BACS HW (Week15)

106070038

2021-06-06

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 send to customers of e-commerce websites. However, we only eigenvalue > 1 criteria and the screeplot to find a suitable number of components. Let's perform a parallel analysis as well this week:

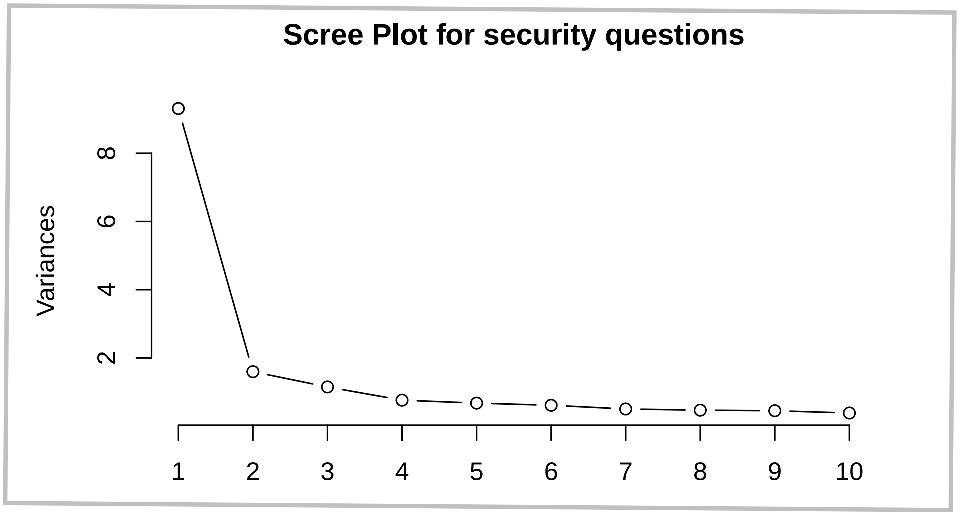
```
# install.packages('readxl')
library('readxl')
security_questions <- read_excel("security_questions.xlsx", sheet = "data")
head(security_questions)
```

```
## # A tibble: 6 x 18
##      Q1    Q2    Q3    Q4    Q5    Q6    Q7    Q8    Q9   Q10   Q11   Q12   Q13
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     7     5     5     7     7     4     4     7     5     7     5     7     5
## 2     5     5     6     6     6     5     5     7     5     6     6     6     6
## 3     6     6     6     6     7     6     6     6     5     7     6     6     5
## 4     5     5     5     5     5     5     5     5     5     5     5     5     4
## 5     7     7     7     7     7     4     5     7     6     7     6     7     6
## 6     6     5     4     5     4     4     4     5     6     2     5     5     5
## # ... with 5 more variables: Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>,
## #   Q18 <dbl>
```

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

- **visualization with scree plot of data**

```
pca <- prcomp(formula = ~.,
              data = security_questions,
              scale = TRUE)
plot(pca, type="line",
     main="Scree Plot for security questions")
```

