Université de Poitiers

Projet de Analyse de Données N^o2

Analyse de données écologiques et démographiques

Rapport

Diversité florale à Bornéo et répartition des médecins libéraux en France

Élève :
Soufiane Lmezouari

Enseignant :
Dr. Poinas Arnaud

10 janvier 2025

TABLE DES MATIÈRES

CHAPITRE 1

PROBLÈME 1 : ÉTUDE DE LA DIVERSITÉ FLORALE À BORNÉO

L'objectif de cet exercice est d'analyser la diversité florale dans le nord-ouest de Bornéo en se basant sur les données issues de l'article "Habitat Patterns in Tropical Rain Forests : A Comparison of 105 Plots in Northwest Borneo". Ces données regroupent des observations de différentes espèces d'arbres collectées dans 105 terrains distincts, chacun étant caractérisé par des conditions géographiques et écologiques spécifiques.

1.1 I) Chargement et premier nettoyage des données

1.1.1 Question 1 : Charger les données

Nous avons commencer par charger le fichier **Trees.txt**, qui contient notre jeu de données. Ce fichier sera importé sur `R` à l'aide de la commande `read.table`, avec les paramètres appropriés :

```
dat = read.table("trees.txt",sep=",", header=TRUE,
   stringsAsFactors = FALSE,)
```

1.1.2 Question 2 : Nombre d'individus et de variables et leur nombre de modalités dans le jeu de données.

Nombre d'individus et de variables

Pour cela, nous avons utiliser la commande suivante :

```
dim(dat)
```

Ainsi, le jeu de données contient 2 variables et 47813 individus .

Nombre de modalités

Pour avoir le nombre de modalités des variables nous avons fait :

```
nombre_modalites_1= length(unique(dat$Espèce))
nombre_modalites_2=length(unique(dat$Terrain))
print(paste("Modalité de la variable Terrain :", nombre_
   modalites_2))
print(paste("Modalité de la variable Espece :", nombre_
   modalites_1))
```

Nous avons donc trouver :
Modalité de la variable Terrain : 128
Modalité de la variable Espece : 1342
Mais en tenant compte du fait que Les terrains sont nommés sous la forme "Lettre-Chiffre" on remarque qu'il y'a 2 modalités qui ne sont pas sous cette forme, ce sont les modalités : "11" et "ENA". On peut donc dire qu'il y'a 126 modalités de la variable terrain sous forme de "Lettre-Chiffre".

1.1.3 Question 3 : Variable tab contenant la table de contingence du nombre d'observations de chaque espèce d'arbre dans chaque terrain

```
tab =table(dat$Terrain, dat$Espèce)
```

1.1.4 Question 4 : Nombre d'observations d'arbres pour chaque terrain

Nous avons utilisé la fonction rowsums pour sommer les lignes

```
observations_par_terrain=rowSums(tab)
```

1.1.5 Question 5 : Retrait des terrains avec moins de 10 observations et vérification

Nous avons retiré les terrains avec moins de 10 observations et verifié la dimension de notre jeu de données avec :

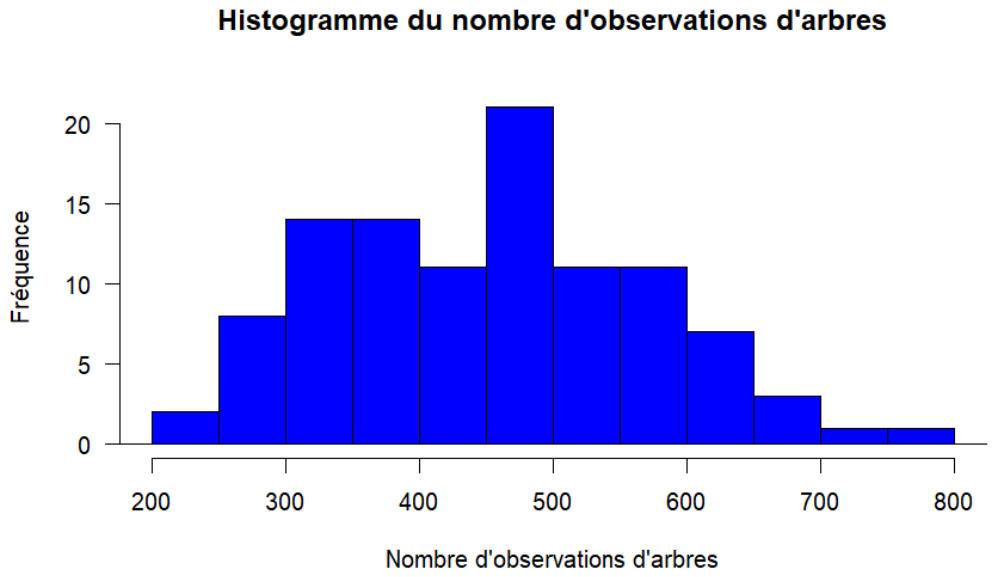
```
tab=tab[observations_par_terrain >= 10, ]
dim(tab)
```

1.1.6 Question 6 : Histogramme du nombre d'observations d'arbres pour ces 105 terrains et comparaison

Nous avons utilisé ces paramètres pour une meilleure comparaison de l'histogramme avec celui de la figure 5a de l'article.

```
hist(observations_par_terrain,
breaks = 13,
main = "Histogramme du nombre d'observations d'arbres",
xlab = "Nombre d'observations d'arbres",
ylab = "Fréquence",
col = "blue",
border = "black",
las = 1,
xlim=c(200,800))
```

L'histogramme trouvé est exactement le même que celui de l'article.



