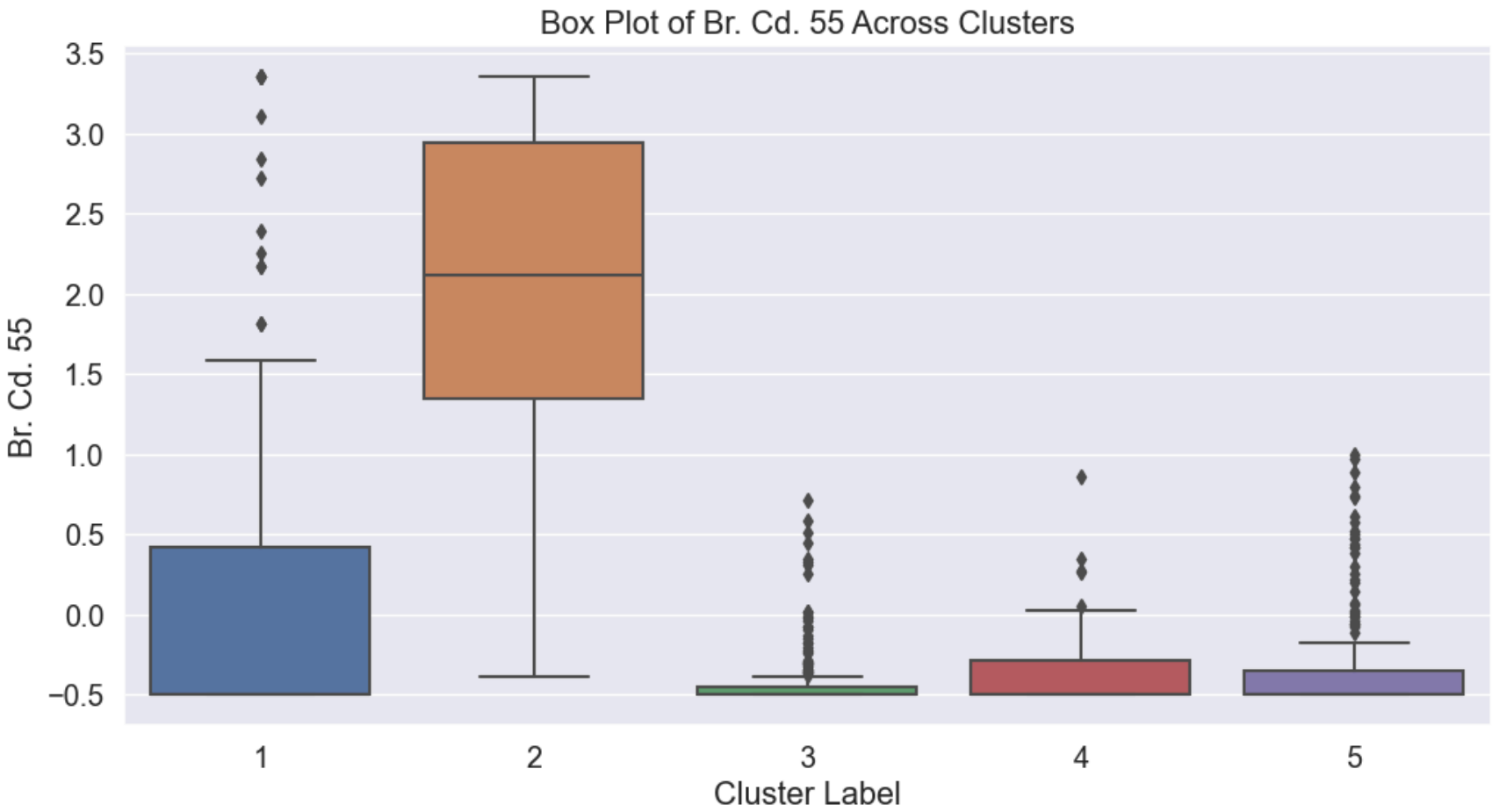
	Sample Index	Cluster Label
0	0	5
1	1	5
2	2	2
3	3	1
4	4	4
...	...	...
595	595	5
596	596	3
597	597	5
598	598	3
599	599	5

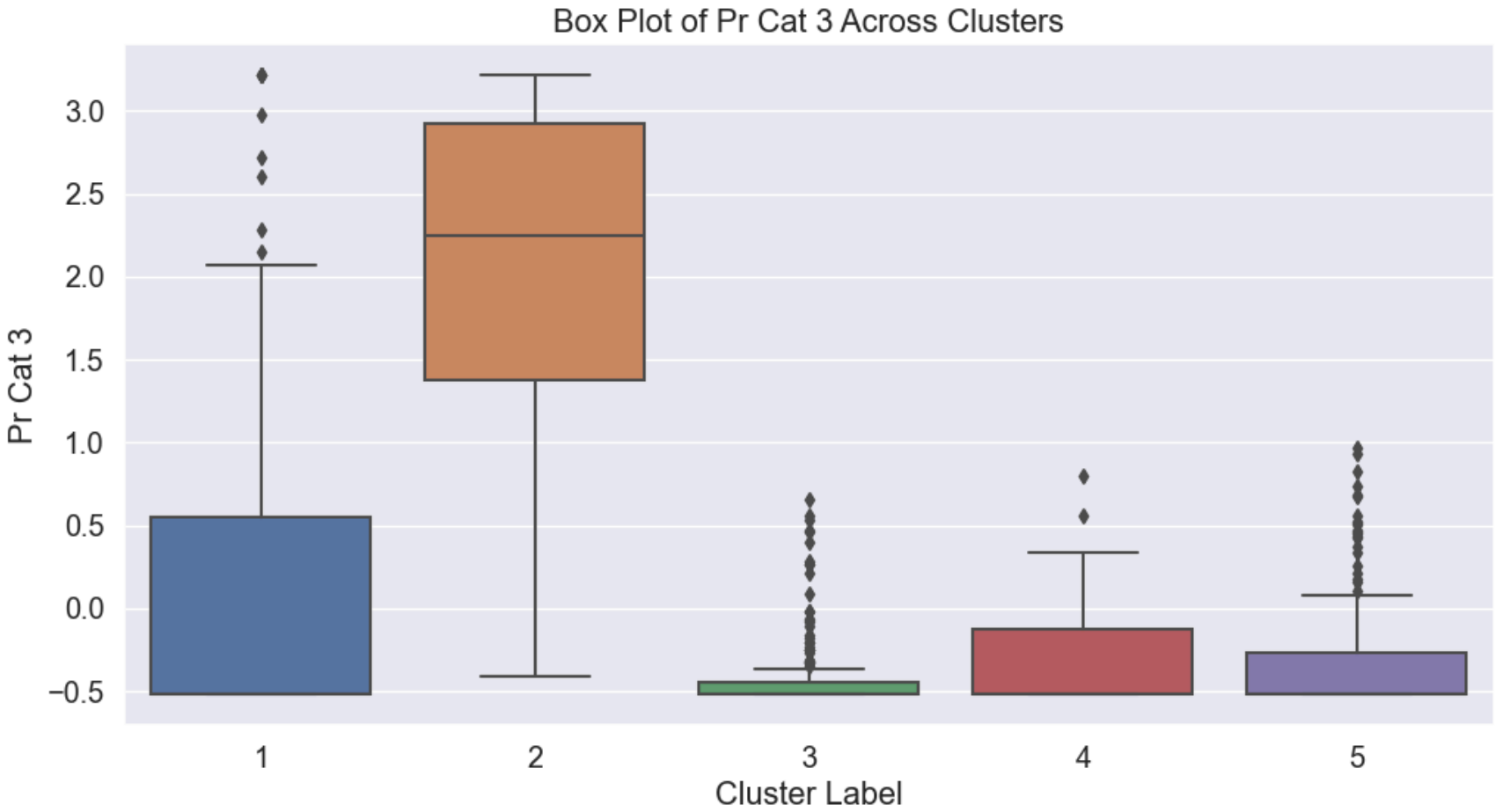
600 rows × 2 columns

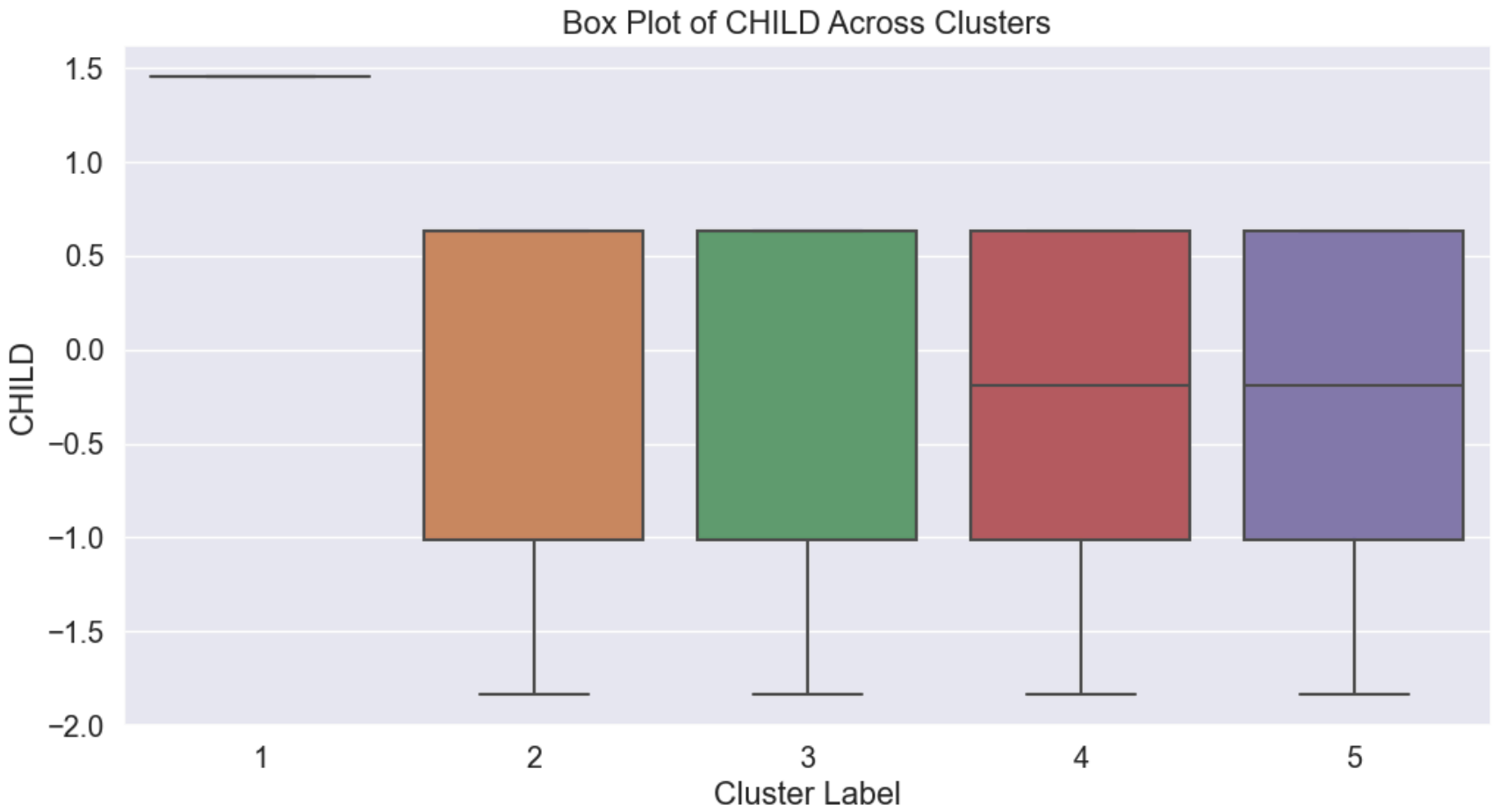
In [34]:

```
# Assign labels
filtered_df['Cluster Label'] = cluster_labels

# List of columns to visualize
columns_to_visualize = ['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'CHILD', 'Trans / Brand Runs', 'SEC', 'Vol/Tran', 'Pur '

# Loop through each column and create visualizations for each cluster
for column in columns_to_visualize:
    plt.figure(figsize=(12, 6))
    sns.boxplot(data=filtered_df, x='Cluster Label', y=column)
    plt.title(f'Box Plot of {column} Across Clusters')
    plt.xlabel('Cluster Label')
    plt.ylabel(column)
    plt.show()
```

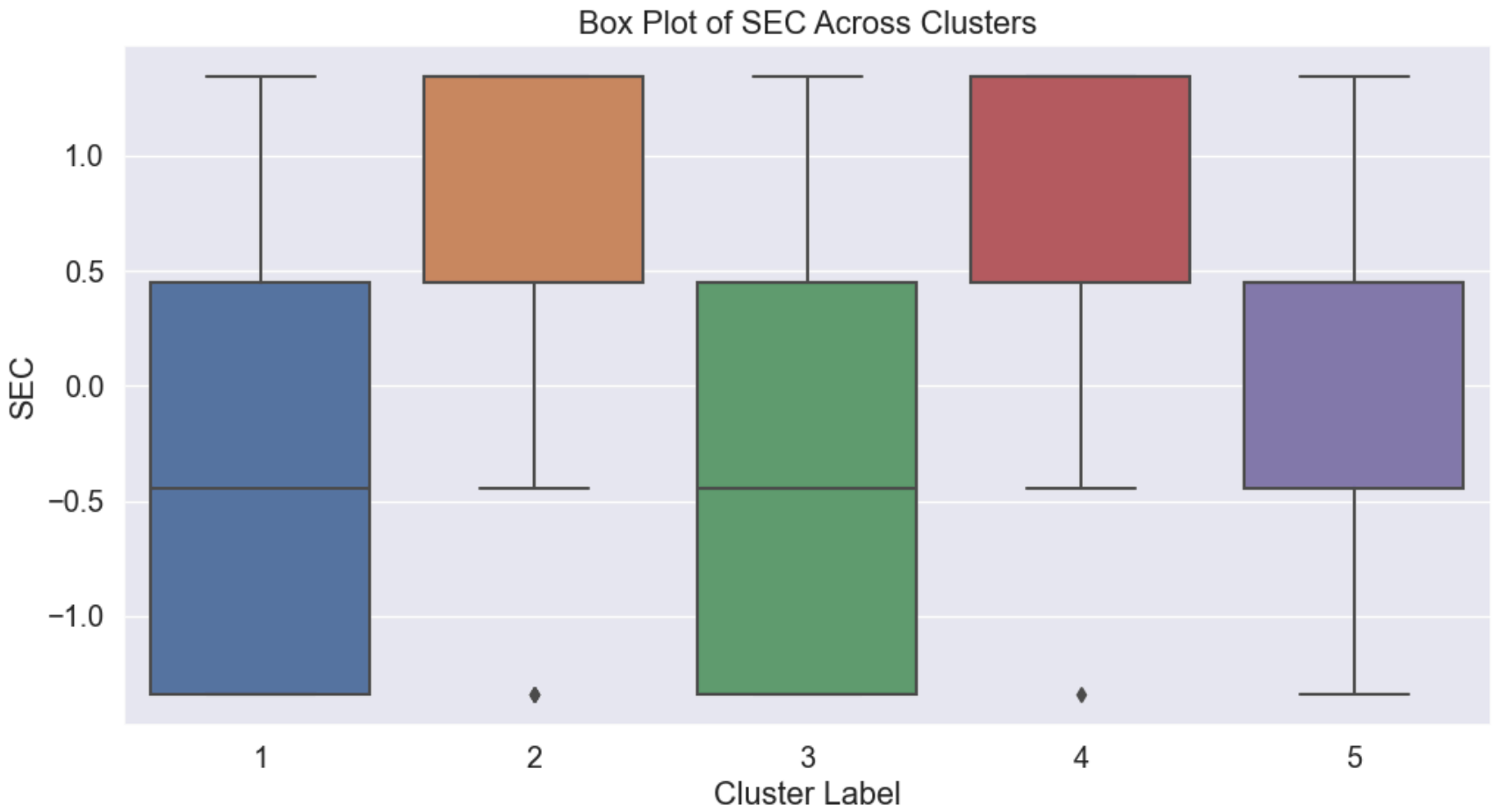

Box Plot of PropCat 14 Across Clusters
PropCat 14
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
Cluster Label

