Ćwiczenia 8

2023-12-18

Zadanie 1

```
df <- MASS::painters
var(df[,1:4])
```

```
##             Composition   Drawing    Colour Expression
## Composition   16.704403  5.870021 -1.855346  12.886792
## Drawing        5.870021 11.951433 -8.313417   9.515723
## Colour        -1.855346 -8.313417 21.638365  -4.452830
## Expression    12.886792  9.515723 -4.452830  23.018868
```

```
(model_pca <- prcomp(~ Composition + Drawing + Colour + Expression,
                     data = df))
```

```
## Standard deviations (1, .., p=4):
## [1] 6.404806 4.573288 2.584339 2.167431
##
## Rotation (n x k) = (4 x 4):
##                    PC1        PC2        PC3        PC4
## Composition  0.4835097 -0.3764049 -0.7838341 -0.1007070
## Drawing      0.4240128  0.1871745  0.2796965 -0.8408024
## Colour      -0.3807729 -0.8452384  0.2108460 -0.3100452
## Expression   0.6644122 -0.3299350  0.5127553  0.4321821
```

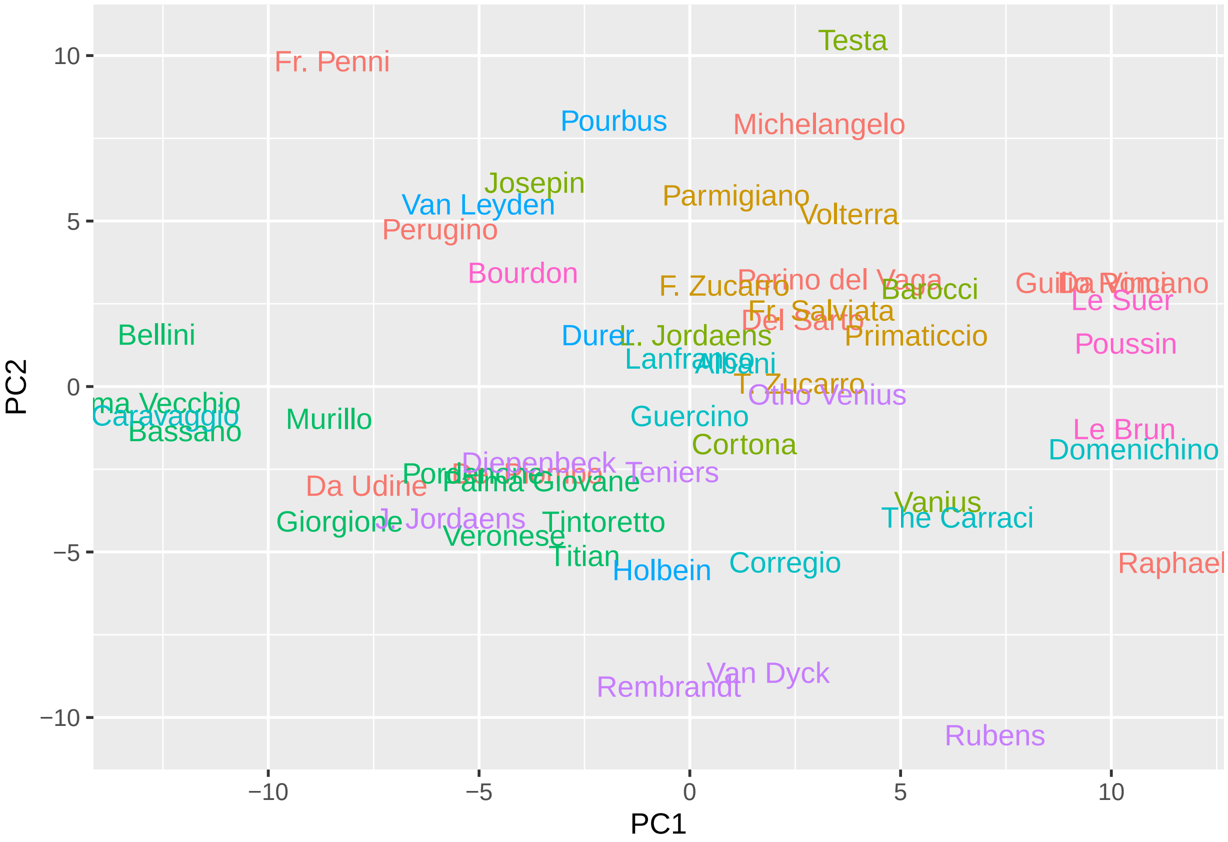
```
summary(model_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     6.4048 4.5733 2.5843 2.16743
## Proportion of Variance 0.5595 0.2853 0.0911 0.06408
## Cumulative Proportion  0.5595 0.8448 0.9359 1.00000
```

```
df_new <- data.frame(
  predict(model_pca, df),
  names = rownames(df)
)

df_new |>
  left_join(
    df |>
      as_tibble(rownames = "names") |>
      select(names, School)
  ) |>
  ggplot(aes(x = PC1, y = PC2, colour = School, group = School)) +
  geom_text(aes(label = names), show.legend = FALSE)
