BACS HW (Week14)

106070038

2021-05-30

Question 1 In the cars dataset, we saw that the number of cylinders does not seem to directly influence mpg when car weight is also considered. But might cylinders have an indirect relationship with mpg through its weight?

Let's check whether weight mediates the relationship between cylinders and mpg, even when other factors are controlled for. Use log.mpg., log.weight., and log.cylinders as your main variables, and keep log.acceleration., model_year, and origin as control variables (see gray variables in diagram).

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

```
# install.packages("logr")
library(logr)
cars <- read.table("auto-data.txt")
names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year","origin", "car_name")
head(cars, 6)
```

```
##   mpg cylinders displacement horsepower weight acceleration model_year origin
## 1  18         8          307      130.0   3504         12.0         70      1
## 2  15         8          350      165.0   3693         11.5         70      1
## 3  18         8          318      150.0   3436         11.0         70      1
## 4  16         8          304      150.0   3433         12.0         70      1
## 5  17         8          302      140.0   3449         10.5         70      1
## 6  15         8          429      198.0   4341         10.0         70      1
##                    car_name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
cars$horsepower <- as.numeric(cars$horsepower)
```

```
## Warning: NAs introduced by coercion
```

```
cars <- na.omit(cars)
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement),
```

```
                                   log(horsepower), log(weight), log(acceleration), model_year, origin))
head(cars_log, 6)
```

```
##   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
```

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)

```
regr_model1 <- lm(log.weight. ~ log.cylinders.  , data = cars_log)
summary(regr_model1)
```

```
##
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35409 -0.09030 -0.00169  0.09271  0.40488
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.60059    0.03710  177.92   <2e-16 ***
## log.cylinders.  0.82187    0.02208   37.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1319 on 390 degrees of freedom
## Multiple R-squared:  0.7804, Adjusted R-squared:  0.7798
## F-statistic:  1386 on 1 and 390 DF,  p-value: < 2.2e-16
```

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?)

```
regr_model2 <- lm(log.mpg. ~ log.weight.  , data = cars_log)
summary(regr_model2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52321 -0.10446 -0.00772  0.10124  0.59445
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5152     0.2365   48.69   <2e-16 ***
## log.weight.  -1.0575     0.0297  -35.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1651 on 390 degrees of freedom
## Multiple R-squared:  0.7648, Adjusted R-squared:  0.7642
## F-statistic:  1268 on 1 and 390 DF,  p-value: < 2.2e-16
```

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

```
regr_both <- lm(log.mpg. ~ log.weight. + log.cylinders.  , data = cars_log)
summary(regr_both)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59128 -0.10270 -0.00582  0.09948  0.61682
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.03408    0.41151  24.383  < 2e-16 ***
## log.weight.    -0.81932    0.06196 -13.222  < 2e-16 ***
## log.cylinders. -0.25085    0.05765  -4.351 1.73e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1615 on 389 degrees of freedom
## Multiple R-squared:  0.7757, Adjusted R-squared:  0.7745
## F-statistic: 672.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

- Answer: Because log.weight. has bigger estimate, it is more significance affecting log.mpg. than log.cylinders., so cylinders is the indirect factor.

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

i. 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.?

```
set.seed(1)
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])
}
indirect <- replicate(2000, boot_mediation(regr_model1, regr_model2, cars_log))
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.9333552 -0.8084441
```

Question 2 Let's revisit the issue of multicollinearity of main effects (between cylinders, displacement, horsepower, and weight) we saw in the cars dataset. Start by recreating the cars_log dataset, which log-transforms all variables except model year and origin.

Important: remove any rows that have missing values. ## 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 (Give this smaller data frame an appropriate name – what might they jointly mean?)

