

# Mortalité Hospitaliere

## load data deces

Dans cette partie nous allons travailler sur des données, et utiliser un jeu de données issue de la base de données mysql.

Le jeu de données que nous allons utiliser est un extrait de l'enquête sur la mortalité hospitaliere réalisée par la DSP de djelfa en 2021. Il contient 13946 deces et 14 variables.

Un data frame (ou tableau de données, ou table) est un type d'objet R qui contient des données au format tabulaire, avec les observations en ligne et les variables en colonnes, comme dans une feuille de tableur de type Excel.

Si on se contente d'exécuter le nom de notre tableau de données R va, comme à son habitude, nous l'afficher dans la console, ce qui est tout sauf utile.

Une autre manière d'afficher le contenu du tableau est de cliquer sur l'icône en forme de tableau à droite du nom de l'objet dans l'onglet Environment :

Ou d'utiliser la fonction View :

Il est important de comprendre que l'objet data contient l'intégralité des données du tableau. On voit donc qu'un objet peut contenir des données de types très différents (simple nombre, texte, vecteur, tableau de données entier), et être potentiellement de très grande taille

Un data frame peut être manipulé comme les autres objets vus précédemment. On peut par exemple faire : ce qui va entraîner la copie de l'ensemble de nos données dans un nouvel objet nommé data. Ceci peut paraître parfaitement inutile mais a en fait l'avantage de fournir un objet avec un nom beaucoup plus court, ce qui diminuera la quantité de texte à saisir par la suite.

Pour résumer, comme nous avons désormais décidé de saisir nos commandes dans un script et non plus directement dans la console, les premières lignes de notre fichier de travail sur les données de l'enquête sur la mortalité hospitaliere pourraient donc ressembler à ceci :

```
source("connection_db.R")
source("sig.R")
```

## inspect data deces

Un tableau étant un objet comme un autre, on peut lui appliquer des fonctions. Par exemple, nrow et ncol retournent le nombre de lignes et de colonnes du tableau.

La fonction dim renvoie ses dimensions, donc les deux nombres précédents.

La fonction names retourne les noms des colonnes du tableau, c'est-à-dire la liste de nos variables.

Enfin, la fonction str renvoie un descriptif plus détaillé de la structure du tableau. Elle liste les différentes variables, indique leur type et affiche les premières valeurs.

À noter que sous RStudio, on peut afficher à tout moment la structure d'un objet en cliquant sur l'icône de triangle sur fond bleu à gauche du nom de l'objet dans l'onglet Environment.

```
nrow(data)
```

```
## [1] 6175
```

```
ncol(data)
```

```
## [1] 14
```

```
dim(data)
```

```
## [1] 6175 14
```

```
names(data)
```

```
## [1] "DINS"          "WILAYAR"        "COMMUNER"       "LD"
## [5] "STRUCTURED"    "SERVICEHOSPIT" "DUREEHOSPIT"    "SEX"
## [9] "Years"         "Days"           "Profession"     "CD"
## [13] "CODECIMO"      "CODECIM"
```

```
#sum(is.na(data))
```

```
#colMeans(is.na(data))
```

## view data structure deces

str

```
#str(data)
```

## glimpse

```
glimpse(data)
```

```
## Rows: 6,175
## Columns: 14
## $ DINS      <date> 2020-01-27, 2020-01-24, 2020-01-24, 2020-01-24, 2020-01-~
## $ WILAYAR   <int> 17000, 17000, 17000, 17000, 17000, 17000, 17000, 17000, ~
## $ COMMUNER  <int> 935, 935, 947, 920, 935, 935, 935, 948, 935, 935, 935, 9~
## $ LD        <fct> SSP, DOM, SSP, SSP, SSP, SSP, SSP, SSP, SSP, SSP, SSP, S~
## $ STRUCTURED <fct> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, ~
## $ SERVICEHOSPIT <fct> 7, 20, 20, 10, 20, 20, 10, 20, 20, 20, 7, 16, 20, 18, 20~
## $ DUREEHOSPIT <int> 0, 0, 4, 5, 0, 1, 0, 2, 0, 1, 0, 20, 0, 4, 7, 18284, 4, ~
## $ SEX       <fct> M, M, M, F, M, F, M, F, M, M, M, F, M, M, M, F, F, M, M, ~
## $ Years     <int> 71, 85, 77, 0, 81, 31, 0, 31, 0, 93, 31, 61, 63, 48, 64, ~
## $ Days      <int> 26100, 31101, 28125, 5, 29638, 11678, 0, 11599, 0, 34017~
## $ Profession <fct> 0, 0, 0, 12, 0, 0, 12, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ CD        <fct> CN, CI, CN, CN, CN, CN, CN, CN, CI, CN, CN, CN, CN, CN, CN, ~
## $ CODECIMO  <int> 10, 18, 10, 16, 6, 18, 16, 18, 1, 4, 14, 10, 9, 10, 2, 9~
## $ CODECIM   <int> 751, 1370, 751, 1145, 517, 1333, 1155, 1370, 35, 407, 10~
```

skimr

```
#library(skimr)
```

```
#skim(data)
```

## modalité des variable deces

```
unique(data$LD)
```

```
## [1] SSP DOM AAP VP SSPV  
## Levels: AAP DOM SSP SSPV VP
```

```
unique(data$SEX)
```

```
## [1] M F  
## Levels: F M
```

```
# Les facteurs prennent leurs valeurs dans un ensemble de modalités prédéfinies  
# et ne peuvent enprendre d'autres
```

```
levels(data$SEX)
```

```
## [1] "F" "M"
```

```
unique(data$Profession)
```

```
## [1] 0 12 1 13 7 11 14 3 6 4 20 19 16 5  
## Levels: 0 1 3 4 5 6 7 9 11 12 13 14 15 16 19 20
```

```
unique(data$CD)
```

```
## [1] CN CI CV  
## Levels: CI CN CV
```

## Accéder aux valeurs des variables d'un tableau

Une opération très importante est l'accès aux variables du tableau (à ses colonnes) pour pouvoir les manipuler, effectuer des calculs, etc. On utilise pour cela l'opérateur \$, qui permet d'accéder aux colonnes du tableau. Ainsi, si l'on tape :

R va afficher l'ensemble des valeurs de la variable sexe dans la console, ce qui est à nouveau fort peu utile. Mais cela nous permet de constater que d\$sexe est un vecteur de chaînes de caractères tels qu'on en a déjà rencontré précédemment.

La fonction table\$colonne renvoie donc la colonne nommée colonne du tableau table, c'est-à-dire un vecteur, en général de nombres ou de chaînes de caractères.

```
## data$DINS
```

## look\_for

```
#look_for(data)
```

## tbl\_summary

```
tbl_summary(data)
```

Characteristic	N = 6,175
Date du décès	2020-01-01 to 2021-12-31
Wilaya de résidence	17,150 (1,992)
Commune de résidence	934 (110)
Lieux du décès	
AAP	348 (5.6%)

Characteristic	N = 6,175
DOM	497 (8.0%)
SSP	5,237 (85%)
SSPV	6 (<0.1%)
VP	87 (1.4%)
Structure sanitaire	
1	1,591 (26%)
2	1,205 (20%)
3	905 (15%)
4	746 (12%)
5	727 (12%)
6	312 (5.1%)
7	206 (3.3%)
8	0 (0%)
18	411 (6.7%)
19	72 (1.2%)
Service d'hospitalisation	
0	128 (2.1%)
1	11 (0.2%)
2	77 (1.2%)
3	1 (<0.1%)
4	0 (0%)
5	40 (0.6%)
6	465 (7.5%)
7	424 (6.9%)
8	15 (0.2%)
9	4 (<0.1%)
10	1,225 (20%)
11	5 (<0.1%)
12	1 (<0.1%)
14	0 (0%)
15	138 (2.2%)
16	173 (2.8%)
17	0 (0%)
18	468 (7.6%)
19	1 (<0.1%)
20	2,342 (38%)
21	639 (10%)
24	1 (<0.1%)
25	17 (0.3%)
Durée d'hospitalisation	881 (3,912)
Sexe	
F	2,566 (42%)
M	3,609 (58%)
Age (Année)	49 (34)
Age (Jours)	17,869 (12,569)
Profession	
0	2,951 (48%)
1	1,980 (32%)
3	3 (<0.1%)
4	4 (<0.1%)
5	1 (<0.1%)
6	11 (0.2%)

Characteristic	N = 6,175
7	16 (0.3%)
9	0 (0%)
11	36 (0.6%)
12	200 (3.2%)
13	266 (4.3%)
14	6 (<0.1%)
15	0 (0%)
16	4 (<0.1%)
19	205 (3.3%)
20	492 (8.0%)
Cause du décès	
CI	351 (5.7%)
CN	5,621 (91%)
CV	203 (3.3%)
Capitre CIM10	9 (6)
Titre CIM10	690 (474)

## summary data deces

```
summary(data)
```

```
##          DINS          WILAYAR          COMMUNER          LD
##  Min.   :2020-01-01  Min.   : 1000  Min.   :   1.0  AAP : 348
## 1st Qu.:2020-07-19 1st Qu.:17000 1st Qu.: 916.0  DOM : 497
## Median :2021-02-10 Median :17000 Median : 924.0  SSP :5237
## Mean   :2021-01-23 Mean   :17150 Mean   : 934.1  SSPV: 6
## 3rd Qu.:2021-08-07 3rd Qu.:17000 3rd Qu.: 935.0  VP  : 87
## Max.   :2021-12-31 Max.   :47000 Max.   :2291.0
##
##  STRUCTURED  SERVICEHOSPIT  DUREEHOSPIT  SEX  Years
## 1      :1591  20      :2342  Min.   : -3580.0  F:2566  Min.   : 0.00
## 2      :1205  10      :1225  1st Qu.: 0.0  M:3609  1st Qu.: 0.00
## 3      : 905  21      : 639  Median : 1.0  Median : 60.00
## 4      : 746  18      : 468  Mean   : 881.3  Mean   : 48.52
## 5      : 727  6      : 465  3rd Qu.: 5.0  3rd Qu.: 79.00
## 18     : 411  7      : 424  Max.   :29220.0  Max.   :100.00
## (Other): 590  (Other): 612
##  Days  Profession  CD  CODECIMO  CODECIM
## Min.   : -279.0  0      :2951  CI: 351  Min.   : 0.000  Min.   : 0.0
## 1st Qu.: 325.5  1      :1980  CN:5621 1st Qu.: 2.000 1st Qu.: 217.0
## Median :22257.0 20     : 492  CV: 203  Median :10.000 Median : 709.0
## Mean   :17868.9 13     : 266  Mean   : 9.201 Mean   : 690.2
## 3rd Qu.:28883.5 19     : 205  3rd Qu.:16.000 3rd Qu.:1154.0
## Max.   :36857.0 12     : 200  Max.   :21.000 Max.   :2039.0
## (Other): 81
```

## view data deces

Si on souhaite afficher seulement les premières ou dernières valeurs d'une variable, on peut utiliser les fonctions `head` et `tail`.

Le deuxième argument numérique permet d'indiquer le nombre de valeurs à afficher.

head

```
head(data,6)
```

```
##          DINS WILAYAR COMMUNER LD STRUCTURED SERVICEHOSPIT DUREEHOSPIT SEX
## 1 2020-01-27  17000      935 SSP          3              7            0    M
## 2 2020-01-24  17000      935 DOM          3             20            0    M
## 3 2020-01-24  17000      947 SSP          3             20            4    M
## 4 2020-01-24  17000      920 SSP          3             10            5    F
## 5 2020-01-22  17000      935 SSP          3             20            0    M
## 6 2020-01-22  17000      935 SSP          3             20            1    F
##   Years  Days Profession CD CODECIMO CODECIM
## 1    71 26100          0  CN         10    751
## 2    85 31101          0  CI         18   1370
## 3    77 28125          0  CN         10    751
## 4     0     5         12  CN         16   1145
## 5    81 29638          0  CN          6    517
## 6    31 11678          0  CN         18   1333
```

tail

```
tail(data,6)
```

```
##          DINS WILAYAR COMMUNER LD STRUCTURED SERVICEHOSPIT DUREEHOSPIT SEX
## 6170 2021-05-06  17000      929 SSP          19             21          -300    M
## 6171 2021-04-28  17000      929 VP           19             21          -326    M
## 6172 2021-01-19  17000      929 DOM          19             21          -414    M
## 6173 2021-04-15  17000      929 SSP          19             20            0    M
## 6174 2021-05-22  17000      929 DOM          19             21          -302    M
## 6175 2021-12-24  17000      916 SSP          5             21            0    F
##   Years  Days Profession CD CODECIMO CODECIM
## 6170     0     2          1  CN          0     0
## 6171    17   6447          1  CI          0     0
## 6172    82 29970          1  CN          2   253
## 6173    15   5839          1  CN          0     0
## 6174    84 30823          1  CN          0     0
## 6175    37 13739          1  CN          0     0
```

View

```
#View(data)
```

## Analyse univariée

On a donc désormais accès à un tableau de données data, dont :

- les lignes sont des observations (des individus enquêtés)
- les colonnes des variables (des caractéristiques de chacun de ces individus), et on sait accéder à ces variables grâce à l'opérateur \$.

Si on souhaite analyser ces variables, les méthodes et fonctions utilisées seront différentes selon qu'il s'agit

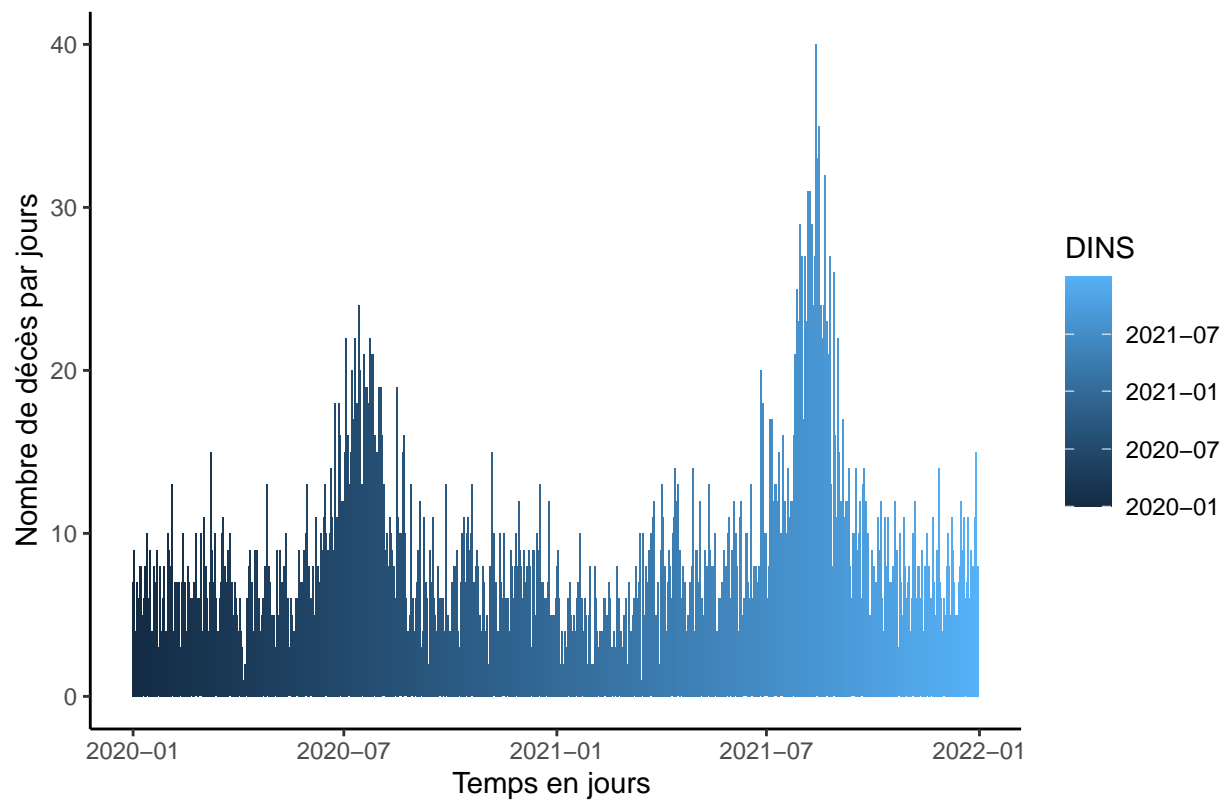
- d'une variable quantitative (variable numérique pouvant prendre un grand nombre de valeurs)
- d'une variable qualitative (variable pouvant prendre un nombre limité de valeurs appelées modalités).

