Factor Analysis on K-Medoid clusters

Factor Analysis on 5th cluster

In [1]:
```python
import pandas as pd
from factor_analyzer import FactorAnalyzer
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity, calculate_kmo
import seaborn as sns
import matplotlib.pyplot as plt

#First the unnecessary columns are removed:

df = pd.read_excel('Asansol MRBQ Data.xlsx')
df = df.drop(columns = ['TOT2'])

df_2 = pd.read_excel('clustered data KMedoids.xlsx')

clusters = list(df_2['Cluster'])

df['Cluster'] = clusters
df
```

Out[1]:

	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	...	Q 28	Q 29	Q 30	Q 31	Q 32	Q 33	Q 34	Q 35	Q 36	Cluster
0	6	5	6	3	6	7	7	1	6	6	...	2	1	6	2	2	6	7	2	4	2
1	4	4	3	2	2	5	2	1	1	3	...	1	4	3	4	3	4	3	3	3	1
2	6	7	6	5	3	4	1	1	4	1	...	1	5	5	1	6	6	1	6	6	4
3	1	1	2	6	2	1	1	1	2	1	...	2	2	1	1	1	2	1	2	2	2
4	4	7	4	6	6	4	4	3	3	3	...	7	1	7	1	3	1	3	1	3	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
476	5	1	5	3	5	3	1	1	5	2	...	1	5	2	1	5	3	1	3	3	4
477	6	7	5	5	6	5	1	1	5	1	...	1	6	2	1	5	1	1	2	2	4
478	2	4	3	3	6	2	1	1	4	2	...	1	5	2	1	2	2	1	1	1	0
479	5	7	5	5	7	3	2	1	5	1	...	2	6	2	1	5	2	1	3	2	1
480	6	1	3	4	4	2	2	1	4	2	...	1	4	2	1	4	2	1	1	1	1

481 rows × 37 columns

In [2]:
```python
# Only the rows beliongaing to the 5th cluster are filtered out.

cluster_5 = df.loc[df['Cluster']==4]
cluster_5
```

Out[2]:

	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	...	Q 28	Q 29	Q 30	Q 31	Q 32	Q 33	Q 34	Q 35	Q 36	Cluster
2	6	7	6	5	3	4	1	1	4	1	...	1	5	5	1	6	6	1	6	6	4
9	2	3	5	6	5	4	2	6	2	3	...	4	5	5	6	5	2	2	2	3	4
11	1	2	1	7	1	7	5	1	1	1	...	1	1	1	2	2	2	1	2	3	4
21	6	4	4	6	2	6	3	4	5	5	...	3	5	5	6	3	5	3	4	4	4
25	4	3	3	6	4	4	3	4	5	5	...	5	6	3	6	6	3	5	7	2	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
473	7	7	6	6	4	3	3	1	3	1	...	1	7	3	1	6	1	1	3	3	4
474	6	1	2	3	5	2	1	1	5	1	...	1	4	2	1	4	3	1	2	2	4
475	6	7	1	1	6	1	1	1	6	1	...	1	5	3	1	4	4	1	3	3	4
476	5	1	5	3	5	3	1	1	5	2	...	1	5	2	1	5	3	1	3	3	4
477	6	7	5	5	6	5	1	1	5	1	...	1	6	2	1	5	1	1	2	2	4

117 rows × 37 columns

In [4]:
```python
cluster_5 = cluster_5.drop(columns=['Cluster'])
cluster_5
```

Out[4]:

	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	...	Q 27	Q 28	Q 29	Q 30	Q 31	Q 32	Q 33	Q 34	Q 35	Q 36
2	6	7	6	5	3	4	1	1	4	1	...	5	1	5	5	1	6	6	1	6	6
9	2	3	5	6	5	4	2	6	2	3	...	6	4	5	5	6	5	2	2	2	3
11	1	2	1	7	1	7	5	1	1	1	...	7	1	1	1	2	2	2	1	2	3
21	6	4	4	6	2	6	3	4	5	5	...	5	3	5	5	6	3	5	3	4	4
25	4	3	3	6	4	4	3	4	5	5	...	6	5	6	3	6	6	3	5	7	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
473	7	7	6	6	4	3	3	1	3	1	...	5	1	7	3	1	6	1	1	3	3
474	6	1	2	3	5	2	1	1	5	1	...	4	1	4	2	1	4	3	1	2	2
475	6	7	1	1	6	1	1	1	6	1	...	6	1	5	3	1	4	4	1	3	3
476	5	1	5	3	5	3	1	1	5	2	...	5	1	5	2	1	5	3	1	3	3
477	6	7	5	5	6	5	1	1	5	1	...	6	1	6	2	1	5	1	1	2	2