```
new_cars_log <- cbind(cars_log['log.cylinders.'], cars_log['log.displacement.'])
new_cars_log <- cbind(new_cars_log, cars_log['log.horsepower.'])
new_cars_log <- cbind(new_cars_log, cars_log['log.weight.'])
# new_cars_log
```

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)

```
round( cor(new_cars_log), 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
```

```
cars_log_pca <- prcomp(new_cars_log, scale. = FALSE)
cars_log_pca
```

```
## Standard deviations (1, .., p=4):
## [1] 0.73122637 0.15173927 0.09535464 0.07272012
##
## Rotation (n x k) = (4 x 4):
##                          PC1         PC2        PC3         PC4
## log.cylinders.    -0.3944484  0.32615343 -0.6895416  0.51241263
## log.displacement. -0.7221160  0.36134848  0.1626248 -0.56703525
## log.horsepower.   -0.4322835 -0.87289692 -0.2158783 -0.06766477
## log.weight.       -0.3689037 -0.03319916  0.6719242  0.64134686
```

```
# biplot(cars_log_pca)
scores = cars_log_pca$x
summary(cars_log_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
```

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?

- Answer: The standard deviation of PC1 is 1.9072, and the values od PC1: log.mpg.= 0.4924630 (Positive), log.displacement.= -0.5054964 (Negative), log.horsepower.= -0.4941301 (Negative), log.weight.= -0.5077293 (Negative)
- It means that log.mpg. has positive effect toward PC1 and others have negative effects.

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...

```
PC1 <- cars_log_pca$x
PC1 <- PC1[,1]
cars_log$PC1 <- PC1
# cars_log
```

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

```
regr_pc1 <- lm(log.mpg. ~ PC1  , data = cars_log)
summary(regr_pc1)
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61931 -0.09520 -0.00968  0.11160  0.67446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.098313   0.008071  383.90   <2e-16 ***
## PC1         0.410632   0.011051   37.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1598 on 390 degrees of freedom
## Multiple R-squared:  0.7797, Adjusted R-squared:  0.7792
## F-statistic:  1381 on 1 and 390 DF,  p-value: < 2.2e-16
```

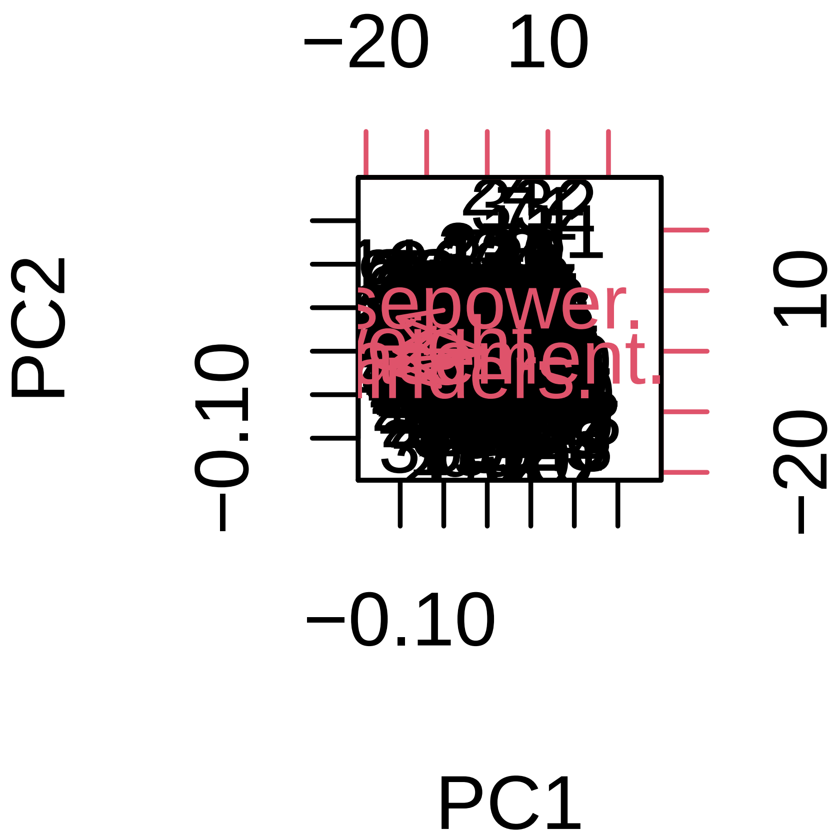
iii. Try running the regression again over the same independent variables, but this time with everything standardized. How important is this new column relative to other columns?

