Mortalité Hospitaliere

# load dada 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 7 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] 13946

ncol(data)

## [1] 14

dim(data)

## [1] 13946 14

names(data)

## [1] "DINS" "WILAYAR" "COMMUNER" "LD"   
## [5] "STRUCTURED" "SERVICEHOSPIT" "DUREEHOSPIT" "SEX"   
## [9] "Years" "Days" "Profession" "CD"   
## [13] "CODECIM0" "CODECIM"

## view data structure deces

### str

str(data)

## 'data.frame': 13946 obs. of 14 variables:  
## $ DINS : Date, format: "2020-01-27" "2019-08-10" ...  
## $ WILAYAR : int 17000 17000 17000 17000 17000 17000 17000 17000 17000 17000 ...  
## $ COMMUNER : int 935 917 935 947 920 935 935 935 917 917 ...  
## $ LD : Factor w/ 5 levels "AAP","DOM","SSP",..: 3 3 2 3 3 3 3 3 2 3 ...  
## $ STRUCTURED : Factor w/ 10 levels "1","2","3","4",..: 3 6 3 3 3 3 3 3 6 6 ...  
## $ SERVICEHOSPIT: Factor w/ 23 levels "0","1","2","3",..: 8 20 20 20 11 20 20 11 21 20 ...  
## $ DUREEHOSPIT : int 0 1 0 4 5 1 0 1 0 1 ...  
## $ SEX : Factor w/ 2 levels "F","M": 2 2 2 2 1 1 2 1 1 1 ...  
## $ Years : int 71 56 85 77 0 84 80 0 88 36 ...  
## $ Days : int 26100 20517 31101 28125 5 31044 29454 1 32391 13366 ...  
## $ Profession : Factor w/ 16 levels "0","1","3","4",..: 1 1 1 1 10 11 1 10 1 1 ...  
## $ CD : Factor w/ 3 levels "CI","CN","CV": 2 2 1 2 2 2 2 2 1 2 ...  
## $ CODECIM0 : int 10 9 18 10 16 9 1 17 0 10 ...  
## $ CODECIM : int 751 690 1370 751 1145 675 10 1271 0 751 ...

### glimpse

glimpse(data)

## Rows: 13,946  
## Columns: 14  
## $ DINS <date> 2020-01-27, 2019-08-10, 2020-01-24, 2020-01-24, 2020-01~  
## $ WILAYAR <int> 17000, 17000, 17000, 17000, 17000, 17000, 17000, 17000, ~  
## $ COMMUNER <int> 935, 917, 935, 947, 920, 935, 935, 935, 917, 917, 935, 9~  
## $ LD <fct> SSP, SSP, DOM, SSP, SSP, SSP, SSP, SSP, DOM, SSP, SSP, S~  
## $ STRUCTURED <fct> 3, 6, 3, 3, 3, 3, 3, 3, 6, 6, 3, 6, 3, 3, 3, 3, 3, 3, 5,~  
## $ SERVICEHOSPIT <fct> 7, 20, 20, 20, 10, 20, 20, 10, 21, 20, 20, 21, 20, 10, 2~  
## $ DUREEHOSPIT <int> 0, 1, 0, 4, 5, 1, 0, 1, 0, 1, 0, 1, 1, 0, 2, 2, 0, 1, 12~  
## $ SEX <fct> M, M, M, M, F, F, M, F, F, F, M, M, F, M, F, M, M, M, M,~  
## $ Years <int> 71, 56, 85, 77, 0, 84, 80, 0, 88, 36, 81, 45, 31, 0, 31,~  
## $ Days <int> 26100, 20517, 31101, 28125, 5, 31044, 29454, 1, 32391, 1~  
## $ Profession <fct> 0, 0, 0, 0, 12, 13, 0, 12, 0, 0, 0, 0, 0, 12, 0, 0, 12, ~  
## $ CD <fct> CN, CN, CI, CN, CN, CN, CN, CN, CI, CN, CN, CN, CN, CN, ~  
## $ CODECIM0 <int> 10, 9, 18, 10, 16, 9, 1, 17, 0, 10, 6, 9, 18, 16, 18, 1,~  
## $ CODECIM <int> 751, 690, 1370, 751, 1145, 675, 10, 1271, 0, 751, 517, 6~

### skimr

#library(skimr)  
#skim(data)

## modalité des variable deces

unique(data$LD)

## [1] SSP DOM VP AAP 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 13 1 19 11 20 15 16 3 7 6 5 4 14 9   
## 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

## summary data deces

summary(data)

