script_P5

brunop31

06/07/2020

```
library(dplyr)
library(FactoMineR)
library(factoextra)
library(gridExtra)

select<-dplyr::select
```

```
#J'importe les données de la FAO et je sélectionne les colonnes
#qui m'intéresse

pop_2008<-read.csv("pop_2008.csv", encoding = "UTF-8")%>%
  select("Zone", "pop2008" = "Valeur")


pop_2018<-read.csv("pop_2018.csv", encoding = "UTF-8")%>%
  select("Zone", "pop2018" = "Valeur")

prot_ani_hab<-read.csv("prot_ani_habitant.csv", encoding = "UTF-8")%>%
  select("Zone", "prot_ani" = "Valeur")


prot_hab<-read.csv("prot_habitant.csv", encoding = "UTF-8")%>%
  select("Zone", "prot" = "Valeur")

kcal_hab<-read.csv("kcal_habitant.csv", encoding = "UTF-8")%>%
  select("Zone", "kcal" = "Valeur")

pib_hab<-read.csv("pib_habitant.csv", encoding = "UTF-8")%>%
  select("Zone", "pib" = "Valeur")%>%group_by(Zone)%>%
  summarise_if(is.numeric, round)
```

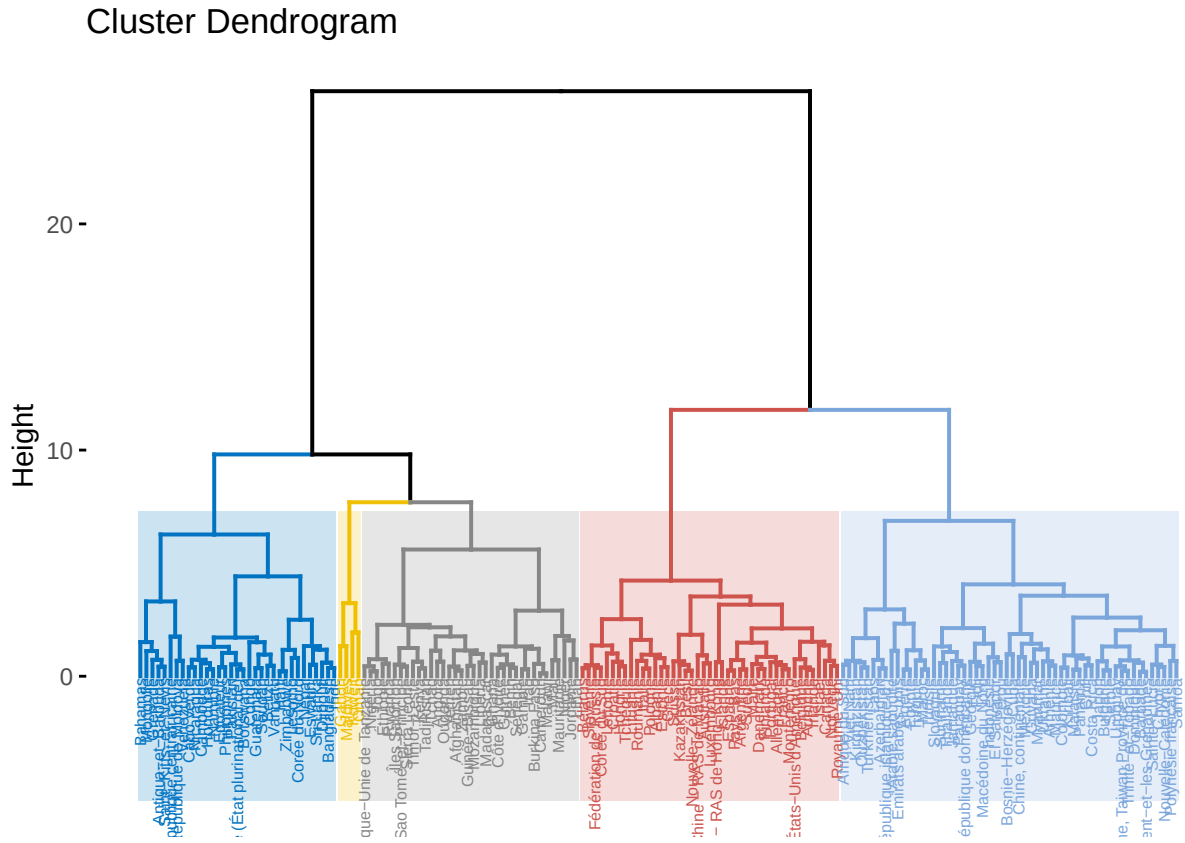
```
#Je regroupe les données dans un même tableau

df<-left_join(prot_ani_hab, pop_2008)%>%
  left_join(pop_2018)%>%left_join( prot_hab)%>%
  left_join(kcal_hab)%>%left_join(pib_hab)

#Je vérifie si il y a des données non renseigné

sapply(df,function(x) sum(is.na(x)))
```

```
##     Zone prot_ani  pop2008  pop2018     prot     kcal      pib
```

```
##        0        0        1        0        0        0        1
```

```
na_table<-filter(df, is.na(pop2008)|is.na(pib))

#Je complete les données non renseigné et je raccourci
#les noms de pays trop long
df$Zone<-df$Zone%>%as.vector()

df["138","Zone"]<-"Royaume-Uni"
df["135","Zone"]<-"Corée du Nord"
df["131","Zone"]<-"Corée du Sud"
df["133","Zone"]<-"Laos"
df["168","Zone"]<-"Venezuela"
df["150","pop2008"]<-   33060
df["35","pib"]<-24971

#Je fais les calculs pour obtenir les colonnes voulues

df<-df%>%
  mutate(pop_diff = round((pop2018 -pop2008)*100/pop2008,1),
         prot_ani_prct = round(prot_ani*100/prot,0))%>%
  select(-"prot_ani", -"pop2018", -"pop2008")
```

```
#Je construit mon dendrogramme à partir des variables demandées

df_table<-df
row.names(df_table)<-df_table$Zone
df_table<-select(df_table, -"Zone",-"pib")

df_table.cr<-scale(df_table, center = T, scale = T)

d.df_table<- dist(df_table.cr)

cah.ward <- hclust(d.df_table,method="ward.D2", )

#Je déssine mon arbre
fviz_dend(cah.ward,
          k = 5,
          cex = 0.4,                     # Label size
          palette = "jco",               # Color palette see ?ggpubr::ggpar
          rect = TRUE, rect_fill = TRUE, # Add rectangle around groups
          rect_border = "jco",           # Rectangle color
          labels_track_height = 5      # Augment the room for labels
          )
```




```
#Je récupère les cluster former par mon algorithme
groupes.cah <- cutree(cah.ward,k=5)

a<-as.data.frame(sort(groupes.cah))%>%select(clust = "sort(groupes.cah)")

#Je crée un fichier csv pour enregistrer mes groupes
write.csv(a, file = "liste_pays.csv")

a$Zone<-row.names(a)

df<-df%>%inner_join(a)
```

