ACP

Imane HADIK

ACP sur données réelles environnementales

Préparation de la data et calcul statistique

Pour appliquer une ACP, nous avons besoin des données quantitatives. Pour cela, nous allons supprimer de notre base de données les variables qualitatives,et remplacer les valeurs manquantes par la moyenne de la variable associée. Après importation de la nouvelle base de données,nous passons au traitement statistique sur l'ensemble de l'échantillon en calculant la matrice variance-covariance, les vecteurs écart type, la moyenne et la médiane des variables.

```
library(readxl)
data<-read_excel("newdata.xlsx")
##Résultats statistiques
print("matrice var-cov")
```

```
## [1] "matrice var-cov"
```

```
Var_cov<-var(data)
Var_cov
```

```
##                          B          T          E          X      9_ane     10_ane
## B                 1092.762   1943.447   4060.882   18438.07   1893.358   6509.247
## T                 1943.447  14596.592  31284.053  103599.09   7137.519  32527.150
## E                 4060.882  31284.053 183737.847  511385.31  28783.211 191077.665
## X                18438.068 103599.088 511385.310 1516365.42  90653.168 550112.652
## 9_ane             1893.358   7137.519  28783.211   90653.17   9624.806  33825.784
## 10_ane            6509.247  32527.150 191077.665  550112.65  33825.784 225365.776
## 13_ane           39607.256  91204.895 219807.699  915721.96  94202.514 324685.723
## 14_ane           65906.682 144389.199 326752.706 1409659.42 149019.620 511700.146
## 1_M_2_PA          8002.635  44243.896 219612.201  633639.88  39146.260 228561.149
## BTM              11185.156 109855.371 796659.168 2126123.03 112189.612 833795.038
## FormicAcid       10998.883  10037.976  37822.917  165467.18  21646.371  58456.696
## aceticacid        3626.384  10666.477   7605.827   45369.38   4579.265  10930.421
## NonaDecanoicAc   15031.980  -1402.488  17728.472  125639.64  23303.884  52642.969
## Tot_OcNoDecana   19094.153  41828.314  90730.293  381017.58  49150.034 135919.558
##                     13_ane     14_ane   1_M_2_PA        BTM FormicAcid
## B                  39607.26   65906.68   8002.635   11185.16   10998.88
## T                  91204.89  144389.20  44243.896  109855.37   10037.98
## E                 219807.70  326752.71 219612.201  796659.17   37822.92
## X                 915721.96 1409659.42 633639.882 2126123.03  165467.18
## 9_ane              94202.51  149019.62  39146.260  112189.61   21646.37
## 10_ane            324685.72  511700.15 228561.149  833795.04   58456.70
```

```
## 13_ane         2196931.27 3285470.01 397491.823  633054.16  365995.02
## 14_ane         3285470.01 5616209.86 624244.108  903732.61  600428.34
## 1_M_2_PA        397491.82  624244.11 338676.298  965666.65   79388.57
## BTM             633054.16  903732.61 965666.654 3657296.01  118291.08
## FormicAcid      365995.02  600428.34  79388.571  118291.08  297007.47
## aceticacid       97792.07  134430.15  24990.076   22492.11  181674.16
## NonaDecanoicAc  355247.98  687663.81  77738.132   63500.60  231260.79
## Tot_OcNoDecana  900414.68 1384993.21 188339.209  245457.22  201250.85
##                aceticacid NonaDecanoicAc Tot_OcNoDecana
## B                3626.384      15031.980       19094.15
## T               10666.477      -1402.488       41828.31
## E                7605.827      17728.472       90730.29
## X               45369.381     125639.642      381017.58
## 9_ane            4579.265      23303.884       49150.03
## 10_ane          10930.421      52642.969      135919.56
## 13_ane          97792.070     355247.980      900414.68
## 14_ane         134430.147     687663.812     1384993.21
## 1_M_2_PA        24990.076      77738.132      188339.21
## BTM             22492.110      63500.599      245457.22
## FormicAcid     181674.159     231260.793      201250.85
## aceticacid     497569.029      35293.562       42905.64
## NonaDecanoicAc  35293.562     546756.522      232474.54
## Tot_OcNoDecana  42905.643     232474.541      637398.31
```

```
# ecart type data
print("vecteur ecart type")
```

```
## [1] "vecteur ecart type"
```

```
ecart_type<-sqrt(diag(Var_cov))
ecart<-c(ecart_type)#vecteur ecart type
ecart
```

```
##             B             T             E             X         9_ane
##      33.05695     120.81636     428.64653    1231.40790     98.10610
##        10_ane        13_ane        14_ane       1_M_2_PA           BTM
##     474.72705    1482.20487    2369.85440     581.95902 1912.40582
##    FormicAcid    aceticacid NonaDecanoicAc Tot_OcNoDecana
##     544.98392     705.38573     739.42986     798.37229
```

```
##moyenne
print("moyenne de chaque variable explicative")
```

```
## [1] "moyenne de chaque variable explicative"
```

```
mean <- c(colMeans(data))
mean
```

```
##             B             T             E             X         9_ane
##      55.78702     208.31002     233.54710     880.68739     104.17355
##        10_ane        13_ane        14_ane       1_M_2_PA           BTM
##     267.22821    1000.06894    1872.59600     311.21119     578.28285
##    FormicAcid    aceticacid NonaDecanoicAc Tot_OcNoDecana
##     484.70289     368.18594     737.96845     644.23034
```

```
#médiane
print("médiane")
```

```
## [1] "médiane"
```

```
mediane<-rep(0,14)
names(mediane)<-c("B","T","E","X","9_ane","10_ane","13_ane","14_ane","1_M_2_PA","BTM","FormicAcid","ace
for (i in 1:14){
  mediane[i]<-median(as.numeric(unlist(data[,i])))
}
mediane
```

```
##             B             T             E             X         9_ane
##      52.12241     183.78986     153.65149     596.32263      74.88167
##        10_ane        13_ane        14_ane       1_M_2_PA           BTM
##     179.96478     240.86632    1056.37024     131.19239     319.75447
##    FormicAcid    aceticacid NonaDecanoicAc Tot_OcNoDecana
##     306.48797     184.86641     547.49311     352.05778
```