```

```
## Joining with `by = join_by(names)`
```



Zadanie 2

```
df <- MASS::Cars93

df <- df |>
  select(where(is.numeric)) |>
  na.omit()

var(df) |>
  diag()
```

```
##        Min.Price            Price        Max.Price         MPG.city
##     7.960125e+01     9.918802e+01     1.310186e+02     3.128892e+01
##      MPG.highway       EngineSize       Horsepower              RPM
##     2.511051e+01     1.008150e+00     2.606615e+03     3.407468e+05
##     Rev.per.mile Fuel.tank.capacity       Passengers           Length
##     2.472901e+05     9.061670e+00     5.024089e-01     2.332955e+02
##        Wheelbase            Width      Turn.circle   Rear.seat.room
##     4.182069e+01     1.364469e+01     1.001581e+01     8.072719e+00
##     Luggage.room           Weight
##     8.987805e+00     3.202836e+05
```

```
(model_pca <- prcomp(df, scale. = TRUE))
```

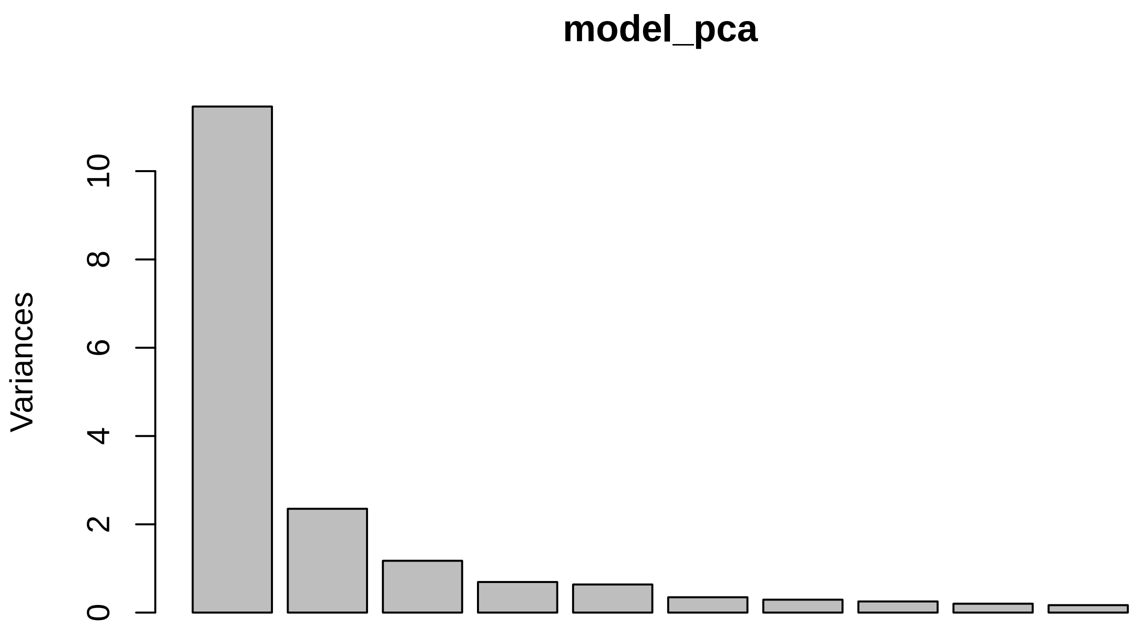
```
## Standard deviations (1, .., p=18):
##  [1] 3.38583089 1.53362511 1.08359582 0.83232398 0.79803716 0.58960668
##  [7] 0.54108330 0.50227972 0.44874332 0.41071188 0.33871859 0.27442422
## [13] 0.26738475 0.24440238 0.20229092 0.18692450 0.14303038 0.00193893
```

```
##
## Rotation (n x k) = (18 x 18):
##                           PC1         PC2         PC3         PC4          PC5
## Min.Price           0.2365632  0.31357211  0.19217662  0.10628571  0.223650270
## Price               0.2235887  0.36638202  0.20093623  