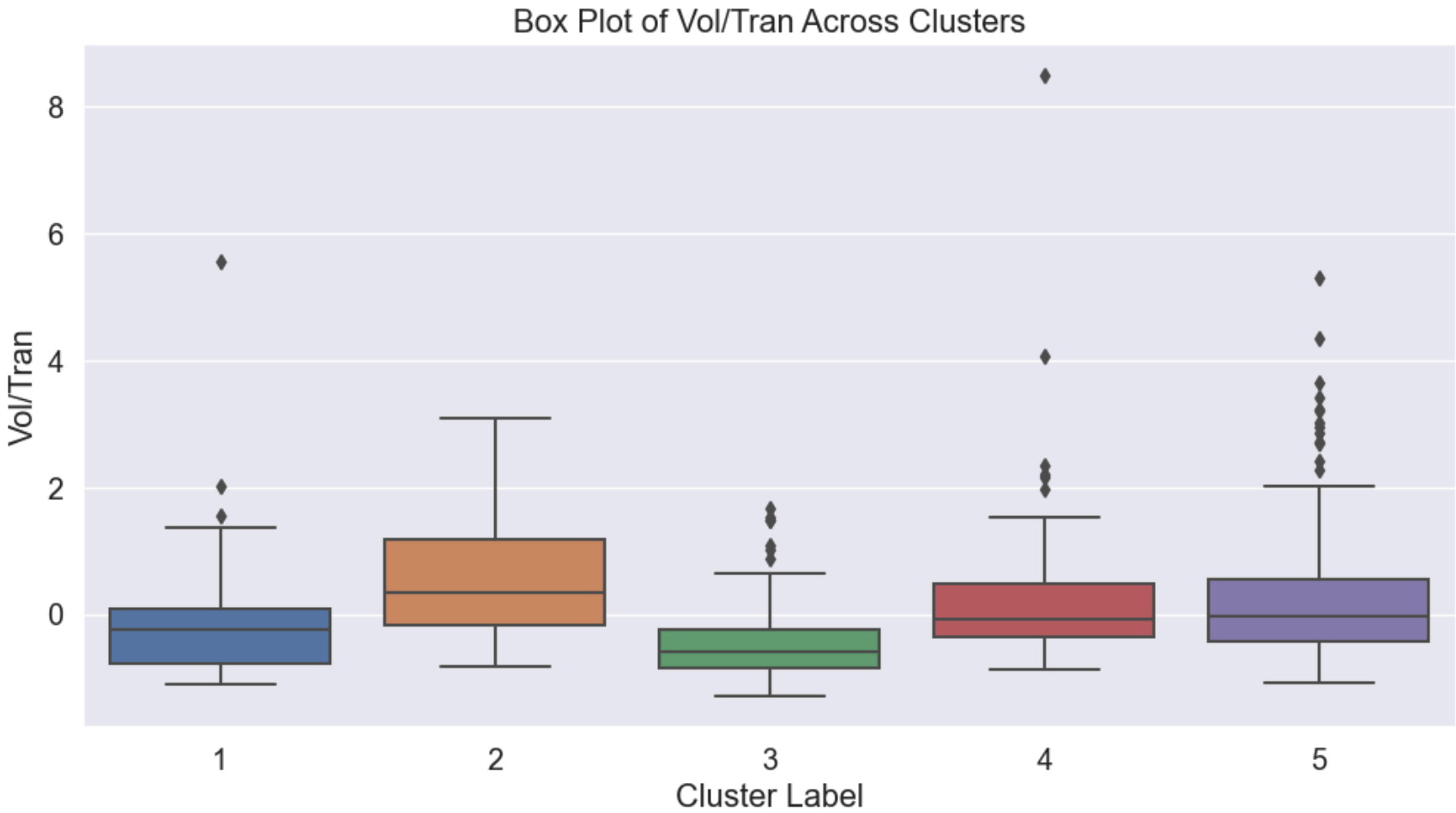

Box Plot of Br. Cd. 55 Across Clusters
Br. Cd. 55
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
Cluster Label

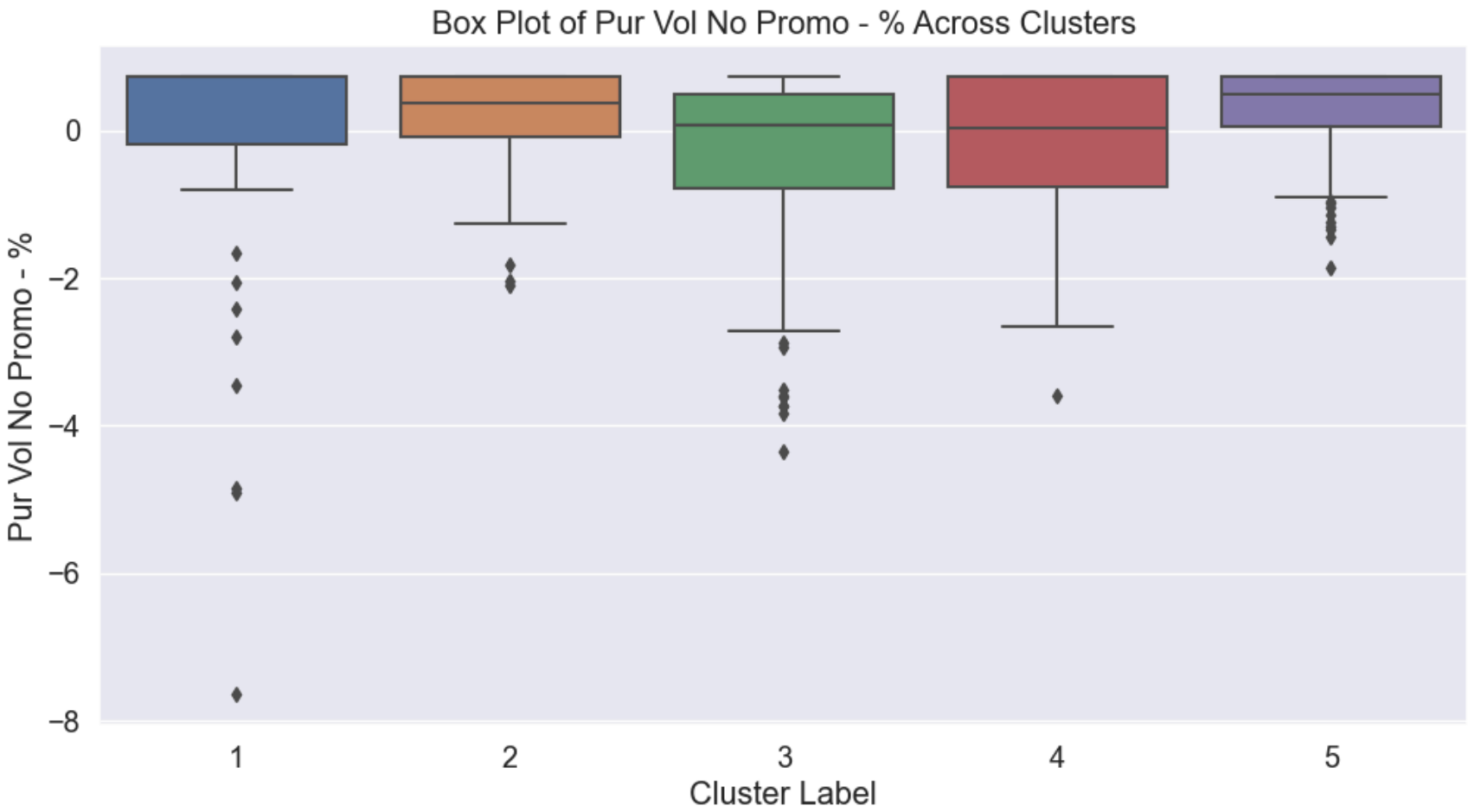

Box Plot of Pr Cat 3 Across Clusters
Pr Cat 3
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
1
2
3
4
5
Cluster Label

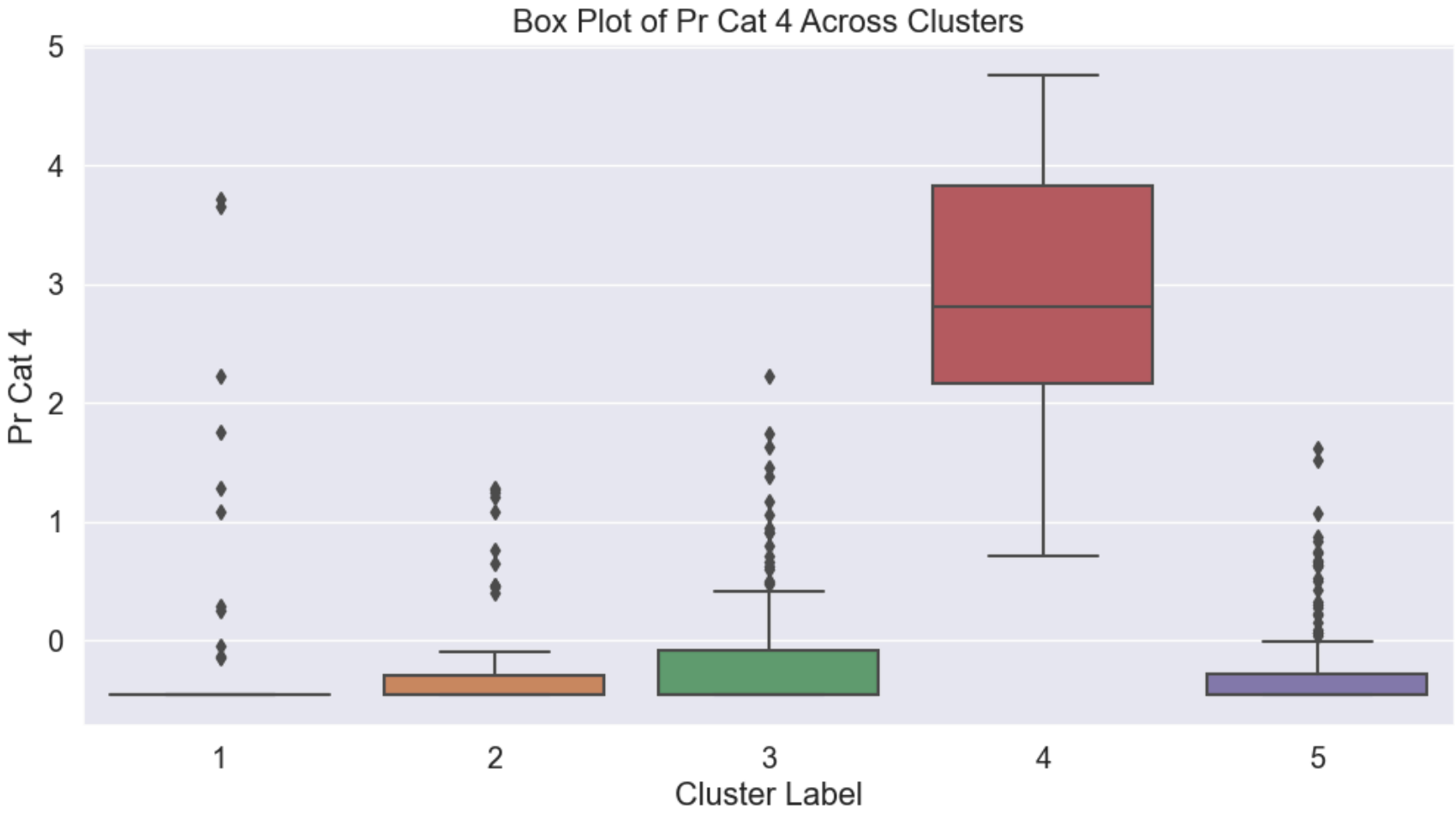

Box Plot of CHILD Across Clusters
CHILD
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
1
2
3
4
5
Cluster Label

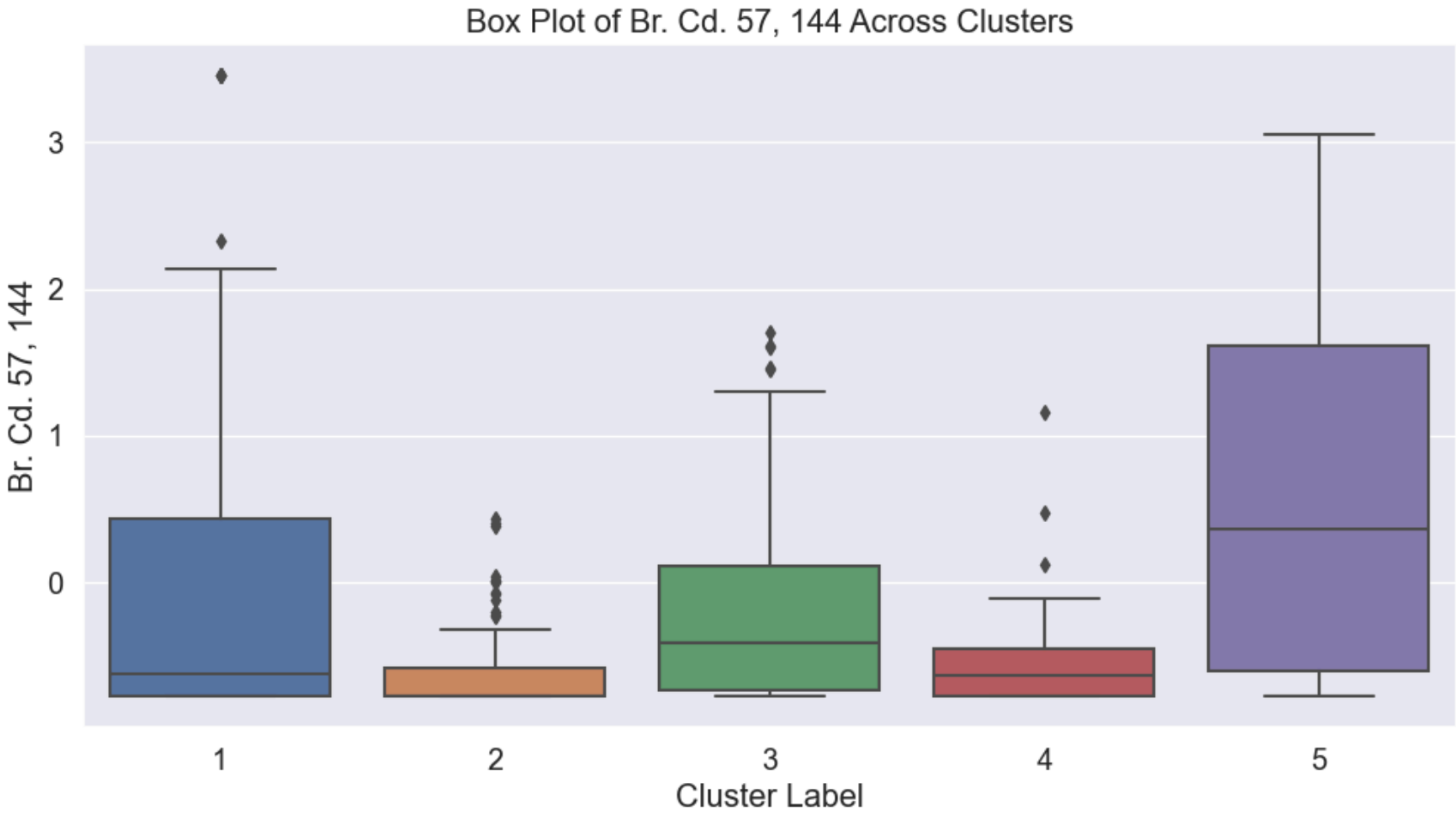

Box Plot of Trans / Brand Runs Across Clusters
Trans / Brand Runs
8
6
4
2
0
1
2
3
4
5
Cluster Label

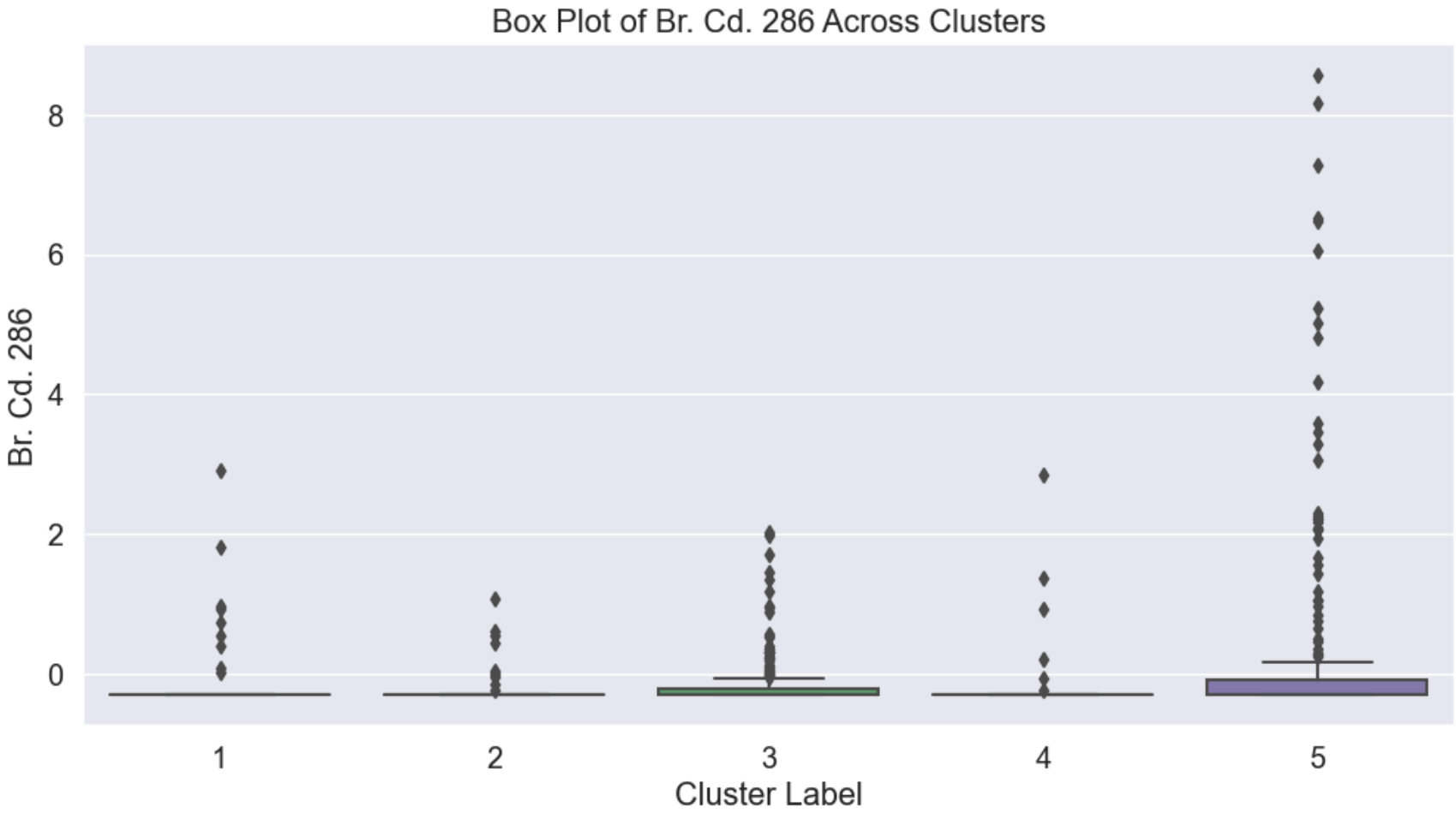

Box Plot of SEC Across Clusters
SEC
1.0
0.5
0.0
−0.5
−1.0
1
2
3
4
5
Cluster Label

