MATH 189 HW7

Zijian Su
Zelong Zhou
Xiangyi Lin

Last Updated: March 03, 2023

Concrete contributions

All problems were done by Zijian Su, Zelong Zhou, Xiangyi Lin. All contributing equally to this assignment. Everyone put in enough effort.

Overview

Places Rated Almanac rated 329 communities in the United States according to the following nine criteria:

1. Climate and Terrain
2. Housing
3. Health Care & the Environment
4. Crime
5. Transportation
6. Education
7. The Arts
8. Recreation
9. Economics

The rating results can be found in Places_Rated.txt. In this dataset, the first 9 columns represent the above 9 variables. The 10th column is the index of communities, ranging from 1 to 329. Note that, except for housing and crime, the higher the score is the better condition the community has. Analyze this dataset according to the following steps.

Packages

```r
#install.packages("rmarkdown")
#install.packages("ggplot2")
#install.packages("ggfortify")
#tinytex::install_tinytex()

#install.packages("scatterplot3d")
```

Question 1

Calculate the eigenvalues and eigenvectors of the covariance matrix of standardized data (each column has mean 0 and variance 1). Calculate the proportion of total variance explained, by each eigenvector and the cumulative proportion of total variance explained by the first k (=1,. . . ,9) eigenvectors.
Report your results in scree plot and cumulative plot. Next, repeat the above steps to raw data, and draw scree plot and cumulative plot. Compare the results obtained from raw and standardized data.

Answer:

Calculate the eigenvalues and eigenvectors of the covariance matrix of standardized data:

```
data <- read.table("Places_Rated.txt")
data <- data[-10]
colnames(data)[1] <- "Climate and Terrain"
colnames(data)[2] <- "Housing"
colnames(data)[3] <- "Health Care & the Environment"
colnames(data)[4] <- "Crime"
colnames(data)[5] <- "Transportation"
colnames(data)[6] <- "Education"
colnames(data)[7] <- "The Arts"
colnames(data)[8] <- "Recreation"
colnames(data)[9] <- "Economics"


data_stand <- scale(data) #standardized data, skip k = 10
cov_matx <-cov(data_stand) #covariance matrix
eigenvalues <- eigen(cov_matx)$values   # eigenvalues
eigenvectors <- eigen(cov_matx)$vectors  # eigenvectors
head(data_stand)
```

```
##      Climate and Terrain     Housing Health Care & the Environment       Crime
## [1,]          -0.1467824 -0.89992576                    -0.9458990 -0.10654981
## [2,]           0.3002069 -0.08743661                     0.4688539 -0.21014653
## [3,]          -0.5854941 -0.42241020                    -0.5660393  0.02504601
## [4,]          -0.5192735 -0.18386205                     0.2445273 -0.98292201
## [5,]           0.9955236  0.01946986                     0.6652643  1.46140045
## [6,]          -0.1550600 -1.05965660                    -0.5441052 -0.65533240
##      Transportation  Education   The Arts Recreation  Economics
## [1,]     -0.1234045 -0.1804514 -0.4641863 -0.5458150  1.9434730
## [2,]      0.4637042 -1.1748623  0.5198122  0.9729596 -1.0838164
## [3,]     -1.1570466 -0.7945547 -0.6276834 -1.2216511 -0.2539168
## [4,]      1.8418937  1.8208394  0.3240034 -0.2834024  0.3122592
## [5,]      1.6179379  0.6580957  0.2897530  0.9482037  0.1859300
## [6,]     -1.2169979  0.4897628 -0.6067885 -1.0248417 -0.2502283
```

```
eigen(cov_matx) #print out the details
```

```
## eigen() decomposition
## $values
## [1] 3.4082918 1.2139762 1.1414791 0.9209178 0.7532849 0.6305619 0.4930477
## [8] 0.3180385 0.1204021
```

```
##
## $vectors
##              [,1]       [,2]         [,3]        [,4]       [,5]        [,6]
##  [1,] -0.2064140  0.2178353  0.689955982  0.13732125  0.3691499 -0.37460469
##  [2,] -0.3565216  0.2506240  0.208172230  0.51182871 -0.2334878  0.14163983
##  [3,] -0.4602146 -0.2994653  0.007324926  0.01470183  0.1032405  0.37384804
##  [4,] -0.2812984  0.3553423 -0.185104981 -0.53905047  0.5239397 -0.08092329
##  [5,] -0.3511508 -0.1796045 -0.146376283 -0.30290371 -0.4043485 -0.46759180
##  [6,] -0.2752926 -0.4833821 -0.229702548  0.33541103  0.2088191 -0.50216981
##  [7,] -0.4630545 -0.1947899  0.026484298 -0.10108039  0.1050976  0.46188072
##  [8,] -0.3278879  0.3844746  0.050852640 -0.18980082 -0.5295406 -0.08991578
##  [9,] -0.1354123  0.4712833 -0.607314475  0.42176994  0.1596201 -0.03260813
##               [,7]        [,8]          [,9]
##  [1,]  0.08470577  0.36230833 -0.0013913515
##  [2,]  0.23063862 -0.61385513 -0.0136003402
##  [3,] -0.01386761  0.18567612  0.7163548935
##  [4,] -0.01860646 -0.43002477  0.0586084614
##  [5,]  0.58339097  0.09359866 -0.0036294527
##  [6,] -0.42618186 -0.18866756 -0.1108401911
##  [7,]  0.02152515  0.20398969 -0.6857582127
##  [8,] -0.62787789  0.15059597  0.0255062915
##  [9,]  0.14974066  0.40480926 -0.0004377942
```

