Ordination Analysis

Nadia Ahmad

2023-04-09

Cluster Analysis on Travel Discrimination

Dataset explanation:
Variables:
Continuos
- Q1_ = Travel frequency
- Q6_15 : checkin experience rate
- Q6_18 : fly experience rate

Categorical
- Q15 = Gender
- Q17 = Race
- Q18 = Religion

Library

```
library(readr)
library(readxl)
library(tidyverse)
library(corrplot)
library(ggfortify)
library(FactoMineR)
library(factoextra)
library(gplots)
library(ggpubr)
library(magrittr)
```

Read the dataset

```
travel <- read_excel("data_ordination.xlsx")
head(travel)
```

```
## # A tibble: 6 x 14
##   Respon~1 UserL~2 Text ~3 Q1    Q1_   Q6_15 Q6_16 Q6_17 Q6_18 Q6_19 Q14   Q15
##   <chr>    <chr>   <chr>   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 "Respon~ "User ~ "Text ~ "How~ "How~ "How~ "How~ "How~ "How~ "How~ "How~ "To ~
## 2 "{\"Imp~ "{\"Im~ "{\"Im~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~
## 3 "1"      "EN"    "Yes"   "<3 ~ "3"   "54"  "50"  "71"  "50"  "51"  "18-~ "Fem~
## 4 "2"      "EN"    "Yes"   "<3 ~ "3"   "52"  "52"  "51"  "53"  "52"  "35-~ "Mal~
## 5 "3"      "EN"    "Yes"   "4-6~ "5"    <NA>  <NA>  <NA>  <NA>  <NA> "65+~ "Mal~
## 6 "4"      "EN"    "Yes"   "<3 ~ "3"   "51"  "53"  "54"  "52"  "57"  "25-~ "Fem~
```

```
## # ... with 2 more variables: Q17 <chr>, Q18 <chr>, and abbreviated variable
## #   names 1: ResponseId, 2: UserLanguage, 3: `Text / Graphic`
```

Dataset contains 231 rows and 14 columns which is still messy. Thus, we'll conduct some data preprocessing steps.

DATA PREPROCESSING

```
# First, drop two first rows.
travel <- travel %>%
  slice(-c(1,2))

# Select used columns
travel_df <- travel[c(1,5,6,9,12,13,14)]
```

```
# CHECK MISSING VALUE----
# Count the missing values by column wise
print("Count of missing values by column wise")
```

```
## [1] "Count of missing values by column wise"
```

```
sapply(travel_df, function(x) sum(is.na(x)))
```

```
## ResponseId        Q1_      Q6_15      Q6_18        Q15        Q17        Q18
##          0         30         72         75         74         72         78
```

```
# Missing value imputation
# Since our data contains 46 missing value, let's impute with mode
# Function to see mode
calc_mode <- function(x){

  # List the distinct / unique values
  distinct_values <- unique(na.omit(x))

  # Count the occurrence of each distinct value
  distinct_tabulate <- tabulate(match(x, distinct_values))

  # Return the value with the highest occurrence
  distinct_values[which.max(distinct_tabulate)]
}

# Impute missing value----
travel_df <- travel_df %>%
  mutate(across(everything(), ~replace_na(.x, calc_mode(.x))))
```

```
# Rename column name
travel_df_clean <- travel_df %>%
       rename(respondent_id =1, travel_frequency = 2, checkin_exp = 3,
              fly_exp = 4, gender = 5, race=6,
              religion = 7)
```

```
head(travel_df_clean)
```

```
## # A tibble: 6 x 7
##   respondent_id travel_frequency checkin_exp fly_exp gender race         relig~1
##   <chr>         <chr>            <chr>       <chr>   <chr>  <chr>        <chr>
## 1 1             3                54          50      Female Asian        Islam
## 2 2             3                52          53      Male   Black of Af~ Islam
## 3 3             5                50          50      Male   Asian        Islam
## 4 4             3                51          52      Female Asian        Islam
## 5 5             3                48          100     Male   Asian        Islam
## 6 6             3                50          50      Male   White        Atheis~
## # ... with abbreviated variable name 1: religion
```

```
# CONVERT DATA TYPE----
# Convert all variables into integer
# Convert column 2 to 6 to numeric
travel_df_clean[,2:4] <- lapply(travel_df_clean[,2:4], as.numeric)
travel_df_clean[,5:7] <- lapply(travel_df_clean[,5:7], as.factor)
travel_df_clean[,5:7] <- lapply(travel_df_clean[,5:7], as.integer)
head(travel_df_clean)
```

```
## # A tibble: 6 x 7
##   respondent_id travel_frequency checkin_exp fly_exp gender  race religion
##   <chr>                   <dbl>       <dbl>   <dbl>  <int> <int>    <int>
## 1 1                           3          54      50      1     2        6
## 2 2                           3          52      53      3     3        6
## 3 3                           5          50      50      3     2        6
## 4 4                           3          51      52      1     2        6
## 5 5                           3          48     100      3     2        6
## 6 6                           3          50      50      3     8        2
```

Correspondence Analysis

Data Exploration

```
# set respondent_id as index
travel_df_clean <- travel_df_clean %>% column_to_rownames(., var = "respondent_id")
head(travel_df_clean)
```

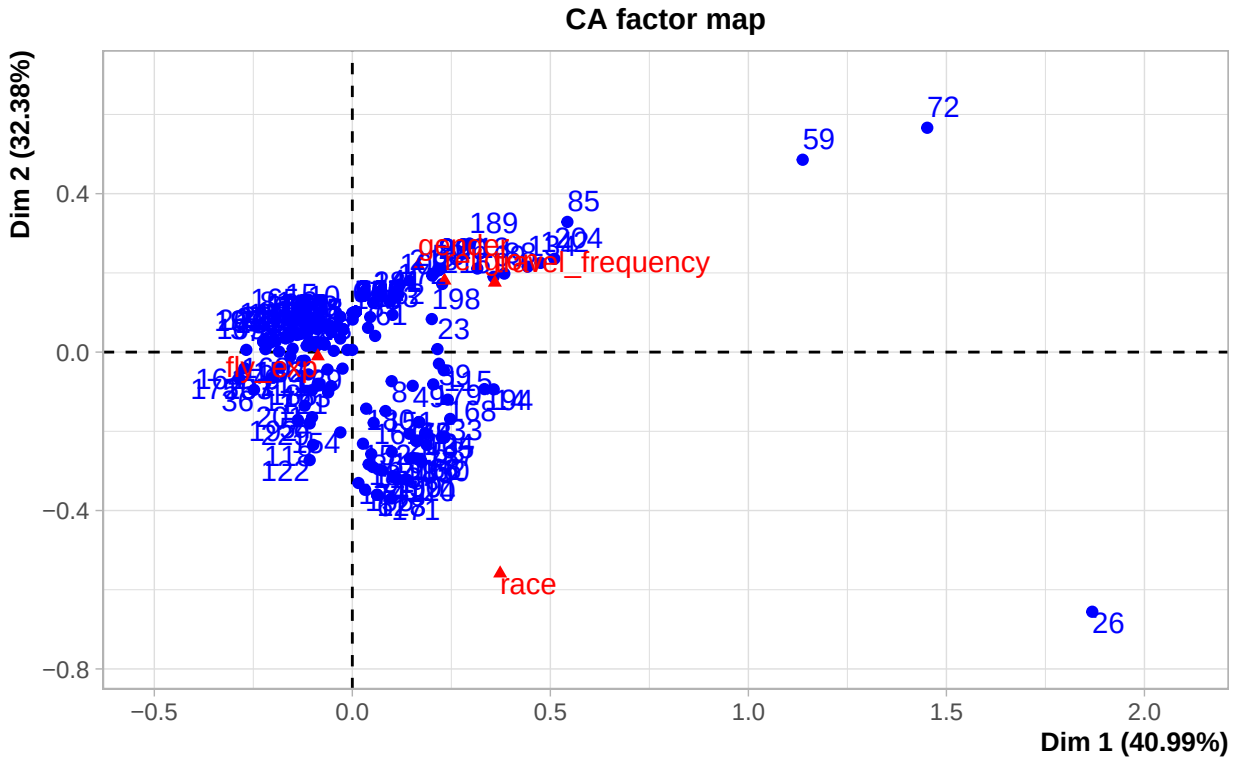
```
##   travel_frequency checkin_exp fly_exp gender race religion
## 1                3          54      50      1    2        6
## 2                3          52      53      3    3        6
## 3                5          50      50      3    2        6
## 4                3          51      52      1    2        6
## 5                3          48     100      3    2        6
## 6                3          50      50      3    8        2
```

```
# convert data into contingency table
df1 = travel_df_clean[,-c(2)]
dt = as.table(as.matrix(df1))

# graph
balloonplot(t(dt[1:15,]), main = "Travel Discriminations", xlab ="",
```

```
          ylab = "", label= FALSE, show.margins = FALSE)
```


```
res_ca <- CA(df1, graph = T)
```




```
print(res_ca)
```

```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  229  categories; the column variable has 5 categories
## The chi square of independence between the two variables is equal to 1047.721 (p-value =  0.00115160
## *The results are available in the following objects:
```

```
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

```
# eigen values
eig_val <- get_eigenvalue(res_ca)
eig_val
```

```
##        eigenvalue variance.percent cumulative.variance.percent
## Dim.1 0.026677504        40.989377                    40.98938
## Dim.2 0.021077130        32.384530                    73.37391
## Dim.3 0.012280015        18.867964                    92.24187
## Dim.4 0.005049297         7.758129                   100.00000
```

```
# Statistical Significance 1
chisq <- chisq.test(travel_df_clean)
chisq
```

```
##
##  Pearson's Chi-squared test
##
## data:  travel_df_clean
## X-squared = 1710.5, df = 1140, p-value < 2.2e-16
```

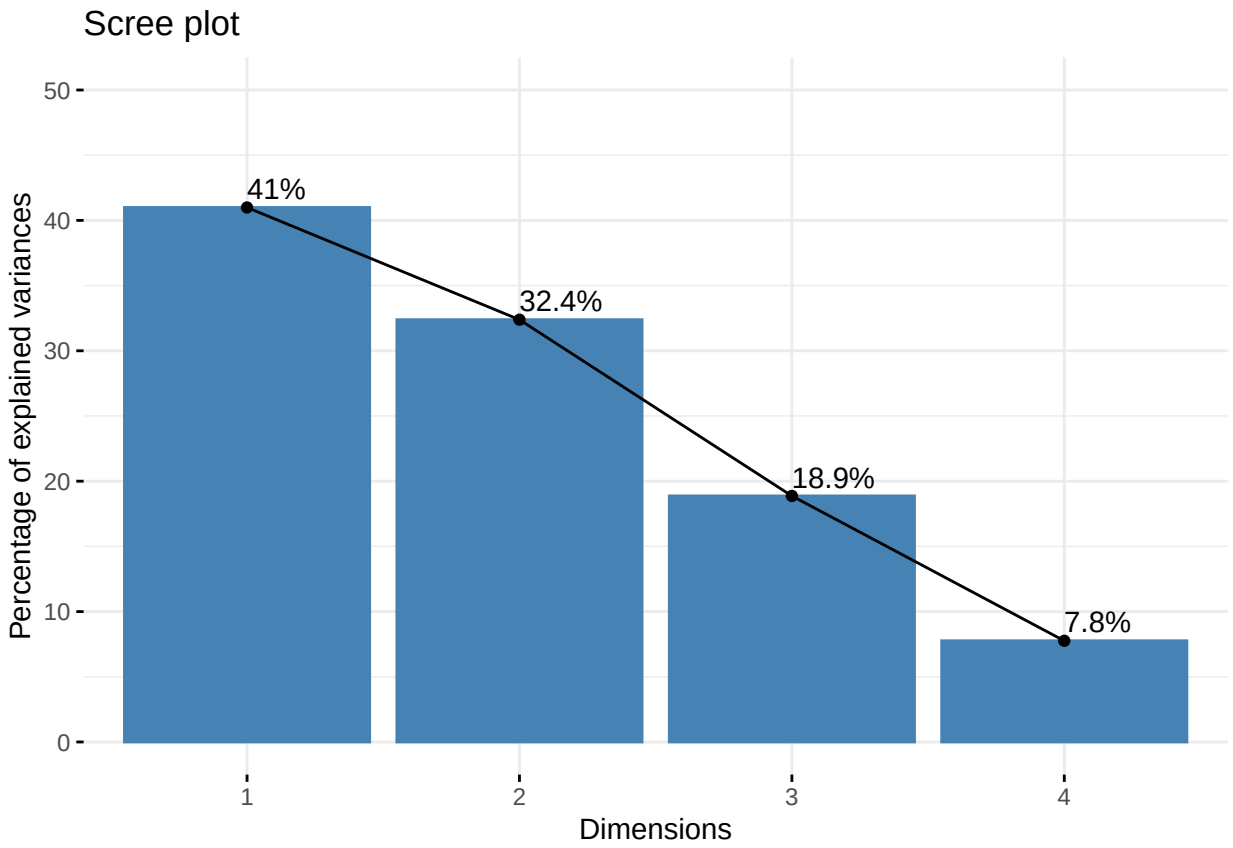
```
# statistical significance 2
summary(res_ca)
```