1.2 II) Deuxième nettoyage des données

1.2.1 Question 1 : Retrait du tab les observations d'arbres qui n'ont pas été identifiés

```
tab = tab[, colnames(tab) != " "]
```

1.2.2 Question 2 : Retrait du tab les observations d'arbres qui ont été mal identifiés

Nous avons utilisé la commande strdetect de la librairie stringr et complété de la manière suivante la commande :

```
library(stringr)
tab = tab[, str_detect(colnames(tab), "\\?", negate = TRUE)
   ]
```

1.2.3 Question 3 : Créeation d'une matrice vide M avec 105 lignes et 10 colonnes

```
M = matrix(NA, nrow = 105, ncol = 10)
```

1.2.4 Question 4 : Stock dans la première ligne de la matrice M des 10 espèces d'arbres les plus abondantes dans le terrain A1

```
especes_A1 = sort(tab["A1", ], decreasing = TRUE)
head(especes_A1)
M[1, ] = names(especes_A1)[1:10]
```

Résultat des 10 espèces d'arbres les plus abondantes dans le terrain A1

Espèce	Abondance
Teijsmanniodendron simplicifolium	12
Eugenia (illegible)	9
Madhuca sandakanensis	9
Mallotus leptophyllus	9
Eugenia valdevenosa	8
Parishia polycarpa	8
Eugenia attenuata	7
Quercus subsesicea	7
Adinandra acuminata	6
Horsfieldia fragilluna	6

1.2.5 Question 5 : Remplissage complet de la matrice M de sorte que la i-ème ligne contienne le nom des 10 espèces d'arbres les plus abondantes dans le i-ème terrain

Pour cela,nous avons fais une boucle pour parcourir toutes les lignes de la matrice M.

```
for (i in 1:105) {
    terrain = rownames(tab)[i]
    especes_terrain = sort(tab[terrain, ], decreasing =
       TRUE)
    M[i, ] = names(especes_terrain)[1:10]
}
```

1.2.6 Question 6 : Le nom des 4 espèces qui apparaissent le plus souvent dans la liste des 10 espèces les plus présentes sur chaque terrain ainsi que le nombre de terrains correspondant

Pour cela, nous avons utilisé les fonctions table et sort

```
Especes_10= as.vector(M)
c = sort(table(Especes_10), decreasing = TRUE)
result <- paste(names(c), c, sep = ": ")
```

Comparaison avec la table 2 de l'article

- Pour l'espèce *Shorea macroptera*, nous avons relevé 28 terrains contre 25 mentionnés dans l'article.
- Pour l'espèce *Koilodepas longifolium*, nous avons relevé 23 terrains contre 24 dans l'article.
- Pour l'espèce *Allantospermum borneense*, nous avons le même nombre de terrains que celui indiqué dans la table 2 de l'article, à savoir 22.
- Pour l'espèce *Vatica micrantha*, nous avons relevé 21 terrains contre 22 mentionnés dans la table 2 de l'article.

1.2.7 Question 7 : Le nom des terrains pour lesquels l'espèce "allantospermum borneense" est parmi les 10 plus présentes.

Nous avons fais cela à l'aide des fonctions which et rownames

```
terrains_allantospermum = which(apply(M, 1, function(x) "
   allantospermum borneense" %in% x))
noms_terrains = rownames(tab)[terrains_allantospermum]
print(noms_terrains)
```

Nous avons obtenu ce résultat qui correspond bien au résultat donné dans la table n°2 de l'article.

```
"G1"  "G14" "G2"  "G5"  "G6"  "G7"  "G8"  "G9"  "K1"
"K2"  "K3"  "L1"  "L3"  "L5"  "M1"  "M2"  "M3"  "M4"
"M5"  "N1"  "N3"  "N4"
```

1.3 III) Clustering hiérarchique des données

1.3.1 Question 1 : Calcul de la dissimilarité de Bray-Curtis entre les deux premières lignes du tableau tab

```
        x=tab[1, ]
        y=tab[2, ]
        a=sum(pmin(x, y))
        b=sum(x + y)
        Bray_curtis=1 - 2 * (a/b)
        print(Bray_curtis)
```

les Résultats otenu est : 0.7688679

1.3.2 Question 2 : Matrice D des dissimilarités de Bray-Curtis entre les 105 terrains et verification de D[1]

Pour cela, nous avons utilisé la fonction bcdist et on a fait un print de D[1] et on a trouvé que le resultat correspond bien à la dissimilarité de Bray-Curtis entre les deux premières lignes du tableau tab

```
        library(ecodist)
        D = bcdist(tab)
        print( D[1])
```

les Résultats otenu est : 0.7688679

1.3.3 Question 3 : la dissimilarité entre les classes utilisée par les auteurs

Les auteurs ont utilisé la dissimilarité moyenne entre les classes

Effectuer le clustering hiérarchique sur les 105 terrains avec cette dissimilarité entre les classes et la dissimilarité de Bray-Curtis entre les terrains. Stocker le résultat dans une variable res

