Master 1 : Mathématiques & Interactions

Projet : Analyse des données

Deux applications des méthodes de l'analyse des données avec R

Auteur :
Ayoub ABRAICH

Encadrant :
Dr. Jean-Renaud PYCKE

12 décembre 2018

Table des matières

Introduction

Dans ce projet, nous allons appliquer deux méthodes d'analyse de données (classification hiérarchique & l'ACP) pour étudier 2 échantillons de données . On commence par une présentation courte des outils thérique, ensuite nous exposons notre analyse via ces deux méthodes en utilisant le langage R . Je me base principalement dans la partie théorique sur les cours de Wikistat .

Lorsqu'on étudie simultanément un nombre important de variables quantitatives, comment en faire un graphique global ? La difficulté vient de ce que les individus étudiés ne sont plus représentés dans un plan, espace de dimension 2, mais dans un espace de dimension plus importante (par exemple 4). L'objectif de l'Analyse en Composantes Principales (ACP) est de revenir à un espace de dimension réduite (par exemple 2) en déformant le moins possible la réalité . Il s'agit donc d'obtenir le résumé le plus pertinent possible des données initiales.

C'est la matrice des variances-covariances (ou celle des corrélations) qui va permettre de réaliser ce résumé pertinent, parce qu'on analyse essentiellement la dispersion des données considérées. De cette matrice, on va extraire, par un procédé mathématique adéquat, les facteurs que l'on recherche, en petit nombre. Ils vont permettre de réaliser les graphiques désirés dans cet espace de petite dimension (le nombre de facteurs retenus), en déformant le moins possible la configuration globale des individus selon l'ensemble des variables initiales (ainsi remplacées par les facteurs).

C'est l'interprétation de ces graphiques qui permettra de comprendre la structure des données analysées. Cette interprétation sera guidée par un certain nombre d'indicateurs numériques et graphiques. L'analyse en Composantes Principales (ACP) est un grand classique de l"analyse des données" en France pour l'étude exploratoire ou la compression d'un grand tableau $n \cdot p$ de données quantitatives.

Concernant la classification non supervisée ou " Clustering " , c'est une méthode de classification qui permet la recherche d'une typologie, ou segmentation, c'est-à-dire d'une partition, ou répartition des individus en classes homogènes, ou catégories. Ceci est fait en optimisant un critère visant à regrouper les individus dans des classes, chacune le plus homogène possible et, entre elles, les plus distinctes possible. Cet objectif est à distinguer des procédures de discrimination,ou encore de classement pour lesquelles une typologie est

a priori connue, au moins pour un échantillon d'apprentissage. Il existe de très nombreuses méthodes de classification non supervisées (Partitioning methods , Hierarchical clustering,Fuzzy clustering,Density-based clustering & Model-based clustering ..) . [WikiStat]

Chapitre 1

Rudiments

1.1 Analyse en Composantes Principales (ACP)

1.1.1 Principe général

D'un point de vue plus "mathématique", l'ACP correspond à l'approximation d'une matrice (n, p) par une matrice de même dimensions mais de rang $q < p$; q étant souvent de petite valeur 2, 3 pour la construction de graphiques facilement compréhensibles.

Soit X un vecteur aléatoire qui admet une varience . Si seulement nous gardons quelques-uns des composantes principales, nous obtenons alors une «simple» explication de la structure de X impliquant un nombre petit de variables aléatoires.
Comme $Var(X)$ est symétrique positive , alors elle s'écrit comme : $Var(X) = ODO^T$ avec O matrice orthogonale : celle des vecteurs propres et D la matrice diagonale des valeurs propores dans le meme ordre.
On considère le vecteur $Y := O^T D$, on trouve que : $Var(Y) = D$, donc les composantes de Y ne sont pas corrélées et en plus les variences des ces composantes sont égales aux valeurs propres de $Var(X)$.
Les composantes de Y sont appelées les principales composantes de X . Puisque une matrice orthogonale est inversible, nous avons aussi $X = OY$. Cela exprime un vecteur aléatoire X arbitraire en tant que combinaison linéaire de variables aléatoires non corrélées (ses composantes principales).
Donc , tout simplement le processus de décomposition spectrale (ie : Trouver des valeurs propres et vecteurs propres) de la matrice de variance de X est appelée ACP !
On note o_j les colonnes de O (= vecteurs propres de $Var(X)$) et λ_j les valeurs propres associées ordonnées dans le sens décroissant et $Y = (y_1, \cdots, y_n)$. On a :

$$X = \sum_{j=1}^{n} y_j \cdot o_j$$

Donc , une somme $\widetilde{X}$ plus petite peut expliquer suffisenemt la majorité de la variance :

$$\widetilde{X} = \sum_{j=1}^{k} y_j \cdot o_j$$

On s'intéresse maintenant à la fraction de contrbution des k premières composantes principales :
On a :

$$\|Y\|^2 := Y^T Y = \sum_{j}^{n} y_j^2$$

et

$$E\left(\|Y - E(Y)\|^2\right) = \sum_{j=1}^{n} \lambda_j$$

On obtient facilement aussi

$$E\left(\|X - E(X)\|^2\right) = \sum_{j=1}^{n} \lambda_j$$

car O est orthogonale donc une rotation qui ne change pas les longueurs . De meme :

$$E\left(\left\|\widetilde{X} - E(\widetilde{X})\right\|^2\right) = \sum_{j=1}^{k} \lambda_j$$

Ainsi, la fraction de la variance de X expliquée par les premières k composantes principales est égale à :

$$f_k := \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j}$$

appelée part d'inertie expliquée .
De façon plus générale , si $M \in M_{n,p}(\mathbb{R})$ est la matrice des données , on peut appliquer le théorème de la Décomposition en Valeurs Singulières qui dit : il existe $U \in M_{n,n}(\mathbb{R})$ et $V \in M_{p,p}(\mathbb{R})$ telles que : $M = UDV^T$ avec $D = diag(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_r}, 0, \cdots, 0)$ et $r := min(n, p)$ et $\forall i\ \lambda_i \in Sp(A^T A) \cap Sp(AA^T)$ avec U la matrice des vecteurs propres de $A^T A$ et V la matrice des vecteurs propres de AA^T. Ainsi , tout simplement l'ACP consiste à diminuer le range r à un $k < r$. D'où la formule de reconstitution :

$$M = \sum_{i=1}^{k} \sqrt{\lambda_i} u_i v_i^T$$

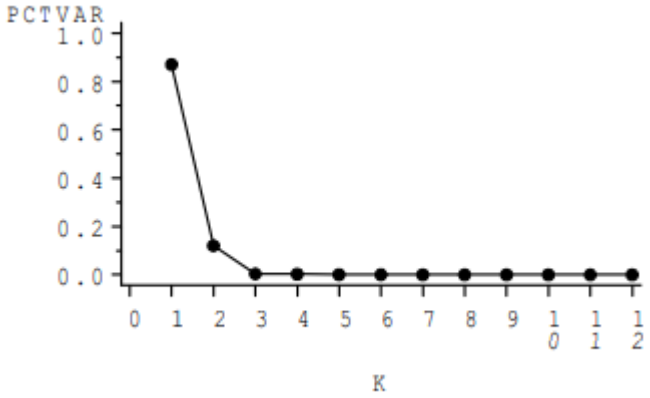


FIGURE 1.1 – Eboulis des valeurs propres

1.1.2 Choix de dimension

La qualité des estimations auxquelles conduit l'ACP dépend, de façon évidente, du choix de k, c'est-à-dire du nombre de composantes retenues pour reconstituer les données, ou encore de la dimension du sous-espace de représentation.De nombreux critères de choix pour q ont été proposés dans la littérature. Nous présentons ici ceux, les plus courants, basés sur une heuristique et un reposant sur une quantification de la stabilité du sous-espace de représentation :

Part d'inertie

La "qualité globale" des représentations est mesurée par f_k . La valeur de k est choisie de sorte que cette part d'inertie expliquée soit supérieure à une valeur seuil fixée a priori par l'utilisateur. C'est souvent le seul critère employé.

Éboulis

C'est le graphique (fig.1.1) présentant la décroissance des valeurs propres. Le principe consiste à rechercher, s'il existe, un "coude" (changement de signe dans la suite des différences d'ordre 2) dans le graphe et de ne conserver que les valeurs propres jusqu'à ce coude.