## DINS WILAYAR COMMUNER LD   
## Min. :0009-11-20 Min. : 1000 Min. : 1.0 AAP : 537   
## 1st Qu.:2018-01-30 1st Qu.:17000 1st Qu.: 916.0 DOM : 699   
## Median :2019-11-24 Median :17000 Median : 924.0 SSP :12548   
## Mean :2019-04-17 Mean :17166 Mean : 935.6 SSPV: 11   
## 3rd Qu.:2021-03-19 3rd Qu.:17000 3rd Qu.: 935.0 VP : 151   
## Max. :2022-05-31 Max. :47000 Max. :2297.0   
## NA's :52   
## STRUCTURED SERVICEHOSPIT DUREEHOSPIT SEX Years   
## 1 :4508 20 :5199 Min. :-4886.0 F:5790 Min. : 0.00   
## 2 :2170 10 :3100 1st Qu.: 0.0 M:8156 1st Qu.: 0.00   
## 4 :2065 18 :1427 Median : 1.0 Median : 52.00   
## 5 :1824 7 :1126 Mean : 591.1 Mean : 43.39   
## 3 :1714 21 : 913 3rd Qu.: 5.0 3rd Qu.: 76.00   
## 6 : 814 15 : 709 Max. :29220.0 Max. :100.00   
## (Other): 851 (Other):1472   
## Days Profession CD CODECIM0 CODECIM   
## Min. : -279 0 :9423 CI: 592 Min. : 0.000 Min. : 0.0   
## 1st Qu.: 52 1 :2415 CN:12992 1st Qu.: 4.000 1st Qu.: 350.0   
## Median :19226 20 : 726 CV: 362 Median :10.000 Median : 711.0   
## Mean :15986 13 : 419 Mean : 9.687 Mean : 723.8   
## 3rd Qu.:28106 12 : 408 3rd Qu.:16.000 3rd Qu.:1155.0   
## Max. :36857 19 : 378 Max. :23.000 Max. :2039.0   
## (Other): 177

## 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 2019-08-10 17000 917 SSP 6 20 1 M  
## 3 2020-01-24 17000 935 DOM 3 20 0 M  
## 4 2020-01-24 17000 947 SSP 3 20 4 M  
## 5 2020-01-24 17000 920 SSP 3 10 5 F  
## 6 2019-11-28 17000 935 SSP 3 20 1 F  
## Years Days Profession CD CODECIM0 CODECIM  
## 1 71 26100 0 CN 10 751  
## 2 56 20517 0 CN 9 690  
## 3 85 31101 0 CI 18 1370  
## 4 77 28125 0 CN 10 751  
## 5 0 5 12 CN 16 1145  
## 6 84 31044 13 CN 9 675

### tail

tail(data,6)

## DINS WILAYAR COMMUNER LD STRUCTURED SERVICEHOSPIT DUREEHOSPIT SEX  
## 13941 2017-05-29 17000 924 SSP 2 10 1 M  
## 13942 2017-01-18 17000 924 SSP 2 20 93 M  
## 13943 2017-01-07 17000 926 SSP 2 0 5561 F  
## 13944 2017-01-05 17000 924 SSP 2 0 4829 M  
## 13945 2017-01-02 17000 924 SSP 2 20 5556 M  
## 13946 2017-01-03 17000 924 SSP 2 0 5192 M  
## Years Days Profession CD CODECIM0 CODECIM  
## 13941 0 1 0 CN 0 0  
## 13942 60 22253 0 CN 2 289  
## 13943 15 5651 0 CN 2 286  
## 13944 15 5587 0 CN 10 754  
## 13945 15 5615 0 CN 9 635  
## 13946 88 32293 0 CN 9 638

### 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).

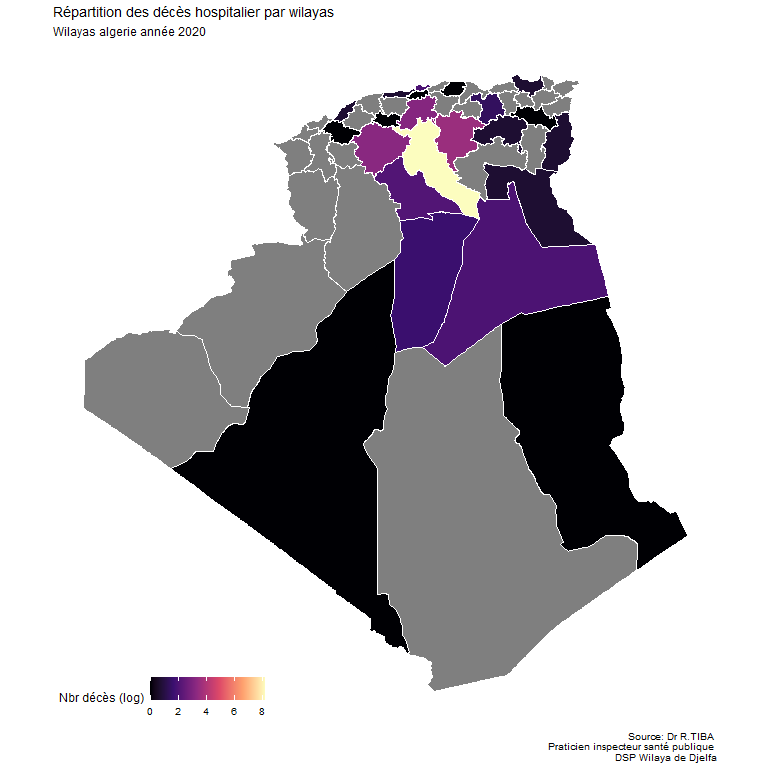
## lieux du deces

### SIG Deces par wilaya de Résidence

data %>%  
 select(DINS,WILAYAR) %>%   
 filter(DINS >= dt1 & DINS >= dt2 ) %>%   
 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