## Temps

```
ts <- data %>%
  select(DINS) %>%
  group_by(DINS) %>%
  summarise(number_year = n()) %>%
  arrange(DINS) #>%

ts %>%
ggplot(aes(x = DINS, y = number_year, fill = DINS)) +
  #geom_line(color="red") +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Temps en jours ",
    y = "Nombre de décès par jours",
    title = paste("Nombre de décès par jours (Wilaya de Djelfa) du ",dt1," au ",dt2)
  )
```

Nombre de décès par jours (Wilaya de Djelfa) du 2020-01-01 au 2021-12-31



## lieux du deces

### SIG Deces par wilaya de Résidence

```
data %>%
  select(DINS,WILAYAR) %>%
  group_by(WILAYAR) %>%
```

```

summarise(number_wil = n()) %>%
arrange(desc(number_wil)) %>%
mutate(CC_1=recode(WILAYAR,
  "1000" = "1",
  "2000" = "2",
  "3000" = "3",
  "4000" = "4",
  "5000" = "5",
  "6000" = "6",
  "7000" = "7",
  "8000" = "8",
  "9000" = "9",
  "10000" = "10",
  "11000" = "11",
  "12000" = "12",
  "13000" = "13",
  "14000" = "14",
  "15000" = "15",
  "16000" = "16",
  "17000" = "17",
  "18000" = "18",
  "19000" = "19",
  "20000" = "20",
  "21000" = "21",
  "22000" = "22",
  "23000" = "23",
  "24000" = "24",
  "25000" = "25",
  "26000" = "26",
  "27000" = "27",
  "28000" = "28",
  "29000" = "29",
  "30000" = "30",
  "31000" = "31",
  "32000" = "32",
  "33000" = "33",
  "34000" = "34",
  "35000" = "35",
  "36000" = "36",
  "37000" = "37",
  "38000" = "38",
  "39000" = "39",
  "40000" = "40",
  "41000" = "41",
  "42000" = "42",
  "43000" = "43",
  "44000" = "44",
  "45000" = "45",
  "46000" = "46",
  "47000" = "47",
  "48000" = "48"
)) -> df

```

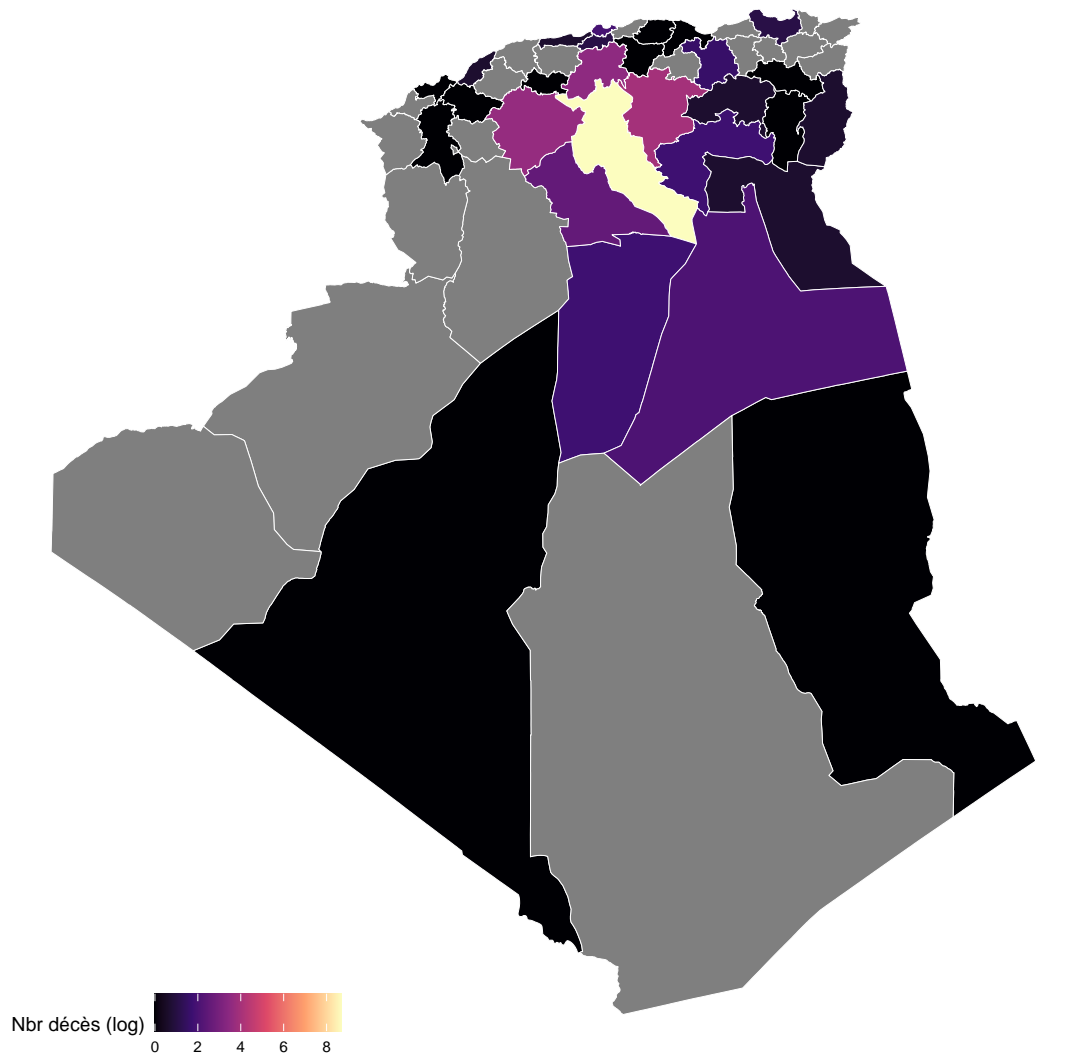


## SIG Deces par wilaya de Résidence:map

```
deces_w_djelfa <-left_join(w_algeria, df, by='CC_1')
wdjelfa <- deces_w_djelfa %>%
  select(NAME_1,CC_1,number_wil) %>%
  mutate(number_wil = log(number_wil))

ggplot()+
  geom_sf(data=wdjelfa,aes(fill=number_wil), color = "white", lwd = 0.05) +
  scale_fill_viridis_c(option = "magma", name = "Nbr décès (log)") +
  theme_map() +
  theme(legend.direction="horizontal") +
  labs(title = "Répartition des décès hospitalier par wilayas",
        subtitle = paste("Wilayas algerie du ",dt1," au ",dt2),
        caption = "Source: Dr R.TIBA \n Praticien inspecteur santé publique \n DSP Wilaya de Djelfa")+
  coord_sf(crs = "+proj=robin")
```

Répartition des décès hospitalier par wilayas  
Wilayas algerie du 2020-01-01 au 2021-12-31



Source: Dr R.TIBA  
Praticien inspecteur santé publique  
DSP Wilaya de Djelfa

## SIG Deces par commune de Résidence

```
data %>%
  select(DINS,WILAYAR,COMMUNER) %>%
  filter(WILAYAR == 17000) %>%
  group_by(COMMUNER) %>%
  summarise(number_com = n()) %>%
  arrange(desc(number_com)) %>%
  mutate(CC_2=recode(COMMUNER,
    "916" = "1701",#djelfa
    "917" = "1714",#el idrissia
    "919" = "1703",#919 El Guedid
    "920" = "1726",#920 Charef
```

```

"923" = "1727",#923 Beni Yacoub
"924" = "1731",#924 Ain Oussera
"925" = "1721",#925 Guernini
"926" = "1719",#926 Sidi Ladjel
"927" = "1733",#927 Hassi Fedoul
"928" = "1711",#928 El Khemis
"929" = "1708",#929 Birine
"931" = "1732",#931 Benhar
"932" = "1720",#932 Had-Sahary
"933" = "1709",#933 Bouira Lahdab
"934" = "1735",#934 Ain Fekka
"935" = "1704",#935 Hassi Bahbah
"939" = "1728",#939 Zaafrane
"940" = "1716",#940 Hassi el Euch
"941" = "1705",#941 Ain Maabed
"942" = "1725",#942 Dar Chioukh
"946" = "1713",#946 MLiliha
"947" = "1712",#947 Sidi Baizid
"948" = "1717",#948 Messad
"951" = "1718",#951 Guettara
"952" = "1729",#952 Deldoul
"953" = "1706",#953 Sed Rahal
"954" = "1722",#954 Selmana
"956" = "1724",#956 Oum Laadham
"957" = "1702",#957 Mouadjebar
"958" = "1730",#958 Ain el Ibel
"962" = "1710",#962 Zaccar
"963" = "1715",#963 Douis
"964" = "1723",#964 Ain Chouhada
"965" = "1736",#965 Tadmit
"967" = "1707",#967 Faïdh el Botma
"968" = "1734",#968 Amourah
)) -> dfc

```

```

deces_c_djelfa <- left_join(wc_algeria, dfc, by='CC_2')
#x = djelfa, y = dataext, by.x = "IDC", by.y = "IDC"

```

```

cdjelfa <- deces_c_djelfa %>%
  filter(NAME_1=="Djelfa") %>%
  select(NAME_1,CC_2,number_com) %>%
  mutate(number_com = log(number_com))

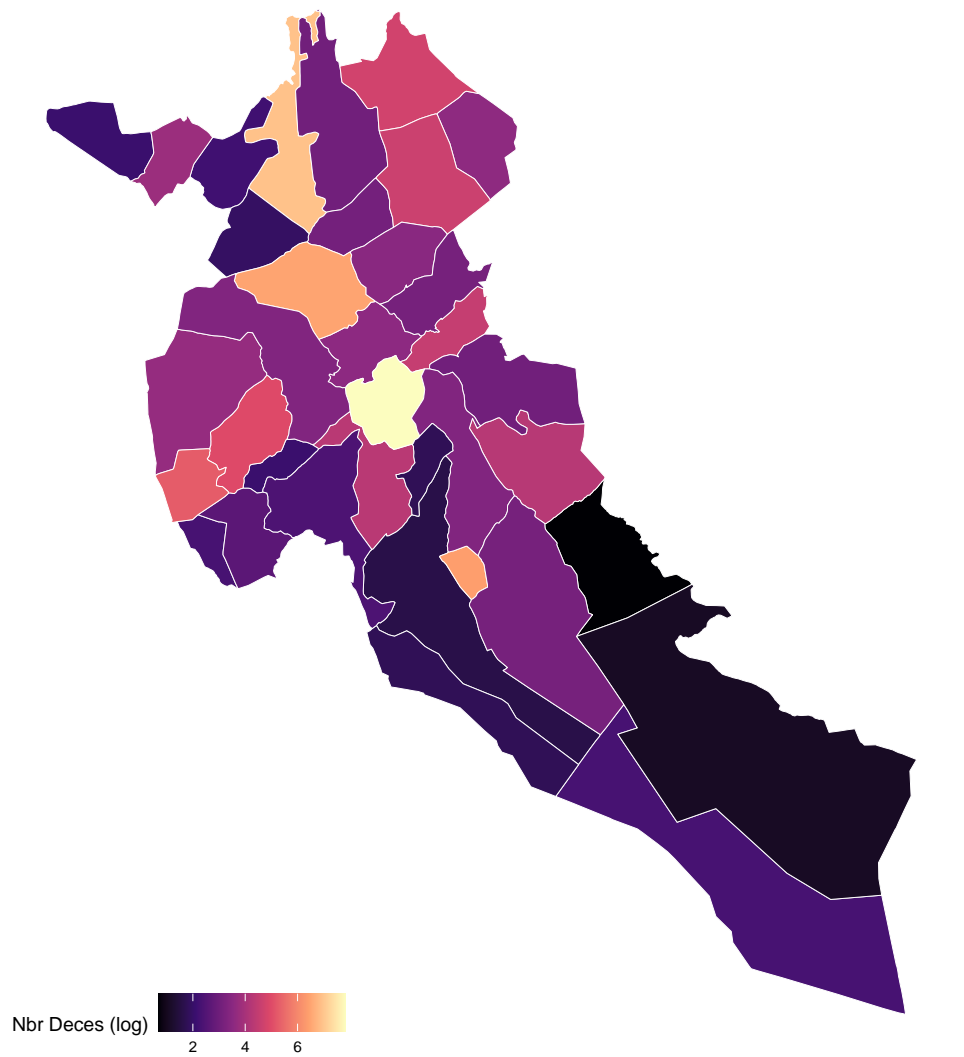
```

```

ggplot()+
  geom_sf(data=cdjelfa,aes(fill=number_com), color = "white", lwd = 0.05) +
  scale_fill_viridis_c(option = "magma", name = "Nbr Deces (log)") +
  theme_map() +
  theme(legend.direction="horizontal") +
  labs(title = "Répartition des décès hospitalier par communes",
        subtitle = paste("Communes Wilaya de djelfa du ",dt1," au ",dt2),
        caption = "Source: Dr R.TIBA \n Praticien inspecteur santé publique \n DSP Wilaya de Djelfa")+
  coord_sf(crs = "+proj=robin")

```

# Répartition des décès hospitalier par communes Communes Wilaya de djelfa du 2020-01-01 au 2021-12-31



Source: Dr R.TIBA  
Praticien inspecteur santé publique  
DSP Wilaya de Djelfa

lieux:table

```
lieux <- table(data$LD)#,useNA = "always"
sort(lieux)
```

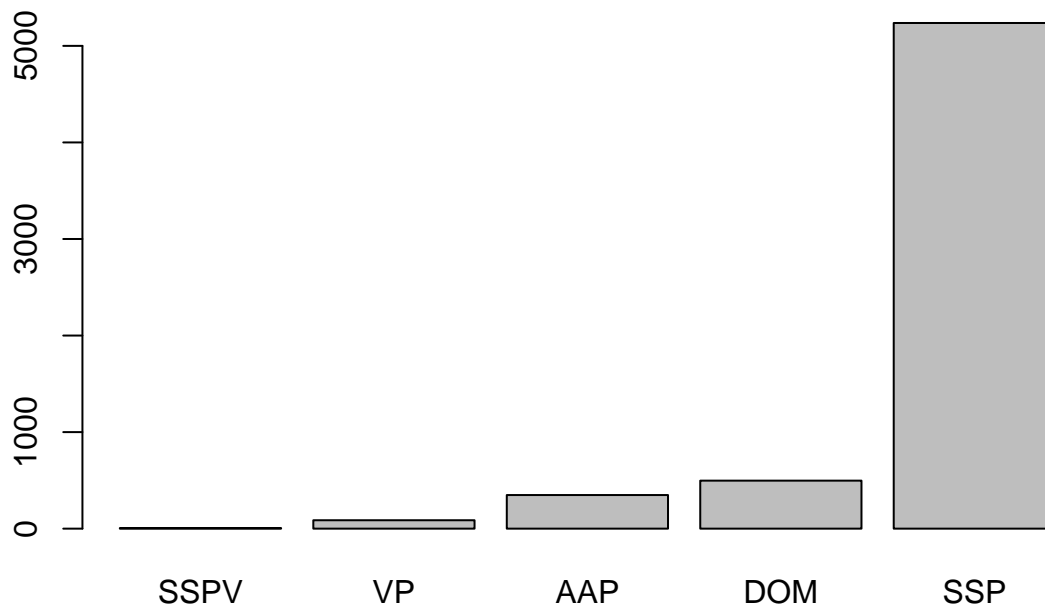
```
##
## SSPV  VP  AAP  DOM  SSP
##    6   87  348  497  5237
```

```
summary(lieux)
```

```
## Number of cases in table: 6175
## Number of factors: 1
```

lieux:graphic:r-base

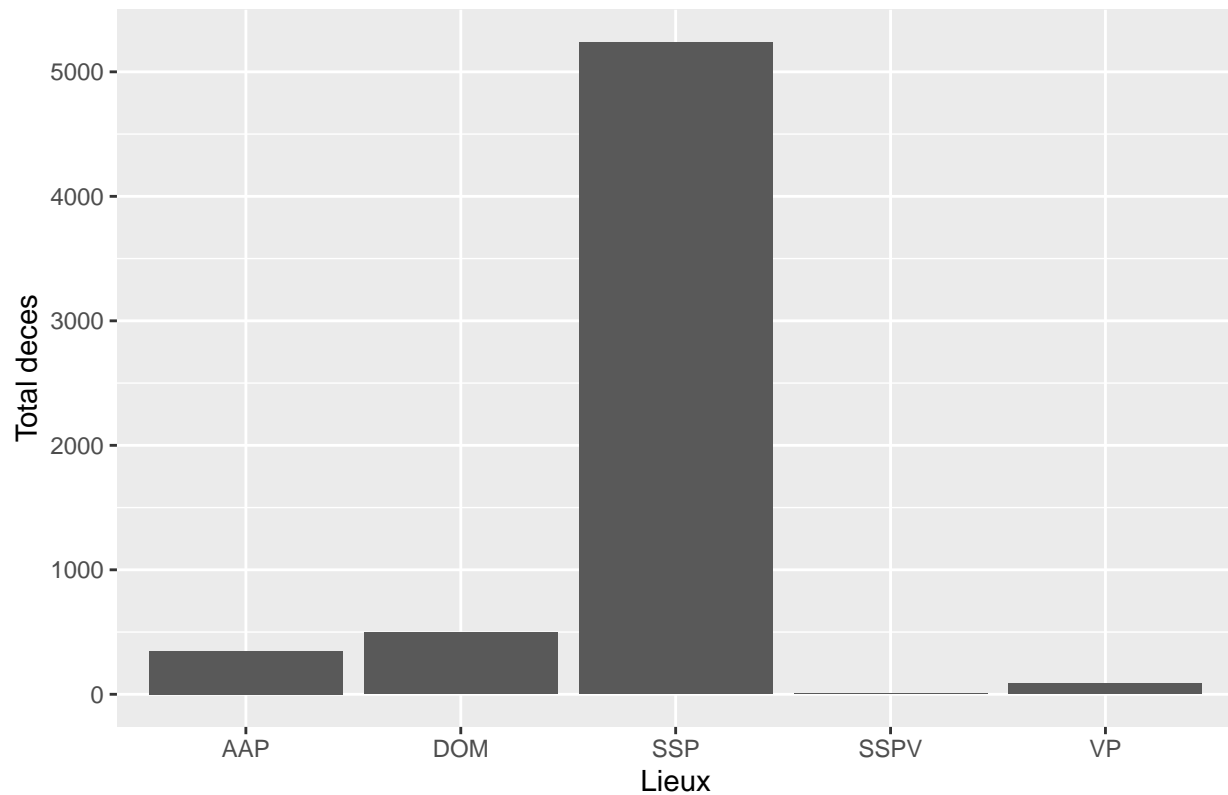
```
barplot(sort(lieux))
```



lieux:graphic:r-ggplot2

```
ggplot(data = data, mapping = aes(x = LD)) +  
  geom_bar()+  
  labs(title = "Répartition des décès par lieux",  
        x = "Lieux",  
        y = "Total deces")
```