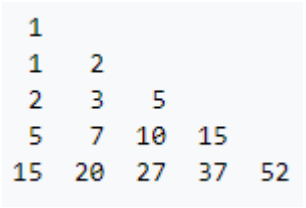


FIGURE 1.2 – Exemple de triangle de Bell

1.2 La classification non supervisée

1.2.1 Les méthodes

Un calcul de combinatoire montre que le nombre de partitions possibles d'un ensemble de n éléments croît exponentiellement avec n ; le nombre de partitions de n éléments en k classes est le nombre de Stirling, le nombre total de partitions est celui de Bell :

$$B_n = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!} = \sum_{k=0}^{n-1}\binom{n-1}{k}B_k$$

avec $B_0 = 1$. Les nombres de Bell peuvent facilement être calculés en créant le triangle dit de Bell, également appelé tableau de Aitken ou triangle de Peirce : Les nombres de Bell apparaissent à gauche et à droite du triangle. Pour $n = 20$ il est de l'ordre de 10^{13}. Il n'est donc pas question de chercher à optimiser le critère sur toutes les partitions possibles. Les méthodes se limitent à l'exécution d'un algorithme itératif convergeant vers une bonne partition et correspondant en général à un optimum local. Plusieurs choix sont laissés à l'initiative de l'utilisateur :

- une mesure d'éloignement (dissemblance, dissimilarité ou distance) entre individus ;
- le critère d'homogénéité des classes à optimiser : il est, dans le cas de variables quantitatives, généralement défini à partir de la trace d'une matrice de variances-covariances (la trace correspond alors à l'inertie de la partition).
- la méthode : classification ascendante hiérarchique, ré-allocation dynamique et DBSCAN sont les plus utilisées, seules ou combinées ;
- le nombre de classes : c'est un point délicat !

1.2.2 Utilisation pratique

Concrètement, il peut arriver que les données à traiter soient directement sous la forme d'une matrice d'un indice de ressemblance ou de dissemblance. Il est alors facile de la transformer en une matrice de dissemblances normées avant d'aborder une classification. Nous précisons ci-dessous les autres cas :

Algorithm *classification ascendante hiérarchique*

Initialiser classes par les singletons
Calculer la matrice de leurs distances deux à deux
repeat
Regrouper les deux classes les plus proches au sens de la distance entre classes choisie
Mettre à jour le tableau de distances en remplaçant les deux classes regroupées par la nouvelle et en calculant sa distance avec chacune des autres classes.
until Agrégation en une seule classe

FIGURE 1.3 – Algorihthme : CAH

— Données quantitatives : Il est nécessaire de définir une matrice M de produit scalaire sur $\mathbb{R}^p$ ($< X, Y >_M := X^T MY$) .Le choix $M = I$ est courant mais pour réduire les variables hétérogènes comme en ACP,il faut choisir la matrice diagonale composée des inverses des écarts-type ou la métrique dite de Mahalanobis (inverse de la matrice des variances-covariances) qui permet d'atténuer la structure de corrélation.
— Données qualitatives : Dans le cas particulier des variables qualitatives binaires de nombreux indices de ressemblances ont été proposés dans la littérature : Concordance , Jaccard et Dice .. Dans le cas plus général de p variables qualitatives,la distance la plus utilisée est celle, euclidienne dite du χ^2
— Mélange quantitatif, qualitatif : Soit on rend tout qualitatif par découpage en classes ou bien on rend tout quantitatif à l'aide d'une ACM (Analyse des Correspondances Multiples) ou on utilise la métrique de Gower qui permet de mixer les types de variables mais celle-ci reste très peu utilisée.

1.2.3 Classification ascendante hiérarchique

Algorithme

Fig.1.3

Résultats

— Le choix du nombre de classes k comme le choix de la dimension en ACP, délicat à opérer. Plusieurs heuristiques ont été proposées selon les critères précédents ou encore suivant le graphe de décroissance de la distance interclasses qui est aussi la décroissance de la variance inter-classe dans le cas du saut de Ward. La recherche

d'un "coude" dans ce graphe est une indication heuristique du choix de k .

— Il existe aussi des indicateurs de qualité d'une CAH : corrélation cophénétique , silhouette et statiqtique du gap etc ..

Chapitre 2

Applications

2.1 Classification des clients grossistes

2.1.1 Description des données

J'ai téléchargé ce jeu de données à partir du référentiel UCI Machine Learning Repository. L'ensemble de données fait référence aux clients d'un distributeur en gros. Il inclut les dépenses annuelles en unités monétaires pour diverses catégories de produits.

2.1.2 Objecif

L'objectif est d'utiliser diverses techniques de clustering pour segmenter les clients. Le clustering est un algorithme d'apprentissage non supervisé qui tente de regrouper des données en fonction de leur similarité. Ainsi, il n'y a pas de résultat à prévoir, et l'algorithme tente simplement de trouver des modèles dans les données.

2.1.3 Exploration et analyse des données

1. La tête et la structure des données d'origine :

```
client <- read.csv('Wholesale.csv')
head(client)
str(client)
```

2. K-Means Clustering : On prépare les données pour les analyser. On supprime les valeurs manquantes et on supprime aussi les colonnes «Channel» et «Region» car elles ne sont pas utiles pour la classification .

```
client1<- client
```

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
2	3	12669	9656	7561	214	2674	1338
2	3	7057	9810	9568	1762	3293	1776
2	3	6353	8808	7684	2405	3516	7844
1	3	13265	1196	4221	6404	507	1788
2	3	22615	5410	7198	3915	1777	5185
2	3	9413	8259	5126	666	1795	1451

```
'data.frame':   440 obs. of  8 variables:
 $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
 $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
 $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 600
6 ...
 $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 11093
...
 $ Grocery         : int  7561 9568 7684 4221 7198 5126 6975 9426 6192 18881
...
 $ Frozen          : int  214 1762 2405 6404 3915 666 480 1669 425 1159 ...
 $ Detergents_Paper: int  2674 3293 3516 507 1777 1795 3140 3321 1716 7425 ...
 $ Delicassen      : int  1338 1776 7844 1788 5185 1451 545 2566 750 2098 ...
```

FIGURE 2.1 – La tête et la structure des données

```
client1<- na.omit(client1) # Supression des valeurs manquantes
client1$Channel <- NULL
client1$Region <- NULL
```

3. On normalise les variables :

```
client1 <- scale(client1)
```

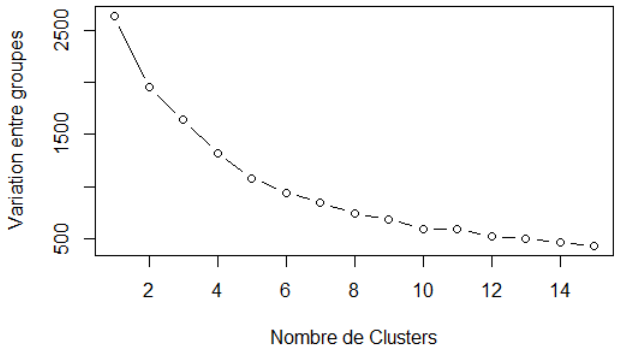
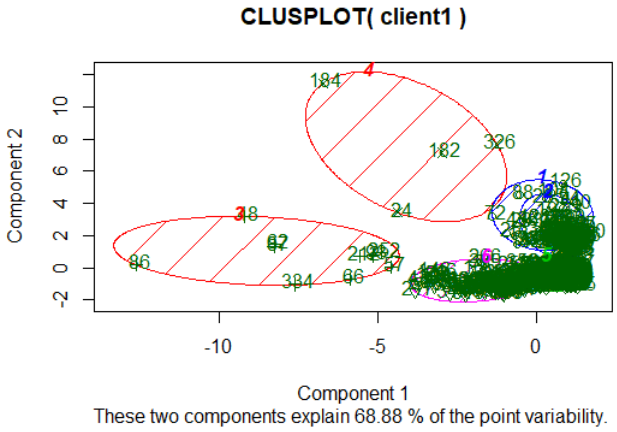


FIGURE 2.2 – Nombre des Clusters

4. On détermine le nombre des "Clusters" :

```
wss <- (nrow(client1)-1)*sum(apply(client1,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(client1,centers=i)$
    withinss)
plot(1:15, wss, type="b", xlab="Nombre de Clusters",ylab="
    Variation entre groupes")
```

avec : " Variations entre groupes " $= \sum(Y_{ij} - y_j)^2$ où Y_{ij} est le score d'individu i dans le groupe j et y_j est la moyenne du groupe j .
Le choix correct de k est souvent ambigu, mais à partir du graphique ci-dessus, je vais essayer mon analyse par le choix de 6 clusters .



