HW14

108048110

2022-05-18

BACS HW - Week 14

Prerequisite

```
library(ggpubr)
library(ggplot2)
library(plot3D)
library(rgl)
library(factoextra)
library(FactoMineR)
library(magrittr)
library(psych)
```

```
path = 'data/security_questions.xlsx'
questions <- readxl::read_excel(path, sheet=1, col_names = c("Index", "Questions"))
data <- readxl::read_excel(path, sheet=2)
```

Question 1)

Parallel analysis

a. Show a single visualization with scree plot of data, scree plot of simulated noise, and a horizontal line showing the $eigenvalue = 1$ cutoff.

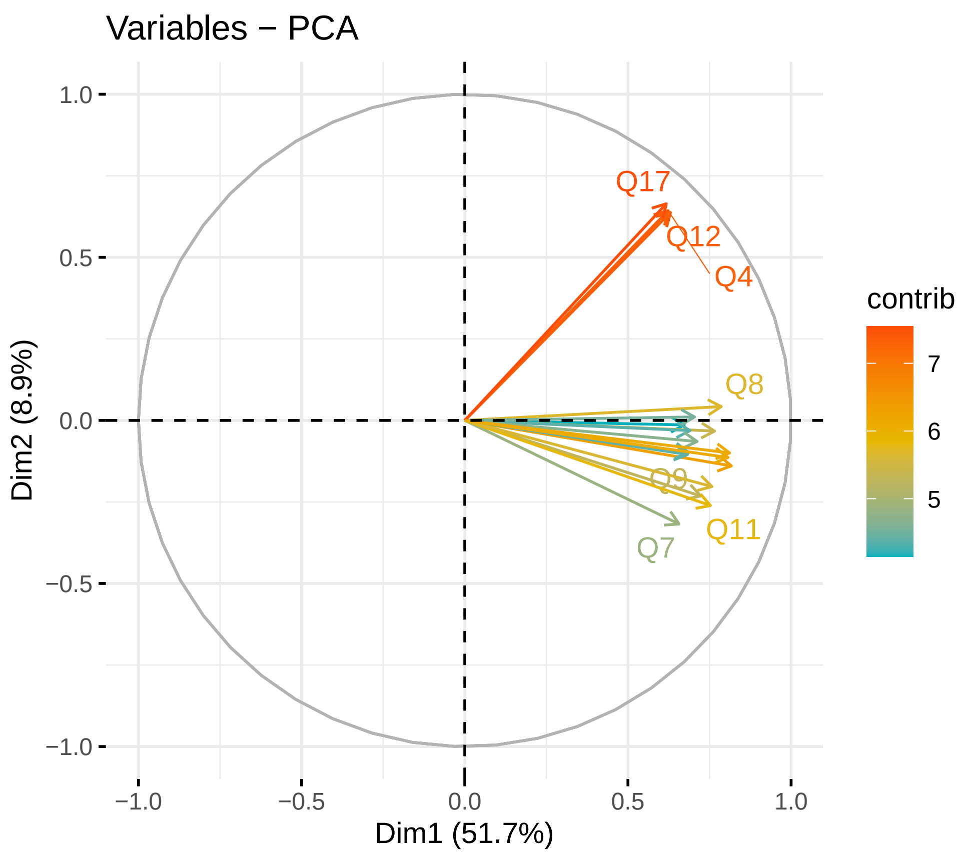
```
data_pca <- prcomp(data, scale.=TRUE)
data_pca |> sapply(head)
```

```
## $sdev
## [1] 3.0513855 1.2634603 1.0721745 0.8729123 0.8216697 0.7820893
##
## $rotation
```

```
##          PC1         PC2          PC3          PC4         PC5         PC6
## Q1 -0.2677422  0.11034169 -0.001973491  0.126220668 -0.04846842  0.18267305
## Q2 -0.2204272  0.01088697  0.083171536  0.258122218  0.09388792  0.79729886
## Q3 -0.2508767  0.02587854  0.083648794 -0.399268076 -0.06176634  0.13431707
## Q4 -0.2042919 -0.50898177  0.100759585  0.040690031 -0.07291314 -0.06834342
## Q5 -0.2261544  0.02474527 -0.505845415  0.052574743 -0.19320785  0.14933383
## Q6 -0.2237681  0.08280509  0.193281966 -0.004209098  0.61134877  0.05513614
##            PC7         PC8         PC9        PC10        PC11          PC12
## Q1 -0.47564502  0.01187767 -0.15894574  0.02559547 -0.26143355  0.3655136121
## Q2  0.10381142  0.37048403  0.01890634 -0.01758985  0.14151163 -0.1423173350
## Q3  0.29794768 -0.04536194  0.04616097  0.62920376 -0.21541155  0.0711375730
## Q4  0.07323286 -0.08271823  0.03401181  0.13146697 -0.18277248  0.0001075882
## Q5  0.19273010 -0.18894882  0.21869003 -0.09878156  0.09015446  0.0962621836
## Q6 -0.06503361 -0.53842306  0.33147646  0.04348905  0.23018884  0.1679270706
##           PC13       PC14         PC15        PC16        PC17        PC18
## Q1 -0.09437152  0.2153828  0.107191422 -0.26663363 -0.15892454  0.49709414
## Q2 -0.01439656 -0.1415103 -0.124321587  0.04539846  0.01378516 -0.07954338
## Q3  0.07897104  0.3827506 -0.173199162  0.10905667 -0.08731092 -0.07451547
## Q4  0.32083974 -0.5371817 -0.009053271 -0.26266355 -0.39030988  0.02091260
## Q5  0.41176540  0.1377995  0.420108616 -0.20508811  0.26389562 -0.07356419
## Q6 -0.06866003 -0.1222959 -0.076584623 -0.04426883  0.11718533  0.02443898
##
## $center
##       Q1       Q2       Q3       Q4       Q5       Q6
## 5.604938 5.049383 5.306173 4.992593 5.760494 4.762963
##
## $scale
##       Q1       Q2       Q3       Q4       Q5       Q6
## 1.409647 1.653838 1.465810 1.592819 1.412259 1.451565
##
## $x
##             PC1        PC2         PC3        PC4         PC5        PC6
## [1,] -1.7754125 -2.0169459 -1.63959104  0.5352081 -0.08797596 -0.3728346
## [2,] -1.3437632 -0.8882234 -0.25368424 -0.4570310  0.23962874 -0.4268643
## [3,] -1.7933115 -0.4260365  0.07691544  0.4141432 -0.60889140  0.5624084
## [4,]  0.9013133 -0.2983700  0.36032074  0.7691251  0.40805995 -0.1145464
## [5,] -2.9135336 -1.3469746 -0.14286375 -0.1077017 -1.47129093  0.8579121
## [6,]  1.2061151 -0.3409259  1.94682489 -0.1299549  0.16378728 -0.1768219
##              PC7         PC8        PC9        PC10        PC11        PC12
## [1,] -0.79963845  0.31011634 -0.1350201 -0.18107617 -0.37536871 -0.07135772
## [2,]  0.25929669 -0.10719126 -0.6186849 -0.03219035  0.13518587 -0.28714217
## [3,]  0.29407731 -1.08068173  0.3747303 -0.02571319 -0.06443770  0.20395036
## [4,] -0.07167346 -0.08161187  0.4527110  0.15952336 -0.07864503 -0.12495990
## [5,] -0.33413881  0.28492699  0.2001644 -0.09883888 -0.56009509  0.05191722
## [6,] -1.44430904  1.24026221 -0.2124737 -0.28195730 -0.24564942 -0.04531787
##              PC13        PC14       PC15        PC16         PC17        PC18
## [1,]  0.001813884  0.25632295  0.2039214  0.05929609 -0.450507689  0.25009281
## [2,]  0.167932157 -0.05901637 -0.4252439  0.07265085  0.255354655 -0.48814544
## [3,] -0.320481103 -0.16741905 -0.2202692 -0.24785872  0.409204449 -0.37686973
## [4,]  0.133981171  0.24263900 -0.2105912  0.39901391 -0.343677682 -0.23329700
## [5,]  0.111733382  0.34393812 -0.6789966 -0.21330516  0.009084729 -0.39442816
## [6,]  0.618654281 -0.08565330  0.3726984 -0.36311736  0.616667407 -0.07356653
```