```
##
## Call:
## CA(X = df1, graph = T)
##
## The chi square of independence between the two variables is equal to 1047.721 (p-value =  0.00115160
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4
## Variance               0.027   0.021   0.012   0.005
## % of var.             40.989  32.385  18.868   7.758
## Cumulative % of var.  40.989  73.374  92.242 100.000
##
## Rows (the 10 first)
##                     Iner*1000    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## 1                 |     0.096 | -0.095  0.131  0.363 |  0.020  0.008  0.017 |
## 2                 |     0.057 | -0.047  0.035  0.162 |  0.003  0.000  0.001 |
## 3                 |     0.048 |  0.008  0.001  0.006 |  0.102  0.200  0.872 |
## 4                 |     0.106 | -0.109  0.177  0.444 |  0.017  0.006  0.011 |
```

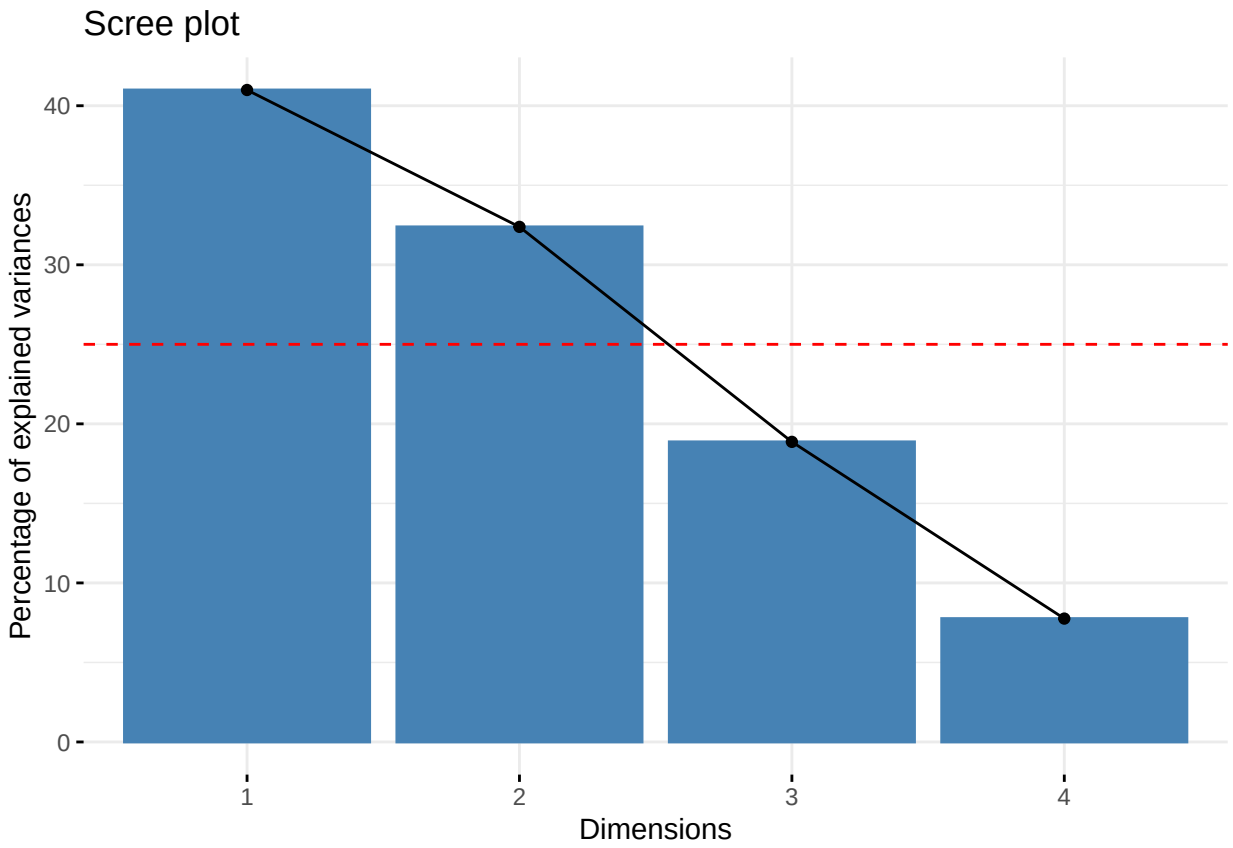
```
## 5                |     0.509 | -0.268  1.903  0.996 |  0.006  0.001  0.000 |
## 6                |     0.669 |  0.063  0.060  0.024 | -0.360  2.527  0.796 |
## 7                |     0.096 | -0.095  0.131  0.363 |  0.020  0.008  0.017 |
## 8                |     0.305 |  0.099  0.278  0.243 | -0.073  0.192  0.133 |
## 9                |     0.112 | -0.115  0.202  0.481 |  0.016  0.005  0.009 |
## 10               |     0.043 | -0.030  0.015  0.092 |  0.089  0.167  0.816 |
##                   Dim.3    ctr   cos2
## 1                 0.043  0.057  0.073 |
## 2                 0.101  0.354  0.760 |
## 3                 0.021  0.014  0.036 |
## 4                 0.038  0.047  0.055 |
## 5                 0.015  0.013  0.003 |
## 6                 0.022  0.017  0.003 |
## 7                 0.043  0.057  0.073 |
## 8                -0.113  0.786  0.316 |
## 9                 0.036  0.043  0.047 |
## 10                0.012  0.005  0.015 |
##
## Columns
##                    Iner*1000    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## travel_frequency |    17.184 |  0.360 33.799  0.525 |  0.175 10.175  0.125 |
## fly_exp          |     6.014 | -0.087 21.853  0.969 | -0.011  0.408  0.014 |
## gender           |     7.846 |  0.166  3.490  0.119 |  0.219  7.665  0.206 |
## race             |    21.273 |  0.373 24.387  0.306 | -0.559 69.201  0.686 |
## religion         |    12.766 |  0.233 16.471  0.344 |  0.181 12.551  0.207 |
##                   Dim.3    ctr   cos2
## travel_frequency -0.294 49.039  0.350 |
## fly_exp          -0.011  0.763  0.016 |
## gender            0.223 13.603  0.213 |
## race              0.058  1.298  0.007 |
## religion          0.231 35.296  0.340 |
```

Scree Plot

```
fviz_screeplot(res_ca, addlabels = TRUE, ylim = c(0, 50))
```



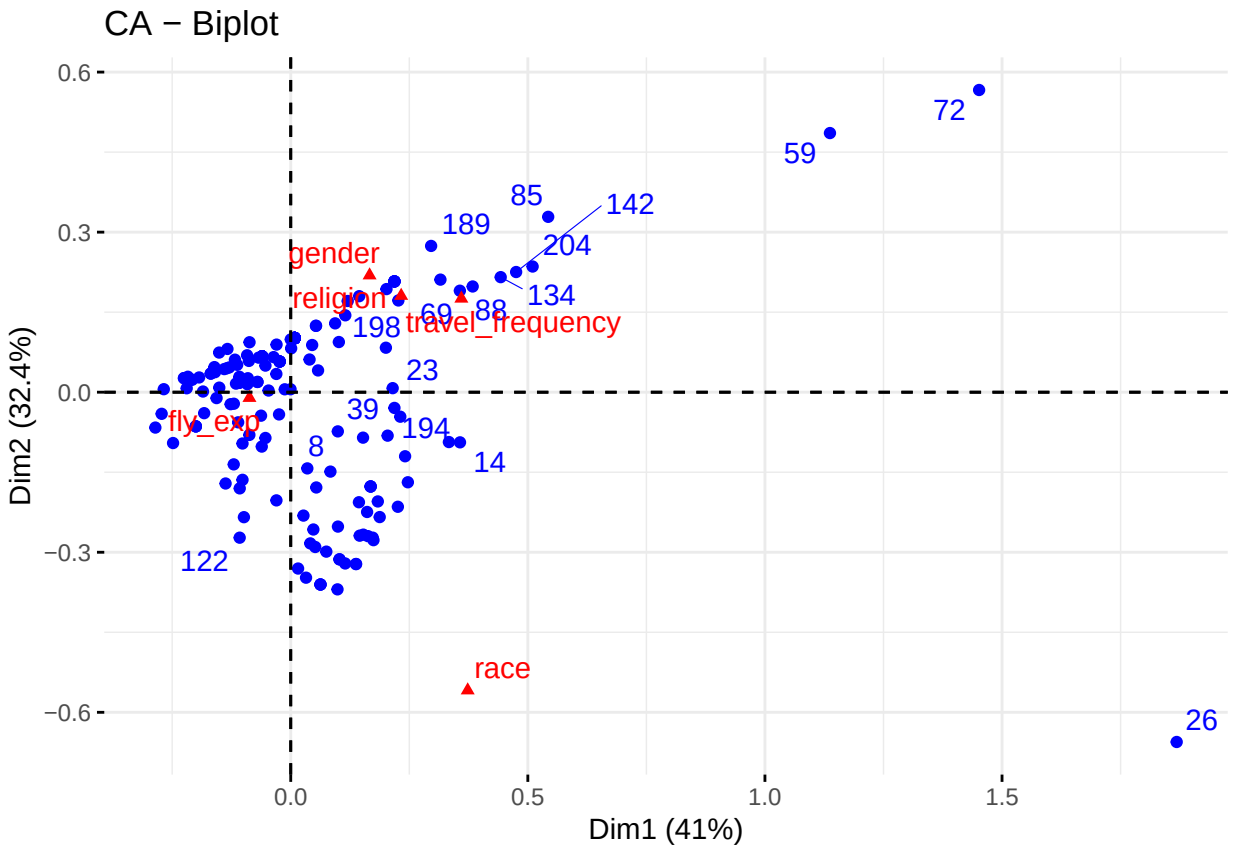

```
fviz_screeplot(res_ca) +
 geom_hline(yintercept=25, linetype=2, color="red")
```



BiPlot

```
# repel= TRUE to avoid text overlapping
#options(ggrepel.max.overlaps = Inf)
fviz_ca_biplot(res_ca, repel = TRUE)
```

```
## Warning: ggrepel: 212 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Graph of row Variables

```
row <- get_ca_row(res_ca)
row
```

```
## Correspondence Analysis - Results for rows
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the rows"
## 2 "$cos2"    "Cos2 for the rows"
## 3 "$contrib" "contributions of the rows"
## 4 "$inertia" "Inertia of the rows"
```

```
# Coordinates
head(row$coord)
```

