FA FCM

July 1, 2024

1 Factor Analysis on Fuzzy C-Means clustered data

[10]:
```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_FCM.xlsx')

clusters = list(df_2['Cluster'])

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

[10]:
```
     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  \
0      6    5    6    3    6    7    7    1    6     6  …     2     1     6
1      4    4    3    2    2    5    2    1    1     3  …     1     4     3
2      6    7    6    5    3    4    1    1    4     1  …     1     5     5
3      1    1    2    6    2    1    1    1    2     1  …     2     2     1
4      4    7    4    6    6    4    4    3    3     3  …     7     1     7
..   …  …  …  …  …  …  …  …  …   …  …   …   …   …
476    5    1    5    3    5    3    1    1    5     2  …     1     5     2
477    6    7    5    5    6    5    1    1    5     1  …     1     6     2
478    2    4    3    3    6    2    1    1    4     2  …     1     5     2
479    5    7    5    5    7    3    2    1    5     1  …     2     6     2
480    6    1    3    4    4    2    2    1    4     2  …     1     4     2

     Q 31  Q 32  Q 33  Q 34  Q 35  Q 36  Cluster
0       2     2     6     7     2     4        0
1       4     3     4     3     3     3        1
```

```
2        1     6     6     1     6     6        0
3        1     1     2     1     2     2        1
4        1     3     1     3     1     3        0
..   ...   ...   ...   ...   ...   ...     ...
476      1     5     3     1     3     3        0
477      1     5     1     1     2     2        0
478      1     2     2     1     1     1        1
479      1     5     2     1     3     2        0
480      1     4     2     1     1     1        0

[481 rows x 37 columns]
```

1.1 Factor Analysis on 1st cluster

[11]:
```python
# Only the rows beliongging to the 1st cluster are filtered out.

cluster_1 = df.loc[df['Cluster']==0]
cluster_1
```

[11]:
```
     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  \
0      6    5    6    3    6    7    7    1    6     6  ...     2     1     6
2      6    7    6    5    3    4    1    1    4     1  ...     1     5     5
4      4    7    4    6    6    4    4    3    3     3  ...     7     1     7
5      4    4    6    6    5    7    5    6    5     6  ...     7     6     6
9      2    3    5    6    5    4    2    6    2     3  ...     4     5     5
..   ...  ...  ...  ...  ...  ...  ...  ...  ...   ...  ...   ...   ...   ...
475    6    7    1    1    6    1    1    1    6     1  ...     1     5     3
476    5    1    5    3    5    3    1    1    5     2  ...     1     5     2
477    6    7    5    5    6    5    1    1    5     1  ...     1     6     2
479    5    7    5    5    7    3    2    1    5     1  ...     2     6     2
480    6    1    3    4    4    2    2    1    4     2  ...     1     4     2

     Q 31  Q 32  Q 33  Q 34  Q 35  Q 36  Cluster
0       2     2     6     7     2     4        0
2       1     6     6     1     6     6        0
4       1     3     1     3     1     3        0
5       5     5     5     6     6     6        0
9       6     5     2     2     2     3        0
..    ...   ...   ...   ...   ...   ...      ...
475     1     4     4     1     3     3        0
476     1     5     3     1     3     3        0
477     1     5     1     1     2     2        0
479     1     5     2     1     3     2        0
480     1     4     2     1     1     1        0

[272 rows x 37 columns]
```

[12]:
```python
cluster_1 = cluster_1.drop(columns=['Cluster'])
cluster_1
```

[12]:
```
     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  \
0      6    5    6    3    6    7    7    1    6     6  …     6     2     1
2      6    7    6    5    3    4    1    1    4     1  …     5     1     5
4      4    7    4    6    6    4    4    3    3     3  …     3     7     1
5      4    4    6    6    5    7    5    6    5     6  …     5     7     6
9      2    3    5    6    5    4    2    6    2     3  …     6     4     5
..   …  …  …  …  …  …  …  …  …   …  …   …   …   …
475    6    7    1    1    6    1    1    1    6     1  …     6     1     5
476    5    1    5    3    5    3    1    1    5     2  …     5     1     5
477    6    7    5    5    6    5    1    1    5     1  …     6     1     6
479    5    7    5    5    7    3    2    1    5     1  …     5     2     6
480    6    1    3    4    4    2    2    1    4     2  …     5     1     4

     Q 30  Q 31  Q 32  Q 33  Q 34  Q 35  Q 36
0       6     2     2     6     7     2     4
2       5     1     6     6     1     6     6
4       7     1     3     1     3     1     3
5       6     5     5     5     6     6     6
9       5     6     5     2     2     2     3
..    …   …   …   …   …   …   …
475     3     1     4     4     1     3     3
476     2     1     5     3     1     3     3
477     2     1     5     1     1     2     2
479     2     1     5     2     1     3     2
480     2     1     4     2     1     1     1

[272 rows x 36 columns]
```

[13]:
```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_1)

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: [0.9209071  0.90156302 0.94753082 0.94008835 0.92498357
0.92647844
 0.93353923 0.9355644  0.94176582 0.94386255 0.74446973 0.7455008
 0.93694343 0.91465609 0.88678158 0.94873599 0.91060643 0.9423133
 0.95020049 0.90180435 0.9355367  0.91234301 0.89662411 0.86126901
```

