Homework 6

Matt Viana

Table of contents

> **Important**
>
> Please read the instructions carefully before submitting your assignment.
>
> 1. This assignment requires you to only upload a `PDF` file on Canvas
> 2. Don't collapse any code cells before submitting.
> 3. Remember to make sure all your code output is rendered properly before uploading your submission.
>
> ⚠️ Please add your name to the author information in the frontmatter before submitting your assignment ⚠️

In this assignment, we will perform various tasks involving principal component analysis (PCA), principal component regression, and dimensionality reduction.

We will need the following packages:

```
packages <- c(
  "tibble",
  "dplyr",
  "readr",
  "tidyr",
  "purrr",
  "broom",
  "magrittr",
  "corrplot",
  "car",
  "janitor",
  "ggplot2",
  "reshape2"
)
# renv::install(packages)
sapply(packages, require, character.only=T)
```

Question 1

70 points

Principal component anlaysis and variable selection

1.1 (5 points)

The `data` folder contains a `spending.csv` dataset which is an illustrative sample of monthly spending data for a group of 5000 people across a variety of categories. The response variable, `income`, is their monthly income, and objective is to predict the `income` for a an individual based on their spending patterns.

Read the data file as a tibble in R. Preprocess the data such that:

1. the variables are of the right data type, e.g., categorical variables are encoded as factors
2. all column names to lower case for consistency
3. Any observations with missing values are dropped

```
path <- "data/spending.csv"
df <- read_csv(path) %>%
  janitor::clean_names() %>%
  mutate_if(is.character, as.factor) %>%
  na.omit()
```

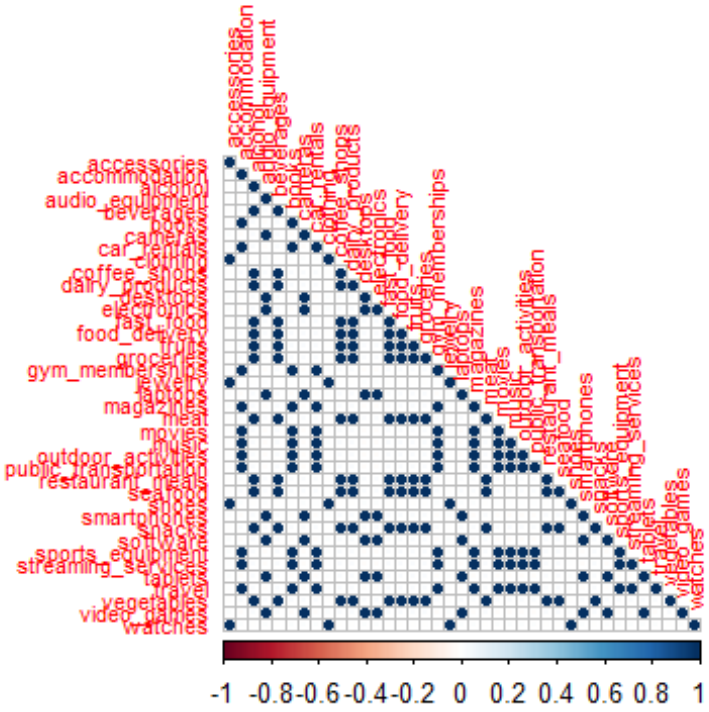
```
Rows: 5000 Columns: 40
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
dbl (40): accessories, accommodation, alcohol, audio_equipment, beverages, b...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

1.2 (5 points)

Visualize the correlation between the variables using the `corrplot()` function. What do you observe? What does this mean for the model?

```
df_x <- df %>%
  select(-income) %>%
  cor() %>%
  corrplot(method = "circle", type = "lower", tl.cex = 0.7)
```



1.3 (5 points)

Run a linear regression model to predict the `income` variable using the remaining predictors. Interpret the coefficients and summarize your results.

