Utilisation de *R* pour l'analyse des données des bases médico-administratives

Séquence 1

Version 2023

L'outil statistique R (www.r-project.org) est devenu incontournable lorsqu'il s'agit d'analyser des données en santé. Libre, gratuit, adossé à une immense communauté d'utilisateurs, ce logiciel permet entre autres de réaliser des analyses statistiques, économiques, spatiales. Il facilite également la génération automatique de rapport d'analyses et la recherche reproductible.

Objectifs

- Prendre en main le logiciel R et l'environnement de programmation RStudio
- Gestion et analyses descriptives de données de santé

Nolwenn Le Meur, PhD

Enseignant chercheur

Département MéTiS - Méthodes quanTitatives en Santé publique

EHESP | UMR CNRS 6051 | INSERM U1309

Tél: +33 (0) 2 99 02 25 14

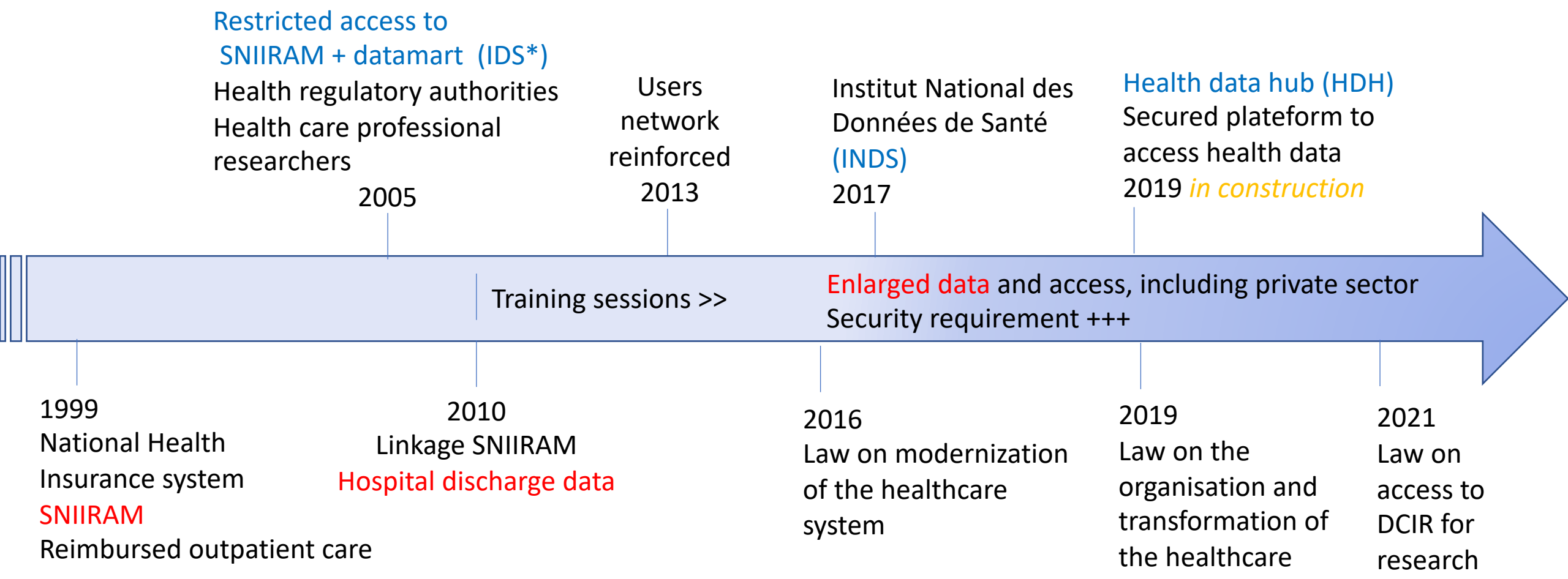
Nolwenn.lemeur@ehesp.fr

Utilisation de R pour l'analyse des données des bases medico-administratives

Nolwenn Le Meur

EHESP | UMR CNRS 6051 | INSERM U1309

French health big data experience



*IDS: Institut des données de santé – Institut for Health data – creation 2004

Package *TraMineR*

Le Meur *et al. BMC Health Services Research* (2015) 15:200
DOI 10.1186/s12913-015-0857-5

BMC Health Services Research

RESEARCH ARTICLE Open Access

Mining care trajectories using health administrative information systems: the use of state sequence analysis to assess disparities in prenatal care consumption

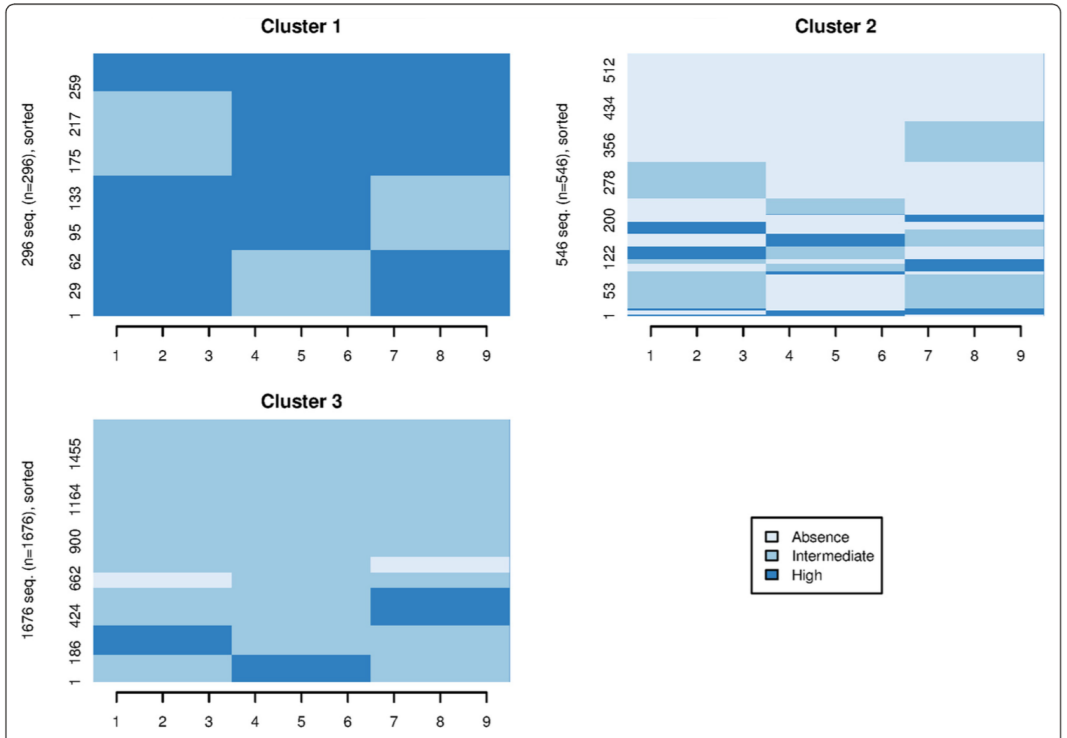


Fig. 3 Homogeneous groups of prenatal care trajectories. Each panel represents a group of homogeneous trajectories computed based on clustering analysis. Each cluster includes women with similar sequences of care trajectories (representing specific visits during pregnancy). Cluster 1 gathers women with mostly high level of care consumption. Cluster 2 includes women with low or absence of care. Cluster 3 includes women with mostly intermediate level of care. Each horizontal line (Y-axis) represents a pregnant woman. The X-axis represents the gestational month

Table 3 Importance of education and employment status for pregnant women care trajectories

Variables	OR [IC95%]		
	"High" level (Clusters 1 vs 2-3) N = 296	"Absence" level (Clusters 2 vs 1-3) N = 546	"Intermediate" level (Clusters 3 vs 1-2) N = 1676
Age			
14-20 year/old	0.25 [0.1-0.61]**	2.17 [1.53-3.08]***	NS
21-35 year/old	1	1	1
>35 year/old	NS	1.32 [1.01-1.72]*	NS
Population type			
Single women	NS	1.37 [1.10-1.70]**	0.82 [0.68-0.99]*
Education			
No Diploma	0.71 [0.54-0.92]**	NS	NS
Technical Education	NS	NS	0.81 [0.67-0.97]*
Employment			
Unemployed	NS	1.27 [1.03-1.55]*	0.76 [0.63-0.93]**
Blue-collar	NS	1.42 [1.15-1.73]***	NS
Precarious job	NS	NS	0.82 [0.69-0.98]*
Artisan	NS	NS	0.79 [0.67-0.97]*
Housing			
House built after 1999	1.34 [1.03-1.73]*	NS	NS

The results of the logistic regression models describe how each cluster membership relates to specific covariates
***p-value <0.001; **p-value <0.01; *p-value <0.05; NS: not significant

Recherche de groupes homogènes de trajectoire **Package *TraMineR***

Use of state sequence analysis for care pathway analysis: The example of multiple sclerosis.

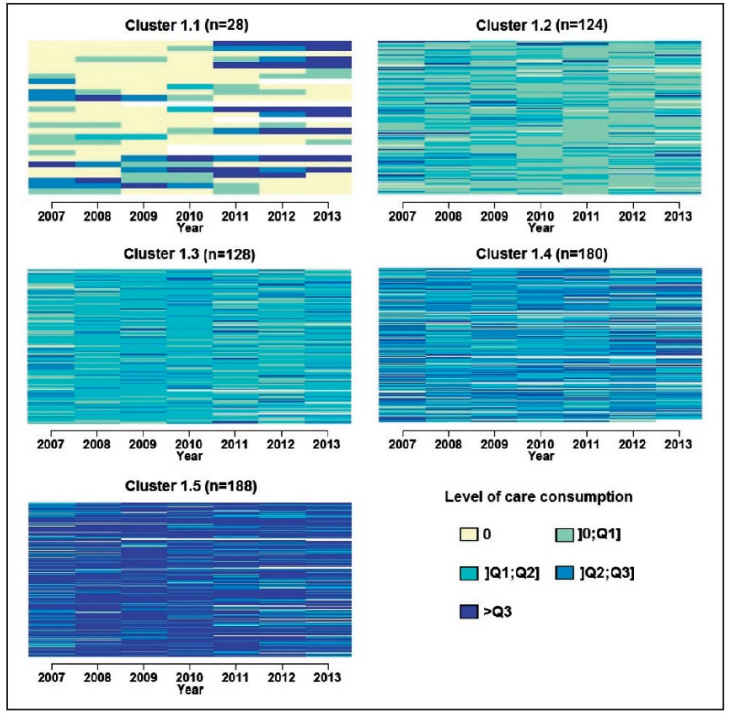


Figure 3. Index plots of clusters obtained for the care pathways of patients identified in 2007 after clustering with indel costs fixed at 0.9 and transition-based substitution costs ($n = 648$).

Roux J, Grimaud O, Leray E. Stat Methods Med Res. 2019 Jun;28(6):1651-1663.

Categorical state sequence analysis and regression tree to identify determinants of care trajectory in chronic disease: Example of end-stage renal disease.

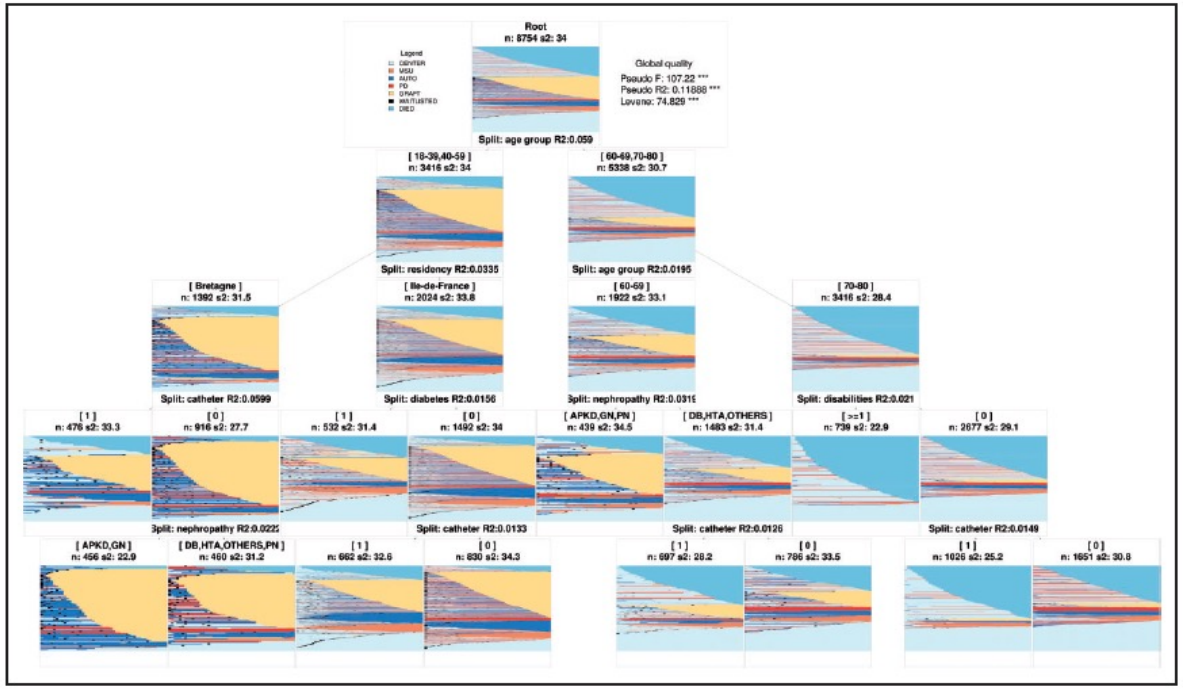


Figure 3. Determinants of homogeneous groups of RRT sequences among ESRD patients. In each panel, each row represents one patient care trajectory over the 48-month follow-up, sorted according to the final state (treatment modality). The RRT modalities can be in-center hemodialysis [CENTER] in aqua blue, hemodialysis in medical unit [MSU] in orange, autonomous hemodialysis [AUTO] in dark blue, peritoneal dialysis [PD] in red, death [DIED] in grey blue, transplantation [GRAFT] in yellow, and registration point to the transplantation waiting list [WAITLISTED] in black. Pseudo R2 or R2 shows the proportion of explained variance by the covariate. S2 is the variance of residuals. The pseudo F provides the statistical significance of the segmentation. The Levene's test gives the significance of variance equality within groups/panels.
APKD: autosomal dominant polycystic kidney disease; catheter: first dialysis with catheter; DN: diabetic nephropathy; GN: glomerulonephritis; HTA: hypertensive and vascular nephropathy; OTHERS: other causes or unknown; PN: pyelonephritis.

Le Meur N, Vigneau C, Lefort M, Lebbah S, Jais JP, Daugas E, Bayat S. Stat Methods Med Res. 2019 Jun;28(6):1731-1740.

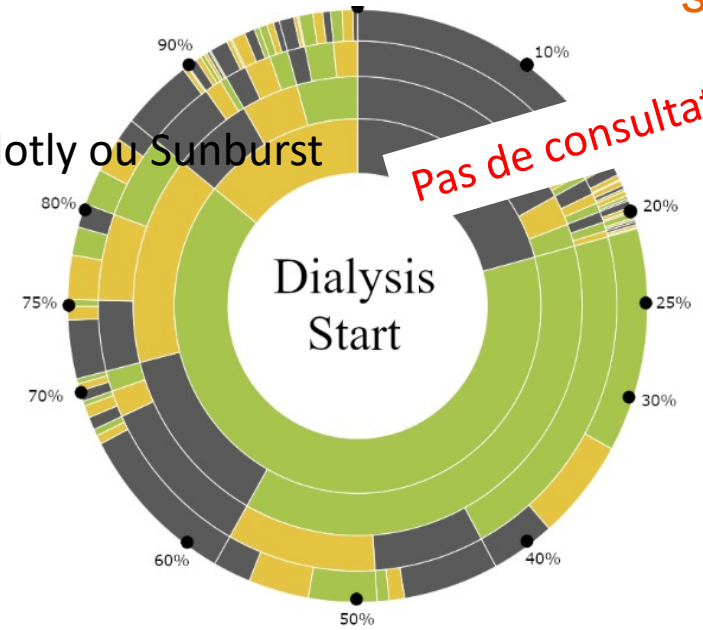
Trajectoire de soins avant dialysis chez les insuffisants rénaux terminaux. Packages Plotly ou Sunburst

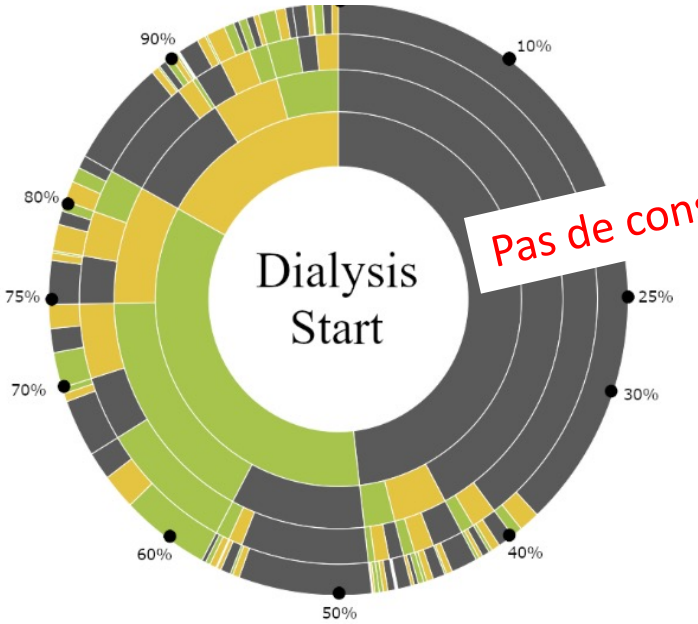
Programmé (N=6175)