```
 0.82508106 0.8576399  0.95164895 0.93848302 0.93043655 0.93244785
 0.91306673 0.93764588 0.9428565  0.95169303 0.89544991 0.88988072]
Overall KMO: 0.9187187783680684
```

[14]:
```python
# The original correlation matrix for the first cluster:
cluster_1.corr()
```

[14]:
```
            Q 1       Q 2       Q 3       Q 4       Q 5       Q 6       Q 7  \
Q 1   1.000000  0.611145  0.536439  0.451371  0.662514  0.353351  0.000252
Q 2   0.611145  1.000000  0.413283  0.353525  0.543867  0.328198  0.027848
Q 3   0.536439  0.413283  1.000000  0.715246  0.536924  0.691026  0.405227
Q 4   0.451371  0.353525  0.715246  1.000000  0.500011  0.707062  0.354751
Q 5   0.662514  0.543867  0.536924  0.500011  1.000000  0.448158  0.086396
Q 6   0.353351  0.328198  0.691026  0.707062  0.448158  1.000000  0.460062
Q 7   0.000252  0.027848  0.405227  0.354751  0.086396  0.460062  1.000000
Q 8  -0.047011 -0.020523  0.281511  0.358614  0.082219  0.389183  0.439656
Q 9   0.487837  0.265413  0.447333  0.389971  0.525932  0.309269  0.141359
Q 10  0.034854 -0.015442  0.320118  0.309839  0.121921  0.449345  0.500136
Q 11  0.180783  0.215693  0.193080  0.130597  0.205732  0.124380  0.362856
Q 12  0.185422  0.050522  0.213670  0.143700  0.181887  0.157220  0.030079
Q 13  0.476476  0.328260  0.455396  0.360316  0.439321  0.395124  0.204253
Q 14  0.406413  0.419068  0.383940  0.288649  0.520380  0.257571  0.097345
Q 15  0.239816  0.119858  0.373019  0.356097  0.300512  0.353061  0.245102
Q 16 -0.000034  0.044639  0.341406  0.390950  0.128343  0.524097  0.579709
Q 17  0.165137  0.044844  0.489949  0.414838  0.090047  0.539121  0.418637
Q 18  0.150894  0.057754  0.400165  0.414910  0.135882  0.514529  0.462670
Q 19  0.521765  0.449162  0.443088  0.390627  0.633370  0.382155  0.148532
Q 20  0.364396  0.436525  0.319481  0.316979  0.506534  0.256201  0.044972
Q 21 -0.075981 -0.049475  0.330520  0.404021  0.134105  0.404023  0.478368
Q 22  0.503304  0.309067  0.452856  0.409219  0.582057  0.439330  0.094245
Q 23  0.090327  0.125830  0.374209  0.340278  0.212422  0.406144  0.439989
Q 24  0.109388  0.094408  0.269515  0.257995  0.106628  0.353892  0.307217
Q 25 -0.102258 -0.056793  0.132058  0.146975  0.065595  0.242990  0.280408
Q 26 -0.059758  0.009209  0.200802  0.197502  0.066270  0.252892  0.336340
Q 27  0.570397  0.357852  0.499664  0.437650  0.573452  0.391571  0.138374
Q 28 -0.081404 -0.032421  0.252108  0.316663  0.119412  0.416676  0.491267
Q 29  0.558160  0.419067  0.449959  0.438765  0.636853  0.348811 -0.059332
Q 30  0.232873  0.258507  0.443426  0.473901  0.314026  0.526931  0.383202
Q 31  0.047346 -0.024348  0.442967  0.390839  0.133832  0.558858  0.549415
Q 32  0.521104  0.362997  0.533495  0.530099  0.585191  0.459213  0.179548
Q 33  0.283219  0.115358  0.503604  0.447520  0.365067  0.476985  0.339434
Q 34  0.022674  0.005051  0.348834  0.347117  0.138527  0.465901  0.540933
Q 35  0.278414  0.219473  0.428476  0.445734  0.337660  0.423777  0.324043
Q 36  0.283216  0.251638  0.450620  0.473326  0.381210  0.479602  0.309350

