Analyse en Composantes Principales (ACP) Rapport

Erwan

12/02/2026

```
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(scales)
library(tidyverse)# ensemble de packages utiles pour manipuler et visualiser les données
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.1     ✓ readr     2.1.6
## ✓ lubridate 1.9.5     ✓ stringr   1.6.0
## ✓ purrr     1.2.1     ✓ tibble    3.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ readr::col_factor() masks scales::col_factor()
## ✖ purrr::discard()    masks scales::discard()
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::lag()        masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become erro
rs
```

```
library(janitor)#  fonctions pratiques pour nettoyer les noms de colonnes (clean_names, etc.)
```

```
##
## Attachement du package : 'janitor'
##
## Les objets suivants sont masqués depuis 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(skimr) #résumé statistique rapide du jeu de données
library(FactoMineR) #réalisation de l'ACP
library(factoextra) #visualisations et extraction des résultats de l'ACP
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(missMDA)  #imputation des valeurs manquantes adaptée à l'ACP
```

#Import et aperçu

```
# Importer les données depuis le fichier CSV puis nettoyer les noms de colonnes
df_raw <- readr::read_csv("db.csv") |> janitor::clean_names()
```

```
## Rows: 2370 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (8): id_global, nom_pays, pays, trimestre, sequence_adn, smoke, alcohol,...
## dbl (6): annee, charge_virale, CD4, anticorps, CA_15_3, CA_15_3_apres
##
## 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.
```

```
# Aperçu de la structure du tableau (types, quelques valeurs) pour vérifier l'import
glimpse(df_raw)
```

```
## Rows: 2,370
## Columns: 14
## $ id_global     <chr> "AY611706.4.1.1987.AL", "AY611699.24.3.1989.AL", "AY6116…
## $ nom_pays      <chr> "ALBANIA", "ALBANIA", "ALBANIA", "ALBANIA", "ALBANIA", "…
## $ pays          <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "A…
## $ annee         <dbl> 1987, 1989, 1990, 1990, 1990, 1990, 1990, 1992, 1993, 19…
## $ trimestre     <chr> "Q1", "Q1", "Q1", "Q1", "Q2", "Q2", "Q4", "Q4", "Q2", "Q…
## $ sequence_adn  <chr> "----------------------------------------------------------…
## $ smoke         <chr> "fumeur léger", "fumer modéré", "gros fumeur", NA, "gros…
## $ alcohol       <chr> "régulièrement", "occasionnellement", "quotidiennement",…
## $ csp           <chr> "Employés", "Employés", "Cadres et professions intellect…
## $ charge_virale <dbl> 0.101, NA, NA, NA, 0.000, NA, NA, 0.000, NA, 0.000, 0.54…
## $ cd4           <dbl> 155, NA, NA, NA, 466, NA, NA, 660, NA, NA, NA, NA, 864, …
## $ anticorps     <dbl> 0.2111, NA, NA, NA, 0.7672, NA, NA, NA, NA, 0.7791, 1.10…
## $ ca_15_3       <dbl> 52, 79, NA, 76, 68, 58, 64, 76, 51, NA, 77, 73, 82, 69, …
## $ ca_15_3_apres <dbl> 49, 91, NA, 61, 73, 71, 52, 70, 42, NA, 74, 92, 84, 75, …
```

```
# Résumé descriptif plus complet (NA, statistiques, distribution) pour repérer anomalies et manquants
skimr::skim(df_raw)
```

Data summary

Name	df_raw
Number of rows	2370
Number of columns	14
Column type frequency:	
character	8
numeric	6

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
id_global	0	1.00	18	22	0	2370	0
nom_pays	0	1.00	5	18	0	29	0
pays	0	1.00	2	2	0	29	0
trimestre	0	1.00	2	2	0	4	0
sequence_adn	173	0.93	558	15558	0	1967	0
smoke	946	0.60	10	18	0	5	0
alcohol	1172	0.51	6	17	0	4	0
csp	946	0.60	8	49	0	6	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
annee	0	1.00	1993.63	3.41	1980	1992.00	1995.00	1996.00	1998.00	
charge_virale	1395	0.41	0.40	0.39	0	0.07	0.42	0.55	3.67	
cd4	1499	0.37	803.33	287.17	1	656.00	850.00	1003.50	1375.00	
anticorps	1418	0.40	0.49	0.37	0	0.10	0.48	0.85	1.17	
ca_15_3	613	0.74	67.50	10.56	32	61.00	68.00	75.00	108.00	
ca_15_3_apres	708	0.70	73.64	18.39	-4	62.00	75.00	87.00	133.00	