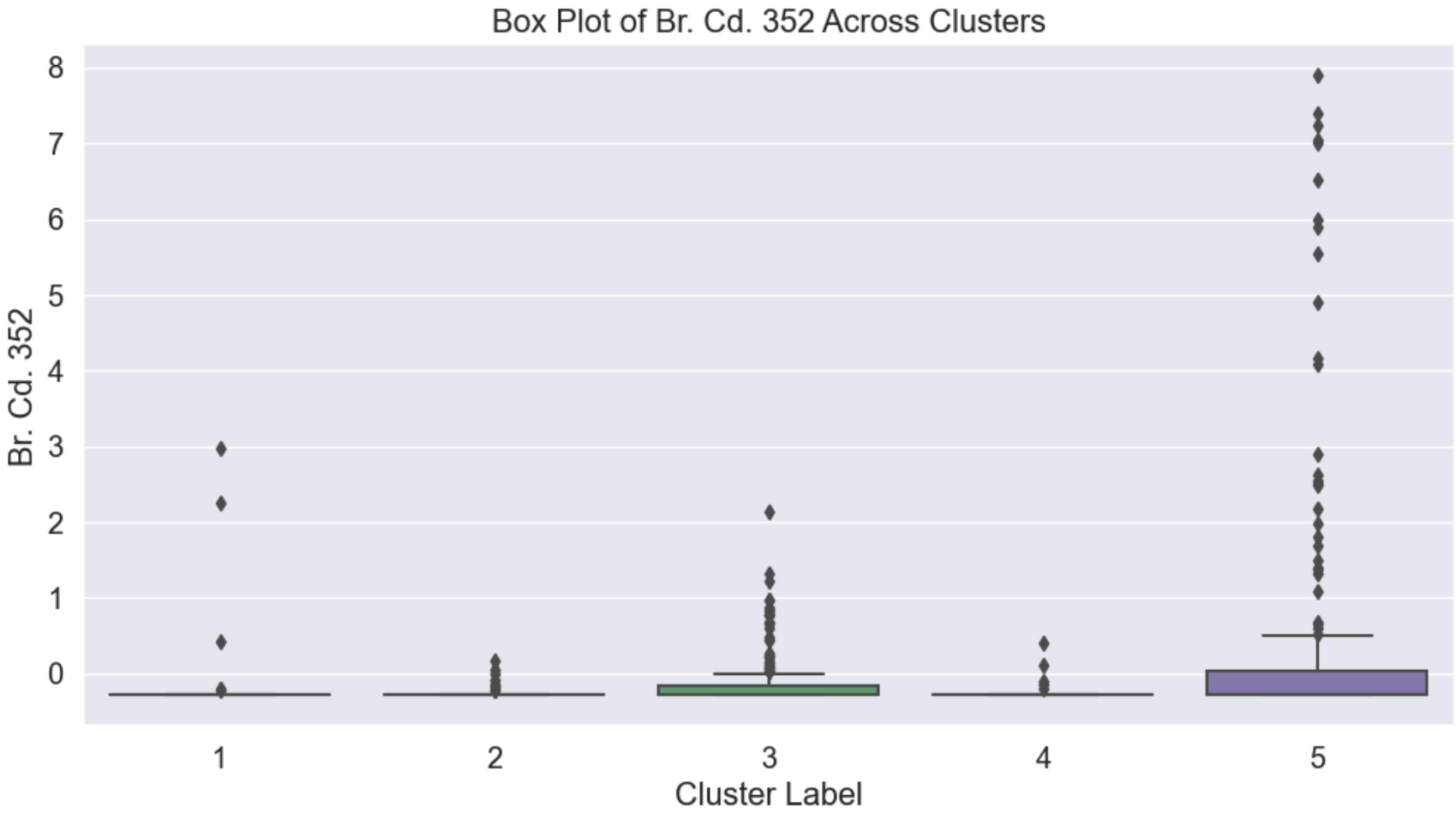

Box Plot of Vol/Tran Across Clusters
Vol/Tran
8
6
4
2
0
1
2
3
4
5
Cluster Label

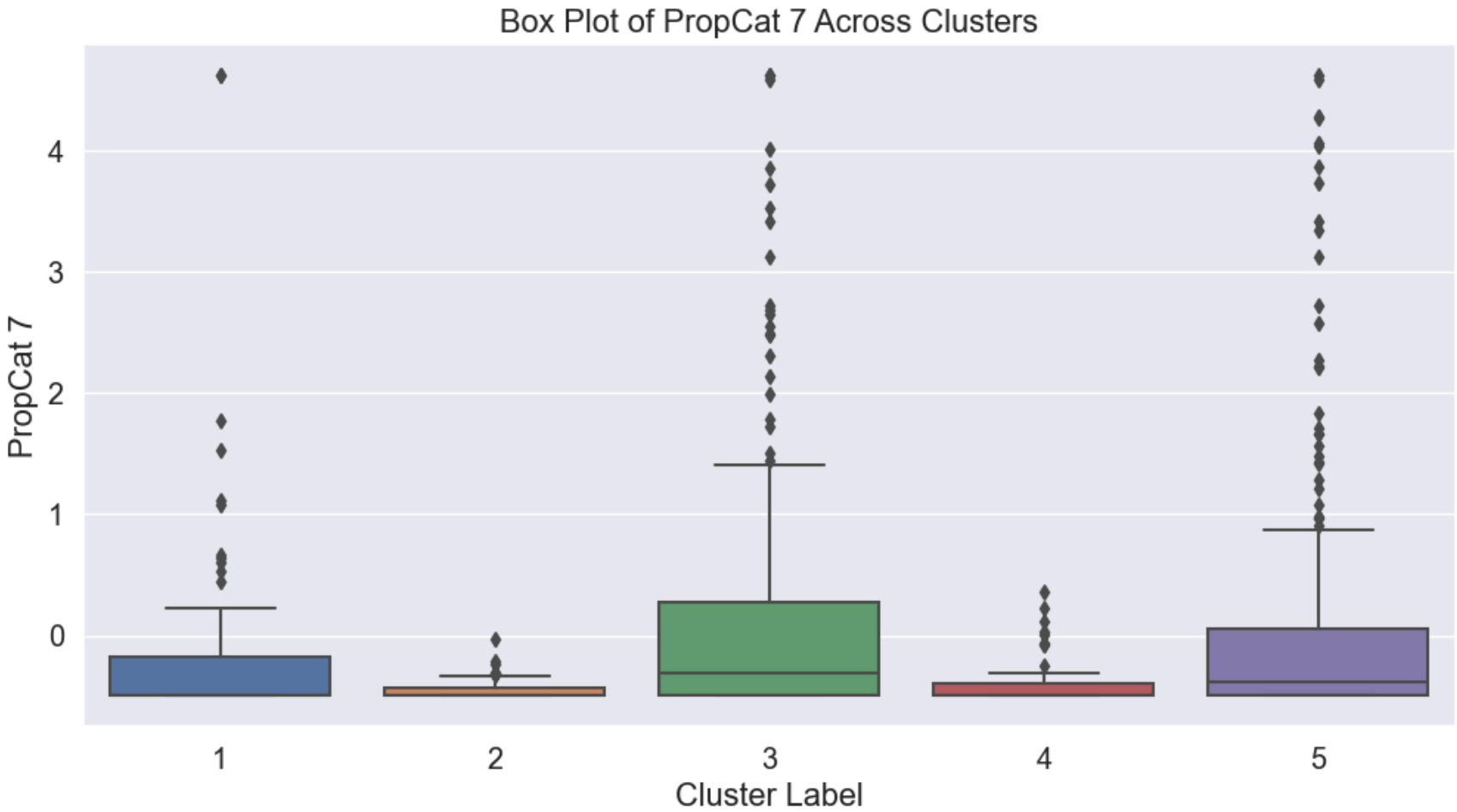

Box Plot of Pur Vol No Promo - % Across Clusters
Pur Vol No Promo - %
0
−2
−4
−6
−8
1
2
3
4
5
Cluster Label

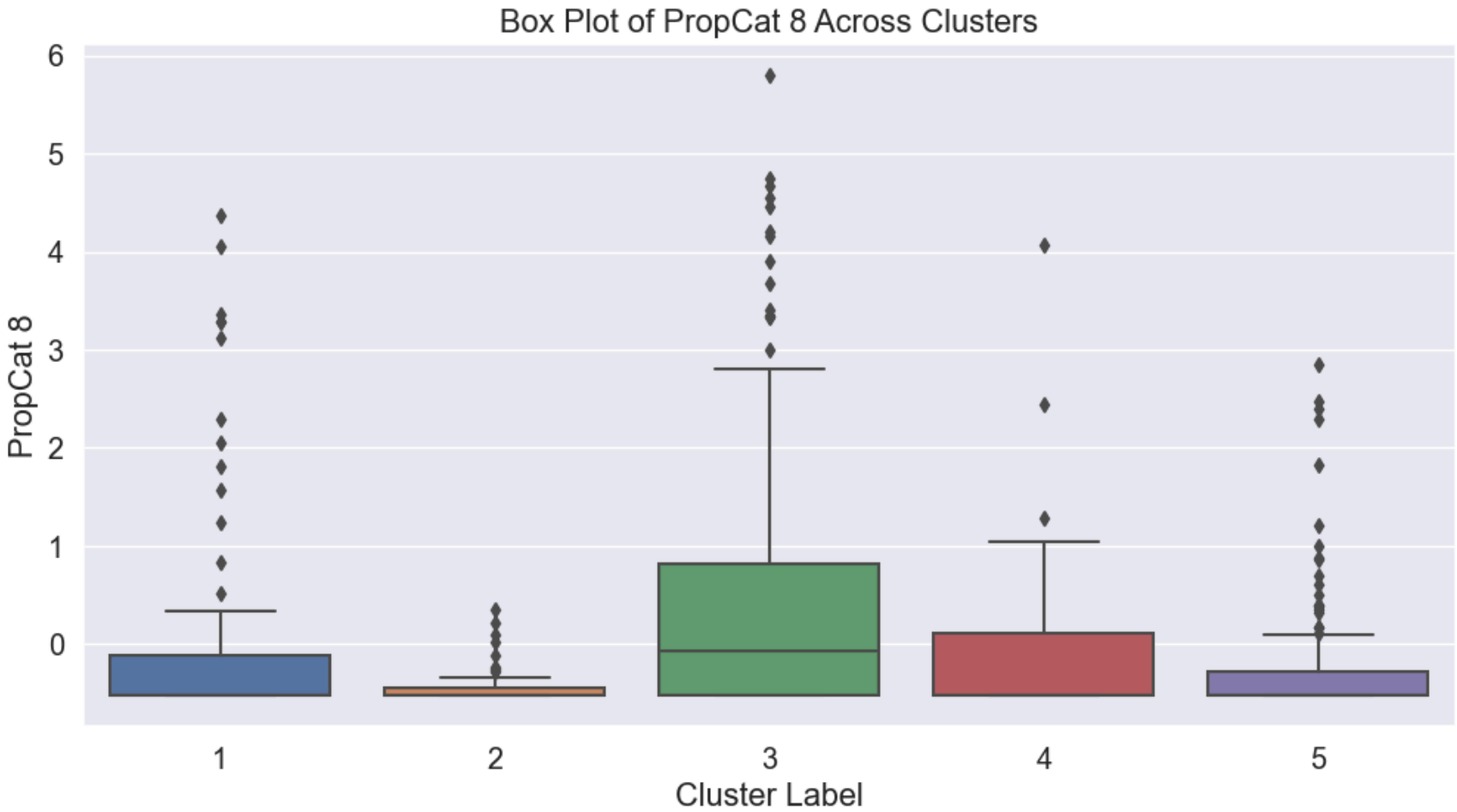

Box Plot of Pr Cat 4 Across Clusters
Pr Cat 4
5
4
3
2
1
0
1
2
3
4
5
Cluster Label

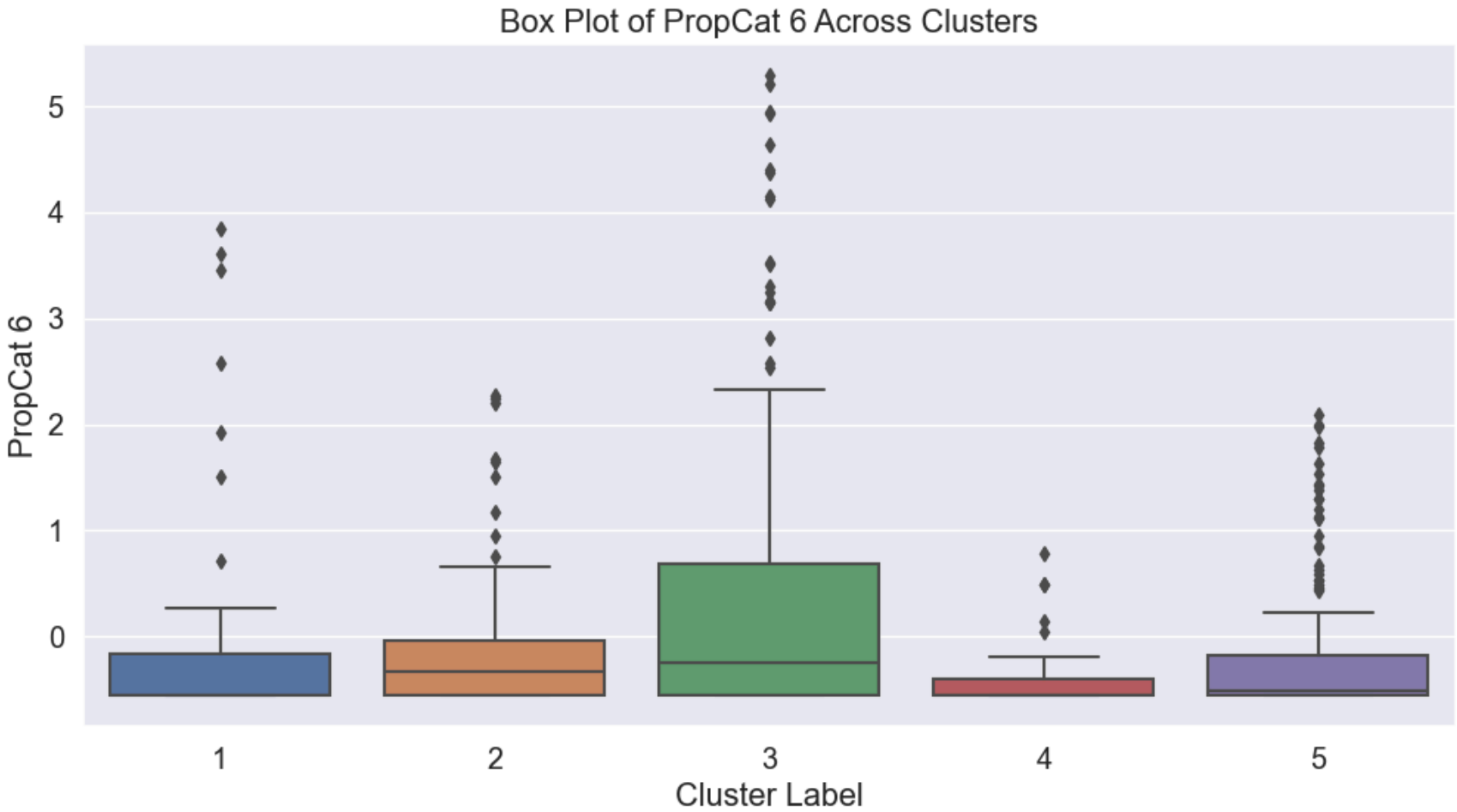

Box Plot of Br. Cd. 57, 144 Across Clusters
Br. Cd. 57, 144
3
2
1
0
1
2
3
4
5
Cluster Label

