HW14

108078517

5/30/2021

Question 1) Cylinders might have an indirect relationship with mpg through its weight?

a. Let's try computing the direct effects first:

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

```
weight_cylinders_regr =
  summary(lm(log.weight.~log.cylinders., data = cars_log))
weight_cylinders_regr$coefficients
```

```
##                 Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)    6.6036502 0.03711549 177.92166  0.000000e+00
## log.cylinders. 0.8201241 0.02212817  37.06244 8.330974e-131
```

ii. Model 2: Regress log.mpg. over log.weight. and all control variables and report the coefficient

```
mpg_weight_regr =  summary(lm(log.mpg.~log.weight. + log.acceleration. + model_year +
factor(origin), data = cars_log))
mpg_weight_regr$coefficients
```

```
##                      Estimate  Std. Error    t value      Pr(>|t|)
## (Intercept)        7.43115547 0.312247834  23.798902  4.173116e-78
## log.weight.       -0.87660818 0.028697020 -30.547011 1.006403e-105
## log.acceleration.  0.05150802 0.036652496   1.405307  1.607219e-01
## model_year         0.03273393 0.001695554  19.305742  7.558672e-59
## factor(origin)2    0.05799137 0.017885258   3.242412  1.286685e-03
## factor(origin)3    0.03233252 0.018278851   1.768849  7.769672e-02
```

b. What is the indirect effect of cylinders on mpg?

```
weight_cylinders_regr$coefficients[2]*mpg_weight_regr$coefficients[2]
```

