HW_Week13_108020033

Che-Wei, Chang

2023-05-09 helped by 108020024

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.

```
# Import the data
cars <- read.table("auto-data.txt", header = FALSE, na.strings = "?")
names(cars) <- c("mpg",  "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")

# Create data frame
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement),
                                  log(horsepower), log(weight), log(acceleration),
                                  model_year, origin))
cars_log  <- cars_log[complete.cases(cars_log),]
```

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?)

```
# Create a new data frame of four log-transformed variables with high multicollinearity
select_name <- c("log.cylinders.", "log.displacement.", "log.horsepower.", "log.weight.")
new_df <- subset(cars_log, select = select_name)
new_df <- new_df[complete.cases(new_df),]
```

(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)

```
# Show the variances of the four variables
new_df_eigen <- eigen(cor(new_df))
new_df_eigen$values[1] / sum(new_df_eigen$values)
```

```
## [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)

```
new_df_eigen$vectors
```

```
##            [,1]        [,2]        [,3]       [,4]
## [1,] -0.4979145  0.53580374 -0.52633608 -0.4335503
## [2,] -0.5122968  0.25665246  0.07354139  0.8162556
## [3,] -0.4856159 -0.80424467 -0.34193949 -0.0210980
## [4,] -0.5037960 -0.01530917  0.77500928 -0.3812031
```

```
# By the table, We can find that pc1 equally captures cylinders, displacement, horsepower,
# and weight; pc2 captures mostly horsepower; pc3 mostly caputure weight; pc4 captures displacement.
```

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.)

```
# Store the scores of the first principal component as a new column
pca <- prcomp(cars_log, scale. = FALSE)
cars_log$PC1 <- pca$x[, 1]
```

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

```
# Create linear model
lm_model <- lm(log.mpg. ~ PC1 + log.acceleration. +
                 model_year + factor(origin), data = cars_log)

# Show the summary of the linear model
summary(lm_model)
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1 + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42623 -0.05333  0.00096  0.04864  0.39217
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       340.82898    8.58642  39.694  < 2e-16 ***
## PC1                 4.47778    0.11240  39.838  < 2e-16 ***
## log.acceleration.  -0.28591    0.03331  -8.584 2.27e-16 ***
## model_year         -4.43313    0.11232 -39.469  < 2e-16 ***
## factor(origin)2    -0.22934    0.01869 -12.269  < 2e-16 ***
## factor(origin)3    -0.41664    0.02269 -18.364  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09413 on 386 degrees of freedom
## Multiple R-squared:  0.9244, Adjusted R-squared:  0.9234
## F-statistic: 943.4 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?

```r
# Standardized the data
cars_log_std <- scale(cars_log)
cars_log_std <- as.data.frame(cars_log_std)

# Create linear model
lm_model_std <- lm(log.mpg. ~ PC1 + log.acceleration. + model_year +
                   factor(origin), data = cars_log_std)

# Show the summary of linear model
summary(lm_model_std)
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1 + log.acceleration. + model_year +
##     factor(origin), data = cars_log_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25347 -0.15685  0.00284  0.14303  1.15331
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       0.36393    0.02535  14.356  < 2e-16 ***
## PC1                              48.75357    1.22380  39.838  < 2e-16 ***
## log.acceleration.                -0.15215    0.01772  -8.584 2.27e-16 ***
## model_year                      -48.02528    1.21679 -39.469  < 2e-16 ***
## factor(origin)0.525710525810929  -0.67445    0.05497 -12.269  < 2e-16 ***
## factor(origin)1.76714743013553   -1.22529    0.06672 -18.364  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2768 on 386 degrees of freedom
## Multiple R-squared:  0.9244, Adjusted R-squared:  0.9234
## F-statistic: 943.4 on 5 and 386 DF,  p-value: < 2.2e-16
```

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.

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.

```
# Import the library
library(readxl)
```

```
# Read the data
sec_que_q <- read_excel("security_questions.xlsx", sheet = "questions")
sec_que_d <- read_excel("security_questions.xlsx", sheet = "data")
```

a. How much variance did each extracted factor explain?

```
pca_sq_d <- prcomp(sec_que_d, scale. = TRUE)
var_ext <- pca_sq_d$sdev^2 / sum(pca_sq_d$sdev^2)
var_ext
```

```
##  [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
```