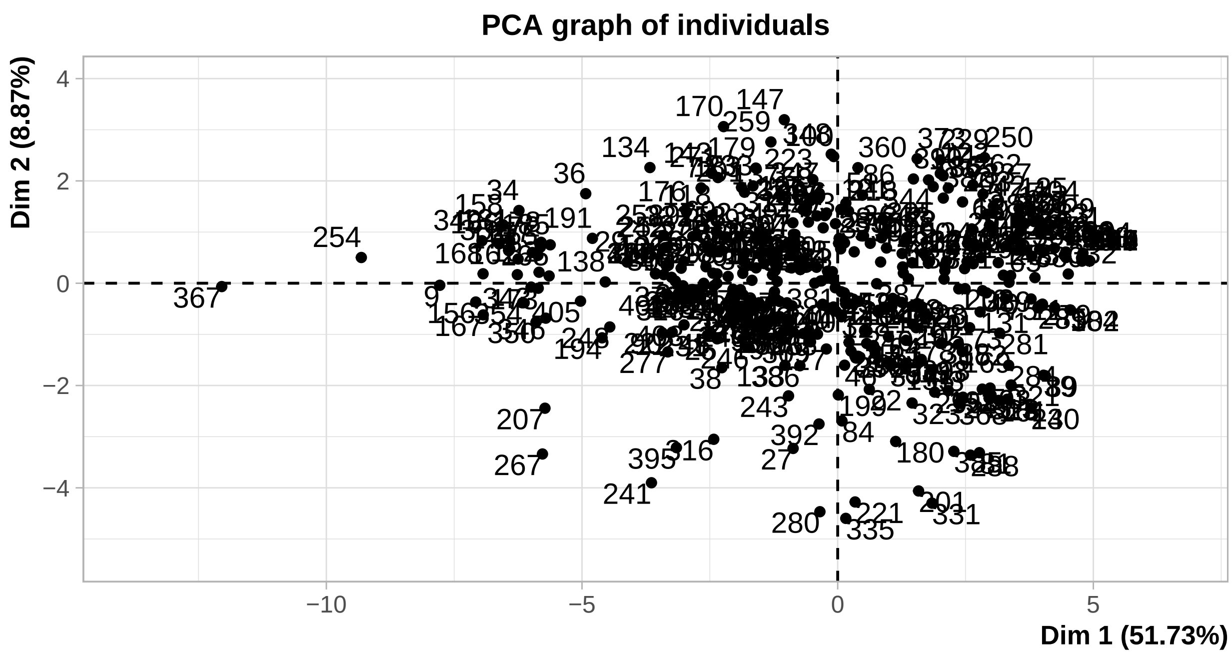
```r
data_pca$rotation = -data_pca$rotation
```

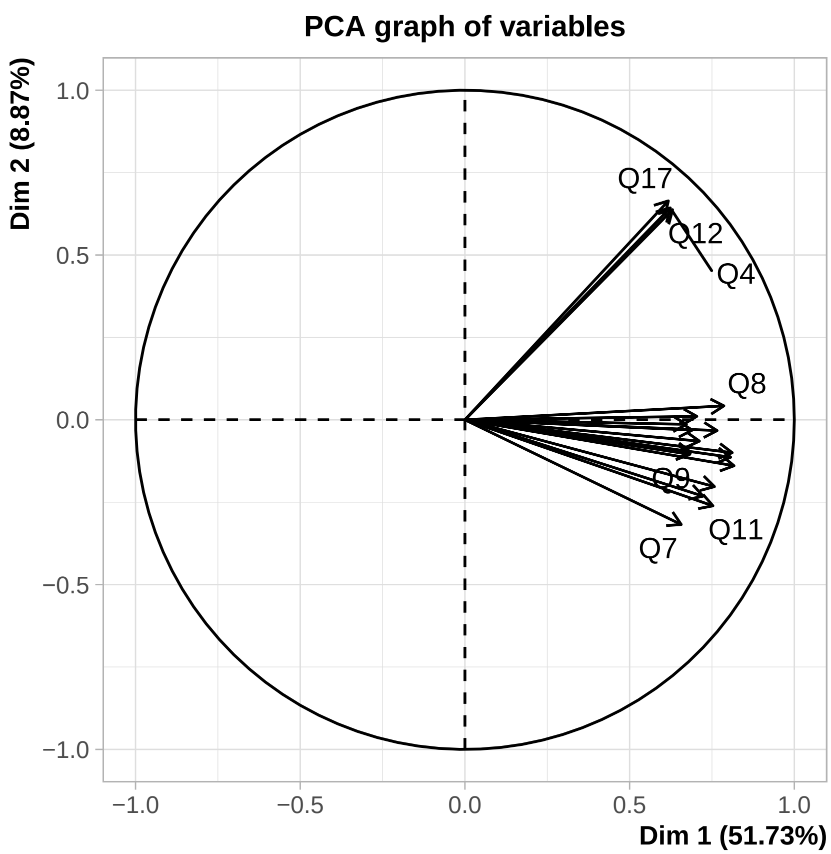
```r
fviz_pca_var(data_pca,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
)
```



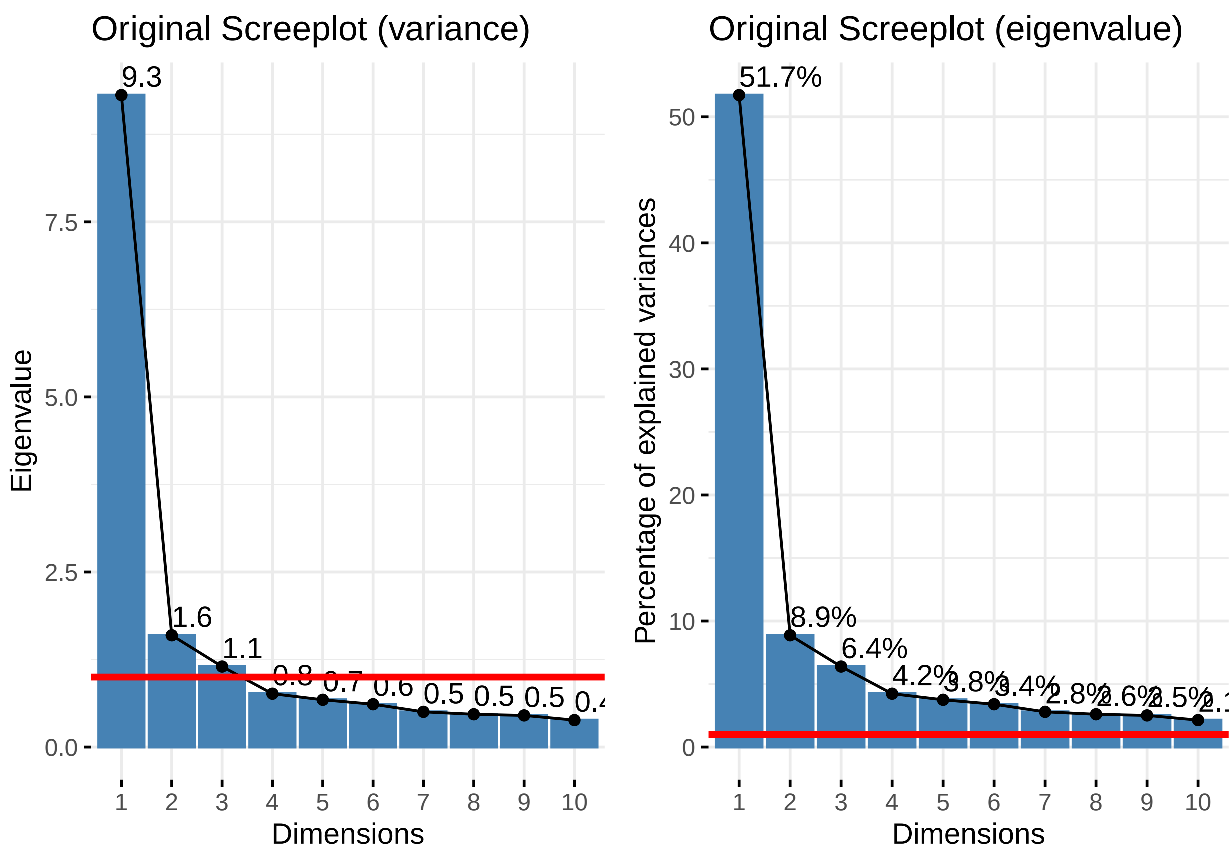

```r
data_pca <- prcomp(data, scale.=TRUE)
```

```r
# Simulating the noise and taking its eigenvalues
sim_noise <- function(n, p){
  noise <- data.frame(replicate(p, rnorm(n))) # 33*10
  correlation <- cor(noise) # 10*10
  eigen(correlation)$values # 10, how much variance can be explained from the noise.
}
eigen_noise <- replicate(100, sim_noise(33, 10)) #100*10
eigen_noise_mean <- apply(eigen_noise, 1, mean)

res.pca <- PCA(data)
```

