Student ID: 112077423

```r
library(ggplot2)
library(compstatslib)
library(data.table)
library(tidyr)
library(lsa)
```

```
## Warning:     'lsa'           R     4.3.3
```

```r
library(readxl)
```

```
## Warning:     'readxl'           R     4.3.3
```

```r
cars <- read.table("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))
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
## 1          2.484907         70      1
## 2          2.442347         70      1
## 3          2.397895         70      1
## 4          2.484907         70      1
## 5          2.351375         70      1
## 6          2.302585         70      1
```

```r
# print rows with missing values
print(cars_log[!complete.cases(cars_log),])
```

```
##     log.mpg. log.cylinders. log.displacement. log.horsepower. log.weight.
## 33  3.218876       1.386294          4.584967              NA    7.623642
## 127 3.044522       1.791759          5.298317              NA    7.963808
## 331 3.711130       1.386294          4.442651              NA    7.514800
## 337 3.161247       1.386294          4.941642              NA    7.974189
## 355 3.540959       1.386294          4.605170              NA    7.749322
## 375 3.135494       1.386294          5.017280              NA    8.017967
##     log.acceleration. model_year origin
```

```
## 33           2.944439         71      1
## 127          2.833213         74      1
## 331          2.850707         80      2
## 337          2.660260         80      1
## 355          2.760010         81      2
## 375          3.020425         82      1
```

```r
# delete rows with missing values
cars_log <- cars_log %>% drop_na()
```

Question 1(a)

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

```r
cor(cars_log)
```

```
##                      log.mpg. log.cylinders. log.displacement. log.horsepower.
## log.mpg.            1.0000000     -0.8215060        -0.8600904      -0.8501157
## log.cylinders.     -0.8215060      1.0000000         0.9469109       0.8265831
## log.displacement. -0.8600904      0.9469109         1.0000000       0.8721494
## log.horsepower.   -0.8501157      0.8265831         0.8721494       1.0000000
## log.weight.       -0.8745110      0.8833950         0.9428497       0.8739558
## log.acceleration.  0.4652735     -0.5039591        -0.5242124      -0.7162923
## model_year         0.5772748     -0.3368039        -0.3297774      -0.3970777
## origin             0.5605076     -0.5822814        -0.6714876      -0.4829311
##                   log.weight. log.acceleration. model_year     origin
## log.mpg.           -0.8745110         0.4652735  0.5772748  0.5605076
## log.cylinders.      0.8833950        -0.5039591 -0.3368039 -0.5822814
## log.displacement.   0.9428497        -0.5242124 -0.3297774 -0.6714876
## log.horsepower.     0.8739558        -0.7162923 -0.3970777 -0.4829311
## log.weight.         1.0000000        -0.4249531 -0.2870883 -0.6088750
## log.acceleration.  -0.4249531         1.0000000  0.3130780  0.2275799
## model_year         -0.2870883         0.3130780  1.0000000  0.1815277
## origin             -0.6088750         0.2275799  0.1815277  1.0000000
```

```r
features <- cars_log[,c('log.cylinders.', 'log.displacement.', 'log.weight.', 'log.horsepower.')]
```

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

```r
features_eigen <- eigen(cor(features))
features_eigen$values[1] / sum(features_eigen$values)
```

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

PC1 captures ~92% of variance of the dataset.

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
tmp <- features_eigen$vectors
rownames(tmp) <- c('log.cylinders.', 'log.displacement.', 'log.weight.', 'log.horsepower.')
tmp[,1]
```

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

The first principal component is the longest dimension of the data. Eigenvectors of the covariance matrix are actually the directions of the axes where there is the most variance (most information). Since the first principal component is strongly correlated with log.displacement. and log.weight. (>0.5), we can say that this principal component is primarily a measure of the displacement and weight.

Question 1(b)

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

```r
scores <- prcomp(features, scale.=TRUE)$x
cars_log$pc_scores <- scores[,'PC1']
```

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

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

```
##
## Call:
## lm(formula = log.mpg. ~ pc_scores + log.acceleration. + model_year +
##     factor(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 ***
## pc_scores          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
```

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
model <- lm(scale(log.mpg.) ~ scale(pc_scores) + scale(log.acceleration.) +
            scale(model_year) + factor(origin), data=cars_log)
summary(model)
```

```
##
## Call:
## lm(formula = scale(log.mpg.) ~ scale(pc_scores) + scale(log.acceleration.) +
##     scale(model_year) + factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50385 -0.17791 -0.00538  0.18591  1.37608
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.01589    0.02563  -0.620    0.536
## scale(pc_scores)          0.82112    0.02851  28.804  < 2e-16 ***
## scale(log.acceleration.) -0.10190    0.02220  -4.589 6.02e-06 ***
## scale(model_year)         0.31611    0.01961  16.122  < 2e-16 ***
## factor(origin)2           0.02433    0.05775   0.421    0.674
## factor(origin)3           0.05790    0.05704   1.015    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3526 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
```

We can see that pc_scores has the highest coefficient among all variables which means that it has a larger influence.