Visite chez un néphrologue 2 ans avant la première séance de dialyse

En Urgence (N=2681)

Packages Plotly ou Sunburst





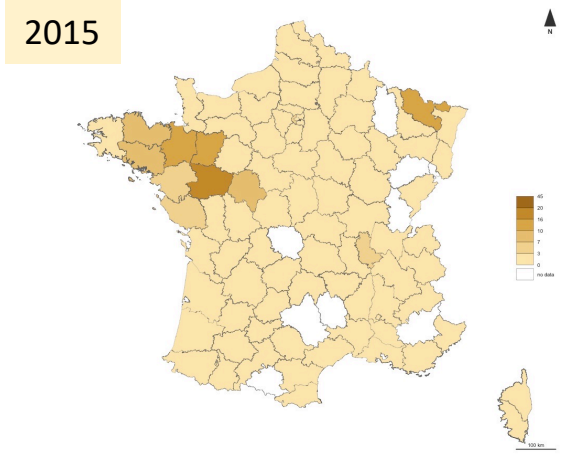
Nombre médian de visites : 6

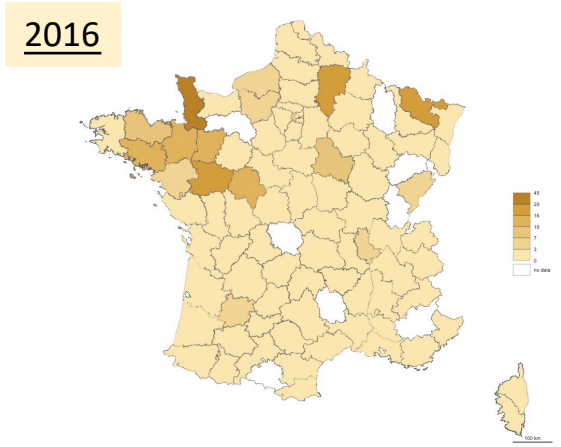
Nombre médian de visites : 2

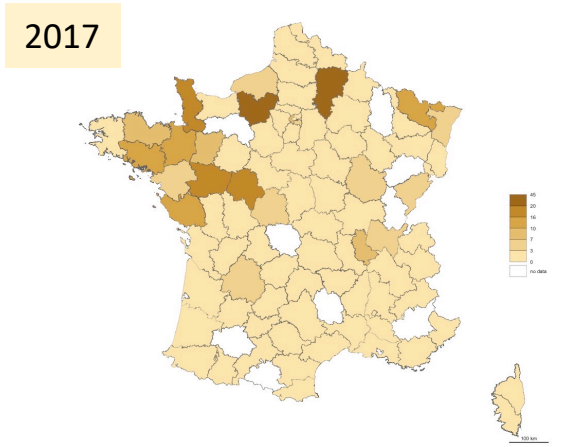
Raffray M, et al. 2019. Int J Environ Res Public Health

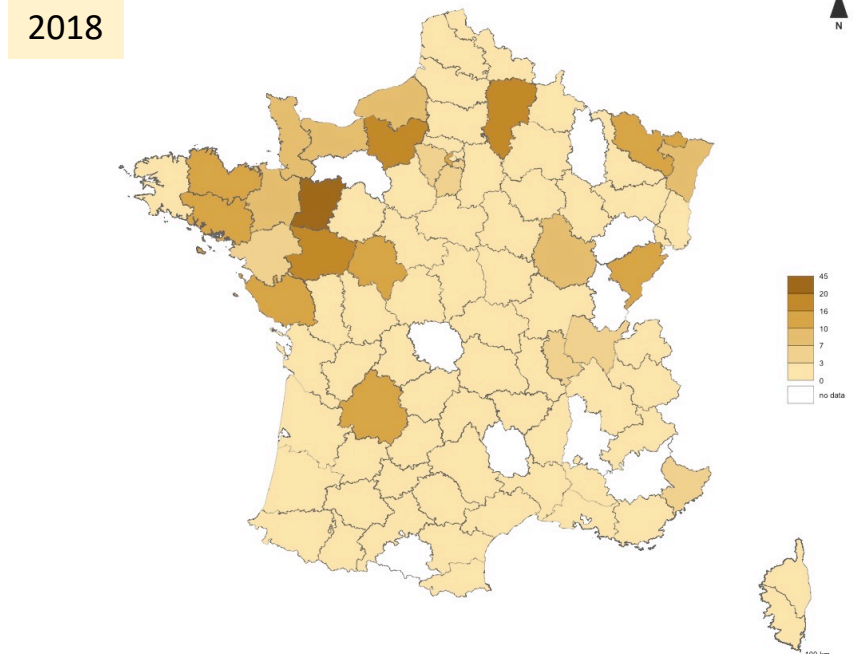
Dynamique géographique de l'évolution des interventions endovasculaires en ambulatoire pour AOMI en France métropolitaine de 2015 à 2019

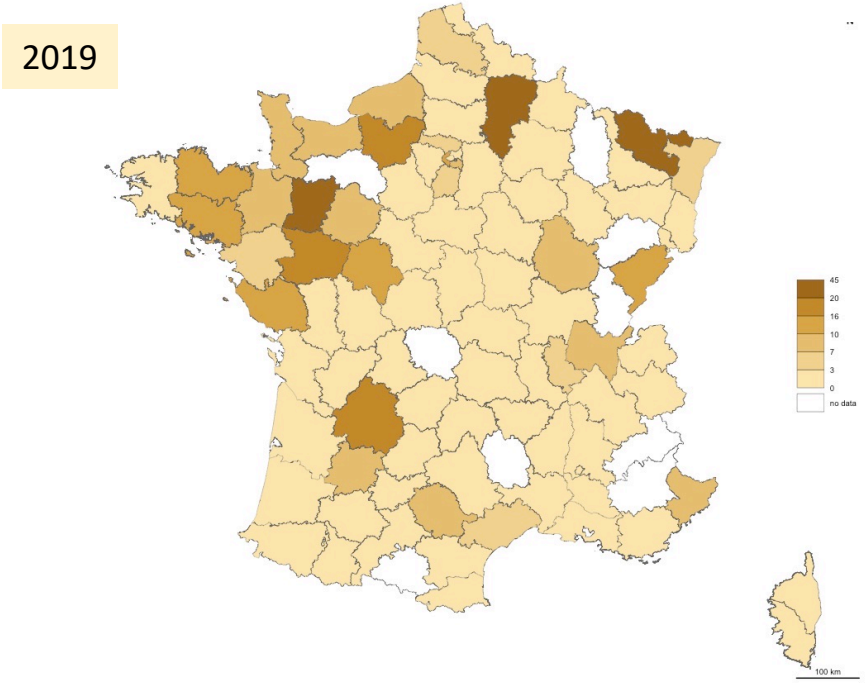
Packages sf et mapsf







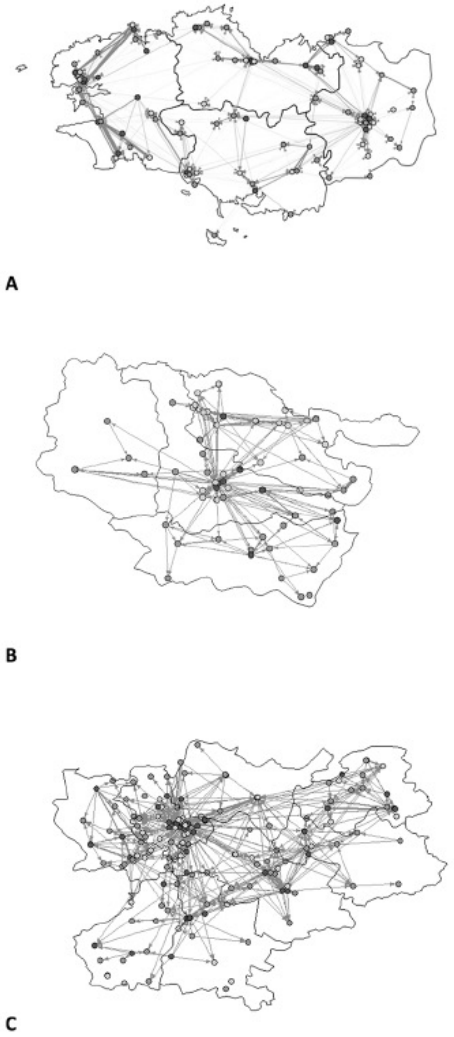




Le Meur, N., Padilla C., Ghoroubi N., Lamirault G., Chatellier G., and Gouëffic Y. 2022. "Geographical Disparities of Endovascular Revascularisations in Ambulatory Setting in France from 2015 to 2019." *European Journal of Vascular and Endovascular Surgery.* https://doi.org/10.1016/j.ejvs.2022.03.015.

Figure 1. Diversité des maillages territoriaux des établissements MCO-SSR.

Chaque panel est une région : (A) Bretagne (B) Lorraine (C) : Rhône Alpes. Les nœuds représentent les établissements. Les couleurs des nœuds symbolisent leur statut juridique : en blanc les établissements privés à but non lucratif ; en gris les établissements publics et en noir les établissements privé à but lucratif. Les arcs représentent les transferts de patients entre établissements (au moins 3).

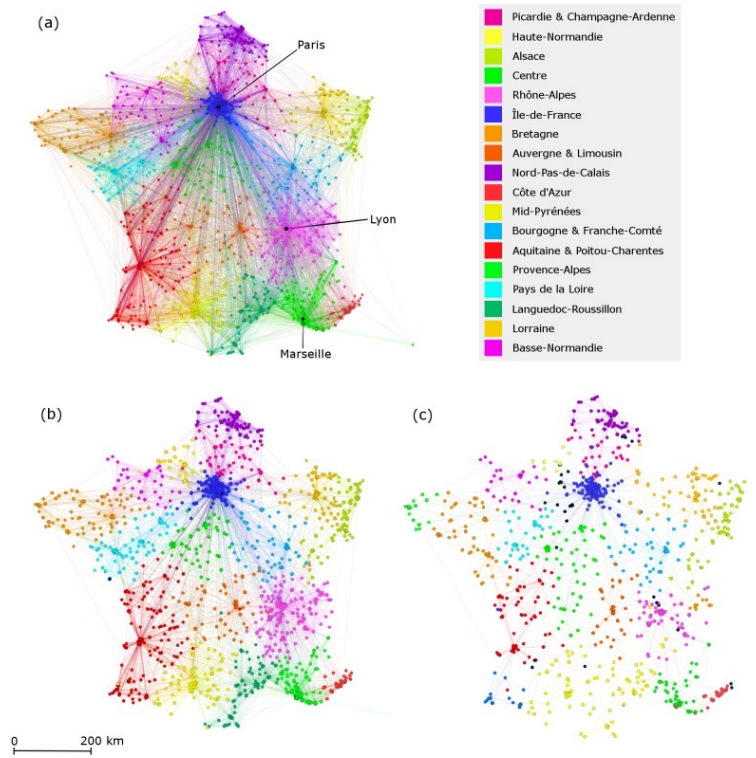


Modélisation des flux de patients : approches par graphe

Packages *igraph et sna*

Le Meur N, Ferrat L, Gao F, Quidu F, Louazel M. Maillage territorial des établissements de santé : apport des modèles issus de la théorie des graphes. *Journal de gestion et d'économie médicales* 2017;**35**(4):197.

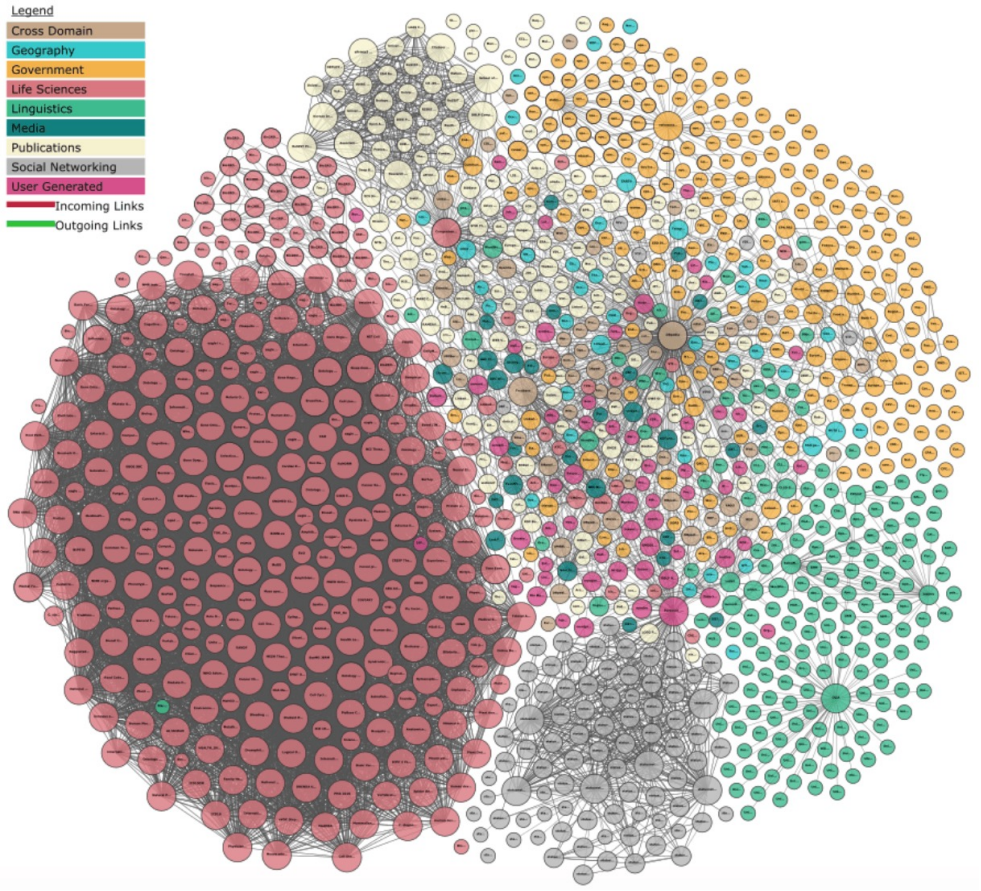
Spread of hospital-acquired infections: A comparison of healthcare networks.



Nekkab N, Astagneau P, Temime L, Crépey P PLoS Comput Biol. 2017 Aug 24;13(8):e1005666.

Data driven

Health care Data and The Web of linked Data – package *queryMed*



The Semantic Web
Linking Open Data cloud diagram 2017–08–22 (http://lod-cloud.net/)

R package *queryMed*: Semantic Web functions for linking pharmacological and medical knowledge to data.

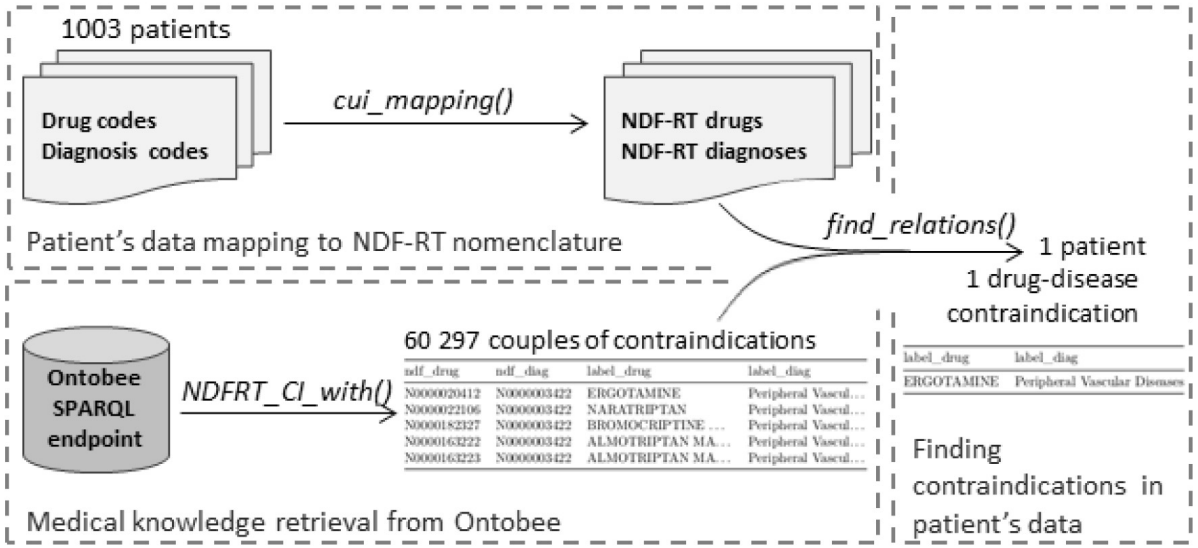


Figure 1. queryMed use in a pharmacovigilance oriented scenario.

Rivault Y, Dameron O, Le Meur N. Bioinformatics. 2019 Sep 1;35(17):3203-3205.

