HW14

106022103

2021/5/29

Assist

- 106000199
  - Discussion about the PCA.
  - Bootstrap function.

Set up

import libary

```
library(ggplot2)
require(qqplotr)
library(plyr)
library(gridExtra)
library(ggcorrplot)
library(magrittr)
library(ggpubr)
library(car)
library(corrplot)
library(openxlsx) # install.packages("openxlsx")
library(psycho) #install.packages("psycho")
```

Q1

Read file

```
cars <- read.table("data/auto-data.txt", header=FALSE, na.strings = "?")
names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
cars[,'origin']<-factor(cars[,'origin']) # convert to factor
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement), log(horsepower), log(wei
cars_log <- cars_log[complete.cases(cars_log), ] # remove missing value
cars_log_keep <- cars_log[,c("log.mpg.", "log.weight.", "log.cylinders.", "log.acceleration.", "model_y
```

a.

```
model1 <- lm(log.weight. ~ log.cylinders.,  data = cars_log)
summary(model1)
```

i. Model 1: Regress `log.weight.` over `log.cylinders.` only and report the coefficient

```
##
```

```
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35409 -0.09030 -0.00169  0.09271  0.40488
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.60059    0.03710  177.92   <2e-16 ***
## log.cylinders.   0.82187    0.02208   37.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1319 on 390 degrees of freedom
## Multiple R-squared:  0.7804, Adjusted R-squared:  0.7798
## F-statistic:  1386 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
model2 <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + origin,  data = cars_log)
summary(model2)
```

ii. Model 2: Regress `log.mpg.` over `log.weight.` and all control variables and report the coefficient (check whether weight has a significant direct effect on mpg with other variables statistically controlled?)

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38259 -0.07054  0.00401  0.06696  0.39798
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.410974   0.316806  23.393  < 2e-16 ***
## log.weight.       -0.875499   0.029086 -30.101  < 2e-16 ***
## log.acceleration.  0.054377   0.037132   1.464  0.14389
## model_year         0.032787   0.001731  18.937  < 2e-16 ***
## origin2            0.056111   0.018241   3.076  0.00225 **
## origin3            0.031937   0.018506   1.726  0.08519 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1163 on 386 degrees of freedom
## Multiple R-squared:  0.8845, Adjusted R-squared:  0.883
## F-statistic: 591.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

b. What is the indirect effect of cylinders on mpg? (use the product of slopes between model 1 & 2)

```r
sprintf("The indrect effect of cylinders on mpg is %.4f",
        (model1$coefficients["log.cylinders."] * model2$coefficients["log.weight."]))
```

```
## [1] "The indrect effect of cylinders on mpg is -0.7195"
```

c. Let's bootstrap for the confidence interval of the indirect effect of cylinders on mpg

```r
boot_indirect <- function(df){
  df_boot <- df[sample(nrow(df),nrow(df),replace=TRUE),]
  model1 <- model1 <- lm(log.weight. ~ log.cylinders.,  data = df_boot)
  model2 <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + origin,  data = df_boot)
  return (model1$coefficients["log.cylinders."] * model2$coefficients["log.weight."])
}

indirect_boots <- replicate(2000, boot_indirect(cars_log))
quantile(indirect_boots, probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7827218 -0.6613995
```

