HW_13

110078509

20220514

Question 1

Note: Yellow Part is Answers or Important details.

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

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

```
engine <-  data.frame(log.mpg. = cars_log$log.mpg.,
                      log.cylinders. =  cars_log$log.cylinders.,
                      log.displacement. =cars_log$log.displacement.,
                      log.horsepower. = cars_log$log.horsepower.)

head(engine,3)

--------------------------------------------------------------------------------------------

##   log.mpg. log.cylinders. log.displacement. log.horsepower.
## 1 2.890372       2.079442          5.726848        4.867534
## 2 2.708050       2.079442          5.857933        5.105945
## 3 2.890372       2.079442          5.762051        5.010635
```

---

aii. How much variance of the four variables is explained by their first principal component?

(a summary of the prcomp() shows it, but try computing this from the eigenvalues alone)

```
var <- eigen(cor(engine))$values
denominator  <- sum (var)
result_manual <- var[1]/denominator
result_manual

--------------------------------------------------------------------------------------------

## [1] 0.8974062 (Calculate it manually)
```

- Explains:

It indicates that 89.74 % variance of 4 variable provided is explained by the first component. (PC1)

- DEMO: By prcomp()

```
# PCA 前記得標準化
engine_pca<- prcomp(engine,scale=T ) #相關矩陣分解
summary(engine_pca)#方差解釋度

--------------------------------------------------------------------------------------------

## Importance of components:
##                           PC1     PC2     PC3     PC4
```

```
## Standard deviation     1.8946 0.46234 0.38683 0.21674
## Proportion of Variance 0.8974 0.05344 0.03741 0.01174
## Cumulative Proportion  0.8974 0.95085 0.98826 1.00000
```

- DEMO: By prcomp()

```
prcomp.variance <- engine_pca$sdev ^2
result_prcomp <- prcomp.variance[1]/sum(prcomp.variance)
result_prcomp # Same value as we calculate it manually

## [1] 0.8974062  (Using prcomp function)
```

---

aiii. 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?

(i.e., think what concept the first principal component captures or represents)

```
engine_pca$rotation#[,1]

------------------------------------------------------------------------------------------

##                          PC1        PC2          PC3         PC4
## log.mpg.           0.4915415  0.5540404 -0.668490521  0.06742885
## log.cylinders.    -0.5011877  0.6122954  0.077765261 -0.60651280
## log.displacement. -0.5129165  0.3549228 -0.004150334  0.78161962
## log.horsepower.   -0.4940794 -0.4383645 -0.739632181 -0.12909816

------------------------------------------------------------------------------------------
##  After Abs

abs(engine_pca$rotation[,1])

------------------------------------------------------------------------------------------

##   log.mpg.    log.cylinders. log.displacement.   log.horsepower.
##    0.4915415      0.5011877      0.5129165         0.4940794
```

We can tell the ratio of 4 variable (log.mpg. etc) are pretty close after absolute. The last of the variable have the negative std to the first princple. Therefore, it would say PCA1 seize the positive relation to the log.mpg. I'll called it 'Efficiency'.

---

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

bi. Store the scores of the first principal component as a new column of cars_log

```
cars_log_bi <- cars_log
std.PC1<-as.numeric(engine_pca$x[,1])
cars_log_bi$std.PC1 <- -std.PC1
```

