

# R - Notes techniques

*JcB*

*28/01/2015*

## Contents

<b>1</b>	<b>Notes techniques</b>	<b>2</b>
<b>2</b>	<b>Notes Github</b>	<b>2</b>
<b>3</b>	<b>Notes <a href="#">Knitr</a></b>	<b>2</b>
<b>4</b>	<b>Notes Pandoc</b>	<b>2</b>
4.1	Transformer un fichier <b>md</b> en <b>pdf</b> : . . . . .	2
<b>5</b>	<b>Notes Latex</b>	<b>3</b>
5.1	Ecriture d'un nombre et virgule . . . . .	3
<b>6</b>	<b>R - RStudio</b>	<b>3</b>
6.1	Utilisation du nom de l'ordinateur pour définir un chemin d'accès: . . . . .	3
6.2	Référencer le travail . . . . .	3
6.3	Librairies . . . . .	4
<b>7</b>	<b>Outils de présentation</b>	<b>9</b>
<b>8</b>	<b>Analyse de la variance (one-way)</b>	<b>9</b>
<b>9</b>	<b>Boxplot</b>	<b>10</b>
9.1	Pour supprimer la notation scientifique: . . . . .	10
9.2	Pour imposer un préfixateur de milliers: . . . . .	10
9.3	Barchart . . . . .	10
9.4	Histogramme de qualité . . . . .	10
9.5	Tracer 2 histogrammes en miroir . . . . .	11
9.6	Packages installés . . . . .	11
9.7	#### Franciser et réorganiser la semaine . . . . .	11
<b>10</b>	<b># radial.plot</b>	<b>12</b>
<b>11</b>	<b>Dataframe</b>	<b>12</b>
11.1	Réordonner les colonnes . . . . .	12

<b>12 Remarques sur “format”</b>	<b>12</b>
<b>13 graphe avec 2 axes y pour les passages et le taux d’hospitalisation.</b>	<b>12</b>
13.1 Ajouter une légende à un graphique . . . . .	13
13.2 Transformer Latin 1 (windows) en UTF8 (linux) . . . . .	13
13.3 %in% et !%in% . . . . .	13

## 1 Notes techniques

- Notes Github
- Notes Knitr
- Notes Pandoc
- Notes Latex
- Notes R et RStudio

## 2 Notes Github

git push origin master:

- git push: on pousse le dépôt git
- origin: dans le dépôt d’origine
- master: branche master

Si on travaille toujours sur le dépôt master de la branche origin, un simple push suffit.

## 3 Notes **Knitr**

Pour inclure des options de **Knitr** dans un document RStudio:

```
# ```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE) # si one ne met pas library(knitr)
set_alias(w = "fig.width", h = "fig.height")
```

ou

```
# ```{r Declarations, echo=FALSE, include=FALSE}
library(knitr)
# set global chunk options
opts_chunk$set(echo = FALSE, cache=TRUE, warning=FALSE, tidy=FALSE, fig.width=8, fig.height=6)
```

## 4 Notes Pandoc

### 4.1 Transformer un fichier md en pdf:

```
FILE<-"EquilibreAB" system(paste("pandoc -o", FILE, ".pdf", FILE, ".md", sep=""))
```

## 5 Notes Latex

source: Arnaud Gazagnes, Latex pour le prof. de Maths

### 5.1 Ecriture d'un nombre et virgule

#### 5.1.0.1 La commande `\np` (pp 58) Package *numprint*

`\Sexpr{\np{3,14}}` permet d'écrire la valeur sans qu'apparaisse un blanc entre 3 et 14.

#### 5.1.0.2 La commande `\DecimalMathComma` (pp 58) Mettre cette expression dans le préambule. Ne fonctionne qu'avec l'option *français* du package *babel*.

