Homework 6

Jacob Adams

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"
```

```
)
# 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)
```

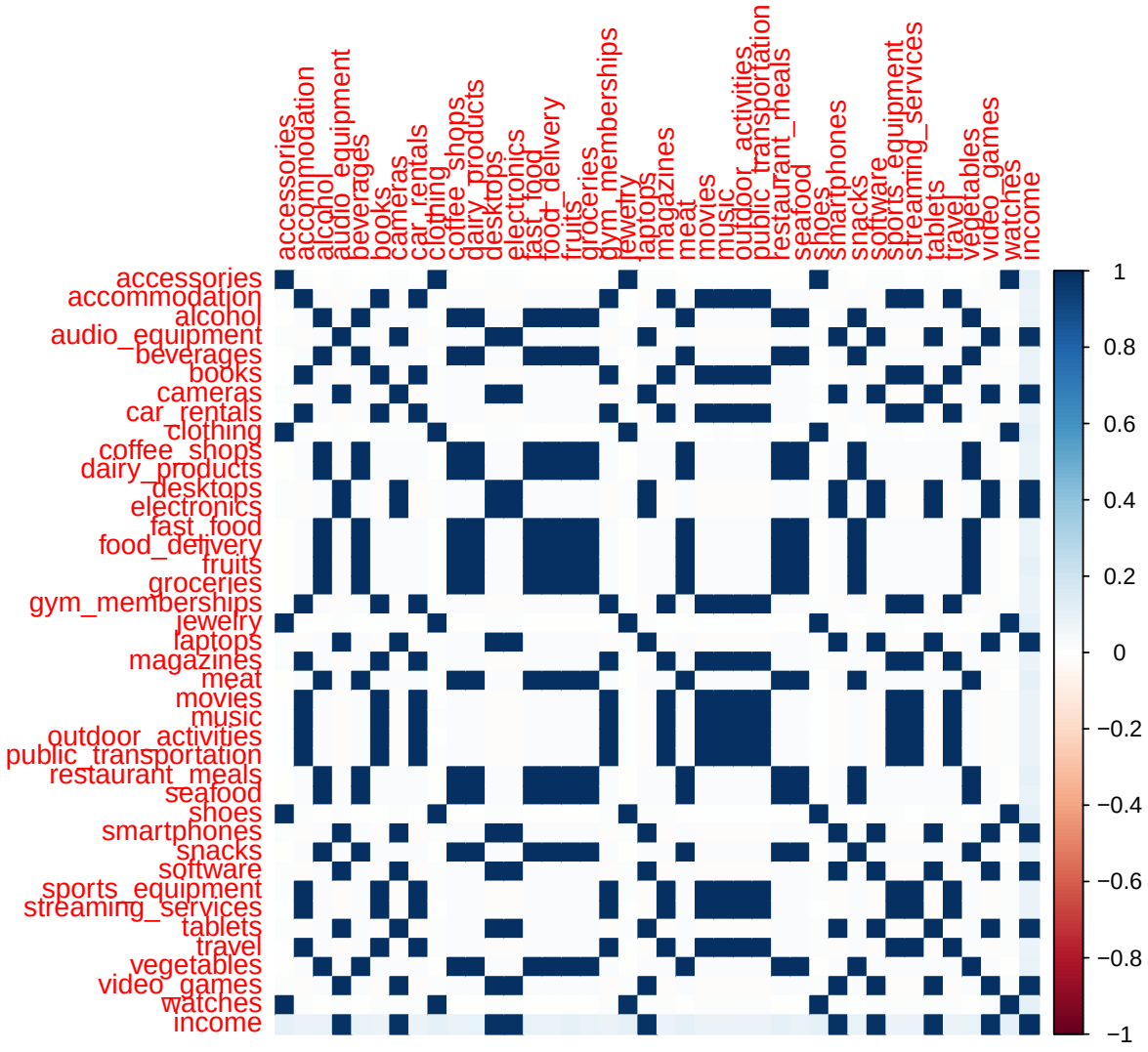
```
Rows: 5000 Columns: 40
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (40): accessories, accommodation, alcohol, audio_equipment, beverages, b...