117 rows × 36 columns

In [5]:
```python
# Next the KMO (Kiaser-Meyer-Olkin) test was used to determine whether the cluster is suitable for factor analysis
# The closer the overall KMO is to 1, the better suited the dataset.

kmo_all, kmo_model = calculate_kmo(cluster_5)

print("KMO per variable:", kmo_all)
print("Overall KMO:", kmo_model)

# Since the cluster has a reasonable KMO value, going ahead with the Factor Analysis
```

```
KMO per variable: [0.8753624  0.71022385 0.93372181 0.85828974 0.85034883 0.86167122
 0.88889427 0.91029094 0.89244858 0.86228494 0.63324518 0.60895361
 0.90776095 0.82460758 0.79200947 0.9073914  0.9015926  0.89589254
 0.83896281 0.7610136  0.91539386 0.88522506 0.88711132 0.82239618
 0.7937945  0.80342156 0.8742284  0.90747866 0.8272541  0.8778996
 0.88076142 0.89549342 0.85383294 0.90660564 0.80319907 0.82362279]
Overall KMO: 0.8622989636604216
```

In [6]:
```python
# The original correlation matrix for the third cluster:
cluster_5.corr()
```

Out[6]:

	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	...	Q 27	Q 28	Q 29	Q 30	Q 31	Q 32	Q 33	Q
Q 1	1.000000	0.549596	0.455565	0.430469	0.661759	0.204717	-0.099678	-0.200118	0.580848	-0.069881	...	0.514901	-0.172580	0.706379	0.181452	-0.160434	0.606780	0.225443	-0.100
Q 2	0.549596	1.000000	0.317573	0.238204	0.545300	0.228705	0.019986	-0.134238	0.262989	-0.113943	...	0.205570	-0.129923	0.550601	0.110339	-0.134234	0.266028	-0.047341	-0.097
Q 3	0.455565	0.317573	1.000000	0.792832	0.524963	0.736146	0.364779	0.252652	0.431952	0.306397	...	0.415121	0.255846	0.505488	0.509330	0.372167	0.606943	0.475279	0.297
Q 4	0.430469	0.238204	0.792832	1.000000	0.400708	0.758694	0.474953	0.282873	0.340266	0.304449	...	0.374682	0.331698	0.406586	0.447028	0.388240	0.571392	0.422799	0.303
Q 5	0.661759	0.545300	0.524963	0.400708	1.000000	0.318334	-0.011559	-0.096733	0.586682	0.031859	...	0.487347	-0.069026	0.756054	0.181720	-0.052887	0.539596	0.195562	-0.057
Q 6	0.204717	0.228705	0.736146	0.758694	0.318334	1.000000	0.550445	0.376906	0.242402	0.506419	...	0.320262	0.396051	0.278704	0.460704	0.553147	0.413676	0.413179	0.442
Q 7	-0.099678	0.019986	0.364779	0.474953	-0.011559	0.550445	1.000000	0.630134	0.021620	0.436633	...	-0.002610	0.573789	-0.186565	0.473245	0.619036	0.076680	0.421652	0.571
Q 8	-0.200118	-0.134238	0.252652	0.282873	-0.096733	0.376906	0.630134	1.000000	-0.007126	0.605115	...	-0.012610	0.728812	-0.181167	0.520657	0.666529	0.118412	0.419423	0.668
Q 9	0.580848	0.262989	0.431952	0.340266	0.586682	0.242402	0.021620	-0.007126	1.000000	0.147352	...	0.504755	0.037797	0.539855	0.284533	0.094327	0.526853	0.393403	-0.004
Q 10	-0.069881	-0.113943	0.306397	0.304449	0.031859	0.506419	0.436633	0.605115	0.147352	1.000000	...	-0.015188	0.579217	0.007247	0.357953	0.638239	0.135948	0.441113	0.527
Q 11	0.148237	0.276292	0.072018	0.124567	0.214306	0.025936	0.240867	0.282218	0.221657	0.166466	...	0.146010	0.202029	0.165838	0.129177	0.105890	0.156099	0.060334	0.128
Q 12	0.021801	0.001922	0.095299	0.064396	0.034803	0.206357	0.078850	0.173455	0.145391	0.375477	...	0.159923	0.021329	0.138509	0.001852	0.120610	0.088896	-0.042631	0.103
Q 13	0.486696	0.321482	0.467029	0.404974	0.539106	0.443985	0.197499	0.046384	0.446520	0.229283	...	0.386758	0.071273	0.413056	0.285125	0.221324	0.353399	0.366070	0.167
Q 14	0.468119	0.471484	0.430450	0.321120	0.630385	0.227147	0.004110	-0.079610	0.392438	0.081625	...	0.228510	-0.058132	0.506464	0.206954	-0.037176	0.306078	0.162282	-0.013
Q 15	0.029757	0.054173	0.315195	0.295027	0.154048	0.365463	0.369771	0.459505	0.278613	0.577794	...	0.075582	0.360261	0.193011	0.285436	0.467437	0.179769	0.189077	0.306