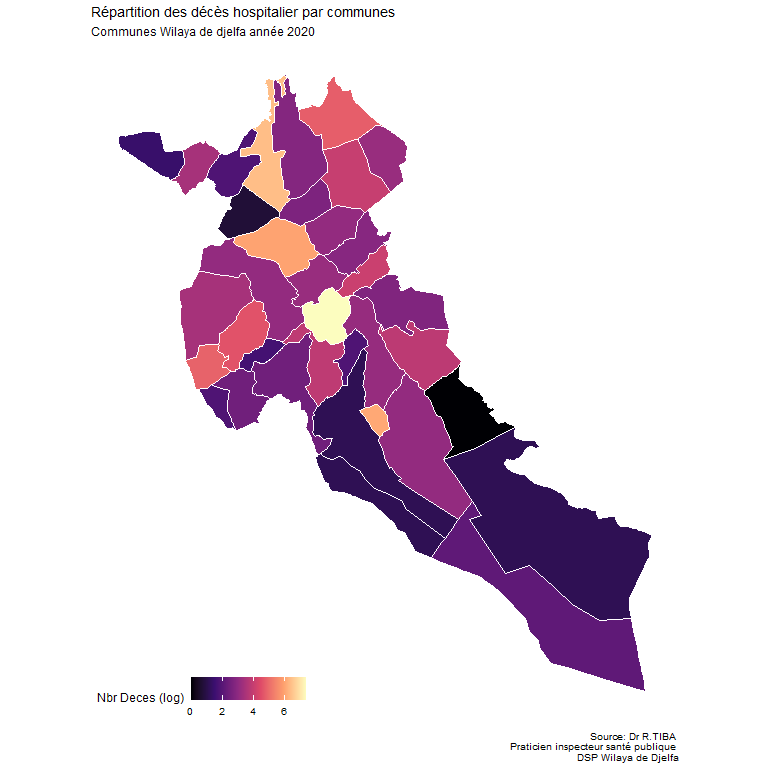
deces\_w\_djelfa <-left\_join(w\_algeria, df, by='CC\_1')  
wdjelfa <- deces\_w\_djelfa %>%   
 select(NAME\_1,CC\_1,number\_wil) %>%   
 mutate(number\_wil = log(number\_wil))  
  
ggplot()+  
 geom\_sf(data=wdjelfa,aes(fill=number\_wil), color = "white", lwd = 0.05) +  
 scale\_fill\_viridis\_c(option = "magma", name = "Nbr décès (log)") +  
 theme\_map() +   
 theme(legend.direction="horizontal") +  
 labs(title = "Répartition des décès hospitalier par wilayas",   
 subtitle = "Wilayas algerie année 2020",   
 caption = "Source: Dr R.TIBA \n Praticien inspecteur santé publique \n DSP Wilaya de Djelfa")+  
 coord\_sf(crs = "+proj=robin")



### SIG Deces par commune de Résidence

data %>%  
 select(DINS,WILAYAR,COMMUNER) %>%   
 filter(DINS >= dt1 & DINS >= dt2 & 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 Faidh 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 = "Communes Wilaya de djelfa année 2020",   
 caption = "Source: Dr R.TIBA \n Praticien inspecteur santé publique \n DSP Wilaya de Djelfa")+  
 coord\_sf(crs = "+proj=robin")



### lieux:table

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

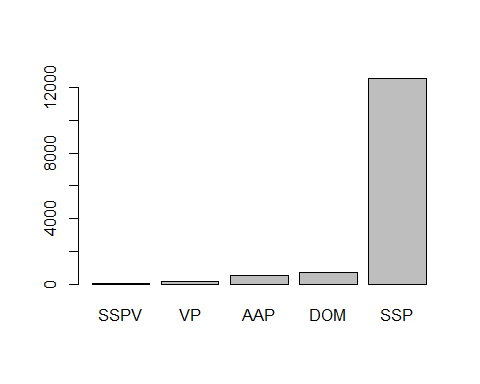
##   
## SSPV VP AAP DOM SSP   
## 11 151 537 699 12548

summary(lieux)

## Number of cases in table: 13946   
## Number of factors: 1

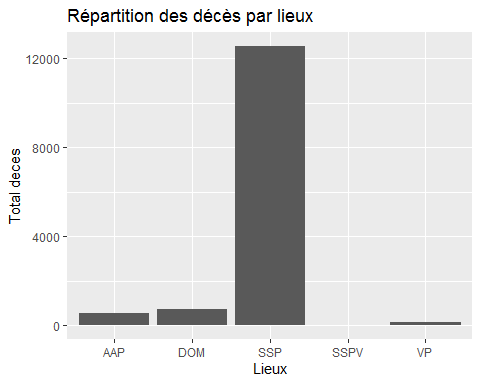
### lieux:graphic:r-base

barplot(sort(lieux))



### lieux:grafic:r-ggplot2

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



## structure sanitaire

### structure:table

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

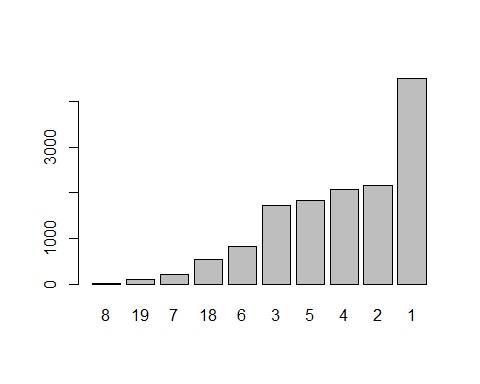
##   
## 8 19 7 18 6 3 5 4 2 1   
## 1 97 206 547 814 1714 1824 2065 2170 4508

summary(structure)