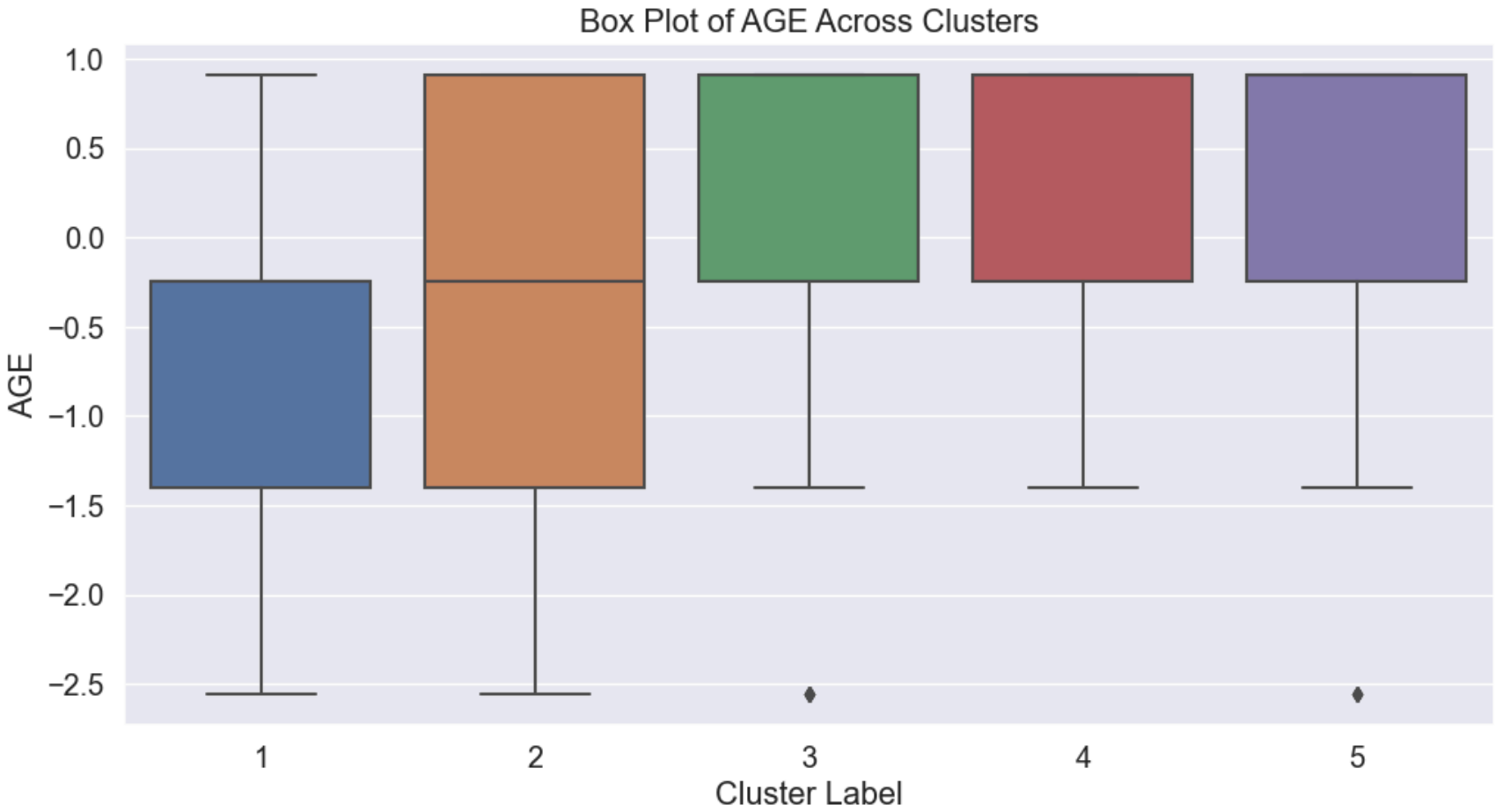

Box Plot of Br. Cd. 286 Across Clusters
Br. Cd. 286
8
6
4
2
0
1
2
3
4
5
Cluster Label

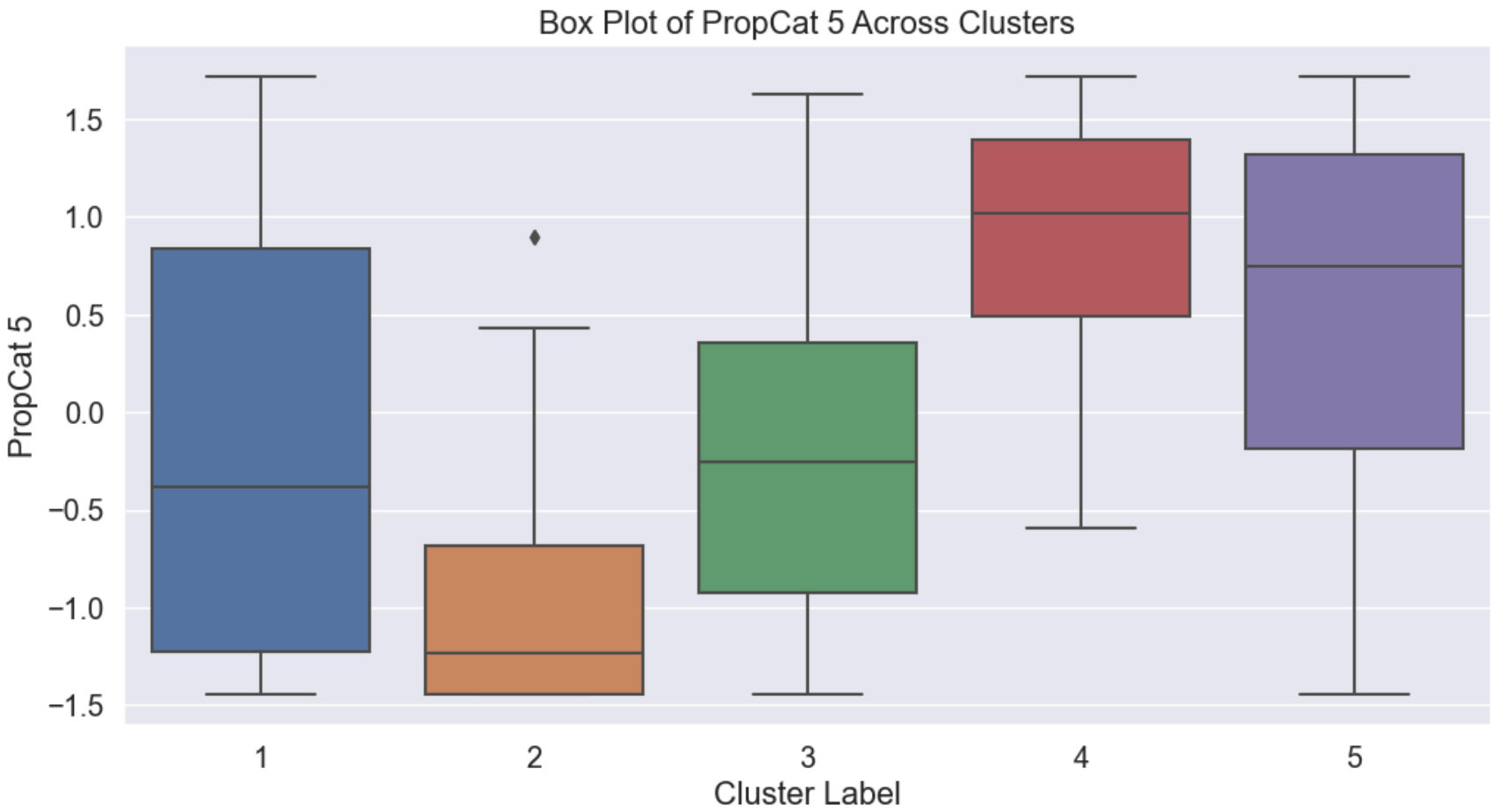

Box Plot of Br. Cd. 352 Across Clusters
Br. Cd. 352
8
7
6
5
4
3
2
1
0
1
2
3
4
5
Cluster Label

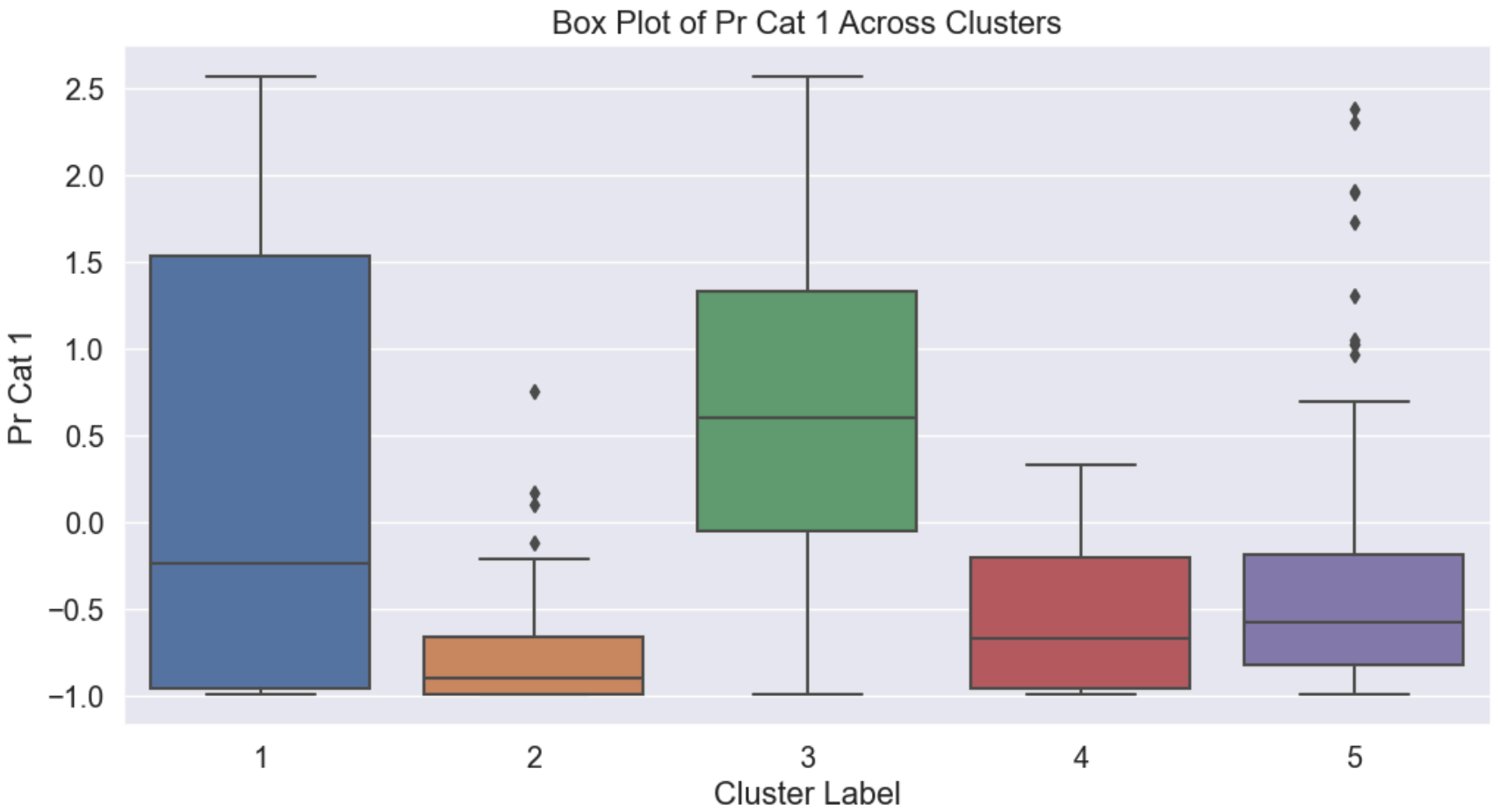

Box Plot of PropCat 7 Across Clusters
PropCat 7
4
3
2
1
0
1
2
3
4
5
Cluster Label

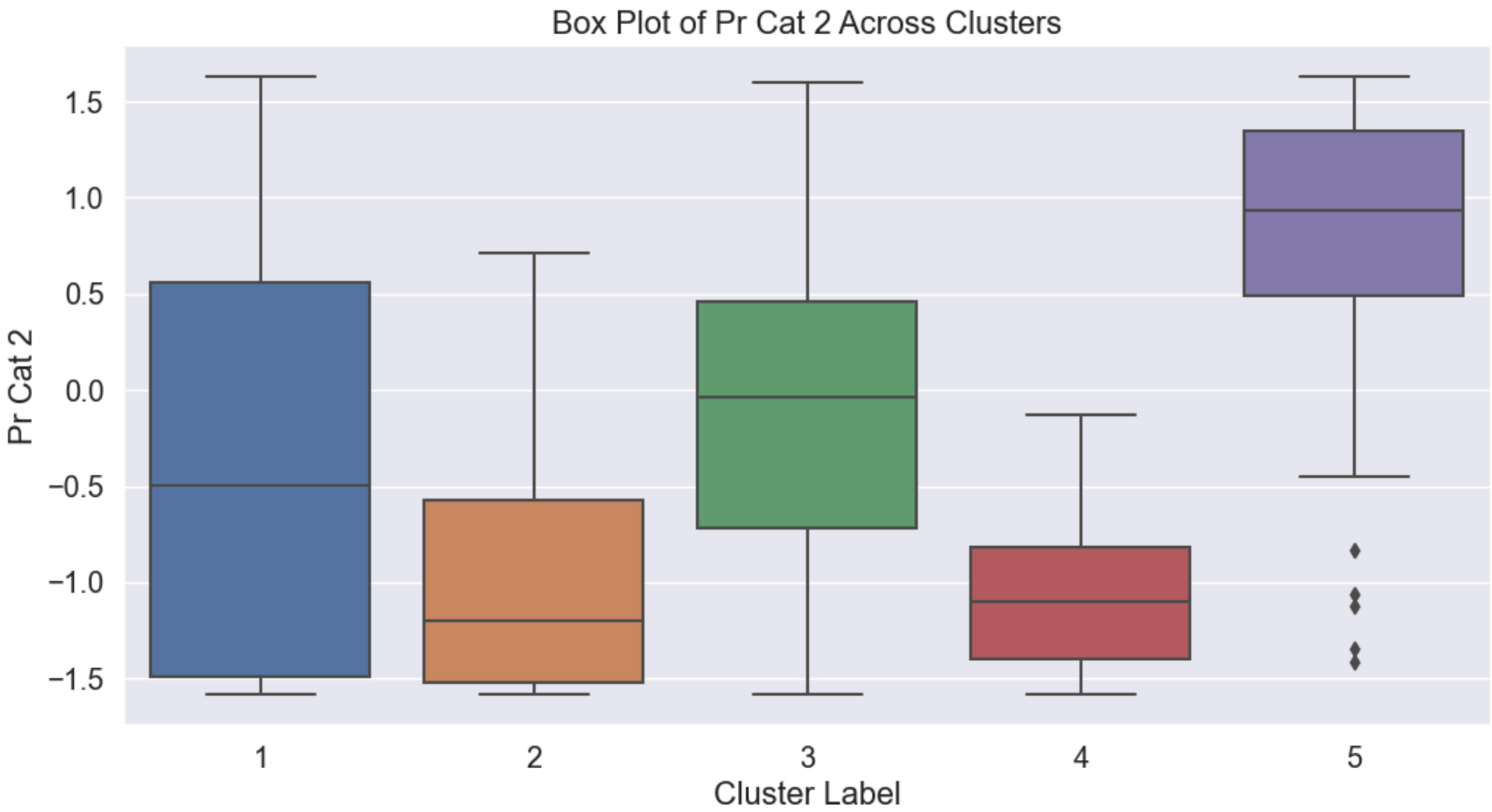

Box Plot of PropCat 8 Across Clusters
PropCat 8
6
5
4
3
2
1
0
1
2
3
4
5
Cluster Label

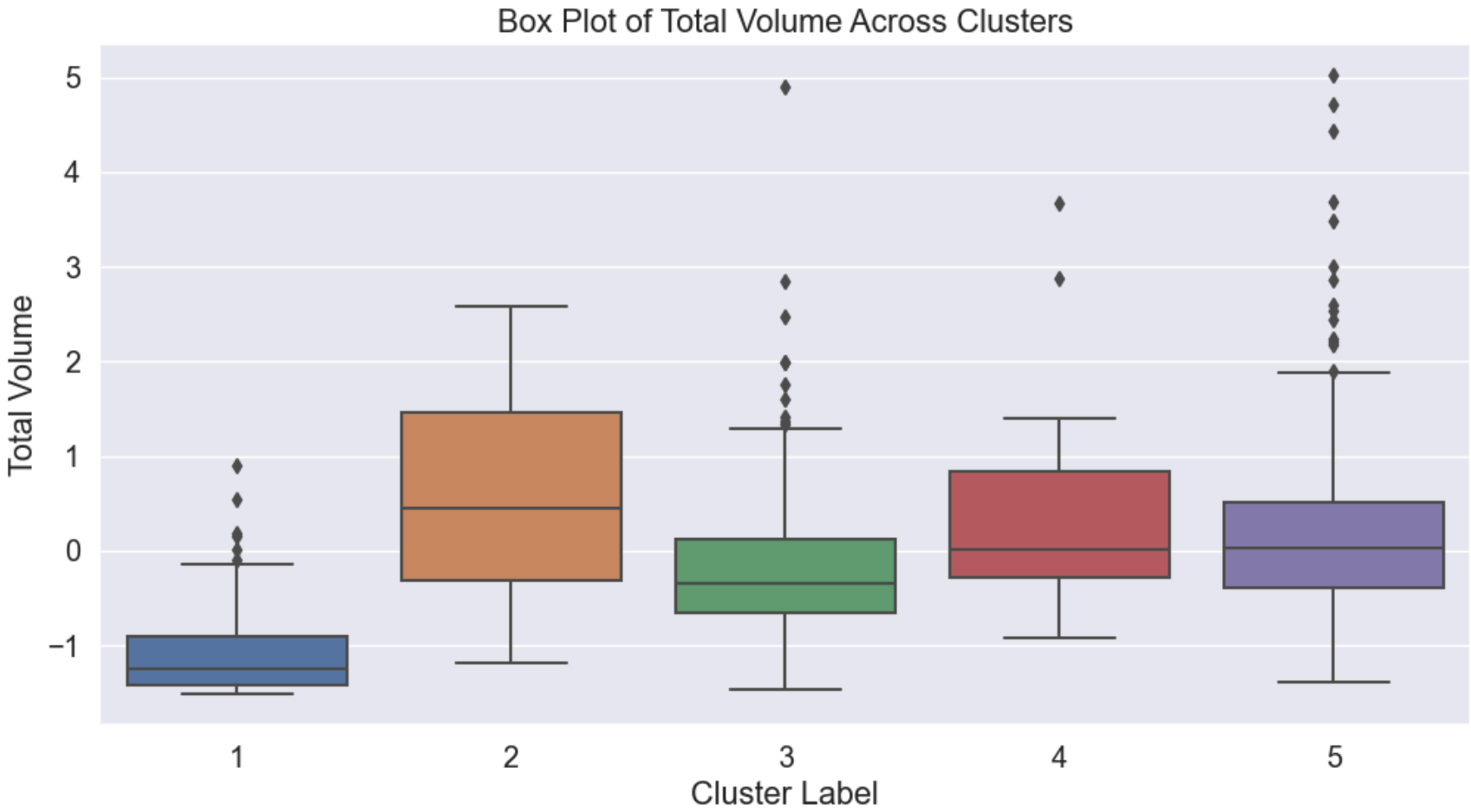

Box Plot of PropCat 6 Across Clusters
PropCat 6
5
4
3
2
1
0
1
2
3
4
5
Cluster Label

