Chapitre 4

Analyse Factorielle des Correspondances (AFC)

Nos allons procéder par traiter un exempleélémentaire et non réaliste, puis nous nous intéresserons à l'exemple traité en cours et enfin nousétudierons une situation pratique.

1. Lancer ensuite `R` et modifier le répertoire de travail en allant dans `Fichier -> Changer le Répertoire Courant` et en choisissant le répertoire `Bureau/TP_M1MIASHS_SSD_AD` qui aété créé.
2. Ouvrir une fenêtre d'éditeur `Fichier -> Nouveau Script`.
3. Sauver le fichier dans le répertoire courant sous le nom `TP3.R : Fichier -> Sauver sous`
4. Penser à sauvegarder régulièrement le contenu du fichier TP3.R en appuyant sur les touches `Ctrl et S`.

4.1 Pratique de l'AFC

L'analyse factorielle des correspondances (AFC ou CA pour correspondence analysis en anglais) est une extension de l'analyse en composantes principales (cf. chapitre Analyse en composantes principales) pour analyser l'association entre deux variables qualitatives (ou catégorielles). L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables catégorielles. Le tableau de contingence contient les fréquences formées par les deux variables.

L'AFC retourne les coordonnées des éléments des colonnes et des lignes du tableau de contingence. Ces coordonnées permettent de visualiser graphiquement l'association entre

les éléments de lignes et de colonnes dans un graphique à deux dimensions.

Lors de l'analyse d'un tableau de contingence, une question typique est de savoir si certains éléments lignes sont associés à certains éléments colonnes. L'analyse factorielle par correspondance est une approche géométrique pour visualiser les lignes et les colonnes d'une table de contingence dans un graphique en nuage de points, de sorte que les positions des points lignes et celles des points colonnes correspondent à leurs associations dans le tableau.

Dans le chapitre actuel, nous montrerons comment calculer et interpréter l'AFC en utilisant deux packages `R` :

— `FactoMineR` pour l'analyse
— `factoextra` pour la visualisation des données.

De plus, nous montrerons comment révéler les éléments les plus importants expliquant les variations dans le jeu de données. Nous continuons en expliquant comment faire une AFC avec une table de données contenant des lignes et des colonnes supplémentaires. Ceci est important, si vous voulez faire des prédictions avec l'AFC. Les dernières sections de ce guide décrivent également comment filtrer les résultats de l'AFC afin de ne conserver que les éléments les plus contributifs. Enfin, nous allons voir comment traiter les valeurs extrêmes.

4.1.1 Calcul

- *Packages R.* Plusieurs fonctions de différents packages sont disponibles dans le logiciel `R` pour calculer l'AFC :

  ```
  CA() [package FactoMineR],
  ca() [package ca],
  dudi.coa() [package ade4],
  corresp() [package MASS],
  ```

  et

  ```
  epCA() [package ExPosition]
  ```

  Peu importe la fonction que vous décidez d'utiliser, vous pouvez facilement extraire et visualiser les résultats de l'AFC en utilisant les fonctions `R` fournies dans le package `factoextra`.

  Ici, nous utiliserons les deux packages `FactoMineR` (pour l'analyse) et `factoextra` (pour la visualisation basée sur `ggplot2`).

  Installez les deux packages comme suit :

  ```
  install.packages(c("FactoMineR", "factoextra"))
  ```

  Chargez-les dans `R`, en tapant ceci :

```
library("FactoMineR")
library("factoextra")
```

- *Format des données.* Les données doivent être un tableau de contingence. Nous utiliserons le jeu de données `housetasks` disponible dans le package `factoextra` de `R`.

```
data(housetasks)
# head(housetasks)
```

Les données correspondent à un tableau de contingence contenant 13 taches ménagères et leur répartition dans le couple :
— les lignes sont les différentes tâches
— les valeurs sont les fréquences des tâches effectuées :
  — par la femme seulement (colonne "wife")
  — alternativement (colonne "alternatively")
  — par le mari seulement (colonne "husband")
  — ou conjointement (colonne "jointly")
Les données sont illustrées dans l'image suivante :

	Wife	Alternating	Husband	Jointly
Laundry	156	14	2	4
Main_meal	124	20	5	4
Dinner	77	11	7	13
Breakfeast	82	36	15	7
Tidying	53	11	1	57
Dishes	32	24	4	53
Shopping	33	23	9	55
Official	12	46	23	15
Driving	10	51	75	3
Finances	13	13	21	66
Insurance	8	1	53	77
Repairs	0	3	160	2
Holidays	0	1	6	153

FIGURE 4.1 – Tableau de contingence sur les tâches ménagères

- *Graphique du tableaux de contingence et test de* `chi2`. Le tableau de contingence ci-dessus n'est pas très gros. Par conséquent, il est facile d'inspecter et d'interpréter

visuellement les profils des lignes et des colonnes :
Il est évident que, les tâches ménagères - `Laundry`, `Main_Meal` et `Dinner` - sont plus fréquemment effectuées par l'épouse. Les tâches `Repairs` et `Driving` sont fréquemment associées à "husband" `Holidays` est fréquemment associé à la colonne "jointly" Le tableau de contingence peut être visualisé en utilisant les fonctions `balloonplot()` [package `gplots`] et `mosaicplot()` [package `garphics`] :

```
library("gplots")
# 1. convertir les données en tant que table
dt <- as.table(as.matrix (housetasks))
# 2. Graphique
balloonplot(t (dt), main = "housetasks", xlab = "", ylab = "", label =
FALSE, show.margins = FALSE)
```

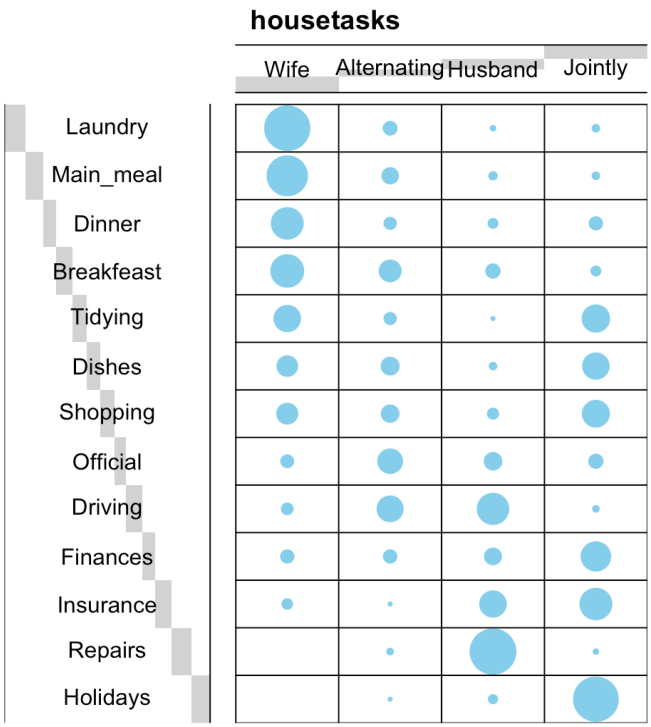

FIGURE 4.2 – Visualisation du tableau de contingence des taches ménagères

Notez que les totaux des lignes et des colonnes sont affichés par défaut dans les marges inférieure et de droite, respectivement. Ces valeurs sont cachées, dans le graphique ci-dessus, en utilisant l'argument `show.margins = FALSE`.
Pour un petit tableau de contingence, vous pouvez utiliser le test de `chi2` pour évaluer s'il existe une dépendance significative entre les catégories des lignes et des colonnes :

```
chisq <- chisq.test (housetasks)
chisq
#
# Pearson's Chi-squared test
#
# data : housetasks
# X-squared = 2000, df = 40, p-value <2e-16
```

Dans notre exemple, les variables de ligne et de colonne sont statistiquement significativement associées (`p-value = r chisq$p.value`).

- *Code R pour calculer l'AFC.* Fonction `R` : `CA()` [`FactoMiner`].
  Format simplifié : `CA (X, ncp = 5, graph = TRUE)`
  `X` : un data frame (tableau de contingence)
  `ncp` : nombre de dimensions à conserver dans les résultats finaux.
  `graph` : une valeur logique. Si TRUE le graphique est affiché.

  Pour calculer l'AFC, tapez ceci :
  ```
  library ("FactoMineR")
  res.ca <- CA (housetasks, graph = FALSE)
  ```

  Le résultat de la fonction `CA()` est une liste comprenant :
  `print(res.ca)`

  ```
  # **Results of the Correspondence Analysis (CA)**
  # The row variable has 13 categories ; the column variable has 4 categories
  # The chi square of independence between the two variables is equal to
  1944 (p-value = 0 ).
  # *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"
```

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

4.1.2 Visualisation et interprétation

Nous utiliserons les fonctions suivantes [dans `factoextra`] pour aider à l'interprétation et à la visualisation de l'analyse factorielle des correspondances :
`get_eigenvalue(res.ca)` : Extraction des valeurs propres/variances expliquées par chaque axe principal
`fviz_eig(res.ca)` : visualisation des valeurs propres
`get_ca_row(res.ca)`, `get_ca_col(res.ca)` : Extraction des résultats pour les lignes et les colonnes, respectivement.
`fviz_ca_row(res.ca)`, `fviz_ca_col(res.ca)` : Visualisation des résultats pour les lignes et les colonnes, respectivement.
`fviz_ca_biplot (res.ca)` : Créez un biplot des lignes et des colonnes.

