BACS HW13

109090046 assisted by 109090035 109090023

2023-05-11

Load data

```
auto <- read.table("D:/下載/auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleratio
n", "model_year", "origin", "car_name")
```

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.

Important: remove any rows that have missing values.

a.

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

i. Create a new data.frame of the four log-transformed variables with high multicollinearity (Give this smaller data frame an appropriate name – what might they jointly mean?)

```
# Creating a new data frame with the four log-transformed variables
cars_log <- with(auto, data.frame(log(mpg), log(cylinders), log(displacement), log(horsep
ower), log(weight), log(acceleration), model_year, origin))

# remove na
cars_log<-na.omit(cars_log)

cars_log_regr <- lm(log.mpg. ~ log.cylinders. + log.displacement. + log.horsepower. + log.
weight. + log.acceleration. + model_year + factor(origin), data = cars_log ,na.action = n
a.exclude)

# vif from car package
vif(cars_log_regr)

##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    10.456738  1        3.233688
## log.displacement. 29.625732  1        5.442952
## log.horsepower.   12.132057  1        3.483110
## log.weight.       17.575117  1        4.192269
## log.acceleration.  3.570357  1        1.889539
## model_year         1.303738  1        1.141814
## factor(origin)     2.656795  2        1.276702
```

There are four variables (cylinders, displacement, horsepower, and weight) have high multicollinearity.

```
new_cars_log <-with(auto, data.frame(log(cylinders), log(displacement),
log(horsepower),log(weight)), na.rm=TRUE)
```

```
new_cars_log <- na.omit(new_cars_log)
head(new_cars_log)

##   log.cylinders. log.displacement. log.horsepower. log.weight.
## 1       2.079442          5.726848        4.867534    8.161660
## 2       2.079442          5.857933        5.105945    8.214194
## 3       2.079442          5.762051        5.010635    8.142063
## 4       2.079442          5.717028        5.010635    8.141190
## 5       2.079442          5.710427        4.941642    8.145840
## 6       2.079442          6.061457        5.288267    8.375860
```

ii. How much variance of the four variables is explained by their first principal component? (a summary of the prcomp() shows it, but try computing this from the eigenvalues alone)

```
# Principal components analysis
pca_cars <- prcomp(new_cars_log, scale. = TRUE)
summary(pca_cars)

## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.9168 0.43316 0.32238 0.18489
## Proportion of Variance 0.9186 0.04691 0.02598 0.00855
## Cumulative Proportion  0.9186 0.96547 0.99145 1.00000

# Eigenvalues
eigenvalues <- eigen(cor(new_cars_log))$values
var_explained <- eigenvalues[1] / length(eigenvalues)
var_explained

## [1] 0.9185647
```

iii. 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? (i.e., think what concept the first principal component captures or represents)

```
loadings <- pca_cars$rotation
print(loadings[, 1])

##    log.cylinders. log.displacement.   log.horsepower.       log.weight.
##        -0.4979145        -0.5122968        -0.4856159        -0.5037960
```

The first principal component capture all 4 variables (cylinders, displacement, horsepower, and weight) at almost same level (0.5) and they are all negative.

The sign of a loading indicates the direction of the correlation between the original variable and the component. If all loadings are positive, it could mean that the first principal component represents the overall size or power of the car. This would be consistent with the fact that cylinders, displacement, horsepower, and weight are all measures of size or power. If some loadings are negative, the interpretation would be more complex and depend on the specific loadings.

b.

Let's revisit our regression analysis on cars_log:

i. Store the scores of the first principal component as a new column of cars_log cars_log$new_column_name <- …scores of PC1… Give this new column a name suitable for what it captures (see 1.a.i.)

```
cars_log$car_power <- predict(pca_cars)[, 1]
head(cars_log)
```

```
##   log.mpg. log.cylinders. log.displacement. log.horsepower. log.weight.
## 1 2.890372       2.079442          5.726848        4.867534    8.161660
## 2 2.708050       2.079442          5.857933        5.105945    8.214194
## 3 2.890372       2.079442          5.762051        5.010635    8.142063
## 4 2.772589       2.079442          5.717028        5.010635    8.141190
## 5 2.833213       2.079442          5.710427        4.941642    8.145840
## 6 2.708050       2.079442          6.061457        5.288267    8.375860
##   log.acceleration. model_year origin car_power
## 1          2.484907         70      1 -2.036645
## 2          2.442347         70      1 -2.593998
## 3          2.397895         70      1 -2.237767
## 4          2.484907         70      1 -2.192902
## 5          2.351375         70      1 -2.097313
## 6          2.302585         70      1 -3.337215
```