            Q 8       Q 9      Q 10  …      Q 27      Q 28      Q 29  \
Q 1  -0.047011  0.487837  0.034854  …  0.570397 -0.081404  0.558160
```

Q 2	-0.020523	0.265413	-0.015442	…	0.357852	-0.032421	0.419067
Q 3	0.281511	0.447333	0.320118	…	0.499664	0.252108	0.449959
Q 4	0.358614	0.389971	0.309839	…	0.437650	0.316663	0.438765
Q 5	0.082219	0.525932	0.121921	…	0.573452	0.119412	0.636853
Q 6	0.389183	0.309269	0.449345	…	0.391571	0.416676	0.348811
Q 7	0.439656	0.141359	0.500136	…	0.138374	0.491267	-0.059332
Q 8	1.000000	0.086730	0.486507	…	0.084398	0.647102	-0.051429
Q 9	0.086730	1.000000	0.196706	…	0.600069	0.127604	0.547635
Q 10	0.486507	0.196706	1.000000	…	0.165609	0.492961	0.039407
Q 11	0.212141	0.213452	0.273222	…	0.198764	0.191859	0.071537
Q 12	0.032188	0.419082	0.197787	…	0.247379	0.025237	0.361952
Q 13	0.129675	0.412501	0.282676	…	0.376145	0.137838	0.353035
Q 14	0.015226	0.368821	0.086993	…	0.353460	0.075813	0.408340
Q 15	0.314183	0.508604	0.305493	…	0.282564	0.281625	0.403726
Q 16	0.605040	0.076600	0.462014	…	0.113052	0.602478	-0.084348
Q 17	0.374738	0.193188	0.453891	…	0.201244	0.370706	0.103546
Q 18	0.504119	0.180906	0.500673	…	0.206589	0.466555	0.090975
Q 19	0.028743	0.487013	0.150043	…	0.568412	0.136564	0.544730
Q 20	0.105435	0.350382	0.024540	…	0.364816	0.138965	0.417698
Q 21	0.590250	0.248852	0.460124	…	0.100385	0.680979	0.058024
Q 22	-0.040219	0.583014	0.173864	…	0.545837	0.108048	0.602924
Q 23	0.554178	0.166971	0.359277	…	0.137839	0.606417	0.026613
Q 24	0.297322	0.175562	0.253465	…	0.228123	0.286945	0.049127
Q 25	0.317492	0.186282	0.321592	…	0.001322	0.460418	0.015489
Q 26	0.362839	0.241233	0.338156	…	0.039305	0.426197	0.027852
Q 27	0.084398	0.600069	0.165609	…	1.000000	0.073653	0.577117
Q 28	0.647102	0.127604	0.492961	…	0.073653	1.000000	-0.051564
Q 29	-0.051429	0.547635	0.039407	…	0.577117	-0.051564	1.000000
Q 30	0.431348	0.311424	0.366580	…	0.353201	0.514693	0.199340
Q 31	0.559332	0.195949	0.520931	…	0.180885	0.643417	0.071820
Q 32	0.178810	0.459202	0.178996	…	0.579147	0.239603	0.548554
Q 33	0.387183	0.447906	0.435710	…	0.402109	0.406421	0.313973
Q 34	0.613860	0.116209	0.500898	…	0.139256	0.626621	0.008776
Q 35	0.370329	0.420070	0.353483	…	0.434674	0.383426	0.345044
Q 36	0.367189	0.393750	0.320406	…	0.453896	0.337846	0.307016

	Q 30	Q 31	Q 32	Q 33	Q 34	Q 35	Q 36
Q 1	0.232873	0.047346	0.521104	0.283219	0.022674	0.278414	0.283216
Q 2	0.258507	-0.024348	0.362997	0.115358	0.005051	0.219473	0.251638
Q 3	0.443426	0.442967	0.533495	0.503604	0.348834	0.428476	0.450620
Q 4	0.473901	0.390839	0.530099	0.447520	0.347117	0.445734	0.473326
Q 5	0.314026	0.133832	0.585191	0.365067	0.138527	0.337660	0.381210
Q 6	0.526931	0.558858	0.459213	0.476985	0.465901	0.423777	0.479602
Q 7	0.383202	0.549415	0.179548	0.339434	0.540933	0.324043	0.309350
Q 8	0.431348	0.559332	0.178810	0.387183	0.613860	0.370329	0.367189
Q 9	0.311424	0.195949	0.459202	0.447906	0.116209	0.420070	0.393750
Q 10	0.366580	0.520931	0.178996	0.435710	0.500898	0.353483	0.320406

Q 11	0.202617	0.184417	0.067874	0.101374	0.218493	0.218662	0.190363
Q 12	-0.012264	0.134203	0.059190	0.221357	0.000363	0.084114	0.066342
Q 13	0.272522	0.239403	0.342241	0.331059	0.223206	0.209816	0.278121
Q 14	0.233606	0.019556	0.388472	0.168059	0.046302	0.214935	0.207655
Q 15	0.204311	0.409715	0.246837	0.359876	0.221097	0.243500	0.243028
Q 16	0.497697	0.599326	0.238612	0.444502	0.635752	0.366259	0.388694
Q 17	0.343758	0.584937	0.215246	0.470166	0.386221	0.249990	0.285796
Q 18	0.321659	0.646825	0.251895	0.359516	0.499154	0.304668	0.333889
Q 19	0.367164	0.108597	0.536262	0.307095	0.131796	0.301469	0.306740
Q 20	0.324278	0.016104	0.478513	0.198566	0.051913	0.265732	0.304001
Q 21	0.472606	0.646747	0.268345	0.480065	0.565243	0.475993	0.435430
Q 22	0.270894	0.130501	0.485256	0.431511	0.077598	0.291844	0.289428
Q 23	0.448688	0.578575	0.228520	0.388305	0.655929	0.421319	0.445235
Q 24	0.164589	0.310621	0.214018	0.211288	0.334914	0.215418	0.236413
Q 25	0.172723	0.304573	0.094600	0.340969	0.384789	0.292309	0.295519
Q 26	0.245902	0.371494	0.183001	0.303099	0.337777	0.354591	0.398114
Q 27	0.353201	0.180885	0.579147	0.402109	0.139256	0.434674	0.453896
Q 28	0.514693	0.643417	0.239603	0.406421	0.626621	0.383426	0.337846
Q 29	0.199340	0.071820	0.548554	0.313973	0.008776	0.345044	0.307016
Q 30	1.000000	0.473593	0.411172	0.541542	0.523889	0.520047	0.568911
Q 31	0.473593	1.000000	0.264281	0.515438	0.632747	0.427123	0.396445
Q 32	0.411172	0.264281	1.000000	0.458386	0.229827	0.510706	0.475944
Q 33	0.541542	0.515438	0.458386	1.000000	0.464997	0.583451	0.561120
Q 34	0.523889	0.632747	0.229827	0.464997	1.000000	0.496438	0.513626
Q 35	0.520047	0.427123	0.510706	0.583451	0.496438	1.000000	0.816932
Q 36	0.568911	0.396445	0.475944	0.561120	0.513626	0.816932	1.000000

[36 rows x 36 columns]

[15]:
```
# The eigenvalues are calculated for the first cluster

