# . *Rapport De Mini-Projet*

## *En Entrepôt et Fouille de Données*

## *(EFD)*

***Domaine : Informatique***
***Option :*** *SITW*

BOUMESSAOUD ABDELKADER

# THEME

CLASSIFICATION NON SUPERVISEE ET OLAP « CUBES »

Encadré par : Mme. Safia Nait Bahloul Université Oran1

# Table des matières

# Liste des figures

# Chapitre 1

# Application de K-means sur le DataSet : USArrests

## 1.1 Le DataSet

### 1.1.1 Préambule

Cet ensemble de données contient des statistiques sur les arrestations pour 100 000 habitants pour agression, meurtre et viol dans chacun des 50 États américains en 1973. Le pourcentage de la population vivant dans les zones urbaines est également indiqué.

### 1.1.2 Description du DataSet

| Variables | Type | Explication |
|---|---|---|
| Murder | Numérique | Arrestations pour meurtre (par 100 000 habitants) |
| Assult | Numérique | Arrestations pour voies de fait (par 100 000 habitants) |
| Rape | Numérique | Arrestations pour viol (par 100 000 habitants) |
| UrbanPop | Numérique | Pourcentage de population urbaine |

*Figure 1 : tableau de description du dataset USArrests*

## 1.2 Expression Du Problème

On veut analyser divers États Américains sur leurs similarités à partir du DataSet USArrests afin de les regrouper (clustering) en conséquence.

## 1.3 Classification Automatique Avec K-Means

### 1.3.1 Chargement Du DataSet

Nous commençons par standardiser les données du DataSet pour les rendrent des variables comparables.

```
#Chargement Du DataSet
DS <- scale(USArrests)
head(DS)
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

*Figure 2 : code R pour le chargement du dataset*

### 1.3.2 Déterminer le nombre optimal pour K (nombre de classes)

Déterminer le nombre optimal de classes dans un ensemble de données est fondamental dans le clustering k-means, qui oblige l'utilisateur à spécifier le nombre K.

Sauf que ce nombre est subjectif et dépend de la méthode utilisée pour mesurer les similarités et des paramètres utilisés pour le partitionnement.

La solution que nous utiliserons sera d'appliquer les 3 méthodes de partitionnement: Elbow, Silhouette et Gap Statistic. Puis comparerons les résultats donnés.

## Méthode Elbow

Elle consiste à calculer la variance des différents volumes de clusters envisagés, puis à placer les variances obtenues sur un graphique.

## Méthode Silhouette

Elle se base sur le coefficient de silhouette, qui est une mesure de qualité d'une partition d'un ensemble de données en classification automatique. Pour chaque point, son coefficient de silhouette est la différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation).

## Méthode Gap Statistic

Elle est basée sur la comparaison de la variation totale intra-cluster pour différentes valeurs de k avec leurs valeurs attendues sous une distribution de référence nulle des données

```r
#Déterminer le nombre optimal pour K (nombre de classes)
#Methode Elbow
fviz_nbclust(DS, kmeans, method = "wss") + geom_vline(xintercept = 4, linetype = 2) + labs(subtitle =
"Methode Elbow")

#Methode Silhouette
fviz_nbclust(DS, kmeans, method = "silhouette") + labs(subtitle = "Methode Silhouette")