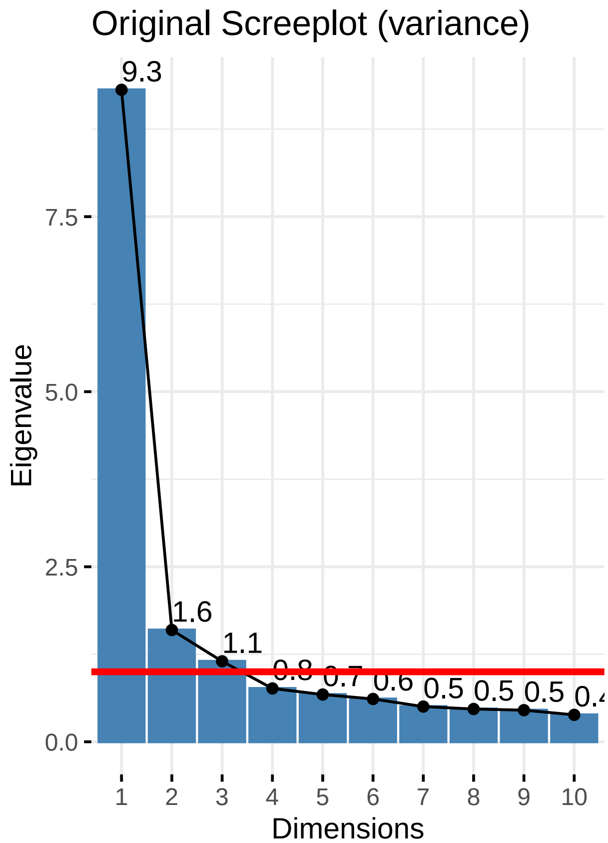

PCA graph of individuals
Dim 2 (8.87%)
Dim 1 (51.73%)
4
2
0
−2
−4
−10
−5
0
5

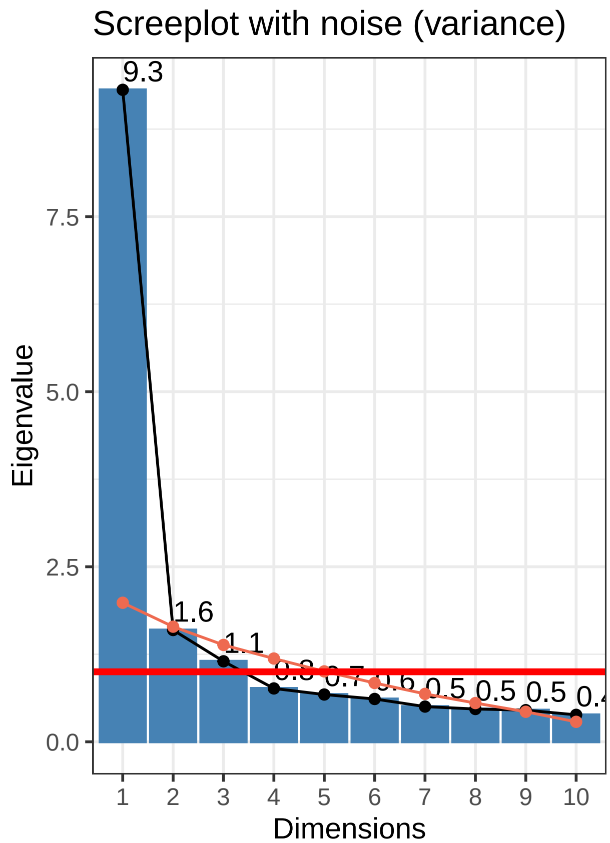



```
p1 <- fviz_eig(res.pca,
               choice = 'eigenvalue',
               addlabels=TRUE,
               main='Original Screeplot (variance)')
p1 <- p1+geom_hline(yintercept = 1,color="red", lwd=1.2)


p2 <- fviz_eig(res.pca,
               choice = 'variance',
               addlabels=TRUE,
               main='Original Screeplot (eigenvalue)')
p2 <- p2+geom_hline(yintercept = 1,color="red", lwd=1.2)

color_list <- "coral2"
p3 <- fviz_eig(res.pca,
               choice = 'eigenvalue',
               addlabels=TRUE,
               main='Screeplot with noise (variance)')+
  geom_point(aes(x=1:10, y=eigen_noise_mean),color=color_list)+
  geom_line(aes(x=1:10, y=eigen_noise_mean),color=color_list)+
  theme_bw()
p3 <- p3+geom_hline(yintercept = 1,color="red", lwd=1.2)

ggarrange(p1, p2, ncol=2, nrow=1)
```




```
ggarrange(p1, p3, ncol=2, nrow=1)
```





b. How many dimensions would you retain if we used Parallel Analysis?

- After conducting ***Horn's Parallel Analysis***, we can easily observe from the plot that the first PC is the only component with a higher eigenvalues compare to the average `noise`. As a result, by comparing variance extracted from the original data versus noise, we can conclude that only a single dimension, the first PC, will be retained in our model.

Question 2) Factor model

Composite measurement model vs. Factor model analysis

Comparison

- In simple terms, the main difference between composite model and factor model is that the latter tends to examine *`model fittness`* by interpreting whether the factors in our model are related to the answer to our question or not; whereas, composite models consist of a set of interrelated composites, all of which emerge as linear combinations of the observable variables.

- Furthermore, factor analysis requires a larger amount of data, while composite measurement model can be done with just few samples.