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.
```

```
df <- df %>%
  rename_all(tolower) %>%
  drop_na()
df <- as_tibble(df)
```

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?

```
corrlelation_matrix <- cor(df)
corrplot(corrlelation_matrix, method = "color")
```



There is a large amount of collinearity between about half of all the predictor variables. As a result, this is very troublesome for the model, since collinearity often leads to inconsistent models.

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.

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

```
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
```

##Interpretation of coefficientients: Accessories: Each unit increase in spending on accessories leads to an increase of 0.299876 in the response to income.

Accommodation: Each unit increase in spending on accommodation leads to an increase of 0.113632 in the response to income.

Alcohol: Spending on alcohol does not significantly affect the response to income.

Audio Equipment: Each unit increase in spending on audio equipment leads to an increase of 0.602004 in the response to income.

Beverages: Spending on beverages does not significantly affect the response to income.

Books: Each unit increase in spending on books leads to an increase of 0.070530 in the response to income.

Cameras: Each unit increase in spending on cameras leads to an increase of 0.461827 in the response to income.

Car Rentals: Each unit increase in spending on car rentals leads to an increase of 0.124875 in the response to income.

Clothing: Each unit increase in spending on clothing leads to an increase of 0.504228 in the response to income.

Coffee Shops: Spending on coffee shops does not significantly affect the response to income.

Dairy Products: Spending on dairy products does not significantly affect the response to income.

Desktops: Each unit increase in spending on desktops leads to an increase of 0.391673 in the response to income.

Electronics: Each unit increase in spending on electronics leads to an increase of 1.079627 in the response to income.

Fast Food: Each unit increase in spending on fast food leads to an increase of 0.077531 in the response to income.

Food Delivery: Spending on food delivery does not significantly affect the response to income.

Fruits: Spending on fruits does not significantly affect the response to income.

Groceries: Each unit increase in spending on groceries leads to an increase of 0.077694 in the response to income.

Gym Memberships: Each unit increase in spending on gym memberships leads to an increase of 0.141168 in the response to income.

Jewelry: Each unit increase in spending on jewelry leads to an increase of 0.213726 in the response to income.

Laptops: Each unit increase in spending on laptops leads to an increase of 0.594328 in the response to income.

Magazines: Each unit increase in spending on magazines leads to an increase of 0.080762 in the response to income.

Meat: Each unit increase in spending on meat leads to an increase of 0.081262 in the response to income.

Movies: Each unit increase in spending on movies leads to an increase of 0.110296 in the response to income.

Music: Each unit increase in spending on music leads to an increase of 0.159925 in the response to income.

Outdoor Activities: Each unit increase in spending on outdoor activities leads to an increase of 0.087846 in the response to income.

Public Transportation: Spending on public transportation does not significantly affect the response to income.

Restaurant Meals: Spending on restaurant meals does not significantly affect the response to income.

Seafood: Spending on seafood does not significantly affect the response to income.

Shoes: Each unit increase in spending on shoes leads to an increase of 0.463185 in the response to income.

Smartphones: Each unit increase in spending on smartphones leads to an increase of 0.780150 in the response to income.

Snacks: Spending on snacks does not significantly affect the response to income.

Software: Each unit increase in spending on software leads to an increase of 0.408500 in the response to income.

Sports Equipment: Spending on sports equipment does not significantly affect the response to income.

Streaming Services: Each unit increase in spending on streaming services leads to an increase of 0.150614 in the response to income.

Tablets: Each unit increase in spending on tablets leads to an increase of 0.637266 in the response to income.

Travel: Each unit increase in spending on travel leads to an increase of 0.129161 in the response to income.

Vegetables: Spending on vegetables does not significantly affect the response to income.

Video Games: Each unit increase in spending on video games leads to an increase of 0.863309 in the response to income.

Watches: Each unit increase in spending on watches leads to an increase of 0.145853 in the response to income.

Non-significant predictors (p-value > 0.05):

Alcohol Beverages Coffee Shops Dairy Products Food Delivery Fruits Public Transportation Seafood Snacks Sports Equipment —

1.3 (5 points)

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