See output above for eigenvalues and eigenvectors for scaled data.

Calculate the proportion of total variance explained, by each eigenvector and the cumulative proportion of total variance explained by the first k (=1,...,9) eigenvectors:

```
eigenvalues <- eigen(cov_matx)$values
prop_var <- eigenvalues / sum(eigenvalues)
prop_var
```

```
## [1] 0.37869909 0.13488624 0.12683102 0.10232420 0.08369832 0.07006243 0.05478308
## [8] 0.03533761 0.01337801
```

See output above for the proportion of total variance explained by each eigenvector.

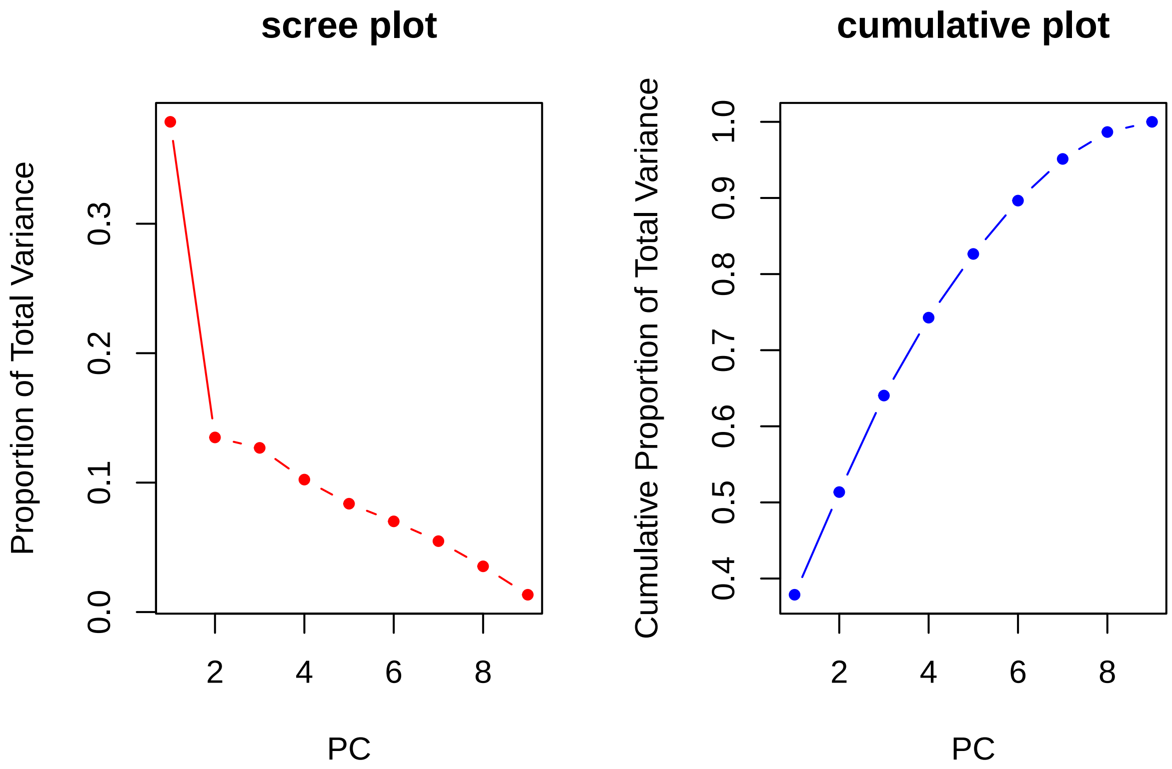
```
cum_prop_var <- cumsum(prop_var)
cum_prop_var
```

```
## [1] 0.3786991 0.5135853 0.6404163 0.7427405 0.8264389 0.8965013 0.9512844
## [8] 0.9866220 1.0000000
```

See output above for the cumulative proportion of total variance explained by the first k (=1,...,9) eigenvectors.

