

# Analyse de données avec R

Ahmadou H. DICKO

Dakar R User Group

Janvier 2014

- 1 Intro
- 2 Présentation de R
- 3 Méthodes factorielles
- 4 Méthodes de classification
- 5 Un exemple d'analyse non supervisé

# Introduction

- Avec la révolution numérique on vit un déluge de données (facebook, twitter, etc.)
- Les données à traiter sont de plus en plus volumineux et liés
- L'analyse exploratoire (EDA) est une étape nécessaire pour comprendre les données et leurs liaisons potentielles
- L'analyse factorielle des données et les méthodes de classification présente une cadre efficace d'exploration de telle données
- Cette analyse passe par des logiciels spécialisés de plus en plus en performant
- R est l'un des langage pour l'analyse de données les plus utilisés dans le monde

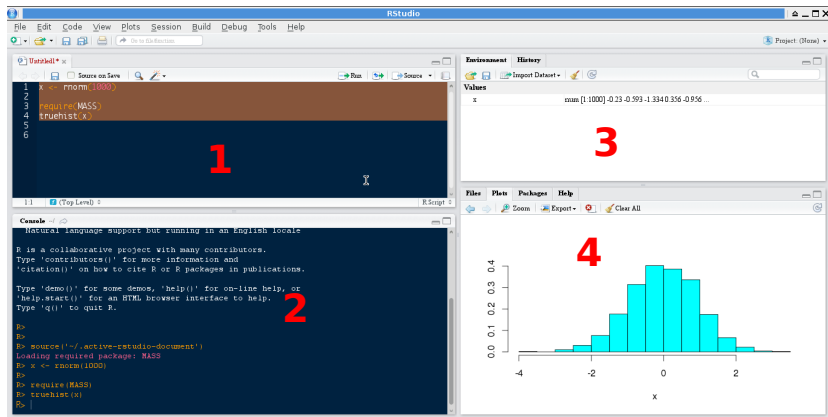
- 1 Intro
- 2 **Présentation de R**
- 3 Méthodes factorielles
- 4 Méthodes de classification
- 5 Un exemple d'analyse non supervisé

- R est un langage interprété qui tirent ses racines de S, Lisp, schemes
- Quelques dates importantes :
  - ▶ 1990 : Ross Ihaka et Robert Gentleman développent R
  - ▶ 1996 : le projet devient open source
  - ▶ 2013 : R 3.0 est sorti et il y a plus de 5000 packages (add-ons)

- Un package R est en ensemble de fonctionnalité généralement organisé autour d'un thème (ou pas)
- Travailler sur R nécessite d'écrire des lignes de commandes (pas toujours)
- Rstudio est un environnement de travail pour R (mais y en d'autres... ESS)

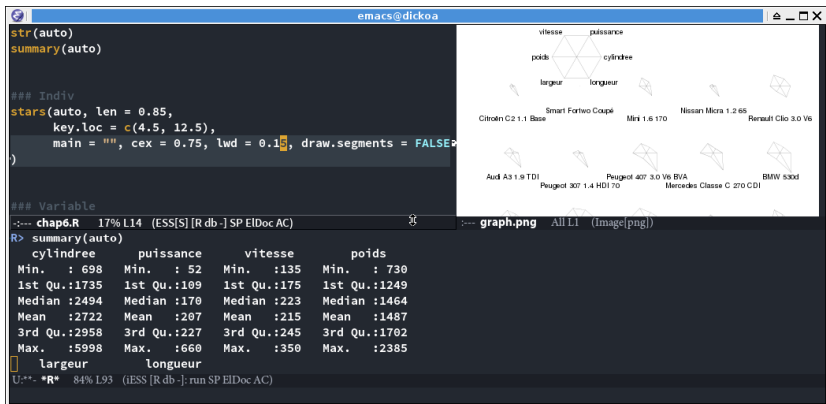
# Environnement de travail

## ● Rstudio



# Environnement de travail

## Emacs ESS



The screenshot shows the Emacs ESS interface. The left pane contains R code for summarizing car data and plotting a graph. The right pane displays the resulting graph, which is a radar chart comparing various car models across six performance metrics: vitesse, puissance, cylindree, longueur, largeur, and poids.

```
str(auto)
summary(auto)

### Indiv
stars(auto, len = 0.85,
      key.loc = c(4.5, 12.5),
      main = "", cex = 0.75, lwd = 0.15, draw.segments = FALSE)

### Variable
--- chap6.R 17% L14 (ESS[S] [R db -] SP ElDoc AC)
R> summary(auto)
  cylindree  puissance    vitesse    poids
Min.   : 698   Min.   : 52   Min.   :135   Min.   : 730
1st Qu.:1735   1st Qu.:109   1st Qu.:175   1st Qu.:1249
Median :2494   Median :170   Median :223   Median :1464
Mean   :2722   Mean   :207   Mean   :215   Mean   :1487
3rd Qu.:2958   3rd Qu.:227   3rd Qu.:245   3rd Qu.:1702
Max.   :5998   Max.   :660   Max.   :350   Max.   :2385

  largeur  longueur
U>*- *R* 84% L93 (iESS [R db -]: run SP ElDoc AC)
```

The graph on the right is a radar chart titled 'graph.png'. It compares ten car models across six metrics. The models are: Citroën C2 1.1 Base, Smart Fortwo Coupé, Mini 1.6 170, Nissan Micra 1.2 65, Renault Clio 3.0 V6, Audi A3 1.9 TDI, Peugeot 307 1.4 HDI 70, Peugeot 407 3.0 V6 BVA, Mercedes Classe C 270 CDI, and BMW 530d. The metrics are: vitesse, puissance, cylindree, longueur, largeur, and poids.



