Principal Component Analysis

Theroy related to PCA can be found in this link. In this document we will only show corresponding MATLAB codes.

```matlab
food = xlsread('D:/MATLAB/data/pca_abdi_food.csv','C2:I13');
[n,p] = size(food);
cent_food = food-mean(food);
scale_food = cent_food./std(cent_food);
```

Covariance PCA:

Using built in function:

```matlab
[loadings, factor_score, variances_using_built_in_function] = pca(food);
format bank
loadings
```

```
loadings = 7×7
        0.07        0.58        0.40        0.11       -0.17 ···
        0.33        0.41       -0.29        0.61        0.43
        0.30       -0.10       -0.34       -0.40        0.57
        0.75       -0.11        0.07       -0.29       -0.28
        0.47       -0.24        0.38        0.33       -0.06
        0.09        0.63       -0.23       -0.41       -0.24
       -0.06        0.14        0.66       -0.31        0.57
```

```matlab
factor_score
```

```
factor_score = 12×7
     -635.05     -120.89       21.14      -68.97        9.50 ···
     -488.56     -142.33     -132.37       34.91      -31.54
      112.03     -139.75       61.86       44.19      100.40
     -520.01       12.05       -2.85      -13.70       -3.52
     -485.94        1.17      -65.75       11.51       27.18
      588.17     -188.44       71.85       28.56       12.14
     -333.95      144.54       34.94       10.07      -28.51
      -57.51       42.86       26.26      -46.55        0.33
      571.32     -206.76       38.45        3.69      -94.73
      -39.38      264.47      126.43      -12.74      -14.45
        :
        :
        :
```

```matlab
variances_using_built_in_function
```

```
variances_using_built_in_function = 7×1
      274831.02
       26415.99
        6254.11
        2299.90
        2090.20
         338.39
          65.81
```

Using SVD (Manually):

Total variance = sum of variance of each column.

```
total_variance = sum(var(cent_food))
```

```
total_variance =
    312295.43
```

```
[u,sigma,v] = svd(cent_food);
loadings_manual_svd = v
```

```
loadings_manual_svd = 7×7
         0.07        -0.58        -0.40         0.11         0.17 ···
         0.33        -0.41         0.29         0.61        -0.43
         0.30         0.10         0.34        -0.40        -0.57
         0.75         0.11        -0.07        -0.29         0.28
         0.47         0.24        -0.38         0.33         0.06
         0.09        -0.63         0.23        -0.41         0.24
        -0.06        -0.14        -0.66        -0.31        -0.57
```

```
factor_score_manual_svd = cent_food * v
```

```
factor_score_manual_svd = 12×7
      -635.05       120.89       -21.14       -68.97        -9.50 ···
      -488.56       142.33       132.37        34.91        31.54
       112.03       139.75       -61.86        44.19      -100.40
      -520.01       -12.05         2.85       -13.70         3.52
      -485.94        -1.17        65.75        11.51       -27.18
       588.17       188.44       -71.85        28.56       -12.14
      -333.95      -144.54       -34.94        10.07        28.51
       -57.51       -42.86       -26.26       -46.55        -0.33
       571.32       206.76       -38.45         3.69        94.73
       -39.38      -264.47      -126.43       -12.74        14.45
       ⋮
```

```
variances_manual_svd = diag(sigma.^2)/(n-1)
```

```
variances_manual_svd = 7×1
    274831.02
     26415.99
      6254.11
      2299.90
      2090.20
       338.39
        65.81
```

Square of Matrix sigma contains eigen values of $\mathbf{X}^T\mathbf{X}$. So it has to be divied by $(n-1)$ to obtain eigen values of covariance matrix.