Calcul de la corrélation entre les variables, et graphique de corrélation

On vérifie dans cette étape que nos données sont adaptées à la réduction de dimension. Pour ce faire, on se base premièrement sur le graphique des corrélations et ensuite sur les deux tests KMO et Bartlett, pour confirmer qu'il existe une forte corrélation entre les variables et que nous pouvons donc faire une ACP.

```
##correlation
print("matrice de corrélation")
```

```
## [1] "matrice de corrélation"
```

```
mcor<-cor(data)
mcor
```

```
##                        B           T          E          X      9_ane
## B              1.0000000  0.48661341 0.28658818 0.45295041 0.58381324
## T              0.4866134  1.00000000 0.60408487 0.69635108 0.60217893
## E              0.2865882  0.60408487 1.00000000 0.96882871 0.68445348
## X              0.4529504  0.69635108 0.96882871 1.00000000 0.75038657
## 9_ane          0.5838132  0.60217893 0.68445348 0.75038657 1.00000000
## 10_ane         0.4147859  0.56712173 0.93900225 0.94103487 0.72628636
## 13_ane         0.8083579  0.50931231 0.34596754 0.50171081 0.64782583
## 14_ane         0.8412885  0.50429806 0.32166086 0.48304835 0.64095241
## 1_M_2_PA       0.4159851  0.62926738 0.88036902 0.88419523 0.68564904
## BTM            0.1769291  0.47546165 0.97183647 0.90283087 0.59796614
## FormicAcid     0.6105231  0.15245326 0.16190939 0.24656206 0.40486050
## aceticacid     0.1555193  0.12516087 0.02515478 0.05223171 0.06617183
## NonaDecanoicAc 0.6149735 -0.01569916 0.05593389 0.13798369 0.32124422
## Tot_OcNoDecana 0.7234893  0.43364981 0.26512309 0.38755882 0.62751247
##                    10_ane     13_ane     14_ane   1_M_2_PA        BTM
## B              0.41478588 0.80835795 0.84128854 0.41598512 0.17692907
```

```
## T              0.56712173 0.50931231 0.50429806 0.62926738 0.47546165
## E              0.93900225 0.34596754 0.32166086 0.88036902 0.97183647
## X              0.94103487 0.50171081 0.48304835 0.88419523 0.90283087
## 9_ane          0.72628636 0.64782583 0.64095241 0.68564904 0.59796614
## 10_ane         1.00000000 0.46143547 0.45483083 0.82730572 0.91840717
## 13_ane         0.46143547 1.00000000 0.93533591 0.46081601 0.22333283
## 14_ane         0.45483083 0.93533591 1.00000000 0.45262693 0.19940600
## 1_M_2_PA       0.82730572 0.46081601 0.45262693 1.00000000 0.86767039
## BTM            0.91840717 0.22333283 0.19940600 0.86767039 1.00000000
## FormicAcid     0.22594701 0.45308873 0.46489603 0.25031213 0.11349801
## aceticacid     0.03264121 0.09353383 0.08041709 0.06087634 0.01667337
## NonaDecanoicAc 0.14996828 0.32413535 0.39242577 0.18065281 0.04490563
## Tot_OcNoDecana 0.35861836 0.76090227 0.73201592 0.40536186 0.16076456
##                FormicAcid aceticacid NonaDecanoicAc Tot_OcNoDecana
## B               0.6105231 0.15551930     0.61497351     0.72348933
## T               0.1524533 0.12516087    -0.01569916     0.43364981
## E               0.1619094 0.02515478     0.05593389     0.26512309
## X               0.2465621 0.05223171     0.13798369     0.38755882
## 9_ane           0.4048605 0.06617183     0.32124422     0.62751247
## 10_ane          0.2259470 0.03264121     0.14996828     0.35861836
## 13_ane          0.4530887 0.09353383     0.32413535     0.76090227
## 14_ane          0.4648960 0.08041709     0.39242577     0.73201592
## 1_M_2_PA        0.2503121 0.06087634     0.18065281     0.40536186
## BTM             0.1134980 0.01667337     0.04490563     0.16076456
## FormicAcid      1.0000000 0.47258812     0.57388033     0.46253924
## aceticacid      0.4725881 1.00000000     0.06766621     0.07618725
## NonaDecanoicAc  0.5738803 0.06766621     1.00000000     0.39379750
## Tot_OcNoDecana  0.4625392 0.07618725     0.39379750     1.00000000
```

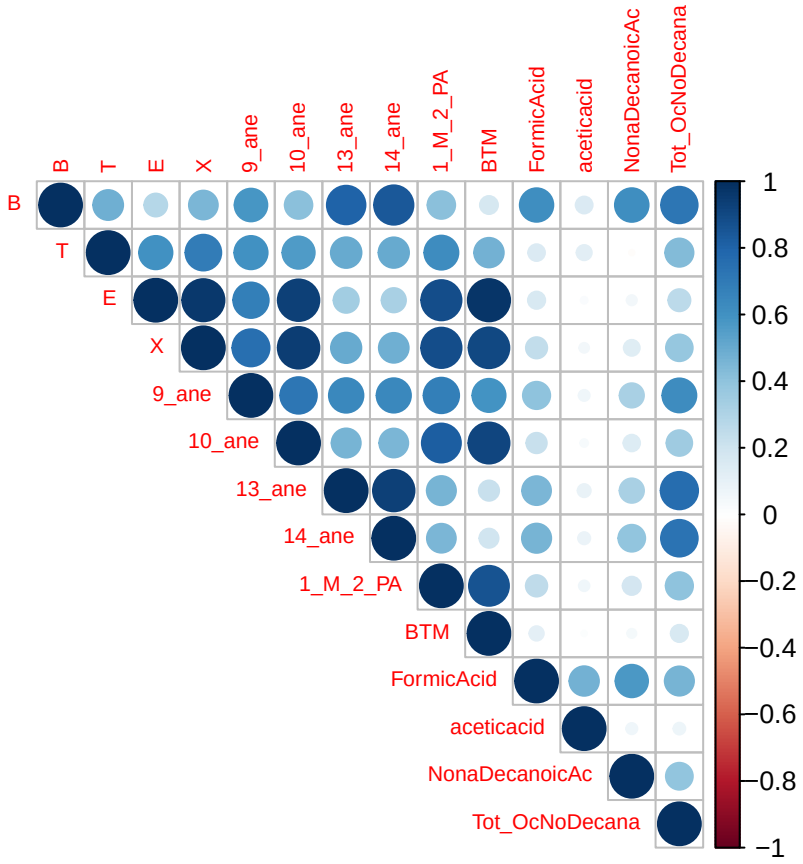
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
print("graphe des corrélations")
```

```
## [1] "graphe des corrélations"
```

```
corrplot(mcor, method="circle", type = "upper", number.cex = 0.6, tl.cex = 0.7)
#test KMO et Bartlett
library(psych)
```




```
KMO(mcor)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = mcor)
## Overall MSA =  0.8
## MSA for each item =
##              B              T              E              X          9_ane
##           0.86           0.84           0.73           0.78           0.94
##         10_ane         13_ane         14_ane       1_M_2_PA            BTM
##           0.81           0.85           0.83           0.79           0.66
##     FormicAcid     aceticacid NonaDecanoicAc Tot_OcNoDecana
##           0.80           0.43           0.67           0.88
```

```
cortest.bartlett(mcor)
```

```
## Warning in cortest.bartlett(mcor): n not specified, 100 used
```

```
## $chisq
## [1] 1958.261
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

D'après le graphique de corrélation ci-dessus, nous remarquons qu'il y a une forte corrélation entre les variables. Nous pouvons confirmer ce résultat à l'aide des deux tests KMO (une valeur de 0.8»0.5) et bartlette(p-value très significative =0). Nous pouvons procéder alors par une ACP.