Access the Row Component

```
##          Dim 1        Dim 2      Dim 3        Dim 4
## 1 -0.095188971  0.020388659 0.04269789 -0.117010259
## 2 -0.046764152  0.003153387 0.10140830  0.032326184
## 3  0.008191715  0.101506038 0.02060529  0.031844872
## 4 -0.108873411  0.017475296 0.03825072 -0.114348866
## 5 -0.267720111  0.005612860 0.01479668  0.003454174
## 6  0.062723171 -0.360460710 0.02243885  0.169894843
```

```r
# Cos2: quality on the factore map
head(row$cos2)
```

```
##        Dim 1        Dim 2       Dim 3        Dim 4
## 1 0.36256599 0.0166337859 0.072950212 0.5478500136
## 2 0.16168669 0.0007351952 0.760317705 0.0772604127
## 3 0.00568234 0.8724916942 0.035952980 0.0858729861
## 4 0.44398844 0.0114387113 0.054803374 0.4897694786
## 5 0.99635265 0.0004379449 0.003043544 0.0001658592
## 6 0.02410159 0.7959862503 0.003084543 0.1768276161
```

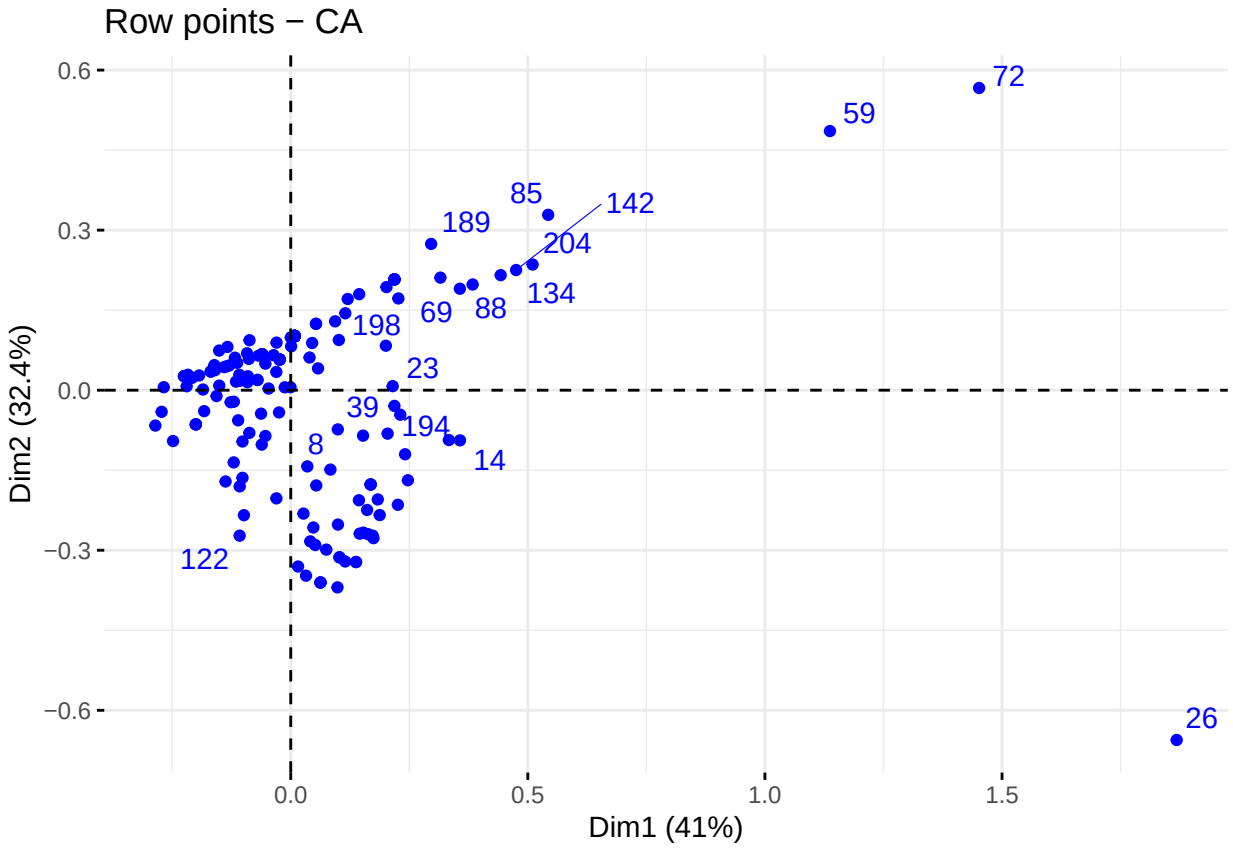
```r
# Contributions to the principal components
head(row$contrib)
```

```
##        Dim 1        Dim 2      Dim 3       Dim 4
## 1 0.13081208 0.0075960113 0.05717861 1.044327643
## 2 0.03462724 0.0001992875 0.35374071 0.087420844
## 3 0.00103128 0.2004214912 0.01417524 0.082341762
## 4 0.17664710 0.0057603076 0.04736837 1.029534477
## 5 1.90260980 0.0010584974 0.01262590 0.001673364
## 6 0.06046204 2.5274140830 0.01681025 2.343697352
```

```r
fviz_ca_row(res_ca, repel = TRUE)
```

Coordinates Row Points

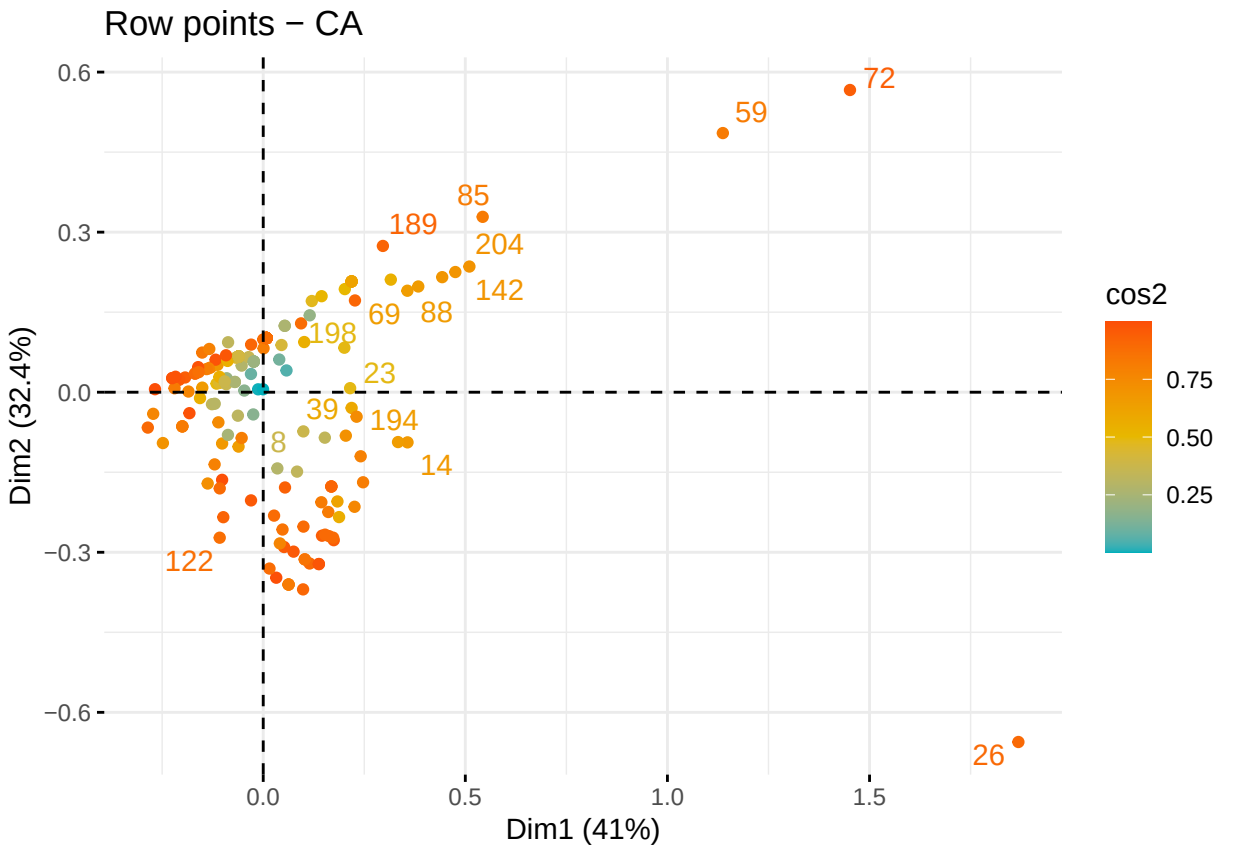
```
## Warning: ggrepel: 212 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



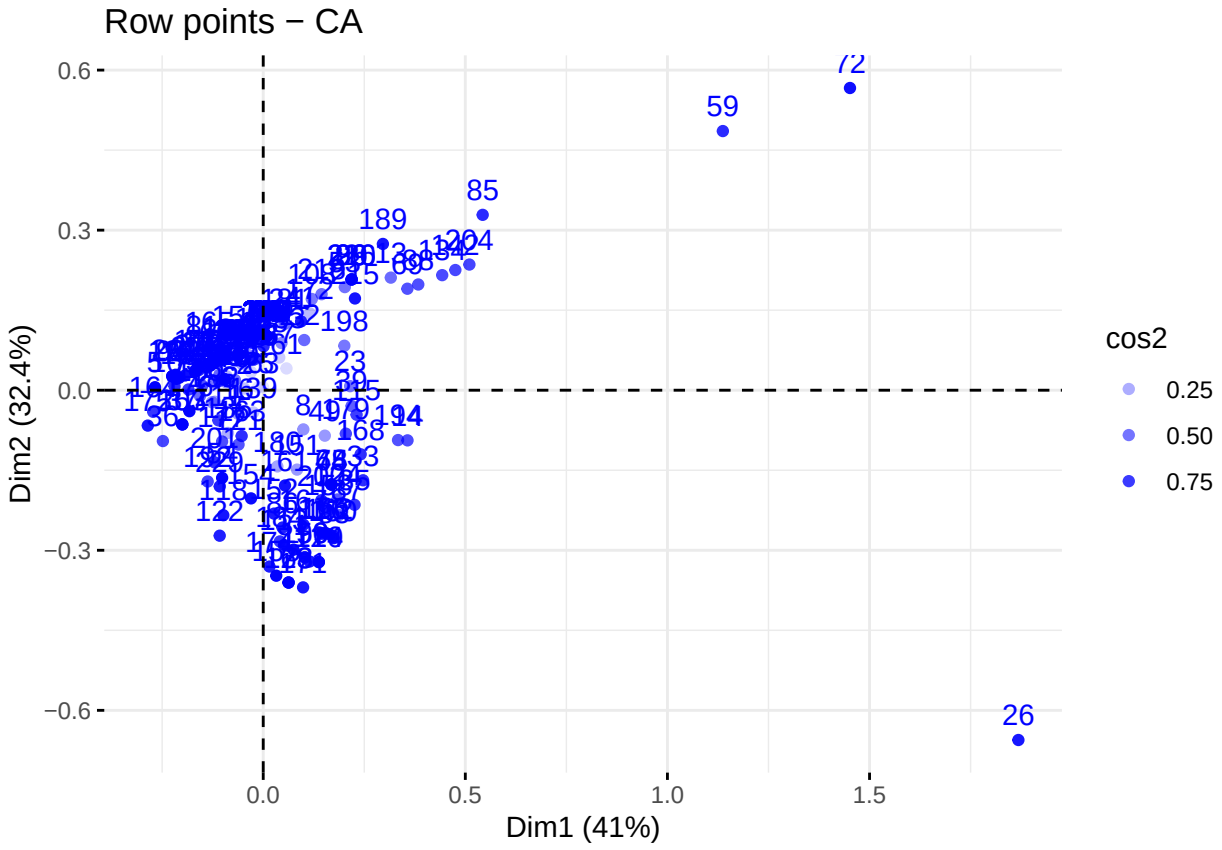
Quality of Representation of Rows