Nettoyage / sélection

```
# Préparer le jeu de données pour l'analyse
df <- df_raw |>
  mutate(
    # Conversion des variables qualitatives en facteurs (catégories) pour éviter des erreurs d'interprétation
    pays = as.factor(nom_pays),
    trimestre = as.factor(trimestre),
    smoke = as.factor(smoke),
    alcohol = as.factor(alcohol),
    csp = as.factor(csp)
  ) |>
  select(
    # Sélection des colonnes utiles
    id_global, nom_pays, annee, trimestre, smoke, alcohol, csp,
    charge_virale, cd4, anticorps, ca_15_3, ca_15_3_apres,
    sequence_adn
  )
# verification rapide du resultat
glimpse(df)
```

```
## Rows: 2,370
## Columns: 13
## $ id_global     <chr> "AY611706.4.1.1987.AL", "AY611699.24.3.1989.AL", "AY6116…
## $ nom_pays      <chr> "ALBANIA", "ALBANIA", "ALBANIA", "ALBANIA", "ALBANIA", "…
## $ annee         <dbl> 1987, 1989, 1990, 1990, 1990, 1990, 1990, 1992, 1993, 19…
## $ trimestre     <fct> Q1, Q1, Q1, Q1, Q2, Q2, Q4, Q4, Q2, Q2, Q3, Q1, Q1, Q2, …
## $ smoke         <fct> fumeur léger, fumer modéré, gros fumeur, NA, gros fumeur…
## $ alcohol       <fct> régulièrement, occasionnellement, quotidiennement, NA, r…
## $ csp           <fct> "Employés", "Employés", "Cadres et professions intellect…
## $ charge_virale <dbl> 0.101, NA, NA, NA, 0.000, NA, NA, 0.000, NA, 0.000, 0.54…
## $ cd4           <dbl> 155, NA, NA, NA, 466, NA, NA, 660, NA, NA, NA, NA, 864, …
## $ anticorps     <dbl> 0.2111, NA, NA, NA, 0.7672, NA, NA, NA, NA, 0.7791, 1.10…
## $ ca_15_3       <dbl> 52, 79, NA, 76, 68, 58, 64, 76, 51, NA, 77, 73, 82, 69, …
## $ ca_15_3_apres <dbl> 49, 91, NA, 61, 73, 71, 52, 70, 42, NA, 74, 92, 84, 75, …
## $ sequence_adn  <chr> "-----------------------------------------------------…
```

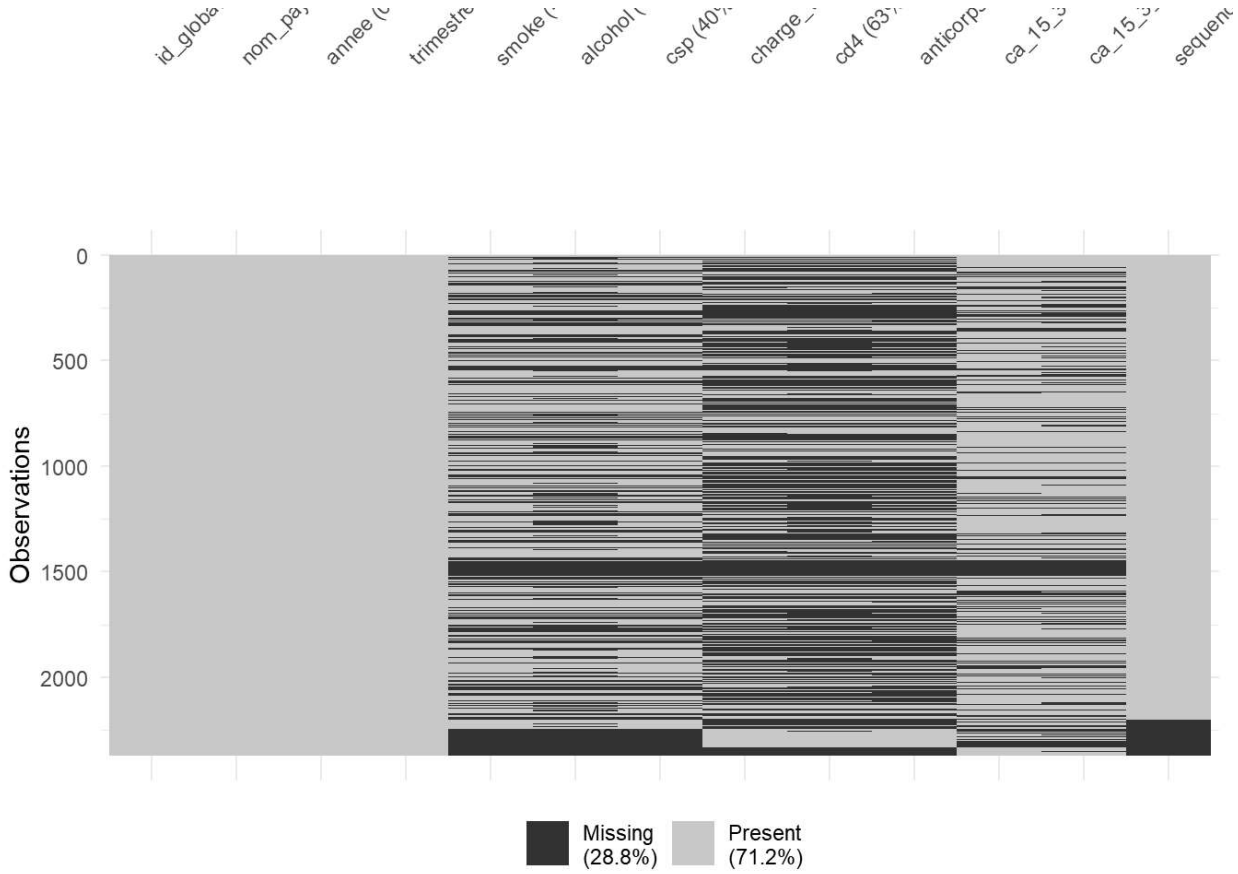
Valeurs manquantes (diagnostic)

```
# 9) Calculer le taux de valeurs manquantes (NA) pour chaque variable
#    -> summarise(across()) : applique la même formule à toutes les colonnes
#    -> mean(is.na(.)) : proportion de NA
#    -> pivot_longer : passer au format long (plus simple à trier/afficher)
na_rate <- df |> summarise(across(everything(), ~mean(is.na(.)))) |> pivot_longer(everything())
# Trier du plus manquant au moins manquant pour identifier les variables problématiques
na_rate |> arrange(desc(value))
```

```
## # A tibble: 13 × 2
##    name           value
##    <chr>          <dbl>
##  1 cd4           0.632
##  2 anticorps     0.598
##  3 charge_virale 0.589
##  4 alcohol       0.495
##  5 smoke         0.399
##  6 csp           0.399
##  7 ca_15_3_apres 0.299
##  8 ca_15_3       0.259
##  9 sequence_adn  0.0730
## 10 id_global     0
## 11 nom_pays      0
## 12 annee         0
## 13 trimestre     0
```