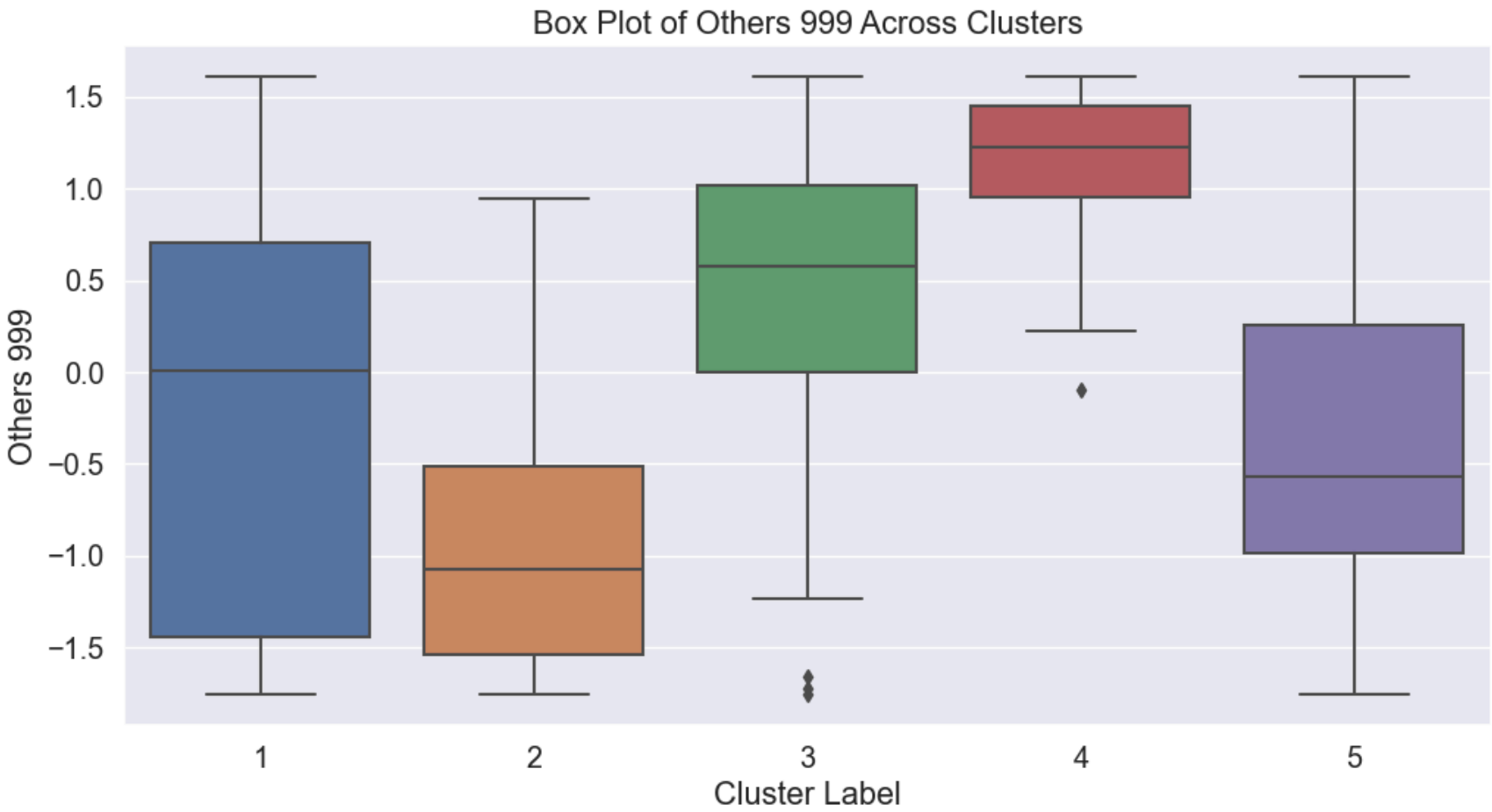

Box Plot of AGE Across Clusters
AGE
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
1
2
3
4
5
Cluster Label

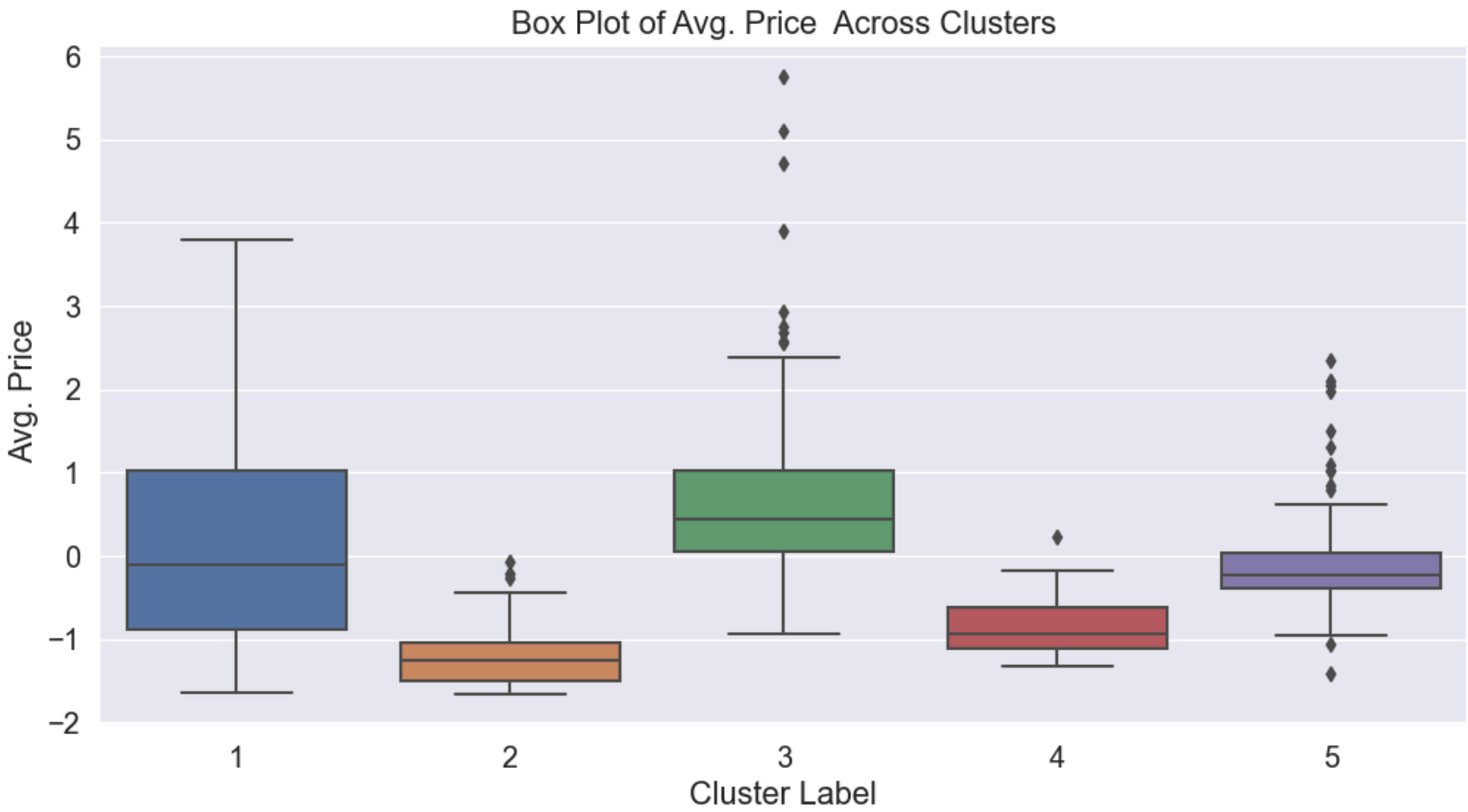

Box Plot of PropCat 5 Across Clusters
PropCat 5
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
1
2
3
4
5
Cluster Label

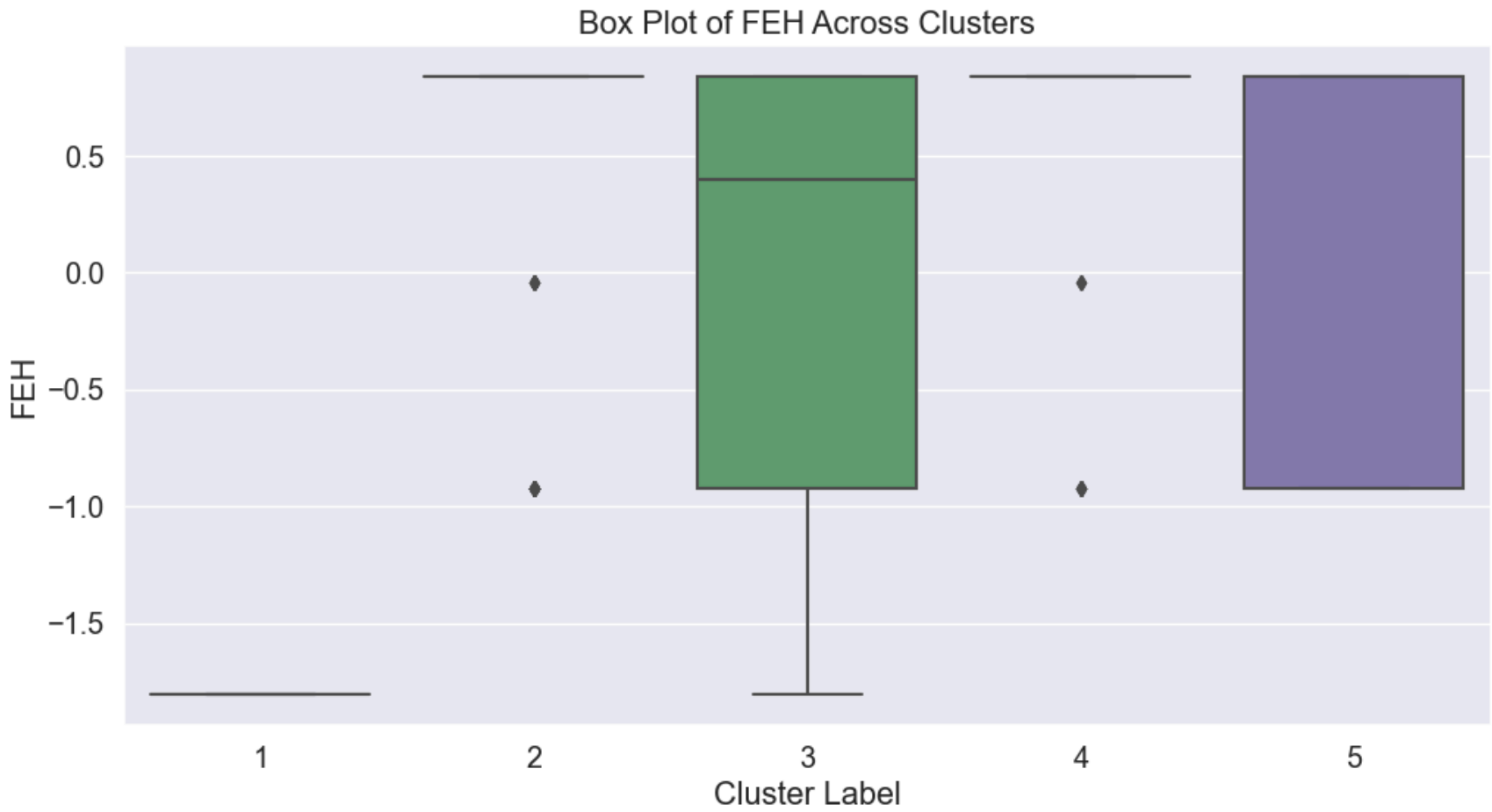

Box Plot of Pr Cat 1 Across Clusters
Pr Cat 1
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
1
2
3
4
5
Cluster Label

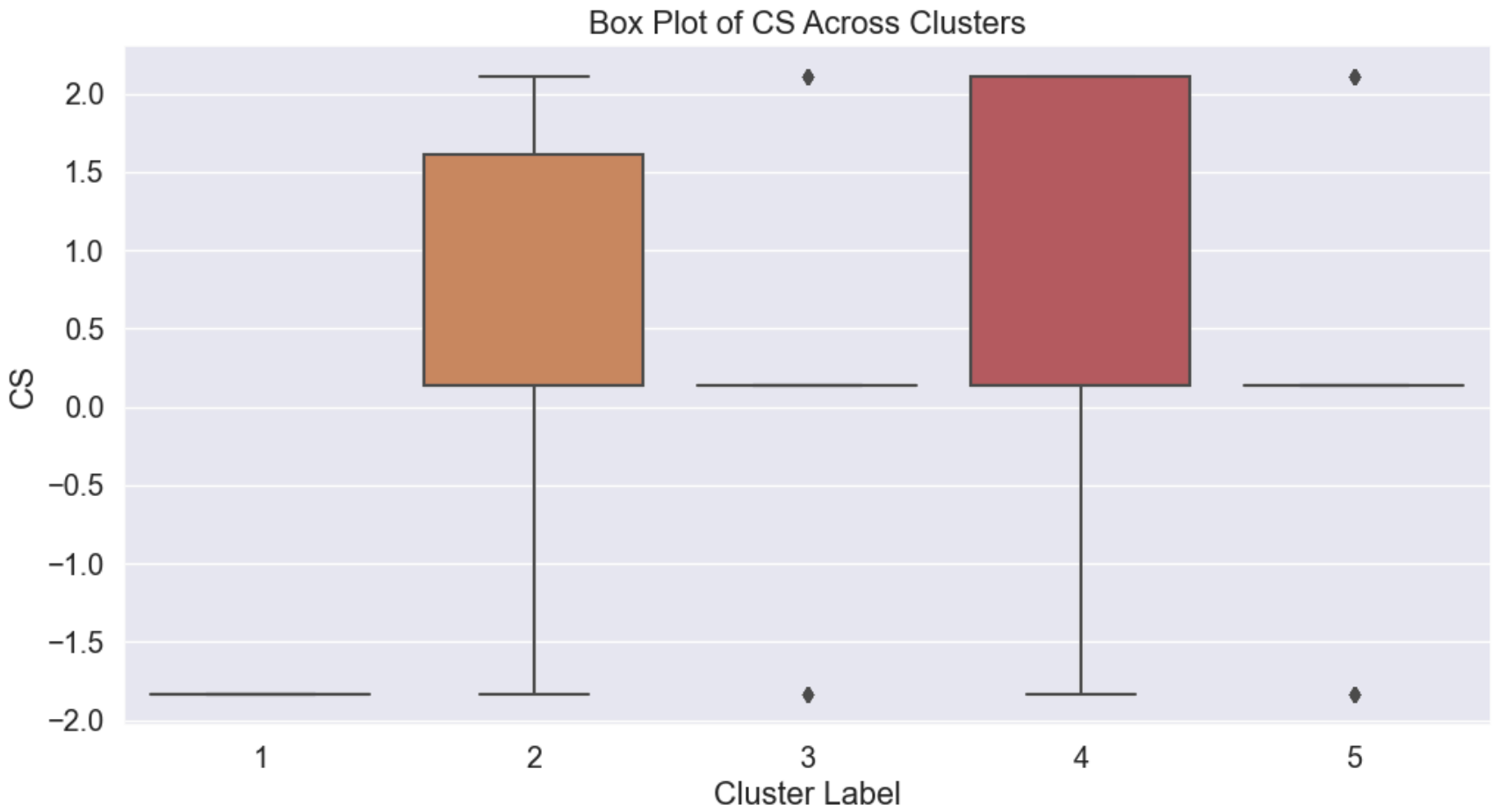

Box Plot of Pr Cat 2 Across Clusters
Pr Cat 2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
1
2
3
4
5
Cluster Label