#### 5.1.0.3 Symbole % Dans un commentaire comme dans xtable utiliser un double échappement:

`\\%`

## 6 R - RStudio

- Tools/project options/Sweave weave Rnw file using *choisir knitr*
- symbole plus ou moins:

`$\pm \Sexpr{x}$` (ne pas oublier les parenthèses)

### 6.1 Utilisation du nom de l'ordinateur pour définir un chemin d'accès:

```
if(as.character(Sys.info()["nodename"]) == "MacBook-Air-de-JCB.local")
  file.reg <- "~/Documents/FEDORU/Codes_regroupement_ORUMIP/Regroupements ORUMiP Thésaurus SFMU.csv" else
  file.reg <- "~/Documents/Resural/FEDORU/Codes_regroupement_ORUMIP/Regroupement_ORUMIP/Regroupements ORUMiP Thésaurus SFMU.csv"
reg <- read.csv(file.reg, skip = 1)
```

### 6.2 Référencer le travail

`citation('lubridate')`

To cite lubridate in publications use:

Garrett Golemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL <http://www.jstatsoft.org/v40/i03/>.

A BibTeX entry for LaTeX users is

```
@Article{, title = {Dates and Times Made Easy with {lubridate}}, author = {Garrett Golemund and Hadley Wickham}, journal = {Journal of Statistical Software}, year = {2011}, volume = {40}, number = {3}, pages = {1-25}, url = {http://www.jstatsoft.org/v40/i03/}, }
```

`citation()`

To cite R in publications use:

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.

A BibTeX entry for LaTeX users is

```
@Manual{, title = {R: A Language and Environment for Statistical Computing}, author = {{R Core Team}}, organization = {R Foundation for Statistical Computing}, address = {Vienna, Austria}, year = {2014}, url = {http://www.R-project.org/}, }
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also ‘citation(“pkgname”)’ for citing R packages.

## 6.3 Librairies

**6.3.0.4 gdata** gdata est utile pour la méthode `drop.levels()` qui supprime les levels inutiles: (ref: [http://rwiki.sciviews.org/doku.php?id=tips:data-manip:drop\\_unused\\_levels](http://rwiki.sciviews.org/doku.php?id=tips:data-manip:drop_unused_levels))

**6.3.0.5 RCurl** Permet de lire des données via HTTP. Avant de l’installer il faut installer la librairie *libcurl4-openssl-dev*:

`sudo apt-get install libcurl4-openssl-dev`

ref: <http://www.omegahat.org/RCurl/FAQ.html>

### 6.3.0.6 Chunks

- pour supprimer les messages inutiles `{r message=FALSE}`
- pour supprimer les warnings: `{r warning=FALSE}`
- pour obtenir l’équivalent de `Sepr`: in the source code I have calculated  $2 + 2 = 4$ . Ref: Getting Started with R Markdown, knitr, and Rstudio, <http://jeromyanglim.blogspot.com.au/2012/05/getting-started-with-r-markdown-knitr.html>

### 6.3.0.7 A NE FAIRE QU’UNE FOIS AU DEBUT DE LA SESSION:

- `rm(list=ls(all=TRUE))` pour vider complètement la mémoire
- `loadhistory(file = “.Rhistory”)`
- `timestamp()`
- A FAIRE SI ON CHANGE DE WORKING DIRECTORY
- sauvegarde: `savehistory(file = “.Rhistory”)`

### 6.3.0.8 Légendes graphes:

- Pour que les légendes de l’axe des Y soient perpendiculaires a ce dernier, rajouter `las = 1`
- Pour que les légendes de l’axe des X soient perpendiculaires a ce dernier, rajouter `las = 2`
- Pour que les légendes soient perpendiculaires aux 2 axes, rajouter `las = 3`
- Par défaut `las = 0`
- pour ne pas encadrer les légendes: `bty = “n”` (bty = box type)

### 6.3.0.9 L'utilisation de la méthode *SweaveInput*

- provoque un erreur si le fichier à inclure comporte des caractères accués (même enregistrés en UTF8)
- lui préférer `<>= @`

Création d'un tableau avec *cbind* et une matrice (et *xtable* pour pdf): voir exhaustivité des données

### 6.3.1 Tables

**6.3.1.1 Tables trop larges** Les tables trop larges pour tenir sur une largeur de page, on la scinde en deux (pas trouvé d'autre solution). Exemple dans `sau_rpu.Rnw`.

**6.3.1.2 Transformer une *table* en *dataframe*:** `as.data.frame.matrix(ma_table)`

**6.3.1.3 Modifier l'entête de colonne d'un tableau** Lorsqu'un tableau n'a qu'une seule colonne, R l'intitule *c* (count) par défaut. Pour changer l'intitulé ajouter `dnn = "%"`:  
`tca<-round(prop.table(table(c, dnn="%"))*100,3)`

**6.3.1.4 Renommer les lignes et/ou les colonnes**

- `rownames(a) <- c("un", "deux")`
- `colnames(a) <- c("un", "deux")`

**6.3.1.5 *tapply* et *xtable*** Affichage avec formatage des chiffres:

```
t<-tapply(as.Date(d1$ENTREE),d1$FINESSE,length)
t2 <- data.frame(as.character(row.names(t)), as.numeric(t))
names(t2) <- c("Etablissement", "RPU")
print(xtable(t2, caption=c("Passages par service d'urgence","Nombre de passages par service d'urgence"))
```

Formatage simple:

```
print(xtable(x),format.args = list(big.mark = " ", decimal.mark = ","))
```

On divise la colonne AGE en 3 groupes puis on calcule l'effectif de chaque groupe avec *tapply*. Pour obtenir un affichage latex avec *xtable*, il faut transformer le vecteur en *dataframe*. On obtient un affichage en colonne. Pour obtenir l'affichage en ligne, utiliser la transposée *t(t)*.

```
tranche_age<-cut(d1$AGE,breaks = c(-1,15,75,max(d1$AGE,na.rm=T)),labels=c("15 ans et moins","16 à 74 ans","75 ans et plus"))
t <- tapply(d1$presence,tranche_age,mean,na.rm=TRUE)
t
xtable(as.data.frame(t))
xtable(as.data.frame(t(t)), caption=c("Durée de passage (mn) en fonction de l'âge","Durée de passage et age"), label="tab:age_dp")
```

Note that `latex.environments = ""` avoids the default option of tables being centered, which in this example would lead to the caption not being directly under the table. Most importantly, `table.placement = NULL` is required to ensure that the default table placement options [ht] being inserted after `\begin{marginable}`. This is necessary because the *marginable* environment does not allow placement options like [ht].

```
> library(xtable)
> x <- matrix(rnorm(6), ncol = 2)
> x.small <- xtable(x, label = 'tabsmall', caption = 'A margin table')
> print(x.small, floating.environment = 'marginable', latex.environments = "", table.placement = NULL)
```

**6.3.1.6 xprop.table(x) [mes\_fonctions.R]** Affiche un vecteur de FACTOR sur deux lignes: n et %

```
xprop.table(table(tranche_age))
a 15 ans et moins 16 à 74 ans 75 ans et plus
n 74739.00 215694.00 49895.00
% 21.96 63.38 14.66
```

**6.3.1.7 sélectionner des variables avec TRUE et FALSE**

- `v<-c("ajax","troie","voiture")`
- `x<-nchar(v)<6`
- `x [1] TRUE TRUE FALSE`
- `v [1] "ajax" "troie" "voiture"`
- `v[x][1] "ajax" "troie"`
- `v[!x][1] "voiture"`

**6.3.1.8 Version:**

```
sessionInfo()
toLatex(sessionInfo())
```

Voir aussi les aides suivantes: - person - citation - bibentry - citEntry - toBibtex(object, ...) - toLatex(object, ...)

**6.3.1.9 Copyright**

```
mtext("@ RESURAL 2013",cex=0.6,side=4,line=-1,adj=0.1)
```

**6.3.1.10 Informations de session** sessionInfo() toLatex(sessionInfo(), locale = FALSE)

**6.3.1.11 xtable** Crée des tables en latex:

<>=

Attention pour results mettre *asis* (et non pas *tex* qui correspond à sweawe)

```
xtable(c, caption = c("intitulé long", "intitulé court"), label = "tab1",align="|l|r|l|r|",digits = c(0,0,2,0,0),
table.placement = "ht",caption.placement = "top")
```

table [6.3.1.17](#), page 7.

*xtable* génère une erreur si l'intitulé d'une colonne est . Dans ce cas il faut transformer les en "NA" et refaire le tableau (voir sortie.rnw par exemple).

Pour supprimer les commentaires commençant par %: print(xtable, comment = FALSE) [stackoverflow](#)

**6.3.1.12 Pour présenter en latex un tableau type summary:** stargazer(as.data.frame(a2[1:length(a)]),summary=TRUE, digits=2,median=TRUE,iqr=TRUE)