Visualisation des manquants

```
# Visualiser la matrice de valeurs manquantes (où se trouvent les NA dans le tableau)
# utile pour voir si les manquants sont concentrés sur certaines variables/individus
naniar::vis_miss(df)
```



ACP sur les biomarqueurs (quanti)

Matrice quantitative + estimation du nb de dimensions pour imputation

```
# Isoler les variables quantitatives qui serviront à l'ACP
X_quanti <- df |>
  select(charge_virale, cd4, anticorps, ca_15_3, ca_15_3_apres)
# Vérifier les statistiques de base
summary(X_quanti)
```

```
##  charge_virale        cd4           anticorps          ca_15_3
##  Min.   :0.000   Min.   :   1.0   Min.   :0.00010   Min.   : 32.0
##  1st Qu.:0.072   1st Qu.: 656.0   1st Qu.:0.09932   1st Qu.: 61.0
##  Median :0.418   Median : 850.0   Median :0.47685   Median : 68.0
##  Mean   :0.396   Mean   : 803.3   Mean   :0.48878   Mean   : 67.5
##  3rd Qu.:0.554   3rd Qu.:1003.5   3rd Qu.:0.85415   3rd Qu.: 75.0
##  Max.   :3.672   Max.   :1375.0   Max.   :1.16990   Max.   :108.0
##  NA's   :1395    NA's   :1499     NA's   :1418      NA's   :613
##  ca_15_3_apres
##  Min.   : -4.00
##  1st Qu.: 62.00
##  Median : 75.00
##  Mean   : 73.64
##  3rd Qu.: 87.00
##  Max.   :133.00
##  NA's   :708
```

```r
# Estimation du nombre optimal de composantes pour l'imputation PCA
set.seed(123) # Fixe l'aléatoire pour obtenir des résultats reproductibles
ncp_est <- missMDA::estim_ncpPCA(X_quanti, ncp.max = 5)
ncp_est$ncp  # Nombre de composantes retenu
```

```
## [1] 0
```

Imputation PCA (missMDA) puis ACP (FactoMineR)

```r
# 13) Imputer les valeurs manquantes puis réaliser l'ACP
set.seed(123)
imp <- missMDA::imputePCA(X_quanti, ncp = ncp_est$ncp)
# Lancer l'ACP sur les données imputées :
res_pca <- FactoMineR::PCA(
  imp$completeObs,
  scale.unit = TRUE,
  graph = FALSE
)
# Afficher un résumé des résultats (inertie, coordonnées, contributions)
res_pca
```

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 2370 individuals, described by 5 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

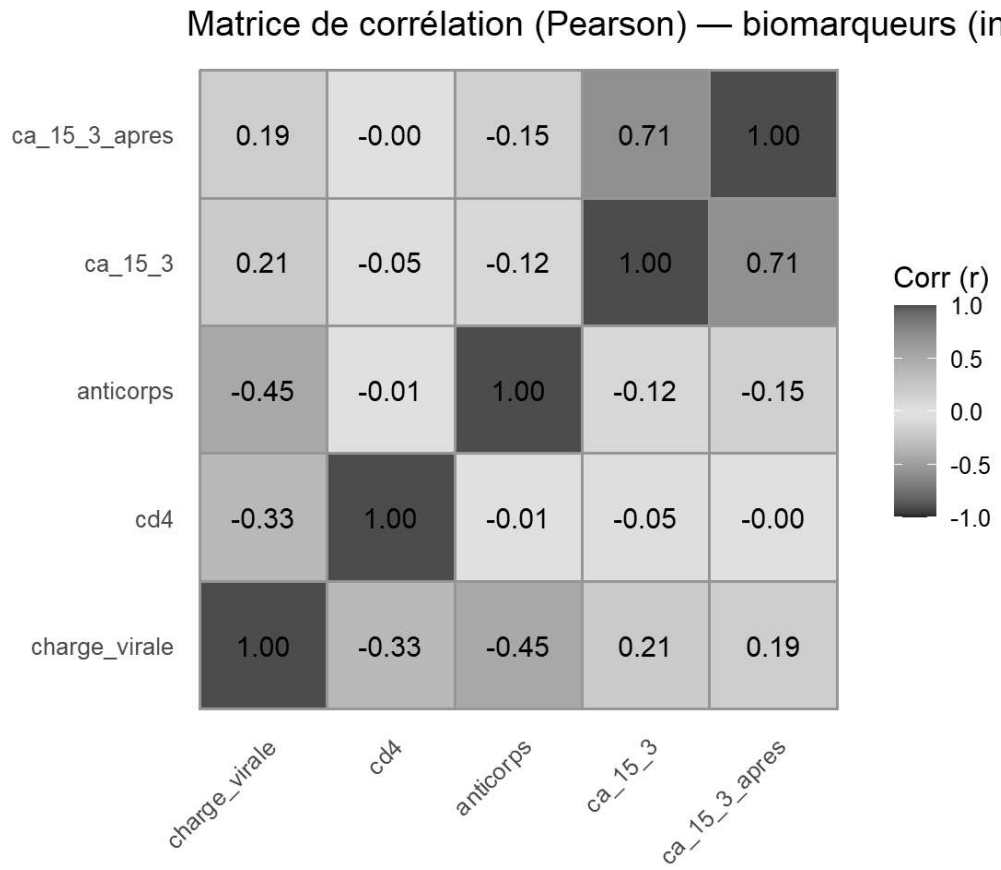
Matrice de corrélation

```r
# Matrice utilisée dans l'ACP (imputée)
X_imp <- as.data.frame(imp$completeObs)

# Matrice de corrélation
cor_mat <- cor(X_imp, method = "pearson")

# Mise en format long pour ggplot
cor_long <- as.data.frame(cor_mat) %>%
  rownames_to_column("var1") %>%
  pivot_longer(-var1, names_to = "var2", values_to = "r") %>%
  mutate(
    var1 = factor(var1, levels = colnames(cor_mat)),
    var2 = factor(var2, levels = colnames(cor_mat))
  )

# Heatmap ggplot
ggplot(cor_long, aes(var1, var2, fill = r)) +
  geom_tile(color = "green", linewidth = 0.6) +
  geom_text(aes(label = sprintf("%.2f", r)), size = 4) +
  scale_fill_gradient2(
    low = "blue", mid = "yellow", high = "red",
    midpoint = 0, limits = c(-1, 1), name = "Corr (r)"
  ) +
  coord_equal() +
  labs(
    title = "Matrice de corrélation (Pearson) — biomarqueurs (imputés)",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid = element_blank(),
    legend.position = "right"
  )
```



