

Date de naissance, âge et sexe

JcB

31/10/2014

Contents

1	Introduction	1
1.1	Recommandations FEDORU 2014	1
1.2	Tranches d'âge pédiatriques:	2
1.3	Tranches d'âge adultes	4
2	Sexe (RPU)	6
3	Résumé RPU	7
4	Population d'Alsace	8
4.1	Population alsacienne:	9
4.2	Tranches d'âge	13
4.3	Comparaison des sex-ratio (RPU - pop.Alsacienne)	13
5	Taux de passage	15

1 Introduction

Ce document adapte les recommandations de la FEDORU ¹.

La **date de naissance** du patient, attendue au format "JJ/MM/AAAA" au sein des RPU, est automatiquement croisée avec la date d'entrée aux urgences pour permettre le calcul de l'âge. Les âges de plus de 120 ans sont considérés comme des valeurs aberrantes et sont exclus de l'analyse. Il en va de même pour les âges négatifs.

Le **sexe** sont au format "M/F/I".

1.1 Recommandations FEDORU 2014

La FEDORU recommande l'harmonisation des analyses des données selon les formats suivants:

- Age = date d'entrée - date de naissance
- Critères d'exclusion: age négatif et age > 120 ans (43800 jours)
- nourissons < 28 jours

¹Collecte et usages des RPU. Recommandations sur la production, les définitions, la qualité et l'exploitation des données des Résumés de Passage aux Urgences. FEDORU 2014 - V01-10/2014

- Tranches d'âge pour les moins de 18 ans
- < 28 jours
- [28 jours à 1 an[
- [1 an à 5 ans[
- [5 ans à 10 ans[
- [10 à 15 ans[
- [15 ans à 18 ans[
- Tranches d'âge pour les adultes:
- [18-30ans[
- [30-45ans[
- [45-65ans[
- [65-75ans[
- [75-85ans[
- 85ans

ref: [Introduction à R] pp 68 pour définir les intervalles

```
ped_jours <- c(1, 28, c(1, 5, 10, 15, 18) * 365) # bornes pour la pédiatrie en jour
ped_jours
```

```
## [1] 1 28 365 1825 3650 5475 6570
```

```
adultes_jours <- c(18, 30, 45, 65, 75, 85, 120) * 365
adultes_jours
```

```
## [1] 6570 10950 16425 23725 27375 31025 43800
```

```
adultes_ans <- c(18, 30, 45, 65, 75, 85, 120)
```

```
library(lubridate)
load("~/Documents/Resural/Stat Resural/RPU_2014/rpu2014d0112_c.Rda")
dx <- d14
N <- nrow(dx)
```

1.2 Tranches d'âge pédiatriques:

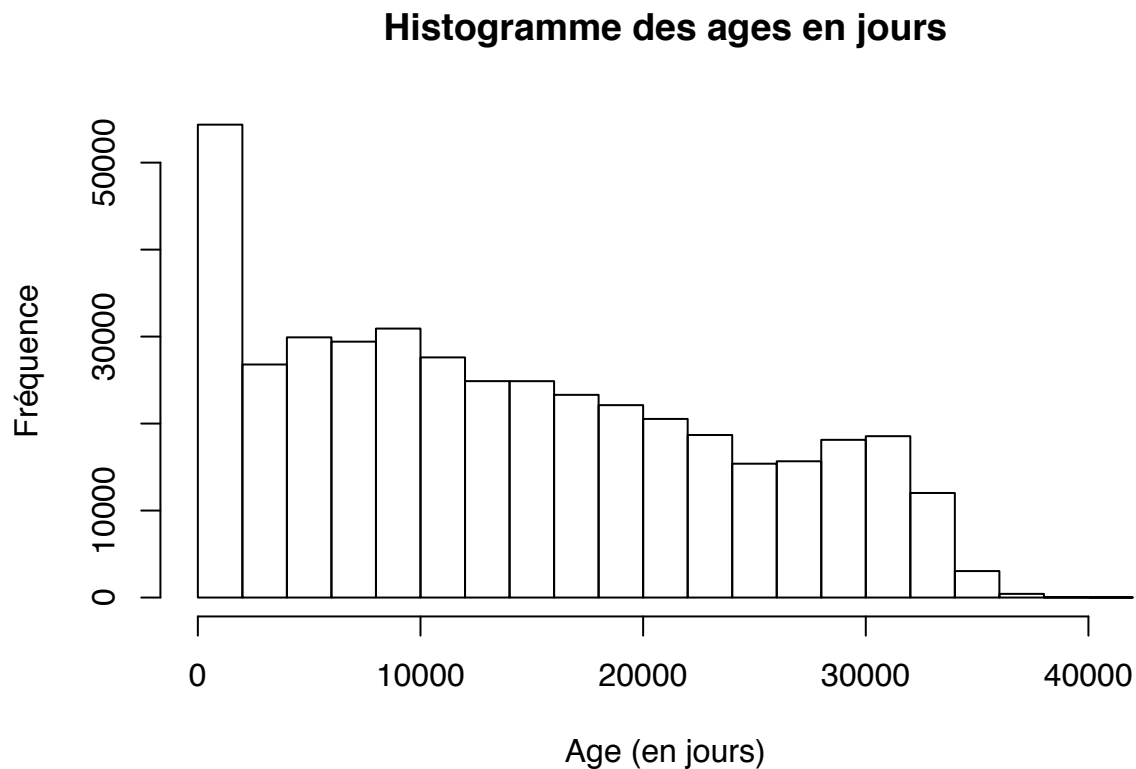
```
# age est un objet de type difftime exprimé en jours
age <- as.Date(dx$ENTREE) - as.Date(dx$NAISSANCE)
age <- as.numeric(age)

# on retire les valeurs aberrantes: mois de 1 jour et plus de 120 ans
age <- age[age > 1 & age <= 365*120]
age_an <- round(age/365, 0)

# nombre de valeurs aberrantes
N - length(age)
```

```
## [1] 36
```