0.15007578  0.252363864
## Max.Price           0.2046061  0.39311451  0.19946984  0.17783473  0.264652954
## MPG.city           -0.2508675 -0.07549828  0.23745363  0.47174532 -0.124901436
## MPG.highway        -0.2294848 -0.14869274  0.22446421  0.59029226 -0.167185181
## EngineSize          0.2774693 -0.06340695 -0.11075087  0.17992672  0.016958339
## Horsepower          0.2393367  0.28765236 -0.09308763  0.08081898 -0.316326750
## RPM                -0.1226287  0.40729181  0.24500114 -0.21188604 -0.692825544
## Rev.per.mile       -0.2384668  0.14749521  0.24971475 -0.02385120  0.006080179
## Fuel.tank.capacity  0.2649115  0.09298933 -0.03533988 -0.04934373 -0.126991580
## Passengers          0.1809153 -0.29782294  0.38998149 -0.40823463  0.015609265
## Length              0.2690792 -0.13994120  0.00240815  0.04710825 -0.184403877
## Wheelbase           0.2705009 -0.10364357  0.13692512  0.14411962 -0.131536108
## Width               0.2604737 -0.13710791 -0.19431745  0.20341267 -0.210021545
## Turn.circle         0.2387195 -0.16165683 -0.19382974  0.11745847 -0.226217345
## Rear.seat.room      0.1728066 -0.23548324  0.57357217 -0.12410365 -0.084326585
## Luggage.room        0.2117860 -0.28317377  0.22052048  0.12648759  0.108473092
## Weight              0.2870144  0.04939767 -0.08825964  0.02330044 -0.102315329
##                             PC6          PC7         PC8          PC9
## Min.Price          -0.118628778 -0.036351038 -0.07281125 -0.048920335
## Price              -0.082373196 -0.050054203  0.06925714  0.109903128
## Max.Price          -0.050375360 -0.059021657  0.18034937  0.228946353
## MPG.city           -0.094589040 -0.017339260  0.08252648 -0.079552540
## MPG.highway        -0.032489716  0.042917474  0.11432599  0.009599702
## EngineSize         -0.079081895  0.142134318 -0.07742345 -0.287001582
## Horsepower         -0.044618452  0.158232222 -0.07775345 -0.084554838
## RPM                 0.056148256  0.171686007  0.08734200  0.233198555
## Rev.per.mile        0.392418478 -0.685683754 -0.10684765 -0.209414982
## Fuel.tank.capacity  0.465065190 -0.094845883 -0.15912464 -0.268096103
## Passengers         -0.153348641 -0.121870286  0.54755520  0.054008225
## Length              0.037324121 -0.235363311  0.22801956 -0.079930348
## Wheelbase           0.005542024 -0.047342604  0.18310747 -0.222849059
## Width               0.074162188 -0.066646481  0.25454579 -0.107271765
## Turn.circle        -0.287611343 -0.515944617 -0.31027988  0.557678305
## Rear.seat.room     -0.258256417  0.102005517 -0.57024795 -0.199902804
## Luggage.room        0.635271510  0.288174445 -0.09650183  0.484919192
## Weight              0.038730603  0.009675358 -0.01374334 -0.102333921
##                           PC10         PC11        PC12        PC13        PC14
## Min.Price           0.04087067  0.165334930  0.36092681  0.27139603  0.49127117
## Price               0.03787475 -0.021550315 -0.03714548  0.07902704  0.01884847
## Max.Price           0.03377878 -0.163366312 -0.34485529 -0.07517967 -0.34910532
## MPG.city           -0.18649032 -0.112577331  0.07366452 -0.01019728  0.05463064
## MPG.highway        -0.01959230 -0.225367682  0.09037666  0.06534624 -0.08047188
## EngineSize         -0.34260922  0.090124280  0.23525890  0.03514347  0.16057320
## Horsepower         -0.47610975  0.190869554  0.07773163 -0.21801530 -0.32656933
## RPM                 0.11867016  0.039321008  0.02533825  0.09601253  0.19393839
## Rev.per.mile       -0.20359883  0.323176041 -0.02112680 -0.09742811 -0.03440289
## Fuel.tank.capacity -0.02636424 -0.724521960 -0.03256203  0.14844967  0.10410733
## Passengers         -0.41396039 -0.170369304  0.08068325 -0.01457985  0.06090916
## Length              0.43418211  0.134114514  0.43651216  0.32623010 -0.45860601
## Wheelbase           0.41581116  0.073495089 -0.10796393 -0.66293470  0.32009617
## Width              -0.07593601  0.291989605 -0.61741481  0.42773866  0.17221293
## Turn.circle        -0.08486897 -0.140066087  0.02987431 -0.07937277  0.14817042
## Rear.seat.room      0.08965264  0.056422947 -0.24923024  0.13080204 -0.17394372
## Luggage.room       -0.07549012  0.214693046  0.09295886 -0.04505816  0.02365900
## Weight             -0.02860522 -0.002363461  0.10361182 -0.25697911 -0.21442585
##                            PC15         PC16         PC17          PC18