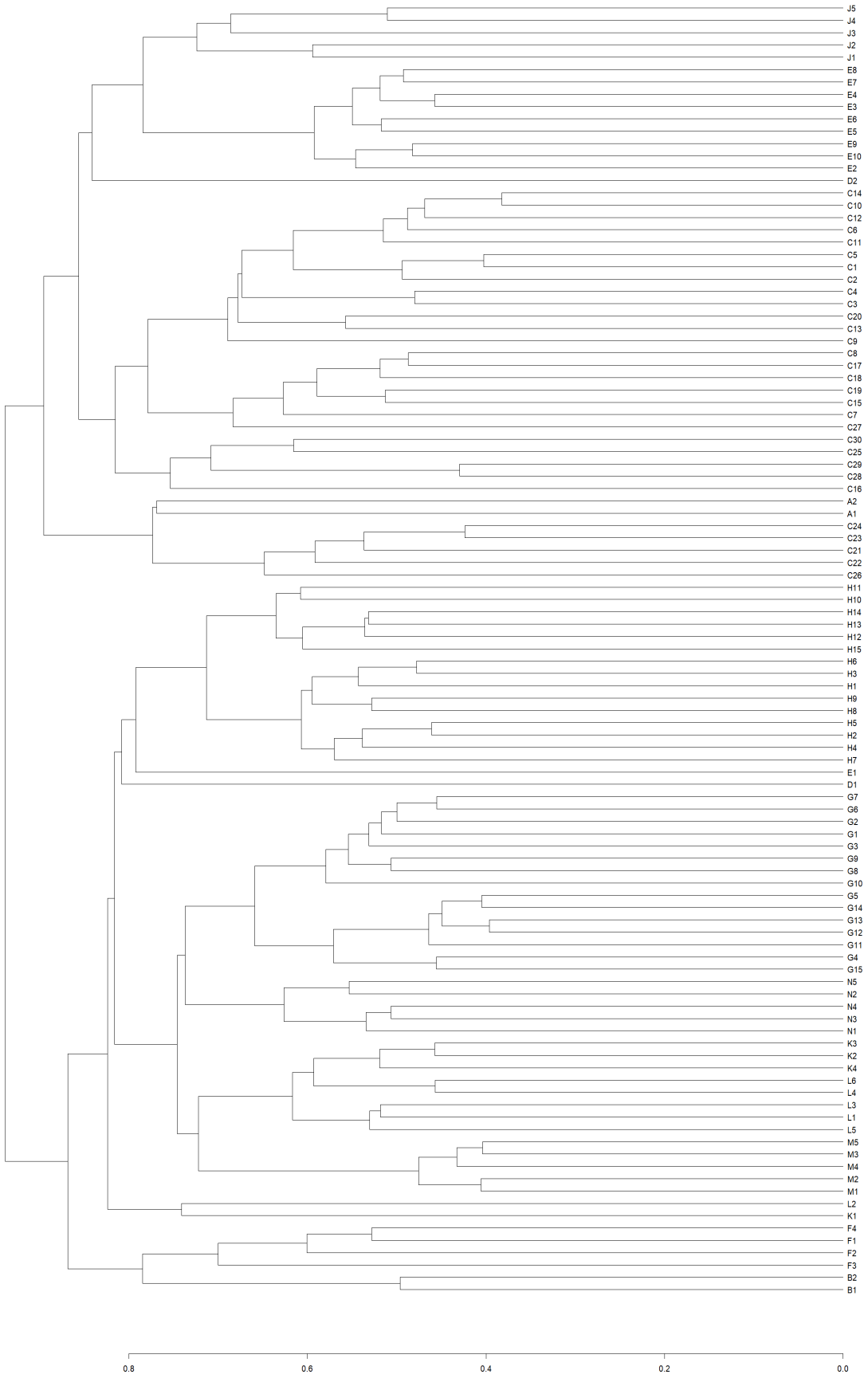
```
        res=hclust(as.dist(D), method = "average")
```

1.3.4 Question 4 : dendrogramme tourné à l'horizontal

Nous avons modifié les dimensions du graphe pour que les noms des individus soient lisibles.

```
        {r,fig.width=20,fig.height=30,fig.align="center"}
        plot(as.dendrogram(res), horiz = TRUE, main = "Dendrogramme
            horizontal",ylim=c(0,length(labels(as.dendrogram(res)))
           ),cex.main=2,cex.lab=1.5)
```

Dendrogramme horizontal



1.3.5 Question 5 : Fusion de la classe H1,H2,H3,H4,H5,H6,H7,H8,H9 et la dissimilarité entre ces deux classes, et comparaison avec la figure n°6 de l'article

La classe **{H1, H2, H3, H4, H5, H6, H7, H8, H9}** a été fusionnée avec la classe **{H10, H11, H12, H13, H14, H15}** et la dissimilarité entre ces deux classes est de : **0,72**.

Sur la **figure 6** de l'article, on observe bien une fusion entre ces deux classes et leur dissimilarité est approximativement la même que ce qu'on a trouvé.

1.3.6 Question 6 : Fusion des classes M1,M2,M3,M4,M5 et K1 et leur dissimilarité, et comparaison avec la figure n°6 de l'article

• La classe **{M1, M2, M3, M4, M5}** a été fusionnée avec la classe **{K2, K3, K4, L1, L3, L4, L5, L6}** et la dissimilarité entre ces deux classes est de **0,75**.

Sur la **figure 6** de l'article, on observe bien une fusion entre ces deux classes et leur dissimilarité est approximativement la même que ce qu'on a trouvé.

• La classe **{K1}** a été fusionnée avec la classe **{L2}** et la dissimilarité entre ces deux classes est de **0,77**.

Sur la **figure 6** de l'article, cette fois-ci il y a une différence entre les résultats. Les auteurs ont fusionné la classe **{K1}** avec la classe **{N1, N2, N3, N4, N5}** avec une dissimilarité approximative de **0,78**.

CHAPITRE 2

PROBLÈME 2 : ÉTUDE DE LA RÉPARTITION DES MÉDECINS LIBÉRAUX EN FRANCE

L'objectif de cet exercice est s'inscrit dans le cadre d'une analyse approfondie visant à étudier la répartition des médecins libéraux en France, en s'appuyant sur le jeu de données "Medecin liberaux 2023". L'objectif principal est de comprendre comment les différentes professions médicales sont distribuées entre les régions et départements français. Cette analyse repose sur plusieurs étapes, allant de la construction de la matrice des données à l'application de méthodes statistiques avancées, telles que l'Analyse en Composantes Principales (ACP), les k-means et le clustering hiérarchique.

2.1 I) Création du jeu de données

2.1.1 Question 1 : Charger les données

Nous commencerons par charger le fichier nommé **Medecin_liberaux_2023**, qui contient notre jeu de données. Ce fichier sera importé dans `R` à l'aide de la commande `read.table`, avec les paramètres appropriés :

```
Medecin_liberaux <- read.table("Medecin_liberaux_2023.txt",
   header = TRUE, sep = ",", dec = ".", stringsAsFactors =
   FALSE, encoding = "UTF-8")
```

2.1.2 Question 2 : Le nombre de professions, régions et départements distincts dans le jeu de données.

Pour cela, nous utilisons les commandes suivantes :

```
nb_professions <- length(unique(Medecin_liberaux$profession_
    sante))
nb_regions <- length(unique(Medecin_liberaux$libelle_region))
nb_departements <- length(unique(Medecin_liberaux$libelle_
    departement))

cat(nb_professions, nb_regions, nb_departements)
```

Ainsi, le jeu de données contient 31 professions, 18 régions et 102 départements distincts.