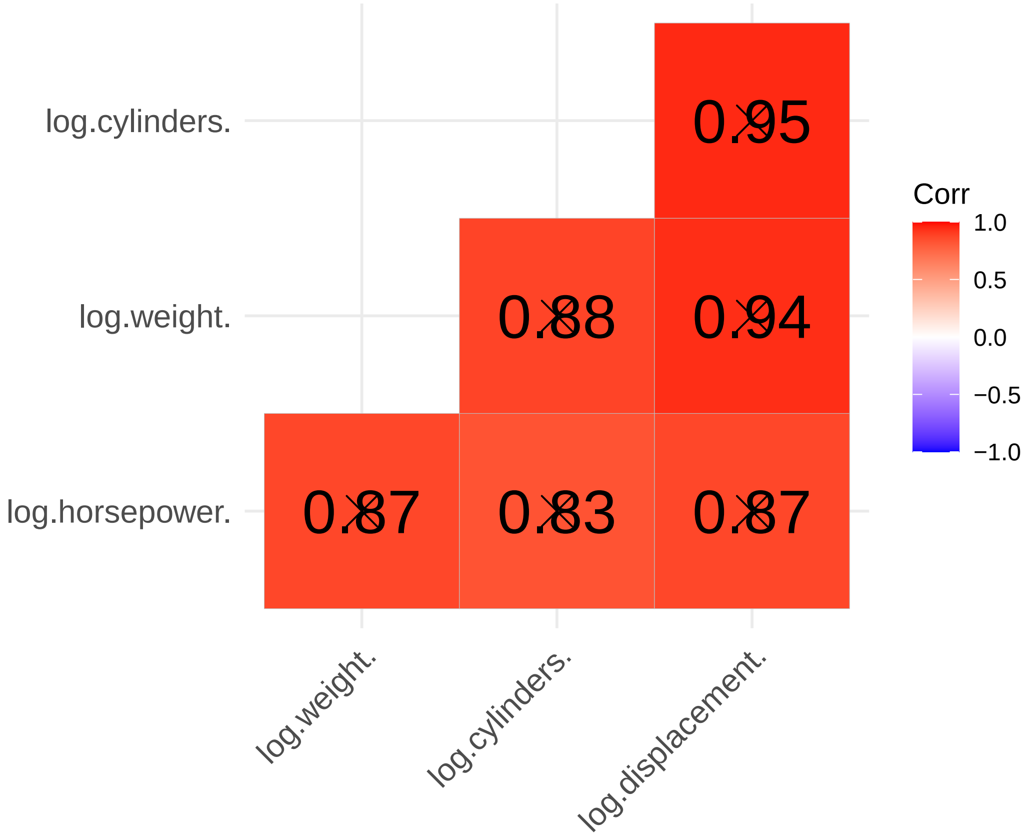
Q2

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

```r
cars_log_colinear <- cars_log[,c("log.cylinders.","log.displacement.","log.horsepower.","log.weight.")]
cor_m <- cor(cars_log_colinear)

cor_raw <- round(cor_m,2)
p.raw_mat <- cor_pmat(cor_m)
ggcorrplot(t(cor_raw), hc.order = TRUE,
  type = "lower", p.mat = t(p.raw_mat), lab = TRUE,lab_size = 8)
```

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




```
eigenvalue <- eigen(cor_m)$values
eigenvectors <- eigen(cor_m)$vectors

# compute from eigenvalues
eigenvalue / sum(eigenvalue)
```

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

```
## [1] 0.918564696 0.046906929 0.025981967 0.008546408
```

```
# check with summary of pca
pca <- prcomp(cars_log_colinear,scale. = T)
summary(pca)
```

```
## 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
```

iii. Looking at the values and valence (positive/negative) of the first principal component's eigenvector, what would you call the information captured by this component? ANSWER: Because of the high co-linearity between the four variables, the value of the first principal component is higher and all components are positive.

b. Let's revisit our regression analysis on cars_log:

```
cars_log$pc1 <- pca$x[,1]
```

i. Store the scores of the first principal component as a new column of cars_log

```
model3 <- lm(log.mpg. ~ pc1 + log.acceleration. + model_year + origin,  data = cars_log)
summary(model3)
```

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

```
##
## Call:
## lm(formula = log.mpg. ~ pc1 + log.acceleration. + model_year +
##     origin, data = cars_log)
##
## 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 ***
## pc1                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 ***
## origin2            0.008272   0.019636   0.421    0.674
## origin3            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
```

```
cars_log_std <- cars_log[-9]
cars_log_colinear_std <- scale(cars_log_colinear)
cars_log_std[,names(cars_log_colinear)] <- cars_log_colinear_std
pca_std <- prcomp(cars_log_colinear_std,scale. = T)
cars_log_std$pc1_std <- pca_std$x[,1]
model4 <- lm(log.mpg. ~ pc1_std + log.acceleration. + model_year + origin,  data = cars_log_std)
summary(model4)
```

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?

```
##
## Call:
## lm(formula = log.mpg. ~ pc1_std + log.acceleration. + model_year +
##     origin, data = cars_log_std)
##
## 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 ***
## pc1_std            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 ***
## origin2            0.008272   0.019636   0.421    0.674    
## origin3            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
```

ANSWER: The importance are same.