## Min.Price          -0.145192595 -0.297292548 -0.095095533  3.627524e-01
## Price               0.001836030  0.004007958  0.006679143 -8.081320e-01
## Max.Price           0.119707541  0.242291382  0.092459569  4.640196e-01
## MPG.city           -0.573651555  0.419269052 -0.231558714  9.905556e-04
## MPG.highway         0.369961515 -0.474888433  0.223401583 -1.416659e-04
## EngineSize          0.389482934  0.514757686  0.367006061 -2.836525e-03
## Horsepower          0.076952448 -0.232966035 -0.470065761  1.517246e-03
## RPM                 0.057585379  0.170615470  0.209713222 -1.248060e-03
## Rev.per.mile        0.095163835  0.007444787  0.113401457 -1.043012e-03
## Fuel.tank.capacity -0.009552335 -0.005689988 -0.138555716  1.549858e-03
## Passengers          0.004377548 -0.087382391 -0.002895105 -5.097950e-04
## Length              0.030620146  0.150669008 -0.138507356 -2.059213e-04
## Wheelbase           0.120615243  0.025898897 -0.163354922 -4.815855e-04
## Width              -0.100086229 -0.093458920  0.016507987 -1.012727e-04
## Turn.circle         0.027818095  0.036619177 -0.007627861  5.167766e-04
## Rear.seat.room     -0.007599741 -0.009596261  0.009426879  1.134059e-03
## Luggage.room       -0.034021351  0.042927652 -0.038306870 -8.891512e-05
## Weight             -0.548675679 -0.245206692  0.631964232 -8.171935e-04
```

```
print(summary(model_pca))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.3858 1.5336 1.08360 0.83232 0.79804 0.58961 0.54108
