HW 14

110078509

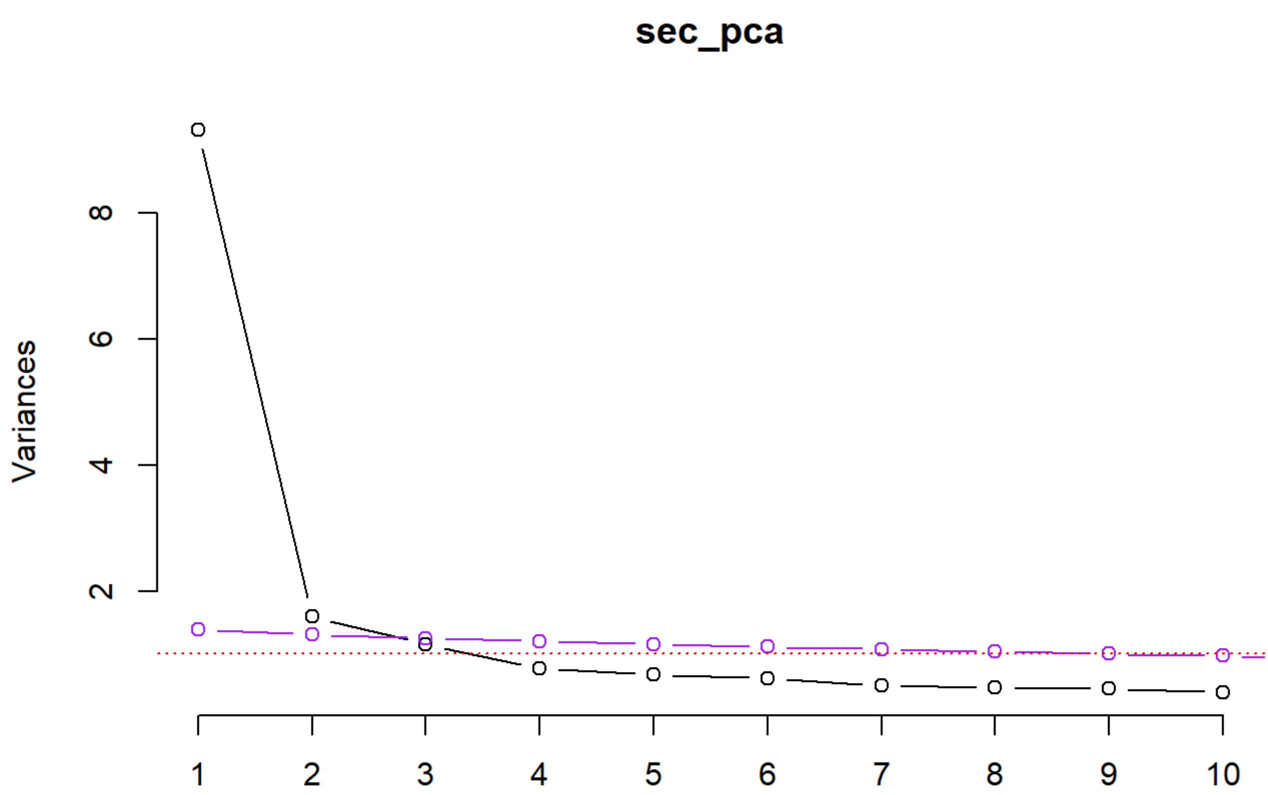
20220522

```
df_question <- read.xlsx("security_questions.xlsx", sheet = 1)
df_data <- read.xlsx("security_questions.xlsx", sheet = 2)

sec_pca <- prcomp(df_data, scale. = T)
```

Question 1.

a. Show a single visualization with scree plot of data, scree plot of simulated noise (use average eigenvalues of ≥ 100 noise samples), and a horizontal line showing the eigenvalue = 1 cutoff.

```
##  1. Function to run a PCA on n × p dataframe of random values
set.seed(100)

sim_noise_ev <- function(n, p) {
noise <- data.frame(replicate(p, rnorm(n)))
eigen(cor(noise))$values
}

## 2. Repeat this k times
n <- dim(df_data)[1];
p <- dim(df_data)[2];

evalues_noise <- replicate(100, sim_noise_ev(n, p))

## 3. Average each of the noise eigenvalue
evalues_mean <- apply(evalues_noise, 1, mean)

## 4. ScreePlot
sec_pca <- prcomp(df_data, scale. = T)
screeplot(sec_pca, type="lines")
lines(evalues_mean, type="b", col = "purple")
abline(h=1, lty="dotted", col = "red")
```



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

Ans:

2 dimensions. Because only the first 2 dimensions are higher than the noise performance.

