Pondérations normalisées avec R

Elodie Baril

2023-06-29

L’utilisation des poids normalisés sur la taille de l’échantillon s’avère obligatoire pour certaines analyses, en particulier pour obtenir des estimations robustes de la variance. La présente fiche présente avec les packages questionr et survey, deux manières d’introduire ces poids dans des analyses.

Fonctions utilisées dans la fiche

| Package | Fonction |
| --- | --- |
| **questionr** | wtd.table - cprop - lprop |
| **survey** | svydesign - svytable - svymean - svyglm |
| **Base R** | mean |

Documentation sur les packages:

* <https://juba.github.io/questionr/>
* <https://www.rdocumentation.org/packages/survey/versions/4.1-1>

Activation des packages et ouverture de la table d’exemple HDV

#install.packages(questionr)  
#install.packages(survey)  
  
library(questionr)  
library(survey)  
  
#Utilisation de la table d'exemple Histoire de vie (hdv2003)   
data("hdv2003")  
  
# Extrait du fichier  
hdv2003[1:5,1:5]

id age sexe nivetud poids  
1 1 28 Femme Enseignement superieur y compris technique superieur 2634.398  
2 2 23 Femme <NA> 9738.396  
3 3 59 Homme Derniere annee d'etudes primaires 3994.102  
4 4 34 Homme Enseignement superieur y compris technique superieur 5731.662  
5 5 71 Femme Derniere annee d'etudes primaires 4329.094

La variable pondérée du fichier s’appelle **poids**

# Package questionr

Le package questionr permet d’obtenir rapidement des tableaux de fréquences ou de pourcentages avec des poids normalisés. L’opération de normalisation est passée directement dans une option.

## Fréquences pondérées

* Fonction **wtd.table()**

table(hdv2003$sexe)

Homme Femme   
 899 1101

# tris à plat avec l'option weights  
wtd.table(hdv2003$sexe, weights = hdv2003$poids)

Homme Femme   
5149382 5921844

# tris croisé  
wtd.table(hdv2003$qualif,hdv2003$sexe,weights=hdv2003$poids)

Homme Femme  
Ouvrier specialise 532365.8 479015.0  
Ouvrier qualifie 1163988.6 296252.0  
Technicien 378956.7 115658.0  
Profession intermediaire 438737.8 375560.7  
Cadre 779354.9 623574.6  
Employe 525068.8 2591911.0  
Autre 129695.9 145912.6

**Pour normaliser la pondération on ajoute l’option normwt=TRUE**.

wtd.table(hdv2003$qualif,hdv2003$sexe,weights=hdv2003$poids, normwt=TRUE)

Homme Femme  
Ouvrier specialise 96.17106 86.53332  
Ouvrier qualifie 210.27275 53.51747  
Technicien 68.45794 20.89344  
Profession intermediaire 79.25731 67.84446  
Cadre 140.78927 112.64780  
Employe 94.85286 468.22473  
Autre 23.42937 26.35888

## Pourcentages pondérés

res=wtd.table(hdv2003$qualif,hdv2003$sexe, weights=hdv2003$poids, normwt=TRUE)

* On utilise les fonctions cprop() et lprop:
  + cprop(): pourcentages colonnes
  + lprop(): pourcentages lignes

cprop(res)

Homme Femme Ensemble  
Ouvrier specialise 13.5 10.4 11.8   
Ouvrier qualifie 29.5 6.4 17.0   
Technicien 9.6 2.5 5.8   
Profession intermediaire 11.1 8.1 9.5   
Cadre 19.7 13.5 16.4   
Employe 13.3 56.0 36.3   
Autre 3.3 3.2 3.2   
Total 100.0 100.0 100.0

lprop(res)

Homme Femme Total  
Ouvrier specialise 52.6 47.4 100.0  
Ouvrier qualifie 79.7 20.3 100.0  
Technicien 76.6 23.4 100.0  
Profession intermediaire 53.9 46.1 100.0  
Cadre 55.6 44.4 100.0  
Employe 16.8 83.2 100.0  
Autre 47.1 52.9 100.0  
Ensemble 46.0 54.0 100.0

# Le package survey

* Plus riche, le package **survey** permet d’appliquer la pondération, normalisée ou non, à un ensemble plus vaster de fonction.
* Pour les modèles, l’application de la pondération normalisée (**obligatoire**) permet d’obtenir une estimation robuste de la variance des paramètres.
* La normalisation de la pondération se fait par la création d’une nouvelle variable.

