HW13

110077443

5/10/2022

Credit: 109077424

Please note that all code in this document is presented in a grey box and the output reflected below each box

- The below code allows lengthy lines of comments to display neatly within the grey box (wrapping it)

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

1) Let's revisit the issue of multicollinearity of main effects and try to apply principal components to it

```
# Importing data
auto <- read.table("auto-data.txt", header = FALSE, na.strings = "?")

# Renaming variables
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower",
    "weight", "acceleration", "model_year", "origin", "car_name")

# log-transform
cars_log <- with(auto, data.frame(log(mpg), log(cylinders), log(displacement),
    log(horsepower), log(weight), log(acceleration), model_year,
    origin))

# Removing rows with missing values
cars_log <- na.omit(cars_log)
```

a) Let's analyze the principal components of the four collinear variables

i) Create a new data.frame of the four log-transformed variables with high multicollinearity

```
# Creating new data frame
correlated_var <- with(cars_log, data.frame(log.cylinders., log.displacement.,
    log.horsepower., log.weight.))

# Checking correlation in table
```

```r
cor_correlated <- cor(correlated_var)
require(knitr)  # For creating tables with kable function
kable(cor_correlated, caption = "Correlated Variables", align = "c")
```

Table 1: Correlated Variables

	log.cylinders.	log.displacement.	log.horsepower.	log.weight.
log.cylinders.	1.0000000	0.9469109	0.8265831	0.8833950
log.displacement.	0.9469109	1.0000000	0.8721494	0.9428497
log.horsepower.	0.8265831	0.8721494	1.0000000	0.8739558
log.weight.	0.8833950	0.9428497	0.8739558	1.0000000

- All variables are highly correlated as seen in the table

ii) How much variance of the four variables is explained by their first principal component?

```r
# Computing eigenvalues
eigen <- eigen(cor_correlated)
eigen$values[1]/sum(eigen$values)  # 91.86%
```

```
ANSWER > [1] 0.9185647
```

```r
# Double checking values using prcomp function
pca_cvar <- prcomp(correlated_var, scale. = TRUE)
summary(pca_cvar)
```

```
> Importance of components:
>                           PC1     PC2     PC3     PC4
> Standard deviation     1.9168 0.43316 0.32238 0.18489
> Proportion of Variance 0.9186 0.04691 0.02598 0.00855
> Cumulative Proportion  0.9186 0.96547 0.99145 1.00000
```

ANSWER ##

- The first principal component explains **91.86%** of the total variation in the dataset

iii) Looking at the values and valence (positiveness/negativeness) of the first principal component's eigenvector, what would you call the information captured by this component?

```r
eigen$vectors[, 1]  # Print PC1 values only
```

```
ANSWER > [1] -0.4979145 -0.5122968 -0.4856159 -0.5037960
```

```r
pca_cvar  # Checking all principal components
```

```
> Standard deviations (1, .., p=4):
> [1] 1.9168356 0.4331601 0.3223785 0.1848936
>
> Rotation (n x k) = (4 x 4):
>                          PC1         PC2         PC3        PC4
> log.cylinders.    -0.4979145 -0.53580374  0.52633608  0.4335503
> log.displacement. -0.5122968 -0.25665246 -0.07354139 -0.8162556
> log.horsepower.   -0.4856159  0.80424467  0.34193949  0.0210980
> log.weight.       -0.5037960  0.01530917 -0.77500928  0.3812031
```

ANSWER ##

- The slope of PC1 relative to the original dimensions is shown here as the individual weights of the eigenvector
- The vector values represent the relationship of PC1 compared to the original items
- PC1 is a vector in 4 dimensions in this case, representing the most orthogonal variance / uncorrelated variance

b) Let's revisit our regression analysis on cars_log:

i) Store the scores of the first principal component as a new column of cars_log

```
# Storing scores
pca_scores <- pca_cvar$x

# Adding column with stored scores of PC1
cars_log$pca1_scores <- pca_scores[, "PC1"]
head(cars_log)  # Checking column was added correctly
```

```
##   log.mpg. log.cylinders. log.displacement. log.horsepower. log.weight.
## 1 2.890372       2.079442          5.726848        4.867534    8.161660
## 2 2.708050       2.079442          5.857933        5.105945    8.214194
## 3 2.890372       2.079442          5.762051        5.010635    8.142063
## 4 2.772589       2.079442          5.717028        5.010635    8.141190
## 5 2.833213       2.079442          5.710427        4.941642    8.145840
## 6 2.708050       2.079442          6.061457        5.288267    8.375860
##   log.acceleration. model_year origin pca1_scores
## 1          2.484907         70      1   -2.036645
## 2          2.442347         70      1   -2.593998
## 3          2.397895         70      1   -2.237767
## 4          2.484907         70      1   -2.192902
## 5          2.351375         70      1   -2.097313
## 6          2.302585         70      1   -3.337215
```

ii) Regress mpg over the column with PC1 scores

```
regr_pc1 <- (lm(data = cars_log, log.mpg. ~ pca1_scores + log.acceleration. +
    model_year + factor(origin)))
summary(regr_pc1)
```

```
>
> Call:
> lm(formula = log.mpg. ~ pca1_scores + log.acceleration. + model_year +
>     factor(origin), data = cars_log)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -0.51137 -0.06050 -0.00183  0.06322  0.46792
>
> Coefficients:
>                    Estimate Std. Error t value Pr(>|t|)
> (Intercept)        1.398114   0.166554   8.394 8.99e-16 ***
> pca1_scores        0.145663   0.005057  28.804  < 2e-16 ***
> log.acceleration. -0.191482   0.041722  -4.589 6.02e-06 ***
> model_year         0.029180   0.001810  16.122  < 2e-16 ***
> factor(origin)2    0.008272   0.019636   0.421    0.674
> factor(origin)3    0.019687   0.019395   1.015    0.311
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> Residual standard error: 0.1199 on 386 degrees of freedom
> Multiple R-squared:  0.8772,  Adjusted R-squared:  0.8756
> F-statistic: 551.6 on 5 and 386 DF,  p-value: < 2.2e-16
```

iii) running regression again over the same independent variables, but this time with everything standardized. How important is this new column relative to other columns?

```r
# Standardizing cars_log data set
cars_log_std <- as.data.frame(scale(cars_log, center = TRUE,
    scale = TRUE))

# Regression including pca1_scores
summary(lm(data = cars_log_std, log.mpg. ~ pca1_scores + log.acceleration. +
    model_year + factor(origin)))
```

```
>
> Call:
> lm(formula = log.mpg. ~ pca1_scores + log.acceleration. + model_year +
>     factor(origin), data = cars_log_std)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -1.50385 -0.17791 -0.00538  0.18591  1.37608
>
> Coefficients:
>                                 Estimate Std. Error t value Pr(>|t|)
> (Intercept)                     -0.01589    0.02563  -0.620    0.536
> pca1_scores                      0.82112    0.02851  28.804  < 2e-16 ***
> log.acceleration.               -0.10190    0.02220  -4.589 6.02e-06 ***
> model_year                       0.31611    0.01961  16.122  < 2e-16 ***
> factor(origin)0.525710525810929  0.02433    0.05775   0.421    0.674
> factor(origin)1.76714743013553   0.05790    0.05704   1.015    0.311
```

```
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> Residual standard error: 0.3526 on 386 degrees of freedom
> Multiple R-squared:  0.8772,  Adjusted R-squared:  0.8756
> F-statistic: 551.6 on 5 and 386 DF,  p-value: < 2.2e-16
```

ANSWER ##

- The pca_scores column is **very significant** as the p-value is near 0. Thus, we can **reject the null hypothesis** that pca_scores has no effect on mpg.