#Methode Gap statistic
set.seed(123)
fviz_nbclust(DS, kmeans, nstart = 25,  method = "gap_stat", nboot = 500) + labs(subtitle = "Methode
Gap Statistic")
```

*Figure 3 : code R pour les méthodes de détermination de K*

## Résultat Et Conclusion

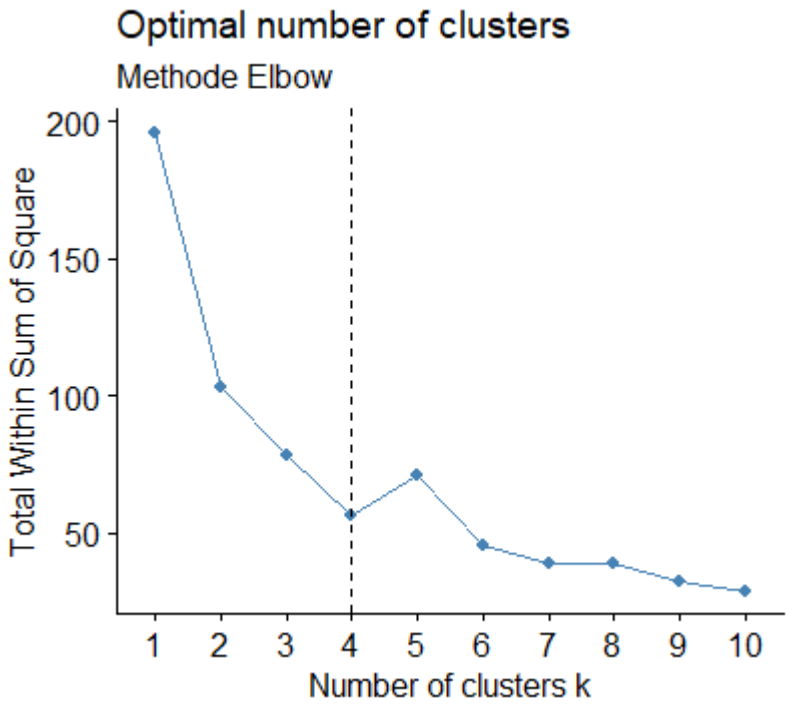


*Figure 4 : courbe du nombre K selon la méthode Elbow*

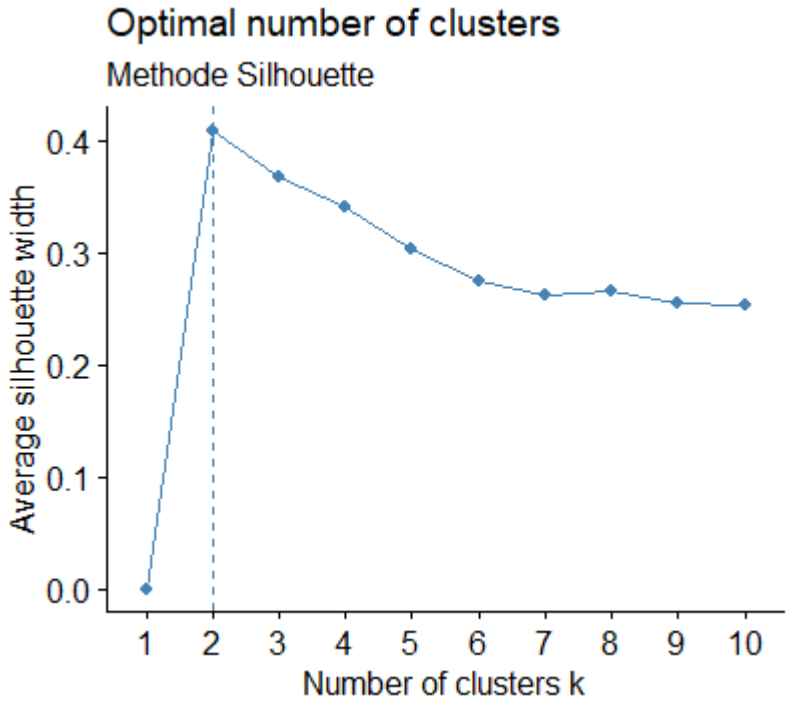


*Figure 5 : courbe du nombre K selon la methode Silhouette*

