Projet Traitement numérique de données

TANRIVERDI Ambre 21607950

04/27/2020

Covid-19

```
# Graphiques
library(dplyr)
library(ggplot2)
# Heatmap et corrélation
library(corrplot)
library("pheatmap")
library("RColorBrewer")
library(Hmisc)
# Affichage des cartes
library(viridisLite)
library(viridis)
library(ggplot2)
library(dplyr)
library(maps)
# Assemblage des cartes
library(gridExtra, warn.conflicts = FALSE)
library(grid)
library(lattice)
# MAE et MSE
library(MLmetrics, warn.conflicts = FALSE)
# ACP
library(psy)
library(factoextra)
library(FactoMineR)
# CAH
library(dendextend)
library(cluster)
```

Librairies utilisées

Partie 1 : Analyses descriptives

1

```
covid = read.csv("donnees.csv", sep = ";")
head(covid)
```

Chargement du jeu de données

```
##                maille_nom duree_jours   latitude longitude deces_total
## 1                     Ain          15 46,2475706 5,1307681         431
## 2                   Aisne          15 49,4769199 3,4417368        1641
## 3                  Allier          15 46,3115552 3,4167655         138
## 4 Alpes-de-Haute-Provence          15 44,0778716 6,2375947          55
## 5         Alpes-Maritimes          15 43,9466791  7,179026         766
## 6                 Ardèche          15  44,759629 4,5624426         472
##   reanimation_total hospitalises_total gueris_total
## 1               446               1691         1457
## 2               602               3563         3204
## 3               266                745          959
## 4                68                452          783
## 5              1088               3484         2939
## 6               282               1473         2249
```

2

```
dim(covid)
```

Présentation du jeu de données

```
## [1] 100   8
```

```
summary(covid)
```

```
##                    maille_nom  duree_jours          latitude       longitude
##  Ain                     : 1   Min.   :15.00   46,6613966: 2   -0,7532809: 2
##  Aisne                   : 1   1st Qu.:15.00   -12,8275  : 1   0,4502368 : 2
##  Allier                  : 1   Median :15.00   -21,115141: 1   6,2375947 : 2
##  Alpes-de-Haute-Provence: 1   Mean   :15.42   14,6817939: 1   -0,3962844: 1
##  Alpes-Maritimes         : 1   3rd Qu.:15.00   16,265    : 1   -0,4330578: 1
##  Ardèche                 : 1   Max.   :20.00   3,933889  : 1   -0,4502368: 1
##  (Other)                 :94                   (Other)   :93   (Other)   :91
##   deces_total    reanimation_total hospitalises_total  gueris_total
##  Min.   :    0   Min.   :   12.0   Min.   :  113.0    Min.   :   79.0
##  1st Qu.:  135   1st Qu.:  185.5   1st Qu.:  840.8    1st Qu.:  573.5
##  Median :  355   Median :  410.5   Median : 1439.5    Median : 1166.0
##  Mean   : 1047   Mean   : 1000.2   Mean   : 4357.9    Mean   : 2962.3
##  3rd Qu.: 1016   3rd Qu.: 1069.0   3rd Qu.: 3819.5    3rd Qu.: 3238.5
##  Max.   :10285   Max.   :12013.0   Max.   :45669.0    Max.   :23513.0
##
```

```
glimpse(covid)
```

```
## Observations: 100
## Variables: 8
## $ maille_nom         <fct> Ain, Aisne, Allier, Alpes-de-Haute-Provence, Alp...
## $ duree_jours        <int> 15, 15, 15, 15, 15, 15, 15, 16, 15, 16, 16, 15, ...
## $ latitude           <fct> "46,2475706", "49,4769199", "46,3115552", "44,07...
## $ longitude          <fct> "5,1307681", "3,4417368", "3,4167655", "6,237594...
## $ deces_total        <int> 431, 1641, 138, 55, 766, 472, 118, 13, 618, 478,...
## $ reanimation_total  <int> 446, 602, 266, 68, 1088, 282, 254, 70, 340, 263,...
## $ hospitalises_total <int> 1691, 3563, 745, 452, 3484, 1473, 1061, 259, 253...
## $ gueris_total       <int> 1457, 3204, 959, 783, 2939, 2249, 544, 174, 1371...
```

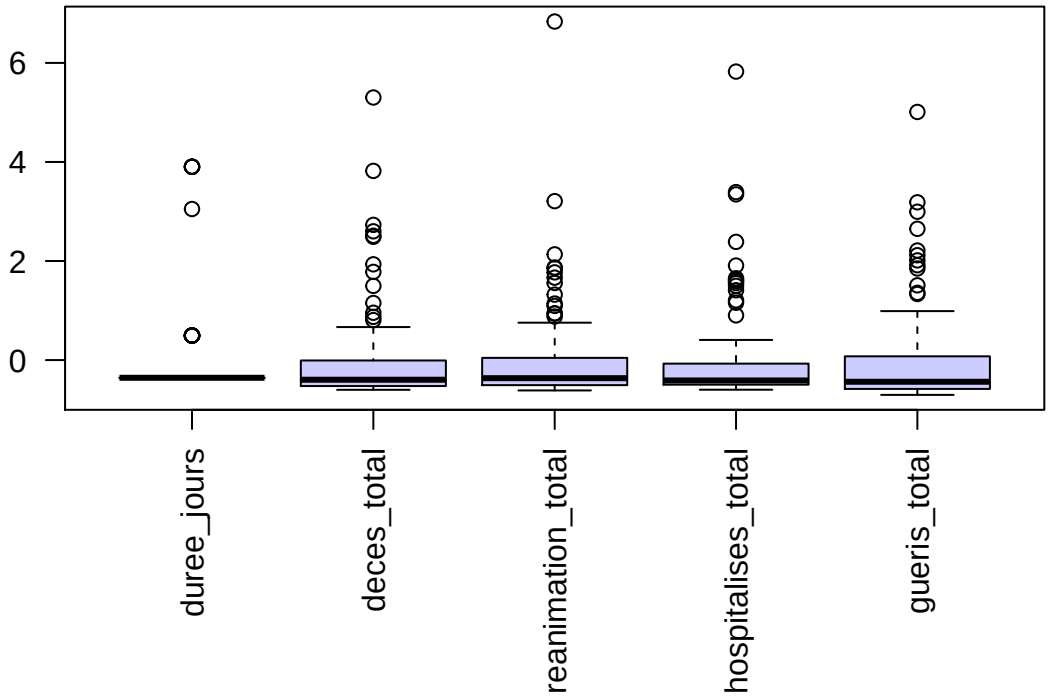
Présentation du jeu de données : - dimensions : 100 lignes et 8 colonnes - types de variables : nous avons des variables qualitatives nominales (maille_nom, latitude, longitude) et des variables quantitatives discrètes (les autres variables)

```
cov = covid[,-c(3, 4)]
head(cov)
```

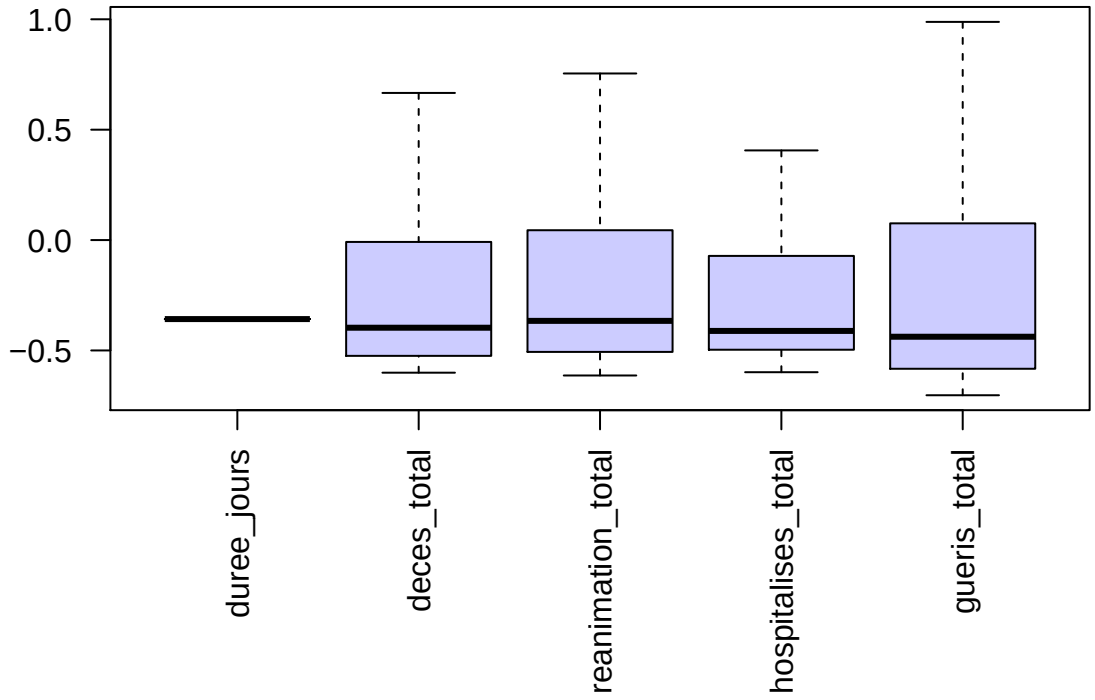
Analyse descriptive

```
##                maille_nom duree_jours deces_total reanimation_total
## 1                     Ain          15         431               446
## 2                   Aisne          15        1641               602
## 3                  Allier          15         138               266
## 4 Alpes-de-Haute-Provence          15          55                68
## 5         Alpes-Maritimes          15         766              1088
## 6                 Ardèche          15         472               282
##   hospitalises_total gueris_total
## 1               1691         1457
## 2               3563         3204
## 3                745          959
## 4                452          783
## 5               3484         2939
## 6               1473         2249
```

```
# Boîtes à moustaches sur toutes les variables numériques
par(mar=c(8,6,4,1))
# On met les données à la même échelle pour la lisibilité
sca = scale(cov[, -1], center = T, scale = T)
boxplot(sca, las = 2, col=rgb(0.8, 0.8, 1))
```

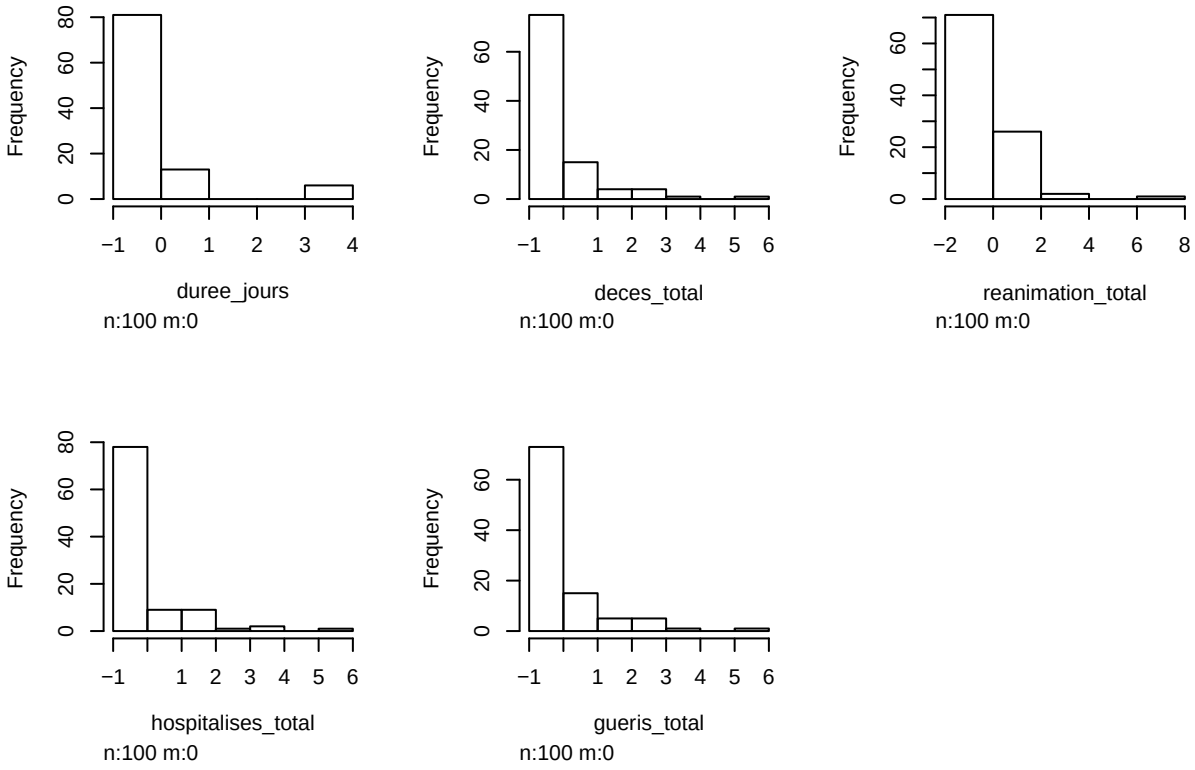



```
boxplot(sca, las = 2, col=rgb(0.8, 0.8, 1), outline = FALSE)
```



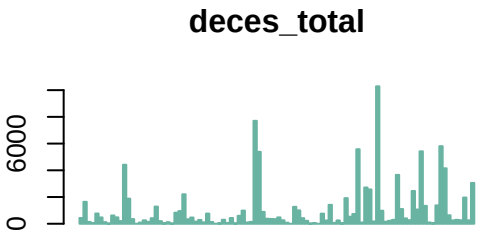
Nous avons des données aberrantes pour toutes les variables. Les centrages sont plutôt similaires, et la dispersion également, en dehors de duree_jours.

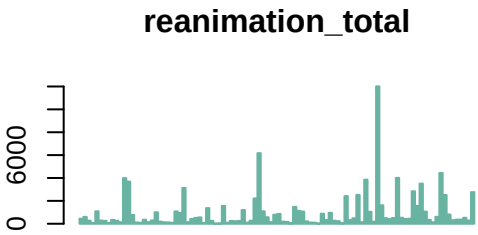
```
# Histogramme pour toutes les variables
par(mfrow = c(2, 3))
hist.data.frame(as.data.frame(sca))
```

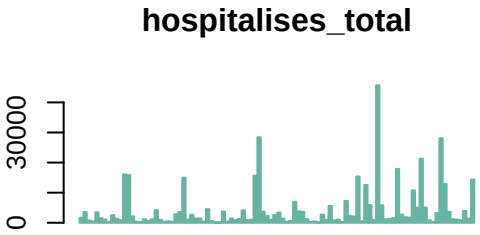


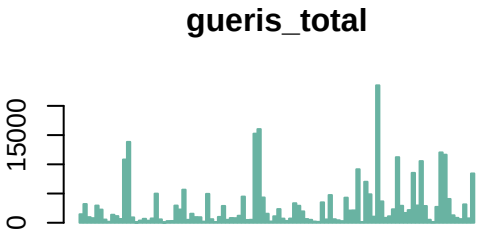
Nous avons une grande asymétrie à droite pour chaque variable, et quelques données aberrantes (surtout pour duree_jours). La dispersion et le centrage sont similaires.

```
# Diagramme en barres
par(mfrow=c(2,2))

cov_names = names(cov)

for (i in c(3:length(colnames(cov)))) {
  p = cov[, i]
  barplot(p, main = cov_names[i], col = "#69b3a2", border = "#69b3a2")
}
```