```
cars_log_pca_stand <- prcomp(new_cars_log, scale. = TRUE)
cars_log_pca_stand
```

```
## Standard deviations (1, .., p=4):
## [1] 1.9168356 0.4331601 0.3223785 0.1848936
##
## Rotation (n x k) = (4 x 4):
```

```
##                          PC1         PC2         PC3        PC4
## log.cylinders.    -0.4979145 -0.53580374  0.52633608  0.4335503
## log.displacement. -0.5122968 -0.25665246 -0.07354139 -0.8162556
## log.horsepower.   -0.4856159  0.80424467  0.34193949  0.0210980
## log.weight.       -0.5037960  0.01530917 -0.77500928  0.3812031
```

```
biplot(cars_log_pca_stand)
```




```
scores = cars_log_pca_stand$x
summary(cars_log_pca_stand)
```

```
## 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
```

```
PC1_stand <- cars_log_pca_stand$x
PC1_stand <- PC1_stand[,1]
cars_log$PC1_stand <- PC1_stand
# cars_log
regr_pc1_stand_onlypc1 <- lm(log.mpg. ~ PC1_stand , data = cars_log)
summary(regr_pc1_stand_onlypc1)
```

```
##
## Call:
## lm(formula = log.mpg. ~ PC1_stand, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58902 -0.09469 -0.01141  0.10420  0.62440
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.098313   0.007892  392.60   <2e-16 ***
## PC1_stand   0.157612   0.004122   38.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1563 on 390 degrees of freedom
## Multiple R-squared:  0.7894, Adjusted R-squared:  0.7889
## F-statistic:  1462 on 1 and 390 DF,  p-value: < 2.2e-16
```

- The estimator before standardized: 0.167457
- The estimator after standardized: 0.157612

Question 3 Please download the Excel data file security_questions.xlsx from Canvas. In your analysis, you can either try to read the data sheet from the Excel file directly from R (there might be a package for that!) or you can try to export the data sheet to a CSV file before reading it into R.

```r
# install.packages('readxl')
library('readxl')
security_questions <- read_excel("security_questions.xlsx", sheet = "data")
```

a. How much variance did each extracted factor explain?

```r
eigen(cor(security_questions))
```

```
## eigen() decomposition
## $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
##
## $vectors
##             [,1]         [,2]         [,3]         [,4]         [,5]
##  [1,] -0.2677422  0.110341691 -0.001973491  0.126220668 -0.048468417
```

