BACS HW13 - 109006234

Credit: 109006278

May 14th 2023

Loading the data

```
cars <- read.table("G:/My Drive/111_2_BACS/HW13/auto-data.txt",
    header=FALSE, na.strings = "?")
names(cars) <- c("mpg", "cylinders", "displacement",
                "horsepower", "weight", "acceleration",
                "model_year", "origin", "car_name")
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement),
        log(horsepower), log(weight), log(acceleration), model_year, origin))
cars_log <- na.omit(cars_log)
```

Problem 1

(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

```
trf_var <- cars_log[, c("log.cylinders.", "log.displacement.",
        "log.horsepower.", "log.weight.")]
summary(trf_var)
```

```
##  log.cylinders.  log.displacement. log.horsepower.  log.weight.
##  Min.   :1.099   Min.   :4.220     Min.   :3.829   Min.   :7.386
##  1st Qu.:1.386   1st Qu.:4.654     1st Qu.:4.317   1st Qu.:7.708
##  Median :1.386   Median :5.017     Median :4.538   Median :7.939
##  Mean   :1.653   Mean   :5.128     Mean   :4.588   Mean   :7.959
##  3rd Qu.:2.079   3rd Qu.:5.618     3rd Qu.:4.836   3rd Qu.:8.193
##  Max.   :2.079   Max.   :6.120     Max.   :5.438   Max.   :8.545
```

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

```
round(cor(trf_var), 2)
```

```
##                   log.cylinders. log.displacement. log.horsepower. log.weight.
## log.cylinders.              1.00              0.95            0.83        0.88
## log.displacement.           0.95              1.00            0.87        0.94
## log.horsepower.             0.83              0.87            1.00        0.87
## log.weight.                 0.88              0.94            0.87        1.00
```

```r
cars_pca <- eigen(cor(trf_var))
cars_pca$values
```

```
## [1] 3.67425879 0.18762771 0.10392787 0.03418563
```

```r
cars_pca$values[1]/sum(cars_pca$values)
```

```
## [1] 0.9185647
```

The initial principal component accounts for 91.86% of the variation in the four variables.

(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?

```r
cars_eigenvec <- cars_pca$vectors
cars_eigenvec[,1]
```

```
## [1] -0.4979145 -0.5122968 -0.4856159 -0.5037960
```

The values for all variables are roughly -0.5, indicating that the initial principal component has an adverse effect on mpg. Hence, it is plausible that the first principal component pertains to the engine size.

(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

```r
cars_pca <- prcomp(trf_var)
summary(cars_pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     0.7312 0.15174 0.09535 0.07272
## Proportion of Variance 0.9346 0.04025 0.01589 0.00924
## Cumulative Proportion  0.9346 0.97486 0.99076 1.00000
```

```r
scores <- cars_pca$x |> round(3)
head(scores, 5)
```

```
##      PC1    PC2    PC3    PC4
## 1 -0.796  0.105 -0.121 -0.010
## 2 -1.013 -0.058 -0.116 -0.067
## 3 -0.876 -0.007 -0.159 -0.052
## 4 -0.843 -0.023 -0.167 -0.027
## 5 -0.811  0.035 -0.150 -0.016
```

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

```
cars_log$engine_size_PC1 <- -1*scores[,"PC1"]
pc1 <- cars_log$engine_size_PC1
reg_pca <- lm(log.mpg. ~ pc1 + log.acceleration. + model_year + factor(origin),
        data=as.data.frame(scale(cars_log)))
summary(reg_pca)
```

```
##
## Call:
## lm(formula = log.mpg. ~ pc1 + log.acceleration. + model_year +
##     factor(origin), data = as.data.frame(scale(cars_log)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.57642 -0.18084  0.00397  0.18484  1.49751
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.004182   0.026913   0.155    0.877
## pc1                           -1.138369   0.041499 -27.431  < 2e-16 ***
## log.acceleration.             -0.101044   0.023016  -4.390 1.46e-05 ***
## model_year                     0.316820   0.020273  15.628  < 2e-16 ***
## factor(origin)0.525710525810929 -0.031933   0.060992  -0.524    0.601
## factor(origin)1.76714743013553   0.006647   0.060339   0.110    0.912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3644 on 386 degrees of freedom
## Multiple R-squared:  0.8689, Adjusted R-squared:  0.8672
## F-statistic: 511.7 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?