## Number of cases in table: 13946   
## Number of factors: 1

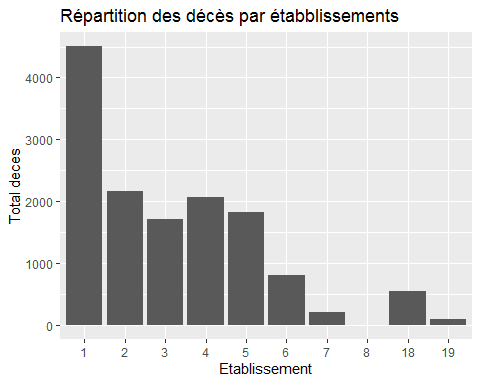
### structure:graphic r-base

barplot(sort(structure))



### structure:graphic:r-ggplot2

ggplot(data = data, mapping = aes(x = STRUCTURED)) +  
 geom\_bar()+  
 labs(title = "Répartition des décès par étabblissements",  
 x = "Etablissement",   
 y = "Total deces")



## service du deces

### <service:table>

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

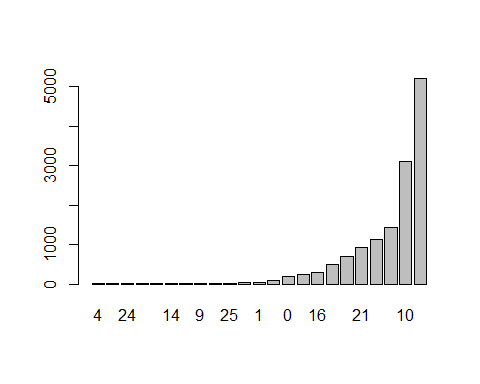
##   
## 4 17 24 12 19 14 3 9 11 25 8 1 5 0 2 16   
## 1 1 1 2 4 5 7 7 14 18 27 51 94 183 250 302   
## 6 15 21 7 18 10 20   
## 505 709 913 1126 1427 3100 5199

summary(service)

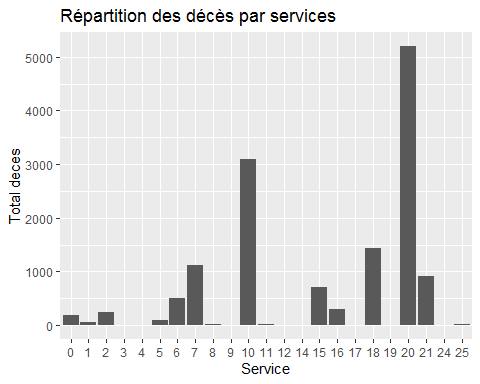
## Number of cases in table: 13946   
## Number of factors: 1

### <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")



## age en année

### age:table

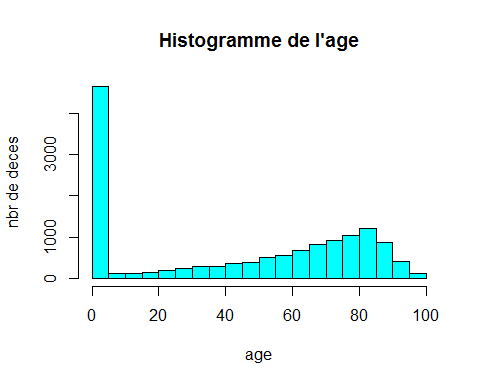
#clag <- c(0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100)  
clag <- c(0,10,20,30,40,50,60,70,80,90,100)  
data$age20cl <- cut(data$Years,clag,include.lowest = TRUE)  
table(data$age20cl)

##   
## [0,10] (10,20] (20,30] (30,40] (40,50] (50,60] (60,70] (70,80]   
## 4770 272 440 591 741 1059 1501 1943   
## (80,90] (90,100]   
## 2090 539

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

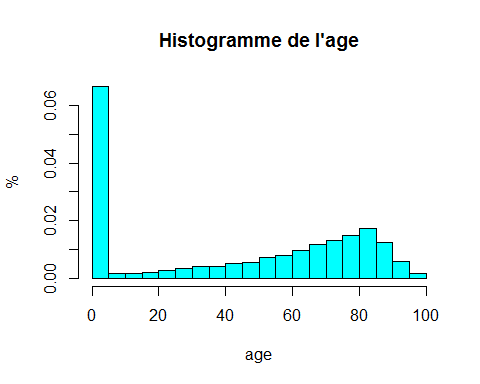
### age:grafic:r-base

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



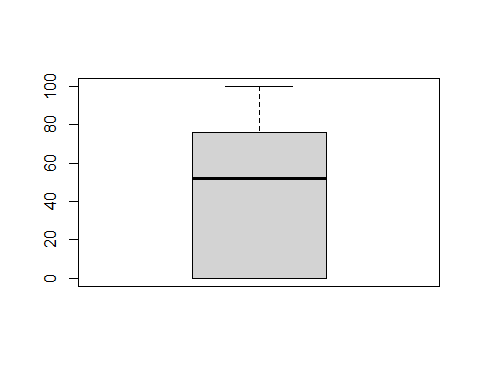
### age:grafic:r-base

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



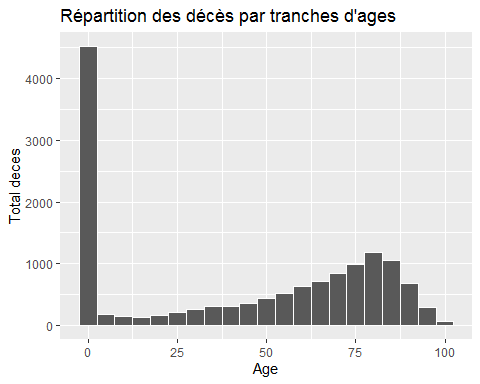
### age:grafic:r-base

boxplot(data$Years)

