

RPU Quotidiens

JcB

10/01/2014

Contents

Fichier RPU quotidien	1
Descriptif	1
Commentaires:	4
En pratique	4
si un seul fichier	4
Si on a une collection de fichiers:	4
exhaustivité des données	5
Résumé activité	6

Fichier RPU quotidien

Depuis février 2014, Alsace e-sante transmet quotidiennement un fichier contenant les RPU des 7 derniers jours (j-7 à j-1). Les données correspondant à J-7 sont considérées comme consolidées. Elles peuvent être extraites et stockées. Les données sont transmises de manière habituelle, c'est à dire un fichier .sql qu'il faut transcoder en R pour le nettoyer avant stockage.

Au mois de mai 2014 la clinique des 3 frontières (C3F) a changé de N°FINESS. (voir le paragraphe C3F)

Méthode rapide: voir En Pratique

Descriptif

1. Le fichier des données est récupéré sur le serveur de test des HUS. Il est déposé dans le dossier de stockage (/home/jcb/Documents/Resural/Stat Resural/Archives_Sagec/dataQ) et dézippé.
2. le nom du fichier est construit de la manière suivante:
 - `date.jour <- "2014-02-21"`
 - `file <- paste0("rpu_", date.jour, "_dump.sql")`
 - *date.jour est du type AAAA-MM-JJ*
2. le fichier est ensuite transféré dans la base de données **archives** dans la table ****RPU__**** via R
 - il est important que le répertoire de travail temporaire soit positionné dans le dossier *dataQ*

```
wd <- getwd()
setwd("~/Documents/Resural/Stat Resural/Archives_Sagec/dataQ")
system(paste0("mysql -u root -pmarion archives < ", file))
setwd(wd)
```

3. Lecture des données dans R

```
library("RMySQL")
con<-dbConnect(MySQL(),group = "archives")
rs<-dbSendQuery(con,paste("SELECT * FROM RPU__ ",sep=""))
dx<-fetch(rs,n=-1,encoding = "UTF-8")
max(dx$ENTREE)
min(dx$ENTREE)
```

4. nettoyage des données

- suppression de la colonne 16: `dx<-dx[,-16]`
- transcodage des FINESS (vérification nombre hôpitaux)
- transformation en facteurs
- création d'une colonne AGE (alertes age < 0 et age > 120)

5. sauvegarde des données

- jour à sauvegarder: `jour <- as.Date(min(dx$ENTREE))`
- `dday <- dx[as.Date(dx$ENTREE) == jour,]`
- fichier du jour: `write.table(dday, paste0(date.jour,".csv"), sep=',', quote=TRUE, na="NA", row.names=FALSE,col.names=TRUE)`
- fichier général: `write.table(dday, "RPU2014.csv", sep=',', quote=TRUE, na="NA", append = TRUE, row.names=FALSE,col.names=TRUE)`

6. fonctions helpers

`source("quot_utils.R")` ou `source("Preparation/RPU Quotidiens/quot_utils.R")` en mode console.

- **rpu_jour**: fonction principale. En entrée on donne la date ISO souhaitée et en sortie retourne un dataframe avec les données correspondantes. Le WD doit pointer sur le dossier contenant le fichier .sql correspondant. Ce fichier doit être dézipé.
- **finess2hop**: transforme le code FINESS en nom court d'hôpital
- **parse_rpu**:

séquence:

- `date.jour <- "2014_02"`
- `dx <- parse_rpu(date.jour)`
- `dx$FINESS <- as.factor(finess2hop(dx$FINESS))`
- `summary(dx$FINESS)` fait un décompte des RPU par établissement sur la période => permet de vérifier si anomalies quantitatives. Suppose de disposer d'un historique moyenne, écart-type par type de jour.
- `dx <- rpu2factor(dx)`

```

#' Méthode générale
#' Préalable: disposer d'une base de donnée MySQL avec une table appelée "archives". Cette base doit être
#'@ data date.jour nom du fichier. Pour une utilisation courante il s'agit de la date du jour au format
parse_rpu <- function(date.jour){
  library("RMySQL")
  file <- paste0("rpu_", date.jour, "_dump.sql")
  wd <- getwd()
  setwd("~/Documents/Resural/Stat Resural/Archives_Sagec/dataQ")
  system(paste0("mysql -u root -pmarion archives < ", file))
  con<-dbConnect(MySQL(),group = "archives")
  rs<-dbSendQuery(con,paste("SELECT * FROM RPU__ ",sep=""))
  dx<-fetch(rs,n=-1,encoding = "UTF-8")
  dx<-dx[,-16]
  dx$FINESS <- as.factor(finess2hop(dx$FINESS))

  dx$AGE<-floor(as.numeric(as.Date(dx$ENTREE)-as.Date(dx$NAISSANCE))/365)

  dx$EXTRACT <- as.Date(dx$EXTRACT)
  setwd(wd)
}

#' Transformation du code Finess et nom court d'hôpital
finess2hop <- function(a){
  # a<-dx$FINESS
  a[a=="670000397"]<-"Sel"
  a[a=="680000684"]<-"Col"
  a[a=="670016237"]<-"Odi"
  a[a=="670780204"]<-"Odi" # Finess juridique
  a[a=="670000272"]<-"Wis"
  a[a=="680000700"]<-"Geb"
  a[a=="670780055"]<-"Hus"
  a[a=="670000025"]<-"Hus" # NHC
  a[a=="670783273"]<-"Hus" # HTP
  a[a=="680000197"]<-"3Fr"
  a[a=="680000627"]<-"Mul"
  a[a=="670000157"]<-"Hag"
  a[a=="680000320"]<-"Dia"
  a[a=="680000395"]<-"Alk"
  a[a=="670000165"]<-"Sav"
  a[a=="680000494"]<-"Ros"
  a[a=="670780162"]<-"Dts"
  a[a=="670780212"]<-"Ane"
  a[a=="680000601"]<-"Tan"
  return(a)
}

```

contrôles quotidiens

- nlevels(dx\$FINESS) si différent de 14 => problème
- nb moyen et écart-type de RPU par établissement et par jour

```

date1 <- "2014-03-01"
date2 <- "2014-03-05"

```

```
p <- seq(as.Date(date1), as.Date(date2), 1)
for(i in 1:length(p)){
  x <- parse_rpu(p[i])
  table(x$FINESS, as.Date(x$ENTREE))
}
```

Commentaires:

```
r <- table(as.Date(a$ENTREE), a$FINESS)
r <- r[,-13] # supprime la colonne 13 qui est totalement vide ?
r
```

- altkirch: toujours des trous inexplicés: 1/1, 5/1, 11 et 12/1, 16/1, 18/1, 2/3
- mulhouse: 15/1, 7/2, 5-6-7/3 zéro rpu
- ste odile: 16 au 31/1 pas de rpu
- sélestat: 22 et 23/2 pas de RPU
- diaconat strasbourg: 1-2-3-4/3 puis plus rien
- roosvelt: depuis le 5/2 OK

En pratique

- dézipper le fichier du jour dans */home/jcb/Documents/Resural/Stat Resural/Archives_Sagec/dataQ*
- charger le fichier **quot_utils.R** pour disposer des routines
- répéter l'étape **rj** autant de fois qu'il y a de fichiers à analyser
- assembler les fichiers avec **assemble()**

```
source("Preparation/RPU Quotidiens/quot_utils.R")
```

si un seul fichier

```
rj <- rpu_jour("2014-11-18")
dj <- assemble(comment = TRUE)

# Exhaustivité des RPU du jour:
table(as.Date(rj$ENTREE), rj$FINESS)
```

Si on a une collection de fichiers:

NB: SUPPRIMER LE FICHIER */home/jcb/Documents/Resural/Stat Resural/Archives_Sagec/dataQ/arch*

```
file.to.delete <- "/home/jcb/Documents/Resural/Stat Resural/Archives_Sagec/dataQ/archivesCsv/rpu2014.da
file.remove(file.to.delete)
```

```
date1 <- "2014-12-08"
date2 <- "2015-01-07"
```

```
mc <- substr(date1, 6, 7)
ac <- substr(date1, 1,4)
```

```

p <- seq(as.Date(date1), as.Date(date2), 1)
for(i in 1:length(p)){
  dx <- rpu_jour(p[i])
}
dx <- assemble(comment = TRUE)

min(as.Date(dx$ENTREE))
max(as.Date(dx$ENTREE))

```

Sélectionner une période particulière (ie. mai 2014)

```

d12 <- dx[as.Date(dx$ENTREE) >= "2014-12-01" & as.Date(dx$ENTREE) < "2015-01-01",]
max(as.Date(d12$ENTREE))
min(as.Date(d12$ENTREE))
d12 <- normalise(d12)
save(d12, file="rpu2014d12_provisoire.Rda")
rm(dx)

load("rpu2014d0111.Rda")

dx <- rbind(dx, d12)
min(as.Date(dx$ENTREE))
max(as.Date(dx$ENTREE))

save(dx, file="rpu2014d0112_provisoire.Rda")

table(as.Date(d12$ENTREE), d12$FINESS)

```

Eventuellement sauvegarder au format .Rda

```

dx <- normalise(dx)
dx$FINESS <- factor(dx$FINESS) # supprime les facteurs vides
dx <- dx[dx$ENTREE >= "2014-04-01" & dx$ENTREE < "2014-05-01",]
save(dx, file="rpu2014d04_provisoire.Rda")

```

et assembler le tout (d1 = fichier .Rda des mois précédents)

```

a <- rbind(d1,dx)
save(a, file="rpu2014d0103_provisoire.Rda")

```

Pour fabriquer les courbes interactives d'activité, voir le projet **dygraph**.

exhaustivité des données

On forme une table en croisant FINESS et ENTREE:

```

dx$FINESS <- factor(dx$FINESS) # supprime les facteurs vide
rpu <- table(as.Date(dx$ENTREE), dx$FINESS)

```

write.csv(rpu, file="exhaustivite_rpu.csv") enregistre la table au format .csv mais à la différence de write.table, une case vide est ajoutée dans le header pour éviter le décalage des colonnes.

Préparation des fichiers pour Dygraph -> voir *Préparation des fichiers pour Dygraph.Rmd*

Résumé activité

On crée un tableau Finess x Jour de l'année permettant de voir rapidement où sont les “trous”. A partir du tableau **rpu** on crée un dataframe **a** comptant deux colonnes: la date du jour et le nombre de RPU correspondants pour l'ensemble des SU d'Alsace. Ce dataframe peut être utilisé pour **Dygraph**.

On peut aussi l'utiliser pour tracer le graphe correspondant.

```
“{ activite}
```

```
s <- rowSums(rpu) b <- rownames(rpu) a <- as.data.frame(cbind(b,s)) m <- rowMeans(rpu)
```

```
colnames(a) <- c(“Date”, “RPU”) aDate <- as.Date(aDate) aRPU <- as.numeric(as.character(aRPU))  
# transforme les facteurs en nombre
```

```
plot(aDate, aRPU, type=“l”, ylab=“nombre de RPU”, xlab=paste0(“Année”, year(Sys.Date())),  
main=“Activité des SU d'Alsace en nombre de RPU”)
```

```
library(“xts”) library(“lubridate”)
```

```
x <- as.xts(aRPU, aDate)
```

```
z <- as.zoo(x) plot(z, col=“blue”, ylab=“nombre de RPU”, xlab=paste0(“Année”, year(Sys.Date())),  
main=“Activité des SU d'Alsace en nombre de RPU”) lines(rollmean(z, 7), col=“red”)
```

```
plot.xts2(x, ylab=“nombre de RPU”, xlab=paste0(“Année”, year(Sys.Date()))), main=“Activité des SU  
d'Alsace en nombre de RPU”, col=“blue”, minor.ticks = FALSE) lines(rollmean(x, 7), col=“red”) “ Activité  
2013-2014 =====
```

Voir **Activités_2013-2014.Rmd**