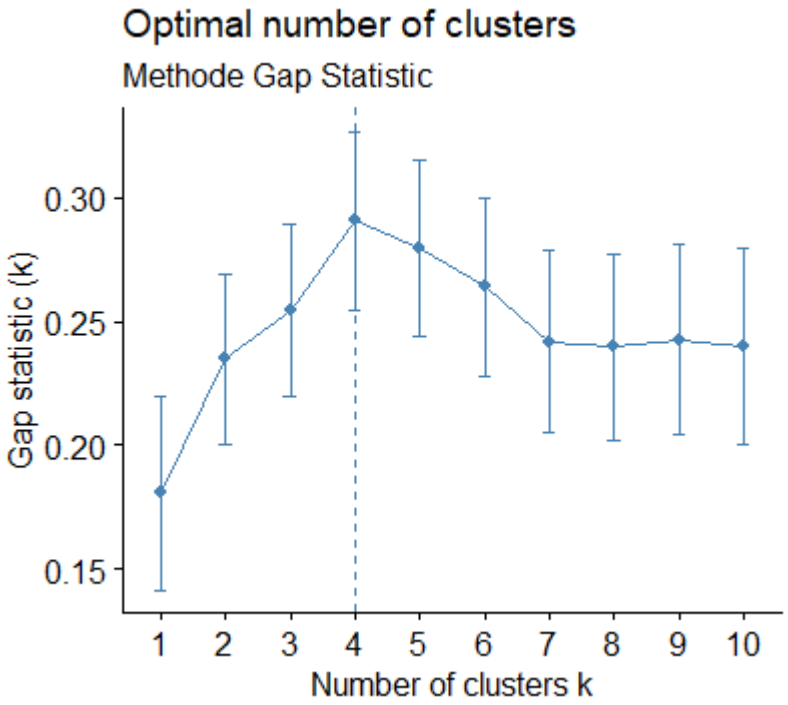


*Figure 6 : courbe du nombre K selon la méthode Gap Statistic*

>Méthode Elbow : solution à 4 classes suggérée.

>Méthode silhouette : solution à 2 classes suggérée.

>Méthode Gap Statistic : solution à 4 classes suggérée.

Selon ces observations, il est possible de définir k = 4 comme le nombre optimal de classes dans les données du DataSet.

### 1.3.3 Clustering avec K-Means et interprétation des résultats