```
regr_pca_std <- lm(scale(log.mpg.) ~ scale(pc1) + scale(log.acceleration.) +
        model_year + factor(origin), data=as.data.frame(scale(cars_log)))
summary(regr_pca_std)
```

```
##
## Call:
## lm(formula = scale(log.mpg.) ~ scale(pc1) + scale(log.acceleration.) +
##     model_year + factor(origin), data = as.data.frame(scale(cars_log)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.57642 -0.18084  0.00397  0.18484  1.49751
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.004200   0.026914   0.156    0.876
## scale(pc1)                    -0.832370   0.030344 -27.431  < 2e-16 ***
## scale(log.acceleration.)      -0.101044   0.023016  -4.390 1.46e-05 ***
## model_year                     0.316820   0.020273  15.628  < 2e-16 ***
## factor(origin)0.525710525810929 -0.031933   0.060992  -0.524    0.601
```

```
## factor(origin)1.76714743013553   0.006647   0.060339   0.110    0.912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3644 on 386 degrees of freedom
## Multiple R-squared:  0.8689, Adjusted R-squared:  0.8672
## F-statistic: 511.7 on 5 and 386 DF,  p-value: < 2.2e-16
```

PC1 is highly significant both prior to and after standardization. The reason for this could be that the data follows a normal distribution, resulting in no significant deviation.

Problem 2

```r
security <- read_excel(
        "G:/My Drive/111_2_BACS/HW13/security_questions.xlsx",
        sheet = "data")
```

(a) How much variance did each extracted factor explain?

```r
security_eigen <- eigen(cor(security))
security_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
```

```r
security_pca <- prcomp(security, scale. = TRUE)
sec_rotation <- security_pca$rotation[, 1:3] |> round(2)
summary(security_pca)
```

```
## 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
```

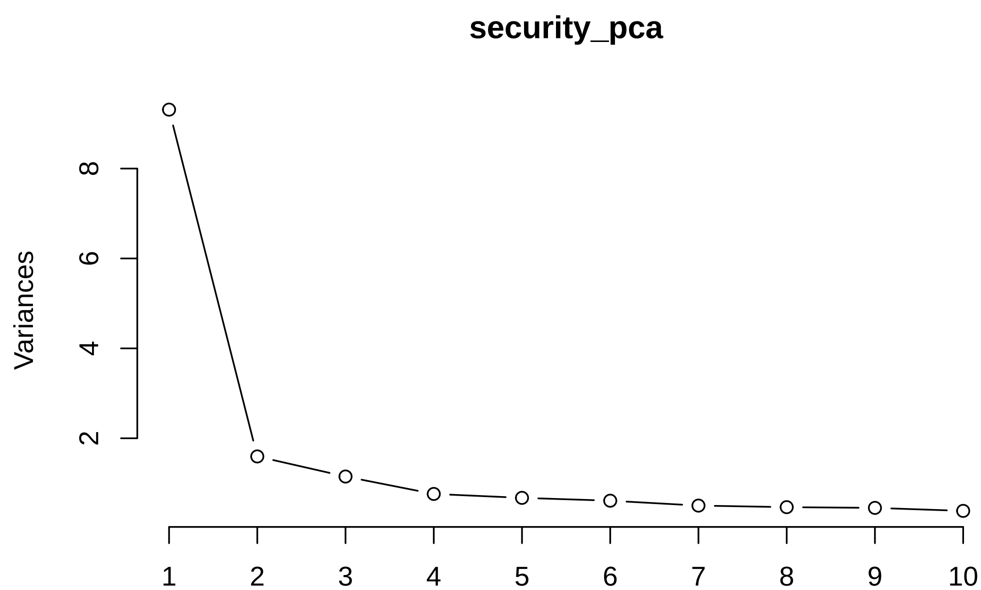
```r
security_eigen$values[1] / sum(security_eigen$values)
```