Centrer et réduire les données pour faire une ACP

Nous remarquons que nos données sont dans des ordres de grandeurs différentes.Il est nécessaire alors de centrer réduire les données avant d'effectuer une ACP.

```
Xcentre<- data.frame()
for(i in 1:14){
  for(j in 1:139){
    Xcentre[j,i] <- (data[j,i] - mean[i])/ecart_type[i]} #our new data
}
```

Réduction de dimension et ACP

Nous cherchons d'abord le nombre optimal d'axes à extraire. On sait que la recherche d'axes portant le maximum d'inertie équivaut à la construction de nouvelles variables (auxquelles sont associés ces axes) de variance maximale. Nous effectuons alors un changement de repère où le premier axe apporte le plus possible d'inertie totale du nuage, le deuxième axe le plus possible d'inertie non prise en compte par le premier axe, et ainsi de suite. l'Inertie totale = la somme des variances des variables étudiées, dans notre cas il s'agit de variables centrées réduites donc l'inertie totale=p=14. Nous pouvons vérifier facilement sur R que la somme des valeurs propres = 14

```
#facteurs à extraire dans l'ACP
library(paran)
```

```
## Le chargement a nécessité le package : MASS
```

```
print("Graphique des valeurs propres et nombre optimal de facteurs à extraire")
```

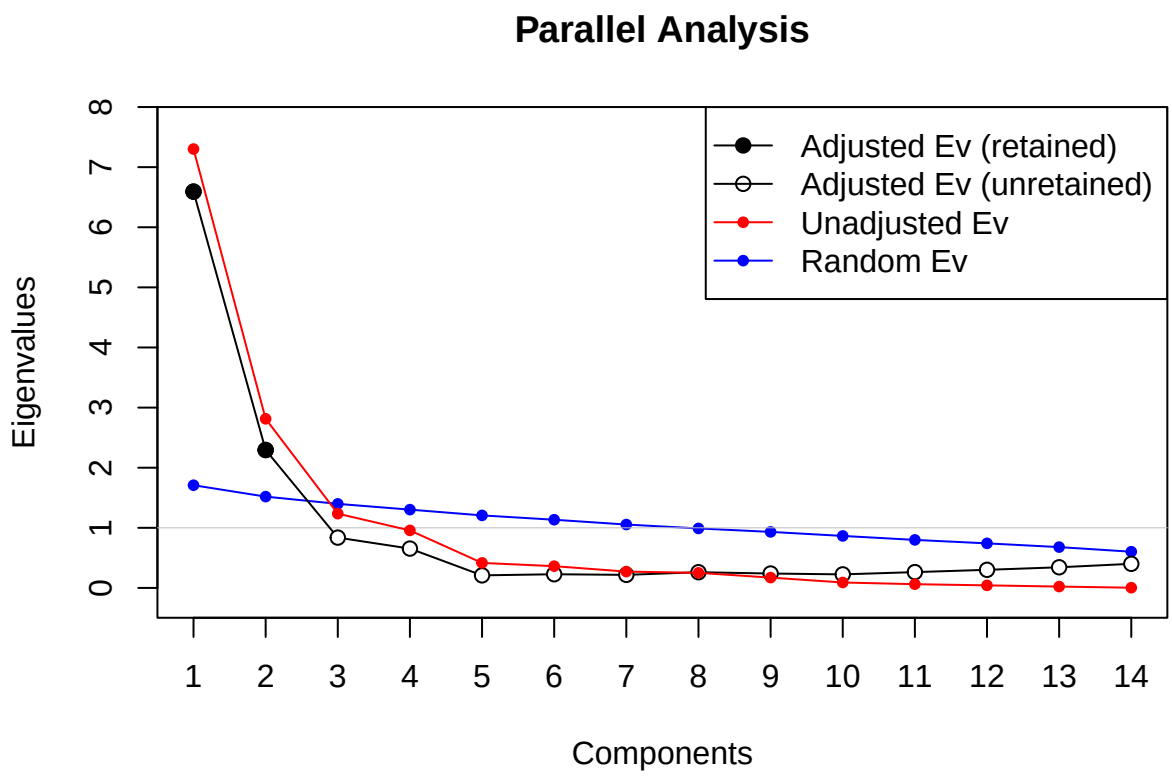
```
## [1] "Graphique des valeurs propres et nombre optimal de facteurs à extraire"
```

```
P<-paran(x= Xcentre,cfa = FALSE,graph = TRUE,color = c("black","red","blue"),centile = 95)
```

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
##
##
## Results of Horn's Parallel Analysis for component retention
## 420 iterations, using the 95 centile estimate
##
## ------------------------------------------------
## Component   Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## ------------------------------------------------
## 1           6.592808    7.301591      0.708782
## 2           2.294269    2.813064      0.518795
```

```
## ------------------------------------------------
##
## Adjusted eigenvalues > 1 indicate dimensions to retain.
## (2 components retained)

## Warning in if (color == FALSE) {: la condition a une longueur > 1 et seul le
## premier élément est utilisé
```




```
val.extr<- P$Retained #nombre de facteurs optimal à extraire
valeursP<-P$Ev##valeurs propres
print("Valeurs propres")
```

```
## [1] "Valeurs propres"
```

```
valeursP
```

```
##  [1] 7.301591426 2.813064643 1.235401775 0.956988431 0.415870586 0.361835185
##  [7] 0.271754346 0.251179266 0.171429214 0.090305751 0.062098038 0.042566337
## [13] 0.022408722 0.003506281
```

```
valeursPadj<-P$AdjEv##valeurs propres ajustées
print("somme des valeurs propres")
```

```
## [1] "somme des valeurs propres"
```

```
sum(valeursP)
```

```
## [1] 14
```

Si nous optons pour la méthode basée sur le critère de Kaiser, nous retenons les axes associés à des valeurs propres supérieurs à 1. Par conséquent, les trois premiers axes sont retenus. Par ailleurs en utilisant la fonction paran(), nous obtenons que 2 axes à extraire (en se basant sur les valeurs propres ajustées).Nous pouvons confirmer ce résultat à l'aide du critère du coude qui est une autre méthode permettant la détermination du nombre optimal des axes, Le graphe ci-dessous nous indique que seulement les 2 premiers axes sont retenus.