```
model <- lm(income ~ ., data = df)
summary(model)
```

```
Call:
lm(formula = income ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.6875 -1.6569  0.0427  1.6633  9.5623 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -0.077509   0.121730  -0.637 0.524330    
accessories           0.299876   0.031786   9.434  < 2e-16 ***
accommodation         0.113632   0.031262   3.635 0.000281 ***
alcohol              -0.005958   0.033266  -0.179 0.857873    
audio_equipment       0.602004   0.033483  17.979  < 2e-16 ***
beverages             0.043335   0.034111   1.270 0.204000    
books                 0.070530   0.033238   2.122 0.033892 *  
```

```
cameras                 0.461827   0.033572  13.756  < 2e-16 ***
car_rentals             0.124875   0.032809   3.806 0.000143 ***
clothing                0.504228   0.026055  19.352  < 2e-16 ***
coffee_shops            0.048839   0.034909   1.399 0.161864
dairy_products          0.024548   0.032715   0.750 0.453082
desktops                0.391673   0.033393  11.729  < 2e-16 ***
electronics             1.079627   0.030035  35.946  < 2e-16 ***
fast_food               0.077531   0.033014   2.348 0.018893 *
food_delivery          -0.004903   0.034257  -0.143 0.886188
fruits                  0.059089   0.033321   1.773 0.076237 .
groceries               0.077694   0.031601   2.459 0.013981 *
gym_memberships         0.141168   0.033410   4.225 2.43e-05 ***
jewelry                 0.213726   0.032834   6.509 8.30e-11 ***
laptops                 0.594328   0.032548  18.260  < 2e-16 ***
magazines               0.080762   0.033694   2.397 0.016571 *
meat                    0.081262   0.032367   2.511 0.012083 *
movies                  0.110296   0.033326   3.310 0.000941 ***
music                   0.159925   0.033398   4.788 1.73e-06 ***
outdoor_activities      0.087846   0.032356   2.715 0.006651 **
public_transportation   0.061138   0.033022   1.851 0.064169 .
restaurant_meals        0.066129   0.033225   1.990 0.046611 *
seafood                 0.061318   0.033786   1.815 0.069596 .
shoes                   0.463185   0.029613  15.641  < 2e-16 ***
smartphones             0.780150   0.031538  24.737  < 2e-16 ***
snacks                  0.007464   0.033229   0.225 0.822290
software                0.408500   0.034102  11.979  < 2e-16 ***
sports_equipment        0.033328   0.033969   0.981 0.326574
streaming_services      0.150614   0.031902   4.721 2.41e-06 ***
tablets                 0.637266   0.033133  19.234  < 2e-16 ***
travel                  0.129161   0.031457   4.106 4.09e-05 ***
vegetables             -0.066111   0.033162  -1.994 0.046257 *
video_games             0.863309   0.031392  27.501  < 2e-16 ***
watches                 0.145853   0.033467   4.358 1.34e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.434 on 4960 degrees of freedom
Multiple R-squared:  0.9999,    Adjusted R-squared:  0.9999
F-statistic: 1.834e+06 on 39 and 4960 DF,  p-value: < 2.2e-16
```

1.3 (5 points)

Diagnose the model using the `vif()` function. What do you observe? What does this mean for the model?

```r
vif_values <- vif(model)
print(vif_values)
```

```
          accessories          accommodation                alcohol
            152.06821              681.15504              387.23376
     audio_equipment              beverages                  books
            1755.56441              914.69186              192.91781
               cameras           car_rentals               clothing
             785.43147              423.55906              282.25143
          coffee_shops        dairy_products               desktops
             425.39644             2336.74847              776.75697
           electronics             fast_food         food_delivery
            3927.16511             1519.85171              921.68162
                fruits              groceries        gym_memberships
            1550.05678             3136.80325              438.30224
               jewelry                laptops              magazines
              72.38215             1658.76990              198.53619
                  meat                 movies                  music
            2284.43676              437.28082              437.03990
   outdoor_activities public_transportation       restaurant_meals
             411.17302              427.77815             1540.26240
               seafood                  shoes           smartphones
            1594.08027              233.33301             2772.27822
                snacks               software       sports_equipment
             868.24282              810.28919              201.00255
   streaming_services                tablets                 travel
             709.25592             1718.78339              690.69616
            vegetables           video_games                watches
            1536.40686             2745.64421               75.56457
```

1.4 (5 points)

Perform PCA using the `princomp` function in R. Print the summary of the PCA object.