Dans les sections suivantes, nous allons illustrer chacune de ces fonctions.

- *Significativité statistique.* Pour interpréter l'AFC, la première étape consiste à évaluer s'il existe une dépendance significative entre les lignes et les colonnes.
  Une méthode rigoureuse consiste à utiliser la statistique de `chi2` pour examiner l'association entre les modalités des lignes et celles des colonnes. Cela apparaî t en haut du rapport généré par la fonction summary (`res.ca`) ou `print (res.ca)`, voir la section `ref(r-code-to-comput-ca)`. Une statistique de `chi2` élevée signifie un lien fort entre les lignes et les colonnes.
  Dans notre exemple, l'association est très significative (`chi-square` : 1944.456, $p = 0$).

```
# Statistiques de khi2
chi2 <- 1944.456
# Degré de liberté
df <- (nrow (housetasks) - 1) * (ncol (housetasks) - 1)
# p value
pval <- pchisq (chi2, df = df, lower.tail = FALSE)
pval
# [1] 0
```

- *Valeurs propres/Variances.* L'examination des valeurs propres permet de déterminer le nombre d'axes principaux à considérer. Les valeurs propres correspondent à la quantité d'informations retenue par chaque axe. Elles sont grandes pour le premier axe et petites pour l'axe suivant.

Les valeurs propres et la proportion de variances expliquées (capturées) par les différents axes peuvent être extraites à l'aide de la fonction :
`get_eigenvalue() [factoextra package]`.

```
library ("factoextra")
eig.val <- get_eigenvalue (res.ca)
eig.val
# eigenvalue variance.percent cumulative.variance.percent # Dim.1 0.543
48.7 48.7
# Dim.2 0.445 39.9 88.6
# Dim.3 0.127 11.4 100.0
```

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, 48.69% plus 39.91% estégal à 88.6%, 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 80% des variances totales expliquées, utilisez le nombre de dimensions nécessaires pour y parvenir.
Notez qu'une analyse est bonne lorsque les premières dimensions représentent une grande partie de la variabilité.
Dans notre analyse, les deux premiers axes expliquent 88.6% 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.
Le graphique des valeurs propres peut être produit à l'aide de la fonction `fviz_eig()` ou `fviz_screeplot()` [package `factoextra`].

```
fviz_screeplot (res.ca, addlabels = TRUE, ylim = c(0, 50))
```

Le point auquel le graphique des valeurs propres montre un virage (appelé "coude") peut être considéré comme indiquant le nombre optimal d'axes principaux à retenir. Il estégalement possible de calculer une valeur propre moyenne au-dessus de laquelle l'axe doit être conservé dans le résultat.
Nos données contiennent 13 lignes et 4 colonnes.
Si les donnéesétaient aléatoires, la valeur attendue de la valeur propre pour chaque

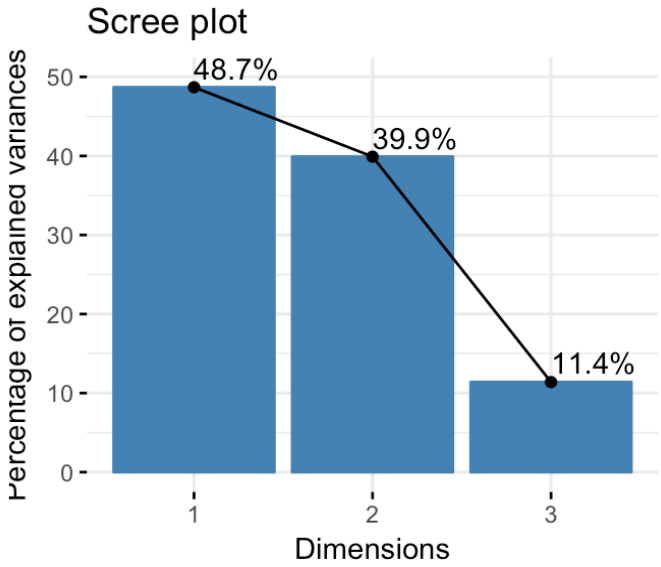

FIGURE 4.3 – Graphique des valeurs propres

axe serait `1 / (nrow (housetasks) -1)` $= 1/12 = 8,33\%$ en termes de lignes.
De même, l'axe moyen devrait représenter `1 / (ncol (housetasks) -1)` $= 1/3 = 33,33\%$ en termes de 4 colonnes.
Selon (M. T. Bendixen 1995) : *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.*
Le code `R` ci-dessous, montre le graphique des valeurs propres avec une droite en pointillée rouge spécifiant la valeur propre moyenne :

```
fviz_screeplot (res.ca) + geom_hline (yintercept = 33.33, linetype = 2,
color = "red")
```

Selon le graphique ci-dessus, seules les dimensions 1 et 2 doivent être considérées pour l'interprétation de la solution. La dimension 3 explique seulement $11,4\%$ de l'inertie totale, ce qui est inférieur à la valeur moyenne des axes ($33,33\%$) et trop petit pour être conservé pour une analyse plus approfondie.
Notez que vous pouvez utiliser plus de 2 dimensions. Cependant, il est peu probable que les dimensions supplémentaires contribuent de manière significative à l'interprétation de la nature de l'association entre les lignes et les colonnes.
Les dimensions 1 et 2 expliquent environ $48,7\%$ et $39,9\%$ de l'inertie totale, respectivement. Cela correspond à un total cumulé de $88,6\%$ de l'inertie totale retenue par les 2 dimensions. Plus la rétention estélevée, plus la subtilité contenue dans les données d'origine est conservée dans la solution de l'AFC à faible dimension (cf. Bendixen, 2003).

- *Biplot.* La fonction `fviz_ca_biplot()` [package `factoextra`] peut être utilisée pour dessiner le biplot des lignes et des colonnes.

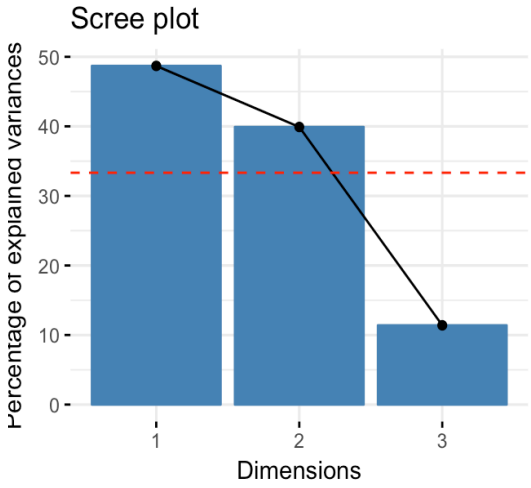

FIGURE 4.4 – Graphique des valeurs propres avec la valeur propre moyenne

```
# repel = TRUE pouréviter le chevauchement de texte fviz_ca_biplot (res.ca,
repel = TRUE)
```

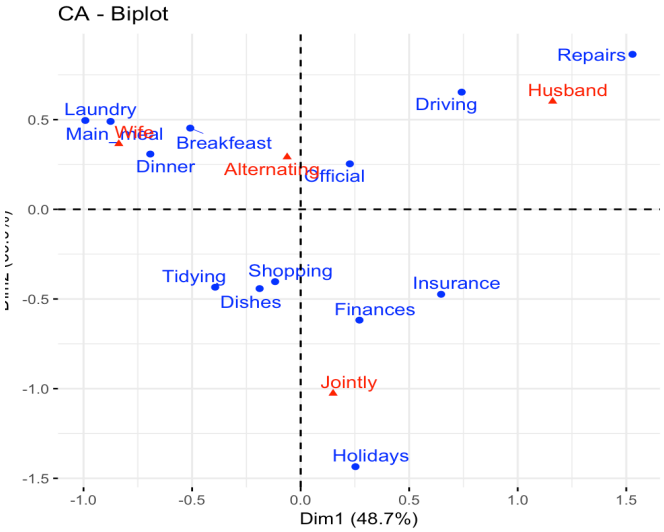

FIGURE 4.5 – Biplot lignes-colonnes

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.

Ce graphique montre que :

— Les lignes `Dinner, Breakfeast` et `Laundry` sont associées le le plus à la colonne "Wife"

— Les lignes Driving et `Repairs` sont associées le plus à la colonne "Husband".

— ...