Résultats

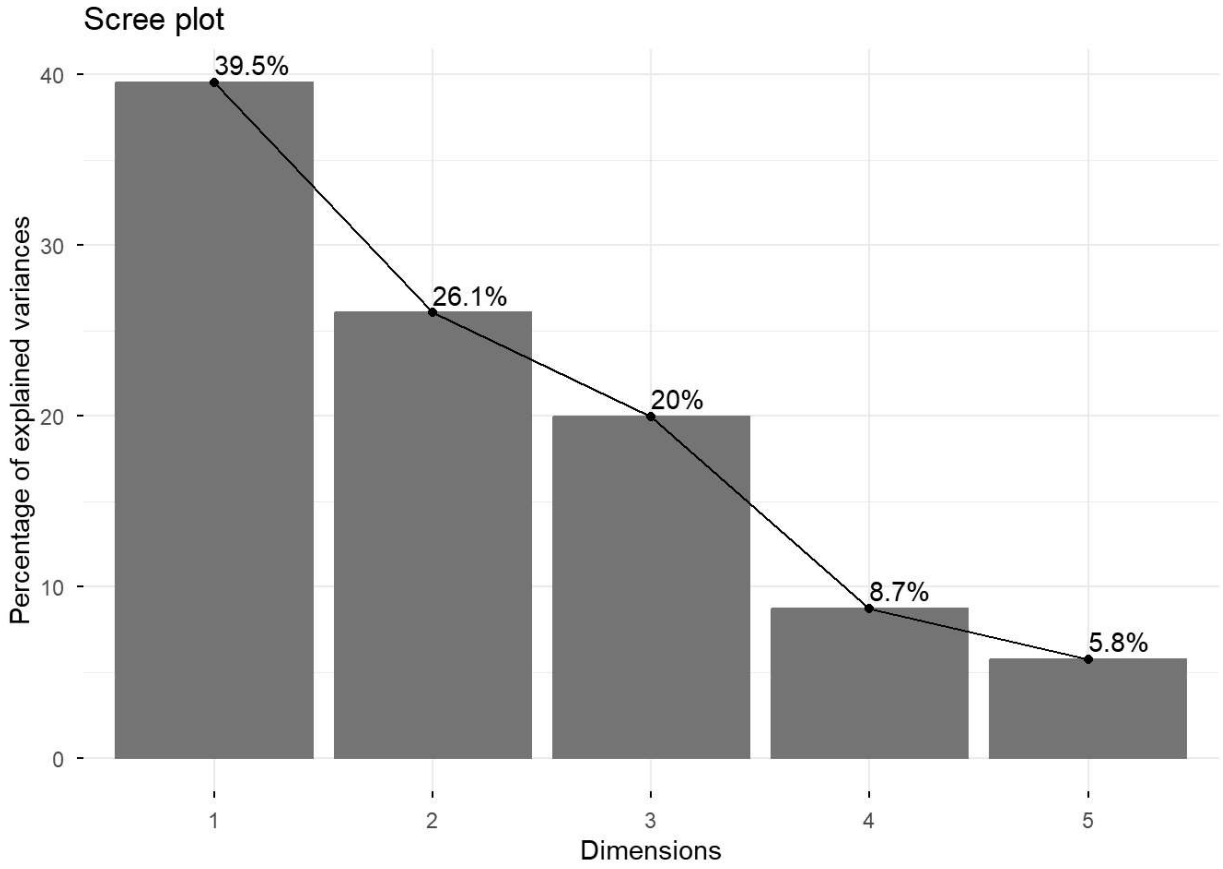
principaux de l'ACP # Variance expliquée (scree plot)

```
# 14) Extraire les valeurs propres (eigenvalues) : part de variance expliquée par chaque axe
eig <- get_eigenvalue(res_pca)
eig
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.9753262        39.506524                    39.50652
## Dim.2  1.3025045        26.050090                    65.55661
## Dim.3  0.9988168        19.976336                    85.53295
## Dim.4  0.4358038         8.716076                    94.24903
## Dim.5  0.2875487         5.750974                   100.00000
```

```
# Graphique des valeurs propres (scree plot) avec étiquettes :
# permet de voir combien d'axes expliquent la majorité de la variance
fviz_eig(res_pca, addlabels = TRUE)
```

```
## Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
## Ignoring empty aesthetic: `width`.
```