```r
#Analyse K-Means
DSK <- kmeans(DS, 4) # K = 4
fviz_cluster(DSK, data = DS)
```

*Figure 7 : code R pour l'analyse K-Means*

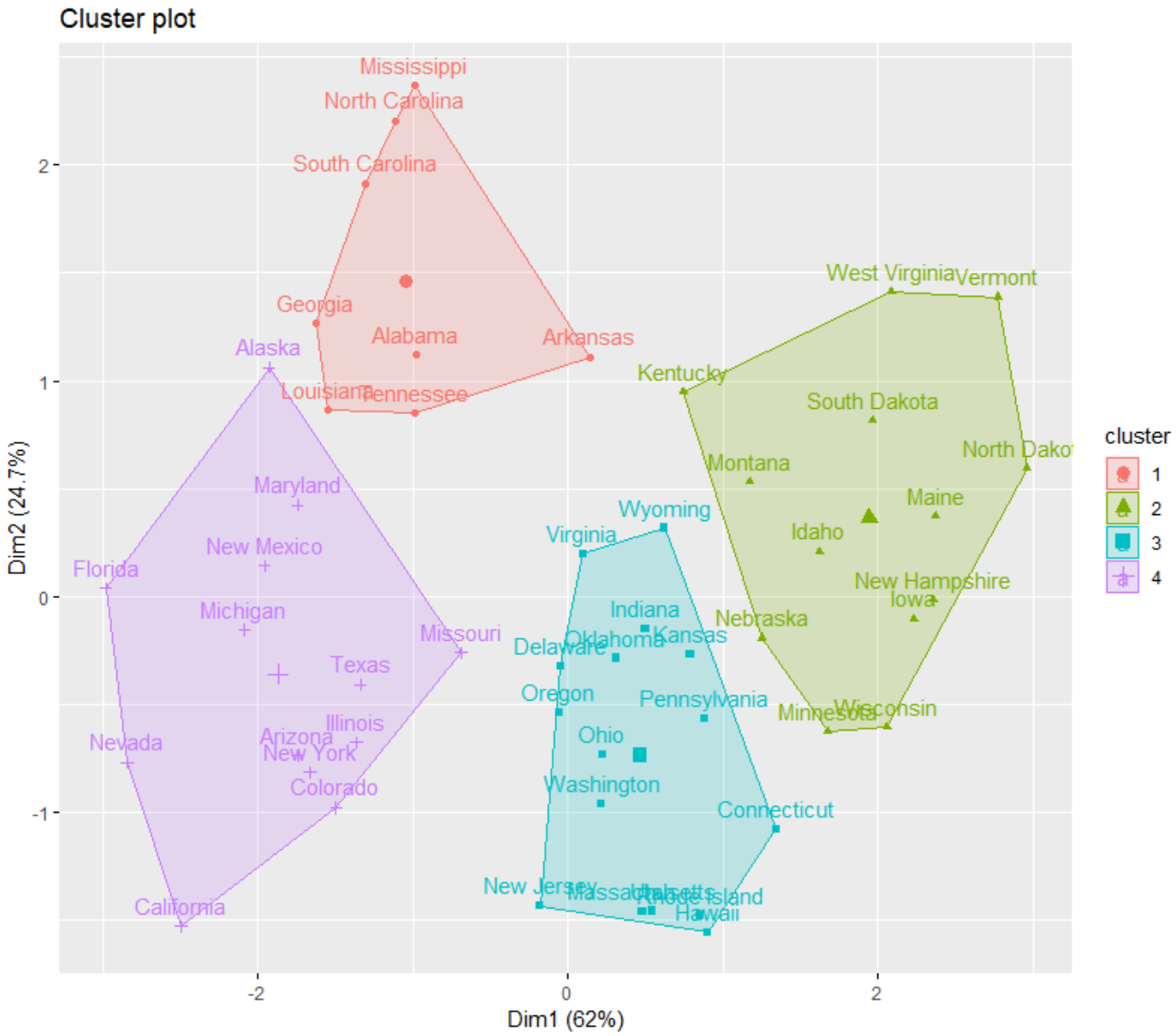


*Figure 8 : Factor Map des clusters*

## Interprétation

Classe 1 : Cluster des individus : Georgia, Mississippi, North et South Carolina… ect, et l'on remarque :

- Fortes valeurs pour les variables Murder et Assault (La plus extrême a la moins extrême).
- Faible valeurs pour la variable UrbanPop.

Classe 2 : Cluster des individus : Iowa, Maine, New Hampshire, South Dakota, West Virginia… ect, et l'on remarque :

- Faibles valeurs pour les variables Assault, Rape, Murder et UrbanPop (La plus extrême a la moins extrême).

Classe 3 : Cluster des individus : Hawaï, Massachusetts, New jersey, Utah… ect, et l'on remarque :

- Fortes valeurs pour la variable UrbanPop.
- Faible valeurs pour la variable Murder.

Classe 4 : Cluster des individus : Alaska, California, Florida, Michigan… ect, et l'on remarque :

- Forte valeurs pour les variables Rape, Assault, UrbanPop et Murder (La plus extrême a la moins extrême).

# Chapitre 2

# Représentation et opération sur un cube OLAP

## 2.1 Représentation En Base De Cube

### 2.1.1 Example choisis

Dans notre exemple on veut modéliser les transactions de ventes de **produit** (prod) informatique dans des **villes** (localisation) Algériennes et Françaises durant les **années** (dates) 2022 et 2023. L'exemple est illustré dans le schéma suivant :

### 2.1.2 Création du Cube

**Création des tables de dimensions**