```
library( factoextra)
```

```
## Le chargement a nécessité le package : ggplot2
```

```
##
## Attachement du package : 'ggplot2'
```

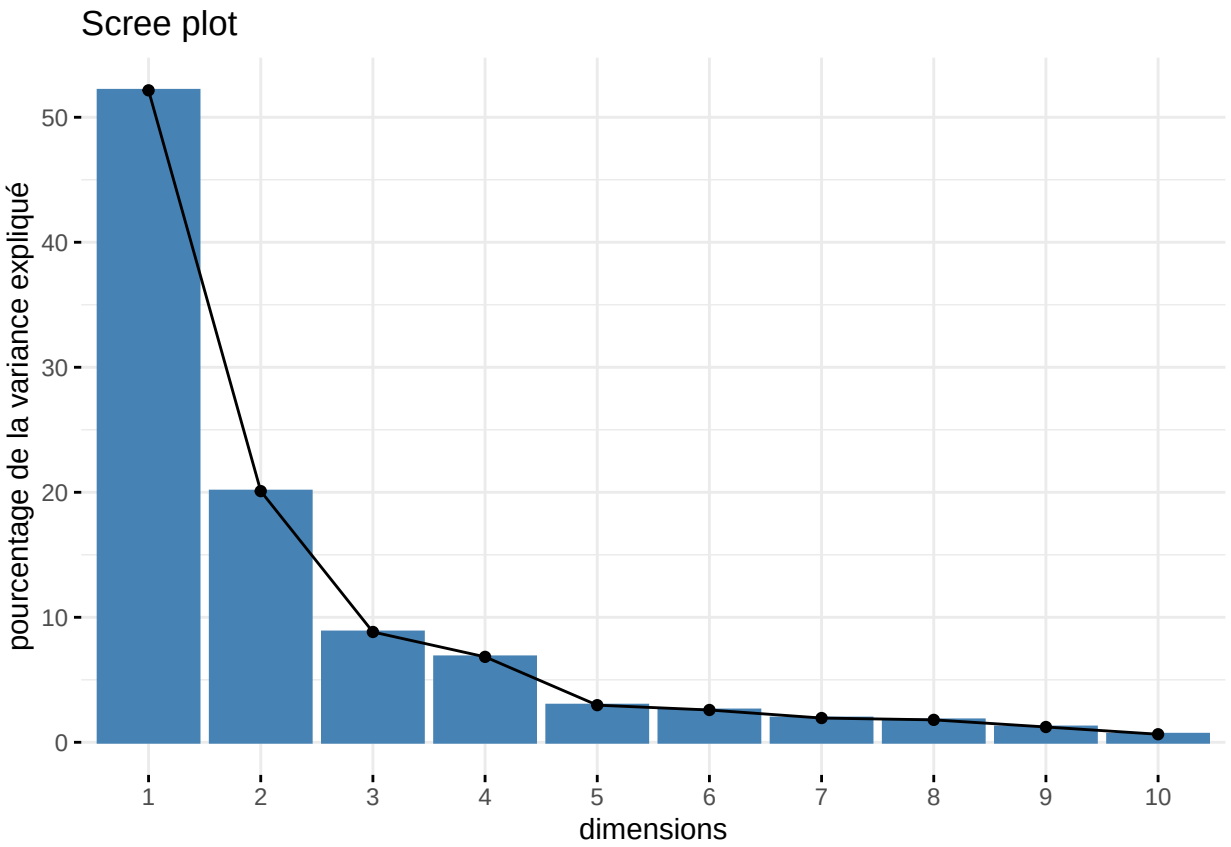
```
## Les objets suivants sont masqués depuis 'package:psych':
##
##     %+%, alpha
```

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

```
library( ggrepel)
library( ggplot2)
acp<-prcomp(Xcentre,scale=TRUE)#acp
print("Diagramme des valeurs propres")
```

```
## [1] "Diagramme des valeurs propres"
```

```
fviz_eig(acp, xlab="dimensions",ylab="pourcentage de la variance expliqué")
```



Ensuite, nous obtenons une matrice qui illustre les coordonnées factorielles des variables. Cela nous aide à savoir quelles sont les variables qui participent le plus à la formation d'un axe.

```
##coordonnées
print("Coordonnées factorielles des variables dans les composantes retenues")
```

```
## [1] "Coordonnées factorielles des variables dans les composantes retenues"
```

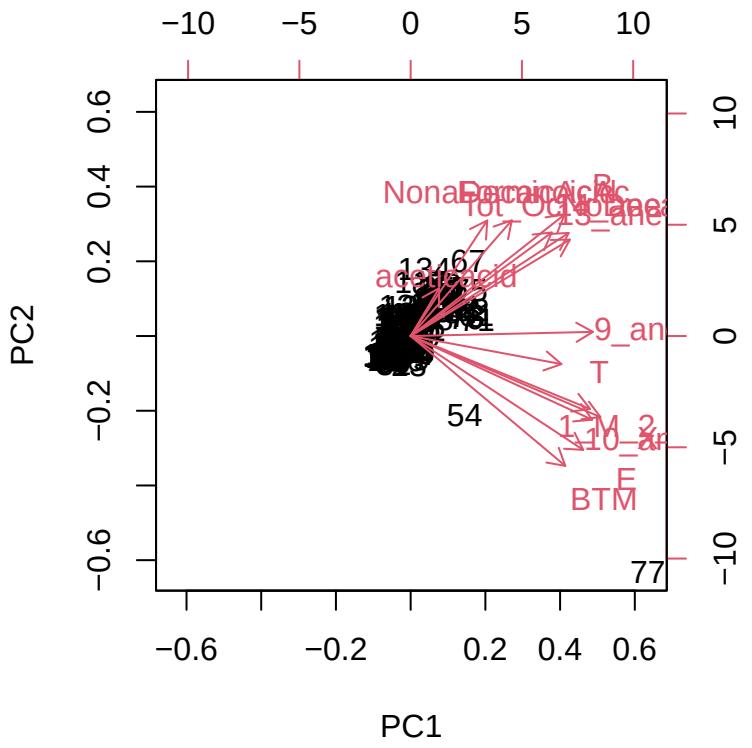
```
acp$rotation[,1:2]
```

```
##                       PC1         PC2
## B              0.27128443  0.34492654
## T              0.26587592 -0.07968041
## E              0.30422765 -0.32337752
## X              0.33456897 -0.23014750
## 9_ane          0.32147021  0.01176707
## 10_ane         0.32011228 -0.23769361
## 13_ane         0.28082404  0.27396049
## 14_ane         0.27861305  0.29285578
## 1_M_2_PA       0.31644307 -0.20716974
## BTM            0.27268123 -0.36885693
## FormicAcid     0.17830744  0.32854752
## aceticacid     0.05032867  0.13614128
## NonaDecanoicAc 0.13497935  0.32817098
## Tot_OcNoDecana 0.24876564  0.29416221
```

Représentation des individus et des variables dans le premier plan factoriel

Nous pouvons représenter les individus et les variables dans le premier plan factoriel.Les deux graphes ci-dessous illustrent ces résultats:

```
biplot(acp)
```




```
cat("\n\n")
```

Qualité de représentation des variables

A partir du cercle de corrélation suivant, nous pouvons affirmer les résultats de corrélation des variables vu précedemment et voir la contribution de chaque variable dans la constitution des composantes principales. On observe que *les varibales : X, 9_ane,1_M_2_PA, 10_ane et E sont les plus proches du bord du cercle de corrélation et du 1er axe donc ils contribuent le plus dans la constitution de la première composantes. Les variables qui contribuent le moins dans cet axe sont : aceticacid,FormicAcid, NonaDecanoicAc.

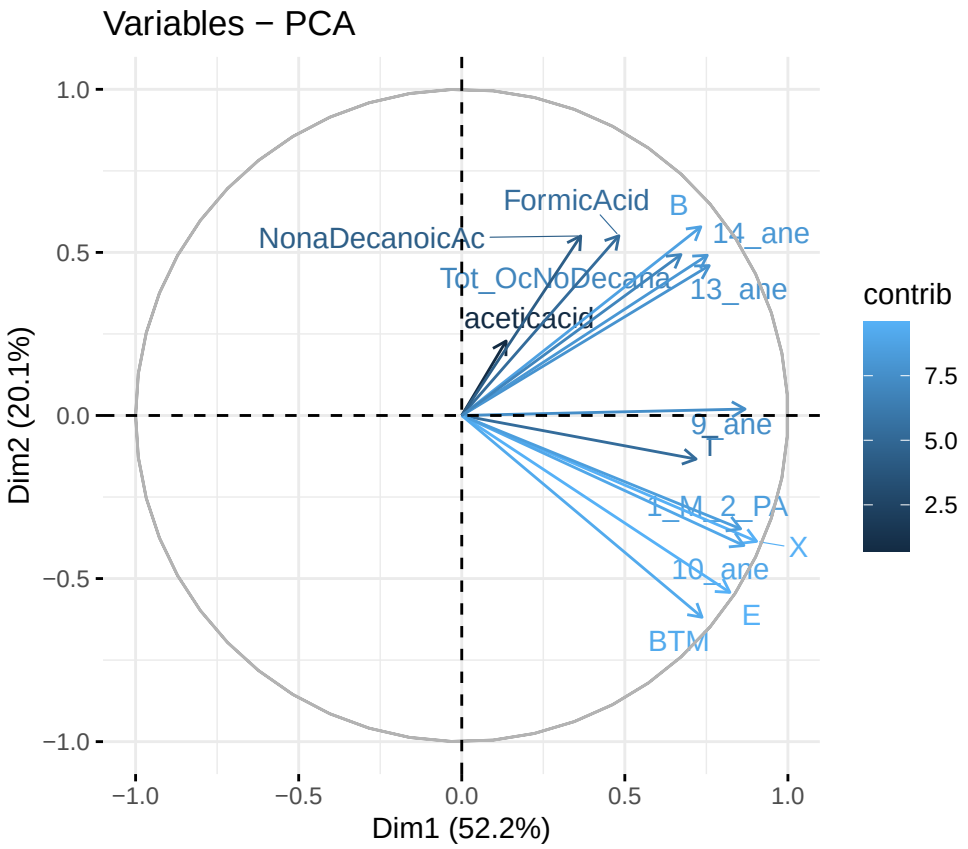
*les variables B,13_ane,14_ane,Tot_OcNoDecana, FormicAcid étant regroupées dans le même sens, ces variables sont corrélées positivement,et elles contribuent fortement et positivement à la construction des deux axes et donc elles sont bien représentées dans les deux axes.

- les variables BTM,X,E,1_M_2_PA regroupées dans le même sens sont fortement corrélées, ces variables sont corrélés potivement avec le premier axe et négativement avec le deuxième axe. Dans ce dernier, nous constatons que les variables : 9_ane (elle est ~perpendiculaire),T et aceticacid sont mal représentées représentées.

```
##Qualité de représentation des variables
print("Cercle de corrélation")
```

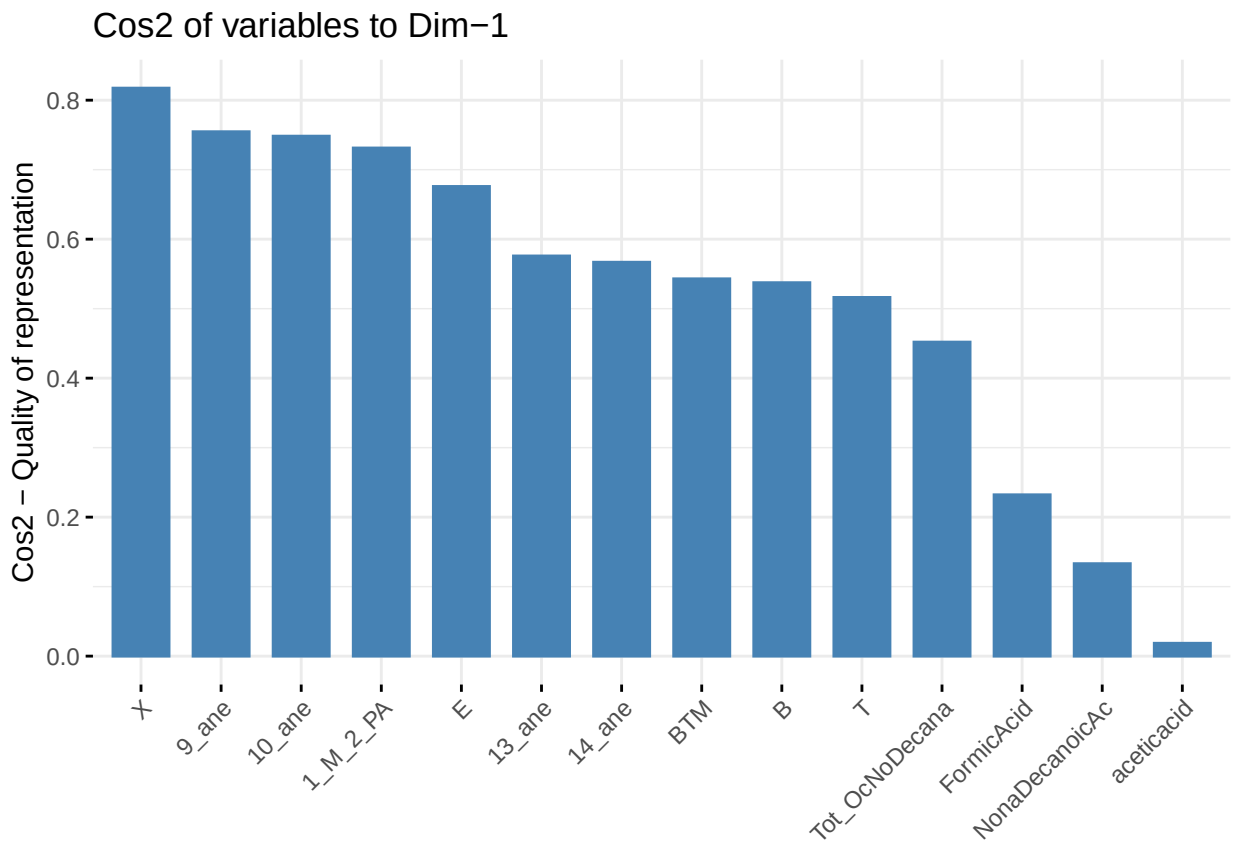
```
## [1] "Cercle de corrélation"
```