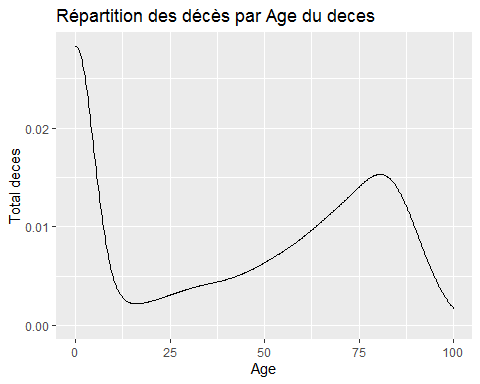


### age:grafic:t-ggplot2

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")

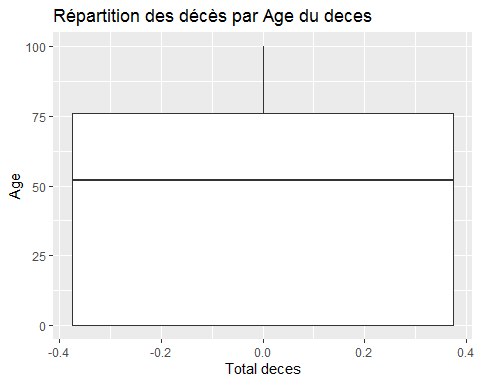
 ### age:grafic:t-ggplot2

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")

 ## sexe

### sexe:table

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

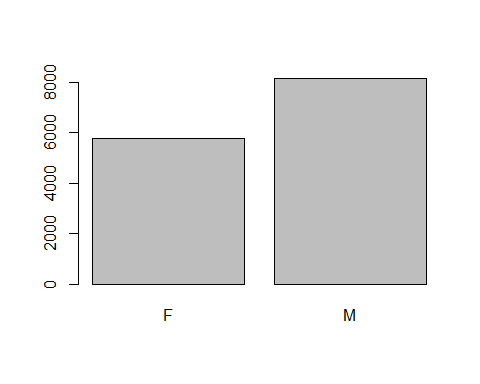
##   
## F M   
## 5790 8156

summary(sexe)

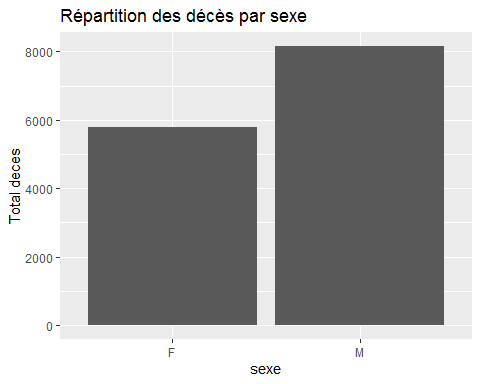
## Number of cases in table: 13946   
## Number of factors: 1

### 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")



## Profession

### profession:table

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

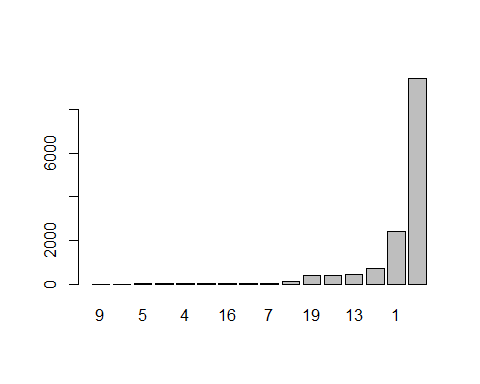
##   
## 9 15 5 3 4 14 16 6 7 11 19 12 13 20 1 0   
## 1 1 4 5 5 7 12 17 31 94 378 408 419 726 2415 9423

summary(Profession)

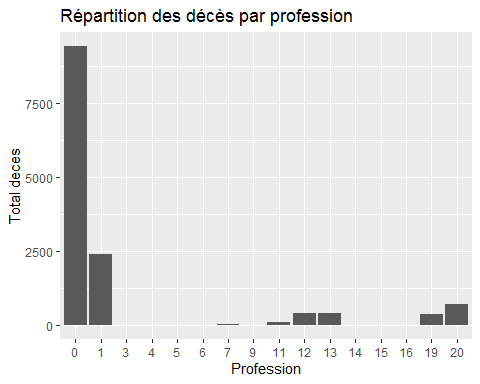
## Number of cases in table: 13946   
## Number of factors: 1

### 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")



# 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

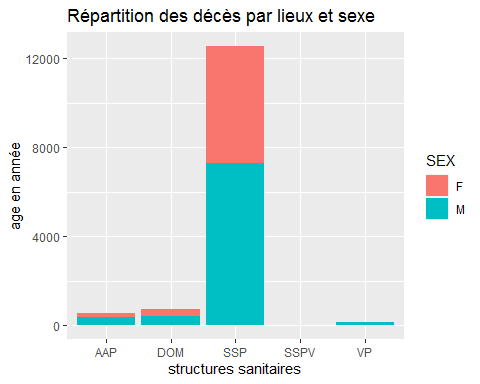
Quand on veut croiser deux variables qualitatives, on fait un tableau croisé. Comme pour un tri à plat ceci s’obtient avec la fonction table de R, mais à laquelle on passe cette fois deux variables en argument.