```
pca <- princomp(df %>% select(-income), cor = TRUE)
summary(pca)

Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4       Comp.5
Standard deviation     3.6201099 3.4479976 2.9939875 2.2288727 0.1125697569
Proportion of Variance 0.3360307 0.3048381 0.2298452 0.1273814 0.0003249218
Cumulative Proportion  0.3360307 0.6408688 0.8707140 0.9980954 0.9984202743
                           Comp.6       Comp.7       Comp.8       Comp.9
Standard deviation     0.0960605322 0.0708312069 0.0691539249 0.0670242037
Proportion of Variance 0.0002366058 0.0001286426 0.0001226222 0.0001151857
Cumulative Proportion  0.9986568801 0.9987855227 0.9989081448 0.9990233306
                          Comp.10      Comp.11      Comp.12      Comp.13
Standard deviation     0.0653196274 5.099363e-02 0.0498072940 4.762347e-02
Proportion of Variance 0.0001094014 6.667565e-05 0.0000636094 5.815371e-05
Cumulative Proportion  0.9991327320 9.991994e-01 0.9992630170 9.993212e-01
                          Comp.14      Comp.15      Comp.16      Comp.17
Standard deviation     0.0469865879 4.611213e-02 0.0459026903 4.552808e-02
```

```
Proportion of Variance 0.0000566087 5.452125e-05 0.0000540271 5.314888e-05
Cumulative Proportion  0.9993777794 9.994323e-01 0.9994863278 9.995395e-01
                            Comp.18      Comp.19      Comp.20      Comp.21
Standard deviation     4.516751e-02 3.944038e-02 0.0358645643 3.505209e-02
Proportion of Variance 5.231037e-05 3.988573e-05 0.0000329812 3.150383e-05
Cumulative Proportion  9.995918e-01 9.996317e-01 0.9996646540 9.996962e-01
                            Comp.22      Comp.23      Comp.24      Comp.25
Standard deviation     3.460809e-02 3.435268e-02 3.297822e-02 3.240319e-02
Proportion of Variance 3.071076e-05 3.025915e-05 2.788623e-05 2.692223e-05
Cumulative Proportion  9.997269e-01 9.997571e-01 9.997850e-01 9.998119e-01
                            Comp.26      Comp.27      Comp.28      Comp.29
Standard deviation     3.135574e-02 2.976920e-02 2.508623e-02 2.460025e-02
Proportion of Variance 2.520981e-05 2.272321e-05 1.613638e-05 1.551723e-05
Cumulative Proportion  9.998371e-01 9.998599e-01 9.998760e-01 9.998915e-01
                            Comp.30      Comp.31      Comp.32      Comp.33
Standard deviation     2.426600e-02 2.374599e-02 2.334190e-02 2.283049e-02
Proportion of Variance 1.509843e-05 1.445825e-05 1.397036e-05 1.336491e-05
Cumulative Proportion  9.999066e-01 9.999211e-01 9.999350e-01 9.999484e-01
                            Comp.34      Comp.35      Comp.36      Comp.37
Standard deviation     2.119139e-02 1.968544e-02 1.937808e-02 1.742835e-02
Proportion of Variance 1.151475e-05 9.936319e-06 9.628464e-06 7.788395e-06
Cumulative Proportion  9.999599e-01 9.999699e-01 9.999795e-01 9.999873e-01
                            Comp.38      Comp.39
Standard deviation     1.677847e-02 1.464440e-02
Proportion of Variance 7.218385e-06 5.498931e-06
Cumulative Proportion  9.999945e-01 1.000000e+00
```

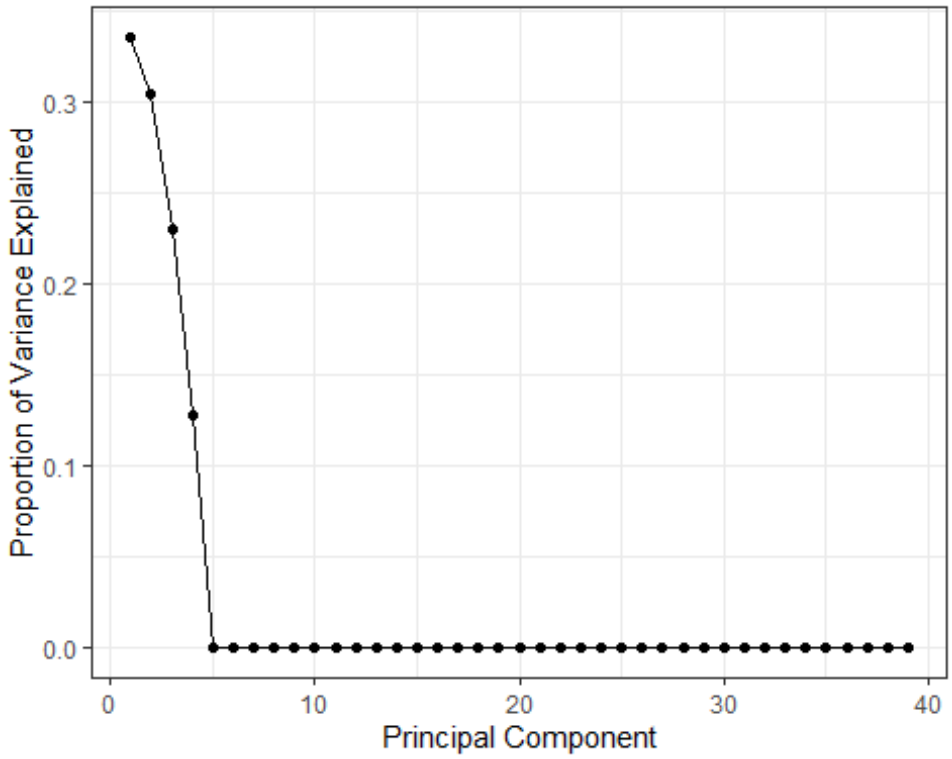
1.5 (5 points)

Make a screeplot of the proportion of variance explained by each principal component. How many principal components would you choose to keep? Why?

