

apprentissage d'ACM (STHDA)

Je note très bien que la totalité de ce TP se trouve sur le lien suivant : <http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/75-acm-analyse-des-correspondances-multiples-avec-r-l-essen>

L'objectif c'est de faire apprendre l'ACF en utilisant R notebook. Rq: J'ai changé quelques codes avec des commentaires de plus. ##Introduction : L'Analyse des Correspondances Multiples (ACM ou MCA pour multiple correspondence analysis) est une extension de l'analyse factorielle des correspondances pour résumer et visualiser un tableau de données contenant plus de deux variables catégorielles. On peut aussi la considérer comme une généralisation de l'analyse en composantes principales lorsque les variables à analyser sont catégorielles plutôt que quantitatives (Abdi and Williams 2010).

L'ACM est généralement utilisée pour analyser des données d'enquête ou de sondage.

L'objectif est d'identifier:

Un groupe de personnes ayant un profil similaire dans leurs réponses aux questions Les associations entre les catégories des variables. [source STHDA]

Calcul :

Pour le calcul nous allons utiliser la fonction MCA() incluse dans le package FactoMiner, et pour les visualisation nous allons opter pour factoextra. ##Charger les libraries :

```
library(FactoMiner)
```

```
## Warning: package 'FactoMiner' was built under R version 3.4.2
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.4.2
```

```
## Loading required package: ggplot2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

Format des données :

Nous allons utiliser les données poison dans le package FactoMiner

```
data(poison)
```

```
str(poison)
```

```
## 'data.frame':   55 obs. of  15 variables:
## $ Age       : int   9 5 6 9 7 72 5 10 5 11 ...
## $ Time       : int   22 0 16 0 14 9 16 8 20 12 ...
## $ Sick       : Factor w/ 2 levels "Sick_n","Sick_y": 2 1 2 1 2 2 2 2 2 2 ...
## $ Sex        : Factor w/ 2 levels "F","M": 1 1 1 1 2 2 1 1 2 2 ...
## $ Nausea     : Factor w/ 2 levels "Nausea_n","Nausea_y": 2 1 1 1 1 1 1 2 2 1 ...
## $ Vomiting   : Factor w/ 2 levels "Vomit_n","Vomit_y": 1 1 2 1 2 1 2 2 1 2 ...
## $ Abdominals: Factor w/ 2 levels "Abdo_n","Abdo_y": 2 1 2 1 2 2 2 2 2 1 ...
## $ Fever      : Factor w/ 2 levels "Fever_n","Fever_y": 2 1 2 1 2 2 2 2 2 2 ...
## $ Diarrhae   : Factor w/ 2 levels "Diarrhea_n","Diarrhea_y": 2 1 2 1 2 2 2 2 2 2 ...
## $ Potato     : Factor w/ 2 levels "Potato_n","Potato_y": 2 2 2 2 2 2 2 2 2 2 ...
## $ Fish       : Factor w/ 2 levels "Fish_n","Fish_y": 2 2 2 2 2 1 2 2 2 2 ...
## $ Mayo       : Factor w/ 2 levels "Mayo_n","Mayo_y": 2 2 2 1 2 2 2 2 2 2 ...
## $ Courgette  : Factor w/ 2 levels "Courg_n","Courg_y": 2 2 2 2 2 2 2 2 2 2 ...
```

```
## $ Cheese      : Factor w/ 2 levels "Cheese_n","Cheese_y": 2 1 2 2 2 2 2 2 2 ...
## $ Icecream    : Factor w/ 2 levels "Icecream_n","Icecream_y": 2 2 2 2 2 2 2 2 2 ...
```

Ces données proviennent d'une enquête menée auprès d'enfants de l'école primaire qui ont subi des intoxications alimentaires. Ils ont été interrogés sur leurs symptômes et sur ce qu'ils ont mangé.

Les données contiennent 55 lignes (individus) et 15 colonnes (variables). Nous n'utiliserons que certains des individus (enfants) et variables pour effectuer l'ACM. Les coordonnées des individus et des variables restantes seront prédites.

Nos données contiennent donc des:

Individus actifs (lignes 1:55): individus qui sont utilisés dans l'ACM. Variables actives (colonnes 5:15): variables utilisées dans l'ACM. Variables supplémentaires: elles ne participent pas à l'ACM. Les coordonnées de ces variables seront prédites. Variables quantitatives supplémentaires (quanti.sup): Colonnes 1 et 2 correspondant aux colonnes age et time, respectivement. Variables qualitatives supplémentaires (quali.sup): Colonnes 3 et 4 correspondant aux colonnes Sick et Sex, respectivement. Ces variables seront utilisées pour colorer les individus par groupes. Nous commençons par extraire les individus actifs et les variables actives pour l'ACM:

```
poison.active <- poison[1:55, 5:15]
```

Résumé des données :

La fonction `summary()` peut être utilisée pour calculer la fréquence des catégories des variables.

Résumés statistiques:

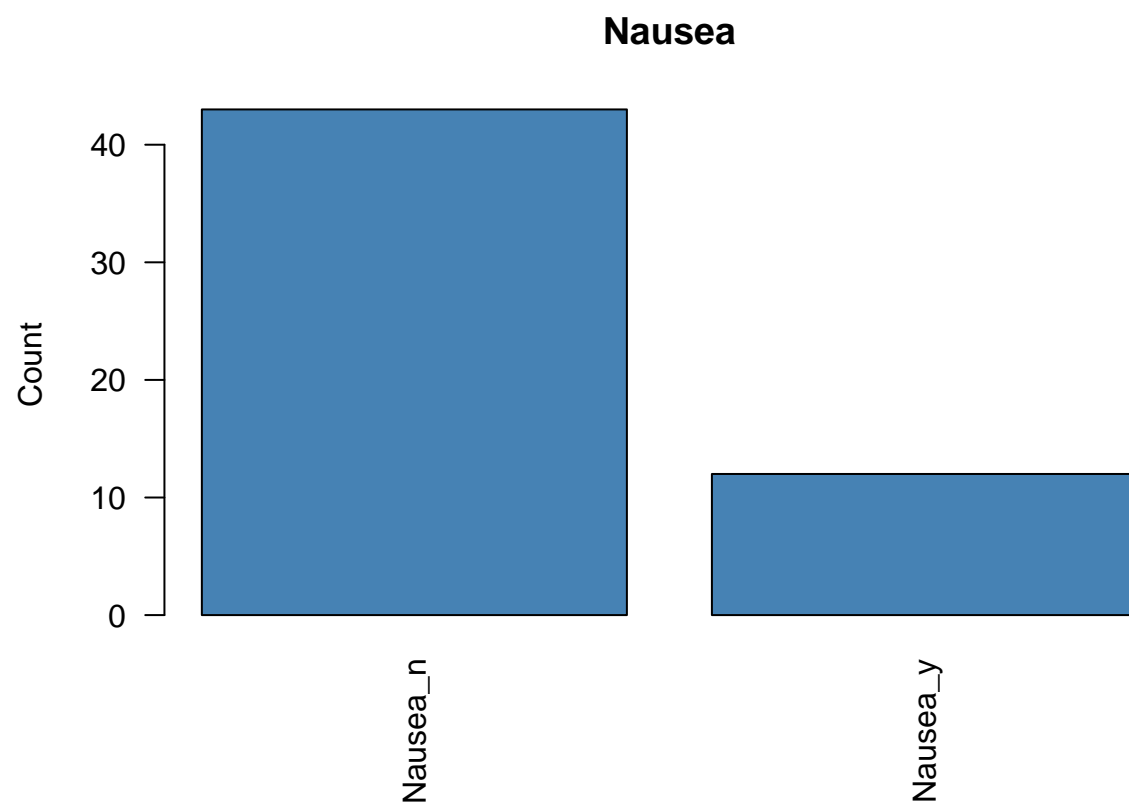
```
summary(poison.active)
```

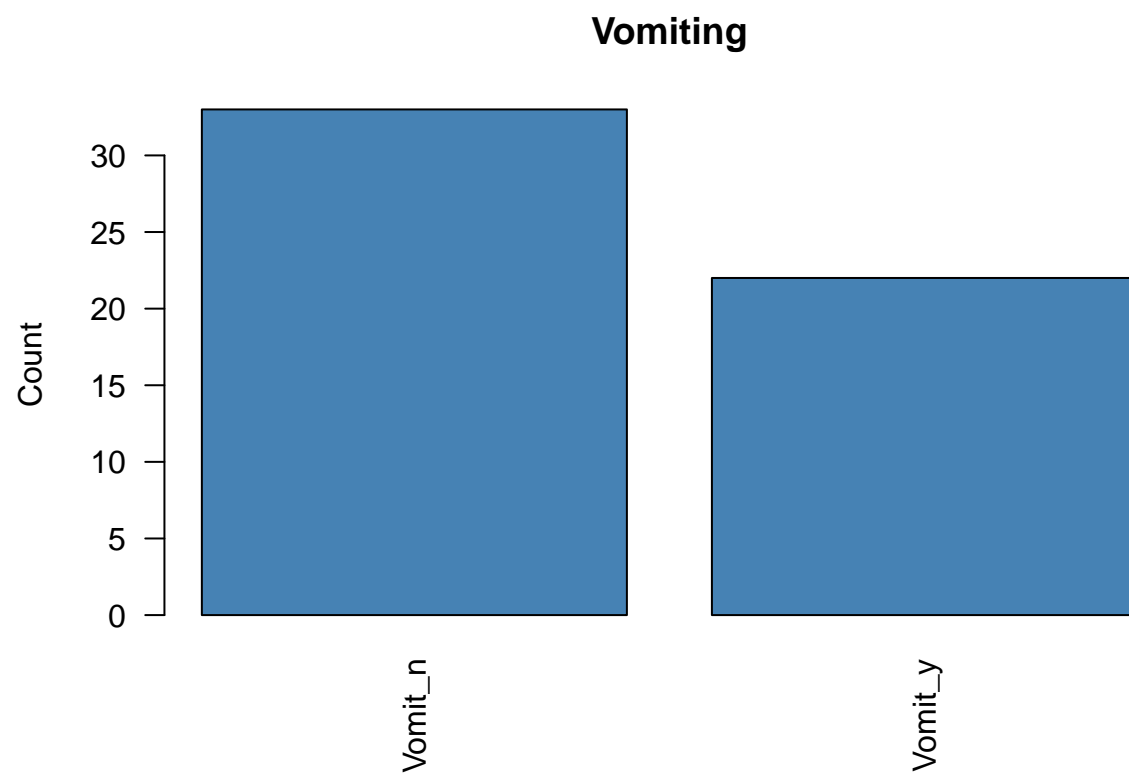
```
##      Nausea      Vomiting      Abdominals      Fever      Diarrhae
## Nausea_n:43 Vomit_n:33  Abdo_n:18  Fever_n:20  Diarrhea_n:20
## Nausea_y:12 Vomit_y:22  Abdo_y:37  Fever_y:35  Diarrhea_y:35
##      Potato      Fish      Mayo      Courgette      Cheese
## Potato_n: 3  Fish_n: 1  Mayo_n:10  Courg_n: 5  Cheese_n: 7
## Potato_y:52  Fish_y:54  Mayo_y:45  Courg_y:50  Cheese_y:48
##      Icecream
## Icecream_n: 4
## Icecream_y:51
```

