Homework13_BACS

109090035

5/10/2023

Question 1) Let's revisit the issue of multicollinearity of main effects (between cylinders, displacement, horsepower, and weight) we saw in the cars dataset, and try to apply principal components to it. Start by recreating the cars_log dataset, which log-transforms all variables except model year and origin.

Important: remove any rows that have missing values.

```
library(dplyr)
# load the car dataset
cars <- read.table("/Users/user/Downloads/auto-data.txt", header=FALSE, na.strings = "?")

names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "origi
n","model")

# Remove rows with missing values
cars <- cars[complete.cases(cars), ]

# Apply log transformation
cars_log <- cars %>%
  mutate_at(vars(mpg, cylinders, displacement, horsepower, weight,acceleration), log)

# Check the result
head(cars_log)
```

```
       mpg cylinders displacement horsepower   weight acceleration model_year
1 2.890372  2.079442     5.726848   4.867534 8.161660     2.484907         70
2 2.708050  2.079442     5.857933   5.105945 8.214194     2.442347         70
3 2.890372  2.079442     5.762051   5.010635 8.142063     2.397895         70
4 2.772589  2.079442     5.717028   5.010635 8.141190     2.484907         70
5 2.833213  2.079442     5.710427   4.941642 8.145840     2.351375         70
6 2.708050  2.079442     6.061457   5.288267 8.375860     2.302585         70
  origin                       model
1      1 chevrolet chevelle malibu
2      1          buick skylark 320
3      1         plymouth satellite
4      1              amc rebel sst
5      1                ford torino
6      1           ford galaxie 500
```

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(Give this smaller data frame an appropriate name – what might they jointly mean?)

```
# Assuming 'cars_log' is your original data frame
df_collinear <- cars_log[, c("cylinders", "displacement", "horsepower", "weight")]
head(df_collinear)
```

```
  cylinders displacement horsepower   weight
1  2.079442     5.726848   4.867534 8.161660
2  2.079442     5.857933   5.105945 8.214194
3  2.079442     5.762051   5.010635 8.142063
4  2.079442     5.717028   5.010635 8.141190
5  2.079442     5.710427   4.941642 8.145840
6  2.079442     6.061457   5.288267 8.375860
```

All variables are interrelated, as they all contribute to a car's overall performance and efficiency.

ii. 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)

```
# Compute the principal components
pca <- prcomp(df_collinear, center = TRUE, scale. = TRUE)

# Compute the proportion of variance explained by the first principal component
explained_variance <- summary(pca)$importance[2, 1]

print(paste("The first principal component explains", round(explained_variance*100, 2), "% of the variance."))
```

```
[1] "The first principal component explains 91.86 % of the variance."
```

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

```
# Get the eigenvectors (loadings)
loadings <- pca$rotation

# Print the loadings of the first principal component
print(loadings[, 1])
```

```
   cylinders displacement   horsepower       weight
  -0.4979145   -0.5122968   -0.4856159   -0.5037960
```

All the values are negative and roughly of equal magnitude. This suggests that the first principal component might represent a concept where all these variables decrease together.

Given that these variables are all related to the power and size of the car (i.e., the number of cylinders, the engine displacement, the horsepower, and the weight), this principal component could potentially capture something like “inverse of power and size” of the cars, since a higher value of the component corresponds to lower values of all these variables.

This is just a rough interpretation based on the information provided.

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 cars_log$new_column_name <- …scores of PC1… Give this new column a name suitable for what it captures (see 1.a.i.)

```
# Assuming your PCA object is named 'pca'
cars_log$Power_Size_PC1 <- pca$x[,1]
head(cars_log)
```

```
       mpg cylinders displacement horsepower   weight acceleration model_year
1 2.890372  2.079442     5.726848   4.867534 8.161660     2.484907         70
2 2.708050  2.079442     5.857933   5.105945 8.214194     2.442347         70
3 2.890372  2.079442     5.762051   5.010635 8.142063     2.397895         70
4 2.772589  2.079442     5.717028   5.010635 8.141190     2.484907         70
5 2.833213  2.079442     5.710427   4.941642 8.145840     2.351375         70
6 2.708050  2.079442     6.061457   5.288267 8.375860     2.302585         70
  origin                       model Power_Size_PC1
1      1 chevrolet chevelle malibu      -2.036645
2      1         buick skylark 320      -2.593998
3      1        plymouth satellite      -2.237767
4      1             amc rebel sst      -2.192902
5      1               ford torino      -2.097313
6      1          ford galaxie 500      -3.337215
```