## Proportion of Variance 0.6369 0.1307 0.06523 0.03849 0.03538 0.01931 0.01627
## Cumulative Proportion  0.6369 0.7675 0.83278 0.87127 0.90665 0.92596 0.94223
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.50228 0.44874 0.41071 0.33872 0.27442 0.26738 0.24440
## Proportion of Variance 0.01402 0.01119 0.00937 0.00637 0.00418 0.00397 0.00332
## Cumulative Proportion  0.95624 0.96743 0.97680 0.98317 0.98736 0.99133 0.99465
##                           PC15    PC16    PC17     PC18
## Standard deviation     0.20229 0.18692 0.14303 0.001939
## Proportion of Variance 0.00227 0.00194 0.00114 0.000000
## Cumulative Proportion  0.99692 0.99886 1.00000 1.000000
```

```
plot(model_pca)
```



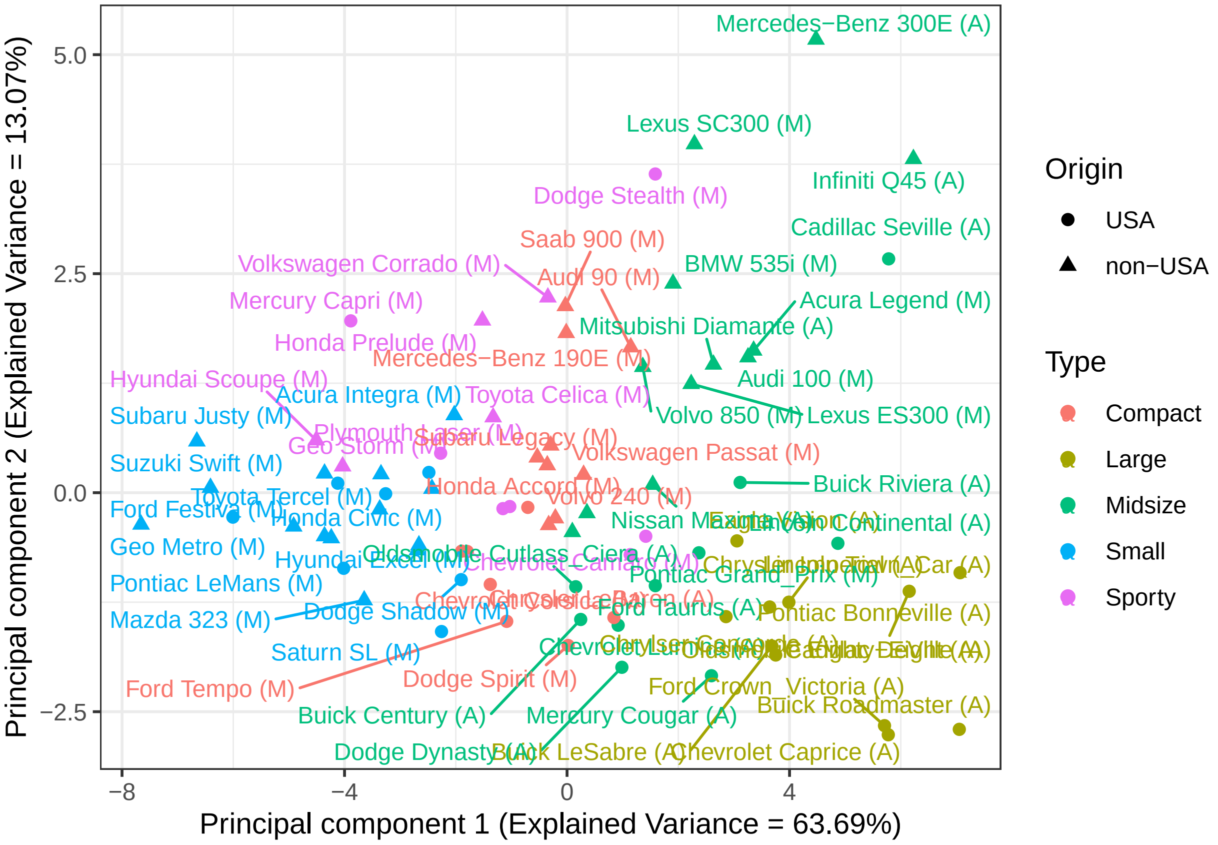
```
df <- cbind(model_pca$x,
            df) |>
  left_join(MASS::Cars93)
```

```
## Joining with `by = join_by(Min.Price, Price, Max.Price, MPG.city, MPG.highway,
## EngineSize, Horsepower, RPM, Rev.per.mile, Fuel.tank.capacity, Passengers,
## Length, Wheelbase, Width, Turn.circle, Rear.seat.room, Luggage.room, Weight)`
```

```
model_sd <- (model_pca |> summary())$importance[2, 1:2]

df |>
  select(PC1, PC2, Origin, Type, Man.trans.avail, Make) |>
  transform("Make" = paste0(
    Make, " (", ifelse(Man.trans.avail == "Yes", "M", "A"), ")"
  )) |>
  ggplot(aes(x = PC1,
             y = PC2,
             col   = Type,
             shape = Origin,
             label = Make)) +
