bacs_hw11

110071010

2024-05-05

109048231 helped me get to know that it is the ratios that are required in Question 1aii and Question 2a.

Question 1

Let's revisit the issue of multicollinearity of main effects (between cylinders, displacement, horsepower, and weight) we saw in the cars dataset, and try to apply principal components to it. Start by recreating the cars_log dataset, which log-transforms all variables except model year and origin.

1a

Let's analyze the principal components of the four collinear variables

```
cars <- read.table("auto-data.txt", header = FALSE, na.strings = "?")
cars1 <- na.omit(cars)
names(cars1) <- c("mpg", "cylinders", "displacement", "horsepower", "we
ight","acceleration", "model_year", "origin", "car_name")
cars1_log <- with(cars1, data.frame(log(mpg), log(cylinders), log(displ
acement), log(horsepower), log(weight), log(acceleration), model_year,
origin))
```

i

Create a new data.frame of the four log-transformed variables with high multicollinearity

```
construct <- data.frame(cars1_log$log.cylinders.,cars1_log$log.displace
ment.,cars1_log$log.horsepower.,cars1_log$log.weight.)
```

ii

How much variance of the four variables is explained by their first principal component?

```
construct_eigen <- eigen(cov(construct))
eigen(cov(construct))$values[1]/sum(eigen(cov(construct))$values)

## [1] 0.9346169
```

iii

Question

Looking at the values and valence (positiveness/negativeness) of the first principal component's eigenvector, what would you call the information captured by this component?

Answer

each principal component's direction of variance (from Week 11 handout)

```
construct_eigen$vectors
```

```
##            [,1]        [,2]       [,3]        [,4]
## [1,] -0.3944484  0.32615343  0.6895416  0.51241263
## [2,] -0.7221160  0.36134848 -0.1626248 -0.56703525
## [3,] -0.4322835 -0.87289692  0.2158783 -0.06766477
## [4,] -0.3689037 -0.03319916 -0.6719242  0.64134686
```

1b

i

Store the scores of the first principal component as a new column of cars_log
cars_log$new_column_name <- ...scores of PC1...

```
cars1_log$construct <- construct_eigen$vectors[,1]
```

ii

Regress mpg over the column with PC1 scores (replacing cylinders, displacement, horsepower, and weight), as well as acceleration, model_year and origin

```
lm(log.mpg. ~ construct + log.acceleration.+ model_year + origin, data
= cars1_log)
```

```
##
## Call:
## lm(formula = log.mpg. ~ construct + log.acceleration. + model_year +

##     origin, data = cars1_log)
##
## Coefficients:
##       (Intercept)          construct  log.acceleration.         mode
l_year
##          -1.36568            0.01599            0.43916            0.
03932
##            origin
##           0.18167
```

iii

Instruction

Try running the regression again over the same independent variables, but this time with everything standardized. How important is this new column relative to other columns?

Answer

It appears important as its after-standardization coefficient (6.680e-03) outweigh other variables'.

```
sdc <- data.frame(scale(cars1_log))
lm(log.mpg.~construct+log.acceleration.+model_year+origin,data=sdc)

##
## Call:
## lm(formula = log.mpg. ~ construct + log.acceleration. + model_year +

##     origin, data = sdc)
##
## Coefficients:
##       (Intercept)          construct  log.acceleration.          mode
l_year
##         9.857e-16          6.680e-03          2.337e-01           4.2
60e-01
##            origin
##         4.304e-01
```

Question 2

A group of researchers is studying how customers who shopped on e-commerce websites over the winter holiday season perceived the security of their most recently used e-commerce site. Based on feedback from experts, the company has created eighteen questions (see 'questions' tab of excel file) regarding security considerations at e-commerce websites. Over 400 customers responded to these questions (see 'data' tab of Excel file). The researchers now wants to use the results of these eighteen questions to reveal if there are some underlying dimensions of people's perception of online security that effectively capture the variance of these eighteen questions. Let's analyze the principal components of the eighteen items.

2a

Question

How much variance did each extracted factor explain?

Answer

See “Cumulative Proportion” below

```
sq <- read.csv('security_questions.csv')
pca <- prcomp(sq)
summary(pca)

## Importance of components:
##                           PC1     PC2    PC3     PC4     PC5    PC6
    PC7
## Standard deviation     4.5803 2.01574 1.6194 1.30124 1.25295 1.2341
1.07068
## Proportion of Variance 0.5097 0.09871 0.0637 0.04113 0.03814 0.0370
0.02785
## Cumulative Proportion  0.5097 0.60836 0.6721 0.71319 0.75133 0.7883
0.81618
##                            PC8    PC9    PC10    PC11    PC12   PC13
    PC14
## Standard deviation     1.03349 0.9940 0.93530 0.88795 0.81779 0.8166
 0.76556
## Proportion of Variance 0.02595 0.0240 0.02125 0.01915 0.01625 0.0162
 0.01424
## Cumulative Proportion  0.84213 0.8661 0.88738 0.90653 0.92278 0.9390
 0.95322
##                           PC15    PC16    PC17    PC18
## Standard deviation     0.74400 0.72833 0.65653 0.64084
## Proportion of Variance 0.01345 0.01289 0.01047 0.00998
## Cumulative Proportion  0.96667 0.97955 0.99002 1.00000
```