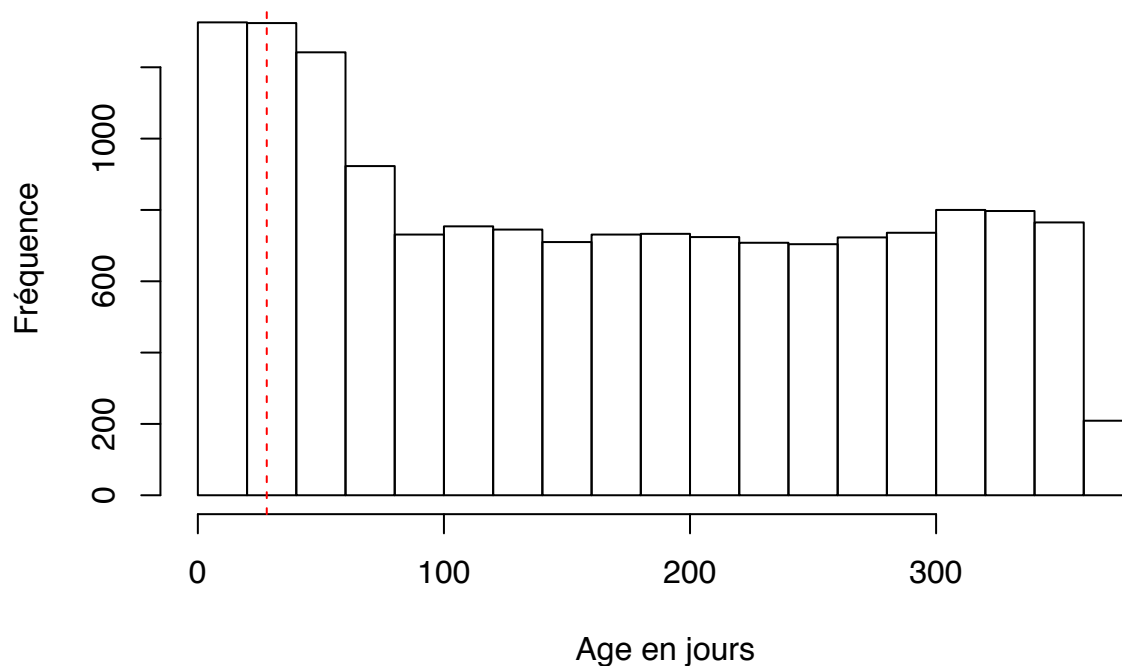
```
# histogramme de age
hist(age, main="Histogramme des ages en jours", xlab = "Age (en jours)", ylab = "Fréquence")
```



```
# les moins de 1 an = age1
age1 <- age[age < 366]
n_age1 <- length(age1) # nombre de moins d'un an
p_age1 <- round(length(age1)*100/N, 2) # proportion de moins de 1 an
hist(age1, main = "Histogramme des moins de 1 an", xlab = "Age en jours", ylab = "Fréquence")
abline(v=28, lty=2, col="red") # limite nourissons
```

```
# nourissons age < 28 jours
nour <- age[age < 28]
n_nour <- length(nour)
```

Histogramme des moins de 1 an



```
# classes FEDORU pour la pédiatrie. La borne inférieure est incluse (include.lowest) et la borne sup es
lab <- c("<28j", "28j-1an[", "1-5ans[", "5-10ans[", "10-15ans[", "15-18ans[")
ped <- cut(age, ped_jours, include.lowest = TRUE, right = FALSE, labels = lab)

table(ped)
```

```
## ped
##      <28j  28j-1an[   1-5ans[   5-10ans[  10-15ans[  15-18ans[
##      1791   13554    36287    24738    27012    15819
```

```
barplot(table(ped), main = "Pédiatrie")
```

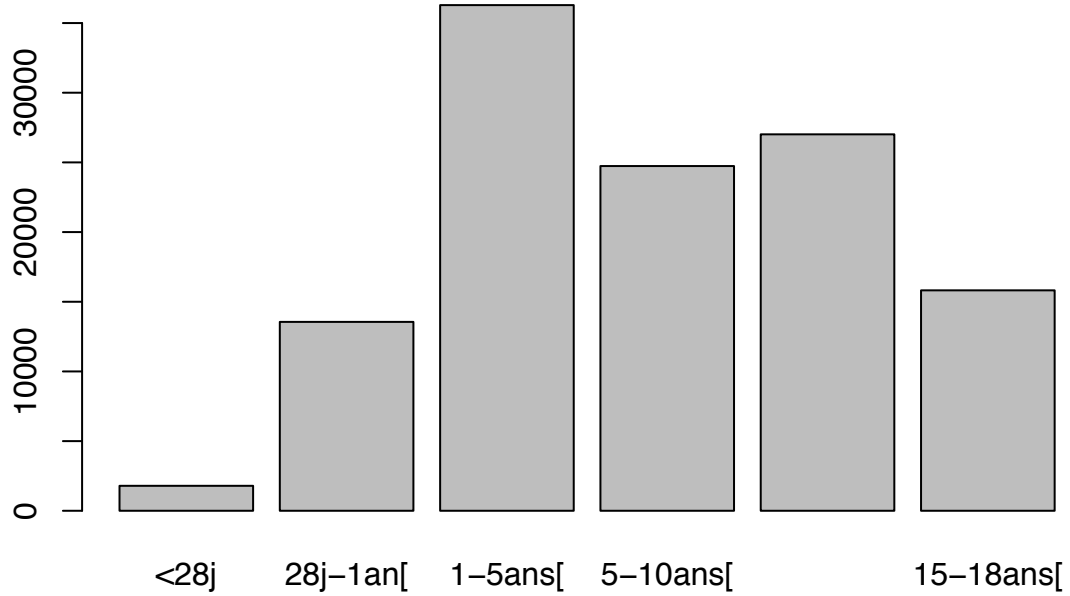
1.3 Tranches d'âge adultes

```
lab_adultes <- c("18-30ans", "30-45ans", "45-65ans", "65-75ans", "75-85ans", ">85ans")
adultes <- cut(age, adultes_jours, include.lowest = TRUE, right = FALSE, labels = lab_adultes)
adultes_rpu <- table(adultes)

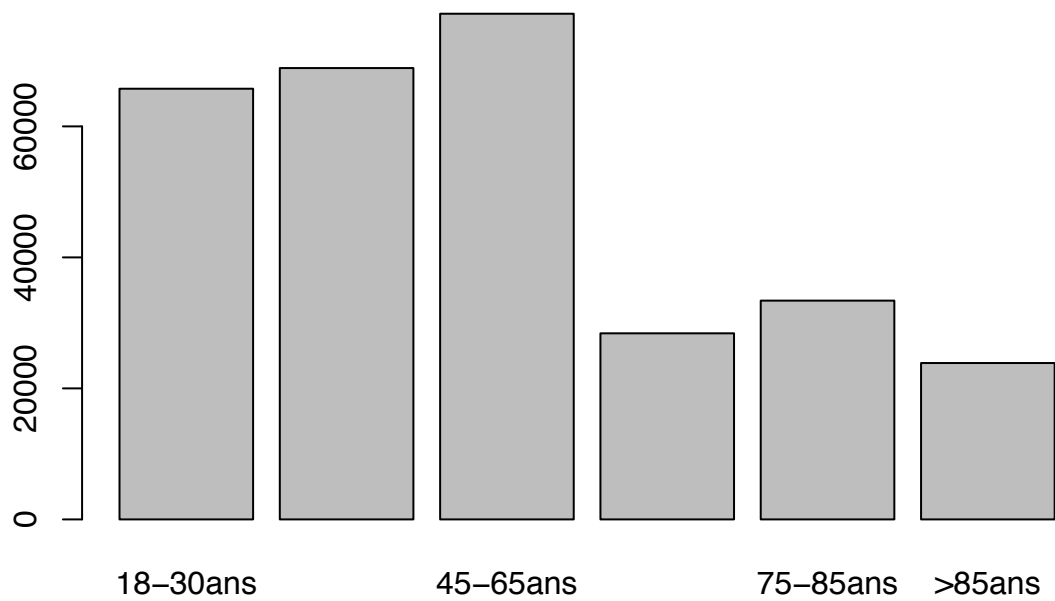
barplot(adultes_rpu, main="RPU adultes en fonction des classes d'âge de la FEDORU")
```

```
# idel en %
barplot(adultes_rpu*100/N, main="RPU adultes en fonction des classes d'âge de la FEDORU")
```