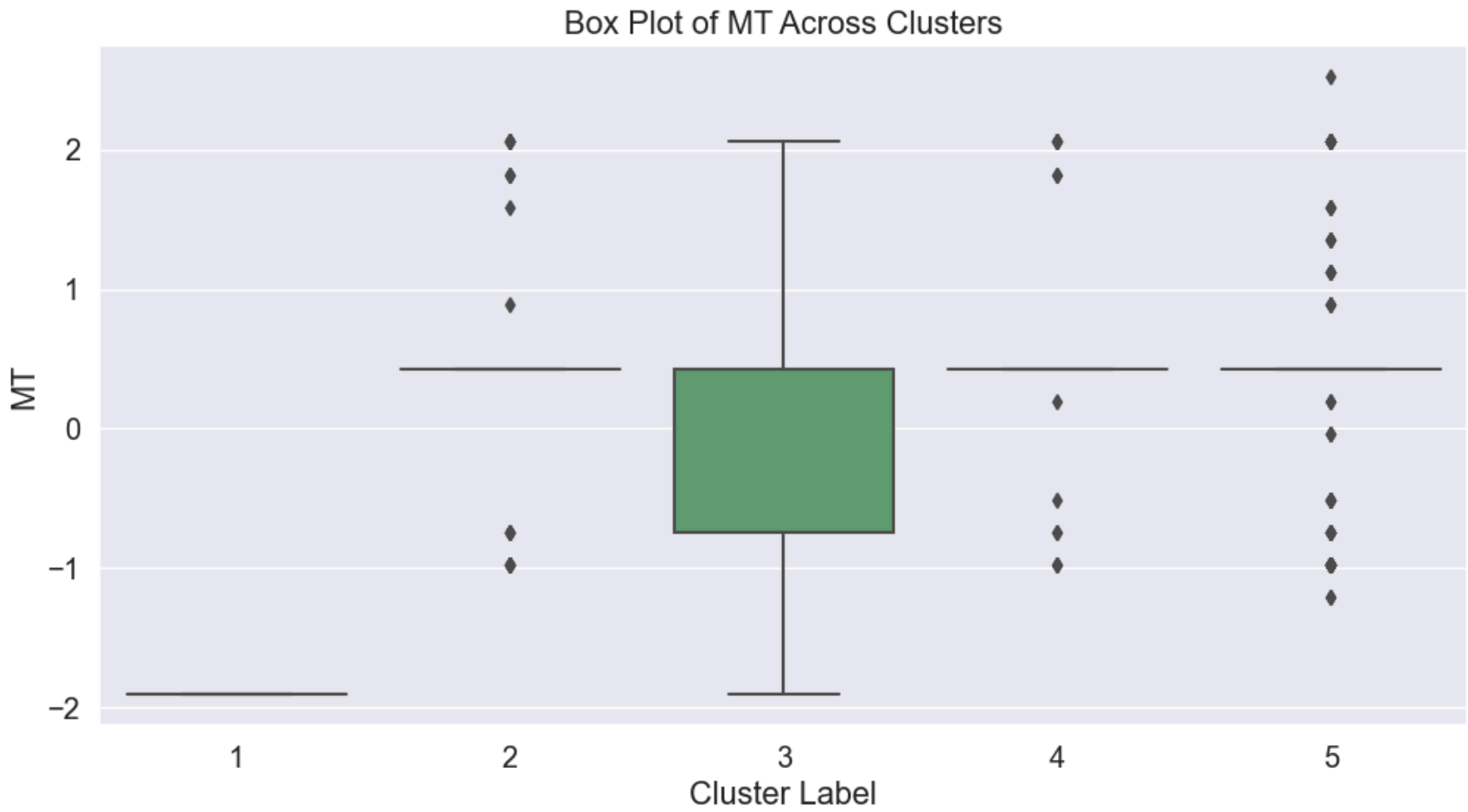

Box Plot of Total Volume Across Clusters
Total Volume
5
4
3
2
1
0
−1
1
2
3
4
5
Cluster Label

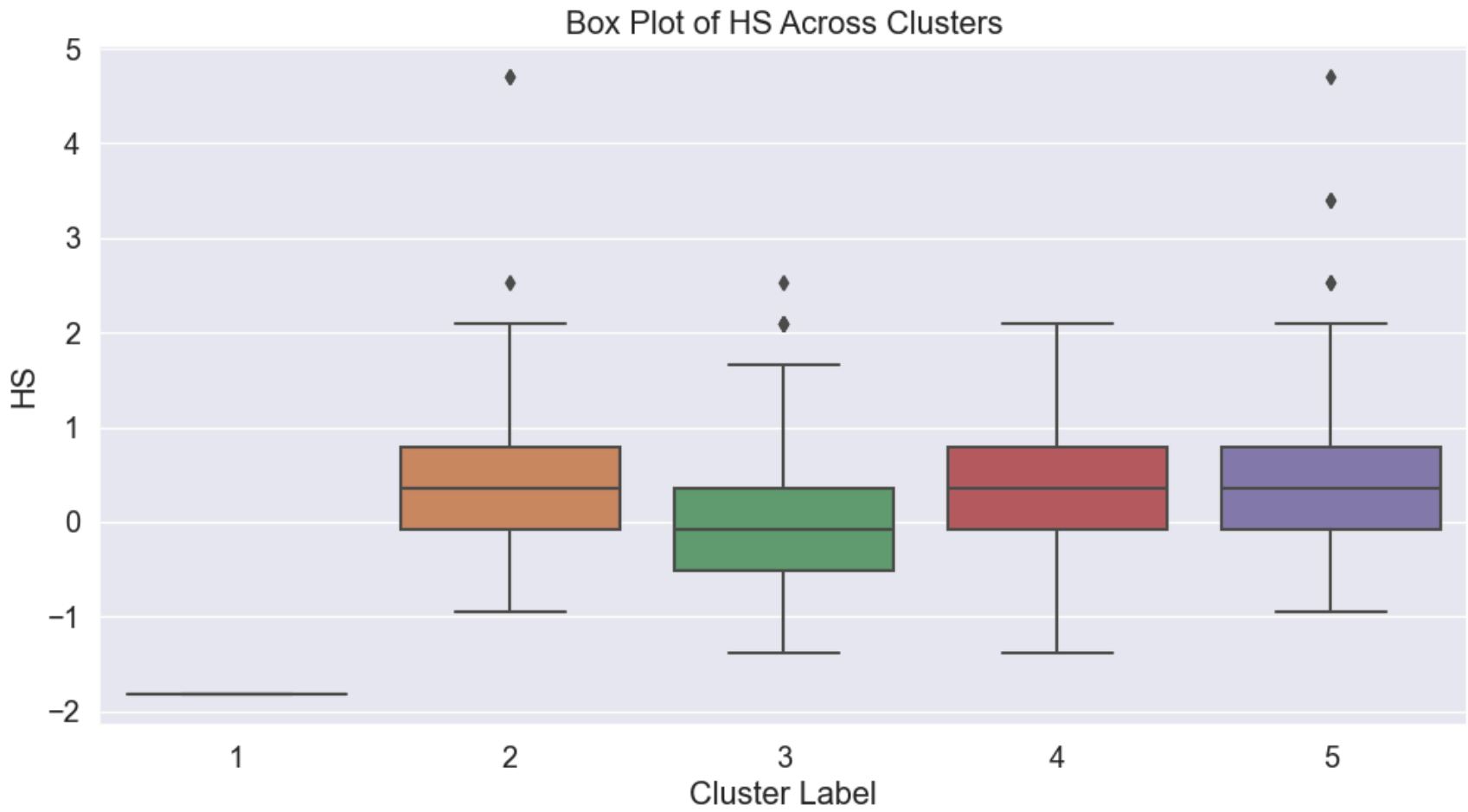

Box Plot of Others 999 Across Clusters
Others 999
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
1
2
3
4
5
Cluster Label

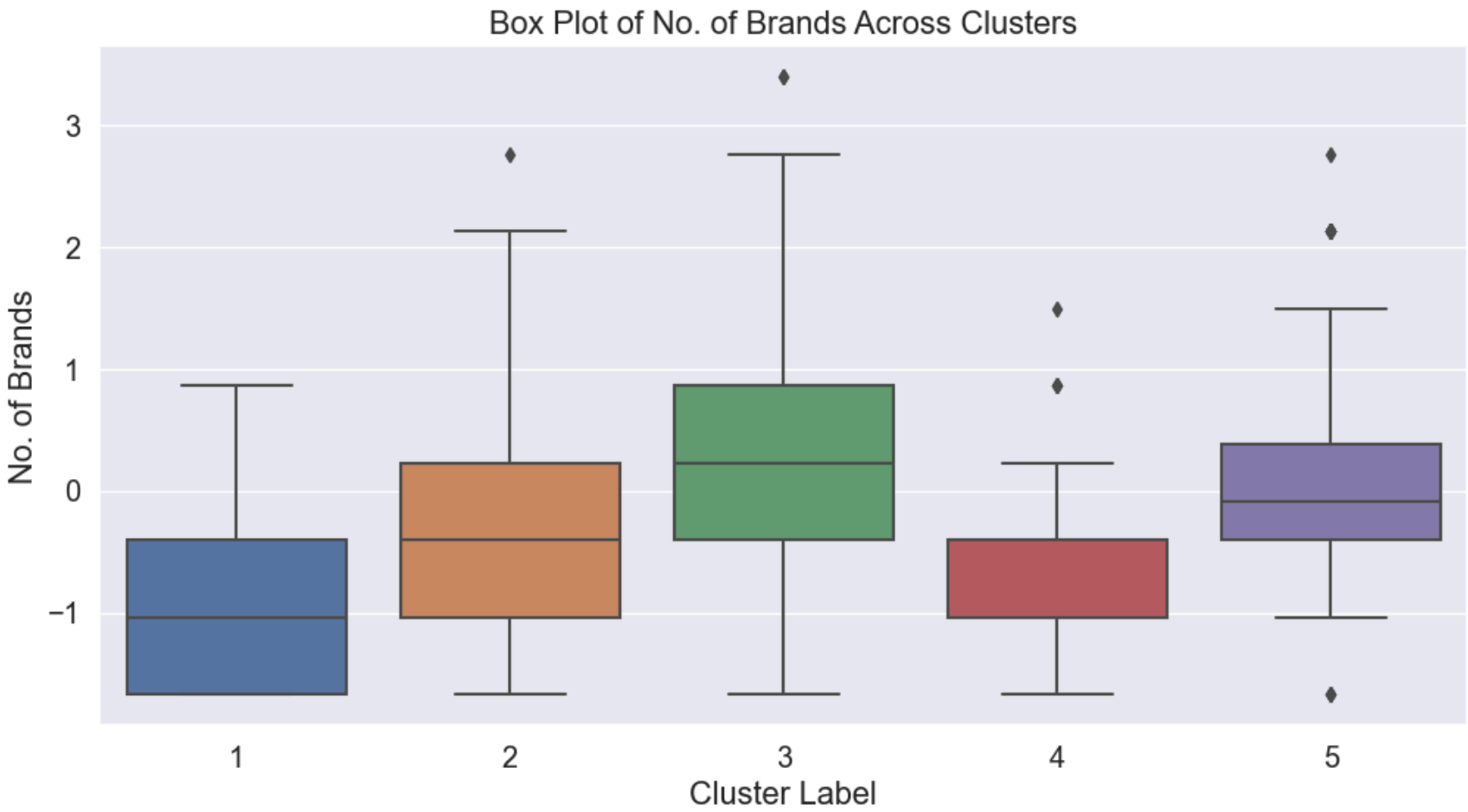

Box Plot of Avg. Price Across Clusters
Avg. Price
6
5
4
3
2
1
0
−1
−2
1
2
3
4
5
Cluster Label

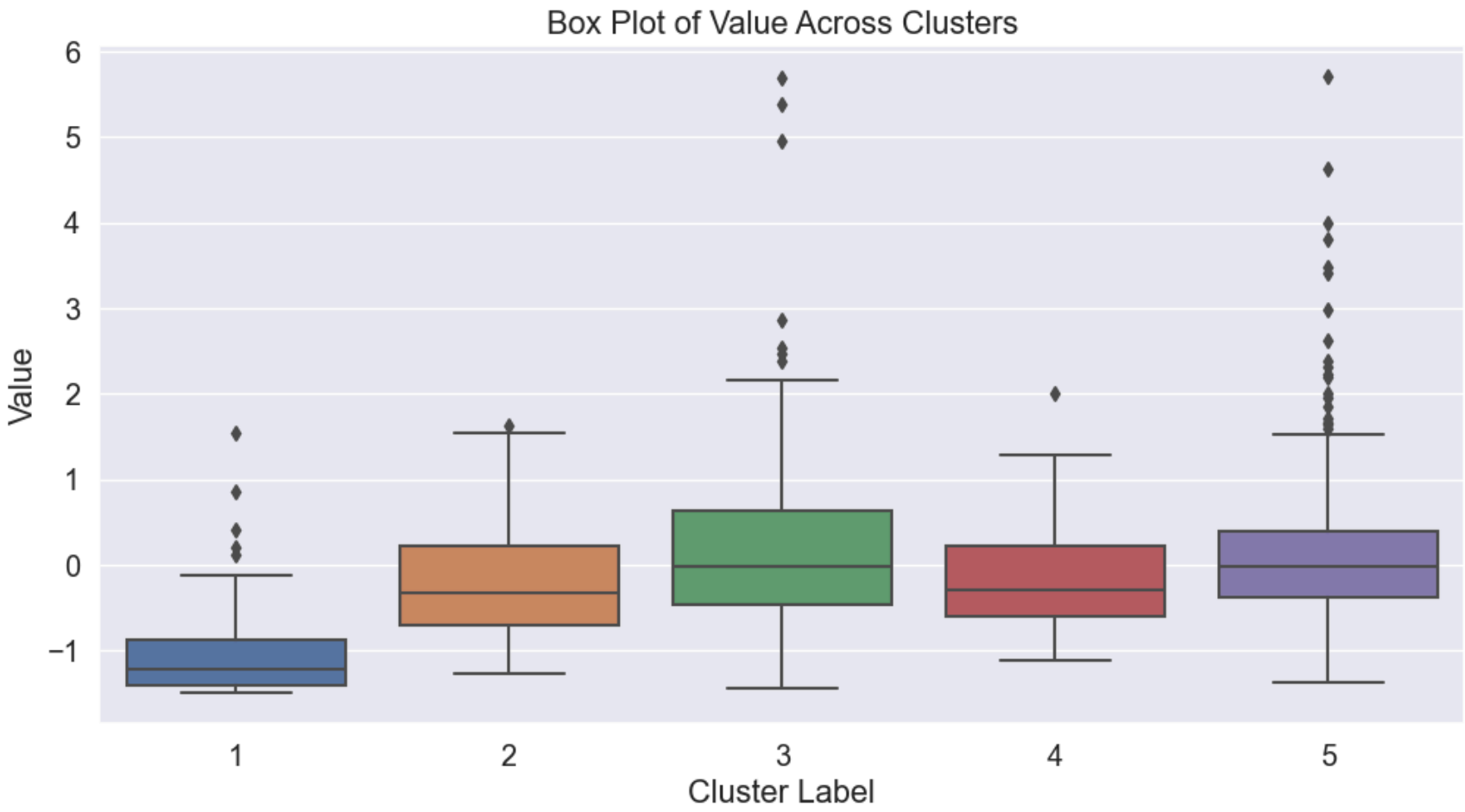

Box Plot of FEH Across Clusters
FEH
0.5
0.0
−0.5
−1.0
−1.5
1
2
3
4
5
Cluster Label

