

# « R » pour les épidémiologistes débutants

## Document supplémentaire

Bernard Branger, 2026, NANTES

[branger44@gmail.com](mailto:branger44@gmail.com) – Tél 06 32 70 33 80

Site Epi-Sûre : <https://www.episure.fr/>

### Aspect de RStudio avec l'éditeur de texte (script) visible

The screenshot shows the RStudio desktop application. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top toolbar has icons for file operations like Open, Save, and Print, along with a 'Go to file/function' search bar and an 'Addins' dropdown.

The main workspace consists of several panes:

- Script Editor:** Shows an R script named 'decl\_temoins.R' with the following code:

```
1 attach(decl_temoins)
2 table (castem, primi)
3
```
- Environment Browser:** Displays the global environment with the following objects:

Object	Type	Value
\$ fiche	num [1:296]	13 779 3...
\$ castem	num [1:296]	2 1 1 2 ...
\$ dateacc	POSIXct[1:296], form...	
\$ hacc	num [1:296]	2 7 19 0...
\$ macc	num [1:296]	40 26 4 ...
\$ hmacc	num [1:296]	2.67 7.4...
\$ geste	num [1:296]	1 1 1 1 ...
\$ parite	num [1:296]	1 1 1 1 ...
- Console:** Shows R version 4.5.1 running on a Windows system. It displays messages from the 'read\_fun' function about unsupported dates and an error message: "Erreur : objet 'castem' introuvable".
- Help Viewer:** Shows the documentation for the 'epi.2by2' function from the 'epiR' package, titled "Summary measures for count data presented in a 2 by 2 table". The description states: "Computes summary measures of risk and a chi-squared test for difference in the observed proportions from count data presented in a 2 by 2 table. With multiple strata the function returns crude and Mantel-Haenszel adjusted measures of association and chi-squared tests of homogeneity".

## Avec le fichier visible sous forme de tableau

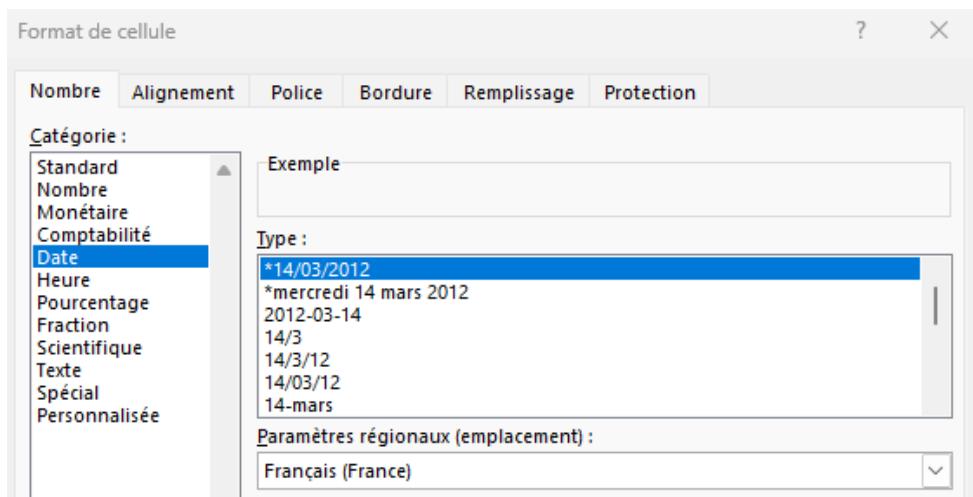
The screenshot shows the RStudio interface with the following components:

- Data View:** Displays a table titled "decl\_temoins" with 296 rows and 59 columns. The columns are: fiche, castem, dateacc, hacc, macc, hmacc, geste, parite, and others.
- Environment View:** Shows the global environment with the "decl\_temoins" object defined as a data frame with 296 observations and 59 variables.
- Console View:** Contains R code and its output. The code includes:
  - Message d'avertissement: "NA inserted for an unsupported date prior to 1900"
  - Execution of `View(decl\_temoins)` and `table(castem, primi)`.
  - An error message: "Erreur : objet 'castem' introuvable".
  - Execution of `attach(decl\_temoins)` followed by `table(castem, primi)` which outputs a 2x2 table:

castem	1	2
1	113	33
2	56	93
- Help View:** Displays the documentation for the `epi.2by2` function from the `epiR` package, specifically the "Summary measures for count data presented in a 2 by 2 table" section.

## Import et manipulation des dates (et des heures)

La manipulation des dates peut être un casse-tête selon les logiciels et les formats. Le mieux est de bien définir les dates sous Excel : sélectionner une colonne, bouton droit, Format de cellule, choisir le format Européen de type (dd/mm/yyyy). Et si possible de calculer les durées et délais dans Excel avant l'importation avec les formules de "Date/Heure".



Dans notre fichier, l'import du fichier Excel a bien pris en compte le format "Date" des variables `dateacc` et `datedec` (imposition du format dans Excel du type dd/mm/yyyy) avec la nuance que "R" a un format "**yyy-mm-dd**". Dans le cas contraire, on peut transformer une variable au départ codée en date avec un autre format ou en caractère.

```
> date3 <- as.Date(dateacc, format="%d/%m/%y")
> list (date3)
[[1]]
[1] "2010-11-23" "2010-02-18" "2010-03-06" "2010-01-07"
[5] "2010-04-02" "2010-03-24" "2010-04-16" "2010-07-25"

> date4 <- as.Date(datedec, format="%d/%m/%y")
> list (date4)
[[1]]
[1] NA          NA          NA          "2010-01-06"
[5] NA          NA          "2010-04-15" "2010-07-25"
```

On peut ensuite faire une différence de dates en jours entre deux dates (ici en jours "`days`", sinon "`years`" pour année).