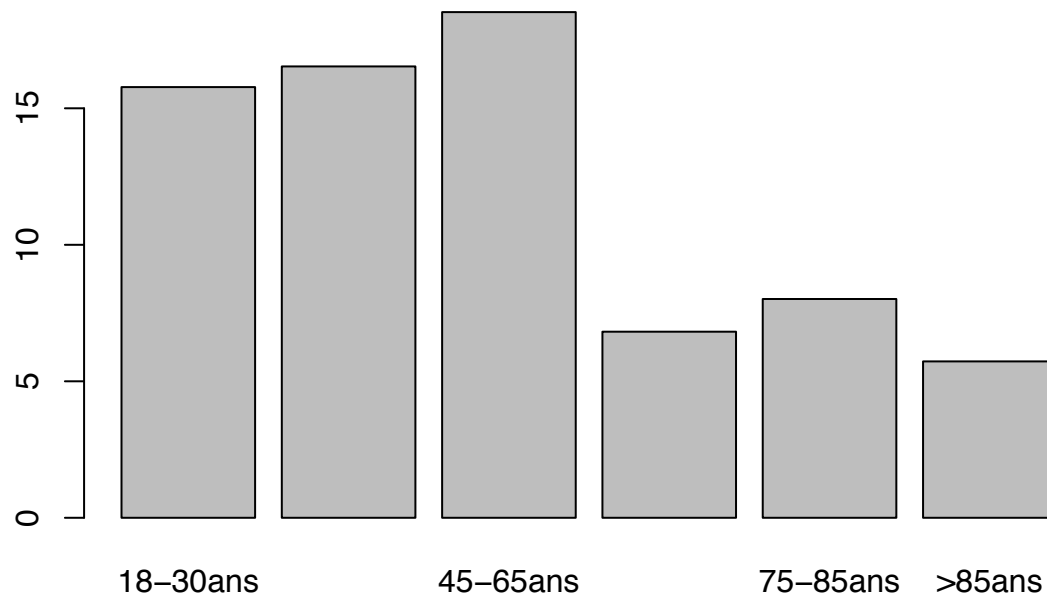
Pédiatrie



RPU adultes en fonction des classes d'âge de la FEDORU

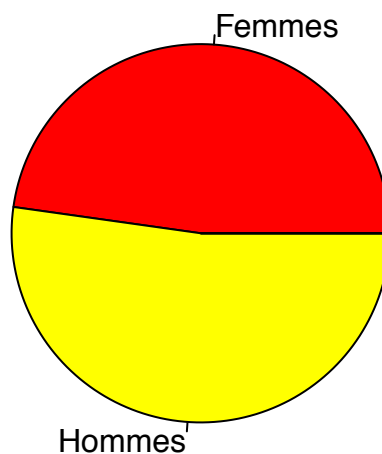


RPU adultes en fonction des classes d'âge de la FEDORU



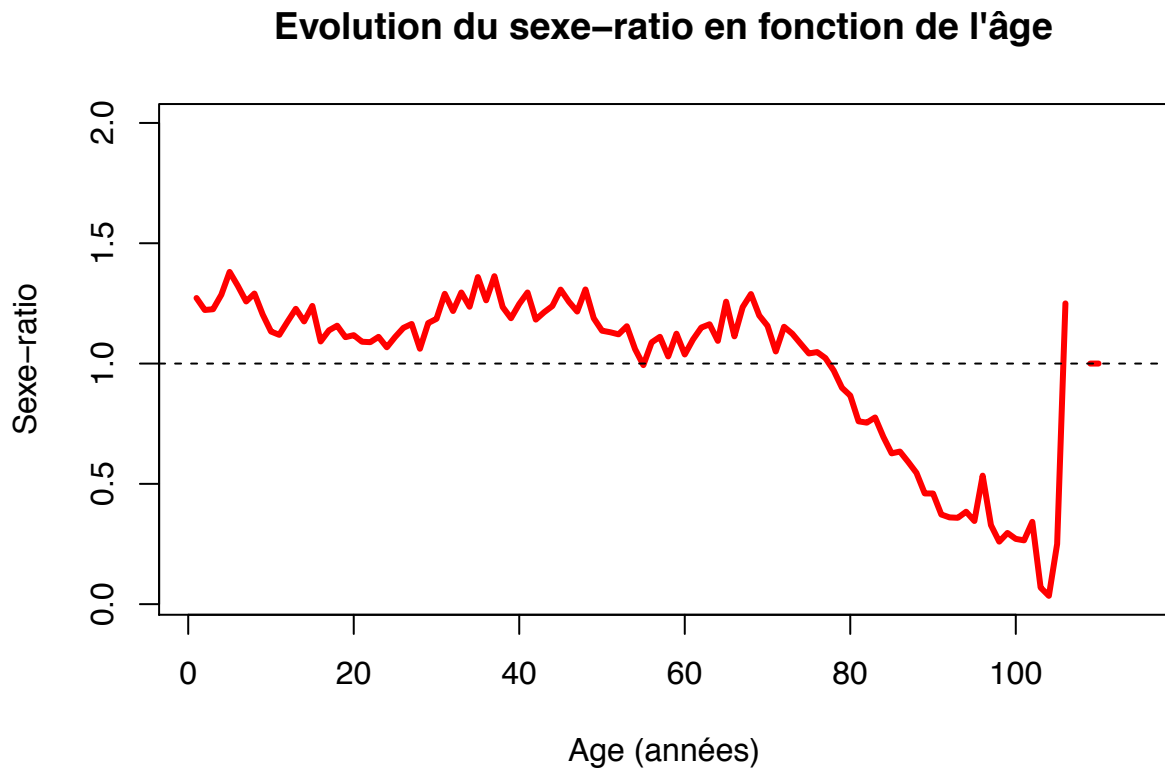
2 Sexe (RPU)

```
sexe <- table(dx$SEXE)
h_rpu <- sexe[2]
f_rpu <- sexe[1]
sex_ratio <- round(h_rpu/f_rpu, 2)
r_masculinite <- round(h_rpu/(h_rpu + f_rpu), 2)
pie(sexe, col=c("red", "yellow"), labels=c("Femmes", "Hommes"))
```



```
# calcul du vecteur sex-ratio: pour chaque tranche d'age de 1 an, on calcule de sr. On ne retient que l
a <- tapply(as.Date(dx$ENTREE), list(dx$SEXE, dx$AGE), length)
a <- a[1:2,]
sr_age_rpu <- a[2,]/a[1,]
```

```
plot(sr_age_rpu, type="l", main="Evolution du sexe-ratio en fonction de l'âge", ylab="Sexe-ratio", xlab="Age (années)", lty=2)
abline(h=1, lty=2)
```