```
##  [2,] -0.2204272  0.010886972  0.083171536  0.258122218  0.093887919
##  [3,] -0.2508767  0.025878543  0.083648794 -0.399268076 -0.061766335
##  [4,] -0.2042919 -0.508981768  0.100759585  0.040690031 -0.072913141
##  [5,] -0.2261544  0.024745268 -0.505845415  0.052574743 -0.193207848
##  [6,] -0.2237681  0.082805088  0.193281966 -0.004209098  0.611348765
##  [7,] -0.2151891  0.251398450  0.302354487  0.327318232  0.008596733
##  [8,] -0.2576225 -0.033526840 -0.320109219  0.076017162  0.209097752
##  [9,] -0.2369512  0.183342667  0.189853454 -0.124795087  0.025138160
## [10,] -0.2248660  0.078103267 -0.496820932 -0.034236123 -0.249119125
## [11,] -0.2467645  0.206580870  0.160903091  0.264607608 -0.210724202
## [12,] -0.2065785 -0.504591429  0.113342400  0.060346524  0.052819352
## [13,] -0.2333066  0.051159791  0.078658760 -0.602543012 -0.030357718
## [14,] -0.2659342  0.078910404  0.146232765 -0.362581586 -0.086718158
## [15,] -0.2307289 -0.008373326 -0.310161141  0.069411508  0.513508897
## [16,] -0.2482681  0.160524168  0.170839887  0.204337585 -0.342722070
## [17,] -0.2023781 -0.525747030  0.102652280  0.080754652 -0.157376900
## [18,] -0.2643810  0.089915229 -0.060800871  0.051492827 -0.024214541
##                [,6]        [,7]         [,8]         [,9]       [,10]
##  [1,]  0.1826730451  0.47564502  0.011877666 -0.158945743 -0.02559547
##  [2,]  0.7972988590 -0.10381142  0.370484027  0.018906337  0.01758985
##  [3,]  0.1343170710 -0.29794768 -0.045361944  0.046160967 -0.62920376
##  [4,] -0.0683434170 -0.07323286 -0.082718228  0.034011814 -0.13146697
##  [5,]  0.1493338250 -0.19273010 -0.188948821  0.218690034  0.09878156
##  [6,]  0.0551361412  0.06503361 -0.538423059  0.331476460 -0.04348905
##  [7,] -0.0562329401 -0.45399251 -0.229822767 -0.236185029  0.31439194
##  [8,] -0.2005009349  0.06635056  0.204619876 -0.232217507  0.08234563
##  [9,] -0.2696485391 -0.12766155  0.452229009  0.595761520  0.25923949
## [10,]  0.0232597277 -0.15613131 -0.250158309  0.141066357  0.09604999
## [11,] -0.1928970917  0.01757216 -0.170741343 -0.289466716 -0.12972901
## [12,] -0.0454546580  0.03110171  0.005586284  0.007633808  0.16822370
## [13,]  0.0949114194  0.03589479 -0.013028375 -0.281562536  0.49131061
## [14,] -0.0006735609  0.07224998  0.032286752 -0.224017714 -0.12173004
## [15,] -0.2572918341 -0.15806779  0.305772284 -0.250812042 -0.19230189
## [16,] -0.2189544787  0.03885431  0.186064954  0.134618480 -0.21266262
## [17,] -0.0527365890 -0.02827931 -0.038609734  0.023978170  0.09198523
## [18,] -0.0327588454  0.58413134 -0.079484842  0.184214340 -0.01232082
##              [,11]         [,12]       [,13]       [,14]        [,15]
##  [1,]  0.261433547  0.3655136121 -0.09437152  0.21538278  0.107191422
##  [2,] -0.141511628 -0.1423173350 -0.01439656 -0.14151031 -0.124321587
##  [3,]  0.215411545  0.0711375730  0.07897104  0.38275058 -0.173199162
##  [4,]  0.182772484  0.0001075882  0.32083974 -0.53718169 -0.009053271
##  [5,] -0.090154465  0.0962621836  0.41176540  0.13779948  0.420108616
##  [6,] -0.230188841  0.1679270706 -0.06866003 -0.12229591 -0.076584623
##  [7,]  0.441121206  0.0404427953 -0.01046519  0.03486607  0.164646045
```