5. On ajuste le modèle et on affiche la moyenne de cluster :

```
fit <- kmeans(client1, 6) # Ajuster le mod le
aggregate(client1,by=list(fit$cluster),FUN=mean)    # la moyenne
    de cluster
client1 <- data.frame(client, fit$cluster) # ajouter l'
   affectation de cluster
```

6. On affiche les résultats : Avec cet analyse, presque 70% des informations sur les données multivariées sont capturées par ce graphique des composantes 1 et 2.

```
library(cluster)
clusplot(client1, fit$cluster, color=TRUE,shade=TRUE, labels=2,
    lines=0)
```

7. Détection de valeurs aberrantes : Premièrement, les données sont partitionnées en k groupes en les affectant aux centres de cluster le plus proche, comme suit :

```
client2 <- client[, 3:8]
kmeans.result <- kmeans(client2, centers=6)
kmeans.result$centers
```

Ensuite, on calcule la distance entre chaque objet et son centre de groupe, puis on choisit ceux qui ont les distances les plus grandes en tant que points aberrants et on affiche les ID des points aberrants.

```
ID <- kmeans.result$cluster      # affiche les ID des clusters
```

```
centers <- kmeans.result$centers [ ID , ]
distances <- sqrt(rowSums((client2 - centers)^2)) # calcule les
    distances
outliers <- order(distances, decreasing=T)[1:5] # on prend les
   5 premi res distances
```

Laissez-moi rendre cela plus significatif : On regrouppe tout dans "Client2"

```
> print(client2[outliers,])
     Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
182 112151 29627   18148  16745             4948       8550
87   22925 73498   32114    987            20070        903
326  32717 16784   13626  60869             1272       5609
86   16117 46197   92780   1026            40827       2944
184  36847 43950   20170  36534              239      47943
```

```
print(client2[outliers,])
```

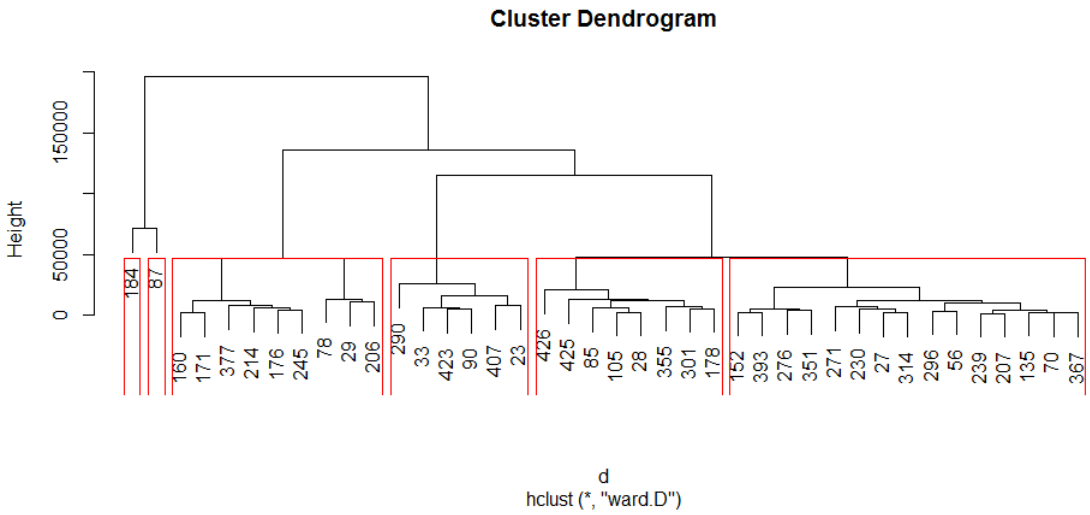
8. Hierarchical Clustering : On commence d'abord par extraire un échantillon de 40 enregistrements à partir des données client, de sorte que le graphe de classification ne soit pas saturé. Comme auparavant, les variables «Région» et «Channel» sont supprimées des données. Après cela, j'applique un clustering hiérarchique aux données.

```
idx <- sample(1:dim(client)[1], 40)
clientSample <- client[idx,]
clientSample$Region <- NULL
clientSample$Channel <- NULL
```

Il existe un large éventail de méthodes de classification hiérarchique. J'ai entendu dire que la méthode de Ward était une bonne idée, alors on va l'essayer :

```
d <- dist(clientSample, method = "euclidean") # matrice des
    distances
fit <- hclust(d, method="ward")
plot(fit) # affichage du dendogramme
groups <- cutree(fit, k=6) # couper l'arbre en 6 clusters
rect.hclust(fit, k=6, border="red") # dessiner le dendogramme avec
    des bordures rouges autour des 6 clusters
```

Interprétation : je commence par 40 points de données, chacun étant affecté à des clusters distincts ; les deux clusters les plus proches sont ensuite fusionnés jusqu'à ce qu'il ne reste plus qu'un cluster. La hauteur dans le dendrogramme à laquelle deux clusters sont fusionnées représente la distance entre deux clusters dans l'espace de données. L'observation du dendrogramme permet de choisir le nombre de groupes qui peut le mieux représenter différents groupes.



2.2 Exploration des attaques terroristes dans le monde

2.2.1 Description des données

La base de données Gloabl Terrorism (GDS) est une base de données publique contenant des informations sur les attaques terroristes dans le monde de 1970 à 2017. Elle contient plus de 150 000 attaques terroristes dans le monde entier entre 1970 et 2017 . Il y a 137 colonnes dans le jeu de données. Pour le rendre parfait, je vais devoir faire quelques sous-ensembles, en garadnt que les colonnes dont j'ai besoin .

2.2.2 Objectif

L'objectif est d'éxplorer les attaques terroristes dans le monde 1970 à 2017 en essayant de réduire le nombre des covariables avec l'ACP .

2.2.3 Exploration et analyse des données

```
# Loading
library(ggplot2)
library(dplyr)
library(ggthemes)
library("readxl")
library("FactoMineR")
library("factoextra")

# Import data : Source -> Global terrorism datbase sur https://www.
    start.umd.edu/gtd/
```

```
databrut <- read_excel("gdb.xlsx")
head(databrut)
str(databrut)
# Sel ction des features int ressantes pour la suite puis on vas
    les  nettoyer :
data <- na.omit(databrut[,c("iyear","imonth", "iday", "country_txt"
    , "region_txt", "provstate", "city", "latitude", "longitude", "
    attacktype1_txt", "targtype1_txt", "corp1", "target1", "natlty1_
    txt", "gname", "weaptype1_txt", "weapsubtype1_txt")])
head(data)
```

— Application de l'ACP :

```
# Pour appliquer l'ACP il nous faut des donnees quantitatives :
dataPCA <-  na.omit(databrut[,c("iyear","imonth", "iday","latitude"
    , "longitude")])
head(dataPCA)
str(dataPCA)
# PCA :

data.pca <- PCA(dataPCA, graph = FALSE)
# Le r sultat de la fonction PCA() est une liste, contenant les
      lments    suivants:
print(data.pca)

####################################
# Visualisation des valeurs propres :
####################################
# 1) Valeurs propres :
eig.val <- get_eigenvalue(data.pca)
> eig.val
       eigenvalue variance.percent cumulative.variance.percent
Dim.1   1.1690902         23.38180                    23.38180
Dim.2   1.0075547         20.15109                    43.53290
Dim.3   0.9976403         19.95281                    63.48570
Dim.4   0.9938479         19.87696                    83.36266
Dim.5   0.8318668         16.63734                   100.00000
>
# 2) Le graphique des valeurs propres :
fviz_eig(data.pca, addlabels = TRUE, ylim = c(0, 50))
```

```
# Remarque : Les 3 premieres PC expliquent 64% de la variance !

####################################
# Graphique des variables
####################################
## 1) R sultats :
var <- get_pca_var(data.pca)
## coordonn es des variables:
head(var$coord)
```