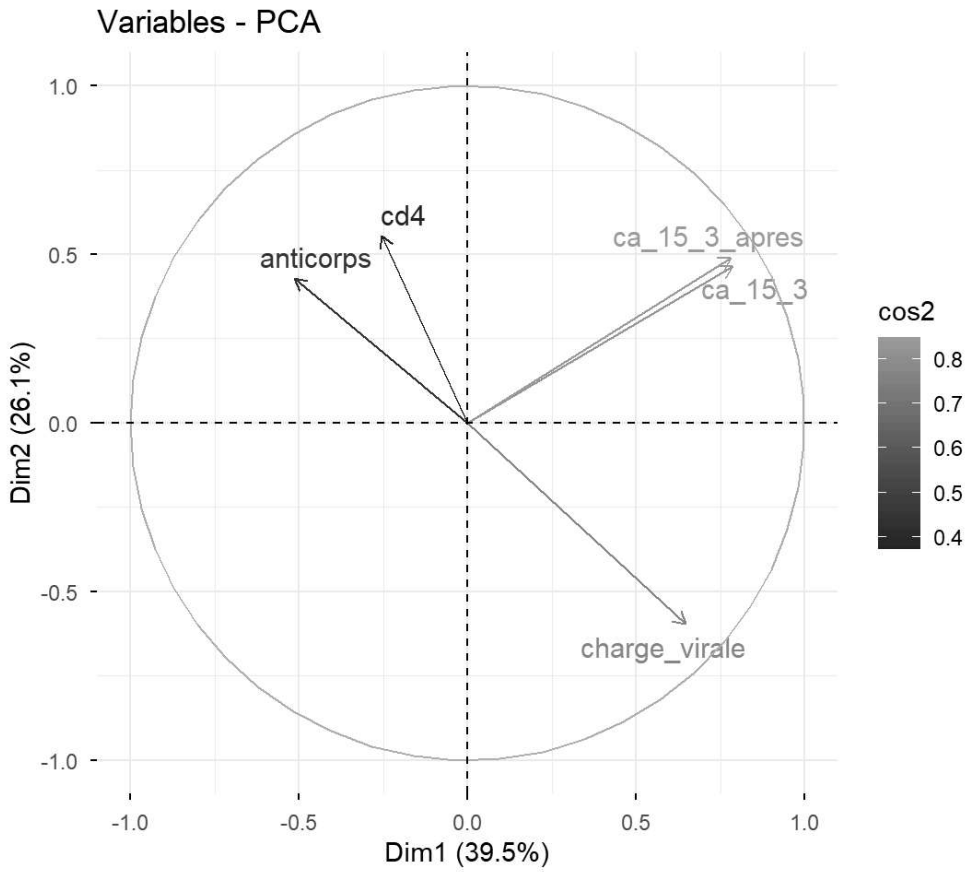


Cercle des corrélations (variables)

```
# Carte des variables : représentation des variables dans le plan factoriel
# col.var = "cos2" colore selon la qualité de représentation sur les axes
# repel = TRUE évite le chevauchement des étiquettes
fviz_pca_var(res_pca, col.var = "cos2", repel = TRUE)
```

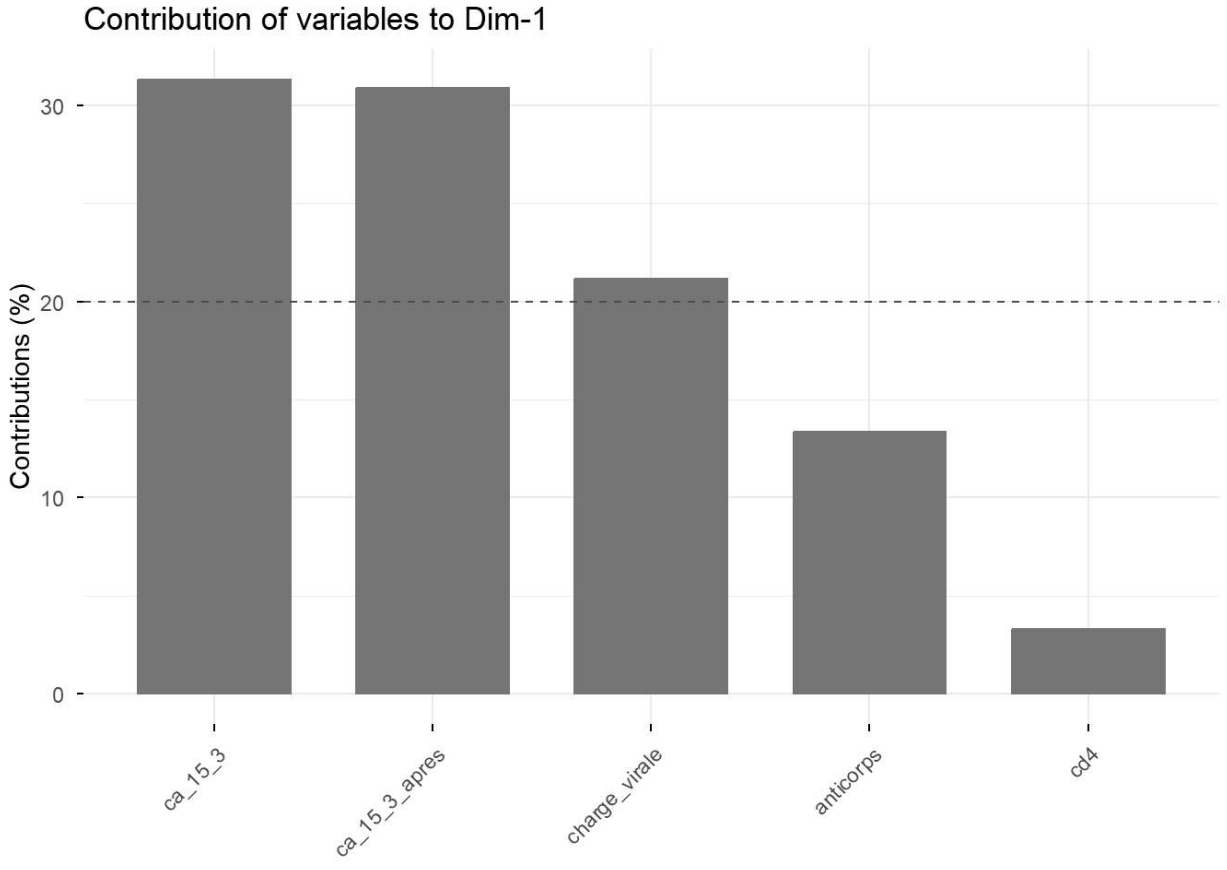
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once per session.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once per session.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