bii. Regress mpg over the column with PC1 scores (replacing cylinders, displacement, horsepower, and weight), as well as acceleration, model_year and origin

```
summary(lm(log.mpg.~std.PC1+log.cylinders.+log.displacement.+log.acceleration.+log.horsep
ower.+log.weight.+model_year+factor(origin),data=cars_log_bi))

------------------------------------------------------------------------------------------
## Call:
## lm(formula = log.mpg. ~ std.PC1 + log.cylinders. + log.displacement. +
##     log.acceleration. + log.horsepower. + log.weight. + model_year +
##     factor(origin), data = cars_log_bi)

## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.584e-13 -7.040e-16  7.550e-16  2.008e-15  6.442e-15

## Coefficients:
##                     Estimate Std. Error    t value Pr(>|t|)
## (Intercept)       -6.780e+00  9.557e-14 -7.095e+13   <2e-16 ***
## std.PC1           -6.918e-01  4.195e-15 -1.649e+14   <2e-16 ***
## log.cylinders.     1.147e+00  1.040e-14  1.103e+14   <2e-16 ***
## log.displacement.  6.661e-01  7.956e-15  8.372e+13   <2e-16 ***
## log.acceleration.  7.265e-15  7.153e-15  1.016e+00   0.3104
## log.horsepower.    9.954e-01  1.037e-14  9.599e+13   <2e-16 ***
## log.weight.       -7.659e-15  1.073e-14 -7.140e-01   0.4756
## model_year         5.890e-16  2.789e-16  2.112e+00   0.0353 *
## factor(origin)2    1.194e-15  2.501e-15  4.780e-01   0.6332
## factor(origin)3    6.877e-16  2.464e-15  2.790e-01   0.7803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.34e-14 on 382 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.796e+28 on 9 and 382 DF,  p-value: < 2.2e-16
```

---

biii. Try running the regression again over the same independent variables, but this time with everything standardized.

```
# Standardize the features
cars_log2 <- cars_log
std.PC1<-as.numeric(engine_pca$x[,1])

cars_log2.scale <- as.data.frame(scale(cars_log2[,c(1:7)]))
cars_log2.scale$origin<-cars_log$origin
cars_log2.scale$std.PC1 <- -std.PC1

summary(lm(log.mpg.~std.PC1+log.cylinders.+log.displacement.+log.acceleration.+log.horsep
ower.+log.weight.+model_year+factor(origin),data=cars_log2.scale))

------------------------------------------------------------------------------------------
## Call:
## lm(formula = log.mpg. ~ std.PC1 + log.cylinders. + log.displacement. +
##     log.acceleration. + log.horsepower. + log.weight. + model_year +
##     factor(origin), data = cars_log2.scale)
##
```

```
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -2.770e-14 -2.354e-16  5.930e-17  2.767e-16  1.672e-14 
## 
## Coefficients:
##                      Estimate Std. Error    t value Pr(>|t|)    
## (Intercept)         6.840e-16  1.375e-16  4.976e+00 9.86e-07 ***
## std.PC1            -2.034e+00  5.413e-16 -3.758e+15  < 2e-16 ***
## log.cylinders.      1.020e+00  4.056e-16  2.514e+15  < 2e-16 ***
## log.displacement.   1.043e+00  5.469e-16  1.908e+15  < 2e-16 ***
## log.acceleration.  -5.716e-17  1.670e-16 -3.420e-01    0.732    
## log.horsepower.     1.005e+00  4.595e-16  2.188e+15  < 2e-16 ***
## log.weight.         4.086e-16  3.892e-16  1.050e+00    0.294    
## model_year          2.084e-17  1.326e-16  1.570e-01    0.875    
## factor(origin)2    -7.587e-17  3.228e-16 -2.350e-01    0.814    
## factor(origin)3    -9.654e-17  3.179e-16 -3.040e-01    0.762    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.73e-15 on 382 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1 
## F-statistic: 1.452e+31 on 9 and 382 DF,  p-value: < 2.2e-16
```

Explains:

The std.PC1, log.cylinders, log.displacement., log.horsepower.is significant on mpg. log.acceleration., log.weight. , model_year, origin are not significant. The R square = SSR/SST = 1, means it perfect fit the linear model. The reason is that we regress it with its residual (std.PC1).

Question 2

Q2 Load security_questions.xlsx

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

a. How much variance did each extracted factor explain?