R and Data in RStudio environment

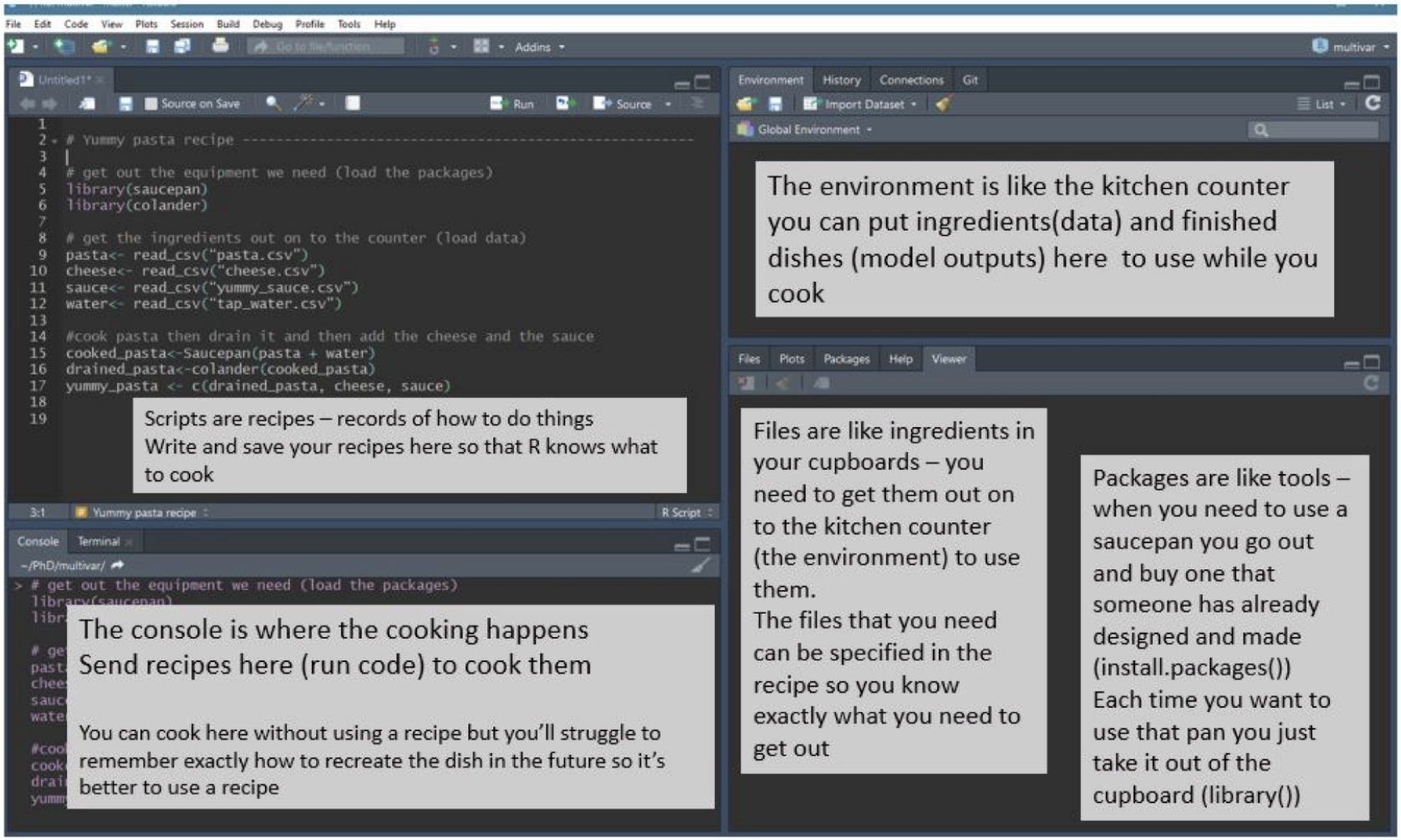
Nolwenn Le Meur

May-June 2023

R and RStudio

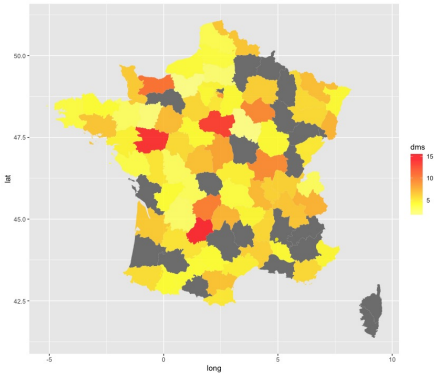
R is the engine - RStudio is the user friendly environment (GUI)

What is in your kitchen ?



The cake

Length of stay for endovascular surgery in France



Proportion of hospitals with length of stay below or above the national average by legal status.

	public	non lucratif	privé
below average	42.86	91.41	66.67
above average	57.14	8.59	33.33

Let's cook !

```
# Where am I?
getwd()
# Open the fridge (set your working directory)
setwd("~/Documents/Projets/01_EHESP/04_Cours/10_FC-R/data")
```

Or create a **R project** (recommended)

- create a shortcut to the path of your project working directory (setwd() no more needed)
- link to your versioning repository (if created)
- custom able environment

Write your recipe (R script *.R file)

To do what?
- save your code (and rerun it later on)
- comment your action/analysis and interpretation (#)

Do not forget to:
- comment your script (#)
- write simple and explicit variable name
- make it short

```
# Project: R teaching class
# N. Le Meur - created April 2023
# Last update 15 May 2023

# loading required libraries
library(tidyverse)
library(ggplot2)
```

Read data files in R

```r
# Take out the raw ingredients from the fridge and make them available on the kitchen counter
ccam20 <- read.csv("Open_ccam_20.CSV", header=T,
                   sep=";", na.strings =".")
ccam20[1:3, 1:3]
```

```
##     finess FinessGeo     acte
## 1 10007300  10007300 BAFA0060
## 2 10007300  10007300 BAFA0080
## 3 10007300  10007300 BAMA0040
```

- assignments (“<-”).
- object name: any roman letters, digits, “.” and “_” (non-initial position)
- avoid using system names.
- R is case sensitive: *myData* and *Mydata* are different names.

Access big data or infrastructure

Read big data files with **data.table** library and the *fread()* function

```
# OPEN_MEDIC_2017.CSV - 400 Mo, 1806702 rows and 21 columns
library(data.table)
med2017 <- fread("OPEN_MEDIC_2017.CSV", header=T,
                 dec = ",", nrows=150)
```

More details at https://stackoverflow.com

Connect to SQL database with the *odbc* package, which is a DBI compliant interface to using ODBC drivers.

```
library(odbc)
con <- dbConnect(odbc(),
                 Driver = "SQL Server",
                 Server = "mysqlhost",
                 Database = "mydbname",
                 UID = "myuser",
                 PWD = rstudioapi::askForPassword("Database password"),
                 Port = 1433)
```

More details at https://db.rstudio.com/getting-started/

What object have we got?

Structure:

```
str(ccam20)
```

```
## 'data.frame':    313491 obs. of  10 variables:
##  $ finess       : chr  "10007300" "10007300" "10007300" "10007300" ...
##  $ FinessGeo    : chr  "10007300" "10007300" "10007300" "10007300" ...
##  $ acte         : chr  "BAFA0060" "BAFA0080" "BAMA0040" "BCFA0030" ...
##  $ nb_sejsea    : int  15 13 16 16 644 28 19 12 46 348 ...
##  $ nb_actes     : int  15 13 17 16 644 28 19 12 46 348 ...
##  $ dms_globale  : chr  "0" "0" "0" "0" ...
##  $ nb_sej_0_nuit: int  15 13 16 16 644 28 19 12 46 348 ...
##  $ nb_acte_ambu : int  15 13 17 16 644 28 19 12 46 348 ...
##  $ dep          : chr  "1" "1" "1" "1" ...
##  $ reg          : int  84 84 84 84 84 84 84 84 84 84 ...
```

Or look at the **Environment panel**

R objects

- Everything in R is an object
- Every object has a class (type/mode)
- Every class of object has its specific methods (functions)
- To list your collection of objects in the work space use the function call *ls()*
- Common objects: vectors, matrices, arrays, lists, data frames, factors, ts, functions

Every class of object has its specific methods

```
summary(ccam20)
```

```
##     finess            FinessGeo             acte             nb_sejsea
##  Length:313491      Length:313491      Length:313491      Min.   :   11
##  Class :character   Class :character   Class :character   1st Qu.:   18
##  Mode  :character   Mode  :character   Mode  :character   Median :   35
##                                                           Mean   :  168
##                                                           3rd Qu.:   97
##                                                           Max.   :44413
##
##     nb_actes       dms_globale        nb_sej_0_nuit      nb_acte_ambu
##  Min.   :   11.0   Length:313491      Min.   :   11.0   Min.   :   11.0
##  1st Qu.:   19.0   Class :character   1st Qu.:   17.0   1st Qu.:   18.0
##  Median :   37.0   Mode  :character   Median :   32.0   Median :   33.0
##  Mean   :  224.6                      Mean   :  160.8   Mean   :  164.2
##  3rd Qu.:  108.0                      3rd Qu.:   83.0   3rd Qu.:   85.0
##  Max.   :50068.0                      Max.   :44237.0   Max.   :44237.0
##                                       NA's   :173871    NA's   :173871
##      dep                reg
##  Length:313491      Min.   : 1.00
##  Class :character   1st Qu.:27.00
##  Mode  :character   Median :52.00
##                     Mean   :51.54
##                     3rd Qu.:76.00
##                     Max.   :93.00
##
```

Data format

- vector
- matrix
- data frame (data table or tibble)
- list

Data types

- numeric (integer and double)
- logical (TRUE/FALSE)
- character (string)
- factor (categorical)
- NA (missing)

Vector (data format)

```r
# character vector
region<- c("GUADELOUPE", "LA REUNION","ILE_DE_FRANCE",
           "CENTRE-VAL-DE-LOIRE", "BOURGOGNE_FRANCHE_COMTE",
           "NORMANDIE", "NORD-PAS-DE-CALAIS",
           "ALSACE-CHAMPAGNE-ARDENNE-LORRAINE",
           "PAYS-DE-LA-LOIRE","BRETAGNE",
           "AQUITAINE-LIMOUSIN-POITOU-CHARENTES",
           "LANGUEDOC-ROUISSILLON-MIDI-PYRENEES","AUVERGNE-RHONE-ALPES",
           "PROVENDE-ALPES-COTE-D-AZUR")
# numeric vector
id <- c(1,6,78,9)
id <- c(1:4)
id <- seq(1,10,2)
```

Matrix (data format)

2 dimensional numerical object

```
mat1 <- matrix(c(1:4), nrow=2,
               dimnames = list(c("exposed", "non-exposed"),
                               c("ill", "non-ill")))
mat1
```

```
##             ill non-ill
## exposed       1       3
## non-exposed   2       4
```

Data frame and Cie (data format)

Rectangle data format with variables of different types

```
str(ccam20)
```

```
## 'data.frame':    313491 obs. of  10 variables:
##  $ finess       : chr  "10007300" "10007300" "10007300" "10007300" ...
##  $ FinessGeo    : chr  "10007300" "10007300" "10007300" "10007300" ...
##  $ acte         : chr  "BAFA0060" "BAFA0080" "BAMA0040" "BCFA0030" ...
##  $ nb_sejsea    : int  15 13 16 16 644 28 19 12 46 348 ...
##  $ nb_actes     : int  15 13 17 16 644 28 19 12 46 348 ...
##  $ dms_globale  : chr  "0" "0" "0" "0" ...
##  $ nb_sej_0_nuit: int  15 13 16 16 644 28 19 12 46 348 ...
##  $ nb_acte_ambu : int  15 13 17 16 644 28 19 12 46 348 ...
##  $ dep          : chr  "1" "1" "1" "1" ...
##  $ reg          : int  84 84 84 84 84 84 84 84 84 84 ...
```

- To access the variables in a data.frame use **'$'** or **indexing method [** ***row-wise*** **,** ***column-wise*****]**
- **data.table** and **tibble** inherit from data.frame
- **data.table** from the *data.table* package is optimized for big data management
- **tibble** from the *tidyverse* suite of packages is optimized for data management steps

Indexing matrices and data frame

Think battleship !

`m[2, ]` - Select a row

`m[ , 1]` - Select a column

`m[2, 3]` - Select an element

Accessing value of the matrix

Indexing

Vectors

```
region[1]
region[1:5]
region[-10]
```

Matrices and data frame

```
ccam20[2, ]
ccam20[ ,1]
ccam20[2:10, 3]
ccam20[seq(2,100,2), 3]
ccam20[which(ccam$reg==53), ]

# using colnames
ccam20[2,"acte"]
```

Lists (data format)

- like a vector
- combine objects of any type and structure (e.g. list of vectors, vectors and data.frame, list of lists ???)
- lists are often used for output of statistical routines in R

Lists (data format)

```r
test.result<- t.test(x=1:10, y = c(7:20))
typeof(test.result)
```

```
## [1] "list"
```

```r
str(test.result)
```

```
## List of 10
##  $ statistic  : Named num -5.43
##   ..- attr(*, "names")= chr "t"
##  $ parameter  : Named num 22
##   ..- attr(*, "names")= chr "df"
##  $ p.value    : num 1.86e-05
##  $ conf.int   : num [1:2] -11.05 -4.95
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num [1:2] 5.5 13.5
##   ..- attr(*, "names")= chr [1:2] "mean of x" "mean of y"
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "difference in means"
##  $ stderr     : num 1.47
##  $ alternative: chr "two.sided"
##  $ method     : chr "Welch Two Sample t-test"
##  $ data.name  : chr "1:10 and c(7:20)"
##  - attr(*, "class")= chr "htest"
```

Indexing Lists

```
test.result[1]
```

```
## $statistic
##        t
## -5.43493
```

```
test.result[[1]]
```

```
##        t
## -5.43493
```

```
test.result$p.value
```

```
## [1] 1.855282e-05
```

Factor (variable type)

For categorical variables

```
table(ccam20$reg)
```

```
##
##     1     4    11    24    27    28    32    44    52    53    75    76    84
##  3922  3373 51075 10264 11419 14720 27947 26763 16547 15230 30692 31009 38545
##    93
## 31985
```

```
ccam20$reg <- factor(ccam20$reg,  labels=region)
table(ccam20$reg)
```

```
##
##                          GUADELOUPE                           LA REUNION
##                                3922                                 3373
##                        ILE_DE_FRANCE                CENTRE-VAL-DE-LOIRE
##                                51075                                10264
##             BOURGOGNE_FRANCHE_COMTE                            NORMANDIE
##                                11419                                14720
##                  NORD-PAS-DE-CALAIS   ALSACE-CHAMPAGNE-ARDENNE-LORRAINE
##                                27947                                26763
##                    PAYS-DE-LA-LOIRE                            BRETAGNE
##                                16547                                15230
## AQUITAINE-LIMOUSIN-POITOU-CHARENTES LANGUEDOC-ROUISSILLON-MIDI-PYRENEES
##                                30692                                31009
##                AUVERGNE-RHONE-ALPES          PROVENDE-ALPES-COTE-D-AZUR
##                                38545                                31985
```

Logical (variable type)

Indicate if a condition is TRUE or FALSE as the result of a logical expression.

```
str(ccam20$dms_globale > 7)
```

```
##  logi [1:313491] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
table(ccam20$dms_globale > 7)
```

```
##
##  FALSE   TRUE
## 277524  35967
```

Logical operators

- **&** for AND
- | for OR
- **!** for NOT

```
nrow(ccam20)
```

```
## [1] 313491
```

```
dms7up_bzh <- ccam20[ccam20$dms_globale > 7 & ccam20$reg == "BRETAGNE", ]
nrow(dms7up_bzh)
```

```
## [1] 1808
```

- Logical operators return logical
- Ensemblist operation allows filtering or selection of subset

Missing values

- **NA** when an element is missing i.e. “not available” or “not attributed”
- The function *is.na(x)* gives a logical vector of the same size as *x* with value TRUE if and only if the corresponding element in *x* is NA.

```
x <- c(NA, 3, 5, NA,10)
is.na(x)
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE
```

```
sum(is.na(x))
```

```
## [1] 2
```

To replace *NA* by a given value

```
x[is.na(x)] <- 0
x
```

```
## [1]  0  3  5  0 10
```

Writing output data files

```
ccam20small <- ccam20[, c(1,3,6)]
write.table(ccam20small,
            file = "ccam20small.csv", sep=";")
```

Save R data object as *.Rdata

```
# save preprocess (clean) data for next time (always keep original somewhere)
save(ccam20small, file= "ccam20small.Rdata")
#---------------------------
# On my next working session,  load *.Rdata in R
load("ccam20small.Rdata")
```

Read Cookbook : Search Help pages

Searching for help within R:

```
apropos("mean")
?mean
help(mean)
example(mean)
help.search("mean")
help.start()
```

Searching for help on the Internet (my short list):

https://www.rstudio.com/

http://www.R-project.org

https://www.r-bloggers.com/

https://stackoverflow.com/

Reference books:

https://bookdown.org/

https://epirhandbook.com/en/index.html

Statistical summary with R

Nolwenn Le Meur

May-June 2023

Outline

- Data profiling and data management
- Create new variables
- Summary functions
- Basic graphs