Q 16	-0.137052	-0.053562	0.342519	0.326162	-0.071698	0.493178	0.698744	0.712298	-0.014390	0.498484	...	0.007461	0.679246	-0.228407	0.532880	0.747343	0.118056	0.496481	0.682
Q 17	-0.052695	-0.050990	0.372955	0.424624	-0.063581	0.550282	0.568591	0.543926	0.082754	0.488300	...	0.150150	0.549876	-0.132640	0.450386	0.605775	0.144487	0.556404	0.520
Q 18	0.003144	-0.088390	0.319888	0.385959	-0.027609	0.532209	0.569139	0.606632	0.163727	0.503300	...	0.202839	0.553500	-0.102112	0.383598	0.658039	0.228859	0.417508	0.667
Q 19	0.517930	0.323722	0.348623	0.249058	0.591855	0.200058	-0.114469	-0.096060	0.516444	-0.080272	...	0.501712	0.045767	0.606326	0.312730	0.015497	0.479940	0.258799	-0.066
Q 20	0.379978	0.416729	0.326301	0.239785	0.497703	0.218258	0.036729	-0.047361	0.359785	0.054143	...	0.276639	-0.094222	0.381392	0.195988	-0.128698	0.354719	0.081894	-0.190
Q 21	-0.101512	-0.072628	0.390386	0.408711	0.089007	0.476268	0.608079	0.708364	0.146457	0.629588	...	-0.020519	0.748766	-0.091719	0.537970	0.763195	0.242544	0.495262	0.655
Q 22	0.629101	0.418886	0.478828	0.454709	0.628581	0.429232	-0.007304	-0.089384	0.582652	0.097442	...	0.583716	-0.043928	0.640141	0.227787	0.038928	0.569528	0.244713	-0.088
Q 23	-0.017275	0.020882	0.343457	0.267648	-0.024093	0.398119	0.571434	0.508295	0.125208	0.421687	...	-0.043879	0.572043	-0.133212	0.453575	0.678138	0.041324	0.413378	0.738
Q 24	0.041720	-0.067512	0.269165	0.253294	-0.089518	0.389190	0.420389	0.328246	0.184364	0.271735	...	0.252759	0.284611	-0.112175	0.207894	0.385127	0.083609	0.281198	0.348
Q 25	-0.176460	-0.030078	0.215616	0.135805	-0.074689	0.285925	0.398337	0.477140	-0.035665	0.398396	...	-0.186422	0.536759	-0.215221	0.244400	0.459107	-0.030526	0.310113	0.574
Q 26	-0.116310	0.034225	0.263305	0.204791	0.022293	0.272588	0.399203	0.496278	0.057570	0.431354	...	-0.174604	0.548810	-0.164557	0.287890	0.548549	0.054399	0.323101	0.534
Q 27	0.514901	0.205570	0.415121	0.374682	0.487347	0.320262	-0.002610	-0.012610	0.504755	-0.015188	...	1.000000	-0.052087	0.582642	0.306043	0.042470	0.558214	0.312844	-0.030
Q 28	-0.172580	-0.129923	0.255846	0.331698	-0.069026	0.396051	0.573789	0.728812	0.037797	0.579217	...	-0.052087	1.000000	-0.181606	0.570820	0.815864	0.188606	0.495023	0.724
Q 29	0.706379	0.550601	0.505488	0.406586	0.756054	0.278704	-0.186565	-0.181167	0.539855	0.007247	...	0.582642	-0.181606	1.000000	0.195845	-0.098875	0.548835	0.118248	-0.142
Q 30	0.181452	0.110339	0.509330	0.447028	0.181720	0.460704	0.473245	0.520657	0.284533	0.357953	...	0.306043	0.570820	0.195845	1.000000	0.618772	0.451918	0.702871	0.521
Q 31	-0.160434	-0.134234	0.372167	0.388240	-0.052887	0.553147	0.619036	0.666529	0.094327	0.638239	...	0.042470	0.815864	-0.098875	0.618772	1.000000	0.170919	0.553498	0.732
Q 32	0.606780	0.266028	0.606943	0.571392	0.539596	0.413676	0.076680	0.118412	0.526853	0.135948	...	0.558214	0.188606	0.548835	0.451918	0.170919	1.000000	0.481921	0.107
Q 33	0.225443	-0.047341	0.475279	0.422799	0.195562	0.413179	0.421652	0.419423	0.393403	0.441113	...	0.312844	0.495023	0.118248	0.702871	0.553498	0.481921	1.000000	0.386
Q 34	-0.100133	-0.097992	0.297167	0.303419	-0.057710	0.442916	0.571728	0.668646	-0.004165	0.527677	...	-0.030077	0.724151	-0.142238	0.521452	0.732762	0.107653	0.386140	1.000
Q 35	0.351244	0.164994	0.501885	0.467918	0.256214	0.418414	0.297134	0.282779	0.405021	0.416696	...	0.369089	0.382746	0.310377	0.539953	0.445986	0.534646	0.595249	0.433
Q 36	0.313080	0.176416	0.543733	0.455045	0.284927	0.478940	0.275859	0.302398	0.360979	0.394537	...	0.410087	0.322371	0.305811	0.576429	0.408650	0.506619	0.619911	0.414