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

```
lm1 <- lm(log.mpg. ~ car_power + log.acceleration. + model_year + factor(origin), data =
cars_log, na.action = na.exclude)

summary(lm1)

##
## Call:
## lm(formula = log.mpg. ~ car_power + log.acceleration. + model_year +
##     factor(origin), data = cars_log, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51137 -0.06050 -0.00183  0.06322  0.46792
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.398114   0.166554   8.394 8.99e-16 ***
## car_power          0.145663   0.005057  28.804  < 2e-16 ***
## log.acceleration. -0.191482   0.041722  -4.589 6.02e-06 ***
## model_year         0.029180   0.001810  16.122  < 2e-16 ***
## factor(origin)2    0.008272   0.019636   0.421    0.674
## factor(origin)3    0.019687   0.019395   1.015    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1199 on 386 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8756
## F-statistic: 551.6 on 5 and 386 DF,  p-value: < 2.2e-16
```

iii. 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?

```
# Standardizing the variables
cars_log_standardized <- scale(cars_log , center = TRUE , scale = FALSE)

# Make it as data.frame
cars_log_standardized <- as.data.frame(cars_log_standardized)

# Running the regression on standardized variables
```

```
lm2 <- lm(log.mpg. ~ car_power + log.acceleration. + model_year + factor(origin), data =
cars_log_standardized, na.action = na.exclude)
summary(lm2)

##
## Call:
## lm(formula = log.mpg. ~ car_power + log.acceleration. + model_year +
##     factor(origin), data = cars_log_standardized, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51137 -0.06050 -0.00183  0.06322  0.46792
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -0.005403   0.008714  -0.620    0.536
## car_power                       0.145663   0.005057  28.804  < 2e-16 ***
## log.acceleration.              -0.191482   0.041722  -4.589 6.02e-06 ***
## model_year                      0.029180   0.001810  16.122  < 2e-16 ***
## factor(origin)0.423469387755102  0.008272   0.019636   0.421    0.674
## factor(origin)1.4234693877551    0.019687   0.019395   1.015    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1199 on 386 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8756
## F-statistic: 551.6 on 5 and 386 DF,  p-value: < 2.2e-16
```

The importance of the `car_power` variable relative to the other predictors in the standardized regression can be assessed by comparing the coefficients, which now represent the change in `mpg` associated with a one-standard-deviation increase in the predictor. This can help in understanding the relative importance of the predictors.

Question 2)

Please download the Excel data file security_questions.xlsx from Canvas. In your analysis, you can either try to read the data sheet from the Excel file directly from R (there might be a package for that!) or you can try to export the data sheet to a CSV file before reading it into R.

```
# security_questions <- read_excel("D:/下載/security_questions.xlsx", sheet= 1)
data <- read_excel("D:/下載/security_questions.xlsx", sheet= 2)

#head(security_questions)
head(data)

## # A tibble: 6 × 18
##      Q1    Q2    Q3    Q4    Q5    Q6    Q7    Q8    Q9   Q10   Q11   Q12   Q13
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     7     5     5     7     7     4     4     7     5     7     5     7     5
## 2     5     5     6     6     6     5     5     7     5     6     6     6     6
## 3     6     6     6     6     7     6     6     6     5     7     6     6     5
## 4     5     5     5     5     5     5     5     5     5     5     5     5     4
## 5     7     7     7     7     7     4     5     7     6     7     6     7     6
## 6     6     5     4     5     4     4     4     5     6     2     5     5     5
```

```
## # … with 5 more variables: Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>,
## #   Q18 <dbl>
```

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.

a.

How much variance did each extracted factor explain?

```
# Principal components analysis
pca_result <- prcomp(data, scale. = TRUE)
summary(pca_result)

## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.0514 1.26346 1.07217 0.87291 0.82167 0.78209 0.70921
## Proportion of Variance 0.5173 0.08869 0.06386 0.04233 0.03751 0.03398 0.02794
## Cumulative Proportion  0.5173 0.60596 0.66982 0.71216 0.74966 0.78365 0.81159
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.68431 0.67229 0.6206 0.59572 0.54891 0.54063 0.51200
## Proportion of Variance 0.02602 0.02511 0.0214 0.01972 0.01674 0.01624 0.01456
## Cumulative Proportion  0.83760 0.86271 0.8841 0.90383 0.92057 0.93681 0.95137
##                           PC15   PC16   PC17   PC18
## Standard deviation     0.48433 0.4801 0.4569 0.4489
## Proportion of Variance 0.01303 0.0128 0.0116 0.0112
## Cumulative Proportion  0.96440 0.9772 0.9888 1.0000

# Eigenvalues
eigenvalues <- eigen(cor(data))$values
var_explained <- eigenvalues / length(eigenvalues)
var_explained

##  [1] 0.51727518 0.08868511 0.06386435 0.04233199 0.03750784 0.03398131
##  [7] 0.02794364 0.02601549 0.02510951 0.02139980 0.01971565 0.01673928
## [13] 0.01623763 0.01456354 0.01303216 0.01280357 0.01159706 0.01119690
```

b.