Factor analysis - Principal

- The `principal` function in r is simply doing a PCA for n principal components of either a correlation or covariance matrix.
- Different from `princomp`, `principal` returns a subset of the ***best n factors***, and the eigenvectors are rescaled by square root of the eigenvlaues to produce the components' **loadings**.

```r
data_principal <- principal(data, nfactor=ncol(data), rotate="none", scores=TRUE)
data_principal |> sapply(head)
```

```
## $values
## [1] 9.3109533 1.5963320 1.1495582 0.7619759 0.6751412 0.6116636
##
## $rotation
## [1] "none"
##
## $n.obs
## [1] 405
##
## $communality
## Q1 Q2 Q3 Q4 Q5 Q6
##  1  1  1  1  1  1
##
## $loadings
##          PC1         PC2          PC3          PC4         PC5         PC6
## Q1 0.8169846 -0.13941235 -0.002115927  0.110179575 -0.03982503  0.14286663
## Q2 0.6726084 -0.01375526  0.089174403  0.225318062  0.07714486  0.62355887
## Q3 0.7655215 -0.03269651  0.089686106 -0.348526020 -0.05075153  0.10504794
## Q4 0.6233733  0.64307826  0.108031860  0.035518829 -0.05991052 -0.05345065
## Q5 0.6900841 -0.03126466 -0.542354570  0.045893140 -0.15875304  0.11679238
## Q6 0.6828029 -0.10462094  0.207232000 -0.003674174  0.50232679  0.04312138
##            PC7          PC8         PC9        PC10        PC11          PC12
## Q1 -0.33733445 -0.008127994 -0.10685731 -0.01588562 -0.15574104 -2.006357e-01
## Q2  0.07362459 -0.253525551  0.01271050  0.01091700  0.08430123  7.812005e-02
## Q3  0.21130888  0.031041586  0.03103346 -0.39051021 -0.12832484 -3.904844e-02
## Q4  0.05193782  0.056604827  0.02286573 -0.08159391 -0.10888112 -5.905673e-05
## Q5  0.13668703  0.129299377  0.14702268  0.06130797  0.05370677 -5.283971e-02
## Q6 -0.04612279  0.368447741  0.22284763 -0.02699113  0.13712796 -9.217760e-02
##            PC13        PC14         PC15        PC16         PC17         PC18
## Q1  0.051019769 -0.11027592  0.051916385 -0.12800198  0.072610801  0.223163787
## Q2  0.007783168  0.07245324 -0.060213096  0.02179430 -0.006298283 -0.035709939
## Q3 -0.042693857 -0.19596819 -0.083886137  0.05235450  0.039891358 -0.033452727
## Q4 -0.173454555  0.27503687 -0.004384802 -0.12609608  0.178328111  0.009388433
## Q5 -0.222611404 -0.07055330  0.203472631 -0.09845601 -0.120570886 -0.033025662
## Q6  0.037119449  0.06261547 -0.037092490 -0.02125200 -0.053540633  0.010971554
##
## $fit
## [1] 1
##
## $fit.off
## [1] 1
##
## $fn
```

```
## [1] "principal"
##
## $Call
## principal(r = data, nfactors = ncol(data), rotate = "none", scores = TRUE)
##
## $uniquenesses
##            Q1            Q2            Q3            Q4            Q5
##  2.220446e-16  1.776357e-15  2.220446e-16  4.440892e-16 -2.220446e-16
##            Q6
##  6.661338e-16
##
## $complexity
##       Q1       Q2       Q3       Q4       Q5       Q6
## 2.152023 2.756187 2.594462 3.027717 3.127357 3.285651
##
## $chi
## [1] 2.115614e-26
##
## $EPVAL
## [1] NA
##
## $R2
## PC1 PC2 PC3 PC4 PC5 PC6
##   1   1   1   1   1   1
##
## $objective
## [1] 0
##
## $residual
##               Q1            Q2           Q3           Q4            Q5
## Q1  1.110223e-16 -1.110223e-16 1.110223e-16 5.551115e-17  0.000000e+00
## Q2 -1.110223e-16  1.665335e-15 9.436896e-16 2.775558e-16  1.665335e-16
## Q3  1.110223e-16  9.436896e-16 3.330669e-16 2.220446e-16  3.330669e-16
## Q4  5.551115e-17  2.775558e-16 2.220446e-16 4.440892e-16  2.775558e-16
## Q5  0.000000e+00  1.665335e-16 3.330669e-16 2.775558e-16 -2.220446e-16
## Q6  2.220446e-16  4.440892e-16 1.110223e-16 5.551115e-17  1.665335e-16
##              Q6            Q7           Q8           Q9          Q10
## Q1 2.220446e-16 -2.220446e-16 1.110223e-16 1.110223e-16 2.220446e-16
## Q2 4.440892e-16  4.440892e-16 9.436896e-16 3.885781e-16 7.771561e-16
## Q3 1.110223e-16  2.775558e-16 7.771561e-16 7.771561e-16 7.771561e-16
## Q4 5.551115e-17  2.775558e-16 2.775558e-16 5.551115e-16 3.885781e-16
## Q5 1.665335e-16  7.216450e-16 3.330669e-16 2.220446e-16 2.220446e-16
## Q6 6.661338e-16  6.661338e-16 4.440892e-16 3.330669e-16 1.665335e-16
##             Q11           Q12          Q13          Q14          Q15
## Q1 0.000000e+00  2.220446e-16 0.000000e+00 3.330669e-16 1.110223e-16
## Q2 8.326673e-16  4.440892e-16 0.000000e+00 3.885781e-16 7.771561e-16
## Q3 4.440892e-16  2.775558e-16 1.110223e-16 4.440892e-16 8.326673e-16
## Q4 4.996004e-16 -3.330669e-16 0.000000e+00 1.110223e-16 5.551115e-17
## Q5 1.110223e-16  5.551115e-17 1.665335e-16 2.775558e-16 3.330669e-16
## Q6 1.665335e-16  6.661338e-16 2.220446e-16 4.440892e-16 1.110223e-16
##             Q16           Q17           Q18
## Q1 0.000000e+00  1.665335e-16 -1.110223e-16
## Q2 8.881784e-16 -1.110223e-16  3.885781e-16
## Q3 3.330669e-16  1.665335e-16  8.881784e-16
```

```
## Q4 4.996004e-16 -2.220446e-16  3.330669e-16
## Q5 3.330669e-16  3.885781e-16  5.551115e-16
## Q6 1.110223e-16  3.330669e-16  2.220446e-16
##
## $rms
## [1] 4.131712e-16
##
## $factors
## [1] 18
##
## $dof
## [1] -18
##
## $null.dof
## [1] 153
##
## $null.model
## [1] 12.40251
##
## $criteria
## objective
##         0        NA        NA
##
## $STATISTIC
## [1] 0
##
## $PVAL
## [1] NA
##
## $weights
##           PC1         PC2          PC3          PC4         PC5         PC6
## Q1 0.08774447 -0.08733293 -0.001840643  0.144597191 -0.05898771  0.23357059
## Q2 0.07223840 -0.00861679  0.077572759  0.295702345  0.11426479  1.01944740
## Q3 0.08221731 -0.02048228  0.078017889 -0.457397691 -0.07517173  0.17174136
## Q4 0.06695053  0.40284745  0.093976850  0.046614110 -0.08873777 -0.08738570
## Q5 0.07411530 -0.01958532 -0.471793912  0.060229123 -0.23514052  0.19094218
## Q6 0.07333329 -0.06553834  0.180270992 -0.004821903  0.74403222  0.07049853
##            PC7         PC8         PC9        PC10       PC11          PC12
## Q1 -0.67066434 -0.01735717 -0.23642509 -0.04124033 -0.4388535 -0.6658844490
## Q2  0.14637516 -0.54139874  0.02812238  0.02834138  0.2375474  0.2592705089
## Q3  0.42010927  0.06628869  0.06866249 -1.01379518 -0.3615990 -0.1295968250
## Q4  0.10325908  0.12087848  0.05059114 -0.21182420 -0.3068097 -0.0001960018
## Q5  0.27175141  0.27611623  0.32529220  0.15916032  0.1513371 -0.1753682735
## Q6 -0.09169806  0.78681278  0.49305726 -0.07007108  0.3864048 -0.3059257472
##           PC13       PC14        PC15        PC16        PC17        PC18
## Q1  0.17455946 -0.4206697  0.22131743 -0.55540931  0.34784094  1.10727007
## Q2  0.02662939  0.2763875 -0.25668597  0.09456695 -0.03017183 -0.17718173
## Q3 -0.14607311 -0.7475602 -0.35760318  0.22716973  0.19109894 -0.16598214
## Q4 -0.59345884  1.0491836 -0.01869223 -0.54713947  0.85427811  0.04658252
## Q5 -0.76164448 -0.2691398  0.86739552 -0.42720735 -0.57759300 -0.16386318
## Q6  0.12700079  0.2388593 -0.15812377 -0.09221387 -0.25648559  0.05443748
##
## $r.scores
##               PC1           PC2           PC3           PC4           PC5
```

```
## PC1  1.000000e+00 -4.163336e-17  0.000000e+00  1.457168e-16 -2.602085e-16
## PC2 -1.214306e-17  1.000000e+00 -1.309716e-16 -9.540979e-17 -5.204170e-17
## PC3  2.515349e-17 -1.188286e-16  1.000000e+00 -2.740863e-16 -2.634611e-16
## PC4 -1.023487e-16 -5.767956e-17 -3.582204e-16  1.000000e+00  3.382711e-17
## PC5 -2.302845e-16 -6.353425e-17 -1.758576e-16  5.659535e-17  1.000000e+00
## PC6 -1.773755e-16  4.165505e-16  2.645453e-16 -6.804453e-16 -5.031782e-16
##               PC6           PC7           PC8           PC9          PC10
## PC1 -2.359224e-16 -6.661338e-16  2.775558e-16 -8.326673e-17  2.463307e-16
## PC2  4.362830e-16 -5.551115e-17  3.469447e-18 -1.249001e-16 -1.669671e-16
## PC3  3.812055e-16  2.289835e-16 -1.517883e-16 -1.040834e-16 -8.326673e-17
## PC4 -6.643991e-16 -7.008283e-16  5.204170e-17 -3.348016e-16  3.786034e-16
## PC5 -4.009380e-16 -4.163336e-16  1.409463e-16  5.204170e-17  1.008308e-16
## PC6  1.000000e+00 -1.977585e-16  5.412337e-16 -1.066855e-16  5.209591e-16
##              PC11          PC12          PC13          PC14          PC15
## PC1 -2.775558e-16 -7.771561e-16  1.110223e-16 -6.938894e-17  1.665335e-16
## PC2 -1.457168e-16 -5.551115e-17  3.469447e-17  7.285839e-17 -8.326673e-17
## PC3 -3.955170e-16 -1.110223e-16  6.765422e-16  2.233456e-16 -5.551115e-16
## PC4 -2.706169e-16  3.885781e-16 -4.267420e-16  1.487525e-16  1.457168e-16
## PC5 -6.149595e-16  3.365364e-16  1.899522e-16  9.473759e-16 -1.539567e-16
## PC6 -3.642919e-16 -9.645063e-16 -6.713380e-16 -6.245005e-16  1.577731e-15
##              PC16         PC17          PC18
## PC1 -1.110223e-16 2.498002e-16  5.551115e-16
## PC2 -2.498002e-16 7.632783e-17 -2.359224e-16
## PC3  8.187895e-16 4.510281e-16  4.024558e-16
## PC4  1.110223e-16 1.196959e-16  8.222589e-16
## PC5 -2.671474e-16 6.990936e-16 -6.591949e-17
## PC6 -5.759282e-16 6.765422e-17  8.604228e-16
## 
## $Vaccounted
##                             PC1        PC2        PC3        PC4        PC5
## SS loadings           9.3109533 1.59633195 1.14955822 0.76197591 0.67514118
## Proportion Var        0.5172752 0.08868511 0.06386435 0.04233199 0.03750784
## Cumulative Var        0.5172752 0.60596029 0.66982464 0.71215663 0.74966447
## Proportion Explained  0.5172752 0.08868511 0.06386435 0.04233199 0.03750784
## Cumulative Proportion 0.5172752 0.60596029 0.66982464 0.71215663 0.74966447
##                              PC6        PC7        PC8        PC9      PC10
## SS loadings           0.61166360 0.50298552 0.46827880 0.45197111 0.3851964
## Proportion Var        0.03398131 0.02794364 0.02601549 0.02510951 0.0213998
## Cumulative Var        0.78364578 0.81158943 0.83760491 0.86271442 0.8841142
## Proportion Explained  0.03398131 0.02794364 0.02601549 0.02510951 0.0213998
## Cumulative Proportion 0.78364578 0.81158943 0.83760491 0.86271442 0.8841142
##                             PC11       PC12       PC13       PC14       PC15
## SS loadings           0.35488164 0.30130710 0.29227731 0.26214370 0.23457883
## Proportion Var        0.01971565 0.01673928 0.01623763 0.01456354 0.01303216
## Cumulative Var        0.90382986 0.92056915 0.93680678 0.95137031 0.96440247
## Proportion Explained  0.01971565 0.01673928 0.01623763 0.01456354 0.01303216
## Cumulative Proportion 0.90382986 0.92056915 0.93680678 0.95137031 0.96440247
##                             PC16       PC17      PC18
## SS loadings           0.23046423 0.20874714 0.2015441
## Proportion Var        0.01280357 0.01159706 0.0111969
## Cumulative Var        0.97720604 0.98880310 1.0000000
## Proportion Explained  0.01280357 0.01159706 0.0111969
## Cumulative Proportion 0.97720604 0.98880310 1.0000000
## 
```

```
## $Structure
##          PC1         PC2          PC3          PC4         PC5         PC6
## Q1 0.8169846 -0.13941235 -0.002115927  0.110179575 -0.03982503  0.14286663
## Q2 0.6726084 -0.01375526  0.089174403  0.225318062  0.07714486  0.62355887
## Q3 0.7655215 -0.03269651  0.089686106 -0.348526020 -0.05075153  0.10504794
## Q4 0.6233733  0.64307826  0.108031860  0.035518829 -0.05991052 -0.05345065
## Q5 0.6900841 -0.03126466 -0.542354570  0.045893140 -0.15875304  0.11679238
## Q6 0.6828029 -0.10462094  0.207232000 -0.003674174  0.50232679  0.04312138
##            PC7          PC8         PC9        PC10        PC11          PC12
## Q1 -0.33733445 -0.008127994 -0.10685731 -0.01588562 -0.15574104 -2.006357e-01
## Q2  0.07362459 -0.253525551  0.01271050  0.01091700  0.08430123  7.812005e-02
## Q3  0.21130888  0.031041586  0.03103346 -0.39051021 -0.12832484 -3.904844e-02
## Q4  0.05193782  0.056604827  0.02286573 -0.08159391 -0.10888112 -5.905673e-05
## Q5  0.13668703  0.129299377  0.14702268  0.06130797  0.05370677 -5.283971e-02
## Q6 -0.04612279  0.368447741  0.22284763 -0.02699113  0.13712796 -9.217760e-02
##            PC13        PC14         PC15        PC16         PC17         PC18
## Q1  0.051019769 -0.11027592  0.051916385 -0.12800198  0.072610801  0.223163787
## Q2  0.007783168  0.07245324 -0.060213096  0.02179430 -0.006298283 -0.035709939
## Q3 -0.042693857 -0.19596819 -0.083886137  0.05235450  0.039891358 -0.033452727
## Q4 -0.173454555  0.27503687 -0.004384802 -0.12609608  0.178328111  0.009388433
## Q5 -0.222611404 -0.07055330  0.203472631 -0.09845601 -0.120570886 -0.033025662
## Q6  0.037119449  0.06261547 -0.037092490 -0.02125200 -0.053540633  0.010971554
##
## $scores
##             PC1       PC2        PC3        PC4        PC5        PC6
## [1,]  0.5818382 1.5963666 -1.5292203  0.6131293 -0.1070697 -0.4767162
## [2,]  0.4403780 0.7030085 -0.2366072 -0.5235704  0.2916363 -0.5458000
## [3,]  0.5877040 0.3371982  0.0717378  0.4744385 -0.7410415  0.7191103
## [4,] -0.2953784 0.2361530  0.3360654  0.8811024  0.4966228 -0.1464620
## [5,]  0.9548232 1.0660996 -0.1332467 -0.1233820 -1.7906111  1.0969491
## [6,] -0.3952680 0.2698351  1.8157724 -0.1488750  0.1993347 -0.2260891
##             PC7        PC8        PC9        PC10       PC11        PC12
## [1,] -1.1274984 -0.4531817 -0.2008367  0.29175628 -0.6301099  0.12999789
## [2,]  0.3656110  0.1566416 -0.9202677  0.05186623  0.2269288  0.52310913
## [3,]  0.4146520  1.5792306  0.5573955  0.04142999 -0.1081679 -0.37155216
## [4,] -0.1010603  0.1192617  0.6733886 -0.25702964 -0.1320169  0.22764912
## [5,] -0.4711391 -0.4163718  0.2977361  0.15925268 -0.9401995 -0.09458162
## [6,] -2.0364905 -1.8124301 -0.3160456  0.45429950 -0.4123576  0.08255907
##             PC13       PC14       PC15       PC16        PC17       PC18
## [1,] -0.00335515 -0.5006310  0.4210351  0.1235163  0.98603412  0.5570782
## [2,] -0.31062493  0.1152664 -0.8779982  0.1513349 -0.55889923 -1.0873370
## [3,]  0.59279546  0.3269905 -0.4547884 -0.5163004 -0.89563299 -0.8394719
## [4,] -0.24782563 -0.4739046 -0.4348062  0.8311631  0.75221340 -0.5196657
## [5,] -0.20667378 -0.6717545 -1.4019200 -0.4443238 -0.01988391 -0.8785831
## [6,] -1.14432785  0.1672917  0.7695079 -0.7563891 -1.34971082 -0.1638684
```

```r
# ss_loadings = eigenvalues
ss_loadings <- sum(data_principal$loadings[,"PC1"]^2)
ss_loadings
```

```
## [1] 9.310953
```

```r
data_principal$values[1]
```

```
## [1] 9.310953
```