2b

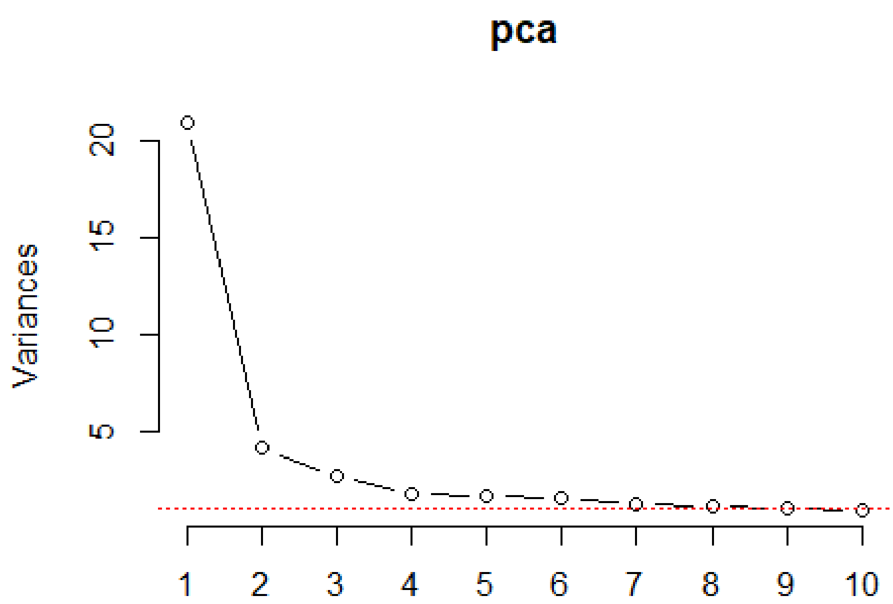
How many dimensions would you retain, according to the two criteria we discussed?

(Eigenvalue ≥ 1 and Scree Plot – can you show the screenplot with eigenvalue=1 threshold?)

```
sq2 <- (pca$sdev)^2
above_one <- sq2 >= 1 # Eigenvalue ≥ 1
below_one <- !above_one # Eigenvalue < 1
retained_dim <- sq2[above_one]
retained_dim

## [1] 20.979225  4.063219  2.622295  1.693233  1.569875  1.523076  1.1
46356
## [8]  1.068102

screeplot(pca, type="lines")
abline(h=1,col='red',lty=3) # red line : eigenvalue = 1
```



2c

Question

Can you interpret what any of the principal components mean? Try guessing the meaning of the first two or three PCs looking at the PC-vs-variable matrix (ungraded)

```
pcs <- pca$rotation
pc1 <- pcs[,1]
pc2 <- pcs[,2]

print("Principal Component 1:")
print(pc1)

## [1] "Principal Component 1:"
##         Q1         Q2         Q3         Q4         Q5         Q6
      Q7
## -0.2491083 -0.2463737 -0.2431477 -0.2221963 -0.2106025 -0.2155338 -0.
2427124
##         Q8         Q9        Q10        Q11        Q12        Q13
     Q14
## -0.2629719 -0.2446115 -0.2199303 -0.2463162 -0.2239165 -0.2167467 -0.
2302322
##        Q15        Q16        Q17        Q18
## -0.2408316 -0.2569632 -0.2276085 -0.2346175
```

```
print("Principal Component 2:")
print(pc2)

## [1] "Principal Component 2:"
##          Q1          Q2          Q3          Q4          Q5
 Q6
## -0.10493359 -0.03148303 -0.03436924  0.49469712 -0.03181945 -0.08661
273
##          Q7          Q8          Q9         Q10         Q11
Q12
## -0.29259568  0.01570353 -0.19064006 -0.08075312 -0.20422392  0.48758
805
##         Q13         Q14         Q15         Q16         Q17
Q18
## -0.04986984 -0.07039120 -0.01016821 -0.16718491  0.53251587 -0.07959
100
```

Answer

PC1 appear to be a general trend as the weights are all negative, hovering around -0.23.