— Le graphique, dans Figure 4.5, représente une analyse symétrique montrant les profils lignes et colonnes simultanément dans un espace commun. Dans ce cas, seule la distance entre les points lignes ou la distance entre les points colonnes peut être vraiment interprétée.

— La distance entre les points lignes et les points colonnes n'a pas de sens dans l'absolue ! Vous ne pouvez faire que des observations générales.

— Pour interpréter la distance entre les points colonnes et lignes, les profils colonnes doivent être représentés dans l'espace des profils lignes ou vice versa. Ce type de graphique est appelé biplot em asymétrique et est discuté à la fin de cet article.

La prochaineétape, pour l'interprétation, est de déterminer les points lignes et colonnes qui contribuent le plus à la définition des différents axes principaux retenus dans le modèle.

- *Graphique des points lignes.*

— *Résultats.* La fonction `get_ca_row()` [`factoextra`] est utilisée pour extraire les résultats pour les lignes. Cette fonction renvoie une liste contenant les coordonnées, les `cos2` et les contributions des lignes :

tt 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"
```

Les composants de la fonction `get_ca_row()` peuvent être utilisés dans le graphique des lignes comme suit :

— `row$coord` : coordonnées des lignes. Utilisées pour créer le nuage de points.

— `row$cos2` : qualité de représentation des lignes.

— `var$contrib` : contribution des lignes (en %) à la définition des dimensions.

Notez qu'il est possible de visualiser des points lignes et de les colorer en fonction de (i) soit de leurs cosinus carré (qualité de représentation), (ii) soit de leurs contributions à la définition des dimensions (`contrib`).

Inspection des différentséléments :

```
# Coordonnées
head(row$coord)
# Cos2 : qualité de représentation
head(row$cos2)
# Contributions
head(row$contrib)
```

Dans cette section, nous décrivons comment visualiser uniquement les points lignes. Ensuite, nous mettons enévidence les lignes selon (i) leurs qualités de représentation (ii) leurs contributions aux dimensions.

— *Coordonnées des points lignes*. Le code `R` ci-dessous affiche les coordonnées de chaque point ligne pour chacune des dimensions (1, 2 et 3) :

```
head(row$coord)
# Dim 1 Dim 2 Dim 3
# Laundry -0.992 0.495 -0.3167
# Main_meal -0.876 0.490 -0.1641
# Dinner -0.693 0.308 -0.2074
# Breakfeast -0.509 0.453 0.2204
# Tidying -0.394 -0.434 -0.0942
# Dishes -0.189 -0.442 0.2669
```

Utilisez la fonction `fviz_ca_row()` [dans `factoextra`] pour visualiser uniquement les points lignes : `fviz_ca_row(res.ca, repel = TRUE)`

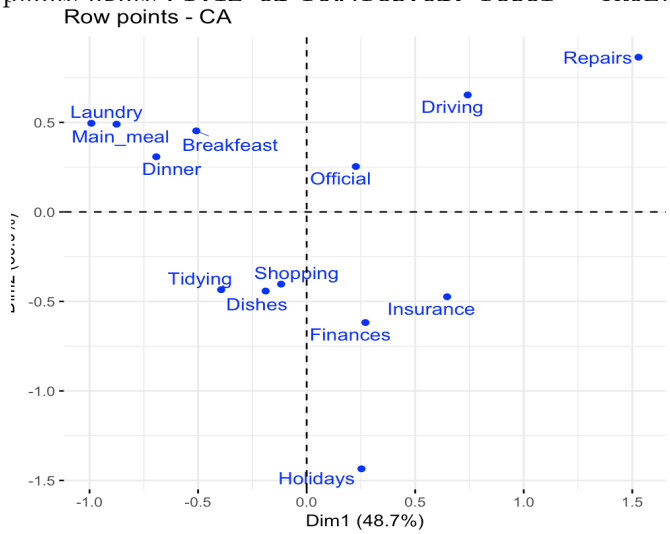

FIGURE 4.6 – Visualisation des points lignes

Il est possible de modifier la couleur et la forme des points lignes en utilisant les

arguments `col.row` et `shape.row` comme suit : `fviz_ca_row (res.ca, col.row = "steelblue", shape.row = 15)`
Le graphique, dans Figure 4.6, montre les relations entre les points lignes :
— Les lignes avec un profil similaire sont regroupées.
— Les lignes corrélées négativement sont positionnées sur des côtés opposés de l'origine de du graphique (quadrants opposés).
— La distance entre les points lignes et l'origine mesure la qualité des points lignes sur le graphique. Les points lignes qui sont loin de l'origine sont bien représentés sur le graphique.

— *Qualité de représentation des lignes.* Le résultat de l'analyse montre que le tableau de contingence est bien représenté dans un espace de faibles dimensions en utilisant l'AFC. Les deux dimensions 1 et 2 sont suffisantes pour conserver 88.6% de l'inertie totale (variation) contenue dans les données.
Cependant, tous les points ne sont pas aussi bien représentes dans les deux dimensions.
Rappelons que la qualité de représentation des lignes sur le graphique est appelée cosinus carré (`cos2`).

Le `cos2` mesure le degré d'association entre les lignes/colonnes et un axe particulier. Le `cos2` des points lignes peut être extrait comme suit :

```
head(row$cos2, 4)
# Dim 1 Dim 2 Dim 3
# Laundry 0.740 0.185 0.0755
# Main_meal 0.742 0.232 0.0260
# Dinner 0.777 0.154 0.0697
# Breakfeast 0.505 0.400 0.0948
```

Les valeurs de `cos2` sont comprises entre 0 et 1. La somme des `cos2` pour les lignes sur toutes les dimensions de l'AFC estégale à 1.
La qualité de représentation d'une ligne ou d'une colonne dans n dimensions est simplement la somme des cosinus carré de cette ligne ou colonne sur les n dimensions.
Si un point ligne est bien représenté par deux dimensions, la somme des `cos2` est proche de 1. Pour certainséléments lignes, plus de 2 dimensions sont nécessaires pour représenter parfaitement les données.
Il est possible de colorer les points lignes par leurs `cos2` à l'aide de l'argument `col.row = "cos2"`. Cela produit un gradient de couleurs, qui peut être personnalisé à l'aide de l'argument `gradient.cols`. Par exemple, `gradient.cols = c ("white", "blue", "red")` signifie que :
— les variables à faible valeur cos2 seront colorées en “white” (blanc)
— les variables avec des valeurs moyennes de cos2 seront colorées en “blue” (bleu)
— les variables avec des valeursélevées de cos2 seront colorées en “red” (rouge)

```
# Colorer en fonction du cos2 :
```

```
fviz_ca_row (res.ca, col.row = "cos2",
gradient.cols = c ("#00AFBB", "#E7B800", "#FC4E07"),
repel = TRUE)
```

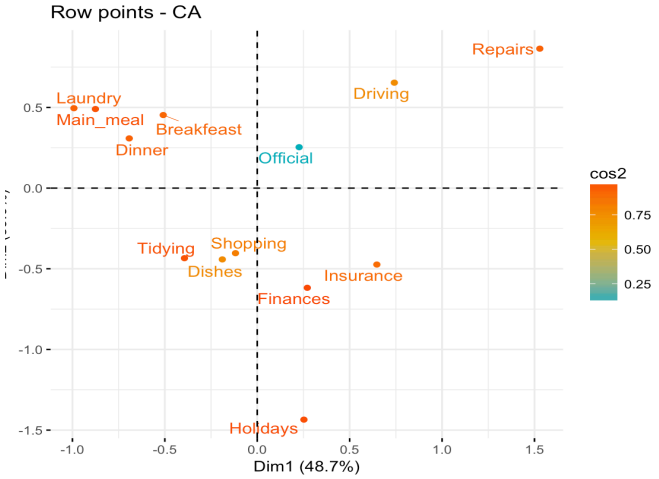

FIGURE 4.7 – Coloration des points lignes en fonctions de leurs cos2

Noter qu'il estégalement possible de modifier la transparence des points lignes en fonction de leurs cos2 à l'aide de l'option `alpha.row = "cos2"`. Par exemple, on peut taper ceci :

```
# Change la transparence par cos2
fviz_ca_row (res.ca, alpha.row = "cos2")
```

On peut visualier le `cos2` des points lignes sur toutes les dimensions en utilisant le package `corrplot` : `library("corrplot")`

```
corrplot(row$cos2, is.corr = FALSE)
```

Il estégalement possible de créer un bar plot du cos2 des lignes en utilisant la fonction `fviz_cos2()` [`factoextra`] :

```
# Cos2 des lignes sur Dim.1 et Dim.2
fviz_cos2(res.ca, choice = "row", axes = 1 :2)
```

Noter que tous les points lignes sauf Official sont bien représentés par les deux premières dimensions. Cela implique que la position du point correspondant à l'élément Official sur le graphique doit être interprétée avec prudence. Plus de dimensions sont probablement nécessaires pour interpréter l'élément Official.

— *Contributions des lignes aux dimensions.* La contribution des lignes (en %) à la définition des dimensions peut être extraite comme suit :