2.1.3 Question 3 : Identifier et compter les pédiatres ayant exercé en Nouvelle-Aquitaine en 2023

Pour identifier et compter les pédiatres ayant exercé en Nouvelle-Aquitaine en 2023, nous appliquons un filtre sur les données en sélectionnant uniquement les lignes correspondant à la profession `Pédiatres` et à la région `Nouvelle-Aquitaine`. Ensuite, nous calculons le total des effectifs à l'aide de la commande `sum()`.
Le code utilisé est le suivant :

```
pediatres_NA <- Medecin_liberaux[Medecin_liberaux$profession_
    sante == "Pédiatres" &
Medecin_liberaux$libelle_region == "Nouvelle-Aquitaine", ]
total_pediatres_NA <- sum(pediatres_NA$effectif)
cat(total_pediatres_NA)
```

Ainsi, en 2023, un total de 430 pédiatres a exercé dans la région Nouvelle-Aquitaine.

2.1.4 Question 4 : Répartition des professions médicales par région

Pour analyser la distribution des professions par région, nous avons créé une matrice vide où les lignes correspondent aux régions et les colonnes aux professions. Cette matrice sera remplie ultérieurement avec les effectifs des différentes professions dans chaque région.
Le code utilisé pour créer cette matrice est le suivant :

```
regions <- unique(Medecin_liberaux$libelle_region)
professions <- unique(Medecin_liberaux$profession_sante)
M <- matrix(0, nrow = length(regions), ncol = length(
    professions))
rownames(M) <- regions
colnames(M) <- professions
```

Cette matrice vide, initialisée avec des zéros, contient :

— Lignes (Régions) : Les noms des régions uniques extraites du jeu de données.

— Colonnes (Professions) : Les noms des professions uniques extraites du jeu de données.

2.1.5 Question 5 : Complétion de la matrice des médecins par profession et région

Pour compléter la matrice `M`, nous utilisons une double boucle qui parcourt les régions et les professions. Chaque cellule de la matrice est remplie avec le total des effectifs des médecins correspondant à la région et à la profession correspondantes.

Le code utilisé est le suivant :

```
for (region in regions) {
    for (profession in professions) {
        M[region, profession] <- sum(Medecin_liberaux$effectif[
           Medecin_liberaux$libelle_region == region &
        Medecin_liberaux$profession_sante == profession])
    }
}
```

Une fois la matrice remplie, nous vérifions que la commande suivante renvoie le résultat attendu de la question précédente :

```
M["Nouvelle-Aquitaine", "Pédiatres"]
```

Le résultat est : **430**, confirmant que la matrice a été correctement remplie.

Cette matrice complète permet de visualiser le nombre de médecins de chaque profession pour chaque région, offrant ainsi une vue d'ensemble des effectifs.

2.1.6 Question 6 : Chargement des données de population régionale au 1er janvier 2023.

2.1.7 Chargement des données de population par région

Nous avons utilisé le fichier `Pop_Reg_2023.txt` pour charger les données contenant la population de chaque région de France au 1er janvier 2023. Ces données sont ensuite renommées pour faciliter leur manipulation.

Le code utilisé est le suivant :

```
Pop_Reg <- read.table("Pop_Reg_2023.txt", header = FALSE, sep =
    ";", stringsAsFactors = FALSE)
# Renommez les colonnes de Pop_Reg
colnames(Pop_Reg) <- c("region", "population")
head(Pop_Reg)
```

Les premières lignes des données chargées sont :

region	population
Auvergne-Rhône-Alpes	8,197,325
Bourgogne-Franche-Comté	2,786,296
Bretagne	3,429,882
Centre-Val de Loire	2,572,278
Corse	351,255
Grand Est	5,562,262

Ces données permettent d'analyser la répartition de la population par région en France au 1er janvier 2023.

2.1.8 Question 7 : Ajout des ratios médecins par 10 000 habitants dans la matrice M et vérification des résultats.

Avant la mise à jour, nous avons vérifié que toutes les régions présentes dans la matrice `M` se retrouvent dans les données de population `Pop_Reg`.

Le code utilisé est le suivant :

```
missing_regions <- setdiff(regions, Pop_Reg$region)
if (length(missing_regions) > 0) {
    stop(paste("Les régions suivantes sont manquantes dans Pop_
       Reg :", paste(missing_regions, collapse = ", ")))
}
```

La mise à jour de `M` pour contenir les effectifs par 10 000 habitants a été réalisée comme suit :

```
for (region in regions) {
    population <- Pop_Reg$population[Pop_Reg$region == region]
    M[region, ] <- (M[region, ] / population) * 10000
}
```

Vérification pour la Nouvelle-Aquitaine Pour la profession `Pédiatres` dans la région `Nouvelle-Aquitaine`, nous avons vérifié que la valeur dans `M["Nouvelle-Aquitaine", "Pédiatres"]` correspond au résultat attendu. Le code utilisé est le suivant :

```
population_NA <- Pop_Reg$population[Pop_Reg$region == "Nouvelle
   -Aquitaine"]

Résultat_attendu <- (total_pediatres_NA / population_NA) *
   10000
Résultat_obtenu <- M["Nouvelle-Aquitaine", "Pédiatres"]

if (isTRUE(all.equal(Résultat_attendu, Résultat_obtenu))) {
    print("La vérification est correcte : les valeurs
       correspondent.")
```

Les résultats de la vérification confirment que la matrice `M` a été correctement ajustée. Pour la Nouvelle-Aquitaine, la valeur obtenue correspond au résultat attendu.

2.2 Visualisation et partitionnement des données.

Pour accorder la même importance à chaque profession, il est nécessaire de centrer et réduire les données. Cela permet d'uniformiser les échelles des différentes professions en normalisant leurs valeurs.

2.2.1 Question 1 : Transformation et centrage.

La matrice `M` est transformée en un jeu de données (`data.frame`) et centrée-réduite grâce à la commande `scale()`.

Le code utilisé est le suivant :

```
M_scaled <- as.data.frame(scale(M))
```

2.2.2 question 2 : Analyse des covariance entre professions médicales

Nous avons calculé la matrice de corrélation des données normalisées, puis visualisé ces corrélations à l'aide de la fonction `corrplot`.