2) Let's analyze the principal components of the eighteen items from the data file *security_questions.xlsx*

```r
# Importing data
require(readxl)  # Used for read_excel function to import 'xls' and 'xlsx' files
sec <- read_excel("security_questions.xlsx", sheet = "data")
```

a) How much variance did each extracted factor explain?

```r
pca_sec <- prcomp(sec, scale. = TRUE)
var_explained = pca_sec$sdev^2/sum(pca_sec$sdev^2)
var_explained  # Print
```

```
>  [1] 0.51727518 0.08868511 0.06386435 0.04233199 0.03750784 0.03398131
>  [7] 0.02794364 0.02601549 0.02510951 0.02139980 0.01971565 0.01673928
> [13] 0.01623763 0.01456354 0.01303216 0.01280357 0.01159706 0.01119690
```

```r
# Double checking values match
summary(pca_sec)  # Print - values match
```

```
> Importance of components:
>                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
> Standard deviation     3.0514 1.26346 1.07217 0.87291 0.82167 0.78209 0.70921
> Proportion of Variance 0.5173 0.08869 0.06386 0.04233 0.03751 0.03398 0.02794
> Cumulative Proportion  0.5173 0.60596 0.66982 0.71216 0.74966 0.78365 0.81159
>                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
> Standard deviation     0.68431 0.67229 0.6206 0.59572 0.54891 0.54063 0.51200
> Proportion of Variance 0.02602 0.02511 0.0214 0.01972 0.01674 0.01624 0.01456
> Cumulative Proportion  0.83760 0.86271 0.8841 0.90383 0.92057 0.93681 0.95137
>                           PC15   PC16   PC17   PC18
> Standard deviation     0.48433 0.4801 0.4569 0.4489
> Proportion of Variance 0.01303 0.0128 0.0116 0.0112
> Cumulative Proportion  0.96440 0.9772 0.9888 1.0000
```

b) How many dimensions would you retain according to the two criteria (Eigenvalue >= 1 and Scree Plot)?

```
# Eigenvalue >= 1 Criteria
eigen_sec <- eigen(cor(sec))
eigen_sec$values  # Print
```

```
##  [1] 9.3109533 1.5963320 1.1495582 0.7619759 0.6751412 0.6116636 0.5029855
##  [8] 0.4682788 0.4519711 0.3851964 0.3548816 0.3013071 0.2922773 0.2621437
## [15] 0.2345788 0.2304642 0.2087471 0.2015441
```

```
eigen_sec$values >= 1  # boolean results
```

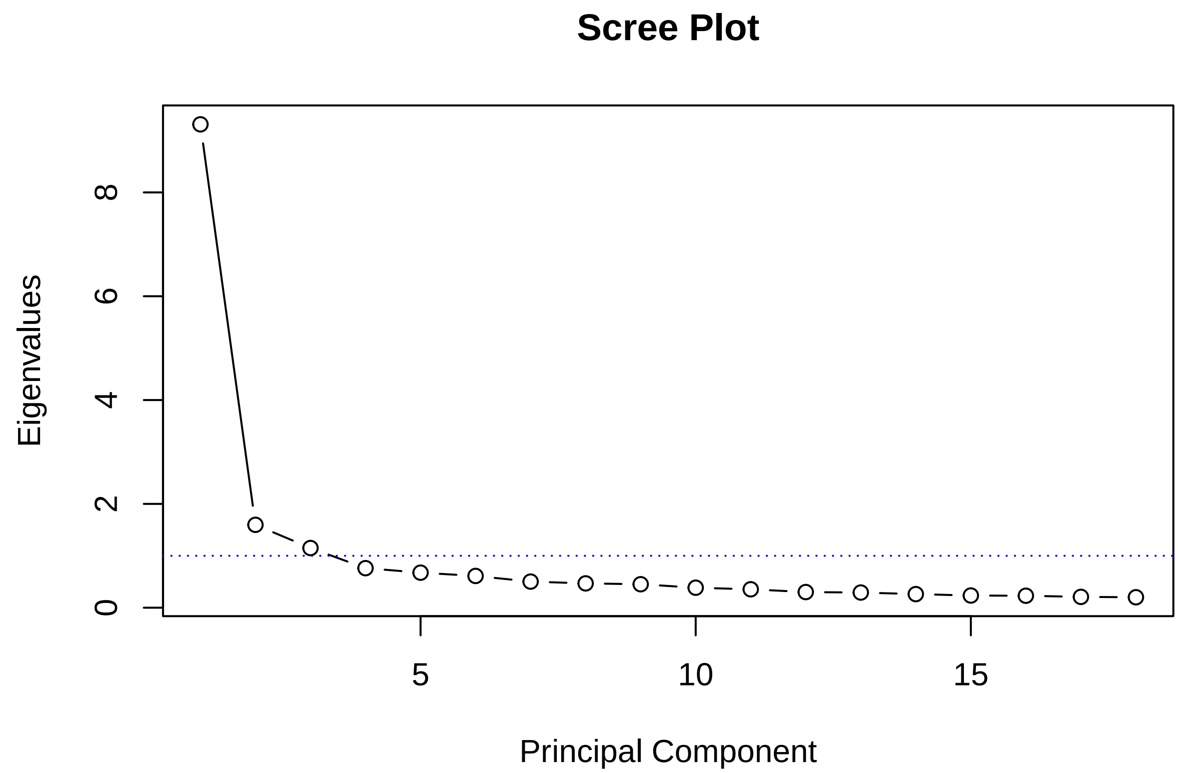
```
##  [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE
```