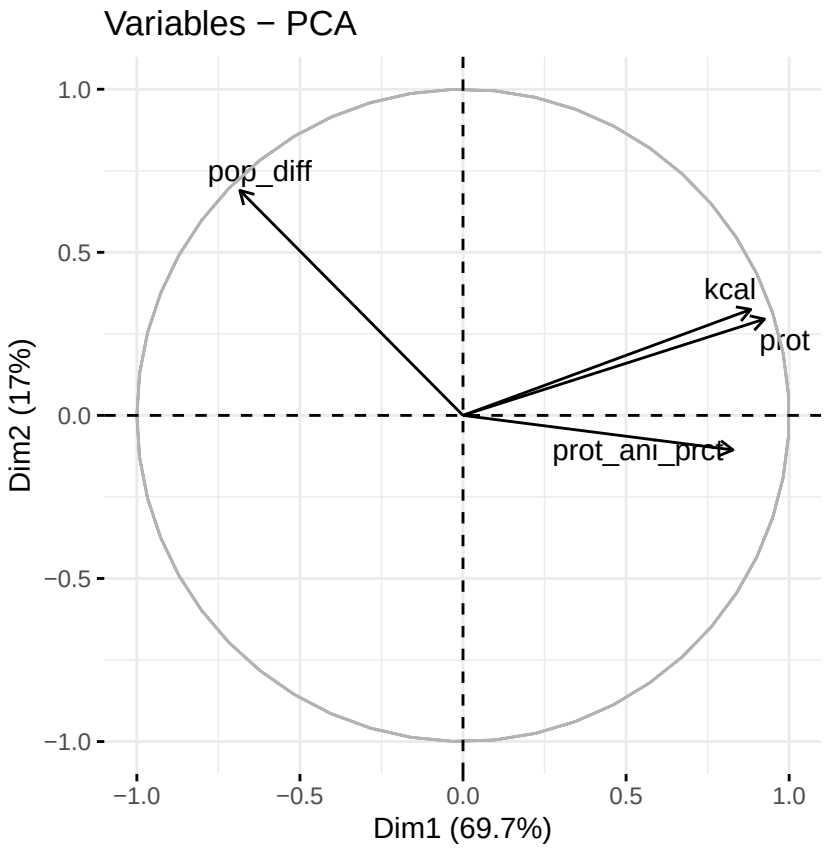
```
## Joining, by = "Zone"
```

```
#Je calcul les centre des clusters
centroide<-select(df, -"Zone")%>%
  group_by(clust)%>%
  summarise_if(is.numeric, function(x) round(mean(x),1))

#Je crée un fichier csv pour enregistrer mes clusters
write.csv(centroide, file = "centroide.csv", row.names = FALSE)
```