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

origin

```
model <- lm(mpg ~ Power_Size_PC1 + acceleration + model_year + origin, data = cars_log)
summary(model)
```

```

Call:
lm(formula = mpg ~ Power_Size_PC1 + acceleration + model_year +
    origin, data = cars_log)

Residuals:
     Min       1Q   Median       3Q      Max
-0.51070 -0.06039 -0.00161  0.06271  0.46795

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.386083   0.166466   8.327 1.45e-15 ***
Power_Size_PC1  0.145547   0.004886  29.786  < 2e-16 ***
acceleration   -0.191608   0.041645  -4.601 5.71e-06 ***
model_year      0.029210   0.001776  16.444  < 2e-16 ***
origin          0.009815   0.009680   1.014    0.311
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1198 on 387 degrees of freedom
Multiple R-squared:  0.8772,    Adjusted R-squared:  0.876
F-statistic: 691.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

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

other columns?

```
# Standardizing variables
cars_log$Power_Size_PC1_std <- scale(cars_log$Power_Size_PC1)
cars_log$acceleration_std <- scale(cars_log$acceleration)
cars_log$model_year_std <- scale(cars_log$model_year)
cars_log$origin_std <- scale(cars_log$origin)

# Running the regression again
model_std <- lm(mpg ~ Power_Size_PC1_std + acceleration_std + model_year_std + origin_std, data = cars_log)
summary(model_std)
```

```

Call:
lm(formula = mpg ~ Power_Size_PC1_std + acceleration_std + model_year_std +
    origin_std, data = cars_log)

Residuals:
     Min       1Q   Median       3Q      Max
-0.51070 -0.06039 -0.00161  0.06271  0.46795

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.098313   0.006049 512.220  < 2e-16 ***
Power_Size_PC1_std  0.278990   0.009366  29.786  < 2e-16 ***
acceleration_std   -0.034673   0.007536  -4.601 5.71e-06 ***
model_year_std      0.107602   0.006543  16.444  < 2e-16 ***
origin_std          0.007906   0.007797   1.014    0.311
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1198 on 387 degrees of freedom
Multiple R-squared:  0.8772,    Adjusted R-squared:  0.876
F-statistic: 691.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

The new column, Power_Size_PC1_std, has a coefficient estimate of 0.278990 and a t-value of 29.786, which is highly statistically significant (p < 2e-16).

This suggests that Power_Size_PC1_std, which encapsulates the combined effect of the four original, highly-collinear variables (cylinders, displacement, horsepower, and weight), is a significant predictor of miles per gallon (mpg).

The t-value can give us an idea of the relative importance of each predictor. The larger the absolute value of the t-value, the more “important” the predictor is in contributing to the response variable variation. Here, Power_Size_PC1_std has the second largest t-value, surpassed only by the intercept, indicating that it is an important predictor.

Question 2) Please download the Excel data file security_questions.xlsx from Canvas. In your analysis, you can either try to read the data sheet from the Excel file directly from R (there might be a package for that!) or you can try to export the data sheet to a CSV file before reading it into R.

```
questions <- read_excel("/Users/user/Downloads/security_questions.xlsx", sheet = 1)
data <- read_excel("/Users/user/Downloads/security_questions.xlsx", sheet = 2)

head(questions)
```

```
# A tibble: 6 × 2
  Q1    I am convinced that this site respects the confidentiality of the tran…¹
  <chr> <chr>
1 Q2    All communications with this site are restricted to the site and me
2 Q3    This site checks the information communicated with me for accuracy
3 Q4    This site provides me with some evidence to protect against its denial …
4 Q5    The transactions I send are transmitted to the real site to which I wan…
5 Q6    This site checks all communications between the site and me for protect…
6 Q7    This site never sells my personal information in their computer databas…
# ℹ abbreviated name:
#   ¹`I am convinced that this site respects the confidentiality of the transactions received from me`
```

```
head(data)
```

```
# A tibble: 6 × 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
# ℹ 5 more variables: Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>, Q18 <dbl>
```