# Base du langage : R calculatrice ?

```
x <- c(pi, 4*pi / 3)
```

```
log(-cos(x))
```

---

[1] 0.00000 -0.69315

---

R output

# Base du langage : Collection d'éléments (1/2)

```
c(1, 2, 10)
```

---

[1] 1 2 10

---

```
c("Ali", "Modou", "Marie")
```

---

[1] "Ali" "Modou" "Marie"

---

## Base du langage : Collection d'éléments (2/2)

```
c(1, 2, "a")
```

R output

```
[1] "1" "2" "a"
```

```
list("Ali", 10, "Marie")
```

R output

```
[[1]]
```

```
[1] "Ali"
```

```
[[2]]
```

```
[1] 10
```

```
[[3]]
```

```
[1] "Marie"
```

# Base du langage : table (rectangulaire)

```
df <- data.frame(nom = c("Ali", "Modou", "Marie"),  
  taille = c(170, 185, 165))
```

```
df$nom
```

R output

```
[1] Ali   Modou Marie  
Levels: Ali Marie Modou
```

```
df[df$taille > 165, ]
```

R output

	nom	taille
1	Ali	170
2	Modou	185

- R dispose de fonctions pour faire des analyses basiques :
  - ▶ `prcomp`, `princomp`, etc.
  - ▶ `hclust`, `kmeans`, etc.
- Il existe de nombreux packages mais les principaux sont :
  - ▶ `FactoMineR`
  - ▶ `ade4`
- Nous utiliserons principalement `FactoMineR` pour le reste des analyses

- 1 Intro
- 2 Présentation de R
- 3 Méthodes factorielles**
- 4 Méthodes de classification
- 5 Un exemple d'analyse non supervisé

# Analyse en composantes principales

- Analyse exploratoire des tables de variables quantitatives
- Positionner les individus par rapport à leurs proximités
- Positionner les variables par rapport à leurs corrélations
- Chercher un petit nombre de variables non corrélées qui résument l'information de la table originale



# Mise en oeuvre sous R

```
auto <- read.csv("data/auto.csv", row.names = 1)
summary(auto)
```

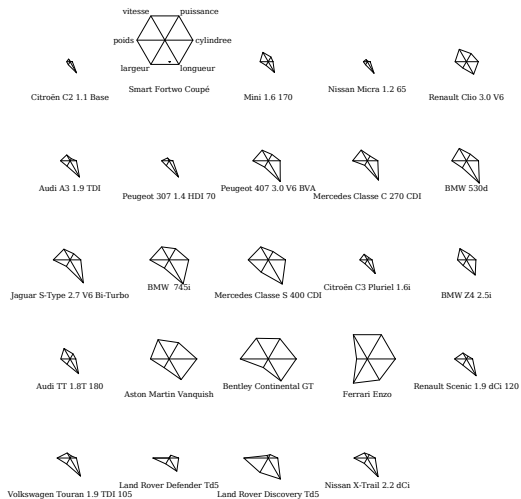
R output			
cylindree	puissance	vitesse	poids
Min. : 698	Min. : 52	Min. :135	Min. : 730
1st Qu.:1735	1st Qu.:109	1st Qu.:175	1st Qu.:1249
Median :2494	Median :170	Median :223	Median :1464
Mean :2722	Mean :207	Mean :215	Mean :1487
3rd Qu.:2958	3rd Qu.:227	3rd Qu.:245	3rd Qu.:1702
Max. :5998	Max. :660	Max. :350	Max. :2385
largeur	longueur		
Min. :1515	Min. :2500		
1st Qu.:1742	1st Qu.:3921		
Median :1792	Median :4325		
Mean :1838	Mean :4278		
3rd Qu.:1860	3rd Qu.:4701		
Max. :2650	Max. :5038		

# ACP : Mise en oeuvre sous R

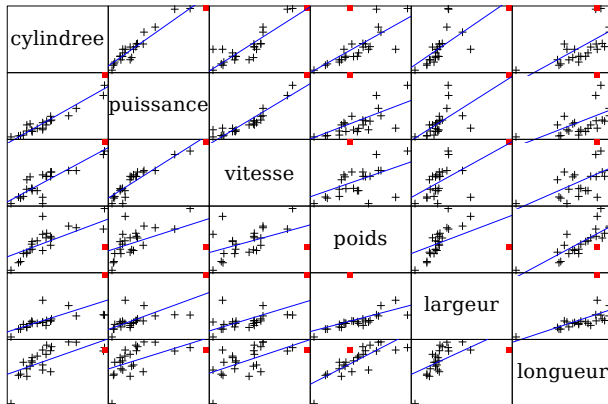
```
head(auto)
```

R output					
	cylindree	puissance	vitesse	poids	largeur
Citroën C2 1.1 Base	1124	61	158	932	1659
Smart Fortwo Coupé	698	52	135	730	1515
Mini 1.6 170	1598	170	218	1215	1690
Nissan Micra 1.2 65	1240	65	154	965	1660
Renault Clio 3.0 V6	2946	255	245	1400	1810
Audi A3 1.9 TDI	1896	105	187	1295	1765
	longueur				
Citroën C2 1.1 Base	3666				
Smart Fortwo Coupé	2500				
Mini 1.6 170	3625				
Nissan Micra 1.2 65	3715				
Renault Clio 3.0 V6	3812				
Audi A3 1.9 TDI	4203				