```r
#Creation des tables de dimensions
table_ville <-
  data.frame(cle=c("ALG", "ORN", "TLC", "MRS", "NIC"),
             name=c("Alger", "Oran", "Tlemcen", "Marseille", "Nice"),
             pays=c("DZ", "DZ", "DZ", "FR", "FR"))

table_mois <-
  data.frame(cle=1:12,
             desc=c("Jan", "Fev", "Mar", "Avr", "Mai", "Jun", "Jul", "Aut", "Sep", "Oct", "Nov",
"Dec"),
             trimestre=c("T1","T1","T1","T2","T2","T2","T3","T3","T3","T4","T4","T4"))

table_prod <-
  data.frame(cle=c("Imprimante", "Tablette", "Laptop", "Ecran"),
             prix=c(225, 570, 1120, 360))
```

*Figure 9 : code R pour la création des tables de dimensions*

## Résultat

```
#Resultat
head(table_ville)

  cle      name pays
1 ALG     Alger   DZ
2 ORN      Oran   DZ
3 TLC   Tlemcen   DZ
4 MRS Marseille   FR
5 NIC      Nice   FR

head(table_mois)

  cle desc trimestre
1   1  Jan        T1
2   2  Fev        T1
3   3  Mar        T1
4   4  Avr        T2
5   5  Mai        T2
6   6  Jun        T2

head(table_prod)

       cle prix
1 Imprimante  225
2   Tablette  570
3     Laptop 1120
4      Ecran  360
```

*Figure 10 : résultat R pour la création des tables de dimensions*

## Fonction de création d’une table de ventes aléatoire

```r
#Fonction de creation d’une table de ventes aléatoire
gen_ventes <- function(nbr_cells)
{
  loc <- sample(table_ville$cle, nbr_cells, replace=T, prob=c(2,2,1,1,1))
  date_mois <- sample(table_mois$cle, nbr_cells, replace=T)
  date_annee <- sample(c(2022, 2023), nbr_cells, replace=T)
  prod <- sample(row.names(table_prod), nbr_cells, replace=T, prob=c(1, 3, 2,4))
  unit <- sample(c(1,2), nbr_cells, replace=T, prob=c(10, 3))
  montant <- unit*table_prod[prod,]$prix
  ventes <- data.frame(mois=date_mois,
                       annee=date_annee,
                       loc=loc,
                       prod=table_prod[prod,]$cle,
                       unit=unit,
                       montant=montant)
  #Trier par date de vente
  ventes <- ventes[order(ventes$annee, ventes$mois),]
  row.names(ventes) <- NULL
  return(ventes)
}

#creation d’une table de ventes aléatoire
table_ventes <- gen_ventes(50)
```

*Figure 11 : code R de la fonction de création d’une table de ventes aléatoire*

## Résultat

```
#Resultat
head(table_ventes)

  mois annee loc       prod unit montant
1    1  2022 ORN      Ecran    1     360
2    1  2022 ALG Imprimante    1     225
3    1  2022 ALG   Tablette    2    1140
4    1  2022 MRS     Laptop    2    2240
5    2  2022 TLC   Tablette    1     570
6    4  2022 MRS   Tablette    1     570
```

*Figure 12 : résultat R de la création d'une table de ventes aléatoire*

## Création du Cube

Maintenant, nous transformons cette table de faits en un cube à plusieurs dimensions. Chaque cellule du cube représente une valeur agrégée pour une combinaison unique de chaque dimension.