La fonction `summary()` renvoie la taille des catégories des variables.

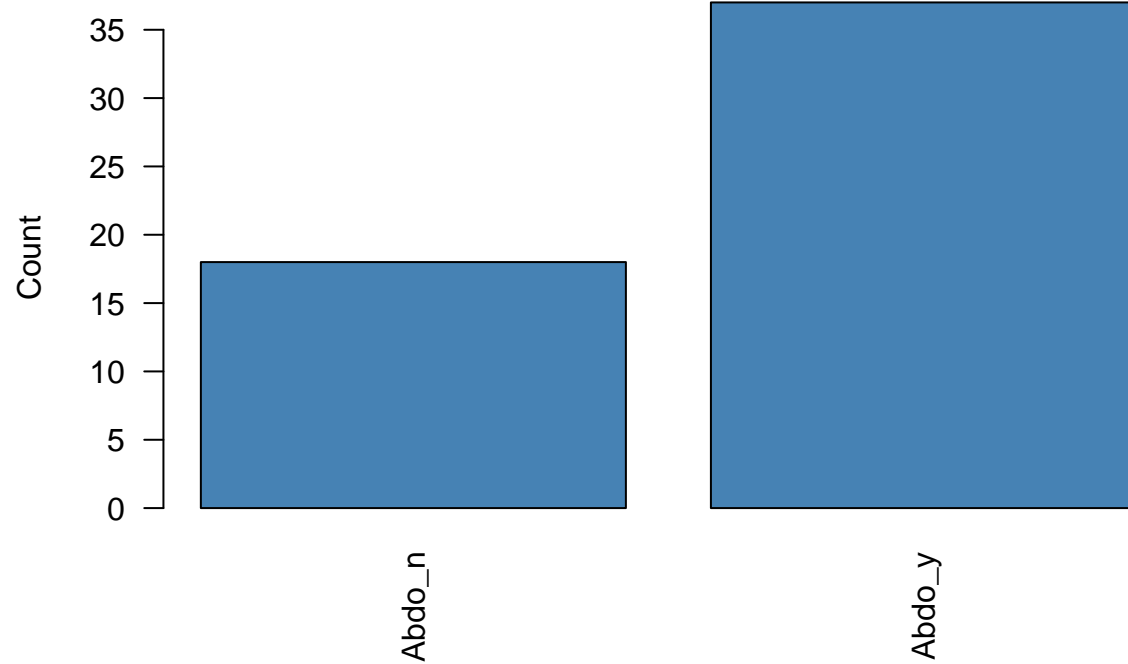
Il est également possible de visualiser la fréquence des catégories des variables. Le code R ci-dessous, montre les 4 premières colonnes:

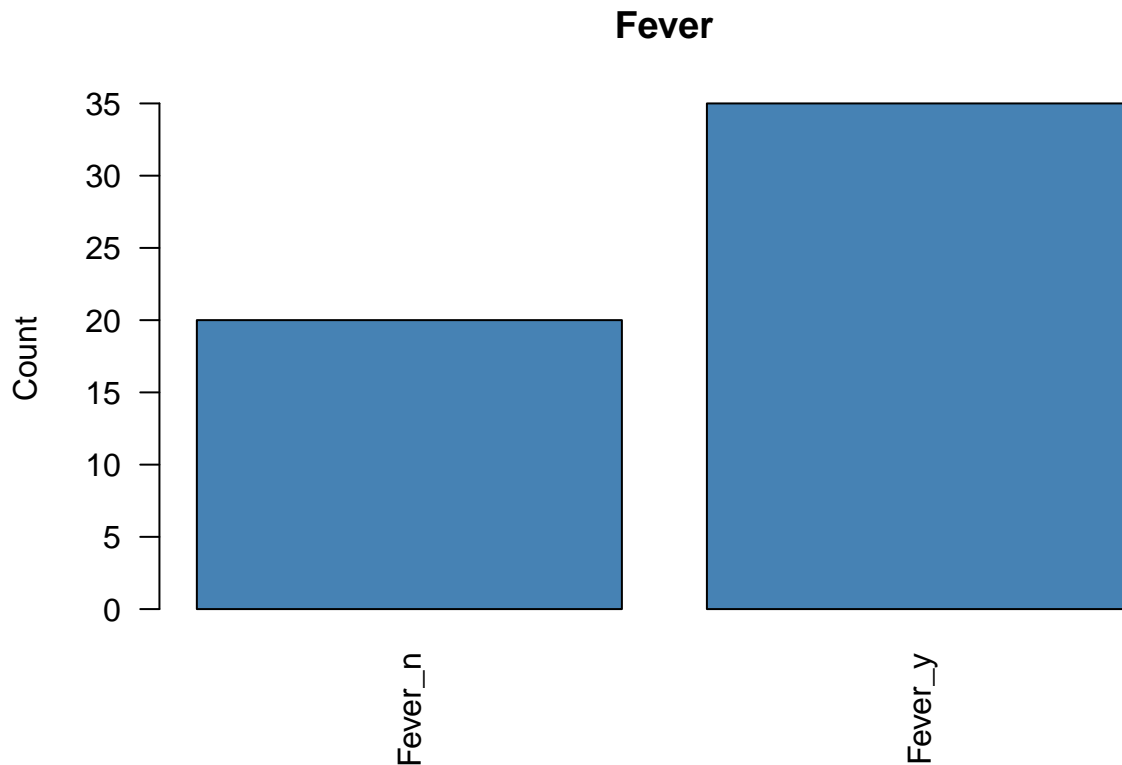
```
for (i in 1:4) {
  plot(poison.active[,i], main = colnames(poison.active)[i],
       ylab = "Count", col="steelblue", las = 2)
}
```





Abdominals





Les graphiques ci-dessus peuvent être utilisés pour identifier les catégories à très faible fréquence. Ce type de variables peut fausser l'analyse et doit être supprimé.

code R pour l'ACM :

Dans le code R ci-dessous, l'ACM est effectuée uniquement sur les individus / variables actifs:

```
res.mca <- MCA (poison.active, graph = FALSE)
```

Le résultat de l'ACM est une liste comprenant:

```
print(res.mca)
```

```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 55 individuals, described by 11 variables
## *The results are available in the following objects:
##
##   name                description
## 1 "$eig"              "eigenvalues"
## 2 "$var"              "results for the variables"
## 3 "$var$coord"        "coord. of the categories"
## 4 "$var$cos2"         "cos2 for the categories"
## 5 "$var$contrib"      "contributions of the categories"
## 6 "$var$v.test"       "v-test for the categories"
## 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 "$call"             "intermediate results"
## 12 "$call$marge.col"   "weights of columns"
## 13 "$call$marge.li"    "weights of rows"
```

L'objet créé avec la fonction `MCA()` contient de nombreuses informations trouvées dans de nombreuses listes et matrices différentes. Ces valeurs sont décrites dans la section suivante.

Visualisation et interprétation :

La proportion des variances retenues par les différentes dimensions (axes) peut être extraite à l'aide de la fonction `get_eigenvalue()` [factoextra package] comme suit:

```
eig.val <- get_eigenvalue(res.mca)
eig.val
```

##	eigenvalue	variance.percent	cumulative.variance.percent
## Dim.1	0.33523140	33.523140	33.52314
## Dim.2	0.12913979	12.913979	46.43712
## Dim.3	0.10734849	10.734849	57.17197
## Dim.4	0.09587950	9.587950	66.75992
## Dim.5	0.07883277	7.883277	74.64319
## Dim.6	0.07108981	7.108981	81.75217
## Dim.7	0.06016580	6.016580	87.76876
## Dim.8	0.05577301	5.577301	93.34606
## Dim.9	0.04120578	4.120578	97.46663
## Dim.10	0.01304158	1.304158	98.77079
## Dim.11	0.01229208	1.229208	100.00000

Les dimensions sont ordonnées de manière décroissante et listées en fonction de la quantité de variance expliquée. La dimension 1 explique la plus grande variance, suivie de la dimension 2 et ainsi de suite.

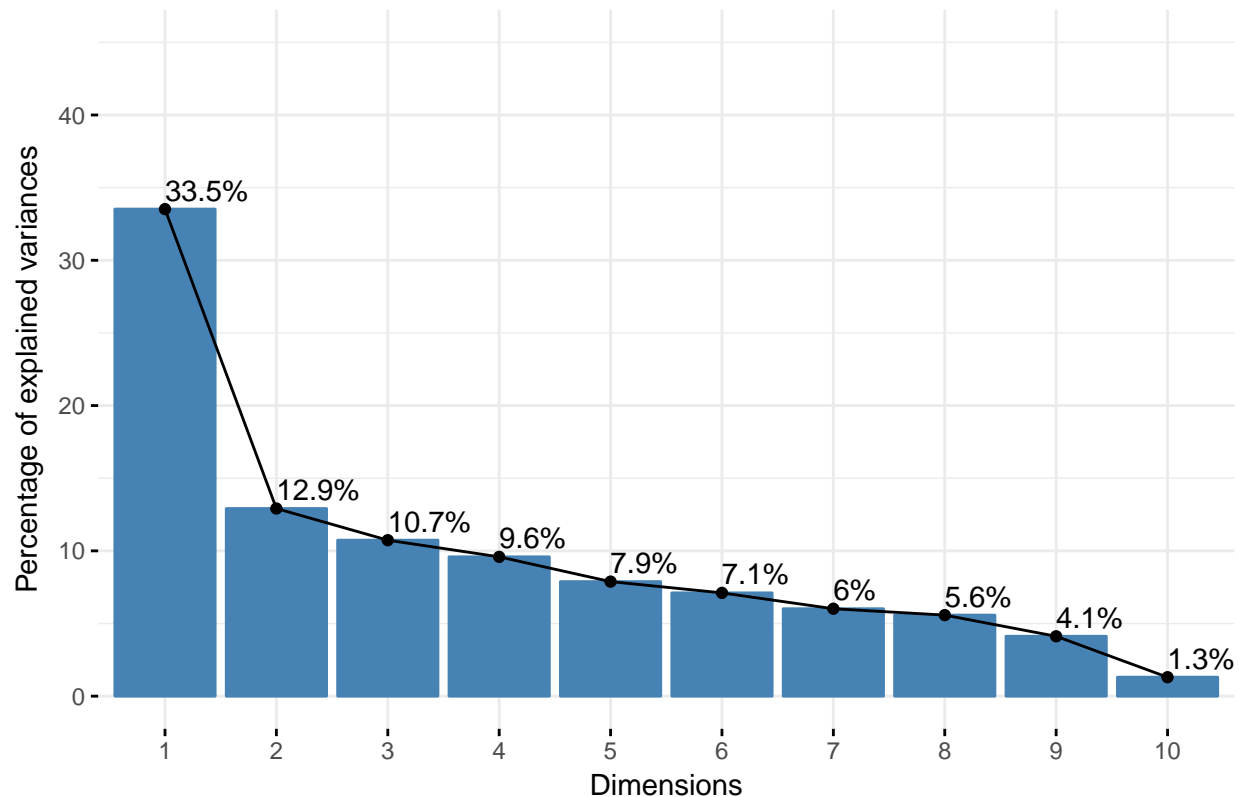
Le pourcentage cumulé expliqué est obtenu en ajoutant les proportions successives de variances expliquées pour obtenir le total courant. Par exemple, 33.52% plus 12.91% est égal à 46.43%, et ainsi de suite. Par conséquent, environ 88.6% de la variance totale est expliquée par les deux premières dimensions.

Les valeurs propres peuvent être utilisées pour déterminer le nombre d'axes à retenir. Il n'y a pas de «règle générale» pour choisir le nombre de dimensions à conserver pour l'interprétation des données. Cela dépend de la question et du besoin du chercheur. Par exemple, si vous êtes satisfait avec 70% des variances totales expliquées, utilisez le nombre de dimensions nécessaires pour y parvenir. Note : Notez qu'une analyse est bonne lorsque les premières dimensions représentent une grande partie de la variabilité. Dans notre analyse, les cinq premiers axes expliquent 74.64% de la variance totale. C'est un pourcentage acceptable.

Une autre méthode pour déterminer le nombre de dimensions est de regarder le graphique des valeurs propres (scree plot), ordonnées de la plus grande à la plus petite valeur. Le nombre d'axes est déterminé par le point point, au-delà duquel les valeurs propres restantes sont toutes relativement petites et de tailles comparables. Pour visualiser les pourcentages de variances expliquées par chaque dimension de l'ACM, utilisez la fonction `fviz_eig()` ou `fviz_screplot()` [package factoextra]:

```
fviz_screplot (res.mca, addlabels = TRUE, ylim = c (0, 45))
```

Scree plot



Il est également possible de calculer une valeur propre moyenne au-dessus de laquelle l'axe doit être conservé dans le résultat. Nos données contiennent 55 lignes et 15 colonnes.

Si les données étaient aléatoires, la valeur attendue de la valeur propre pour chaque axe serait $1 / (\text{nrow}(\text{poison.active}) - 1) = 1/54 = 1.85\%$ en termes de lignes.

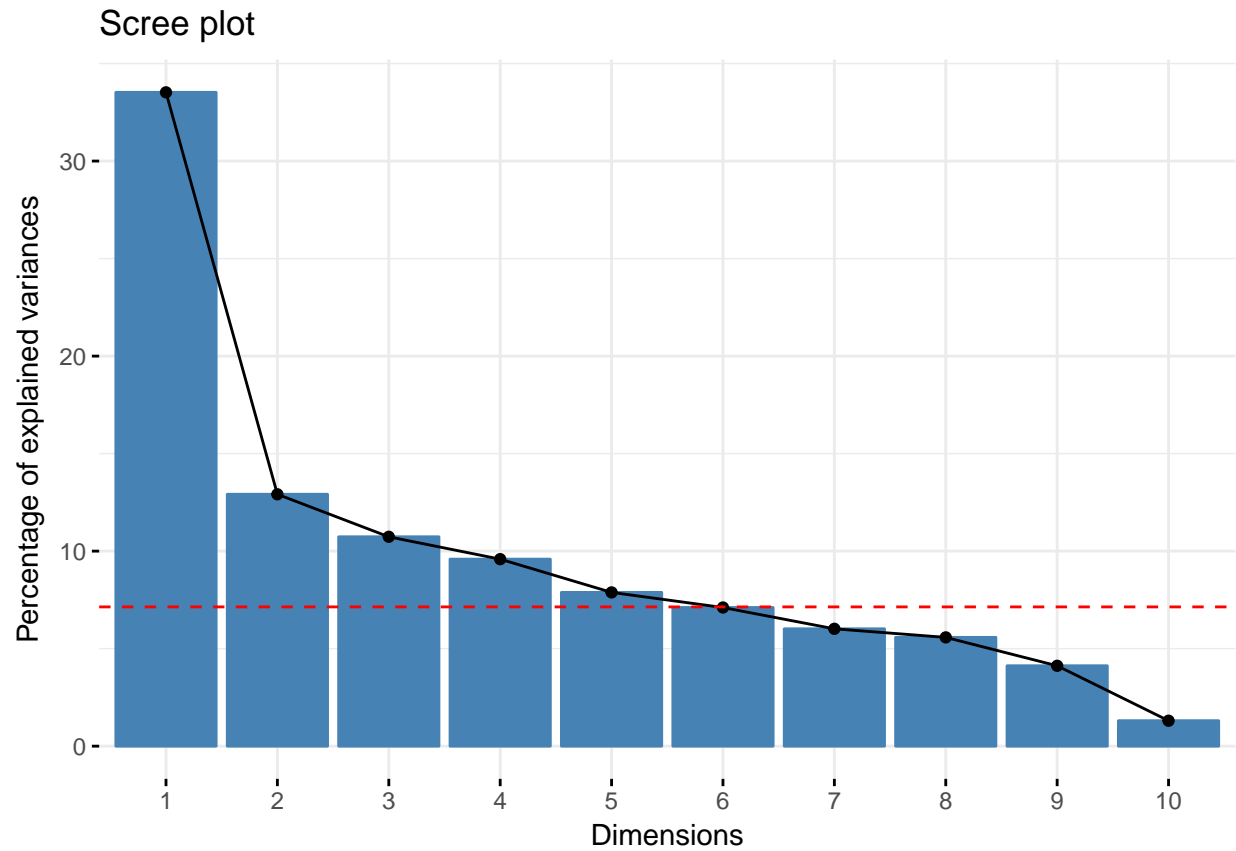
De même, l'axe moyen devrait représenter $1 / (\text{ncol}(\text{poison.active}) - 1) = 1/14 = 7.14\%$ en termes de 15 colonnes.

Tout axe avec une contribution supérieure au maximum de ces deux pourcentages devrait être considéré comme important et inclus dans la solution pour l'interprétation des données.

Donc on doit retenir selon M. T. Bendixen 6 axes principales. Ces 6 axes ont pour inertie totale : 81.75% (une valeur acceptable).

Le code R ci-dessous, montre le graphique des valeurs propres avec une droite en pointillée rouge spécifiant la valeur propre moyenne:

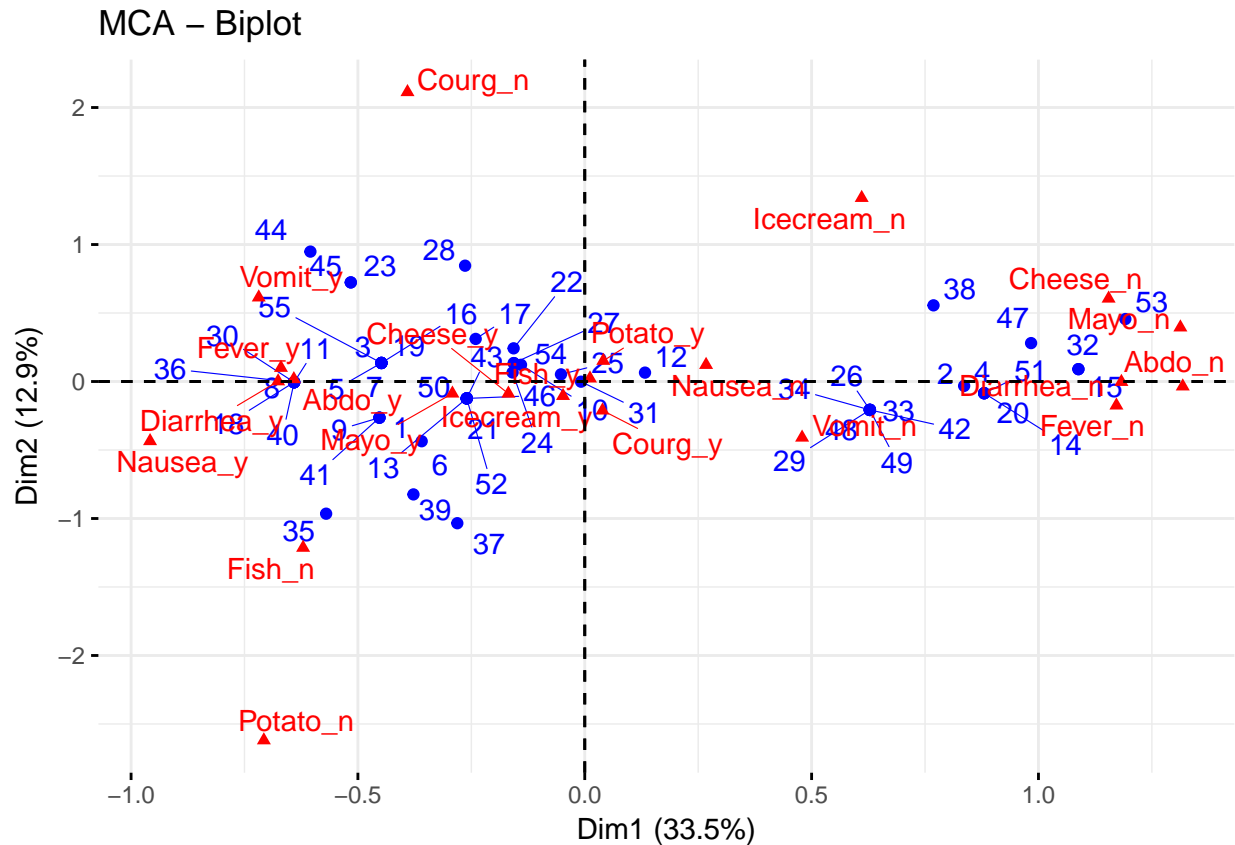
```
fviz_screplot (res.mca) +
  geom_hline (yintercept = 7.14, linetype = 2, color = "red")
```

Biplot :

La fonction `fviz_mca_biplot()` [factoextra] permet de visualiser le biplot des individus et des variables:

```
fviz_mca_biplot (res.mca, repel = TRUE,  
                 ggtheme = theme_minimal())
```



Dans le graphique ci-dessus, les lignes sont représentées par des points bleus et des colonnes par des triangles rouges.

La distance entre les points lignes ou entre les points colonnes donne une mesure de leur similitude (ou dissemblance). Les points lignes avec un profil similaire sont proches sur le graphique. Il en va de même pour les points colonnes. Le graphique montre que : - Les lignes 34,42,33, 26, 48, 49 et 29 sont associés le plus à la colonne Vomit_n.4 - etc...

Graphique des variables :

Résultats :

La fonction `get_mca_var()` [factoextra] sert à extraire les résultats pour les catégories des variables. Cette fonction renvoie une liste contenant les coordonnées, les cos2 et les contributions des catégories:

```
var <- get_mca_var(res.mca)
var
```

```
## Multiple Correspondence Analysis Results for variables
## =====
##   Name      Description
## 1 "$coord"   "Coordinates for categories"
## 2 "$cos2"    "Cos2 for categories"
## 3 "$contrib" "contributions of categories"
```

Les composants de `get_mca_var()` peuvent être utilisés dans le graphique des variables comme suit:

varcoord : coordonnées des variables pour créer un nuage de points varcos2: qualité de représentation des variables. var\$contrib: contributions (en pourcentage) des variables à la définition des dimensions.

Les différents composants peuvent être consultés comme suit:

```
# Coordonnées
print('les coordonnées :')
```

```
## [1] "les coordonnées :"
```

```
var$coord
```

```
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Nausea_n    0.26739087  0.121390290 -0.265583253  0.03376130  0.073704999
## Nausea_y   -0.95815062 -0.434981874  0.951673323 -0.12097801 -0.264109581
## Vomit_n     0.47902794 -0.409194649  0.084492799  0.27361142  0.052452504
## Vomit_y    -0.71854191  0.613791974 -0.126739198 -0.41041713 -0.078678757
## Abdo_n      1.31802207 -0.035745005 -0.005094243 -0.15360951 -0.069869870
## Abdo_y     -0.64119993  0.017389462  0.002478280  0.07472895  0.033990747
## Fever_n     1.17183098 -0.174895110  0.097275290 -0.18967098 -0.018478570
## Fever_y    -0.66961770  0.099940063 -0.055585880  0.10838342  0.010559183
## Diarrhea_n  1.18282245 -0.002756840 -0.082979839 -0.24123007 -0.104910816
## Diarrhea_y -0.67589854  0.001575337  0.047417051  0.13784575  0.059949038
## Potato_n   -0.70730851 -2.619110755  2.138637690  0.37461658  1.355830579
## Potato_y    0.04080626  0.151102544 -0.123382944 -0.02161249 -0.078220995
## Fish_n     -0.62079031 -1.213389338 -3.691000244  5.56976822  0.154864481
## Fish_y      0.01149612  0.022470173  0.068351856 -0.10314386 -0.002867861
## Mayo_n      1.31277835  0.394914774  0.417481883  0.37330832 -0.316756981
## Mayo_y     -0.29172852 -0.087758839 -0.092773752 -0.08295741  0.070390440
## Courg_n    -0.39073240  2.112852279  0.728297960  0.32368182  1.125993240
## Courg_y     0.03907324 -0.211285228 -0.072829796 -0.03236818 -0.112599324
## Cheese_n    1.15462219  0.605560528  0.283133231  0.36031062  1.681357074
## Cheese_y   -0.16838240 -0.088310910 -0.041290263 -0.05254530 -0.245197907
## Icecream_n  0.61038541  1.340331612  1.917576423  1.71261979 -1.458589548
## Icecream_y -0.04787337 -0.105124048 -0.150398151 -0.13432312  0.114399180
```

```
# Cos2: qualité de représentation
print('la qualité de représentation :')
```

```
## [1] "la qualité de représentation :"
```

```
var$cos2
```

```
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Nausea_n    0.256200730 5.280258e-02 2.527485e-01 0.004084375 0.0194661965
## Nausea_y    0.256200730 5.280258e-02 2.527485e-01 0.004084375 0.0194661965
## Vomit_n     0.344201648 2.511604e-01 1.070855e-02 0.112294813 0.0041268978
## Vomit_y     0.344201648 2.511604e-01 1.070855e-02 0.112294813 0.0041268978
## Abdo_n      0.845115652 6.215864e-04 1.262496e-05 0.011479077 0.0023749291
## Abdo_y      0.845115652 6.215864e-04 1.262496e-05 0.011479077 0.0023749291
## Fever_n     0.784678768 1.747903e-02 5.407133e-03 0.020557189 0.0001951186
## Fever_y     0.784678768 1.747903e-02 5.407133e-03 0.020557189 0.0001951186
## Diarrhea_n  0.799467973 4.342953e-06 3.934659e-03 0.033252541 0.0062893025
## Diarrhea_y  0.799467973 4.342953e-06 3.934659e-03 0.033252541 0.0062893025
## Potato_n    0.028862615 3.957543e-01 2.638714e-01 0.008096399 0.1060544169
## Potato_y    0.028862615 3.957543e-01 2.638714e-01 0.008096399 0.1060544169
## Fish_n      0.007136678 2.726507e-02 2.522867e-01 0.574487370 0.0004441298
## Fish_y      0.007136678 2.726507e-02 2.522867e-01 0.574487370 0.0004441298
```

```
## Mayo_n      0.382974888 3.465726e-02 3.873136e-02 0.030968690 0.0222966634
## Mayo_y      0.382974888 3.465726e-02 3.873136e-02 0.030968690 0.0222966634
## Courg_n     0.015267181 4.464145e-01 5.304179e-02 0.010476992 0.1267860776
## Courg_y     0.015267181 4.464145e-01 5.304179e-02 0.010476992 0.1267860776
## Cheese_n    0.194418058 5.347760e-02 1.169065e-02 0.018932629 0.4122652350
## Cheese_y    0.194418058 5.347760e-02 1.169065e-02 0.018932629 0.4122652350
## Icecream_n  0.029221204 1.409011e-01 2.883999e-01 0.230044435 0.1668614487
## Icecream_y  0.029221204 1.409011e-01 2.883999e-01 0.230044435 0.1668614487
```