```
head(row$contrib)
# Dim 1 Dim 2 Dim 3
# Laundry 18.287 5.56 7.968
```

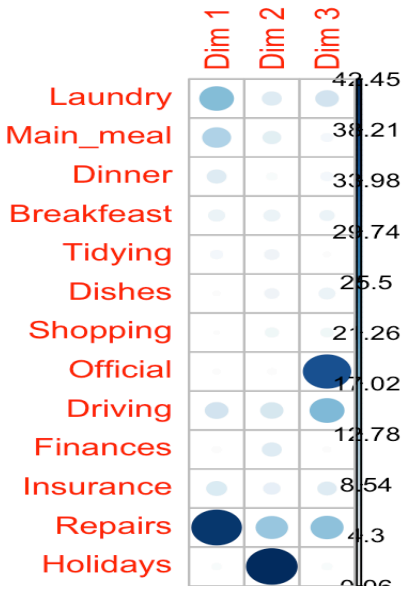

FIGURE 4.8 – Visualisation du `cos2`

```
# Main_meal 12.389 4.74 1.859
# Dinner 5.471 1.32 2.097
# Breakfeast 3.825 3.70 3.069
# Tidying 1.998 2.97 0.489
# Dishes 0.426 2.84 3.634
```

Les lignes avec des valeursélevées, contribuent le mieux à la définition des dimensions.

— Les lignes qui contribuent le plus à Dim.1 et Dim.2 sont les plus importantes pour expliquer la variabilité dans le jeu de données.
— Les lignes qui ne contribuent pas beaucoup à aucune dimension ou qui contribuent aux dernières dimensions sont moins importantes.

Il est possible d'utiliser la fonction `corrplot()` [package `corrplot`] pour mettre enévidence les points lignes les plus contributifs pour chaque dimension :

```
library("corrplot")
corrplot(row$contrib, is.corr=FALSE)
```

La fonction `fviz_contrib()` [package `factoextra`] peut être utilisée pour créer un bar plot des contributions des lignes. Si vos données contiennent de nombreuses lignes, vous pouvez décider de ne montrer que les lignes les plus contributives. Le code `R` ci-dessous montre le top 10 des lignes les plus contributives aux dimensions :

```
# Contributions des lignes à la dimension 1
```

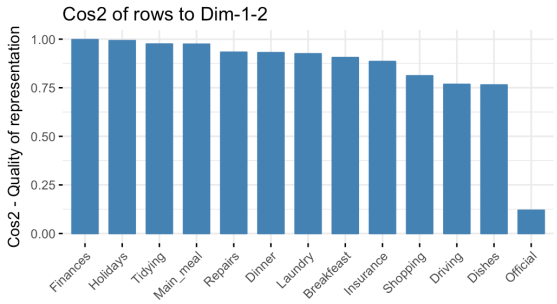

FIGURE 4.9 – Barplot de cos2

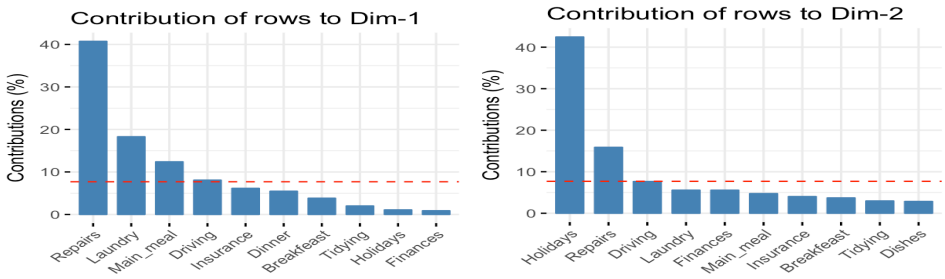

FIGURE 4.10 – Contributions des lignes en %

```
fviz_contrib(res.ca, choice = "row", axes = 1, top = 10)
# Contributions des lignes à la dimension 2
fviz_contrib(res.ca, choice = "row", axes = 2, top = 10)
```

La contribution totale aux dimensions 1 et 2 peut être obtenue comme suit :

```
# Contribution totale aux dimensions 1 et 2
fviz_contrib (res.ca, choice = "row", axes = 1 :2, top = 10)
```

La droite en pointillée rouge, sur le graphique ci-dessus, indique la valeur moyenne attendue, si les contributions étaient uniformes. Le calcul de la contribution attendue, sous l'hypothèse nulle, a été détaillée dans le chapitre concernant l'analyse en composantes principales.

On peut voir que : les lignes `Repairs, Laundry, Main_meal` et `Driving` sont les plus importants dans la définition de la première dimension. Les lignes `Holidays` et `Repairs` contribuent le plus à la dimension 2.

Les lignes les plus importantes (ou contributives) peuvent être mise en avant sur le graphique comme suit :

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

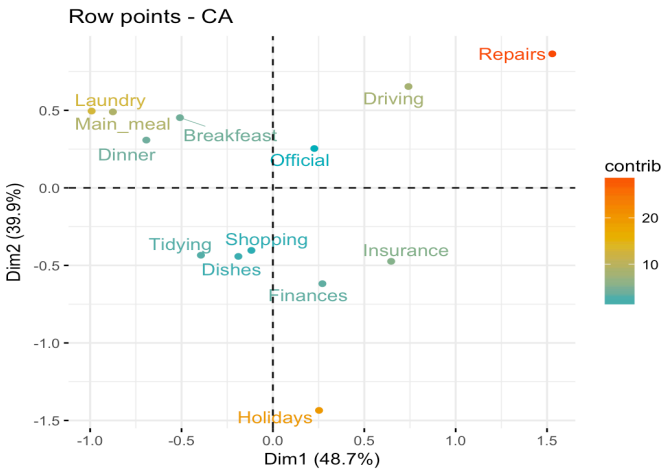

FIGURE 4.11 – Coloration des points lignes en fonctions de leurs cos2

Le graphique donne une idée de la contribution des lignes aux différents pôles des dimensions. Il est évident que les catégories `Repair` et `Driving` ont une contribution importante au pôle positif de la première dimension, tandis que les catégories `Laundry` et `Main_meal` ont une contribution majeure au pôle négatif de la première dimension, ...

En d'autres termes, la dimension 1 est principalement définie par l'opposition de `Repair` et `Driving` (pôle positif) avec `Laundry` et `Main_meal` (pôle négatif).

Il est à noter qu'il est également possible de contrôler la transparence des points lignes en fonction de leurs contributions en utilisant l'option `alpha.row = "contrib"`. Par exemple, on peut taper ceci :

```
# Changez la transparence par les valeurs de contrib
fviz_ca_row (res.ca, alpha.row = "contrib", repel = TRUE)
```

- *Graphique des colonnes*
  — *Résultats.* La fonction `get_ca_col()` [`factoextra`] sert à extraire les résultats pour les colonnes. Cette fonction renvoie une liste contenant les coordonnées, le `cos2`, la contribution et l'inertie des colonnes :

```
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"
```

Le résultat, pour les colonnes, donne les mêmes informations que celles décrites pour les lignes. Pour cette raison, nous allons simplement afficher le résultat pour les colonnes dans cette section avec seulement un commentaire très bref.
Pour accéder aux différents composants, on peut utiliser ceci :

```
# Coordonnées
head(col$coord)
# Qualité de représentation
head(col$cos2)
# Contributions
head(col$contrib)
```

— *Graphiques : qualité et contribution.*
La fonction `fviz_ca_col()` est utilisée pour produire le graphique des colonnes. Pour créer un graphique simple, on peut taper :

```
fviz_ca_col (res.ca)
```

Comme les points lignes, il estégalement possible de colorer les points colonnes en fonction de leurs cos2 :

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

Le code `R` ci-dessous crée un `bar-plot` du `cos2` des colonnes :

```
fviz_cos2 (res.ca, choice = "col", axes = 1 :2)
```

Rappelons que la valeur du `cos2` est comprise entre 0 et 1. Un `cos2` proche de 1 correspond à une colonne/ligne bien représentée sur le graphique de l'AFC.
Il faut noter que, seul la colonne `Alternating` n'est pas très bien représentée sur les deux premières dimensions. La position de cet élément doit être interprétée avec prudence dans l'espace formé par les dimensions 1 et 2.

Pour visualiser la contribution des colonnes aux deux premières dimensions, on peut taper ceci :