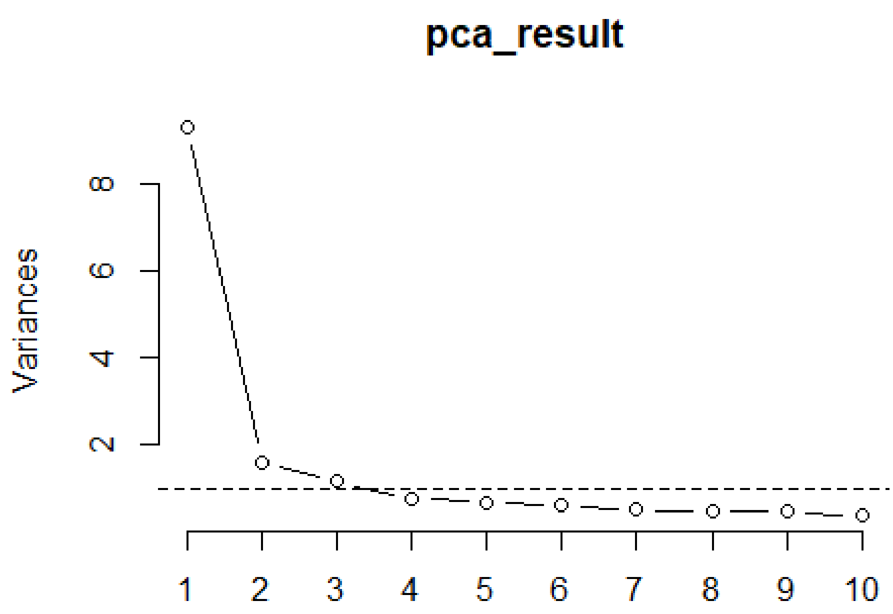
How many dimensions would you retain, according to the two criteria we discussed? (Eigenvalue ≥ 1 and Scree Plot – can you show the screeplot with eigenvalue=1 threshhold?)

```
# Eigenvalue ≥ 1
eigenvalues[eigenvalues >= 1]

## [1] 9.310953 1.596332 1.149558
```

ANS: retain 3 dimensions

```
screeplot(pca_result, type="lines") # Scree Plot : Q1~Q3 above the threshold
abline(h=1, lty="dashed")
```



c.

(ungraded) 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

---

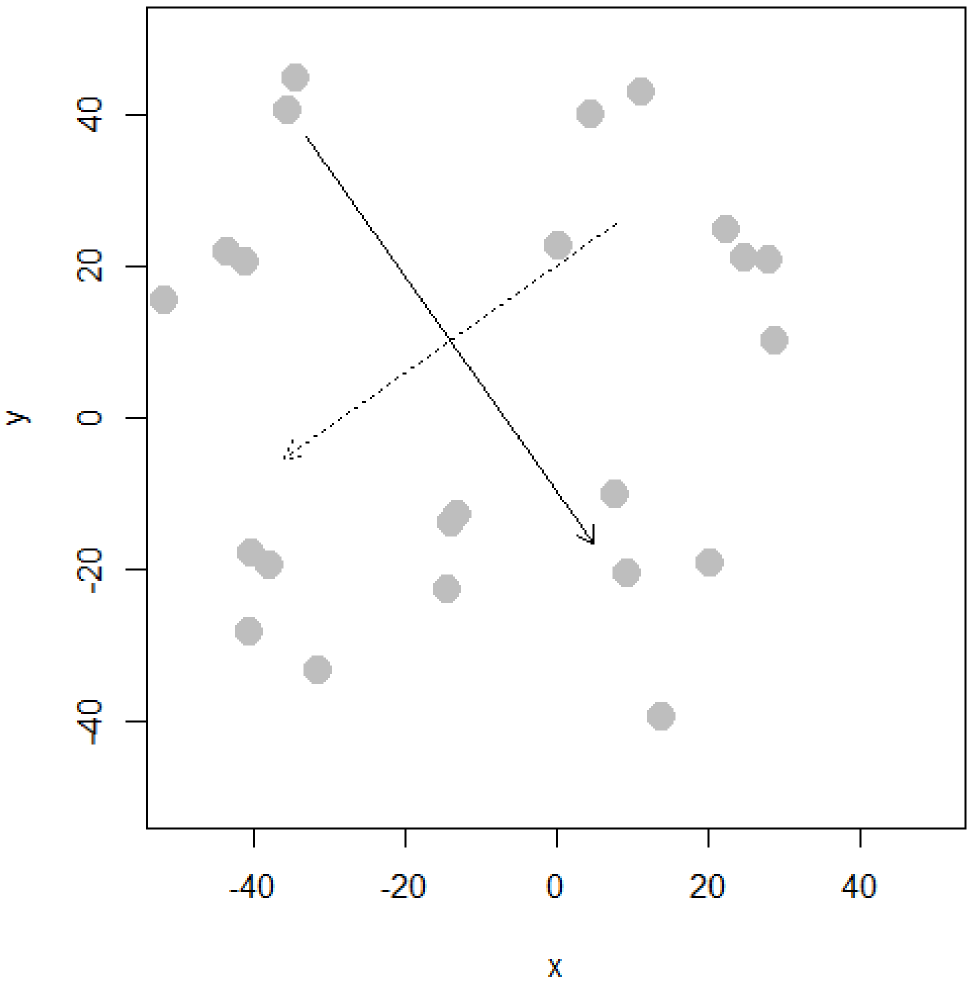
Question 3)