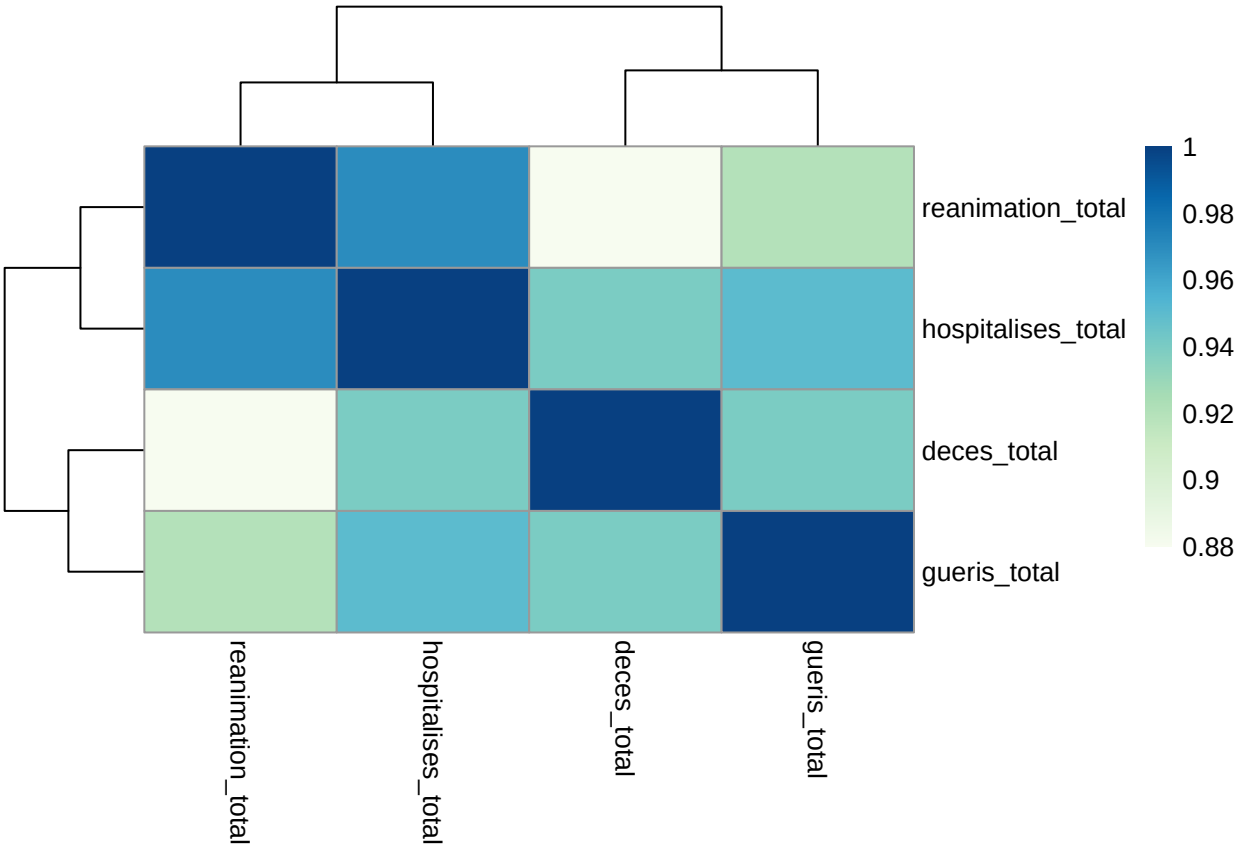




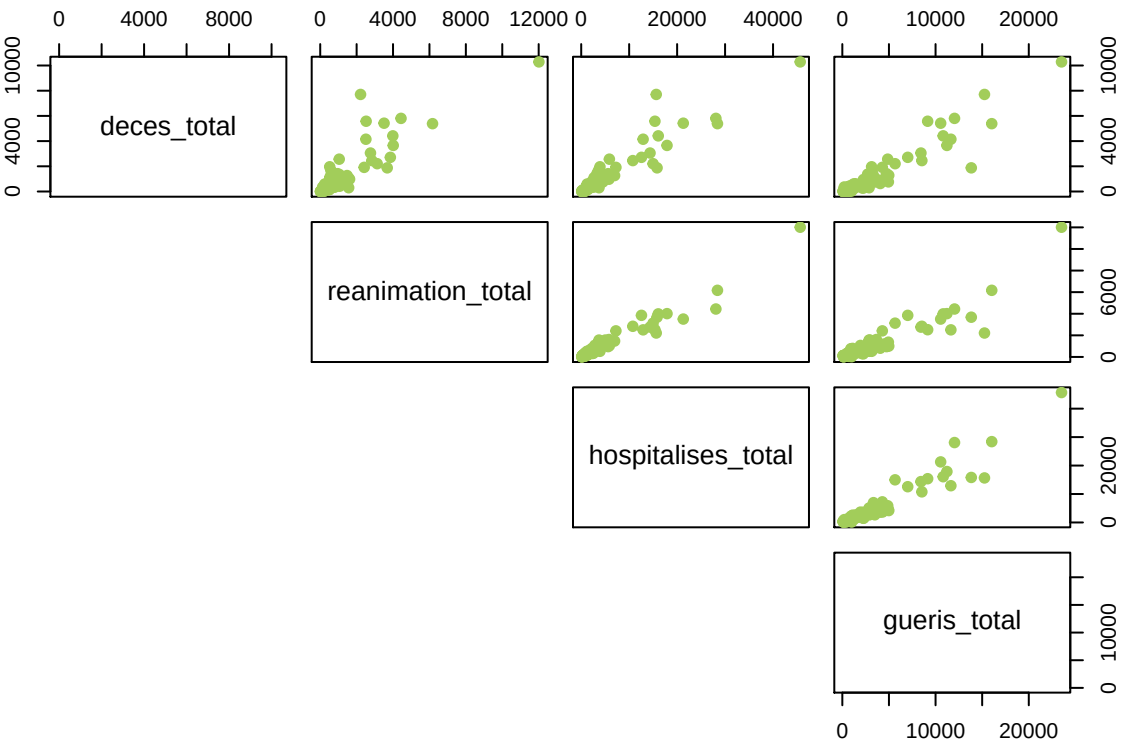


Les valeurs des quatres variables sont visuellement similaires entre elles.

```
# Heatmap des variables
corr = signif(cor(cov[, -c(1, 2)]), 2)

col <- colorRampPalette(brewer.pal(9, "GnBu"))(256)
pheatmap(corr, col = col)
```

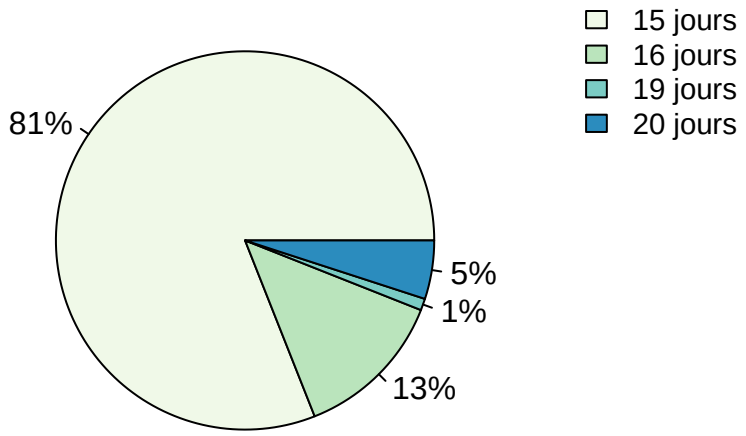



```
pairs(cov[,-c(1, 2)], pch = 19, lower.panel = NULL, col = "darkolivegreen3")
```



Toutes les variables numériques sont fortement corrélée positivement entre elles.

```r
# Diagramme en camembert
tbl = table(cov[, 2])
slices = as.vector(tbl)
pct = round(slices/sum(slices)*100)
pct = paste(pct,"%",sep="")

names = paste(names(tbl), "jours", sep = " ")
cols = brewer.pal(n = length(tbl), name = 'GnBu')

pie(slices, labels = pct, main = "Fréquence de la durée en jours", col = cols)
legend("topright", legend = names, cex=0.9, bty = "n", fill = cols)
```



Nous voyons que 81% des département ont récolté 15 donnée, 13% ont 16 données, 5% en ont 19, et seulement 1% en a 20.

3

```r
cov_dbl = covid
# conversion de la latitude et de la longitude en variable numérique
cov_dbl$longitude <- gsub(',', '.', cov_dbl$longitude)
cov_dbl$latitude <- gsub(',', '.', cov_dbl$latitude)
cov_dbl$longitude=as.numeric(cov_dbl$longitude)
cov_dbl$latitude=as.numeric(cov_dbl$latitude)

head(cov_dbl, 2)
```

Affichage des données sur des cartes géographiques.

```
##   maille_nom duree_jours latitude longitude deces_total reanimation_total
## 1        Ain          15 46.24757  5.130768         431               446
## 2      Aisne          15 49.47692  3.441737        1641               602
##   hospitalises_total gueris_total
## 1               1691         1457
## 2               3563         3204
```

Pour afficher la carte de France avec les régions d'Outre-Mer sans représentation des océans, j'ai décidé de réaliser chaque carte à part et de les combiner à la fin.

```
# Sélection de la carte de France (hors régions d'Outre-Mer)
fr <- map_data("world") %>% filter(region=="France")
data <- world.cities %>% filter(country.etc=="France")

# Sélection des départements se trouvant en France Métropolitaine
outre_mer = c("Mayotte", "Guyane", "Guadeloupe", "Martinique", "La Réunion")

metropoli = cov_dbl
metropoli = metropoli[!(metropoli$maille_nom %in% outre_mer),]
```

Dans cette partie, nous préparons la carte de la France métropolitaine.

```
# deces_total
metropoli_deces = ggplot() +
  geom_polygon(data = fr, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = metropoli, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alpha
  scale_size_continuous(range=c(1,10), name = "") +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5, breaks = c(50, 400,
  theme_void() +
  guides( colour = guide_legend(title="Nombre de personnes")) +
  coord_map()

# reanimation_total
metropoli_reanimation = ggplot() +
  geom_polygon(data = fr, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = metropoli, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_
  scale_size_continuous(range=c(1,10), name = "") +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5, breaks = c(50, 400,
  theme_void() +
  guides( colour = guide_legend(title="Nombre de personnes")) +
  coord_map()

# hospitalises_total
metropoli_hospitalises = ggplot() +
  geom_polygon(data = fr, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = metropoli, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalises
  scale_size_continuous(range=c(1,10), name = "") +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5, breaks = c(400, 3000
  theme_void() +
  guides( colour = guide_legend(title="Nombre de personnes")) +
  coord_map()

# gueris_total
metropoli_gueris = ggplot() +
  geom_polygon(data = fr, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point(data = metropoli, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), alph
  scale_size_continuous(range=c(1,10), name = "") +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5, breaks = c(400, 3000
  theme_void() +
  guides(colour = guide_legend(title="Nombre de personnes")) +
  coord_map()
```

Dans cette partie, nous préparons les cartes de régions d'Outre-Mer

```
outre_mer = c("Mayotte", "Guyane", "Guadeloupe", "Martinique", "La Réunion")

# création de sous data frames pour chaque département
for (i in outre_mer){
  copy = cov_dbl[cov_dbl$maille_nom == i,]
  if (i == "La Réunion")
    assign("reunion", copy)
  assign(tolower(i), copy)
}

# Sélection des cartes de chaque département
may <- map_data("world") %>% filter(region=="Mayotte")
guy <- map_data("world") %>% filter(region=="Guyana")
gua <- map_data("world") %>% filter(region=="Guadeloupe")
mar <- map_data("world") %>% filter(region=="Martinique")
reu <- map_data("world") %>% filter(region=="Reunion")
```

```
# Mayotte
# taille du graphique
options(repr.plot.width = 14, repr.plot.height = 8)

# deces_total
mayotte_deces = ggplot() +
  geom_polygon(data = may, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = mayotte, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alpha=
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Mayotte") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# reanimation_total
mayotte_reanimation = ggplot() +
  geom_polygon(data = may, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = mayotte, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_to
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Mayotte") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# hospitalises_total
mayotte_hospitalises = ggplot() +
  geom_polygon(data = may, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = mayotte, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalises_
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Mayotte") +
```

```
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# gueris_total
mayotte_gueris = ggplot() +
  geom_polygon(data = may, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = mayotte, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), alpha
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Mayotte") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))
```

```
# Guyane
# taille de la carte
options(repr.plot.width = 14, repr.plot.height = 8)
# ajustement des coordonnées GPS pour corriger l'emplacement du point sur la carte
guyane[, 4] = -59

# deces_total
guyane_deces = ggplot() +
  geom_polygon(data = guy, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guyane, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alpha=0
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Guyane") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# reanimation_total
guyane_reanimation = ggplot() +
  geom_polygon(data = guy, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guyane, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_tota
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Guyane") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# hospitalises_total
guyane_hospitalises = ggplot() +
  geom_polygon(data = guy, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guyane, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalises_to
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
```

```
  labs(subtitle = "Guyane") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# gueris_total
guyane_gueris = ggplot() +
  geom_polygon(data = guy, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guyane, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), alpha=
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Guyane") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))
```

```
# Guadeloupe
# taille de la carte
options(repr.plot.width = 14, repr.plot.height = 8)

# deces_total
guadeloupe_deces = ggplot() +
  geom_polygon(data = gua, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guadeloupe, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alp
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Gouadeloupe") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# reanimation_total
guadeloupe_reanimation = ggplot() +
  geom_polygon(data = gua, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guadeloupe, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Gouadeloupe") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# hospitalises_total
guadeloupe_hospitalises = ggplot() +
  geom_polygon(data = gua, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guadeloupe, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalis
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Gouadeloupe") +
```

```
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# gueris_total
guadeloupe_gueris = ggplot() +
  geom_polygon(data = gua, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = guadeloupe, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), al
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Gouadeloupe") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))
```

```
# Martinique
# taille de la carte
options(repr.plot.width = 14, repr.plot.height = 8)

# deces_total
martinique_deces = ggplot() +
  geom_polygon(data = mar, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = martinique, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alph
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Martinique") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# reanimation_total
martinique_reanimation = ggplot() +
  geom_polygon(data = mar, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = martinique, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Martinique") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# hospitalises_total
martinique_hospitalises = ggplot() +
  geom_polygon(data = mar, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = martinique, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalise
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Martinique") +
  theme_void() +
```

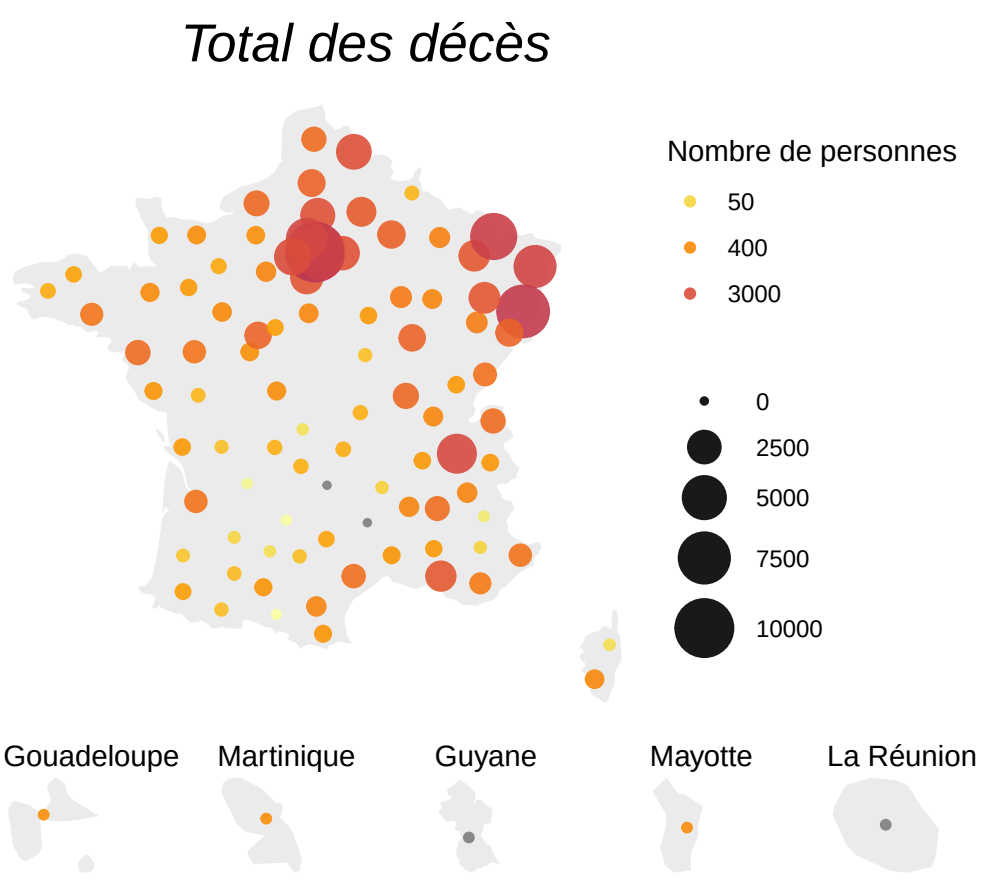
```
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# gueris_total
martinique_gueris = ggplot() +
  geom_polygon(data = mar, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = martinique, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), al
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "Martinique") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))
```

```
# La Réunion
# taille de la carte
options(repr.plot.width = 14, repr.plot.height = 14)

# deces_total
reunion_deces = ggplot() +
  geom_polygon(data = reu, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = reunion, aes(x=longitude, y=latitude, size=deces_total, color=deces_total), alpha=0
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "La Réunion") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# reanimation_total
reunion_reanimation = ggplot() +
  geom_polygon(data = reu, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = reunion, aes(x=longitude, y=latitude, size=reanimation_total, color=reanimation_to
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "La Réunion") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# hospitalises_total
reunion_hospitalises = ggplot() +
  geom_polygon(data = reu, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = reunion, aes(x=longitude, y=latitude, size=hospitalises_total, color=hospitalises_
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "La Réunion") +
  theme_void() +
  coord_map() +
```

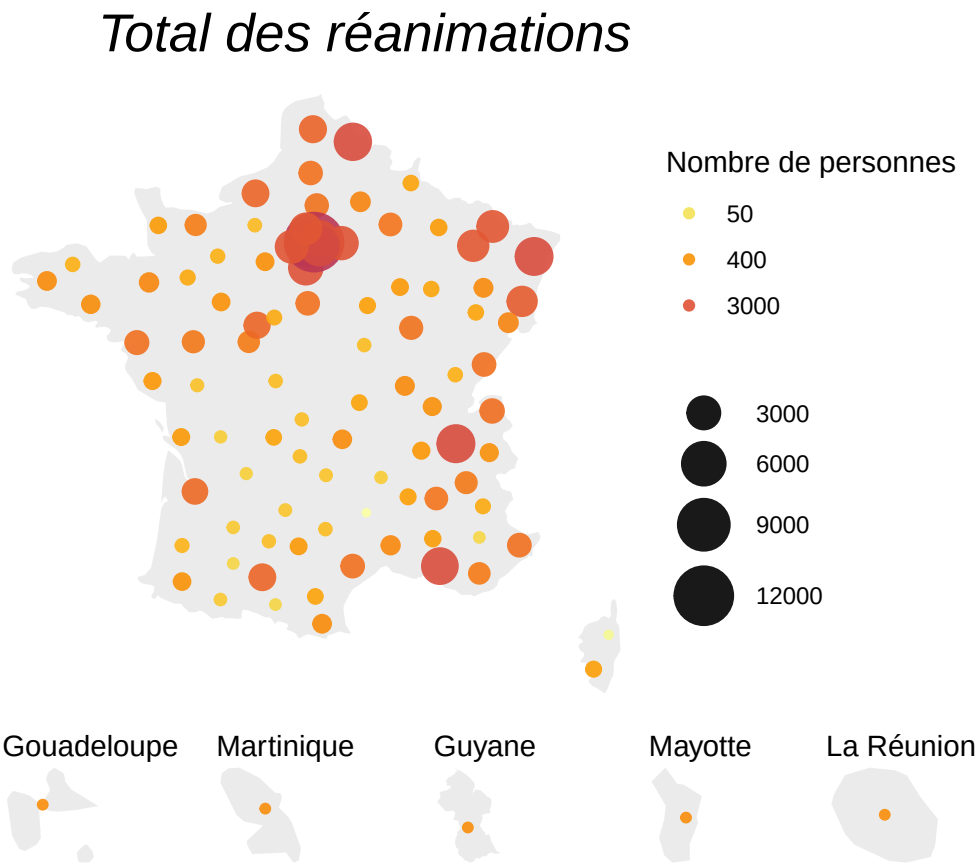
```
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))

# gueris_total
reunion_gueris = ggplot() +
  geom_polygon(data = reu, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data = reunion, aes(x=longitude, y=latitude, size=gueris_total, color=gueris_total), alpha
  scale_size_continuous(range=c(0, 2)) +
  scale_color_viridis(trans="log", option = "inferno", direction = -1, begin = 0.5) +
  labs(subtitle = "La Réunion") +
  theme_void() +
  coord_map() +
  theme(legend.position="none",
        plot.subtitle = element_text(hjust = 0))
```