```
##  [8,]  0.218910615  0.3074295739  0.08286262 -0.07220809 -0.517381497
##  [9,]  0.125837984 -0.1387657899  0.06167134  0.06636535 -0.103891810
## [10,]  0.006787801 -0.1568738426 -0.54451920 -0.17543121 -0.275471410
## [11,] -0.395639123 -0.4128696157  0.22239835  0.14404891 -0.308218564
## [12,] -0.072388580 -0.1181594259 -0.39416050  0.46427132  0.147423769
## [13,] -0.306206763  0.1388173302  0.19909498  0.01118762 -0.042881369
## [14,]  0.134853427 -0.2306763906 -0.29401321 -0.38305994  0.322075542
## [15,] -0.178156051 -0.1589461038 -0.01621655  0.01470750  0.336177176
## [16,] -0.383866578  0.4817217034 -0.17169894 -0.17403268  0.168614520
## [17,] -0.083760590  0.0503178068  0.03431935  0.09260499 -0.096523523
## [18,]  0.229097907 -0.3832085961  0.19580495  0.02702597  0.077981920
##             [,16]       [,17]       [,18]
##  [1,] -0.26663363  0.15892454  0.49709414
##  [2,]  0.04539846 -0.01378516 -0.07954338
##  [3,]  0.10905667  0.08731092 -0.07451547
##  [4,] -0.26266355  0.39030988  0.02091260
##  [5,] -0.20508811 -0.26389562 -0.07356419
##  [6,] -0.04426883 -0.11718533  0.02443898
##  [7,]  0.19302912  0.07574440 -0.08656284
##  [8,] -0.08324463 -0.31696165 -0.32212598
##  [9,] -0.19386537 -0.01929777  0.22424357
## [10,]  0.07402245  0.24996841  0.14445897
## [11,] -0.28230295 -0.05599291  0.11746105
## [12,] -0.29758805  0.08367724 -0.38027121
## [13,]  0.11740772  0.26739129 -0.04166051
## [14,] -0.16553236 -0.50553644 -0.01188146
## [15,]  0.18191811  0.22010115  0.21302663
## [16,]  0.17538230  0.09232084 -0.26436304
## [17,]  0.51310849 -0.39101042  0.42651093
## [18,]  0.42203495  0.12287014 -0.30773331
```

- Answer:
  - **Vector** is the direction of variance.
  - **Value** is the variance captured by PC.

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

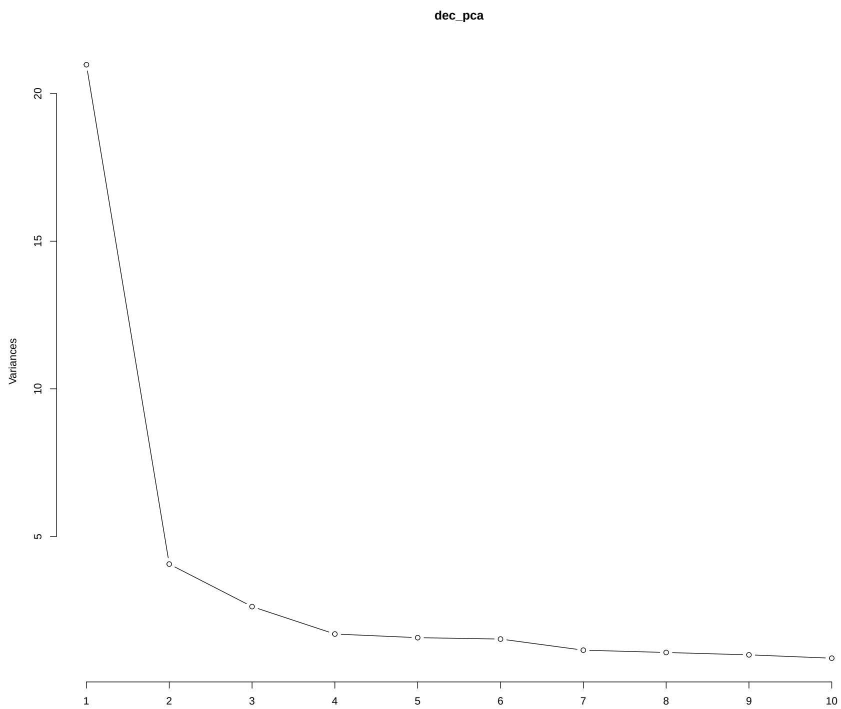
```
eigen(cor(security_questions))$values
```