```
print(vif(lm_fit))
```

```
          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
```

According to the rules of variance inflation factor, any values of "vif" above 5 are considered to have a high degree of multi-collinearity. Thus, we can see this model has an enormous amount of collinearity which will lead to an inneffective model.

1.4 (5 points)

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

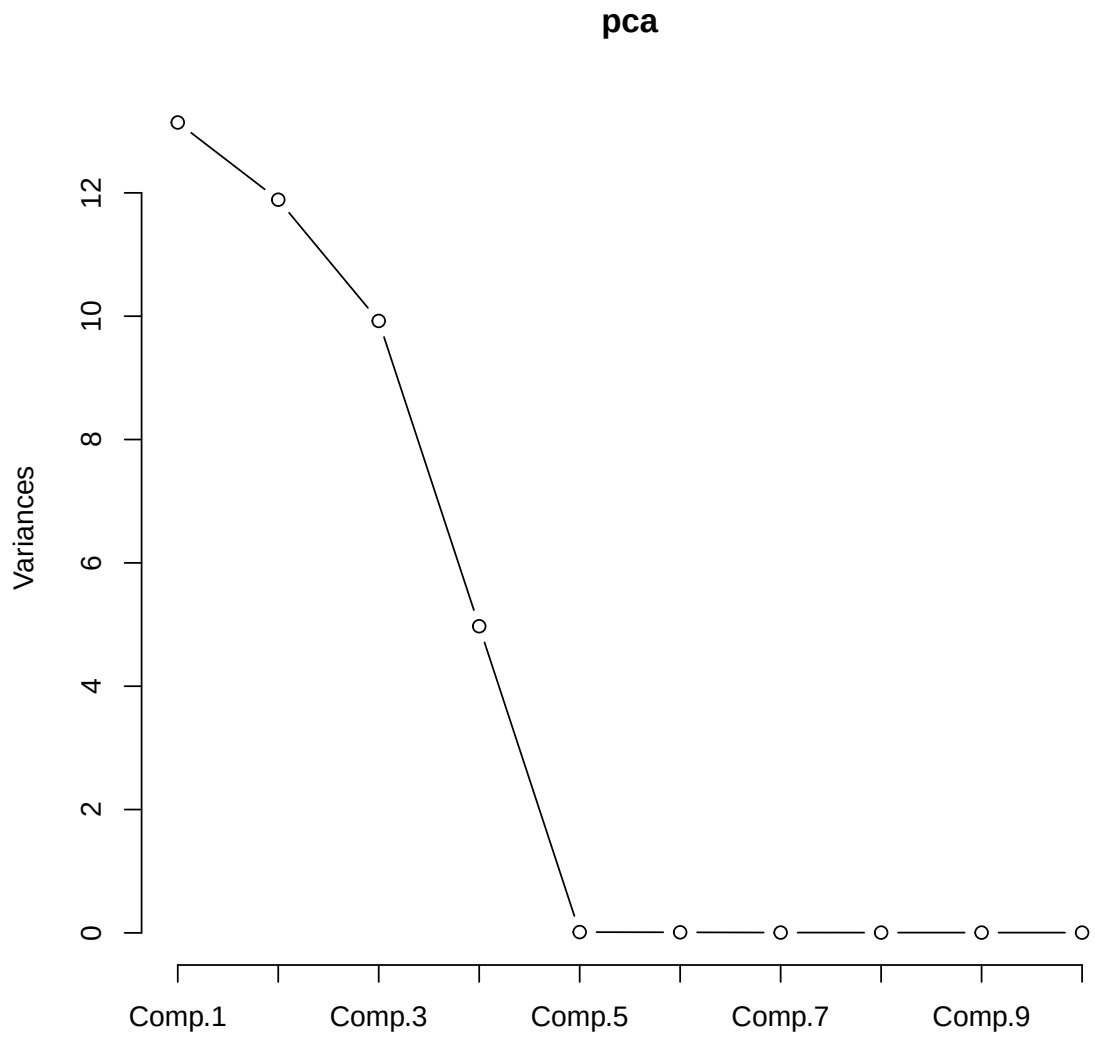
```
pca <- princomp(df, cor = TRUE)
summary(pca)
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4       Comp.5
Standard deviation     3.6250565 3.4480375 3.1501541 2.2298341 0.1125697797
Proportion of Variance 0.3285259 0.2972241 0.2480868 0.1243040 0.0003167989
Cumulative Proportion  0.3285259 0.6257499 0.8738367 0.9981407 0.9984574993
                            Comp.6       Comp.7       Comp.8       Comp.9
Standard deviation     0.0960777376 0.070831618 0.0691545366 0.0670347877
Proportion of Variance 0.0002307733 0.000125428 0.0001195587 0.0001123416
Cumulative Proportion  0.9986882726 0.998813701 0.9989332593 0.9990456008
                           Comp.10       Comp.11       Comp.12       Comp.13
Standard deviation     0.0653221903 5.099557e-02 4.981058e-02 4.762384e-02
Proportion of Variance 0.0001066747 6.501371e-05 6.202734e-05 5.670076e-05
Cumulative Proportion  0.9991522756 9.992173e-01 9.992793e-01 9.993360e-01
                           Comp.14       Comp.15       Comp.16       Comp.17
Standard deviation     4.698663e-02 4.611215e-02 4.590277e-02 4.552849e-02
Proportion of Variance 5.519359e-05 5.315825e-05 5.267662e-05 5.182109e-05
Cumulative Proportion  9.993912e-01 9.994444e-01 9.994970e-01 9.995489e-01
                           Comp.18       Comp.19       Comp.20       Comp.21
Standard deviation     4.516755e-02 3.944182e-02 3.586495e-02 0.0350545840