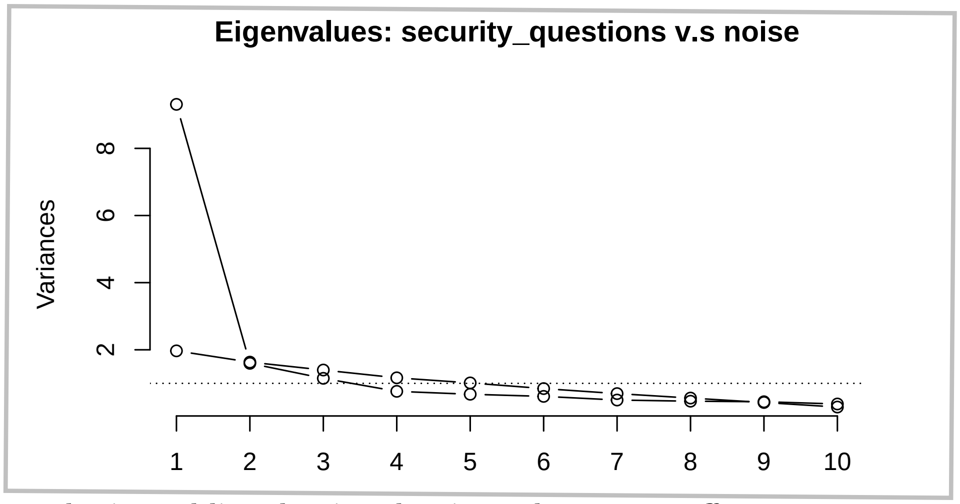


- **scree plot of simulated noise**

```
set.seed(1)

## Function to run a PCA on n   p dataframe of random values
sim_noise_ev <- function(n, p) {
  noise <- data.frame(replicate(p, rnorm(n)))
  return( eigen(cor(noise))$values )
}

## Repeat this k times
evalues_noise <- replicate(100, sim_noise_ev(33, 10))

## Average each of the noise eigenvalues ev over k to produce ev
evalues_mean <- apply(evalues_noise, 1, mean)
dec_pca <- prcomp(security_questions, scale. = TRUE)
screeplot(dec_pca, type="lines", main="Eigenvalues: security_questions v.s noise")
lines(evalues_mean, type="blue")
```

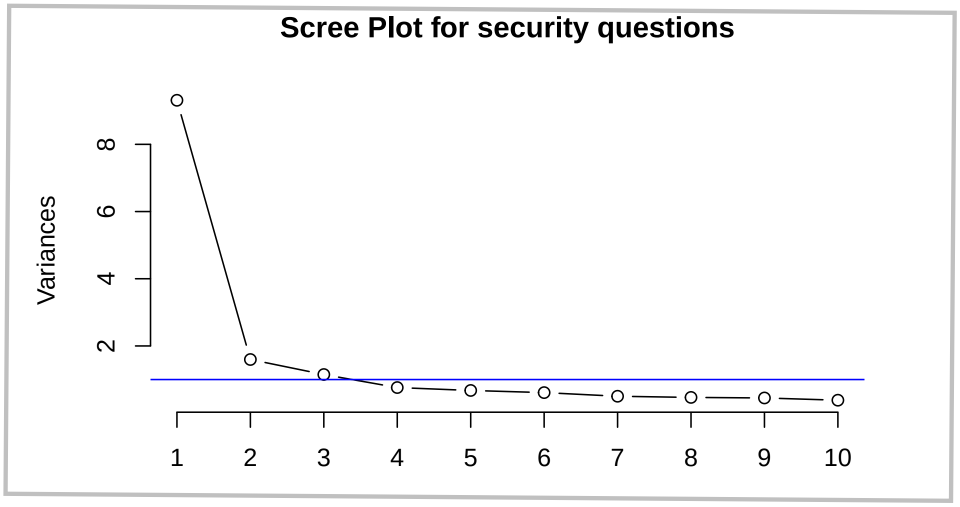
```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): plot type 'blue' will be
## truncated to first character
```

```
abline(h=1, lty="dotted")
```



- **a horizontal line showing the eigenvalue = 1 cutoff**

```
plot(pca, type="line",
     main="Scree Plot for security questions")
abline(h=1, col="blue") # Kaiser eigenvalue-greater-than-one rule
```



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

- **Parallel Analysis**: Parallel analyis is an alternative technique that compares the scree of factors of the observed data with that of a random data matrix of the same size as the original.

- Answer: Based on (a), I will retain 2 dimensions.

```
# install.packages("psych")
# library(psych)
# fa.parallel(security_questions,n.obs=NULL,fm="minres",fa="both",nfactors=1,
# main="Parallel Analysis Scree Plots for security_questions",
# n.iter=20,error.bars=FALSE,se.bars=FALSE,SMC=FALSE,ylabel=NULL,show.legend=TRUE,
# sim=TRUE,quant=.95,cor="cor",use="pairwise",plot=TRUE,correct=.5)
```

Question 2 Earlier, we examined the eigenvectors of the security dataset. Now, let's examine factor loadings

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

```
# install.packages("psych")
library(psych)
dec_pca3_orig <- principal(security_questions, nfactor=3, rotate="none", scores=TRUE)
dec_pca3_orig$loadings
```

```
##
## Loadings:
##     PC1    PC2    PC3
## Q1   0.817 -0.139
## Q2   0.673
## Q3   0.766
## Q4   0.623  0.643  0.108
## Q5   0.690        -0.542
## Q6   0.683 -0.105  0.207
## Q7   0.657 -0.318  0.324
## Q8   0.786        -0.343
## Q9   0.723 -0.232  0.204
## Q10  0.686        -0.533
## Q11  0.753 -0.261  0.173
## Q12  0.630  0.638  0.122
## Q13  0.712
## Q14  0.811         0.157
## Q15  0.704        -0.333
## Q16  0.758 -0.203  0.183
## Q17  0.618  0.664  0.110
## Q18  0.807 -0.114
##
##                  PC1   PC2   PC3
## SS loadings    9.311 1.596 1.150
## Proportion Var 0.517 0.089 0.064
## Cumulative Var 0.517 0.606 0.670
```