```
fviz_pca_var(acp,col.var = "contrib", radient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE)
```

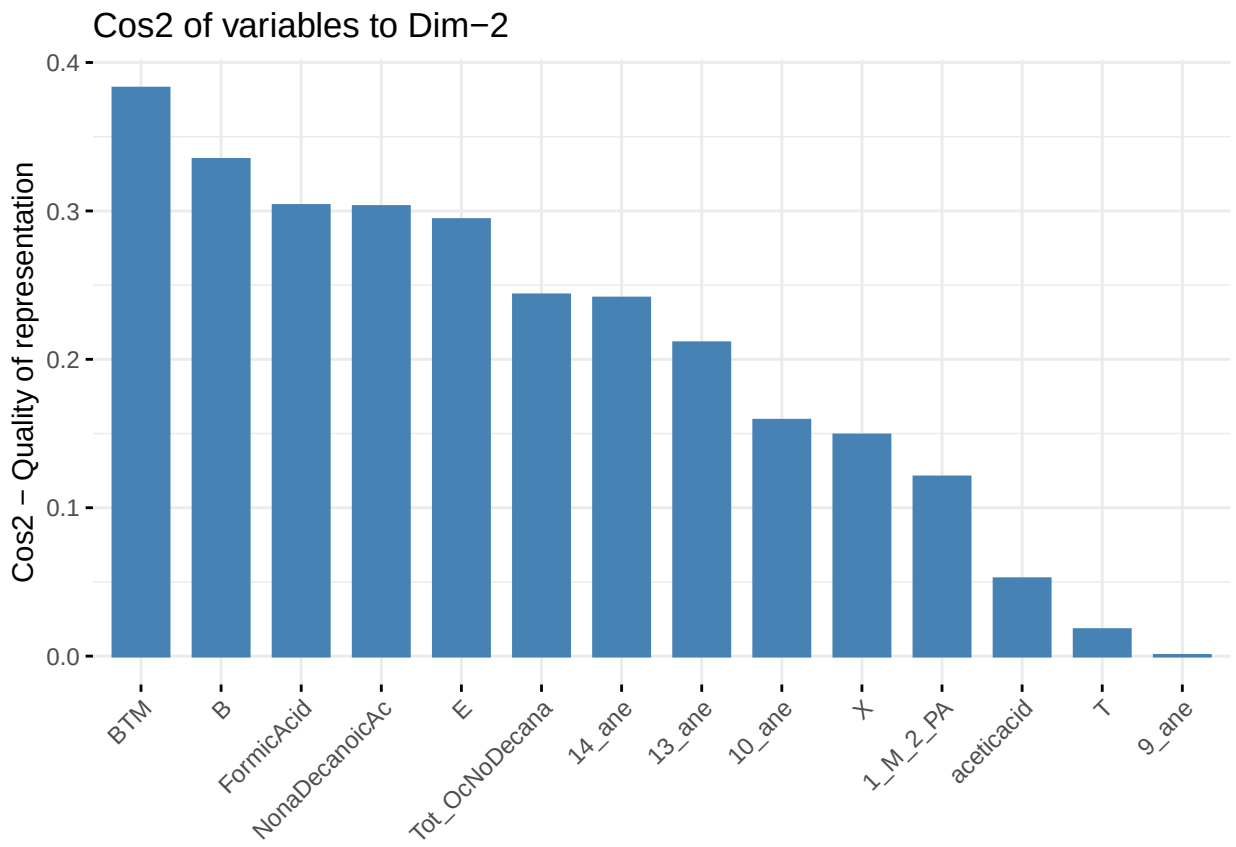


La qualité de représentation des variables sur le plan de l'ACP s'appelle cos2 (cosinus carré). Un cos2 élevé indique une bonne représentation de la variable sur l' axe principal en question. Un faible cos2 indique que la variable n'est pas bien représentée par l'axe principal. Nou confirmons les résultats de cercle de corrélation par les 2 diagrammes ci-dessous, qui repérsentent les valeurs cos2 des variables sur les deux axes. Comme dernière remarque, la variable aceticacid est mal représentée dans les deux axes.

```
fviz_cos2(acp, choice="var", axes = 1 )##Qualité de la représentation axe1
```




```
fviz_cos2(acp, choice="var", axes = 2 )##Qualité de la représentation ax2
```



Qualité de représentation des individus

En utilisant le cos2 pour évaluer la réprésentation des individus dans les deux composantes principales, nous remarquons que : * Certaines individus(en vert ayant un cos2 le plus petit) sont mal représentés par les deux axes(ils sont proches de L'origine du plan).

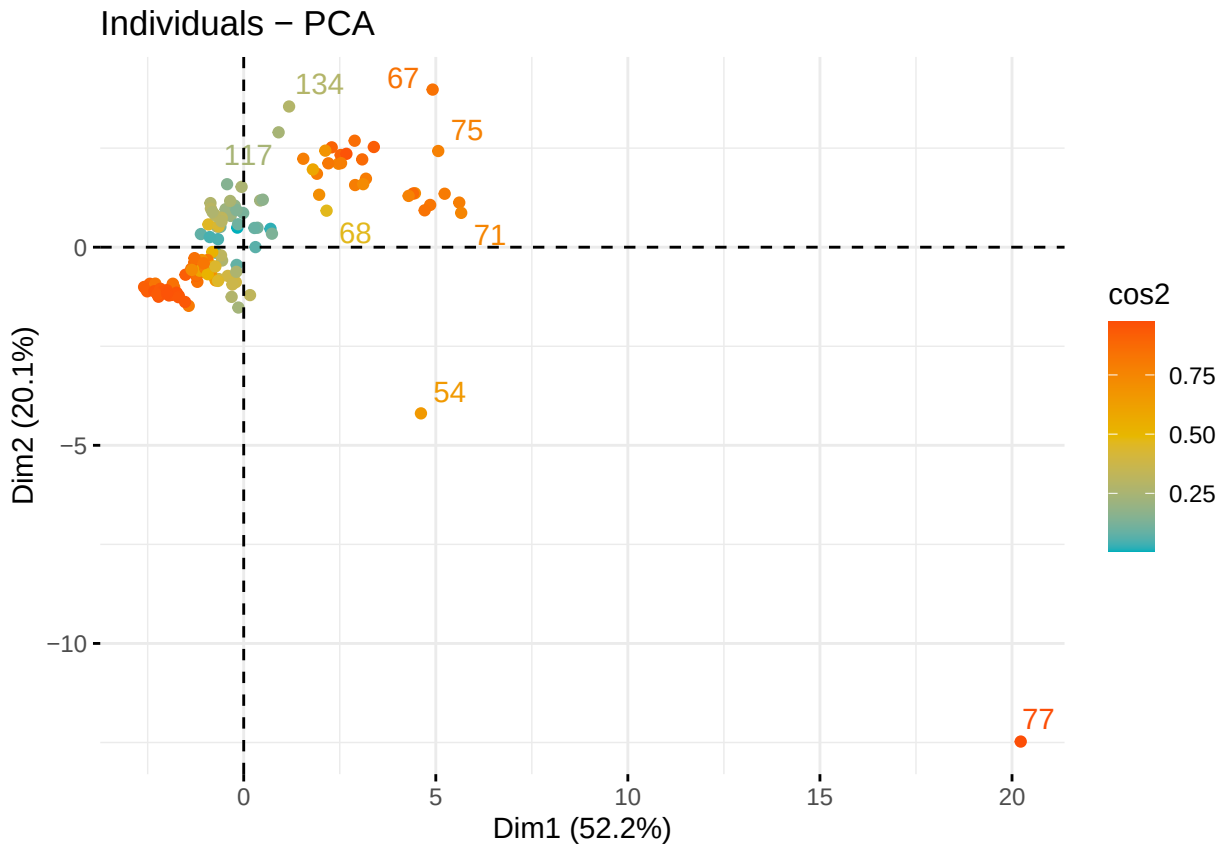
- L'individu 77 est un point aberrant.
- Des individus qui sont bien représentés(positivement par exemple les individus 65,75,61,68,71) dans le côté droit du plan. Ces individus est lié positivement aux composés chimiques qui expliquent le mieux les deux axes et qui contribuent positivement à la construction des deux axes, parmi ces composés nous pouvons citer :B,13_ane,14_ane,Tot_OcNoDecana.
- L'individu 54 ayant un cos2 moyennement élevé , est positivement représenté dans le premier axe et négativement représenté dans le deuxième axe . Cet individu est lié potivement aux composés chimiques : BTM,X,E,1_M_2_PA
- les autres individus sont représentés fortement et négativement dans les deux axes.