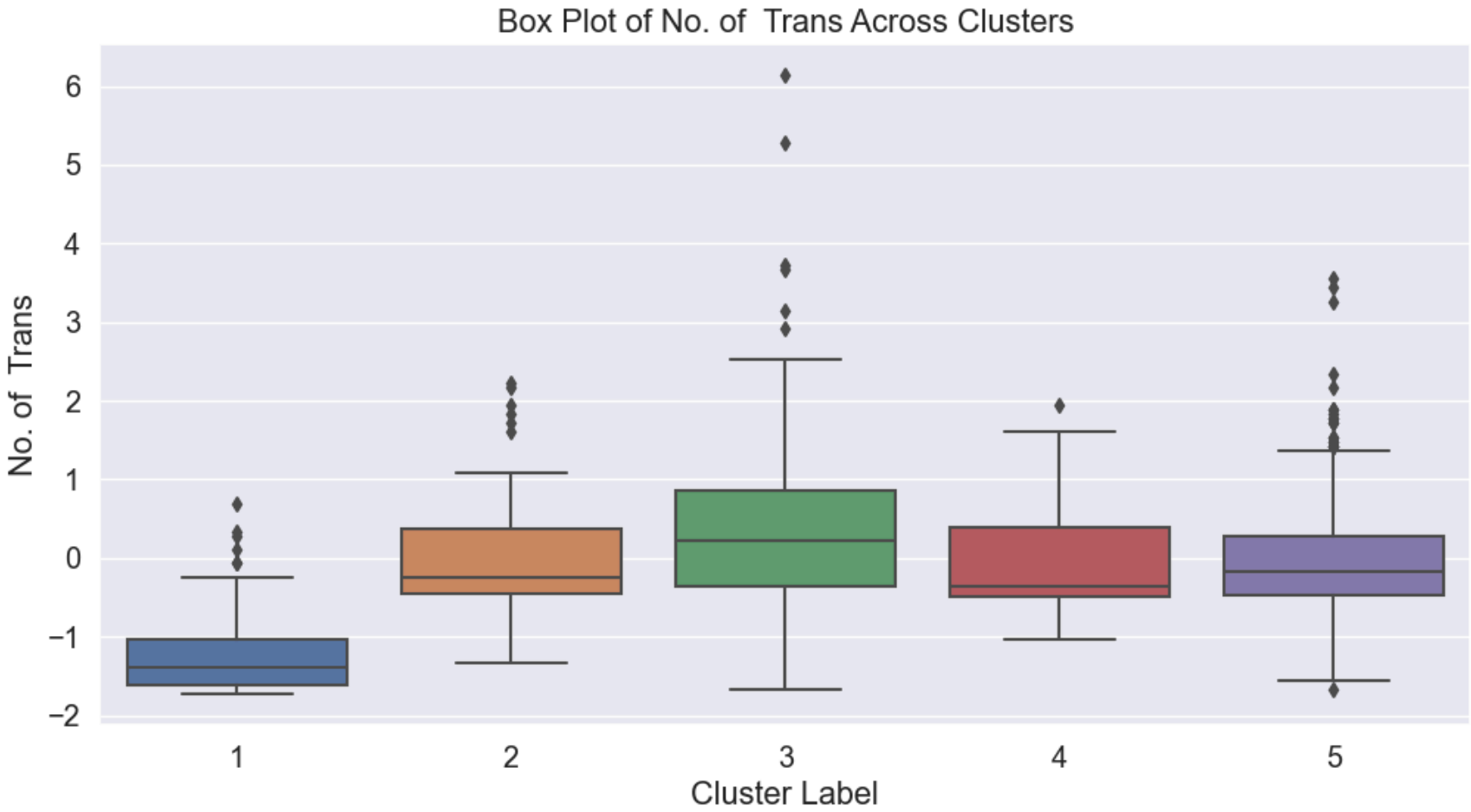

Box Plot of CS Across Clusters
CS
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
1
2
3
4
5
Cluster Label


Box Plot of MT Across Clusters
MT
2
1
0
−1
−2
1
2
3
4
5
Cluster Label

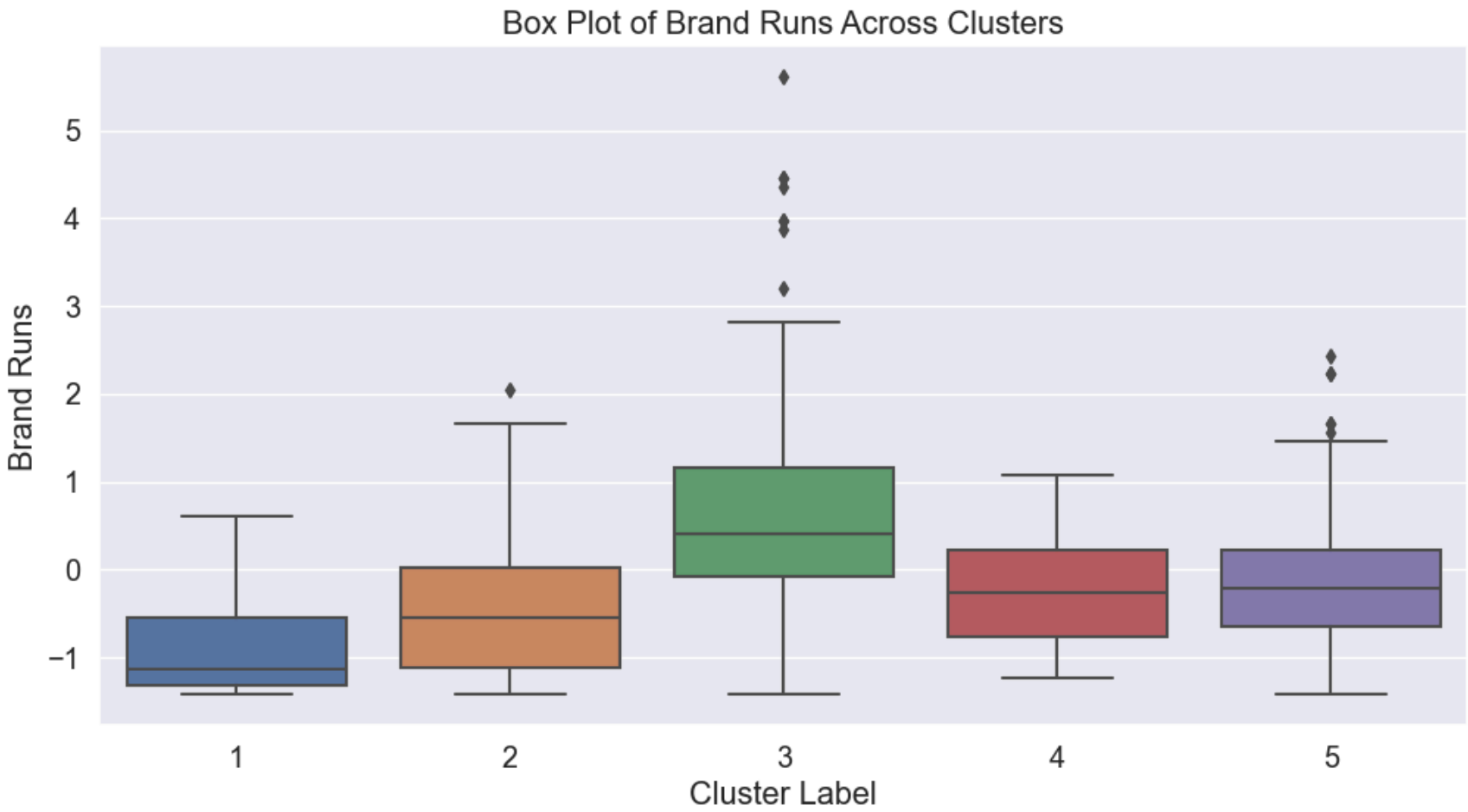

Box Plot of HS Across Clusters
HS
5
4
3
2
1
0
−1
−2
1
2
3
4
5
Cluster Label

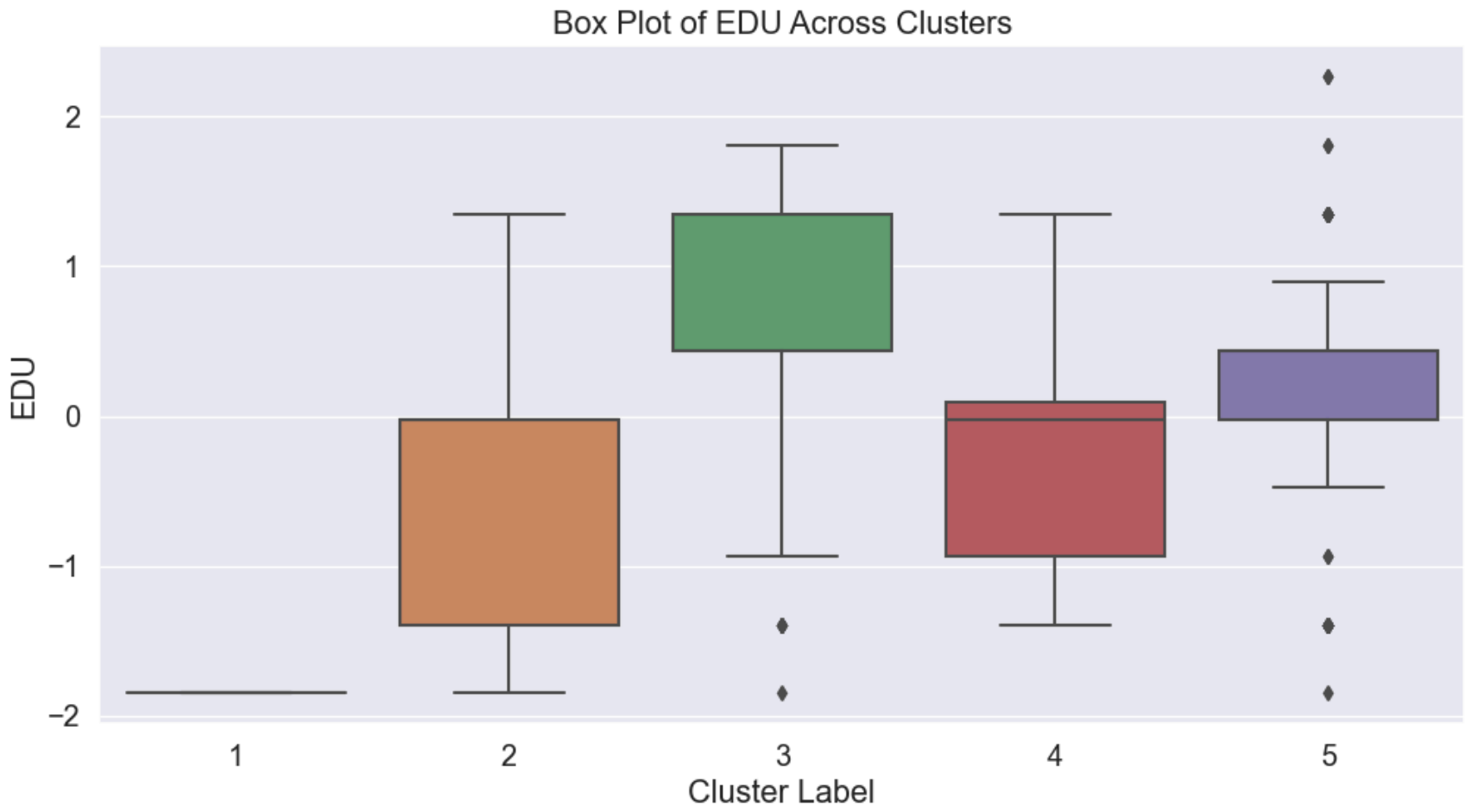

Box Plot of No. of Brands Across Clusters
No. of Brands
3
2
1
0
−1
1
2
3
4
5
Cluster Label


Box Plot of Value Across Clusters
Value
6
5
4
3
2
1
0
−1
1
2
3
4
5
Cluster Label


Box Plot of No. of Trans Across Clusters
No. of Trans
6
5
4
3
2
1
0
−1
−2
1
2
3
4
5
Cluster Label


Box Plot of SEX Across Clusters
SEX
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
1
2
3
4
5
Cluster Label


Box Plot of Brand Runs Across Clusters
Brand Runs
5
4
3
2
1
0
−1
1
2
3
4
5
Cluster Label


Box Plot of EDU Across Clusters
EDU
2
1
0
−1
−2
1
2
3
4
5
Cluster Label

In [35]:
```
# List of columns to visualize
columns_to_visualize = ['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'CHILD', 'Trans / Brand Runs', 'SEC', 'Vol/Tran',
                        'Pur Vol No Promo - %', 'Pr Cat 4', 'Br. Cd. 57, 144', 'Br. Cd. 286', 'Br. Cd. 352',
                        'PropCat 7', 'PropCat 8', 'PropCat 6', 'AGE', 'PropCat 5', 'Pr Cat 1', 'Pr Cat 2',
                        'Total Volume', 'Others 999', 'Avg. Price ', 'FEH', 'CS', 'MT', 'HS', 'No. of Brands',
                        'Value', 'No. of  Trans', 'SEX', 'Brand Runs', 'EDU', 'Affluence Index']

# Calculate the mean for each column across clusters
mean_values = filtered_df.groupby('Cluster Label')[columns_to_visualize].mean()

# Display the mean values as a table
print(mean_values)
```

```
                PropCat 14  Br. Cd. 55  Pr Cat 3     CHILD  Trans / Brand Runs  \
Cluster Label
1                 0.223202    0.234814  0.211376  1.452736           -0.347624
2                 2.105569    2.080112  2.116046 -0.158534            0.896193
3                -0.423353   -0.414882 -0.427214 -0.101809           -0.210812
4                -0.299102   -0.340120 -0.287865 -0.266626           -0.005349
5                -0.337869   -0.332378 -0.336176 -0.264427            0.008951

                    SEC  Vol/Tran  Pur Vol No Promo - %  Pr Cat 4  \
Cluster Label
1              -0.263067 -0.116618             -0.019914 -0.183359
2               0.749385  0.528135              0.176389 -0.228762
3              -0.472769 -0.465910             -0.300415 -0.204044
4               0.894427  0.464961             -0.239506  2.966138
5               0.109611  0.226621              0.303563 -0.285608

                Br. Cd. 57, 144  ...        CS        MT        HS  \
Cluster Label                    ...
1                      0.047514  ... -1.837792 -1.905900 -1.823913
2                     -0.591407  ...  0.428015  0.440272  0.516372
3                     -0.249048  ...  0.162973  0.069413  0.026419
4                     -0.535156  ...  0.583108  0.498676  0.391286
5                      0.570490  ...  0.163802  0.296581  0.309069

                No. of Brands     Value  No. of  Trans       SEX  Brand Runs  \
Cluster Label
1                   -0.757410 -1.017366      -1.208908 -2.682741   -0.876470
2                   -0.180763 -0.155549      -0.001045  0.341260   -0.434011
3                    0.414225  0.197580       0.397848  0.367081    0.601048
4                   -0.576152 -0.153117      -0.092339  0.368751   -0.188318
5                    0.015901  0.225181       0.013715  0.313044   -0.128516

                    EDU  Affluence Index
Cluster Label
1              -1.847808        -1.492908
2              -0.384169        -0.458109
3               0.593884         0.614339
4              -0.300236        -0.318724
5               0.208697         0.100150

[5 rows x 33 columns]
```

In [36]:

```
# Create a dictionary to store variables in each cluster
cluster_variable_dict = {cluster: [] for cluster in set(cluster_labels)}
```

```
# Iterate through each column and check which cluster it belongs to
for column in columns_to_visualize:
    cluster_membership = filtered_df.groupby('Cluster Label')[column].sum()
    for cluster, sum_value in cluster_membership.items():
        if sum_value > 0:
            cluster_variable_dict[cluster].append(column)

# Print the variables for each cluster
for cluster, variables in cluster_variable_dict.items():
    print(f'Cluster {cluster}: {variables}')
```

```
Cluster 1: ['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'CHILD', 'Br. Cd. 57, 144', 'PropCat 8', 'Pr Cat 1', 'Avg. Price ']
Cluster 2: ['PropCat 14', 'Br. Cd. 55', 'Pr Cat 3', 'Trans / Brand Runs', 'SEC', 'Vol/Tran', 'Pur Vol No Promo - %', 'T
otal Volume', 'FEH', 'CS', 'MT', 'HS', 'SEX']
Cluster 3: ['PropCat 7', 'PropCat 8', 'PropCat 6', 'AGE', 'Pr Cat 1', 'Others 999', 'Avg. Price ', 'FEH', 'CS', 'MT',
'HS', 'No. of Brands', 'Value', 'No. of  Trans', 'SEX', 'Brand Runs', 'EDU', 'Affluence Index']
Cluster 4: ['SEC', 'Vol/Tran', 'Pr Cat 4', 'AGE', 'PropCat 5', 'Total Volume', 'Others 999', 'FEH', 'CS', 'MT', 'HS',
'SEX']
Cluster 5: ['Trans / Brand Runs', 'SEC', 'Vol/Tran', 'Pur Vol No Promo - %', 'Br. Cd. 57, 144', 'Br. Cd. 286', 'Br. Cd.
352', 'PropCat 7', 'AGE', 'PropCat 5', 'Pr Cat 2', 'Total Volume', 'FEH', 'CS', 'MT', 'HS', 'No. of Brands', 'Value',
'No. of  Trans', 'SEX', 'EDU', 'Affluence Index']
```

In [37]:

```
# Iterate through each cluster and attach a mean value to each variable
for cluster_label in range(6):
    cluster_data = filtered_df[filtered_df['Cluster Label'] == cluster_label]

    if not cluster_data.empty:
        cluster_mean = cluster_data[columns_to_visualize].mean()
        print(f"Cluster {cluster_label}:")
        for column, mean_value in cluster_mean.items():
            print(f"{column}: {mean_value:.4f}")
        print("\n")
    else:
        print(f"Cluster {cluster_label}: (No data)\n")
```

Cluster 0: (No data)

Cluster 1:
PropCat 14: 0.2232
Br. Cd. 55: 0.2348
Pr Cat 3: 0.2114
CHILD: 1.4527
Trans / Brand Runs: -0.3476
SEC: -0.2631
Vol/Tran: -0.1166
Pur Vol No Promo - %: -0.0199
Pr Cat 4: -0.1834
Br. Cd. 57, 144: 0.0475
Br. Cd. 286: -0.1370
Br. Cd. 352: -0.1843
PropCat 7: -0.0898
PropCat 8: 0.1066
PropCat 6: -0.1316
AGE: -0.5868
PropCat 5: -0.1439
Pr Cat 1: 0.3198
Pr Cat 2: -0.3573
Total Volume: -1.0573
Others 999: -0.1655
Avg. Price : 0.1849
FEH: -1.8063
CS: -1.8378
MT: -1.9059
HS: -1.8239
No. of Brands: -0.7574
Value: -1.0174
No. of Trans: -1.2089
SEX: -2.6827
Brand Runs: -0.8765
EDU: -1.8478
Affluence Index: -1.4929