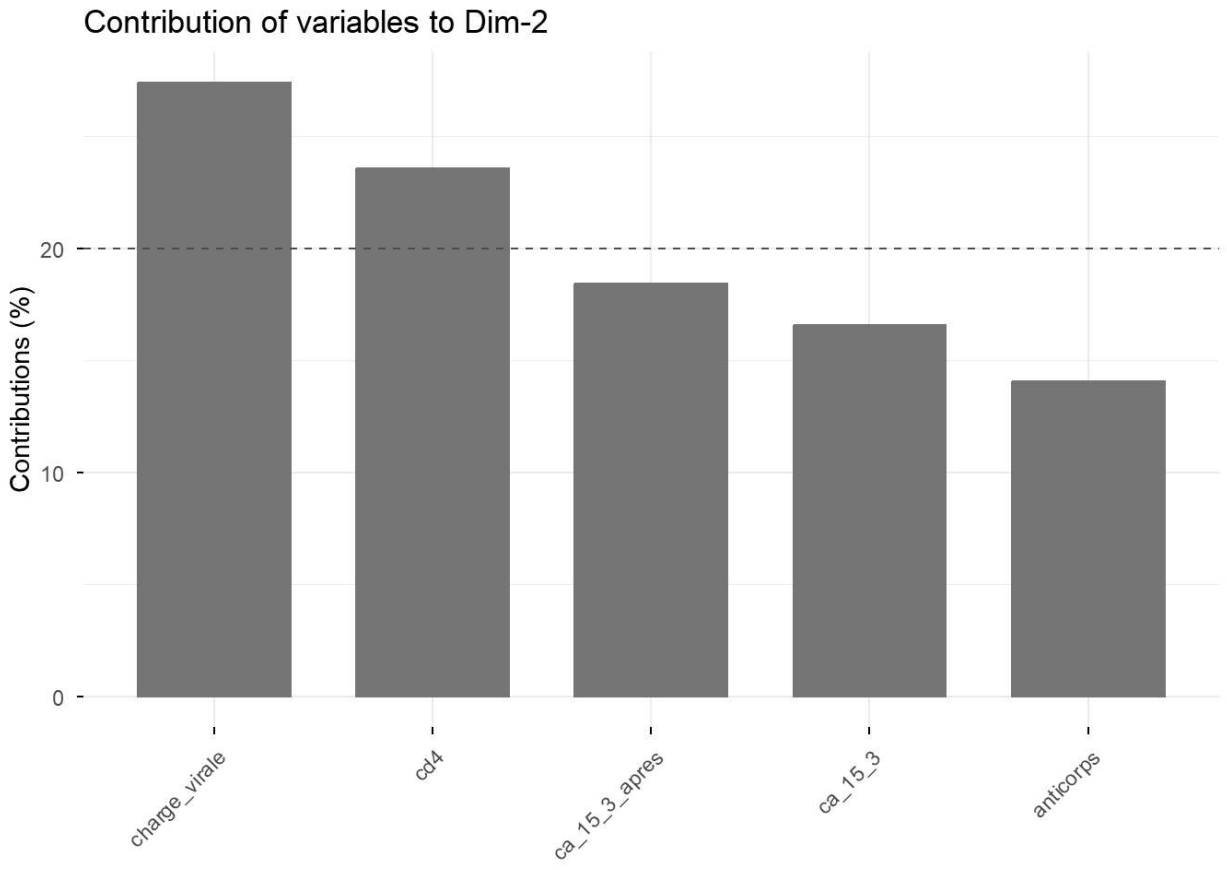


Contributions des variables

```
# Contributions des variables à l'axe 1 : top 10
fviz_contrib(res_pca, choice = "var", axes = 1, top = 10)
```