# ACP : Mise en oeuvre sous R

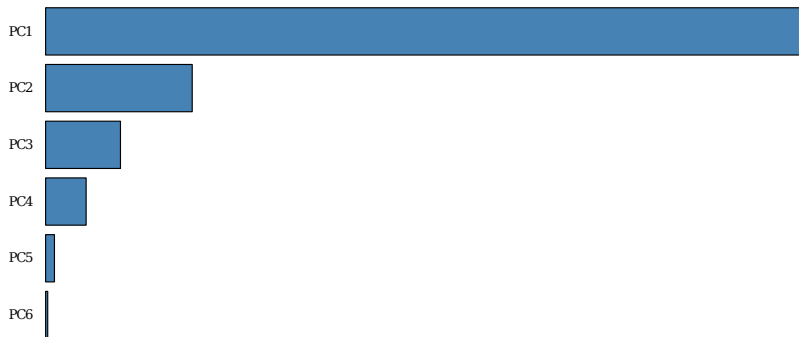


# ACP : Mise en oeuvre sous R



# ACP : Mise en oeuvre sous R

```
library(FactoMineR)
auto_acp <- PCA(auto, ncp = 2, graph = FALSE)
```



# ACP : Mise en oeuvre sous R

```
auto_acp$eig
```

---

	eigenvalue	percentage of variance
comp 1	4.411268	73.52113
comp 2	0.853410	14.22350
comp 3	0.435664	7.26107
comp 4	0.235871	3.93118
comp 5	0.051437	0.85728
comp 6	0.012351	0.20586

	cumulative percentage of variance
comp 1	73.521
comp 2	87.745
comp 3	95.006
comp 4	98.937
comp 5	99.794
comp 6	100.000

---

# ACP : Mise en oeuvre sous R

```
cbind(  
  dist = auto_acp$ind$dist^2,  
  coord = auto_acp$ind$coord,  
  cos2 = auto_acp$ind$cos2  
)[,1:4, ]
```

---

	dist	Dim.1	Dim.2	Dim.1	Dim.2
Citroën C2 1.1 Base	7.1023	-2.5959	-0.50997	0.94882	0.036618
Smart Fortwo Coupé	20.9302	-4.1501	-1.66591	0.82291	0.132595
Mini 1.6 170	2.9343	-1.3819	-0.81572	0.65082	0.226768
Nissan Micra 1.2 65	6.6117	-2.5133	-0.40359	0.95540	0.024635

---

# ACP : Mise en oeuvre sous R

```
auto_acp$var$coord
```

---

R output

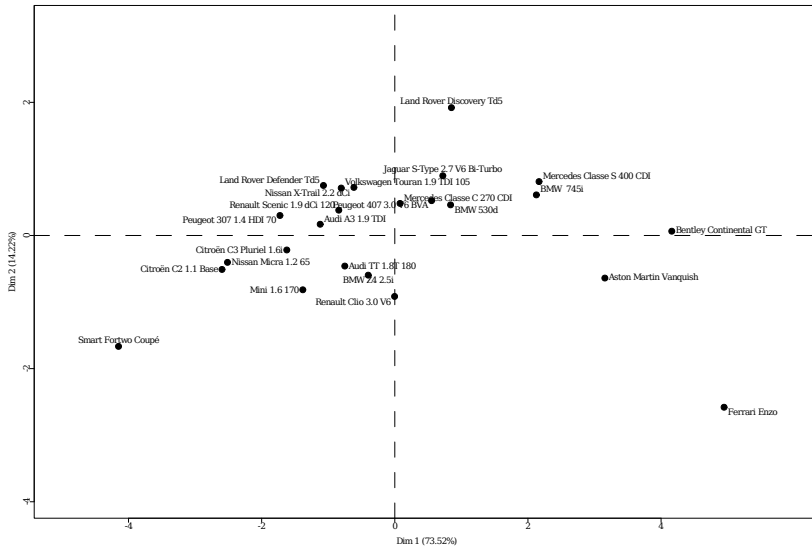
---

	Dim.1	Dim.2
cylindree	0.96241	-0.12694
puissance	0.92332	-0.35266
vitesse	0.88611	-0.33874
poids	0.75691	0.57571
largeur	0.80123	-0.11102
longueur	0.79526	0.50440