```
scree_plot <- qplot(c(1:length(pca$sdev)), pca$sdev^2/sum(pca$sdev^2),
                    xlab = "Principal Component",
                    ylab = "Proportion of Variance Explained") +
  geom_line() +
  theme_bw()

Warning: `qplot()` was deprecated in ggplot2 3.4.0.

scree_plot
```



1.6 (5 points)

By setting any factor loadings below 0.2 to 0, summarize the factor loadings for the principal components that you chose to keep.

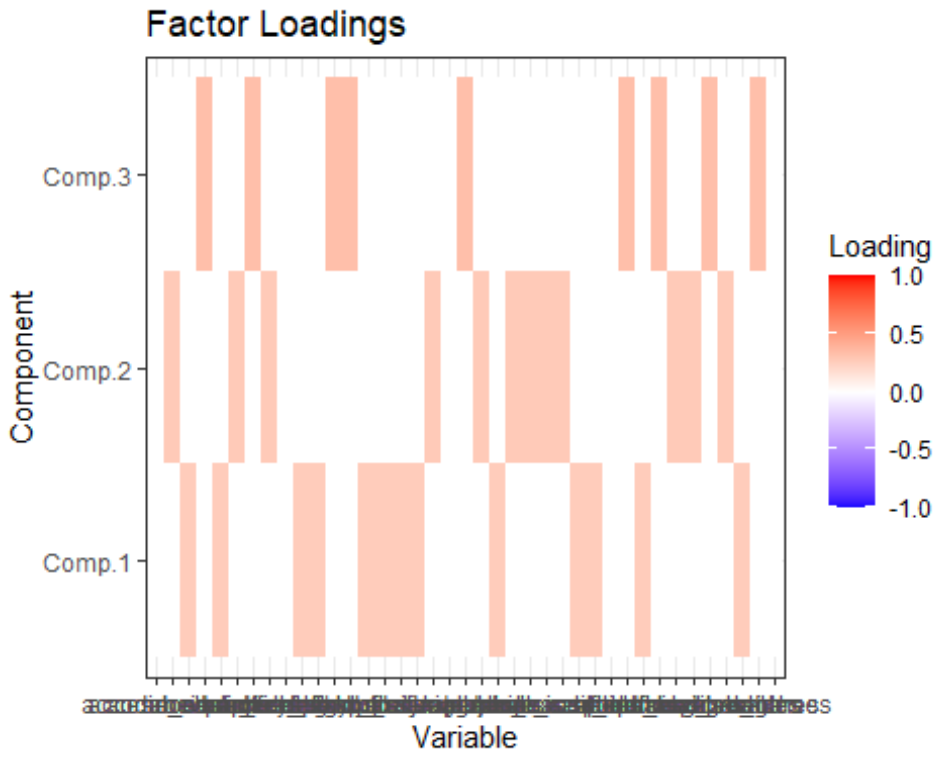
```
num_components <- 3