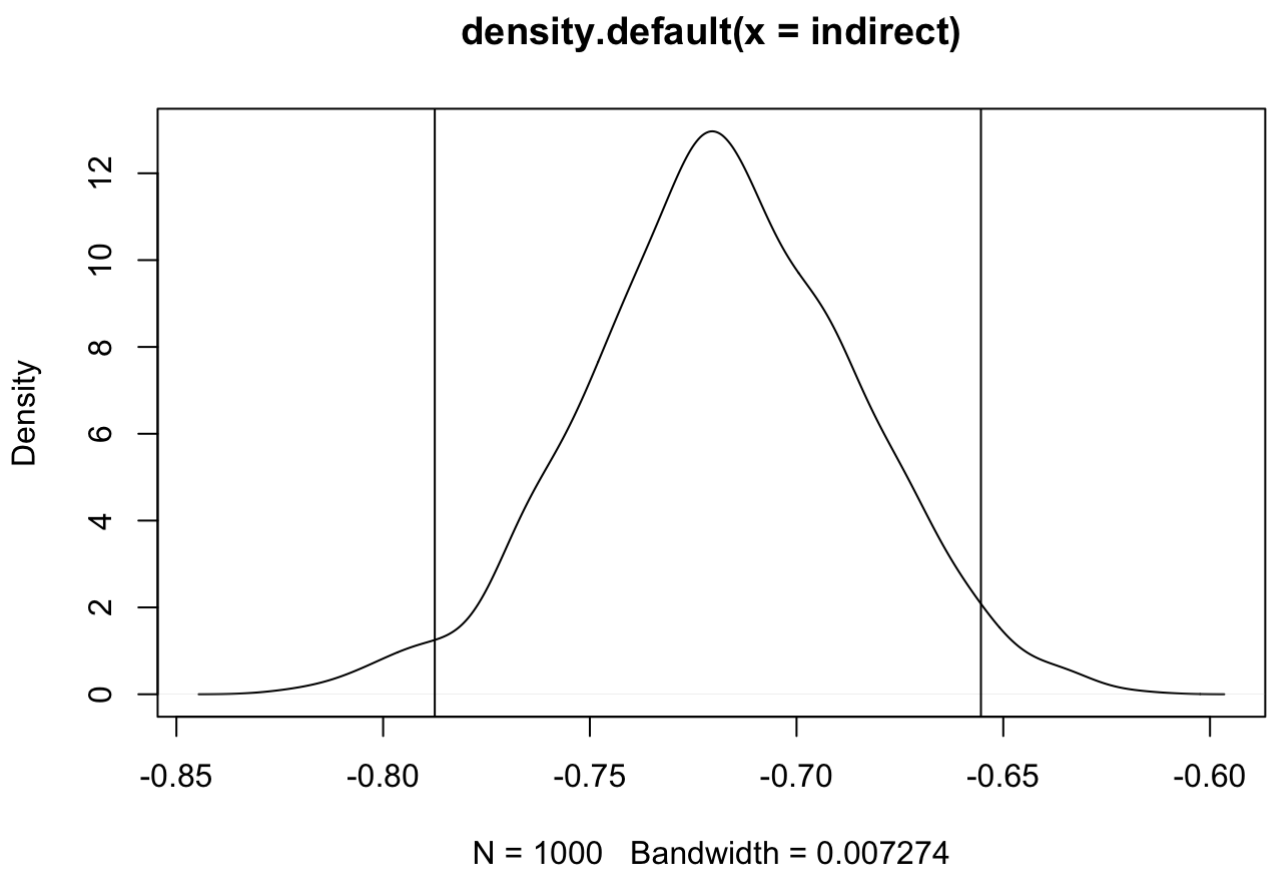
```
## [1] -0.7189275
```

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

i. Bootstrap to get indirect effects: what is its 95% CI of the indirect effect of log.cylinders. on log.mpg.?

```
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)
  return(regr1$coefficients[2] * regr2$coefficients[2])
}
```

```
set.seed(32)
indirect <- replicate(1000,
                      boot_mediation(weight_cylinders_regr, mpg_weight_regr, cars_lo
g))
plot(density(indirect))
abline(v=quantile(indirect, probs=c(0.025, 0.975)))
```




```
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7875109 -0.6553990
```

Q2) Let's revisit the issue of multicollinearity of main effects (between cylinders, displacement, horsepower, and weight) we saw in the cars dataset.

```
cars_log <- cars_log[complete.cases(cars_log), ]
```

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

```
cars_performance = cars_log[,c("log.weight.", "log.displacement.",
                               "log.horsepower.", "log.cylinders.")]
```

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

```
eigen(cor(cars_performance))
```

```
## eigen() decomposition
## $values
## [1] 3.67425879 0.18762771 0.10392787 0.03418563
##
## $vectors
##            [,1]        [,2]        [,3]       [,4]
## [1,] -0.5037960 -0.01530917  0.77500928 -0.3812031
## [2,] -0.5122968  0.25665246  0.07354139  0.8162556
## [3,] -0.4856159 -0.80424467 -0.34193949 -0.0210980
## [4,] -0.4979145  0.53580374 -0.52633608 -0.4335503
```

PC1 captures 3.67425879 times the variance in the cars_log.

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?

When PC1 is increased by one unit, log.weight. will be decreased by 0.5 units. log.displacement. will be reduced by 0.51 units, log.horsepower. will be reduced by 0.48 units, and log.cylinders. will be reduced by 0.49 units.

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 cars_log$new_column_name <- …scores of PC1…

```
cars_performance_pca <- prcomp(cars_performance, scale. = TRUE)
cars_log$PC1 = cars_performance_pca$x[,1]
```

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

```
summary(lm(log.mpg.~PC1 + log.acceleration. + model_year + factor(origin),
           data = cars_log))
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1 + 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 ***
## 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 ***
## 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
```

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?

```
cars_standardized = data.frame(scale(cars_log, center = TRUE, scale = TRUE))
summary(lm(log.mpg.~PC1 + log.acceleration. + model_year + factor(origin),
           data = cars_standardized))
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1 + log.acceleration. + model_year +
##     factor(origin), data = cars_standardized)
##
## 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
## PC1                              0.82112    0.02851  28.804  < 2e-16 ***
## log.acceleration.               -0.10190    0.02220  -4.589 6.02e-06 ***
## model_year                       0.31611    0.01961  16.122  < 2e-16 ***
## factor(origin)0.525710525810929  0.02433    0.05775   0.421    0.674
## factor(origin)1.76714743013553   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
```