A group of researchers is studying how customers who shopped on e-commerce websites over the winter holiday season perceived the security of their most recently used e-commerce site. Based on feedback from experts, the company has created eighteen questions (see 'questions' tab of excel file) regarding security considerations at e-commerce websites. Over 400 customers responded to these questions (see 'data' tab of Excel file). The researchers now wants to use the results of these eighteen questions to reveal if there are some underlying dimensions of people's perception of online security that effectively capture the variance of these eighteen questions. Let's analyze the principal components of the eighteen items.

a. How much variance did each extracted factor explain?

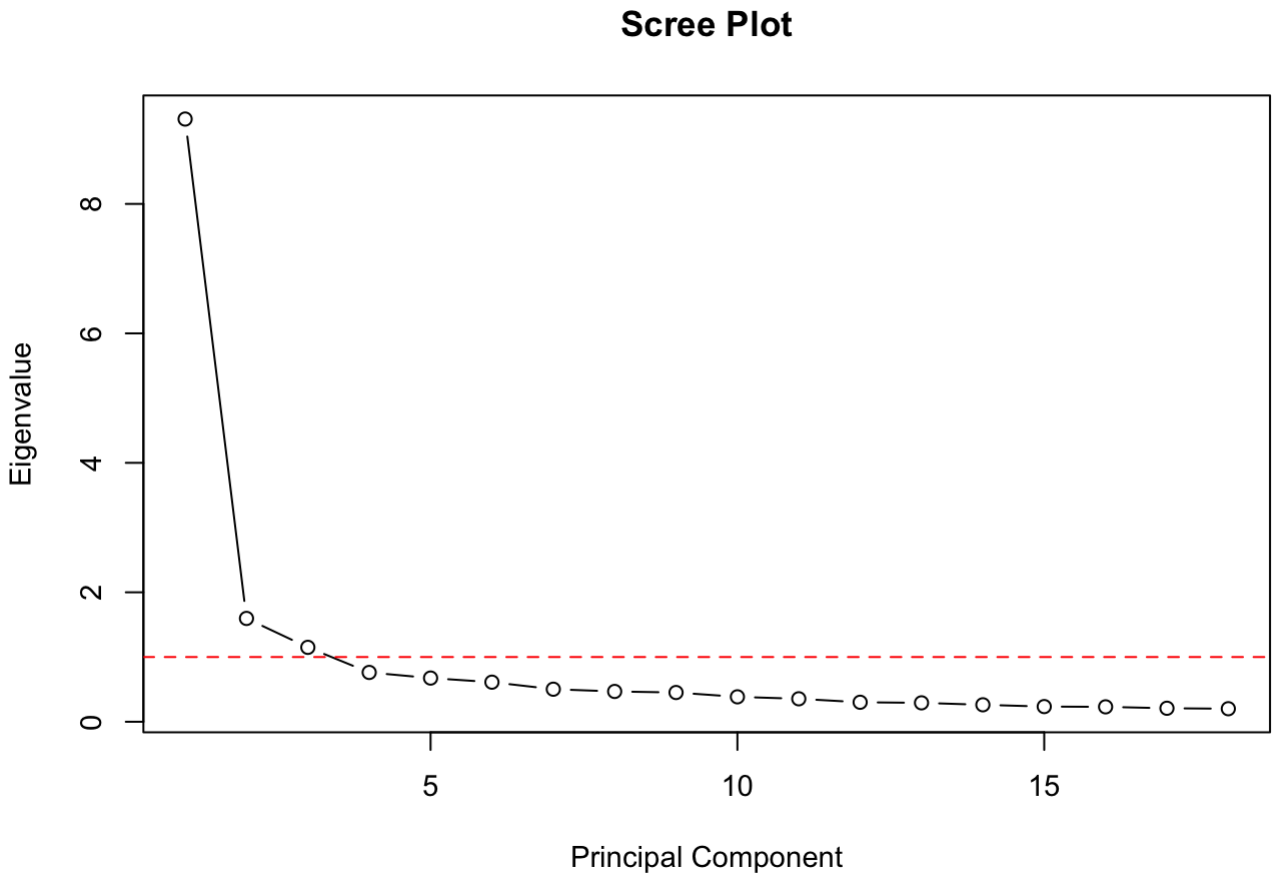
```
# Run PCA on the data
pca_result <- prcomp(data, center = TRUE, scale. = TRUE)

# Print summary of the PCA result
print(summary(pca_result))
```

```
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
```