Dans cette partie, nous allons rassembler les différentes cartes

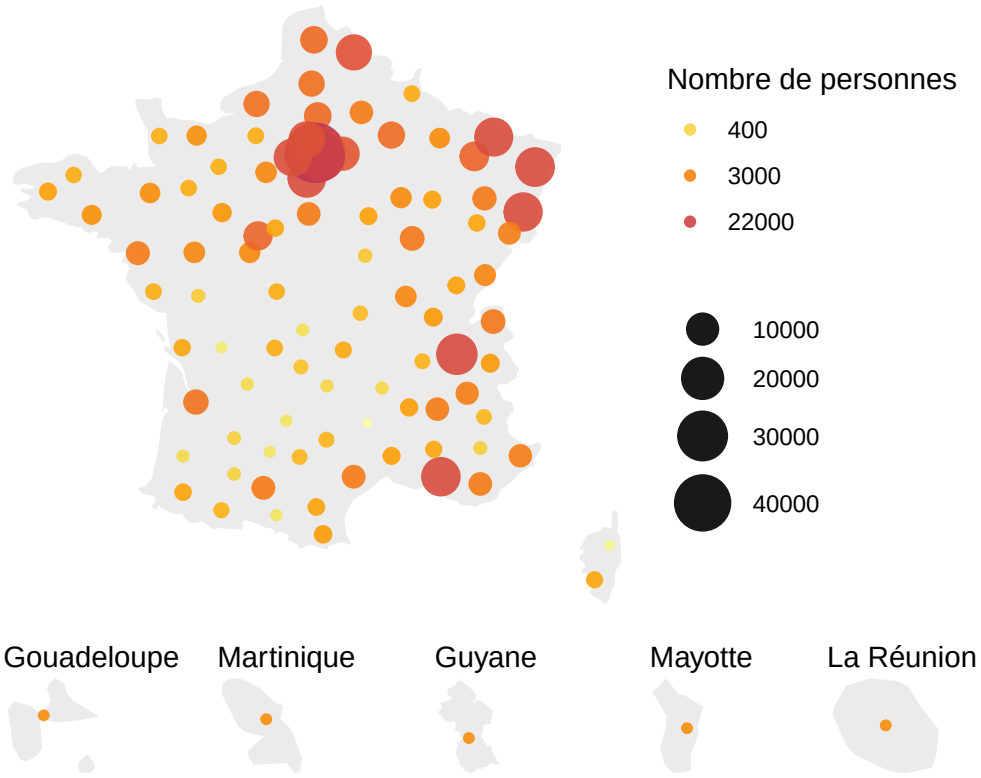
```
# deces_total
suppressWarnings(grid.arrange(guadeloupe_deces, martinique_deces, guyane_deces, mayotte_deces, reunion_
             nrow = 4, heights=c(1.5, 1.5, 1.5, 1),
             layout_matrix = rbind(c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 1, 2, 3, 4, 5, NA)),
             top=textGrob("Total des décès",gp=gpar(fontsize=20,font=3))))
```



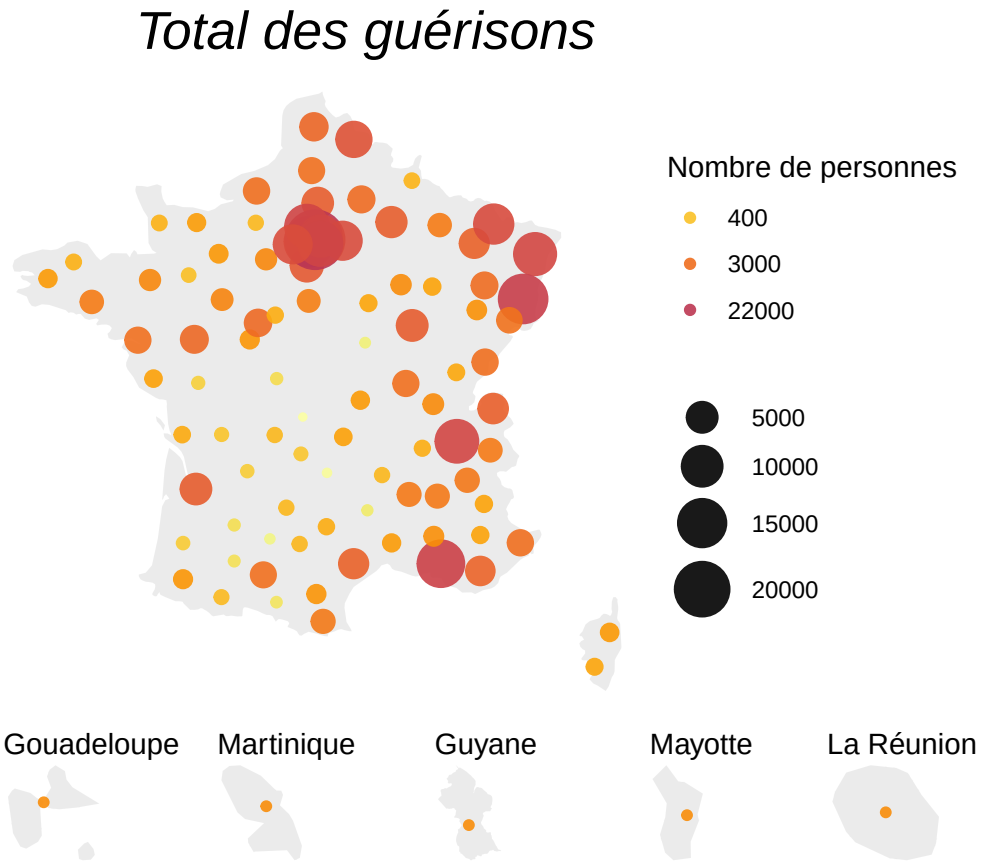
```
# reanimation_total
suppressWarnings(grid.arrange(guadeloupe_reanimation, martinique_reanimation, guyane_reanimation, mayot
             nrow = 4, heights=c(1.5, 1.5, 1.5, 1),
             layout_matrix = rbind(c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 1, 2, 3, 4, 5, NA)),
             top=textGrob("Total des réanimations",gp=gpar(fontsize=20,font=3))))
```




```
# hospitalises_total
suppressWarnings(grid.arrange(guadeloupe_hospitalises, martinique_hospitalises, guyane_hospitalises, ma
             nrow = 4, heights=c(1.5, 1.5, 1.5, 1),
             layout_matrix = rbind(c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 1, 2, 3, 4, 5, NA)),
             top=textGrob("Total des hospitalisations",gp=gpar(fontsize=20,font=3))))
```



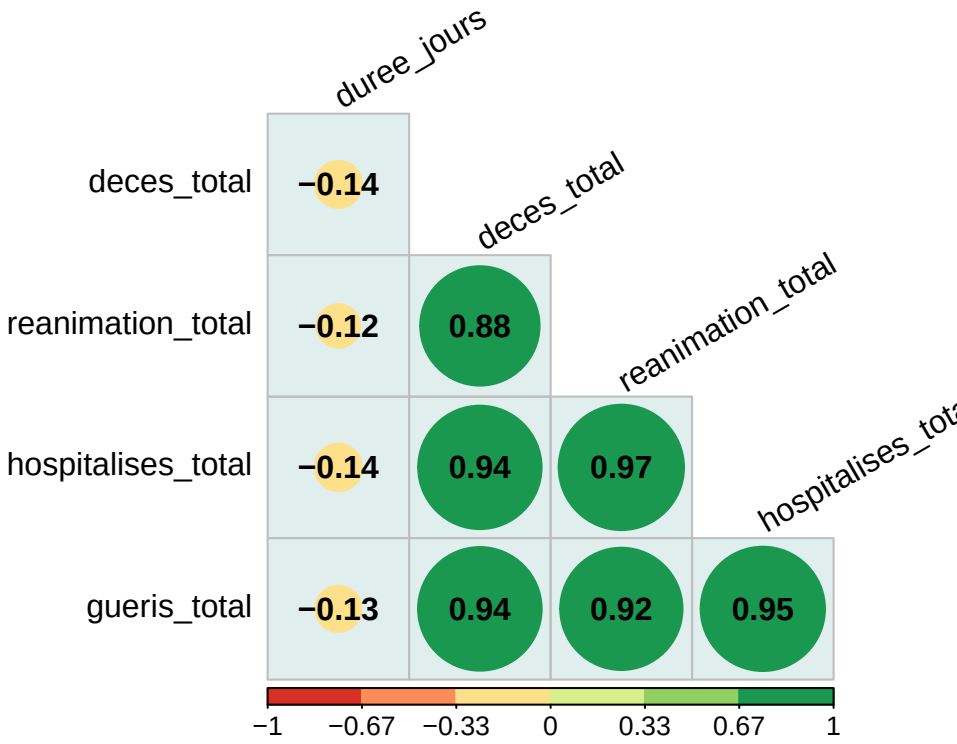

```
# gueris_total
suppressWarnings(grid.arrange(guadeloupe_gueris, martinique_gueris, guyane_gueris, mayotte_gueris, reuni
             nrow = 4, heights=c(1.5, 1.5, 1.5, 1),
             layout_matrix = rbind(c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 6, 6, 6, 6, 6, NA),
                                   c(NA, 1, 2, 3, 4, 5, NA)),
             top=textGrob("Total des guérisons",gp=gpar(fontsize=20,font=3))))
```



Partie 2 : Prédiction du nombre de décès

1

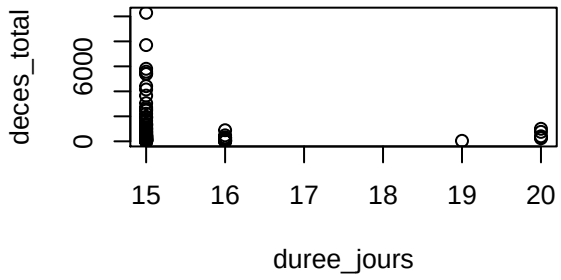
```
cov_cor <- cor(cov[-1])
corrplot(cov_cor,
         method = "circle", type = "lower", diag = FALSE,
         tl.srt = 30, tl.col = "black",
         bg= "azure2", col = brewer.pal(6,"RdYlGn"), addCoef.col = "black")
```

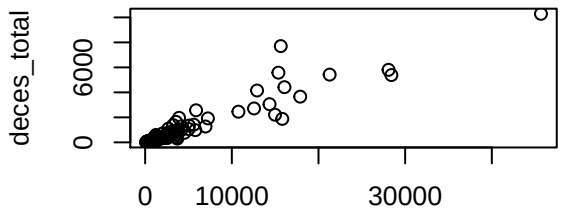


Corrélation des variables entre elles
Toutes les variables sont fortement corrélées entre elles en dehors de duree_jours, comme vu dans la partie 1. Cette corrélation est positive : lorsque l'une d'elle augmente, l'autre aussi.

2

```
par(mfrow=c(2,2))

plot(deces_total ~ duree_jours, data = cov)
plot(deces_total ~ reanimation_total, data = cov)
plot(deces_total ~ hospitalises_total, data = cov)
plot(deces_total ~ gueris_total, data = cov)
```





Nuages de points entre chaque variable et deces_total
Les variables qui vont nous permettre d'expliquer au mieux deces_total sont hospitalises_total et gueris_total.

3

```
sample <- sample.int(n = nrow(cov), size = floor(.80*nrow(cov)))
train <- cov[sample, ]
test  <- cov[-sample, ]
# vérification des dimensions
dim(train)
```

Division du jeu de données en deux ensembles apprentissage/test (80%-20%)

```
## [1] 80  6
```

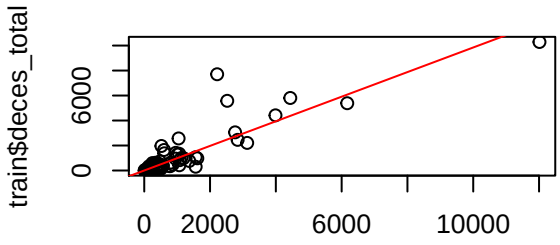
```
dim(test)
```

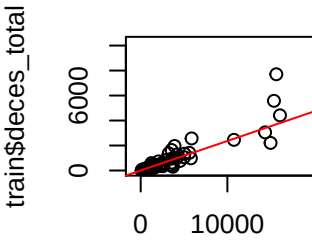
```
## [1] 20  6
```

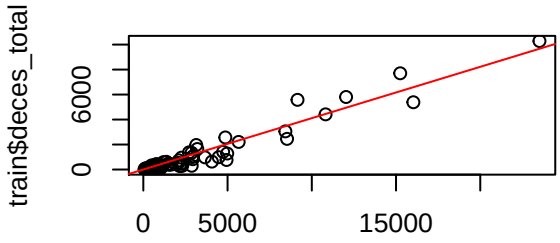
4

```
par(mfrow = c(2, 2))

reg_reanimation <- lm(deces_total ~ -1 + reanimation_total, data=train)
plot(train$reanimation_total, train$deces_total)
```

```
abline(reg_reanimation, col='red')

reg_hospitalises <- lm(deces_total ~ -1 + hospitalises_total, data=train)
plot(train$hospitalises_total, train$deces_total)
abline(reg_hospitalises, col='red')

reg_gueris <- lm(deces_total ~ -1 + gueris_total, data=train)
plot(train$gueris_total, train$deces_total)
abline(reg_gueris, col='red')
```







Régression linéaire pour deces_total

Les points de chaque diagramme forment un nuage élliptique. Le diagramme de gueris_total est celui avec le moins de résidus, et le diagramme de reanimation_total a le plus de résidus.

5

```
print("Coefficients de chaque variable")
```

Analyse des résultats

```
## [1] "Coefficients de chaque variable"
```

```
reg_reanimation$coefficients
```

```
## reanimation_total
##         0.9845781
```

```r
reg_hospitalises$coefficients
```

```
## hospitalises_total
##          0.2362509
```

```r
reg_gueris$coefficients
```

```
## gueris_total
##    0.4110227
```

```r
print(toupper("***** Résultats pour la variable reanimation_total *****"))
```

```
## [1] "***** RÉSULTATS POUR LA VARIABLE REANIMATION_TOTAL *****"
```

```r
summary(reg_reanimation)
```

```
##
## Call:
## lm(formula = deces_total ~ -1 + reanimation_total, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1542.7  -184.1   -23.1    89.8  5521.2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## reanimation_total  0.98458    0.05074   19.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 851.4 on 79 degrees of freedom
## Multiple R-squared:  0.8266, Adjusted R-squared:  0.8244
## F-statistic: 376.5 on 1 and 79 DF,  p-value: < 2.2e-16
```

```r
print(toupper("***** Résultats pour la variable hospitalises_total *****"))
```

```
## [1] "***** RÉSULTATS POUR LA VARIABLE HOSPITALISES_TOTAL *****"
```

```r
summary(reg_hospitalises)
```

```
##
## Call:
## lm(formula = deces_total ~ -1 + hospitalises_total, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1332.2  -119.2   -34.3    42.3  4006.1
##
## Coefficients:
```

```
##                    Estimate Std. Error t value Pr(>|t|)
## hospitalises_total 0.236251   0.008395   28.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 615.8 on 79 degrees of freedom
## Multiple R-squared:  0.9093, Adjusted R-squared:  0.9081
## F-statistic: 791.9 on 1 and 79 DF,  p-value: < 2.2e-16
```

```
print(toupper("***** Résultats pour la variable gueris_total *****"))
```

```
## [1] "***** RÉSULTATS POUR LA VARIABLE GUERIS_TOTAL *****"
```

```
summary(reg_gueris)
```

```
##
## Call:
## lm(formula = deces_total ~ -1 + gueris_total, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1267.86  -271.84   -89.46    -1.26  1814.91
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## gueris_total   0.4110     0.0112    36.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 481.2 on 79 degrees of freedom
## Multiple R-squared:  0.9446, Adjusted R-squared:  0.9439
## F-statistic:  1347 on 1 and 79 DF,  p-value: < 2.2e-16
```

Le coefficient de régression de reanimation_total est le plus élevé, et celui de hospitalises_total est le moins élevé. La variable gueris_total est celle avec le moins d'erreur résiduelle.

6

```
# Train
print("RÉANIMATIONS")
```

Mean Absolute Error et Mean Squared Error

```
## [1] "RÉANIMATIONS"
```

```
print("apprentissage")
```

```
## [1] "apprentissage"
```

```
rea_y_est_train = predict(reg_reanimation)
MAE_rea = MAE(train$deces_total, rea_y_est_train)
MSE_rea = MSE(train$deces_total, rea_y_est_train)
MAE_rea + MSE_rea
```

```
## [1] 716269.4
```

```
print("test")
```

```
## [1] "test"
```

```
rea_y_est_test = predict(reg_reanimation , test)
MAE_rea_test = MAE(test$deces_total, rea_y_est_test )
MSE_rea_test = MSE(test$deces_total, rea_y_est_test )
MAE_rea_test + MSE_rea_test
```

```
## [1] 599108.9
```

```
print("HOSPITALISATIONS")
```

```
## [1] "HOSPITALISATIONS"
```

```
print("apprentissage")
```

```
## [1] "apprentissage"
```

```
hospi_y_est_train = predict(reg_hospitalises)
MAE_hos = MAE(train$deces_total,hospi_y_est_train)
MSE_hos = MSE(train$deces_total, hospi_y_est_train)
MAE_hos + MSE_hos
```

```
## [1] 374731.1
```

```
print("test")
```

```
## [1] "test"
```

```
hospi_y_est_test = predict(reg_hospitalises, test)
MAE_hos_test = MAE(test$deces_total,hospi_y_est_test )
MSE_hos_test = MSE(test$deces_total,hospi_y_est_test )
MAE_hos_test + MSE_hos_test
```

```
## [1] 286146.6
```

```
print("GUÉRISONS")
```

```
## [1] "GUÉRISONS"
```

```r
print("apprentissage")
```

```
## [1] "apprentissage"
```

```r
guer_y_est_train = predict(reg_gueris)
MAE_gue = MAE(train$deces_total, guer_y_est_train)
MSE_gue = MSE(train$deces_total, guer_y_est_train)
MAE_gue + MSE_gue
```

```
## [1] 228971
```

```r
print("test")
```

```
## [1] "test"
```

```r
guer_y_est_test = predict(reg_gueris, test)
MAE_gue_test = MAE(test$deces_total, guer_y_est_test )
MSE_gue_test = MSE(test$deces_total, guer_y_est_test )
MAE_gue_test + MSE_gue_test
```

```
## [1] 940462.2
```

7

Comparaison des MAE et MSE C'est avec la variable hospitalises_total que les erreurs sont les plus faibles. Les erreurs de l'ensemble de test sont sistématiquement plus faibles que celles de l'ensemble d'apprentissage, ce qui est positif. La différence est flagrante avec la variable reanimation_total. #### Précision des tests

```r
# Création des jeux de données selon chaque test
rea = cbind(test[, c(1, 3)], rea_y_est_test)
hospi = cbind(test[, c(1, 3)], hospi_y_est_test)
guer = cbind(test[, c(1, 3)], guer_y_est_test)

# Calcul du nombre de prédictions justes (en arrondissant à la centaine supérieure)
rea_trouve = 0
hospi_trouve = 0
guer_trouve = 0
for (i in 1:20){
  rea[i,2] = ceiling(rea[i, 2]/100) * 100
  rea[i,3] = ceiling(rea[i, 3]/100) * 100
  if (rea[i,2] == rea[i,3])
    rea_trouve = rea_trouve + 1
  hospi[i,2] = ceiling(hospi[i, 2]/100) * 100
  hospi[i,3] = ceiling(hospi[i, 3]/100) * 100
  if (hospi[i,2] == hospi[i,3])
    hospi_trouve = hospi_trouve + 1
  guer[i,2] = ceiling(guer[i, 2]/100) * 100
  guer[i,3] = ceiling(guer[i, 3]/100) * 100
  if (guer[i,2] == guer[i,3])
```

```
    guer_trouve = guer_trouve + 1
}

# Calcul de la précision pour chaque test
rea_precision = rea_trouve / 20
hospi_precision = hospi_trouve / 20
guer_precision = guer_trouve / 20

# Affichage des résultats
print("Précision pour les prédictions avec reanimation_total :")
```

```
## [1] "Précision pour les prédictions avec reanimation_total :"
```

```
rea_precision
```

```
## [1] 0
```

```
print("Précision pour les prédictions avec hospitalises_total :")
```

```
## [1] "Précision pour les prédictions avec hospitalises_total :"
```

```
hospi_precision
```

```
## [1] 0.25
```

```
print("Précision pour les prédictions avec gueris_total :")
```

```
## [1] "Précision pour les prédictions avec gueris_total :"
```

```
guer_precision
```

```
## [1] 0.1
```

La précision de la variable reanimation_total est celle qui est la plus élevée. C'est donc celle qu'il convient d'utiliser. Cela fait sens, étant donné que la plupart des personnes étant en réanimation décèdent.