```
# Color by cos2 values: quality on the factor map
fviz_ca_row(res_ca, col.row = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE)
```

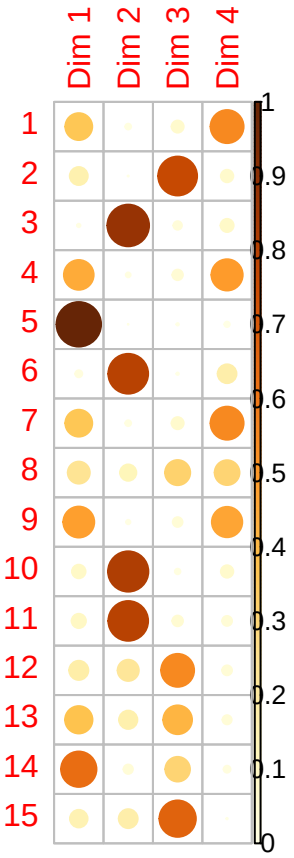
```
## Warning: ggrepel: 213 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
# Change the transparency by cos2 values
fviz_ca_row(res_ca, alpha.row="cos2")
```

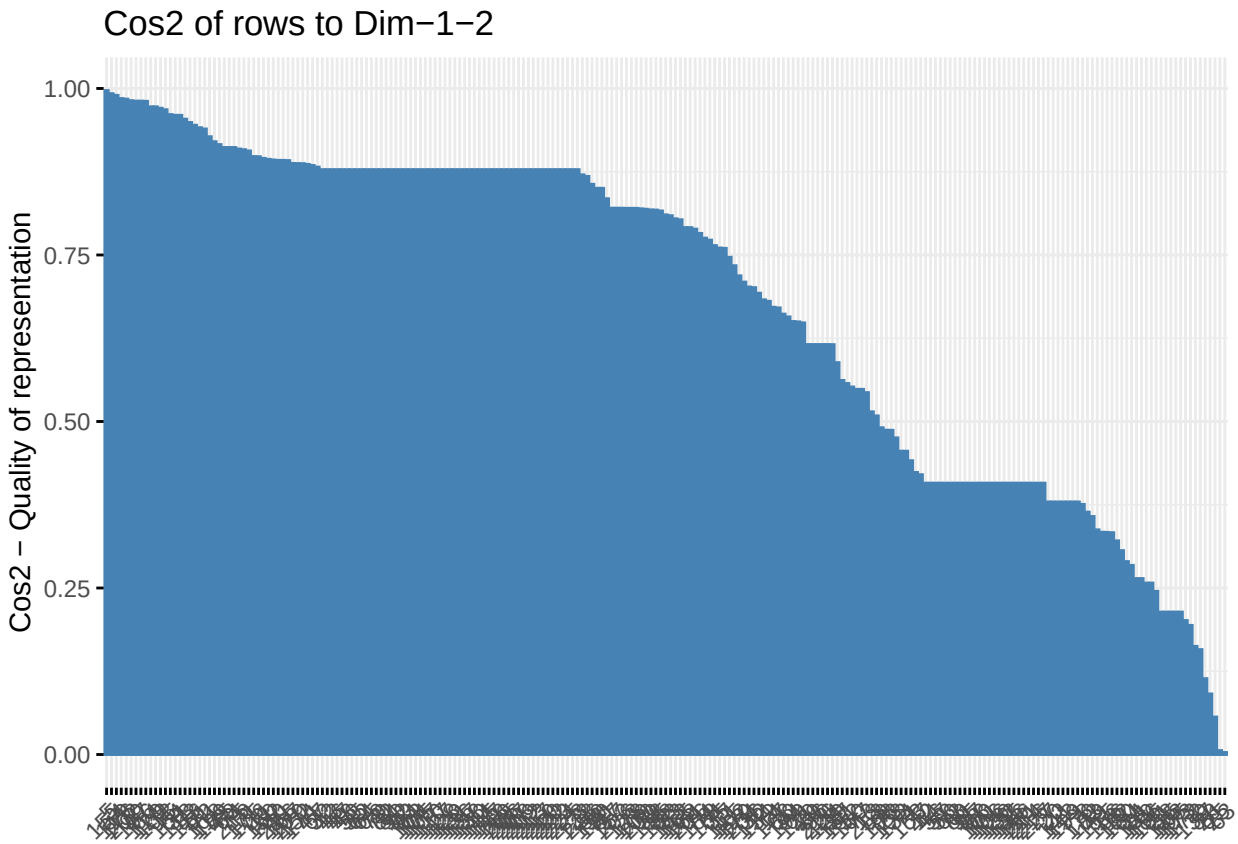



```
corrplot(row$cos2[1:15,], is.corr=FALSE)
```



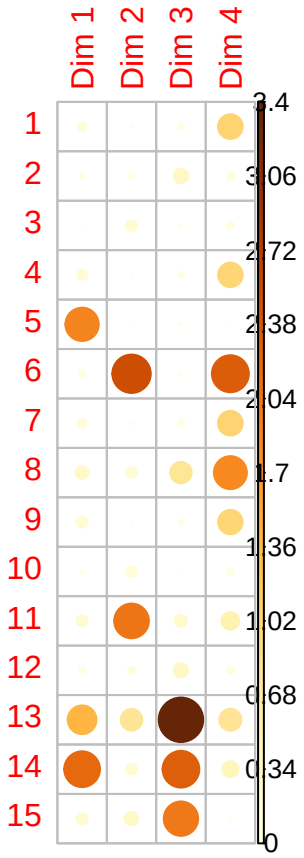
Corrplot

```
# Cos2 of rows on Dim.1 and Dim.2
fviz_cos2(res_ca, choice = "row", axes = 1:2)
```

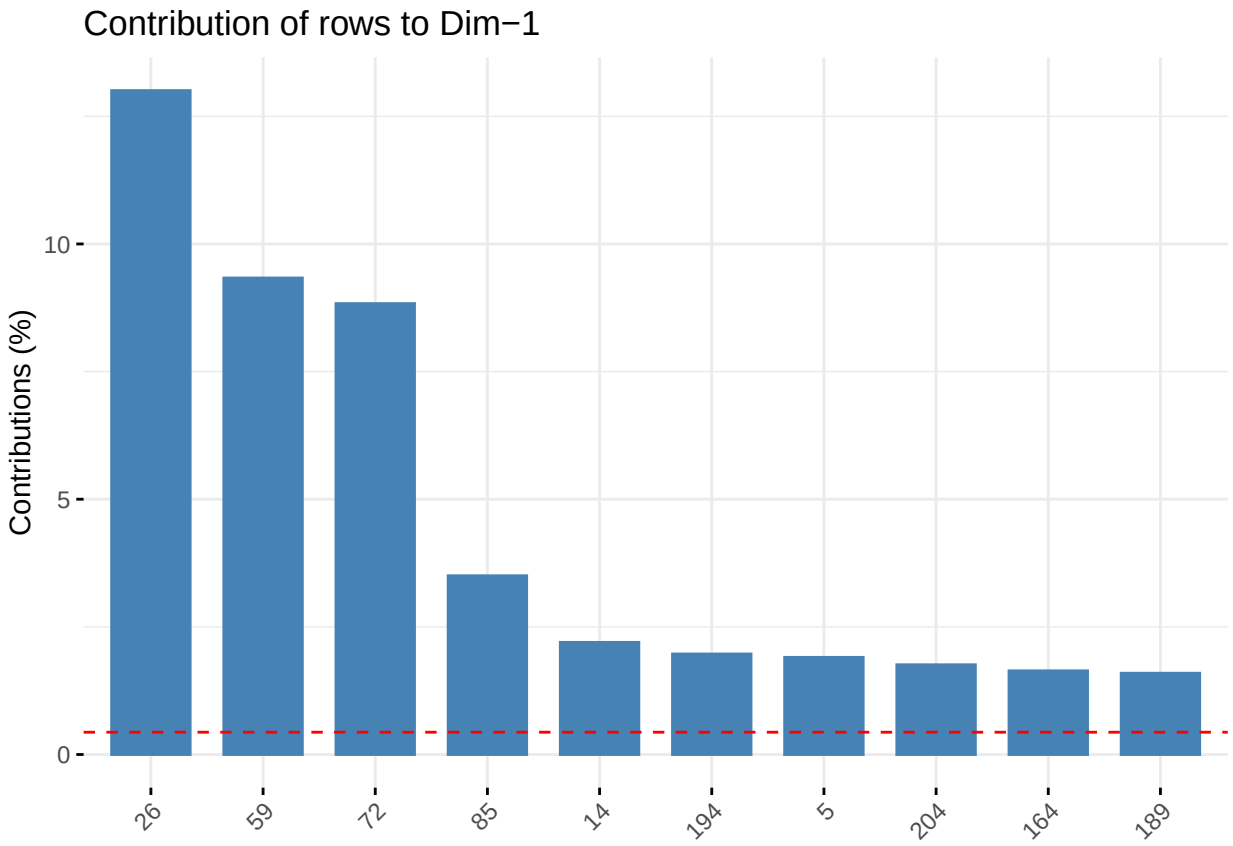


Contributions of rows to the dimensions

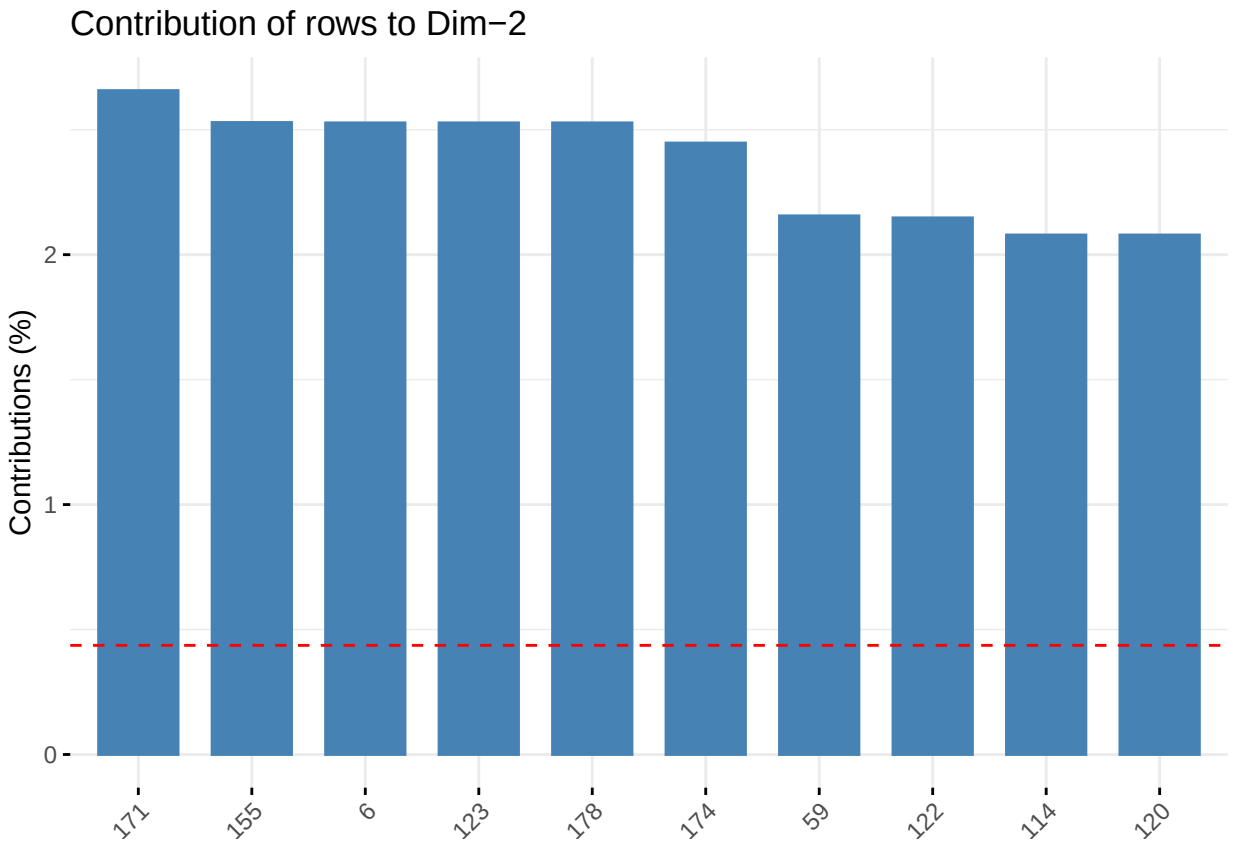
```
corrplot(row$contrib[1:15,], is.corr=FALSE)
```



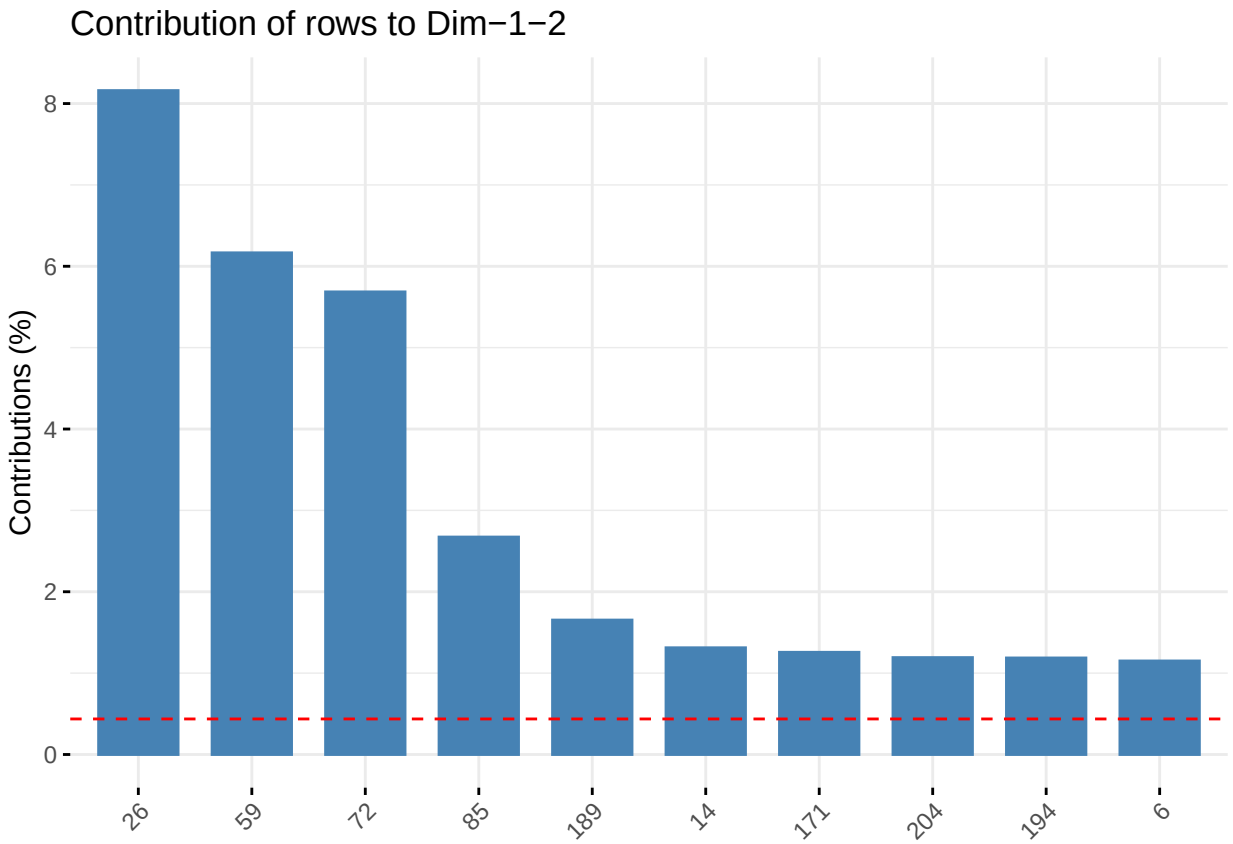