fa_1 = FactorAnalyzer(n_factors=3, rotation='oblimin').fit(cluster_1)
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 6 eigenvalues are greater than 1, the no. of␣
↪factors is take to be 6.
```

[15]:

	Eigenvalues
0	11.999670
1	5.451338

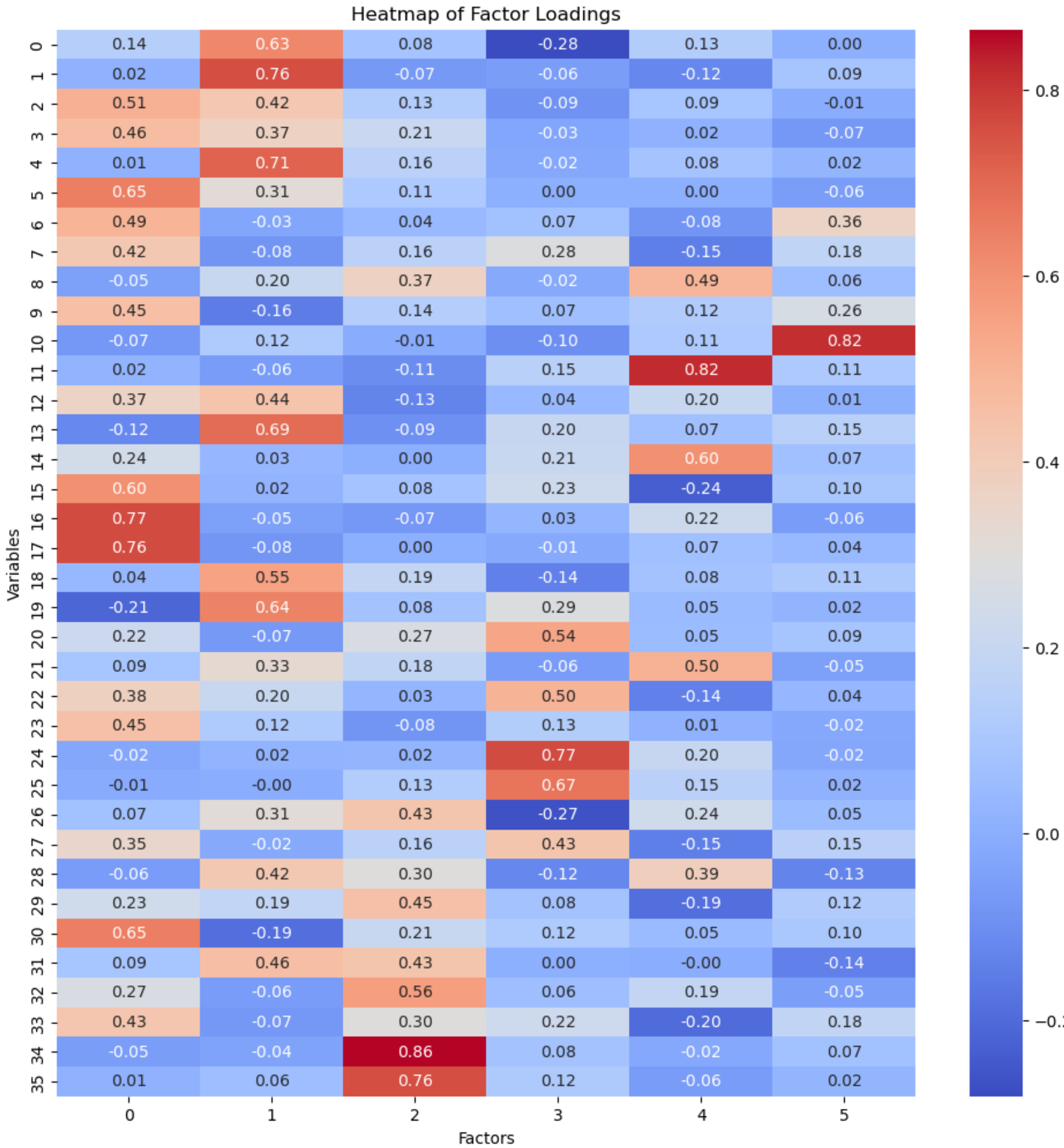
```
2       2.052597
3       1.743498
4       1.359914
5       1.156193
6       0.979314
7       0.908193
8       0.814096
9       0.724375
10      0.622492
11      0.605617
12      0.565693
13      0.533879
14      0.506021
15      0.472187
16      0.443909
17      0.435632
18      0.393177
19      0.382287
20      0.354526
21      0.325318
22      0.311970
23      0.309090
24      0.295990
25      0.273204
26      0.262555
27      0.261823
28      0.232045
29      0.225362
30      0.200658
31      0.187232
32      0.178389
33      0.167315
34      0.137778
35      0.126666
```

[17]:

```
# The factor loadings for the 6 Factors are presented in the form of a heatmap:

fa_2 = FactorAnalyzer(n_factors=6, rotation='oblimin').fit(cluster_1)
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()
```



[19]:
```
corr_mat_factors_1 = pd.DataFrame(fa_2.phi_)
corr_mat_factors_1