Pour simplifier l'affichage, les noms des professions ont été abrégés.

```
library(corrplot)
Calculer la matrice de corrélation
cov_matrix <- cov(M_scaled, use = "pairwise.complete.obs")

rownames(cov_matrix) <- abbreviate(rownames(cov_matrix),
   minlength = 10)
colnames(cov_matrix) <- abbreviate(colnames(cov_matrix),
   minlength = 10)
```

TABLE 2.1 – Liste des professions médicales et leurs abréviations

Nom complet	Abréviation	Nom complet	Abréviation
Allergologues	Allergolgs	Néphrologues	Néphrologs
Anesthésistes-réanimateurs	Ansthssts-	Ophtalmologues	Ophtalmlgs
Autres médecins	Autrsmdcns	Orthophonistes	Orthphnsts
Cardiologues	Cardiologs	Orthoptistes	Orthoptsts
Chirurgiens	Chirurgins	Oto-rhino-laryngologistes	Ot-rhn-lry
Dermatologues	Dermatolgs	Pneumologues	Pneumologs
Endocrinologues	Endocrnlgs	Psychiatres	Psychiatrs
Chirurgiens-dentistes	Edchrrgns-	Pédiatres	Pédiatres
Médecins généralistes	Ensmbldsmg	Podologues	Pdcrs-pdlg
Gynécologues	Gynclgsmeo	Radiologues	Radiologus
Hépato-gastro-entérologues	Hpt-gstr-n	Radiothérapeutes	Radithrpts
Infirmiers	Infirmiers	Rhumatologues	Rhumatolgs
Kinésithérapeutes	Mssrs-knst	Sages-femmes	Sages-fmms
Médecins nucléaires	Mdcnsnclrs	Stomatologues	Stomatolgs
Pathologistes	Mdcnspthlg	Médecins vasculaires	Mdcnsvsclr
Neurologues	Neurologus		

Visualisation des corrélations avec `corrplot`

```
# Visualiser la matrice de corrélation
corrplot(cov_matrix, method = "circle")
```

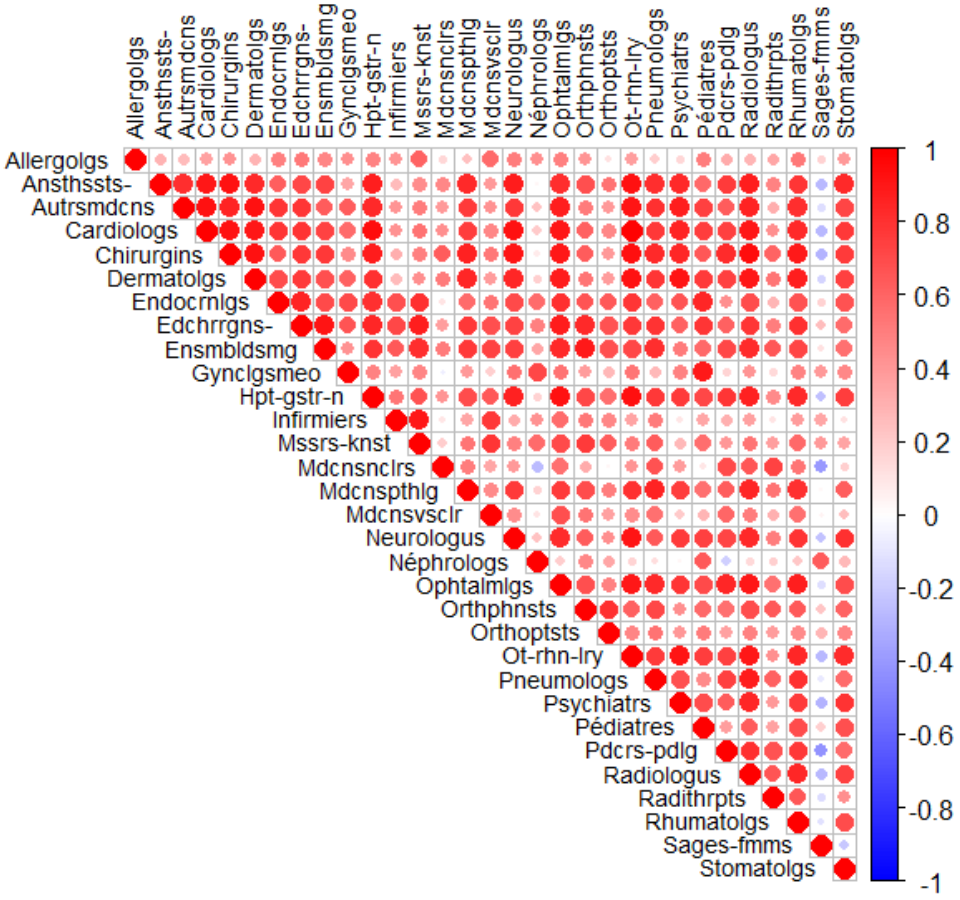


FIGURE 2.1 – Matrice de corrélation

1- Les médecins généralistes sont souvent fortement corrélés positivement avec des spécialités complémentaires comme les infirmiers, les radiologues ou les pédiatres. Cela peut être dû à des interactions ou des collaborations fréquentes entre ces spécialités Ces professions travaillent ensemble pour fournir des soins intégrés.

la même chose pour pour Les spécialités chirurgicales (comme les chirurgiens, anesthésistes, et gynécologues-obstétriciens) montrent également des corrélations élevées.

2- Certaines professions, comme les stomatologues ou les allergologues, montrent peu de corrélations avec d'autres spécialités, reflétant leur rôle très spécialisé et leur interaction limitée avec d'autres domaines.

En résumé Les professions médicales présentent de fortes corrélations lorsqu'elles collaborent ou remplissent des fonctions complémentaires, tandis que les spécialisations plus autonomes affichent des corrélations plus faibles.

2.2.3 Question 3 :Choix du nombre K pour les k-means avec la méthode des silhouettes.

Pour déterminer le nombre optimal de classes K pour le partitionnement des données avec l'algorithme des k-means, nous avons utilisé la méthode des silhouettes.