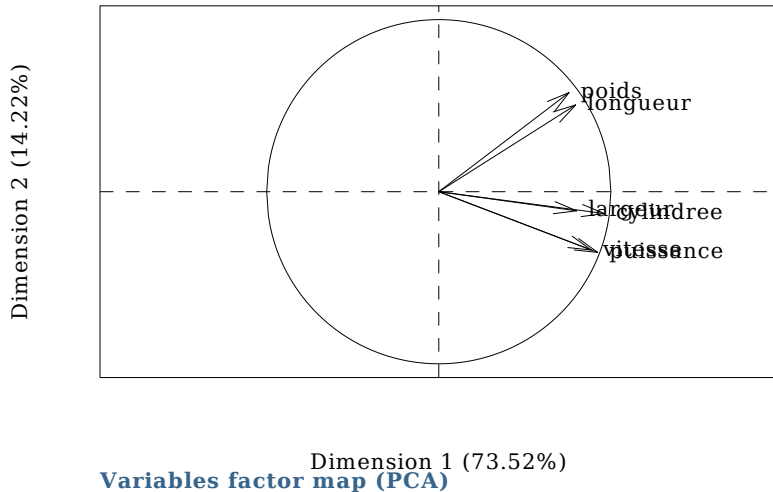
---



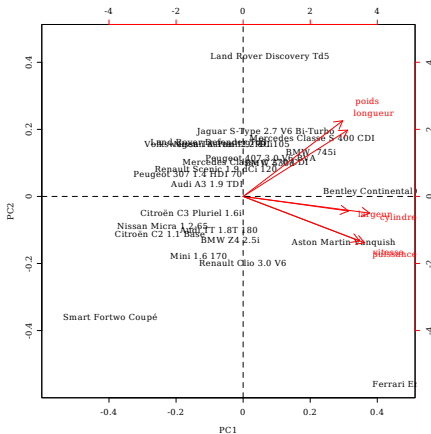
# ACP : Mise en oeuvre sous R



# ACP : Mise en oeuvre sous R



# ACP : Mise en oeuvre sous R



# Analyse factorielles des correspondances

- Analyse exploratoire de tables de contingence
- Mise en relief des écarts à l'indépendance du tableau de contingence
- Allez au delà du  $\chi^2$  et autre test d'indépendance

# AFC : Mise en oeuvre sous R

```
nice <- read.csv("data/nice.csv")
names(nice) <- tolower(names(nice))
str(nice)
```

\_\_\_\_\_ R output \_\_\_\_\_

```
'data.frame':  6 obs. of  7 variables:
 $ csp      : Factor w/ 6 levels "Agriculteur",...: 1 4 2 6 3 5
 $ iut      : int   9 66 77 50 52 55
 $ sciences: int   35 72 139 78 86 103
 $ lsh      : int   44 171 380 155 274 191
 $ droit    : int   24 122 195 152 43 40
 $ economie: int    8 48 69 57 26 25
 $ médecine: int   13 71 233 85 48 46
```

# AFC : Mise en oeuvre sous R

```
library(reshape2)
nice <- melt(nice, id = "csp", variable.name = "filiere")
str(nice)
```

---

R output

---

```
'data.frame':  36 obs. of  3 variables:
 $ csp      : Factor w/ 6 levels "Agriculteur",..: 1 4 2 6 3 5 1 4 2 6 ...
 $ filiere  : Factor w/ 6 levels "iut","sciences",..: 1 1 1 1 1 1 2 2 2 2 ...
 $ value    : int  9 66 77 50 52 55 35 72 139 78 ...
```

---

# AFC : Mise en oeuvre sous R

```
cont_table <- xtabs(value ~ csp + filiere, data = nice)
cont_table
```

---

R output

---

	filiere					
csp	iut	sciences	lsh	droit	economie	médecine
Agriculteur	9	35	44	24	8	13
Cadres	77	139	380	195	69	233
Employés	52	86	274	43	26	48
Indépendants	66	72	171	122	48	71
Ouvriers	55	103	191	40	25	46
Prof. inter.	50	78	155	152	57	85

---



```
chisq.test(cont_table)
```

---

R output

---

Pearson's Chi-squared test

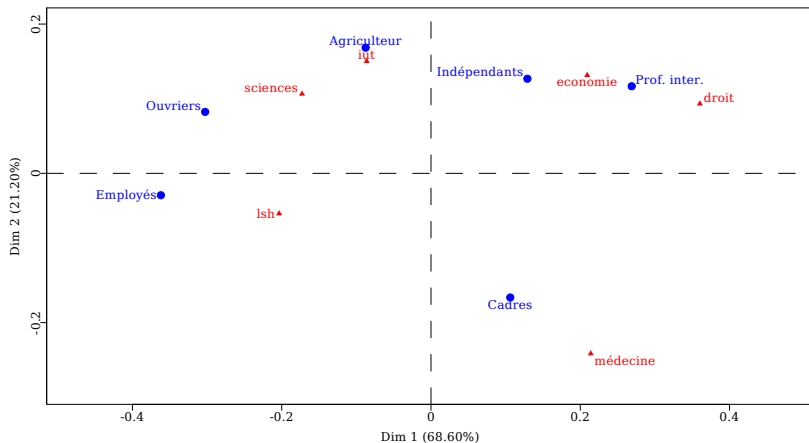
data: cont\_table

X-squared = 256.01, df = 25, p-value < 2.2e-16

---

# AFC : Mise en oeuvre sous R

```
nice_afc <- CA(cont_table, graph = FALSE)
```



# AFC : Mise en oeuvre sous R

```
rbind(  
  nice_afc$row$contrib[,1 :3],  
  nice_afc$col$contrib[,1 :3]  
)
```

	R output		
	Dim 1	Dim 2	Dim 3
Agriculteur	0.58122	6.94491	41.01525
Cadres	7.01907	55.70760	3.12380
Employés	39.44170	0.84044	24.34516
Indépendants	5.22233	16.28117	15.52434
Ouvriers	23.97869	5.74860	15.76031
Prof. inter.	23.75698	14.47727	0.23114
iut	1.30525	12.83230	8.81539
sciences	8.71843	10.71249	65.06416
lsh	28.64715	6.50715	15.86942
droit	42.57064	9.21883	0.78294
economie	5.81289	7.41395	3.06994
médecine	12.94564	53.31529	6.39815