Q3

Read File

```r
data <- read.xlsx(xlsxFile="data/security_questions.xlsx", sheet = 2, colNames = TRUE)
```

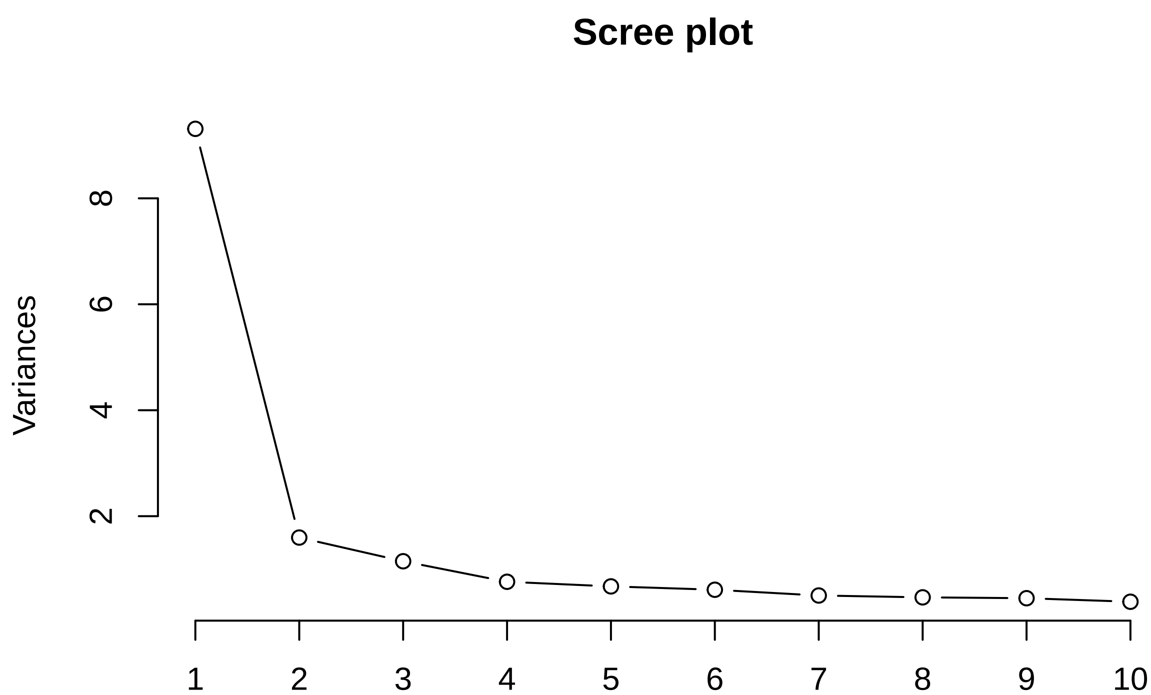
a. How much variance did each extracted factor explain?

```r
summary(prcomp(data,scale. = T))
```

```
## 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
```