## Répartition des décès par lieux



## structure sanitaire

### structure:table

```
structure <- table(data$STRUCTURED)#,useNA = "always"
sort(structure)
```

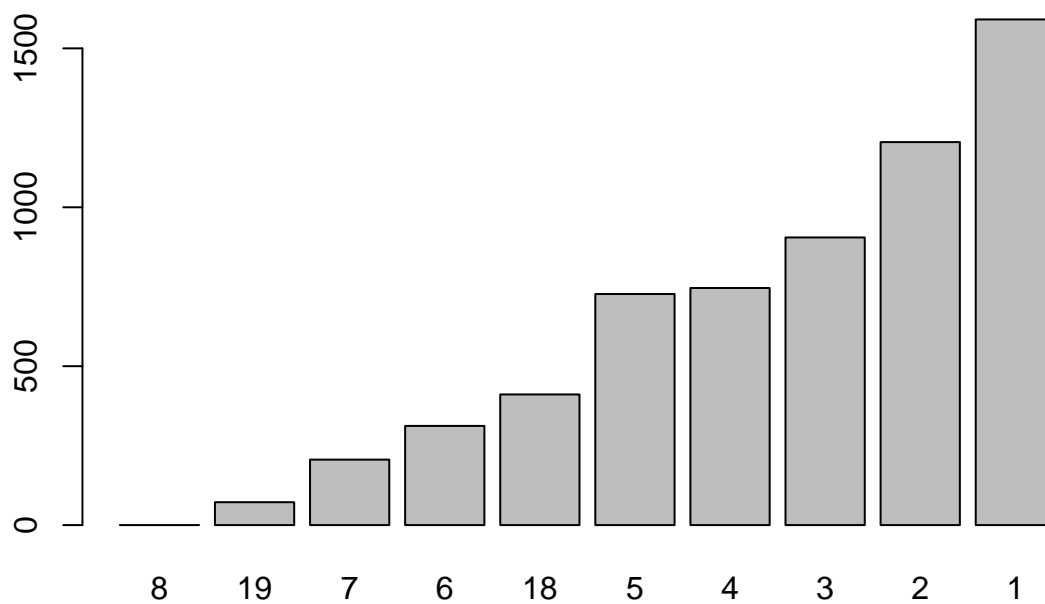
```
##
##      8   19    7    6   18    5    4    3    2    1
##      0   72  206  312  411  727  746  905 1205 1591
```

```
summary(structure)
```

```
## Number of cases in table: 6175
## Number of factors: 1
```

### structure:graphic r-base

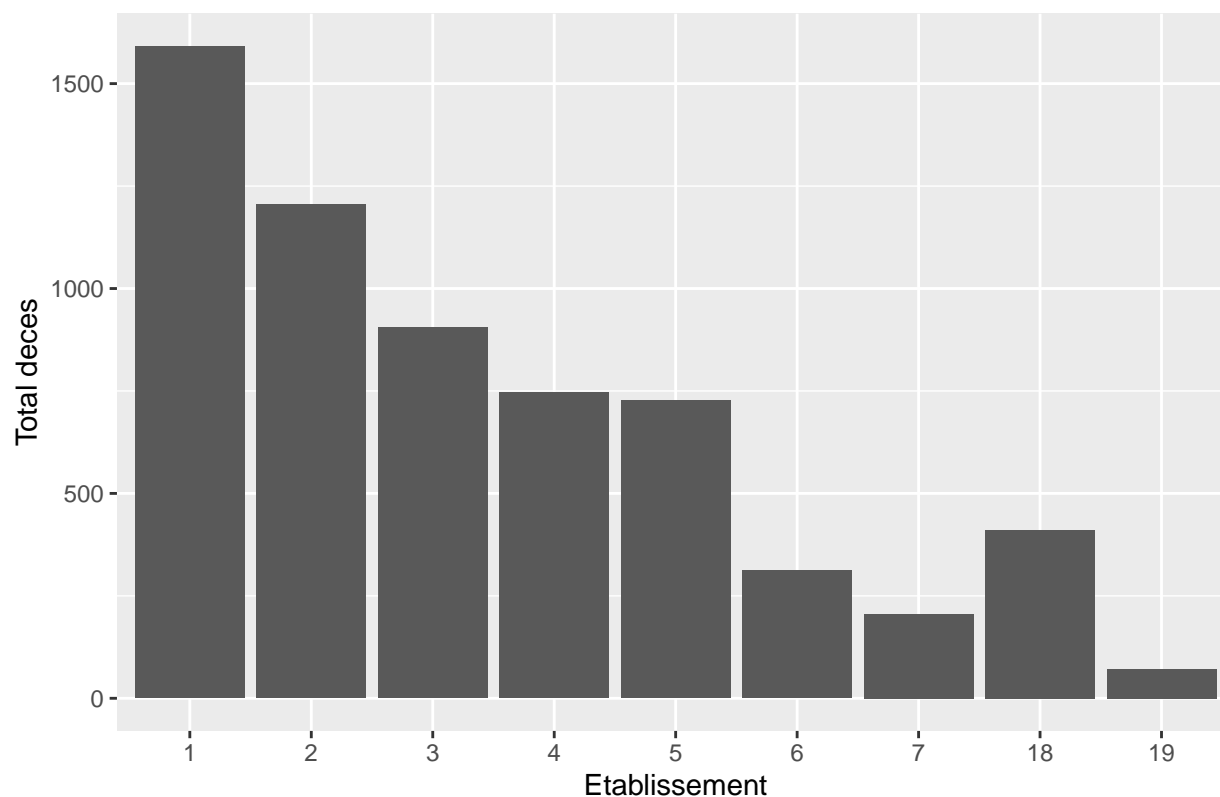
```
barplot(sort(structure))
```



structure:graphic:r-ggplot2

```
ggplot(data = data, mapping = aes(x = STRUCTURED)) +  
  geom_bar()+  
  labs(title = "Répartition des décès par établissements",  
        x = "Etablissement",  
        y = "Total deces")
```

## Répartition des décès par établissements



### service du deces

service:table

```
service <- table(data$SERVICEHOSPIT)#,useNA = "always"
sort(service)
```

```
##
##      4      14      17      3      12      19      24      9      11      1      8      25      5      2      0      15
##      0      0      0      1      1      1      1      4      5      11     15     17     40     77    128    138
##     16      7      6     18     21     10     20
##    173    424    465    468    639   1225   2342
```

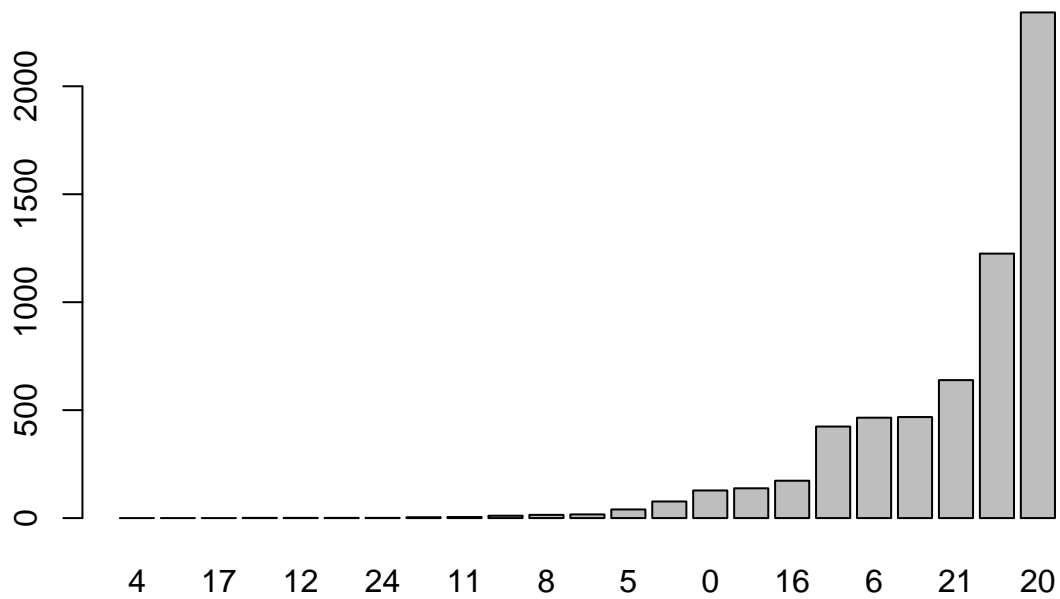
```
summary(service)
```

```
## Number of cases in table: 6175
## Number of factors: 1
```

service:graphic:r-base

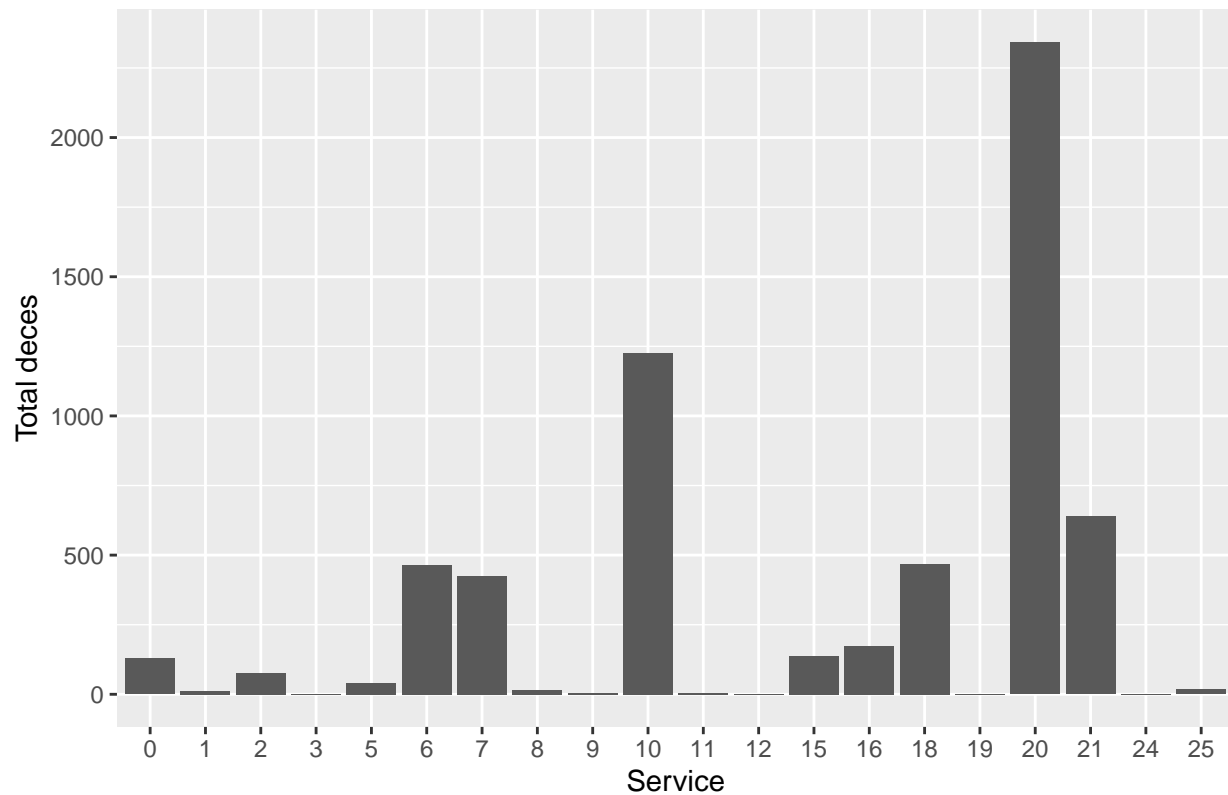
```
barplot(sort(service))
```





```
### service:graphic:r-ggplot2
ggplot(data = data, mapping = aes(x = SERVICEHOSPIT)) +
  geom_bar()+
  labs(title = "Répartition des décès par services",
        x = "Service",
        y = "Total deces")
```

## Répartition des décès par services



## age en année

### age:table

```
#clag <- c(0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100)
clag <- c(0,10,20,30,40,50,60,70,80,90,100)
data$age20cl <- cut(data$Years,clag,include.lowest = TRUE)
table(data$age20cl)
```

```
##
## [0,10] [10,20] [20,30] [30,40] [40,50] [50,60] [60,70] [70,80]
## 1724 108 174 235 343 508 800 936
## (80,90] (90,100]
## 1068 279
```

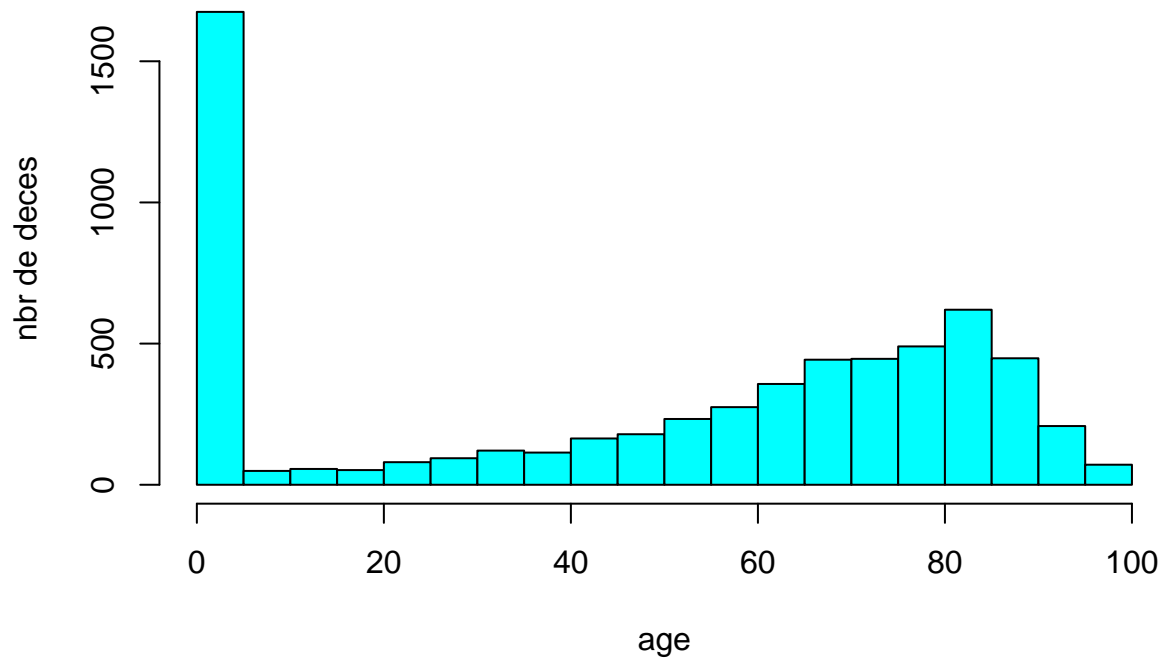
```
#age <- table(data$Years)#,useNA = "always"
#sort(age)
#summary(age)
```