```
# Contributions of rows to dimension 1
fviz_contrib(res_ca, choice = "row", axes = 1, top = 10)
```




```
# Contributions of rows to dimension 2
fviz_contrib(res_ca, choice = "row", axes = 2, top = 10)
```

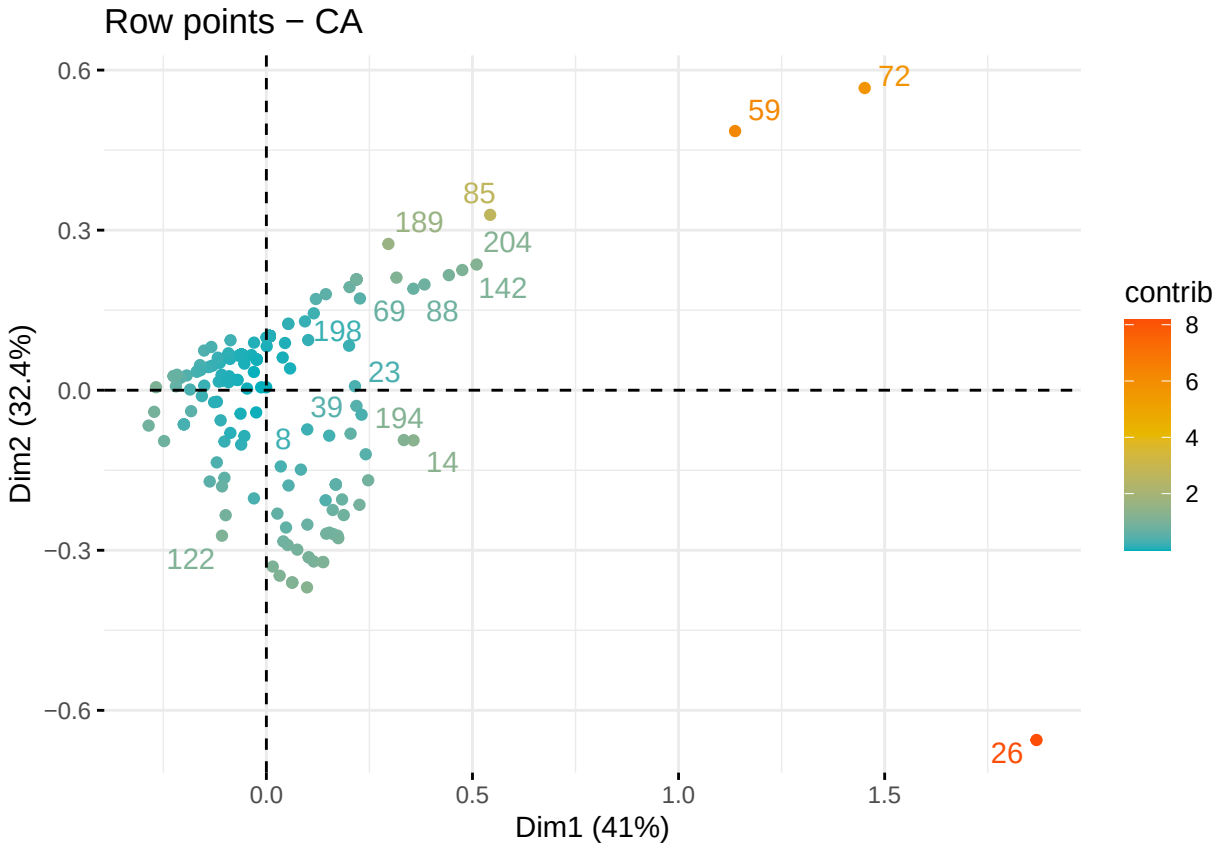



```
# Total contribution to dimension 1 and 2
fviz_contrib(res_ca, choice = "row", axes = 1:2, top = 10)
```



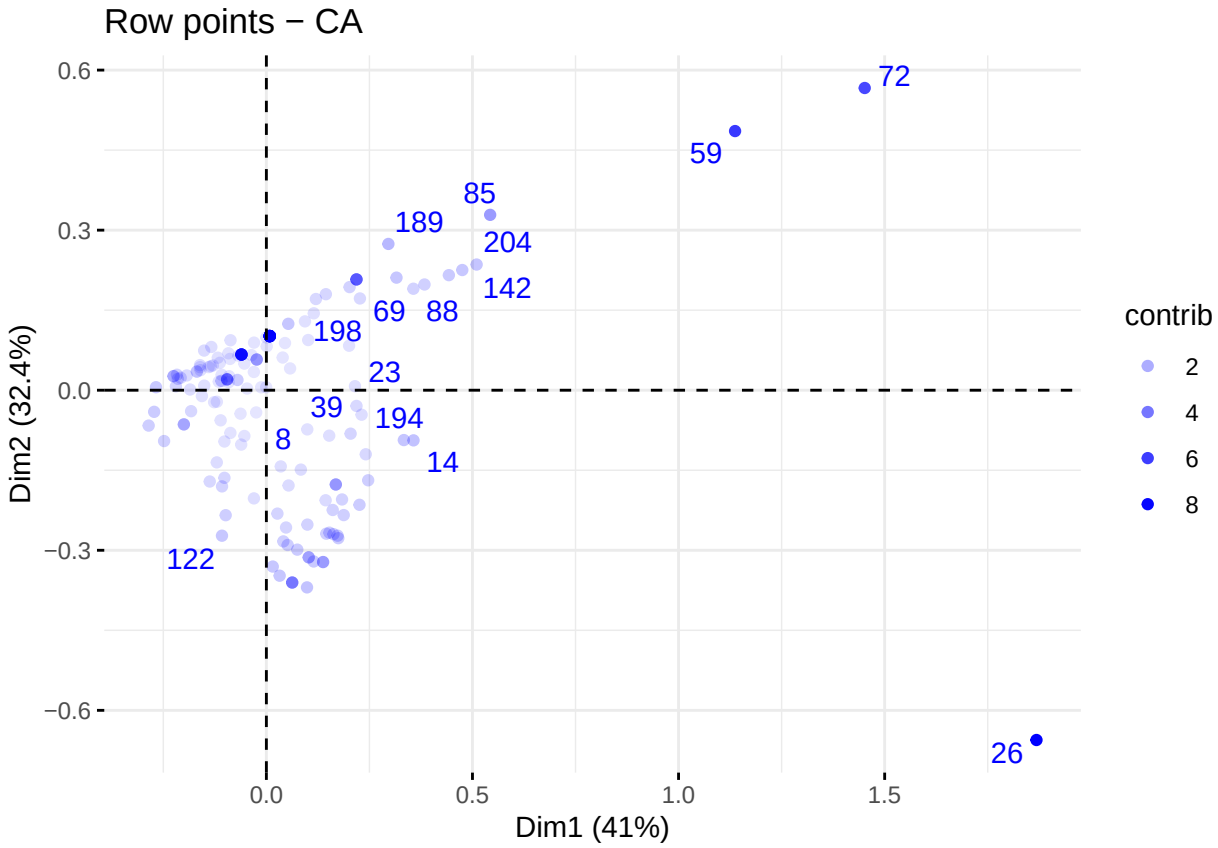

```
fviz_ca_row(res_ca, col.row = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE)
```

```
## Warning: ggrepel: 213 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
# Change the transparency by contrib values
fviz_ca_row(res_ca, alpha.row="contrib",
             repel = TRUE)
```