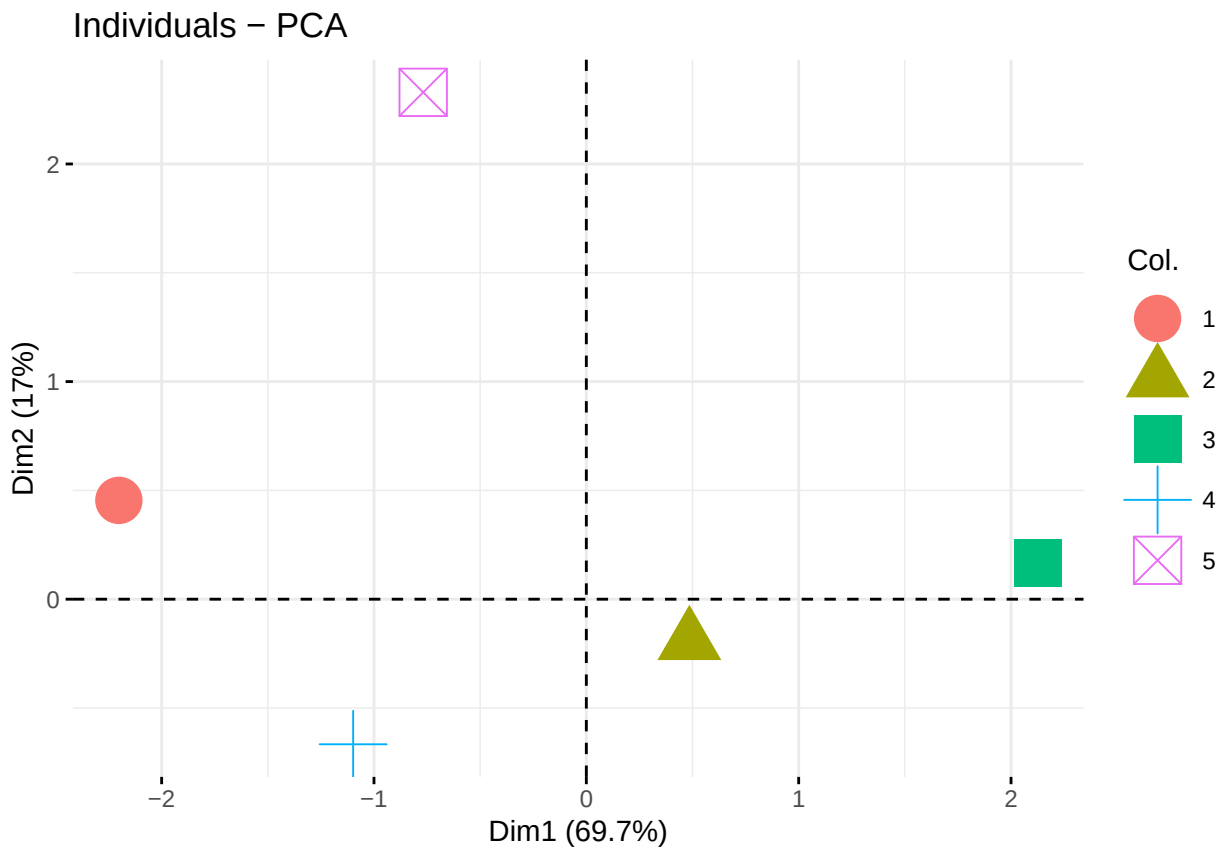
```
#Je renomme les cluster
df$clust<-as.vector(df$clust)
df[df$clust == 1,]$clust<-"pays sous développés"
```

```r
df[df$clust == 2,]$clust<-"autres"
df[df$clust == 3,]$clust<-"occident"
df[df$clust == 4,]$clust<-"pays en transition"
df[df$clust == 5,]$clust<-"pays à forte démographie"
df$clust<-as.factor(df$clust)
```

```r
#Je fais une pca sur mes données
res.pca <- PCA(df_table, graph = FALSE)

#J'affiche le cercle des corrélations
fviz_pca_var(res.pca,repel = TRUE)
```




```r
#J'observe mon premier plan (inertie >85%, je ne regarde pas les autres)
groupes<-groupes.cah%>%as.data.frame()

fviz_pca_ind(res.pca, col.ind = as.character(groupes.cah),
             mean.point = TRUE, geom.ind = c(""), pointsize = 4)
```




```
###Je réalise un test de shapiro wilk sur chaque variable de la
###table de base

w<-ggplot(df, aes(prot))+
  geom_histogram(aes(y=..density..), bins = 30,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(prot)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())

shapiro.test(df$prot)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$prot
## W = 0.97924, p-value = 0.01123
```

```
x<-ggplot(df, aes(kcal))+
  geom_histogram(aes(y=..density..), bins = 30,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(kcal)),
             color="blue", linetype="dashed", size=1)+
```

```
  theme(axis.text.y = element_blank())

shapiro.test(df$kcal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$kcal
## W = 0.98133, p-value = 0.02066
```

```
y<-ggplot(df, aes(pop_diff))+
  geom_histogram(aes(y=..density..), bins = 30,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(pop_diff)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())

shapiro.test(df$pop_diff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$pop_diff
## W = 0.95572, p-value = 3.012e-05
```

```
z<-ggplot(df, aes(prot_ani_prct))+
  geom_histogram(aes(y=..density..), bins = 30,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(prot_ani_prct)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())

shapiro.test(df$prot_ani_prct)
```