### age:grafic:r-base

```
hist(data$Years,
      main = "Histogramme de l'age",
      xlab = "age",
      ylab = "nbr de deces",
      breaks = 15,
```

```
col = "cyan")
```

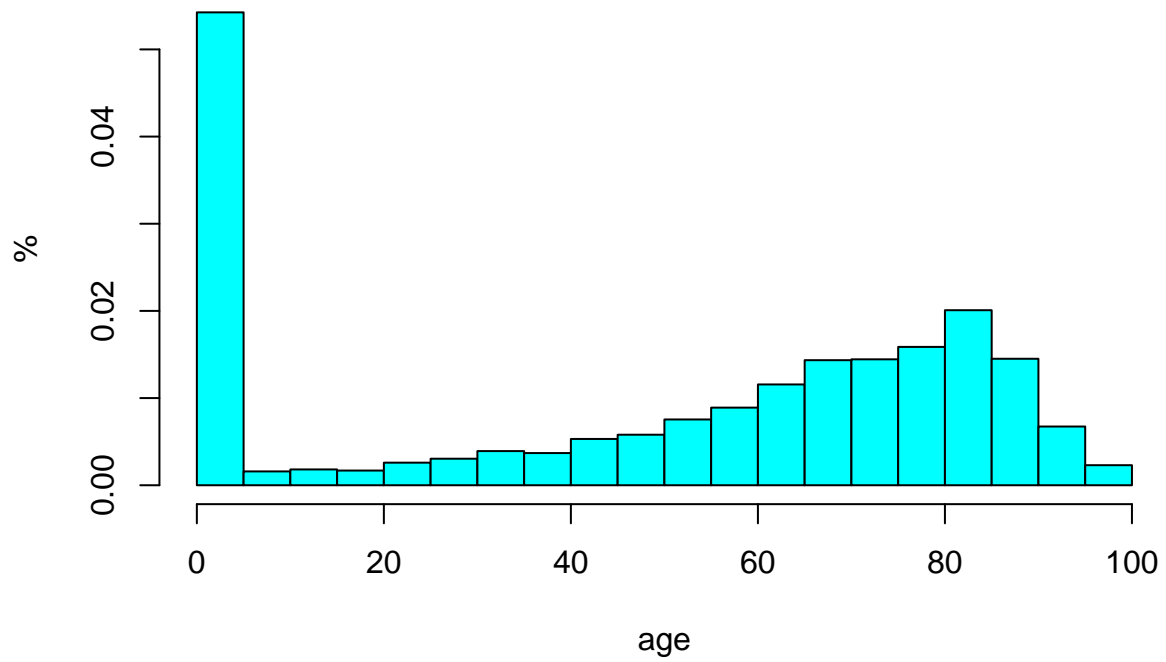
## Histogramme de l'age



age:grafic:r-base

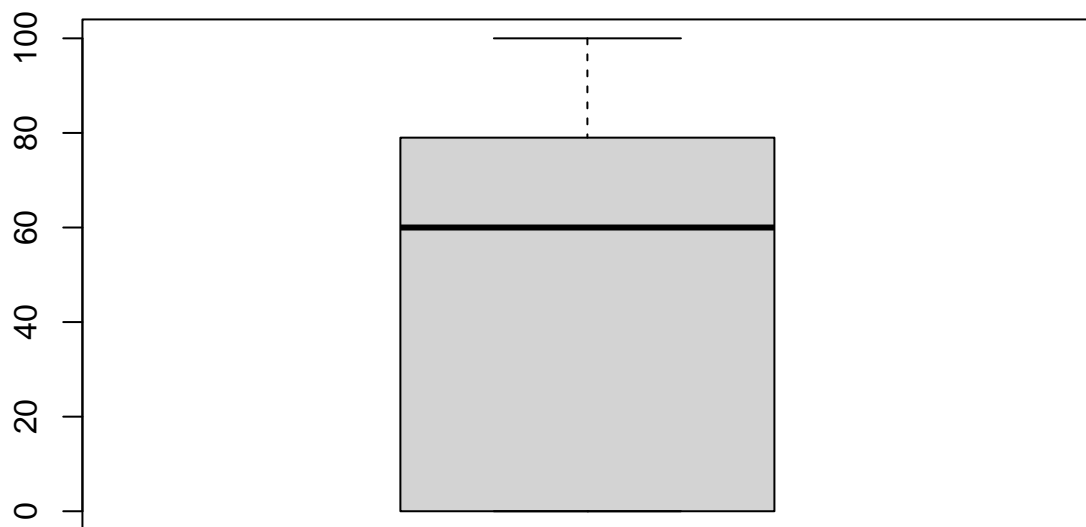
```
hist(data$Years,  
      main = "Histogramme de l'age",  
      xlab = "age",  
      ylab = "%",  
      breaks = 15,  
      col = "cyan",  
      probability = TRUE)
```

## Histogramme de l'age



age:grafic:r-base

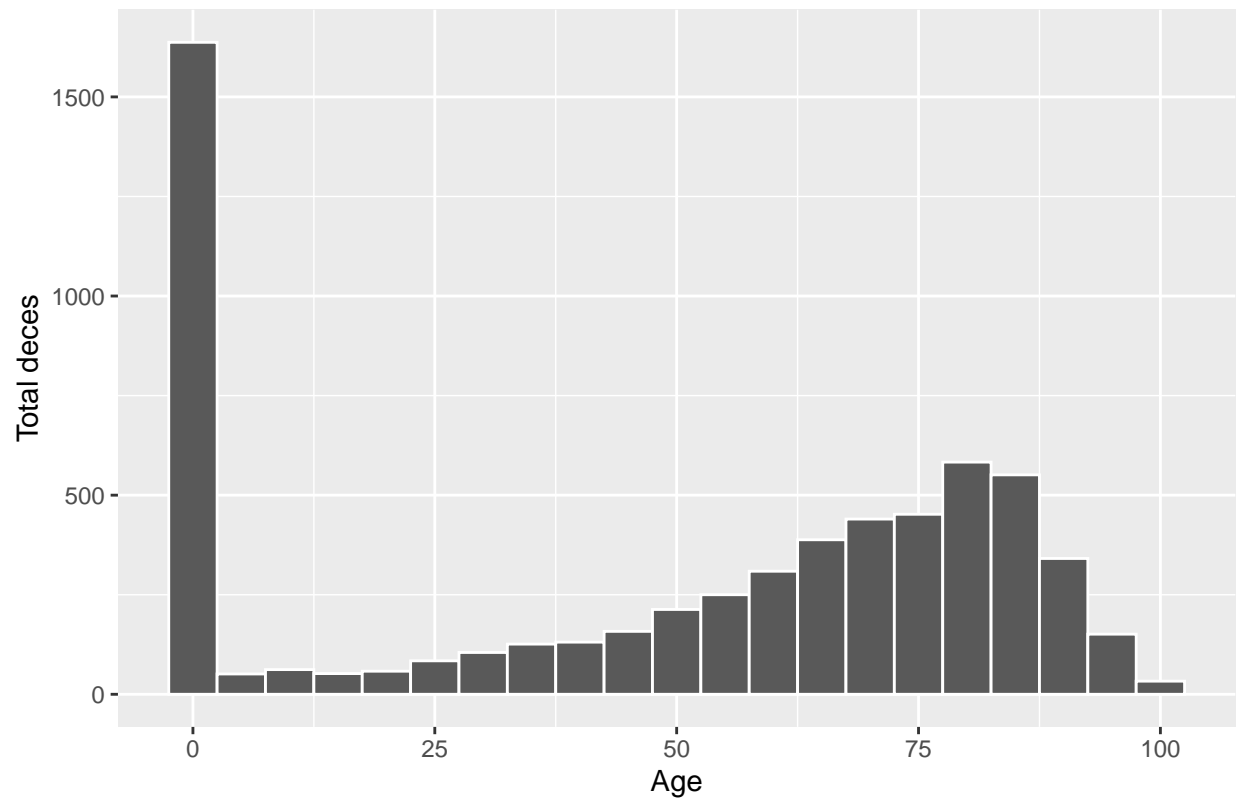
```
boxplot(data$Years)
```



age:grafic:t-ggplot2

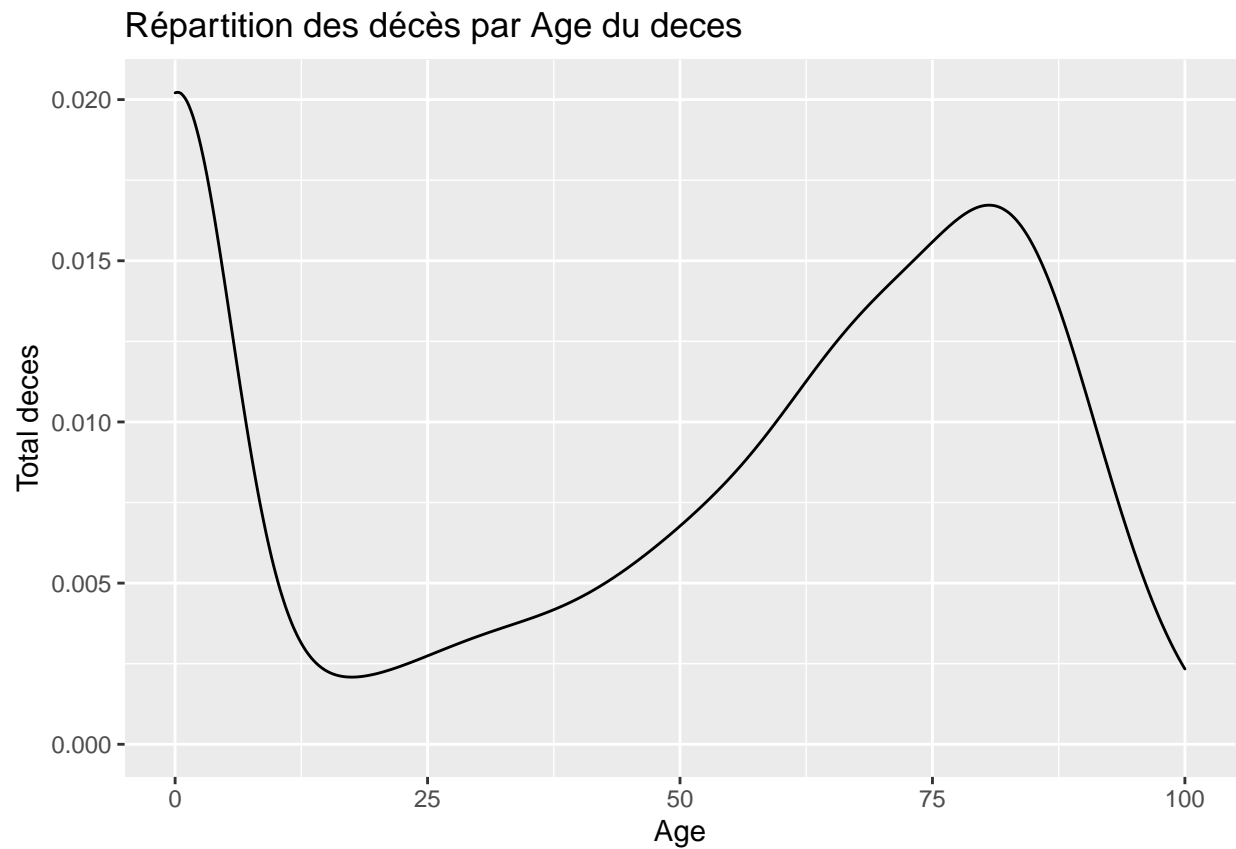
```
ggplot(data = data, mapping = aes(x = Years)) + #, fill = data$SEX  
  geom_histogram(binwidth = 5, color = "white")+  
  labs(title = "Répartition des décès par tranches d'ages",  
        x = "Age",  
        y = "Total deces")
```

Répartition des décès par tranches d'âges



```
### age:grafic:t-ggplot2
```

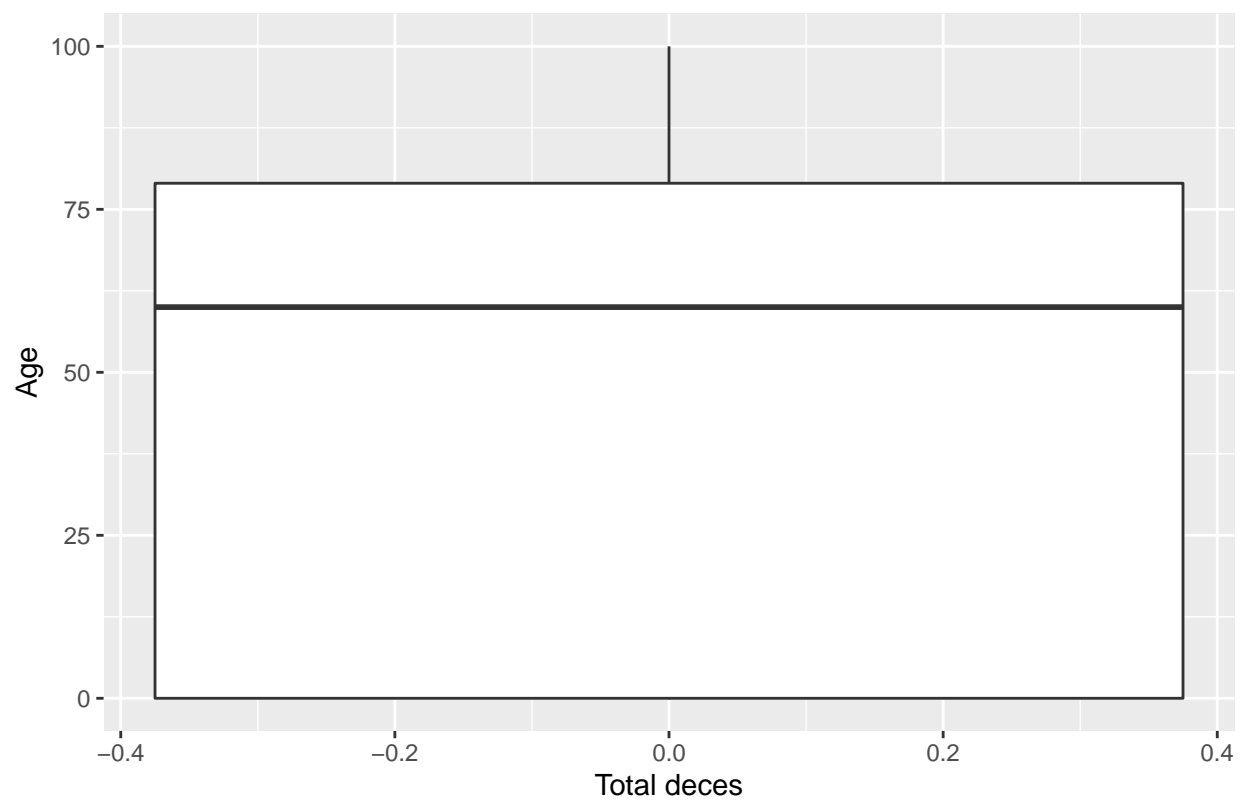
```
ggplot(data = data, mapping = aes(x = Years)) + #, fill = data$SEX  
  geom_density()+  
  labs(title = "Répartition des décès par Age du deces",  
        x = "Age",  
        y = "Total deces")
```



age:grafic:t-ggplot2

```
ggplot(data = data, mapping = aes(y = Years)) + #, fill = data$SEX
  geom_boxplot()+
  labs(title = "Répartition des décès par Age du deces",
        x = "Total deces",
        y = "Age")
```