# The correlation matrix between the 6 Factors themselves:
```

[19]:

	0	1	2	3	4	5
0	1.000000	0.161755	0.103818	0.364421	0.463131	0.261458
1	0.161755	1.000000	0.340573	-0.108607	0.427321	0.072123
2	0.103818	0.340573	1.000000	0.020298	0.170984	0.037264
3	0.364421	-0.108607	0.020298	1.000000	0.271748	0.308157
4	0.463131	0.427321	0.170984	0.271748	1.000000	0.212738

```
5  0.261458  0.072123  0.037264  0.308157  0.212738  1.000000
```

1.2 Factor Analysis on 2nd cluster

[20]:
```python
# Only the rows beliongging to the 2nd cluster are filtered out.

cluster_2 = df.loc[df['Cluster']==1]
cluster_2
```

[20]:
```
     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  \
1      4    4    3    2    2    5    2    1    1     3  …     1     4     3
3      1    1    2    6    2    1    1    1    2     1  …     2     2     1
6      6    7    6    5    3    4    1    1    4     1  …     1     5     5
7      1    2    1    1    2    6    3    1    2     1  …     1     3     1
8      7    6    7    3    3    3    1    4    6     7  …     1     1     1
..   …    …    …    …    …    …    …    …    …     …   …     …     …     …
467    6    6    5    5    6    6    2    1    5     1  …     1     5     2
468    5    7    1    1    6    1    1    1    6     1  …     1     5     3
469    6    6    5    5    5    5    1    1    5     1  …     1     5     2
472    3    3    3    3    6    2    1    1    4     2  …     1     6     2
478    2    4    3    3    6    2    1    1    4     2  …     1     5     2

     Q 31  Q 32  Q 33  Q 34  Q 35  Q 36  Cluster
1       4     3     4     3     3     3        1
3       1     1     2     1     2     2        1
6       1     6     6     1     6     6        1
7       3     4     2     1     2     6        1
8       4     1     3     1     1     1        1
..    …     …     …     …     …     …        …
467     1     5     1     1     1     1        1
468     1     4     4     1     3     2        1
469     1     5     1     1     1     1        1
472     1     1     1     1     1     1        1
478     1     2     2     1     1     1        1

[209 rows x 37 columns]
```

[21]:
```python
cluster_2 = cluster_2.drop(columns=['Cluster'])
cluster_2
```

[21]:
```
     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  \
1      4    4    3    2    2    5    2    1    1     3  …     3     1     4
3      1    1    2    6    2    1    1    1    2     1  …     1     2     2
6      6    7    6    5    3    4    1    1    4     1  …     5     1     5
7      1    2    1    1    2    6    3    1    2     1  …     2     1     3
8      7    6    7    3    3    3    1    4    6     7  …     7     1     1
..   …    …    …    …    …    …    …    …    …     …   …     …     …     …
```

```
467      6    6    5    5    6    6    2    1    5     1  …     5     1     5
468      5    7    1    1    6    1    1    1    6     1  …     5     1     5
469      6    6    5    5    5    5    1    1    5     1  …     5     1     5
472      3    3    3    3    6    2    1    1    4     2  …     2     1     6
478      2    4    3    3    6    2    1    1    4     2  …     4     1     5

     Q 30  Q 31  Q 32  Q 33  Q 34  Q 35  Q 36
1       3     4     3     4     3     3     3
3       1     1     1     2     1     2     2
6       5     1     6     6     1     6     6
7       1     3     4     2     1     2     6
8       1     4     1     3     1     1     1
..    …   …   …   …   …   …   …
467     2     1     5     1     1     1     1
468     3     1     4     4     1     3     2
469     2     1     5     1     1     1     1
472     2     1     1     1     1     1     1
478     2     1     2     2     1     1     1

[209 rows x 36 columns]
```

[22]:
```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_2)