```
#Creation du Cube
revenue_cube <-
  tapply(table_ventes$montant,
         table_ventes[,c("prod", "mois", "annee", "loc")],
         FUN=function(x){return(sum(x))})
```

*Figure 13 : code R de la création du Cube*

## Résultat

| ville | produit | annee | mois | unit | montant |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |

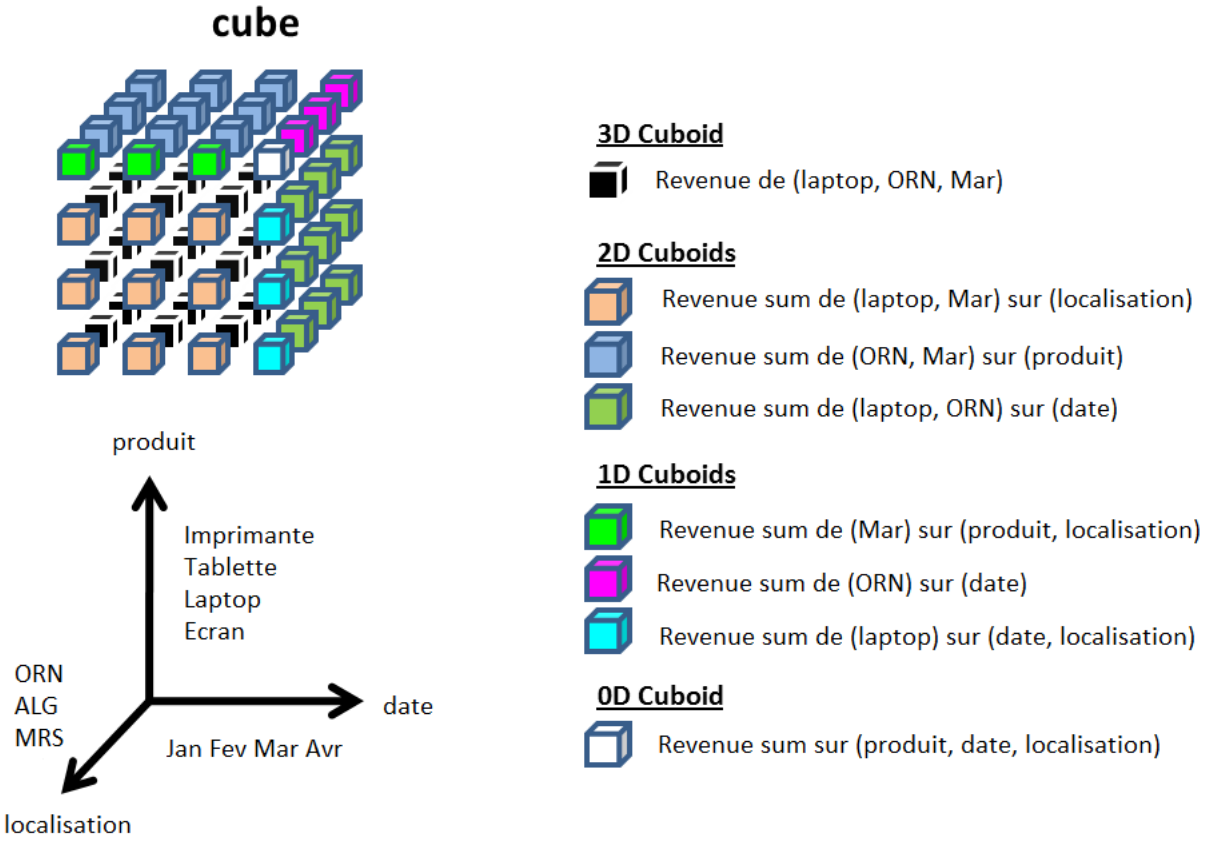


*Figure 14 : Cube Multidimensionnelle*

*Figure 15 : résultat R de la création du Cube (partie 1)*

```
#Resultat (Cellules du Cube)
revenue_cube

, , annee = 2022, loc = ALG

            mois
prod            1    2    3    4   5    6    7   8    9  10   11   12
  Ecran      1440  720 1080  720 360 1800 1800  NA  720 360  360 1440
  Imprimante   NA   NA   NA   NA  NA   NA  225 225  225 450  225   NA
  Laptop       NA   NA   NA 2240  NA 2240 1120  NA 3360  NA 2240 1120
  Tablette   1140 1710 1710  570 570  570 4560 570 2280 570  570 1710

, , annee = 2023, loc = ALG

            mois
prod            1    2    3    4    5    6    7    8    9   10   11   12
  Ecran       720 1080  360 1440 1440  720  720   NA  360 1800  720  720
  Imprimante  225   NA   NA   NA   NA  675   NA  225   NA  225  225  675
  Laptop     1120 1120 2240 2240 1120 3360 5600 1120 4480 1120   NA 2240
  Tablette   2280  570 1710  570 2280  570  570 2850  570 5130 3420 1710

, , annee = 2022, loc = MRS

            mois
prod            1    2    3    4    5    6    7   8    9   10   11   12
  Ecran      1080 1440 1080   NA   NA 1080  360 360 1800   NA   NA 1080
  Imprimante   NA   NA   NA   NA   NA   NA   NA  NA   NA   NA  225   NA
  Laptop       NA 1120 1120   NA   NA 1120 1120  NA 1120   NA 2240   NA
  Tablette   1140  570   NA 1140 1710  570 1710 570   NA 1710  570  570