ANSWER ##

- Based on **Eigenvalue >= 1 Criteria**, we would choose the first three principal components

```
# Screeplot Criteria

# Creating Scree Plot with an eiganvalue = 1 threshold
plot(eigen_sec$values, type = "b", xlab = "Principal Component",
    ylab = "Eigenvalues", main = "Scree Plot")
abline(h = 1, col = "blue", lty = "dotted")  #eigenvalue = 1
```

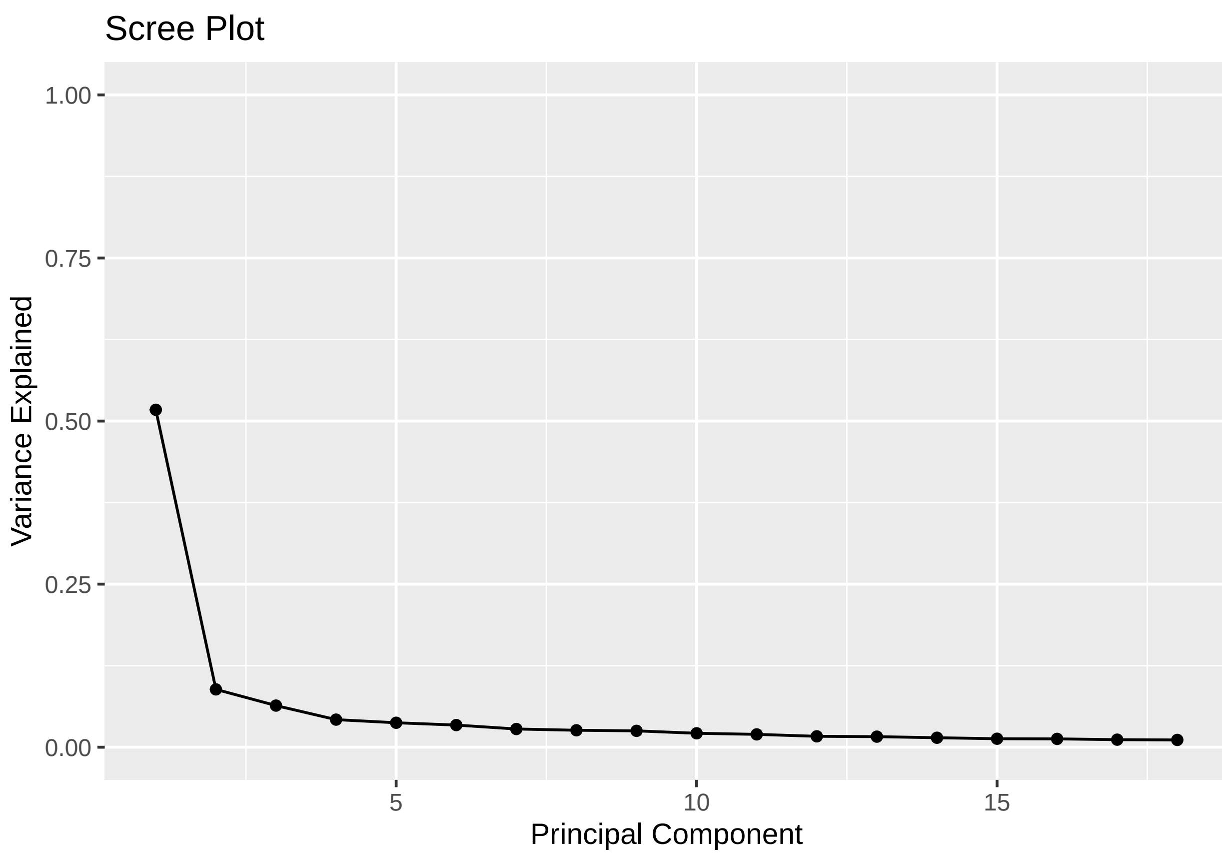


ANSWER ##

- Based on **Screeplot Criteria**, we see that the elbow falls on the 2nd dimension. This indicates the biggest gap in variance explained. Since we only use dimensions that lie above the elbow, we would only retain the **1st dimension**.

```
# Creating additional Scree Plot with variance explained
require(ggplot2)  # Used for visualization

qplot(c(1:18), var_explained) + geom_line() + xlab("Principal Component") +
    ylab("Variance Explained") + ggtitle("Scree Plot") + ylim(0,
    1)
```



- The above plot is an additional feature to show the principal components in relation to the variance explained. As you can see, the scree plot looks the same.

c) (ungraded) Can you interpret what any of the principal components mean?

```r
# Looking at the values of the first three PC's eigenvector
pca_sec$rotation[, 1:3]
```

```
##            PC1          PC2          PC3
## Q1  -0.2677422  0.110341691 -0.001973491
## Q2  -0.2204272  0.010886972  0.083171536
## Q3  -0.2508767  0.025878543  0.083648794
## Q4  -0.2042919 -0.508981768  0.100759585
## Q5  -0.2261544  0.024745268 -0.505845415
## Q6  -0.2237681  0.082805088  0.193281966
## Q7  -0.2151891  