```
## [1] 0.5172752
```

Each of the extracted factors accounts for 51.73% of the variance.

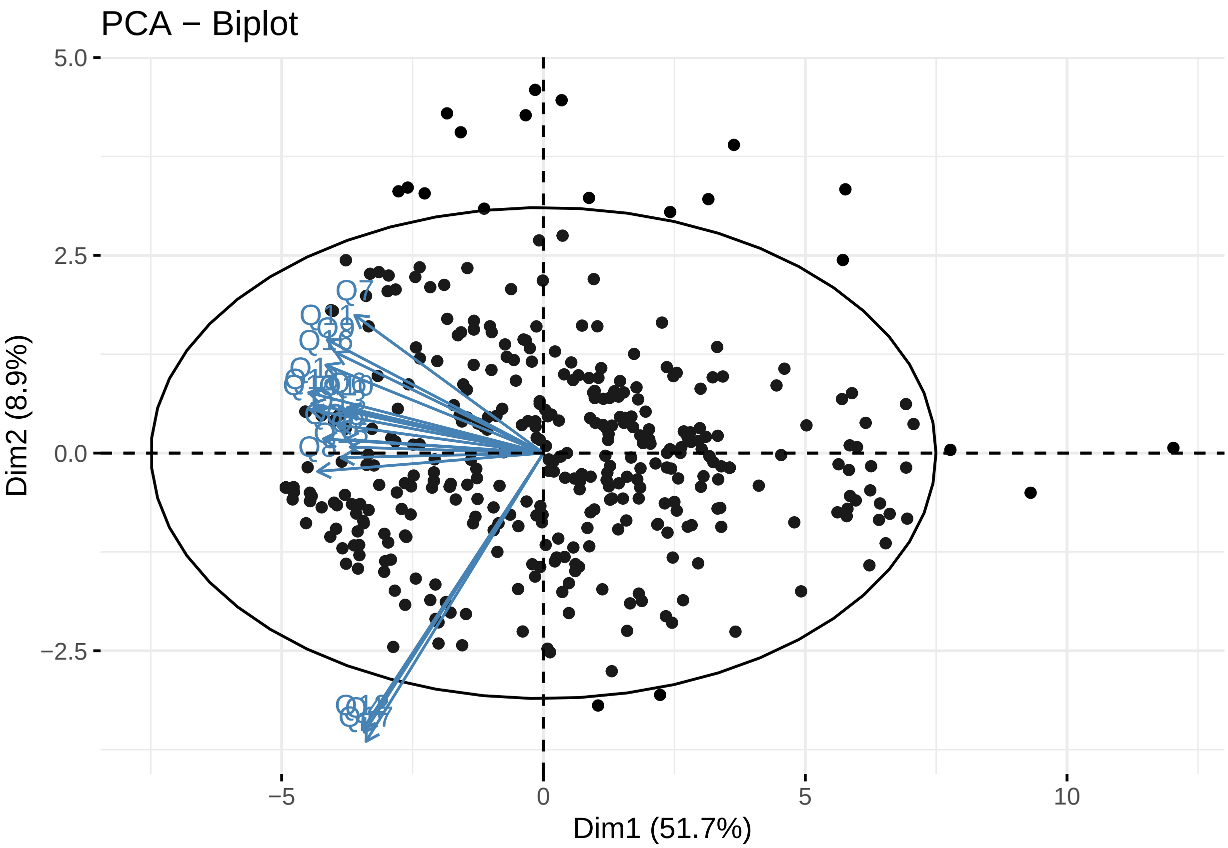
(b) How many dimensions would you retain, according to the two criteria we discussed? (Eigenvalue 1 and Scree Plot – can you show the screeplot with eigenvalue=1 threshhold?)