```
fviz_contrib (res.ca, choice = "col", axes = 1 :2)
```

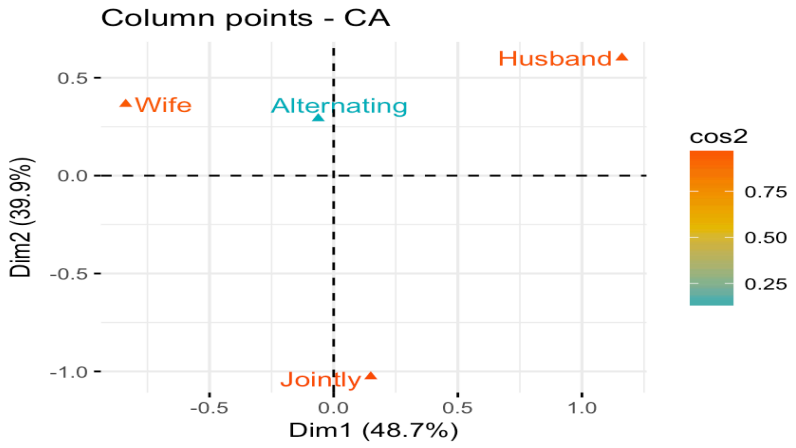

FIGURE 4.12 – Coloration des points colonnes en fonctions de leurs cos2

- *Types de biplots.*
  Le `biplot` permet de représenter simultanément les lignes et les colonnes sur le même graphique. Nous avons déjà décrit comment créer des biplots pour l'AFC. Ici, nous décrivons différents types de biplots pour l'AFC.
  — *Biplot symétrique.*
  Comme mentionné ci-dessus, le graphique standard de l'AFC est un biplot symétrique dans lequel les lignes (points bleus) et les colonnes (triangles rouges) sont représentées dans le même espace à l'aide des coordonnées principales. Ces coordonnées représentent les profils des lignes et des colonnes. Dans ce cas, seule la distance entre les points lignes ou la distance entre les points colonnes peut être vraiment interprétée.
  Avec un biplot symétrique, la distance entre les lignes et les colonnes ne peut pas être interprétée. Seules des conclusions générales peuvent être tirées.
  `fviz_ca_biplot(res.ca, repel = TRUE)`
  On doit noter que, pour interpréter la distance entre les points colonnes et les points lignes, le moyen le plus simple est de créer un biplot asymétrique. Cela signifie que les profils des colonnes doivent être représentés dans l'espace des lignes ou vice versa.
  — *Biplot asymétrique.*
  Pour créer un biplot asymétrique, les points lignes (ou colonnes) sont visualisés

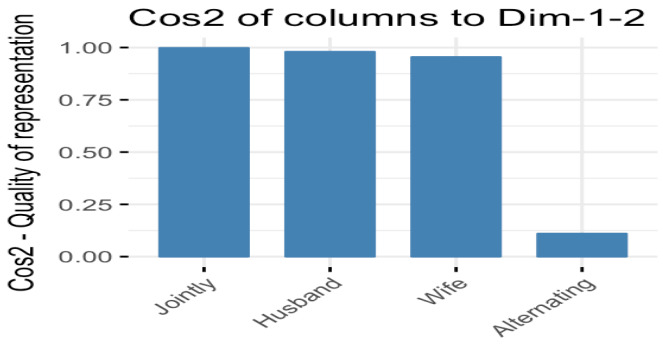

FIGURE 4.13 – Bar plot du cos2

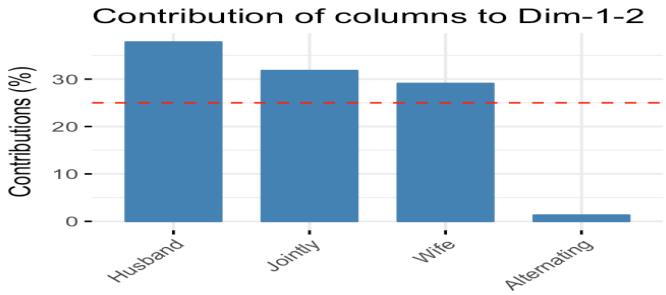

FIGURE 4.14 – Contribution des colonnes en dimension 1-2

à partir des "coordonnées standard" (S) et les points colonnes (ou lignes) sont visualisés à partir des "coordonnées principales" (P) (M. Bendixen 2003).
Pour un axe donné, les coordonnées standards et principales sont liées comme suit :
`P = sqrt (valeur propre) x S`
— `P` : Coordonnée principale d'une ligne (ou d'une colonne) sur l'axe
— valeur propre : Valeur propre de l'axe

Selon la situation, d'autres types de graphiques peuvent être définis à l'aide de l'argument `map` (Nenadic and Greenacre, 2007) dans la fonction `fviz_ca_biplot()` [`factoextra`].
Les options autorisées pour l'argument `map` sont :

1. `"rowprincipal"` ou `"colprincipal"` : Biplots asymétriques. Les lignes sont en

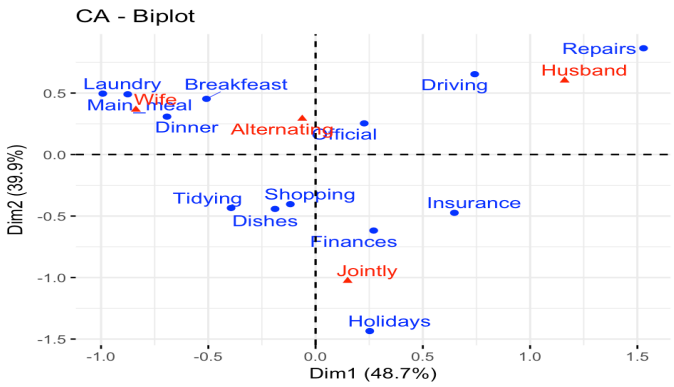

FIGURE 4.15 – Biplot des profils lignes et colonnes

coordonnées principales et les colonnes en coordonnées standard, ou vice versa. La première préserve la métrique des lignes et la deuxième, celle des colonnes.
   - `"rowprincipal"` : les colonnes sont représentées dans l'espace des lignes
   - `"colprincipal"` : les lignes sont représentées dans l'espace des colonnes
2. `"symbiplot"` : Biplot symétrique. Ne conserve pas les métriques des lignes et des colonnes.
3. `"rowgab"` ou `"colgab"` : graphiques asymétriques proposés par Gabriel & Odoroff (Gabriel and Odoroff, 1990) :
   - `"rowgab"` : Les lignes en coordonnées principales et les colonnes en coordonnées standard multipliées par la masse.
   - `"colgab"` : Les colonnes en coordonnées principales et les lignes en coordonnées standard multipliées par la masse.
4. `"rowgreen"` ou `"colgreen"` : Biplots de contribution. Met enévidence, visuellement les points les plus contributifs (Greenacre 2006b).
   - `"rowgreen"` : Les lignes en coordonnées principales et les colonnes en coordonnées standard multipliées par la racine carrée de la masse.
   - `"colgreen"` : Les colonnes en coordonnées principales et les lignes dans les coordonnées standard multipliées par la racine carrée de la masse.

Le code`R` ci-dessous crée un biplot asymétrique standard :

```
fviz_ca_biplot (res.ca, map = "rowprincipal", arrow = c(TRUE, TRUE),
repel = TRUE)
```

Nous avons utilisé, l'argument `arrow`, qui est un vecteur logique, de longueur 2, spécifiant si le graphique doit contenir des points (`FALSE`, par défaut) ou des flèches (`TRUE`). La première valeur définit les lignes et la seconde valeur définit les colonnes. Si l'angle entre deux flèches est aigu, alors il y a une forte association entre les lignes et les colonnes correspondantes.
Pour interpréter la distance entre les lignes et les colonnes, vous devriez projeter

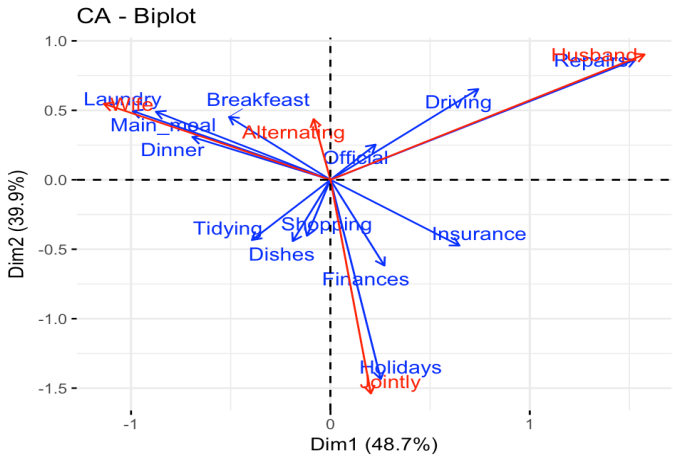

FIGURE 4.16 – Biplot asymétrique

perpendiculairement des points lignes sur la flèche de la colonne.

— *Biplot des contributions.*
Dans le biplot symétrique standard (mentionné dans la section précédente), il est difficile de connaî tre les points les plus contributifs à la solution de l'AFC.
Michael Greenacre a proposé une nouvelle solution (biplot de contribution) qui intègre la contribution des points (M. Greenacre, 2013).
Dans ce graphique, les points qui contribuent très peu à la solution sont proches du centre du biplot et sont relativement peu importants pour l'interprétation.
Un biplot de contribution peut être visualisé en utilisant l'argument `map = "rowgreen"` ou `map = "colgreen"`.
Tout d'abord, vous devez décider si vous voulez analyser les contributions des lignes ou de celles des colonnes.
Dans notre exemple, nous allons interpréter la contribution des lignes. L'argument `map = "colgreen"` est utilisé. Dans ce cas, rappelez-vous que les colonnes sont en coordonnées principales et les lignes sont en coordonnées standard multipliées par la racine carrée de la masse. Pour une ligne donnée, le carré de sa coordonnée sur un axe i correspond exactement à sa contribution à l'axe i.