Data profiling & Data management

PMSI example

```
load("../data/pmsi1112.Rdata")
summary(pmsi)
##    key1PMSI         Num_Anonyme        Num_de_sejour   FINESS_PMSI
##  Length:3831        Length:3831        Min.   :21958   Length:3831
##  Class :character   Class :character   1st Qu.:23218   Class :character
##  Mode  :character   Mode  :character   Median :24334   Mode  :character
##                                        Mean   :24270
##                                        3rd Qu.:25364
##                                        Max.   :26650
##
##       RSA          Mois_Sortie          CMD_Obtenu     GHM_Obtenu
##  Min.   :      4   Length:3831        Min.   : 1.00   Length:3831
##  1st Qu.:   4688   Class :character   1st Qu.: 8.00   Class :character
##  Median :  12071   Mode  :character   Median : 8.00   Mode  :character
##  Mean   :  37256                      Mean   :13.04
##  3rd Qu.:  29735                      3rd Qu.: 8.00
##  Max.   :1200382                      Max.   :90.00
##
##    Nb_de_RUM          Age              Sexe       Mode_Entree
##  Min.   :1.000   Min.   : 30.00   Min.   :1.000   Length:3831
##  1st Qu.:1.000   1st Qu.: 65.00   1st Qu.:1.000   Class :character
##  Median :1.000   Median : 74.00   Median :2.000   Mode  :character
##  Mean   :1.163   Mean   : 72.35   Mean   :1.616
##  3rd Qu.:1.000   3rd Qu.: 80.00   3rd Qu.:2.000
##  Max.   :5.000   Max.   :104.00   Max.   :2.000
##
##   Provenance         Annee_Sortie  Mode_Sortie        Destination
##  Length:3831        Min.   :2011   Length:3831        Length:3831
##  Class :character   1st Qu.:2011   Class :character   Class :character
##  Mode  :character   Median :2011   Mode  :character   Mode  :character
##                     Mean   :2011
##                     3rd Qu.:2012
##                     Max.   :2012
```

Data profiling

Process of examining, analysing, reviewing and summarizing data sets to gain insight into the quality of data

Becareful of numeric value for categorical variables !

```
summary(pmsi[, c("Age", "Sexe", "Nb_Actes","Duree_Sejour")])
##       Age              Sexe          Nb_Actes        Duree_Sejour
##  Min.   : 30.00   Min.   :1.000   Min.   :  0.000   Min.   :  0.000
##  1st Qu.: 65.00   1st Qu.:1.000   1st Qu.:  2.000   1st Qu.:  1.000
##  Median : 74.00   Median :2.000   Median :  4.000   Median :  7.000
##  Mean   : 72.35   Mean   :1.616   Mean   :  5.236   Mean   :  7.665
##  3rd Qu.: 80.00   3rd Qu.:2.000   3rd Qu.:  7.000   3rd Qu.:  9.000
##  Max.   :104.00   Max.   :2.000   Max.   :183.000   Max.   :364.000
```

Data profiling

Specific R libraries (DataExplorer, skmir, visdat ...)

```
library(skimr)
skim(pmsi)
```

Putatunda et al., (2019). Journal of Open Source Software, 4(41), 1509, https://doi.org/10.21105/joss.01509

Proper summary needs correct data type

```
summary(pmsi[, c("Age", "Sexe", "Nb_Actes","Duree_Sejour")])
##       Age              Sexe          Nb_Actes        Duree_Sejour
##  Min.   : 30.00   Min.   :1.000   Min.   :  0.000   Min.   :  0.000
##  1st Qu.: 65.00   1st Qu.:1.000   1st Qu.:  2.000   1st Qu.:  1.000
##  Median : 74.00   Median :2.000   Median :  4.000   Median :  7.000
##  Mean   : 72.35   Mean   :1.616   Mean   :  5.236   Mean   :  7.665
##  3rd Qu.: 80.00   3rd Qu.:2.000   3rd Qu.:  7.000   3rd Qu.:  9.000
##  Max.   :104.00   Max.   :2.000   Max.   :183.000   Max.   :364.000
```

```
pmsi$Sexe <- factor(pmsi$Sexe, levels=c(1, 2),
                           labels=c("Male","Female"))
summary(pmsi[, c("Age", "Sexe", "Nb_Actes","Duree_Sejour")])
##       Age              Sexe         Nb_Actes        Duree_Sejour
##  Min.   : 30.00   Male  :1470   Min.   :  0.000   Min.   :  0.000
##  1st Qu.: 65.00   Female:2361   1st Qu.:  2.000   1st Qu.:  1.000
##  Median : 74.00                 Median :  4.000   Median :  7.000
##  Mean   : 72.35                 Mean   :  5.236   Mean   :  7.665
##  3rd Qu.: 80.00                 3rd Qu.:  7.000   3rd Qu.:  9.000
##  Max.   :104.00                 Max.   :183.000   Max.   :364.000
```

Missing values

- is.na() to be replace with a fixed value
- *Mice* package for imputation

```
table(pmsi$Entree)[c(1:3, 10)]
##
##                  missing from short stay provider      from rehab provider
##                      150                           2                        1
##                 from home
##                      2916
levels(pmsi$Entree)
##  [1] "missing"                   "from short stay provider"
##  [3] "from rehab provider"       "mutation from rehab"
##  [5] "mutation from long stay"   "transfer from short stay"
##  [7] "transfer from rehab"       "transfer from long stay"
##  [9] "transfer from psy"         "from home"
## [11] "from home to ER"           "from medico-sociale"
levels(pmsi$Entree)[1] <- "from home"; table(pmsi$Entree)[c(1:3)]
##
##                 from home from short stay provider      from rehab provider
##                      3066                           2                        1
```

Missing values

- is.na() to be replace with a fixed value
- *mice* package for Imputation Using Multivariate Imputation by Chained Equation

```
library(mice)
# 5 imputation sets with predictive mean matching (pmm)
tempData <- mice(data, m=5, maxit=50, meth='pmm',seed=500)
# fitting linear regression on each set
modelFit1 <- with(tempData, lm(Temp~ Ozone + Solar.R + Wind))
# pool coefficients
summary((pool(modelFit1)))
```

Creating new variables

```r
# Create binary variable as a new vector column
# set all values to 0
pmsi$Ambu <- 0
# replace 0 by 1 when GHM_Obtenu end with "J"
pmsi$Ambu[grep("J$", pmsi$GHM_Obtenu)] <- 1
table(pmsi$Ambu)
##
##    0    1
## 3533  298
```

Transform quantitative variable (I)

Transform into factor

```
# Create age groups
pmsi$Age_grp <- cut(pmsi$Age, seq(30,110,10), include.lowest =TRUE,
                    right=FALSE)
table(pmsi$Age_grp)
##
##   [30,40)   [40,50)   [50,60)   [60,70)   [70,80)   [80,90)  [90,100) [100,110]
##        26       126       355       866      1306      1029       121         2
```

Transform quantitative variable (II)

Transform into logical

```
# Create age groups
pmsi$Old_age <- ifelse(pmsi$Age > 59, TRUE, FALSE)
table(pmsi$Old_age)
##
## FALSE  TRUE
##   507  3324
```

Classical statistical summary

Position parameters

- mean

```
mean(pmsi$Duree_Sejour, na.rm=TRUE)
## [1] 7.6651
```

- median

```
median(pmsi$Duree_Sejour, na.rm=TRUE)
## [1] 7
```

Deviation parameters

- Range

```
range(pmsi$Duree_Sejour, na.rm=TRUE)
## [1]   0 364
IQR(pmsi$Duree_Sejour, na.rm=TRUE)
## [1] 8
```

- Quantiles

```
quantile(pmsi$Duree_Sejour, c(0.05,0.3), na.rm=TRUE)
##  5% 30%
##   0   2
```

- Variance and Standard Deviation

```
var(pmsi$Duree_Sejour, na.rm=TRUE)
sd(pmsi$Duree_Sejour, na.rm=TRUE)
```

Advanced summary (with *tidyverse suite*)

Tidyserve

https://www.tidyverse.org/

An opinionated collection of R packages designed for data science

dplyr and *tidyr* packages

Functions to :

- reshape from wide to long and vice versa
- subset
- summarise
- transform or mutate variables
- merge

Grammar programming with *tidyserve*

The “Piping” syntax

- *dplyr::%>%*
- Passes object on left hand side as first argument (or argument) of function on right hand side.

```
MYDATA %>% filter(THESE) %>% group_by(THAT) %>% summarise(THIS)
```

Wrangling Cheat sheet

Summary by groups

```
library(dplyr)
pmsi %>% group_by(Mode_Entree) %>% summarise("Mean"= mean(Duree_Sejour),
                                             "Median"= median(Duree_Sejour))
```

```
## # A tibble: 5 × 3
##   Mode_Entree  Mean Median
##   <chr>       <dbl>  <dbl>
## 1 0            2.35      0
## 2 6           12.7       9
## 3 7           11.6      10
## 4 8            7.16      7
## 5 X           42.6       0
```

The *group_by()* can have multiple parameters

```
library(dplyr)
pmsi %>% group_by(Mode_Entree, Age_Old, Sexe) %>%
  summarise("Mean"= mean(Duree_Sejour),
            "Median"= median(Duree_Sejour))
```

Filter, group, and summarize

```
pmsi %>% filter(Old_age==TRUE) %>%
  group_by(Mode_Entree) %>%
  summarise("Mean length stay"= mean(Duree_Sejour),
            "Median length stay"= median(Duree_Sejour),
            "n" =length(Duree_Sejour),
            "Women (%)"= sum(Sexe=="Female")/length(Sexe)*100)
## # A tibble: 5 × 5
##   Mode_Entree `Mean length stay` `Median length stay`     n `Women (%)`
##   <chr>                    <dbl>                <dbl> <int>       <dbl>
## 1 0                         2.64                    0    91        60.4
## 2 6                        13.0                     9    20        85
## 3 7                        11.6                    10   113        52.2
## 4 8                         7.33                    7  3049        63.5
## 5 X                        42.6                     0    51        72.5
```

Additional packages with summary function

```
library(Hmisc)
Hmisc::describe(pmsi$Age)
## pmsi$Age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     3831        0       70    0.999    72.35    12.57       52       57
##      .25      .50      .75      .90      .95
##       65       74       80       85       88
##
## lowest :  30  31  32  33  34, highest:  96  97  98  99 104
```

Gmd: Gini's mean difference - mean absolute difference between any pairs of observations.

```
library(psych)
psych::describe(pmsi$Age)
##    vars    n  mean    sd median trimmed   mad min max range  skew kurtosis   se
## X1    1 3831 72.35 11.28     74   73.08 10.38  30 104    74 -0.62     0.33 0.18
```

- Note the use of :: for namespace management

Merging datasets

- Using *merge()* from the *base* library (old fashion way)

```
# combine PMSI data with hospital characteristics
# with hospital Finess as foreign key
pmsi_jur <- merge(pmsi, finess, by.x = "FINESS_PMSI", by.y="FINESS_GEO")
```

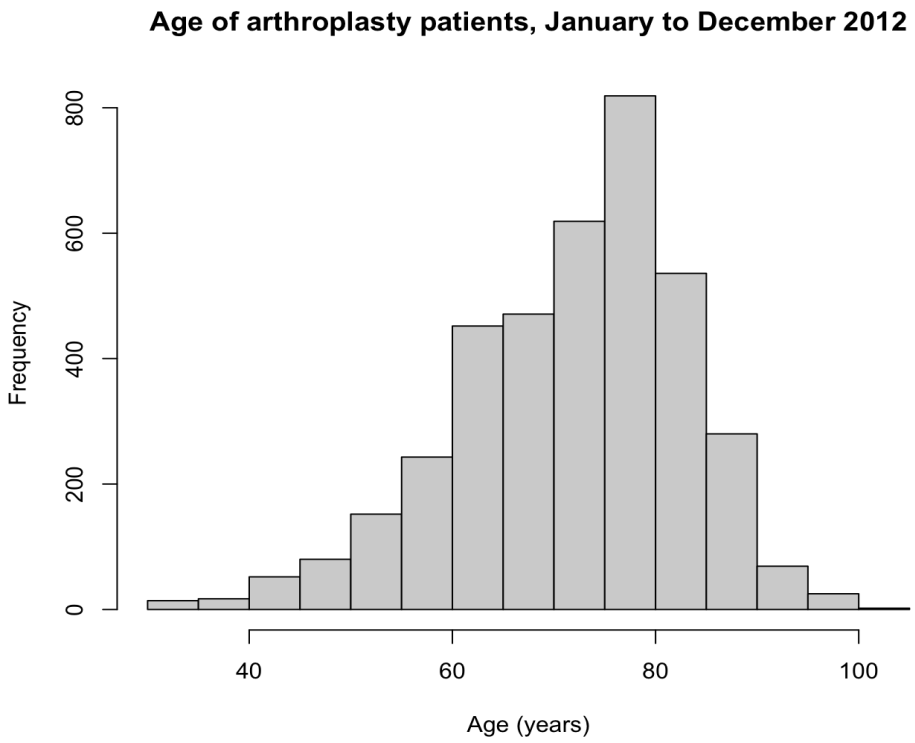
- Using *full_join()* from the *dplyr* library

```
pmsi_jur <- full_join(pmsi, finess, by = c("FINESS_PMSI" = "FINESS_GEO"))
```

Basic graphics

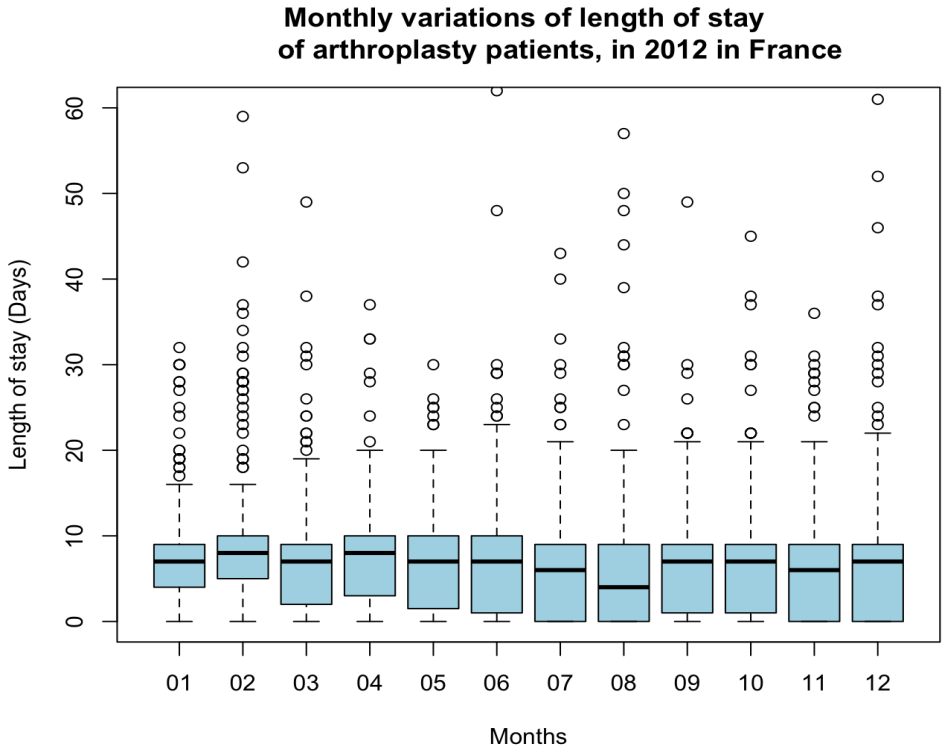
Basic plots: histogram

```
hist(pmsi$Age, xlab="Age (years)",
     main="Age of arthroplasty patients, January to December 2012")
```



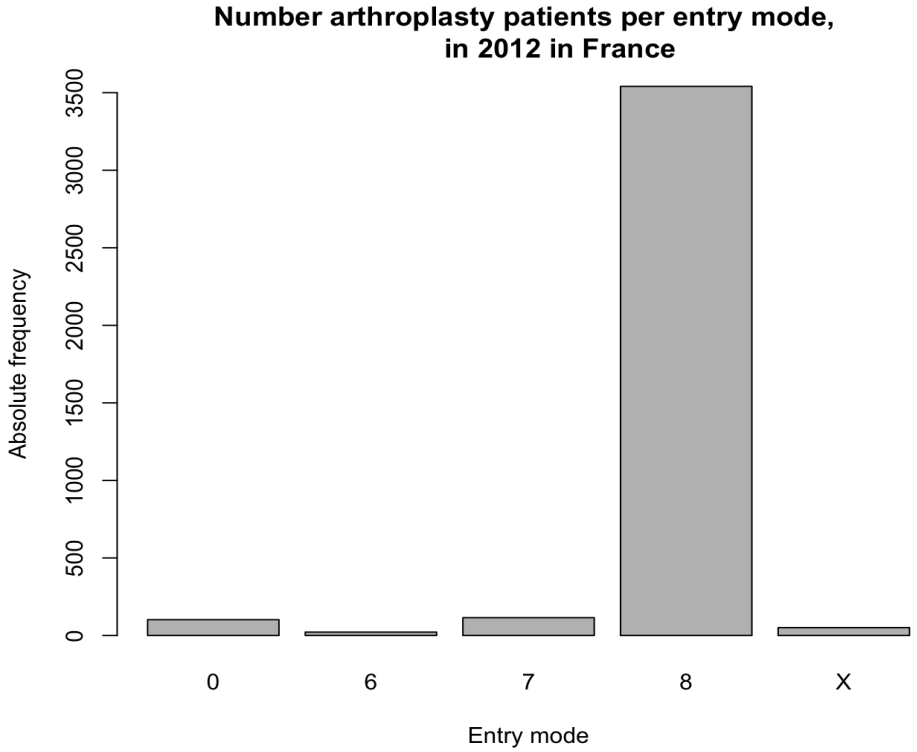
Basic plots: boxplot

```
boxplot(pmsi$Duree_Sejour ~ pmsi$Mois_Sortie, ylim=c(0,60), xlab="Months",
        ylab = "Length of stay (Days)",
        main="Monthly variations of length of stay
        of arthroplasty patients, in 2012 in France",
        col="lightblue")
```