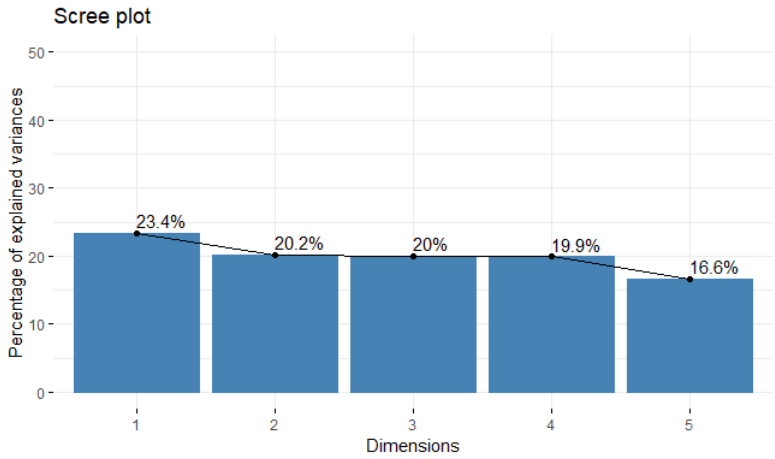


FIGURE 2.3 – Graphique des valeurs propres

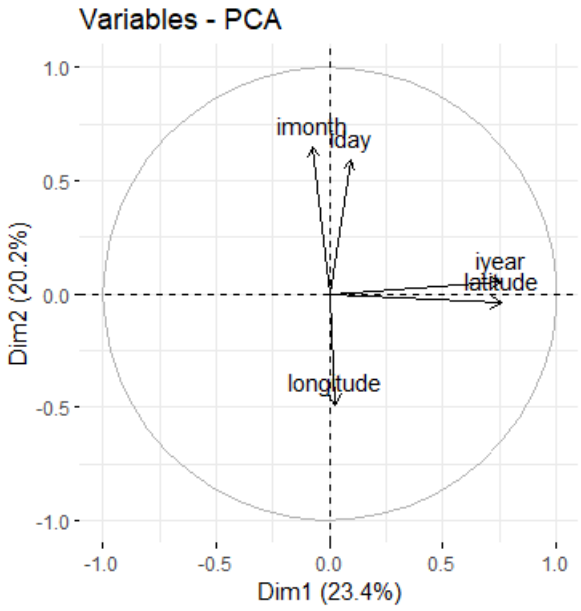


FIGURE 2.4 – Cercle de corrélation

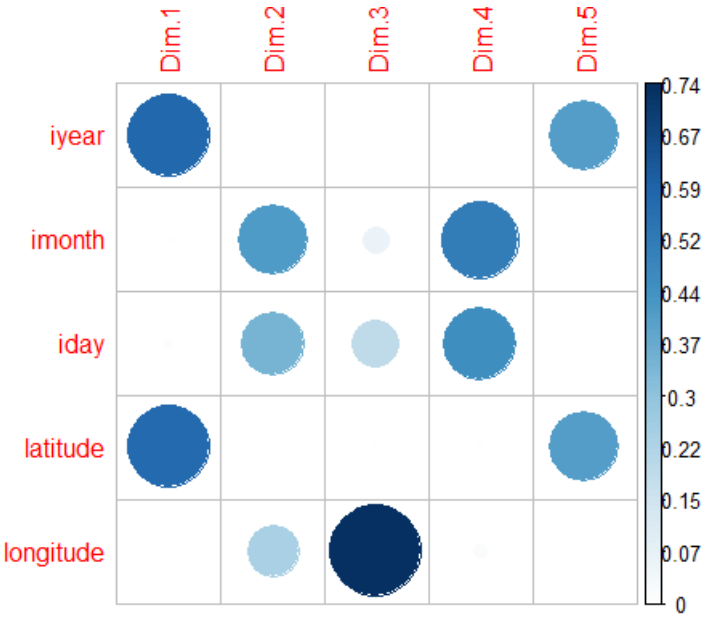


FIGURE 2.5 – Cos2 des variables sur toutes les dimensions

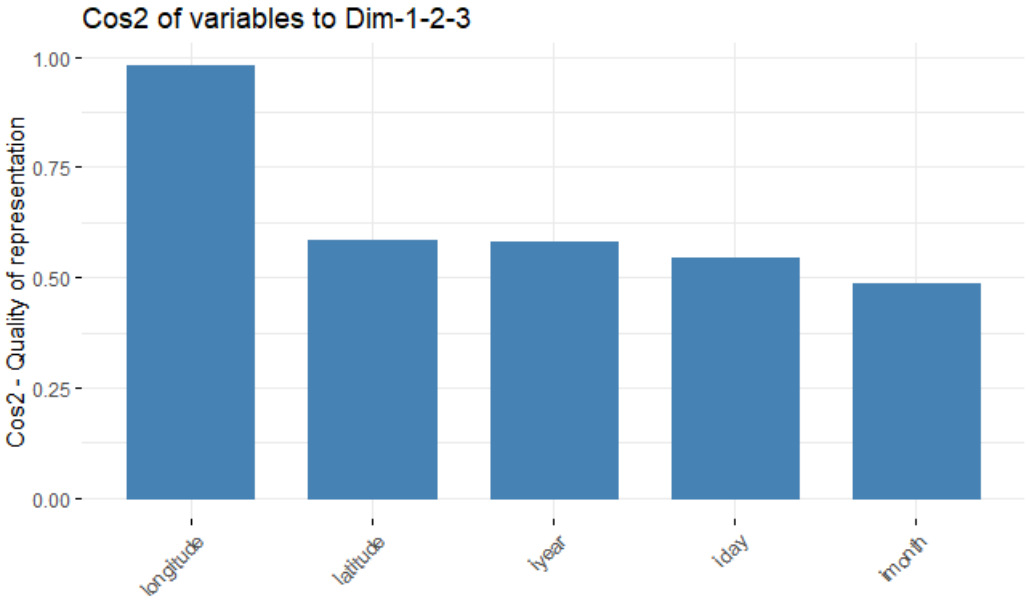


FIGURE 2.6 – Cos2 total des variables sur Dim.1 -> Dim.3

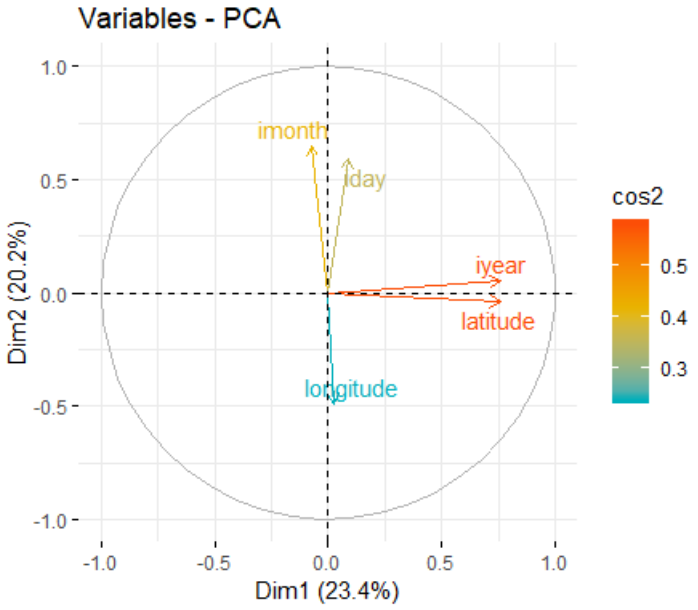


FIGURE 2.7 – Variables en fonction de Cos2

```
# Cos2: qualit  de representation
head(var$cos2)
# Contributions aux composantes principales
head(var$contrib)
## 2) Cercle de correlation :

head(var$cos2, 4)
corrplot(var$cos2, is.corr=FALSE)

# Cos2 total des variables sur Dim.1 -> Dim.3 :
fviz_cos2(data.pca, choice = "var", axes = 1:3)


# Un cos2   lev   indique une bonne repr sentation
# de la variable sur les axes principaux en consid ration.
# Dans ce cas, la variable est positionn e    proximit  de la
   circonf rence
# # du cercle de corr lation.Un faible cos2 indique que la
   variable  n est  pas
# parfaitement repr sent e par les axes principaux. Dans ce cas,
   la variable
# est proche du centre du cercle.

# Colorer en fonction du cos2: qualit  de repr sentation
fviz_pca_var(data.pca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE #  vite  le chevauchement de texte
```