Report your results in scree plot and cumulative plot:

```
par(mfrow = c(1, 2))
plot(prop_var,pch = 20, type = "b", main = "scree plot", xlab = "PC", ylab = "Proportion of Total Variar
plot(cum_prop_var, pch = 20, type = "b", main = "cumulative plot", xlab = "PC", ylab = "Cumulative Propo
```



Repeat the above steps to raw data:

```
cov_matx_2 <-cov(data) #covariance matrix, skip k = 10

eigenvalues2 <- eigen(cov_matx_2)$values
eigenvectors2 <- eigen(cov_matx_2)$vectors

eigen(cov_matx_2)
```

```
## eigen() decomposition
## $values
## [1] 24413668.72  4408004.85  1638039.60  1076355.78   478338.27   240851.80
## [7]    92809.94    66995.90    10962.63
##
## $vectors
##              [,1]         [,2]         [,3]        [,4]         [,5]
##  [1,] -0.006416346  0.015459527 -0.006692298 -0.02631066 -0.016278231
##  [2,] -0.269142181  0.937207188 -0.082641934 -0.17775057  0.083842278
##  [3,] -0.178318724 -0.020539870  0.027761041 -0.02656157  0.159075722
##  [4,] -0.028134276 -0.010901921  0.037610931  0.09903536 -0.116013534
##  [5,] -0.149302463  0.018757344  0.971531831 -0.03839697  0.146649668
##  [6,] -0.025190912 -0.001395877  0.041507669  0.02163938  0.106255968
##  [7,] -0.930859522 -0.282260587 -0.151026851  0.02775471 -0.008673762
##  [8,] -0.069824043  0.103848215  0.149571984  0.06903276 -0.954262248
##  [9,] -0.025130829  0.173359958  0.012743344  0.97453606  0.102240592
##              [,6]        [,7]        [,8]          [,9]
##  [1,] -0.001186617 -0.08140848 -0.04213801  0.9951449417
##  [2,] -0.048638182 -0.02668780 -0.01211847 -0.0229330011
##  [3,]  0.929492918 -0.13706121  0.24135975  0.0013718748
##  [4,] -0.053976191 -0.94477955 -0.26682693 -0.0876894940
##  [5,] -0.092235051  0.01354542  0.04150769  0.0094188168
##  [6,]  0.253188491  0.24115526 -0.92915944 -0.0168655619
##  [7,] -0.167554494  0.04296041 -0.01594931  0.0005985854
##  [8,]  0.173348306  0.12711706 -0.01878071 -0.0050315892
##  [9,]  0.005152175  0.07016097  0.05439799  0.0327178331
```

See output above for eigenvalues and eigenvectors for raw data

```
prop_var2 <- eigenvalues2 / sum(eigenvalues2)
cum_prop_var2 <- cumsum(prop_var2)  # k = 1 to 9
prop_var2
```

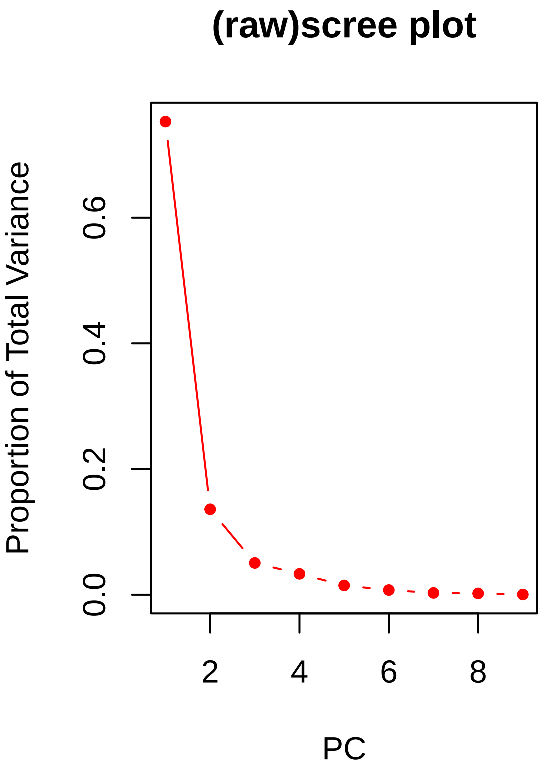
```
## [1] 0.752903473 0.135940329 0.050516197 0.033194192 0.014751677 0.007427731
## [7] 0.002862205 0.002066115 0.000338081
```