Question 2)

a. Looking at the loadings of the first 3 principal components, to which components does each item seem to best belong?

```
# - Performs PCA, reports factor loadings
s_principal <- df_data |> principal(nfactor=10, rotate="none", scores=TRUE)
# s_principal3 <- df_data |> principal(nfactor=3, rotate="none", scores=TRUE)
```

- PC1

```
lpc1<-as.data.frame(s_principal$loadings[,1])
colnames(lpc1) <- 'loading_PC1'
arrange_PC1<-arrange(lpc1,desc(loading_PC1))
arrange_PC1
```

```
##     loading_PC1
## Q1    0.8169846
## Q14   0.8114677
## Q18   0.8067284
## Q8    0.7861054
## Q3    0.7655215
## Q16   0.7575616
## Q11   0.7529735
## Q9    0.7230295
## Q13   0.7119085
## Q15   0.7040428
## Q5    0.6900841
## Q10   0.6861529
## Q6    0.6828029
## Q2    0.6726084
## Q7    0.6566249
## Q12   0.6303505
## Q4    0.6233733
## Q17   0.6175336
```

- PC2

```
lpc2<-as.data.frame(s_principal$loadings[,2])
colnames(lpc2) <- 'loading_PC2'
arrange_PC2<-arrange(lpc2,desc(loading_PC2))
arrange_PC2
```

```
##     loading_PC2
## Q17  0.66426051
## Q4   0.64307826
## Q12  0.63753124
## Q8   0.04235983
## Q15  0.01057936
## Q2  -0.01375526
## Q5  -0.03126466
## Q3  -0.03269651
## Q13 -0.06463837
## Q10 -0.09868038
## Q14 -0.09970016
## Q6  -0.10462094
## Q18 -0.11360432
## Q1  -0.13941235
## Q16 -0.20281591
## Q9  -0.23164618
## Q11 -0.26100673
## Q7  -0.31763196
```

- PC3

```
lpc3<-as.data.frame(s_principal$loadings[,3])
colnames(lpc3) <- 'loading_PC3'
arrange_PC3<-arrange(lpc3,desc(loading_PC3))
arrange_PC3
```

```
##      loading_PC3
## Q7   0.324176779
## Q6   0.207232000
## Q9   0.203556038
## Q16  0.183170175
## Q11  0.172516196
## Q14  0.156787046
## Q12  0.121522834
## Q17  0.110061160
## Q4   0.108031860
## Q3   0.089686106
## Q2   0.089174403
## Q13  0.084335919
## Q1  -0.002115927
## Q18 -0.065189145
## Q15 -0.332546876
## Q8  -0.343212951
## Q10 -0.532678749
## Q5  -0.542354570
```

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

reference: https://ppfocus.com/0/edc2fbae7.html (https://ppfocus.com/0/edc2fbae7.html)

```
# attributes(s_principal)
s_principal$Vaccounted[,c(1:3)] |> round(2)
```

```
##                        PC1  PC2  PC3
## SS loadings           9.31 1.60 1.15
## Proportion Var        0.52 0.09 0.06
## Cumulative Var        0.52 0.61 0.67
## Proportion Explained  0.59 0.10 0.07
## Cumulative Proportion 0.59 0.69 0.76
```

The Cumulative Var represented the total variance explained by PCs. From PC1 to PC3, the Cumulative Var is 0.67.

---

c. Looking at commonality and uniqueness, which items are less than adequately explained by the first 3 principal components?

Reference: https://www.statisticshowto.com/communality/ (https://www.statisticshowto.com/communality/)

1. Communality (H2: 公因子方差) estimates for each item. These are merely the sum of squared factor loadings for that item, range from 0~1. It can be considered as the proportion of common variance found in a particular variable.The higer the better. (即主成分對每個變量的方差解釋度)

```
s_principal3 <- df_data |> principal(nfactor=3, rotate="none", scores=TRUE)

s_principal3$communality |> sort(decreasing = TRUE)
```

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

2. Uniqueness (u2: 成分唯一性), it's the ratio of variance can be explained by principal components.The lower the better. (即方差無法被主成分解釋的比例)

```
s_principal3$uniqueness|> sort(decreasing = FALSE)
```

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

Ans:

The Q2 ranked as the last one in communality and uniqueness as above.

Therefore. the Q2 is the less than adequately explained by the first 3 principal components.