### lieux du deces et le sexe

table(data$LD, data$SEX)

##   
## F M  
## AAP 187 350  
## DOM 305 394  
## SSP 5259 7289  
## SSPV 7 4  
## VP 32 119

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")

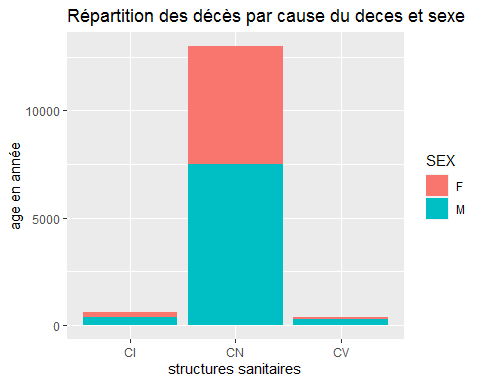


### causes du deces et le sexe

table(data$CD, data$SEX)

##   
## F M  
## CI 238 354  
## CN 5483 7509  
## CV 69 293

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")

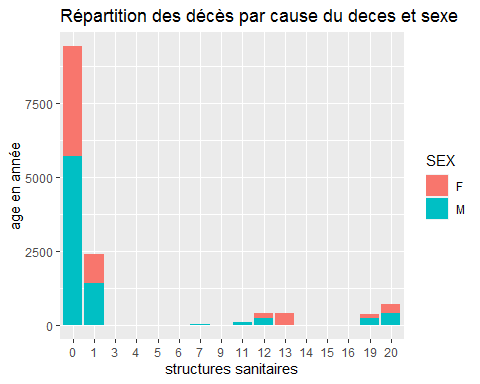


### la profession du deces et le sexe

table(data$Profession, data$SEX)

##   
## F M  
## 0 3715 5708  
## 1 993 1422  
## 3 0 5  
## 4 0 5  
## 5 0 4  
## 6 7 10  
## 7 1 30  
## 9 0 1  
## 11 2 92  
## 12 180 228  
## 13 414 5  
## 14 3 4  
## 15 1 0  
## 16 8 4  
## 19 150 228  
## 20 316 410

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")

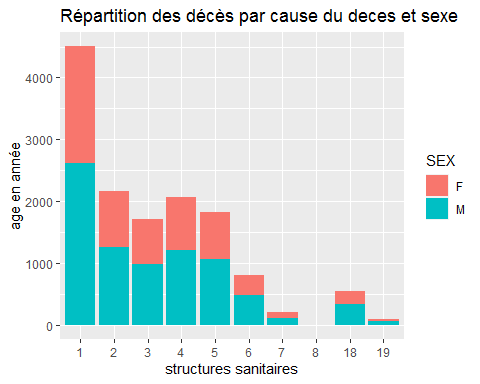


### la structure sanitaire du deces et le sexe

table(data$STRUCTURED, data$SEX)

##   
## F M  
## 1 1895 2613  
## 2 902 1268  
## 3 722 992  
## 4 857 1208  
## 5 762 1062  
## 6 328 486  
## 7 90 116  
## 8 1 0  
## 18 206 341  
## 19 27 70

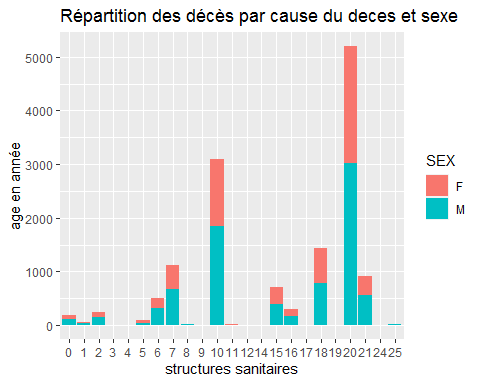
ggplot(data = data) +  
 geom\_bar(aes(x = STRUCTURED, fill = SEX))+  
 labs(title = "Répartition des décès par cause du deces et sexe",  
 x = "structures sanitaires",   
 y = "age en année")

 ### le service du deces et le sexe

table(data$SERVICEHOSPIT, data$SEX)

##   
## F M  
## 0 64 119  
## 1 20 31  
## 2 105 145  
## 3 6 1  
## 4 0 1  
## 5 56 38  
## 6 191 314  
## 7 454 672  
## 8 10 17  
## 9 3 4  
## 10 1248 1852  
## 11 6 8  
## 12 1 1  
## 14 3 2  
## 15 314 395  
## 16 134 168  
## 17 0 1  
## 18 638 789  
## 19 1 3  
## 20 2174 3025  
## 21 357 556  
## 24 0 1  
## 25 5 13

