HW14

106062137

Question 1

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

# create a new data set that log-transforms several variables from our original data set
cars_log <- with(
  auto,
  data.frame(
    log(mpg),
    log(cylinders),
    log(displacement),
    log(horsepower),
    log(weight),
    log(acceleration),
    model_year,
    origin
  )
)
# rename without `log.` in beginning
names(cars_log) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                     "acceleration", "model_year", "origin")
```

a. (i) Model 1: Regress log.weight. over log.cylinders. only and report the coefficient (check whether number of cylinders has a significant direct effect on weight)

```r
w_cy_regr <- lm(weight ~ cylinders, data = cars_log)
summary(w_cy_regr)
```

```
##
## Call:
## lm(formula = weight ~ cylinders, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35473 -0.09076 -0.00147  0.09316  0.40374
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.60365    0.03712  177.92   <2e-16 ***
## cylinders    0.82012    0.02213   37.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1329 on 396 degrees of freedom
## Multiple R-squared:  0.7762, Adjusted R-squared:  0.7757
## F-statistic:  1374 on 1 and 396 DF,  p-value: < 2.2e-16
```

Cylinders does have a significant direct effect on weight.

a. (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?)

```
mpg_w_c_regr <- lm(mpg ~ weight + acceleration + model_year + origin, data = cars_log)
summary(mpg_w_c_regr)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + model_year + origin,
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39581 -0.07037  0.00014  0.06984  0.39638
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.539281   0.314707  23.956   <2e-16 ***
## weight       -0.889384   0.028466 -31.243   <2e-16 ***
## acceleration  0.062145   0.036679   1.694   0.0910 .
## model_year    0.032106   0.001690  18.999   <2e-16 ***
## origin        0.018352   0.009165   2.002   0.0459 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1164 on 393 degrees of freedom
## Multiple R-squared:  0.8836, Adjusted R-squared:  0.8825
## F-statistic: 746.1 on 4 and 393 DF,  p-value: < 2.2e-16
```

Weight also does have a significant direct effect on mpg with other variables statistically controlled.

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

```
mpg_w_c_regr$coefficients[2] * w_cy_regr$coefficients[2]
```

```
##     weight
## -0.7294051
```

The indirect effect of cylinders on mpg is `0.7294`.

c. Let's bootstrap for the confidence interval of the indirect effect of cylinders on mpg. Bootstrap (estimating regression models 1 & 2 each time) to get indirect effects. What is its 95% CI of the indirect effect of log.cylinders. on log.mpg.?

```
# define bootstrap func.
boot_mediation <- function(model1, model2, dataset) {
  boot_index <- sample(1:nrow(dataset), replace=TRUE)
  data_boot <- dataset[boot_index, ]
  regr1 <- lm(model1, data_boot)
  regr2 <- lm(model2, data_boot)
```

```r
  return(regr1$coefficients[2] * regr2$coefficients[2])
}
# use the func.
set.seed(0529)
indirect <- replicate(5, boot_mediation(w_cy_regr, mpg_w_c_regr, cars_log))
quantile(indirect, probs=c(0.025,0.975))
```

```
##       2.5%      97.5%
## -0.7930957 -0.7234140
```

The 95% CI of the indirect effect of cylinders on mpg is from `0.7234` to `0.7931`.

Question 2

```r
# remove rows that contain NAs
cars_log <- cars_log[complete.cases(cars_log),]
```

a. (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?)

```r
multicollinearity_cars_log <- cars_log[,c("cylinders","displacement","horsepower","weight")]
mean(multicollinearity_cars_log[,"cylinders"])
```

```
## [1] 1.653046
```

```r
mean(multicollinearity_cars_log[,"displacement"])
```

```
## [1] 5.127891
```

```r
mean(multicollinearity_cars_log[,"horsepower"])
```

```
## [1] 4.587931
```

```r
mean(multicollinearity_cars_log[,"weight"])
```

```
## [1] 7.95918
```

The mean of the four log-transformed variables is `1.65`, `5.13`, `4.59`, `7.96` respectively.

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

```r
mul_carslog_pca <- prcomp(multicollinearity_cars_log, scale. = TRUE)
summary(mul_carslog_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
```

There is about `0.9186` variance is explained by their first principal component.

a. (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? (i.e., think what the variance of the first principal component means or explains)

```
mul_carslog_pca$rotation[,1]
```

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

Since the first principal component contains all the half variables with negative valence, I would call this component "size" of the car, for example, there are in fact A, B, C, D four segments in car industry, describing whether the car is a "Luxury saloon" or a "Subcompact".

b. (i) Store the scores of the first principal component as a new column of cars_log (cars_log$new_column_name <- ...scores of PC1...)

```
cars_log$size <- mul_carslog_pca$x[,1]
```

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

```
mpg_size_c_regr <- lm(mpg ~ size + acceleration + model_year + origin, data = cars_log)
summary(mpg_size_c_regr)
```