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: [0.94309423 0.92943648 0.91777595 0.92253755 0.90941347
0.92064401
 0.90907125 0.90387413 0.8969244  0.89820828 0.60400922 0.81708066
 0.91046922 0.87204114 0.8370649  0.9024438  0.87130515 0.85080907
 0.9252228  0.87729138 0.89688374 0.90291194 0.77838997 0.65444135
 0.77965894 0.80655507 0.91875317 0.89494731 0.88433582 0.92164014
 0.91200904 0.92703405 0.90904047 0.87299561 0.78257194 0.74464512]
Overall KMO: 0.884602422964242
```

[23]:
```python
# The original correlation matrix for the second cluster:
cluster_2.corr()
```

[23]:
```
           Q 1       Q 2       Q 3       Q 4       Q 5       Q 6       Q 7  \
Q 1   1.000000  0.663600  0.633670  0.553598  0.674052  0.521720  0.219445
Q 2   0.663600  1.000000  0.416016  0.358052  0.666603  0.346847  0.022351
```

Q 3	0.633670	0.416016	1.000000	0.777455	0.587803	0.746727	0.440182
Q 4	0.553598	0.358052	0.777455	1.000000	0.545317	0.704431	0.394233
Q 5	0.674052	0.666603	0.587803	0.545317	1.000000	0.521510	0.169496
Q 6	0.521720	0.346847	0.746727	0.704431	0.521510	1.000000	0.490674
Q 7	0.219445	0.022351	0.440182	0.394233	0.169496	0.490674	1.000000
Q 8	-0.046856	-0.174642	0.256140	0.247410	-0.022005	0.314592	0.444479
Q 9	0.446787	0.373032	0.415481	0.314477	0.546450	0.316199	0.126485
Q 10	0.188534	-0.000640	0.405989	0.330328	0.126488	0.403856	0.568055
Q 11	0.232745	0.249780	0.125895	0.087828	0.308769	0.185823	0.148997
Q 12	0.358511	0.244194	0.384559	0.229203	0.325784	0.260944	0.216552
Q 13	0.549005	0.492591	0.505017	0.415916	0.491249	0.383942	0.288944
Q 14	0.504183	0.561020	0.383301	0.260576	0.521995	0.255307	0.027608
Q 15	0.260733	0.218564	0.299169	0.257653	0.284999	0.268256	0.336636
Q 16	0.103022	-0.027810	0.392224	0.365109	0.135249	0.455739	0.610765
Q 17	0.207519	-0.005815	0.483844	0.347920	0.038450	0.443082	0.489883
Q 18	0.295891	0.154781	0.468238	0.359610	0.149027	0.469672	0.626573
Q 19	0.595403	0.633812	0.417215	0.401607	0.684308	0.401021	0.177142
Q 20	0.483833	0.516244	0.348515	0.309821	0.539190	0.300315	-0.006091
Q 21	0.037861	-0.175175	0.337636	0.299649	0.082296	0.387274	0.457689
Q 22	0.628136	0.542149	0.553910	0.419286	0.656428	0.439023	0.201007
Q 23	0.060867	-0.077538	0.138102	0.150282	0.038731	0.269649	0.464715
Q 24	0.145398	0.028744	0.259344	0.277180	0.150548	0.278853	0.247921
Q 25	-0.046174	-0.171918	0.208604	0.204468	0.020573	0.301715	0.373539
Q 26	-0.010149	-0.171141	0.291695	0.230792	-0.042801	0.253018	0.377454
Q 27	0.619037	0.510946	0.555352	0.445115	0.667025	0.482844	0.233402
Q 28	-0.012257	-0.147048	0.177383	0.273734	0.057758	0.393077	0.509579
Q 29	0.596775	0.634436	0.481335	0.423831	0.765534	0.441347	-0.009410
Q 30	0.363219	0.210191	0.436648	0.449838	0.291881	0.507908	0.406674
Q 31	0.113816	-0.096695	0.408475	0.362183	0.123824	0.532045	0.619287
Q 32	0.664891	0.493549	0.534056	0.538181	0.595650	0.499759	0.248213
Q 33	0.387207	0.146386	0.515976	0.452483	0.274715	0.517962	0.420558
Q 34	-0.048263	-0.182065	0.191840	0.148838	-0.018866	0.286485	0.448094
Q 35	0.326568	0.201667	0.333205	0.340371	0.195541	0.309950	0.205632
Q 36	0.244303	0.160891	0.247748	0.270573	0.155996	0.311864	0.132176

	Q 8	Q 9	Q 10	…	Q 27	Q 28	Q 29	\
Q 1	-0.046856	0.446787	0.188534	…	0.619037	-0.012257	0.596775	
Q 2	-0.174642	0.373032	-0.000640	…	0.510946	-0.147048	0.634436	
Q 3	0.256140	0.415481	0.405989	…	0.555352	0.177383	0.481335	
Q 4	0.247410	0.314477	0.330328	…	0.445115	0.273734	0.423831	
Q 5	-0.022005	0.546450	0.126488	…	0.667025	0.057758	0.765534	
Q 6	0.314592	0.316199	0.403856	…	0.482844	0.393077	0.441347	
Q 7	0.444479	0.126485	0.568055	…	0.233402	0.509579	-0.009410	
Q 8	1.000000	0.113990	0.409876	…	0.043597	0.518140	-0.209445	
Q 9	0.113990	1.000000	0.189110	…	0.470387	0.054800	0.420043	
Q 10	0.409876	0.189110	1.000000	…	0.146641	0.375041	-0.023710	
Q 11	0.121783	0.187942	0.104118	…	0.153237	0.115992	0.215383	

Q 12	-0.013378	0.308257	0.228047	…	0.453755	-0.000621	0.356461
Q 13	0.063354	0.341491	0.203484	…	0.544536	0.021772	0.463392
Q 14	0.012996	0.353479	0.151437	…	0.522965	-0.138154	0.553745
Q 15	0.139932	0.164826	0.177011	…	0.306910	0.180042	0.299977
Q 16	0.469561	0.109763	0.512948	…	0.177354	0.519629	-0.038517
Q 17	0.442075	0.189608	0.430195	…	0.272309	0.284877	-0.014646
Q 18	0.344876	0.215018	0.568961	…	0.247730	0.349221	0.017825
Q 19	0.045804	0.448068	0.150501	…	0.577337	0.109077	0.587084
Q 20	-0.128755	0.469444	0.079628	…	0.492039	-0.054534	0.569179
Q 21	0.444776	0.107926	0.490141	…	0.156063	0.485991	-0.020144
Q 22	0.045424	0.499036	0.192754	…	0.676904	0.018418	0.603587
Q 23	0.245635	0.053032	0.389028	…	0.108780	0.396642	0.024896
Q 24	0.146686	0.141516	0.365874	…	0.167964	0.223394	0.138933
Q 25	0.420099	0.140278	0.288449	…	-0.041421	0.467799	-0.093557
Q 26	0.408893	0.071415	0.376735	…	-0.048876	0.378639	-0.158957
Q 27	0.043597	0.470387	0.146641	…	1.000000	0.012062	0.573034
Q 28	0.518140	0.054800	0.375041	…	0.012062	1.000000	-0.127604
Q 29	-0.209445	0.420043	-0.023710	…	0.573034	-0.127604	1.000000
Q 30	0.331305	0.258582	0.352843	…	0.331438	0.443376	0.255193
Q 31	0.492946	0.093133	0.616452	…	0.176180	0.541338	-0.005700
Q 32	0.157062	0.450537	0.142586	…	0.507567	0.124427	0.491115
Q 33	0.361602	0.213170	0.319188	…	0.405708	0.341123	0.214299
Q 34	0.479293	0.079905	0.347568	…	0.053656	0.506504	-0.119884
Q 35	0.159038	0.120347	0.176378	…	0.221877	0.207969	0.198892
Q 36	0.163307	0.064312	0.103664	…	0.170025	0.226744	0.146438

	Q 30	Q 31	Q 32	Q 33	Q 34	Q 35	Q 36
Q 1	0.363219	0.113816	0.664891	0.387207	-0.048263	0.326568	0.244303
Q 2	0.210191	-0.096695	0.493549	0.146386	-0.182065	0.201667	0.160891
Q 3	0.436648	0.408475	0.534056	0.515976	0.191840	0.333205	0.247748
Q 4	0.449838	0.362183	0.538181	0.452483	0.148838	0.340371	0.270573
Q 5	0.291881	0.123824	0.595650	0.274715	-0.018866	0.195541	0.155996
Q 6	0.507908	0.532045	0.499759	0.517962	0.286485	0.309950	0.311864
Q 7	0.406674	0.619287	0.248213	0.420558	0.448094	0.205632	0.132176
Q 8	0.331305	0.492946	0.157062	0.361602	0.479293	0.159038	0.163307
Q 9	0.258582	0.093133	0.450537	0.213170	0.079905	0.120347	0.064312
Q 10	0.352843	0.616452	0.142586	0.319188	0.347568	0.176378	0.103664
Q 11	0.064791	-0.013774	0.089909	-0.000026	0.188060	-0.048048	-0.024752
Q 12	-0.029048	0.139201	0.201320	0.210424	-0.085255	-0.003662	-0.034759
Q 13	0.300112	0.113650	0.380309	0.219845	0.064827	0.239005	0.190426
Q 14	0.211995	0.027199	0.336528	0.176791	-0.075447	0.076733	0.051100
Q 15	0.244516	0.339622	0.250813	0.245738	0.156494	0.025402	-0.029786
Q 16	0.394348	0.637700	0.177790	0.434433	0.500290	0.152684	0.078020
Q 17	0.352930	0.566202	0.303776	0.590553	0.314986	0.251926	0.212272
Q 18	0.373134	0.589917	0.297900	0.468913	0.331426	0.211714	0.106380
Q 19	0.337943	0.153702	0.500834	0.242763	-0.056973	0.172333	0.131338
Q 20	0.337422	-0.016829	0.379747	0.173862	0.006409	0.184373	0.184597