- nombre d'hommes 217 617
- nombre de femmes 199 110
- sex-ratio 1.1
- rapport de masculinité 0.52

3 Résumé RPU

Variables	intitulé en clair	valeur
N	nombre total de RPU	416 733
age	ages des consultants en jours de 1 jour à 120 ans	<i>vecteur</i>
age_an	age des consultants en années	<i>vecteur</i>
sex_ratio	sexe-ratio	1.1
r_masculinite	taux de masculinité = nb hommes / (hommes + femmes)	0.52
sr_age_rpu	vecteur des sex-ratio	<i>vecteur</i>
h_rpu	nombre d'hommes	217 617
f_rpu	nombre de femmes	199 110
n_age1	nombre de moins d'un an	15 385

Variables	intitulé en clair	valeur
p_age1	proportion de moins d'un an	3.7
n_nour	nombre de nourrissons (< 28 jours)	1 791
adultes	vecteur ages > 18 ans selon FEDORU	<i>vecteur</i>
adultes_rpu	table(adultes)	<i>table</i>
adultes_jours	bornes intervalles adulte en jours	<i>vecteur</i>
adultes_ans	bornes intervalles adulte en années	<i>vecteur</i>
lab_adultes	vecteur de labels pour tranches âge	
ped_jours	bornes intervalles ped. en jours	<i>vecteur</i>
ped	vecteur des tranches d'âge pédiatriques	<i>vecteur</i>
lab	vecteur de labels pour table(ped)	

4 Population d'Alsace

La comparaison de la pyramide des âges de la patientèle des urgences à celle de la population de la même zone géographique est également informative. Elle nécessite pour cela de connaître la structure de la population en Alsace. Pour cela on utilise des données de l'INSEE.

source: http://www.insee.fr/fr/ppp/bases-de-donnees/donnees-detaillees/rp2011/tab-detailles/td-population-11/BTT_TD_POP1B_2011.zip

Ce fichier recense pour chaque commune de France la population par **sexe** et **tranches d'âges de 1 an** de 0 à 100 ans. La version *POP1B2011_* est le fichier légal pour 2014.

Le fichier dézippé fait 220 Mo. Il est stocké à l'adresse suivante:

Après plusieurs essais, la meilleure méthode pour lire ce fichier est d'utiliser **read.csv2**, le séparateur étant le **point-virgule**. L'utilisation de *encoding* est à proscrire car cela crée des fichiers corrompus. L'inconvénient de la méthode est de produire un codage anormal des caractères accentués qu'il faut corriger:

expression	symbole
\xe8	è
\xe9	é

Les instructions qui suivent décrivent la procédure pour lire ce fichier et en extraire les données propres à l'Alsace. Les fichiers suivants sont créés et peuvent être chargés directement pour exploiter les données locales:

fichier	Données
pop_france_2014.Rda	ensemble de la population française
pop_als_2014.Rda	ensemble de la population alsacienne
pop67_2014.Rda	ensemble de la population du Bas-Rhin
pop68_2014.Rda	ensemble de la population du Haut-Rhin


```

file <- "/home/jcb/Documents/Resural/Stat Resural/population_alsace/BTT_TD_POP1B_2011.txt"
pop <- read.csv2(file)

# remplacer les caractères anormaux
pop$LIBGEO <- as.character(pop$LIBGEO)      # transforme le facteur en caractères
pop$LIBGEO <- gsub("\xe8","è", pop$LIBGEO)  # remplacement des caractères anormaux
pop$LIBGEO <- gsub("\xe9","é", pop$LIBGEO)

save(pop, file = "pop_france_2014.Rda")      # toute la France

names(pop)

# on extrait les populations d'Alsace

# population du 67. On supprime les levels exédentaires
pop67_2014 <- pop[substr(pop$CODGEO, 1, 2) == 67,]
pop67_2014$LIBGEO <- factor(pop67_2014$LIBGEO)
pop67_2014$CODGEO <- factor(pop67_2014$CODGEO)

# population du 68. On supprime les levels exédentaires
pop68_2014 <- pop[substr(pop$CODGEO, 1, 2) == 68,]
pop68_2014$LIBGEO <- factor(pop68_2014$LIBGEO)
pop68_2014$CODGEO <- factor(pop68_2014$CODGEO)

# population de la région Alsace
pop_als_2014 <- pop[substr(pop$CODGEO, 1, 2) == 67 | substr(pop$CODGEO, 1, 2) == 68,]
pop_als_2014$LIBGEO <- factor(pop_als_2014$LIBGEO)
pop_als_2014$CODGEO <- factor(pop_als_2014$CODGEO)

# Sauvegarde des données
save(pop_als_2014, file="pop_als_2014.Rda")
save(pop67_2014, file="pop67_2014.Rda")
save(pop68_2014, file="pop68_2014.Rda")

# Chargement des données
load("pop68_2014.Rda")
load("pop67_2014.Rda")
load("pop_als_2014.Rda")