```
stargazer(as.data.frame(data$total),summary=TRUE,digit.separator=" ",digits=2, median=TRUE,
iqr=TRUE, title ="Totalité des passages: résumé des principales caractéristiques")
```

**6.3.1.13 Séparateur de milliers:**

`\np{x}` génère une erreur si x est en notation scientifique

**6.3.1.14 Se débarrasser des NA** Procéder en 2 temps: - créer un vecteur de *logical* avec *!is.na* - extraire les valeurs qui ne sont pas de NA à l'aide du vecteur précédent exemple:

```
vector<-c(1,NA,2)
vector
select<-!is.na(vector)
vector[select]
mean(vector)
mean(vector[select])
```

**6.3.1.15 couleur** The argument *col* will set the colours, you could use this in conjunction with an *ifelse* statement `col<- plot(x,y,xlab="PC1",ylab="PC2", col = ifelse(x < 0,'red','green'), pch = 19 )`

**6.3.1.16 Dates** Transformer une date *R* en date *française*:  
`date<-"2013-02-05"`

```
\Sexpr{format(date,format="%d %B %Y")}
```

**6.3.1.17 tab1** Nécessite le package **epicalc** - `a<-tab1(factor(d1$MODE_ENTREE),sort.group = "decreasing", main="Origine des patients (en %)",ylab="Pourcentage",bar.values = "percent")`

- est un objet composé d'un graphique et d'un tableau
- l'intitulé est accessible par `a$first.line<-"Origine des patients"`
- le tableau est accessible via `aoutput.table.Permetd'utiliserxtable(aoutput.table)`.
- valeurs individuelles: `a$output.table[1,1]`
- noms des lignes: `row.names(a$output.table)`
- nom des colonnes: `colnames(a$output.table)`
- franciser le nom des colonnes: `colnames(a$output.table)<-c("Fréquence","Pourcentage","Pourcentage cumul.")`
- `sort.group = "decreasing"`, affiche les colonnes par ordre décroissant.

Pour imprimer avec *xtable*, utiliser `tab1$output.table` :

```
a <- tab1(heure.AVC, missing=FALSE,main="Heures d'admission des AVC", ylab="Fréquence")
colnames(a$output.table)<-c("Fréquence","Pourcentage","Pourcentage cumul.")
print(xtable(a$output.table, caption=c(paste0("Horaires de passages des AVC en ", anc, "."),"Horaire d
```

**6.3.1.18 Supprimer la notation scientifique** Par défaut R transforme tout nombre supérieur à  $10^5$  en notation scientifique. Voir **?option** puis *scipen* pour explication (et <http://yihui.name/knitr/demo/output/>).  
ex: `options(scipen = 6, digits = 2)` Accepté par knitr et Sweave. A mettre en début de programme.

**6.3.1.19 Mettre un espace comme séparateur de milliers** A mettre en début de document. Source: <http://stackoverflow.com/questions/18965637/set-global-thousand-separator-on-knitr> `knit_hooks$set(inline = function(x) { prettyNum(x, big.mark=" ") })`

### 6.3.1.20 tapply et xtable

```
t <- round(tapply(d1$presence,d1$GRAVITE,mean,na.rm=TRUE),2)
p <- round(prop.table(t)*100,2)
r <- rbind(t,p)
rownames(r) <- c("mn","%")
xtable(r, caption=c("Titre long","Titre court"), label = "duree_gravite")
ou
x <- xtable(r, caption=c("Durée de présence et gravité","Durée de présence et gravité"), label = "duree",
print.xtable {x} permet d'utiliser d'autres arguments notamment longtable.
```

### 6.3.1.21 afficher un tableau en Latex