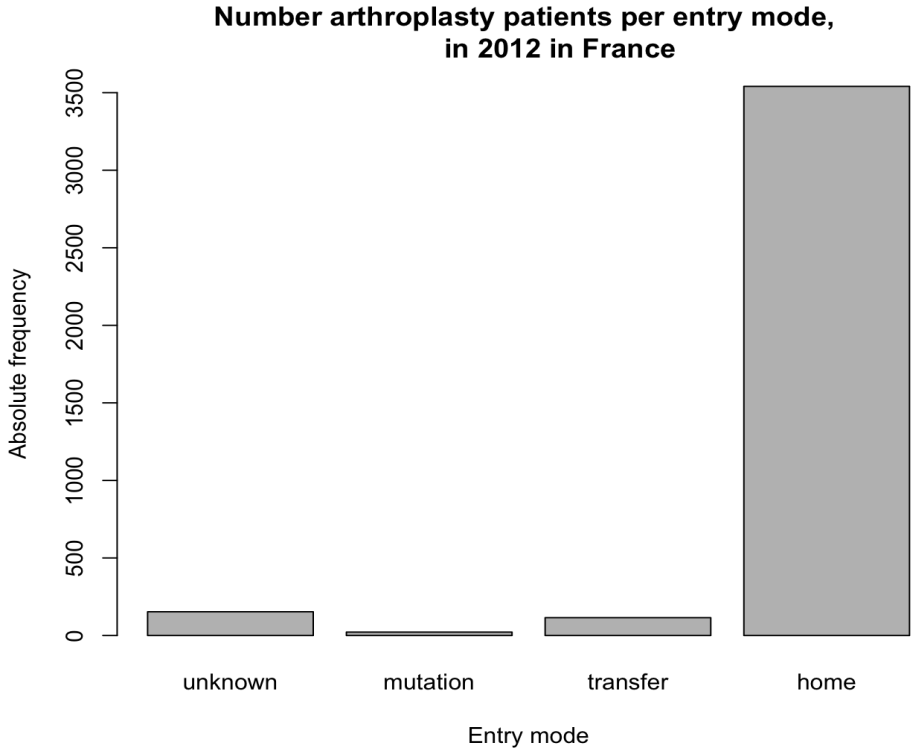
Basic plots: barplot

```
x <- table(pmsi$Mode_Entree)
barplot(x, xlab="Entry mode", ylab="Absolute frequency",
        main="Number arthroplasty patients per entry mode,
        in 2012 in France")
```



Basic plots: barplot

```
pmsi$Mode_Entree <- factor(pmsi$Mode_Entree, labels=c("unknown", "mutation",
                                                      "transfer", "home", "unknown"))
x <- table(pmsi$Mode_Entree)
barplot(x, xlab="Entry mode", ylab="Absolute frequency",
        main="Number arthroplasty patients per entry mode,
        in 2012 in France")
```



Probability distributions and Statistical tests

Nolwenn Le Meur

May-June, 2023

Common distributions

Code	Distribution	Parameters	Defaults
`beta`	beta	shape1, shape2	–, –
`binom`	binomial	size, prob	–, –
`cauchy`	Cauchy	location, scale	0, 1
`chisq`	chi squared	df, ncp	-, 1
`exp`	exponential	rate	1
`f`	F	df1, df2	–, –
`gamma`	gamma	shape, rate, scale	–, 1, 1/rate
`geom`	geometric	prob	–
`hyper`	hyper geometric	m, n, k	–, –, –
`lnorm`	lognormal	meanlog, sdlog	0, 1
`logis`	logistic	location, scale	0, 1
`nbinom`	negative binomial	size, prob, mu	–, –, –
`norm`	normal (Gaussian)	mean, sd	0, 1
`pois`	Poisson	Lambda	1
`t`	Student's t	df, ncp	–,0
`unif`	uniform	min, max	0, 1
`weibull`	Weibull	shape, scale	–, 1
`wilcoxon`	Wilcoxon	m, n	–, –

Common statistical tests

Goal	Measurement (from Gaussian Population)	Rank, Score, or Measurement (from Non- Gaussian Population)	Binomial (Two Possible Outcomes)	Survival Time
Describe one group	Mean, SD	Median, interquartile range	Proportion	Kaplan Meier survival curve
Compare one group to a hypothetical value	One-sample t test	Wilcoxon test	Chi-square or Binomial test	
Compare two unpaired groups	Unpaired t test	Mann-Whitney test	Fisher's test or chi-square	Log-rank test or Mantel-Haenszel
Compare two paired groups	Paired t test	Wilcoxon test	McNemar's test	Conditional proportional hazards regression

Common statistical tests (II)

Goal	Measurement	Rank, Score, or Measurement	Binomial	Survival Time
Compare three or more unmatched groups	One-way ANOVA	Kruskal-Wallis test	Chi-square test regression	Cox proportional hazard
Compare three or more matched groups	Repeated-measures ANOVA	Friedman test	Cochrane Q	Conditional proportional hazards regression
Quantify association between two variables	Pearson correlation	Spearman correlation	Contingency coefficients	
Predict value from another measured variable	Simple linear regression	Nonparametric regression	Simple logistic regression	Cox proportional hazard regression
Predict value from several measured or binomial variables	Multiple linear regression		Multiple logistic regression	Cox proportional hazard regression

Applications

Testing for a certain distribution function

The *ks.test()* function test whether a data vector is drawn from a certain distribution

```
x <- runif(100)
ks.test(x,"pnorm")
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.50117, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Alternative function to test for normality *shapiro.test()*

Compare two unpaired groups

Example : Ranking US hospital based on their management mode

Hospital	Top50	Other	Total	%Top50
Physician	33	18	51	64.7%
Manager	17	32	49	34.7%

H0: proportion of hospital in the top50 directed by physician = proportion of hospital in the top50 directed by manager

H1: proportion of hospital in the top50 directed by physician != proportion of hospital in the top50 directed by manager At α = 5%

Compare two unpaired groups

```r
medecinManager <- matrix(c(33,17,18,32), ncol=2)
medecinManager
##      [,1] [,2]
## [1,]   33   18
## [2,]   17   32
fisher.test(medecinManager)
##
##  Fisher's Exact Test for Count Data
##
## data:  medecinManager
## p-value = 0.00485
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.407342 8.536524
## sample estimates:
## odds ratio
##   3.405563
```

Compare two means

Length of stay comparison between Male and Female

```
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
arthro_genou <- pmsi %>% filter(GHM_Obtenu%in% c("08C241", "08C242"))
arthro_genou  %>% group_by(Sexe) %>%
  summarise("N" = length(Sexe),
            "Mean lenght stay"= mean(Duree_Sejour))
## # A tibble: 2 × 3
##   Sexe       N `Mean lenght stay`
##   <fct>  <int>              <dbl>
## 1 Male     258               8.53
## 2 Female   425               8.88
```

- Parametric or non-parametric test?
- Verification of assumptions

Compare two means : verify equality of variances

```r
var.test(Duree_Sejour~Sexe, data= arthro_genou)
##
##  F test to compare two variances
##
## data:  Duree_Sejour by Sexe
## F = 0.807, num df = 257, denom df = 424, p-value = 0.05888
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6498777 1.0081226
## sample estimates:
## ratio of variances
##          0.8069982
```

Compare two means: parametric test

```r
t1 <- t.test(Duree_Sejour~Sexe, data= arthro_genou)
t1
##
##  Welch Two Sample t-test
##
## data:  Duree_Sejour by Sexe
## t = -1.7501, df = 587.55, p-value = 0.08062
## alternative hypothesis: true difference in means between group Male and group Female is not equal to 0
## 95 percent confidence interval:
##  -0.74886015  0.04312372
## sample estimates:
##   mean in group Male mean in group Female
##             8.527132             8.880000
```

access t.test values

```
names(t1)
##  [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
##  [6] "null.value"  "stderr"      "alternative" "method"      "data.name"
t1$estimate
##   mean in group Male mean in group Female
##             8.527132             8.880000
t1$p.value
## [1] 0.08061819
```

Compare two means: non parametric test

```r
wilcox.test(Duree_Sejour~Sexe, data= arthro_genou)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Duree_Sejour by Sexe
## W = 50378, p-value = 0.07086
## alternative hypothesis: true location shift is not equal to 0
```

Compare 3 means

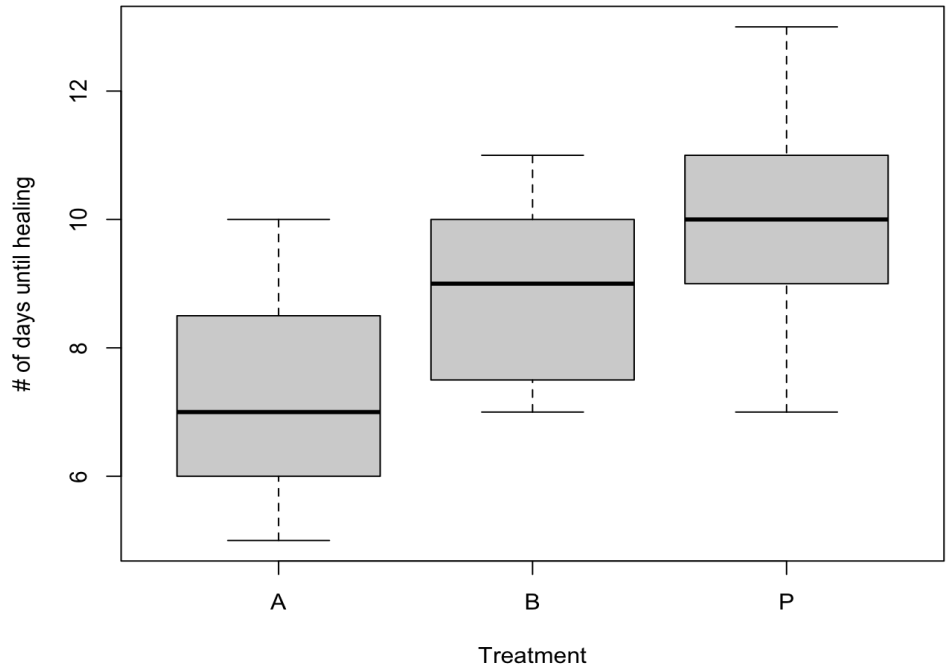
Do the (means of) the quantitative variables depend on the group (given by categorical variable) the individual is in?

Subjects: 25 patients with blisters Treatments: 3 (A, B, Placebo) Measurement: # of days until blisters heal

```
A <- c(5,6,6,7,7,8,9,10)
B <- c(7,7,8,9,9,10,10,11)
P <- c(7,9,9,10,10,10,11,12,13)
## combining data
treat <- data.frame("Day"= c(A,B,P),
                    "Treatment"=c(rep("A", length(A)),
                                  rep("B", length(B)),
                                  rep("P", length(P))))
```

Compare 3 means: visual comparison

```
boxplot(Day~Treatment, data=treat,
        xlab="Treatment", ylab="# of days until healing")
```



Compare 3 means: ANOVA

At its simplest (there are extensions) ANOVA tests the following hypotheses:

Ho = The means of all the groups are equal.

H1 = Not all the means are equal

Does not say how or which one differs. Can follow up with a multiple comparison

Assumptions:

1. Each group is approximately normal (can handle some nonnormality, but not severe outliers) (check this by looking at histograms and/or normal quantile plots)

2. Variances of the groups are approximately equal

Compare 3 means: ANOVA and Tukey's test

```
m <- aov(Day~Treatment, data=treat)
summary(m)
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Treatment    2  34.74  17.368   6.447 0.00626 **
## Residuals   22  59.26   2.694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
TukeyHSD(m)
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Day ~ Treatment, data = treat)
##
## $Treatment
##         diff        lwr      upr     p adj
## B-A 1.625000 -0.4365045 3.686505 0.1407700
## P-A 2.861111  0.8576888 4.864533 0.0044884
## P-B 1.236111 -0.7673112 3.239533 0.2879573
```

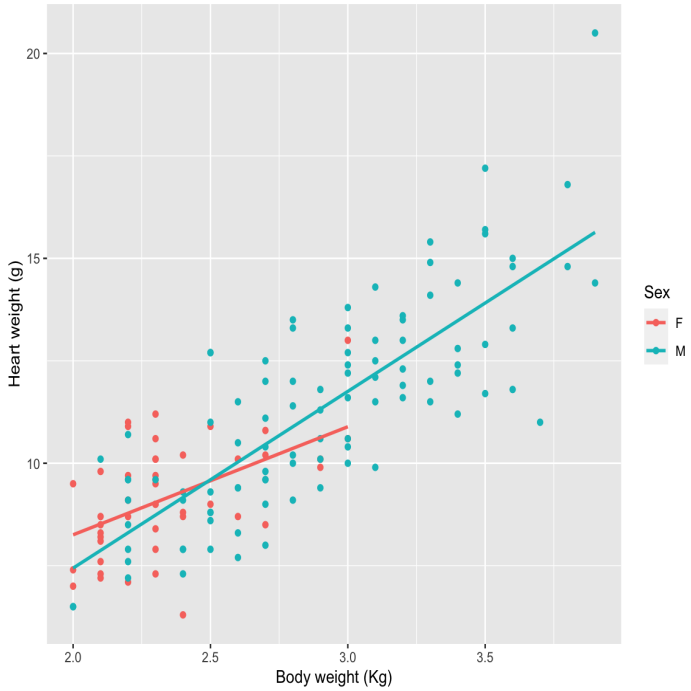
Regression models

Nolwenn Le Meur

May-June 2023

Regression models

Does the explanatory variable x significantly affect the response variable y?



https://stats.idre.ucla.edu/other/dae/

Simple linear regression: Example

Here : In cats, is the heart weight associated with (predict) body weight ?

```
library(MASS)
data(cats)
head(cats)
##   Sex Bwt Hwt
## 1   F 2.0 7.0
## 2   F 2.0 7.4
## 3   F 2.0 9.5
## 4   F 2.1 7.2
## 5   F 2.1 7.3
## 6   F 2.1 7.6
```

Simple linear regression: Example

```
# simple linear regression
lmB <- lm(Hwt~Bwt, data=cats)
summary(lmB)
##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5694 -0.9634 -0.0921  1.0426  5.1238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3567     0.6923  -0.515    0.607
## Bwt           4.0341     0.2503  16.119   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.452 on 142 degrees of freedom
## Multiple R-squared:  0.6466, Adjusted R-squared:  0.6441
## F-statistic: 259.8 on 1 and 142 DF,  p-value: < 2.2e-16
```

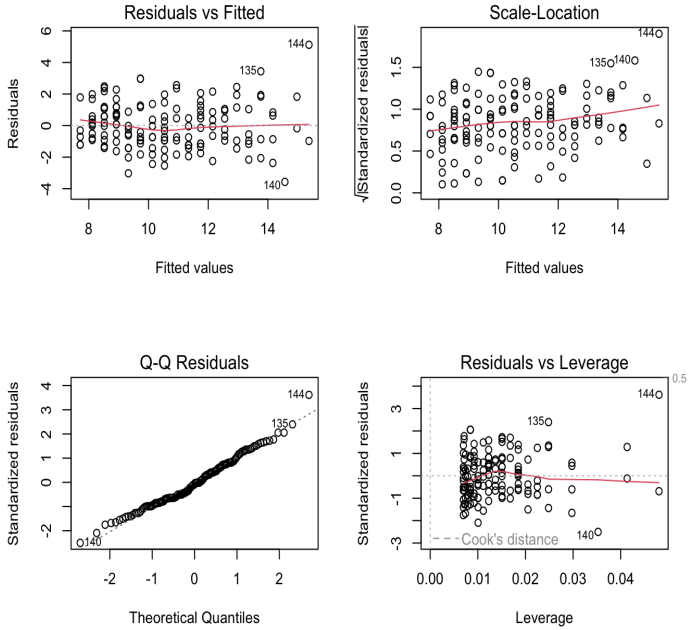
Residual plots

To verify assumptions and quality of the model

1. linearity and additivity of the predict relationship
2. homoscedasticity (constant variance) of errors and independence (lack of correlation) of errors (in particular, no correlation between consecutive errors in the case of time series data)
3. normality of the error distribution
4. checking for outliers

Diagnostic plots

```
layout(matrix(c(1:4), nrow=2))
plot(lmB)
```




```
dev.off()
## null device
##           1
```

Output of a linear regression

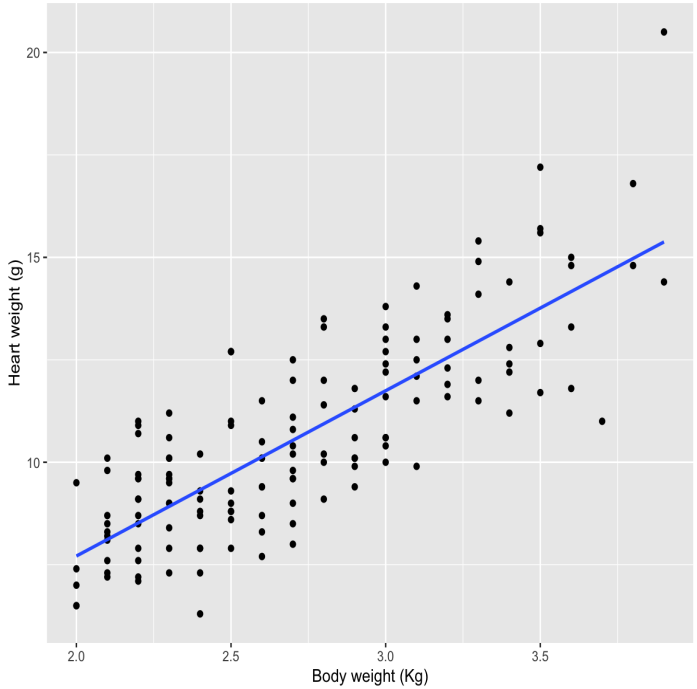
- Printed output of the model fit is minimal
- Value of a fitted model object is stored in an object (list format)
- Information about the fitted model can be displayed, extracted and plotted using indexing or function calls
- the `broom` package for optimized, standardized display of models

The functions are:

- coef(obj) - regression coefficients
- resid(obj) - residuals
- fitted(obj) - fitted values
- summary(obj) - analysis summary
- predict(obj,newdata = ndat) - predict for new data
- deviance(obj) - residual sum of squares
- plot(obj) - produce diagnostic plots
- formula(obj) - extract the model formula

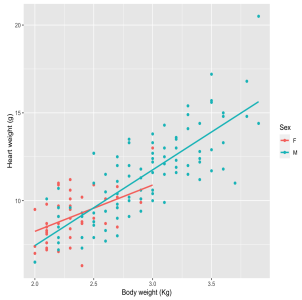
Heart weight and body weight

```
lmB <- lm(Hwt ~ Bwt, data=cats)
library(ggplot2)
ggplot(cats, aes(x=Bwt, y=Hwt)) +
  geom_point() + stat_smooth(method = "lm",se=F) +
  xlab("Body weight (Kg)") + ylab("Heart weight (g)")
## `geom_smooth()` using formula = 'y ~ x'
```



Heart weight and body weight and sex

```
lmBxS <- lm(Hwt~Bwt*Sex, data=cats)
ggplot(cats, aes(x=Bwt, y=Hwt, strata=Sex,color=Sex)) +
  geom_point() + stat_smooth(method = "lm",  se=F)+
  xlab("Body weight (Kg)") + ylab("Heart weight (g)")
## `geom_smooth()` using formula = 'y ~ x'
```



Regression for prediction

What is the predicted heart weight with for the new male cats?