Proportion of Variance 5.100269e-05 3.889143e-05 3.215737e-05 0.0000307206
Cumulative Proportion  9.995999e-01 9.996388e-01 9.996709e-01 0.9997016390
                           Comp.22       Comp.23       Comp.24       Comp.25
Standard deviation     3.460923e-02 3.435492e-02 3.298824e-02 3.240375e-02
Proportion of Variance 2.994497e-05 2.950651e-05 2.720561e-05 2.625007e-05
Cumulative Proportion  9.997316e-01 9.997611e-01 9.997883e-01 9.998145e-01
                           Comp.26       Comp.27      Comp.28       Comp.29
Standard deviation     3.157795e-02 2.977572e-02 0.025086252 2.460209e-02
Proportion of Variance 2.492918e-05 2.216484e-05 0.000015733 1.513158e-05
Cumulative Proportion  9.998395e-01 9.998616e-01 0.999877373 9.998925e-01
                           Comp.30       Comp.31       Comp.32       Comp.33
Standard deviation     2.426620e-02 2.374610e-02 2.334392e-02 2.283050e-02
Proportion of Variance 1.472121e-05 1.409693e-05 1.362347e-05 1.303079e-05
Cumulative Proportion  9.999072e-01 9.999213e-01 9.999349e-01 9.999480e-01
                           Comp.34       Comp.35       Comp.36       Comp.37
Standard deviation     2.119155e-02 1.983574e-02 1.937935e-02 1.742907e-02
Proportion of Variance 1.122705e-05 9.836418e-06 9.388978e-06 7.594316e-06
Cumulative Proportion  9.999592e-01 9.999690e-01 9.999784e-01 9.999860e-01
                           Comp.38       Comp.39       Comp.40
Standard deviation     1.677847e-02 1.476855e-02 7.708148e-03
Proportion of Variance 7.037927e-06 5.452751e-06 1.485389e-06
Cumulative Proportion  9.999931e-01 9.999985e-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?

```
screeplot(pca, type = "l")
```



I would keep 4 principal components based off of the screeplot. This is because only 4 components seem to explain the high degree of variance before it levels off.

