資料分析方法-HW6

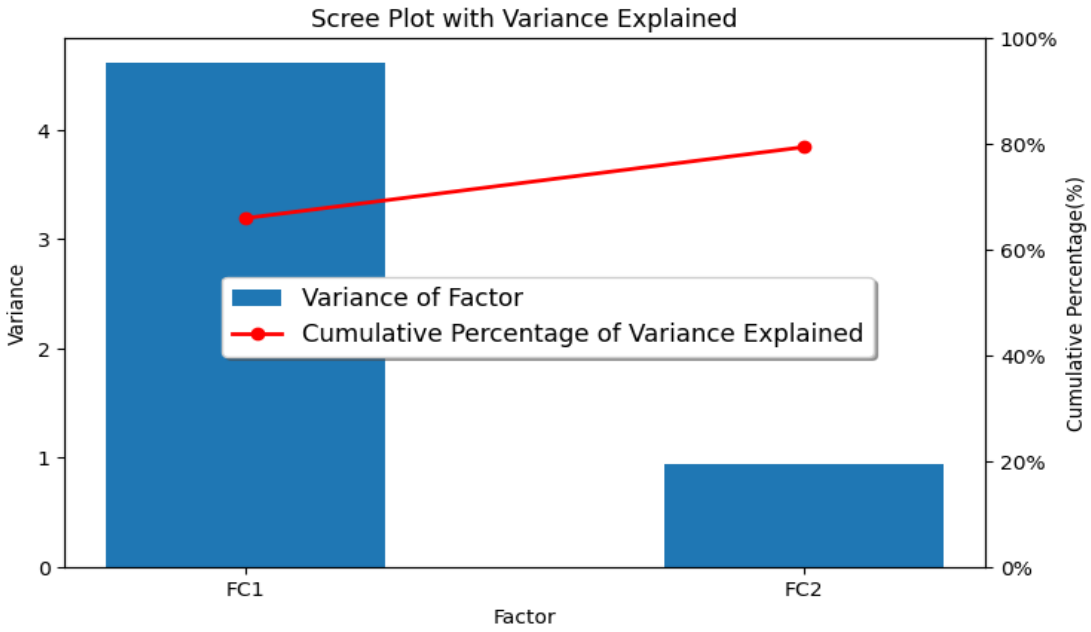
經研一 江彥亨 R11323040

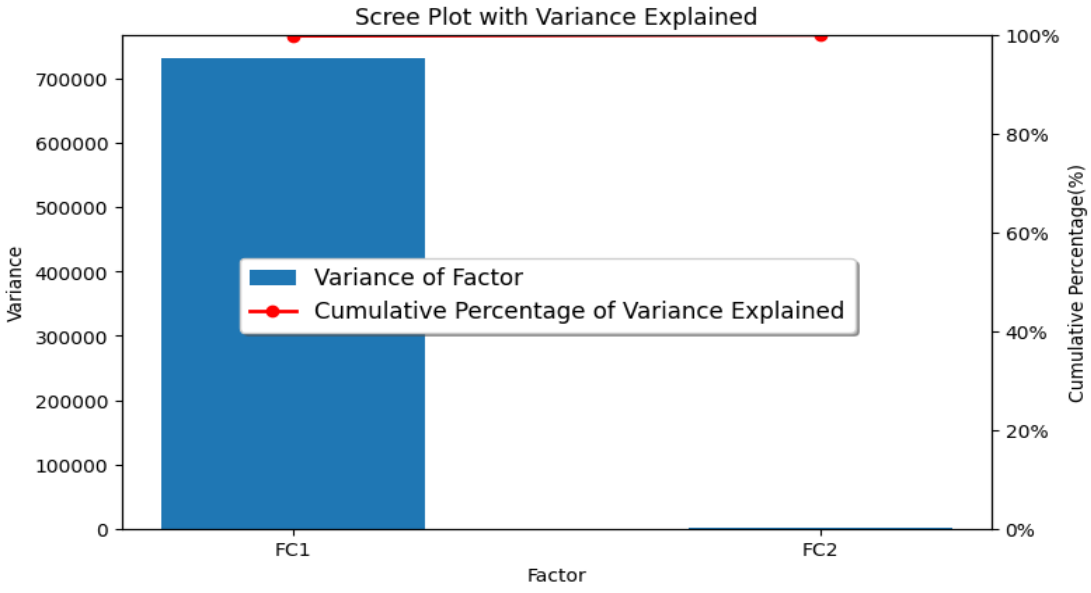
1. a

```python
def FA_func(X=None, isCorrMX=False, n_factors=None):
    n = len(X) # sample size
    p = len(X.columns) # number of variables
    X = (np.eye(n)-np.ones((n,n))/int(n)) @  X # center X
    Z = X / np.std(X, axis=0) # standardize X

    S = np.cov(X, rowvar=False) # covariance matrix of X
    R = np.corrcoef(X, rowvar=False) # correlation matrix of X

    if isCorrMX == True:
        eig_vals, eig_vecs = np.linalg.eig(R) # spectral decomposition
    else:
        eig_vals, eig_vecs = np.linalg.eig(S)

    A = np.sqrt(np.diag(eig_vals)) @ eig_vecs.T # loading matrix
    q = n_factors
    A = A[:q] # choose q factors

    h_square = np.diag(A.T @ A) # communality vector

    if isCorrMX == True:
        psi = np.diag(R - A.T @ A) # uniqueness vector
    else:
        psi = np.diag(S - A.T @ A)

    if isCorrMX == True:
        F = Z @ np.linalg.inv(np.diag(psi)) @ A.T @ np.linalg.inv(A @ np.linalg.inv(np.diag(psi)) @ A.T) # estimate factor matrix
    else:
        F = X @ np.linalg.inv(np.diag(psi)) @ A.T @ np.linalg.inv(A @ np.linalg.inv(np.diag(psi)) @ A.T)

    if isCorrMX == True:
        var_contributed = eig_vals/p # proportions of total variance contributed by each factor
    else:
        var_contributed = eig_vals/np.trace(S)
```

```python
#Scree plot
fig, ax1 = plt.subplots(figsize=(8,5))

FC = pd.Series({f"FC{i+1}":eig_vals[i] for i in range(q)}) # Variance of each factor
ax1.bar(FC.keys(), FC.values, width=0.5, align='center',
        label='Variance of Factor')
ax1.set_title('Scree Plot with Variance Explained')
ax1.set_xlabel('Factor')
ax1.set_ylabel('Variance')
if len(FC.keys()) > 10:
    ax1.xaxis.set_major_locator(ticker.MultipleLocator(500))

ax2 = ax1.twinx()
variance_ratio = var_contributed.cumsum()
ax2.plot(FC.keys(), variance_ratio[:q], 'o-', linewidth=2, c='r',
         label='Cumulative Percentage of Variance Explained')
ax2.set_ylabel('Cumulative Percentage(%)')
ax2.set_ylim(0, 1)
ax2.yaxis.set_major_formatter(ticker.PercentFormatter(1.0))

fig.legend(loc='center', fontsize=12, shadow=True)

return A, F, h_square, psi, var_contributed
```