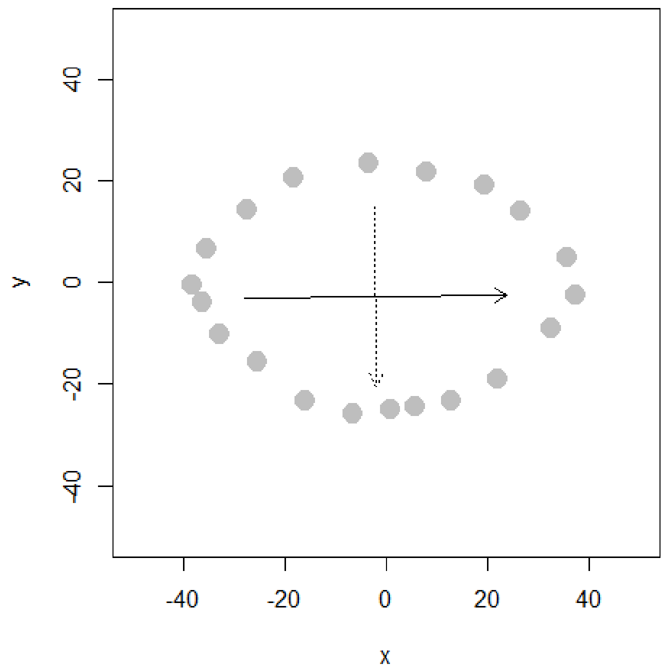
PC2 has strong positive loadings for Q4 ,Q12, Q17. Based on the security-questions data, these questions are confidentiality-related.

Question 3

Let's simulate how principal components behave interactively: run the interactive_pca() function from the compstatslib package we have used earlier:

3a

Create an oval shaped scatter plot of points that stretches in two directions – you should find that the principal component vectors point in the major and minor directions of variance (dispersion). Show this visualization.

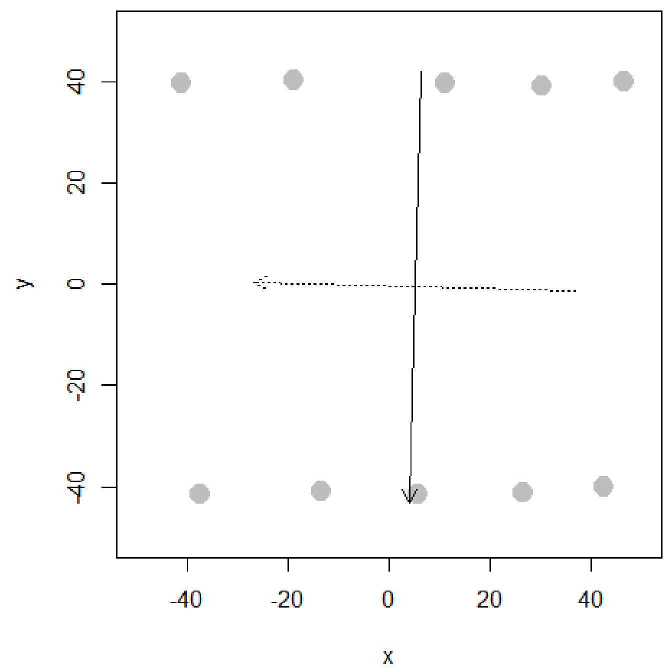



```
# compstatslib::interactive_pca()
# Click on the plot to create data points; hit [esc] to stop$points
#             x          y
# 1  -38.449096  -0.3139867
# 2  -35.594672   6.8220742
# 3  -27.602284  14.5290200
# 4  -18.468126  20.8087536
# 5   -3.625119  23.6631779
# 6    7.792578  21.9505233
# 7   19.210276  19.3815414
# 8   26.346337  14.2435775
# 9   35.480495   5.1094196
# 10  37.193149  -2.3120837
# 11  32.340628  -8.8772598
# 12  21.779258 -18.8677450
# 13  12.930542 -23.1493815
# 14   5.509039 -24.2911513
# 15   0.941960 -24.8620362
# 16  -6.764986 -25.7183635
# 17 -16.184586 -23.1493815
# 18 -25.604186 -15.4424358
# 19 -33.025690 -10.0190295
# 20 -36.450999  -3.7392959
#
# $pca
# Standard deviations (1, .., p=2):
```

```
# [1] 25.79274 17.65758
#
# Rotation (n x k) = (2 x 2):
#             PC1          PC2
# x 0.999967513  0.008060619
# y 0.008060619 -0.999967513
```

3b

Can you create a scatterplot whose principal component vectors do NOT seem to match the major directions of variance? Show this visualization.




```
# compstatslib::interactive_pca()
# Click on the plot to create data points; hit [esc] to stop$points
#              x         y
# 1  -41.303520  39.93340
# 2  -19.039010  40.50428
# 3   10.932445  39.93340
# 4   30.057088  39.36251
# 5   46.327307  40.21884
# 6   42.331113 -39.99049
# 7   26.346337 -41.13225
# 8    5.509039 -41.41770
# 9  -13.615604 -40.84681
# 10 -37.592769 -41.41770
#
```

```
# $pca
# Standard deviations (1, .., p=2):
# [1] 42.67473 31.79428
#
# Rotation (n x k) = (2 x 2):
#           PC1         PC2
# x -0.02676013 -0.99964188
# y -0.99964188  0.02676013
```