Business Analytics Using Statistical Modeling
Assignment 13

Question 1

Due to strong multicollinearity between cylinders, displacement, horsepower, and weight, we removed them all except for weight in our earlier regression model.

This time, let's try to capture as much variance of all these independent variables as possible.

Let's start by recreating the `cars_log` dataset, which log-transforms all variables except model year and origin.

```
cars <- read.table("../10-auto-data.txt", header = FALSE, na.strings = "?")[-9]
names(cars) <- c("mpg", "cyl", "disp", "hp", "wei", "acc", "year", "ori")
cars <- na.omit(cars)
cars_log <- with(cars, data.frame(log(mpg), log(cyl), log(disp), log(hp), log(wei),
                                  log(acc), year, ori))
```

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

i. Give this smaller data frame an appropriate name (think what they jointly mean)

```
multicol_vars <- cars_log[, c("log.cyl.", "log.disp.", "log.hp.", "log.wei.")]
```

ii. Check the correlation table of these four variables to confirm they are indeed collinear

```
cor(multicol_vars)
```

```
##            log.cyl. log.disp.   log.hp.  log.wei.
## log.cyl.  1.0000000 0.9469109 0.8265831 0.8833950
## log.disp. 0.9469109 1.0000000 0.8721494 0.9428497
## log.hp.   0.8265831 0.8721494 1.0000000 0.8739558
## log.wei.  0.8833950 0.9428497 0.8739558 1.0000000
```

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

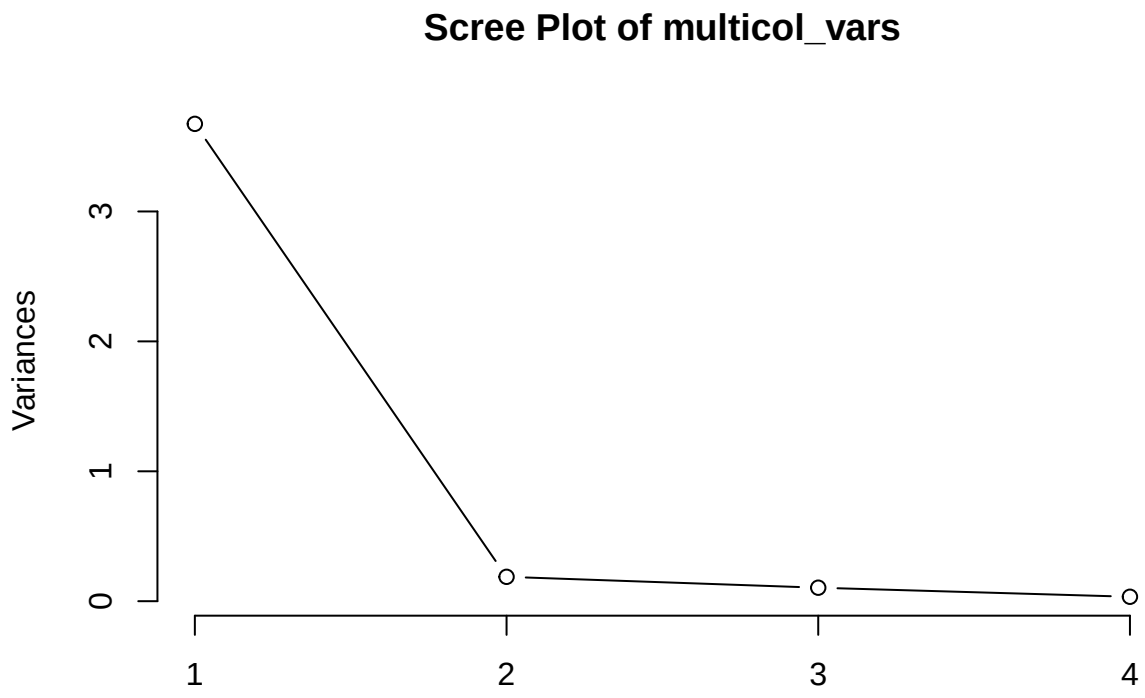
i. How many principal components are needed to summarize these four variables?

```
eigens <- eigen(cor(multicol_vars))
eigens$values
```

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

```
multicol_vars_pca <- prcomp(multicol_vars, scale. = TRUE)
```

```
screeplot(multicol_vars_pca, type = 'l', main = 'Scree Plot of multicol_vars')
```



Based on both the eigenvalues and scree plot criteria, we only need 1 principal components to summarize these 4 variables.