```r
screeplot(security_pca, type="lines")
```



(c) 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

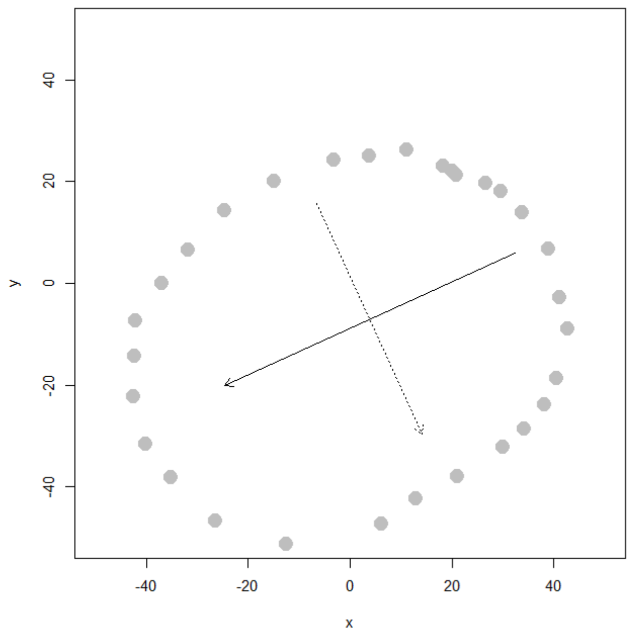
```r
fviz_pca_biplot(security_pca, label = "var", addEllipses = T,
        itle = "PCA - Biplot of Security Questions")
```

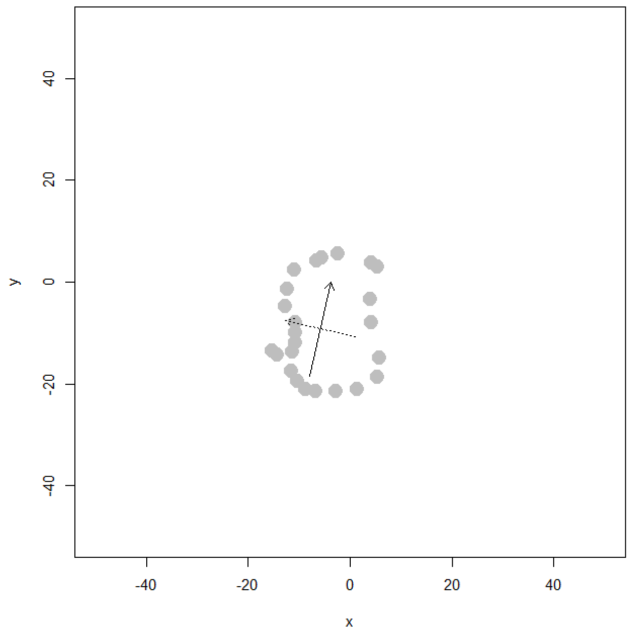


Eigenvalues that are approximately -0.2 are present in PC1, indicating that this dimension represents security confidentiality as a whole and includes all questions from Q1 to Q18. In PC2, Q4, Q12, and Q17 exhibit a strong negative correlation, potentially due to their association with transaction record-keeping by the website. Negative values for these questions imply that individuals may lack confidence in the website's ability to maintain accurate transaction records. Conversely, PC3 displays a substantial negative correlation with Q5, Q8, Q10, and Q15, which relate to the website's identity verification process prior to granting access. Negative values for these questions indicate that people may be skeptical about the website's ability to prevent unauthorized access to their accounts.

Problem 3

(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.





40
20
0
-20
-40
y
-40
-20
0
20
40
x

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