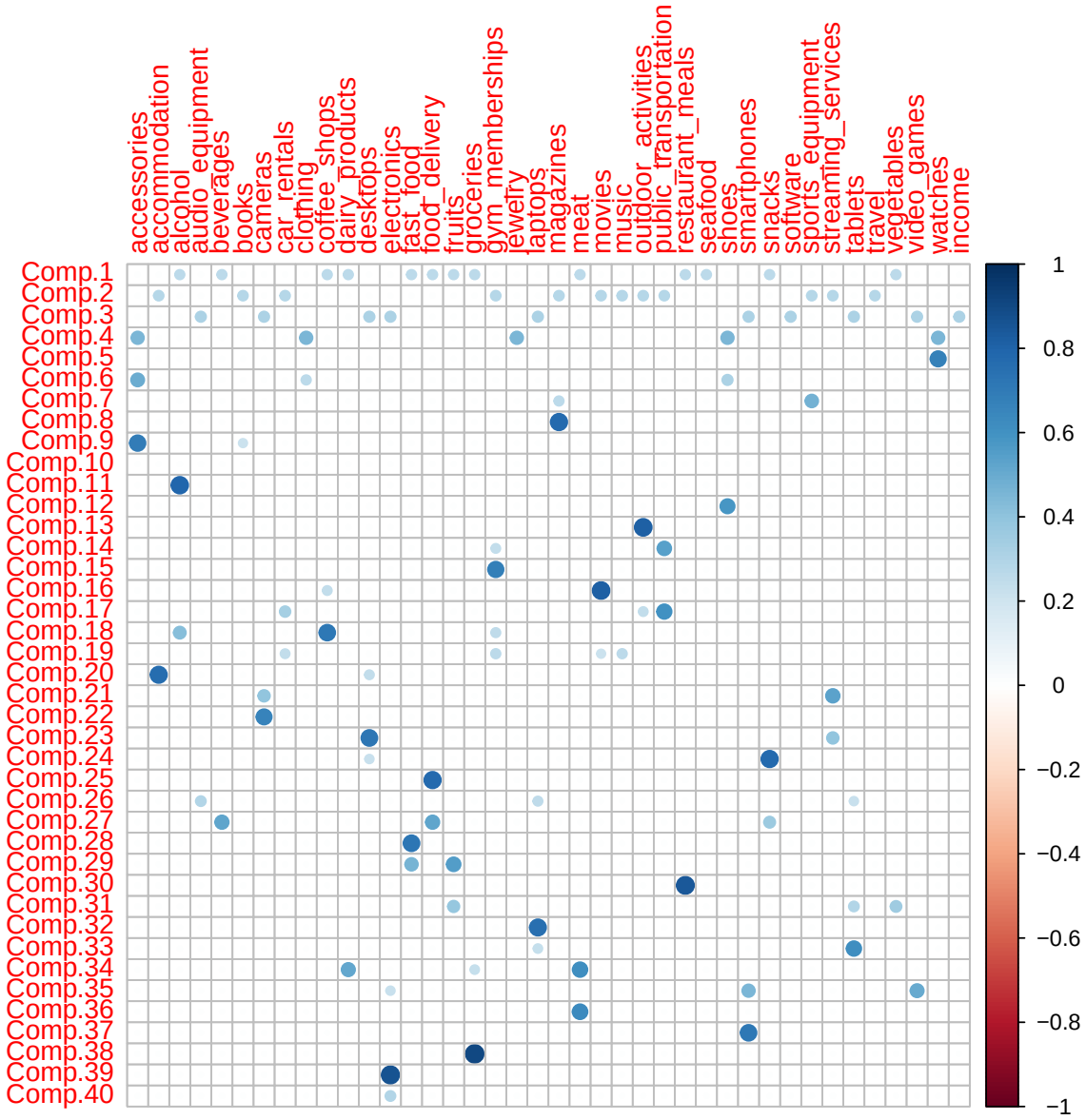
1.6 (5 points)