1. b



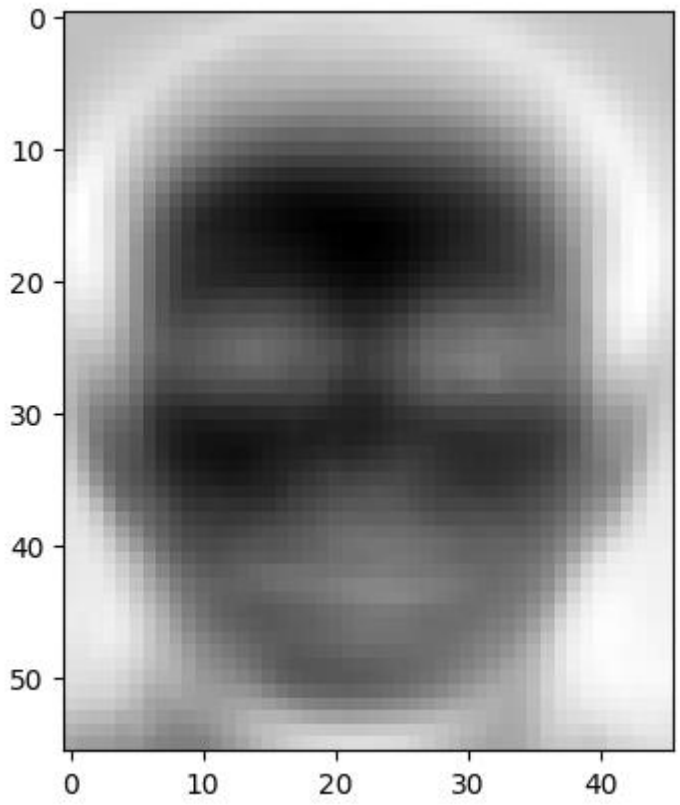


As shown in two graphs above, using the covariance matrix produces different results from using the correlation matrix. Similar to PCA, the loading matrix in FA is derived by spectral decomposition, so the results of FA can also be affected by scale transformation. Therefore, FA is scale variant.

2.a

```
2 factors are needed to explain 50% of total variance

4 factors are needed to explain 60% of total variance

7 factors are needed to explain 70% of total variance

17 factors are needed to explain 80% of total variance

50 factors are needed to explain 90% of total variance
```

2.b



3.a

```
MSE for 1 component : 14.707019459759698
MSE for 2 component : 10.991148636934089
MSE for 3 component : 10.134825206774277
MSE for 4 component : 10.365345908979883
MSE for 5 component : 9.905154651852952
MSE for 6 component : 9.931316994140339
MSE for 7 component : 9.787059754377932
```

	Var. Explained X	Cum. Var. Exp. X	Var. Explained Y	Cum. Var. Exp. Y
component1	0.645987	0.645987	0.739030	0.739030
component2	0.108755	0.754742	0.051653	0.790683
component3	0.069599	0.824341	0.010682	0.801366
component4	0.083421	0.907763	0.007090	0.808455
component5	0.067271	0.975033	0.002697	0.811152
component6	0.015834	0.990868	0.002128	0.813280
component7	0.005799	0.996667	0.000283	0.813563

Based on the testing results, the mean square error decreases as the more components(iterations) are included in the model. In addition, four components can explain 90% of the covariance of in the training data of X. However, even if we include all components, only 81% of the covariance in the training data of y can be explained.

3.b

```
MSE for 1 component : 14.564937014187791
MSE for 2 component : 13.810169516452783
MSE for 3 component : 13.77723945925516
MSE for 4 component : 13.675114874108456
MSE for 5 component : 13.236399213903418
MSE for 6 component : 13.104339221197298
```

	Var. Explained X	Cum. Var. Exp. X	Var. Explained Y	Cum. Var. Exp. Y
component1	0.727791	0.727791	0.582734	0.582734
component2	0.147511	0.875302	0.010434	0.593168
component3	0.073056	0.948357	0.009376	0.602544
component4	0.017890	0.966247	0.009976	0.612520
component5	0.024390	0.990637	0.002015	0.614534
component6	0.006030	0.996667	0.001124	0.615658

Based on the testing results, including more components in the model also leads to a decrease in mean squared error, but the rate of decrease is smaller than in 3.a. Additionally, just three components can explain 95% of the covariance of in the training data of X. However, when all components are included, only 61% of the covariance in the training data of y can be explained. The reason behind this is that we have reduced the number of explanatory variables while increasing the number of explained variables in model, which implies that we are using less information to predict more complex targets. Therefore, it is expected to have more prediction errors in the target variables.