Partie 3: Clustering des départements selon la dynamique de propagation du virus

1

```
pca = PCA(cov[-1],graph = FALSE, scale.unit = TRUE)

eigen_val = get_eigenvalue(pca)
eigen_val
```

Réalisation d'une ACP

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 3.82582812       76.5165624                     76.51656
## Dim.2 0.97614484       19.5228967                     96.03946
## Dim.3 0.12883768        2.5767535                     98.61621
## Dim.4 0.05164074        1.0328149                     99.64903
## Dim.5 0.01754862        0.3509725                    100.00000
```

```
dimensions = dim(eigen_val)[1]

dim1 = eigen_val[1]
dim2 = eigen_val[2]

dim1_val = round(100 * dim1 / 5 , 1)
dim2_val = round(100 * dim2 / 5 , 1)
dim1_val
```

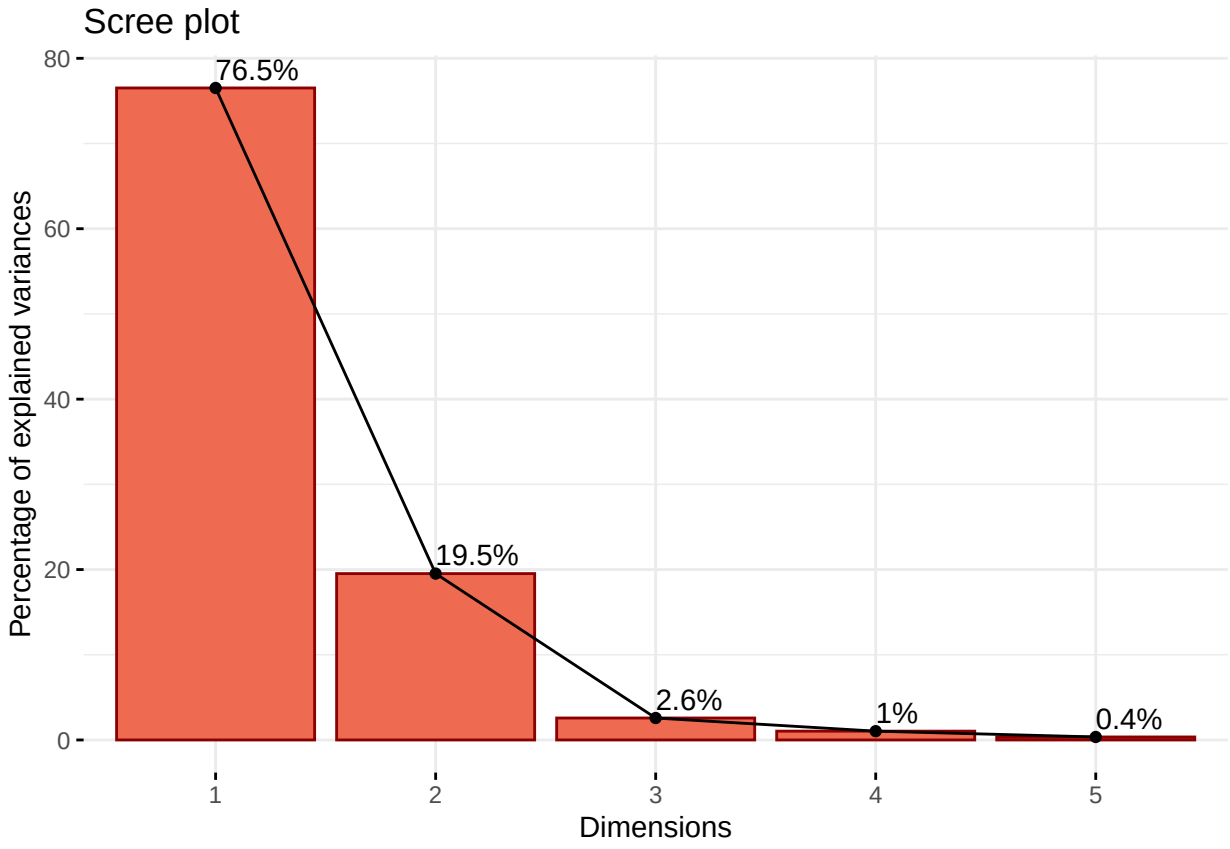
```
## [1] 76.5
```

```
dim2_val
```

```
## [1] 19.5
```

La variation est expliquée par la dimension 1 à environ 76.5% et par la dimension 2 à environ 19.5% (soit environ 96% pour les deux). On vérifie cela tout simplement avec le diagramme ci-dessous : il présente un coude après la dimension 2.

```
fviz_eig(pca, addlabels = TRUE, barfill = "coral2", barcolor = "darkred")
```



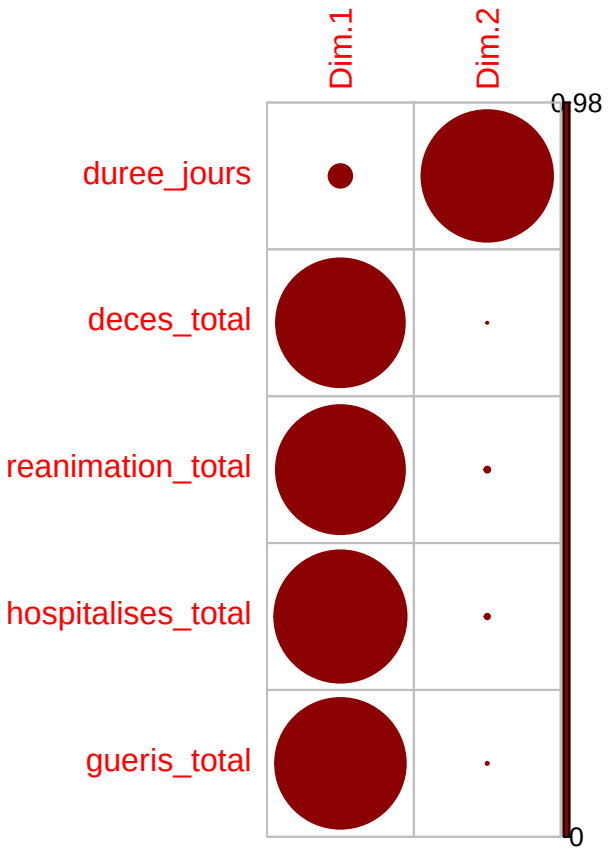
Étude des variables

```
variables = get_pca_var(pca)
variables$contrib[, c(1, 2)]
```

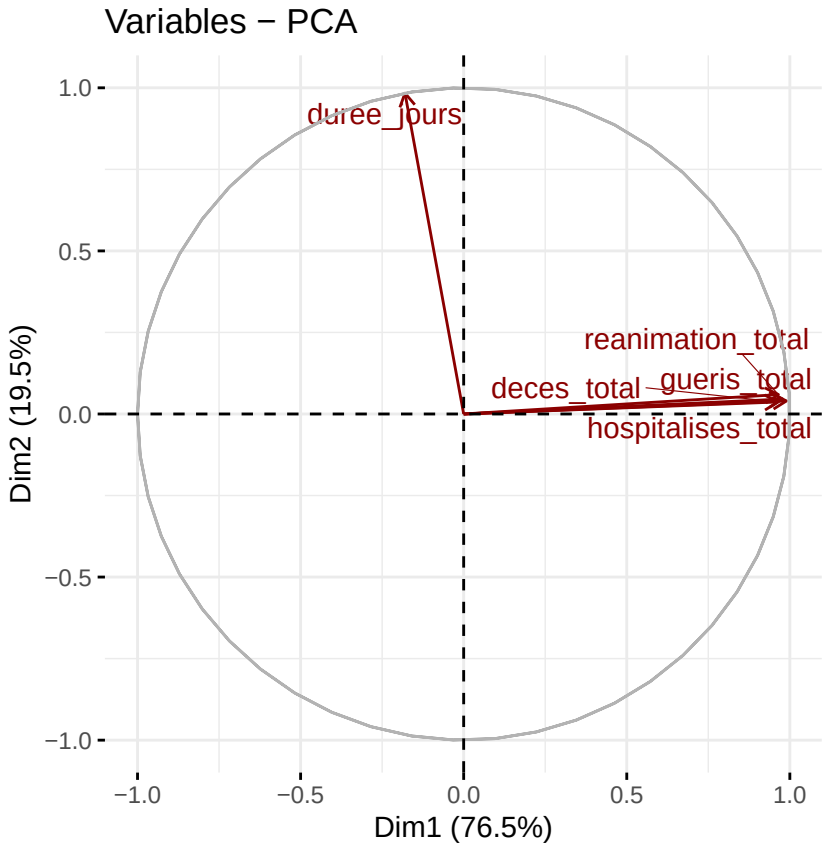
```
##                          Dim.1      Dim.2
## duree_jours          0.8470359 99.1211173
## deces_total         24.2533318  0.1385958
## reanimation_total   24.3398011  0.3619282
## hospitalises_total  25.5817181  0.1631504
## gueris_total        24.9781131  0.2152082
```

Les variables contribuant le plus à la dimension 1 sont reanimation_total puis gueris_total, suivies de près par deces_total et hospitalises_total. La participation de duree_jours est très faible. Pour la dimension 2, c'est tout l'inverse : duree_jours a une participation quasiment totale, et toutes les autres variables ont une participation extrêmement faible.

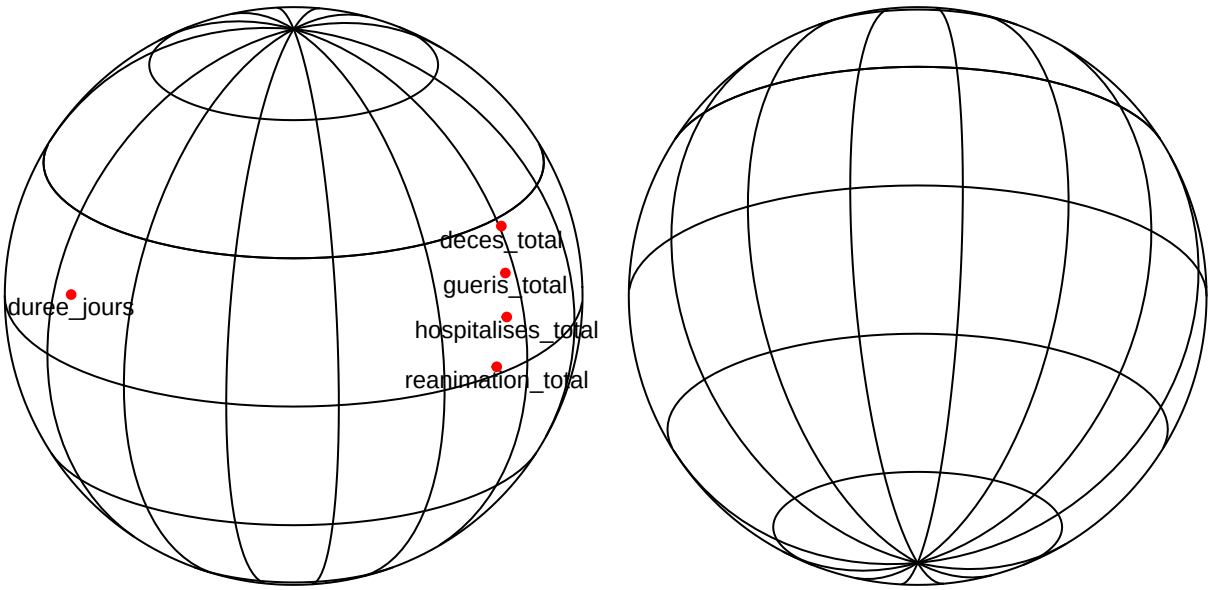
```
corrplot(variables$cos2[, c(1, 2)],is.corr = FALSE, col = "darkred")
```




```
fviz_pca_var(pca, col.var = "darkred", repel = TRUE)
```

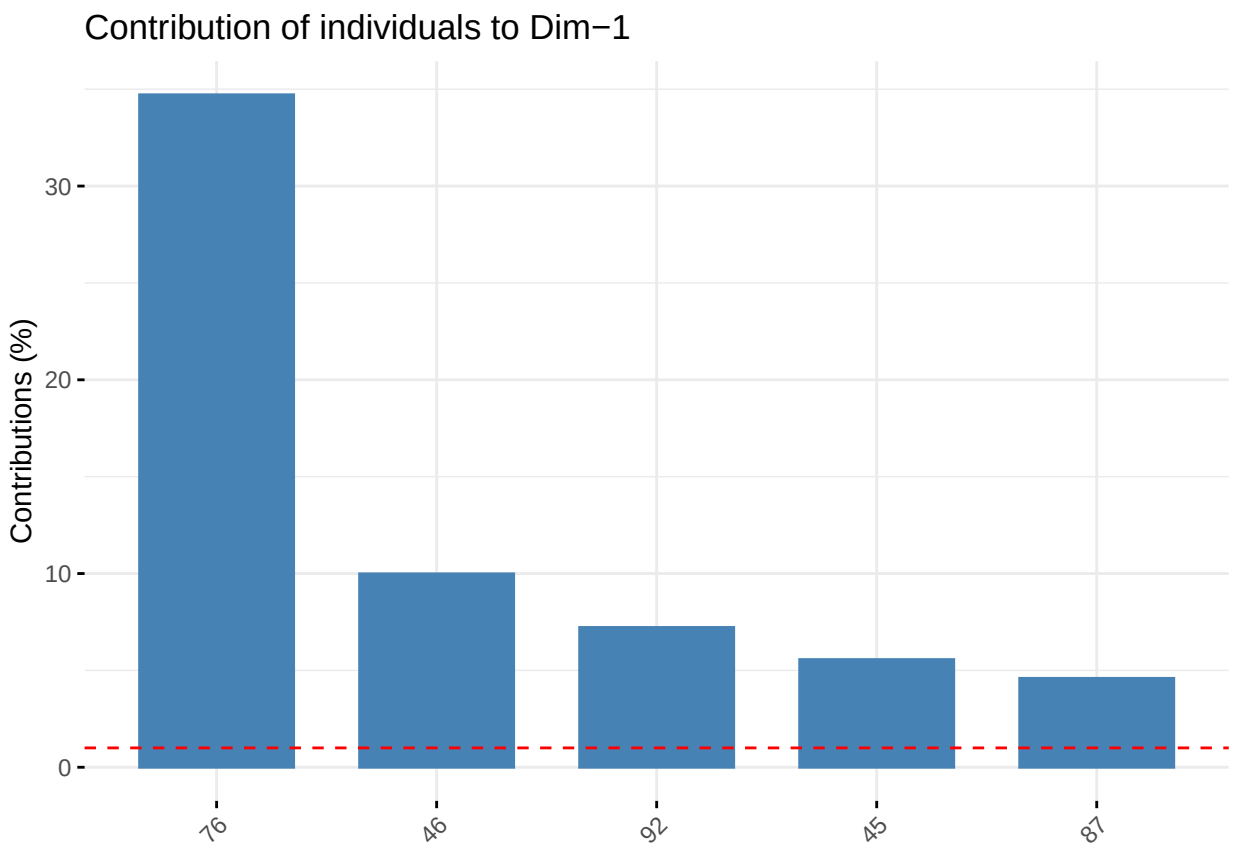



