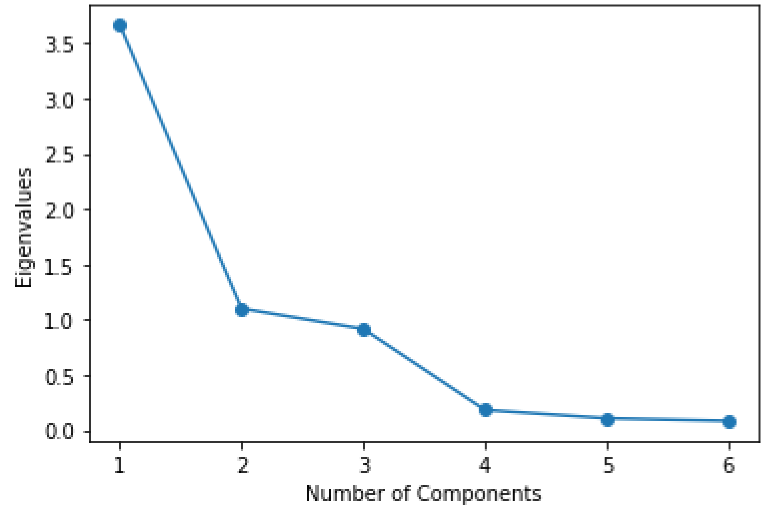
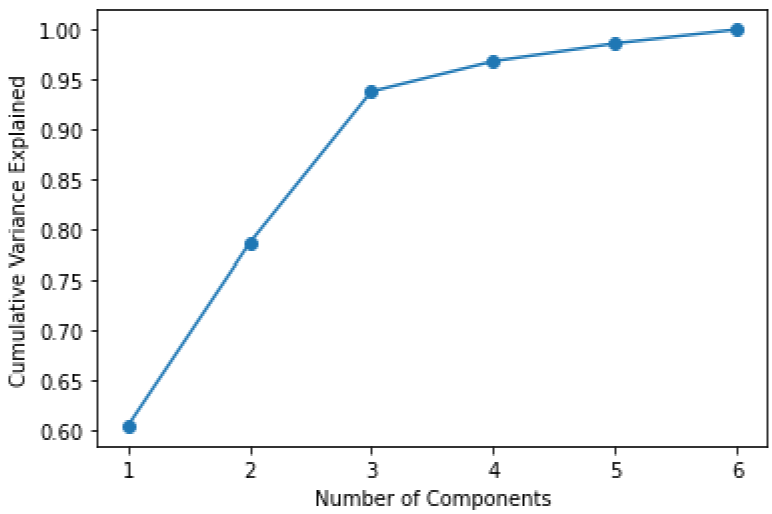
Assignment 3

a) To decide how many factors we should extract from given variables, we first performed principal component analysis on our (standardized) data and examined the eigenvalues and cumulative variance explained:





We can see that the first 3 PC components explain 95% of the variance in our data. Thus, we will run our factor model with 3 factors below.

b) After performing EFA with three factors (and no rotation) we got the following table:

	Factor1	Factor2	Factor3
SumSquared Loadings	3.513462	0.955243	0.822169
Prop Var	0.585577	0.159207	0.137028
Cum Var	0.585577	0.744784	0.881812

So, we can see that our three factors, in total, account for almost 88% of the information from the original set of six variables.

c) We got the following loading matrix in our initial model with no rotation:

	Factor1	Factor2	Factor3
X1	0.754413	0.560205	0.206392
X2	0.689907	0.550507	0.092118
X3	0.800428	-0.110269	-0.433047
X4	0.819374	-0.150528	-0.540724
X5	0.738433	-0.347792	0.292411
X6	0.781671	-0.427292	0.453509

However, using the above loading matrix, it is hard to uniquely associate factors with the original variables. This is because the factor 1 has very high correlations with all of the 6 original variables, and the factor 2 also has high correlations with the variables X1 and X2. Thus we decided to apply "varimax" rotation, and got the following loading matrix:

	Factor1	Factor2	Factor3
X1	0.186471	0.240682	0.912615
X2	0.237926	0.141489	0.843141
X3	0.839018	0.268286	0.253862
X4	0.941584	0.238376	0.207390
X5	0.286080	0.794917	0.194969
X6	0.202832	0.959848	0.191857

Now, we can see clear separation and clusters being formed, meaning that the factor 1 has high correlations with the variables X3 and X4; the factor 2 has high correlations with the variables X5 and X6; the factor 3 has high correlations with the variables X1 and X2.

d) We found the factor scores for the first two observation in our sample using our factor model (with "varimax" rotation):

	Factor1	Factor2	Factor3
0	0.037051	-0.037844	0.309890
1	-0.169629	1.854914	0.333625

We can see that in terms of the factor 3, both observations are very similar to each other, however the observation "0" has relatively high score for factor 1 compared to the observation "1", while the observation "1" has relatively higher score for factor 2 comparted to the observation "0".

Appendix

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

```
data = pd.read_csv("SIX_VARIABLES.csv")
data.head()
```

```
data.describe()
```

```
normalizer = StandardScaler()
data_scaled = normalizer.fit_transform(data)
```

```
pca = PCA()
pca.fit(data_scaled)
eigenvalues = pca.explained_variance_
cumulative_variance_explained = np.cumsum(pca.explained_variance_ratio_)
```

```
plt.plot(range(1, len(eigenvalues)+1), eigenvalues, '-o')
plt.xlabel('Number of Components')
plt.ylabel('Eigenvalues')
plt.show()
```

```
eigenvalues
```

```
array([3.66599572, 1.10231253, 0.91677737, 0.18325152, 0.10801864,
       0.08425029])
```

```
plt.figure
plt.plot(range(1, len(cumulative_variance_explained)+1), cumulative_variance_
explained, '-o')
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Variance Explained')
plt.show()
```

```
print(pca.explained_variance_ratio_)
```

```
[0.60488929 0.18188157 0.15126827 0.0302365  0.01782308 0.0139013 ]
```

```
#!pip install factor_analyzer
```

```
from factor_analyzer import FactorAnalyzer
```

Factor Analysis with No Rotation

```
FA = FactorAnalyzer(n_factors = 3, use_smc = True, rotation = None)
common_factors = FA.fit(data_scaled)
```

```
columns = ["Factor1","Factor2","Factor3"]
index = ["SumSquared Loadings","Prop Var","Cum Var"]
```

```
variance_df = pd.DataFrame(common_factors.get_factor_variance(),
                           columns = columns,
                           index = index)
variance_df

loadings = pd.DataFrame(common_factors.loadings_, columns = ["Factor1","Facto
r2","Factor3"],
                        index = data.columns)
loadings
```

Factor Analysis with Varimax Rotation

```
FA = FactorAnalyzer(n_factors = 3, use_smc = True, rotation = 'varimax')
common_factors = FA.fit(data_scaled)

variance_df = pd.DataFrame(common_factors.get_factor_variance(),
                           columns = columns,
                           index = index)
variance_df

loadings = pd.DataFrame(common_factors.loadings_, columns = columns,
                        index = data.columns)
loadings

factor_scores = FA.transform(data_scaled)
factor_scores = pd.DataFrame(factor_scores, columns = columns)
factor_scores.head()
```

Factor Analysis Using Oblique Rotation

```
FA = FactorAnalyzer(n_factors = 3, use_smc = True, rotation = 'promax')
common_factors = FA.fit(data_scaled)

variance_df = pd.DataFrame(common_factors.get_factor_variance(),
                           columns = columns,
                           index = index)
variance_df

loadings = pd.DataFrame(common_factors.loadings_, columns = columns,
                        index = data.columns)
loadings
```