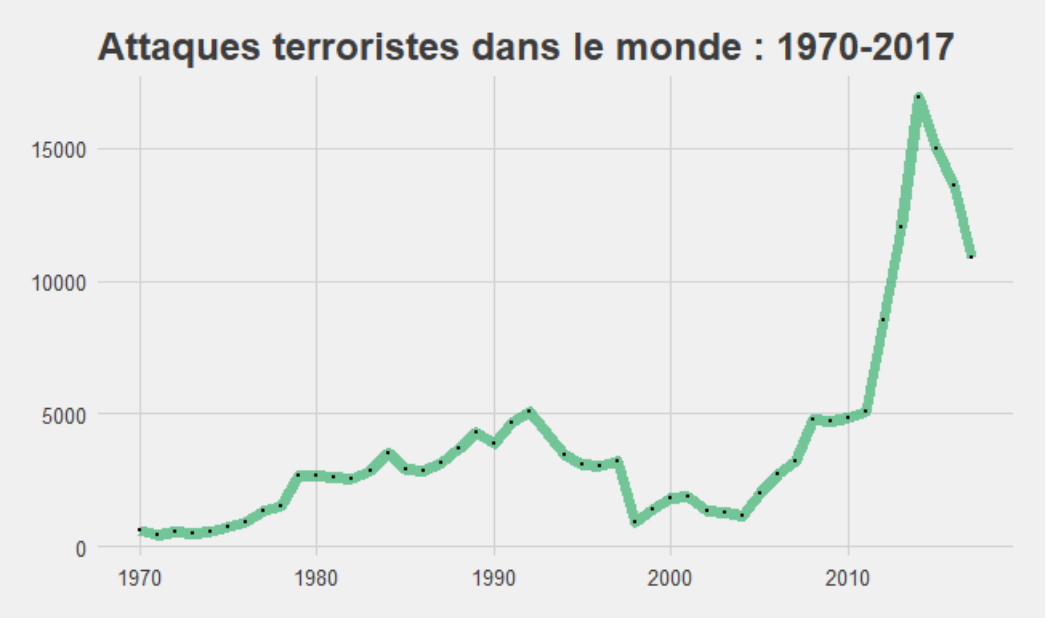


FIGURE 2.8 – Tendances d'attaques terroristes dans le monde : 1970-2017

```
)

# Contributions des variables aux axes principaux :
corrplot(var$contrib, is.corr=FALSE)


# Contributions des variables     PC1
fviz_contrib(data.pca, choice = "var", axes = 1, top = 3)
# Contributions des variables     PC2
fviz_contrib(data.pca, choice = "var", axes = 2, top = 3)

fviz_contrib(data.pca, choice = "var", axes = 1:2, top = 3)

#######
```

— Attaques terroristes dans le monde : 1970-2017 :

```
terrorism <- data
# Tendances d'attaques terroristes dans le monde : 1970-2017
by_year <- terrorism %>% group_by(iyear) %>% dplyr::summarise(n=n()
    )
ggplot(aes(x = iyear, y = n), data = by_year) +
  geom_line(size = 2.5, alpha = 0.7, color = "mediumseagreen") +
  geom_point(size = 0.5) + xlab("Ann e") + ylab("Nombre d'
      attentats terroristes") +
  ggtitle("Attaques terroristes dans le monde : 1970-2017") + theme
      _fivethirtyeight()
```

