hw14

111078513

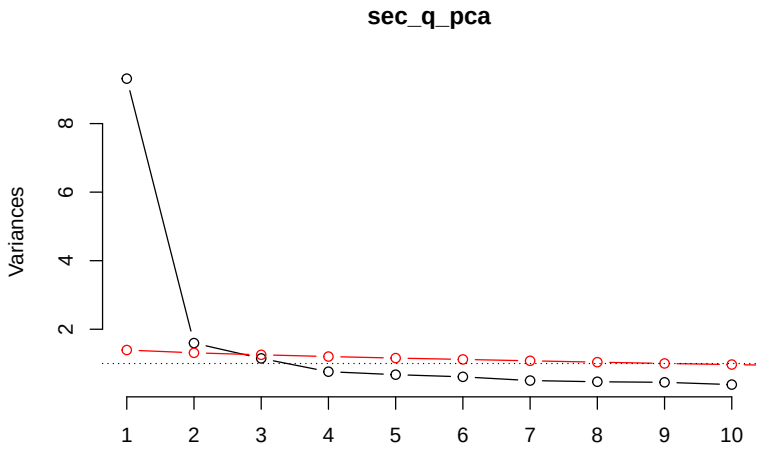
Help By 108078467

Question 1) Earlier, we examined a dataset from a security survey sent to customers of e-commerce websites. However, we only used the eigenvalue > 1 criteria and the screeplot "elbow" rule to find a suitable number of components. Let's perform a parallel analysis as well this week:

```
library(readxl)
sec_q <- data.frame(read_excel("security_questions.xlsx", sheet = "data", col_names = T))
```

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

```
sec_q_pca <- prcomp(sec_q, scale. = TRUE)
sim_noise_ev <- function(n, p) {
  noise <- data.frame(replicate(p, rnorm(n)))
  eigen(cor(noise))$values
}
evalues_noise <- replicate(100, sim_noise_ev(405, 18))
evalues_mean <- apply(evalues_noise, 1, mean)
screeplot(sec_q_pca, type="lines")
lines(evalues_mean, type="b", col = "red")
abline(h=1, lty="dotted")
```



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

Only one PC is significantly higher than noise, after that there is another PC gets as more Variance as noise. Based on the context above, I will retain 2 PCs by doing Parallel Analysis.

Question 2) Earlier, we treated the underlying dimensions of the security dataset as composites and examined their eigenvectors (weights). Now, let's treat them as factors and examine factor loadings (use the principal() method from the psych package)

```
library(psych)
```

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

```
sec_q_principal <- principal(sec_q, nfactor = 3, rotate = "none", scores = TRUE)
sec_q_principal
```

```
## Principal Components Analysis
## Call: principal(r = sec_q, nfactors = 3, rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1   PC2   PC3   h2   u2 com
## Q1  0.82 -0.14  0.00 0.69 0.31 1.1
## Q2  0.67 -0.01  0.09 0.46 0.54 1.0
## Q3  0.77 -0.03  0.09 0.60 0.40 1.0
## Q4  0.62  0.64  0.11 0.81 0.19 2.1
## Q5  0.69 -0.03 -0.54 0.77 0.23 1.9
## Q6  0.68 -0.10  0.21 0.52 0.48 1.2
## Q7  0.66 -0.32  0.32 0.64 0.36 2.0
## Q8  0.79  0.04 -0.34 0.74 0.26 1.4
## Q9  0.72 -0.23  0.20 0.62 0.38 1.4
## Q10 0.69 -0.10 -0.53 0.76 0.24 1.9
## Q11 0.75 -0.26  0.17 0.66 0.34 1.4
## Q12 0.63  0.64  0.12 0.82 0.18 2.1
## Q13 0.71 -0.06  0.08 0.52 0.48 1.0
## Q14 0.81 -0.10  0.16 0.69 0.31 1.1
## Q15 0.70  0.01 -0.33 0.61 0.39 1.4
## Q16 0.76 -0.20  0.18 0.65 0.35 1.3
## Q17 0.62  0.66  0.11 0.83 0.17 2.0
## Q18 0.81 -0.11 -0.07 0.67 0.33 1.1
##
##                        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  0.77 0.13 0.10
## Cumulative Proportion 0.77 0.90 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 3 components are sufficient.
##
```

```
## The root mean square of the residuals (RMSR) is  0.05
##  with the empirical chi square  258.65  with prob <  1.4e-15
##
## Fit based upon off diagonal values = 0.99
```

It seems like all the items belong to PC1.