```
##  [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
```

ii. Scree plot

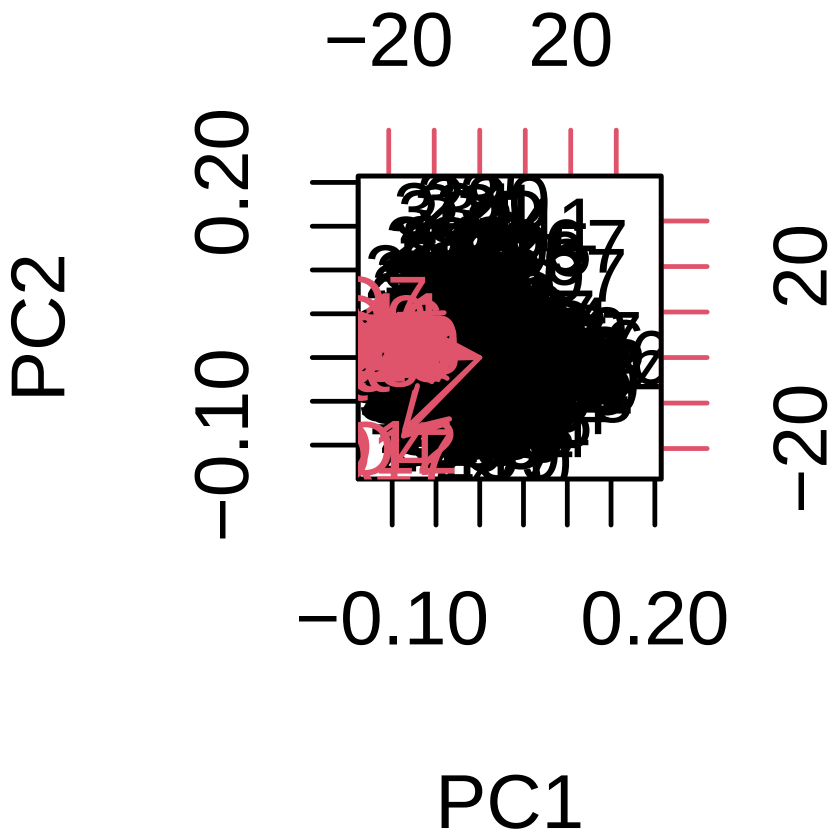
```
dec_pca <- prcomp(security_questions, scale. = FALSE)
screeplot(dec_pca, type="lines")
```



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 - Answer: According to the matrix below, Q8 and Q16 dominate the biggest part of PC1, because they have biggest values in PC1.

```
biplot(dec_pca)
```




```
dec_pca
```

```
## Standard deviations (1, .., p=18):
##  [1] 4.5803084 2.0157428 1.6193501 1.3012428 1.2529464 1.2341295 1.0706802
##  [8] 1.0334903 0.9939914 0.9352967 0.8879475 0.8177929 0.8165971 0.7655616
## [15] 0.7439983 0.7283268 0.6565257 0.6408405
##
## Rotation (n x k) = (18 x 18):
##            PC1         PC2         PC3          PC4          PC5          PC6
## Q1  -0.2491083  0.10493359 -0.01019162  0.113078870 -0.035391881  0.103159433
## Q2  -0.2463737  0.03148303  0.11456415  0.648476743 -0.571725684  0.246277381
## Q3  -0.2431477  0.03436924  0.04365275 -0.314076786 -0.261958856  0.168882550
## Q4  -0.2221963 -0.49469712  0.10946792 -0.018335298  0.100069964  0.008570492
## Q5  -0.2106025  0.03181945 -0.44037452  0.116979931  0.073573622  0.247536637
## Q6  -0.2155338  0.08661273  0.14733810 -0.026219947 -0.292968008 -0.450990599
## Q7  -0.2427124  0.29259568  0.38364287  0.263800203  0.265621587 -0.181647774
## Q8  -0.2629719 -0.01570353 -0.34767565  0.040328044  0.092332220 -0.262238070
## Q9  -0.2446115  0.19064006  0.17013737 -0.300517171  0.006980713 -0.093223592
## Q10 -0.2199303  0.08075312 -0.47509118  0.005463849  0.146246096  0.284192183
## Q11 -0.2463162  0.20422392  0.16581527  0.080363503  0.339754822  0.051893655
## Q12 -0.2239165 -0.48758805  0.11494519  0.021145489  0.017479776 -0.104632794
## Q13 -0.2167467  0.04986984  0.02002763 -0.446079939 -0.321683837  0.149204902
```