a. To which components does each item seem to best belong? (3 components)

```r
# function that find good loadings in each PC.
find_unique <- function(data_principal, n){
  for (i in 1:n){
    if (max(data_principal$loadings[,i]) > 0.7){
      important_ones <- data_principal$loadings[,i][data_principal$loadings[,i]>=0.7]
      important_ones <- sort(important_ones, decreasing=TRUE)
      cat("Component", i, "has good loadings:", names(important_ones))
    }else{
      important_ones <- sort(data_principal$loadings[,i], decreasing = TRUE)[1]
      cat("Component", i, "has good no loadings.\n")
      cat('The largest lambda value is: ', important_ones)
    }
    name <- names(important_ones)
    cat("\nBest fit:", name)
    cat('\n')
    cat('\n')
  }
}

head(data_principal$loadings[,1:3])
```

```
##          PC1         PC2          PC3
## Q1 0.8169846 -0.13941235 -0.002115927
## Q2 0.6726084 -0.01375526  0.089174403
## Q3 0.7655215 -0.03269651  0.089686106
## Q4 0.6233733  0.64307826  0.108031860
## Q5 0.6900841 -0.03126466 -0.542354570
## Q6 0.6828029 -0.10462094  0.207232000
```

```r
data_principal$Vaccounted[, 1:5]
```

```
##                             PC1        PC2        PC3        PC4        PC5
## SS loadings           9.3109533 1.59633195 1.14955822 0.76197591 0.67514118
## Proportion Var        0.5172752 0.08868511 0.06386435 0.04233199 0.03750784
## Cumulative Var        0.5172752 0.60596029 0.66982464 0.71215663 0.74966447
## Proportion Explained  0.5172752 0.08868511 0.06386435 0.04233199 0.03750784
## Cumulative Proportion 0.5172752 0.60596029 0.66982464 0.71215663 0.74966447
```

```r
for (i in 1:18){
  cat(paste0('Best belong for Question ', i),': ')
  cat(names(sort(data_principal$loadings[i,], decreasing=TRUE))[1])
  cat('\n')
}
```

```
## Best belong for Question 1 : PC1
## Best belong for Question 2 : PC1
## Best belong for Question 3 : PC1
## Best belong for Question 4 : PC2
## Best belong for Question 5 : PC1
## Best belong for Question 6 : PC1
## Best belong for Question 7 : PC1
## Best belong for Question 8 : PC1
## Best belong for Question 9 : PC1
## Best belong for Question 10 : PC1
## Best belong for Question 11 : PC1
## Best belong for Question 12 : PC2
## Best belong for Question 13 : PC1
## Best belong for Question 14 : PC1
## Best belong for Question 15 : PC1
## Best belong for Question 16 : PC1
## Best belong for Question 17 : PC2
## Best belong for Question 18 : PC1
```

```
find_unique(data_principal, 18)
```

```
## Component 1 has good loadings: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
## Best fit: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
##
## Component 2 has good no loadings.
## The largest lambda value is:  0.6642605
## Best fit: Q17
##
## Component 3 has good no loadings.
## The largest lambda value is:  0.3241768
## Best fit: Q7
##
## Component 4 has good no loadings.
## The largest lambda value is:  0.2857201
## Best fit: Q7
##
## Component 5 has good no loadings.
## The largest lambda value is:  0.5023268
## Best fit: Q6
##
## Component 6 has good no loadings.
## The largest lambda value is:  0.6235589
## Best fit: Q2
##
## Component 7 has good no loadings.
## The largest lambda value is:  0.3219782
## Best fit: Q7
##
## Component 8 has good no loadings.
## The largest lambda value is:  0.3684477
## Best fit: Q6
##
## Component 9 has good no loadings.
## The largest lambda value is:  0.4005233
```

```
## Best fit: Q9
##
## Component 10 has good no loadings.
## The largest lambda value is:  0.3049279
## Best fit: Q13
##
## Component 11 has good no loadings.
## The largest lambda value is:  0.2356899
## Best fit: Q11
##
## Component 12 has good no loadings.
## The largest lambda value is:  0.2266301
## Best fit: Q11
##
## Component 13 has good no loadings.
## The largest lambda value is:  0.2943817
## Best fit: Q10
##
## Component 14 has good no loadings.
## The largest lambda value is:  0.2750369
## Best fit: Q4
##
## Component 15 has good no loadings.
## The largest lambda value is:  0.2034726
## Best fit: Q5
##
## Component 16 has good no loadings.
## The largest lambda value is:  0.2463264
## Best fit: Q17
##
## Component 17 has good no loadings.
## The largest lambda value is:  0.1783281
## Best fit: Q4
##
## Component 18 has good no loadings.
## The largest lambda value is:  0.2231638
## Best fit: Q1
```