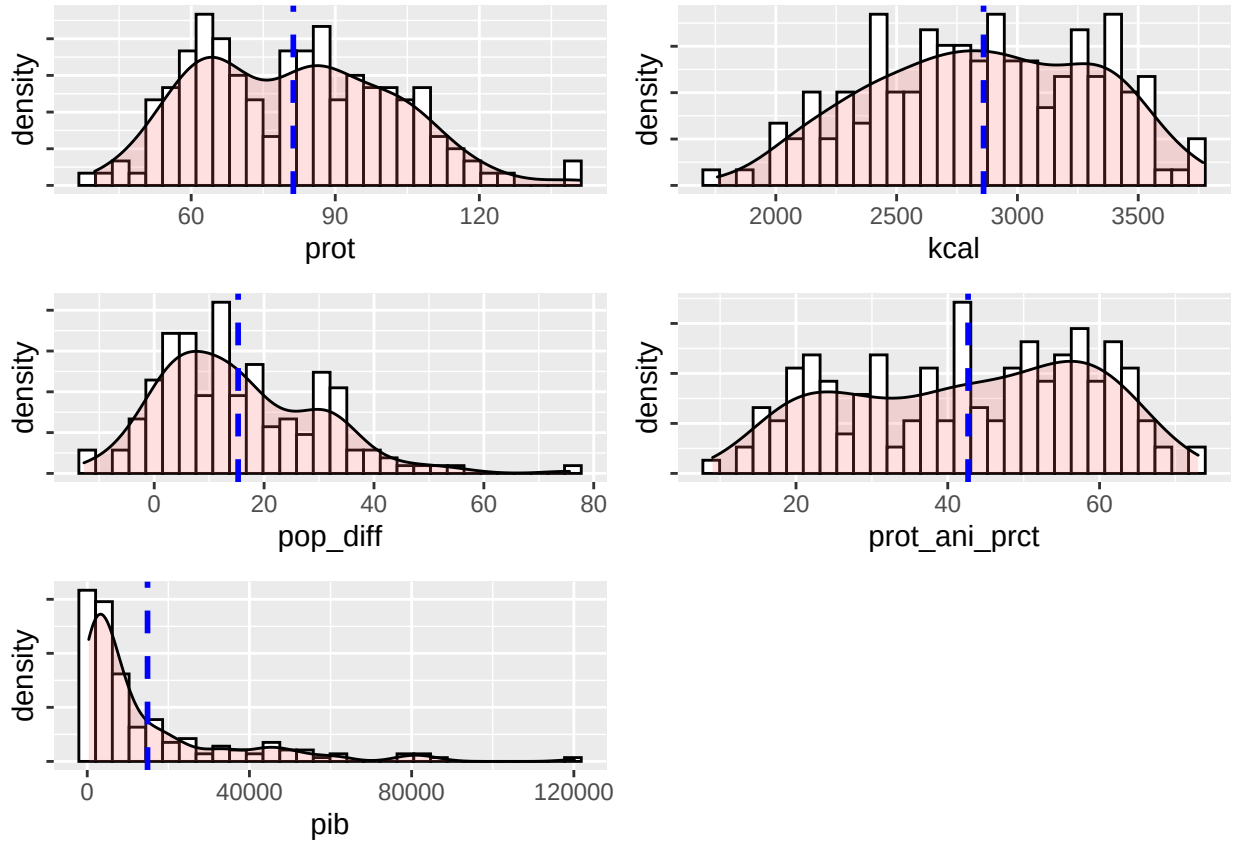
```
##
##  Shapiro-Wilk normality test
##
## data:  df$prot_ani_prct
## W = 0.95562, p-value = 2.944e-05
```

```
v<-ggplot(df, aes(pib))+
  geom_histogram(aes(y=..density..), bins = 30,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(pib)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())

shapiro.test(df$pib)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$pib
## W = 0.70032, p-value < 2.2e-16
```

```
###On rejette l'hypothèse nulle au seuil de 1%
###Les variables retenues sont kcal et prot
```

```
#J'affiche les courbe de densité de chacune des variables
grid.arrange(w,x,y,z,v, ncol=2, nrow = 3)
```

