BACS HW14

109090046 assisted by 109090035 109090023

2023-05-17

Let's reconsider the security questionnaire from last week, where consumers were asked security related questions about one of the e-commerce websites they had recently used.

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:

```
data <- read_excel("D:/下載/security_questions.xlsx", sheet= 2)
```

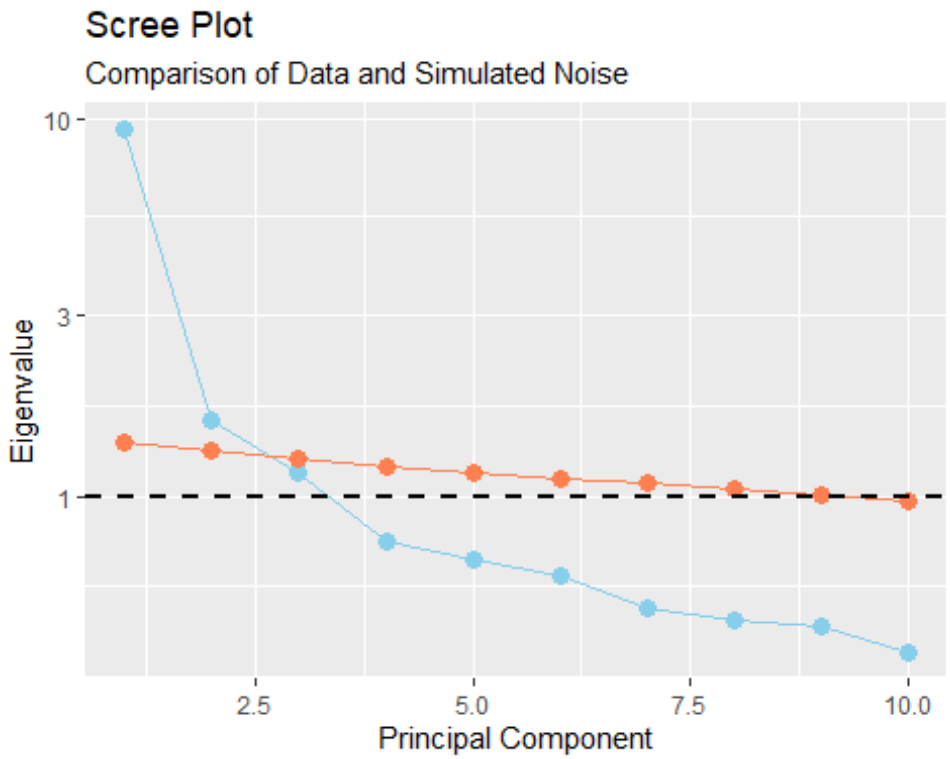
a. Show a single visualization with scree plot of data, scree plot of simulated noise (use average eigenvalues of ≥ 100 noise samples), and a horizontal line showing the eigenvalue = 1 cutoff.

```
# Simulated noise eigenvalues
set.seed(42)
sim_noise <- function(n, p)
{
  noise <- data.frame(replicate(p, rnorm(n)))
  eigen(cor(noise))$values
}
evalues_noise <- replicate(100, sim_noise(nrow(data), ncol(data)))
evalues_mean <- apply(evalues_noise, 1, mean)
pca <- prcomp(data, scale. = TRUE)

# Create a data frame for plotting
plot_data <- data.frame(
  Component = 1:length(pca$sdev),
  Variance = pca$sdev^2,
  Noise = evalues_mean[1:length(pca$sdev)]
)

# Filter the data for the first 10 components
plot_data <- plot_data[1:10, ]

# Create the plot
ggplot(plot_data, aes(x = Component)) +
  geom_point(aes(y = Variance), color = "skyblue", size=3) +
  geom_point(aes(y = Noise), color = "coral", size=3) +
  geom_line(aes(y = Variance), color = "skyblue") +
  geom_line(aes(y = Noise), color = "coral") +
  geom_hline(yintercept = 1, linetype = "dashed", linewidth=0.8) +
  scale_y_continuous(trans = 'log10') +
  labs(x = "Principal Component",
       y = "Eigenvalue",
       title = "Scree Plot",
       subtitle = "Comparison of Data and Simulated Noise")
```



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

```
retain_dims <- sum(plot_data$Variance > plot_data$Noise)

retain_dims

## [1] 2
```

There are only first two dimensions that higher than the noise eigenvalues, so we retain those dimensions.