, , annee = 2023, loc = MRS

            mois
prod            1    2    3   4    5    6    7   8    9   10  11   12
  Ecran       720 1080 1080 720  360  360   NA 360   NA 1080  NA  720
  Imprimante 225  450   NA  NA   NA   NA   NA  NA   NA   NA 225   NA
  Laptop      NA 3360   NA  NA   NA 1120   NA  NA 2240 1120  NA 3360
  Tablette   570   NA   NA 570 2280   NA 1140  NA  570   NA  NA   NA
```

```
, , annee = 2022, loc = NIC

            mois
prod            1    2    3    4   5    6    7    8   9   10   11  12
  Ecran        NA   NA  360  360 360  720  360 1080 360  360   NA 720
  Imprimante   NA  225   NA  450  NA  225   NA  225  NA   NA   NA 225
  Laptop     1120 1120 1120 1120  NA   NA 2240 1120  NA 1120 1120  NA
  Tablette   1140 1140 1140 1140 570 1710 1710   NA 570 1140 1710  NA

, , annee = 2023, loc = NIC

            mois
prod            1    2    3    4    5    6    7   8   9   10  11   12
  Ecran       360  360   NA   NA  720 1080  720 720 360  360  NA   NA
  Imprimante   NA   NA   NA   NA   NA   NA   NA  NA  NA   NA  NA   NA
  Laptop       NA 1120 1120 1120   NA 1120   NA  NA  NA 1120  NA   NA
  Tablette    570   NA   NA   NA 1140  570 1140 570 570   NA 570 2850

, , annee = 2022, loc = ORN

            mois
prod            1    2    3    4    5    6    7    8    9   10   11   12
  Ecran      1440  720 1080  720 1440 3240 1440  360  720 1080 2160 1080
  Imprimante   NA  450   NA   NA  450  675  225  225  225   NA  225  450
  Laptop     1120   NA 4480 1120   NA 2240 3360   NA 2240   NA 4480   NA
  Tablette   1140 1140 2280 1140 3420  570 1710 2850   NA 1140 2850 1140
```

*Figure 16 : résultat R de la création du Cube (partie 2)*

```
, , annee = 2023, loc = ORN

            mois
prod           1    2    3    4    5    6    7    8    9   10   11   12
  Ecran     1800 1800  720  360  720 1080 1080  720 2160   NA 1800 2880
  Imprimante   NA   NA   NA   NA  450  450  450   NA   NA   NA   NA   NA
  Laptop    1120 1120 2240 2240 1120   NA   NA 1120 2240 1120 3360   NA
  Tablette   570 1710   NA  570 1140 1140  570 2850 1710   NA 2280 1710

, , annee = 2022, loc = TLC

            mois
prod           1    2   3    4    5   6    7  8    9   10 11  12
  Ecran      360  720 360 1080 2160  NA   NA NA 1080 1440 NA 720
  Imprimante  NA   NA 225   NA  225  NA  450 NA  225   NA NA 450
  Laptop      NA 1120  NA   NA   NA  NA 3360 NA   NA 1120 NA  NA
  Tablette    NA   NA  NA   NA   NA 570   NA NA   NA  570 NA  NA

, , annee = 2023, loc = TLC

            mois
prod           1   2   3   4  5  6    7    8    9   10   11   12
  Ecran       NA 360  NA 360 NA NA   NA   NA   NA  360   NA 1080
  Imprimante 225 450  NA  NA NA NA   NA   NA  450   NA   NA  225
  Laptop      NA  NA  NA  NA NA NA   NA 2240 2240   NA 3360 1120
  Tablette   570 570 570  NA NA NA 1140   NA 1710 2280   NA 1140
```

*Figure 17 : résultat R de la création du Cube (partie 3)*

## 2.2 Operations Sur Le Cube

Voici quelques opérations courantes d'OLAP

-Dice

-Slice

-Rollup

-Drilldown

### 2.2.1 Dice

"Dice" consiste à limiter chaque dimension à une certaine plage de valeurs, tout en gardant le même nombre de dimensions dans le cube résultant. Par exemple, nous pouvons nous concentrer sur les ventes qui se déroulent en [janvier/février/mars, laptop/tablette, ORN/ALG].