## Répartition des décès par Age du deces



## sexe

sexe:table

```
sexe <- table(data$SEX)#,useNA = "always"  
sort(sexe)
```

##

## F M

## 2566 3609

```
summary(sexe)
```

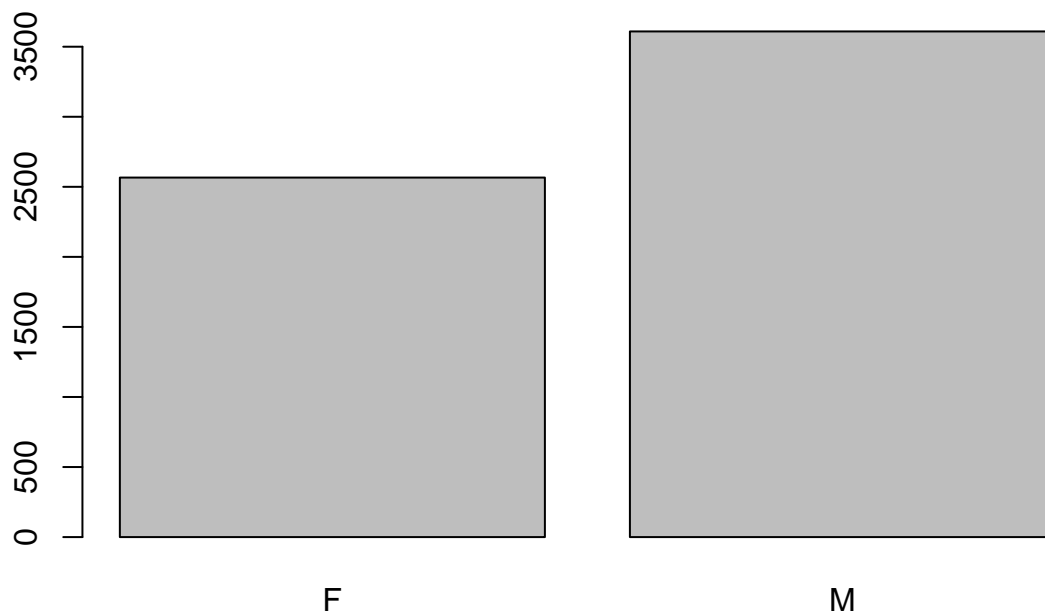
## Number of cases in table: 6175

## Number of factors: 1

sexe:graphic r-base

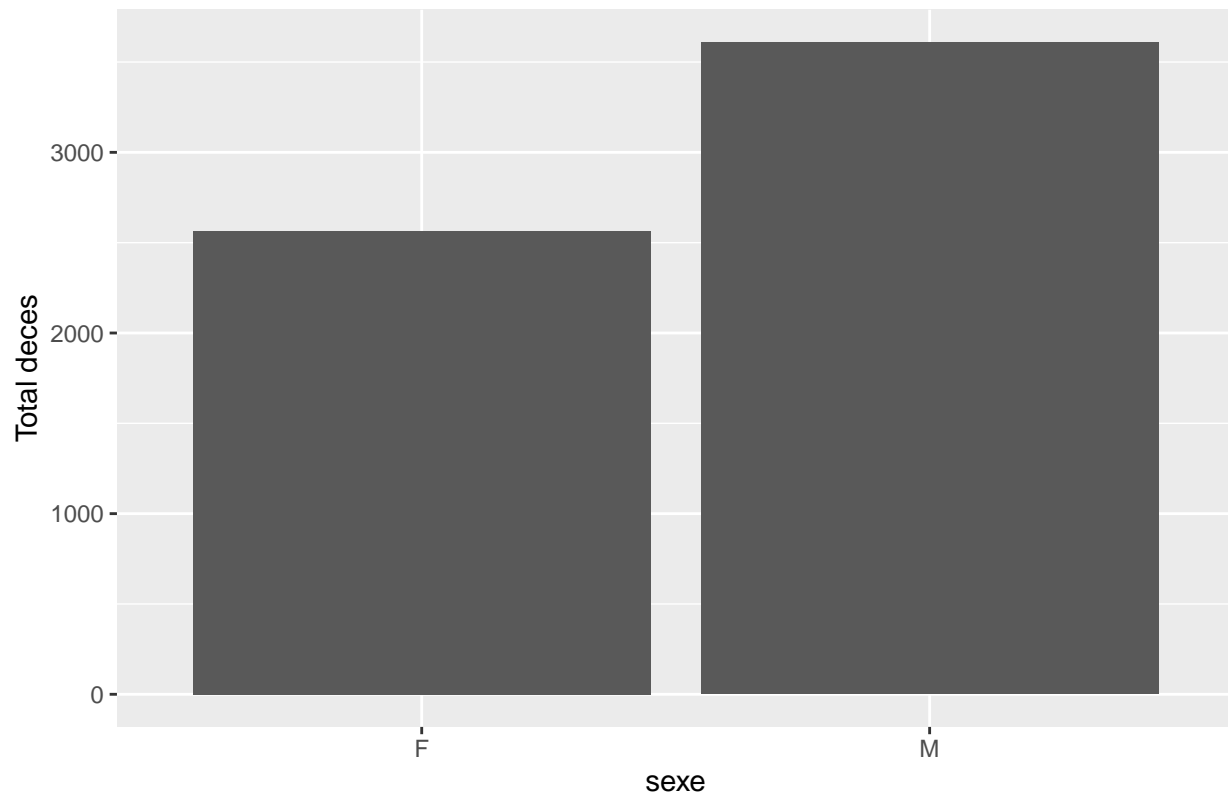
```
barplot(sort(sexe))
```





```
### sexe:graphic r-ggplot2
ggplot(data = data, mapping = aes(x = SEX)) +
  geom_bar()+
  labs(title = "Répartition des décès par sexe",
        x = "sexe",
        y = "Total deces")
```

## Répartition des décès par sexe



## Profession

profession:table

```
Profession <- table(data$Profession)#,useNA = "always"
sort(Profession)
```

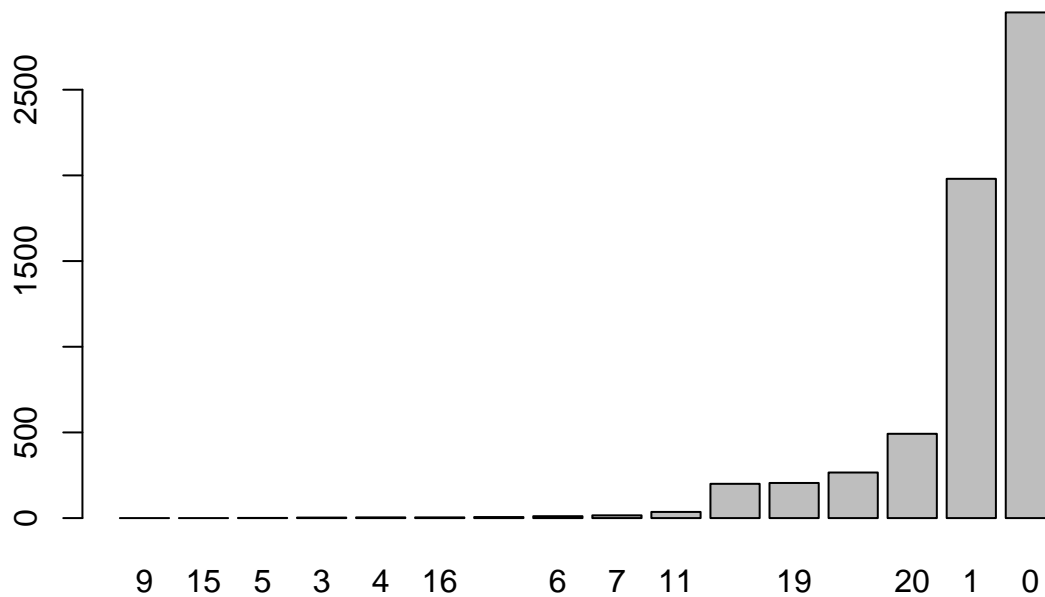
```
##
##      9   15    5    3    4   16   14    6    7   11   12   19   13   20    1    0
##      0    0    1    3    4    4    6   11   16   36   200  205  266  492 1980 2951
```

```
summary(Profession)
```

```
## Number of cases in table: 6175
## Number of factors: 1
```

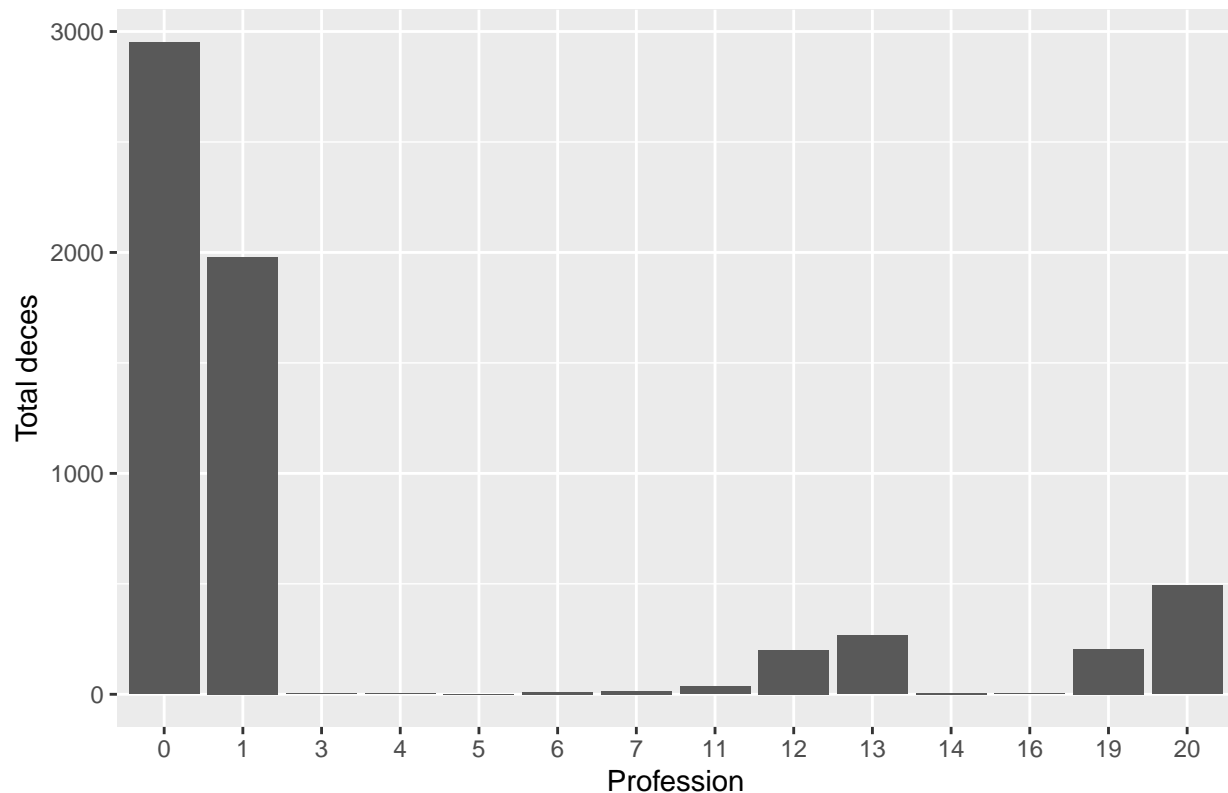
profession:graphic r-base

```
barplot(sort(Profession))
```



```
### profession:graphic r-ggplot2
ggplot(data = data, mapping = aes(x = Profession)) +
  geom_bar()+
  labs(title = "Répartition des décès par profession",
        x = "Profession",
        y = "Total deces")
```

## Répartition des décès par profession



## repartition des deces par chapitre CIM10

```
table(data$CODECIM0)
```

```
##
##      0      1      2      3      4      5      6      8      9     10     11     12     13     14     15     16
## 1137   270   241    38   104     3    75     2   902  1398   120     3     8   124    16   825
##      17     18     19     20     21
##   237   525   102    43     2
```

## repartition des deces par titre CIM10

```
#table(data$CODECIM)
```

## Analyse bivariée

Faire une analyse bivariée, c'est étudier la relation entre deux variables : sont-elles liées ? les valeurs de l'une influencent-elles les valeurs de l'autre ? ou sont-elles au contraire indépendantes ?

À noter qu'on va parler ici d'influence ou de lien, mais pas de relation de cause à effet. Les outils présentés permettent de visualiser ou de déterminer une relation, mais la mise en évidence de liens de causalité proprement dit est nettement plus complexe : il faut en effet vérifier que c'est bien telle variable qui influence telle autre et pas l'inverse, qu'il n'y a pas de "variable cachée", etc.

Là encore, le type d'analyse ou de visualisation est déterminé par la nature qualitative ou quantitative des deux variables.

## Croisement de deux variables qualitatives

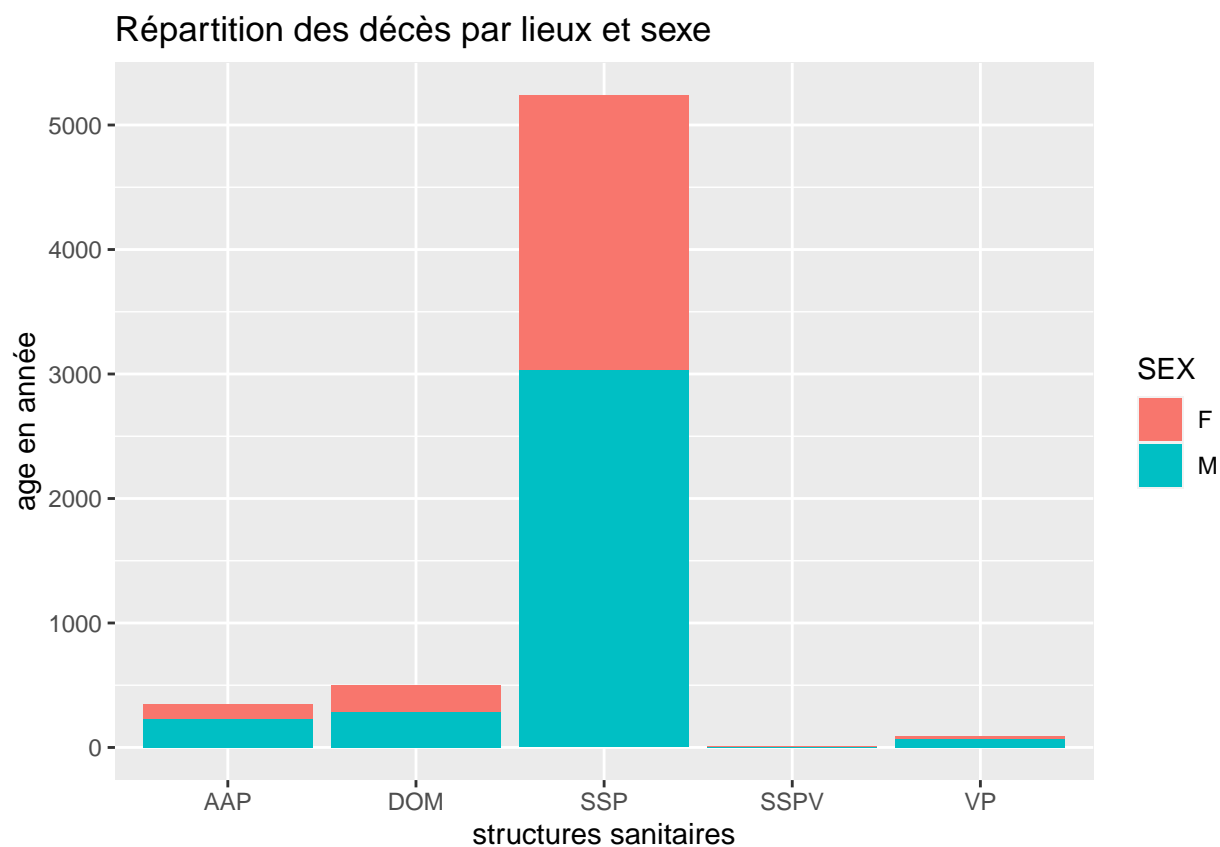
Quand on veut croiser deux variables qualitatives, on fait un tableau croisé. Comme pour un tri à plat ceci s'obtient avec la fonction `table` de R, mais à laquelle on passe cette fois deux variables en argument.