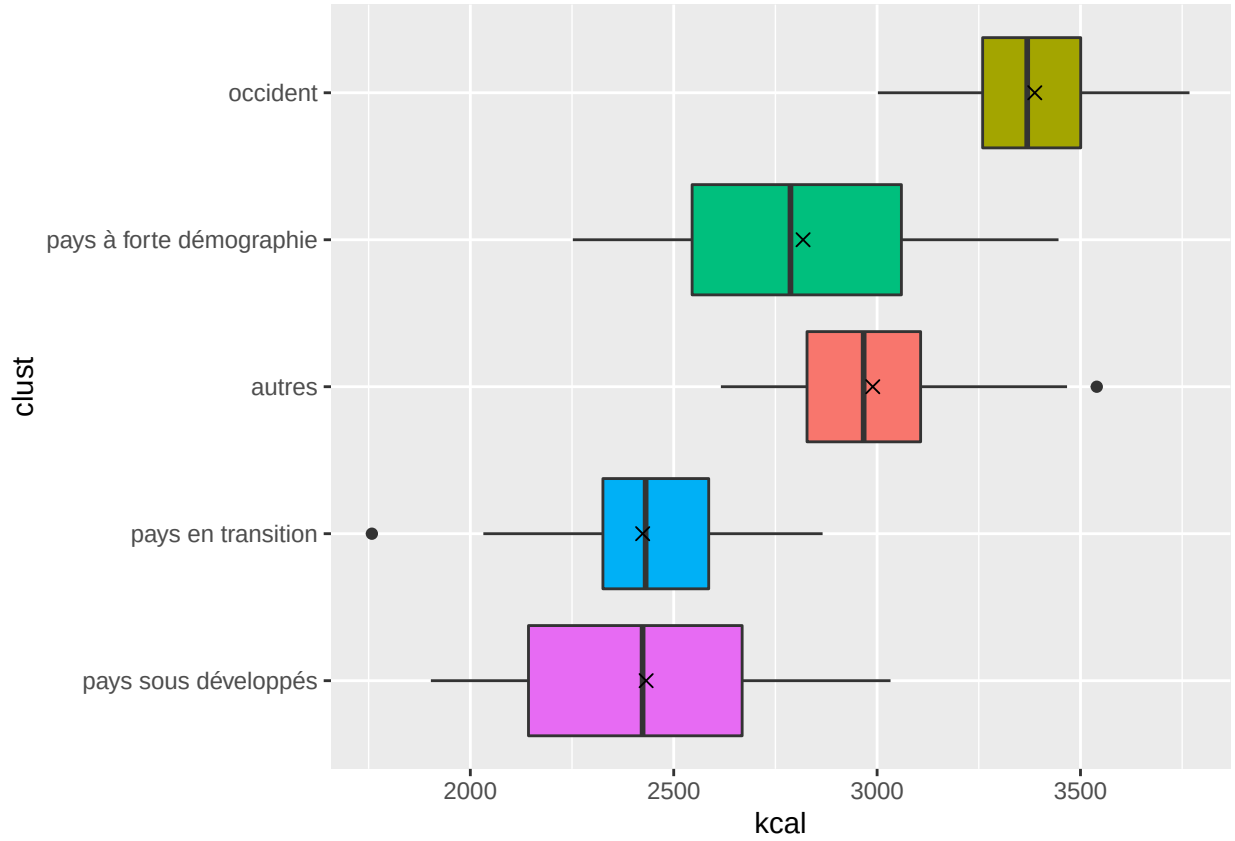



```
#J'extrait les pays de mes groupes dans 5 table différentes
df_1<-df%>%filter(clust == "pays sous développés")%>%
  select(-"clust")

df_2<-df%>%filter(clust == "pays en transition")%>%
  select(-"clust")

df_3<-df%>%filter(clust == "pays à forte démographie")%>%
  select(-"clust")

df_4<-df%>%filter(clust == "autres")%>%
  select(-"clust")
```

```
df_5<-df%>%filter(clust == "occident")%>%
  select(-"clust")
```

```
#J'observe la dispersion de mes groupes selon la variable kcal
ggplot(df, aes(x = clust, y = kcal, fill = clust))+
  geom_boxplot()+
  theme(legend.position="none")+
  stat_summary(fun.y=mean, geom="point", shape=4, size=2)+
  scale_x_discrete(limits=c("pays sous développés",
                            "pays en transition",
                            "autres",
                            "pays à forte démographie",
                            "occident"))+ coord_flip()
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```