Question 2(a)

```r
df <- read_excel('security_questions.xlsx', sheet = 2)
head(df)
```

```
## # A tibble: 6 x 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
## # i 5 more variables: Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>, Q18 <dbl>
```

How much variance did each extracted factor explain?

```r
df_pca <- prcomp(df, scale.=TRUE)
summary(df_pca)$importance[2,]
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
## 0.51728 0.08869 0.06386 0.04233 0.03751 0.03398 0.02794 0.02602 0.02511 0.02140
##    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18
## 0.01972 0.01674 0.01624 0.01456 0.01303 0.01280 0.01160 0.01120
```

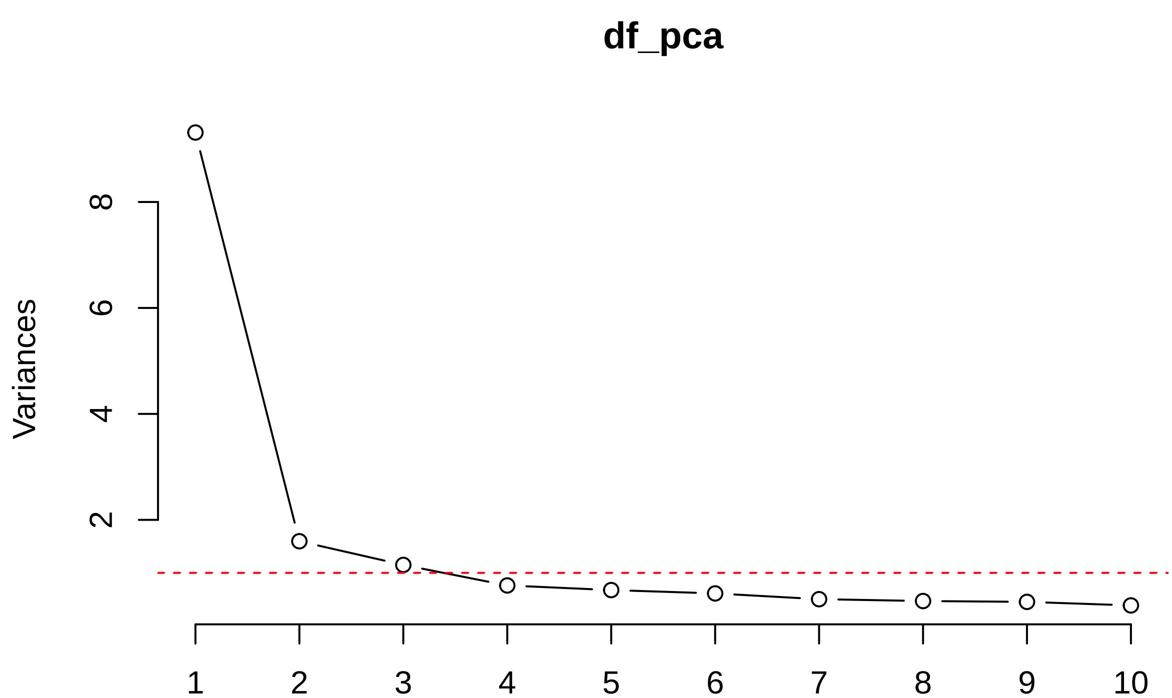
Question 2(b)

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

```r
df_eigen <- eigen(cor(df))
df_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
screeplot(df_pca, type="lines")
abline(h=1, col="red", lty=2)
```



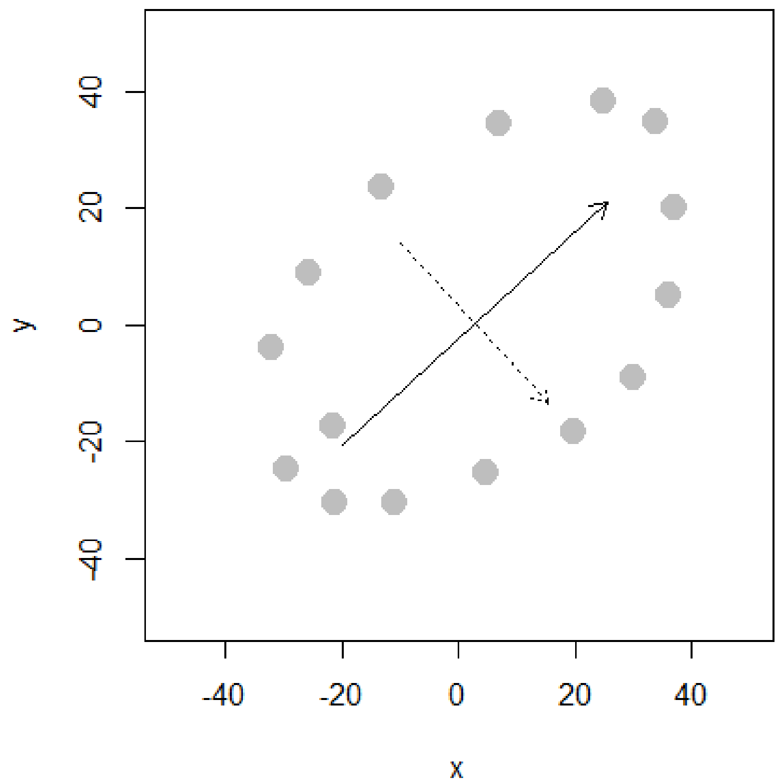
I'd retain only one dimension. Despite the first 3 PCs having eigenvalues >= 1, not each of them follows screeplot criteria (where we only consider factors before the "elbow").

Question 2(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

The main idea of PCA is to reduce the dimensionality of a dataset. Principal components are variables that explain variation in a dataset, where the first principal component explains the most of it, and each remaining component explains the remaining variance in decreasing order.

Question 3(a)



Question 3(b)