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)

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

```
principal <- principal(data, nfactor=10, rotate="none", scores=TRUE)
pc1 <- principal$loadings[,"PC1"]
pc2 <- principal$loadings[,"PC2"]
pc3 <- principal$loadings[,"PC3"]
first3pc <- round(cbind(pc1, pc2, pc3), digits=3)

first3pc

##       pc1    pc2    pc3
## Q1  0.817 -0.139 -0.002
## Q2  0.673 -0.014  0.089
## Q3  0.766 -0.033  0.090
## Q4  0.623  0.643  0.108
## Q5  0.690 -0.031 -0.542
```

```
## Q6  0.683 -0.105  0.207
## Q7  0.657 -0.318  0.324
## Q8  0.786  0.042 -0.343
## Q9  0.723 -0.232  0.204
## Q10 0.686 -0.099 -0.533
## Q11 0.753 -0.261  0.173
## Q12 0.630  0.638  0.122
## Q13 0.712 -0.065  0.084
## Q14 0.811 -0.100  0.157
## Q15 0.704  0.011 -0.333
## Q16 0.758 -0.203  0.183
## Q17 0.618  0.664  0.110
## Q18 0.807 -0.114 -0.065
```

The loadings of principal components represent the correlations between the original variables and the component. High absolute values (either positive or negative) indicate that the original variable contributes significantly to that component. *To determine which component an item best belongs to, we can consider the component where the item has the highest absolute loading.*

- **PC1:** Q1, Q2, Q3, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q13, Q14, Q15, Q16, Q18**
- **PC2:** Q4, Q12, Q17
- **PC3:** None

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

```
summary(pca)$importance[2, c(1:3)]
```

```
##     PC1     PC2     PC3
## 0.51728 0.08869 0.06386
```

```
sum(summary(pca)$importance[2, c(1:3)])
```

```
## [1] 0.66983
```

Sum up the proportions of variance, first 3 PCs captures 66.893%

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

```
commonalities <- rowSums(first3pc^2)
uniqueness <- 1 - commonalities

uniqueness
```

```
##       Q1       Q2       Q3       Q4       Q5       Q6       Q7       Q8
## 0.313186 0.538954 0.404055 0.186758 0.229175 0.479637 0.362251 0.262791
##       Q9      Q10      Q11      Q12      Q13      Q14      Q15      Q16
## 0.381831 0.235514 0.334941 0.181172 0.481775 0.307630 0.393374 0.350738
##      Q17      Q18
## 0.165080 0.331530
```

Q2 with highest uniqueness of 0.538954 is less adequately explained by the first three components.

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

```
# function for evaluate similarity
evaluate_loadings <- function(df, range) {
  return(
    (
```

```
      abs(df[range, 1] - df[range, 2])<0.1 |
      abs(df[range, 2] - df[range, 3])<0.1 |
      abs(df[range, 1] - df[range, 3])<0.1
    ) &
    (
      df[range, 1] < 0.7 &
      df[range, 2] < 0.7 &
      df[range, 3] < 0.7
    )
  )
}

evaluate_loadings(principal$loading, 1:ncol(data))
```

```
##    Q1    Q2    Q3    Q4    Q5    Q6    Q7    Q8    Q9   Q10   Q11   Q12   Q13
## FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   Q14   Q15   Q16   Q17   Q18
## FALSE FALSE FALSE  TRUE FALSE
```

If the difference between each PCs smaller than 0.1 and the PCs are smaller than 0.7, it will return True.