```

4.1 Population alsacienne:

```

path <- "../"
load(paste0(path, "pop68_2014.Rda"))
load(paste0(path, "pop67_2014.Rda"))
load(paste0(path, "pop_als_2014.Rda"))

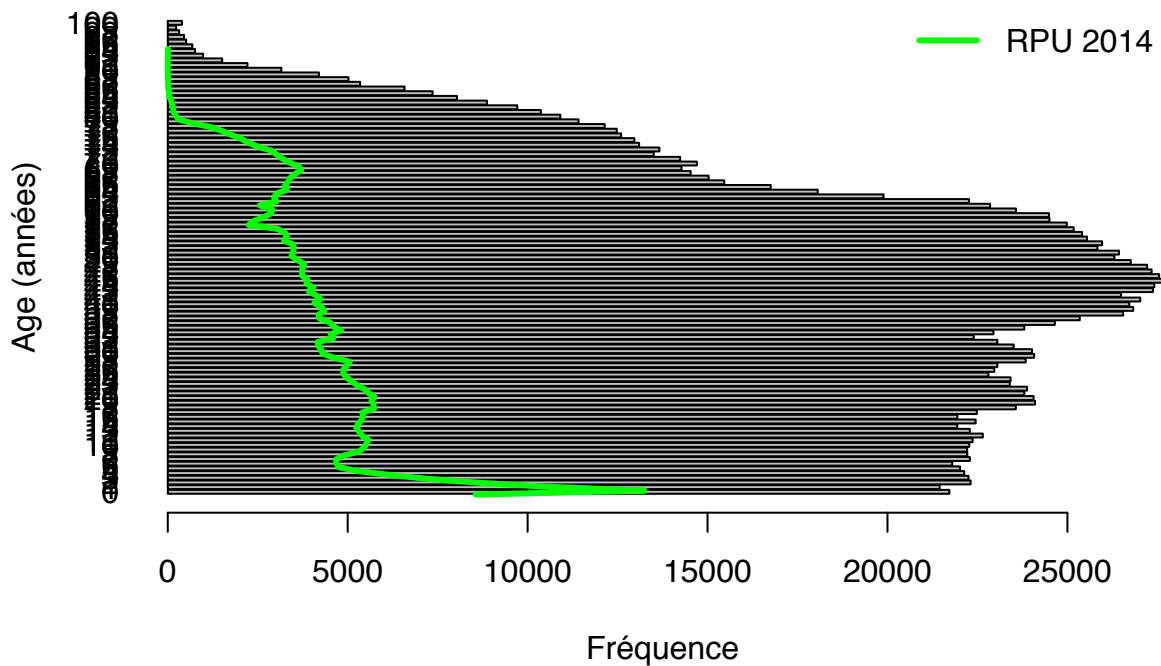
n67_2014 <- sum(pop67_2014$NB)
n68_2014 <- sum(pop68_2014$NB)
nAls_2014 <- sum(pop_als_2014$NB)

# pyramide des ages en Alsace en 2014: vecteur de 101 lignes. Chaque ligne correspond au total de la po
p_ages_als_2014 <- tapply(pop_als_2014$NB, pop_als_2014$AGED100, sum)

```

```
# représentation graphique de la pyramide des ages avec en surimpression la pyramide des ages des RPU (
barplot(p_ages_als_2014, horiz=TRUE, las = 1, main="Pramides des ages en Alsace (2014)", ylab="Age (années)
a <- table(as.factor(age_an))
x <- as.numeric(a)
y <- as.numeric(names(a))
lines(x,y, col = "green", lwd = 3)
legend("topright", legend = "RPU 2014", col = "green", lwd = 3, bty = "n")
```

Pramides des ages en Alsace (2014)



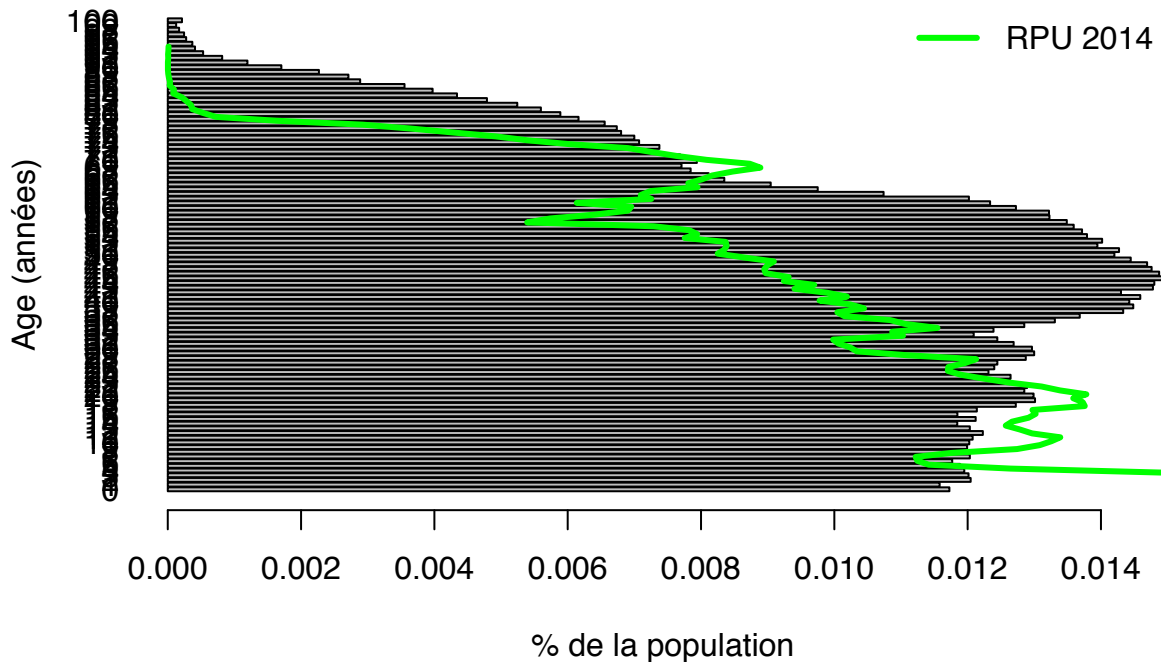
```
# Même représentation mais cette fois on utilise des pourcentages. Pour la pop alsacienne, l'effectif d
barplot(p_ages_als_2014/nAls_2014, horiz=TRUE, las = 1, main="Pramides des ages en Alsace (2014)", ylab="Age (années)
a <- table(as.factor(age_an))
a <- a/N
x <- as.numeric(a)
y <- as.numeric(names(a))
lines(x,y, col = "green", lwd = 3)
legend("topright", legend = "RPU 2014", col = "green", lwd = 3, bty = "n")
```

```
# calcul sur les sexes
```

```
sx <- tapply(pop_als_2014$NB, pop_als_2014$SEXE, sum)
nh_als <- as.numeric(sx[1])
nf_als <- as.numeric(sx[2])
sexR_als <- nh_als/nf_als
masc_als <- nh_als/(nh_als+nf_als)
```

```
# sex ratio par tranche d'age. On utilise tapply en lui passant une liste de deux séparateur, le sexe et
a <- tapply(pop_als_2014$NB, list(pop_als_2014$SEXE, pop_als_2014$AGED100), sum)
```

