

# DimensionReductionWithFAMD

May 4, 2023

```
[45]: #package prince https://github.com/MaxHalford/prince
      #FAMD Factor Analysis of Mixed Data
      #For mixture of categorical and numerical data
```

```
[4]: #Importing the necessary package
import math
import pandas as pd
import numpy as np
from prince import FAMD#Dataset preparation with mixed numerical and categorical
    ↳features
df = pd.read_csv('SurveyAnswers.csv')
```

```
[5]: df=df.set_index('CDR_name')
df.drop(['commercial','opensource'],axis=1,inplace=True)
df
```

```
[5]:      rest_api  another_api  gui  api_n  gui_n  aql  sql  json_xml  \
CDR_name
ehrbase      y             y  y    4.00    0  y   n             2
better       y             y  y    4.00    3  y   n             2
base24       y             y  y    4.00    4  n   n             2
cabo         y             y  y    2.91    3  n   n             2
arenaehr     y             n  y    4.00    0  y   n             2
eweave       n             n  y    0.00    3  y   y             0
ehrcare      y             y  n    3.33    0  y   n             1
clever       y             y  y    4.00    3  n   y             0
ehrdb        y             y  y    4.00    3  y   n             2
rhp          y             y  y    4.00    3  n   y             0
ehrn         y             n  y    3.00    2  y   y             0

      flat_struct  opt_wt  openehrextr_fhnr  add_openehr  add_others
CDR_name
ehrbase          2      2                1            1            1
better           2      2                1            2            4
base24           0      1                1            1            2
cabo             0      1                1            0            0
arenaehr         1      2                1            2            2
```

eweave	0	0	0	1	2
ehrcare	1	1	0	1	1
clever	1	1	1	2	3
ehrdbr	2	1	1	1	3
rhpr	1	1	0	2	4
ehrn	1	1	1	0	1

```
[6]: famd = FAMD(n_components = 2, n_iter = 3, random_state = 101,engine="sklearn")
famd=famd.fit(df)
df_famd = famd.transform(df)
df_famd
```

```
[6]:          0          1
CDR_name
ehrbaser -1.355253 -0.847850
better -1.320558  0.241814
base24 -0.356374  0.962018
cabo -0.102387  0.257373
arenaahr -0.806795 -0.965018
eweave  3.262109 -0.663478
ehrcare -0.209372 -1.446912
clever  0.249242  1.496386
ehrdbr -0.871460  0.025389
rhpr  0.553719  1.558004
ehrn  0.957130 -0.617726
```

```
[7]: famd.explained_inertia_ #variance explained
```

```
[7]: array([0.34520031, 0.20760006])
```

```
[8]: round(sum(famd.explained_inertia_)*100,1)
```

```
[8]: 55.3
```

```
[9]: famd.eigenvalues_
```

```
[9]: array([0.14130948, 0.08498213])
```

```
[10]: import plotly.express as px
fig=px.scatter(df_famd,x=0,y=1,color=df.index)
fig.show()
```

```
[11]: famd = FAMD(n_components = 3, n_iter = 3, random_state = 101,engine="sklearn")
famd = famd.fit(df)
df_famd = famd.transform(df)
df_famd
```

```
[11]:
```

	0	1	2
CDR_name			
ehrbase	-1.355253	-0.847850	-0.114822
better	-1.320558	0.241814	-0.978906
base24	-0.356374	0.962018	1.140254
cabo	-0.102387	0.257373	1.926911
arenaehr	-0.806795	-0.965018	-0.695325
eweave	3.262109	-0.663478	-0.296131
ehrcare	-0.209372	-1.446912	0.404369
clever	0.249242	1.496386	-0.398770
ehrdb	-0.871460	0.025389	-0.213718
rhph	0.553719	1.558004	-0.740811
ehrn	0.957130	-0.617726	-0.033051

```
[12]: famd.explained_inertia_ #variance explained
```

```
[12]: array([0.34520031, 0.20760006, 0.15088997])
```

```
[13]: round(sum(famd.explained_inertia_)*100,1)
```

```
[13]: 70.4
```

```
[14]: famd.eigenvalues_
```

```
[14]: array([0.14130948, 0.08498213, 0.06176757])
```

```
[15]: famd.row_coordinates(df)
```

```
[15]:
```