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

```
paste(round(eigens$values[1] / sum(eigens$values) * 100, 2), '%', sep = '')
```

```
## [1] "91.86%"
```

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 first principal component means)

```
eigens$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
```

Light cars.

c. Let's reduce the four collinear variables into one new variable!

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

ii. Name this column appropriately based on the meaning of this first principal component

```
cars_log$PC1_scores <- multicol_vars_pca$x[, "PC1"]
```

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

```
cars_log_std <- as.data.frame(scale(cars_log[, c(1:7, 9)]))
cars_log_std$ori <- cars_log$ori
```

i. Regress mpg over weight, acceleration, model_year and origin

```
regr_mpg_log1 <- lm(log.mpg. ~ log.wei. + log.acc. + year + factor(ori),
                    data = cars_log_std)
summary(regr_mpg_log1)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.wei. + log.acc. + year + factor(ori),
##     data = cars_log_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.12514 -0.20744  0.01179  0.19691  1.17041
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.04755    0.02434  -1.954  0.05145 .
## log.wei.     -0.72400    0.02405 -30.101  < 2e-16 ***
## log.acc.      0.02894    0.01976   1.464  0.14389
## year          0.35519    0.01876  18.937  < 2e-16 ***
## factor(ori)2  0.16501    0.05364   3.076  0.00225 **
## factor(ori)3  0.09392    0.05442   1.726  0.08519 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3421 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
```

ii. Repeat the regression, but replace weight with the factor scores of the 1st principal component of our collinear independent variables

```
regr_mpg_log2 <- lm(log.mpg. ~ log.acc. + year + factor(ori) + PC1_scores,
                    data = cars_log_std)
summary(regr_mpg_log2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.acc. + year + factor(ori) + PC1_scores,
##     data = cars_log_std)
##
## 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
## log.acc.     -0.10190    0.02220  -4.589 6.02e-06 ***
## year          0.31611    0.01961  16.122  < 2e-16 ***
## factor(ori)2  0.02433    0.05775   0.421    0.674
## factor(ori)3  0.05790    0.05704   1.015    0.311
## PC1_scores    0.82112    0.02851  28.804  < 2e-16 ***
## ---
## 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
```

iii. Use VIF scores to check whether the either regression suffers from multicollinearity

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.2.5
```

```
vif(regr_mpg_log1)
```

```
##                 GVIF Df GVIF^(1/(2*Df))
## log.wei.    1.933208  1        1.390398
## log.acc.    1.304761  1        1.142261
## year        1.175545  1        1.084225
## factor(ori) 1.710178  2        1.143564
```

```
vif(regr_mpg_log2)
```

```
##                 GVIF Df GVIF^(1/(2*Df))
## log.acc.    1.549953  1        1.244971
## year        1.208800  1        1.099454
## factor(ori) 1.845979  2        1.165619
## PC1_scores  2.555002  1        1.598437
```

Either regression does not suffer from multicollinearity.

iv. (ungraded)

Question 2

An online marketing firm is studying how customers who shop on e-commerce websites over the winter holiday season perceive the security of e-commerce sites.

Based on feedback from experts, the company has created eighteen questions (see 'questions' tab of excel file) regarding important security considerations at e-commerce websites.

Over 400 customers responded to these questions (see 'data' tab of Excel file).

Respondents were asked to consider a shopping site they were familiar with when answering questions (site was chosen randomly from those each subject has recently visited).

The company 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.