# AFC : Mise en oeuvre sous R

```
rbind(  
  nice_afc$row$cos2[ ,1 :3],  
  nice_afc$col$cos2[ ,1 :3]  
)
```

	R output		
	Dim 1	Dim 2	Dim 3
Agriculteur	0.071781	0.265039	0.55201612
Cadres	0.285208	0.699478	0.01383267
Employés	0.915171	0.006026	0.06156014
Indépendants	0.413949	0.398790	0.13410216
Ouvriers	0.839285	0.062176	0.06011591
Prof. inter.	0.825516	0.155453	0.00087528
iut	0.150971	0.458651	0.11111768
sciences	0.455903	0.173102	0.37077989
lsh	0.871778	0.061192	0.05262910
droit	0.925526	0.061934	0.00185501
economie	0.678399	0.267374	0.03904481
médecine	0.423402	0.538837	0.02280468

# Analyse des correspondances multiples

- Analyse exploratoire de tables de variables qualitatives
- Mettre en évidence des proximités (non linéaire) entre individus, entre modalités et entre les deux
- Développer une typologie des individus et des modalités

# ACM : Mise en oeuvre sous R

```
chien <- read.csv("data/chien.csv", row.names = 1, colClasses = "factor")
names(chien) <- tolower(names(chien))
str(chien)
```

---

R output

---

```
'data.frame':  27 obs. of  7 variables:
 $ taille      : Factor w/ 3 levels "1","2","3": 3 1 3 2 1 3 1 1 2 3 ...
 $ poids       : Factor w/ 3 levels "1","2","3": 2 1 2 2 1 3 1 1 1 2 ...
 $ vitesse     : Factor w/ 3 levels "1","2","3": 3 1 3 2 1 1 2 1 1 3 ...
 $ intelligence: Factor w/ 3 levels "1","2","3": 2 1 3 2 2 3 3 1 2 2 ...
 $ affection   : Factor w/ 2 levels "1","2": 2 1 2 2 2 1 2 2 2 2 ...
 $ agressivité : Factor w/ 2 levels "1","2": 2 2 2 2 1 2 1 1 2 1 ...
 $ fonction    : Factor w/ 3 levels "1","2","3": 3 2 3 1 1 3 1 1 1 1 ...
```

---

# ACM : Mise en oeuvre sous R

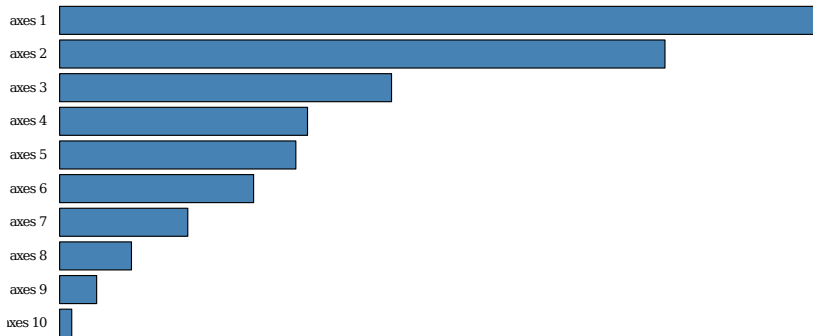
```
library(ade4)  
acm.disjonctif(chien)[1 :6, 1 :5]
```

	R output				
	taille.1	taille.2	taille.3	poids.1	poids.2
beauceron	0	0	1	0	1
basset	1	0	0	1	0
berger allemand	0	0	1	0	1
boxer	0	1	0	0	1
bull-dog	1	0	0	1	0
bull-mastiff	0	0	1	0	0

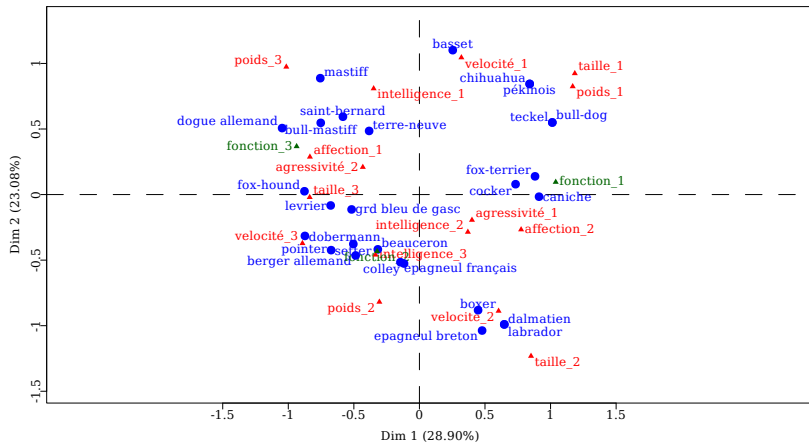


# ACM : Mise en oeuvre sous R

```
chien_acm <- MCA(chien,  
  quali.sup = match("fonction", names(chien)),  
  graph = FALSE)
```



# ACM : Mise en oeuvre sous R



- 1 Intro
- 2 Présentation de R
- 3 Méthodes factorielles
- 4 Méthodes de classification**
- 5 Un exemple d'analyse non supervisé