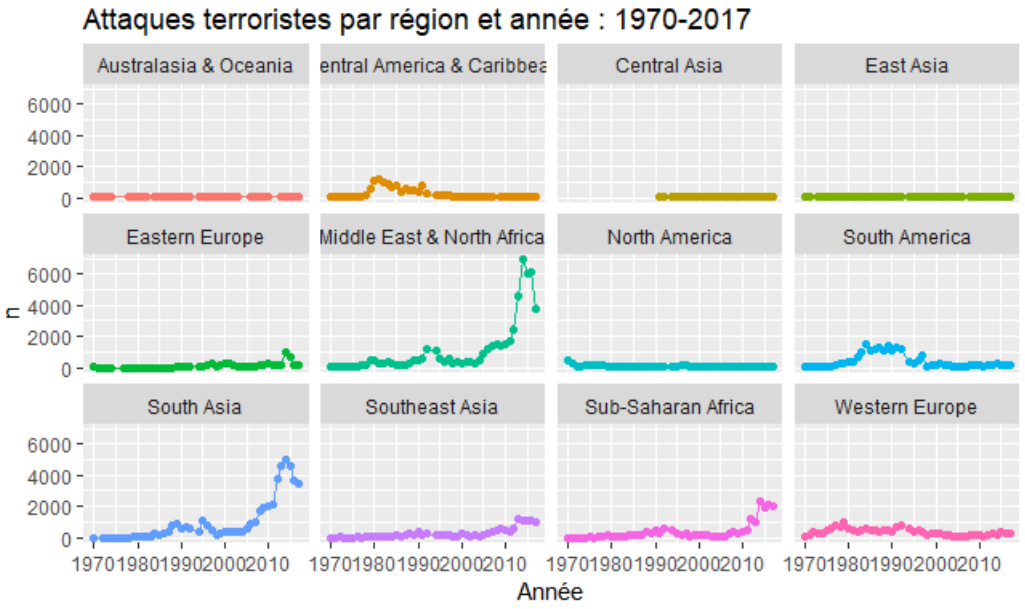


FIGURE 2.9 – Attaques terroristes par région et année : 1970-2017

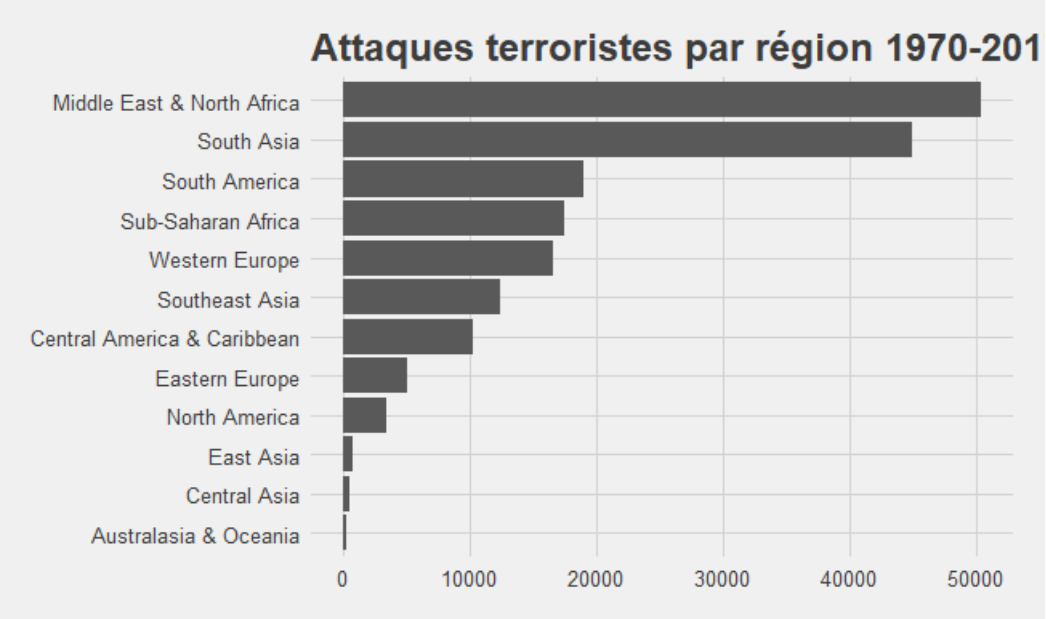


FIGURE 2.10 – Attaques terroristes par région 1970-2017

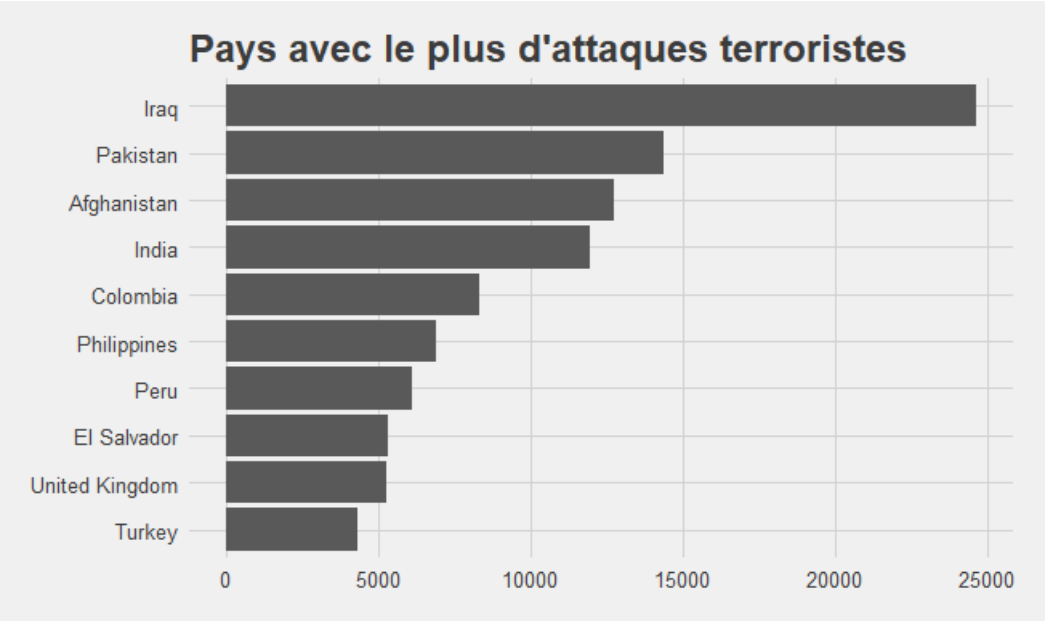


FIGURE 2.11 – Le Pays avec le plus d'attaques terroristes

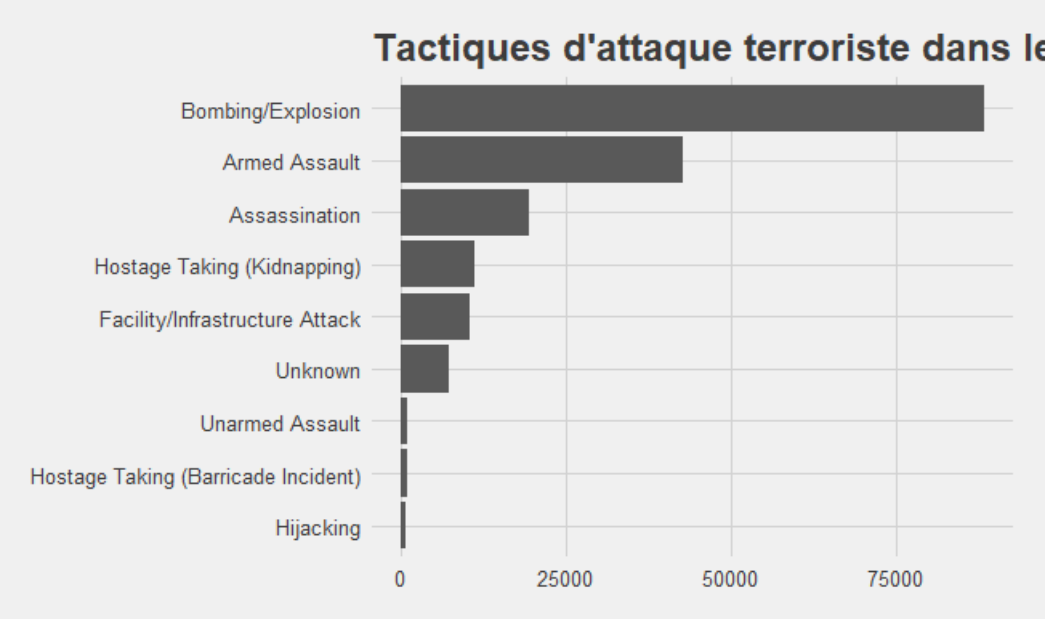


FIGURE 2.12 – Tactiques d'attaque terroriste dans le monde, 1970-2017

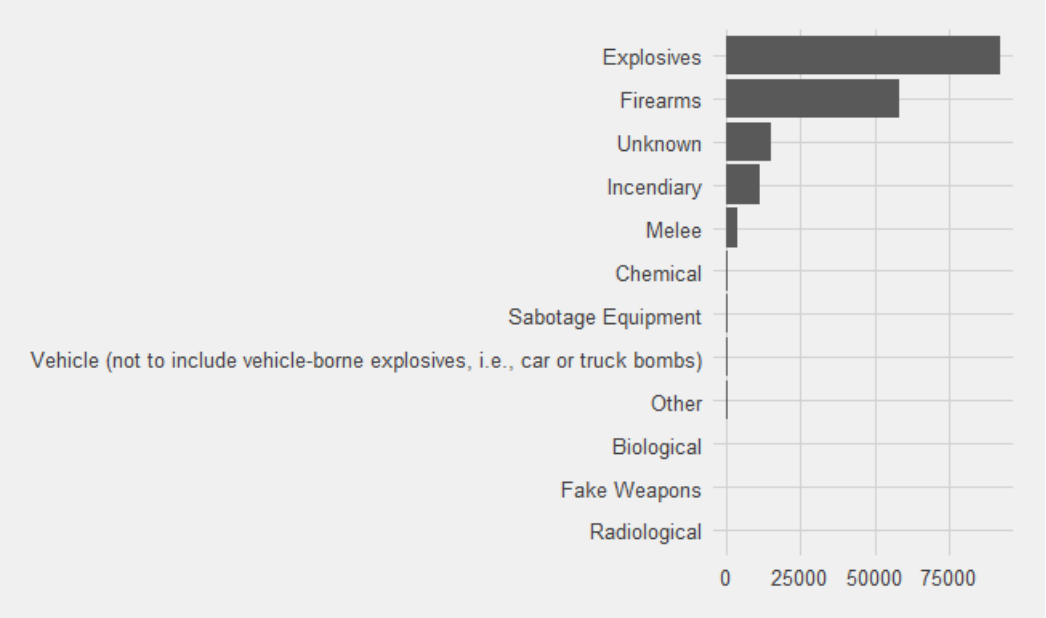


FIGURE 2.13 – Armes d'attaque terroriste dans le monde, 1970-2017

Globalement, les attaques terroristes ont considérablement augmenté depuis 2010 .

— Attaques terroristes par région et année 1970-2017 :

```
by_region <- terrorism %>% group_by(region_txt, iyear) %>% dplyr::
    summarise(n=n())
ggplot(by_region, aes(x = iyear, y = n, colour = region_txt)) +
  geom_line() +
  geom_point() +
  facet_wrap(~region_txt) + xlab('Ann e') +
  ggtitle('Attaques terroristes par r gion et ann e : 1970-2017'
      ) + theme(legend.position="none")
```

L'Amérique centrale était très instable à partir de la fin des années 70, elle s'est améliorée avec le temps et elle est stabilisée depuis environ 1995.
L'Europe occidentale a connu un passé difficile et a connu de nombreuses attaques jusqu'au début des années 2000.
L'Amérique du Sud avait le même schéma et était très dangereuse depuis le début de 1980 jusqu'à tout juste avant 2000.
Le Moyen-Orient, l'Afrique du Nord et l'Asie du Sud avaient connu un climat relativement calme autour de 1980, mais les attaques terroristes dans ces régions ont régulièrement augmenté entre les années 1980 et 2005, et ont considérablement augmenté depuis 2005 .

— Attaques terroristes par région 1970-2017 :

```
# par r gion
by_region_no_year <- terrorism %>% group_by(region_txt) %>% dplyr::
```

```
    summarise(n=n())
ggplot(aes(x=reorder(region_txt, n), y=n), data=by_region_no_year)
    +
  geom_bar(stat = 'identity') +
  ggtitle('Attaques terroristes par r gion 1970-2017') + coord_
     flip() + theme_fivethirtyeight()
```

Une petite fraction des attaques terroristes ont eu lieu dans les pays occidentaux. La plupart des attaques ont été fortement concentrées géographiquement au Moyen-Orient, en Afrique du Nord et en Asie du Sud. Regardons les pays :

— Pays avec le plus d'attentats terroristes entre 1970-2017 :

```
# par pays :
by_country <- terrorism %>% group_by(country_txt) %>% dplyr::
    summarise(n=n())
by_country <- arrange(by_country, desc(n))
top10 <- head(by_country, 10)
top10
ggplot(aes(x=reorder(country_txt, n), y=n), data=top10) +
  geom_bar(stat = 'identity') + xlab('Pays') + ylab("Nombre d'
     attantas") + ggtitle("Pays avec le plus d'attaques terroristes
     ") +
  coord_flip() + theme_fivethirtyeight()
```

L'Irak, l'Afghanistan et le Pakistan, l'Inde ont le plus souffert du terrorisme. Étonnamment, le Royaume-Uni arrive en tête de liste en Europe avec près de 5000 attaques entre 1970 et 2017.