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

```
sum(sec_q_principal$loadings[,1:3]^2)
```

```
## [1] 12.05684
```

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

```
is_h2_less_than_0.55 <- apply(sec_q_principal$loadings[, 1:3]^2, 1, sum) < 0.55
row.names(sec_q_principal$loadings)[is_h2_less_than_0.55]
```

```
## [1] "Q2"  "Q6"  "Q13"
```

We can notice that the Variance we can capture through PC1 to PC3 of "Q2", "Q6", "Q13" are lower than 0.55. So we can say that these three items are less than adequately explained by the first 3 principal components.

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

The variance captured from PC1 and PC2 in 'Q4', 'Q12', and 'Q17' are similar.

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)

The conclusion of the questions from Q1, Q14, and Q18, which the variance of those items are captured over o.8 from PC1, can be made as how confident users believe in the site securing their transactions information.

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

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

```
sec_q_principal_rot <- principal(sec_q, nfactor = 3, rotate = "varimax", scores = TRUE)
compare_each_PC_and_RC <- t(data.frame(sec_q_principal$values[1:3], sec_q_principal_rot$values[1:3]))
row.names(compare_each_PC_and_RC) <-c("PCs", "RCs")
compare_each_PC_and_RC
```

```
##         [,1]     [,2]     [,3]
## PCs 9.310953 1.596332 1.149558
```

```
## RCs 9.310953 1.596332 1.149558
```

From the table we made, we can notice that each RCs and PCs explain the same amounts of variance.

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

```
compare_PCs_and_RCs <- apply(compare_each_PC_and_RC, 1, sum)
compare_PCs_and_RCs
```

```
##      PCs      RCs
## 12.05684 12.05684
```

From the table we made, we can alse notice that RCs and PCs explain the same amounts of variance combined.

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?

```
sec_q_principal_rot$Structure[c(4, 12, 17), c(1, 3)]
```

```
##           RC1       RC2
## Q4  0.2182880 0.8536838
## Q12 0.2327616 0.8542346
## Q17 0.2054021 0.8703910
```

The items 'Q4', 'Q12', and 'Q17' have more clearly differentiated loadings among rotated components.

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)

```
sec_q_principal_rot_structure <- sec_q_principal_rot$Structure[, c(1, 3, 2)]
sec_q_principal_rot_structure_colname <- colnames(sec_q_principal_rot_structure)
sec_q_principal_rot_structure_rowname <- row.names(sec_q_principal_rot_structure)
```

```
for(i in 1:3){
  print(sec_q_principal_rot_structure_colname[i])
  is_RC_greater_than_0.5 <- sec_q_principal_rot_structure[, i] > 0.5
  print(sec_q_principal_rot_structure_rowname[is_RC_greater_than_0.5])
}
```

```
## [1] "RC1"
##  [1] "Q1"  "Q2"  "Q3"  "Q6"  "Q7"  "Q9"  "Q11" "Q13" "Q14" "Q16" "Q18"
## [1] "RC2"
## [1] "Q4"  "Q12" "Q17"
## [1] "RC3"
## [1] "Q5"  "Q8"  "Q10" "Q15"
```

Yes, we can now more easily interpret the "meaning" according to the significant difference between three rotated components.

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

```
sec_q_principal_rot_2 <- principal(sec_q, nfactor = 2, rotate = "varimax", scores = TRUE)
sec_q_principal_rot_2_structure <- sec_q_principal_rot_2$Structure[, c(1, 2)]
sec_q_principal_rot_2_structure_colname <- colnames(sec_q_principal_rot_2_structure)
sec_q_principal_rot_2_structure_rowname <- row.names(sec_q_principal_rot_2_structure)
```

```
for(i in 1:2){
  print(sec_q_principal_rot_2_structure_colname[i])
  is_RC_greater_than_0.5 <- sec_q_principal_rot_2_structure[, i] > 0.5
  print(sec_q_principal_rot_2_structure_rowname[is_RC_greater_than_0.5])
}
```

```
## [1] "RC1"
##  [1] "Q1"  "Q2"  "Q3"  "Q5"  "Q6"  "Q7"  "Q8"  "Q9"  "Q10" "Q11" "Q13" "Q14"
## [13] "Q15" "Q16" "Q18"
## [1] "RC2"
## [1] "Q4"  "Q12" "Q17"
```

We can notice that although RC2 can still explain 'Q4', 'Q12', 'Q17, the most, the RC3 before is combined into RC1. So yes, the meaning of our rotated components change.