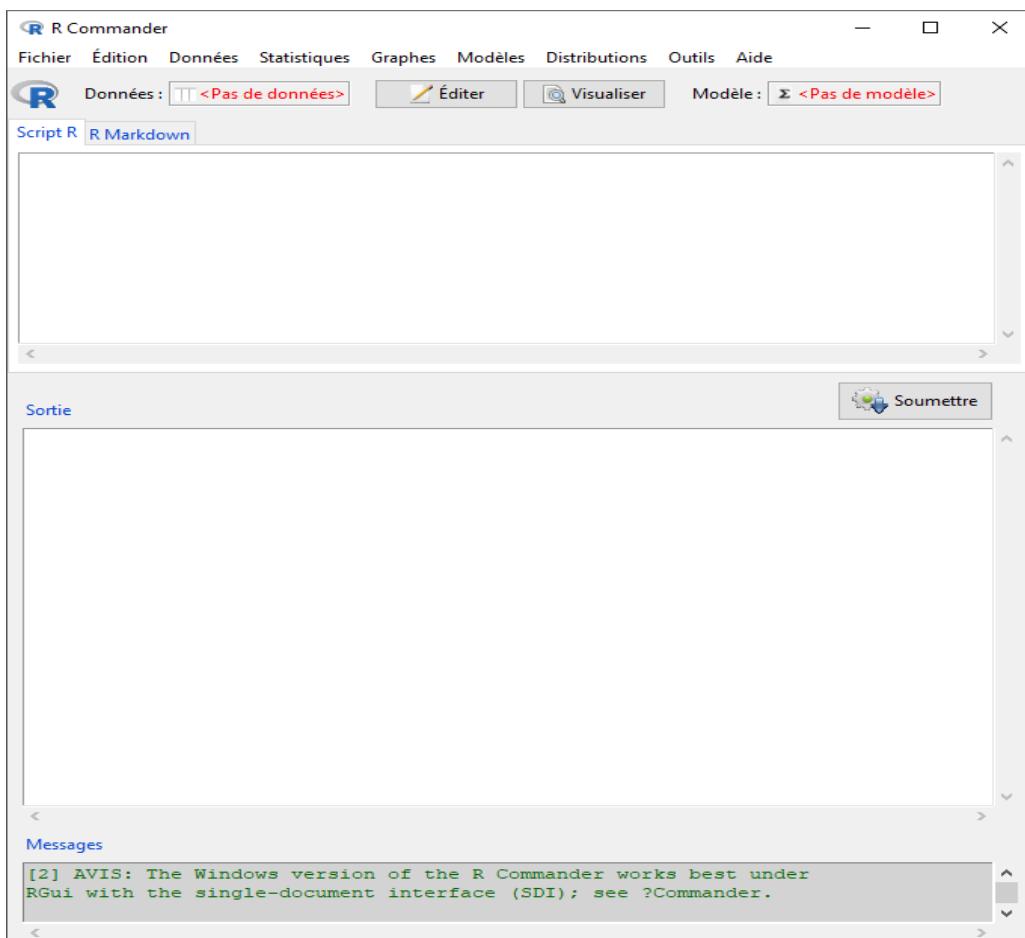
```
> duree <- difftime(dateacc,datedec, units=c("days"))
> summary (duree)
Time differences in days
   Min. 1st Qu. Median    Mean 3rd Qu.    Max.    NA's 
 0.0000  0.0000  1.0000  0.6892  1.0000  5.0000      607
```

## Si on est perdu.... R Commander (cliquable)

**On peut essayer " R commander 2.5.2" qui est plus cliquable, mais limité (pas décrit ici)**

Ecrire dans la ligne de commande install.packages ("Rcmdr") , puis library(Rcmdr) pour lancer le travail et avoir une fenêtre spécifique de travail. Les menus présentés sont assez intuitifs.

**Aspect de la fenêtre d'accueil :**



Cliquer sur les onglets divers pour décrire ou transformer les variables, pour comparer etc...

## Importer un fichier avec une ligne de commandes

N'est envisagée ici que la partie de « R » concernant les « bases de données » ou « structures de données » ou « data frame », déjà saisies et constituées.

Syntaxe assez compliquée ! On va considérer que le fichier a été établi ou exporté sous un tableur. Il faut préalablement, dans le tableur, faire « enregistrer sous » pour un nouveau fichier en \*.txt ou \*.csv. On peut ainsi importer des fichiers en format « texte » avec un espace (" ") comme séparateur (suffixe \*.txt sous Windows), des fichiers « csv » avec une virgule ou un point-virgule comme séparateur (suffixe \*.csv). Attention, l'importation ne concerne que la première feuille du tableur.

Les syntaxes sont les suivantes :

```
> install.packages ("readxl")
> library (readxl)
> read.table("chemin et nom du fichier.txt", sep= "", header=TRUE, dec=".")
> read.csv2("chemin et nom du fichier.csv", sep= ";", header=TRUE, dec=".")
> read_excel("chemin/decl_temoins.xlsx")
```

Les chemins des fichiers doivent comporter des slash normaux comme ceci /, et non pas des anti-slashs comme ceci \ (on peut aussi doubler les anti-slash \\, le tout entre deux guillemets " libra").

Le mot `sep` est pour le séparateur des variables, soit "" pour des espaces, soit ";" pour les points-virgules, soit "," pour des virgules (c'est vous qui décidez lors de "enregistrer sous" dans le tableur).

Le mot `header` permet d'importer le nom des variables de la 1<sup>ère</sup> ligne du tableur : les noms des variables doivent être simples : ne pas commencer par un chiffre, pas de caractères spéciaux (-/%<>....), pas d'accent, minuscules ou majuscules...

Le mot `dec` correspond au format des nombres avec décimales (soit un point, soit une virgule). Sous Excel, par défaut, c'est le format anglo-saxon avec un point. *Sinon, dans Windows, voir Panneau de configuration / Horloge, langue et région/Format/Paramètres supplémentaires/Symbol décimal = point ou virgule.*

On peut ajouter la gestion des valeurs manquantes avec `na="NA"` par exemple.

Pour donner un nom au fichier importé, on peut faire précéder la syntaxe par le nom du fichier (ici « fichier1 ») que l'on traitera. Par exemple

```
> donnees=read.table("chemin et nom du fichier.txt", sep= " ", header=TRUE,
dec=".")
> donnees=read.csv2("chemin et nom du fichier.csv", sep= ";", header=TRUE, d
ec=".")
```