— Tactiques et Armes :

```
# types d'attaques
by_attacktype <- terrorism %>% group_by(attacktype1_txt) %>% dplyr
    ::summarise(n=n())
ggplot(aes(x=reorder(attacktype1_txt, n), y=n), data=by_attacktype)
    +
  geom_bar(stat = 'identity') + xlab("Type d'attaque") + ylab("
     Nombre d'attaques") + ggtitle("Tactiques d'attaque terroriste
     dans le monde, 1970-2017") + coord_flip() +
  theme_fivethirtyeight()
```

```
# armes
by_weapon <- terrorism %>% group_by(weaptype1_txt) %>% dplyr::
    summarise(n=n())
ggplot(aes(x=reorder(weaptype1_txt, n), y=n), data=by_weapon) +
  geom_bar(stat = 'identity') + xlab('Arme') + ylab("Nombre d'
     attaques") + ggtitle("Armes d'attaque terroriste dans le monde
     , 1970-2017") + coord_flip() +
  theme_fivethirtyeight()
```

La tactique d'attaque la plus couramment utilisée de 1970 à 2017 consistait à utiliser une bombe et des explosifs, suivis d'assaut à main armée. Regardons l'année la plus

récente - 2017 :

— Attaques terroristes et victimes dans le monde par mois en 2017 :
Pour obtenir des informations plus détaillées sur les victimes , je suis allé sur le site Web du Département des États-Unis pour télécharger un petit jeu de données contenant des informations sur les victimes.

```
library(xlsx)
library(reshape2)
casualties <- read.xlsx('casualties.xlsx', sheetIndex = 1, header =
    TRUE, stringsAsFactors = F)
casualties <- melt(casualties, id="month")
casualties<-casualties[!casualties$month=="Total",]
casualties$month <- ordered(casualties$month, levels=c('January', '
   February', 'March', 'April', 'May', 'June', 'July', 'August', '
   September', 'October', 'November', 'December'))
ggplot(aes(x=month, y=value, fill=variable), data=casualties) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  ggtitle('Terrorist attacks and casualties worldwide by month,
     2017') +
  theme_fivethirtyeight()
```

Le nombre total de personnes tuées lors d'attaques terroristes a culminé en avril et juillet 2017, et les mois ayant enregistré le plus grand nombre de morts et de blessés étaient les mois de juin, juillet et août et les mois de janvier et mai qui ont enregistré le plus grand nombre d'enlèvements.

— Informations sur les groupes terroristes :
Comme il y avait tellement de valeurs inconnues dans l'ensemble de données d'origine, je dois extraire à nouveau des données du site Web du département d'État américain sur les informations relatives aux groupes terroristes.

```
group <- read.xlsx('group.xlsx', sheetIndex = 1, header=T,
   stringsAsFactors=F)
group <- melt(group, id = 'group_name')
group[5, 1] = "Kurdistan Workers' Party"
group[10, 1] = "Kurdistan Workers' Party"
group[15, 1] = "Kurdistan Workers' Party"
group[20, 1] = "Kurdistan Workers' Party"
ggplot(aes(x=group_name, y=value, fill=variable), data=group) +
  geom_bar(stat = 'identity', position = position_dodge()) +
     ggtitle('Terrorist Groups with the most Attacks Wordwide, 2017
     ') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Outre le nombre d'attaques terroristes qu'ils ont menées, ces cinq groupes terroristes ont été à l'origine du plus grand nombre d'attaques terroristes de 2017. Parmi ces cinq groupes, l'État islamique en Iraq et au Levant (EIIL) est l'organisation terroriste la plus meurtrière au monde. 6000 décès en 2017.

— Cibles / victimes d'attaques terroristes, 2017 :

```
attack2017 <- terrorism[terrorism$iyear==2017, ]
by_target <- attack2017 %>% group_by(targtype1_txt) %>%
  summarise(n=n())
by_target <- arrange(by_target, desc(n))
by_target
ggplot(aes(x=reorder(targtype1_txt, n), y=n), data=by_target) +
  geom_bar(stat = 'identity') + ggtitle('Terrorist Attack Targets/
     Victims, 2017') +
  coord_flip() + theme_fivethirtyeight()
```

— Quels pays / villes étaient les plus dangereux en 2017 ?

```
attack2017_by_city <- attack2017 %>% group_by(country_txt, city)
   %>%
  summarise(n=n())
attack2017_by_city <- arrange(attack2017_by_city, desc(n))
top10_city_2017 <- head(attack2017_by_city, 20)
top10_city_2017
```

Bagdad était la ville la plus dangereuse en 2017, avec environ 1000 attaques terroristes en un an, mais depuis quand cela est devenu dangereux ?

```
baghdad <- terrorism[terrorism$city=='Baghdad', ]
baghdad_year <- baghdad %>% group_by(iyear) %>%
  summarise(n=n())
ggplot(aes(x = iyear, y = n), data = baghdad_year) +
        geom_line(size = 2.5, alpha = 0.7, color = "mediumseagreen"
            ) +
        geom_point(size = 0.5) + xlab("Year") + ylab("Number of
           terrorist Attacks") +
        ggtitle("Terrorist Attacks in Baghdad by Year 1970-2017") +
            theme_fivethirtyeight()
```

```
baghdad_type <- baghdad %>% group_by(attacktype1_txt, iyear) %>%
  summarise(n=n())
ggplot(aes(x=iyear, y=n, fill=attacktype1_txt), data=baghdad_type)
   +
  geom_bar(stat = 'identity') +
  ggtitle('Attack Type in Baghdad') + theme_fivethirtyeight()
```

Bagdad était autrefois un centre culturel et d'apprentissage prestigieux. Depuis l'invasion de la coalition en 2003, elle est devenue l'une des villes les plus dangereuses de la planète.

— Carte thermique des victimes d'attentats terroristes dans le monde en 2017 :

```
gtd <- read.csv("terrorism.csv")
gtd2017 <- gtd[gtd$iyear==2017, ]
gtd2017 <- aggregate(nkill~country_txt,gtd2017,sum)
library(rworldmap)
gtdMap <- joinCountryData2Map( gtd2017,
```

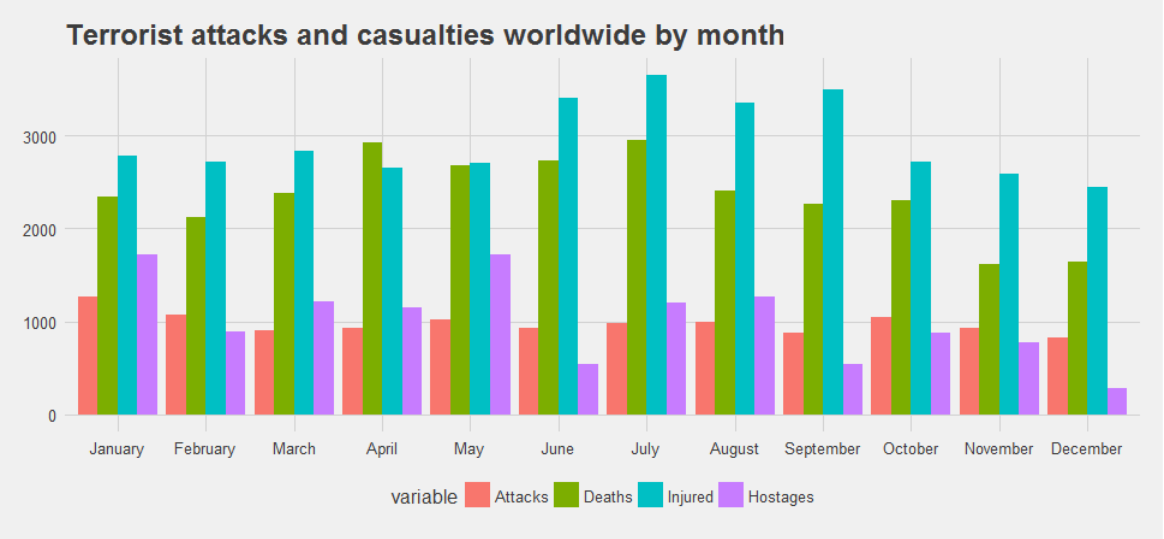


FIGURE 2.14 – Nombre des victimes