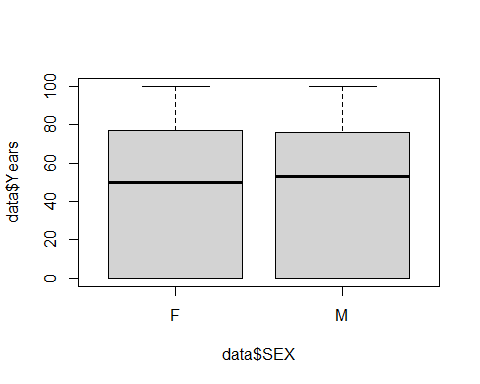
ggplot(data = data) +  
 geom\_bar(aes(x = SERVICEHOSPIT, fill = SEX))+  
 labs(title = "Répartition des décès par cause du deces et sexe",  
 x = "structures sanitaires",   
 y = "age en année") #+theme\_bw()



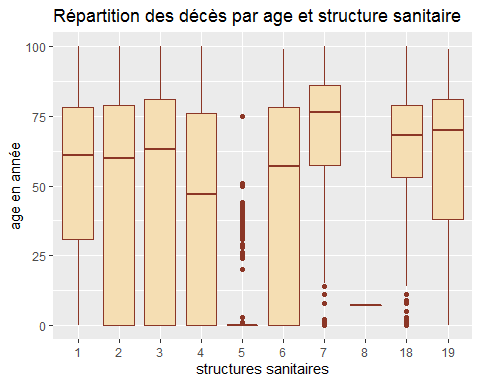
## Croisement d’une variable quantitative et d’une variable qualitative

### 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")



## Croisement de deux variables quantitatives

# Analyse multivariée

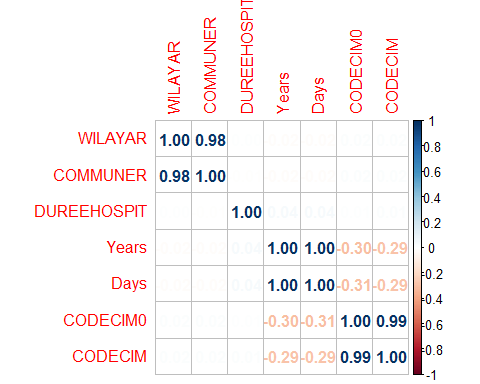
library(corrplot)

## corrplot 0.92 loaded

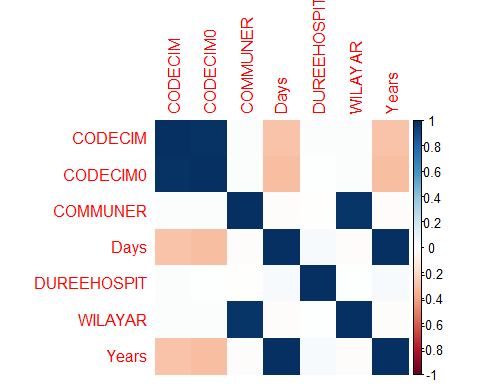
str(data)

## 'data.frame': 13946 obs. of 15 variables:  
## $ DINS : Date, format: "2020-01-27" "2019-08-10" ...  
## $ WILAYAR : int 17000 17000 17000 17000 17000 17000 17000 17000 17000 17000 ...  
## $ COMMUNER : int 935 917 935 947 920 935 935 935 917 917 ...  
## $ LD : Factor w/ 5 levels "AAP","DOM","SSP",..: 3 3 2 3 3 3 3 3 2 3 ...  
## $ STRUCTURED : Factor w/ 10 levels "1","2","3","4",..: 3 6 3 3 3 3 3 3 6 6 ...  
## $ SERVICEHOSPIT: Factor w/ 23 levels "0","1","2","3",..: 8 20 20 20 11 20 20 11 21 20 ...  
## $ DUREEHOSPIT : int 0 1 0 4 5 1 0 1 0 1 ...  
## $ SEX : Factor w/ 2 levels "F","M": 2 2 2 2 1 1 2 1 1 1 ...  
## $ Years : int 71 56 85 77 0 84 80 0 88 36 ...  
## $ Days : int 26100 20517 31101 28125 5 31044 29454 1 32391 13366 ...  
## $ Profession : Factor w/ 16 levels "0","1","3","4",..: 1 1 1 1 10 11 1 10 1 1 ...  
## $ CD : Factor w/ 3 levels "CI","CN","CV": 2 2 1 2 2 2 2 2 1 2 ...  
## $ CODECIM0 : int 10 9 18 10 16 9 1 17 0 10 ...  
## $ CODECIM : int 751 690 1370 751 1145 675 10 1271 0 751 ...  
## $ age20cl : Factor w/ 10 levels "[0,10]","(10,20]",..: 8 6 9 8 1 9 8 1 9 4 ...