The Importance of components & the variance explained each component

```
raw_pca<-prcomp(df_ans, scale. = T)
summary(raw_pca)

----------------------------------------------------------------------------------------

## 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

var_explained  <- raw_pca$sdev^2 /sum(raw_pca$sdev^2); var_explained

##  [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
```

- Ans:

51.7% is explained by their first principal component (PC1).

8.8% is explained by their second principal component (PC2).

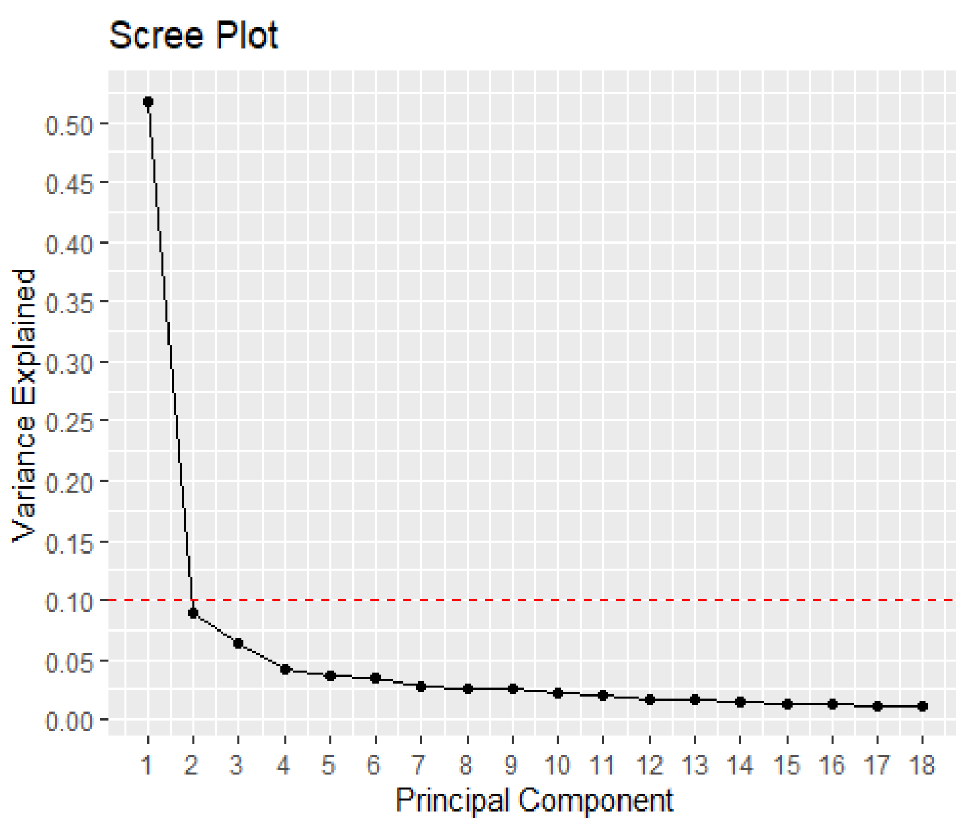
6.3% is explained by their third principal component (PC3).etc…

---

b. How many dimensions would you retain, according to the two criteria we discussed?

- Plot scree Plot via ggplot2 for more clearification.

```
qplot(c(1:18), var_explained) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  scale_x_continuous(breaks = seq(1, 18, by = 1)) +
    scale_y_continuous(breaks = seq(0, 1, by = 0.05))+
    geom_hline(aes(yintercept=.1),  color="red", linetype="dashed")
```



I 'd select only the first one component. Because the rest of them are within 0-10% (under red dashed line).