clean_loadings <- pca$loadings[, 1:num_components]
clean_loadings[abs(clean_loadings) < 0.2] <- 0

loadings_data <- melt(clean_loadings)
colnames(loadings_data) <- c("Variable", "Component", "Loading")

loadings_plot <- ggplot(loadings_data, aes(Variable, Component, fill =
Loading)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1, 1)) +
  labs(title = "Factor Loadings", x = "Variable", y = "Component") +
  theme_bw()

loadings_plot
```



Visualize the factor loadings.

```
df_pca <- cbind(df$income, pca$scores[, 1:num_components])
colnames(df_pca) <- c("income", paste0("PC", 1:num_components))
```

1.7 (15 points)

Based on the factor loadings, what do you think the principal components represent?

Provide an interpreation for each principal component you chose to keep.

1.8 (10 points)

Create a new data frame with the original response variable `income` and the principal components you chose to keep. Call this data frame `df_pca`.

```
num_components <- 3
df_pca <- cbind(df$income, pca$scores[, 1:num_components])
colnames(df_pca) <- c("income", paste0("PC", 1:num_components))
```

Fit a regression model to predict the `income` variable using the principal components you chose to keep. Interpret the coefficients and summarize your results.

```r
num_components <- 3
df_pca <- cbind(df$income, pca$scores[, 1:num_components])
colnames(df_pca) <- c("income", paste0("PC", 1:num_components))

df_pca <- as.data.frame(df_pca)

model_pca <- lm(income ~ ., data = df_pca)
summary(model_pca)
```

```
Call:
lm(formula = income ~ ., data = df_pca)

Residuals:
    Min      1Q  Median      3Q     Max 
-44.345 -18.599  -0.293  18.730  47.846 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 628.17783    0.30371  2068.4   <2e-16 ***
PC1          13.33571    0.08390   159.0   <2e-16 ***
PC2          -1.16303    0.08808   -13.2   <2e-16 ***
PC3          95.58547    0.10144   942.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.48 on 4996 degrees of freedom
Multiple R-squared:  0.9946,    Adjusted R-squared:  0.9946 
F-statistic: 3.044e+05 on 3 and 4996 DF,  p-value: < 2.2e-16
```

Compare the results of the regression model in 1.3 and 1.9. What do you observe? What does this mean for the model?

```r
summary(model)
```

```
Call:
lm(formula = income ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.6875 -1.6569  0.0427  1.6633  9.5623 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -0.077509   0.121730  -0.637 0.524330    
accessories           0.299876   0.031786   9.434  < 2e-16 ***
accommodation         0.113632   0.031262   3.635 0.000281 ***
alcohol              -0.005958   0.033266  -0.179 0.857873    
audio_equipment       0.602004   0.033483  17.979  < 2e-16 ***
```

```
beverages                0.043335   0.034111   1.270 0.204000
books                    0.070530   0.033238   2.122 0.033892 *
cameras                  0.461827   0.033572  13.756  < 2e-16 ***
car_rentals              0.124875   0.032809   3.806 0.000143 ***
clothing                 0.504228   0.026055  19.352  < 2e-16 ***
coffee_shops             0.048839   0.034909   1.399 0.161864
dairy_products           0.024548   0.032715   0.750 0.453082
desktops                 0.391673   0.033393  11.729  < 2e-16 ***
electronics              1.079627   0.030035  35.946  < 2e-16 ***
fast_food                0.077531   0.033014   2.348 0.018893 *
food_delivery           -0.004903   0.034257  -0.143 0.886188
fruits                   0.059089   0.033321   1.773 0.076237 .
groceries                0.077694   0.031601   2.459 0.013981 *
gym_memberships          0.141168   0.033410   4.225 2.43e-05 ***
jewelry                  0.213726   0.032834   6.509 8.30e-11 ***
laptops                  0.594328   0.032548  18.260  < 2e-16 ***
magazines                0.080762   0.033694   2.397 0.016571 *
meat                     0.081262   0.032367   2.511 0.012083 *
movies                   0.110296   0.033326   3.310 0.000941 ***
music                    0.159925   0.033398   4.788 1.73e-06 ***
outdoor_activities       0.087846   0.032356   2.715 0.006651 **
public_transportation    0.061138   0.033022   1.851 0.064169 .
restaurant_meals         0.066129   0.033225   1.990 0.046611 *
seafood                  0.061318   0.033786   1.815 0.069596 .
shoes                    0.463185   0.029613  15.641  < 2e-16 ***
smartphones              0.780150   0.031538  24.737  < 2e-16 ***
snacks                   0.007464   0.033229   0.225 0.822290
software                 0.408500   0.034102  11.979  < 2e-16 ***
sports_equipment         0.033328   0.033969   0.981 0.326574
streaming_services       0.150614   0.031902   4.721 2.41e-06 ***
tablets                  0.637266   0.033133  19.234  < 2e-16 ***
travel                   0.129161   0.031457   4.106 4.09e-05 ***
vegetables              -0.066111   0.033162  -1.994 0.046257 *
video_games              0.863309   0.031392  27.501  < 2e-16 ***
watches                  0.145853   0.033467   4.358 1.34e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.434 on 4960 degrees of freedom
Multiple R-squared:  0.9999,    Adjusted R-squared:  0.9999
F-statistic: 1.834e+06 on 39 and 4960 DF,  p-value: < 2.2e-16
```