```
#je construit les courbes de densitée pour kcal des 2 groupes
x1<-ggplot(filter(df, clust == "pays en transition"), aes(kcal))+
  geom_histogram(aes(y=..density..), bins = 10,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(kcal)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())
```

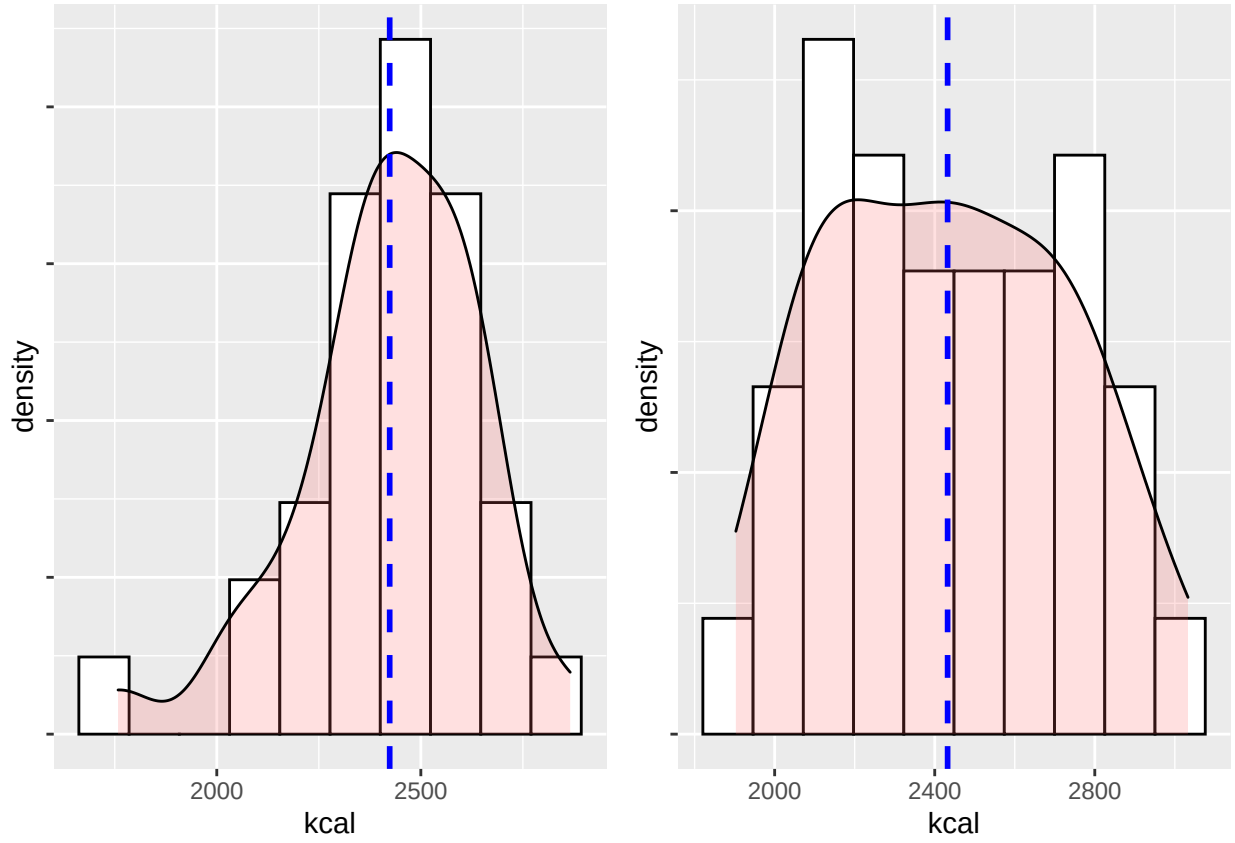
```
x2<-ggplot(filter(df, clust == "pays sous développés"), aes(kcal))+
  geom_histogram(aes(y=..density..), bins = 10,
                 colour="black", fill="white")+
 geom_density(alpha=.2, fill="#FF6666")+
  geom_vline(aes(xintercept=mean(kcal)),
             color="blue", linetype="dashed", size=1)+
  theme(axis.text.y = element_blank())

#J'éffectue un test de normalité pour kcal sur les 2 groupes
shapiro.test(filter(df, clust == "pays sous développés")$kcal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  filter(df, clust == "pays sous développés")$kcal
## W = 0.9698, p-value = 0.4202
```

```
shapiro.test(filter(df, clust == "pays en transition")$kcal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  filter(df, clust == "pays en transition")$kcal
## W = 0.95841, p-value = 0.2329
```