Pramides des ages en Alsace (2014)



```
sr_age <- a[1,]/a[2,]
plot(sr_age, type="l", main="Evolution du sexe-ratio en fonction de l'âge", ylab="Sexe-ratio", xlab="Age", lty=2)
abline(h=1, lty=2)
```

```
# Répartition de la population alsacienne en fonction des classes d'age de la Fedoru:
adultes_als <- array()
adultes_als[1] <- sum(p_ages_als_2014[18:29])
adultes_als[2] <- sum(p_ages_als_2014[30:44])
adultes_als[3] <- sum(p_ages_als_2014[45:64])
adultes_als[4] <- sum(p_ages_als_2014[65:74])
adultes_als[5] <- sum(p_ages_als_2014[75:84])
adultes_als[6] <- sum(p_ages_als_2014[85:101])
names(adultes_als) <- lab_adultes
adultes_als
```

```
## 18-30ans 30-45ans 45-65ans 65-75ans 75-85ans >85ans
## 281385 374784 507286 150229 114559 47760
```

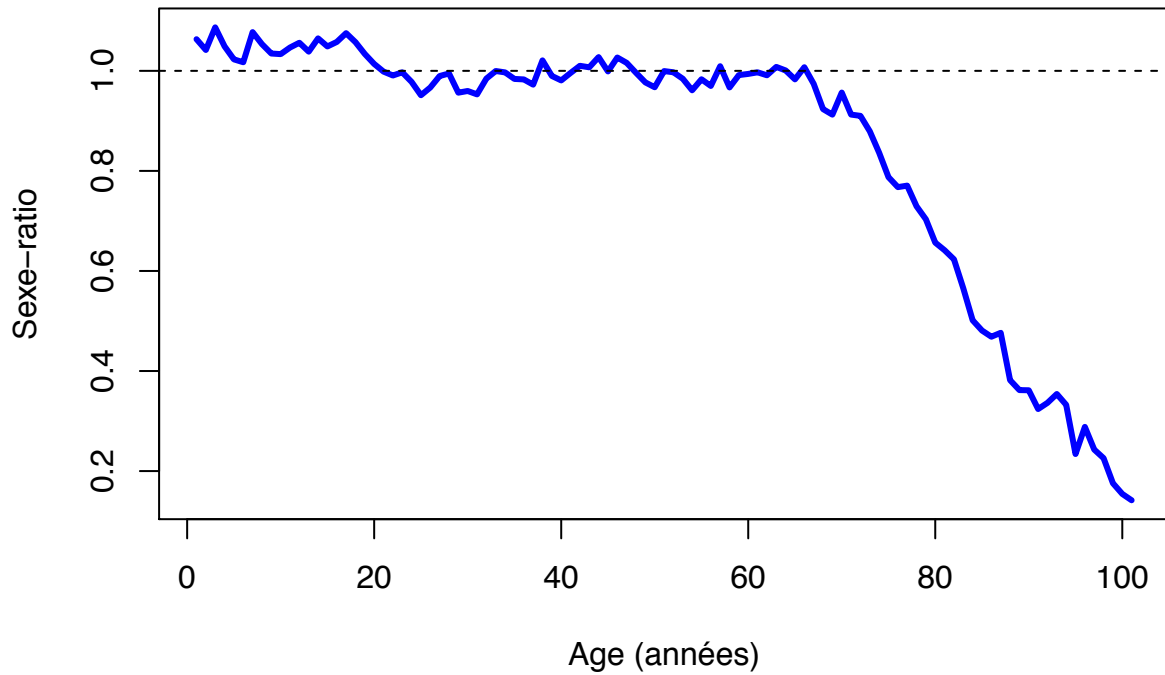
```
barplot(adultes_als, main="Répartition de la population alsacienne adulte\n en fonction des classes d'âge", lty=2)
```

```
# en % de la population
barplot(adultes_als*100/n67_2014, main="Répartition de la population alsacienne adulte\n en fonction de l'âge", lty=2)
```

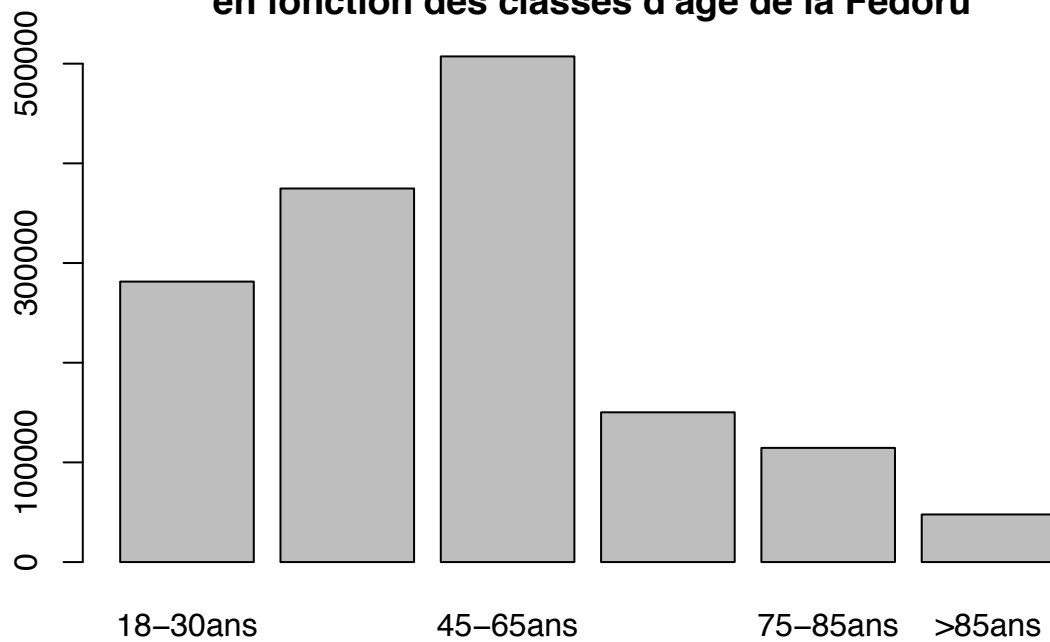
Variables créées:

- n67_2014: population totale du Bas-Rhin en 2014

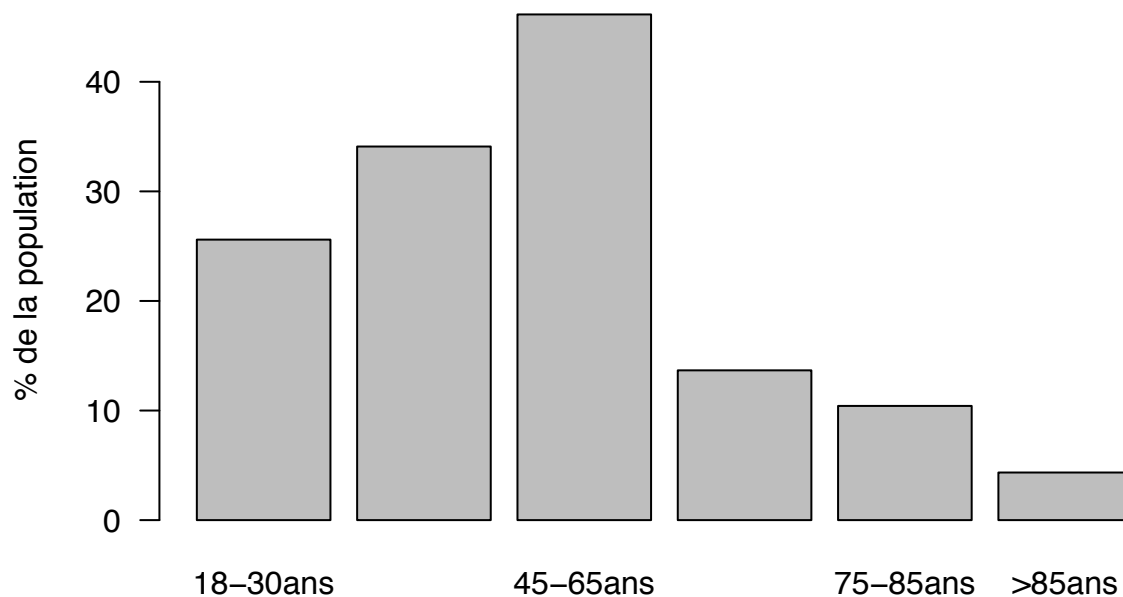
Evolution du sexe-ratio en fonction de l'âge



Répartition de la population alsacienne adulte en fonction des classes d'âge de la Fedoru



Répartition de la population alsacienne adulte en fonction des classes d'âge de la Fedoru



- n68_2014: population totale du Haut-Rhin en 2014
- nAls_2014: population totale d'Alsace en 2014
- p_ages_als_2014: pyramide des ages
- nh_als: nombre d'hommes
- nf_als: nombre de femmes en Alsace
- sexR_als : sex ratio en Alsace
- masc_als: rapport de masculinité en Alsace
- adultes_als: pop.adulte d'Alsace en fonction des classes de la Fedoru

Territoire	population
Bas-Rhin	1 099 269
Haut-Rhin	753 056
Alsace	1 852 325

4.2 Tranches d'age

Répartition des adultes dans la population alsacienne (INSEE). On utilise le découpage de la FEDORU (adultes_ans).

```
adultes_als_insee <- cut(dx$AGE, adultes_ans, include.lowest = TRUE, right = FALSE, labels = lab_adultes)
table(adultes_als_insee)
```

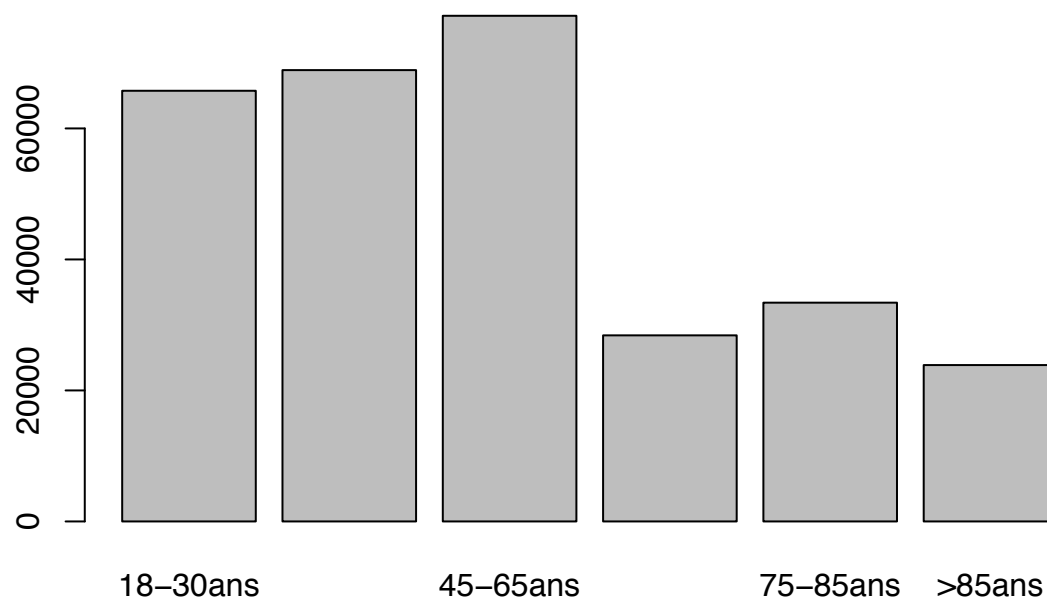
```
## adultes_als_insee
## 18-30ans 30-45ans 45-65ans 65-75ans 75-85ans >85ans
##      65744      68900      77194      28407      33399      23872
```

```
sum(table(adultes_als_insee))
```