```
# Contributions aux axes
```

```
print('la contribution des variables :')
```

```
## [1] "la contribution des variables :"
```

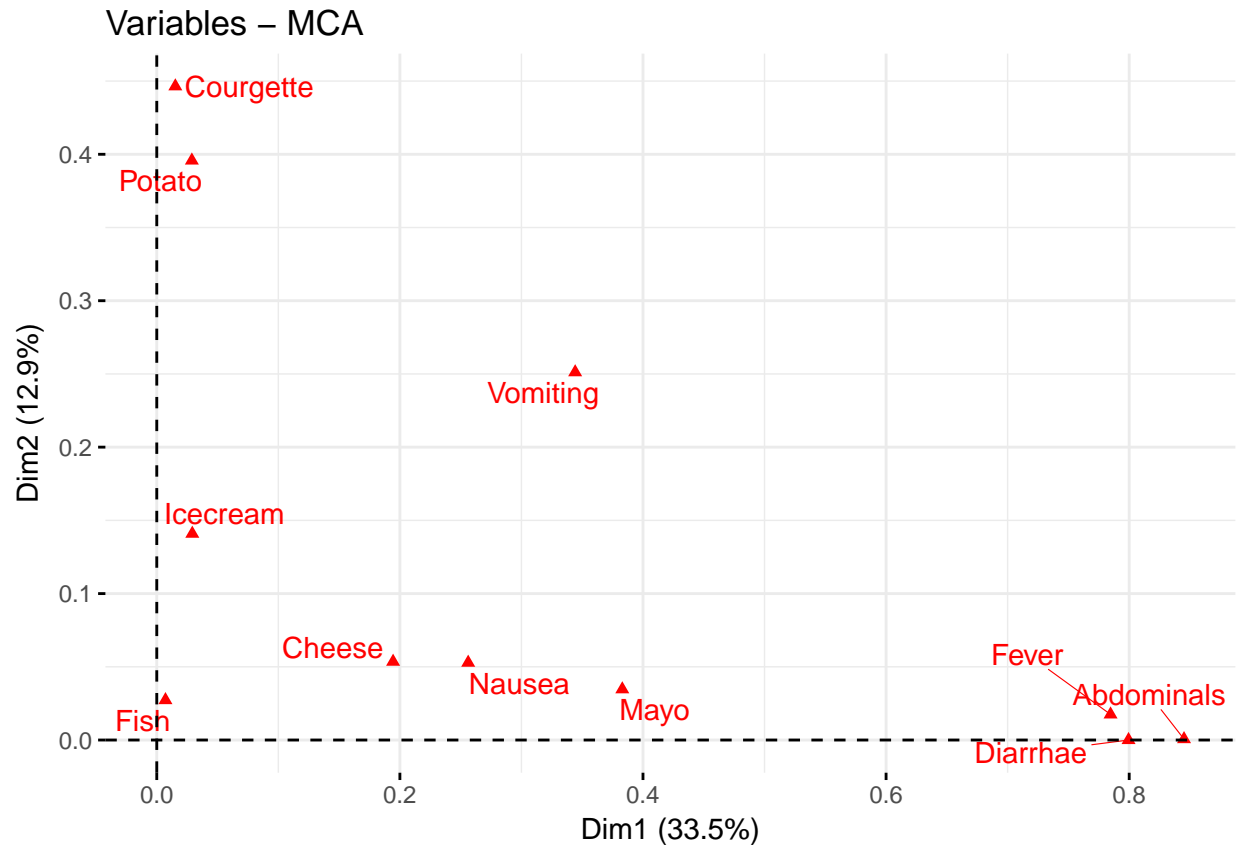
```
var$contrib
```

```
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Nausea_n    1.515868554 8.110001e-01 4.670018e+00 0.08449397 4.897791e-01
## Nausea_y    5.431862319 2.906084e+00 1.673423e+01 0.30277007 1.755042e+00
## Vomit_n     3.733666829 7.072263e+00 3.627455e-01 4.25893721 1.903638e-01
## Vomit_y     5.600500244 1.060839e+01 5.441183e-01 6.38840581 2.855456e-01
## Abdo_n      15.417636578 2.943661e-02 7.192511e-04 0.73219636 1.842427e-01
## Abdo_y       7.500471849 1.432051e-02 3.499060e-04 0.35620363 8.963157e-02
## Fever_n     13.541285078 7.830146e-01 2.913961e-01 1.24036823 1.431873e-02
## Fever_y      7.737877188 4.474369e-01 1.665121e-01 0.70878185 8.182133e-03
## Diarrhea_n  13.796503952 1.945529e-04 2.120430e-01 2.00637332 4.615390e-01
## Diarrhea_y   7.883716544 1.111731e-04 1.211674e-01 1.14649904 2.637366e-01
## Potato_n    0.740012332 2.633986e+01 2.112732e+01 0.72579521 1.156299e+01
## Potato_y     0.042693019 1.519608e+00 1.218884e+00 0.04187280 6.670953e-01
## Fish_n      0.190015838 1.884451e+00 2.097668e+01 53.48021026 5.028535e-02
## Fish_y      0.003518812 3.489724e-02 3.884571e-01 0.99037426 9.312103e-04
## Mayo_n      8.497335124 1.996141e+00 2.683638e+00 2.40244916 2.103731e+00
## Mayo_y      1.888296694 4.435870e-01 5.963641e-01 0.53387759 4.674958e-01
## Courg_n     0.376381955 2.856885e+01 4.083541e+00 0.90307845 1.329167e+01
## Courg_y     0.037638196 2.856885e+00 4.083541e-01 0.09030784 1.329167e+00
## Cheese_n    4.601270591 3.285471e+00 8.640292e-01 1.56664651 4.149118e+01
## Cheese_y    0.671018628 4.791313e-01 1.260043e-01 0.22846928 6.050798e+00
## Icecream_n  0.734798428 9.197484e+00 2.264718e+01 20.22556991 1.784284e+01
## Icecream_y  0.057631249 7.213713e-01 1.776249e+00 1.58631921 1.399439e+00
```

Rq: La dimension1 peut alors être écrite sous la forme : $\text{Dim.1} = \text{Nausea_n} \cdot 0.26739087 - \text{Nausea_y} \cdot 0.95815062 + \dots - \text{Icecream_y} \cdot 0.04787337$

Dans cette section, nous décrirons comment visualiser uniquement les catégories des variables. Ensuite, nous mettrons en évidence les catégories en fonction soit de leurs qualités de représentation, soit de leurs contributions aux dimensions. **Corrélation entre les variables et les axes principaux** : Pour visualiser la corrélation entre les variables et les axes principaux de l'ACM, tapez ceci:

```
fviz_mca_var (res.mca, choice = "mca.cor",
              repel = TRUE,
              ggtheme = theme_minimal ())
```



+ Le graphique ci-dessus permet d'identifier les variables les plus corrélées avec chaque axe. Les corrélations au carré entre les variables et les axes sont utilisées comme coordonnées.

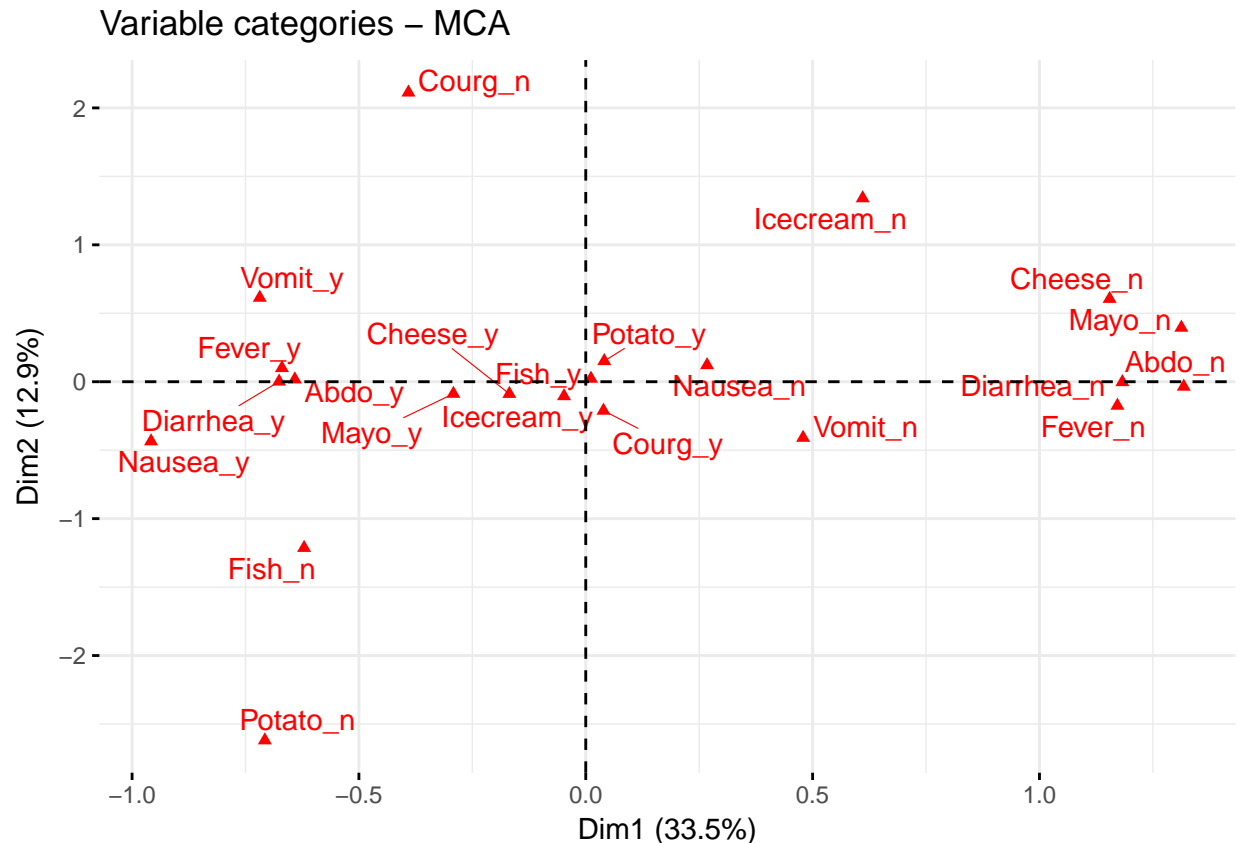
- On constate que les variables Diarrhae, Abdominals et Fever sont les plus corrélées avec la dimension 1. De même, les variables Courgette et Potato sont les plus corrélées avec la dimension 2. ### Coordonnées des catégories des variables : Le code R ci-dessous affiche les coordonnées de chacune des catégories des variables dans chaque dimension (1, 2 et 3):

```
head(round(var$coord, 2), 4)
```

```
##          Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## Nausea_n  0.27  0.12 -0.27  0.03  0.07
## Nausea_y -0.96 -0.43  0.95 -0.12 -0.26
## Vomit_n   0.48 -0.41  0.08  0.27  0.05
## Vomit_y  -0.72  0.61 -0.13 -0.41 -0.08
```

Utilisez la fonction `fviz_mca_var()` [factoextra] pour visualiser uniquement les catégories des variables:

```
fviz_mca_var(res.mca,
  repel = TRUE,
  ggtheme = theme_minimal())
```



Le graphique ci-dessus montre les relations entre les catégories des variables. Il peut être interprété comme suit:

- Les catégories avec un profil similaire sont regroupées (exemple : Cheese_n, Mayo_n, Abdo_n, Fever_n).
- Les catégories corrélées négativement sont positionnées sur les côtés opposés de l'origine du graphique (quadrants opposés)(Exemple : Diarrhea_y, Abdo_y,...)
- La distance entre les catégories et l'origine mesure la qualité des catégories. Les points qui sont loin de l'origine sont bien représentés par l'ACM(exemple : Cheese_n, Courg_n, Potato_n, Diarrhea_y,...).

Qualité de représentation des catégories des variables :

Les deux dimensions 1 et 2 capturent 46% de l'inertie totale (variation) contenue dans les données. Tous les points ne sont pas aussi bien représentés par les deux dimensions.

La qualité de représentation, appelée cosinus carré (\cos^2), mesure le degré d'association entre les catégories des variables et les dimensions. Le \cos^2 peut être extrait comme suit:

```
head(var$cos2, 4)
```

```
##          Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Nausea_n 0.2562007 0.05280258 0.25274850 0.004084375 0.019466197
## Nausea_y 0.2562007 0.05280258 0.25274850 0.004084375 0.019466197
## Vomit_n  0.3442016 0.25116039 0.01070855 0.112294813 0.004126898
## Vomit_y  0.3442016 0.25116039 0.01070855 0.112294813 0.004126898
```

Si une catégorie d'une variable donnée est bien représentée par deux dimensions, la somme des \cos^2 est proche de 1. Pour certains éléments, plus de 2 dimensions sont nécessaires pour représenter parfaitement les données.

Il est possible de colorer les variables en fonction de la valeur de leur \cos^2 à l'aide de l'argument `col.var =`

“cos2”. Cela produit un gradient de couleurs. Dans ce cas, l’argument `gradient.cols` peut être utilisé pour spécifier une palette de couleur personnalisée. Par exemple, `gradient.cols = c(“white”, “blue”, “red”)` signifie que:

- les variables à faible valeur de `cos2` seront colorées en “white” (blanc)
- les variables avec des valeurs moyennes de `cos2` seront colorées en “blue” (bleu)
- les variables avec des valeurs élevées de `cos2` seront colorées en “red” (rouge)

```
# Colorer en fonction du cos2
fviz_mca_var(res.mca, col.var = "cos2",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE,
  ggtheme = theme_minimal())
```