```
## Warning: ggrepel: 213 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Graph of column Variables

```
col <- get_ca_col(res_ca)
col
```

```
## Correspondence Analysis - Results for columns
##  ==================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the columns"
## 2 "$cos2"    "Cos2 for the columns"
## 3 "$contrib" "contributions of the columns"
## 4 "$inertia" "Inertia of the columns"
```

```
# Coordinates of column points
head(col$coord)
```

```
##                       Dim 1       Dim 2       Dim 3        Dim 4
## travel_frequency  0.35983729  0.17549212 -0.29407260  0.004110830
## fly_exp          -0.08707109 -0.01057473 -0.01103846 -0.002262902
## gender            0.16628685  0.21905547  0.22274527  0.328314504
## race              0.37319043 -0.55877696  0.05842279  0.021861741
## religion          0.23290549  0.18071330  0.23131791 -0.131121375
```

```
# Quality of representation
head(col$cos2)
```

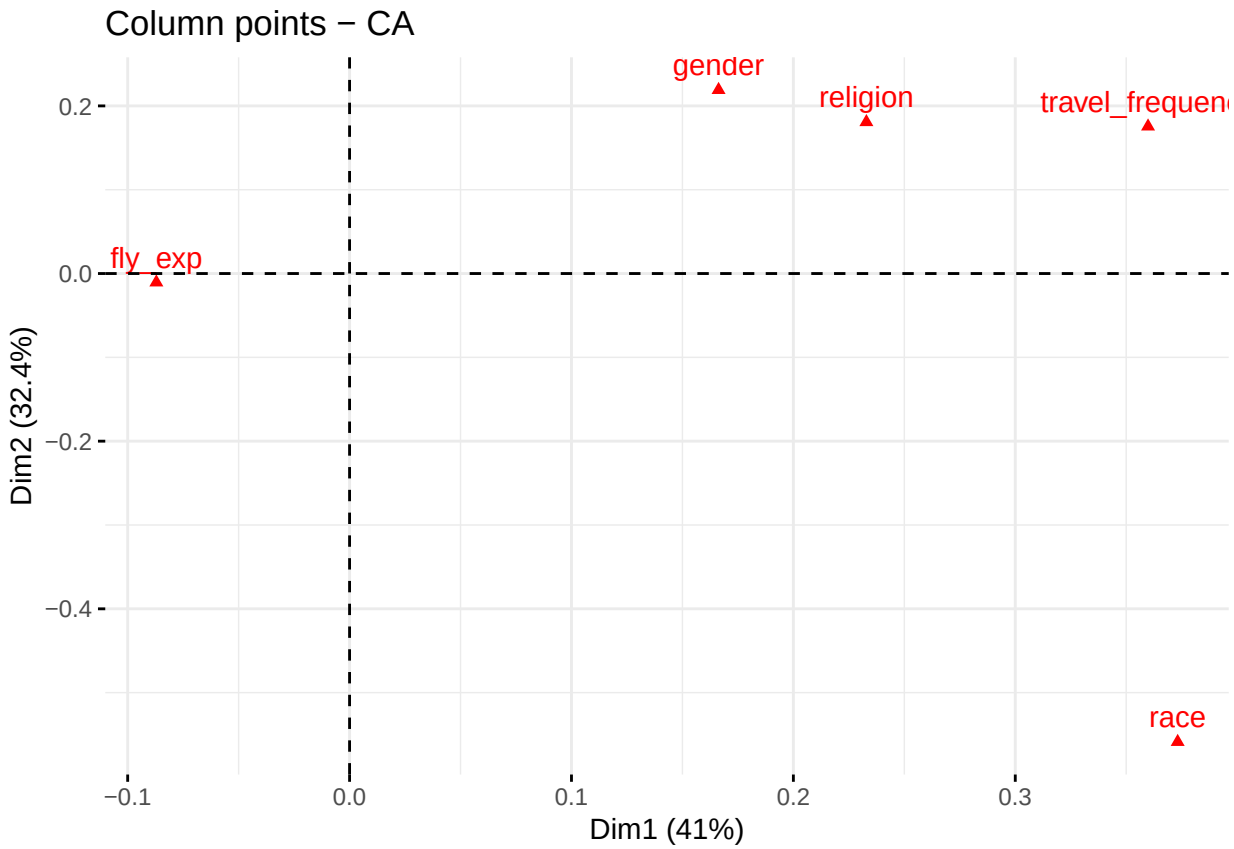
```
##                      Dim 1      Dim 2       Dim 3        Dim 4
## travel_frequency 0.5246981 0.12479938 0.350434044 0.0000684788
## fly_exp          0.9694644 0.01429956 0.015581187 0.0006548099
## gender           0.1186535 0.20590794 0.212903047 0.4625354662
```

```
## race              0.3058255 0.68562987 0.007495098 0.0010494995
## religion          0.3441873 0.20721233 0.339511033 0.1090893527
```

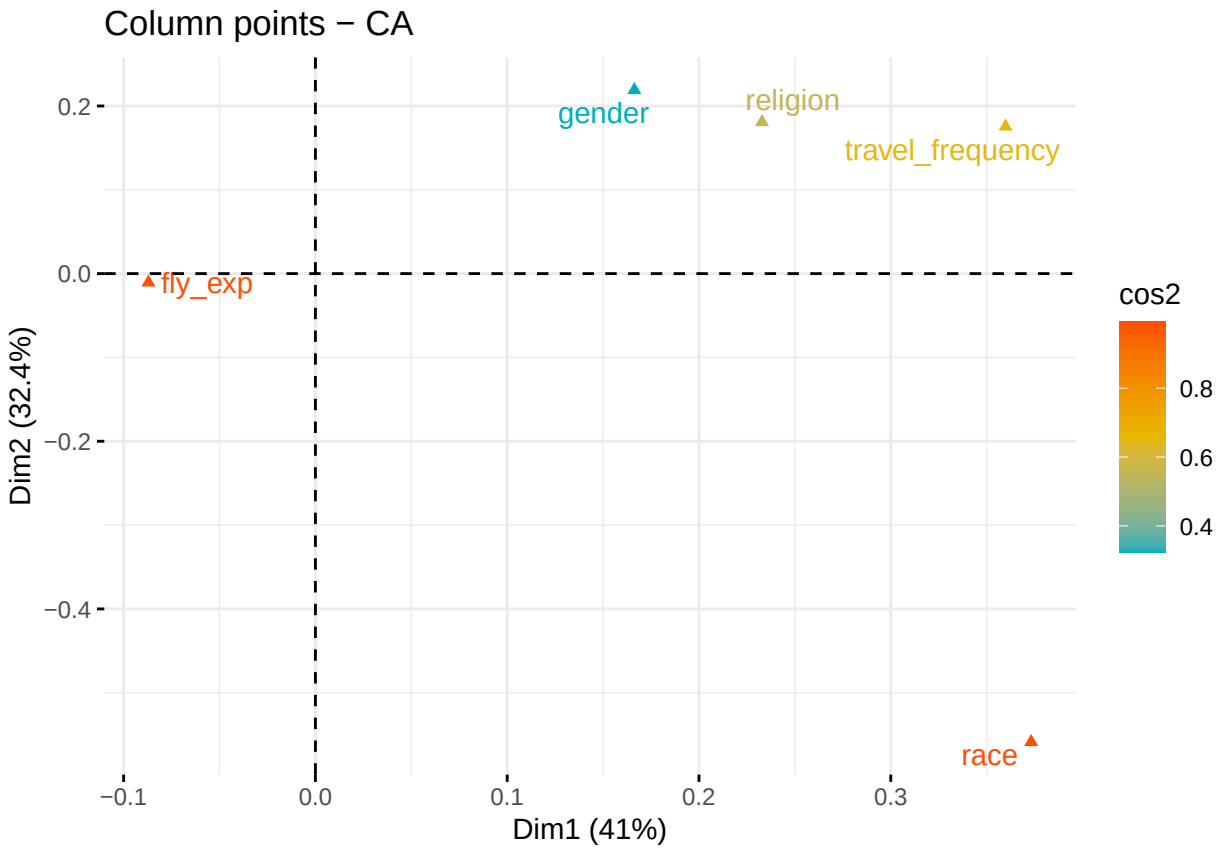
```
# Contributions
head(col$contrib)
```

```
##                      Dim 1      Dim 2     Dim 3       Dim 4
## travel_frequency 33.798765 10.1750715 49.039260  0.02330568
## fly_exp          21.853269  0.4079818  0.763012  0.07798554
## gender            3.489779  7.6652105 13.603336 71.87479601
## race             24.387188 69.2008496  1.298409  0.44216516
## religion         16.470998 12.5508866 35.295982 27.58174760
```