- **Conclusion.** Items are either best belong to PC1 or PC2. Also, only the first PC has good loadings.

b. How much of the total variance of the security dataset do the first 3 PCs capture?

```
data_principal$Vaccounted['Cumulative Var','PC3']
```

```
## [1] 0.6698246
```

c. Which items are less than adequately explained by the first 3 principal components?

- **Tips.** Communality and Uniqueness.

```r
# communality = h^2
data_pca3 <- principal(data, nfactor=3, rotate='none', scores=TRUE)

# variances of variables explained by a specified amount of principal components.
community <- data_pca3$communality
community
```

```
##        Q1        Q2        Q3        Q4        Q5        Q6        Q7        Q8
## 0.6869041 0.4605433 0.5951359 0.8138147 0.7713420 0.5201104 0.6371369 0.7375512
##        Q9       Q10       Q11       Q12       Q13       Q14       Q15       Q16
## 0.6178667 0.7642903 0.6648554 0.8185557 0.5181043 0.6930021 0.6063756 0.6485852
##       Q17       Q18
## 0.8347032 0.6679663
```

```r
# 1-communality, unexplained variance of a certain variable.
data_pca3$uniquenesses
```

```
##        Q1        Q2        Q3        Q4        Q5        Q6        Q7        Q8
## 0.3130959 0.5394567 0.4048641 0.1861853 0.2286580 0.4798896 0.3628631 0.2624488
##        Q9       Q10       Q11       Q12       Q13       Q14       Q15       Q16
## 0.3821333 0.2357097 0.3351446 0.1814443 0.4818957 0.3069979 0.3936244 0.3514148
##       Q17       Q18
## 0.1652968 0.3320337
```

- Think we're dealing with an item with low ***communality*** and high ***uniqueness*** value.

```r
temp = sort(data_pca3$communality)
names(temp[temp<=0.7])
```

```
##  [1] "Q2"  "Q13" "Q6"  "Q3"  "Q15" "Q9"  "Q7"  "Q16" "Q11" "Q18" "Q1"  "Q14"
```

```r
temp = sort(data_pca3$uniquenesses, decreasing = TRUE)
names(temp[temp>=0.3])
```

```
##  [1] "Q2"  "Q13" "Q6"  "Q3"  "Q15" "Q9"  "Q7"  "Q16" "Q11" "Q18" "Q1"  "Q14"
```

- **Conclusion.** Q2 is the least to be adequately explained by the first 3 PC.

d. How many measurement items share similar loadings between 2 or more components?