```
Q 21  0.384850  0.624081  0.144991  0.520480  0.429350  0.184209  0.125008
Q 22  0.247513  0.228598  0.492792  0.454202 -0.001564  0.155867  0.099086
Q 23  0.383424  0.354379  0.053047  0.175478  0.377501  0.220353  0.173677
Q 24  0.190463  0.228432  0.055723  0.127162  0.227534  0.114646  0.072635
Q 25  0.178632  0.418250  0.061861  0.260788  0.292542  0.072564  0.071009
Q 26  0.185908  0.451406  0.044069  0.274714  0.325338  0.142734  0.086074
Q 27  0.331438  0.176180  0.507567  0.405708  0.053656  0.221877  0.170025
Q 28  0.443376  0.541338  0.124427  0.341123  0.506504  0.207969  0.226744
Q 29  0.255193 -0.005700  0.491115  0.214299 -0.119884  0.198892  0.146438
Q 30  1.000000  0.428069  0.459752  0.581296  0.355662  0.476677  0.433052
Q 31  0.428069  1.000000  0.209984  0.566541  0.471261  0.263279  0.181276
Q 32  0.459752  0.209984  1.000000  0.522086  0.105222  0.450865  0.449879
Q 33  0.581296  0.566541  0.522086  1.000000  0.362027  0.525427  0.446435
Q 34  0.355662  0.471261  0.105222  0.362027  1.000000  0.300549  0.305696
Q 35  0.476677  0.263279  0.450865  0.525427  0.300549  1.000000  0.793353
Q 36  0.433052  0.181276  0.449879  0.446435  0.305696  0.793353  1.000000

[36 rows x 36 columns]
```

[24]:
```python
# The eigenvalues are calculated for the second cluster

fa_3 = FactorAnalyzer(n_factors=3, rotation='oblimin').fit(cluster_2)
fa_3.loadings_

eigenvalues_2, _ = fa_3.get_eigenvalues()
eigen_values_2 = list(eigenvalues_2)