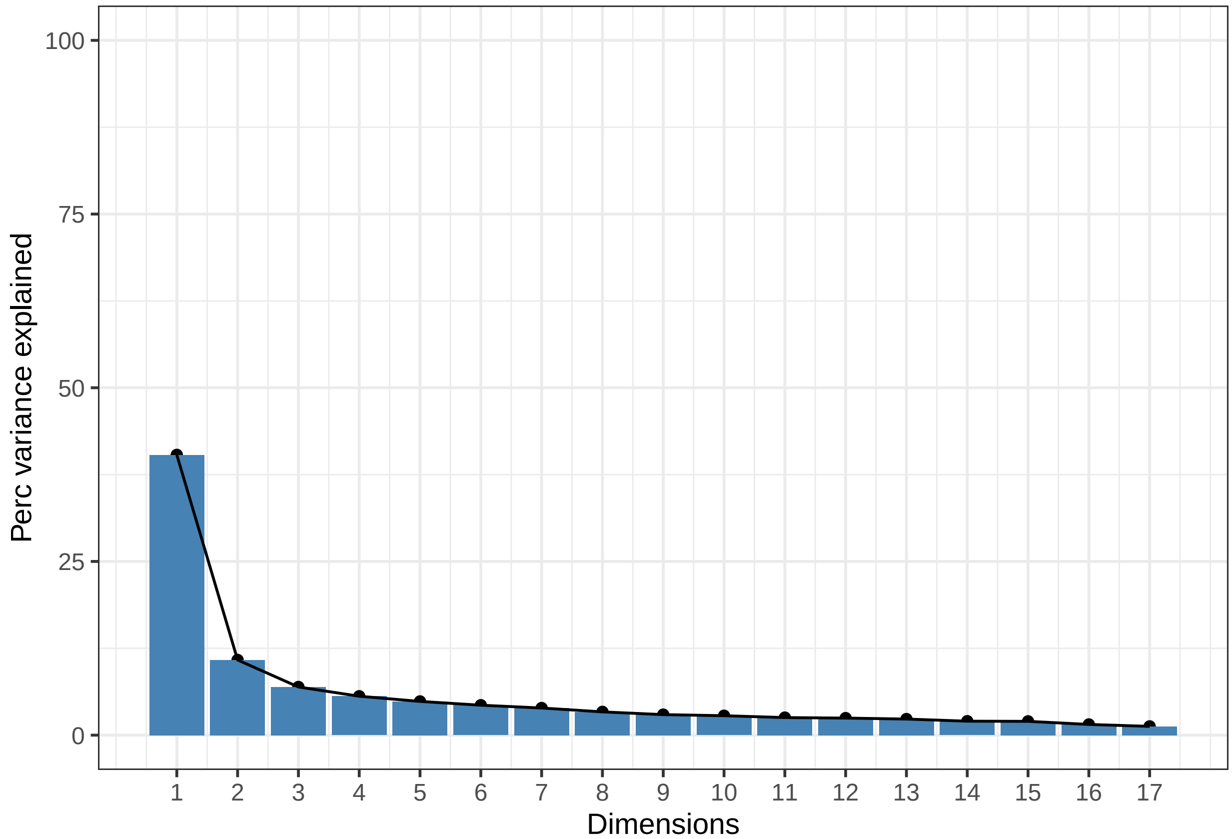
b. How many dimensions would you retain, according to the criteria we discussed?

```r
# Use built-in plot
screeplot(prcomp(data,scale.=TRUE),type = "line",main = "Scree plot")
```




```
# use ggplot
res.pca = prcomp(log2(data[,-ncol(data)]+1))
varExp = (100*res.pca$sdev^2)/sum(res.pca$sdev^2)
varDF = data.frame(Dimensions=1:length(varExp),
                   varExp=varExp)

ggplot(varDF,aes(x=Dimensions,y=varExp)) + geom_point() +
  geom_col(fill="steelblue") + geom_line() +
  theme_bw() + scale_x_continuous(breaks=1:nrow(varDF)) +
  ylim(c(0,100)) + ylab("Perc variance explained")
```