```
sphpca(cov[-1], v=130)
```

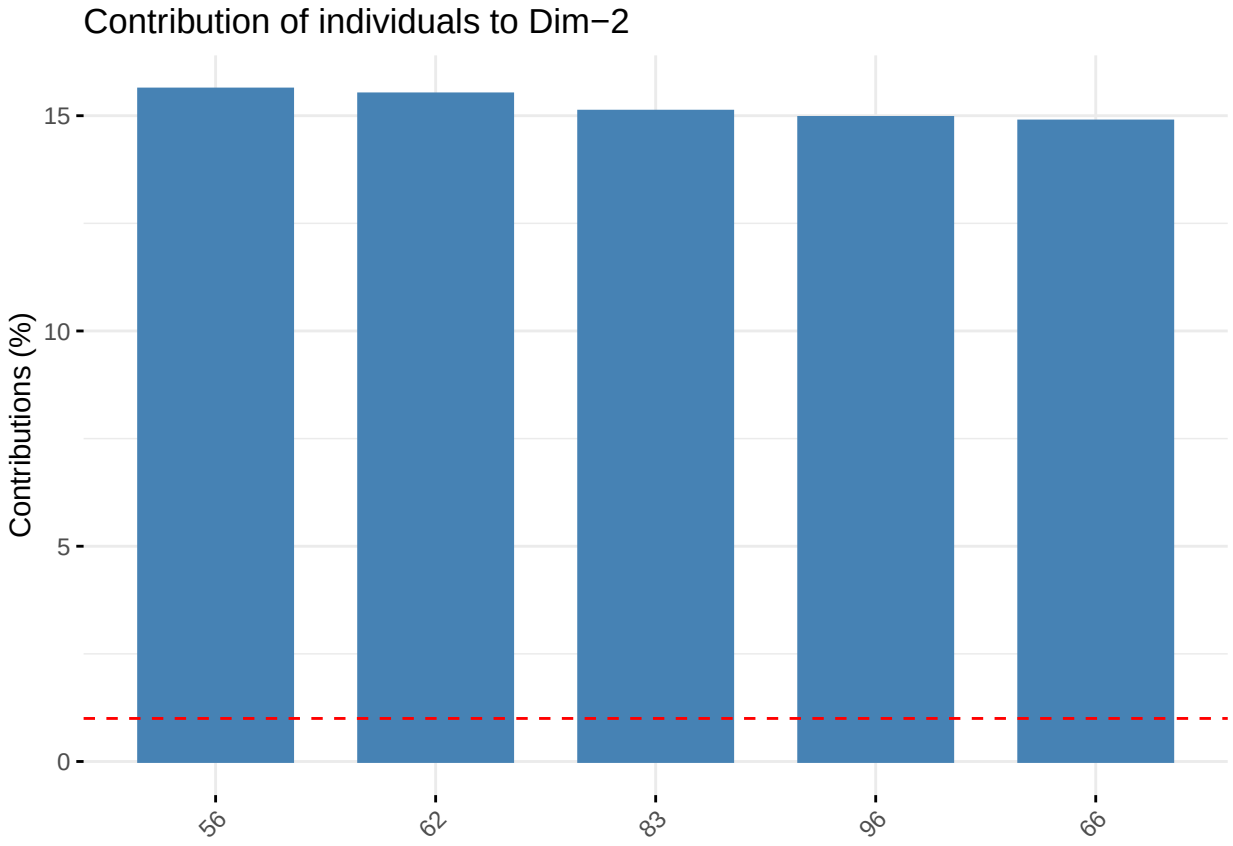


Nous observons ici les corrélations entre les variables. Le diagramme de corrélation confirme ce que nous avons observé pour les contributions. Les deux vues du graphique des variables nous permettent d'appuyer ceci. En effet, duree_jours se trouve dans un angle à environ 90° des autres variables, ce qui signifie que ces deux groupes ne sont pas corrélés. Les autres variables, groupées, montrent une forte corrélation entre elles. ### 2 #### Étude des individus

```
fviz_contrib(pca, choice = "ind", axes = 1, top = 5)
```




```
fviz_contrib(pca, choice = "ind", axes = 2, top = 5)
```




```
top_individus_dim1 = as.vector(cov[c(76, 46, 92, 45, 87),]$maille_nom)
top_individus_dim1
```

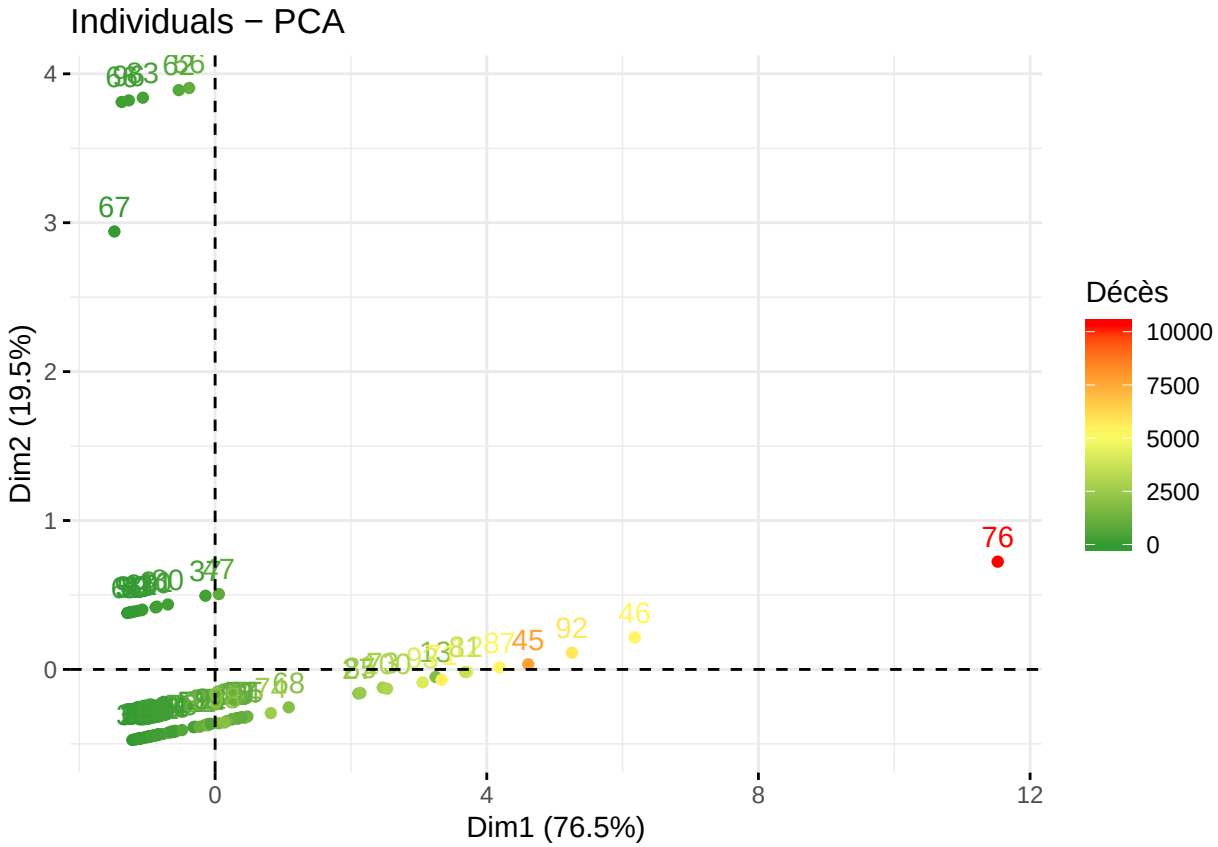
```
## [1] "Paris"              "Hauts-de-Seine"    "Val-de-Marne"
## [4] "Haut-Rhin"          "Seine-Saint-Denis"
```

```
top_individus_dim2 = as.vector(cov[c(56, 62, 83, 96, 66),]$maille_nom)
top_individus_dim2
```

```
## [1] "Loire-Atlantique" "Maine-et-Loire"   "Sarthe"           "Vendée"
## [5] "Mayenne"
```

Les 5 individus contribuant le plus à la dimension 1 sont Paris, les Hauts-de-Seine, le Val-de-Marne, le Haut-Rhin et la Seine-Saint-Denis. Les 5 contribuant le plus à la dimension 2 sont la Loire-Atlantique, le Maine-et-Loire, la Sarthe, la Vendée et la Mayenne.

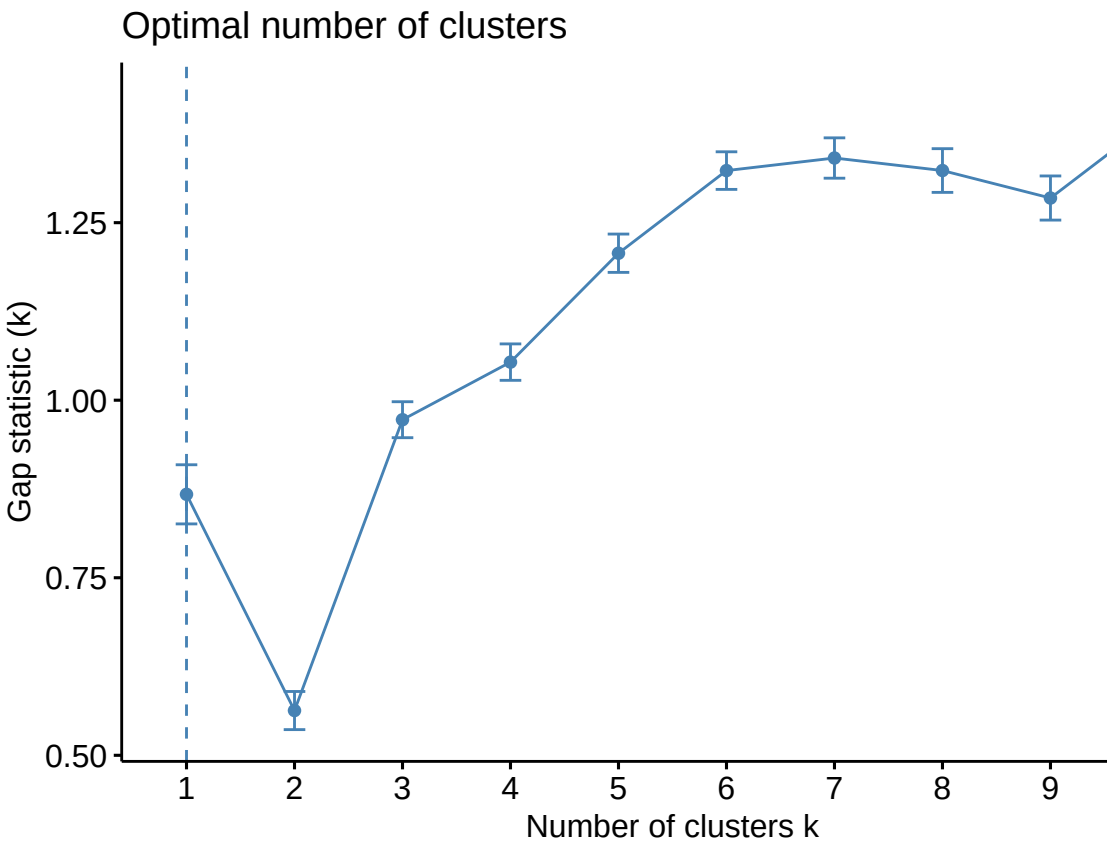
```
fviz_pca_ind (pca, col.ind = cov$deces_total,
              gradient.cols = c("#339933", "#FFFF66", "#ff0000"),
              legend.title = "Décès")
```



Le nuage de points est assez dispersé, cela veut dire qu'il y a une forte variabilité. Sur l'axe horizontal, les individus proches ont des comportement similaires, ceux qui sont éloignés ont des comportements différents : nombre de décès et autre différents, comme nous avons effectivement pu le voir sur les cartes de la France. La position verticale dépend uniquement de durée_jours.

3

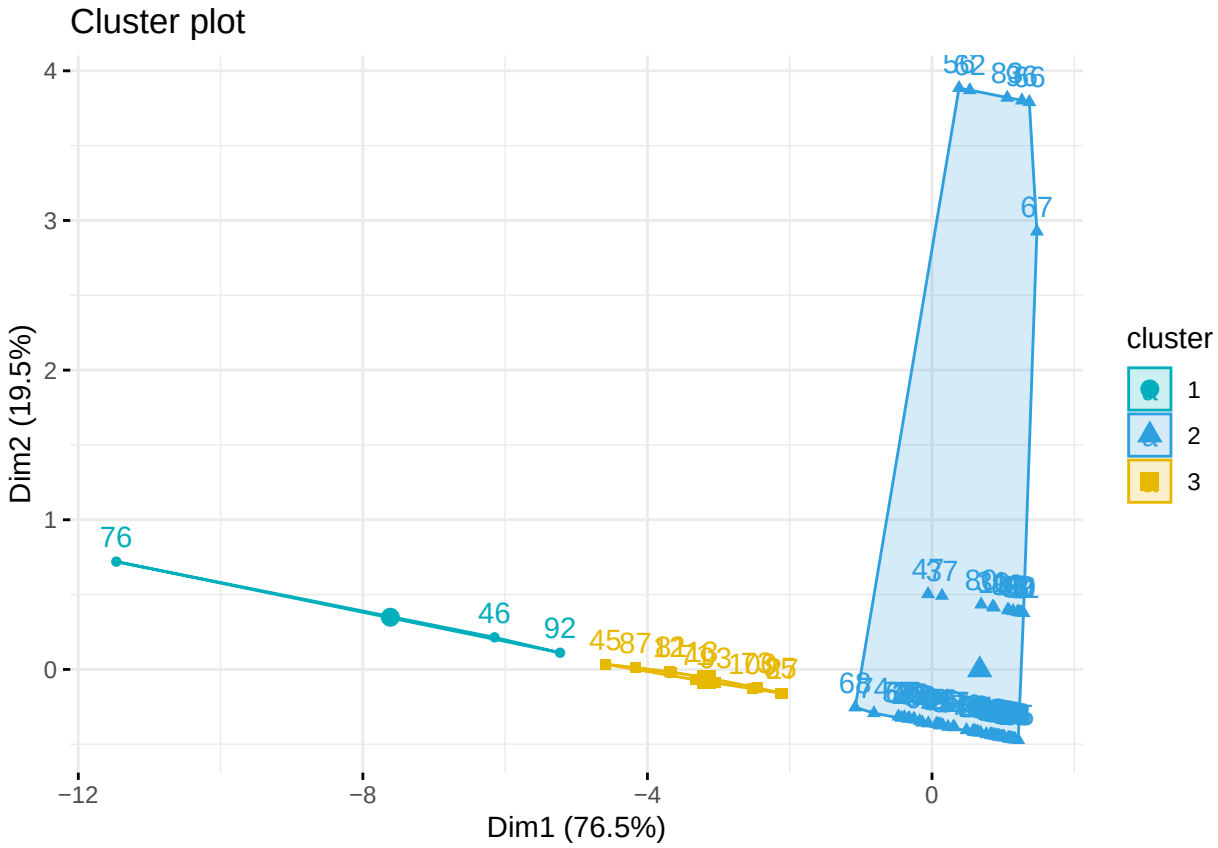
```
fviz_nbclust(scale(cov[, -1]), kmeans, method = "gap_stat")
```



Classification K-means

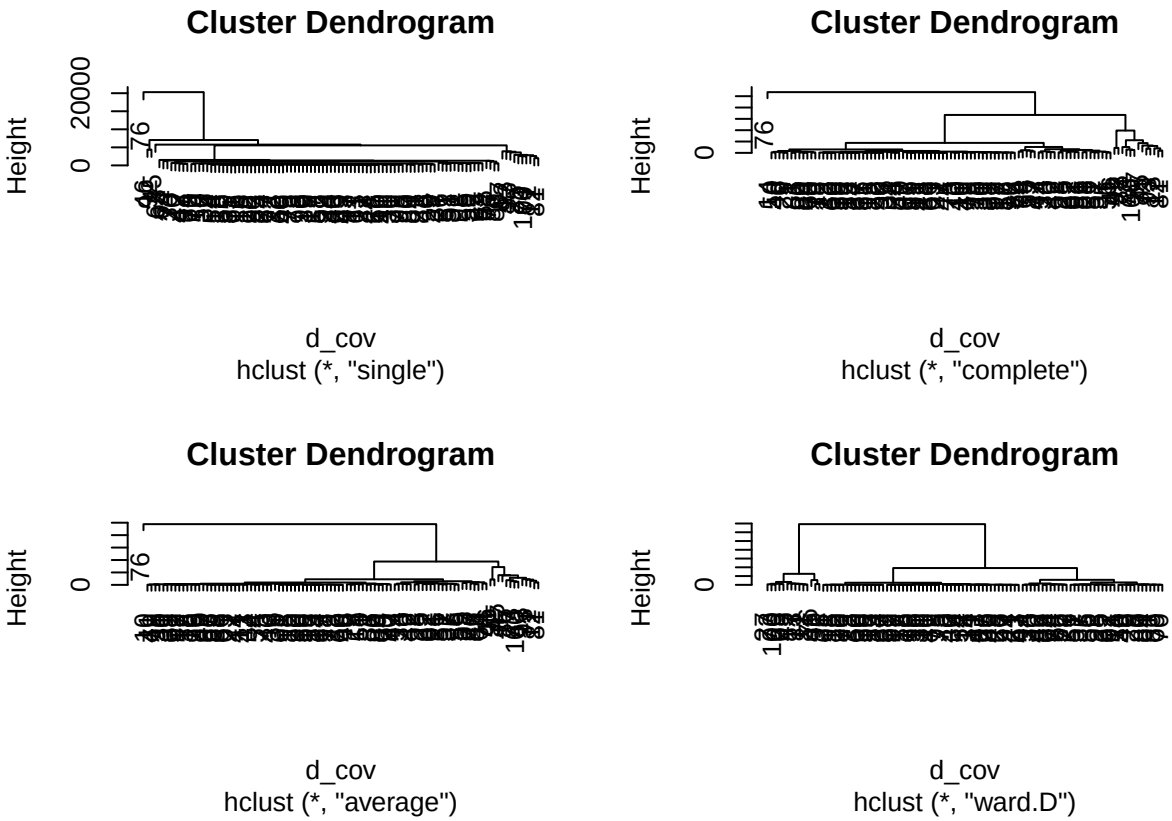
Je choisis de créer 3 clusters, car 1 et 2 n'ont pas de sens/d'intérêt, et au dessus de 3 la lisibilité risque d'être amoindrie. On observe un coude entre 2 et 3.