### Pour sauver le fichier au format "R"

```
> save.image("chemin et nom du fichier.R")
```

Ou **Onglet File / Save As**, choisir l'endroit de la sauvegarde

### Pour charger ultérieurement un fichier "R"

```
> load ("chemin et nom du fichier.R")
```

Ou **Onglet File / Open File** : choisir le fichier en .R

## Quelques graphes (bruts, à enjoliver)

### Données qualitatives

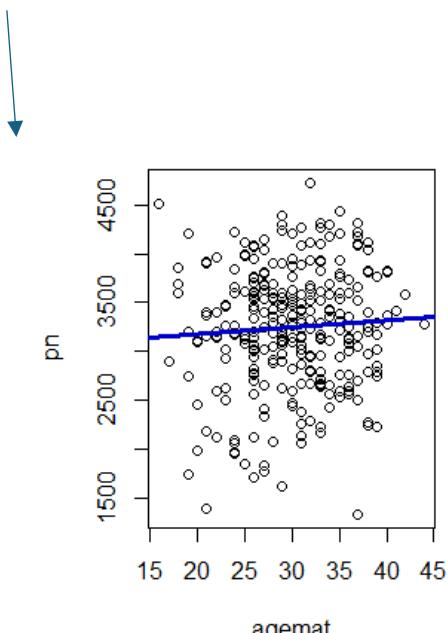
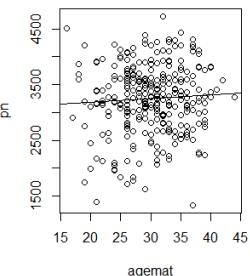
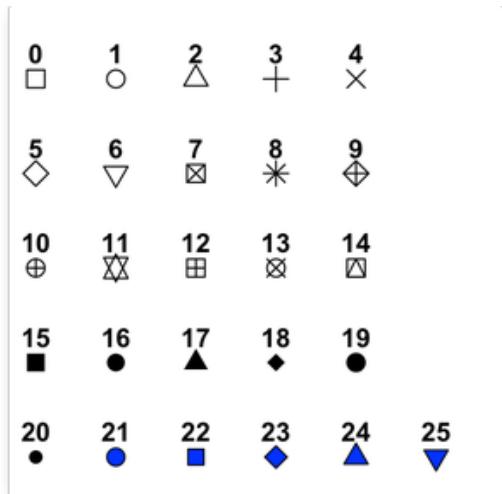
Il n'est guère conseillé dans les articles de présenter des graphes en barres/ bâtons ([> barplot \(x\)](#)) ou en camembert ([> piechart \(x\)](#)) pour résumer des données qualitatives descriptives ou croisées : ce sont des données à montrer dans des tableaux.

### Données quantitatives

Il n'est pas conseillé non plus de montrer des histogrammes ([> hist\(x\)](#)),  
Mais on peut proposer des graphes en points, en box-plot ou en barres.

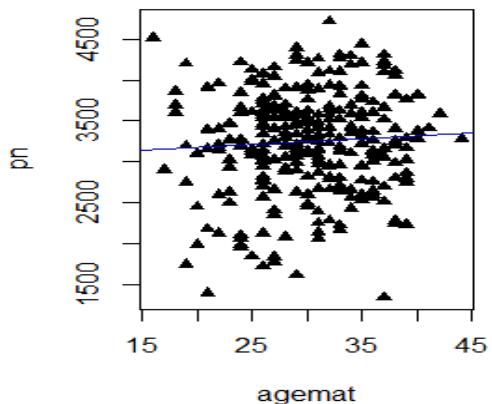
#### Nuages en points (scatterplot) avec la droite de régression

```
> plot (agemat, pn)
> abline (lm (pn~agemat))
> abline (lm (pn~agemat), col="blue3", lwd=3) # avec une couleur bleue et une épaisseur de
3 (défaut = 1)
```



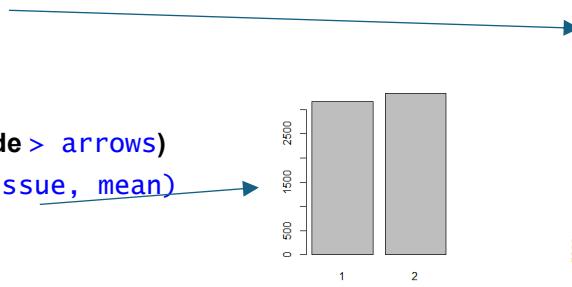
Différents types de points avec `pch = 0 à 25`

```
> plot (agemat, pn, pch=17)
> abline (lm (pn~agemat), col="blue3")
```



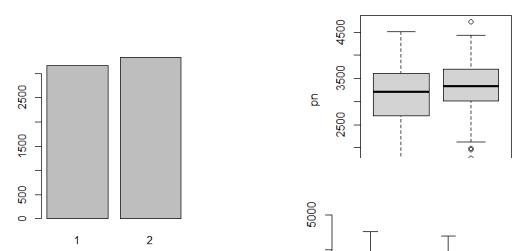
**Box-plot global ou selon des groupes**

```
> boxplot (pn~issue)
```

**Moyenne avec ± 2 écart-types (commande > arrows)**

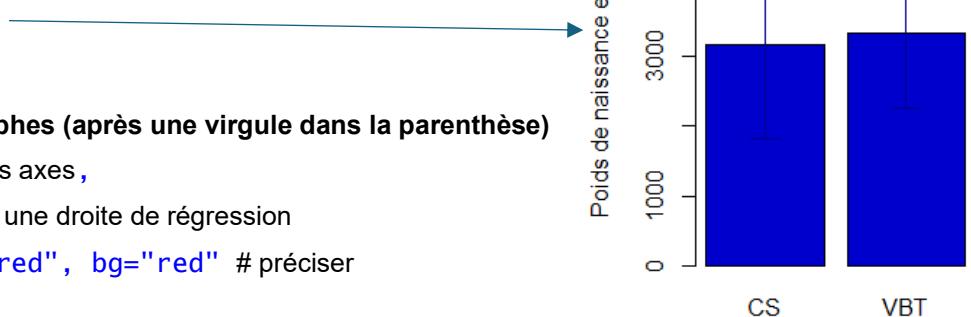
```
> taille_moy=tapply (pn, INDEX=issue, mean)
```

```
> barplot (taille_moy)
```



```
> tab_moy = tapply (pn, INDEX=issue, FUN=mean, na.rm=TRUE)
> tab_et = tapply (pn, INDEX=issue, FUN=sd, na.rm=TRUE)
> limite_inf = tab_moy - 2*tab_et
> limite_sup = tab_moy + 2*tab_et
> graphe1 = barplot (tab_moy, ylim=c(0,5000))
> arrows(x0 = graphe1, y0 = limite_inf, x1 = graphe1, y1 = limite
e_sup, angle=90, code=3,length=0.1)
```

```
> graphe2 = barplot (tab_moy, ylim=c(0,5000), col="blue3"
,xlab="Issue", ylab="Poids de naissance en g")
> arrows(x0 = graphe2, y0 = tab_moy - 2*tab_et, x1 = grap
he2, y1 = tab_moy + 2*tab_et, angle=90, code=3,length=0.1
, col="blue4")
```