```
#qualité de représentation des indiv
print("Représentation des individus dans le premier plan factoriel")
```

```
## [1] "Représentation des individus dans le premier plan factoriel"
```

```
fviz_pca_ind(acp ,col.ind = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE)
```

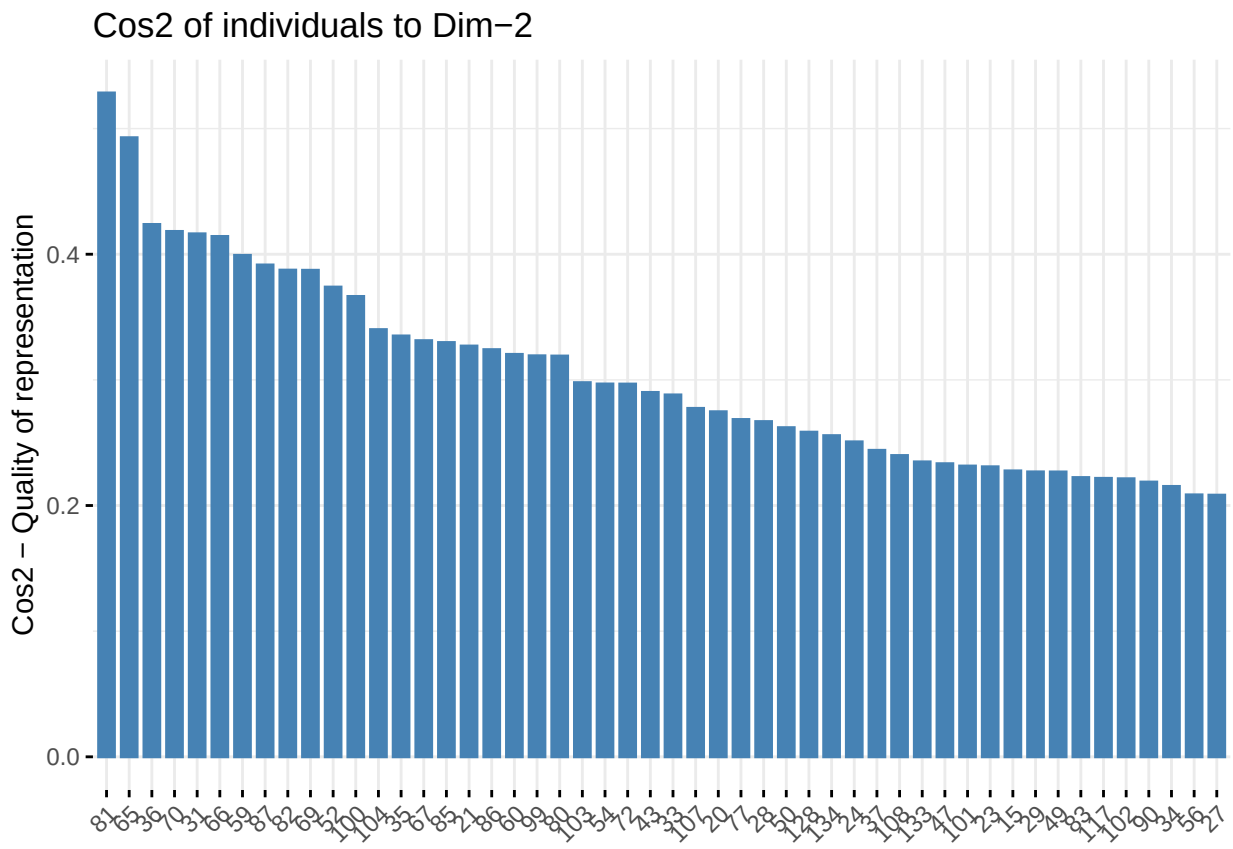
```
## Warning: ggrepel: 131 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