```
set.seed(123)
k_cov <- kmeans(cov[, -1], 3, nstart = 10)

plot_k_means = fviz_cluster(k_cov, data = cov[, -1],
             palette = c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"),
             ggtheme = theme_minimal())
plot(plot_k_means)
```



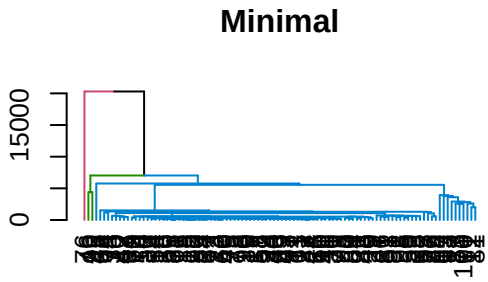
Les trois groupes font sens : nous avons ceux qui sont rouges/oranges (cluster 1), ceux qui sont jaune/vert clair (cluster 3), et ceux qui sont vert (cluster 2). Je valide donc le nombre 3. #### Classification CAH

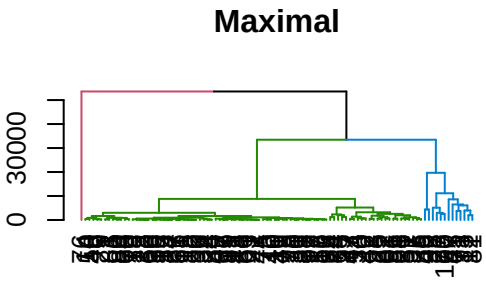
```
d_cov = dist(cov[, -c(1, 2)], method = "euclidean")

par(mfrow = c(2, 2))
# lien minimum
h_mini = hclust(d_cov, method = "single")
plot(h_mini)

# lien maximum
h_maxi = hclust(d_cov, method = "complete")
plot(h_maxi)

# lien moyen
h_moy = hclust(d_cov, method = "average")
plot(h_moy)

# lien ward
h_ward = hclust(d_cov, method = "ward.D")
plot(h_ward)
```

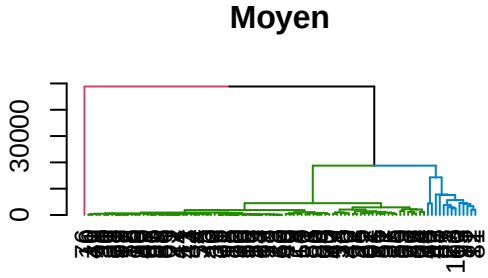


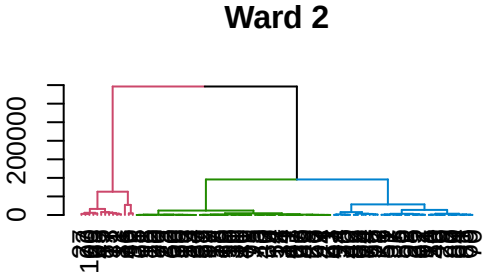
Le dendogramme avec le lien ward est celui qui semble faire les groupes les plus homogènes. Ceci est confirmé avec le coloriage des branches : il n'y a que le lien ward qui ne met pas l'individu Paris seul dans un cluster.

```
par(mfrow = c(2, 2))
# lien minimum
c_mini = cutree(h_mini, 3)
dend_mini <- as.dendrogram(h_mini)
dend <- color_branches(dend_mini, k = 3)
plot(dend, main = "Minimal")

# lien maximum
c_maxi = cutree(h_maxi, 3)
dend_maxi <- as.dendrogram(h_maxi)
dend <- color_branches(dend_maxi, k = 3)
plot(dend, main = "Maximal")

# lien moyen
c_moy = cutree(h_moy, 3)
dend_moy <- as.dendrogram(h_moy)
dend <- color_branches(dend_moy, k = 3)
plot(dend, main = "Moyen")

# lien ward
c_ward = cutree(h_ward, 3)
dend_ward <- as.dendrogram(h_ward)
dend <- color_branches(dend_ward, k = 3)
plot(dend, main = "Ward 2")
```









4

Comparaison des algorithmes

```
print("Kmeans")
```

```
## [1] "Kmeans"
```

```
table(grp2 = k_cov$cluster, grp4 = k_cov$cluster)
```

```
##     grp4
## grp2  1  2  3
##    1  3  0  0
##    2  0 86  0
##    3  0  0 11
```

```
print("Minimal")
```

```
## [1] "Minimal"
```

```
table(grp2 = c_mini, grp4 = c_mini)
```

```
##     grp4
## grp2  1  2  3
##    1 97  0  0
##    2  0  2  0
##    3  0  0  1
```

```
print("Maximal")
```

```
## [1] "Maximal"
```

```
table(grp2 = c_maxi, grp4 = c_maxi)
```

```
##     grp4
## grp2  1  2  3
##    1 86  0  0
##    2  0 13  0
##    3  0  0  1
```

```
print("Moyen")
```

```
## [1] "Moyen"
```

```
table(grp2 = c_moy, grp4 = c_moy)
```

```
##     grp4
## grp2  1  2  3
##    1 86  0  0
##    2  0 13  0
##    3  0  0  1
```

```
print("Ward")
```

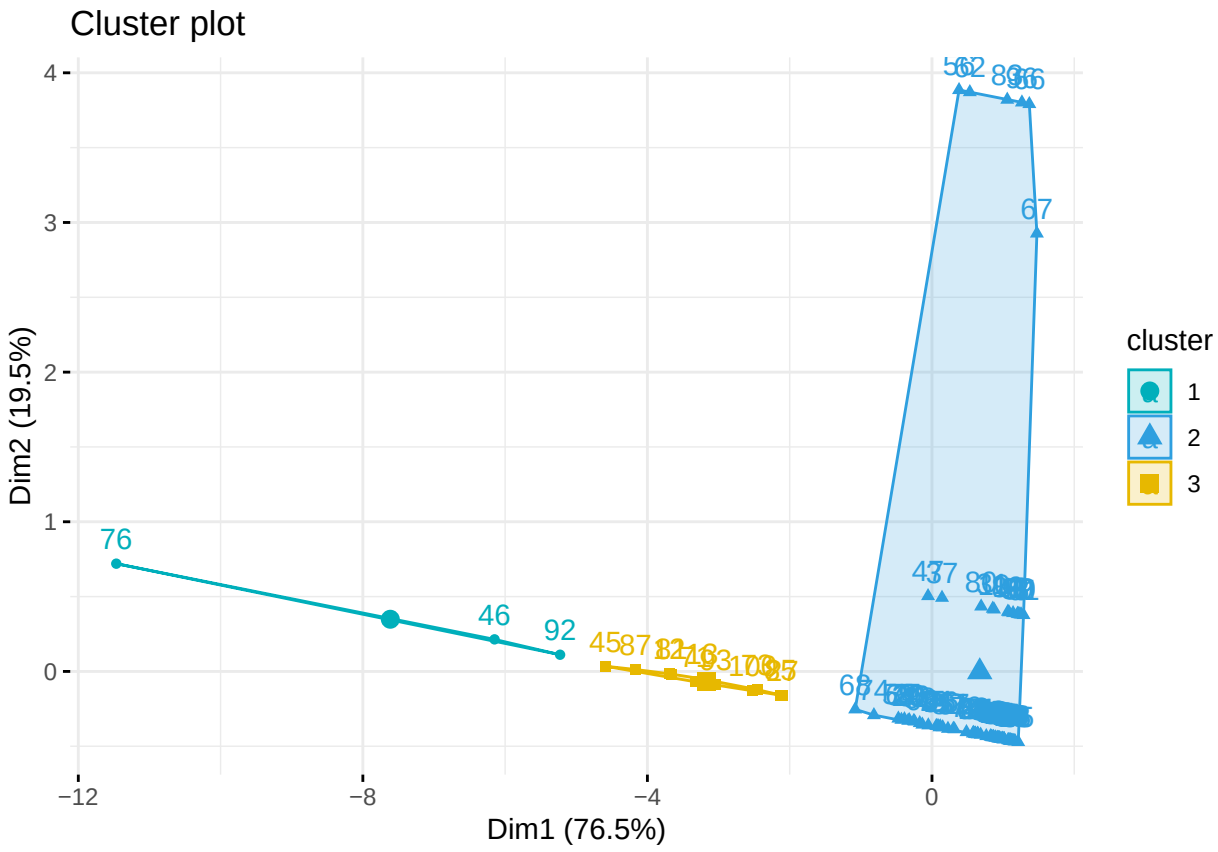
```
## [1] "Ward"
```

```
table(grp2 = c_ward, grp4 = c_ward)
```

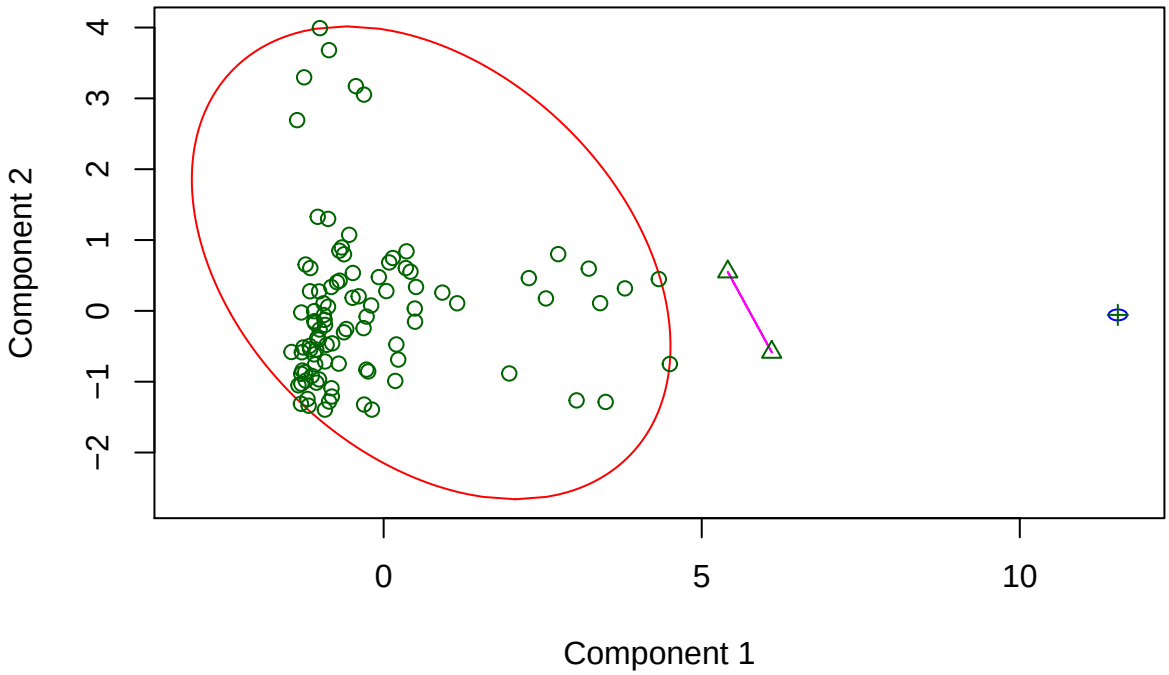
```
##     grp4
## grp2  1  2  3
##    1 36  0  0
##    2  0 50  0
##    3  0  0 14
```

Nous pouvons voir avec ces tables de confusion que les algorithmes K-means et ward sont ceux qui font les groupes les plus homogènes. Ceci se vérifie avec les graphes suivants : on observe aussi que malgré une meilleure répartition avec K-means et ward, c'est l'algorithme K-means qui a fait les groupes les plus cohérents

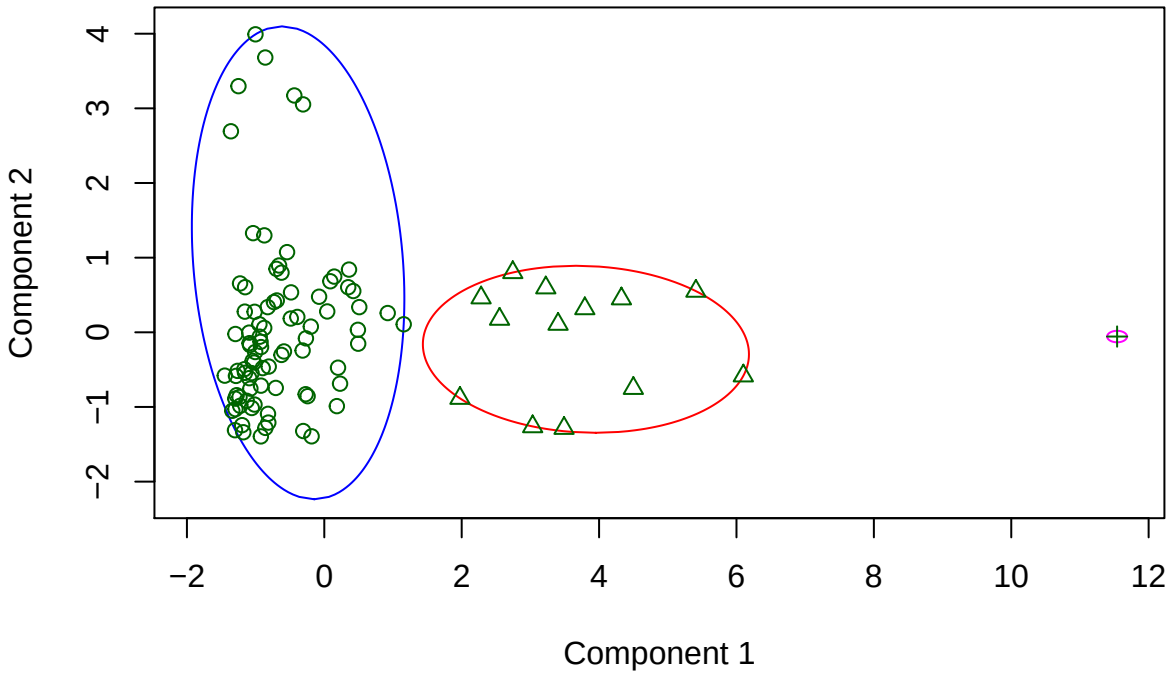
```
plot(plot_k_means)
```




```
clusplot(cov, c_mini, main='Minimum',
         color=TRUE,
         shade=FALSE, lines = 0)
```

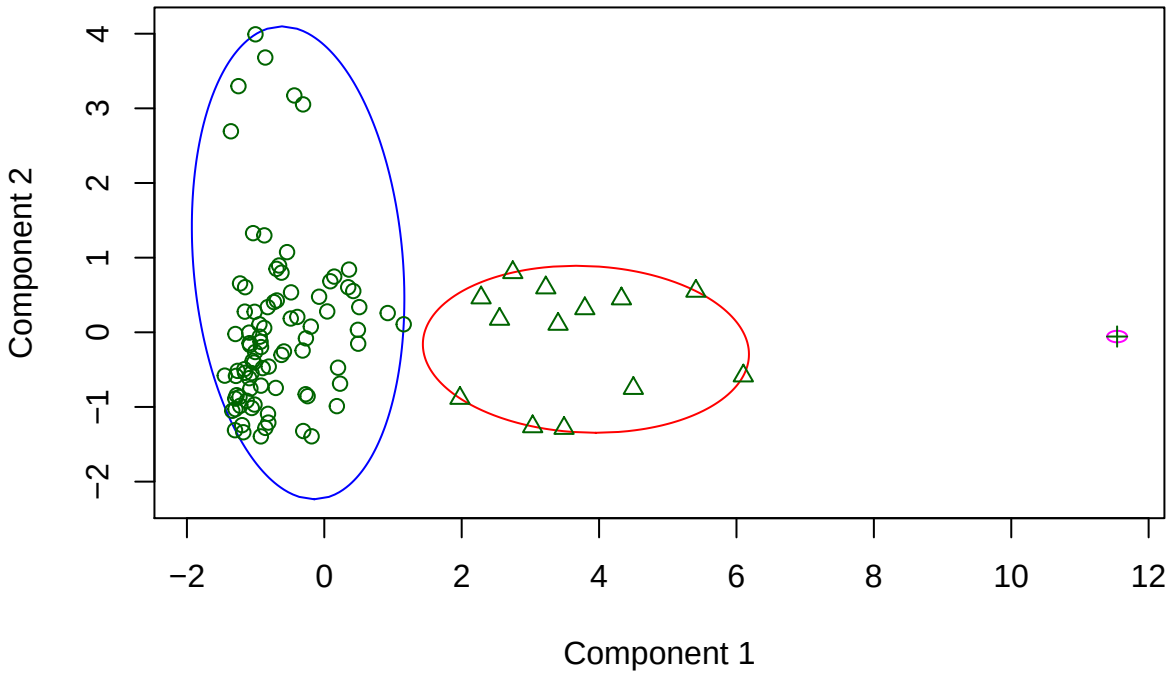



```
clusplot(cov, c_maxi, main='Maximum',
         color=TRUE,
         shade=FALSE, lines = 0)
```

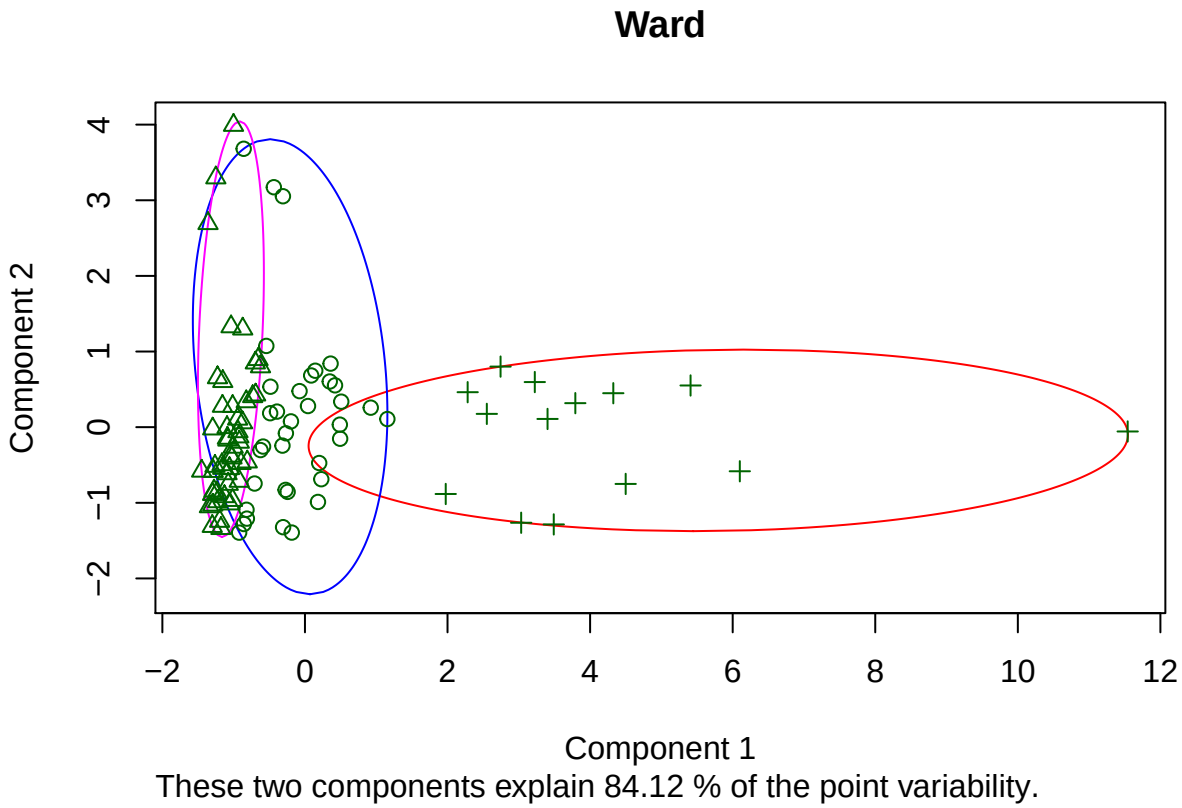


These two components explain 84.12 % of the point variability.