Le nombre optimal de clusters K est celui qui maximise la valeur moyenne des silhouettes.

```r
# Détermination du nombre optimal de clusters avec la méthode des silhouettes
library(factoextra)

fviz_nbclust(M_scaled, kmeans, method = 'silhouette', nstart = 10)
```

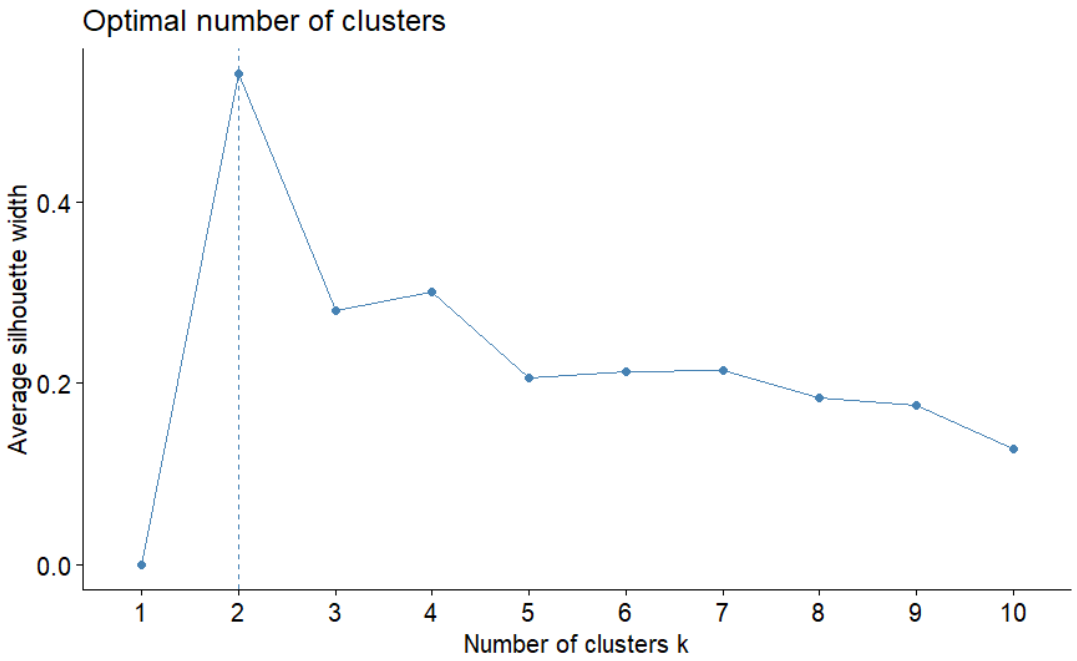


FIGURE 2.2 – Analyse des silhouettes pour le choix optimal de K

le pic maximal observé : $K = 2$, cela signifie que la partition des données en 2 clusters est la plus cohérente.

2.2.4 Question 4 : Classification k-means et répartition des régions par classe.

Nous avons utilisé $K = 2$ classes pour cette analyse. Le calcul des clusters a été effectué avec le code suivant :

```r
# Calcul des distances entre observations
D <- dist(M_scaled)

# Classification k-means avec K = 2
res <- kmeans(M_scaled, 2, nstart = 10)

# Liste des clusters
res$cluster
```

Région	Classe
Guadeloupe	2
Martinique	2
La Réunion	2
Mayotte	1
Île-de-France	2
Centre-Val de Loire	2
Bourgogne-Franche-Comté	2
Normandie	2
Hauts-de-France	2
Grand Est	2
Pays de la Loire	2
Bretagne	2
Nouvelle-Aquitaine	2
Occitanie	2
Auvergne-Rhône-Alpes	2
Provence-Alpes-Côte d'Azur	2
Corse	2
Guyane	1

— **Cluster 1** : Inclut des régions ultramarines comme *Mayotte* et *Guyane*.

— **Cluster 2** : Comprend la majorité des régions métropolitaines, partageant des caractéristiques similaires.

2.2.5 Question 5 : Diagramme des silhouettes et évaluation de la qualité de la classification.

```
sil <- silhouette(res$cluster, D)
library(factoextra)
fviz_silhouette(sil)
```

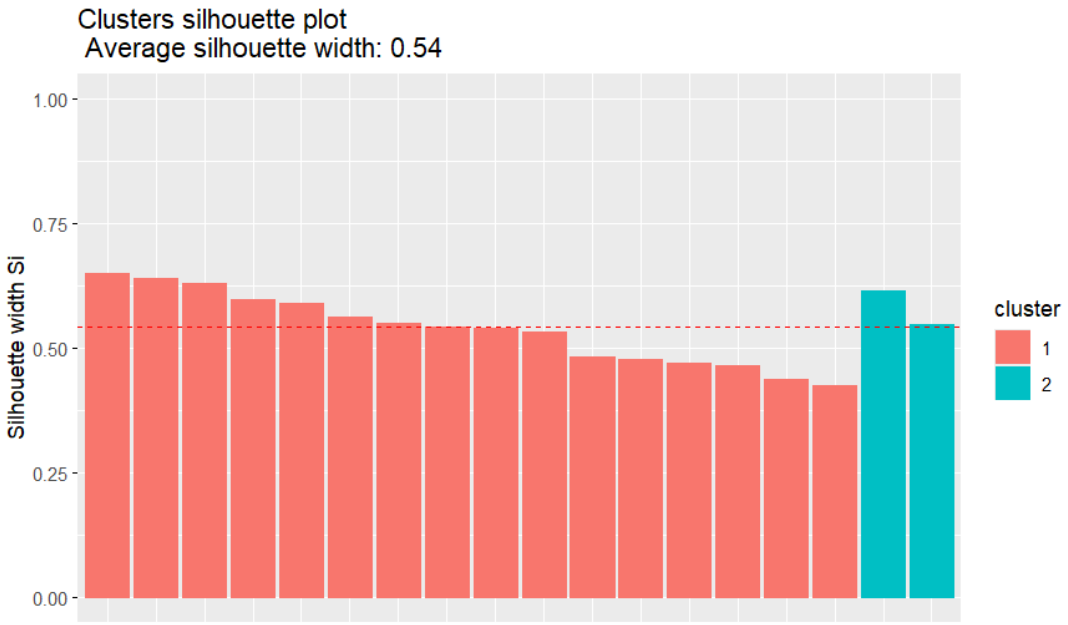


FIGURE 2.3 – Diagramme En Barres Des Silhouettes Pour K=2

Silhouette moyenne (0.54) : Cela suggère une séparation raisonnable entre les clusters.