```
\begin{figure}[ht!]
\centering
<<duree_gravite2,echo=FALSE,fig.width=8>>=
boxplot(d1$presence ~ d1$GRAVITE,,outline=F,main="Durée de présence selon la gravité",ylab="durée de p
@
\caption{Durée de passage en fonction de l'âge}
\label{toucan}
\end{figure}
```

La figure~?? (pp.??) montre une photographie de toucan. Le tilde (~) dans l'exemple ci-dessus est un symbole spécial en Latex. Il représente un espace insécable. Il est utile ici parce qu'il garde « figure » et le numéro quel qu'il soit auquel \ref se rapporte comme un tout, et ne les coupera pas sur une ligne ou une page lors de la production du document.

### 6.3.1.22 tracer une ligne joignant les sommets des barres du barplot NOTE TECHNIQUE:

tracer une ligne joignant les sommets des barres du barplot. On utilise lines avec les valeurs suivantes: - x = abscisse des colonnes. Elles sont contenues dans l'objet barplot. On peut les recueillir explicitement par la fonction *str* (str(x)). - y = ordonnées des barres, récupérées avec la fonction *table* qui agglomère les données par mois Voir aussi: <http://www.ats.ucla.edu/stat/r/faq/barplotplus.htm>

### 6.3.1.23 Renommer des levels prédéfini Notamment pour franciser certains tableau:

```
RPU.jour.semaine <- wday(d1$ENTREE,label=T)
levels(RPU.jour.semaine) <- week.short
```

On réordonne les jours pour que la semaines commence le lundi. Source: <http://www.r-bloggers.com/reorder>  
RPU.jour.semaine <- factor(RPU.jour.semaine, levels(RPU.jour.semaine)[c(2:7,1)])

```
table(RPU.jour.semaine)
```

wday retourne le nom du jour en anglais. *week.short* contient la même liste en français. Par défaut le premier jour de la semaine est le dimanche = 1. En réordonnant les niveaux on le place en dernier.

### 6.3.1.24 notes sur les spatialPolygons

```
str(poly) Formal class 'SpatialPolygons' [package "sp"] with 4 slots ..@ polygons :List of 1 .. ..$
:Formal class 'Polygons' [package "sp"] with 5 slots .. ..@ Polygons :List of 1 .. .. ..$
```



```

:Formal class 'Polygon' [package "sp"] with 5 slots .. .. .. ..@ labpt : num [1:2] 1 10 .. ..
.. .. ..@ area : num 1 .. .. .. ..@ hole : logi FALSE .. .. .. ..@ ringDir: int 1 .. ..
.. .. ..@ coords : num [1:5, 1:2] 0.5 0.5 1.5 1.5 0.5 9.5 10.5 10.5 9.5 9.5 .. .. .. .. attr(,
"dimnames")=List of 2 .. .. .. ..$ : chr [1:5] "s1" "s1" "s1" "s1" ... .. .. ..
.. ..$ : chr [1:2] "x" "y" .. .. ..@ plotOrder: int 1 .. .. ..@ labpt : num [1:2] 1 10 .. .. ..@
ID : chr "g1" .. .. ..@ area : num 1 ..@ plotOrder : int 1 ..@ bbox : num [1:2, 1:2] 0.5 9.5
1.5 10.5 .. .. attr(, "dimnames")=List of 2 .. ..$ : chr [1:2] "r1" "r2" .. ..$ : chr [1:2] "min"
"max" ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots .. ..@ projargs: chr NA

```

I'll explain, stepping through the structure breadth first, and backwards:

@proj4string is the coordinate reference system slot @bbox is the bounding box slot @plotOrder is the order to plot the polygons @polygons is the list of Polygons objects.

@polygons [[1]] is the first (and in this case, only) feature. It is an object of class 'Polygons' (non-spatial, since there's no @proj4string in this part of the structure).

@polygons [[1]]@Polygons is the 'Polygons' slot of class 'Polygons', and is a list of rings that make up the feature.

@polygons [[1]]@Polygons[[1]] is an object of class 'Polygon'.

@polygons [[1]]@Polygons[[1]]@coords is the coordinates of the Polygon:

```
poly@polygons[[1]]@Polygons[[1]]@coords x y s1 0.5 9.5 s1 0.5 10.5 s1 1.5 10.5 s1 1.5 9.5 s1 0.5 9.5
```

## 7 Outils de présentation

Sont également utilisables pour le site internet de Resural: - Interactive presentation with slidify and googleVis: comment faire un diaporama en R en incorporant des graphiques interactifs présentés via googlevis et les incorporer dans son site [http://feedproxy.google.com/~r/RBloggers/~3/4w4oN8fMOa4/?utm\\_source=feedburner&utm\\_medium=email](http://feedproxy.google.com/~r/RBloggers/~3/4w4oN8fMOa4/?utm_source=feedburner&utm_medium=email)

## 8 Analyse de la variance (one-way)

Comparaisons de plusieurs moyennes. Le principe est de comparer une variable *qualitative* par une variable *quantitatives* possédant *plusieurs* niveaux. Par exemple on veut comparer l'âge moyen des patients consultants aux urgences (*variable qualitative*) selon le territoire de santé (*variable quantitative à 4 niveaux*).

```

tapply(d1$AGE,d1$secteur,mean, na.rm=TRUE)
tapply(d1$AGE,d1$secteur,sd, na.rm=TRUE)
tapply(d1$AGE,d1$secteur,median, na.rm=TRUE)
boxplot(d1$AGE ~ d1$secteur, ylab="Age", xlab="Territoire de santé", main="Age des consultants selon 1
age <- d1$AGE
territoire <- d1$secteur
mod <- aov(age ~ territoire)
mod
summary(mod)
plotmeans(age ~ territoire, ylab="Age moyen", p=0.9999999, xlab="Territoire de santé", main = "Age moy
TukeyHSD(mod)
par(las=2)
par(mar=c(5,8,4,2))
plot(TukeyHSD(mod))

```

```
tuk <- glht(mod, linfct=mcp(territoire="Tukey"))
par(las=1)
plot(cld(tuk, levels=0.05), col="lightgray")
```

Impression de qualité:

L'intervalle de confiance a été augmenté à  $p=0.9999999$  pour qu'il soit visible. A  $p = 0.95$  on ne les voit pas car inférieurs à 1 et des messages d'avertissement sont générés. La méthode **plotCI** fait la même chose (voir notamment le premier exemple de cette méthode). “ Explications: - les 3 premières lignes explorent le pb: on calcule la moyenne, l'écart-type et le boxplot pour chaque secteur (territoire de santé) - les 3 lignes suivantes constituent l'Anova proprement dite: `anova <- aov(var.quantitative ~ var.qualitative)` - la ligne `_plotmeans()` trace un graphique avec les moyennes des groupes et l'écart-type. La taille des groupes étant très importante, les SD sont très petits. - TukeyHSD calcule les différences par groupe pris 2 à 2 et dessine le graphe associé.

ref: R in action pp 225-230

## 9 Boxplot

extrait de Uffler2014. Dessine une boxplot et rajoute les moyennes sous forme de points et de textes:

```
means <- tapply(motivation, CSP, mean, na.rm=TRUE)
means
boxplot(motivation ~ CSP, xlab="CSP", ylab="Score de motivation", main="Motivation selon la CSP", col="red",
points(1:length(means), means, pch = 23, cex = 0.75, bg = "red")
text(1:length(means) , means + 10, labels = formatC(means, format = "f", digits = 1), pos = 1, cex = 0.9
```

### 9.1 Pour supprimer la notation scientifique:

```
options(scipen = 6, digits = 2)
```

### 9.2 Pour imposer un préfixateur de milliers:

```
knit_hooks$set(inline = function(x) { prettyNum(x, big.mark=" ") })
```

### 9.3 Bar chart

Bar charts with percentage labels but counts on the y axis

```
library(ggplot2) library(scales) perbar=function(xx){ q=ggplot(data=data.frame(xx),aes(x=xx))+
geom_bar(aes(y = (..count..)),fill="orange")+ geom_text(aes(y = (..count..),label = ifelse((..count..)==0,"",scales::percent((..count../sum(..count..))),stat="bin",colour="darkgreen")) q } perbar(mtcars$ cyl)perbar(mtcars$wt)+coord_flip()
```

### 9.4 Histogramme de qualité

```
heure.AVC <- hour(AVC$ENTREE)
hist(heure.AVC, breaks=24, xlab="Heure de la journée", main="Répartition des AVC dans la journée", ylab="Nombre d'AVC par heure")
```

## 9.5 Tracer 2 histogrammes en miroir

[source](#)