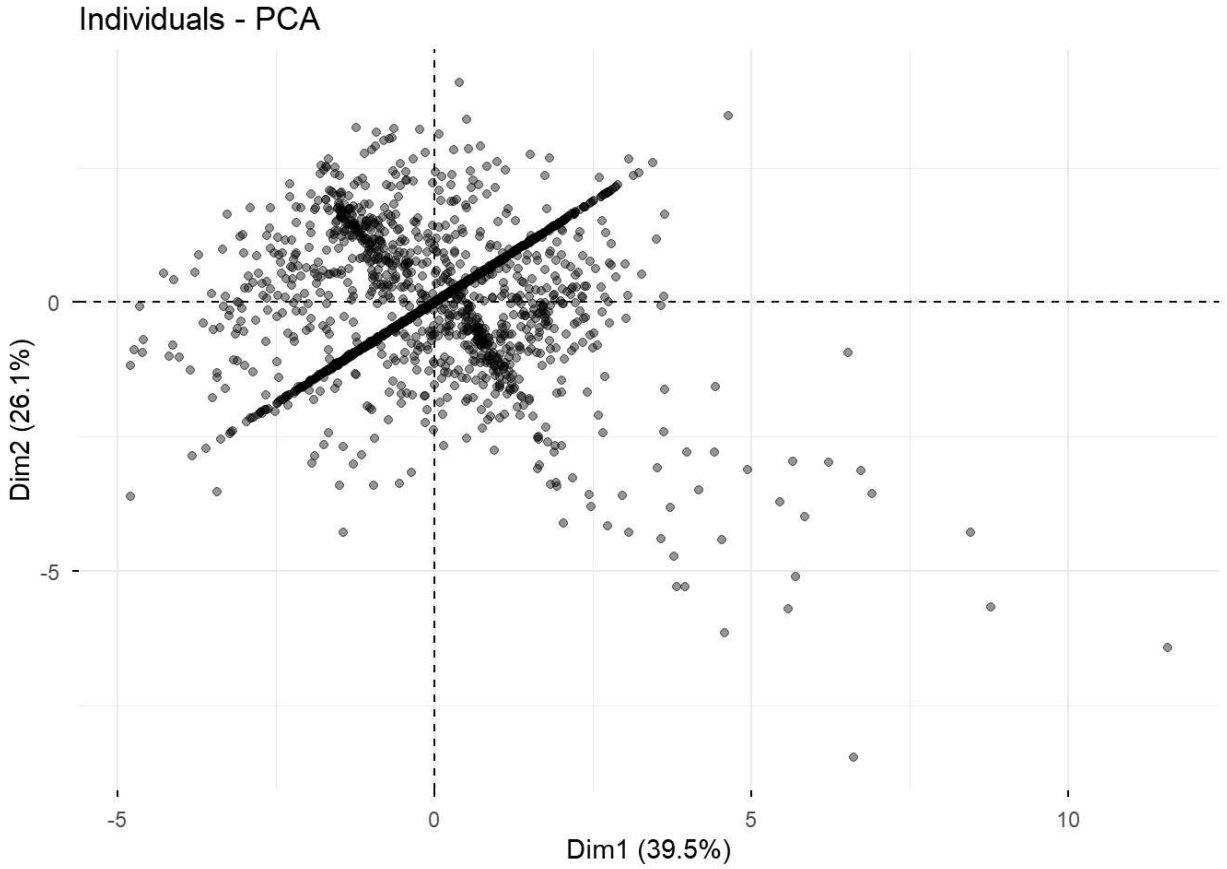


```
# Contributions des variables à l'axe 2 : top 10
fviz_contrib(res_pca, choice = "var", axes = 2, top = 10)
```



Individus (observations) + détection d’atypiques

```
# 19) Carte des individus (points) sur le plan factoriel
fviz_pca_ind(res_pca, geom = "point", alpha.ind = 0.4)
```



Points les plus extrêmes sur Dim1/Dim2 (à interpréter comme profils atypiques)

```
# Identifier des individus "extrêmes" sur la dimension 1
ind <- get_pca_ind(res_pca)
# Trier par valeur absolue de la coordonnée sur Dim.1 (les plus éloignés de l'origine) puis afficher l
es 10 premiers
head(ind$coord[order(abs(ind$coord[,1]), decreasing=TRUE), ], 10)
```

```
##          Dim.1      Dim.2    Dim.3      Dim.4       Dim.5
## 1565 11.570696 -6.4278298 1.959205 6.45147687 -0.22649841
## 1572  8.790426 -5.6755349 1.816114 4.64759487  0.02883189
## 1561  8.463140 -4.2808108 2.022948 3.36715432 -0.45190457
## 880   6.912157 -3.5575087 1.916725 1.39487630  0.30399578
## 878   6.738625 -3.1308486 2.022512 1.79287719  0.58391459
## 2261  6.617267 -8.4565153 1.378470 6.73108962  0.54948721
## 1580  6.525990 -0.9401482 2.752527 0.08996567 -0.94275132
## 975   6.225289 -2.9750343 3.665760 1.60867148  0.15089490
## 763   5.848239 -3.9799075 2.729072 2.05086615  0.46912100
## 756   5.696071 -5.0961105 2.036831 2.13961006  0.33600517
```

Variables qualitatives en supplémentaires

On projette pays, trimestre, smoke, alcohol, csp sur l’espace défini par les biomarqueurs.