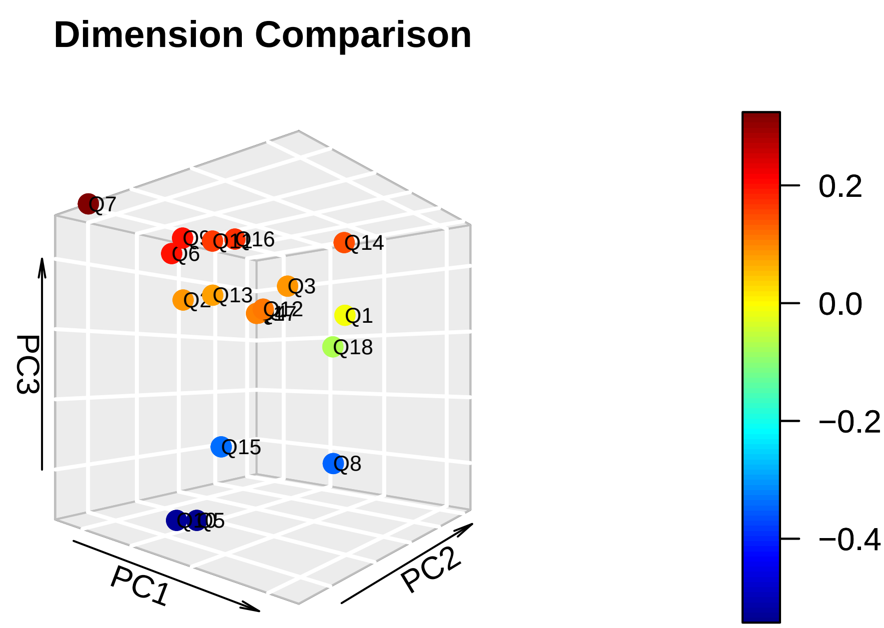
```r
data_principal <- principal(data, nfactor=ncol(data), rotate='none', scores=TRUE)

data_principal$loadings |> round(2)
```

```
## 
## Loadings:
##     PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9   PC10  PC11  PC12
## Q1   0.82 -0.14        0.11        0.14 -0.34       -0.11       -0.16 -0.20
```

```
## Q2   0.67              0.23        0.62        -0.25
## Q3   0.77             -0.35        0.11  0.21              -0.39 -0.13
## Q4   0.62  0.64  0.11                                            -0.11
## Q5   0.69       -0.54       -0.16  0.12  0.14  0.13  0.15
## Q6   0.68 -0.10  0.21        0.50              0.37  0.22        0.14
## Q7   0.66 -0.32  0.32  0.29              0.32  0.16 -0.16  0.20 -0.26
## Q8   0.79       -0.34        0.17 -0.16       -0.14 -0.16       -0.13 -0.17
## Q9   0.72 -0.23  0.20 -0.11       -0.21       -0.31  0.40  0.16
## Q10  0.69 -0.10 -0.53       -0.20        0.11  0.17
## Q11  0.75 -0.26  0.17  0.23 -0.17 -0.15        0.12 -0.19        0.24  0.23
## Q12  0.63  0.64  0.12                                      0.10
## Q13  0.71             -0.53                          -0.19  0.30  0.18
## Q14  0.81 -0.10  0.16 -0.32                          -0.15              0.13
## Q15  0.70       -0.33        0.42 -0.20  0.11 -0.21 -0.17 -0.12  0.11
## Q16  0.76 -0.20  0.18  0.18 -0.28 -0.17       -0.13       -0.13  0.23 -0.26
## Q17  0.62  0.66  0.11       -0.13
## Q18  0.81 -0.11                         -0.41        0.12       -0.14  0.21
##     PC13  PC14  PC15  PC16  PC17  PC18
## Q1        -0.11       -0.13        0.22
## Q2
## Q3        -0.20
## Q4   -0.17  0.28       -0.13  0.18
## Q5   -0.22        0.20 -0.10 -0.12
## Q6
## Q7
## Q8               -0.25       -0.14 -0.14
## Q9                                  0.10
## Q10  0.29       -0.13        0.11
## Q11 -0.12       -0.15 -0.14
## Q12  0.21 -0.24       -0.14       -0.17
## Q13 -0.11                    0.12
## Q14  0.16  0.20  0.16       -0.23
## Q15              0.16        0.10  0.10
## Q16                               -0.12
## Q17                    0.25 -0.18  0.19
## Q18 -0.11              0.20       -0.14
##
##                  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
## SS loadings    9.319 1.587 1.137 0.775 0.667 0.607 0.500 0.470 0.450 0.383
## Proportion Var 0.518 0.088 0.063 0.043 0.037 0.034 0.028 0.026 0.025 0.021
## Cumulative Var 0.518 0.606 0.669 0.712 0.749 0.783 0.811 0.837 0.862 0.883
##                 PC11  PC12  PC13  PC14  PC15  PC16  PC17  PC18
## SS loadings    0.358 0.303 0.287 0.268 0.232 0.233 0.206 0.199
## Proportion Var 0.020 0.017 0.016 0.015 0.013 0.013 0.011 0.011
## Cumulative Var 0.903 0.920 0.936 0.951 0.963 0.976 0.988 0.999
```

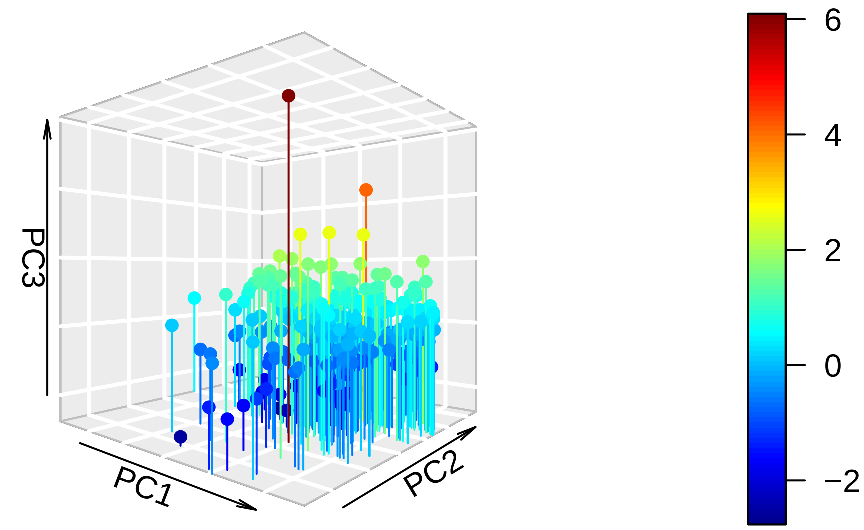
```r
x<- unname(data_pca3$loadings[,1])
y<- unname(data_pca3$loadings[,2])
z<- unname(data_pca3$loadings[,3])

scatter3D(x, y, z,
          phi=0,
          bty='g',
          main="Dimension Comparison",
```

```
          xlab='PC1', ylab='PC2', zlab='PC3',
          cex=2, pch=20,
          ticktype="simple")
text3D(x, y, z, labels=paste0("Q", 1:18), add=TRUE, colkey = FALSE, cex=0.7)
```




```
x<- unname(data_pca3$scores[,1])
y<- unname(data_pca3$scores[,2])
z<- unname(data_pca3$scores[,3])

scatter3D(x, y, z,
          phi=0,
          bty='g',
          main="Subjects scores",
          xlab='PC1', ylab='PC2', zlab='PC3',
          cex=1.2, pch=20,
          ticktype="simple",
          type="h")
```

Subjects scores



- **Ans.** Q4, Q12 and Q17 share similar loadings since about the same amount of variance is explained by each PC.

e. Can you interpret a `meaning` behind the first principal component from the items that load best upon it?

```
find_unique(data_principal, 1)
```

```
## Component 1 has good loadings: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
## Best fit: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
```

```
best_fit_PC1 <- sort(c(1, 14, 18, 8, 3, 16, 11, 9, 13, 15))
questions[best_fit_PC1,]
```

```
## # A tibble: 10 x 2
##    Index Questions
##    <chr> <chr>
##  1 Q1    I am convinced that this site respects the confidentiality of the tran~
##  2 Q3    This site checks the information communicated with me for accuracy
##  3 Q8    This site ascertains my identity before processing the transactions re~
##  4 Q9    I can remove my personal information from this site when I want to
##  5 Q11   This site devotes time and effort to preventing unauthorized access to~
```

```
##  6 Q13   This site provides me with some evidence to protect against its denial~
##  7 Q14   This site devotes time and effort to verify the accuracy of the inform~
##  8 Q15   This site ascertains my identity before sending any messages to me
##  9 Q16   Databases that contain my personal information are protected from unau~
## 10 Q18   This site uses some security controls for the confidentiality of the t~
```

- **Ans.** I'm guessing that consumers perceive the security of e-commerce websites largely on its functionality and security.

Question 3) Rotated Components

a. Does each rotated component (RC) explain the same, or different, amount of variance than the corresponding PCs?

```r
data_principal <- principal(data, nfactor=3, rotate='none', scores=TRUE)
data_r_principal <- principal(data, nfactor=3, rotate='varimax', scores=TRUE)

data_principal$Vaccounted["Proportion Var",]
```

```
##        PC1        PC2        PC3
## 0.51727518 0.08868511 0.06386435
```

```r
data_r_principal$Vaccounted["Proportion Var",]
```

```
##       RC1       RC3       RC2
## 0.3118416 0.1938966 0.1640864
```

- **Ans.** They are all different from the original PC. RC1 is less then the first PC.

b. Do the 3 RCs explain the same, more, or less cumulative variance as the 3 PCs combined?

```r
data_principal$Vaccounted["Cumulative Var",]
```

```
##       PC1       PC2       PC3
## 0.5172752 0.6059603 0.6698246
```

```r
data_r_principal$Vaccounted["Cumulative Var",]
```

```
##       RC1       RC3       RC2
## 0.3118416 0.5057382 0.6698246
```

- **Ans.** They give the same cumulative variance eventually.