Using Eigen decomposition (Manually): (Not Recommended)

```
[vec,val] = eig(cov(cent_food));
```

```
[~, ind] = sort(diag(val),'descend');
loadings_using_eig = vec(:,ind)
```

```
loadings_using_eig = 7×7
        -0.07       0.58       0.40      -0.11      -0.17 ···
        -0.33       0.41      -0.29      -0.61       0.43
        -0.30      -0.10      -0.34       0.40       0.57
        -0.75      -0.11       0.07       0.29      -0.28
        -0.47      -0.24       0.38      -0.33      -0.06
        -0.09       0.63      -0.23       0.41      -0.24
         0.06       0.14       0.66       0.31       0.57
```

```
variances_using_eig = diag(val(ind,ind))
```

```
variances_using_eig = 7×1
     274831.02
      26415.99
       6254.11
       2299.90
       2090.20
        338.39
         65.81
```

```
sum_of_variances_eig = sum(variances_using_eig)
```

```
sum_of_variances_eig =
     312295.43
```

```
% or equivalently
[vec_n,val_n] = eig((1/(n-1))*(cent_food)'*cent_food);
[~, ind_n] = sort(diag(val_n),'descend');
loadings_using_eig_new = vec_n(:,ind_n)
```

```
loadings_using_eig_new = 7×7
         0.07       0.58       0.40       0.11       0.17 ···
         0.33       0.41      -0.29       0.61      -0.43
         0.30      -0.10      -0.34      -0.40      -0.57
         0.75      -0.11       0.07      -0.29       0.28
         0.47      -0.24       0.38       0.33       0.06
         0.09       0.63      -0.23      -0.41       0.24
        -0.06       0.14       0.66      -0.31      -0.57
```

```
variances_using_eig_new = diag(val_n(ind_n,ind_n))
```

```
variances_using_eig_new = 7×1
     274831.02
      26415.99
       6254.11
       2299.90
       2090.20
        338.39
         65.81
```

```
sum_of_variances_eig_new = sum(variances_using_eig_new)
```

```
sum_of_variances_eig_new =
     312295.43
```

```
% Check with total variance of data matrix
toal_variance_of_original_variables = sum(diag(cov(cent_food)))
```

```
toal_variance_of_original_variables =
     312295.43
```

```
% or equivalently
total_variance_of_original_variable_new = sum(var(cent_food))
```

```
total_variance_of_original_variable_new =
     312295.43
```

```
% or equivalently
total_variance_of_original_variable_new_2 = (1/(n-1))*sum(diag(cent_food'*cent_food))
```

```
total_variance_of_original_variable_new_2 =
     312295.43
```

Correlation PCA:

To do this manually, we have to first find correlation matrix and then perform operations on correlation matrix.

Different ways to obtain correlation matrix.

```
cor_matrix_using_function = corr(cent_food)
```

```
cor_matrix_using_function = 7×7
        1.00        0.59        0.20        0.32        0.25 ...
        0.59        1.00        0.86        0.88        0.83
        0.20        0.86        1.00        0.96        0.93
        0.32        0.88        0.96        1.00        0.98
        0.25        0.83        0.93        0.98        1.00
        0.86        0.66        0.33        0.37        0.23
        0.30       -0.36       -0.49       -0.44       -0.40
```

```
std_of_each_column = std(cent_food); % Standard deviation of each column
cor_matrix_another_way = cov(cent_food)./(std_of_each_column'*std_of_each_column)
```

```
cor_matrix_another_way = 7×7
        1.00        0.59        0.20        0.32        0.25 ...
        0.59        1.00        0.86        0.88        0.83
        0.20        0.86        1.00        0.96        0.93
        0.32        0.88        0.96        1.00        0.98
        0.25        0.83        0.93        0.98        1.00
        0.86        0.66        0.33        0.37        0.23
        0.30       -0.36       -0.49       -0.44       -0.40
```

```
cor_matrix_yet_another_way = (1/(n-1))*(scale_food)'*scale_food   % Where, "scale_food = (food-
```

```
cor_matrix_yet_another_way = 7×7
        1.00        0.59        0.20        0.32        0.25 ...
        0.59        1.00        0.86        0.88        0.83
        0.20        0.86        1.00        0.96        0.93
```