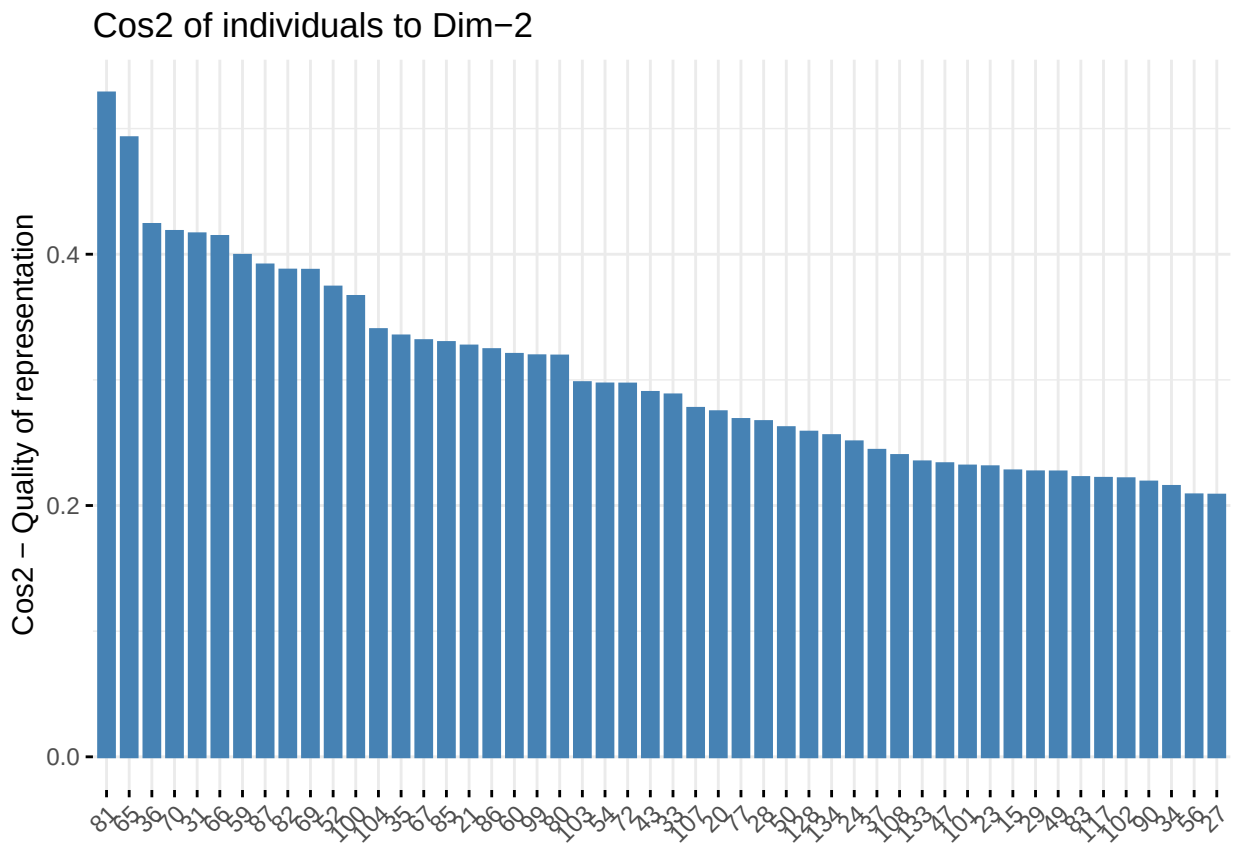

```
cat("\n\n")
```

On montre ci-dessous les diagrammes de cos2 pour les individus.

```
fviz_cos2(acp, choice = "ind",axes = 2,top=50) ##1er axe visualisation de 50 parmi 139
```



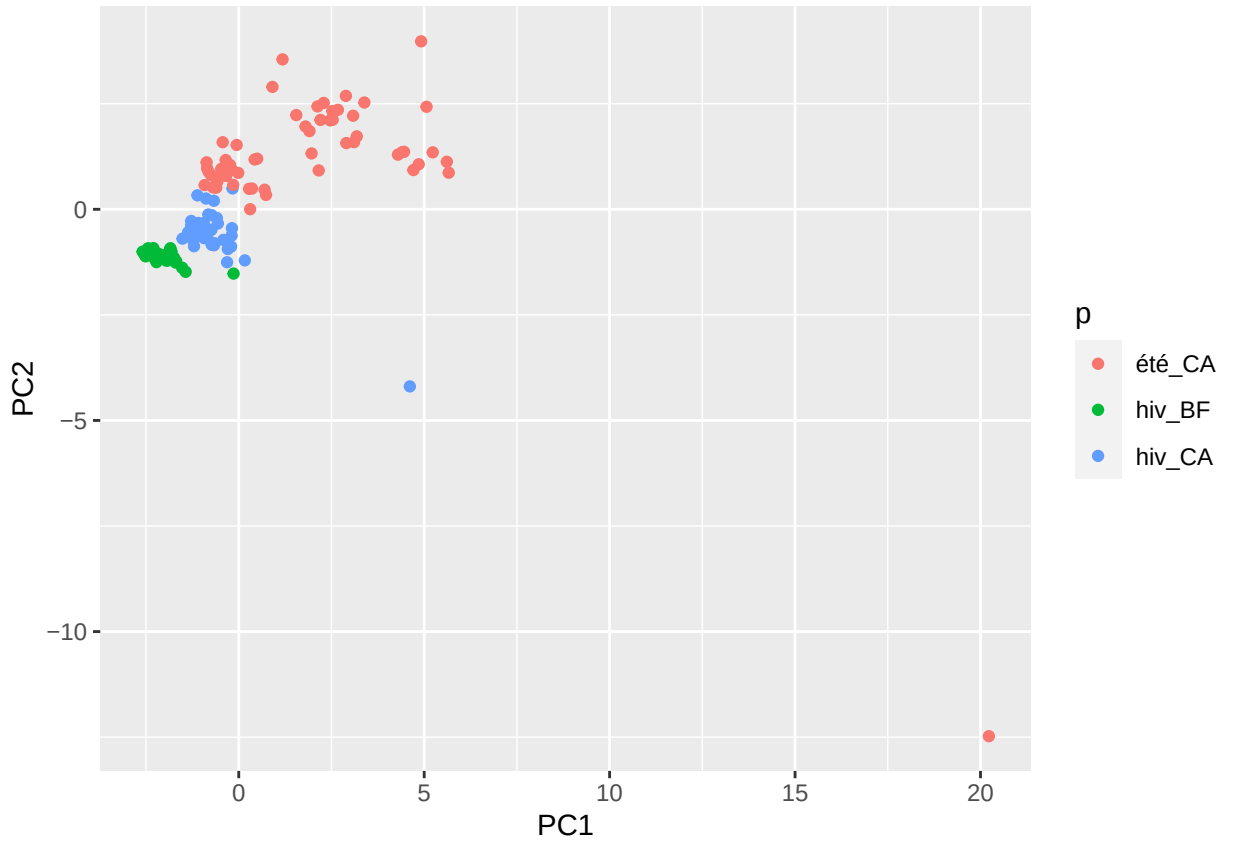

```
fviz_cos2(acp, choice = "ind",axes = 2,top=50)##2ème axe
```




```
cat("\n\n")
```

Dans cette partie, nous avons essayé de regrouper les individus par 4 classes : en période d'été avant 'été_BF' et après 'été_CA' la mise en activités du site, et en période d'hiver avant'hiv_BF' et après'hiv_CA' la mise en activités du site. Dans notre data il y avait apparemment pas de données pour les indiv "été_BF". Nous visualisons ces classes dans les deux composantes principal.

```
library(readxl)
brut<-read_excel("TP4_covC1234_DS19_20.xlsx")
#plot de été (avant/après), hiver(avant/après)
p<- ifelse(brut$SAISON=="hiver"& brut$Campagne%in%c("BF2","BF3"),"hiv_BF",ifelse(brut$SAISON=="hiver"& 
crd<- data.frame(PC1= acp$x[,1],PC2=acp$x[,2],period=p)
ggplot(data= crd,mapping=aes(x=PC1, y=PC2, colour=p))+geom_point()
```



- individus hiver_BF : nous avons vu dans la partie qualité des individus, ques ces individus ont un cos2 élevé ainsi qu'ils sont corrélés négativement avec les deux axes. Nous pouvons dire alors qu'en hiver et avant la mise en activité des site, on remarque pas la présence de composé chimiques.
- individus été_CA : On trouve que parmi ces individus, il y en a ceux qui sont proches de l'origine du plan factoriel (0,0) qui ont un cos2 faible et qui sont donc mal représentés dans ces deux axes, D'où l'absence des composés chimique.Mais pour les autres individus de cette classe, nous savons qu'ils sont lié aux composés chimiques B,13_ane,14_ane,Tot_OcNoDecana, et donc nous avons remarqué la présence de ces composés chimiques.
- individus hiver_CA : de la même manière nous observons qu'il y a des individus qui ont cos2 élevé et d'autres non, et donc il y en a ceux avec qui on remarque la présence des composé chimiques et d'autres non.

Nous ne pouvons pas conclure pour les deux classes été_CA et hiver_CA Nous aurons besoin peut être d'autres variables pour bien interpréter ces individus.