```
clusplot(cov, c_moy, main='Moyen',
         color=TRUE,
         shade=FALSE, lines = 0)
```




```
clusplot(cov, c_ward, main='Ward',
         color=TRUE,
         shade=FALSE, lines = 0)
```



5

Déductions La dynamique de propagation semble liée à la géographie (comme vu dans la partie 1), et est aussi différente selon les départements. Nous avons 3 groupes de départements dans lesquels le nombre de personnes guéries, atteintes et décédées sont similaires. Les départements les plus lourdement touchés sont ceux qui contribuent le plus à la dimension 1, c'est-à-dire au nombre de contaminations, décès, etc. Cela est tout à fait logique. Nous avons les département les plus gravement touchés, ceux qui sont moyennement touchés, et ceux qui sont le moins touchés, avec une distribution assez homogène.

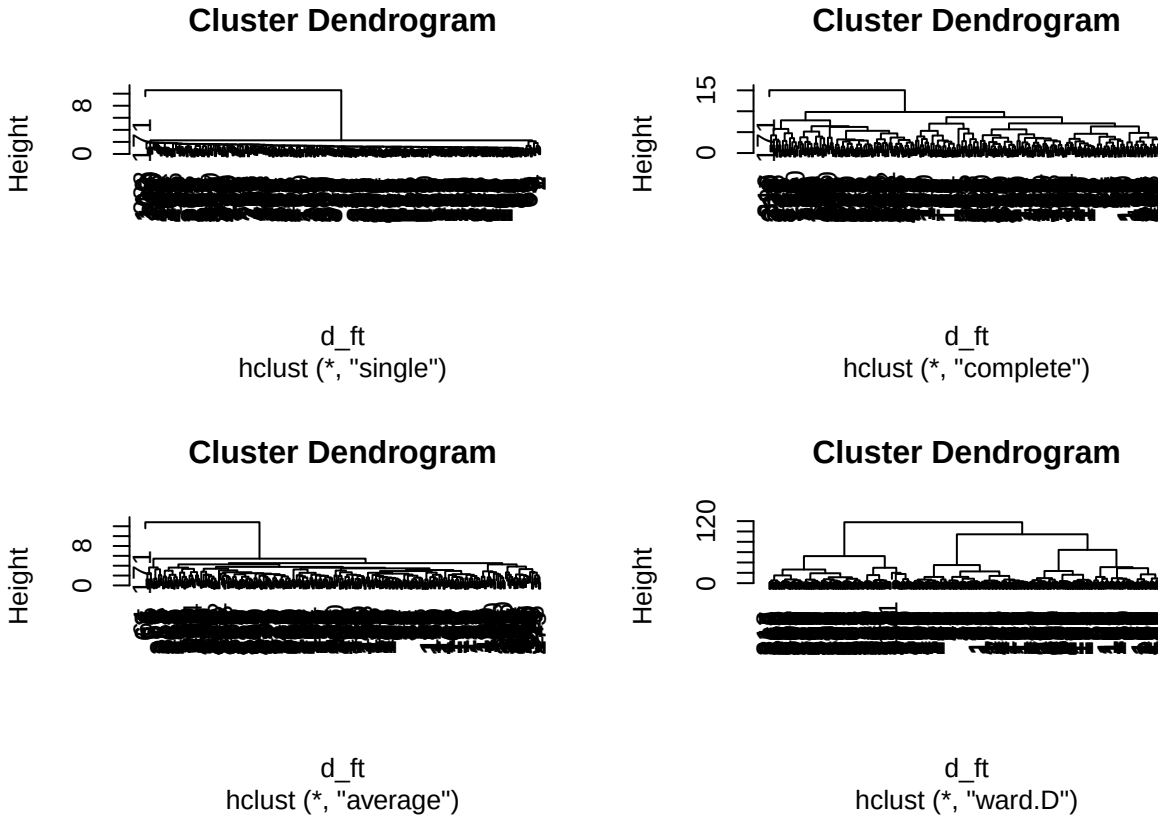
Fatality

```
fatality = read.csv("Fatality.csv", sep = ",")
ft = scale(fatality[, -c(7, 8)])
head(fatality)
```

Préparation des données

```
##   X state year   mrall  beertax  mlda jaild comserd   vmiles unrate   perinc
## 1 1     1 1982 2.12836 1.539379 19.00    no      no 7.233887   14.4 10544.15
## 2 2     1 1983 2.34848 1.788991 19.00    no      no 7.836348   13.7 10732.80
## 3 3     1 1984 2.33643 1.714286 19.00    no      no 8.262990   11.1 11108.79
## 4 4     1 1985 2.19348 1.652542 19.67    no      no 8.726917    8.9 11332.63
## 5 5     1 1986 2.66914 1.609907 21.00    no      no 8.952854    9.8 11661.51
## 6 6     1 1987 2.71859 1.560000 21.00    no      no 9.166302    7.8 11944.00
```

```r
d_ft = dist(ft, method = "euclidean")

par(mfrow = c(2, 2))
# lien minimum
h_mini = hclust(d_ft, method = "single")
plot(h_mini)

# lien maximum
h_maxi = hclust(d_ft, method = "complete")
plot(h_maxi)

# lien moyen
h_moy = hclust(d_ft, method = "average")
plot(h_moy)

# lien ward
h_ward = hclust(d_ft, method = "ward.D")
plot(h_ward)
```

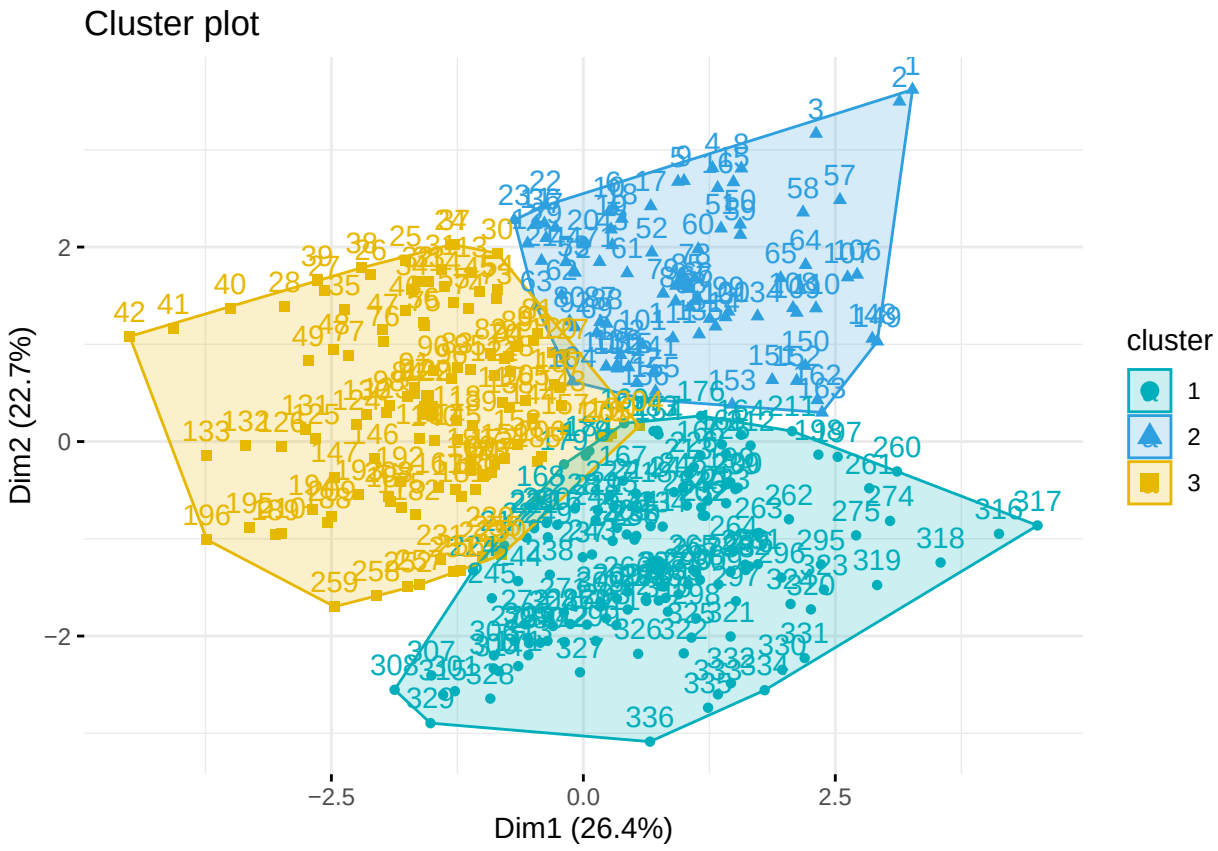


Création de clusters

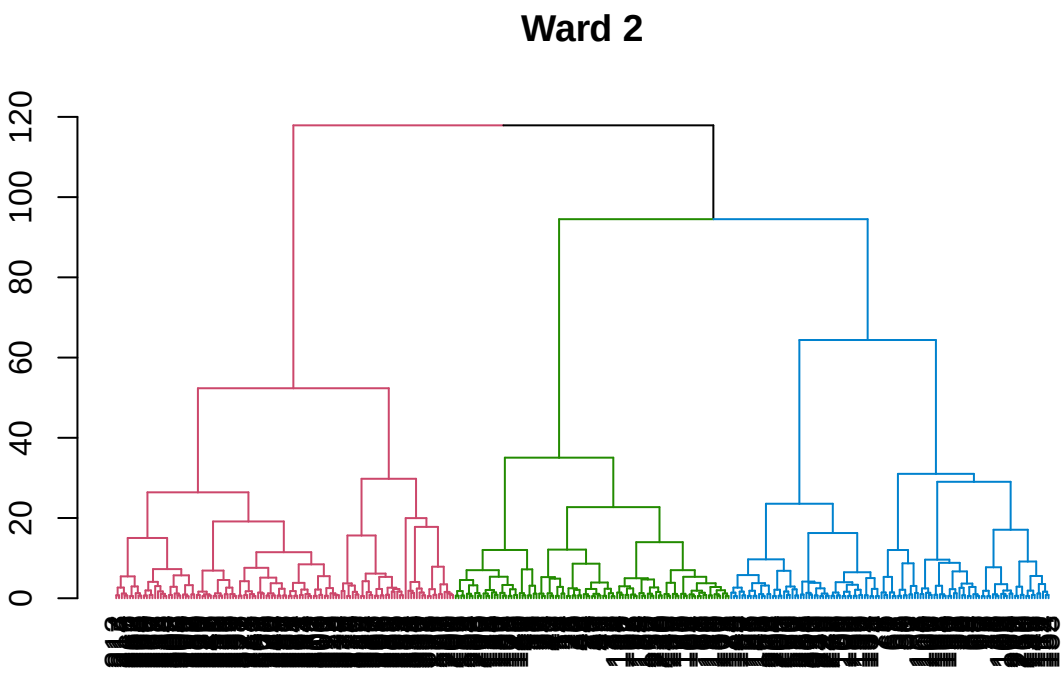
Nous voyons que l'algorithme ward est celui qui a les branches les mieux définies et départagées. C'est donc cet algorithme que je vais garder en clutering hiérarchique.

```r
set.seed(123)
k_ft <- kmeans(ft, 3, nstart = 10)

fviz_cluster(k_ft, data = ft,
```

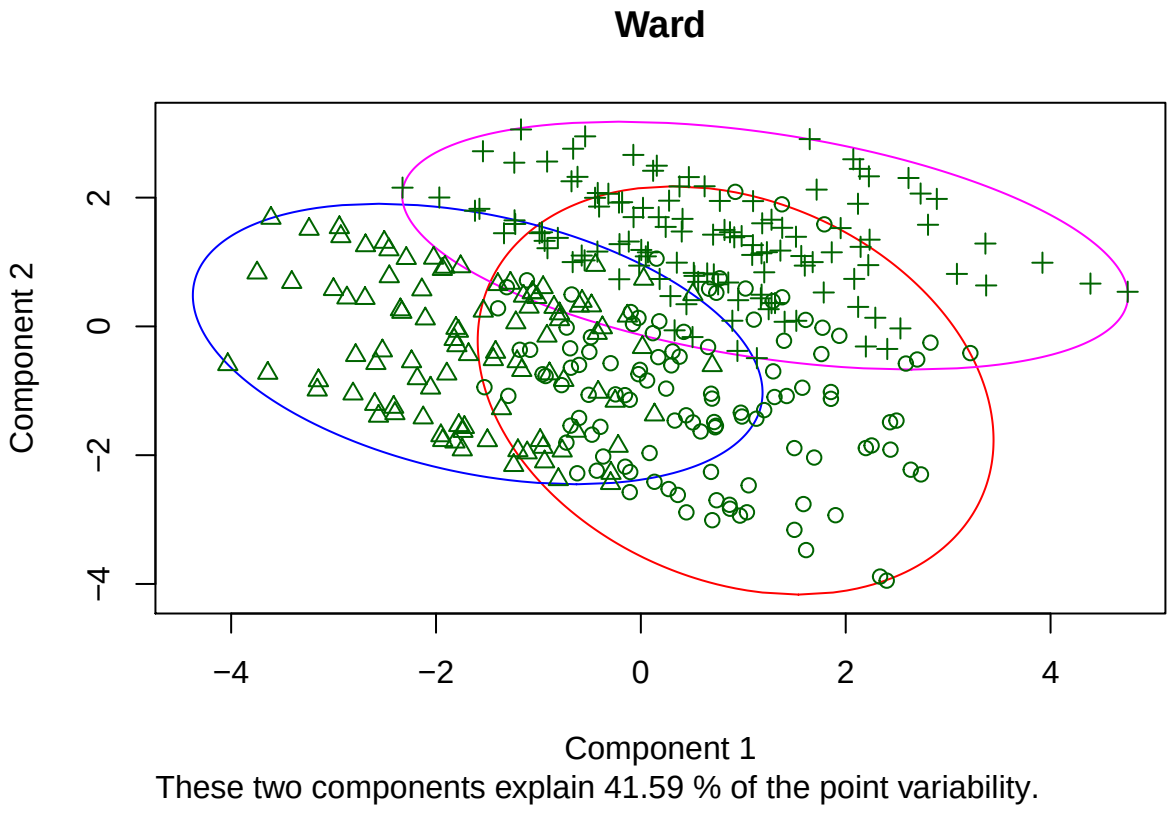
```
          palette = c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"),
          ggtheme = theme_minimal())
```




```
c_ward = cutree(h_ward, 3)
dend_ward <- as.dendrogram(h_ward)
dend <- color_branches(dend_ward, k = 3)
plot(dend, main = "Ward 2")
```




```
clusplot(fatality, c_ward, main='Ward',
         color=TRUE,
         shade=FALSE, lines = 0)
```



En comparant les clusters de l'algorithme K-means et Ward, je vois que les clusters sont assez similaires. Les clusters créés avec K-means étant plus maniables, ce sont eux que je vais garder pour la suite. #### Étude de chaque groupe

```r
# Création de sous data frame par groupe
clust <- cbind(fatality, clusterNum = k_ft$cluster)
clust1<- subset(clust,clust$clusterNum==1)
clust1 = clust1[, -12]
clust2<- subset(clust ,clust$clusterNum==2)
clust2 = clust2[, -12]
clust3<- subset(clust ,clust$clusterNum==3)
clust3 = clust3[, -12]

sum1 = summary(clust1)
sum2 = summary(clust2)
sum3 = summary(clust3)

sum1
```

```
##        X              state            year          mrall
##  Min.   :154.0   Min.   :28.00   Min.   :1982   Min.   :1.046
##  1st Qu.:224.8   1st Qu.:38.75   1st Qu.:1983   1st Qu.:1.744
##  Median :268.5   Median :46.00   Median :1985   Median :2.144
##  Mean   :261.3   Mean   :44.57   Mean   :1985   Mean   :2.223
##  3rd Qu.:302.2   3rd Qu.:51.00   3rd Qu.:1986   3rd Qu.:2.565
##  Max.   :336.0   Max.   :56.00   Max.   :1988   Max.   :4.218
##     beertax             mlda       jaild    comserd       vmiles
##  Min.   :0.04331   Min.   :18.00   no :79   no :116   Min.   : 5.575
##  1st Qu.:0.23917   1st Qu.:20.00   yes:57   yes: 20   1st Qu.: 7.481
##  Median :0.41284   Median :21.00                      Median : 8.166
##  Mean   :0.53784   Mean   :20.43                      Mean   : 8.381
##  3rd Qu.:0.67624   3rd Qu.:21.00                      3rd Qu.: 9.011
##  Max.   :2.06205   Max.   :21.00                      Max.   :26.148
##      unrate           perinc
##  Min.   : 2.800   Min.   :10394
##  1st Qu.: 5.875   1st Qu.:11852
##  Median : 7.400   Median :12893
##  Mean   : 7.651   Mean   :12969
##  3rd Qu.: 9.000   3rd Qu.:13951
##  Max.   :18.000   Max.   :17012
```