c. Refer to Question 2 (d), do those items have more clearly differentiated `loadings` among rotated components?

```
best_fit_PC1
```

```
##  [1]  1  3  8  9 11 13 14 15 16 18
```

```
apply(data_principal$loadings[,1:3], 1, sum)
```

```
##         Q1         Q2         Q3         Q4         Q5         Q6         Q7
## 0.67545634 0.74802751 0.82251113 1.37448338 0.11646488 0.78541392 0.66316968
##         Q8         Q9        Q10        Q11        Q12        Q13        Q14
## 0.48525232 0.69493938 0.05479378 0.66448296 1.38940461 0.73160603 0.86855457
##        Q15        Q16        Q17        Q18
## 0.38207531 0.73791584 1.39185527 0.62793493
```

```
apply(data_r_principal$loadings[,1:3], 1, sum)
```

```
##        Q1        Q2        Q3        Q4        Q5        Q6        Q7        Q8
## 1.3306176 1.1180147 1.2681355 1.2653345 1.2339059 1.0856569 0.9486593 1.3930230
##        Q9       Q10       Q11       Q12       Q13       Q14       Q15       Q16
## 1.1090290 1.2019089 1.1537269 1.2731707 1.1670301 1.3121035 1.2415778 1.1805769
##       Q17       Q18
## 1.2626960 1.3317078
```

- **Ans.** Looking at the table, almost every measurement items increased, instead of `Q4`, `Q12` and `Q17`, which are the only 3 items that are not best belong to PC1 originally.

d. Can you now more easily interpret the `meaning` of the 3 RCs from the items that load best upon each of them?

```
for (i in 1:18){
  cat(paste0('Best belong for Question ', i),': ')
  cat(names(sort(data_r_principal$loadings[i,], decreasing=TRUE))[1])
  cat('\n')
}
```

```
## Best belong for Question 1 : RC1
## Best belong for Question 2 : RC1
## Best belong for Question 3 : RC1
## Best belong for Question 4 : RC2
## Best belong for Question 5 : RC3
## Best belong for Question 6 : RC1
## Best belong for Question 7 : RC1
## Best belong for Question 8 : RC3
## Best belong for Question 9 : RC1
## Best belong for Question 10 : RC3
## Best belong for Question 11 : RC1
## Best belong for Question 12 : RC2
## Best belong for Question 13 : RC1
## Best belong for Question 14 : RC1
## Best belong for Question 15 : RC3
```

```
## Best belong for Question 16 : RC1
## Best belong for Question 17 : RC2
## Best belong for Question 18 : RC1
```

```
find_unique(data_r_principal, 3)
```

```
## Component 1 has good loadings: Q7 Q11 Q16 Q9 Q14
## Best fit: Q7 Q11 Q16 Q9 Q14
##
## Component 2 has good loadings: Q5 Q10 Q8
## Best fit: Q5 Q10 Q8
##
## Component 3 has good loadings: Q17 Q12 Q4
## Best fit: Q17 Q12 Q4
```

```
find_unique(data_principal, 3)
```

```
## Component 1 has good loadings: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
## Best fit: Q1 Q14 Q18 Q8 Q3 Q16 Q11 Q9 Q13 Q15
##
## Component 2 has good no loadings.
## The largest lambda value is:  0.6642605
## Best fit: Q17
##
## Component 3 has good no loadings.
## The largest lambda value is:  0.3241768
## Best fit: Q7
```

- **Ans.** Yes.

e. If we reduced the number of extracted and RCs to 2, does the meaning of our RCs change?

- RC1: concerns of the sites security. (unautorized access)
- RC2: concerns of transaction transmission.
- RC3: concerns of transaction evidential protection.

Ungraded Question

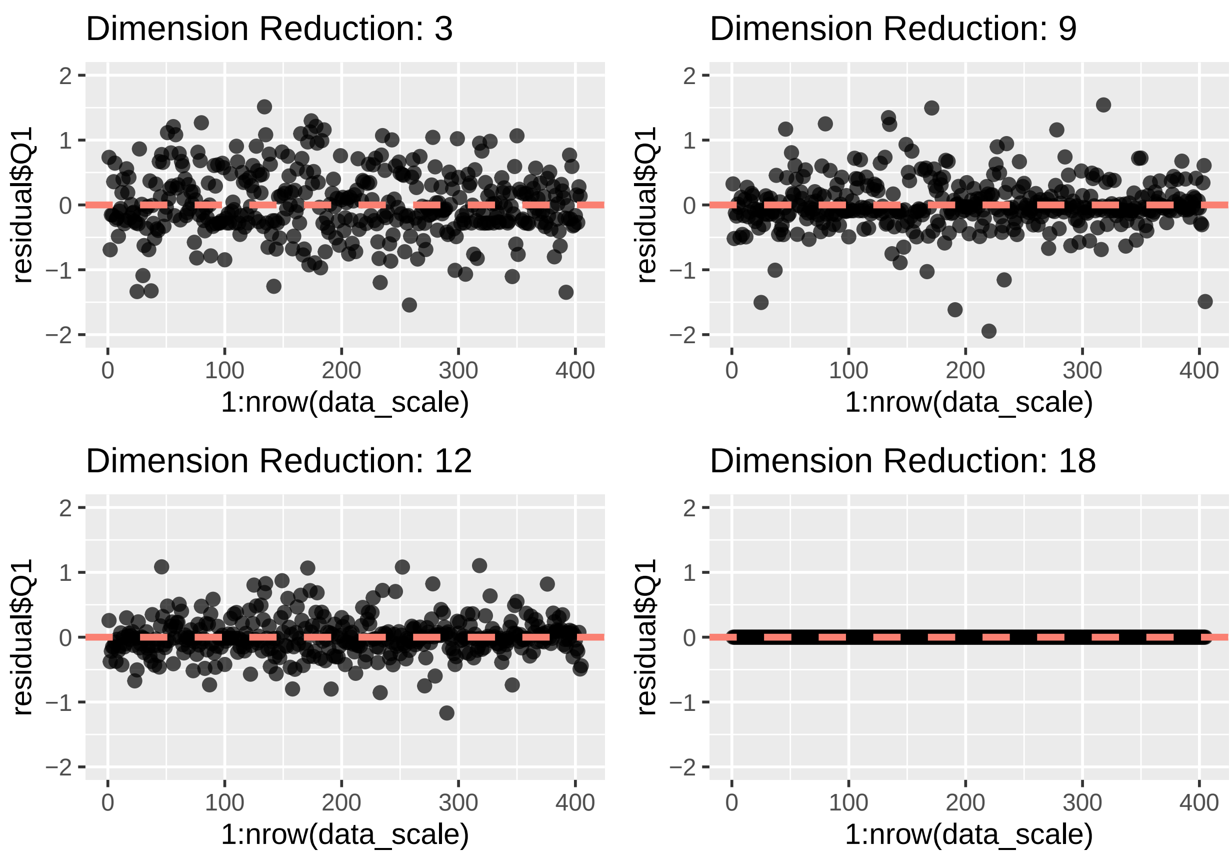
Q. How many components (1-3) do you believe we should extract and analyze to understand this dataset?

Additional Practice

Reproducing data

```r
# reproduce data from various pc dimensions
reproduce_data <- function(original, num_pc){
  pca_results <- prcomp(original, scale=TRUE)
  scores <- pca_results$x[, 1:num_pc]
  weights <- pca_results$rotation[, 1:num_pc]
  reproduction <- scores %*% t(weights)
  return(reproduction)
}

residual_plt <- function(original, reproduction, ndim){
  residual <- as.data.frame(original-reproduction)
  ggplot()+
    aes(x=1:nrow(data_scale), y=residual$Q1)+
    geom_point(alpha=0.7, size=2)+
    ylim(-2,2)+
    geom_hline(yintercept = 0,
               col='salmon',
               lwd=1.2,
               lty=2)+
    ggtitle(paste0("Dimension Reduction: ", ndim))
}

data_scale <- scale(data)
ndim=3
temp = reproduce_data(data_scale, ndim)
p1 <- residual_plt(data_scale, temp,ndim=ndim)
ndim=9
temp = reproduce_data(data_scale, ndim)
p2 <- residual_plt(data_scale, temp,ndim=ndim)
ndim=12
temp = reproduce_data(data_scale, ndim)
p3 <- residual_plt(data_scale, temp,ndim=ndim)
ndim=18
temp = reproduce_data(data_scale, ndim)
p4 <- residual_plt(data_scale, temp,ndim=ndim)

ggarrange(p1, p2, p3, p4, ncol=2, nrow=2)
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



Dimension Reduction: 3
Dimension Reduction: 9
Dimension Reduction: 12
Dimension Reduction: 18
residual$Q1
1:nrow(data_scale)