0.251398450  0.302354487
## Q8  -0.2576225 -0.033526840 -0.320109219
## Q9  -0.2369512  0.183342667  0.189853454
## Q10 -0.2248660  0.078103267 -0.496820932
## Q11 -0.2467645  0.206580870  0.160903091
## Q12 -0.2065785 -0.504591429  0.113342400
## Q13 -0.2333066  0.051159791  0.078658760
## Q14 -0.2659342  0.078910404  0.146232765
```

```
## Q15 -0.2307289 -0.008373326 -0.310161141
## Q16 -0.2482681  0.160524168  0.170839887
## Q17 -0.2023781 -0.525747030  0.102652280
## Q18 -0.2643810  0.089915229 -0.060800871
```

ANSWER ##

- The more heavily an original column weights on a PC, the more related it is to the PC.
- We can see high values in PC2 as highlighted in green.
- This means that these questions are heavily favored for PC2.
- Values are consistent for PC1, so perhaps all questions should be considered.
- These are some underlying dimensions of people's perception of online security that effectively capture the variance of the eighteen questions

3) Let's simulate how principal components behave interactively

```r
# Running
# devtools::install_github('soumyaray/compstatslib')
require(compstatslib)
# interactive_pca() # Using for creating plot
require(knitr)
```

a) Creating an oval shaped scatter plot of points that stretches in two directions

b) Creating a scatterplot whose principal component vectors do NOT seem to match the major directions of variance

- Plots are displayed on the next page

```r
knitr::include_graphics("Rplot_pca1.pdf")  # Importing plot for (a)
```

```r
knitr::include_graphics("Rplot_pca2.pdf")  # Importing plot for (b)
```