```
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 3.2.5
```

```
sec_qs <- read.xlsx('../13-security_questions.xlsx', sheet = 'data')
```

a. How much variance did each extracted factor explain?

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

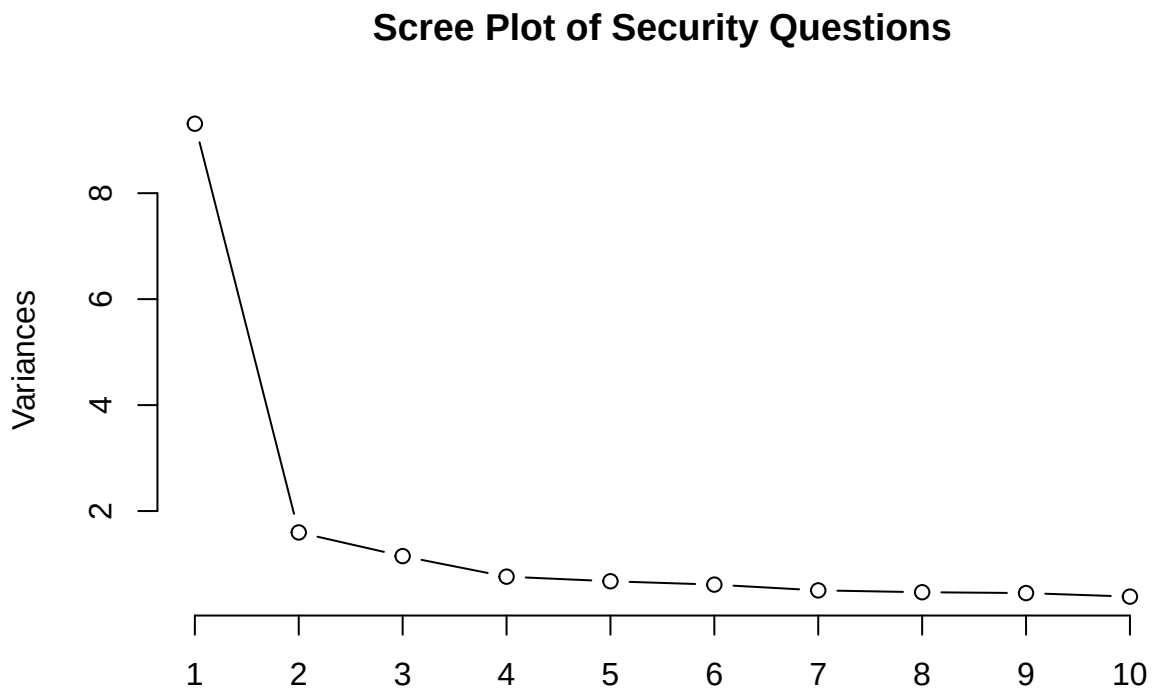
```
## The following object is masked from 'package:car':
##
##     logit
```

```
sec_qs_principal <- principal(sec_qs, nfactor = 18, rotate = 'none', scores = TRUE)
round(sec_qs_principal$Vaccounted[2:3, ], 2)
```

```
##                  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12
## Proportion Var 0.52 0.09 0.06 0.04 0.04 0.03 0.03 0.03 0.03 0.02 0.02 0.02
## Cumulative Var 0.52 0.61 0.67 0.71 0.75 0.78 0.81 0.84 0.86 0.88 0.90 0.92
##                 PC13 PC14 PC15 PC16 PC17 PC18
## Proportion Var 0.02 0.01 0.01 0.01 0.01 0.01
## Cumulative Var 0.94 0.95 0.96 0.98 0.99 1.00
```

b. Show a scree plot of factors extracted

```
sec_qs_pca <- prcomp(sec_qs, scale. = TRUE)
screeplot(sec_qs_pca, type = 'l', main = 'Scree Plot of Security Questions')
```



c. How many factors should we retain in our analysis? (judge using the criteria we've discussed)

```
round(sec_qs_principal$values, 2)
```

```
##  [1] 9.31 1.60 1.15 0.76 0.68 0.61 0.50 0.47 0.45 0.39 0.35 0.30 0.29 0.26
## [15] 0.23 0.23 0.21 0.20
```

Based on both the eigenvalues and scree plot criteria, we should retain 1 factor in our analysis.

d. (ungraded)