```
newCats <- data.frame(Bwt=seq(2,5,0.5), Sex="M")
newCats
##   Bwt Sex
## 1 2.0   M
## 2 2.5   M
## 3 3.0   M
## 4 3.5   M
## 5 4.0   M
## 6 4.5   M
## 7 5.0   M
predict(lmBxS, newCats)
##         1         2         3         4         5         6         7
##  7.441270  9.597609 11.753948 13.910288 16.066627 18.222966 20.379306
```

Step procedures for variable selection

```
hwt.all <- lm(Hwt~ ., data=cats)
hwt.best <- step(hwt.all, direction = "backward")
## Start:  AIC=111.39
## Hwt ~ Sex + Bwt
##
##        Df Sum of Sq    RSS    AIC
## - Sex   1      0.15 299.53 109.47
## <none>              299.38 111.39
## - Bwt   1    405.88 705.26 232.78
##
## Step:  AIC=109.47
## Hwt ~ Bwt
##
##        Df Sum of Sq    RSS    AIC
## <none>              299.53 109.47
## - Bwt   1    548.09 847.63 257.26
```

Generalized linear modeling

The R function to fit a generalized linear model is `glm()`

```
fitted.model <- glm(formula, family=family.generator(distribution.link),
data=data.frame)
```

Family generator and distribution link:

- Binomial: logit, probit, log, cloglog
- Gaussian: identity, log, inverse
- Gamma: identity, inverse, log
- inverse.gaussian: $1/\mathrm{mu}^2$, identity, inverse, log
- Poisson: identity, log, sqrt

Simple logistic regression

Records of high risk pregnant women under a trial on new and old methods of antenatal care in two clinics. The outcome was perinatal mortality.

```
library(epiDisplay)
data(ANCdata)
head(ANCdata)
##   death anc clinic
## 1    no old      A
## 2    no old      A
## 3    no old      A
## 4    no old      A
## 5    no old      A
## 6    no old      A
```

Simple logistic regression (Ctd)

```
glm1 <- glm(death ~ anc, family=binomial(), data=ANCdata)
summary(glm1)
##
## Call:
## glm(formula = death ~ anc, family = binomial(), data = ANCdata)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.0929     0.1563 -13.393   <2e-16 ***
## ancnew       -0.6671     0.2785  -2.395   0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 447.75  on 754  degrees of freedom
## Residual deviance: 441.64  on 753  degrees of freedom
## AIC: 445.64
##
## Number of Fisher Scoring iterations: 5
```

Simple logistic regression (Ctd)

Nice output with *epiDisplay*

```
logistic.display(glm1)
##
## Logistic regression predicting death : yes vs no
##
##                                       OR(95%CI)        P(Wald's test)
## type of antenatal service: new vs old 0.51 (0.3,0.89)  0.017
##
##                                       P(LR-test)
## type of antenatal service: new vs old 0.013
##
## Log-likelihood = -220.8218
## No. of observations = 755
## AIC value = 445.6437
```

Multiple logistic regression

1. univariate test

```
da = fisher.test(ANCdata$death, ANCdata$anc); da$p.value
## [1] 0.01906334
dc = fisher.test(ANCdata$death, ANCdata$clinic); dc$p.value
## [1] 6.010879e-05
```

1. multivariate model

```
glm2 <- glm(death ~ anc+clinic, family=binomial, data=ANCdata)
logistic.display(glm2, simplified = T)
##
##                OR lower95ci upper95ci    Pr(>|Z|)
## ancnew  0.8604452 0.4504932  1.643457 0.648932163
## clinicB 2.6812398 1.4634387  4.912434 0.001410115
```

Model quality: Likelihood ratio test

Model 1: death ~ anc
Model 2: death ~ anc + clinic

H0 quality of fitting is identical in the 2 models
(adding *clinic* does not change the quality of the model)
H1 quality of fitting is different in the 2 models
(adding *clinic* significantly improves the quality of the model)

```
logLik(glm1) ; logLik(glm2)
## 'log Lik.' -220.8218 (df=2)
## 'log Lik.' -215.4681 (df=3)

library(epiDisplay)
lrtest(glm1, glm2)
## Likelihood ratio test for MLE method
## Chi-squared 1 d.f. =  10.70745 , P value =  0.00106705
```

Model quality: AIC and BIC criterion

Adjusted log likelihood criterion ==> the smaller the better

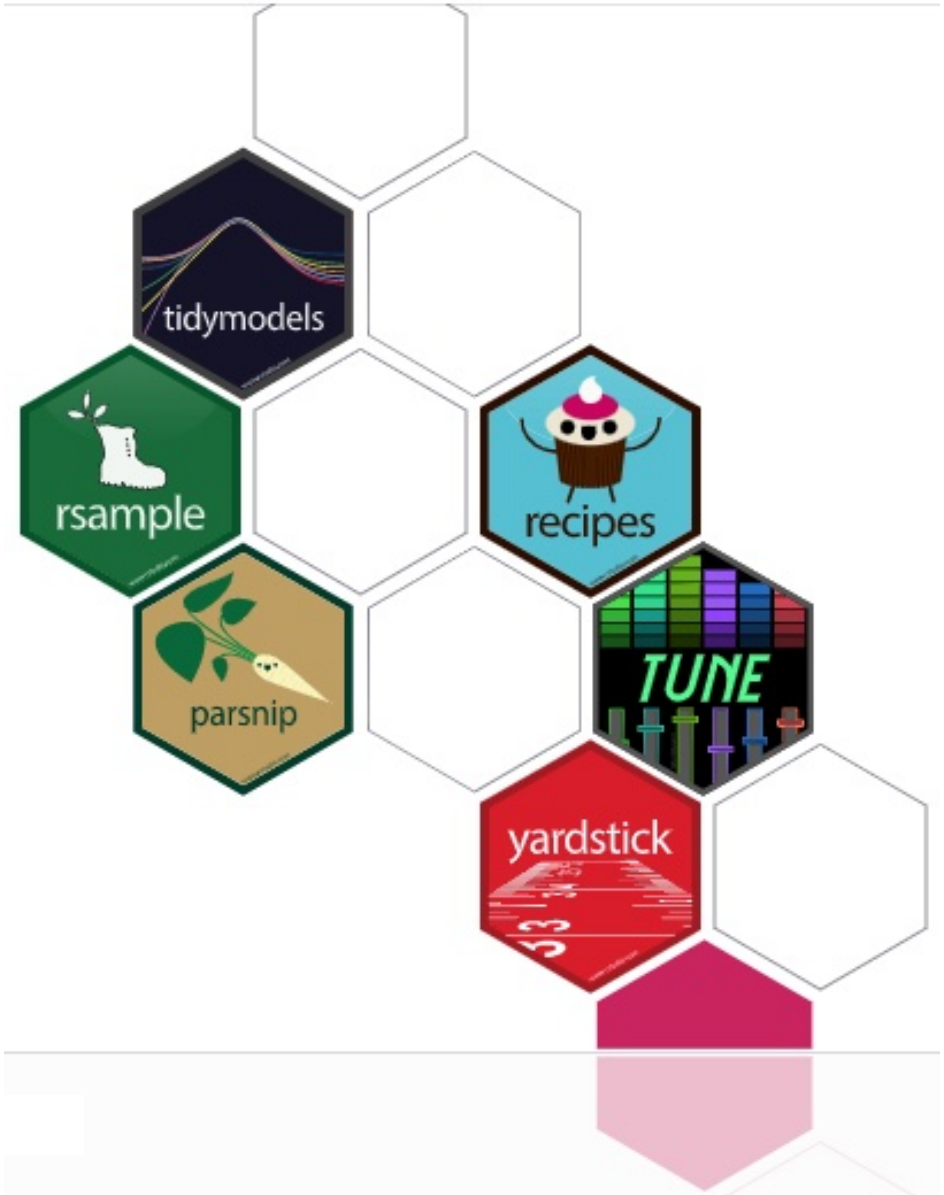
```
AIC(glm1, glm2)
##      df      AIC
## glm1  2 445.6437
## glm2  3 436.9362
BIC(glm1, glm2)
##      df      BIC
## glm1  2 454.8971
## glm2  3 450.8164
```

AIC: best model for prediction (constant weight penalization)

BIC: best model for risk factors selection (sample sized penalization)

tidymodel etc

The tidymodels framework is a collection of packages for modeling and machine learning using tidyverse principles. https://www.tidymodels.org/start/models/



Graphics

Nolwenn Le Meur

May-June 2023

Outline

- Base graphics system
- `ggplot2` library

Graphics device

R initiates a graphics device driver which opens a special graphics window for the display of interactive graphics.

- X11()
- quartz() for MacOS

Base graphics system

Default plotting functions

Plotting commands divided into three basic groups

1. High-level plotting functions create a new plot on the graphics device, possibly with axes, labels, titles and so on.

2. Low-level plotting functions add more information to an existing plot, such as extra points, lines and labels.

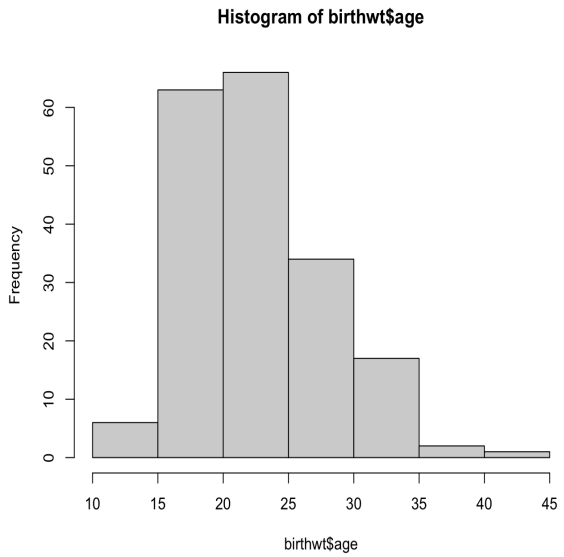
3. Interactive graphics functions allow you to interactively add information to, or extract information from the plots

High-level plotting: histogram

```
library(MASS)
data(birthwt)
```

- Histogram

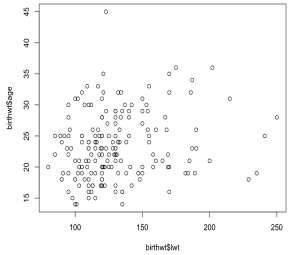
```
hist(birthwt$age)
```



High-level plotting: scatterplot

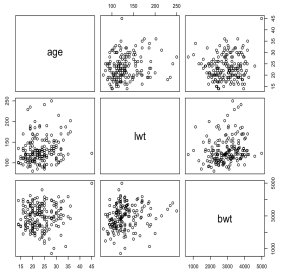
- Scatterplot of two vectors

```
plot(birthwt$lwt, birthwt$age)
```



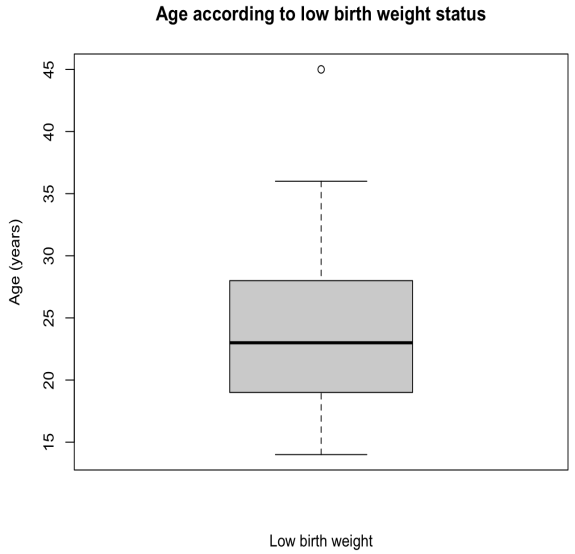
- and more than 2

```
pairs(birthwt[, c("age", "lwt", "bwt")])
```



High-level plotting: boxplot

```
boxplot(age~ low, data=birthwt[birthwt$low==0, ],
        main="Age according to low birth weight status",
        xlab= "Low birth weight", ylab="Age (years)")
```



High-level plotting: arguments example

type= argument controls the type of plot produced, as follows:

- type="p" Plot individual points (the default)
- type="l" Plot lines
- type="b" Plot points connected by lines (both)
- type="o" Plot points overlaid by lines
- type="h" Plot vertical lines from points to the zero axis (high-density)
- type="n" No plotting at all. However axes are still drawn (by default) and the coordinate

High-level plotting: axis and labels arguments

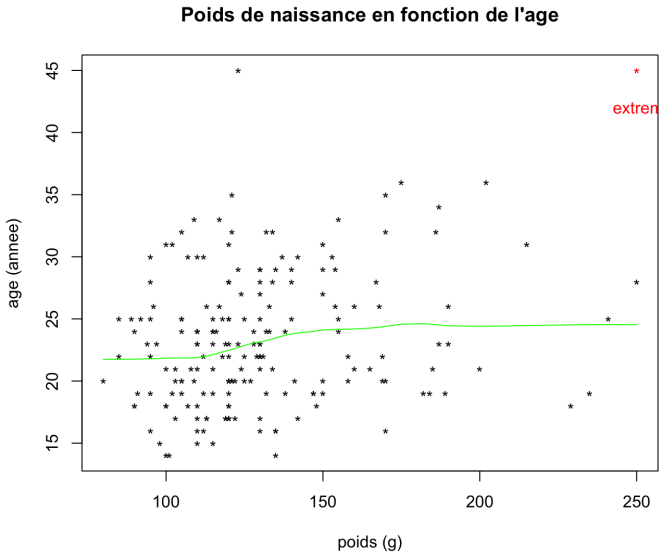
- axes = FALSE : Suppresses generation of axes-useful for adding your own custom axes with the axis() function. The default, axes=TRUE, means include axes.
- xlab = “” ; ylab = “”

Axis labels for the x and y axes. Use these arguments to change the default labels, usually the names of the objects used in the call to the high-level plotting function.

- main = “” : Figure title, placed at the top of the plot in a large font.
- sub = “” : Sub-title, placed just below the x-axis in a smaller font.

Customizable