```r
sum2
```

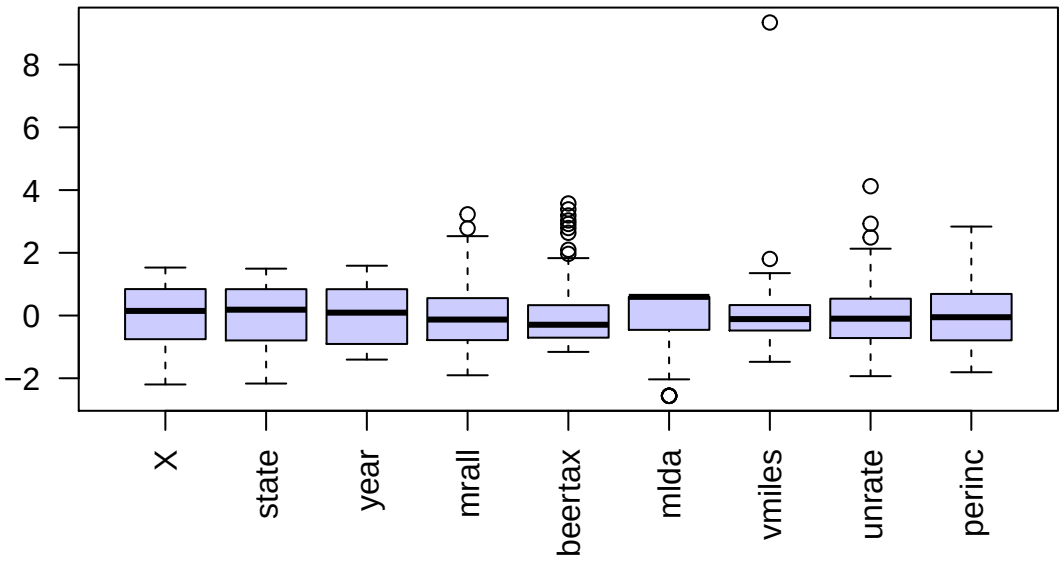
```
##        X              state            year          mrall
##  Min.   :  1.00   Min.   : 1.00   Min.   :1982   Min.   :1.328
##  1st Qu.: 26.00   1st Qu.: 7.00   1st Qu.:1982   1st Qu.:1.859
##  Median : 69.00   Median :16.00   Median :1984   Median :2.261
##  Mean   : 74.52   Mean   :15.63   Mean   :1984   Mean   :2.228
##  3rd Qu.:109.50   3rd Qu.:22.00   3rd Qu.:1985   3rd Qu.:2.545
##  Max.   :163.00   Max.   :30.00   Max.   :1988   Max.   :3.505
##     beertax             mlda       jaild    comserd       vmiles
##  Min.   :0.1032   Min.   :18.00   no :65   no :65   Min.   :5.697
##  1st Qu.:0.3328   1st Qu.:19.00   yes:18   yes:18   1st Qu.:7.004
##  Median :0.5245   Median :20.50                     Median :7.469
##  Mean   :0.7704   Mean   :20.06                     Mean   :7.550
##  3rd Qu.:1.0697   3rd Qu.:21.00                     3rd Qu.:8.029
##  Max.   :2.7208   Max.   :21.00                     Max.   :9.817
```

```
##      unrate           perinc
##  Min.   : 5.000   Min.   : 9514
##  1st Qu.: 7.800   1st Qu.:11424
##  Median : 8.900   Median :12033
##  Mean   : 9.196   Mean   :12444
##  3rd Qu.:10.450   3rd Qu.:13555
##  Max.   :15.500   Max.   :17255
```

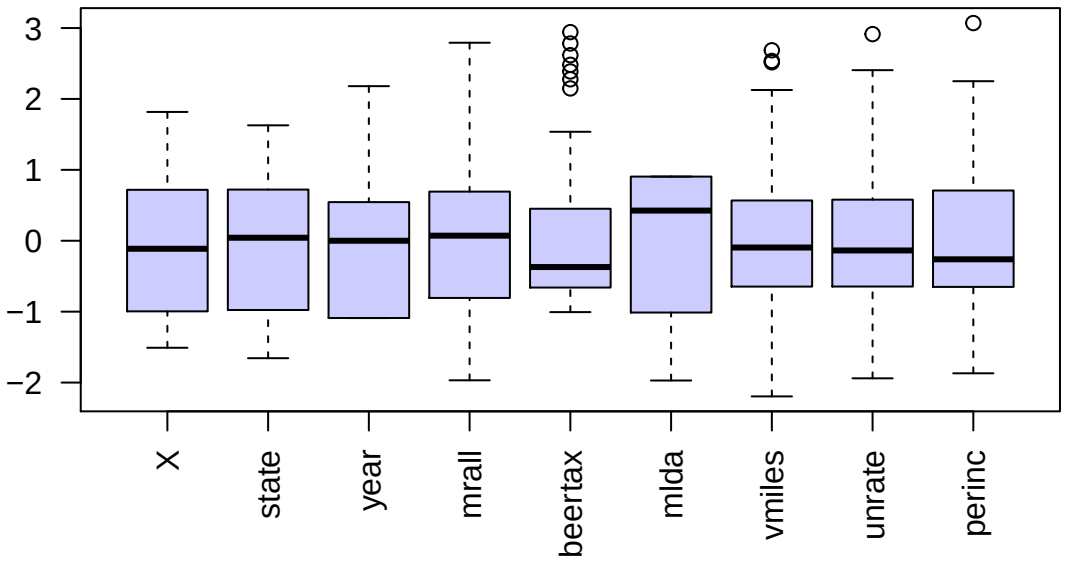
```
sum3
```

```
##        X              state            year          mrall
##  Min.   : 13.0   Min.   : 4.00   Min.   :1982   Min.   :0.8212
##  1st Qu.: 74.0   1st Qu.:17.00   1st Qu.:1985   1st Qu.:1.3817
##  Median :127.0   Median :25.00   Median :1986   Median :1.6720
##  Mean   :127.3   Mean   :23.79   Mean   :1986   Mean   :1.6952
##  3rd Qu.:185.0   3rd Qu.:33.00   3rd Qu.:1987   3rd Qu.:1.9455
##  Max.   :259.0   Max.   :44.00   Max.   :1988   Max.   :2.7673
##     beertax             mlda       jaild    comserd      vmiles
##  Min.   :0.07218   Min.   :19.00   no :98   no :93   Min.   :4.576
##  1st Qu.:0.15934   1st Qu.:21.00   yes:19   yes:24   1st Qu.:6.979
##  Median :0.24000   Median :21.00                     Median :7.708
##  Mean   :0.30226   Mean   :20.76                     Mean   :7.562
##  3rd Qu.:0.36124   3rd Qu.:21.00                     3rd Qu.:8.253
##  Max.   :1.11455   Max.   :21.00                     Max.   :9.816
##      unrate          perinc
##  Min.   :2.400   Min.   :12190
##  1st Qu.:4.600   1st Qu.:14631
##  Median :5.500   Median :15603
##  Mean   :5.681   Mean   :15958
##  3rd Qu.:6.700   3rd Qu.:16985
##  Max.   :9.900   Max.   :22193
```

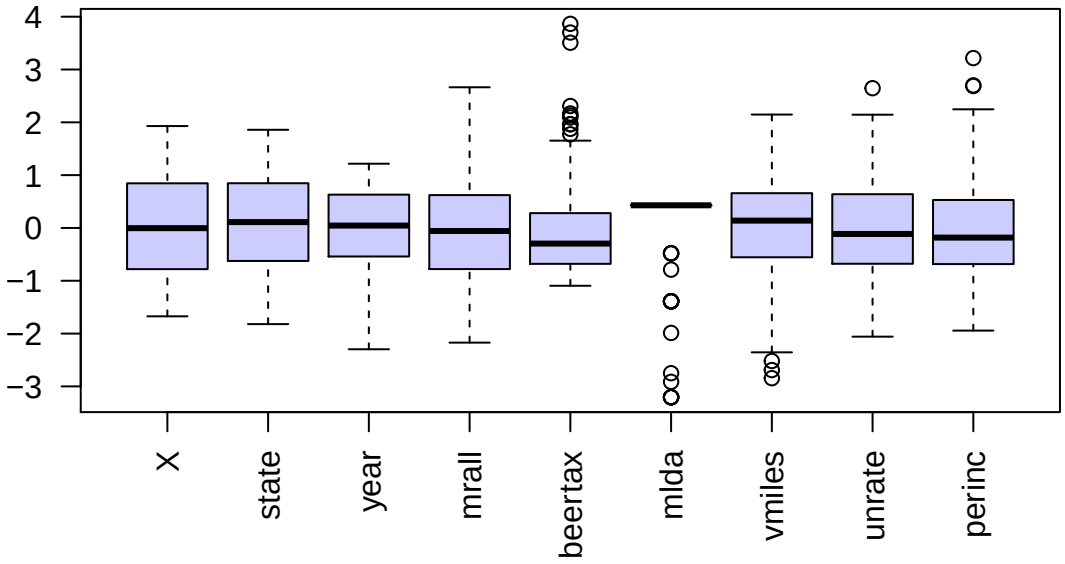
```
# Boxplot sur chaque groupe
par(mar=c(8,6,4,1))
sca1 = scale(clust1[, -c(7, 8)], center = T, scale = T)
boxplot(sca1, las = 2, col=rgb(0.8, 0.8, 1))
```



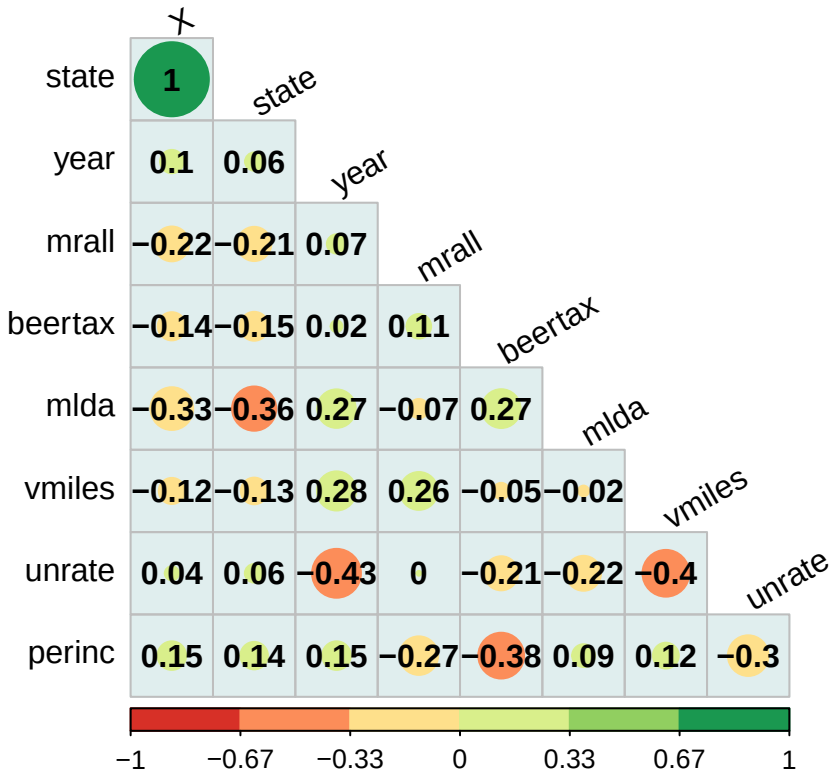
```
par(mar=c(8,6,4,1))
sca2 = scale(clust2[, -c(7, 8)], center = T, scale = T)
boxplot(sca2, las = 2, col=rgb(0.8, 0.8, 1))
```



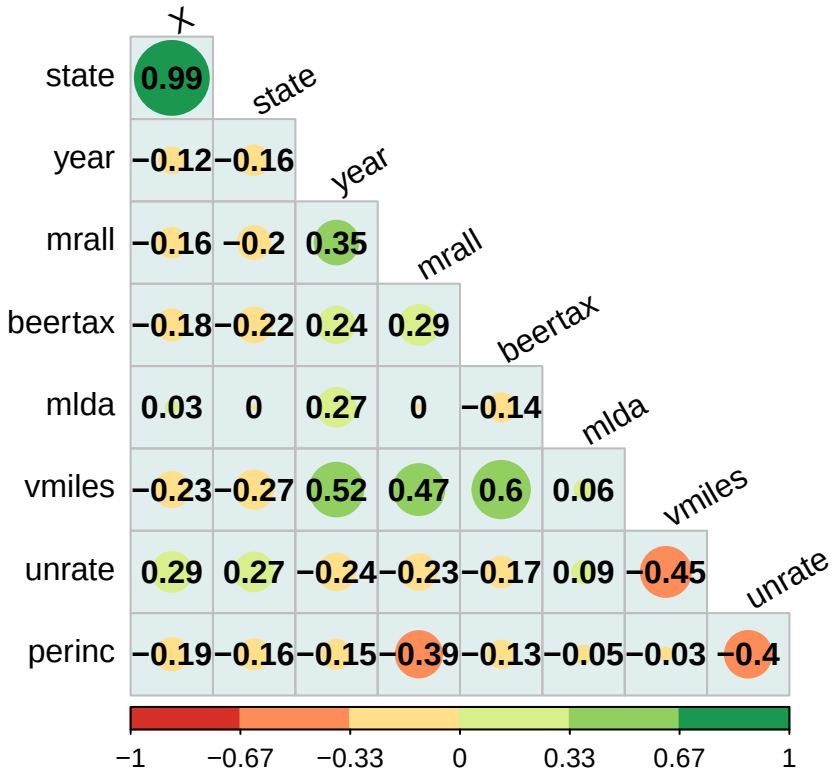

```
par(mar=c(8,6,4,1))
sca3 = scale(clust3[, -c(7, 8)], center = T, scale = T)
boxplot(sca3, las = 2, col=rgb(0.8, 0.8, 1))
```



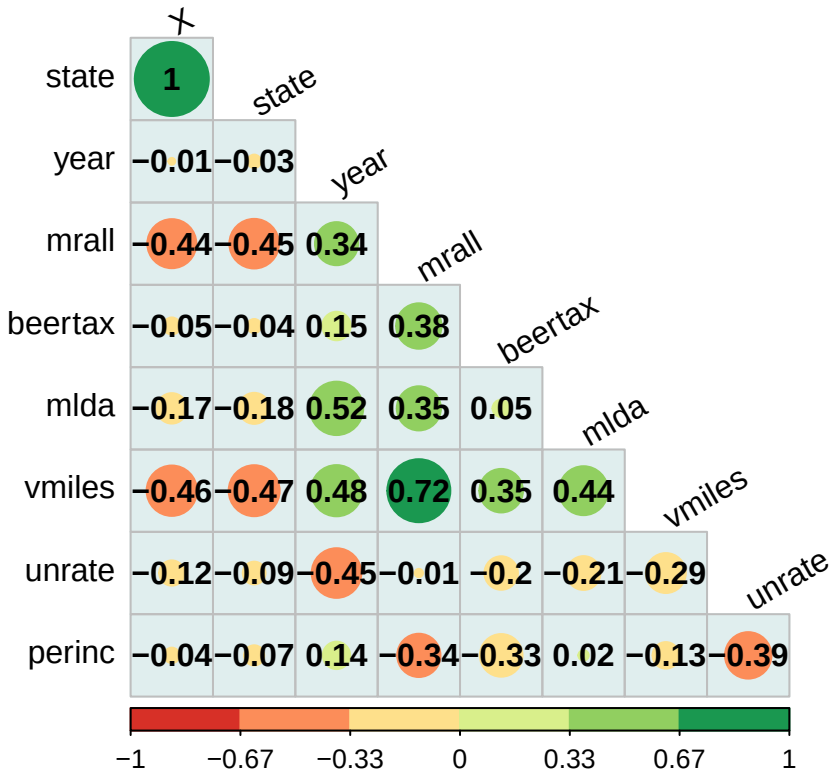

```
# Graphe de corrélation pour chaque groupe
corrplot(cor(clust1[, -c(7, 8)]),
         method = "circle", type = "lower", diag = FALSE,
         tl.srt = 30, tl.col = "black",
         bg= "azure2", col = brewer.pal(6,"RdYlGn"), addCoef.col = "black")
```




```
corrplot(cor(clust2[, -c(7, 8)]),
         method = "circle", type = "lower", diag = FALSE,
         tl.srt = 30, tl.col = "black",
         bg= "azure2", col = brewer.pal(6,"RdYlGn"), addCoef.col = "black")
```



```
corrplot(cor(clust3[, -c(7, 8)]),
         method = "circle", type = "lower", diag = FALSE,
         tl.srt = 30, tl.col = "black",
         bg= "azure2", col = brewer.pal(6,"RdYlGn"), addCoef.col = "black")
```



L'âge minimum pour boire de l'alcool semble t-il lié au nombre de morts dans certains groupes ? Dans le groupe n°3, l'âge minimum pour boire de l'alcool semble effectivement lié au nombre de morts : on a une corrélation positive de 0.35. Cela signifie que quand l'âge minimum augmente, le nombre de morts augmente également

Les groupes où les taxes de bière les plus faibles sont-ils ceux pour lesquels le nombre de morts est le plus élevé ? ceux avec le taux de chômage le plus élevé ? Le groupe 3 est celui avec les taxes les plus faibles, le nombre de morts et le taux de chômage les plus élevés. Le groupe 2 est celui avec les taxes les plus élevée, et le nombre de morts de le taux de chômage les plus faibles. Le groupe1 se situe entre les groupe 2 et 3 pour les taxes, le nombres de décès et le chômage.

Les groupes où l'on boit le plus sont-ils ceux où les revenus sont les plus élevés ? ceux avec le moins de condamnations ? Le groupe 3 est celui avec les revenus les plus élevés, puis le groupe 2, puis le 1. On suppose que les groupes où les taxes sont les plus élevées sont ceux qui boivent le plus, en partant du principe que c'est pour dissuader l'achat de bières. On voit alors que le groupe 3, avec les revenus les plus élevés, est celui qui boit le moins.

```
table(clust1[, c(7, 8)])
```

```
##      comserd
## jaild no yes
```

```
##   no  79   0
##   yes 37  20
```

```
table(clust2[, c(7, 8)])
```

```
##      comserd
## jaild no yes
##   no  60   5
##   yes  5  13
```

```
table(clust3[, c(7, 8)])
```

```
##      comserd
## jaild no yes
##   no  89   9
##   yes  4  15
```

Groupe 1 : jail = 100*57/136 = 42% comserd = 100*20/136 = 15% total = 29%

Groupe 2 : jail = 100*18/83 = 22% comserd = 100*18/83 = 22% total = 22%

Groupe3 : jail = 100*19/117 = 16% comserd = 100*24/117 = 21% total = 19%

Nous pouvons voir que le groupe avec le plus de revenus a le moins de condamnations, et celui avec le moins de revenus a le plus de condamnations.

La mortalité est-elle équivalente pour tous les groupes ? Quels sont les états où la mortalité est la plus élevée ? La mortalité du groupe 3 est en moyenne plus faible que les autres. On observe que c'est le groupe avec les mortalités minimale et maximale les plus faibles.

```
mrtl = head(fatality[order(fatality$mrall, decreasing = TRUE),], 40)
rvn = head(fatality[order(fatality$perinc),], 60)

mrtl = unique(mrtl$state)
rvn = unique(rvn$state)
mrtl
```

```
##  [1] 35 56 30 40 32 45  4 28 48  1
```

```
rvn
```

```
##  [1] 28  5 45 54  1 49 21 47 16 37 46 35 23 22 30
```

```
intersect(mrtl, rvn)
```

```
## [1] 35 30 45 28  1
```

Les 10 états dans lesquels la mortalité est la plus élevée sont les états 1, 4, 28, 30, 32, 35, 40, 45, 48 et 56. Parmi eux, les états 1, 28, 30, 35 et 45 sont dans les 15 états les plus pauvres.