```
##
## Call:
## lm(formula = mpg ~ size + acceleration + model_year + origin,
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51070 -0.06039 -0.00161  0.06271  0.46795
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.386083   0.166466   8.327 1.45e-15 ***
## size          0.145547   0.004886  29.786  < 2e-16 ***
## acceleration -0.191608   0.041645  -4.601 5.71e-06 ***
## model_year    0.029210   0.001776  16.444  < 2e-16 ***
## origin        0.009815   0.009680   1.014    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1198 on 387 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.876
## F-statistic: 691.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

b. (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?

```
mpg_size_c_regr_std <- lm(mpg ~ size + acceleration + model_year + origin,
                          data = data.frame(scale(cars_log))
                        )
summary(mpg_size_c_regr_std)
```

```
##
```

```
## Call:
## lm(formula = mpg ~ size + acceleration + model_year + origin,
##     data = data.frame(scale(cars_log)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50188 -0.17759 -0.00472  0.18442  1.37615
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.413e-16  1.779e-02   0.000    1.000
## size          8.205e-01  2.755e-02  29.786  < 2e-16 ***
## acceleration -1.020e-01  2.216e-02  -4.601 5.71e-06 ***
## model_year    3.164e-01  1.924e-02  16.444  < 2e-16 ***
## origin        2.325e-02  2.293e-02   1.014    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3522 on 387 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.876
## F-statistic: 691.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

After independent variables standardized, the size is now the most significant to mpg with the biggest coefficient `0.8205`.

Question 3

```r
# load data
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 4.0.5
```

```r
se_questions <- read_excel("security_questions.xlsx", sheet = "data")
```

a. How much variance did each extracted factor explain?

```r
se_pca <- prcomp(se_questions, scale. = TRUE)
summary(se_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
```

The proportion of the variances to each extracted factor is decreasing from `0.5173`, `0.08869`, `0.06386`, . . . , to `0.0112`.

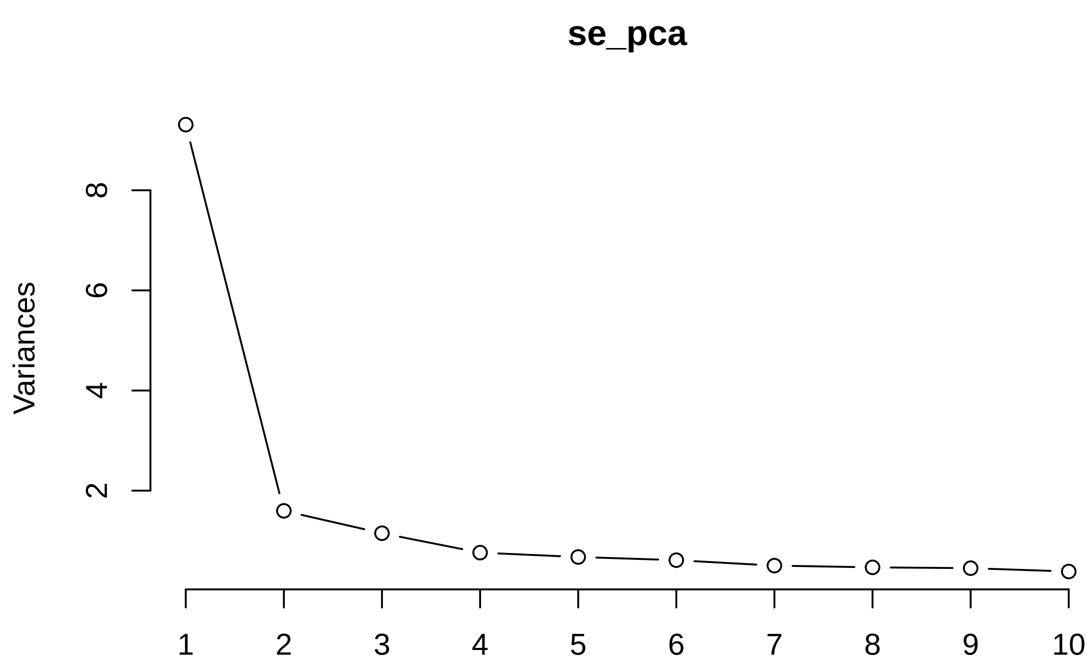
b. How many dimensions would you retain, according to the criteria we discussed? (show a single visualization with scree plot of data, scree plot of noise, eigenvalue = 1 cutoff)

```
# (i) Eigenvalues >= 1
eigen(cor(se_questions))$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
```

If following the eigenvalue = 1 cutoff, we would retain 3 dimensions.

```
# (ii) Scree plot
screeplot(se_pca, type = "lines")
```



If following the screeplot criteria, we would only retain 1 dimension.

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.

```
se_pca$rotation[,1:3]
```

```
##            PC1          PC2          PC3
## Q1  -0.2677422  0.110341691 -0.001973491
## Q2  -0.2204272  0.010886972  0.083171536
## Q3  -0.2508767  0.025878543  0.083648794
## Q4  -0.2042919 -0.508981768  0.100759585
## Q5  -0.2261544  0.024745268 -0.505845415
## Q6  -0.2237681  0.082805088  0.193281966
## Q7  -0.2151891  0.251398450  0.302354487
## Q8  -0.2576225 -0.033526840 -0.320109219
## Q9  -0.2369512  0.183342667  0.189853454
```

```
## Q10 -0.2248660  0.078103267 -0.496820932
## Q11 -0.2467645  0.206580870  0.160903091
## Q12 -0.2065785 -0.504591429  0.113342400
## Q13 -0.2333066  0.051159791  0.078658760
## Q14 -0.2659342  0.078910404  0.146232765
## Q15 -0.2307289 -0.008373326 -0.310161141
## Q16 -0.2482681  0.160524168  0.170839887
## Q17 -0.2023781 -0.525747030  0.102652280
## Q18 -0.2643810  0.089915229 -0.060800871
```

The first component seems to be the average of all questions but in negative valence, representing the abstract of security considerations. The second component weights more in Q4, Q12, and Q17 with negative valence, representing how strong the website providing evidence of transaction correctness. The third component weights more in Q5 and Q10 with negative valence, representing whether the website is from the a real site or a "Phishing" website.