```
## Q14 -0.2302322  0.07039120  0.08257362 -0.284289818 -0.135308697  0.103261518
## Q15 -0.2408316  0.01016821 -0.36477469  0.058951560 -0.037660861 -0.586298049
## Q16 -0.2569632  0.16718491  0.18266079 -0.011994703  0.368720117  0.206049060
## Q17 -0.2276085 -0.53251587  0.12484564  0.021923925  0.160002586  0.099594636
## Q18 -0.2346175  0.07959100 -0.06772757 -0.031117618  0.031291545  0.039199823
##              PC7          PC8         PC9         PC10        PC11         PC12
## Q1  -0.161777961  0.406400022 -0.10173207  0.228171938 -0.32965112  0.186785582
## Q2  -0.173614372 -0.187032826 -0.08474079 -0.029696672  0.08365364 -0.071115345
## Q3   0.238087069 -0.164773872 -0.06230109 -0.512181918 -0.35535402  0.301066787
## Q4   0.100048065 -0.047293432  0.05268594 -0.149947484 -0.38129581 -0.474326908
## Q5   0.176064563 -0.095752546  0.27843152 -0.027127614  0.08224388 -0.253565929
## Q6   0.135676273  0.379985945  0.56706246 -0.200968929  0.16125303 -0.098429974
## Q7   0.577128036 -0.229299464 -0.04354000  0.280019791 -0.12758543  0.064994344
## Q8  -0.082823777 -0.006577311 -0.24229404  0.239339673 -0.26058284 -0.109268250
## Q9  -0.424096814 -0.547415938  0.34979479  0.251211990 -0.02489844 -0.055850943
## Q10  0.242186655 -0.058530327  0.27474852  0.036482062  0.15906944  0.266273842
## Q11  0.006436242  0.263324013 -0.21790280 -0.260845532  0.31143761 -0.160139610
## Q12 -0.035927599  0.025004719  0.03142390  0.181284149  0.27258469  0.576213734
## Q13  0.172323357  0.080846912 -0.29594288  0.323999488  0.40051808 -0.317350881
## Q14  0.057159477  0.077140772 -0.22250445  0.008004421 -0.08646575  0.120367696
## Q15 -0.105246819 -0.188101662 -0.32347965 -0.261151027  0.14189708  0.036687125
## Q16 -0.384781737 -0.017379091 -0.02530682 -0.327383329  0.14008190  0.022363843
## Q17  0.018395680 -0.029062397  0.02191276  0.044040243  0.15960247 -0.074451274
## Q18 -0.235304099  0.377121572  0.16420696  0.194217416 -0.25569636 -0.001216882
##             PC13         PC14        PC15         PC16        PC17         PC18
## Q1   0.159122298 -0.064700974 -0.13888473 -0.165583327  0.65081062 -0.071258661
## Q2  -0.054135974  0.073327632  0.08150110  0.098305676 -0.08486285 -0.009042054
## Q3  -0.009384882 -0.309710953 -0.12235301  0.124714832 -0.06478903 -0.210495813
## Q4  -0.124219004  0.471705447 -0.07761508 -0.066836405  0.09051096 -0.137833257
## Q5   0.013943405 -0.262660625 -0.46394120 -0.301712708 -0.04560851  0.336613178
## Q6   0.205671411  0.036460041  0.03303509  0.140830311  0.03745562  0.089612726
## Q7   0.081272419 -0.020641289  0.03729512 -0.165802640 -0.10016486 -0.052397069
## Q8   0.264470754 -0.060126938 -0.10572974  0.617241240 -0.22787449  0.102040281
## Q9  -0.217460812 -0.055439372 -0.03687258  0.091822697  0.21641200  0.010564421
## Q10 -0.050621341  0.361586581  0.38223023  0.189853026  0.14303642 -0.214945076
## Q11 -0.529433591 -0.097615680 -0.14369308  0.327595975  0.14212726 -0.002200337
## Q12 -0.105325138  0.148931969 -0.43287765 -0.008914802 -0.15612919 -0.001063554
## Q13  0.157756152  0.039106443 -0.08477193 -0.085033666 -0.03574401 -0.305575130
## Q14 -0.113298547  0.268281636  0.24976281 -0.093514879 -0.05472596  0.765336392
## Q15 -0.092597156  0.002895087  0.15271206 -0.405042673  0.11629804 -0.099649142
## Q16  0.562575264  0.193989437 -0.02541730 -0.135389875 -0.20583370 -0.048273865
## Q17  0.146670430 -0.546823863  0.48041689  0.023474494  0.13151189  0.096125742
## Q18 -0.334125489 -0.148188347  0.21058572 -0.269750427 -0.54961840 -0.223834787
```