```
fviz_ca_biplot (res.ca, map = "colgreen", arrow = c (TRUE, FALSE), repel
= TRUE)
```

Dans le graphique ci-dessus, la position des points colonnes est inchangée par rapport

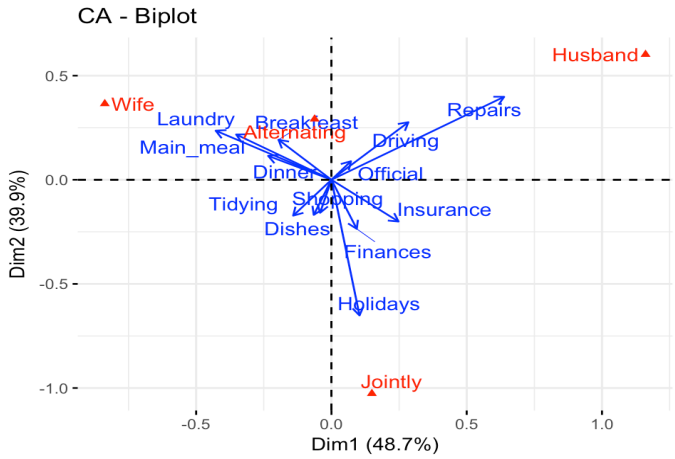

FIGURE 4.17 – Biplot de contributions des lignes

.

à celle du biplot conventionnel. Cependant, les distances entre les points lignes et l'origine du graphique sont liées à leurs contributions aux axes principaux en considération.
Plus une flèche est proche (en termes de distance angulaire) d'un axe, plus la contribution de la ligne sur cet axe par rapport à l'autre axe est importante. Si la flèche est à mi-chemin entre les deux axes, la ligne contribue aux deux axes de manière identique.

— Il est évident que la ligne `Repairs` a une contribution importante au pôle positif de la première dimension, tandis que les lignes `Laundry` et `Main_meal` ont une contribution majeure au pôle négatif de la première dimension.
— La dimension 2 est principalement définie par la ligne `Holidays`.
— La ligne `Driving` contribue aux deux axes de manière identique.

- *Description des dimensions.*
Pour identifier facilement les lignes et les colonnes les plus associées aux axes principaux, vous pouvez utiliser la fonction `dimdesc()` [`FactoMineR`]. Les lignes/colonnes sont triées en fonction de leurs coordonnées dans le résultat de `dimdesc()`.
`# Description de la dimension`
`res.desc <- dimdesc(res.ca, axes = c(1, 2))` Description de la dimension 1 :
`# Description de la dimension 1 par les lignes head(res.desc[[1]]$row,`

```
4)

# coord
# Laundry -0.992
# Main_meal -0.876
# Dinner -0.693
# Breakfeast -0.509

# Description de la dimension 1 par les colonnes
head(res.desc[[1]]$col, 4)

# coord
# Wife -0.8376
# Alternating -0.0622
# Jointly 0.1494
# Husband 1.1609
```

Description de la dimension 2 :

```
# Description de la dimension 2 par les lignes
res.desc[[2]]$row
# Description de la dimension 1 par les colonnes
res.desc[[2]]$col
```

4.1.3 Eléments supplémentaires

- *Format des données*. Nous utiliserons le jeu de données `children` [`FactoMineR`]. Il contient 18 lignes et 8 colonnes :

```
data(children)
# head(children)
```

Les données utilisées ici correspondent à un tableau de contingence décrivant les réponses données par différentes catégories de personnes à la question suivante : Quelles sont les raisons qui peuvent faire hésiter une femme ou un couple à avoir des enfants ?

Seules certaines lignes et colonnes seront utilisées pour effectuer l'analyse factorielle des correspondances. Les coordonnées des autres lignes / colonnes restantes seront prédites après l'AFC.

Dans la terminologie de l'AFC, nos données contiennent des :
— Lignes actives (`lignes 1 :14`) : Lignes utilisées lors de l'AFC.
— Lignes supplémentaires (`15 :18`) : Les coordonnées de ces lignes seront prédites à l'aide des informations de l'AFC obtenues avec les lignes/colonnes actives

	unqualified	cep	bepc	high_school_diploma	university	thirty	fifty	more_fifty
money	51	64	32	29	17	59	66	70
future	53	90	78	75	22	115	117	86
unemployment	71	111	50	40	11	79	88	177
circumstances	1	7	5	5	4	9	8	5
hard	7	11	4	3	2	2	17	18
economic	7	13	12	11	11	18	19	17
egoism	21	37	14	26	9	14	34	61
employment	12	35	19	6	7	21	30	28
finances	10	7	7	3	1	8	12	8
war	4	7	7	6	2	7	6	13
housing	8	22	7	10	5	10	27	17
fear	25	45	38	38	13	48	59	52
health	18	27	20	19	9	13	29	53
work	35	61	29	14	12	30	63	58
comfort	2	4	3	1	4	NA	NA	NA
disagreement	2	8	2	5	2	NA	NA	NA
world	1	5	4	6	3	NA	NA	NA
to_live	3	3	1	3	4	NA	NA	NA

FIGURE 4.18 – Données sur les réponses de différentes catégories de personnes

— Colonnes actives (`colonnes 1 :5`) : Colonnes utilisées pour l'AFC.
— Colonnes supplémentaires (`col.sup 6 :8`) : Comme pour les lignes supplémentaires, les coordonnées de ces colonnes seront également prédites.

- *Spécification dans l'AFC.*
  Pour spécifier des lignes/colonnes supplémentaires, la fonction `CA()` [`FactoMineR`] peut être utilisée comme suit :
  `CA (X, ncp = 5, row.sup = NULL, col = NULL, graph = TRUE)`
  — `X` : data frame (tableau de contingence)
  — `row.sup` : un vecteur numérique spécifiant les positions des lignes supplémentaires dans le tableau.
  — `col.sup` : un vecteur numérique spécifiant les positions des colonnes supplémentaires.
  — `ncp` : nombre de dimensions conservées dans le résultat final.
  — `graph` : une valeur logique. Si TRUE le graphique est montré.
  Par exemple, on peut taper ceci :
  `res.ca <- CA (children, row.sup = 15 :18, col.sup = 6 :8, graph = FALSE)`

- *Biplot des lignes et des colonnes.*
  `fviz_ca_biplot (res.ca, repel = TRUE)`
  — Les lignes actives sont en bleu
  — Les lignes supplémentaires sont en bleu foncé
  — Les colonnes sont en rouge
  — Les colonnes complémentaires sont en noir

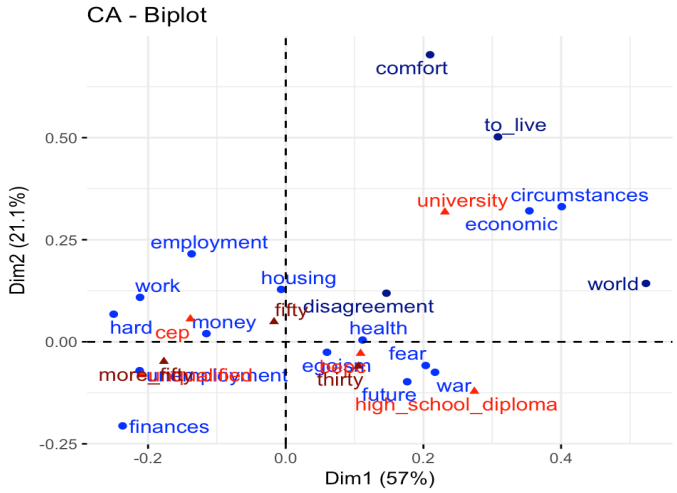

FIGURE 4.19 – Biplot des lignes et des colonnes

Il est également possible de masquer les lignes et les colonnes supplémentaires à l'aide de l'argument invisible :

```
fviz_ca_biplot(res.ca, repel = TRUE, invisible = c("row.sup", "col.sup"))
```

- *Lignes supplémentaires.*
  Résultats prédits (coordonnées et cos2) pour les lignes supplémentaires :