```
plot(birthwt$lwt, birthwt$age, main = "Poids de naissance en fonction de l'age",
     pch="*", xlab="poids (g)", ylab="age (annee)")
lines(lowess(birthwt$lwt, birthwt$age), col="green")
points(max(birthwt$lwt), max(birthwt$age),  pch="*",col="red")
text(max(birthwt$lwt)+2, max(birthwt$age)-3, "extreme", col="red")
```

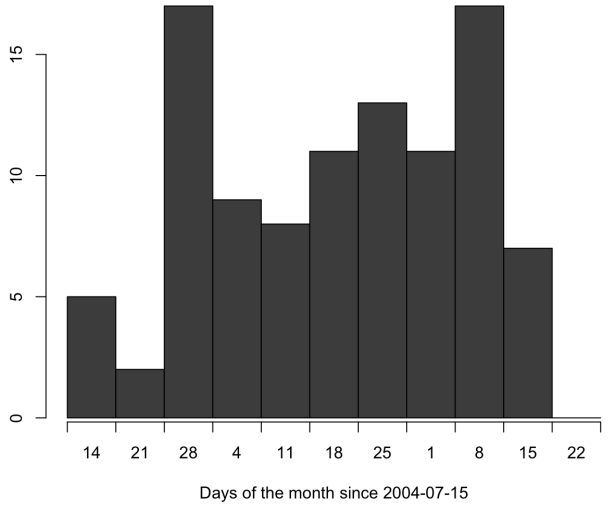


Epidemic curve

- Epidemic curves are histograms with time in x axis
- Time can be in hours, days, weeks, months
- Optimed functions available in `epitools` or `EpiCurve` library

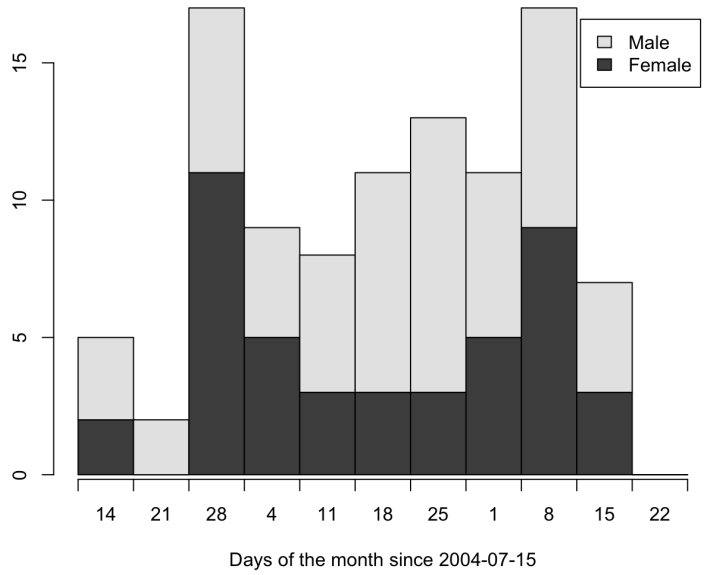
Epidemic curve with *epitools* library

```
library(epitools)
# time period
sampdates <- seq(as.Date("2004-07-15"), as.Date("2004-09-15"), 1)
# sample of 100 dates within the period
x <- sample(sampdates, 100, rep=TRUE)
# epidemic curve
rr<-epicurve.weeks(x, min.date="2004-07-15", axisnames = FALSE,
                   xlab="Days of the month since 2004-07-15")
axis(1, at = rr$xvals, labels = rr$cmday, tick = FALSE, line = 0)
```



Epidemic curve with strata

```
# sample gender data of the same length
xs <- sample(c("Male","Female"), 100, rep=TRUE)
# epidemic curve with strata
epicurve.weeks(x,  min.date="2004-07-15", strata = xs,
               axisnames = FALSE,
               xlab="Days of the month since 2004-07-15", legend=TRUE)
axis(1, at = rr$xvals, labels = rr$cmday, tick = FALSE, line = 0)
```

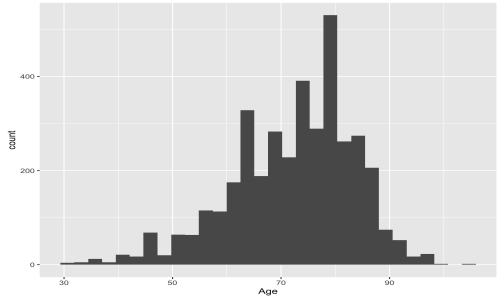


ggplot2 library

ggplot2 : like a grammar of graphics

ggplot2 developed by Hadley Wickham *like a grammar of graphics*

```
library(ggplot2)
b <- ggplot(pmsi, aes(Age))
b + geom_histogram()
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The ggplot2 grammar

- data : the variable to display
- aes : aesthetics mapping, i.e. the dimensions along which the data are displayed
- geom: geometries, i.e. shape to represent the data
- facets : array (row et column) of graphs
- statistics : model or data transformation used to represent/summaryze the data
- coordinates : space (horizontal, vertical, cartasian, polar)
- scales : scales of the axes (linear, logarithm, inverse), filling colors
- themes : background

Link: See ggplot2 cheat sheet for basic use

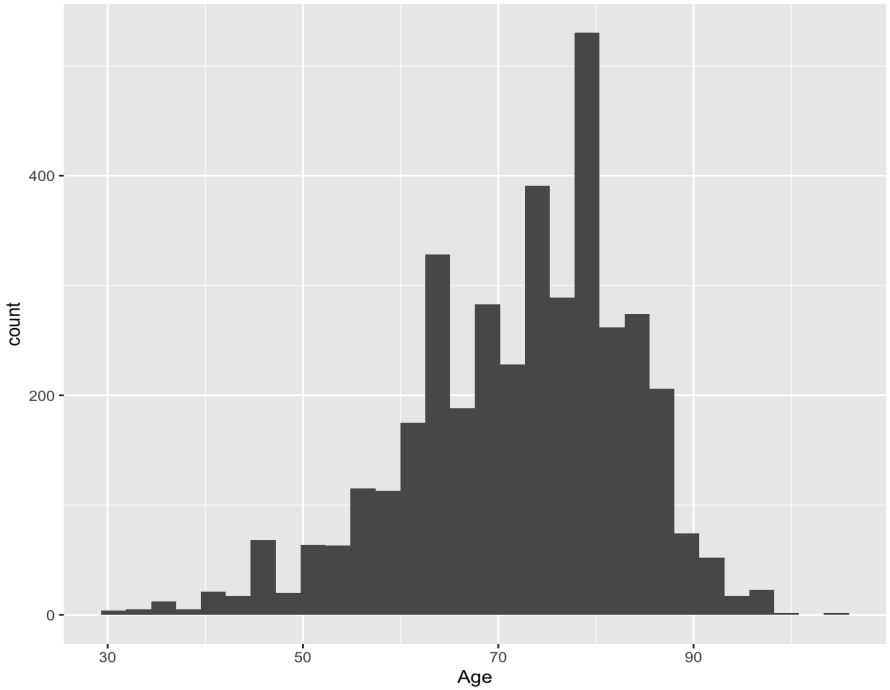
Geometries

`geom_...()` define the graph type, for instance

- basic : `geom_point()`, `geom_line()`, `geom_polygon()`, `geom_bar()`, `geom_text()`
- statistical object : `geom_histogram()`, `geom_boxplot()`, `geom_smooth()`, `geom_density()`, `geom_quantile()`
- vertical intervals : `geom_pointrange()`, `geom_errorbar()`

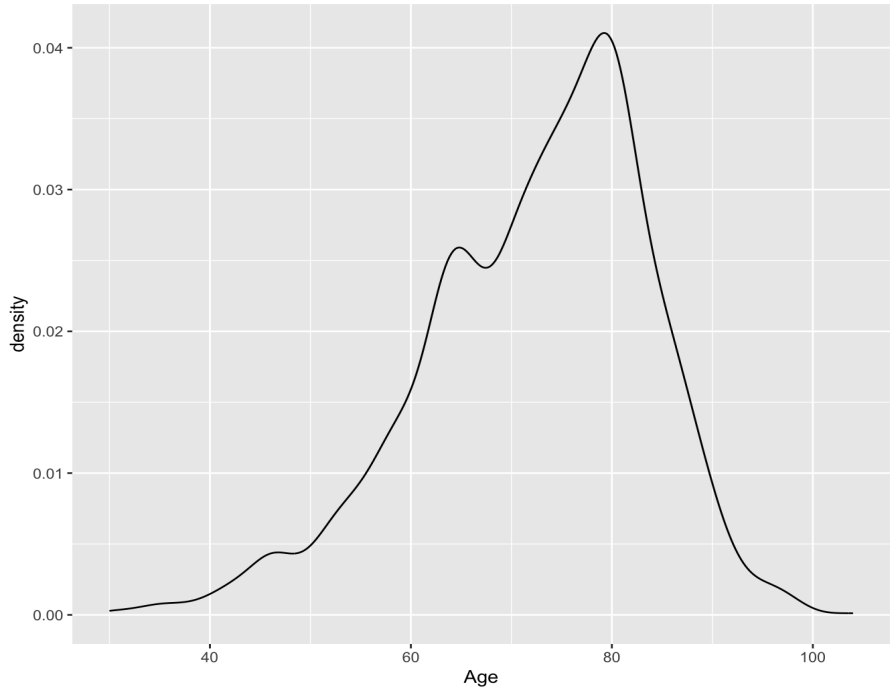
Quantitative variable : histogram

```
ageplot <- ggplot(pmsi, aes(Age))
ageplot + geom_histogram()
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



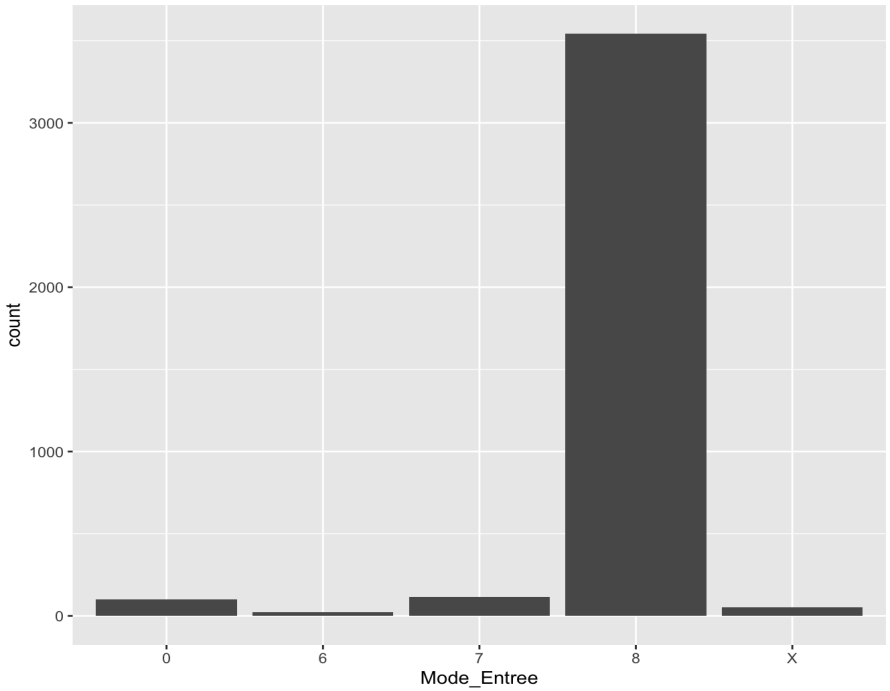
Quantitative variable : density

```
ageplot + geom_density()
```



Qualitative variable

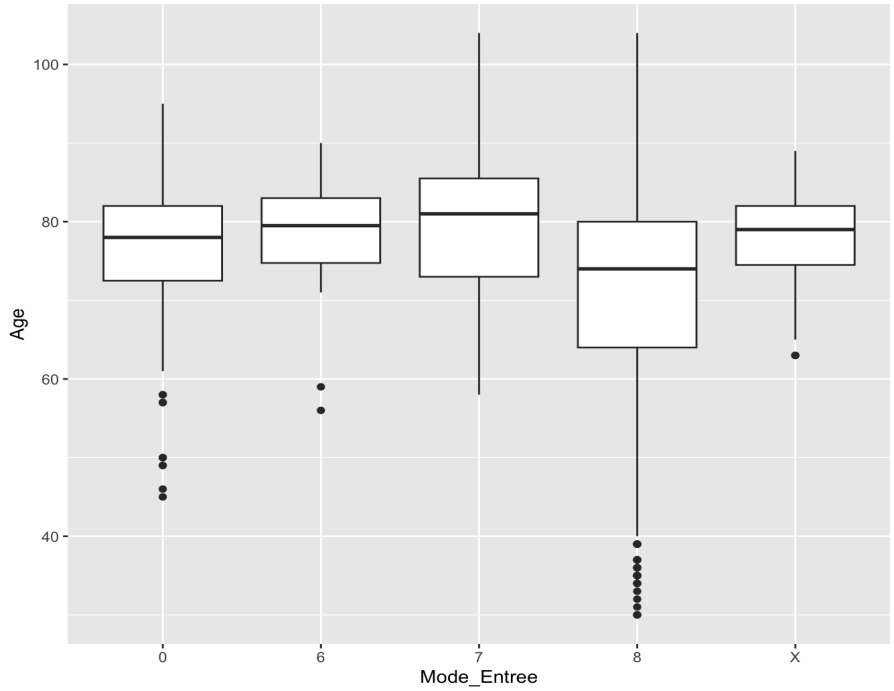
```
ggplot(pmsi, aes(Mode_Entree)) + geom_bar()
```



Bivariate plot

Quanti * Quali

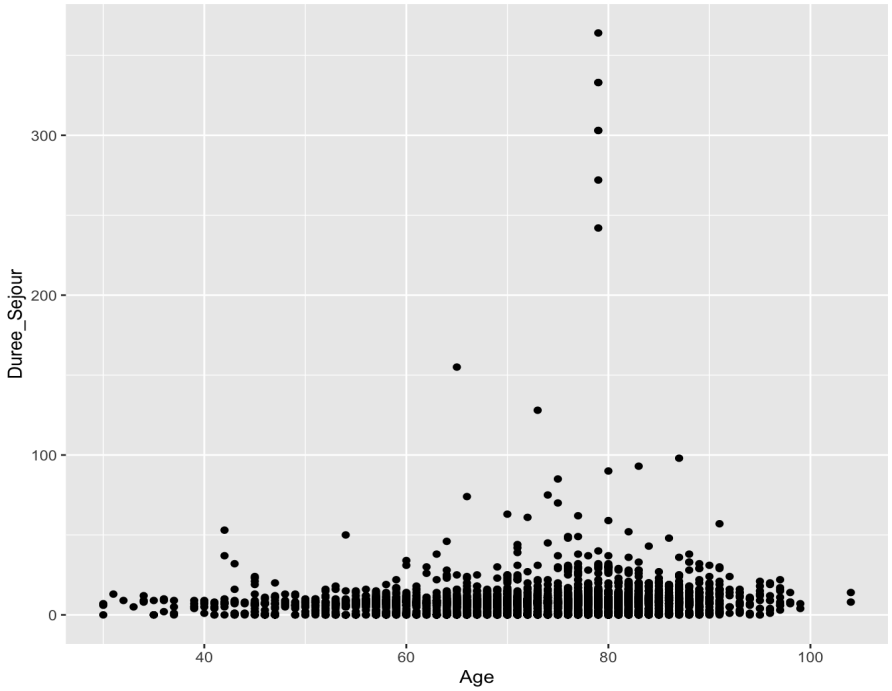
```
ggplot(pmsi, aes(x=Mode_Entree, y=Age)) + geom_boxplot()
```



Bivariate plot

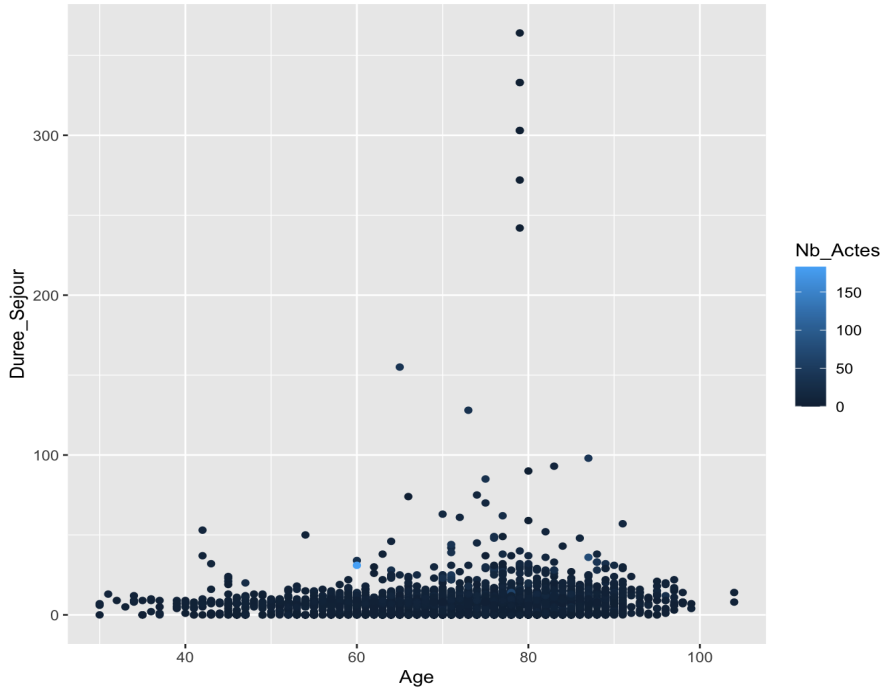
Quanti * Quanti