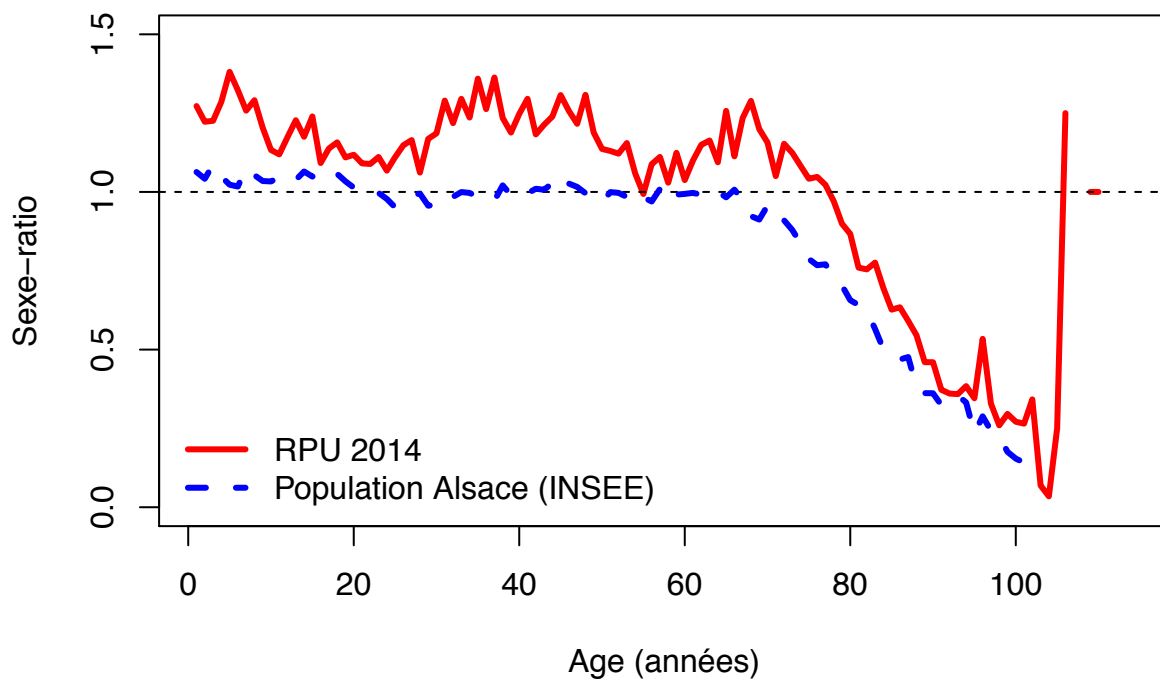
```
## [1] 297516
```

13

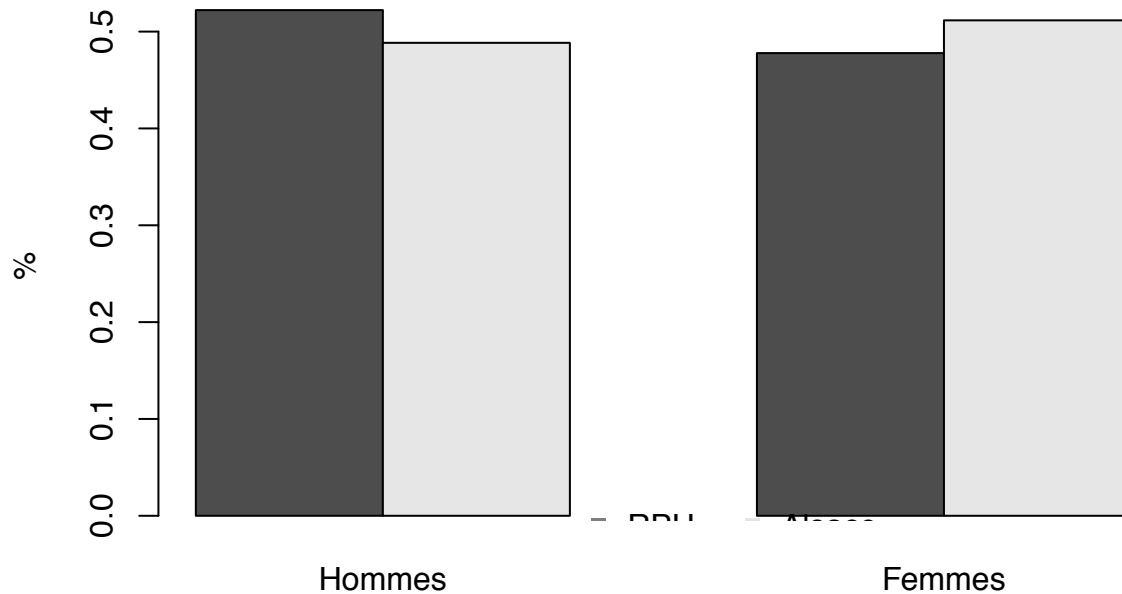
```
barplot(table(adultes_als_insee))
```



Evolution du sexe-ratio en fonction de l'âge



Répartition des passages aux urgences 2014 par sexe



4.3.0.1 Comparaison Passages - Population On combine dans une matrice les effectifs en pourcentages par classes d'âge (FEDORU) des passages ayant donné lieu à un RPU et ceux de la population totale d'Alsace. Le rapport s'inverse pour la classe 85 ans et plus.

```
a <- rbind(adultes_als * 100 / n67_2014, adultes_rpu * 100 / N)
rownames(a) <- c("Alsace", "RPU")
a
```

```
##      18-30ans 30-45ans 45-65ans 65-75ans 75-85ans >85ans
## Alsace      26      34      46      13.7      10      4.3
## RPU         16      17      19       6.8       8      5.7
```

```
barplot(a, beside=TRUE, main="Comparaison RPU et population adulte Alsacienne", ylab="% de la population",
legend("topright", legend=c("Alsace", "RPU"), pch=15, bty="n", col=c("black", "gray"))
```

5 Taux de passage

Le taux de passage: nombre de RPU / population estimée.

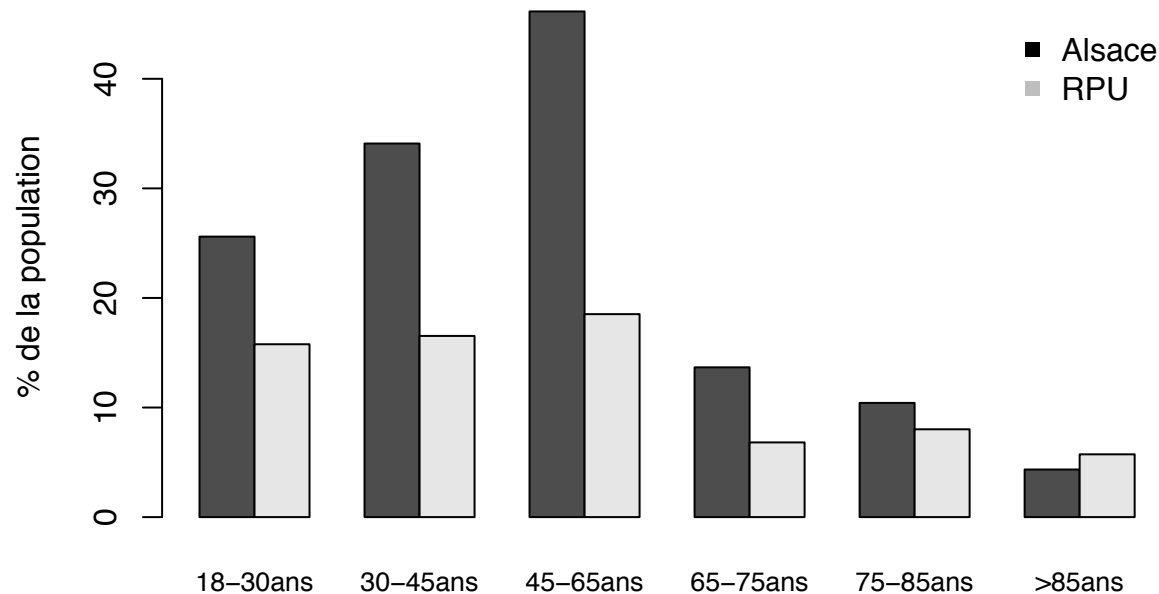
Pour l'ensemble de l'Alsace:

```
tx_passage <- N / nAls_2014
tx_passage
```

```
## [1] 0.22
```

Taux de passage par tranches d'âge:

Comparaison RPU et population adulte Alsacienne



```
tx_passage_age <- as.numeric(adultes_rpu) / adultes_als  
tx_passage_age
```

```
## 18-30ans 30-45ans 45-65ans 65-75ans 75-85ans >85ans  
##    0.23    0.18    0.15    0.19    0.29    0.50
```

A calculer également par SU.