Q4, Q12 and Q17 share similar loadings 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:

The items with the highest loadings on PC1 are Q1, Q3, Q8, Q14, and Q18. If these items share a common theme or concept, that could be interpreted as the 'meaning' of the first component.

All these questions are related to a specific aspect of information and accuracy, so I think the first component could be interpreted as representing that aspect.

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)?

```
rc <- principal(data, nfactors = 3, rotate = "varimax", scores = TRUE)$loadings
rc
```

```
##
## Loadings:
##     RC1   RC3   RC2
## Q1  0.660 0.450 0.221
## Q2  0.544 0.286 0.288
## Q3  0.621 0.337 0.311
## Q4  0.218 0.193 0.854
## Q5  0.244 0.828 0.162
## Q6  0.652 0.199 0.234
## Q7  0.790 0.103
## Q8  0.382 0.706 0.305
```

```
## Q9  0.738 0.234 0.138
## Q10 0.277 0.823 0.102
## Q11 0.757 0.278 0.118
## Q12 0.233 0.186 0.854
## Q13 0.593 0.315 0.259
## Q14 0.719 0.310 0.283
## Q15 0.342 0.656 0.244
## Q16 0.740 0.267 0.174
## Q17 0.205 0.187 0.870
## Q18 0.609 0.495 0.227
##
##                  RC1   RC3   RC2
## SS loadings    5.613 3.490 2.954
## Proportion Var 0.312 0.194 0.164
## Cumulative Var 0.312 0.506 0.670
```

Looking at the proportion variance, we can see RC1is 30% and PC1is 51%, RC2 is 19% and PC2 is 8%. RC3 is 16% and PC3 is 6%. The ratios have large difference, so the are not the same.

Each rotated component (RC) should explain the same amount of variance as the corresponding unrotated principal component (PC). However, the pattern of loadings across variables will be different between the PCs and RCs.

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

ANS: The total amount of variance explained by a given number of components (whether rotated or not) is the same. Rotation does not change the total variance explained, it only changes the distribution of that variance across the components to make the solution more interpretable.

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?

```
rc[c(4,12,17), 1:3]
```

```
##           RC1       RC3       RC2
## Q4  0.2182880 0.1933627 0.8536838
## Q12 0.2327616 0.1861745 0.8542346
## Q17 0.2054021 0.1869028 0.8703910
```

Because the RC2 loadings are over 0.8, I think they have more clearly differentiated.

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)

```
rc[rc[, 1] > 0.7, 1]
```

```
##        Q7        Q9       Q11       Q14       Q16
## 0.7895344 0.7378148 0.7573493 0.7187578 0.7396241
```

For RC1, those questions are all about "personal information protection".

```
rc[rc[, 2] > 0.7, 2]
```

```
##        Q5        Q8       Q10
## 0.8279850 0.7062018 0.8229206
```

For RC2, those questions are all about "transaction processing".

```
rc[rc[, 3] > 0.7, 3]
```

```
##        Q4       Q12       Q17
## 0.8536838 0.8542346 0.8703910
```

For RC3, thosw questions are about "providing evidence to protect against its denial".

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

```
reduced_rc <- principal(data, nfactors = 2, rotate = "varimax", scores = TRUE)
reduced_rc$loadings[,1][reduced_rc$loadings[,1] > 0.7]
```

```
##        Q1        Q7        Q9       Q11       Q14       Q16       Q18
## 0.7830951 0.7284256 0.7451939 0.7855784 0.7591295 0.7615661 0.7616746
```

Yes, when we reduced the number of extracted and rotated components to 2, the number of questions belong to RC1 increase. Also, the meanings change a little. Those question are about personal information and security.

(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.

I think there should be at least three because when we set nFactor to 2, the RC1 are not only about personal information, but also about confidentiality of the transaction.