### lieux du deces et le sexe

```
table(data$LD, data$SEX)
```

```
##  
##           F      M  
## AAP    122   226  
## DOM    214   283  
## SSP   2209  3028  
## SSPV     4     2  
## VP      17    70
```

```
ggplot(data = data) +  
  geom_bar(aes(x = LD, fill = SEX)) +  
  labs(title = "Répartition des décès par lieux et sexe",  
        x = "structures sanitaires",  
        y = "age en année")
```

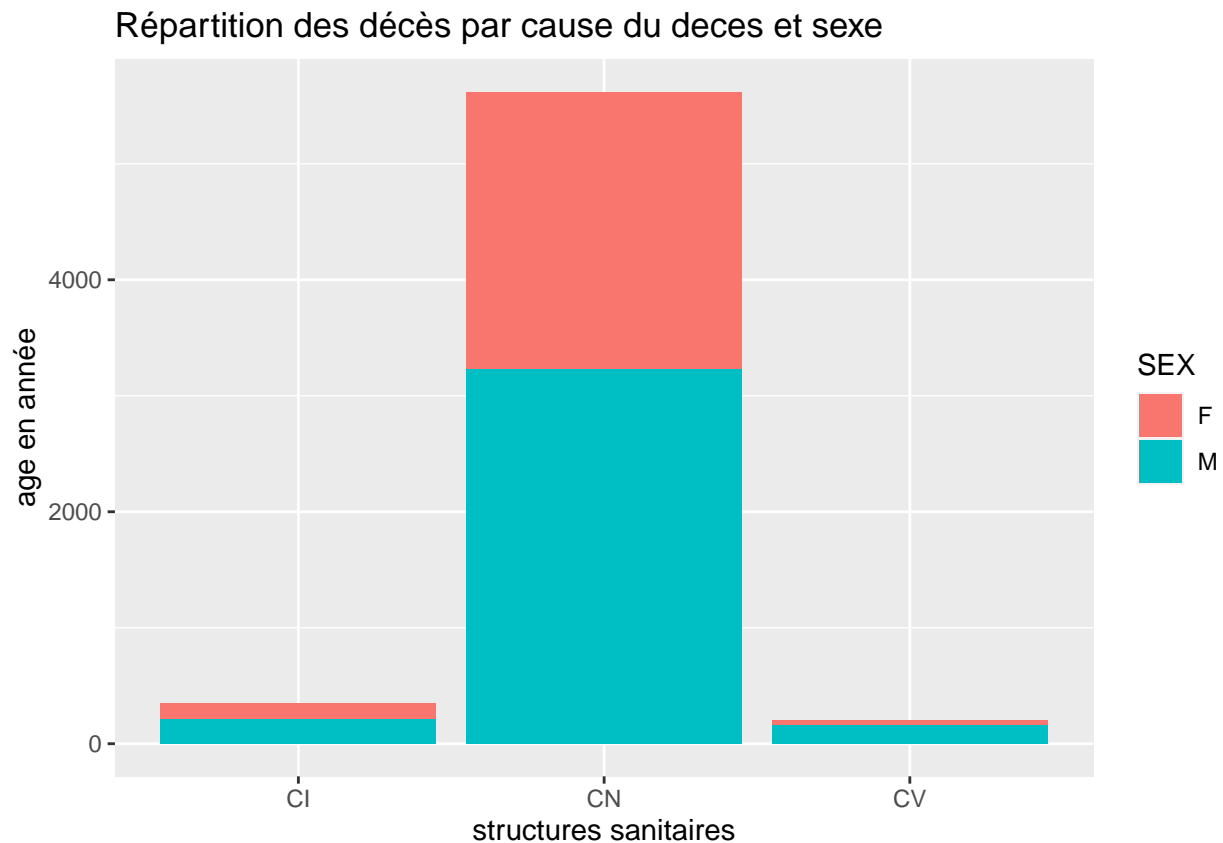


### causes du deces et le sexe

```
table(data$CD, data$SEX)
```

```
##
##           F      M
##  CI   137   214
##  CN  2388  3233
##  CV    41   162

ggplot(data = data) +
  geom_bar(aes(x = CD, fill = SEX))+
  labs(title = "Répartition des décès par cause du deces et sexe",
        x = "structures sanitaires",
        y = "age en année")
```



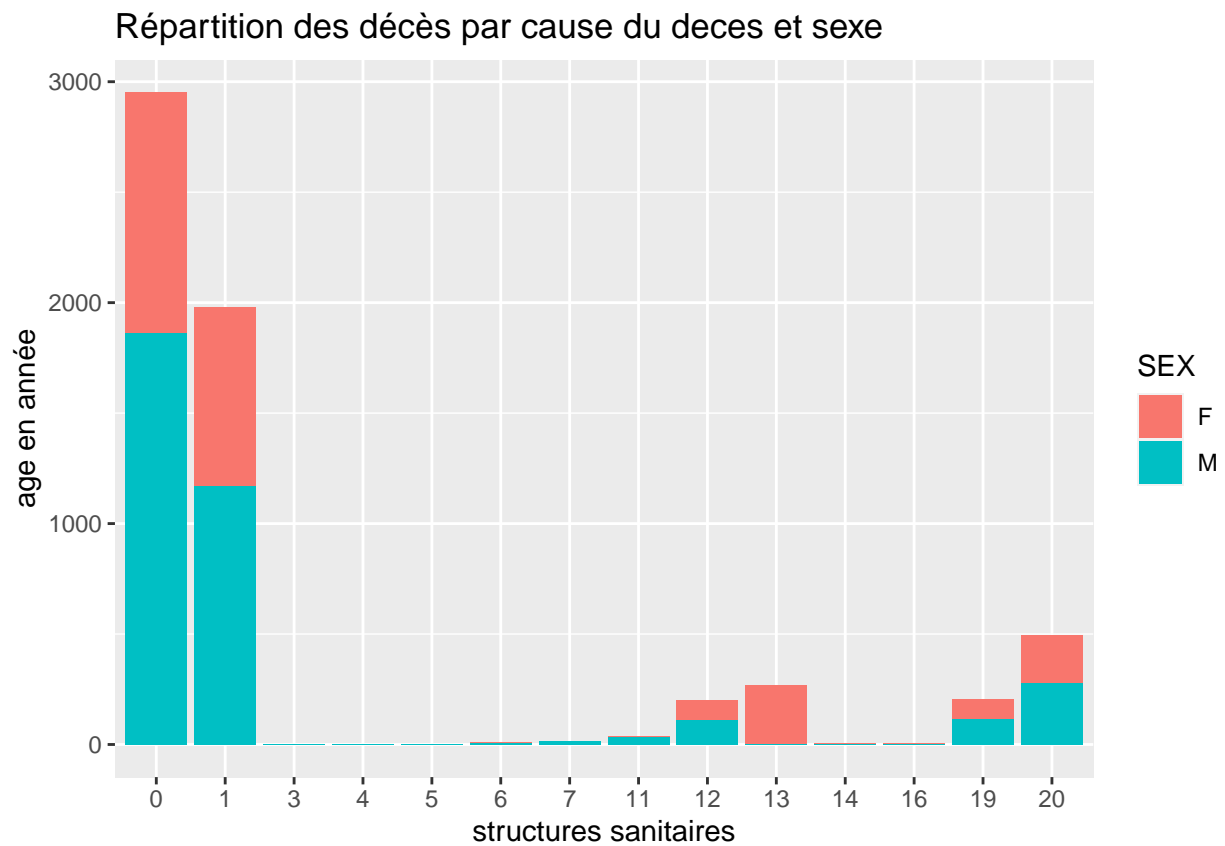
la profession du deces et le sexe

```
table(data$Profession, data$SEX)
```

```
##
##           F      M
##  0  1087  1864
##  1   812  1168
##  3     0     3
##  4     0     4
##  5     0     1
##  6     4     7
##  7     0    16
##  9     0     0
## 11     2    34
```

```
## 12 89 111
## 13 263 3
## 14 2 4
## 15 0 0
## 16 2 2
## 19 91 114
## 20 214 278
```

```
ggplot(data = data) +
  geom_bar(aes(x = Profession, fill = SEX))+
  labs(title = "Répartition des décès par cause du deces et sexe",
        x = "structures sanitaires",
        y = "age en année")
```



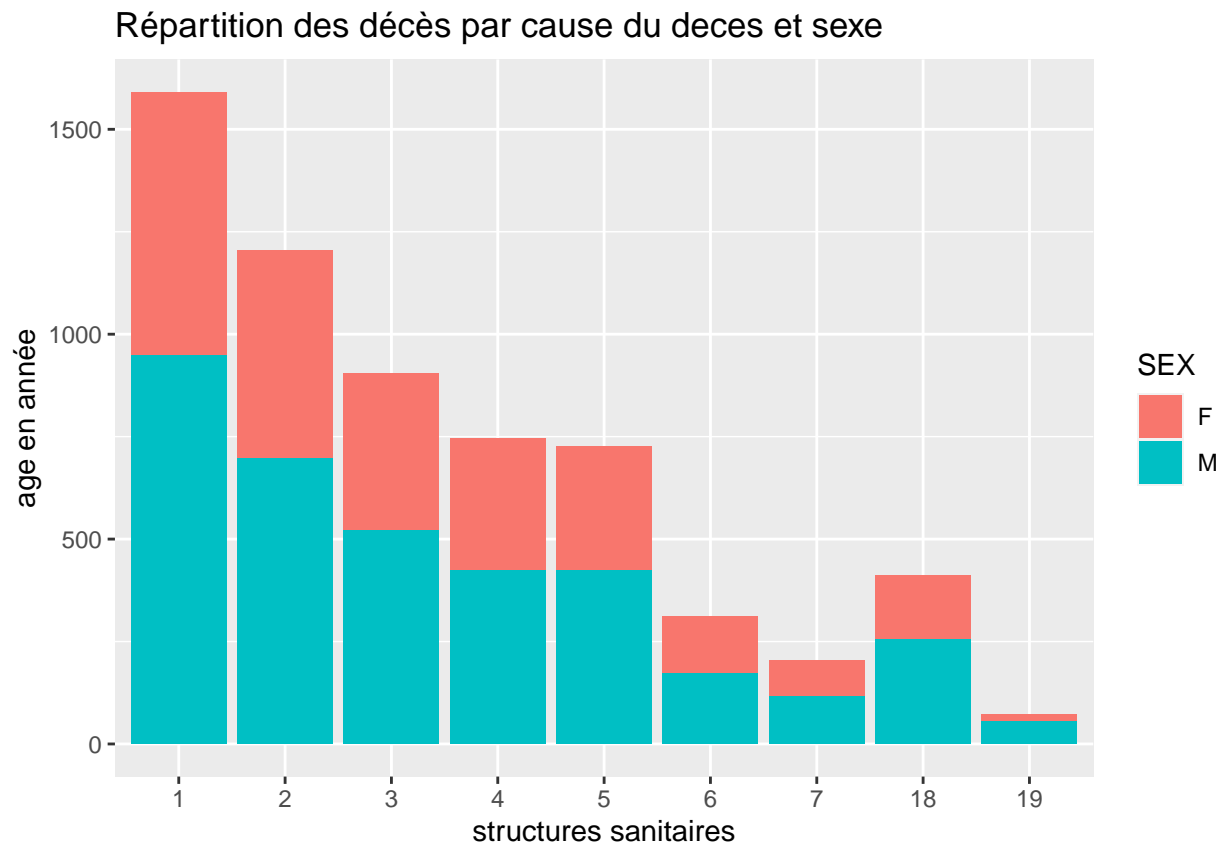
la structure sanitaire du deces et le sexe

```
table(data$STRUCTURED, data$SEX)
```

```
##
##      F      M
## 1 644 947
## 2 508 697
## 3 385 520
## 4 322 424
## 5 303 424
## 6 139 173
## 7  90 116
```

```
##      8      0      0
##     18    157   254
##     19     18    54
```

```
ggplot(data = data) +
  geom_bar(aes(x = STRUCTURED, fill = SEX))+
  labs(title = "Répartition des décès par cause du deces et sexe",
       x = "structures sanitaires",
       y = "age en année")
```



### le service du deces et le sexe

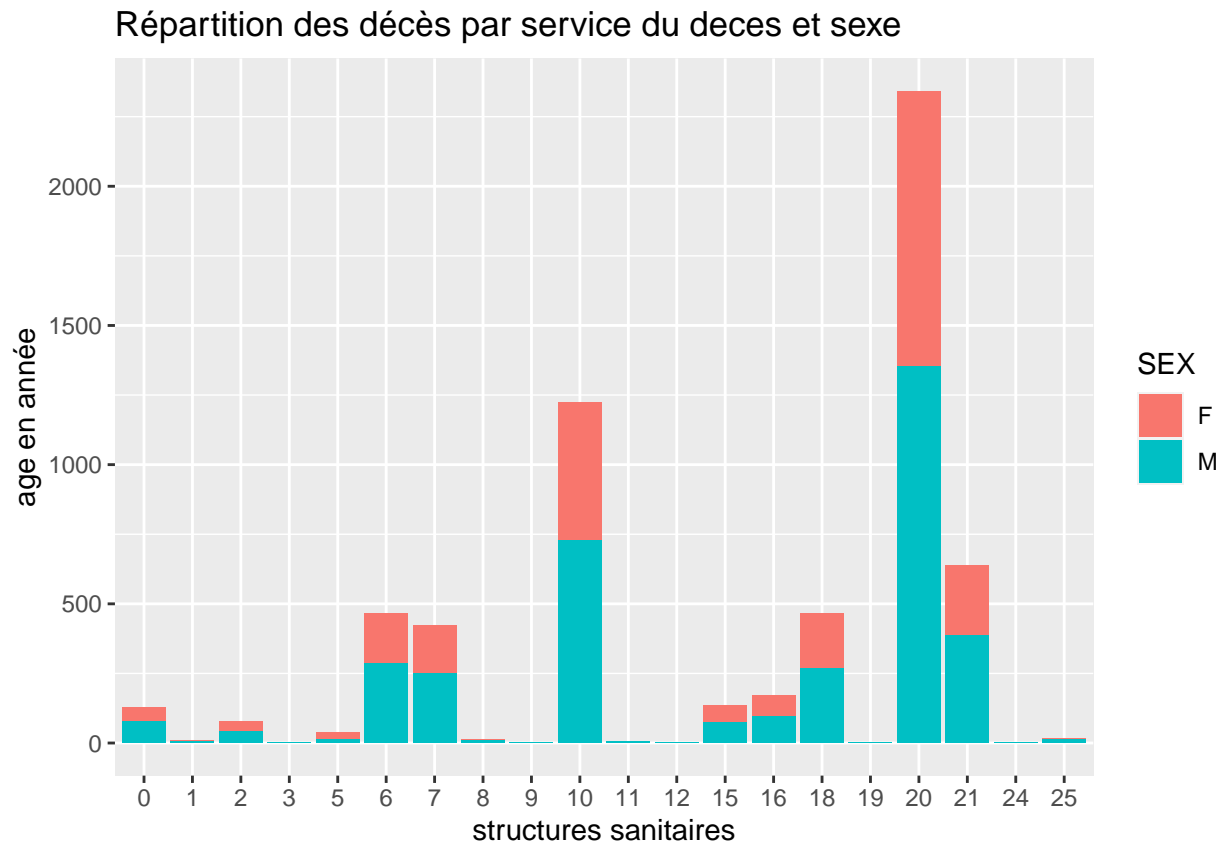
```
table(data$SERVICEHOSPIT, data$SEX)
```

```
##
##      F      M
##     0     49    79
##     1      5      6
##     2     34    43
##     3      0      1
##     4      0      0
##     5     28    12
##     6    178   287
##     7    176   248
##     8      6      9
##     9      3      1
##    10    499   726
##    11      1      4
```



```
## 12 0 1
## 14 0 0
## 15 64 74
## 16 78 95
## 17 0 0
## 18 201 267
## 19 0 1
## 20 988 1354
## 21 252 387
## 24 0 1
## 25 4 13
```

```
ggplot(data = data) +
  geom_bar(aes(x = SERVICEHOSPIT, fill = SEX))+
  labs(title = "Répartition des décès par service du deces et sexe",
        x = "structures sanitaires",
        y = "age en année") #+theme_bw()
```



repartition des deces par CIM10 et sexe

```
table(data$CODECIM0, data$SEX)
```

```
##
##      F      M
## 0 468 669
## 1 120 150
```

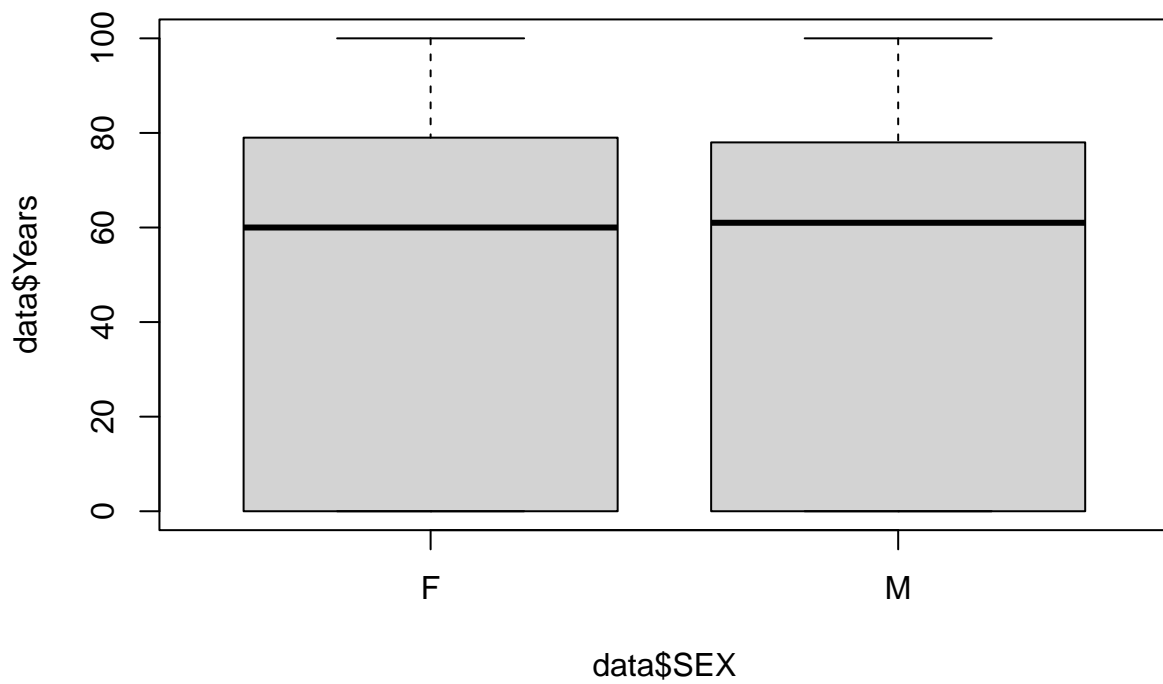
```
## 2 116 125
## 3 16 22
## 4 44 60
## 5 2 1
## 6 29 46
## 8 0 2
## 9 393 509
## 10 556 842
## 11 59 61
## 12 0 3
## 13 4 4
## 14 50 74
## 15 11 5
## 16 329 496
## 17 115 122
## 18 215 310
## 19 31 71
## 20 7 36
## 21 1 1
```

```
#table(data$CODECIM, data$SEX)
```