**Les mises en forme des graphes (après une virgule dans la parenthèse)**

```
axes=TRUE # pour tracer les axes,
abline (lm(x~y)) # trace une droite de régression
col="blue", col.axis="red", bg="red" # préciser
```

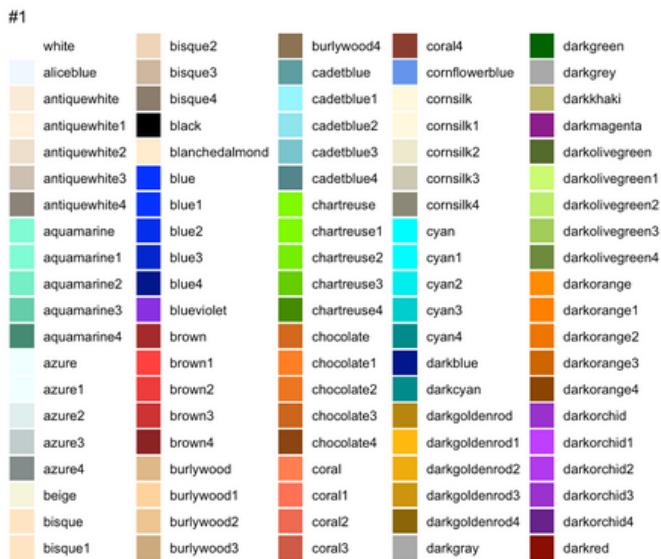
la couleur de fond, des axes

```
xlim ou ylim = c(0, 20) # précise les échelles des axes
xlab ou ylab = "texte" # précise les étiquettes des axes
main="titre" # précise le titre du graphe
sub="sous-titre" # précise le sous-titre du graphe
mfcol=c(3,2 OU mfrow=c(3,2) # positionne plusieurs graphes en 3 lignes et 2 colonnes
code=3 # trace deux traits horizontaux pour ± 2 écart-types
length=0.1 # largeur de ces traits
arrows (x0, y0, x1, y1, angle=90, code=3) # trace les traits de ± 2 écart-types
```

Voir aussi `legend()`, `title ()`, etc...

Voir <https://cran.r-project.org/doc/contrib/Short-refcard.pdf>

**Liste et noms des 20 premières couleurs dans "R" → il y en a 567 différentes !**  
 Voir <https://www.datanovia.com/en/fr/blog/liste-geniale-de-657-noms-de-couleur-dans-r/>



### Graphes avec le package "ggplot2"

Voir <https://larmarange.github.io/analyse-R/graphiques-bivaries-ggplot2.html>

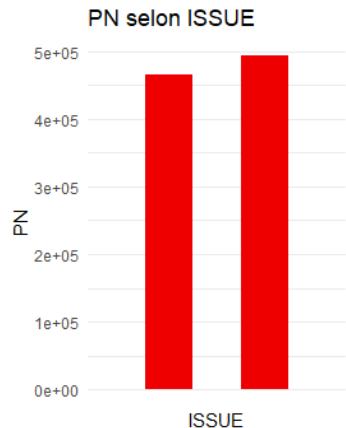
OU <https://www.datanovia.com/en/fr/lessons/ggplot-barres-erreurs/>

```
> install.packages("ggplot2")
> library(ggplot2)
> attach(decl_temoins)
> issue <- as.factor (issue)
> issue <- factor(issue, levels=c(1,2), labels=c("CS", "VBT"))
> summary (issue)
> tab_moy = tapply (pn, INDEX=issue, FUN=mean, na.rm=TRUE)
> tab_sd = tapply (pn, INDEX=issue, FUN=sd, na.rm=TRUE)
> limite_inf = tab_moy - 2*tab_sd
> limite_sup = tab_moy + 2*tab_sd

> graphe11 <- ggplot (decl_temoins, aes (x=issue, y=pn)) +
  geom_bar (stat="identity", fill="red2", width=0.2) +
  labs (title="PN selon ISSUE", x="ISSUE", y="PN") +
  theme_minimal()
> print (graphe11)
```

A développer avec les écart-types sur le graphe :

```
> taille_moy = tapply (taille, INDEX=issue, FUN=mean,
na.rm=TRUE)
> taille_sd = tapply (taille, INDEX=issue, FUN=sd,
na.rm=TRUE)
> graphe12 <- ggplot (decl_temoins, aes (x=issue,y=taille,
fill=issue)) + geom_bar (stat="identity", color="blue3", width=0.7) +
  geom_errorbar(aes(ymin =taille_moy, ymax=taille_moy+2*taille_sd), width=0.2)
> print (graphe12)
```



## Tests diagnostiques et courbe ROC

### Avec les packages "epiR" et "pROC"

Plusieurs packages existent pour calculer la sensibilité et la spécificité d'un seuil d'une variable quantitative vis-à-vis d'une variable de maladie codée en 1 ou 2 (préférable), ou en 0 et 1.

```
> install.packages ("epiR")
> library(epiR)
```

Importer le fichier `decl_temoins.xlsx`

```
> attach (decl_temoins)
```

Détermination d'un âge maternel en deux classes ( $\pm 30$  ans)

```
> issue <- as.factor (issue)
> age30 <- cut (agemat, breaks=c(0,30,45))
> summary (age30)
(0,30] (30,45]
 156     139
```

Tableau de performance du seuil

```
> tab22 <- table (age30, issue)
> addmargins (tab22)
      issue
age30    1   2
(0,30] 80  76 156
(30,45] 67  72 139
           147 148

> epi.tests (tab22)
      Outcome +   Outcome -   Total
Test +          80        76    156
Test -          67        72    139
Total         147       148    295
```

Point estimates and 95% CIs:

Apparent prevalence *	0.53 (0.47, 0.59)
True prevalence *	0.50 (0.44, 0.56)
<b>Sensitivity *</b>	<b>0.54 (0.46, 0.63) # 80 / 147</b>
<b>Specificity *</b>	<b>0.49 (0.40, 0.57) # 72 / 139</b>
<b>Positive predictive value *</b>	<b>0.51 (0.43, 0.59) # 80 / 156</b>
Negative predictive value *	0.52 (0.43, 0.60)
<b>Positive likelihood ratio</b>	<b>1.06 (0.85, 1.31)</b>
Negative likelihood ratio	0.94 (0.74, 1.19)
False T+ proportion for true D- *	0.51 (0.43, 0.60)
False T- proportion for true D+ *	0.46 (0.37, 0.54)
False T+ proportion for T+ *	0.49 (0.41, 0.57)
False T- proportion for T- *	0.48 (0.40, 0.57)
<b>Correctly classified proportion *</b>	<b>0.52 (0.46, 0.57)</b>