Cluster 2:
PropCat 14: 2.1056
Br. Cd. 55: 2.0801
Pr Cat 3: 2.1160
CHILD: -0.1585
Trans / Brand Runs: 0.8962
SEC: 0.7494

Vol/Tran: 0.5281
Pur Vol No Promo - %: 0.1764
Pr Cat 4: -0.2288
Br. Cd. 57, 144: -0.5914
Br. Cd. 286: -0.2333
Br. Cd. 352: -0.2582
PropCat 7: -0.4477
PropCat 8: -0.4450
PropCat 6: -0.0756
AGE: -0.1529
PropCat 5: -0.9832
Pr Cat 1: -0.7677
Pr Cat 2: -0.9875
Total Volume: 0.5710
Others 999: -0.9424
Avg. Price : -1.2017
FEH: 0.4817
CS: 0.4280
MT: 0.4403
HS: 0.5164
No. of Brands: -0.1808
Value: -0.1555
No. of Trans: -0.0010
SEX: 0.3413
Brand Runs: -0.4340
EDU: -0.3842
Affluence Index: -0.4581

Cluster 3:
PropCat 14: -0.4234
Br. Cd. 55: -0.4149
Pr Cat 3: -0.4272
CHILD: -0.1018
Trans / Brand Runs: -0.2108
SEC: -0.4728
Vol/Tran: -0.4659
Pur Vol No Promo - %: -0.3004
Pr Cat 4: -0.2040
Br. Cd. 57, 144: -0.2490
Br. Cd. 286: -0.1468
Br. Cd. 352: -0.1271
PropCat 7: 0.1558
PropCat 8: 0.4169
PropCat 6: 0.3531

AGE: 0.1498
PropCat 5: -0.2733
Pr Cat 1: 0.6837
Pr Cat 2: -0.1233
Total Volume: -0.1449
Others 999: 0.4857
Avg. Price : 0.6607
FEH: 0.0204
CS: 0.1630
MT: 0.0694
HS: 0.0264
No. of Brands: 0.4142
Value: 0.1976
No. of Trans: 0.3978
SEX: 0.3671
Brand Runs: 0.6010
EDU: 0.5939
Affluence Index: 0.6143

Cluster 4:
PropCat 14: -0.2991
Br. Cd. 55: -0.3401
Pr Cat 3: -0.2879
CHILD: -0.2666
Trans / Brand Runs: -0.0053
SEC: 0.8944
Vol/Tran: 0.4650
Pur Vol No Promo - %: -0.2395
Pr Cat 4: 2.9661
Br. Cd. 57, 144: -0.5352
Br. Cd. 286: -0.1454
Br. Cd. 352: -0.2482
PropCat 7: -0.3758
PropCat 8: -0.0434
PropCat 6: -0.3924
AGE: 0.1212
PropCat 5: 0.9394
Pr Cat 1: -0.5565
Pr Cat 2: -1.0754
Total Volume: 0.2988
Others 999: 1.1353
Avg. Price : -0.8467
FEH: 0.5987
CS: 0.5831

MT: 0.4987
HS: 0.3913
No. of Brands: -0.5762
Value: -0.1531
No. of Trans: -0.0923
SEX: 0.3688
Brand Runs: -0.1883
EDU: -0.3002
Affluence Index: -0.3187

Cluster 5:
PropCat 14: -0.3379
Br. Cd. 55: -0.3324
Pr Cat 3: -0.3362
CHILD: -0.2644
Trans / Brand Runs: 0.0090
SEC: 0.1096
Vol/Tran: 0.2266
Pur Vol No Promo - %: 0.3036
Pr Cat 4: -0.2856
Br. Cd. 57, 144: 0.5705
Br. Cd. 286: 0.3128
Br. Cd. 352: 0.3395
PropCat 7: 0.1130
PropCat 8: -0.2940
PropCat 6: -0.2076
AGE: 0.0707
PropCat 5: 0.4833
Pr Cat 1: -0.4119
Pr Cat 2: 0.8362
Total Volume: 0.2300
Others 999: -0.3479
Avg. Price : -0.1232
FEH: 0.2773
CS: 0.1638
MT: 0.2966
HS: 0.3091
No. of Brands: 0.0159
Value: 0.2252
No. of Trans: 0.0137
SEX: 0.3130
Brand Runs: -0.1285
EDU: 0.2087
Affluence Index: 0.1001

Cluster 1:

Purchase Behavior:

Total Volume: The mean value of -1.0573 indicates that customers in Cluster 1 have a relatively low total purchase volume. This suggests that they do not purchase large quantities of products.

Trans / Brand Runs: With a mean value of -0.3476, this variable suggests that, on average, customers in Cluster 1 have fewer transactions relative to the instances of consecutive brand purchases. This may indicate that they tend to make fewer purchases despite showing brand loyalty.

Pr Cat 1: This variable has a mean value of 0.3198, indicating that customers in Cluster 1 tend to make a relatively higher proportion of purchases in Price Category 1. This suggests a preference for lower-priced products.

Avg. Price: The mean value of 0.1849 suggests that, on average, customers in this cluster tend to make purchases at a slightly higher average price compared to other clusters.

CHILD: With a mean value of 1.4527, this variable suggests that households in Cluster 1 tend to have more children. This may impact their purchase behavior, as families with children often have different consumption patterns.

Basis of Purchase:

Pr Cat 3: The mean value of 0.2114 indicates that customers in Cluster 1 make a moderate proportion of purchases in Price Category 3.

Pr Cat 4: This variable has a mean value of -0.1834, suggesting that customers in this cluster tend to make fewer purchases in Price Category 4.

PropCat 5: With a mean value of -0.1439, this variable indicates a lower proportion of purchases related to Product Category 5.

PropCat 6: The mean value of -0.1316 suggests a lower proportion of purchases related to Product Category 6.

PropCat 7: This variable has a mean value of -0.0898, indicating a lower proportion of purchases related to Product Category 7.

PropCat 8: With a mean value of 0.1066, this variable suggests a moderate proportion of purchases related to Product Category 8.

PropCat 14: The mean value of 0.2232 indicates that customers in this cluster make a significant proportion of purchases related to Product Category 14.

Brand Loyalty:

Br. Cd. 55: The mean value of 0.2348 suggests that customers in Cluster 1 have a preference for Brand Code 55, indicating a degree of brand loyalty.

Br. Cd. 57, 144: With a mean value of 0.0475, this variable indicates a smaller proportion of purchases related to Brand Codes 57 and 144.

Br. Cd. 286: The mean value of -0.1370 suggests a lower proportion of purchases related to Brand Code 286.

Br. Cd. 352: With a mean value of -0.1843, this variable indicates a lower proportion of purchases related to Brand Code 352.

Others 999: The mean value of -0.1655 suggests a lower proportion of purchases related to other unspecified brands.

Brand Runs: This variable has a mean value of -0.8765, indicating very little brand loyalty in comparision to other clusters.

In summary, customers in Cluster 1 exhibit a relatively low purchase volume, tend to make purchases at lower price categories, and show very little brand loyalty, except towards Brand Code 55. They also tend to have more children in their households. These characteristics define their purchase behavior, basis of purchase, and brand loyalty within the cluster.

Cluster 2:

Purchase Behavior:

Total Volume: With a mean value of 0.5710, customers in Cluster 2 have a moderate total purchase volume. This suggests that they make purchases in moderate quantities.

Trans / Brand Runs: The mean value of 0.8962 indicates that customers in this cluster have a relatively high number of transactions relative to instances of consecutive brand purchases. This implies that they make more frequent purchases compared to some other clusters.

Avg. Price: With a mean value of -1.2017, this variable suggests that customers in Cluster 2 tend to make purchases at a lower average price compared to other clusters. They may be more price-sensitive.

CHILD: The mean value of -0.1585 suggests that households in Cluster 2 tend to have fewer children. This may impact their purchase behavior, as families with fewer children often have different consumption patterns.

Basis of Purchase:

Pr Cat 3: The mean value of 2.1160 indicates that customers in this cluster make a significant proportion of their purchases in Price Category 3.

Pr Cat 4: With a mean value of -0.2288, this variable suggests that customers in this cluster tend to make fewer purchases in Price Category 4.

PropCat 5: The mean value of -0.9832 indicates a lower proportion of purchases related to Product Category 5.

PropCat 6: With a mean value of -0.0756, this variable suggests a relatively lower proportion of purchases related to Product Category 6.

PropCat 7: The mean value of -0.4477 indicates a lower proportion of purchases related to Product Category 7.

PropCat 8: With a mean value of -0.4450, this variable suggests a lower proportion of purchases related to Product Category 8.

PropCat 14: The mean value of 2.1056 indicates that customers in Cluster 2 make a significant proportion of purchases related to Product Category 14.

Brand Loyalty:

Br. Cd. 55: With a mean value of 2.0801, customers in Cluster 2 have a strong preference for Brand Code 55, indicating a high level of brand loyalty to this specific brand.

Br. Cd. 57, 144: The mean value of -0.5914 suggests a lower proportion of purchases related to Brand Codes 57 and 144.

Br. Cd. 286: With a mean value of -0.2333, this variable indicates a lower proportion of purchases related to Brand Code 286.

Br. Cd. 352: The mean value of -0.2582 suggests a lower proportion of purchases related to Brand Code 352.

Others 999: With a mean value of -0.9424, this variable indicates a lower proportion of purchases related to other unspecified brands.

Brand Runs: 'Brand Runs' has a mean value of -0.4340, it indicates that customers in this cluster are less brand loyal and are more likely to switch between brands more frequently.

In summary, customers in Cluster 2 exhibit moderate purchase behavior with a preference for Brand Code 55 and a significant proportion of purchases related to Product Category 14. They tend to make purchases at lower average prices and have a preference for Price Category 3. Their brand loyalty is particularly strong for Brand Code 55.

Cluster 3:

Purchase Behavior:

Total Volume: With a mean value of -0.1449, customers in Cluster 3 have a relatively low total purchase volume. This suggests that they make fewer purchases or buy products in smaller quantities compared to other clusters.

Trans / Brand Runs: The mean value of -0.2108 indicates that customers in this cluster have a lower number of transactions relative to instances of consecutive brand purchases. This implies that they make fewer purchases and may not be as frequent buyers.

Avg. Price: With a mean value of 0.6607, this variable suggests that customers in Cluster 3 tend to make purchases at a higher average price compared to other clusters. They may not be as price-sensitive.

CHILD: The mean value of -0.1018 suggests that households in Cluster 3 tend to have fewer children. This may impact their purchase behavior, as families with fewer children often have different consumption patterns.

Basis of Purchase:

Pr Cat 3: The mean value of -0.4272 indicates that customers in this cluster make a lower proportion of their purchases in Price Category 3.

Pr Cat 4: With a mean value of -0.2040, this variable suggests that customers in Cluster 3 tend to make fewer purchases in Price Category 4.

PropCat 5: The mean value of -0.2733 indicates a lower proportion of purchases related to Product Category 5.

PropCat 6: With a mean value of 0.3531, this variable suggests a higher proportion of purchases related to Product Category 6.

PropCat 7: The mean value of 0.1558 indicates a moderate proportion of purchases related to Product Category 7.

PropCat 8: With a mean value of 0.4169, this variable suggests a higher proportion of purchases related to Product Category 8.

PropCat 14: The mean value of -0.4234 indicates that customers in Cluster 3 make a lower proportion of purchases related to Product Category 14.

Brand Loyalty:

Br. Cd. 55: With a mean value of -0.4149, customers in Cluster 3 have a lower preference for Brand Code 55, indicating a lower level of brand loyalty to this specific brand.

Br. Cd. 57, 144: The mean value of -0.2490 suggests a lower proportion of purchases related to Brand Codes 57 and 144.

Br. Cd. 286: With a mean value of -0.1468, this variable indicates a lower proportion of purchases related to Brand Code 286.

Br. Cd. 352: The mean value of -0.1271 suggests a lower proportion of purchases related to Brand Code 352.

Others 999: With a mean value of 0.4857, this variable indicates a higher proportion of purchases related to other unspecified brands.

Brand Runs: The mean value of 0.6010 suggests a higher degree of brand loyalty, and customers are less likely to switch brands.

In summary, customers in Cluster 3 exhibit relatively low purchase behavior, with a preference for higher-priced items and a mix of product category preferences. Their brand loyalty is not very strong for Brand Code 55, and they make fewer transactions.

Cluster 4:

Purchase Behavior:

Total Volume: With a mean value of 0.2988, customers in Cluster 4 have a moderate total purchase volume. This suggests that they make a moderate number of purchases or buy products in moderate quantities.

Trans / Brand Runs: The mean value of -0.0053 indicates that customers in this cluster have a relatively low ratio of transactions to instances of consecutive brand purchases. They don't make many purchases compared to brand runs, suggesting less frequent buying behavior.

Avg. Price: With a mean value of -0.8467, this variable suggests that customers in Cluster 4 tend to make purchases at a lower average price compared to other clusters. They may be price-sensitive shoppers.

CHILD: The mean value of -0.2666 suggests that households in Cluster 4 tend to have fewer children. This may impact their purchase behavior, as families with fewer children often have different consumption patterns.

Basis of Purchase:

Pr Cat 3: The mean value of -0.2879 indicates that customers in this cluster make a lower proportion of their purchases in Price Category 3.

Pr Cat 4: With a mean value of 2.9661, this variable suggests that customers in Cluster 4 make a significantly higher proportion of their purchases in Price Category 4.

PropCat 5: The mean value of 0.9394 indicates a significantly higher proportion of purchases related to Product Category 5.

PropCat 6: With a mean value of -0.3924, this variable suggests a lower proportion of purchases related to Product Category 6.

PropCat 7: The mean value of -0.3758 indicates a lower proportion of purchases related to Product Category 7.

PropCat 8: With a mean value of -0.0434, this variable suggests a relatively lower proportion of purchases related to Product Category 8.

PropCat 14: The mean value of -0.2991 indicates that customers in Cluster 4 make a lower proportion of purchases related to Product Category 14.

Brand Loyalty:

Br. Cd. 55: With a mean value of -0.3401, customers in Cluster 4 have a lower preference for Brand Code 55, indicating a lower level of brand loyalty to this specific brand.

Br. Cd. 57, 144: The mean value of -0.5352 suggests a lower proportion of purchases related to Brand Codes 57 and 144.

Br. Cd. 286: With a mean value of -0.1454, this variable indicates a lower proportion of purchases related to Brand Code 286.

Br. Cd. 352: The mean value of -0.2482 suggests a lower proportion of purchases related to Brand Code 352.

Others 999: With a mean value of 1.1353, this variable indicates a higher proportion of purchases related to other unspecified brands.

Brand Runs: The mean value of -0.1883 suggests a lower level of brand loyalty. Customers in this cluster do not exhibit strong brand loyalty but do not frequently switch brands either.

In summary, customers in Cluster 4 exhibit moderate purchase behavior with a preference for lower-priced items in Price Category 4. They have a mixed basis of purchase, with a significant proportion of purchases related to Product Category 5. Their brand loyalty is not very strong, and they make fewer transactions relative to instances of consecutive brand purchases.

Cluster 5:

Purchase Behavior:

Total Volume: With a mean value of 0.2300, customers in Cluster 5 have a moderate total purchase volume. This suggests that they make a moderate number of purchases or buy products in moderate quantities.

Trans / Brand Runs: The mean value of 0.0090 indicates that customers in this cluster have a relatively balanced ratio of transactions to instances of consecutive brand purchases. Their buying behavior is not skewed towards one extreme.

Avg. Price: With a mean value of -0.1232, this variable suggests that customers in Cluster 5 tend to make purchases at a slightly lower average price compared to other clusters. They may have a preference for lower-priced items.

CHILD: The mean value of -0.2644 suggests that households in Cluster 5 tend to have fewer children. This may impact their purchase behavior, as families with fewer children often have different consumption patterns.

Basis of Purchase:

Pr Cat 3: The mean value of -0.3362 indicates that customers in this cluster make a lower proportion of their purchases in Price Category 3.

Pr Cat 4: With a mean value of -0.2856, this variable suggests that customers in Cluster 5 make a lower proportion of their purchases in Price Category 4.

PropCat 5: The mean value of 0.4833 indicates a significantly higher proportion of purchases related to Product Category 5.

Pr Cat 2: With a mean value of 0.8362, this variable suggests a significantly higher proportion of purchases in Price Category 2.

PropCat 6: The mean value of -0.2076 indicates a lower proportion of purchases related to Product Category 6.

Brand Loyalty:

Br. Cd. 55: With a mean value of -0.3324, customers in Cluster 5 have a lower preference for Brand Code 55, indicating a lower level of brand loyalty to this specific brand.

Br. Cd. 57, 144: The mean value of 0.5705 suggests a higher proportion of purchases related to Brand Codes 57 and 144, indicating a preference for these brands.

Br. Cd. 286: With a mean value of 0.3128, this variable indicates a higher proportion of purchases related to Brand Code 286.

Br. Cd. 352: The mean value of 0.3395 suggests a higher proportion of purchases related to Brand Code 352.

In summary, customers in Cluster 5 exhibit moderate purchase behavior with a preference for lower-priced items in Price Categories 3 and 4. They have a mixed basis of purchase, with a significant proportion of purchases related to Product Category 5 and Price Category 2. Their brand loyalty is not very strong, and they have a balanced ratio of transactions to brand runs, indicating a moderate level of brand loyalty and frequency of brand switching.

Member ID could be added to the study to identify individuals in each cluster for marketing purposes.