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

a.

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.

```
library(compstatslib)
# interactive_pca()
```



$pca Standard deviations (1, .., p=2):

[1] 27.84296 16.20438

Rotation (n x k) = (2 x 2):

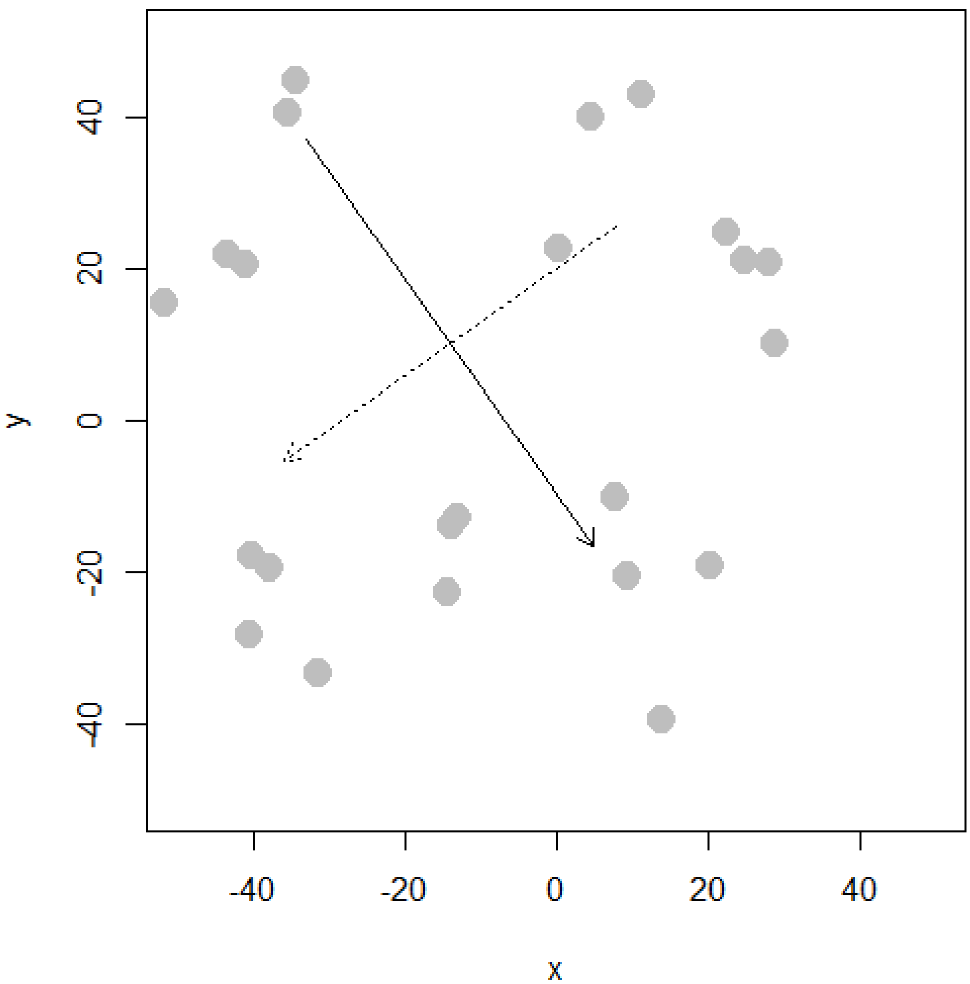
PC1 PC2

x -0.08066056 -0.99674163

y -0.99674163 0.08066056

b.

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



$pca

Standard deviations (1, .., p=2):

[1] 32.85468 26.78596

Rotation (n x k) = (2 x 2):

PC1 PC2

x 0.5779645 -0.8160619

y -0.8160619 -0.5779645