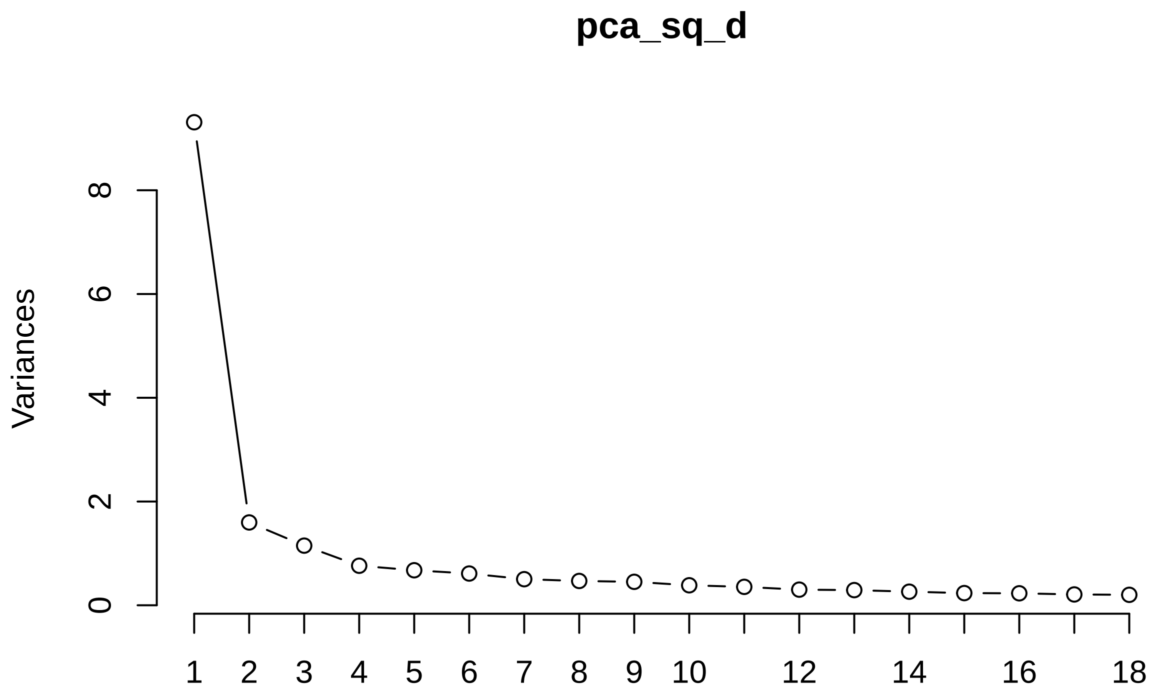
b. How many dimensions would you retain, according to the two criteria we discussed?

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

```
# Check for eigenvalue >= 1
Data_eigen <- eigen(cor(sec_que_d))
Data_eigen$values
```

```
##  [1] 9.3109533 1.5963320 1.1495582 0.7619759 0.6751412 0.6116636 0.5029855
##  [8] 0.4682788 0.4519711 0.3851964 0.3548816 0.3013071 0.2922773 0.2621437
## [15] 0.2345788 0.2304642 0.2087471 0.2015441
```

```
# By using scree plot
screeplot(pca_sq_d, type = "lines", npcs = length(pca_sq_d$sdev))
```



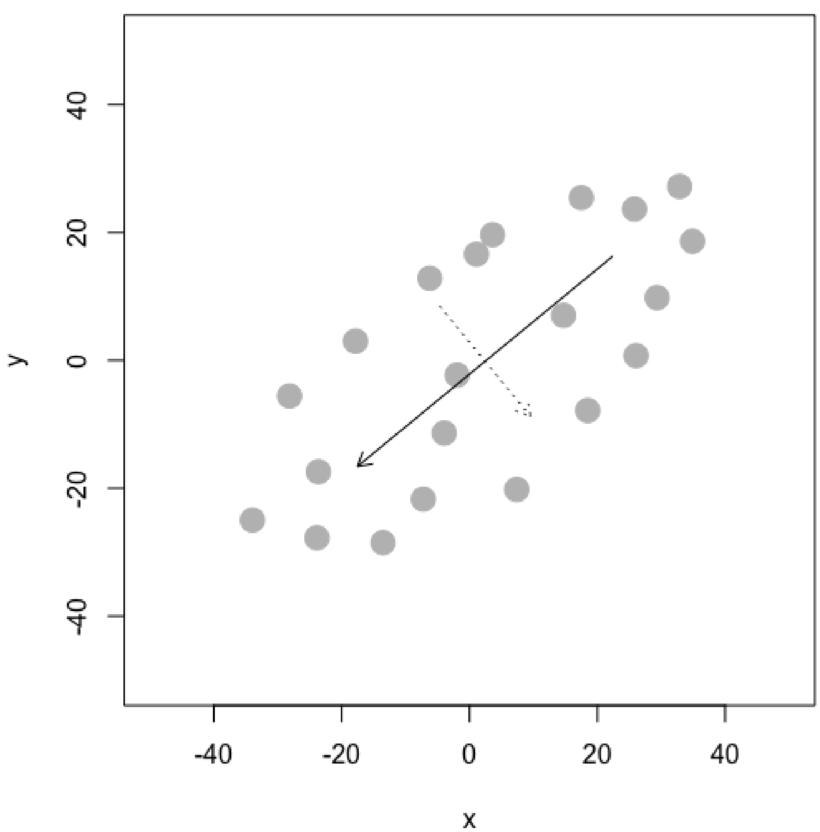

```
# By these two criteria, We can find that only 3 eigenvalues >= 1 in
# eigenvalue >= 1 criteria. Also, 3 points would be retain by using
# eigenvalue = 1 threshold. Therefore, we would retain 3 dimension.
```

Question 3) Let's simulate how principal components behave interactively: run the interactive_pca() function from the compstatslib package we have used earlier:

```
# Import library
library(compstatslib)
```

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.

```
knitr::include_graphics("plot1.png")
```



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

```
knitr::include_graphics("plot2.png")
```