# Classification hierarchique

- Créez des groupes en agrégeant successivement les individus

# CAH : Mise en oeuvre sous R

```
tempsen <- read.csv("data/tempsen.csv", row.names = 1)
tempsen[1 :8, 1 :6]
```

R output						
	Jan	Feb	Mar	Apr	May	Jun
Bakel	34.017	37.017	41.058	43.292	43.442	40.525
Bambey	33.075	35.025	37.792	38.050	37.992	36.592
Cap-skiring	31.950	32.483	31.933	30.208	30.058	31.608
Dakar-yoff	26.158	25.442	25.700	25.358	26.575	29.150
Diourbel	34.083	35.958	38.992	39.908	40.792	38.008
Fatick	34.283	36.183	39.150	39.642	39.192	36.792
Goudiry	34.420	37.360	40.480	41.380	41.900	38.280
Kaolack	34.850	36.908	39.992	40.800	40.275	37.292

# CAH : Mise en oeuvre sous R

```
mat_dist <- dist(tempsn)
as.matrix(mat_dist)[1 :5, 1 :5]
```

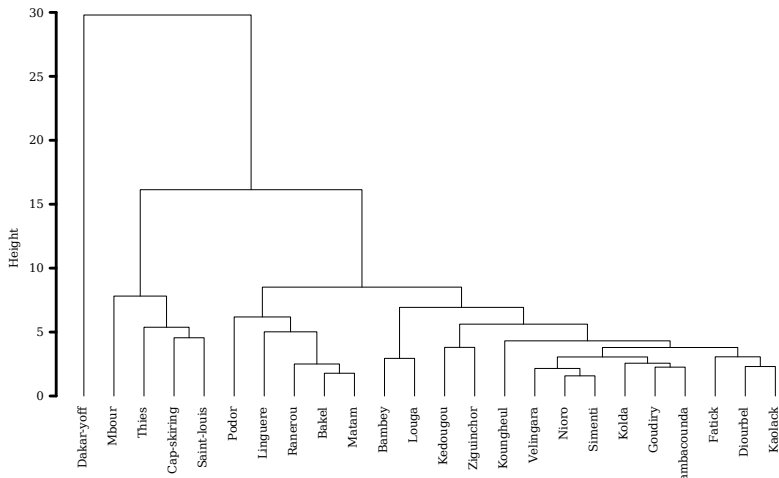
---

	R output				
	Bakel	Bambey	Cap-skiring	Dakar-yoff	Diourbel
Bakel	0.0000	10.1929	25.641	37.296	5.8866
Bambey	10.1929	0.0000	15.561	27.693	4.5734
Cap-skiring	25.6410	15.5614	0.000	13.790	20.0178
Dakar-yoff	37.2961	27.6932	13.790	0.000	32.0160
Diourbel	5.8866	4.5734	20.018	32.016	0.0000

---

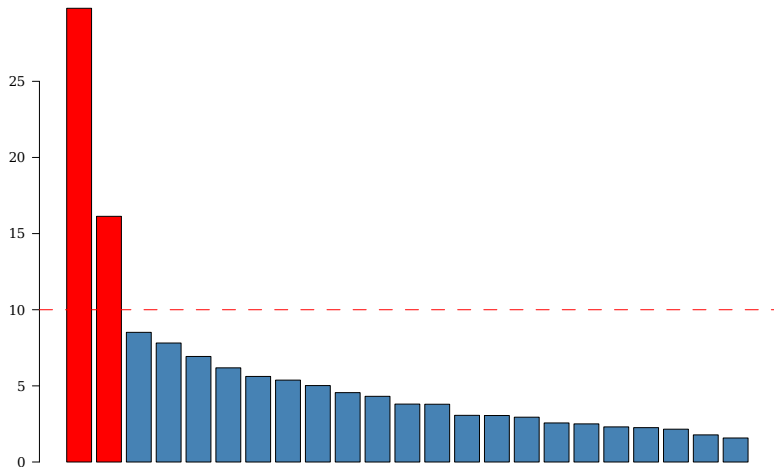
# CAH : Mise en oeuvre sous R

```
tempclust <- hclust(mat_dist, method = "average")
```