Q3) Let's analyze the principal components of the eighteen items.

```
library(readxl)
security = read_excel("security_questions.xlsx", sheet = "data")
```

a. How much variance did each extracted factor explain?

```
eigen_sec = eigen(cor(security) )
eigen_sec$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
```

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)

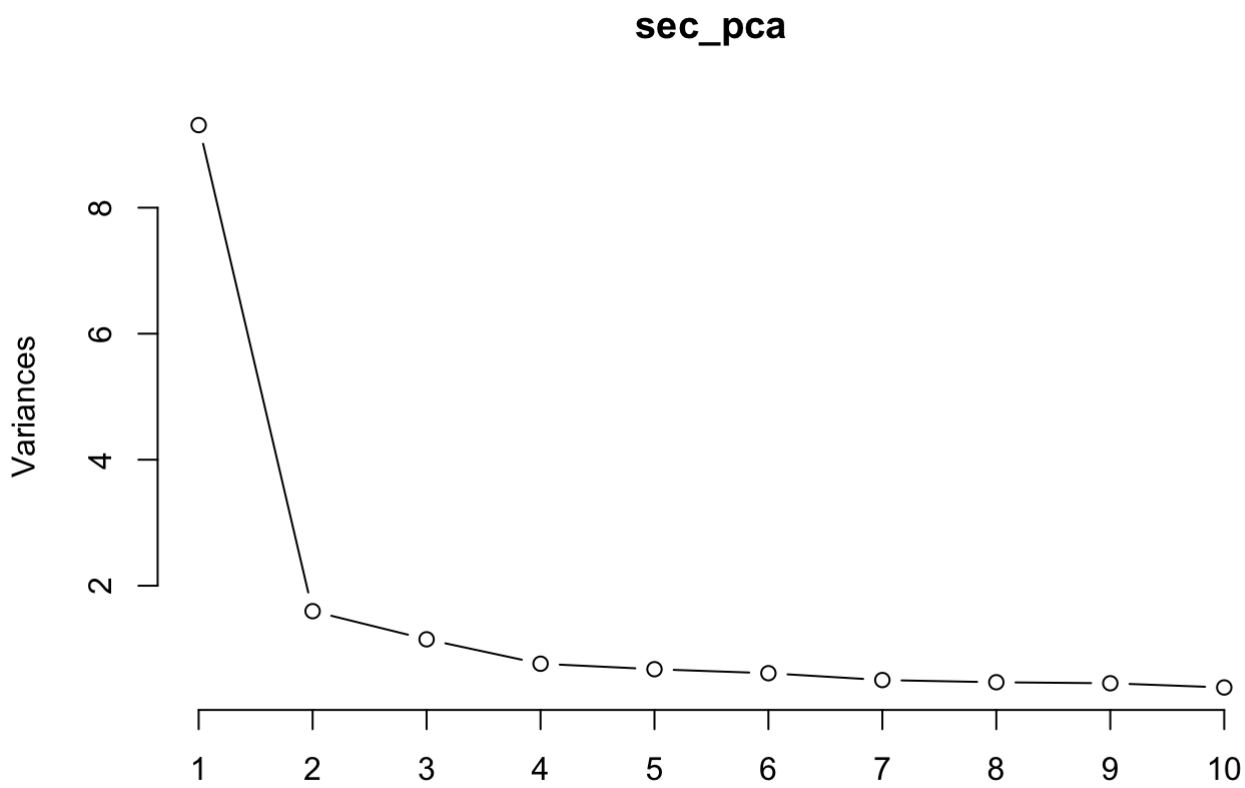
i. Eigenvalues ≥ 1

```
eigen_sec$values[eigen_sec$values>=1] #PC1~3
```

```
## [1] 9.310953 1.596332 1.149558
```

ii. Scree plot

```
sec_pca <- prcomp(security, scale. = TRUE)
screeplot(sec_pca, type="lines")
```



iii. (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

According to graph, we only need to choose first principal components.