---

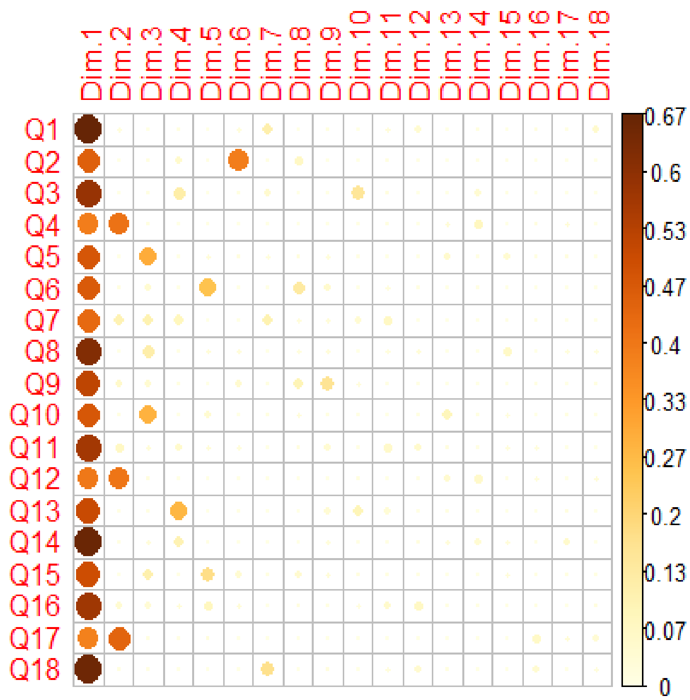
c. (ungraded) Can you interpret what any of the principal components mean? Try guessing the meaning of the first two or three PCs looking at the PC-vs-variable matrix

Reference: https://www.twblogs.net/a/5d419f19bd9eee517423483b

- All PC versus ALL Variable

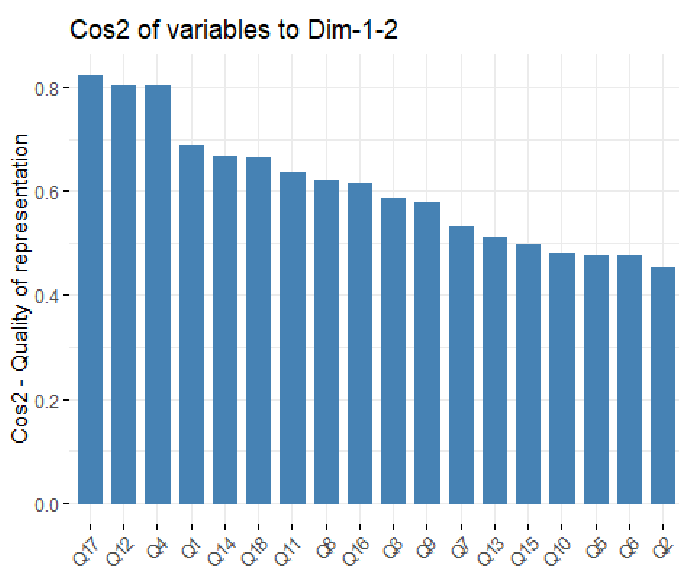
```
var <- get_pca_var(raw_pca)

corrplot(var$cos2, is.corr=FALSE)
```