In the summary, the “Proportion of Variance” row shows how much variance each principal component explains.

b. How many dimensions would you retain, according to the two criteria we discussed?(Eigenvalue ≥ 1 and Scree Plot – can you show the screeplot with eigenvalue=1 threshhold?)

```
# Scree plot
plot(pca_result$sdev^2, type = "b", main = "Scree Plot",
     xlab = "Principal Component", ylab = "Eigenvalue")
abline(h = 1, lty = 2, col = "red")  # Add a horizontal line at y = 1
```



We can visualize this using a scree plot, where we plot the eigenvalues in decreasing order:

The point where the plot bends sharply (the “elbow”) is often used as a cutoff: components to the left of the elbow are retained, and components to the right are discarded.

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

```
# Print the loadings of the first three components
print(pca_result$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
```

The loadings can be interpreted as the correlations between the original variables and the principal components.

Looking at the loadings, we can make some interpretations about the components:

PC1: All the variables have negative loadings on the first principal component, which means that PC1 might represent a general tendency across all the questions. This could be a general level of comfort or perceived security with online shopping, with high scores indicating low comfort or perceived security.

PC2: The loadings on the second principal component are mixed. Variables Q4, Q12 and Q17 have notably high negative loadings on this component, suggesting that these questions might be capturing a different aspect of perceived security that is in some way opposed to the aspects captured by the other questions.

PC3: The loadings on the third principal component are also mixed. Variables Q5, Q8, Q10 and Q15 have high negative loadings, while Q7 has a positive loading. This suggests that these questions might be capturing yet another aspect of perceived security, separate from those captured by PC1 and PC2.

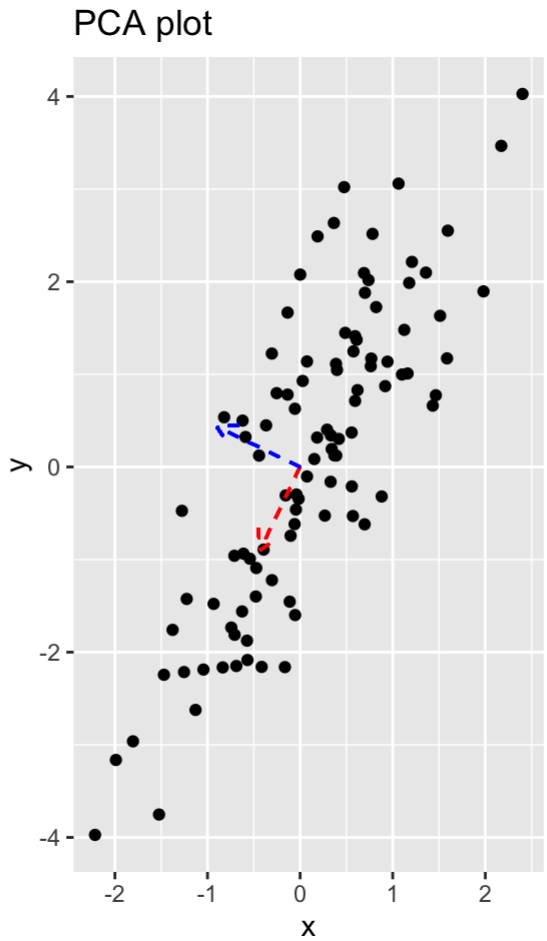
Question 3) Let's simulate how principal components behave interactively: run the interactive_pca() function from the compstatslib package we have used earlier:

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.