---

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

```
s_principal3$loadings |> round(1)
```

```
##
## Loadings:
##     PC1  PC2  PC3
## Q1   0.8 -0.1
## Q2   0.7       0.1
## Q3   0.8       0.1
## Q4   0.6  0.6  0.1
## Q5   0.7      -0.5
## Q6   0.7 -0.1  0.2
## Q7   0.7 -0.3  0.3
## Q8   0.8      -0.3
## Q9   0.7 -0.2  0.2
## Q10  0.7 -0.1 -0.5
## Q11  0.8 -0.3  0.2
## Q12  0.6  0.6  0.1
## Q13  0.7 -0.1  0.1
## Q14  0.8 -0.1  0.2
## Q15  0.7      -0.3
## Q16  0.8 -0.2  0.2
## Q17  0.6  0.7  0.1
## Q18  0.8 -0.1 -0.1
##
##                  PC1   PC2   PC3
## SS loadings    9.480 1.530 1.040
## Proportion Var 0.527 0.085 0.058
## Cumulative Var 0.527 0.612 0.669
```

Ans:

I considered the "similar" indicated the number equal to any of the others after being rounded to the first decimal place. Q4, Q12, Q18 share similar loading between 2 or more components.

---

e. Can you interpret a 'meaning' behind the first principal component from the items that load best upon it? (see the wording of the questions of those items)

Ans:

They're about confidentiality, Accuracy, denial of something, Security of transaction, and Personal Information Security.

Question 3) To improve interpretability of loadings, let's rotate the our principal component axes using the varimax technique to get rotated components (extract and rotate only 3 principal components)

```
principal3.rotate<-df_data |> principal(nfactor=3,rotate="varimax", scores= TRUE )
summary(principal3.rotate)
```

```
##
## Factor analysis with Call: principal(r = df_data, nfactors = 3, rotate = "varimax", scores
= TRUE)
##
## Test of the hypothesis that 3 factors are sufficient.
## The degrees of freedom for the model is 102  and the objective function was  1.28
## The number of observations was  405  with Chi Square =  504.66  with prob <  1.3e-54
##
## The root mean square of the residuals (RMSA) is  0.05
```

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

Ans:

The rotated component (RC) explain the different amount of variance than the corresponding principal components (PCs).

- The rotate One

```
var.pc.r <-  principal3.rotate$Vaccounted
var.pc.r
```

```
##                             RC1       RC3       RC2
## SS loadings           5.6131484 3.4901395 2.9535556
## Proportion Var        0.3118416 0.1938966 0.1640864
## Cumulative Var        0.3118416 0.5057382 0.6698246
## Proportion Explained  0.4655570 0.2894737 0.2449692
## Cumulative Proportion 0.4655570 0.7550308 1.0000000
```

- The original One

```
principal3.none<-df_data |> principal(nfactor=3,rotate="none",scores=TRUE)

var.pc.none <- principal3.none$Vaccounted
var.pc.none
```

```
##                             PC1        PC2        PC3
## SS loadings           9.3109533 1.59633195 1.14955822
## Proportion Var        0.5172752 0.08868511 0.06386435
## Cumulative Var        0.5172752 0.60596029 0.66982464
## Proportion Explained  0.7722546 0.13240049 0.09534487
## Cumulative Proportion 0.7722546 0.90465513 1.00000000
```

b. Together, do the three rotated components explain the same, more, or less cumulative variance as the three principal components combined?

Ans:

After combination, the three rotated components explain less cumulative variance than the none-rotated one.

```
var.pc.r |> apply(1, sum)
```

```
##           SS loadings        Proportion Var        Cumulative Var
##            12.0568434             0.6698246             1.4874044
##  Proportion Explained Cumulative Proportion
##             1.0000000             2.2205878
```

```
var.pc.none |> apply(1,sum)
```

```
##           SS loadings        Proportion Var        Cumulative Var
##            12.0568434             0.6698246             1.7930601
##  Proportion Explained Cumulative Proportion
##             1.0000000             2.6769098
```

c. Looking back at the items that shared similar loadings with multiple principal components (#2d), do those items have more clearly differentiated loadings among rotated components?

Ans:

Yes. In Question 2d, I answered Q4, Q12, Q18 shared similar loadings with multiple principal components.