## Croisement d'une variable quantitative et d'une variable qualitative

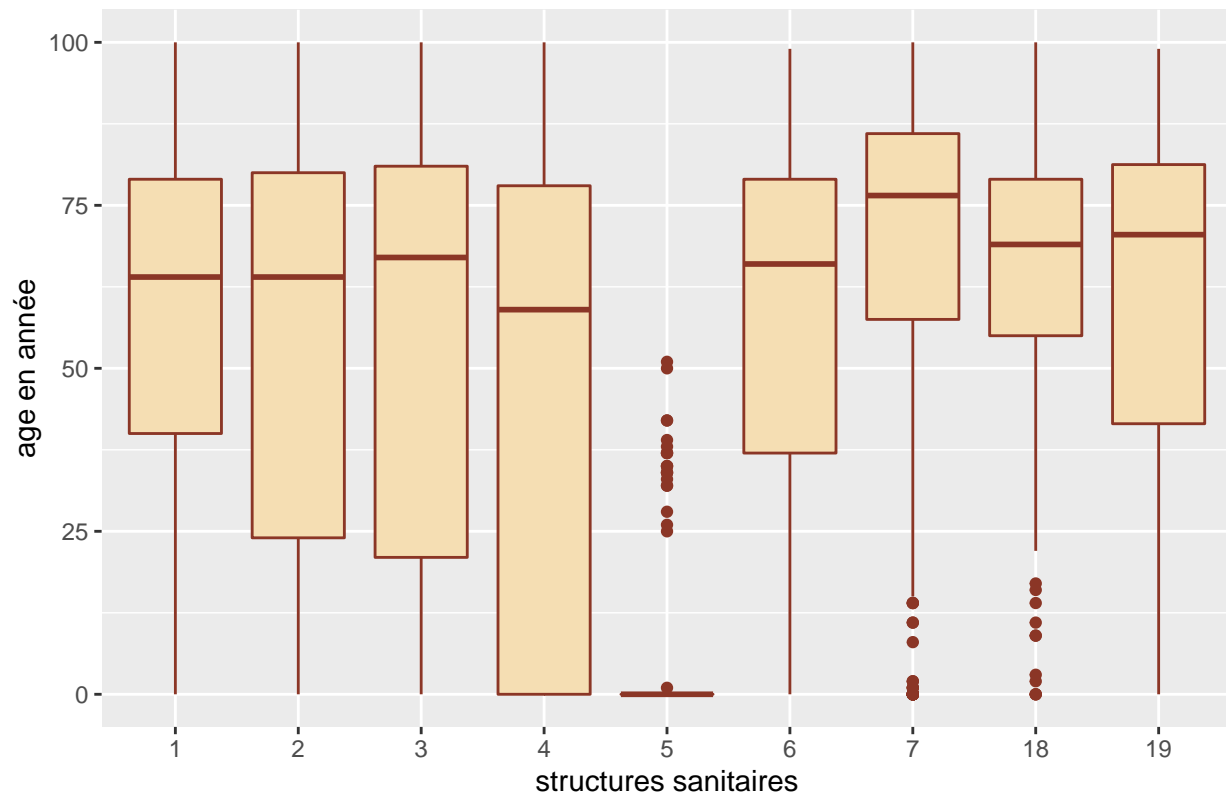
age et sexe

```
boxplot(data$Years ~ data$SEX)
```



```
ggplot(data = data,  
       mapping = aes(x =STRUCTURED,y = Years)) +  
  geom_boxplot(fill = "wheat", color = "tomato4")+  
  labs(title = "Répartition des décès par age et structure sanitaire",  
       x = "structures sanitaires",  
       y = "age en année")
```

## Répartition des décès par âge et structure sanitaire



## Croisement de deux variables quantitatives

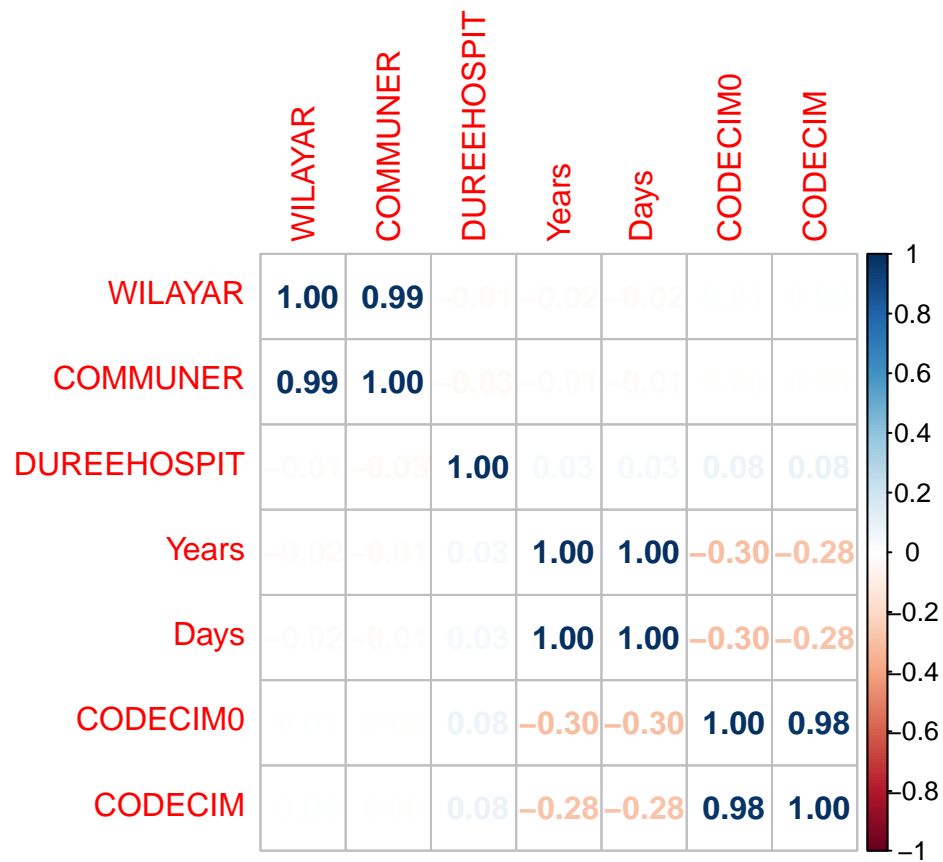
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

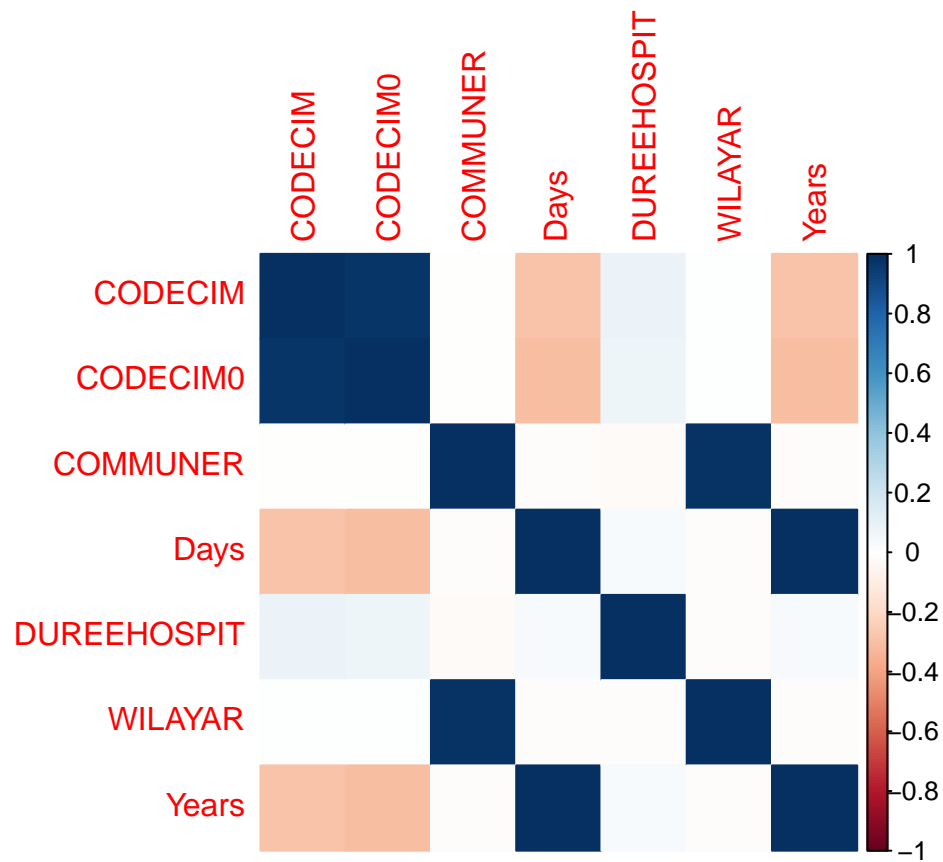
```
#str(data)
```

```
data_n <- data %>%
  select(WILAYAR, COMMUNER, DUREEHOSPIT, Years, Days, CODECIMO, CODECIM)
```

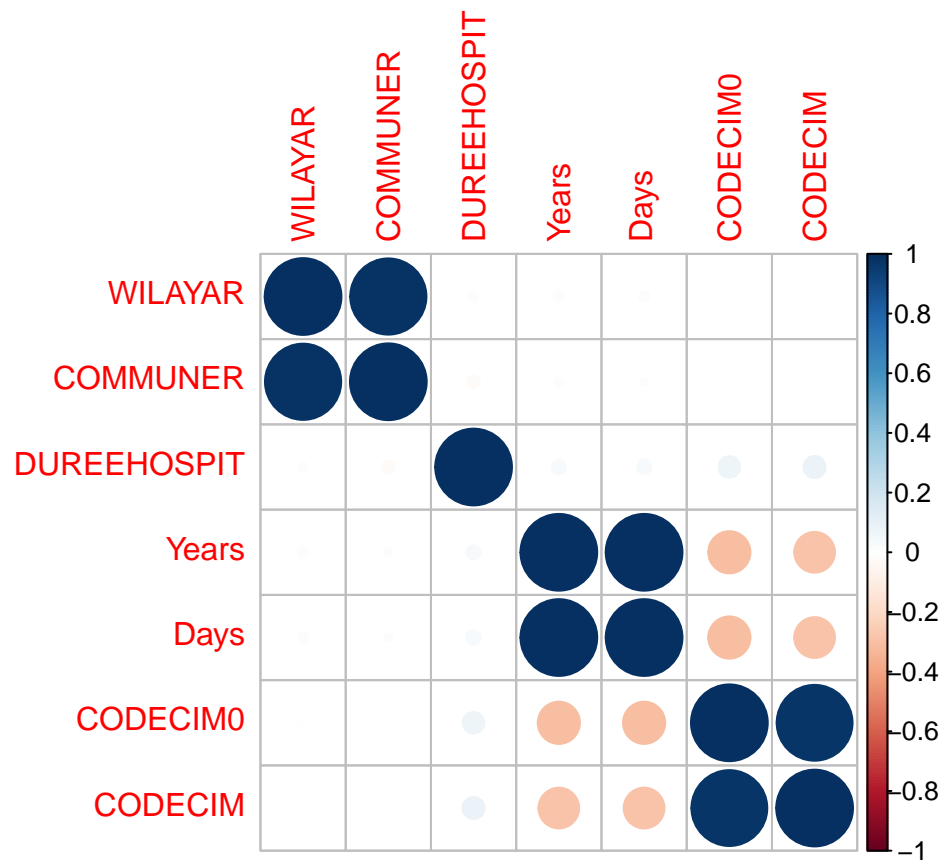
```
M = cor(data_n)
corrplot(M, method = 'number') # colorful number
```



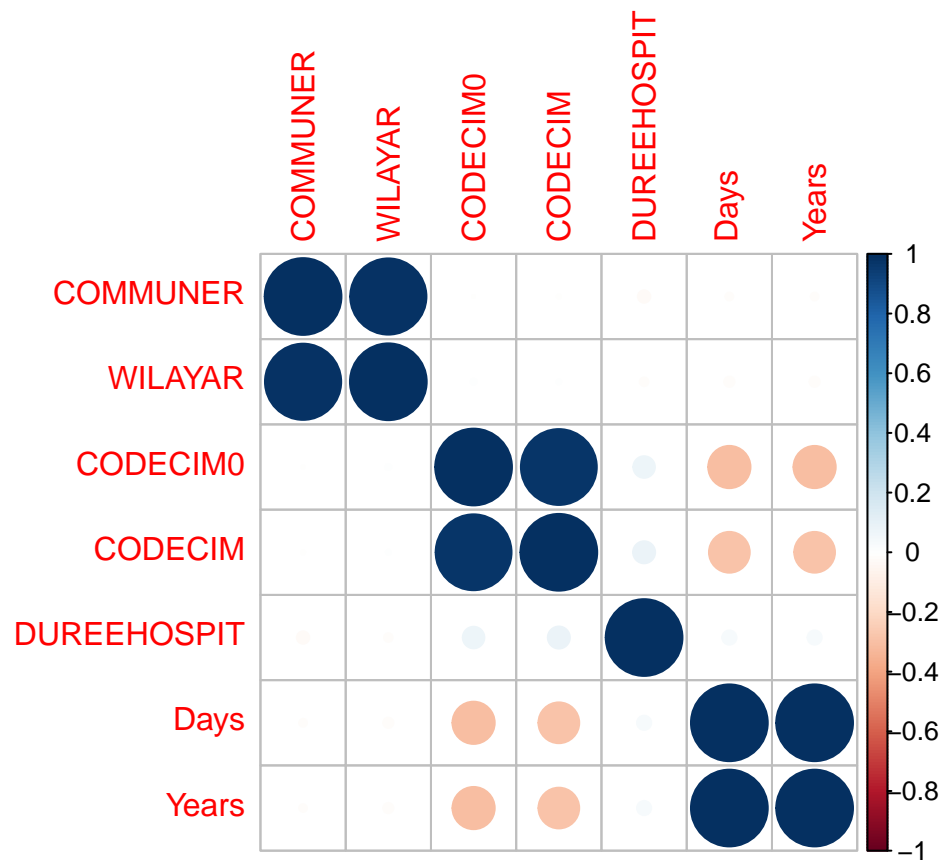
```
corrplot(M, method = 'color', order = 'alphabet')
```



```
corrplot(M) # by default, method = 'circle'
```

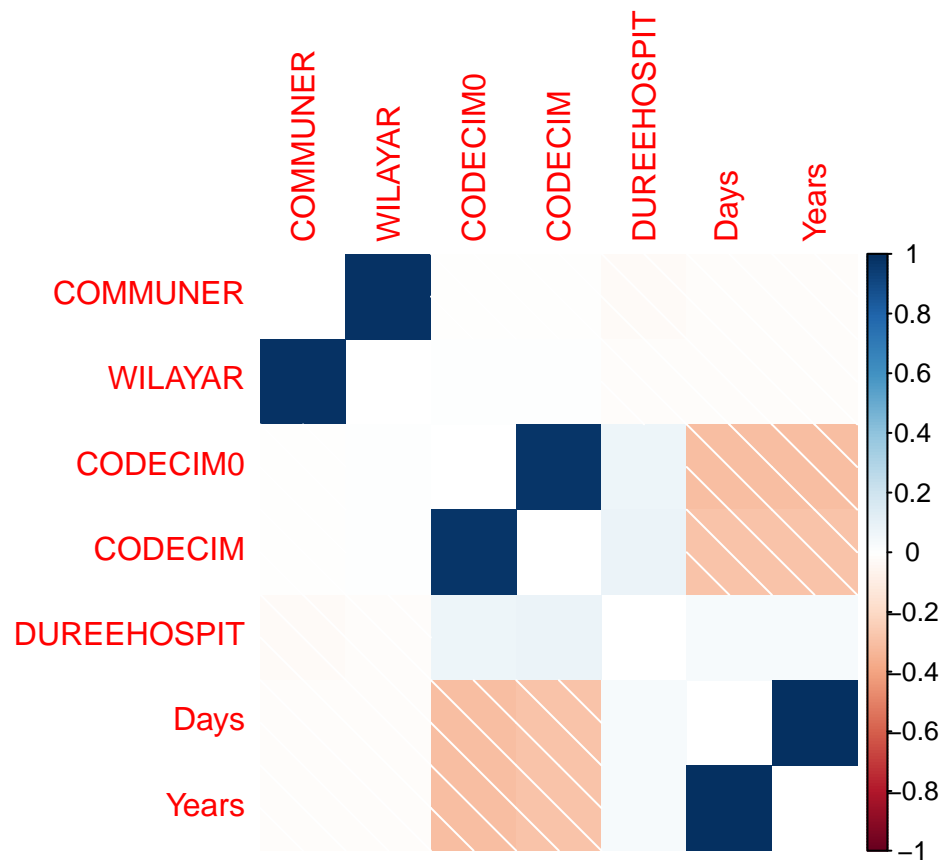


```
corrplot(M, order = 'AOE') # after 'AOE' reorder
```