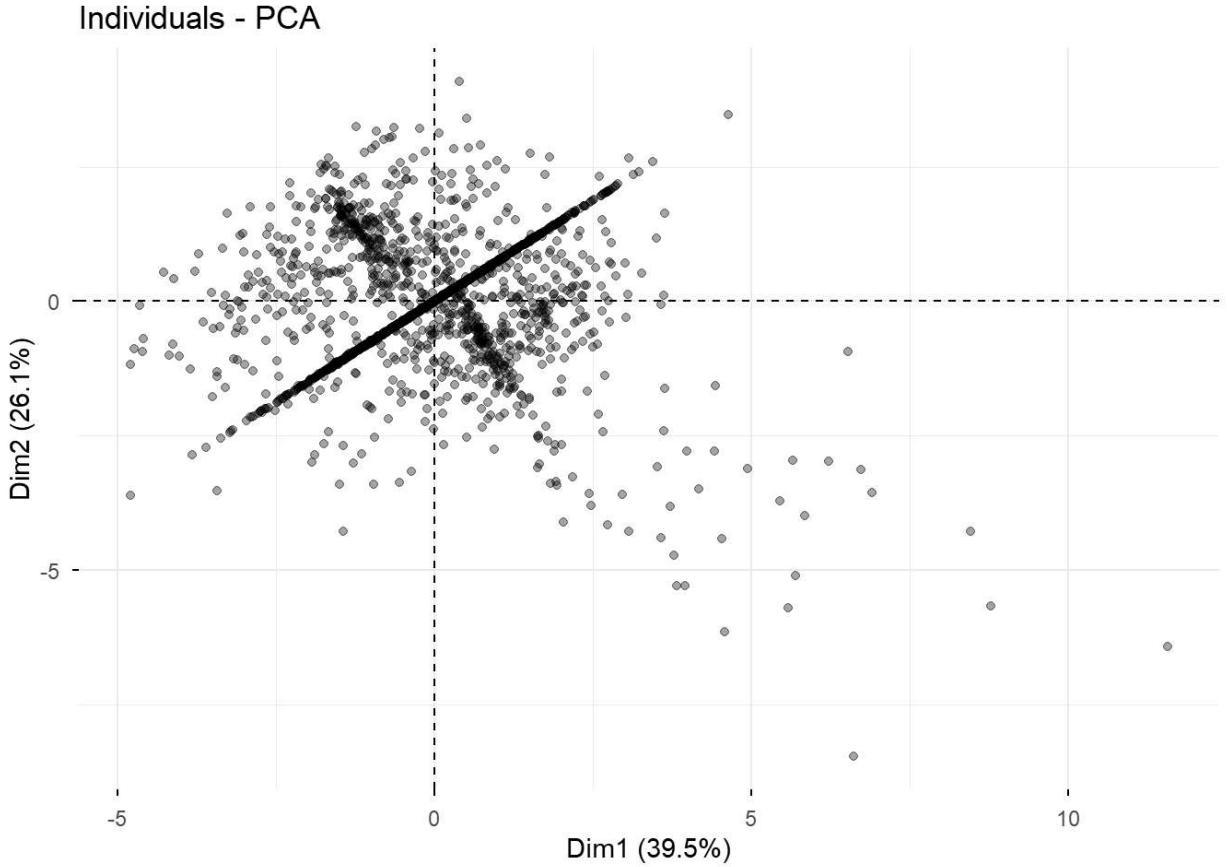
#Fusion scores ACP + variables quali

```
# Construire un tableau de scores : coordonnées ACP + variables descriptives
scores <- as_tibble(res_pca$ind$coord) |>
   # Ajouter des variables qualitatives/temps pour interpréter les positions des individus
  bind_cols(df |> select(nom_pays, trimestre, smoke, alcohol, csp, annee))
# Vérifier le résultat
head(scores)
```

```
## # A tibble: 6 × 11
##       Dim.1     Dim.2     Dim.3     Dim.4     Dim.5 nom_pays trimestre smoke
##       <dbl>     <dbl>     <dbl>     <dbl>     <dbl> <chr>    <fct>     <fct>
## 1 -1.28e+ 0 -3.01e+ 0  1.68e+ 0 -2.93e+ 0  6.88e- 2 ALBANIA  Q1        fumeur l…
## 2  1.33e+ 0  9.99e- 1  2.71e- 1 -1.04e- 1 -1.00e- 1 ALBANIA  Q1        fumer mo…
## 3  7.28e-15  1.36e-14  1.08e-13 -3.34e-14 -6.51e-14 ALBANIA  Q1        gros fum…
## 4  6.73e- 2  2.85e- 2  8.05e- 2  5.42e- 2 -1.24e+ 0 ALBANIA  Q1        <NA>
## 5 -8.05e- 1  3.41e- 1  2.28e+ 0 -1.32e+ 0 -1.46e- 2 ALBANIA  Q2        gros fum…
## 6 -6.80e- 1 -4.99e- 1 -1.69e- 1  2.59e- 2  6.17e- 1 ALBANIA  Q2        <NA>
## # ℹ 3 more variables: alcohol <fct>, csp <fct>, annee <dbl>
```

Visualisation : individus colorés par groupe

```
# Deuxième visualisation des individus (même idée, paramètres légèrement ajustés)
fviz_pca_ind(res_pca, geom = "point", alpha.ind = 0.35)
```



Par pays

```r
library(dplyr)
library(tidyr)
library(ggplot2)

# Récupérer les coordonnées des individus sur les deux premières dimensions
coord2 <- as.data.frame(res_pca$ind$coord)[, 1:2] %>%
  as_tibble() %>%
  mutate(
    # On rattache à chaque individu ses modalités (groupes) pour comparer les profi
    smoke   = df$smoke,
    alcohol = df$alcohol,
    csp     = df$csp
  ) %>%
   # Passer au format long : une colonne "variable" (smoke/alcohol/csp) + une colonne "groupe"
  pivot_longer(cols = c(smoke, alcohol, csp),
               names_to = "variable",
               values_to = "groupe")
# Tracer les individus + ellipses (intervalle de confiance 95%) par groupe,
# séparément pour chaque variable (facettes)
ggplot(coord2, aes(Dim.1, Dim.2)) +
  geom_point(alpha = 0.20, size = 1) +
  stat_ellipse(aes(group = groupe), level = 0.95, type = "norm",
               linewidth = 0.6, alpha = 0.12) +
  facet_wrap(~ variable) +
  theme_minimal() +
  labs(title = "ACP – ellipses par variable (smoke / alcohol / csp)",
       x = "Dim 1", y = "Dim 2")
```