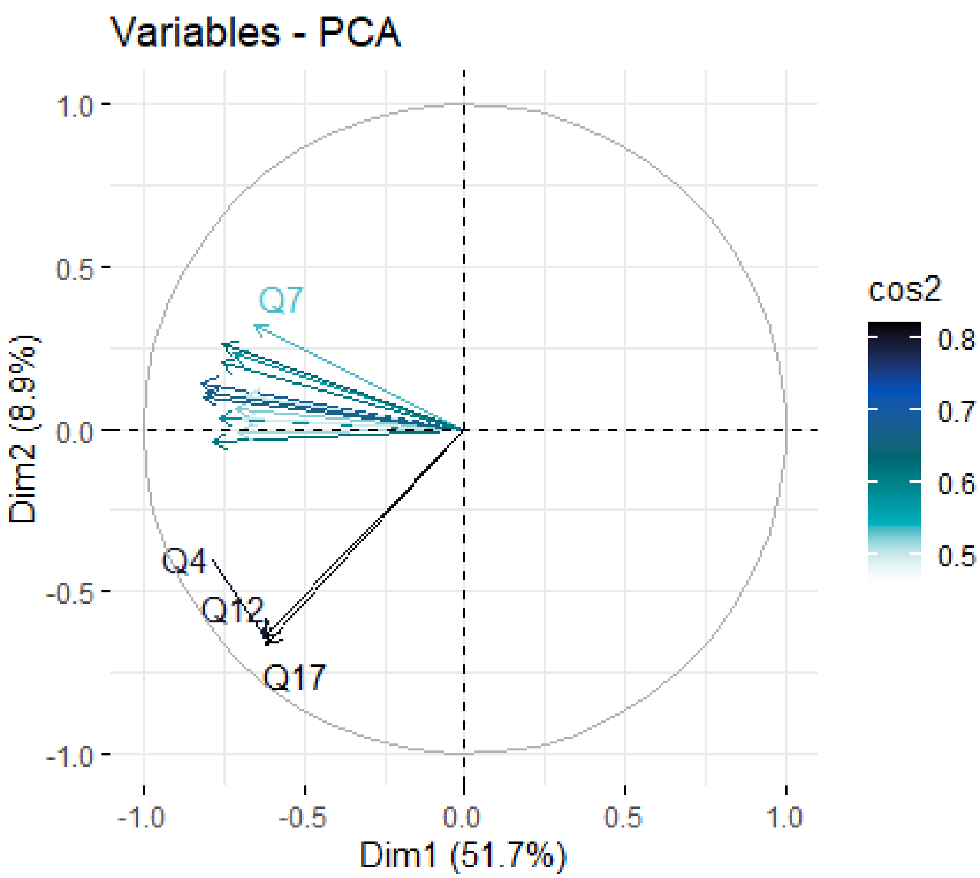
- The First 2 PC versus ALL Variable

```
fviz_cos2(raw_pca, choice = "var", axes = 1:2)
```



```
fviz_pca_var(raw_pca, col.var = "cos2",
 gradient.cols = c("00bb0c", "#00afbb", "#00676f", "#0052bb", 'black'),
 repel = TRUE) #

## Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Explain:

High cos2 indicates that the features have importance effect to o the principal component.

It can be used to measure the usefulness of the questions.

For the plot above,

we can tell that Q4, Q12, Q17 have strong effect to the principal component. And they were designed to ask the questions about the transaction.

ps:

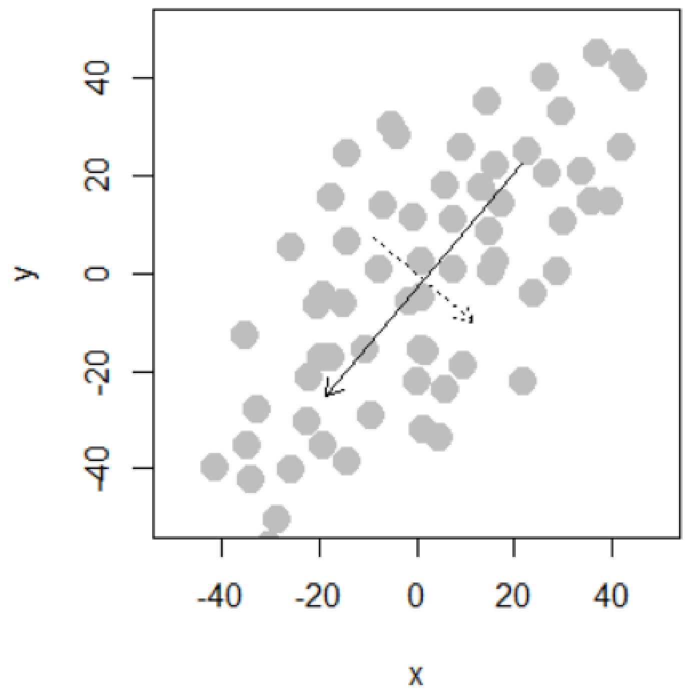
Q4 : "This site provides me with some evidence to protect against its denial of having received a transaction from me"