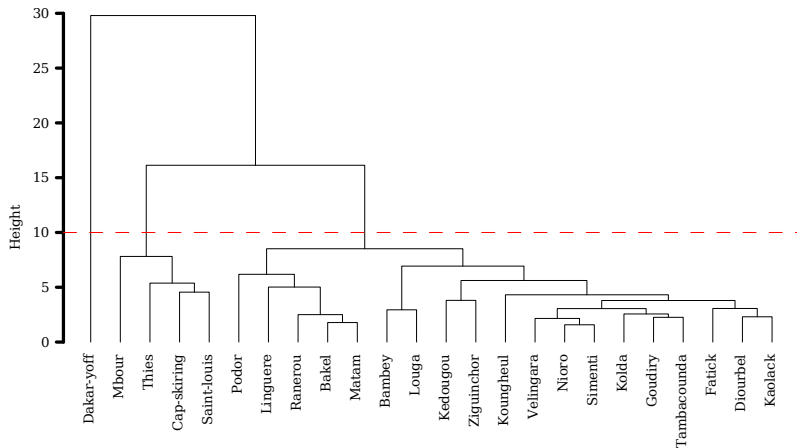


# CAH : Mise en oeuvre sous R

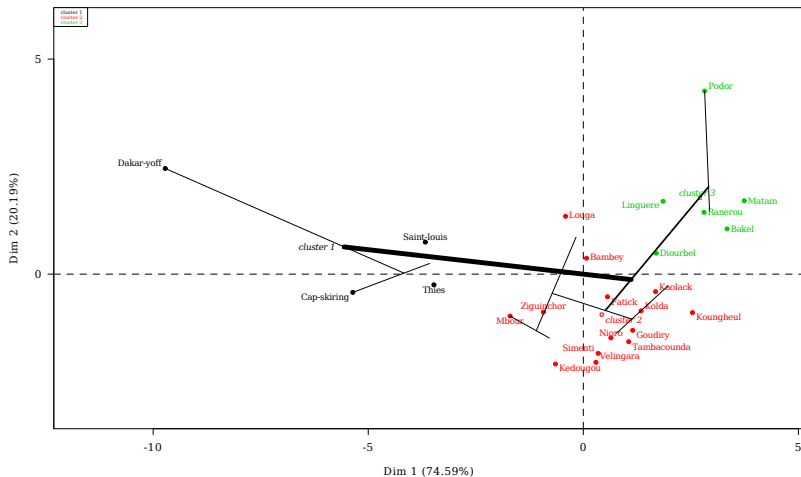


# CAH : Mise en oeuvre sous R

```
temphcut <- cutree(tempclust, k = 3)
```



# CAH : Mise en oeuvre sous R



- 1 Intro
- 2 Présentation de R
- 3 Méthodes factorielles
- 4 Méthodes de classification
- 5 Un exemple d'analyse non supervisé

# Un exemple pour conclure

# Classification des ménages Sénégalais selon leur niveau de vie

- étape 1 : Analyse factorielle : ACM
- étape 2 : K-means sur les axes factorielles
- étape 3 : CAH sur les centres de gravités des groupes formés
- étape 4 : K-means pour consolider les groupes

# Mise en oeuvre sous R

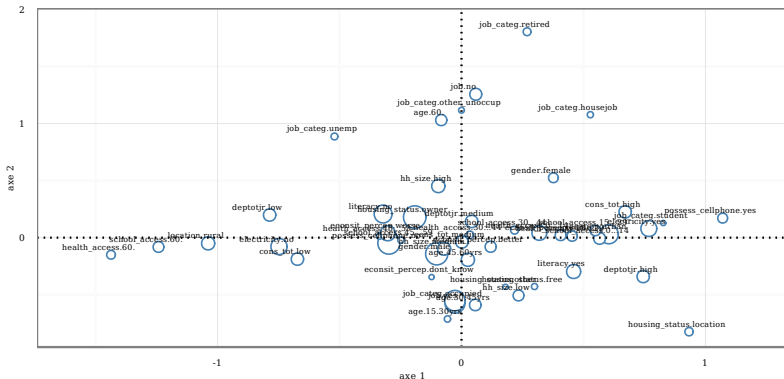
```
household <- readRDS("data/household.rds")
```

```
summary(household)
```

region		location	R output	
			gender	age
dakar	:1598	urban:8576	male :10684	15-30yrs: 775
diourbel	:1200	rural:4989	female: 2881	30-45yrs:3983
kolida	:1200			45-60yrs:5043
saint-louis	:1200			60+ :3764
tamba	:1200			
kaolack	:1199			
(Other)	:5968			
hh_size	literacy	job	housing_status	
low :3563	yes:5582	yes:9312	owner	:10853
medium:4948	no :7983	no :4253	location:	2013
high :5054			free	: 638
			other	: 61

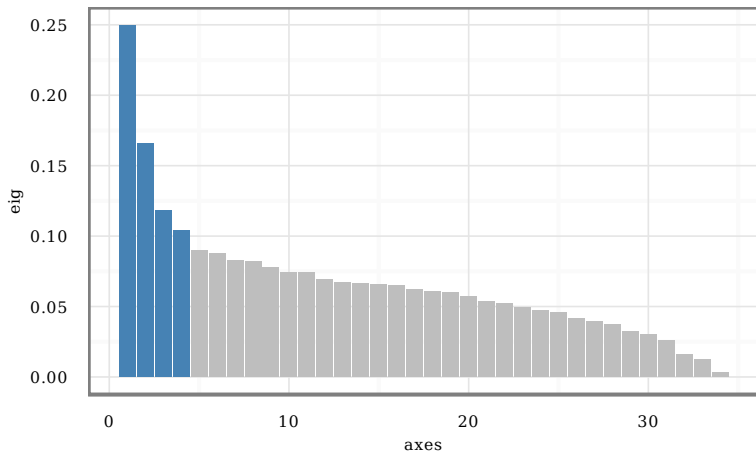
## Mise en oeuvre sous R

```
naxes_max <- sum(sapply(household[, -1], nlevels)) - ncol(household[, -1])
mca <- dudi.acm(household[, -1], nf = naxes_max, scannf = FALSE)
```





# Mise en oeuvre sous R



# Mise en oeuvre sous R

- On va donc travailler avec les 4 premiers axes et vérifier qu'ils sont bien centrés et réduits.

```
mcadata <- mca$l1[,1:4]  
sapply(mcadata, function(x) list(mean = round(mean(x), 2), std = sd(x)))
```