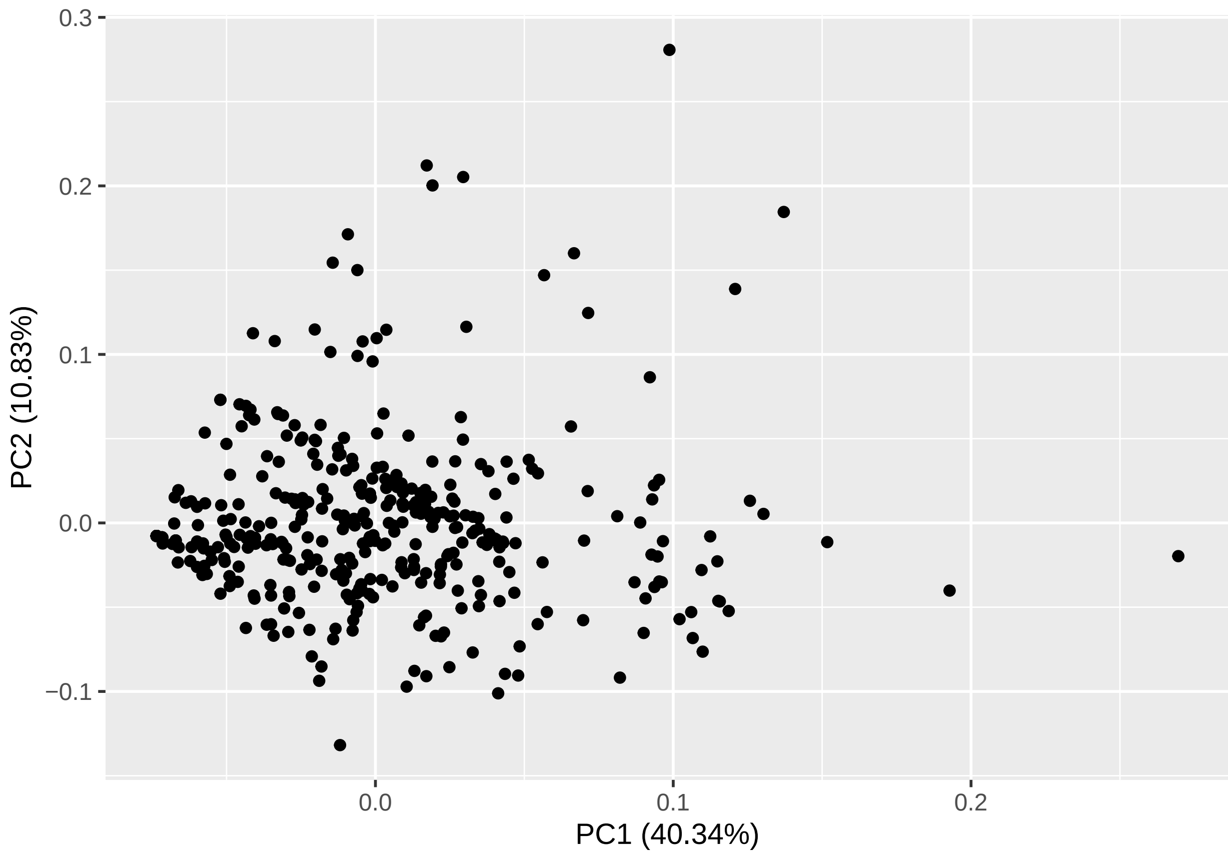



```
eigenvalues <- eigen(cor(data))$values
sprintf("We should retain %d dimensions ", length(eigenvalues[eigenvalues>1]))
```

```
## [1] "We should retain 3 dimensions "
```

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

```
library(patchwork)
library(ggfortify)
autoplot(res.pca,data=data)
```




```
print("The components of PC1")
```

```
## [1] "The components of PC1"
```

```
eigen(cor(data))$vectors[,1]
```

```
##  [1] -0.2677422 -0.2204272 -0.2508767 -0.2042919 -0.2261544 -0.2237681
##  [7] -0.2151891 -0.2576225 -0.2369512 -0.2248660 -0.2467645 -0.2065785
## [13] -0.2333066 -0.2659342 -0.2307289 -0.2482681 -0.2023781 -0.2643810
```

```
print("The components of PC2")
```

```
## [1] "The components of PC2"
```

```
eigen(cor(data))$vectors[,2]
```

```
##  [1]  0.110341691  0.010886972  0.025878543 -0.508981768  0.024745268
##  [6]  0.082805088  0.251398450 -0.033526840  0.183342667  0.078103267
## [11]  0.206580870 -0.504591429  0.051159791  0.078910404 -0.008373326
## [16]  0.160524168 -0.525747030  0.089915229
```

```
print("The components of PC3")
```

```
## [1] "The components of PC3"
```

```
eigen(cor(data))$vectors[,3]
```

```
##  [1] -0.001973491  0.083171536  0.083648794  0.100759585 -0.505845415
##  [6]  0.193281966  0.302354487 -0.320109219  0.189853454 -0.496820932
## [11]  0.160903091  0.113342400  0.078658760  0.146232765 -0.310161141
```

```
## [16]  0.170839887  0.102652280 -0.060800871
```

ANSWER: From the composition of PC1 looks like the average of all the problems, PC2 looks like the opposite of Q4,Q11,Q17, PC3 looks like the opposite of Q5,Q10.

Reference Link

- Read .xlsx file in R
- How to set scree plot scale as same as principal components?
- How To Make Scree Plot in R with ggplot2?
- Sample random rows in dataframe
- standardize.data.frame: Standardize (scale and reduce) Dataframe.