	0	1	2
CDR_name			
ehrbase	-1.355253	-0.847850	-0.114822
better	-1.320558	0.241814	-0.978906
base24	-0.356374	0.962018	1.140254
cabo	-0.102387	0.257373	1.926911
arenaehr	-0.806795	-0.965018	-0.695325
eweave	3.262109	-0.663478	-0.296131
ehrcare	-0.209372	-1.446912	0.404369
clever	0.249242	1.496386	-0.398770
ehrdb	-0.871460	0.025389	-0.213718
rhph	0.553719	1.558004	-0.740811
ehrn	0.957130	-0.617726	-0.033051

```
[19]: %matplotlib notebook
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D
```

```

from matplotlib import interactive, pyplot
from mpl_toolkits.mplot3d import Axes3D
from numpy.random import rand
from pylab import figure

m=rand(3,3) # m is an array of (x,y,z) coordinate triplets

fig = figure()
ax = fig.add_subplot(projection='3d')
listofloc=[]
mycolors=['r','r','b','b','r','k','r','g','r','g','r']
mymarkers=['s','s','o','o','s','v','s','^','s','^','s']

nloc=[0]*len(df_famd)
for i in range(len(df_famd)): #plot each point + its index as text above
    ax.scatter(df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.
→iloc[i,2],color=mycolors[i],marker=mymarkers[i])
    if [df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.iloc[i,2]] in listofloc:
        nloc[listofloc.index([df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.
→iloc[i,2]])]+=1
        pad=0.12*(nloc[listofloc.index([df_famd.iloc[i,0],df_famd.
→iloc[i,1],df_famd.iloc[i,2]])]-1)
    else:
        listofloc.append([df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.iloc[i,2]])
        nloc[listofloc.index([df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.
→iloc[i,2]])]=1
        pad=0
        ax.text(df_famd.iloc[i,0],df_famd.iloc[i,1],df_famd.iloc[i,2]+pad, '%s' %
→(df.index[i]), size=13, zorder=1,
        color=mycolors[i])

ax.set_xlabel(f'component 0 {round(famd.explained_inertia_[0]*100,1)}%',
→fontsize=14)
ax.set_ylabel(f'component 1 {round(famd.explained_inertia_[1]*100,1)}%',
→fontsize=14)
ax.set_zlabel(f'component 2 {round(famd.explained_inertia_[2]*100,1)}%',
→fontsize=14)

plt.show()

```

<IPython.core.display.Javascript object>

```
<IPython.core.display.HTML object>
```

```
[20]: import plotly.express as px
labels = {
    str(i): f"Comp {i+1} ({var:.1f}%)"
    for i, var in enumerate(famd.explained_inertia_ * 100)
}
mycolors=['red','red','blue','blue','red','black','red','green','red','green','red']
#mycolors=[0,0,1,1,0,2,0,3,0,3,0]
mymarkers=['circle','circle','square','square','circle','triangle-down','circle','triangle-up',
#mymarkers=[' ',' ',' ',' ',' ','_',' ','_',' ','_',' ','_']
mycolors=df_famd.index
mysymbols=df_famd.index
mycolor_discrete_map={'ehrbase':'red','better':'red','base24':'blue','cabo':
    ↳'blue','arenaehr':'red',
                        'eweave':'black','ehrcare':'red','clever':'green','ehrd':
    ↳'red','rhp':'green','ehrn':'red'}
mysymbol_map={'ehrbase':'circle','better':'circle','base24':'square','cabo':
    ↳'square','arenaehr':'circle',
                'eweave':'triangle-down','ehrcare':'circle','clever':
    ↳'triangle-up','ehrd':'circle',
                'rhp':'triangle-up','ehrn':'circle'}

fig = px.scatter_matrix(
    df_famd,
    labels=labels,
    dimensions=range(3),
    color_discrete_map=mycolor_discrete_map,
    symbol_map=mysymbol_map,
    color=mycolors,
    symbol=mysymbols
)
fig.update_traces(diagonal_visible=False)
fig.show()
```