```
summary(model_pca)
```

```
Call:
lm(formula = income ~ ., data = df_pca)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-44.345 -18.599  -0.293  18.730  47.846

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 628.17783    0.30371  2068.4   <2e-16 ***
PC1          13.33571    0.08390   159.0   <2e-16 ***
PC2          -1.16303    0.08808   -13.2   <2e-16 ***
PC3          95.58547    0.10144   942.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.48 on 4996 degrees of freedom
Multiple R-squared:  0.9946,    Adjusted R-squared:  0.9946
F-statistic: 3.044e+05 on 3 and 4996 DF,  p-value: < 2.2e-16
```

```
metrics <- rbind(
  data.frame(Model = "Original", RMSE = sqrt(mean(model$residuals^2)), R2 =
summary(model)$r.squared),
  data.frame(Model = "PCA", RMSE = sqrt(mean(model_pca$residuals^2)), R2 =
summary(model_pca)$r.squared)
)

print(metrics)
```

```
     Model      RMSE        R2
1 Original  2.423783 0.9999306
2      PCA 21.466895 0.9945598
```

1.10 (10 points)

Based on your interpretation of the principal components from Question 1.7, provide an interpretation of the regression model in Question 1.9.

—

ⓘ Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.3.3 (2024-02-29 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

other attached packages:
 [1] reshape2_1.4.4 ggplot2_3.5.0  janitor_2.2.0  car_3.1-2      carData_3.0-
5
 [6] corrplot_0.92  magrittr_2.0.3 broom_1.0.5    purrr_1.0.2    tidyr_1.3.1
[11] readr_2.1.5    dplyr_1.1.4    tibble_3.2.1

loaded via a namespace (and not attached):
 [1] utf8_1.2.4        generics_0.1.3    renv_1.0.3        stringi_1.8.3
 [5] hms_1.1.3         digest_0.6.35     evaluate_0.23     grid_4.3.3
 [9] timechange_0.3.0  fastmap_1.1.1     plyr_1.8.9        jsonlite_1.8.8
[13] backports_1.4.1   fansi_1.0.6       scales_1.3.0      codetools_0.2-19
[17] abind_1.4-5       cli_3.6.2         crayon_1.5.2      rlang_1.1.3
[21] bit64_4.0.5       munsell_0.5.1     withr_3.0.0       yaml_2.3.8
[25] parallel_4.3.3    tools_4.3.3       tzdb_0.4.0        colorspace_2.1-0
[29] vctrs_0.6.5       R6_2.5.1          lifecycle_1.0.4   lubridate_1.9.3
[33] snakecase_0.11.1  stringr_1.5.1     bit_4.0.5         vroom_1.6.5
[37] pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.4      glue_1.7.0
[41] Rcpp_1.0.12       xfun_0.43         tidyselect_1.2.1  knitr_1.46
[45] farver_2.1.1      htmltools_0.5.8.1 labeling_0.4.3    rmarkdown_2.26
[49] compiler_4.3.3
```