4

```
        0.32            0.88            0.96            1.00            0.98
        0.25            0.83            0.93            0.98            1.00
        0.86            0.66            0.33            0.37            0.23
        0.30           -0.36           -0.49           -0.44           -0.40
```

Using Built-in function:

```
[loadings_cor,variances_cor] = pcacov(corr(cent_food))
```

```
loadings_cor = 7×7
        0.24            0.62           -0.01           -0.54           -0.04 ···
        0.47            0.10           -0.06           -0.02            0.81
        0.45           -0.21            0.15            0.55            0.07
        0.46           -0.14            0.21           -0.05           -0.41
        0.44           -0.20            0.36           -0.32           -0.22
        0.28            0.52           -0.44            0.45           -0.34
       -0.21            0.48            0.78            0.31            0.07
variances_cor = 7×1
        4.33
        1.83
        0.63
        0.13
        0.06
        0.02
        0.00
```

```
sum(variances_cor)
```

```
ans =
        7.00
```

Note that we no longer get factor scores using this command. To get factor scores we have to use original scaled data and multiply it by loading scores.

```
factor_scores_cor = scale_food*loadings_cor
```

```
factor_scores_cor = 12×7
       -2.86           -0.36            0.40            0.36           -0.23 ···
       -1.89           -1.79           -1.31           -0.16            0.09
       -0.12           -0.73            1.42            0.20            0.44
       -2.04            0.32           -0.11            0.10           -0.01
       -1.69           -0.16           -0.51            0.16            0.18
        1.69           -1.35            0.99           -0.43            0.08
       -0.93            1.37           -0.28           -0.26           -0.09
       -0.25            0.63            0.27            0.29           -0.16
        1.60           -1.74            0.10           -0.40           -0.42
        0.22            2.78            0.57           -0.25           -0.12
      :
      :
```

Using SVD (Manually):

A note is in order here. Eigen values of S^2 are square of eigenvalues of S.

```
[u_cor,sigma_cor,v_cor] = svd(corr(cent_food));
```

```
loadings_cor_svd = v_cor
```

```
loadings_cor_svd = 7×7
        -0.24         0.62         0.01        -0.54        -0.04 · · ·
        -0.47         0.10         0.06        -0.02         0.81
        -0.45        -0.21        -0.15         0.55         0.07
        -0.46        -0.14        -0.21        -0.05        -0.41
        -0.44        -0.20        -0.36        -0.32        -0.22
        -0.28         0.52         0.44         0.45        -0.34
         0.21         0.48        -0.78         0.31         0.07
```

```
variances_cor_svd = diag(sigma_cor)  % This is equivalent to: sqrt(diag(sigma_cor).^2)
```

```
variances_cor_svd = 7×1
         4.33
         1.83
         0.63
         0.13
         0.06
         0.02
         0.00
```

```
sqrt(diag(sigma_cor).^2)
```

```
ans = 7×1
         4.33
         1.83
         0.63
         0.13
         0.06
         0.02
         0.00
```

```
sum_of_variances_cor = sum(diag(sigma_cor))
```

Using Eigenvaectors (Manual):

```
[vec_cor,val_cor] = eig(corr(cent_food));
[~,ind] = sort(diag(val_cor),'descend');
loadings_cor_eig = vec_cor(:,ind)
```

```
loadings_cor_eig = 7×7
         0.24         0.62         0.01        -0.54         0.04 · · ·
         0.47         0.10         0.06        -0.02        -0.81
         0.45        -0.21        -0.15         0.55        -0.07
         0.46        -0.14        -0.21        -0.05         0.41
         0.44        -0.20        -0.36        -0.32         0.22
         0.28         0.52         0.44         0.45         0.34
        -0.21         0.48        -0.78         0.31        -0.07
```

```
variances_cor_eig = diag(val_cor(ind,ind))
```

```
variances_cor_eig = 7×1
         4.33
         1.83
         0.63
         0.13
         0.06
```

```
       0.02
       0.00
```

```
sum_of_variances_cor_eig = sum(variances_cor_eig)
```

sum_of_variances_cor_eig =
       7.00