R output

	RS1	RS2	RS3	RS4
mean	0	0	0	0
std	1	1	1	1

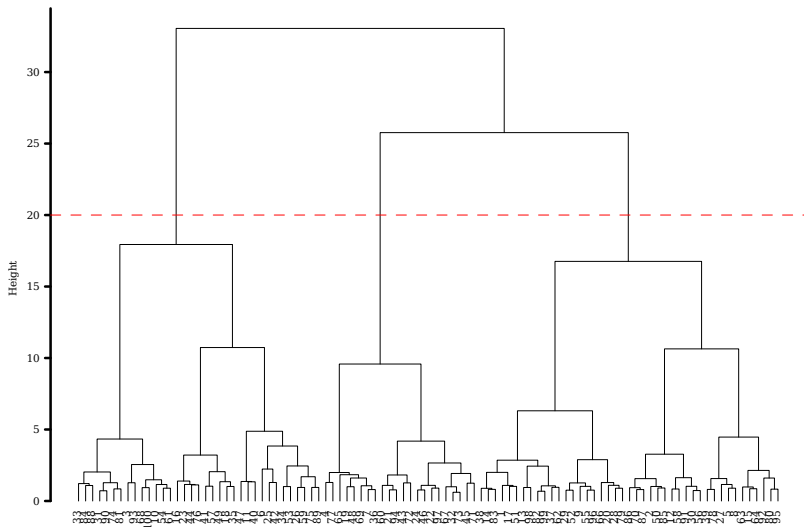
- Utilisation des K-means sur les axes factorielles
- Ne pas oublier d'initialiser le seed pour rendre les résultats reproductibles

```
set.seed(1234)
hh_part <- kmeans(mcadata, centers = 100,
nstart = 20, iter.max = 500)
```

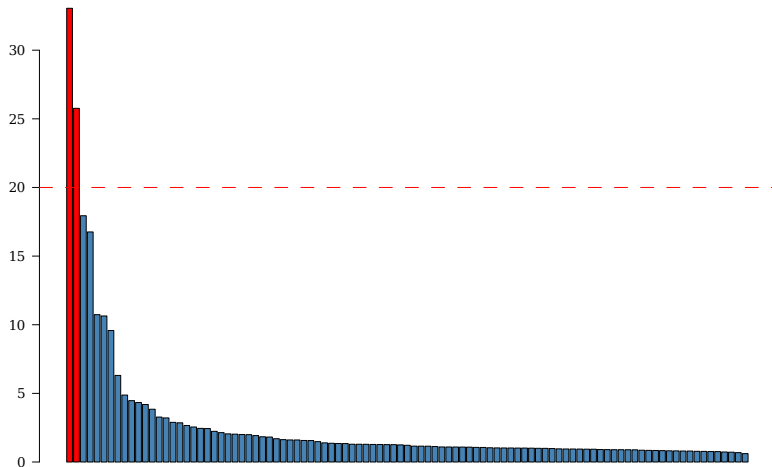
- CAH sur les centres de gravités des classes

```
hh_hclust <- hclust(dist(hh_part$centers), method = "ward")
```

# Mise en oeuvre sous R



# Mise en oeuvre sous R



- Nombre de centre par groupe

```
hh_tree <- cutree(hh_hclust, k = 3)
table(hh_tree)
```

---

R output

---

hh\_tree

1	2	3
43	35	22

---

- Associer à chaque ménage à son nouveau groupe

```
dataclust <- data.frame(kclust = seq_along(hh_tree), hclust = hh_tree)
household$kclust <- hh_part$cluster
household <- merge(household, dataclust, by = "kclust")
```

- Nombre final de ménages par groupe

```
table(household$hclust)
```

---

R output

```
  1    2    3
6894 3253 3418
```

---

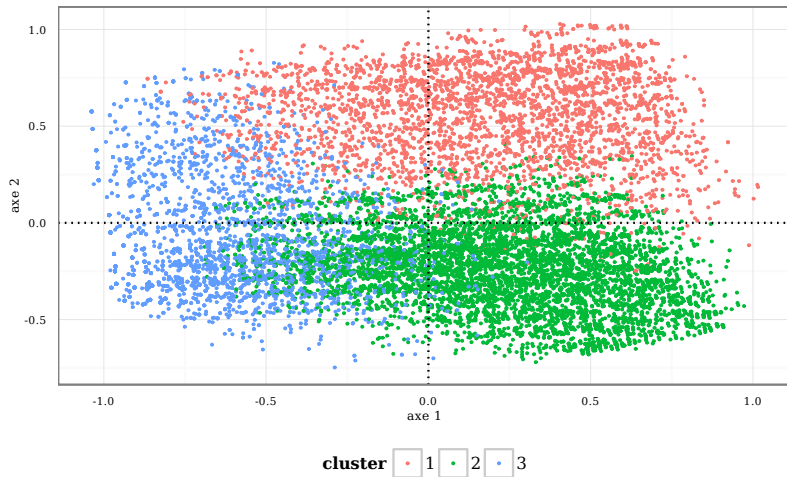
# Mise en oeuvre sous R

- Étape finale de consolidation des groupes précédemment formés.
- Utilisation de la médiane pour rendre les résultats moins sensibles aux valeurs extrêmes

```
centers <- by(mcadata, household$hclust, function(x) apply(x, 2, median))  
centers <- matrix(unlist(centers), ncol = ncol(mcadata), byrow = TRUE)  
consol <- kmeans(mcadata, centers = centers, iter.max = 50, nstart = 10)
```



# Mise en oeuvre sous R



# Conclusion

- R est une option viable et performante pour l'analyse de données
- R est libre et gratuit donc le seul coût est celui d'apprentissage
- Il s'agit juste d'une introduction plusieurs méthodes n'ont pas été explorées