In these case, those items have more clearly differentiated loadings among rotated component. Especially , Q18 is improved a lot. Shown as below:

```
Load_old <- s_principal3$loadings |> round(2)
Load_old[c(4,12,18),]
```

```
##      PC1   PC2   PC3
## Q4  0.62  0.64  0.11
## Q12 0.63  0.64  0.12
## Q18 0.81 -0.11 -0.07
```

```
Load_roate <- principal3.rotate$loadings |> round(2)
Load_roate[c(4,12,18),]
```

```
##      RC1  RC3  RC2
## Q4  0.22 0.19 0.85
## Q12 0.23 0.19 0.85
## Q18 0.61 0.50 0.23
```

d. Can you now more easily interpret the "meaning" of the 3 rotated components from the items that load best upon each of them? (see the wording of the questions of those items)

Ans:

- RC1 (> .7)

```
sort(principal3.rotate$loadings[,1], decreasing = T) #|> unlist()
```

```
##        Q7       Q11       Q16        Q9       Q14        Q1        Q6        Q3 
## 0.7895344 0.7573493 0.7396241 0.7378148 0.7187578 0.6602758 0.6524225 0.6206018 
##       Q18       Q13        Q2        Q8       Q15       Q10        Q5       Q12 
## 0.6090325 0.5931915 0.5437243 0.3819373 0.3417567 0.2768895 0.2441735 0.2327616 
##        Q4       Q17 
## 0.2182880 0.2054021
```

Ans:

Among the value of measurement item s bigger than 0.7, the Q7, Q11, Q16, Q9, Q14 are about Personal information Security.

- RC2 (> .7)

```
sort(principal3.rotate$loadings[,2], decreasing = T)
```

```
##        Q5       Q10        Q8       Q15       Q18        Q1        Q3       Q13 
## 0.8279850 0.8229206 0.7062018 0.6557490 0.4953450 0.4497592 0.3367919 0.3150514 
##       Q14        Q2       Q11       Q16        Q9        Q6        Q4       Q17 
## 0.3100848 0.2860379 0.2779380 0.2669610 0.2335447 0.1991636 0.1933627 0.1869028 
##       Q12        Q7 
## 0.1861745 0.1031417
```

Ans:

Among the value of measurement item s bigger than 0.7, the Q5, Q10, Q8 are about the safety of transaction information.

- RC3 (> .7)

```
sort(principal3.rotate$loadings[,3], decreasing = T)
```

```
##        Q17        Q12         Q4         Q3         Q8         Q2        Q14 
## 0.87039101 0.85423462 0.85368376 0.31074186 0.30488390 0.28825252 0.28326088 
##        Q13        Q15         Q6        Q18         Q1        Q16         Q5 
## 0.25878712 0.24407206 0.23407080 0.22733033 0.22058261 0.17399181 0.16174750 
##         Q9        Q11        Q10         Q7 
## 0.13766953 0.11843957 0.10209878 0.05598322
```

Ans:

Among the value of measurement item s bigger than 0.7, the Q17, Q12, Q4 are about the protection of denial of transaction.

---

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

```
principal2.rotate <- df_data |> principal(nfactor=2, rotate="varimax", scores=TRUE)

lpc1<-as.data.frame(principal2.rotate$loadings[,1])
colnames(lpc1) <- 'loading_PC1'
arrange(lpc1,desc(loading_PC1))
```

```
##     loading_PC1
## Q11   0.7855784
## Q1    0.7830951
## Q18   0.7616746
## Q16   0.7615661
## Q14   0.7591295
## Q9    0.7451939
## Q7    0.7284256
## Q3    0.6865878
## Q8    0.6684679
## Q13   0.6549937
## Q10   0.6488232
## Q6    0.6487494
## Q5    0.6197912
## Q15   0.6118654
## Q2    0.5960420
## Q12   0.2452587
## Q4    0.2364722
## Q17   0.2211505
```

```
lpc2<-as.data.frame(principal2.rotate$loadings[,2])

colnames(lpc2) <- 'loading_PC2'

arrange(lpc2,desc(loading_PC2))
```

```
##     loading_PC2
## Q17  0.87959208
## Q4   0.86384301
## Q12  0.86234333
## Q8   0.41582056
## Q15  0.34843790
## Q3   0.34013157
## Q2   0.31196986
## Q5   0.30504494
## Q14  0.30354960
## Q18  0.28908208
## Q13  0.28631285
## Q1   0.27140703
## Q10  0.24407384
## Q6   0.23725419
## Q16  0.18721908
## Q9   0.14531919
## Q11  0.13401543
## Q7   0.03797881
```

Ans.

Yes, Compare to (Q3.d), both of PC1 and PC2 are changed.

(ungraded) Looking back at all our results and analyses of this dataset (from this week and previous), how many components (1-3) do you believe we should extract and analyze to understand the security dataset? Feel free to suggest different answers for different purposes.

Ans.

I think 2 components would be proper. Because the third one did not performace better than the noise.