```
res.ca$row.sup
# $coord
# Dim 1 Dim 2 Dim 3 Dim 4
# comfort 0.210 0.703 0.0711 0.307
# disagreement 0.146 0.119 0.1711 -0.313
# world 0.523 0.143 0.0840 -0.106
# to_live 0.308 0.502 0.5209 0.256
#
# $cos2
# Dim 1 Dim 2 Dim 3 Dim 4
# comfort 0.0689 0.7752 0.00793 0.1479
# disagreement 0.1313 0.0869 0.17965 0.6021
# world 0.8759 0.0654 0.02256 0.0362
# to_live 0.1390 0.3685 0.39683 0.0956
```

Graphique des lignes actives et supplémentaires : `fviz_ca_row(res.ca, repel =`

```
TRUE)
```

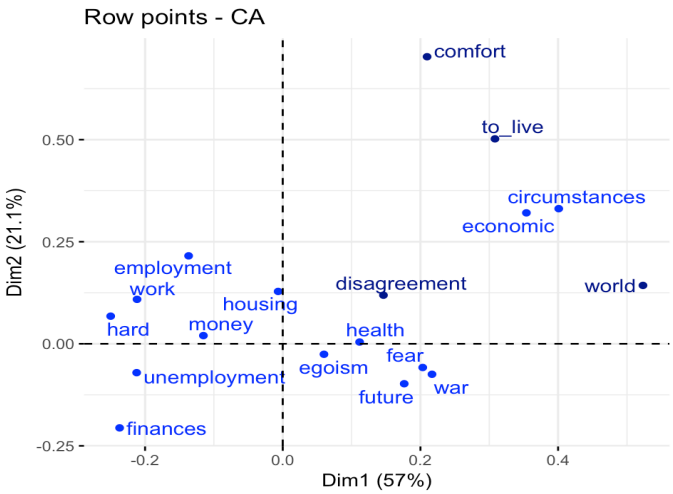

FIGURE 4.20 – Coordonnées et cos2 pour les lignes supplémentaires

Les lignes supplémentaires sont montrées en bleu foncé.

- *Colonnes supplémentaires.*
  Résultats prédits (coordonnées et `cos2`) pour les colonnes supplémentaires :

```
res.ca$col.sup
# $coord
# Dim 1 Dim 2 Dim 3 Dim 4
# thirty 0.1054 -0.0597 -0.1032 0.0698
# fifty -0.0171 0.0491 -0.0157 -0.0131
# more_fifty -0.1771 -0.0481 0.1008 -0.0852
#
# $cos2
# Dim 1 Dim 2 Dim 3 Dim 4
# thirty 0.1376 0.0441 0.13191 0.06028
# fifty 0.0109 0.0899 0.00919 0.00637
# more_fifty 0.2861 0.0211 0.09267 0.06620
```

  Graphique des colonnes actives et supplémentaires : `fviz_ca_col(res.ca, repel = TRUE)`

  Les colonnes supplémentaires sont affichées en noir.

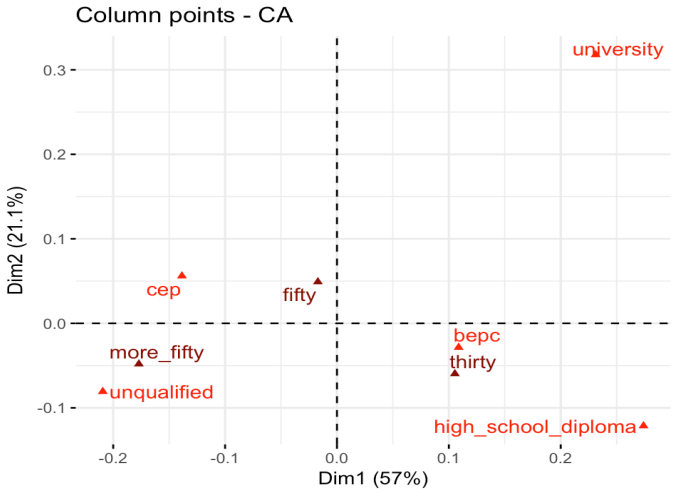

FIGURE 4.21 – Coordonnées et `cos2` pour les colonnes supplémentaires

4.1.4 Filtrer les résultats

Si vous avez beaucoup de lignes/colonnes, il est possible de visualiser seulement certaines d'entre elles en utilisant les arguments `select.row` et `select.col`. Les valeurs autorisées sont `NULL` ou une `liste` contenant le nom des arguments, `cos2` ou `contrib` :

- `name` : est un vecteur de caractères contenant les noms des colonnes/lignes à visualiser
- `cos2` : si `cos2` est dans `[0, 1]`, exemple : 0.6, alors les colonnes/lignes avec un `cos2`> 0.6 sont montrées. Si `cos2`> 1, exemple : 5, les top 5 individus/variables actifs ainsi que les top 5 colonnes/lignes supplémentaires avec le `cos2` le plusélevé sont montrés
- `contrib` : si `contrib`> 1, exemple : 5, alors les top 5 colonnes/lignes avec les contributions les plus importantes sont montrés

```
# Visualiser les lignes avec cos2> = 0.8
fviz_ca_row (res.ca, select.row = list (cos2 = 0.8))
# Top 5 lignes actives et top 4 lignes suppl. avec le cos2 le plusélevé
fviz_ca_row (res.ca, select.row = list (cos2 = 5))
# Sélectionner par noms
name <- list(name = c("employment", "fear", "future"))
fviz_ca_row(res.ca, select.row = name)
# Top 5 des lignes et des colonnes les plus contributives
fviz_ca_biplot(res.ca, select.row = list(contrib = 5), select.col = list(contrib
= 5)) + theme_minimal()
```

4.1.5 Outliers

Si un ou plusieurs "outliers" (valeurs atypiques ou aberrantes) sont présents dans le tableau de contingence, ils peuvent dominer l'interprétation des axes (M. Bendixen, 2003).
Les valeurs aberrantes sont des points qui ont des coordonnées et des contributions très-sélevées en valeur absolue. Sur le graphique, elles sont très loin de l'origine. Dans ce cas, les points lignes et colonnes restants ont tendance à êtreétroitement regroupés dans le graphique, lequel devient difficile à interpréter.
Il n'y a pas de valeurs aberrantes apparentes dans nos données. S'il y avait des valeurs aberrantes dans les données, elles doivent être supprimées ou traitées comme des points supplémentaires lors de l'AFC.

4.1.6 Exportation des résultats

- *Exporter les graphiques en PDF/PNG.* Pour enregistrer les différents graphiques en fichiers pdf ou png, nous commençons par créer des graphiques d'intérêt en tant qu'objet `R` :
  ```
  # Scree plot
  scree.plot <- fviz_eig(res.ca)
  # Biplot des lignes et colonnes
  biplot.ca <- fviz_ca_biplot (res.ca)
  ```
  Ensuite, les graphiques peuvent être exportées dans un seul fichier pdf comme suit (un graphique par page) :
  ```
  library(ggpubr)
  ggexport(plotlist = list(scree.plot, biplot.ca), filename = "CA.pdf")
  ```
  Plus d'options au Chapitre Analyse en composantes principales (section : exportation des résultats).

- *Exporter les résultats vers les fichiers txt/csv.* Fonction `R` facile à utiliser : `write.infile()` [package `FactoMineR`] :
  ```
  # Exporter vers un fichier TXT
  write.infile (res.ca, "ca.txt", sep = "")
  # Exporter vers un fichier CSV
  write.infile (res.ca, "ca.csv", sep = " ;")
  ```

4.1.7 Résumé

En conclusion, nous avons décrit comment réaliser et interpréter l'analyse factorielle des correspondances (AFC) dans `R`. Nous avons calculé l'AFC en utilisant la fonction `CA()` [`FactoMineR`]. Ensuite, nous avons utilisé le package `factoextra` `R` pour produire une visualisation `ggplot2` des résultats de l'AFC.

L'AFC peut être calculée en utilisant d'autres fonctions [packages] :

```
dudi.coa() [ade4]
library("ade4")
res.ca <- dudi.coa(housetasks, scannf = FALSE, nf = 5)
ca() [ca]
library(ca)
res.ca <- ca(housetasks)
corresp() [MASS]
library(MASS)
res.ca <- corresp(housetasks, nf = 3)
epCA() [ExPosition]
library("ExPosition")
res.ca <- epCA(housetasks, graph = FALSE)
```

Peu importe les fonctions que vous décidez d'utiliser, dans la liste ci-dessus, le package `factoextra` peut gérer le résultat.