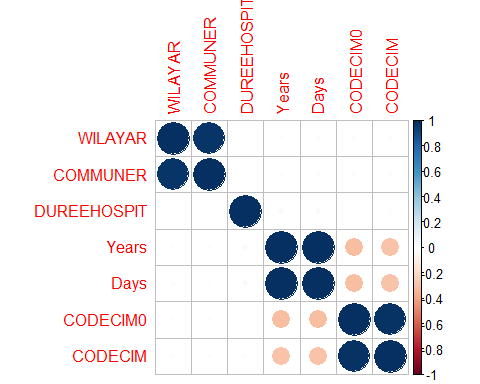
data\_n <- data %>%   
 select(WILAYAR,COMMUNER,DUREEHOSPIT,Years,Days,CODECIM0,CODECIM)   
  
M = cor(data\_n)  
corrplot(M, method = 'number') # colorful number



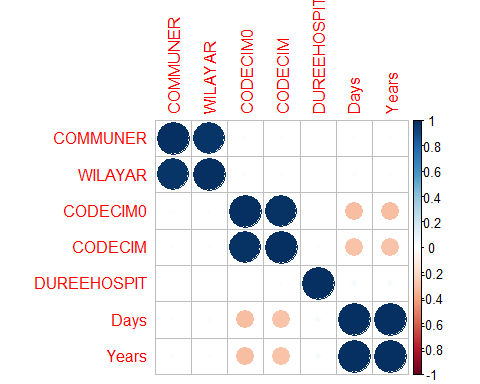
corrplot(M, method = 'color', order = 'alphabet')



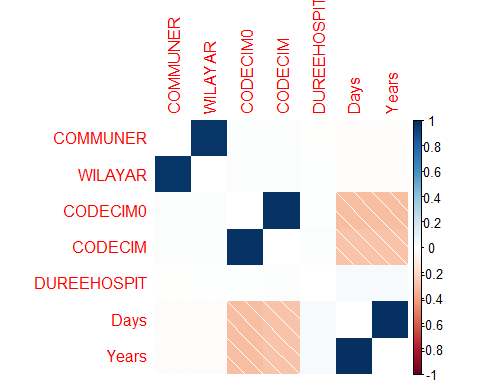
corrplot(M) # by default, method = 'circle'



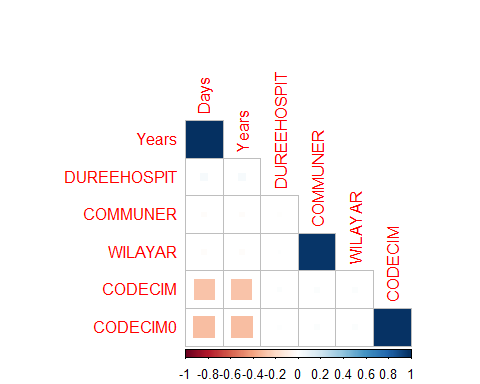
corrplot(M, order = 'AOE') # after 'AOE' reorder



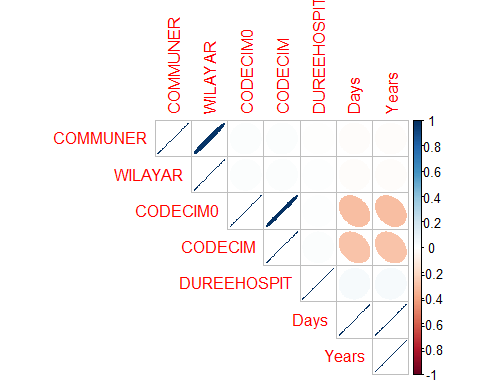
corrplot(M, method = 'shade', order = 'AOE', diag = FALSE)



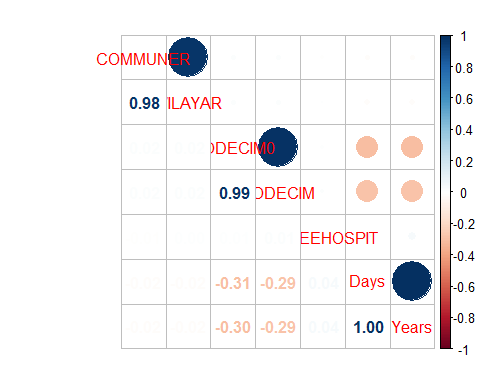
corrplot(M, method = 'square', order = 'FPC', type = 'lower', diag = FALSE)



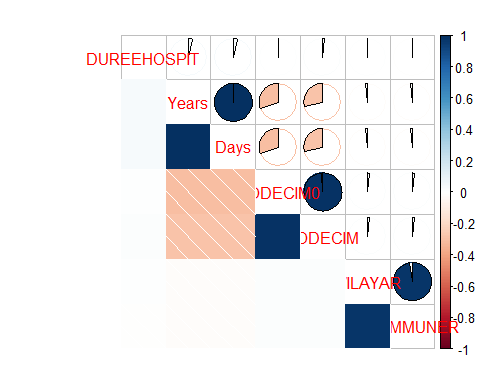
corrplot(M, method = 'ellipse', order = 'AOE', type = 'upper')



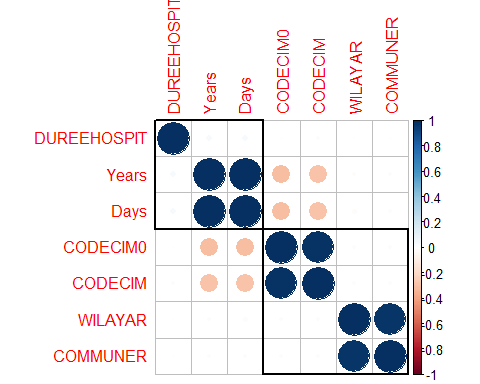
corrplot.mixed(M, order = 'AOE')



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')