```
ggplot(pmsi, aes(x=Age, y=Duree_Sejour)) + geom_point()
```



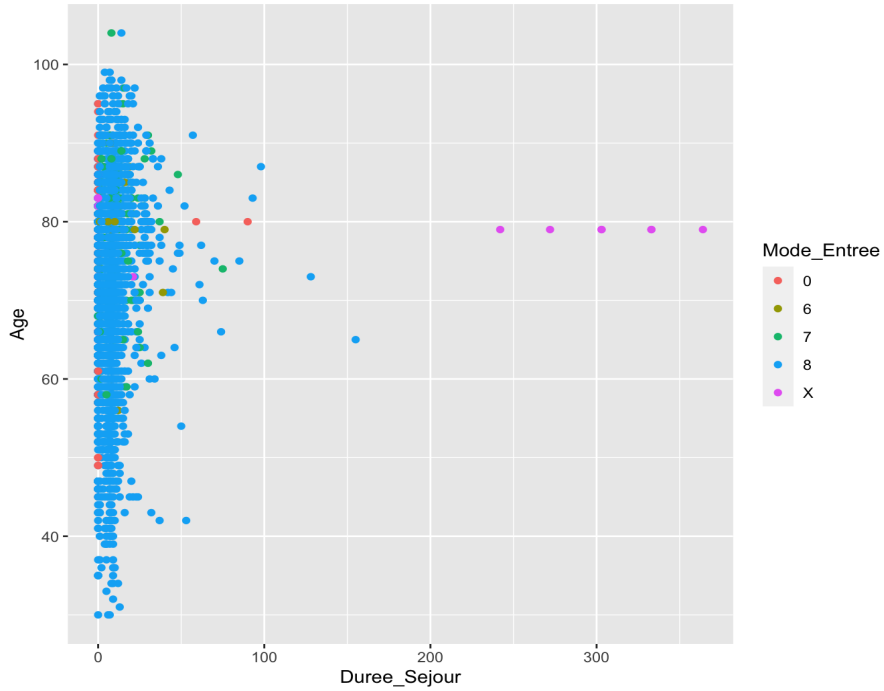
Scatter plot with color function of a third quantitative variable

```
ggplot(pmsi, aes(x=Age, y=Duree_Sejour)) +
  geom_point(aes(colour = Nb_Actes))
```



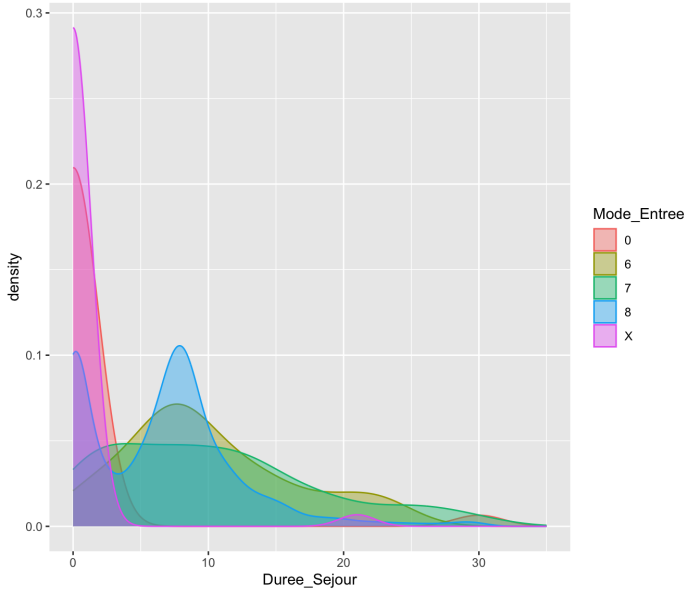
Scatter plot with color function of a thrid qualitative variable

```
ggplot(pmsi, aes(x=Duree_Sejour, y=Age)) +
  geom_point(aes(colour = Mode_Entree))
```



Scatter plot with color function of a thrid quantitative variable

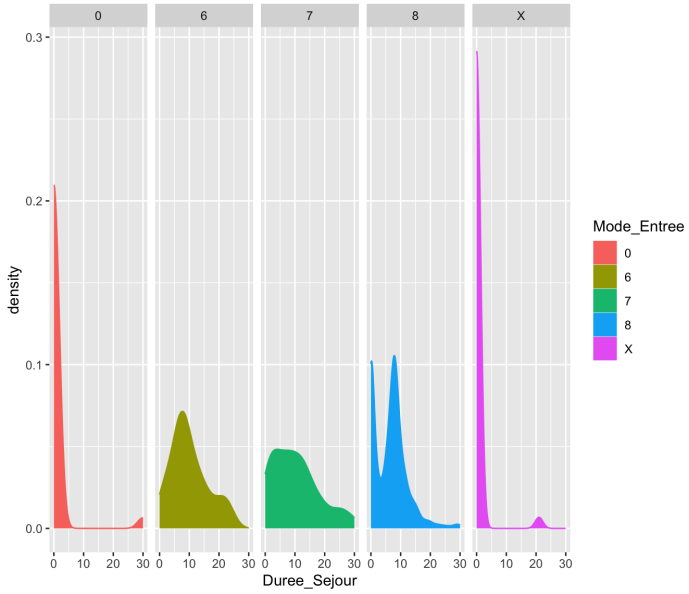
```
library(dplyr)
pmsi30<- pmsi %>% filter(Duree_Sejour<=30)
ggplot(pmsi30,
       aes(x = Duree_Sejour,
           color = Mode_Entree, fill=Mode_Entree)) +
  geom_density(alpha=0.4) + xlim(0, 35)
```



Facet grid

Multi-panel display

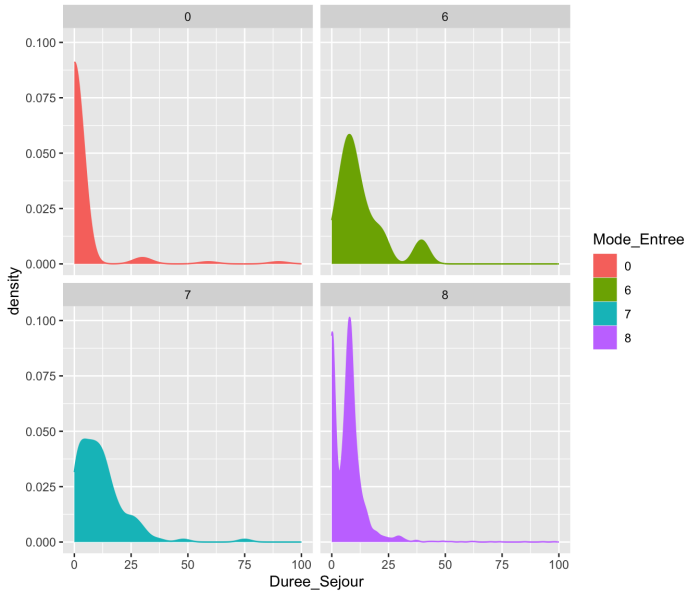
```
p <- ggplot(pmsi, aes(x = Duree_Sejour, color = Mode_Entree,
                      fill=Mode_Entree)) +
  geom_density() + xlim(0, 30)
p + facet_grid(. ~ Mode_Entree)
## Warning: Removed 60 rows containing non-finite values (`stat_density()`).
```



Facet wrap

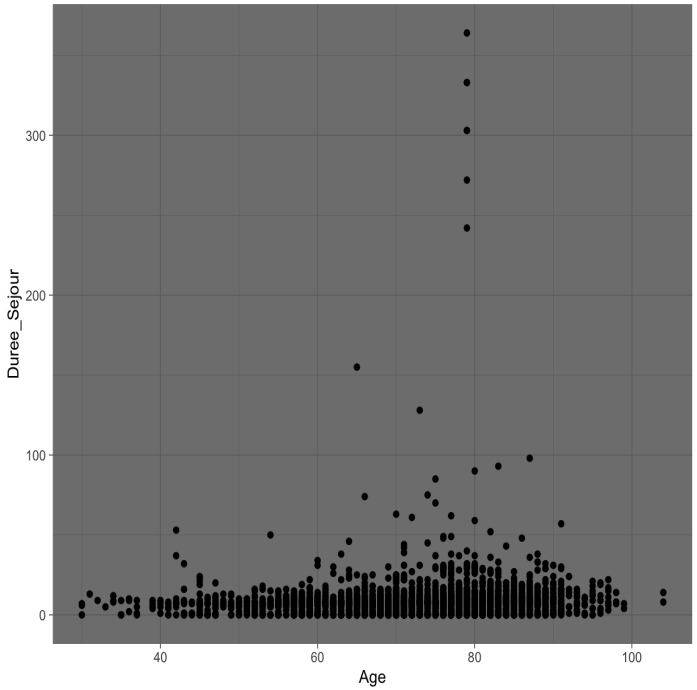
Multi-panel display

```
pmsi2 = subset(pmsi, pmsi$Mode_Entree!="X")
p <- ggplot(pmsi2, aes(x = Duree_Sejour, color = Mode_Entree,
                       fill=Mode_Entree)) +
  geom_density() + xlim(0, 100)
p + facet_wrap(. ~ Mode_Entree)
## Warning: Removed 2 rows containing non-finite values (`stat_density()`).
```



Themes (background)

```
ggplot(pmsi, aes(x = Age, y = Duree_Sejour)) +
  geom_point() + theme_dark()
```



Parameters

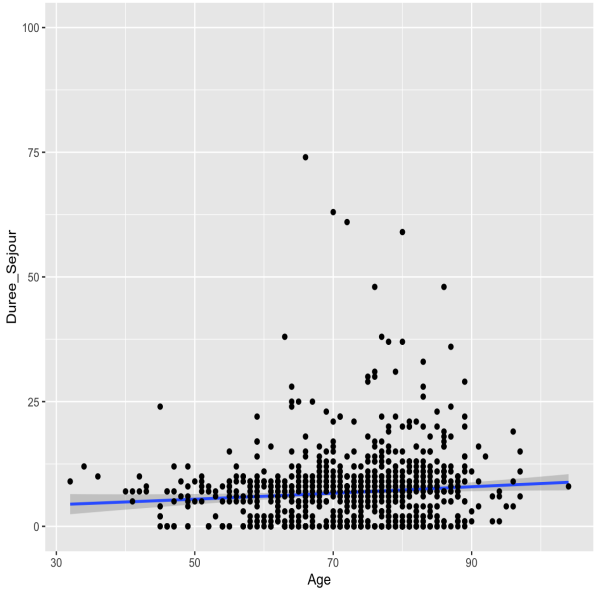
Parameters change the shape of the *geom* and statistics computation:

- geom_smooth(method=lm)
- stat_bin(binwidth = 100)
- stat_summary(fun="mean_cl_boot")
- geom_boxplot(outlier.colour = "red")
- geom_point(colour = "red", size = 5)
- geom_line(linetype = 3)

Parameters

To display a subset of point and a smoothing curve:

```
ggplot(pmsi[sample(nrow(pmsi), size = 1000),],
       aes(x = Age, y = Duree_Sejour)) +
  stat_smooth() + geom_point() + ylim(0, 100)
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



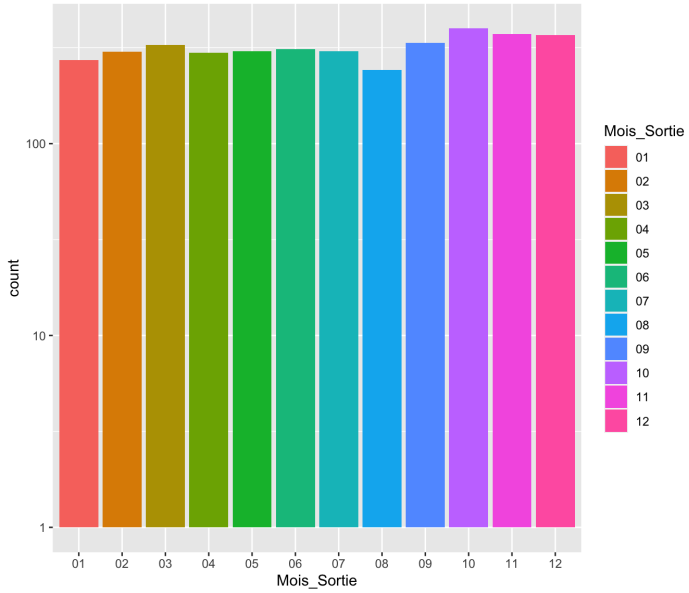
Scales

To change scale in one or more dimension

- Logarithmic scale
- Changing limits
- Changing breaks
- Changing labels
- Changing colours

Scales

```
b <- ggplot(pmsi, aes(x=Mois_Sortie))
b + geom_bar(aes(fill = Mois_Sortie)) +
   scale_y_log10()
```



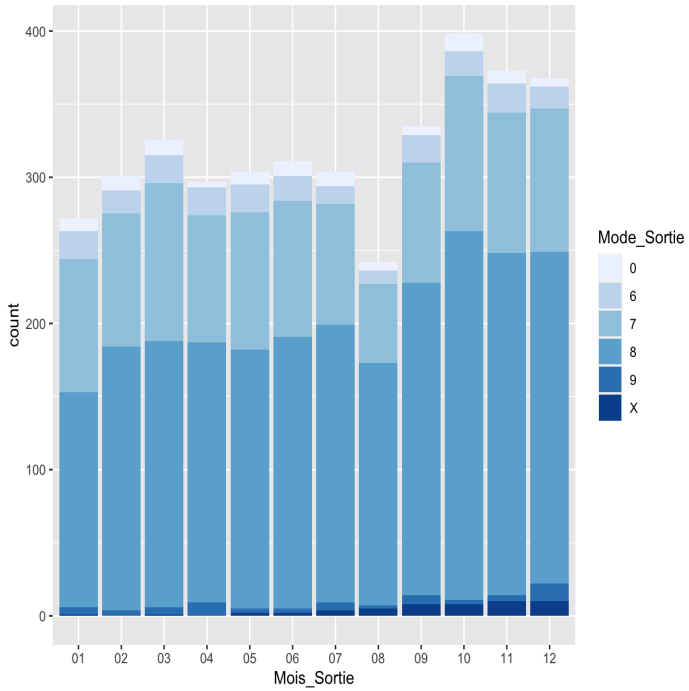
What does that do?

```
b + geom_bar(aes(fill = Mode_Sortie)) + scale_fill_brewer()
```

Scales

What does that do?

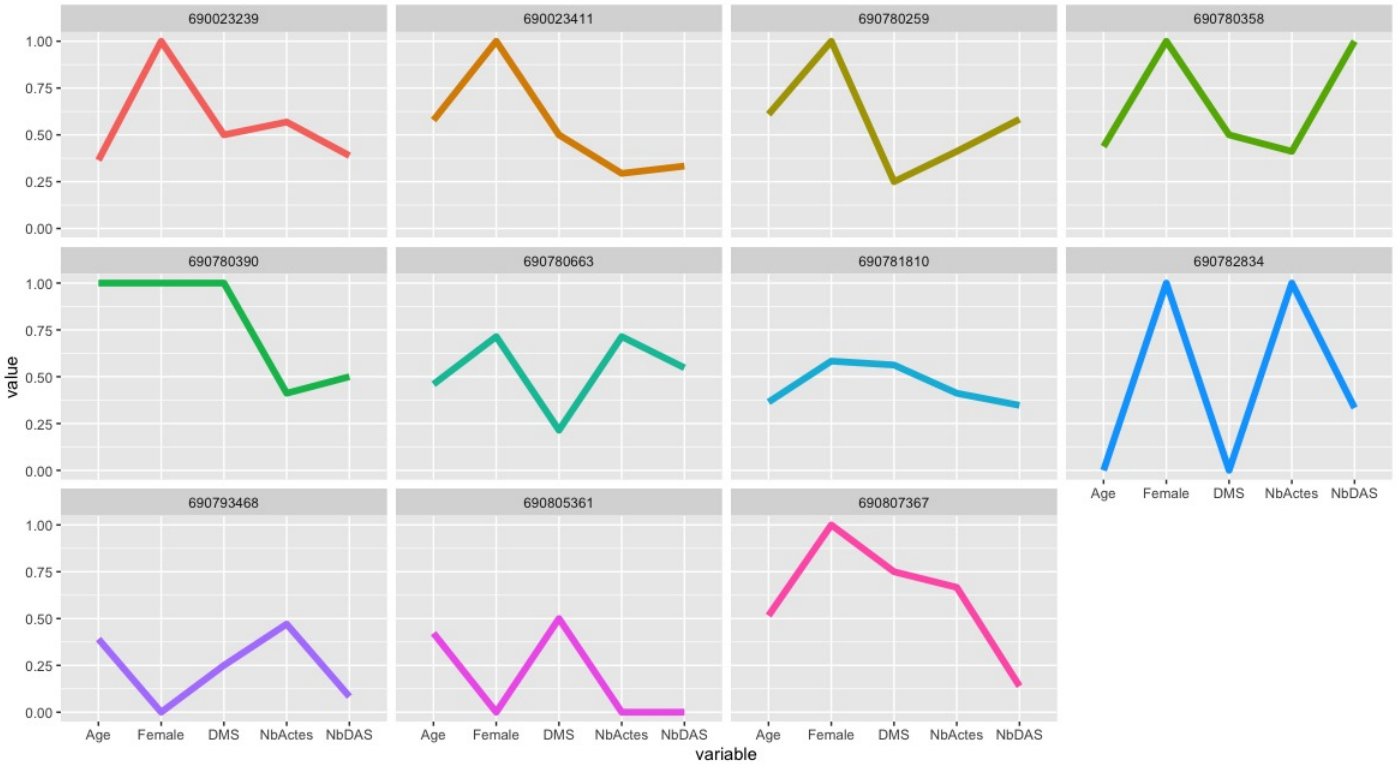
```
b + geom_bar(aes(fill = Mode_Sortie)) + scale_fill_brewer()
```



Parallel Coordinates

Like a flat radar plot

- PMSI data grouped by Finess
- All variables scaled to lie between 0 and 1



References

ggplot2 cheat sheet

https://r-graph-gallery.com/

Datanovia website

Maps

```
library(ggmap)
## ℹ Google's Terms of Service: <https://mapsplatform.google.com>
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
library(maps)
france <- map_data("france")
ggplot(data = france) +
   geom_polygon(aes(x = long, y = lat, fill = region, group = group),
                color = "white") +
   coord_fixed(1.3) +
  guides(fill="none")
```