When we perform PCA on this data, the first principal component (PC1) will be a line that goes through the center of the cloud of points along the direction where the data varies the most. This is the long axis of the oval. The second principal component (PC2) will be a line perpendicular to PC1, going through the center of the data along the direction of the second-most variance. This will be along the short axis of the oval.

In R, the PCA plot might show the vectors (arrows) for PC1 and PC2 superimposed on the scatterplot of the data. PC1 will be a longer vector (since it explains more of the variance), and PC2 will be shorter.

```
# Create correlated data
set.seed(1)
x <- rnorm(100)
y <- 1.5 * x + rnorm(100)

# Combine the data into a data frame
df <- data.frame(x = x, y = y)

# Perform PCA
pca <- prcomp(df)

# Create a scatter plot
ggplot(df, aes(x, y)) +
  geom_point() +
  coord_fixed() +
  geom_segment(aes(x = 0, y = 0, xend = pca$rotation[1, 1], yend = pca$rotation[2, 1]),
               arrow = arrow(length = unit(0.3, "cm")),
               color = "red",
               linetype = "dashed") +
  geom_segment(aes(x = 0, y = 0, xend = pca$rotation[1, 2], yend = pca$rotation[2, 2]),
               arrow = arrow(length = unit(0.3, "cm")),
               color = "blue",
               linetype = "dashed") +
  ggtitle("PCA plot")
```




```
# Load the necessary package
library(compstatslib)
# Run the interactive_pca function
# using interactive_pca()
```

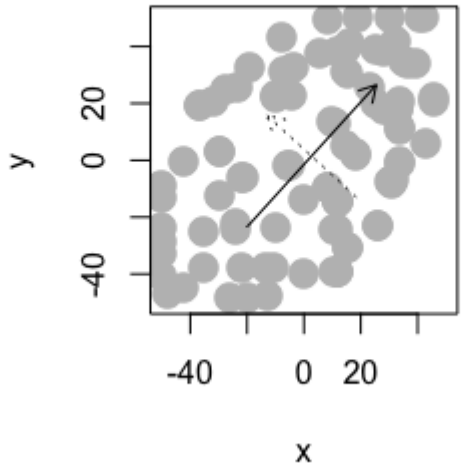
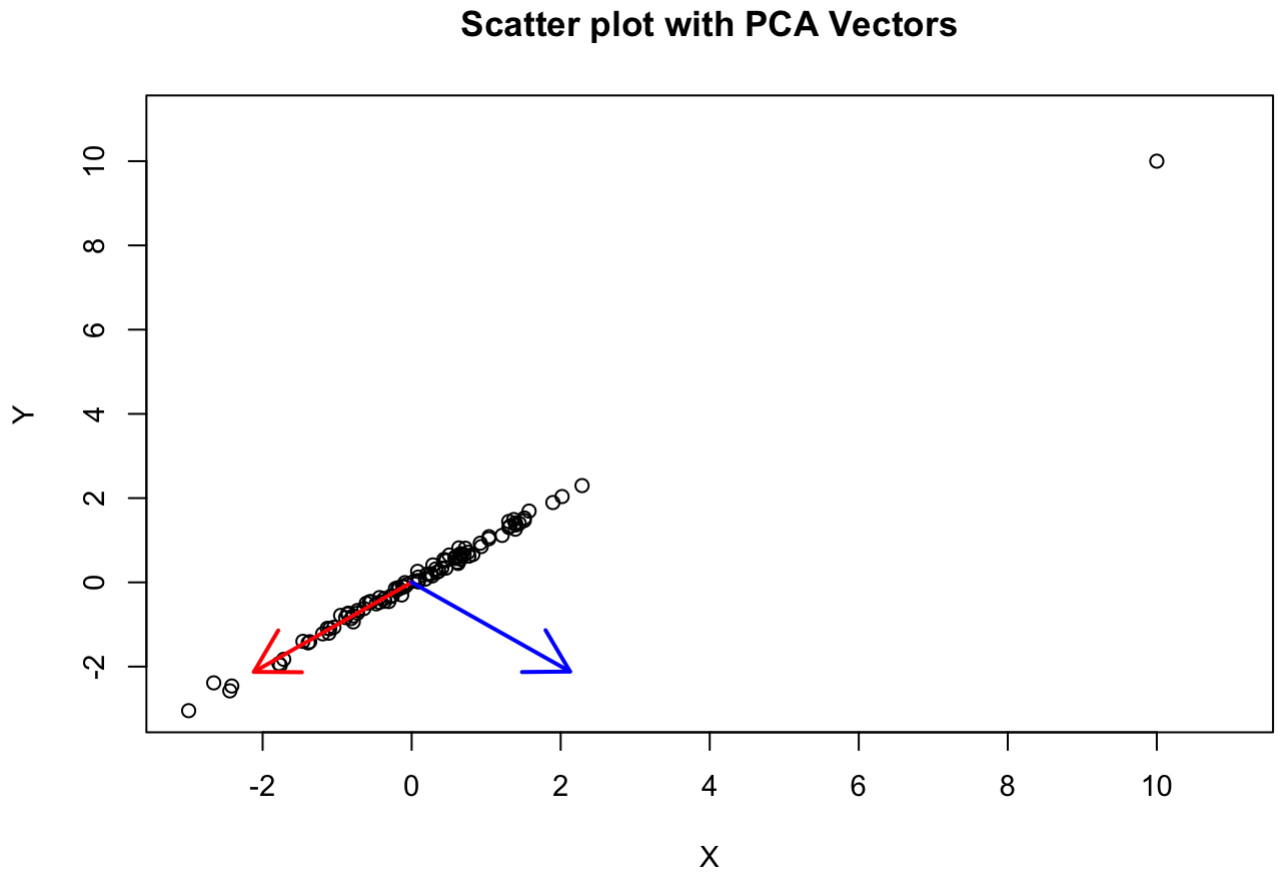