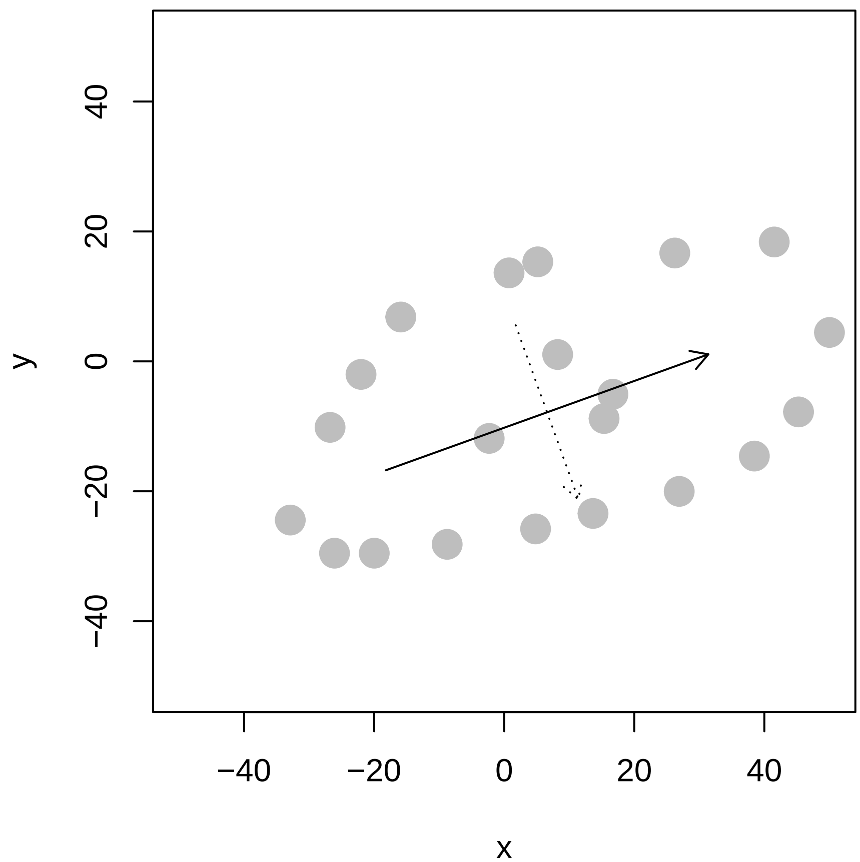


Figure 1: a) PCA Oval Shaped Scatterplot

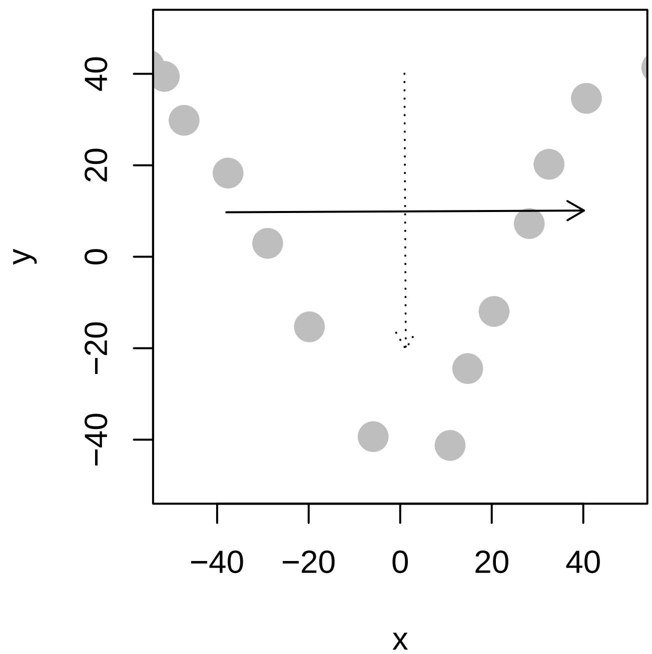


Figure 2: b) PCV Not Matching Major