```
#j'affiche les courbes
grid.arrange(x1,x2, ncol=2, nrow = 1)
```



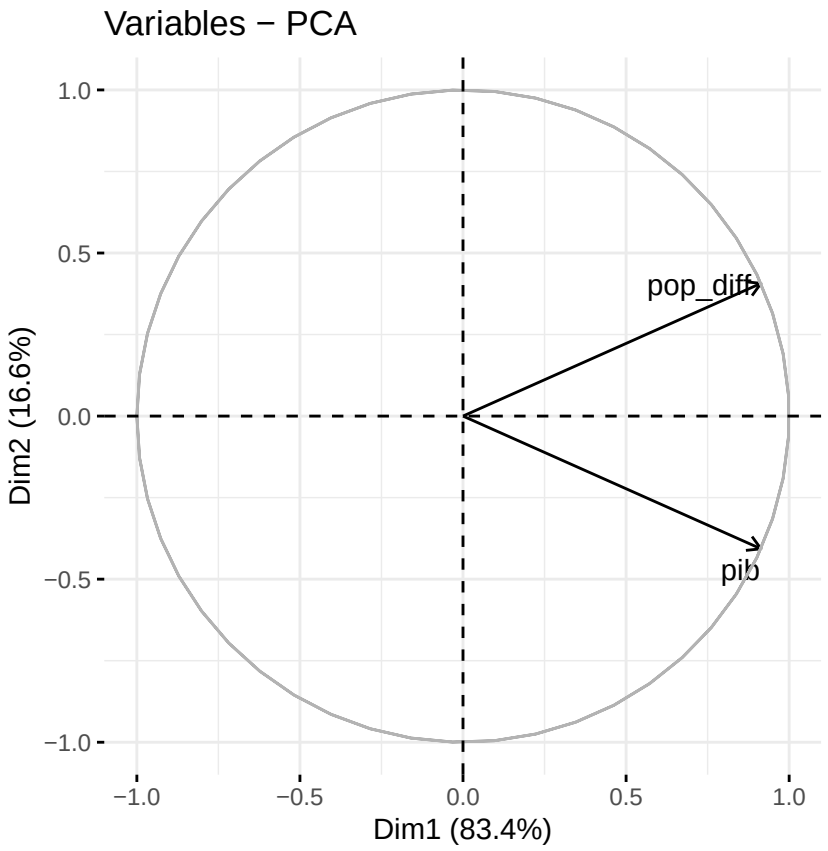

```
###Test de fisher (variance)
var.test(df_1$kcal, df_2$kcal)
```

```
##
##  F test to compare two variances
##
## data:  df_1$kcal and df_2$kcal
## F = 1.7916, num df = 35, denom df = 32, p-value = 0.09905
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8941111 3.5515242
## sample estimates:
## ratio of variances
##           1.791575
```

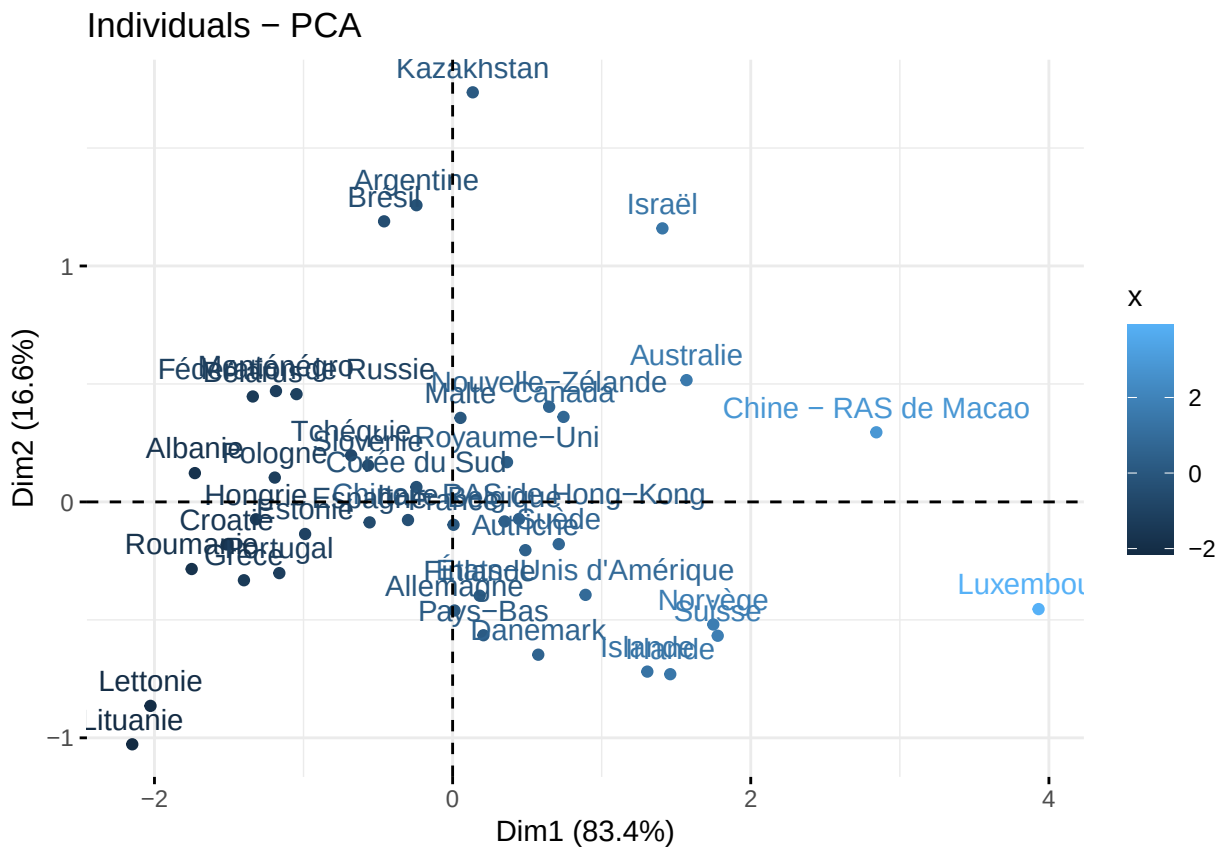
```
###p-value = 0.09905 les variances sont différentes au seuil de 10%
```