Image description

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

we generate data where y is a noisy version of x, so the major direction of variance should be along the line y=x. However, we then add an outlier at (10, 10). As PCA is sensitive to outliers, this will pull one of the principal component vectors towards the outlier, making it appear as though the principal component vectors do not match the major directions of variance in the original data.

```
# Simulate data
set.seed(42)
x <- rnorm(100)
y <- x + rnorm(100, sd = 0.1)
df <- data.frame(x = x, y = y)

# Add an outlier
df <- rbind(df, c(10, 10))

# PCA
pca <- prcomp(df)

# Scatter plot
plot(df, xlab = "X", ylab = "Y", main = "Scatter plot with PCA Vectors", xlim = c(-3, 11), ylim = c(-3, 11))

# Add principal component vectors
arrows(0, 0, pca$rotation[1, 1] * 3, pca$rotation[2, 1] * 3, col = "red", lwd = 2)
arrows(0, 0, pca$rotation[1, 2] * 3, pca$rotation[2, 2] * 3, col = "blue", lwd = 2)
```




```
# Load the necessary package
library(compstatslib)
# Run the interactive_pca function
# using interactive_pca()
```

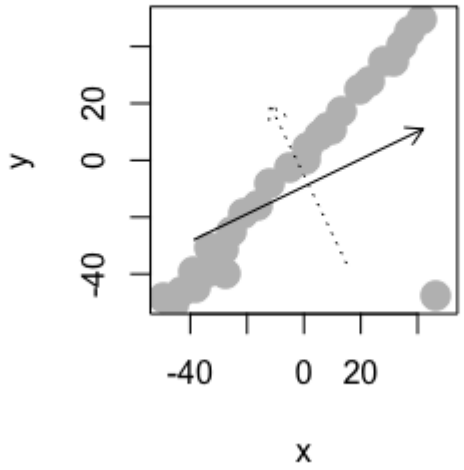


Image description