2.2.6 Question 6 : Clustering hiérarchique : dendrogramme et comparaison avec les k-means.

L'objectif est de regrouper les régions en K classes en utilisant une approche de clustering hiérarchique. Nous utilisons la distance euclidienne entre les régions et la méthode de Ward pour minimiser l'inertie intra-classe à chaque étape de fusion.

```
res_H <- hclust(D, method = "ward.D2")
```

```
library(factoextra)
fviz_nbclust(M_scaled, hcut, method = "silhouette", hc_method =
    "ward.D2", hc_metric = "euclidean")
```

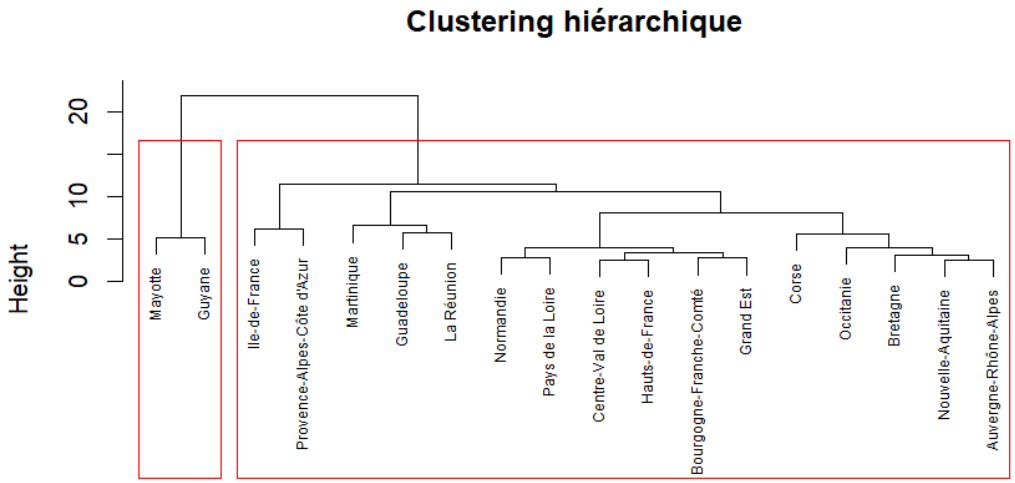


FIGURE 2.4 – Dendrogramme - Clustering Hiérarchique (Ward)

Les régions ont été classées par clustering hiérarchique en deux clusters, avec une partition identique à celle obtenue par k-means :

Cluster 1 : Mayotte et Guyane. Cluster 2 : Guadeloupe, Martinique, La Réunion, Île-de-France, et d'autres régions.

Les résultats du clustering hiérarchique et du k-means sont cohérents, les deux méthodes produisant les mêmes regroupements pour les régions.

2.2.7 Question 7 : ACP à deux dimensions : visualisation des classes k-means et interprétation.

Pour visualiser les régions dans un espace réduit tout en conservant l'essentiel de la variabilité des données, une **ACP** a été réalisée. Les individus (régions) sont représentés dans un plan à deux dimensions et colorés selon les classes obtenues par le partitionnement en k-means.

Réalisation de l'ACP

```
    fviz_pca_ind(res.pca,
    col.ind = as.factor(cutree(res_H, 2)),
    palette = c("#2E9FDF", "#E7B800"),
    addEllipses = TRUE,
    ellipse.type = "convex",
    repel = TRUE,
    pointsize = 4,
    labelsize = 3,
    alpha.ind = 0.9) +
    theme_minimal() +
    ggtitle("Projection des individus selon l'ACP et le clustering"
       ) +
    xlab(paste0("Dim 1 (", round(res.pca$eig[1, 2], 1), "%)")) +
    ylab(paste0("Dim 2 (", round(res.pca$eig[2, 2], 1), "%)"))
```

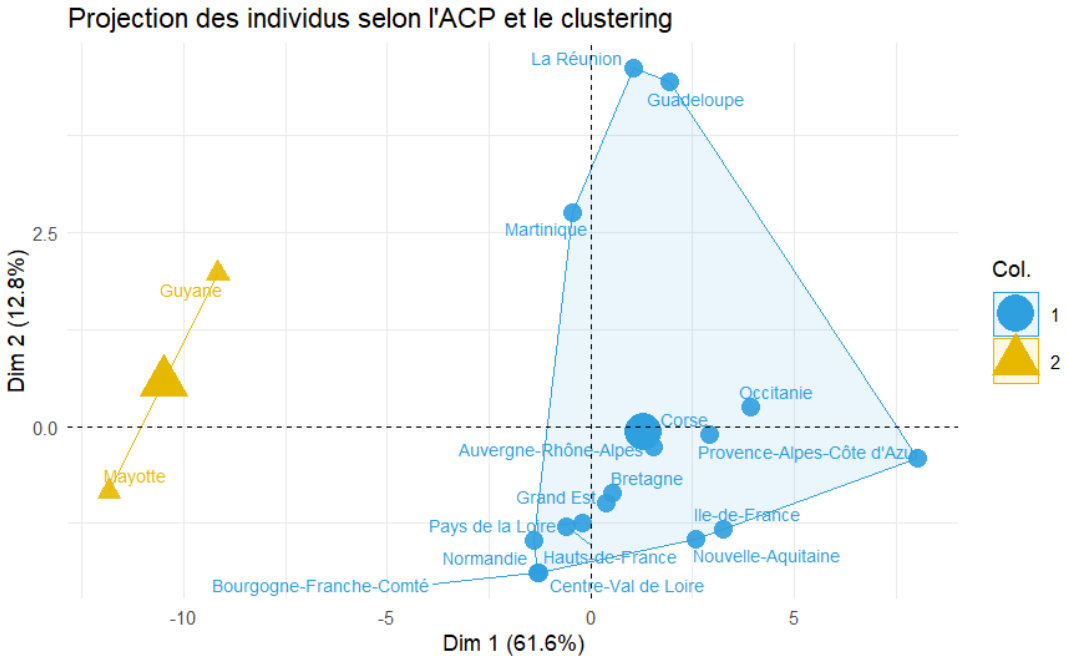


FIGURE 2.5 – Nuage Des Individus Colorié Par Les Classes (K-Means, K=2

Les deux premières dimensions principales expliquent les pourcentages suivants de l'inertie totale des données :