```
                                    nameJoinColumn="country_txt",
                                    joinCode="NAME" )

mapDevice('x11')
mapCountryData( gtdMap,
                nameColumnToPlot='nkill',
                catMethod='fixedWidth',
                numCats=100 )
```

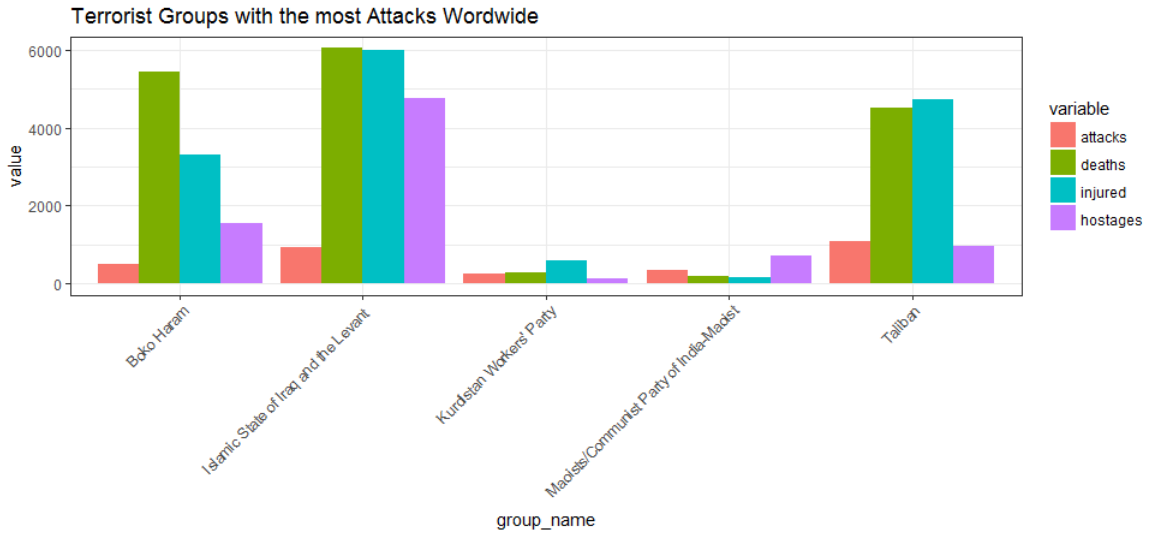


FIGURE 2.15 – Groupes terroristes

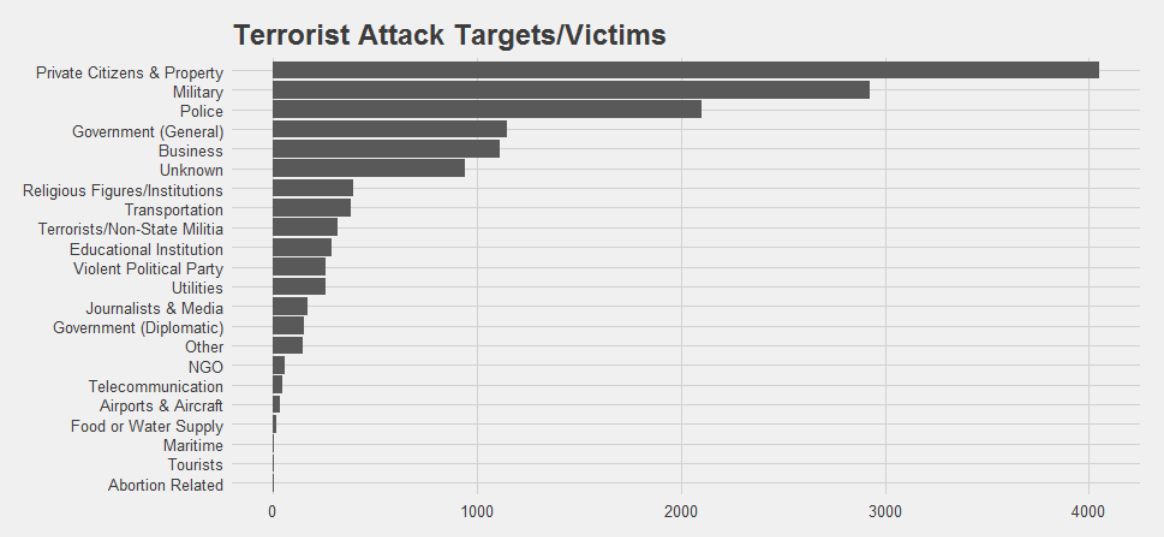


FIGURE 2.16 – Victimes cibles

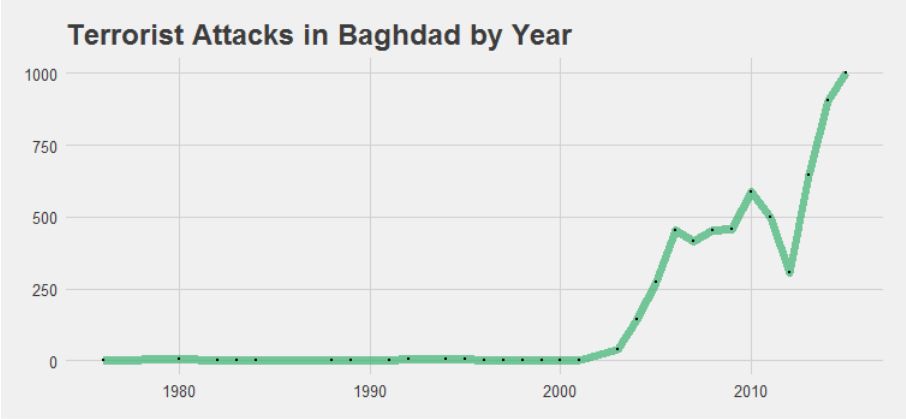


FIGURE 2.17 – Baghdad : Attaques

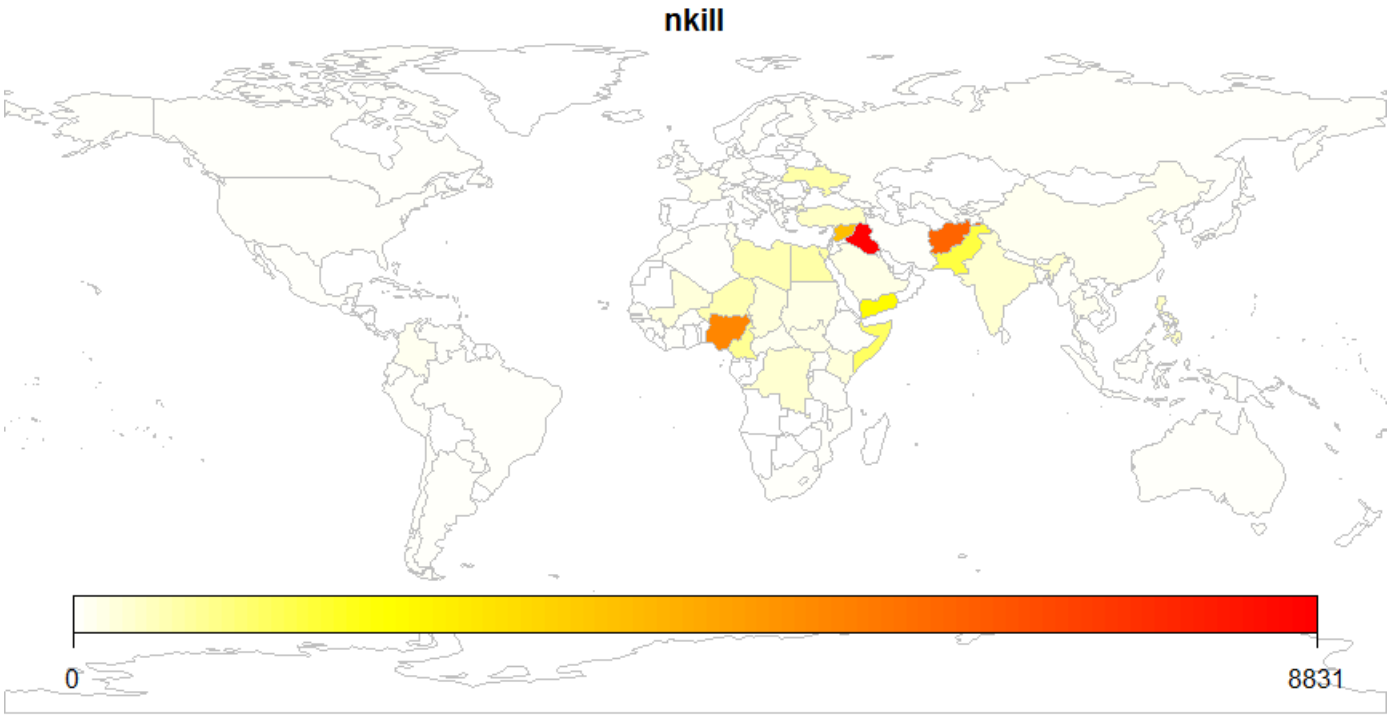


FIGURE 2.18 – Carte thermique des victimes d'attentats terroristes dans le monde