A noter :

PLR ou le LR + : rapport "Vrais positifs / Faux positifs" = Se / (1 – Sp) = (80 / 147) / (76 / 148) = 1.06

Bien classés : (80 + 72) / 295 = 0.52

**Courbe ROC avec l'âge maternel a**

```
> install.packages("pROC")
> library(pROC)
> courbe <- roc(issue, agemat)

Setting levels: control = 1, case = 2
Setting direction: controls < cases
> plot(courbe, main ="Courbe ROC Age maternel")
```

Aire sous la courbe (area under curve = AUC)

```
> courbe_auc <- auc(courbe)
```

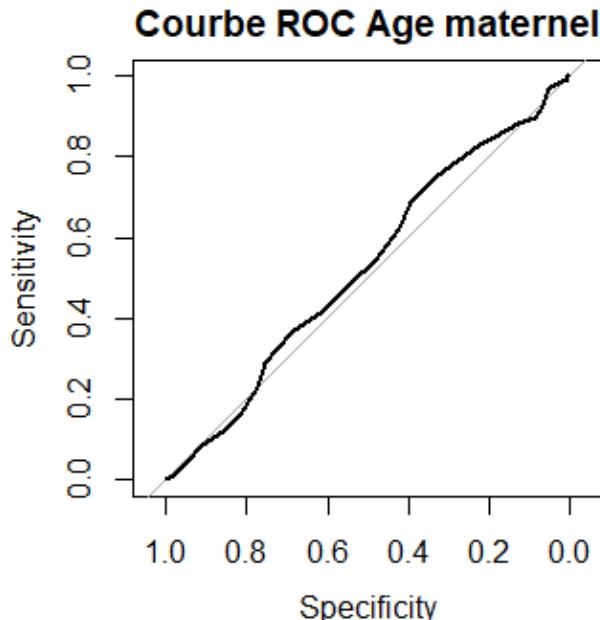
```
> courbe_auc
```

**Area under the curve: 0.5286**

AUC très proche de 0.50 → Age mère pas prédictive de l'issue de l'accouchement

Détermination du meilleur seuil mathématique (point proche du coin sup gauche du graphe (diagonale). Ce n'est pas forcément le seuil important pour le clinicien qui, en fonction de la clinique ou d'un dépistage, peut privilégier plutôt la sensibilité ou plutôt la spécificité :

```
> seuil <- coords(courbe, "best")
> seuil
threshold specificity sensitivity
1      27.5     0.3945578    0.6891892
```

**Exemple avec la taille de la mère**

```
> courbe2 <- roc(issue, taille)
> plot(courbe2, main ="Courbe ROC Taille mère")

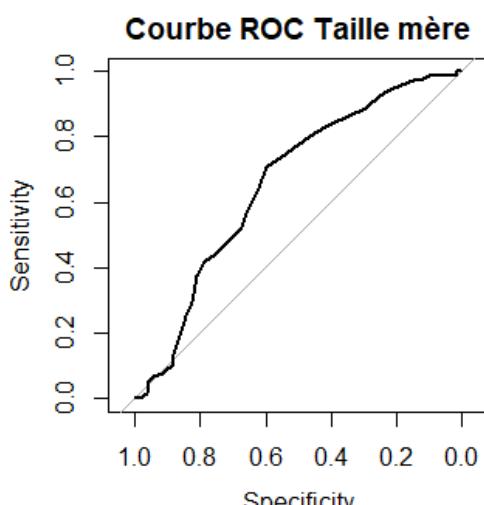
> courbe2_auc <- auc(courbe2)
> courbe2_auc
Area under the curve: 0.661
```

Essai de détermination d'un seuil pour l'obstétricien

Meilleur seuil mathématique

```
> seuil2 <- coords(courbe2, "best")
> seuil2
threshold specificity sensitivity
1      1.615     0.5985915    0.7062937
```

La taille de 1 m 61 comme seuil n'est pas "commode", et cette taille "équilibre" le risque de CS et le risque de VB. En pratique courante, il faut mieux privilégier le fait d'éviter une CS +++, c'est-à-dire augmenter la spécificité et diminuer la sensibilité.



**Proposition de seuil de taille de 1 m 60**

```
> t160 <- cut (taille, breaks=c(0,1.60,1.80))
> tab160 <- table (t160, issue)
> addmargins (tab160)
      issue
t160      1   2 Sum
  (0,1.6]   81  39 120
  (1.6,1.8] 61 104 165
  Sum       142 143 285

> epi.tests (table (t160, issue))
      Outcome +    Outcome -    Total
Test +          81           39        120
Test -          61          104        165
Total          142          143        285
```

Point estimates and 95% CIs:

Apparent prevalence *	0.42 (0.36, 0.48)
True prevalence *	0.50 (0.44, 0.56)
<b>Sensitivity *</b>	<b>0.57 (0.48, 0.65)</b>
<b>Specificity *</b>	<b>0.73 (0.65, 0.80)</b>
<b>Positive predictive value *</b>	<b>0.68 (0.58, 0.76)</b>
Negative predictive value *	0.63 (0.55, 0.70)
Positive likelihood ratio	2.09 (1.54, 2.83)
Negative likelihood ratio	0.59 (0.48, 0.73)
...	

Le seuil de 1 m 60 prédit 81 césariennes sur les 120 femmes "petites" (VPP de 68 %), mais 39 parmi elles accouchent par voie basse.

**Proposition de seuil de taille de 1 m 65**

```
> t165 <- cut (taille, breaks=c(0,1.65,1.80))
> tab165 <- table (t165, issue)
> addmargins (tab165)
      issue
t165      1   2 Sum
  (0,1.65] 107  80 187
  (1.65,1.8] 35  63  98
  Sum       142 143 285

> epi.tests (tab165, issue)
      Outcome +    Outcome -    Total
Test +          107          80        187
Test -          35          63        98
Total          142          143        285
```

Point estimates and 95% CIs:

Apparent prevalence *	0.66 (0.60, 0.71)
True prevalence *	0.50 (0.44, 0.56)
<b>Sensitivity *</b>	<b>0.75 (0.67, 0.82)</b>
<b>Specificity *</b>	<b>0.44 (0.36, 0.53)</b>
<b>Positive predictive value *</b>	<b>0.57 (0.50, 0.64)</b>
...	