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

```
loadings <- pca$loadings
clean_loading <- ifelse(pca$loadings[, 1:40] < 0.2, 0, round(pca$loadings[, 1:40], 2)) %>% as
View(clean_loading)
```

Visualize the factor loadings.

```
clean_loading %>% as.matrix() %>% t() %>% corrplot()
```



1.7 (15 points)

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

Based off the corrplot above I would keep components 1 and 2. These principal components represent a strong correlation between the variables and response variable that may be present in the data.

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

Choosing to keep the principal components 1 and 2. We see in component one that alcohol, beverages, coffe_shop, dairy_products, fast_food, food_delivery, fruits, and groceries, meat, restaurant_meals, seafood, snacks, and vegetables all seem to be valid predictors. In component two we see accommodation, audio_equipment, books, car_rentals, gym_memberships, magazines, movies, music, outdoor_activities, public_transportation, sports_equipment, streaming_services, and travel seem to be valid predictors as well. The other components have a stronger correlation, but far less predictors. In the context of this data set, we should probably choose the larger models.

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`.

```
Z <- predict(pca, df)

df_pca <- Z %>% as_tibble %>% select(Comp.1, Comp.2) %>% mutate(income = df$income)
```

1.9 (10 points)

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

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

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

Residuals:
    Min      1Q  Median      3Q     Max
-584.10 -239.08    1.87  236.97  611.89

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  628.178      4.019 156.286   <2e-16 ***
Comp.1        17.324      1.109  15.625   <2e-16 ***
Comp.2        -1.682      1.166  -1.443    0.149
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 284.2 on 4997 degrees of freedom
Multiple R-squared:  0.04696,   Adjusted R-squared:  0.04658
F-statistic: 123.1 on 2 and 4997 DF,  p-value: < 2.2e-16
```

Interpretation:

Comp.1: For every one unit increase in component one variables, there is a 17.324 increase in income. Comp.2: For every one unit increase in component two variables, there is a 1.682 decrease in income

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

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.

Examining the model we can see there is a high degree of error in the model. This may be because of a previous error where I decided to take more predictors over highly correlated components with less predictors. Component one is much more effective in predicting the income compared to component two. For example, it is a significant predictor where component two is not. Overall, it is still a better model than the original regression model.

—

Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 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 utils     datasets  methods   base

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

loaded via a namespace (and not attached):
 [1] bit_4.0.5         jsonlite_1.8.7    compiler_4.3.1    crayon_1.5.2
 [5] tidyselect_1.2.0  parallel_4.3.1    yaml_2.3.7        fastmap_1.1.1
 [9] R6_2.5.1          generics_0.1.3    knitr_1.43        backports_1.4.1
[13] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.1       utf8_1.2.3
[17] xfun_0.40         bit64_4.0.5       cli_3.6.1         withr_2.5.0
[21] digest_0.6.33     vroom_1.6.3       rstudioapi_0.15.0 hms_1.1.3
[25] lifecycle_1.0.3   vctrs_0.6.5       evaluate_0.21     glue_1.6.2
[29] codetools_0.2-19  abind_1.4-5       fansi_1.0.4       rmarkdown_2.24
[33] tools_4.3.1       pkgconfig_2.0.3   htmltools_0.5.6
```