Q12 : "This site takes steps to make sure that the information in transit is not deleted"

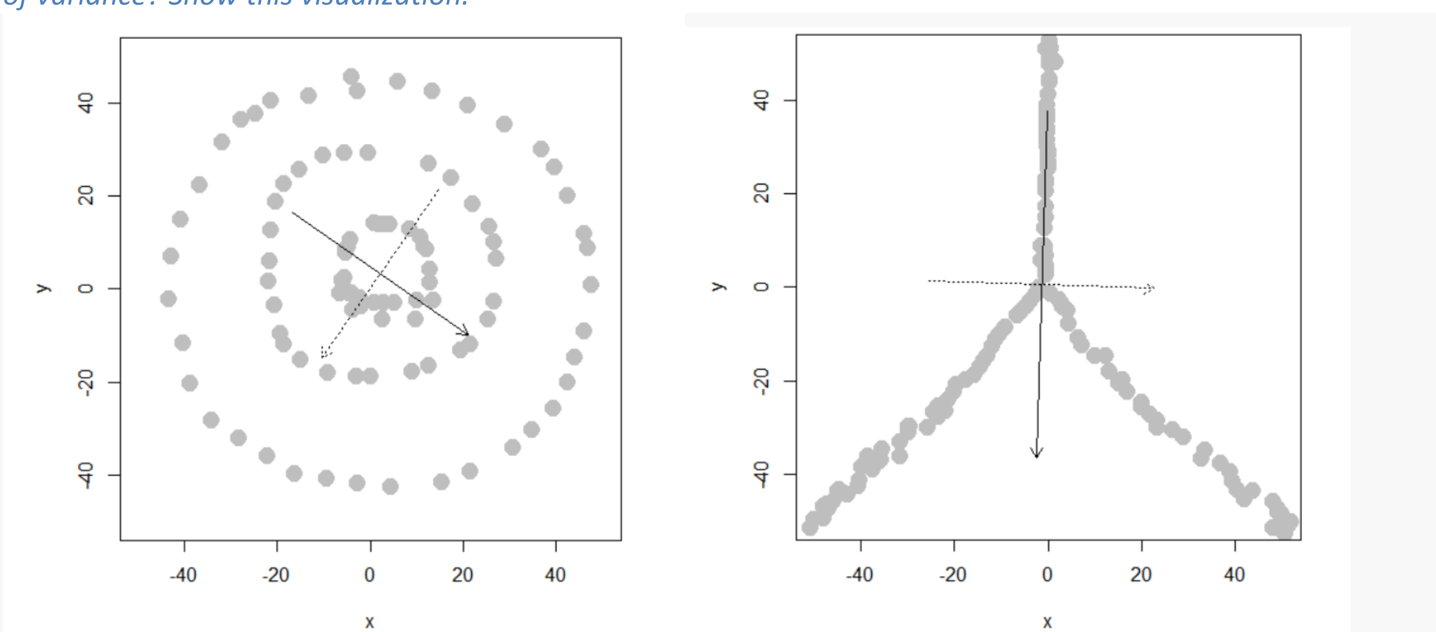
Q17 : "This site provides me with some evidence to protect against its denial of having participated in a transaction after processing it"

Question 3

a. Create an oval shaped scatter plot of points that stretches in two directions – you should find that the principal component vectors point in the major and minor directions of variance (dispersion). Show this visualization.



b. Can you create a scatterplot whose principal component vectors do NOT seem to match the major directions of variance? Show this visualization.



The first plot totally messed up. And the second one can only size the tendency of the upper part of data. Therefore, their principal component vectors doesn't match the major directions of variance.