Le seuil de 1 m 65 prédit 107 césariennes sur les 187 femmes "petites" (VPP de 57 %)

**Proposition de taille de 1 m 55**

```
> t155 <- cut (taille, breaks=c(0,1.55,1.80))
> tab155 <- table (t155, issue)
> addmargins (tab155)
      issue
taille      1   2 Sum
(0,1.55]    31   8 39
(1.55,1.8] 111 135 246
Sum        142 143 285

> epi.tests (tab155)
      Outcome +     Outcome -     Total
Test +          31            8       39
Test -         111           135      246
Total          142           143      285
```

Point estimates and 95% CIs:

Apparent prevalence *	0.14 (0.10, 0.18)
True prevalence *	0.50 (0.44, 0.56)
Sensitivity *	0.22 (0.15, 0.30)
Specificity *	0.94 (0.89, 0.98)
<b>Positive predictive value *</b>	<b>0.79 (0.64, 0.91)</b>
Negative predictive value *	0.55 (0.48, 0.61)
Positive likelihood ratio	3.90 (1.86, 8.19)
Negative likelihood ratio	0.83 (0.75, 0.91)
False T+ proportion for true D- *	0.06 (0.02, 0.11)
False T- proportion for true D+ *	0.78 (0.70, 0.85)
False T+ proportion for T+ *	0.21 (0.09, 0.36)
False T- proportion for T- *	0.45 (0.39, 0.52)
Correctly classified proportion *	0.58 (0.52, 0.64)

Le seuil de 1 m 55 est plus prédictif : 31 césariennes sur 39 femmes "petites" , mais concerne peu de femmes. En fait , plus la population de femmes à risques est grande, plus la VPP est petite.

**Courbe ROC après régression logistique : ensemble de facteurs prédictifs**

```
> library(epiDisplay)
> attach (decl_temoins)

> summary (taille)
  Min. 1st Qu. Median     Mean 3rd Qu.     Max.     NA's
1.450  1.580  1.630  1.629  1.680  1.790      10
> taille=taille*100
> summary (taille)
  Min. 1st Qu. Median     Mean 3rd Qu.     Max.     NA's
145.0 158.0 163.0 162.9 168.0 179.0      10

> summary (bishop)
  Min. 1st Qu. Median     Mean 3rd Qu.     Max.
0        4        5        5        6        9
> summary (imc)
  Min. 1st Qu. Median     Mean 3rd Qu.     Max.     NA's
15.67 20.96 23.44 25.00 27.94 44.41      11
> issue <- as.factor (issue)
> issue2 <- replace (issue, issue == 2, 0)
```

```

> table (issue2)
issue2
  0   1
148 147

> utcicat <- as.factor (utcicat)
> utcicat <- relevel(utcicat, ref="2")
> table (utcicat)
utcicat
  2   1
276 18

> primi <- as.factor (primi)
> primi <- relevel(primi, ref="2")
> table (primi)
primi
  2   1
126 168

> modedec1 <- as.factor (modedec1)
> table (modedec1)
modedec1
  1   2   3
120 149 26

> reg_log2 <- glm (issue2~primi+modedec1+taille+imc+bishop+utcicat, family=binomial)
> logistic.display(reg_log2)

> lroc (reg_log2, graph=TRUE)
$model.description
[1] "issue2 ~ primi + modedec1 + taille + imc + bishop + utcicat"

$auc
[1] 0.8567497

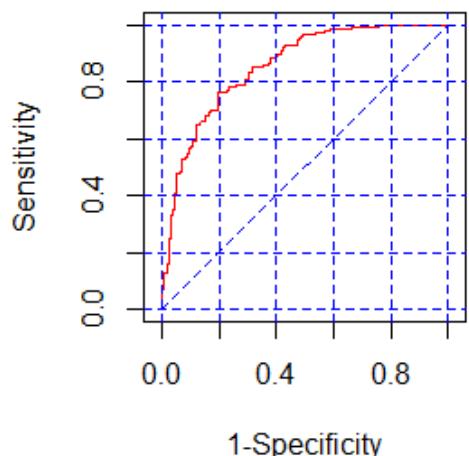
$predicted.table
predicted.prob Non-diseased Diseased
  0.0150          1      0
  0.0201          1      0
  0.0216          1      0
  .....
  0.9924          0      1

$diagnostic.table
  1-Specificity Sensitivity
  1.0000000000 1.0000000000
> 0.992957746 1.0000000000
  .....
> 0.0000000000 0.0000000000

```

Voir Help `lroc` pour améliorer le graphe avec titres, AUC sur la courbe, limites des x et y...

Voir aussi <https://rpubs.com/GjT/5267>, ou <https://rdrr.io/cran/epiDisplay/man/roc.html>



## Etude de la survie (commandes sans les résultats)

Voir [https://epirhandbook.com/fr/new\\_pages/survival\\_analysis.fr.html](https://epirhandbook.com/fr/new_pages/survival_analysis.fr.html)

```
install.packages("survival")
library (survival)
Importer le fichier decl_temoins.xlsx avec la durée du déclenchement en heures comme variable de
durée (deldec) et issue comme variable d'événement.
```

```
attach(decl_temoins)
summary (deldec)
deldec <- na.omit (deldec)
summary (issue)
issue <- replace (issue, issue==2,0) # avec 2 signes = à la suite
issue <- as.factor (issue) # voir événements (statut) peut-être 1 et 2, ou 0 et 1
summary(issue)
```

### Table de survie avec les événements = 1 et censurés = 0

```
survie1 <- survfit(Surv (deldec, issue)~1, data=decl_temoins)
summary (survie1, censored=TRUE)
```

### Graphe de survie : courbe de Kaplan-Meier

```
plot (survie1, mark.time=TRUE, conf.int=FALSE) # avec options de graphes ++
```

### Comparer deux groupes : table, courbes et tests avec un ratio

```
primi <- as.factor (primi)
summary (primi)
primi <- na.omit (primi)
survie_compar <- survfit (formula=Surv (deldec, issue)~primi, data=decl_temoins)
summary (survie_compar, censored=TRUE) # voir pour Moyenne et médiane de survie ? p = 0.50 ?
plot (survie_compar)
test_lr <- survdiff (Surv(deldec, issue)~primi, data=decl_temoins) # Log-rank
test_lr
HR <- coxph (formula=Surv(deldec, issue)~primi) # Hazard ratio (HR)
HR
```

### Modèle de Cox avec deux variables sexe et taille

```
install.packages("survminer")
library (survminer)
ggsurvplot (test_lr, data=decl_temoins, pval=T, risk.table=TRUE) # vérifier
cox1 <- coxph (Surv (deldec, issue) ~ sexe + taille, data=decl_temoins)
cox1 # HR ajustés
risq_prop <- cox.zph(cox1)
risq_prop # vérification des risques proportionnels
```

### Avec sexe, taille et primi

```
cox1 <- coxph (Surv (deldec, issue) ~ sexe + taille + primi, data=decl_temoins)
cox1
risq_prop <- cox.zph(cox1)
risq_prop
```

**Voir aussi les commandes suivantes :** [ggcoxzph](#), [ggforest](#).... Lien internet en début de page

## Propensity score (score de propension) (commandes sans les résultats)

Objectifs : Apparier des cas et des témoins avec le même risque ou un risque voisin d'être exposé.