Première dimension (PCA1) : 61.6

Deuxième dimension (PCA2) : 12.8

▷ les deux dimensions principales capturent ensemble 74.4% de la variance totale des données

Ce graphique représente la projection des régions françaises selon une Analyse en Composantes Principales (ACP) combinée avec un clustering. Voici une interprétation des résultats :

▷ **Clustering :**

⋊ Cluster 1 (en bleu, cercle) : Comprend toutes les régions de la France métropolitaine ainsi que les départements d'outre-mer comme La Réunion, la Guadeloupe et la Martinique. Ces régions partagent des caractéristiques similaires.

⋊ Cluster 2 (en jaune, triangle) : Composé uniquement de la Guyane et de Mayotte. Situées à gauche et séparées du reste des régions, Il semble qu'elles présentent des profils non typiques.

▷ **La Réunion, Guadeloupe et Martinique :** Elles se situent légèrement éloignées des régions métropolitaines, ce qui pourrait refléter des particularités liées à leur situation géographique ou leurs infrastructures.

▷ **Ensemble des régions métropolitaines :** Elles se trouvent regroupées près de l'origine des axes dans le Cluster 1, indiquant des caractéristiques relativement homogènes entre elles.

Le clustering et l'ACP montrent une segmentation nette entre les régions métropolitaines et Guyane, Mayotte.

2.3 Étude des professions médicales séparées par département

2.3.1 Question 1 : Création d'un jeu de données des médecins par profession et département pour 10 000 habitants.

En s'inspirant de ce qui a été fait dans les parties précédentes, on a créé un nouveau matrice M_dept contienne le nombre de médecins de chaque profession et chaque departement.

La suppression de "Tout département" :

```
M_dept <- M_dept[rownames(M_dept) != "Tout département", ]
departments <- departments[departments != "Tout département"]
```

nous avons également exclu "Tout département", car ce département n'existe pas dans les données Pop_Dep_2023 fournies.

Calcul du nombre de médecins pour 10 000 habitants

```
for (dept in departments) {
    population <- Pop_Dep$population[Pop_Dep$departement ==
        dept]
    M_dept[dept, ] <- (M_dept[dept, ] / population) * 10000
}
```

2.3.2 Question 2 : ACP identification des disparités entre départements."

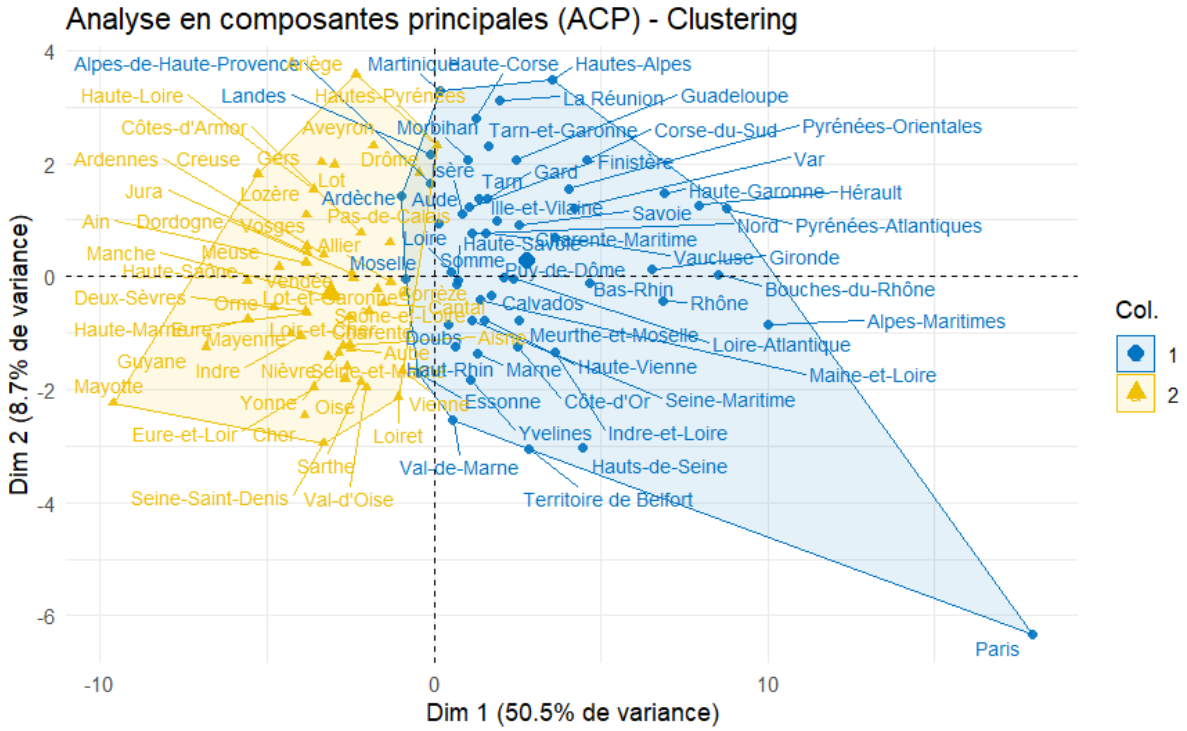


FIGURE 2.6 – Analyse en composantes principales (ACP) - Clustering

Variance expliquée par les dimensions : Dim 1 (50.5%) et Dim 2 (8.7%) les deux dimensions capturent 59.2% de l'inertie totale, ce qui est une raisonable représentation des données originales

⋊ le département de Paris (situé en bas à droite, dans le Cluster 1) se distingue nettement des autres départements en termes de répartition des professions médicales.

Il est probable que cela soit du fait de sa forte densité urbaine, de son statut de capitale et de la présence significative d'experts et d'infrastructures médicales.