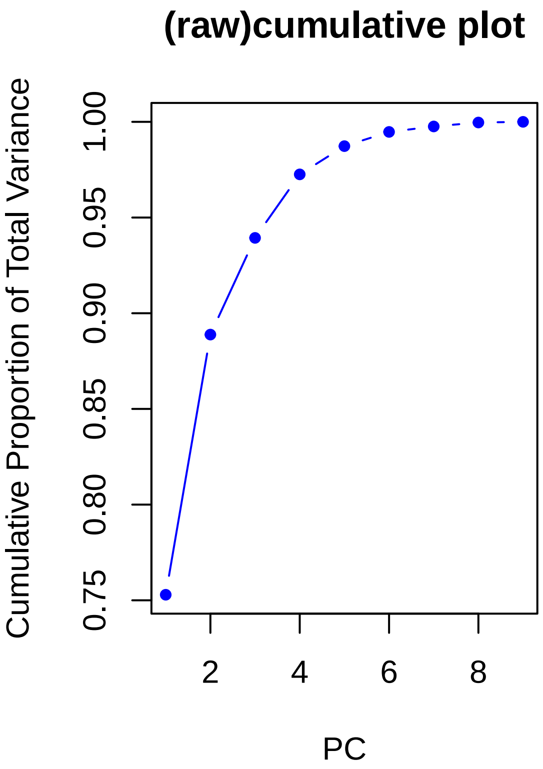
```
cum_prop_var2
```

```
## [1] 0.7529035 0.8888438 0.9393600 0.9725542 0.9873059 0.9947336 0.9975958
## [8] 0.9996619 1.0000000
```

See output above for the proportion of total variance explained by each eigenvector and the cumulative proportion of total variance explained by the first k (=1,. . . ,9) eigenvectors about the raw data.

```
par(mfrow = c(1, 2))
plot(prop_var2, pch = 20, type = "b", main = "(raw)scree plot", xlab = "PC", ylab = "Proportion of Total
plot(cum_prop_var2, pch = 20, type = "b", main = "(raw)cumulative plot", xlab = "PC", ylab = "Cumulative
```





Compare the results obtained from raw and standardized data:

when we using PCA on the raw data, first principal component explains more than 75% of total variation. and for the standardized data, first principal component explains more than 35% of total variation. We think about the first 4 ~5 principal components are enough to describe the data. Because the first two eigenvalues are much larger than others. In the cumulative plot, the values of the proportion of total variance explained in the standardized data are relatively close, so the polt looks like a straight line. In the raw data, the value of PC1 is relatively large. The values of other PCs are relatively small instead. As can be seen from the 2 plots, if we do not standardize the data, then some features may unduly affect our judgment.

Question 2

Apply principal component analysis to the standardized data. Choose the number of principal components (k) according to the scree plot you obtained in Part 1. Report the corresponding principal component loading vectors. Visualize the dataset by projecting the observations onto the plane spanned by the first two principal components.

Answer:

Determine the number of principal components

```
k = which.max(cum_prop_var >= 0.8)
k
```

```
## [1] 5
```

we choose first 5 principal components (k =5). Because the first 5 eigenvalues are much larger than others. Thus, the first 5 PCs are enough to describe the data.

```
loading_vectors <- eigenvectors * sqrt(eigenvalues[1])
loading_vectors
```