Voir <file:///C:/Users/brang/Downloads/pare-1370-randolph.pdf> (2014)

```

install.packages("epitools")
install.packages("epiDisplay")
install.packages("epiR")
install.packages("optmatch")
install.packages ("RItools")
install.packages("MatchIt")
install.packages ("tableone")
library(MatchIt)
library(optmatch)
library(epiDisplay)
library(epiR)
library(epitools)
library(RItools)
library(tableone)

attach (fichier)

score_PS <- glm(var_jugement~var1+var2+var3, family=binomial)
Score_PS
summary(score_PS)
summary(score_PS$fitted.values)
boxplot(score_PS$fitted.values)
hist(score_PS$fitted.values)

# méthode nearest
Match1<-matchit(var_jugement~var1+var2+var3, method="nearest",data=fichier)
summary(Match1)
head (Match1)
Table1<-CreateTableOne(vars=c('var1', 'var2','var3'), data=match.data(Match1),
strata="var_jugement")
Table1

# fichiers des appariés
m.final<- match.data(Match2)
Tableau.final<- table (m.final$var_jugement, m.final$var1)
Tableau.final

```

Commandes suivantes : comparer les variables d'exposition avec la nouvelle classification de la variable de jugement.

## Commandes de différents packages. Commandes dans "R"

```
> ls("package:epitools")
```

```
[1] "ageadjust.direct"      "ageadjust.indirect"   "as.hour"          "as.month"        "as.week"  
[6] "binom.approx"         "binom.exact"       "binom.wilson"     "colorbrewer.data" "colorbrewer.display"  
[11] "colorbrewer.palette"   "colors.matrix"     "colors.plot"      "epicurve.dates"  "epicurve.hours"  
[16] "epicurve.months"      "epicurve.table"    "epicurve.weeks"   "epidate"         "epitab"  
[21] "epitable"              "expand.table"     "expected"        "julian2date"     "kapmeier"  
[26] "oddsratio"             "oddsratio.fisher" "oddsratio.midp"  "oddsratio.small" "oddsratio.wald"  
[31] "or.midp"               "ormidp.test"      "pois.approx"     "pois.byar"       "pois.daly"  
[36] "pois.exact"            "probratio"        "rate2by2.test"   "rateratio"      "rateratio.midp"  
[41] "rateratio.wald"       "ratetable"        "riskratio"       "riskratio.boot" "riskratio.small"  
[46] "riskratio.wald"        "tab2by2.test"     "table.margins
```

```
> ls("package:epiDisplay")
```

```
[1] "aggregate.numeric"      "aggregate.plot"    "alpha"           "alpha"  
[4] "alphaBest"              "cc"                "cci"            "cci"  
[7] "ci"                     "ci.binomial"     "ci.default"     "ci.default"  
[10] "ci.numeric"             "ci.poisson"      "clogistic.display" "clogistic.display"  
[13] "codebook"               "cox.display"     "cs"             "cs"  
[16] "csi"                    "des"              "dotplot"        "dotplot"  
[19] "followup.plot"          "graph.casecontrol" "graph.prospective" "graph.prospective"  
[22] "idr.display"            "kap"              "kap.2.raters"  "kap.2.raters"  
[25] "kap.ByCategory"         "kap.default"     "kap.m.raters"  "kap.m.raters"  
[28] "kap.table"               "labelTable"      "logistic.display" "logistic.display"  
[31] "lookup"                 "lroc"            "lrtest"         "lrtest"  
[34] "lsNoFunction"           "make2x2"        "matchTab"       "matchTab"  
[37] "mhor"                   "mlogit.display" "n.for.2means"   "n.for.2means"  
[40] "n.for.2p"                "n.for.cluster.2means" "n.for.cluster.2p" "n.for.cluster.2p"  
[43] "n.for.equi.2p"          "n.for.lqas"      "n.for.noninferior.2p" "n.for.noninferior.2p"  
[46] "n.for.survey"           "ordinal.or.display" "poisgof"        "poisgof"  
[49] "power.for.2means"        "power.for.2p"    "print.alpha"    "print.alpha"  
[52] "print.cci"               "print.des"       "print.display"  "print.display"  
[55] "print.kap.ByCategory"   "print.kap.table" "print.lrtest"    "print.lrtest"  
[58] "print.matchTab"          "print.n.for.2means" "print.n.for.2p"  "print.n.for.2p"  
[61] "print.n.for.cluster.2means" "print.n.for.cluster.2p" "print.n.for.equi.2p" "print.n.for.equi.2p"  
[64] "print.n.for.lqas"        "print.n.for.noninferior.2p" "print.n.for.summ" "print.n.for.summ"  
[67] "print.power.for.2means"  "print.power.for.2p"  "print.statStack" "print.statStack"  
[70] "print.summ.data.frame"   "print.summ.default" "print.tab1"      "print.tab1"  
[73] "print.tableStack"        "pyramid"         "regress.display" "regress.display"  
[76] "roc.from.table"          "setTitle"        "shapiro.qqnorm" "shapiro.qqnorm"  
[79] "statStack"               "summ"            "summ.data.frame" "summ.data.frame"  
[82] "summ.default"            "summ.factor"     "summ.logical"  "summ.logical"  
[85] "tab1"                    "tableGlm"        "tableStack"    "tableStack"  
[88] "tabpct"
```

```
> ls("package:epiR")
[1] "epi.2by2"          "epi.about"        "epi.asc"          "epi.betabuster"
[5] "epi.blcm.paras"   "epi.bohning"     "epi.ccc"          "epi.conf"
[9] "epi.convgrid"      "epi.cp"          "epi.cpresids"    "epi.descriptives"
[13] "epi.dgamma"        "epi.directadj"   "epi.dms"          "epi.dsl"
[17] "epi.edr"           "epi.empbayes"   "epi.herdtest"    "epi.indirectadj"
[21] "epi.insthaz"       "epi.interaction" "epi.iv"          "epi.kappa"
[25] "epi.ltd"           "epi.mh"          "epi.nomogram"   "epi.occc"
[29] "epi.offset"        "epi.pooled"      "epi.popsize"     "epi.prcc"
[33] "epi.prev"          "epi.psi"         "epi.realarisk"  "epi.RtoBUGS"
[37] "epi.smd"           "epi.smr"         "epi.sscc"         "epi.ssclus1estb"
[41] "epi.ssclus1estb"   "epi.ssclus2estb" "epi.ssclus2estc" "epi.sscohortc"
[45] "epi.sscohortt"     "epi.sscompb"     "epi.sscompc"     "epi.sscomps"
[49] "epi.ssdetect"       "epi.ssdxsesp"   "epi.ssdxtest"    "epi.ssequb"
[53] "epi.ssequc"         "epi.ssninfb"    "epi.ssninfc"    "epi.sssimpleestb"
[57] "epi.sssimpleestc"  "epi.ssstrataestb" "epi.ssstrataestc" "epi.sssupb"
[61] "epi.sssupc"         "epi.ssxsectn"   "epi.tests"       "rsu.adjrisk"
[65] "rsu.dxtest"         "rsu.epinf"       "rsu.pfree.equ"  "rsu.pfree.rs"
[69] "rsu.pstar"          "rsu.sep"         "rsu.sep.cens"   "rsu.sep.pass"
[73] "rsu.sep.rb"          "rsu.sep.rb1rf"   "rsu.sep.rb2rf"   "rsu.sep.rb2st"
[77] "rsu.sep.rbvarse"    "rsu.sep.rs"      "rsu.sep.rs2st"   "rsu.sep.rsfreecalc"
[81] "rsu.sep.rsmult"     "rsu.sep.rspool"  "rsu.sep.rsparse" "rsu.spp.rs"
[85] "rsu.sspfree.rs"      "rsu.sssep.rb2st1rf" "rsu.sssep.rb2st2rf" "rsu.sssep.rbmrg"
[89] "rsu.sssep.rbsrg"    "rsu.sssep.rs"    "rsu.sssep.rs2st" "rsu.sssep.rsfreecalc"
[93] "rsu.sssep.rspool"
```