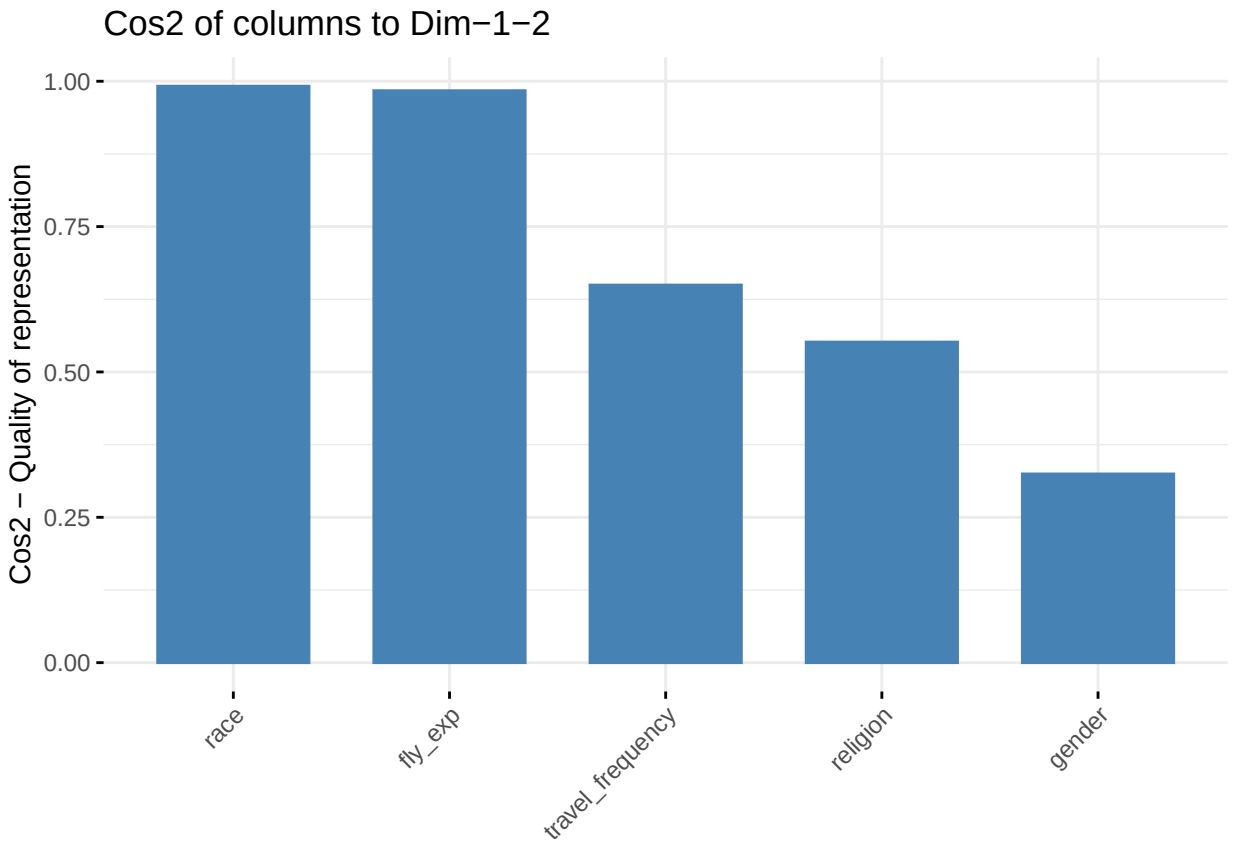
```
fviz_ca_col(res_ca)
```



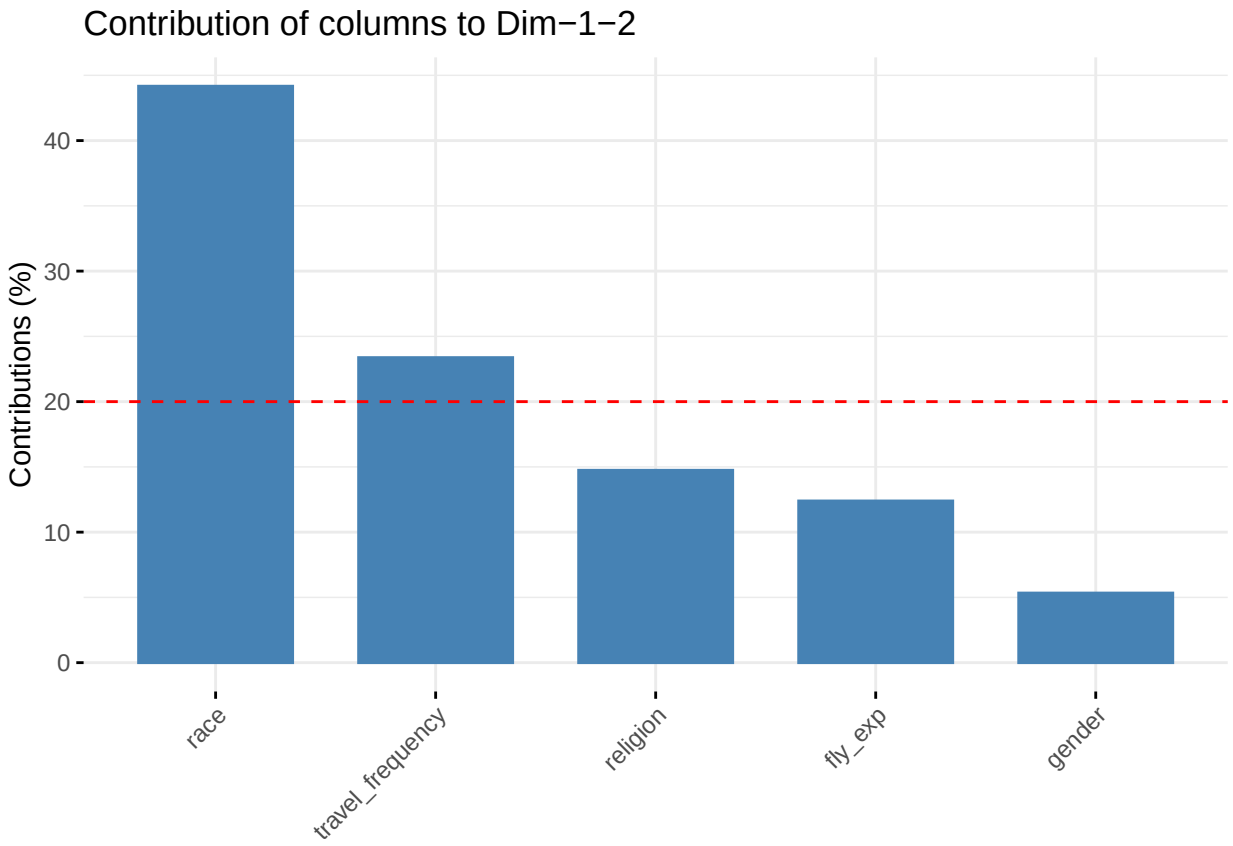

```
fviz_ca_col(res_ca, col.col = "cos2",
            gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
            repel = TRUE)
```




```
fviz_cos2(res_ca, choice = "col", axes = 1:2)
```



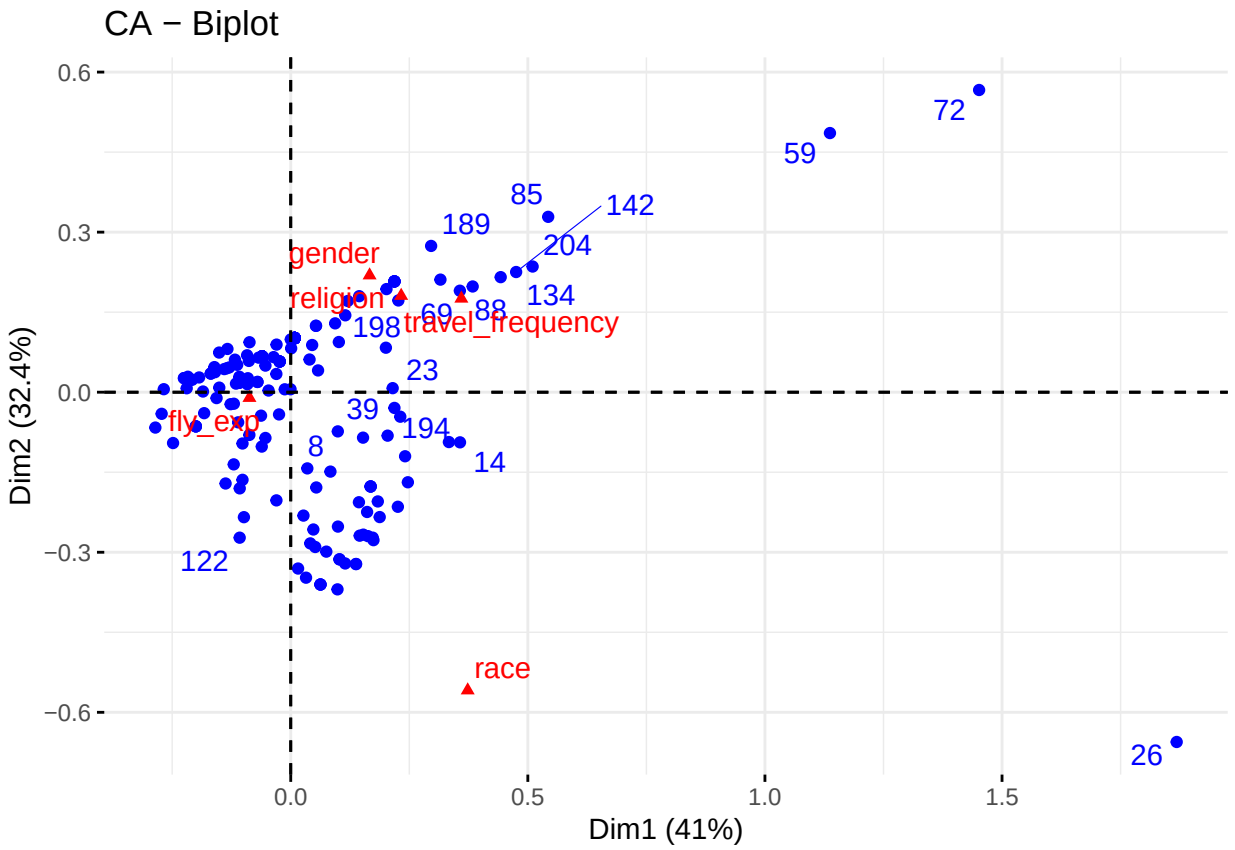

```
fviz_contrib(res_ca, choice = "col", axes = 1:2)
```



Symmetric Biplot

```
fviz_ca_biplot(res_ca, repel = TRUE)
```

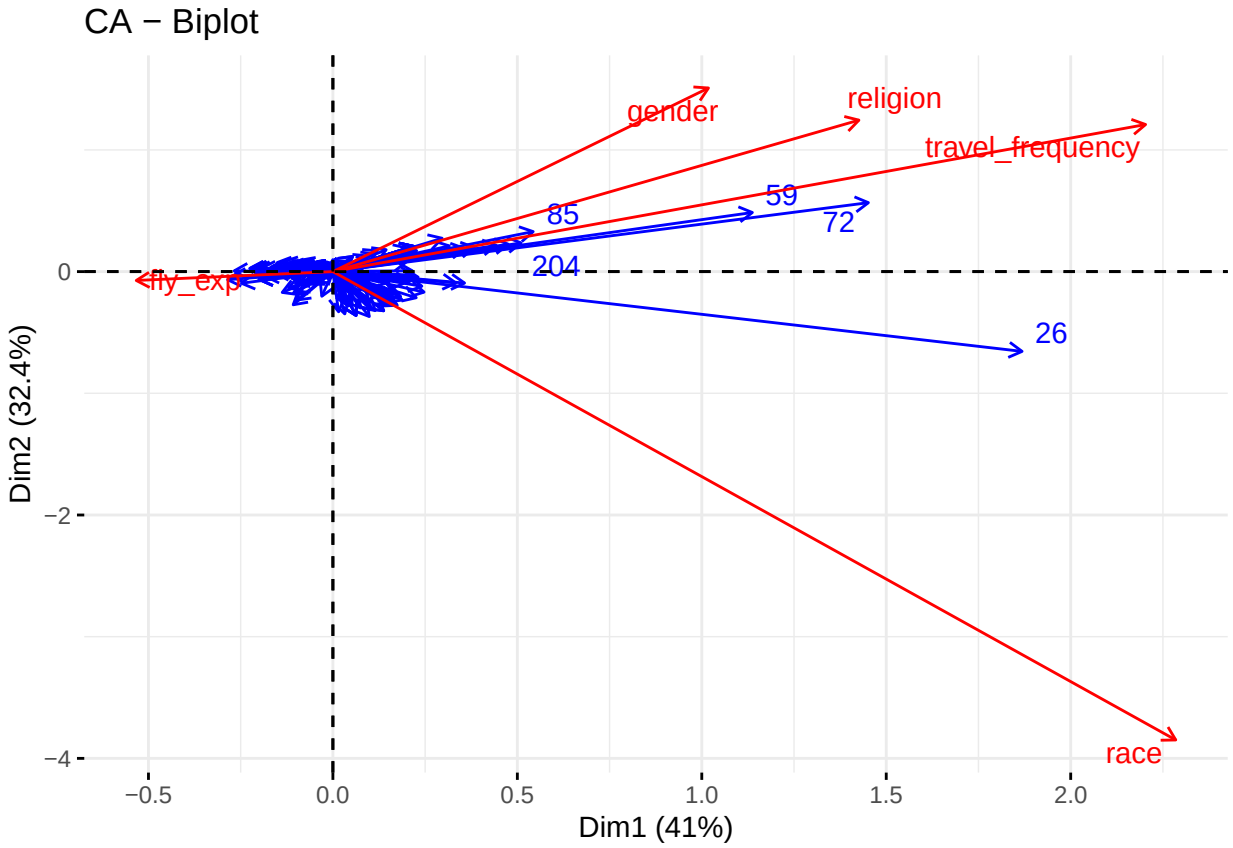
```
## Warning: ggrepel: 212 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Asymmetric Biplot