```
#Je choisis le groupe occident et je réalise une pca par rapport
#au variable pop_diff et pib
df_table_5<-df_5

row.names(df_table_5)<-df_table_5$Zone
df_table_5<-select(df_table_5,"pib","pop_diff")

res.pca <- PCA(df_table_5, graph = FALSE)
```

```
fviz_pca_var(res.pca,repel = TRUE)
```




```
fviz_pca_ind(res.pca, col.ind = "x")
```




```
#J'établis la liste des pays du plus intéressant au moins intéressant
#pour nous.
#La variable x est un mélange égal de la variable pib et de la
#variable pop_diff
pays_liste<-res.pca[["ind"]][["coord"]]%>%as.data.frame()
pays_liste$Zone<-rownames(pays_liste)
pays_liste<-pays_liste%>%select(Zone, Dim.1)%>%arrange(desc(Dim.1))
pays_liste
```

```
##                                          Zone        Dim.1
## Luxembourg                           Luxembourg  3.930617102
## Chine - RAS de Macao         Chine - RAS de Macao  2.842400609
## Suisse                                 Suisse  1.778446933
## Norvège                                Norvège  1.748649719
## Australie                              Australie  1.569058544
## Irlande                                Irlande  1.459909881
## Israël                                 Israël  1.407317487
## Islande                                Islande  1.306202806
## États-Unis d'Amérique       États-Unis d'Amérique  0.891733052
## Canada                                 Canada  0.743860375
## Suède                                  Suède  0.712306321
## Nouvelle-Zélande               Nouvelle-Zélande  0.647511948
## Danemark                               Danemark  0.574614825
## Autriche                               Autriche  0.488283165
## Chine - RAS de Hong-Kong Chine - RAS de Hong-Kong  0.445411038
## Royaume-Uni                         Royaume-Uni  0.364846866
```

```
## Belgique                                  Belgique  0.348661361
## Pays-Bas                                  Pays-Bas  0.206769408
## Finlande                                  Finlande  0.181866706
## Kazakhstan                              Kazakhstan  0.135105077
## Malte                                         Malte  0.052536407
## Allemagne                                Allemagne  0.012742290
## France                                      France  0.006090978
## Argentine                               Argentine -0.242605290
## Corée du Sud                         Corée du Sud -0.242972243
## Italie                                      Italie -0.299398263
## Brésil                                      Brésil -0.459575873
## Espagne                                  Espagne -0.556929279
## Slovénie                                Slovénie -0.566712863
## Tchéquie                                Tchéquie -0.680519213
## Estonie                                  Estonie -0.989388353
## Fédération de Russie         Fédération de Russie -1.047054239
## Portugal                                 Portugal -1.162378856
## Monténégro                             Monténégro -1.184902963
## Pologne                                  Pologne -1.192754433
## Hongrie                                  Hongrie -1.319279643
## Bélarus                                  Bélarus -1.340233107
## Grèce                                      Grèce -1.399711893
## Croatie                                  Croatie -1.515616659
## Albanie                                  Albanie -1.728938036
## Roumanie                                Roumanie -1.750421111
## Lettonie                                Lettonie -2.026545134
## Lituanie                                Lituanie -2.149005444
```

```
shapiro.test(df_1$kcal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_1$kcal
## W = 0.9698, p-value = 0.4202
```

```
shapiro.test(df_2$kcal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_2$kcal
## W = 0.95841, p-value = 0.2329
```