36 rows × 36 columns

In [8]:
```python
# The eigenvalues are calculated for the third cluster

fa_1 = FactorAnalyzer(n_factors=3, rotation='oblimin').fit(cluster_5)
fa_1.loadings_

eigenvalues, _ = fa_1.get_eigenvalues()
eigen_values = list(eigenvalues)

data_ev = pd.DataFrame(eigen_values,columns=['Eigenvalues'])

data_ev

# In this case, since the first 7 eigenvalues are greater than 1, the no. of factors is take to be 7.
```

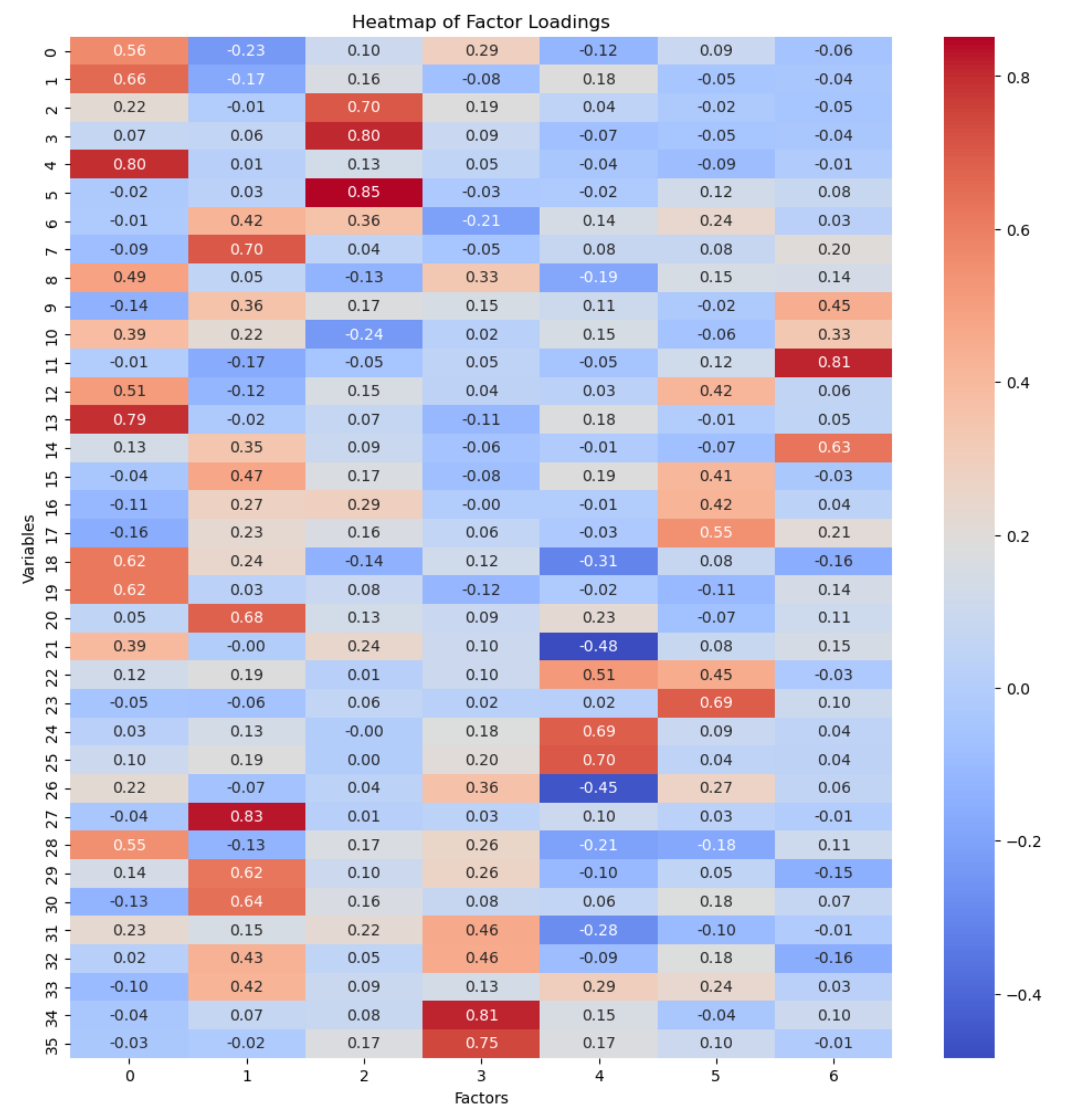
Out[8]:

	Eigenvalues
0	11.622003
1	7.266139
2	1.990538
3	1.758093
4	1.429721
5	1.245185
6	1.065921
7	0.996036
8	0.877576
9	0.711040
10	0.622060
11	0.573227
12	0.535024
13	0.506471
14	0.461963
15	0.410649
16	0.389377
17	0.372894
18	0.332059
19	0.323559
20	0.287605
21	0.258200
22	0.233161
23	0.225169
24	0.215713
25	0.178115
26	0.164256
27	0.162852
28	0.148412
29	0.119574
30	0.113307
31	0.105637
32	0.089093
33	0.078384
34	0.075462
35	0.055524

In [9]:
```python
# The factor loadings for the 7 Factors are presented in the form of a heatmap:

fa_2 = FactorAnalyzer(n_factors=7, rotation='oblimin').fit(cluster_5)
loadings = fa_2.loadings_

plt.figure(figsize=(12, 12))
sns.heatmap(loadings, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap of Factor Loadings')
plt.xlabel('Factors')
plt.ylabel('Variables')
plt.show()
```



In [10]:
```python
corr_mat_factors_5 = pd.DataFrame(fa_2.phi_)
corr_mat_factors_5

# The correlation matrix between the 7 Factors themselves:
```

Out[10]:

	0	1	2	3	4	5	6
0	1.000000	-0.074657	0.216908	0.377905	0.301032	0.410532	0.402661
1	-0.074657	1.000000	0.090221	-0.255720	0.393148	-0.020930	0.340310
2	0.216908	0.090221	1.000000	0.104705	0.078038	0.178342	0.223915
3	0.377905	-0.255720	0.104705	1.000000	-0.061454	0.180937	0.059020
4	0.301032	0.393148	0.078038	-0.061454	1.000000	0.256808	0.423330
5	0.410532	-0.020930	0.178342	0.180937	0.256808	1.000000	0.379578
6	0.402661	0.340310	0.223915	0.059020	0.423330	0.379578	1.000000

In [11]:
```python
# Exporting to Excel

corr_mat_factors_5.to_excel('FA5_KMedoids.xlsx')
```