=====

## Le package **tidyverse** et **dplyr** (script seulement)

```

install.packages ("tidyverse")
library (tidyverse)
library (dplyr)

library(readxl)
Importer le fichier declenche_2025.xlsx
slice (declenche_2025, 1:30)
attach(declenche_2025)
summary (issue)
summary (taille)
taille <- na.omit (taille)
summary (taille)

filter ➔ Nouveau fichier sans le codage =3 de issue
dec10 <- dplyr::filter (declenche_2025, issue<3)
attach (dec10)
summary (dec10)
summary (issue)
issue <- as.factor (issue)
summary (issue)

select ➔ Nouveau fichier avec seulement 3 variables (colonnes)
detach(dec10)
attach (declenche_2025)
dec11 <- dplyr::select (declenche_2025, issue, taille, primi)
# quand une commande existe dans plusieurs packages, signaler le package puis :: avant la commande !
attach (dec11)
head (dec11)
print (1:100)
str (dec11)
summary (taille)
issue <- as.factor (issue)
summary (issue)
primi <- as.factor (primi)
summary (primi)
primi <- na.omit (primi)
summary (primi)
table (dec11$issue, dec11$primi)

```

C'est un "L", pas un "1"

**select** → Même fichier sans la variable primi

```
dec12 <- dplyr::select (dec11, -primi)
attach (dec12)
head (dec12)
summary (taille)
```

**rename** → Renommer une variable (colonne) (**nouvelle = ancienne**)

```
rename (dec11, height=taille)
```

**mutate** → Modifier la valeur d'une variable (taille en cm au lieu de taille en mètres) en créant une nouvelle variable

```
dec13 <- dplyr::mutate (dec12, taille_cm = taille*100)
attach (dec13)
summary (taille_cm)
issue <- as.factor (issue)
issue <- factor(issue, levels=c(1,2), labels=c("CS", "VBT"))
table (issue)
```

**Summarise** → calculer des statistiques (avec un "s", pas un "z" dans ce package...)

```
summarise (taillemoy = mean (taille))
```

**group\_by** → défini des groupes et les analyser

```
group_by (issue)
summarise (taillemoy)
```

## Utilisation du pipe %>%

Cette possibilité permet d'avoir une écriture plus logique, qui introduit les commandes à la suite, progressivement : d'abord le fichier, puis le pipe, puis la fonction.

Voir <https://larmarange.github.io/guide-R/manipulation/pipe.html>

OU [https://book.utilitr.org/03\\_Fiches\\_thematiques/Fiche\\_tidyverse.html](https://book.utilitr.org/03_Fiches_thematiques/Fiche_tidyverse.html)

On peut faire succéder plusieurs pipes --> c'est une **pipeline**. NB : quand une commande a la même écriture dans différents packages, faire précéder la commande du package suivi de : : pour ne considérer que la commande du package en question.

```
attach(decenche_2025)
decenche_2025 %>%
  filter (issue==1) %>%
  dplyr::select (issue, taille, primi) %>%
  dplyr::mutate(taille_cm=taille*100)
```

=====



## Table des matières

<b>Aspect de RStudio avec l'éditeur de texte visible .....</b>	<b>1</b>
<b>Avec le fichier visible sous forme de tableau .....</b>	<b>2</b>
<b>Import et manipulation des dates (et des heures) .....</b>	<b>3</b>
<b>Si on est perdu.... R Commander (cliquable) .....</b>	<b>4</b>
<b>Importer un fichier avec une ligne de commandes .....</b>	<b>5</b>
<b>Quelques graphes (bruts, à enjoliver) .....</b>	<b>6</b>
<b>Tests diagnostiques et courbe ROC.....</b>	<b>9</b>
<b>Etude de la survie (commandes sans les résultats) .....</b>	<b>14</b>
<b>Propensity score (score de propension) (commandes sans les résultats) .....</b>	<b>15</b>
<b>Commandes de différents packages. Commandes dans "R" .....</b>	<b>16</b>
<b>Le package tidyverse et dplyr (script seulement) .....</b>	<b>18</b>