- Answer:
  - Loadings, which include magnitude and direction are easier to interpret than eigenvectors. lambda > 0.70 is considered a good loading, more than half of item variance explained by PC.
  - As a result, PC1 belongs to Q1, Q14, Q18.

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

```
sum(dec_pca3_orig$loadings[,"PC1"]^2) + sum(dec_pca3_orig$loadings[,"PC2"]^2) + sum(dec_pca3_orig$loadi
```

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

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

- Commonality: variance of X100m explained by both principal components
- Uniqueness: Unexplained variance of X100m. u2 = 1 - Communality

- Answer: **Q17**

```
dec_pca3_orig[3]
```

```
## $n.obs
## [1] 405
```

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

- Answer:
  - Q4 share similar loadings between PC1 and PC2.
  - Q5 share similar loadings between PC1 and PC3.
  - Q12 share similar loadings between PC1 and PC2.
  - Q17 share similar loadings between PC1 and PC2.

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

- Some infomation about site and positive meaning.

Question 3 To improve interpretability of loadings, let's rotate the our principal component axes 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)?

- Answer: All are **different**.

```
dec_pca3_original <- principal(security_questions, nfactor=3, rotate="none", scores=TRUE)
dec_pca3_original$loadings
```

```
##
## Loadings:
##     PC1    PC2    PC3
## Q1   0.817 -0.139
## Q2   0.673
## Q3   0.766
## Q4   0.623  0.643  0.108
## Q5   0.690        -0.542
## Q6   0.683 -0.105  0.207
## Q7   0.657 -0.318  0.324
## Q8   0.786        -0.343
## Q9   0.723 -0.232  0.204
## Q10  0.686        -0.533
## Q11  0.753 -0.261  0.173
## Q12  0.630  0.638  0.122
## Q13  0.712
## Q14  0.811         0.157
## Q15  0.704        -0.333
## Q16  0.758 -0.203  0.183
## Q17  0.618  0.664  0.110
## Q18  0.807 -0.114
##
##                  PC1   PC2   PC3
## SS loadings    9.311 1.596 1.150
## Proportion Var 0.517 0.089 0.064
## Cumulative Var 0.517 0.606 0.670
```

```
dec_pca3_rotate <- principal(security_questions, nfactor=3, rotate="varimax", scores=TRUE)
dec_pca3_rotate$loadings
```

```
##
## 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
```

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

- The **same**.

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?

- Answer:
  - Q4 loadings between PC1 and PC2. -> same
  - Q5 loadings between PC1 and PC3. -> smaller
  - Q12 loadings between PC1 and PC2. -> bigger
  - Q17 loadings between PC1 and PC2. -> bigger

d. Can you now 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)

- PC1: some negative word, ex. never, remove, prevent.
- PC2: about "I", "my" and "mine".
- PC3: promise something, ex. make sure and provide me something to protect.

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

- Yes, it will definitely change.

```
dec_pca2_rotate <- principal(security_questions, nfactor=2, rotate="varimax", scores=TRUE)
dec_pca2_rotate$loadings
```

```
##
## Loadings:
##     RC1   RC2
## Q1  0.783 0.271
## Q2  0.596 0.312
## Q3  0.687 0.340
## Q4  0.236 0.864
## Q5  0.620 0.305
## Q6  0.649 0.237
## Q7  0.728
## Q8  0.668 0.416
## Q9  0.745 0.145
## Q10 0.649 0.244
## Q11 0.786 0.134
## Q12 0.245 0.862
## Q13 0.655 0.286
## Q14 0.759 0.304
## Q15 0.612 0.348
## Q16 0.762 0.187
## Q17 0.221 0.880
## Q18 0.762 0.289
##
##                  RC1   RC2
## SS loadings    7.521 3.387
## Proportion Var 0.418 0.188
## Cumulative Var 0.418 0.606
```

(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.

- Answer: **one**, the loading gap between PC1 and PC2 is quite large no matter which approach.