  geom_point(size = 2) +
  ggrepel::geom_text_repel(size = 3.2, max.overlaps = 20) +
  theme_bw() +
  xlab(paste0("Principal component 1 (Explained Variance = ", round(model_sd[1] * 100, 2), "%)")) +
  ylab(paste0("Principal component 2 (Explained Variance = ", round(model_sd[2] * 100, 2), "%)"))
```

```
## Warning: ggrepel: 19 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



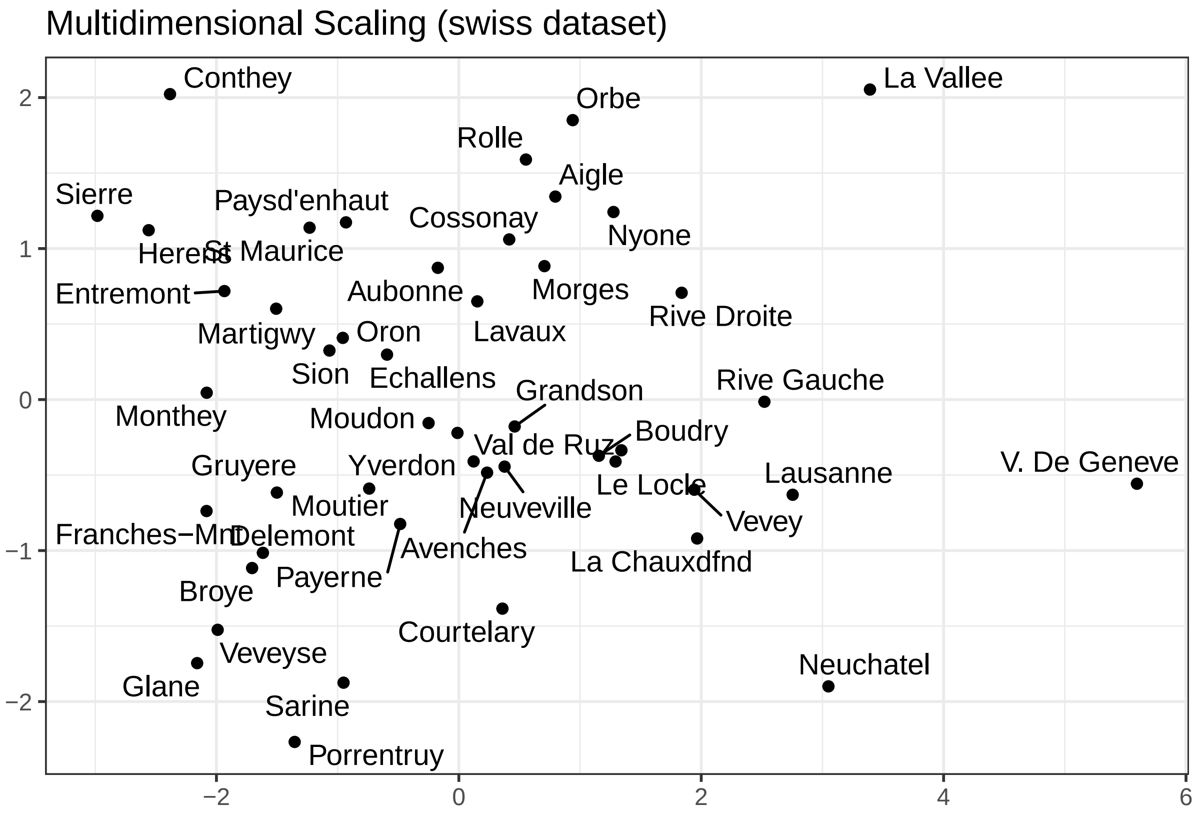
Zadanie 3

```
df <- swiss
mds_model <- cmdscale(dist(scale(df)))
```

```
swiss |>
  scale() |>
  dist() |>
  cmdscale() |>
  as_tibble(rownames = "province") |>
  ggplot(aes(x = V1, y = V2, label = province)) +
  geom_point() +
  ggrepel::geom_text_repel() +
  labs(x = '', y = '', title = 'Multidimensional Scaling (swiss dataset)') +
  theme_bw()
```

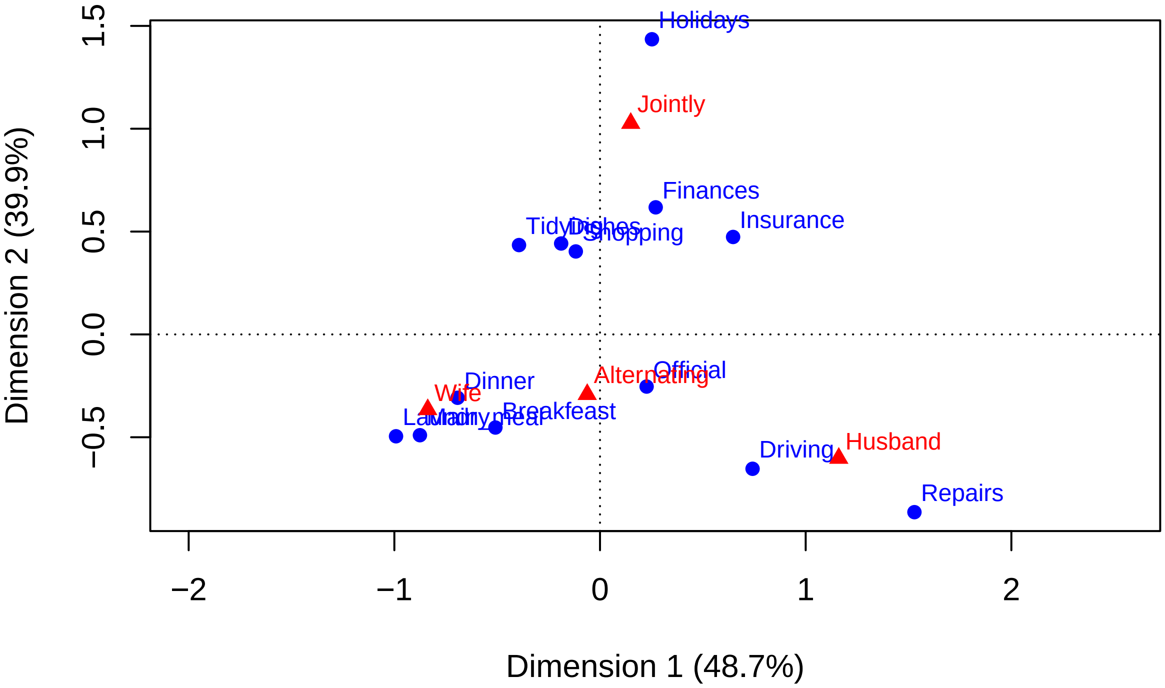
```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



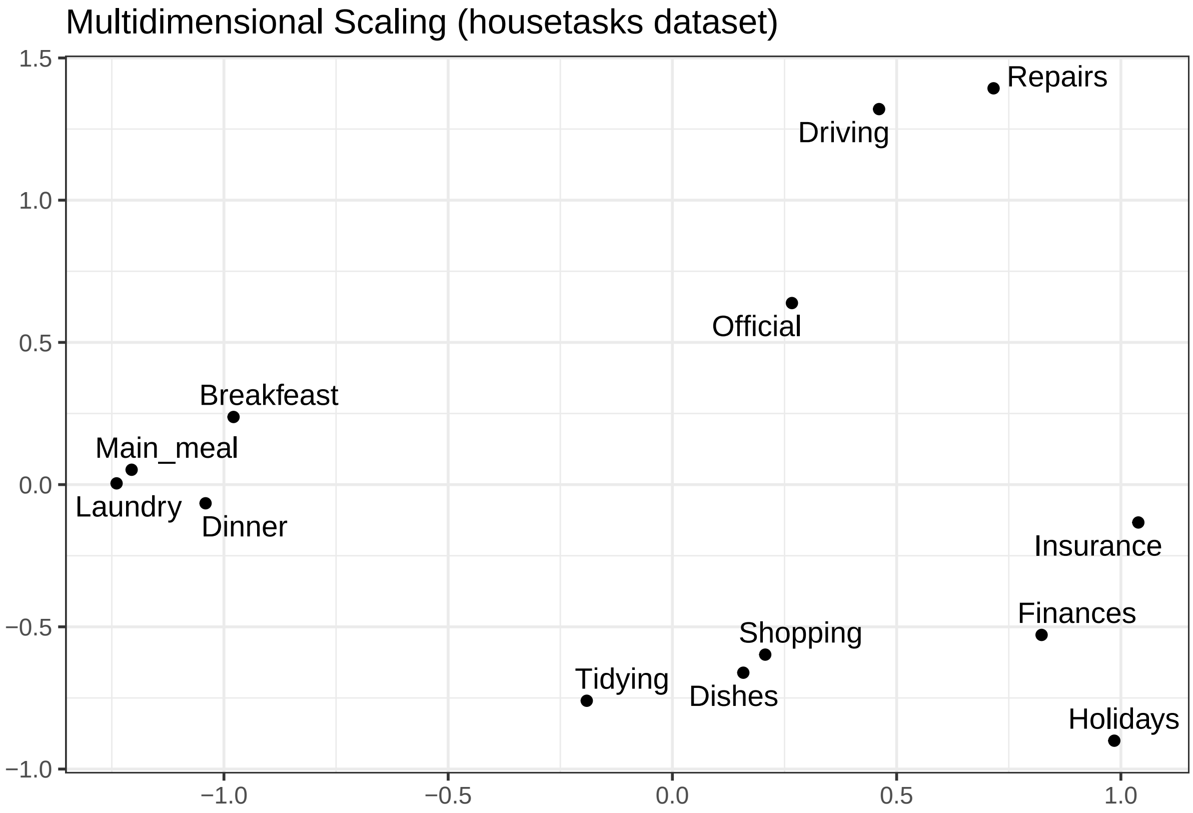
Zadanie 4

```
plot(ca::ca(factoextra::housetasks))
```



```
factoextra::housetasks |>
  t() |>
```

```
scale(center = FALSE) |>
t() |>
dist() |>
cmdscale() |>
as_tibble(rownames = "what") |>
ggplot(aes(x = V1, y = V2, label = what)) +
geom_point() +
ggrepel::geom_text_repel() +
labs(x = '', y = '', title = 'Multidimensional Scaling (housetasks dataset)') +
theme_bw()
```


Multidimensional Scaling (housetasks dataset)
Repairs
Driving
Official
Breakfeast
Main_meal
Laundry
Dinner
Insurance
Finances
Shopping
Tidying
Dishes
Holidays