```
#Dice
revenue_cube[c("Tablette","Laptop"),c("1","2","3"), , c("ORN","ALG")]

, , annee = 2022, loc = ORN

          mois
prod         1    2    3
  Tablette 1140 1140 2280
  Laptop   1120   NA 4480

, , annee = 2023, loc = ORN

          mois
prod         1    2    3
  Tablette  570 1710   NA
  Laptop   1120 1120 2240

, , annee = 2022, loc = ALG

          mois
prod         1    2    3
  Tablette 1140 1710 1710
  Laptop     NA   NA   NA

, , annee = 2023, loc = ALG

          mois
prod         1    2    3
  Tablette 2280  570 1710
  Laptop   1120 1120 2240
```

*Figure 18 : code R de l'opération OLAP Dice*

### 2.2.2 Slice

"Slice" consiste à fixer certaines dimensions pour analyser les dimensions restantes. Par exemple, nous pouvons nous concentrer sur les ventes qui se déroulent en "2022", "Jan", ou nous pouvons nous concentrer sur les ventes qui se déroulent en "2022", "Jan", "Tablette".

```
#Slice
#Donnes du Cube en Jan, 2022
revenue_cube[, "1", "2022",]

            loc
prod          ALG  MRS  NIC  ORN TLC
  Ecran      1440 1080   NA 1440 360
  Imprimante   NA   NA   NA   NA  NA
  Laptop       NA   NA 1120 1120  NA
  Tablette   1140 1140 1140 1140  NA

#Donnes du Cube en Jan, 2022
revenue_cube["Tablette", "1", "2022",]

 ALG  MRS  NIC  ORN  TLC
1140 1140 1140 1140   NA
```

*Figure 19 : code R de l'opération OLAP Slice*

### 2.2.3 Rull-Up

"Rollup" consiste à appliquer une fonction d'agrégation pour réduire un certain nombre de dimensions. Par exemple, nous voulons nous concentrer sur le chiffre d'affaires annuel de chaque produit et réduire la dimension géographique (c'est-à-dire : nous ne nous soucions pas de l'endroit où nous avons vendu notre produit).

```r
#Rollup
apply(revenue_cube, c("annee", "prod"), FUN=function(x) {return(sum(x, na.rm=TRUE))})
```

```
      prod
annee  Ecran Imprimante Laptop Tablette
  2022 47160       7425  54880    59280
  2023 38520       5850  67200    57570
```

*Figure 20 : code R de l'opération OLAP Rull-Up*

### 2.2.4 Drill-Down

"Drilldown" est l'inverse de "Rollup" et applique une fonction d'agrégation à un niveau de granularité plus fin. Par exemple, nous voulons nous concentrer sur les revenus annuels et mensuels de chaque produit et réduire la dimension géographique (c'est-à-dire : nous ne nous soucions pas de l'endroit où nous avons vendu notre produit).

```
#Drilldown
  apply(revenue_cube, c("annee", "mois", "prod"), FUN=function(x) {return(sum(x, na.rm=TRUE))})


, , prod = Ecran

      mois
annee    1    2    3    4    5    6    7    8    9   10   11   12
  2022 4320 3600 3960 2880 4320 6840 3960 1800 4680 3240 2520 5040
  2023 3600 4680 2160 2880 3240 3240 2520 1800 2880 3600 2520 5400

, , prod = Imprimante

      mois
annee    1   2   3   4   5    6   7   8   9  10  11   12
  2022   0 675 225 450 675  900 900 675 675 450 675 1125
  2023 675 900   0   0 450 1125 450 225 450 225 450  900

, , prod = Laptop

      mois
annee    1    2    3    4    5    6     7    8     9   10    11   12
  2022 2240 3360 6720 4480    0 5600 11200 1120  6720 2240 10080 1120
  2023 2240 6720 5600 5600 2240 5600  5600 4480 11200 4480  6720 6720

, , prod = Tablette

      mois
annee    1    2    3    4    5    6    7    8    9   10   11   12
  2022 4560 4560 5130 3990 6270 3990 9690 3990 2850 5130 5700 3420
  2023 4560 2850 2280 1710 6840 2280 4560 6270 5130 7410 6270 7410
```

*Figure 21 : code R de l'opération OLAP Drill-Down*