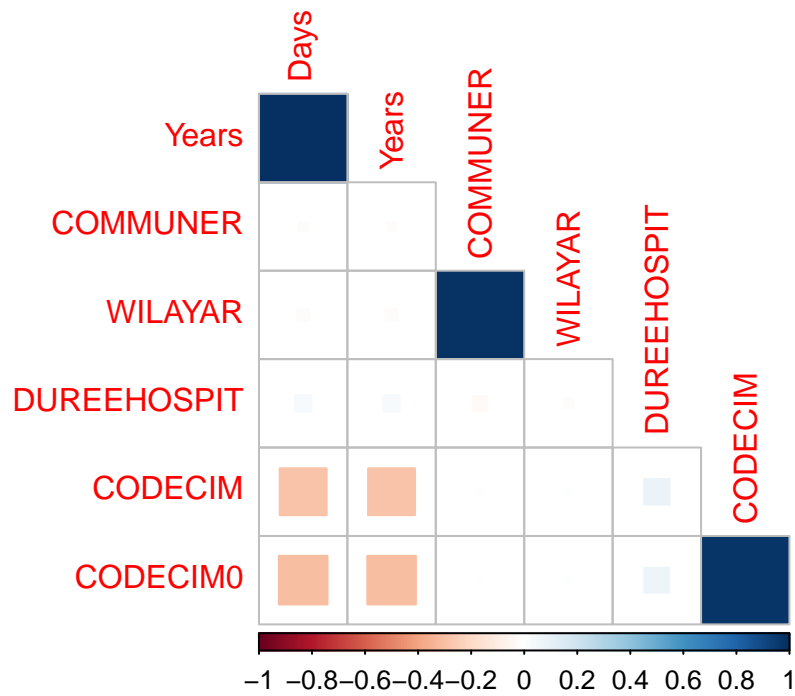


```
corrplot(M, method = 'shade', order = 'AOE', diag = FALSE)
```

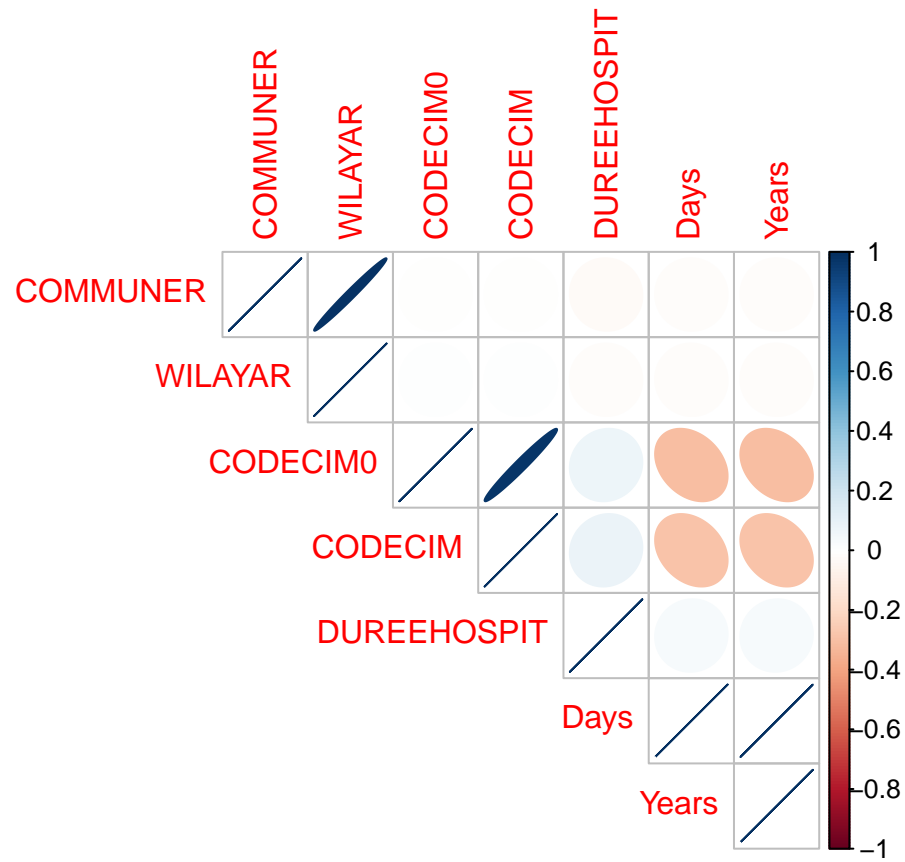




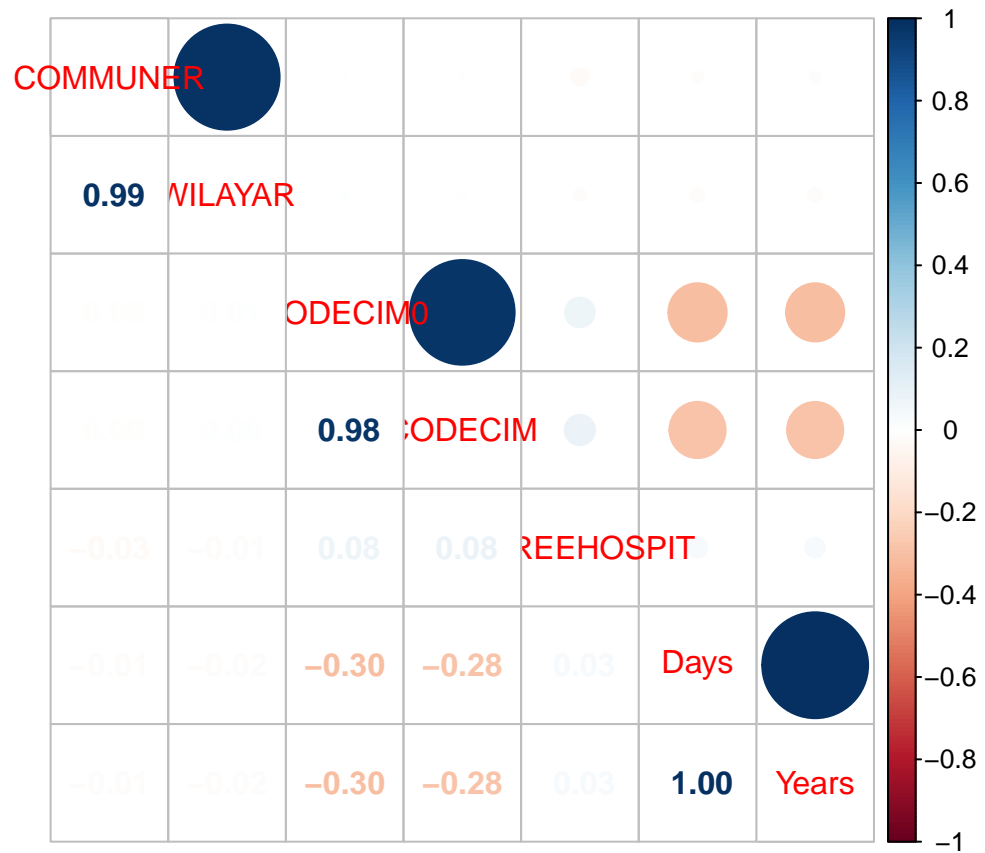
```
corrplot(M, method = 'square', order = 'FPC', type = 'lower', diag = FALSE)
```



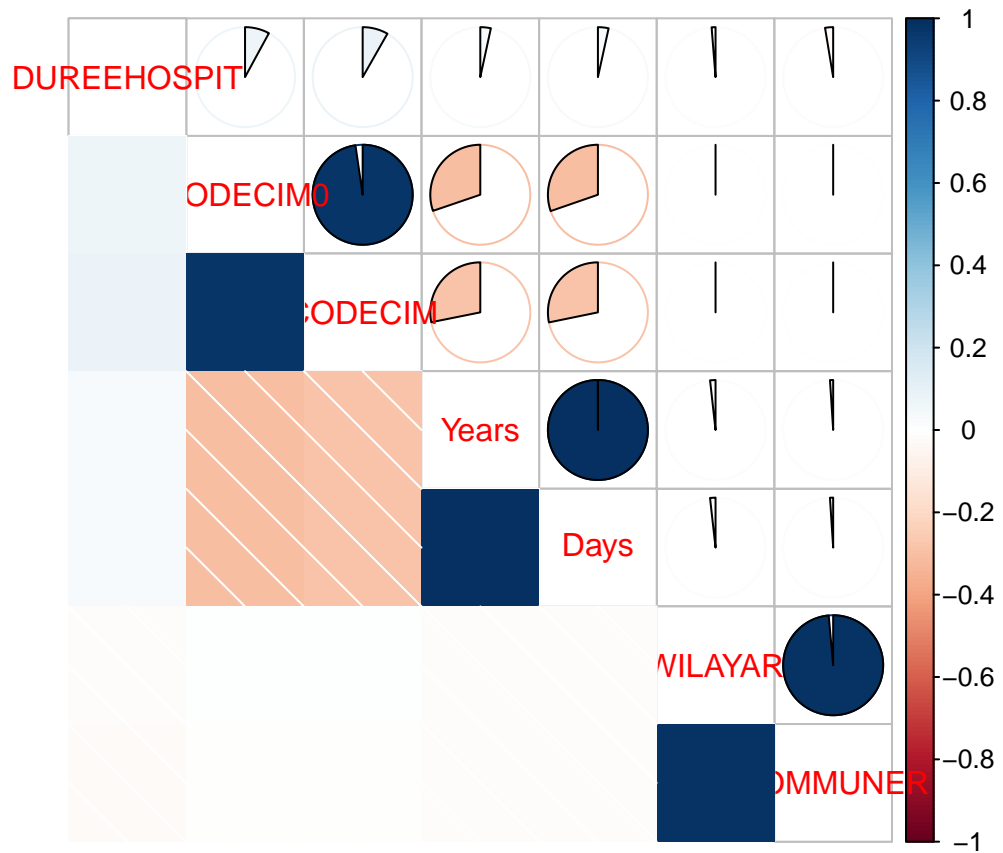
```
corrplot(M, method = 'ellipse', order = 'AOE', type = 'upper')
```



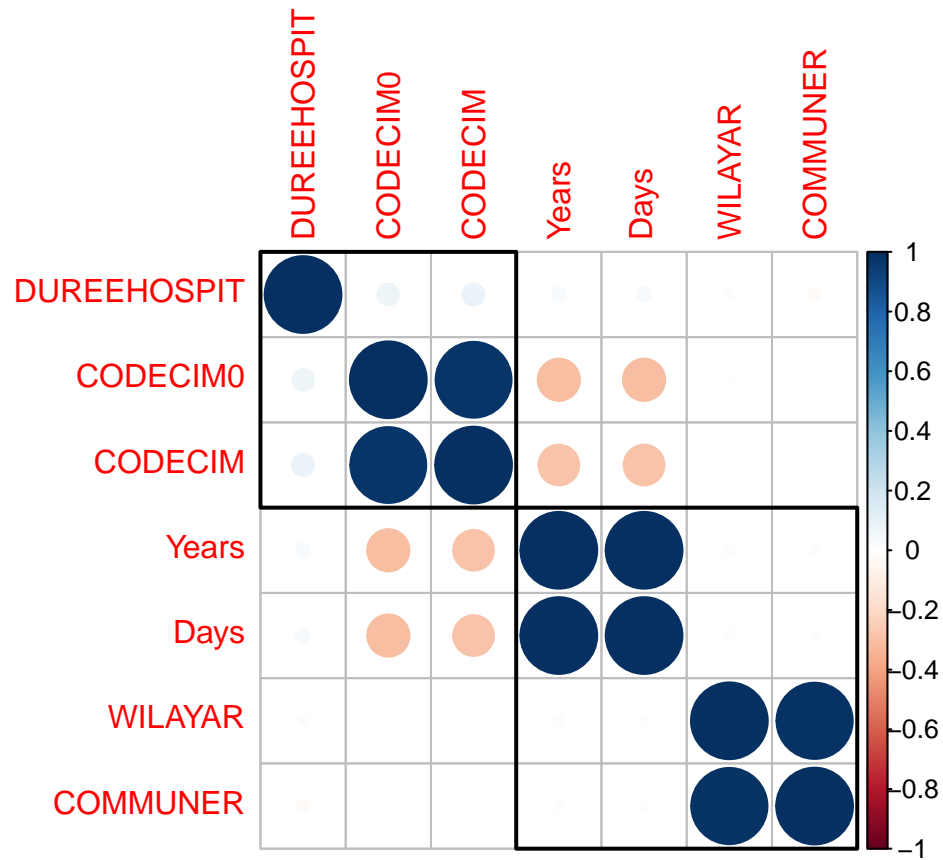
```
corrplot.mixed(M, order = 'AOE')
```



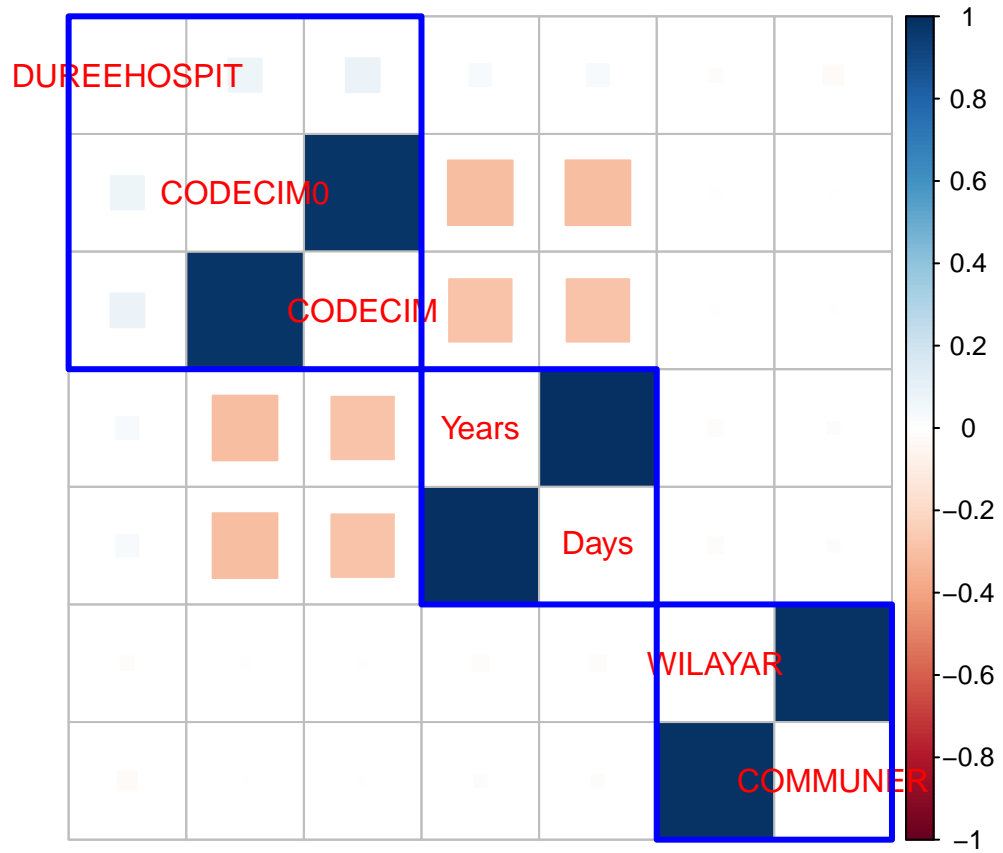
```
corrplot.mixed(M, lower = 'shade', upper = 'pie', order = 'hclust')
```



```
corrplot(M, order = 'hclust', addrect = 2)
```



```
corrplot(M, method = 'square', diag = FALSE, order = 'hclust',
         addrect = 3, rect.col = 'blue', rect.lwd = 3, tl.pos = 'd')
```



Analyse multivariée