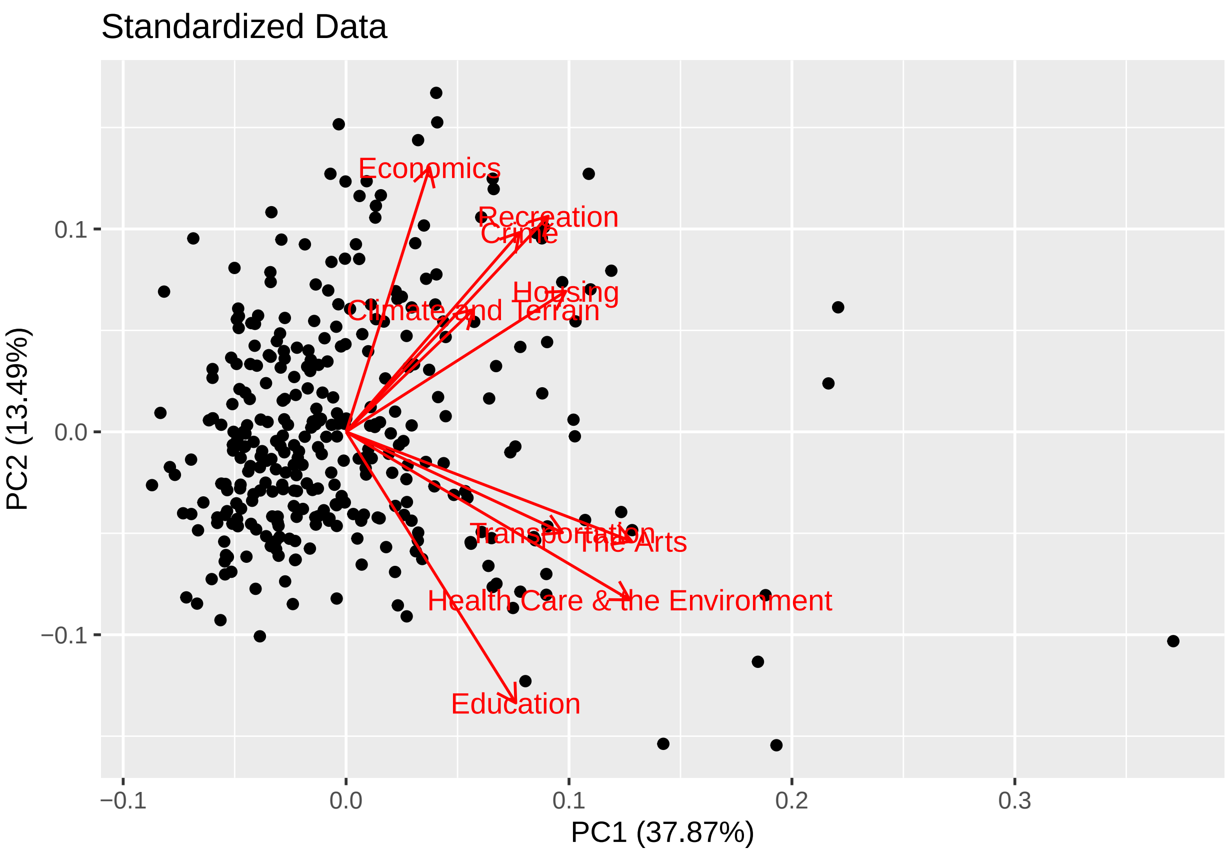
```
##             [,1]       [,2]        [,3]        [,4]       [,5]       [,6]
##  [1,] -0.3810724  0.4021580  1.27376634  0.25351644  0.6815083 -0.6915787
##  [2,] -0.6581945  0.4626910  0.38431840  0.94491562 -0.4310549  0.2614892
##  [3,] -0.8496280 -0.5528596  0.01352295  0.02714187  0.1905981  0.6901818
##  [4,] -0.5193207  0.6560173 -0.34173266 -0.99517124  0.9672744 -0.1493970
##  [5,] -0.6482791 -0.3315779 -0.27023345 -0.55920748 -0.7464903 -0.8632474
##  [6,] -0.5082331 -0.8923987 -0.42406673  0.61922108  0.3855125 -0.9270838
##  [7,] -0.8548708 -0.3596126  0.04889414 -0.18661016  0.1940266  0.8527038
##  [8,] -0.6053322  0.7098001  0.09388190 -0.35040191 -0.9776145 -0.1659986
##  [9,] -0.2499921  0.8700624 -1.12119723  0.77865308  0.2946835 -0.0601997
##              [,7]       [,8]          [,9]
##  [1,]  0.15638007  0.6688777 -0.0025686518
##  [2,]  0.42579487 -1.1332723 -0.0251083490
##  [3,] -0.02560177  0.3427871  1.3225028506
##  [4,] -0.03435042 -0.7938928  0.1082003598
##  [5,]  1.07703071  0.1727977 -0.0067005356
##  [6,] -0.78679818 -0.3483097 -0.2046282785
##  [7,]  0.03973879  0.3765968 -1.2660166063
##  [8,] -1.15916051  0.2780236  0.0470885918
##  [9,]  0.27644461  0.7473410 -0.0008082365
```

above are the corresponding principal component loading vectors.

Visualize the dataset by projecting the observations onto the plane spanned by the first two principal components:

```
library(ggfortify)
pca_1 <- prcomp(data[1:9], scale. = TRUE)

plot1 <- autoplot(pca_1 ,loadings= TRUE, loadings.label = TRUE, main="Standardized Data")
plot1
```


Standardized Data
PC2 (13.49%)
PC1 (37.87%)
0.1
0.0
−0.1
−0.1
0.0
0.1
0.2
0.3
Economics
Recreation
Crime
Housing
Climate and Terrain
Transportation
The Arts
Health Care & the Environment
Education