```
df = data.frame(x = rnorm(100), x2 = rnorm(100, mean=2))
h1 = hist(df$x, plot=FALSE)
h2 = hist(df$x2, plot=FALSE)
h2$counts = - h2$counts
hmax = max(h1$counts)
hmin = min(h2$counts)
X = c(h1$breaks, h2$breaks)
xmax = max(X)
xmin = min(X)
plot(h1, ylim=c(hmin, hmax), col="green", xlim=c(xmin, xmax))
lines(h2, col="blue")
```

Refusal of Care in the Prehospital Setting

## 9.6 Packages installés

source: <http://www.ats.ucla.edu/stat/r/faq/packages.htm>

ou <http://stackoverflow.com/questions/2169118/installing-all-cran-packages-that-are-not-already-installed>

Liste de tous les packages installés sur la machine:

```
a <- installed.packages()
a[1:5, 1]
```

Uniquement la liste des noms

```
a <- .packages(all.available = TRUE)
save(a, file = "my_RPackages.R")
rm(a)
load("my_RPackages.R")
```

Pour installer des packages:

```
install.packages(a)
```

Tous les packages disponibles:

```
a <- available.packages()
```

## 9.7 ##### Franciser et réorganiser la semaine

La se

```

library("gdata")
w<-wday(AVC$ENTREE,label=TRUE)
levels(w) <- week.short
w <- reorder(w,new.order = french.short.week)
wt <- table(w, dnn = "Nombre d'AVC selon le jour de la semaine")
pwt <- round( prop.table(table(w, dnn = "Pourcentage d'AVC selon le jour de la semaine"))*100,2)
a <- rbind(wt, pwt)
rownames(a) <- c("fréquence", "p.cent")
xtable(a, caption = c("Distribution des AVC en fonction du jour de la semaine. La fréquence quotidienne

```

## 10 # radial.plot

```
radial.plot(1-a,rp.type="p",radial.pos=NULL,labels= names(a),line.col=fadeBlue,poly.col=fadeBlue,radial

```

Le paramètre **mar** permet de faire de la place pour le titre et les labels.

## 11 Dataframe

### 11.1 Réordonner les colonnes

Il suffit de recréer le DF en modifiant l'ordre des colonnes

```
df[,c(1,2,3,4)]
```

```
df2[,c(1,3,2,4)]
```

```
pour sauvegarder: write.csv(df2, file="somedf.csv")
```

[source](#)

## 12 Remarques sur “format”

```

a <- 123456.789
a
format(a)
# nombre de chiffres significatifs
format(a, digits = 20)
# nombre MINIMUM de chiffres après la virgule
format(a, nsmall = 3)
format(a, nsmall = 3, big.mark = ' ')
format(a, nsmall = 3, big.mark = ' ', decimal.mark = ",")

```

## 13 graphe avec 2 axes y pour les passages et le taux d’hospitalisation.

```

# premier graphique
plot(d.xts$passages, minor.ticks = FALSE, main = "")
# permet de dessiner un second graphique avec ses propres paramètres

```

```
par(new = T)
# second graphique
plot(d.xts$taux, axes = F, ylim = c(0, 100), col = "blue", main="")
axis(4, ylim = c(0, 100), col = "blue" ) # utilise l'axe de droite. Prévoir plus de marge
```

### 13.1 Ajouter une légende à un graphique

```
legend("topleft", legend = c("Passages","Taux d'hospitalisation"), col = c("black", "blue"), lty = 1, b
```

### 13.2 Transformer Latin 1 (windows) en UTF8 (linux)

La plupart des fichiers disponibles via Open Data ont été préparés avec windows et sont de fait en codage Latin1.

```
# correction des caractères accentués (nom descommunes), les données provenant de windows...
# si les données sont de type factor, il faut d'abord les transcoder en character
b <- as.character(cp67@data$LIB)
Encoding(b) <- "latin1"
cp67@data$LIB <- b
```

### 13.3 %in% et !%in%

L'expression:

```
df1$id %in% idNums1
```

produit un vecteur logique. Pour le négativer il faut négativer la totalité du vecteur:

```
!(df1$id %in% idNums1)
```

[source](#)