```
fviz_ca_biplot(res_ca,
               map ="rowprincipal", arrow = c(TRUE, TRUE),
               repel = TRUE)
```

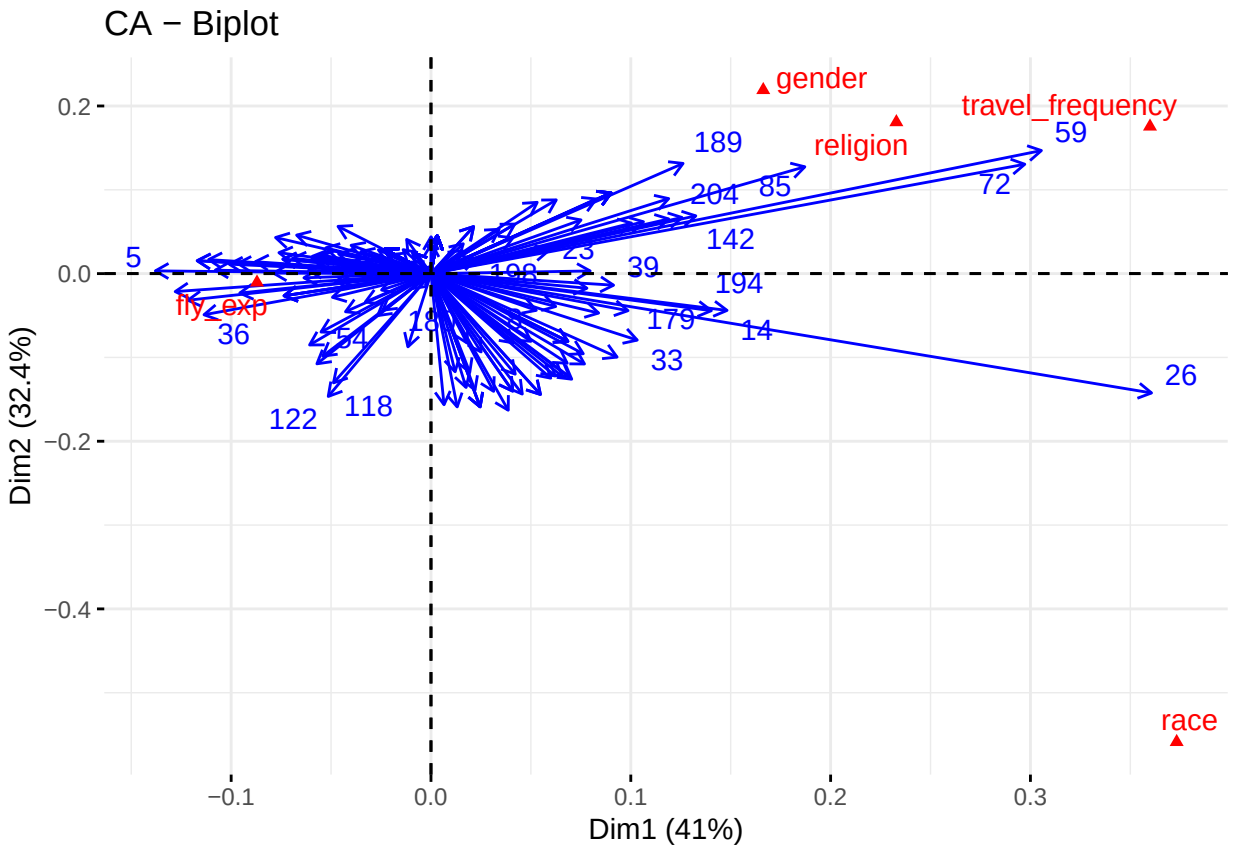
```
## Warning: ggrepel: 224 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Contribution Biplot

```
fviz_ca_biplot(res_ca, map ="colgreen", arrow = c(TRUE, FALSE),
               repel = TRUE)
```

```
## Warning: ggrepel: 208 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Dimension Description

```
# Dimension description
res_desc <- dimdesc(res_ca, axes = c(1,2))
# Description of dimension 1 by row points
head(res_desc[[1]]$row)
```

```
##          coord
## 175 -0.2854792
## 164 -0.2721847
## 5   -0.2677201
## 36  -0.2481467
## 99  -0.2251198
## 197 -0.2251198
```

```
# Description of dimension 1 by column points
head(res_desc[[1]]$col)
```

```
##                       coord
## fly_exp          -0.08707109
## gender            0.16628685
## religion          0.23290549
## travel_frequency  0.35983729
## race              0.37319043
```

```
# Description of dimension 2 by row points
head(res_desc[[2]]$row)
```

```
##          coord
## 26  -0.6555887
```

```
## 171 -0.3695536
## 6   -0.3604607
## 123 -0.3604607
## 178 -0.3604607
## 155 -0.3476451
```

```
# Description of dimension 1 by column points
head(res_desc[[2]]$col)
```

```
##                       coord
## race             -0.55877696
## fly_exp          -0.01057473
## travel_frequency  0.17549212
## religion          0.18071330
## gender            0.21905547
```

Multidimensional Scaling (MDS)

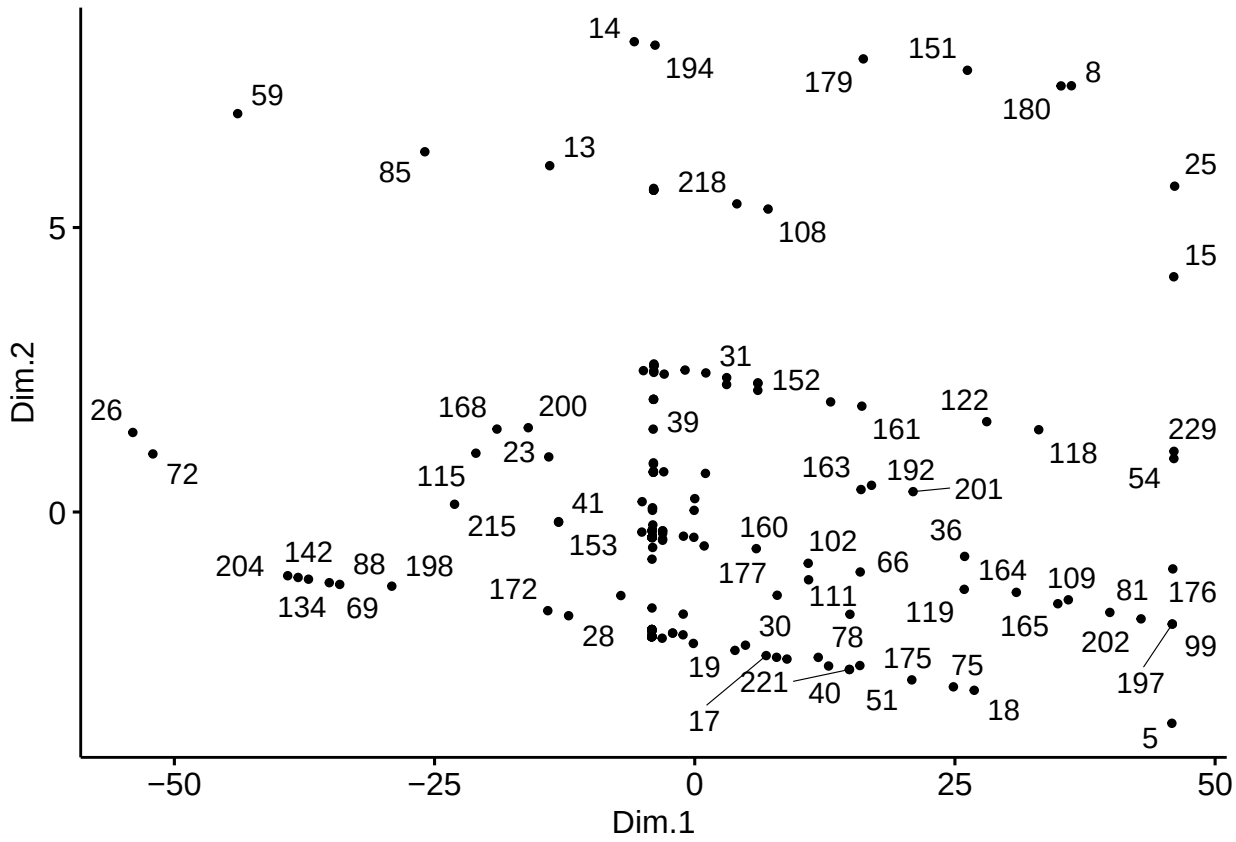
```
# Cmpute MDS
mds <- df1 %>%
  dist() %>%
  cmdscale() %>%
  as_tibble()
```

```
## 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`.
```

```
colnames(mds) <- c("Dim.1", "Dim.2")

# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(df1),
          size = 1,
          repel = TRUE)
```

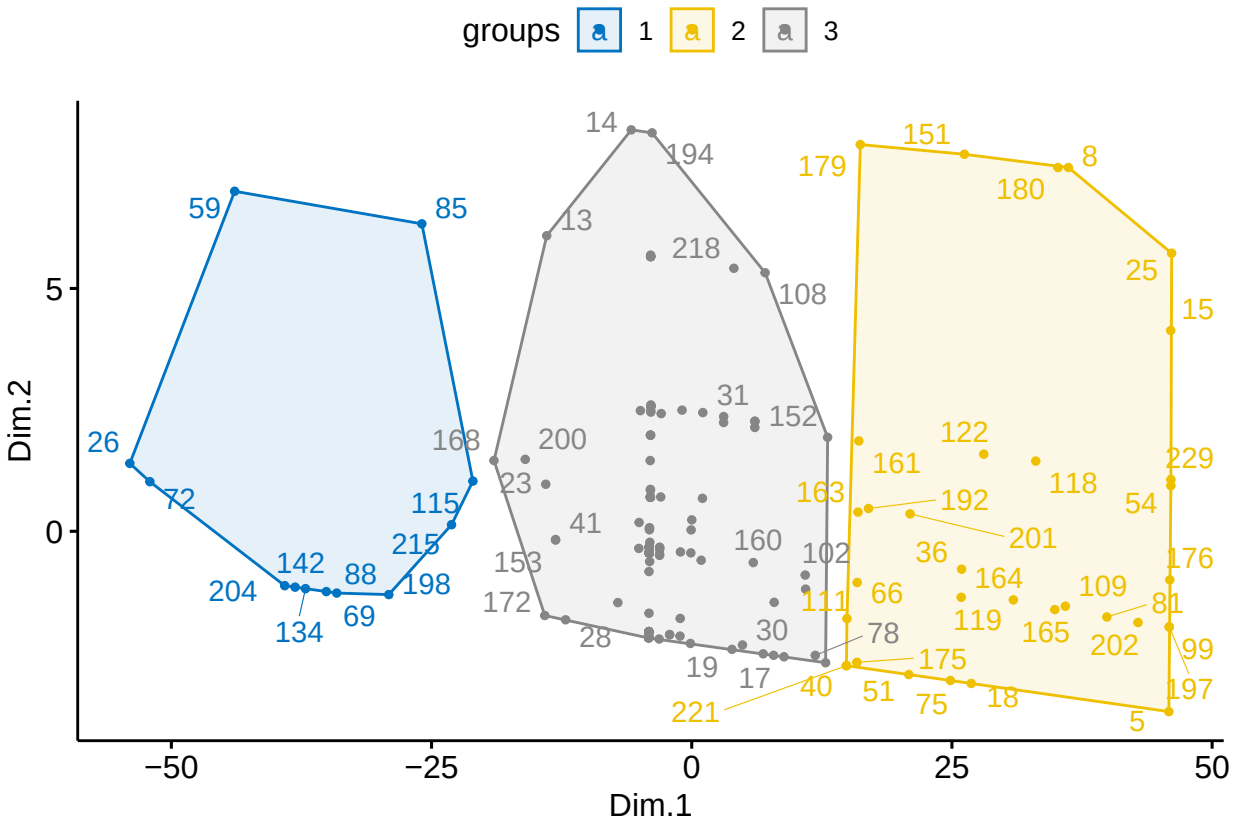
```
## Warning: ggrepel: 162 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Create 3 groups k-means clustering

```
# K-means clustering
clust <- kmeans(mds, 3)$cluster %>%
  as.factor()
mds <- mds %>%
  mutate(groups = clust)
# Plot and color by groups
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(df1),
          color = "groups",
          palette = "jco",
          size = 1,
          ellipse = TRUE,
          ellipse.type = "convex",
          repel = TRUE)
```

```
## Warning: ggrepel: 164 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Cannonical Correspondence Analysis

```
library(vegan)
```

```
#df2 <- travel_df_clean[,2]
#trav_cca <- cca(df2 ~ religion, data= df1)
#trav_cca
#plot(trav_cca)
```