data_ev = pd.DataFrame(eigen_values_2,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.
```

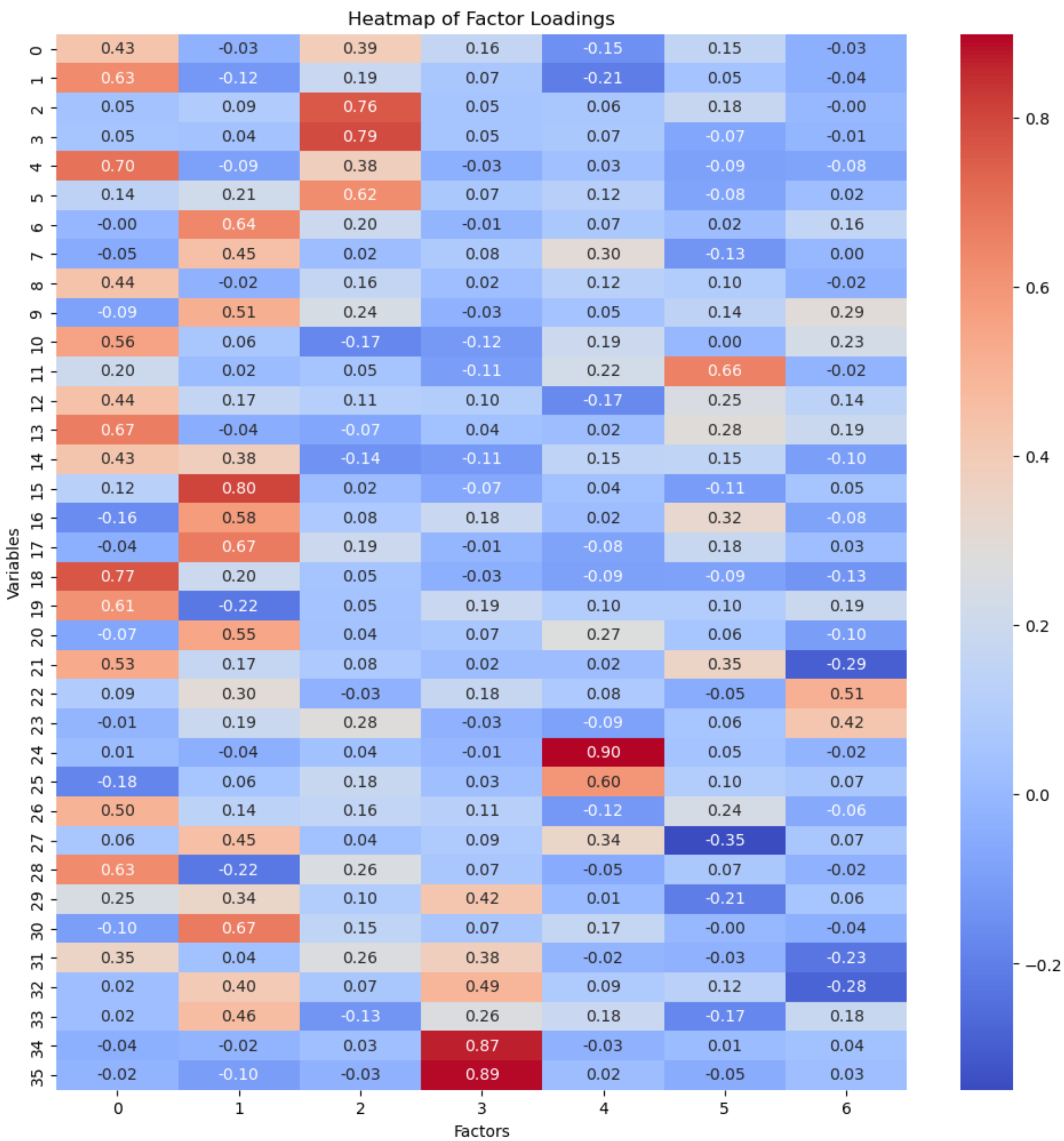
[24]:
```
    Eigenvalues
0     11.022999
1      5.991609
2      2.315060
3      1.612728
4      1.344734
5      1.154913
6      1.101420
7      0.932159
8      0.888823
9      0.794088
10     0.751821
11     0.685319
```

```
12    0.614577
13    0.591640
14    0.562871
15    0.517178
16    0.453080
17    0.425186
18    0.390834
19    0.355367
20    0.344035
21    0.323905
22    0.311847
23    0.289371
24    0.266482
25    0.255437
26    0.232682
27    0.224158
28    0.197623
29    0.184053
30    0.182647
31    0.168328
32    0.152869
33    0.138290
34    0.122785
35    0.099082
```

[25]:
```
# The factor loadings for the 7 Factors are presented in the form of a heatmap:

fa_4 = FactorAnalyzer(n_factors=7, rotation='oblimin').fit(cluster_2)
loadings_2 = fa_4.loadings_

plt.figure(figsize=(12, 12))
sns.heatmap(loadings_2, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap of Factor Loadings')
plt.xlabel('Factors')
plt.ylabel('Variables')
plt.show()
```



[26]:
```
corr_mat_factors_2 = pd.DataFrame(fa_4.phi_)
corr_mat_factors_2

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

[26]:

	0	1	2	3	4	5	6
0	1.000000	0.062777	0.194989	-0.066767	-0.053682	0.289561	0.482286
1	0.062777	1.000000	0.267026	0.080565	0.430410	0.090928	0.362779
2	0.194989	0.267026	1.000000	-0.019520	0.110874	0.007209	0.391672
3	-0.066767	0.080565	-0.019520	1.000000	0.136942	-0.113843	-0.088577
4	-0.053682	0.430410	0.110874	0.136942	1.000000	0.007220	0.126321

```
5  0.289561  0.090928  0.007209 -0.113843  0.007220  1.000000  0.254751
6  0.482286  0.362779  0.391672 -0.088577  0.126321  0.254751  1.000000
```

[27]:
```python
# Exporting the relevant data to excel:

corr_mat_factors_1.to_excel('FA_1_FCM.xlsx')
corr_mat_factors_2.to_excel('FA_2_FCM.xlsx')
```