```
fviz_eig(res.ca) # Scree plot
fviz_ca_biplot(res.ca) # Biplot des lignes et colonnes
```

4.1.8 Autres lectures

Pour les bases mathématiques de l'AFC, referez-vous aux cours, articles et livres suivants (en anglais) :
Exploratory Multivariate Analysis by Example Using `R` (book) (Husson, Le, and Pagès 2017).
Principal component analysis (article). (Abdi and Williams 2010). *https ://goo.gl/1Vtwq1*.

4.2 Travail personnel

Exercice 31.

1. Charger le jeu de données `USArrests` dans `R` avec la commande `load`. Afficher les données. Quelle est la classe de cet objet ?
2. Calculer avec les fonctions `princomp` et `prcomp` les composantes principales de l'ACP (les scores des 50états) et afficher avec la fonction `head` les résultats pour les 5 premiersétats.
3. La fonction `gsvd` réalise la décomposition en valeur singulières généralisée d'une matrice réelle Z de dimension $n \times p$ avec les métriques diagonales $N = diag(r)$ sur $\mathbb{R}^n$ et $M = diag(c)$ sur $\mathbb{R}^p$. Le code de cette fonction est le suivant :

```
# fonction SVD generalisee avec metriques diagonales
gsvd <- function(Z,r,c) {
#---entree--------
# Z matrice numerique de dimension (n,p) et de rang k
# r poids de la metrique des lignes
N=diag(r)
# c poids de la metrique des colonnes
M=diag(c)
#---sortie--------
# d vecteur de taille k contenant les valeurs singulieres (racines carres
des valeurs propres)
# U matrice de dimension (n,k) des vecteurs propres de de ZMZ'N
# V matrice de dimension (p,k) des vecteurs propres de de Z'NZM
#--------------
k <- qr(Z)$rank
colnames<-colnames(Z)
rownames<-rownames(Z)
Z <- as.matrix(Z)
Ztilde <- diag(sqrt(r)) %*% Z %*% diag(sqrt(c))
e <- svd(Ztilde)
U <-diag(1/sqrt(r))%*%e$u[,1 :k] # Attention : ne s'ecrit comme cela que
parceque N et M sont diagonales !
V <-diag(1/sqrt(c))%*%e$v[,1 :k]
d <- e$d[1 :k]
rownames(U) <- rownames
rownames(V) <- colnames if (length(d)>1)
colnames(U) <- colnames (V) <- paste("dim", 1 :k, sep = "")
return(list(U=U,V=V,d=d))
}
```

(a) Standardiser les données USArrests avec la fonction scale.

(b) Calculer avec la fonction gsvd les composantes principales de l'ACP et afficher avec la fonction head les résultats pour les premiersétats.

(c) Comparer avec les résultats trouvés avec les fonctions princomp et prcomp.

(d) Comparer avec les résultats trouvés avec les fonctions PCA du package FactoMineR.

Exercice 32. *Etude des liens entre des catégories socio-professionnelles et le type d'hébérgement*

Rappelons que le but de l'Analyse Factorielle des Correspondances (AFC) est de détecter des liens entre variables qualitatives, et de positionner les individus par rapport à ces liens.

On considère par exemple un ensemble de 18282 individus pour lesquels on connaît la CSP (modalités agriculteur AGRI, cadre supérieur CADR, inactif INAC, et ouvrier OUVR) et le choix de l'hébergement pour les vacances HEB (modalités camping CAMP, HOTEL, location LOCA, et résidence secondaire RESI). Le tableau des données brutes serait de la forme :

CSB/HEB	CAMP	HOTEL	LOCA	RESI
AGRI	239	155	129	0
CADR	1003	1556	1821	1521
INAC	682	1944	967	1333
OUVR	2594	1124	2176	1038

Dans cet exemple, le but de l'AFC sera de représenter les éventuels liens entre la CSP et le type d'hébergement choisi HEB. Cetteétude de lien peut se passer de l'analyse par AFC, via le calcul de la statistique du khi-deux, ou le calcul des profils-lignes et des profils-colonnes. D'autres méthodes d'étude de ce type de lien existent. Cependant, l'avantage qu'a l'AFC par rapport à ces méthodes est la hiérarchisation des différents types de lien, la représentation, s'il y a lieu, des individus par rapport à ces liens, et des représentations graphiques qui facilitent la communication de l'information.

1. La statistique du khi-deux et le test associé.
2. Donner les profils-ligne et colonne. commenter.
3. Faire une AFC.

   Remarque 4.2.1. Rappelons que l'AFC consiste à faire une ACP bien choisie du tableau des profils-ligne, ce qui estéquivalent à l'ACP du tableau des profils- colonne. Le vocabulaire employé sera donc assez voisin de celui de l'ACP. Cependant, leur calcul et usage diffèrera.

4. Vérifier que la statistique du khi-deuxégale la somme des valeurs propres multipliée par n.
5. Donner les coordonnées des modalités et faire leur représentation graphique.
6. Donner les contributions et cosinus carrés.
7. Y'a-t-il un effet Guttman ? Commenter.

Exercice 33. (*Données Smoke*)
Il s'agit d'un tableau de contingence donnant les fréquences de 4 catégories de fumeur (en colonne) pour 5 catégories de salarié (en ligne) dans une entreprise fictive. Les catégories en ligne sont :

— SM=Senior Managers,
— JM=Junior Managers,
— SE=Senior Employees,
— JE=Junior Employees,

— SC=Secretaries.

1. Charger le jeu de données `Smoke` du package `ca` dans `R` avec la commande `load`. Afficher les données.
2. `AFC` et `SVD` généralisée.
   (a) Construire la matrice F des fréquences, les vecteurs r et c des distributions marginales et la matrice Z desécarts à l'indépendance.
   (b) Calculer avec la fonction `gsvd` les matrices X et Y d des coordonnées factorielles des profil-lignes et colonnes de l'AFC.
   (c) Représenter avec la fonction `plot` les profil-lignes et les profil-colonnes sur le premier plan factoriel de l'AFC.
   (d) Quel est le pourcentage d'inertie expliquée par le premier plan factoriel de l'AFC
3. Retrouver ces résultats avec le package `FactoMineR` et la fonction `CA`.

Exercice 34. (*Données textuelles*)
Il s'agit ici de proposer une méthodologie d'analyse textuelle pour identifier les auteurs de deux fragements de texte anonymes. On connaît pour chacun de ces fragments de texte la fréquence d'apparition de certaines lettres. On supposeégalement que les auteurs de ces textes appartiennent à la liste suivante d'écrivains du 17ème et 18ème siècles :
Charles Darwin, René Descartes, Thomas Hobbes, Mary Shelley et Mark Twain.
Ainsi, 3 échantillons de 1000 caractères de textes de ces auteurs ontété examinés. La fréquence d'apparition de 16 lettres pour chacun de ces 15échantillons est donnée dans un tableau de contingence.

1. Récupérez les données et charger le jeu de données dans `R` avec la commande `read.csv`. Afficher les données.
2. On considère dans un premier temps le tableau de contingence des 15échantillons dont on connaît les auteurs. Effectuer un test du χ^2 d'indépendence pour répondre à la question : les distributions des lettres sont-elles significativement différentes d'unéchantillon à l'autre ? Vous pouvez utiliser la fonction `chisq.test`
3. Effectuer une AFC avec la fonction la fonction `CA` de `FactoMineR` et interpréter les résultats.
4. Effectuer une AFC avec la fonction `CA` de `FactoMineR` en ajoutant les deux textes inconnus en lignes supplémentaires.
5. Faire avec la fonction `hclust` une classification ascendante hiérarchique de Ward des 17échantillons décrits par leurs coordonnées factorielles sur les 4 premières dimensions de l'AFC. Quelle est la partition en 4 classes ?

Bibliographie

[1] Abdi, Hervé, and Lynne J. Williams. 2010. "Principal Component Analysis." John Wiley and Sons, Inc. WIREs Comp Stat 2 : 43359. *http ://staff.ustc.edu.cn/žwp/teach/MVA/abdi-awPCA2010.pdf.*

[2] Bendixen, Mike. 2003. "A Practical Guide to the Use of Correspondence Analysis in Marketing Research." Marketing Bulletin 14. *http ://marketing-bulletin.massey.ac.nz/V14/MB_V14_T2_Bendixen.pdf.*

[3] Bendixen, Mike T. 1995. "Compositional Perceptual Mapping Using Chisquared Trees Analysis and Correspondence Analysis." Journal of Marketing Management 11 (6) : 57181. doi :10.1080/0267257X.1995.9964368.

[4] Gabriel, K. Ruben, and Charles L. Odoroff. 1990. "Biplots in Biomedical Research." Statistics in Medicine 9 (5). Wiley Subscription Services, Inc., A Wiley Company : 46985. doi :10.1002/sim.4780090502.

[5] Greenacre, Michael. 2013. "Contribution Biplots." Journal of Computational and Graphical Statistics 22 (1) : 10722. *http ://dx.doi.org/10.1080/10618600.2012.702494.*

[6] Husson, Francois, Sebastien Le, and Jérôme Pagès. 2017. Exploratory Multivariate Analysis by Example Using `R`. 2nd ed. Boca Raton, Florida : Chapman ; Hall/CRC. *http ://factominer.free.fr/bookV2/index.html.*

[7] Nenadic, O., and M. Greenacre. 2007. "Correspondence Analysis in `R`, with Two- and Three-Dimensional Graphics : The ca Package." Journal of Statistical Software 20 (3) : 1-13. http ://www.jstatsoft.org.