## Création de la variable de pondération normalisée

On divise tout simplement chaque poids individuel par la moyenne des poids (fonction mean())

hdv2003$poids\_norm=hdv2003$poids/mean(hdv2003$poids)

Comme la normalisation se fait sur la taille de l’échantillon, la somme des poids normalisés est bien égale à l’effectif de celui-ci

sum(hdv2003$poids\_norm)

[1] 2000

## Design de l’échantillon

* On est ici dans la situation la plus simple simple, le design se réduisant à l’application d’un poids d’échantillonage.
* Avant d’utiliser la fonction d’analyse on générère un objet renseignant les divers éléments entrant dans le design de l’enquête (ici seulement la variable *hdv2003$poids\_norm*) avec la fonction **svydesign**

dfw <- svydesign(ids = ~1, data = hdv2003, weights = ~ hdv2003$poids)

## Application du design à des fonctions analytiques

* On ajoute aux fonctions l’objet généré avec svydesign (ici **dfw**)

**Tableaux de fréquence**

* Fonction **svytable**

svytable(~sexe, dfw)

sexe  
 Homme Femme   
5149382 5921844

svytable(~ sexe + qualif, dfw)

qualif  
sexe Ouvrier specialise Ouvrier qualifie Technicien Profession intermediaire  
 Homme 532365.8 1163988.6 378956.7 438737.8  
 Femme 479015.0 296252.0 115658.0 375560.7  
 qualif  
sexe Cadre Employe Autre  
 Homme 779354.9 525068.8 129695.9  
 Femme 623574.6 2591911.0 145912.6

tab <- svytable(~ sexe + qualif, dfw)

lprop(tab) # % ligne pondérée

qualif  
sexe Ouvrier specialise Ouvrier qualifie Technicien  
 Homme 13.5 29.5 9.6   
 Femme 10.4 6.4 2.5   
 Ensemble 11.8 17.0 5.8   
 qualif  
sexe Profession intermediaire Cadre Employe Autre Total  
 Homme 11.1 19.7 13.3 3.3 100.0  
 Femme 8.1 13.5 56.0 3.2 100.0  
 Ensemble 9.5 16.4 36.3 3.2 100.0

cprop(tab) # % colonne pondérée

qualif  
sexe Ouvrier specialise Ouvrier qualifie Technicien Profession intermediaire  
 Homme 52.6 79.7 76.6 53.9   
 Femme 47.4 20.3 23.4 46.1   
 Total 100.0 100.0 100.0 100.0   
 qualif  
sexe Cadre Employe Autre Ensemble  
 Homme 55.6 16.8 47.1 46.0   
 Femme 44.4 83.2 52.9 54.0   
 Total 100.0 100.0 100.0 100.0

**Moyenne pondérée**

* Fonction svymean et svyby

svymean(~age, dfw)

mean SE  
age 46.347 0.5284

* Si on souhaite calculer une moyenne par sous groupe, on utilise la fonction svyby à laquelle on ajoute l’argument svymean en option

svyby(~age, ~sexe, dfw, svymean)

sexe age se  
Homme Homme 45.20200 0.7419450  
Femme Femme 47.34313 0.7420836

* Modèle logistique avec svyglm

options(show.signif.stars=FALSE)  
summary(svyglm(cuisine ~ age + sexe + occup + qualif, dfw, family=binomial))

Warning in eval(family$initialize): nombre de succès non entier dans un glm  
binomial !

Call:  
svyglm(formula = cuisine ~ age + sexe + occup + qualif, design = dfw,   
 family = binomial)  
  
Survey design:  
svydesign(ids = ~1, data = hdv2003, weights = ~hdv2003$poids)  
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)  
(Intercept) -0.151065 0.324985 -0.465 0.642  
age -0.012303 0.006407 -1.920 0.055  
sexeFemme 0.922651 0.160463 5.750 1.06e-08  
occupChomeur 0.344982 0.255175 1.352 0.177  
occupEtudiant, eleve -0.677345 1.124417 -0.602 0.547  
occupRetraite -0.256705 0.242383 -1.059 0.290  
occupRetire des affaires 0.633727 1.100048 0.576 0.565  
occupAu foyer 0.249697 0.286245 0.872 0.383  
occupAutre inactif -0.231741 0.382382 -0.606 0.545  
qualifOuvrier qualifie 0.134447 0.252713 0.532 0.595  
qualifTechnicien -0.395892 0.357399 -1.108 0.268  
qualifProfession intermediaire 0.220994 0.293482 0.753 0.452  
qualifCadre -0.265439 0.256451 -1.035 0.301  
qualifEmploye 0.018206 0.222582 0.082 0.935  
qualifAutre -0.510097 0.401470 -1.271 0.204  
  
(Dispersion parameter for binomial family taken to be 0.9331289)  
  
Number of Fisher Scoring iterations: 4

Vous pouvez retrouver dans la documentation l’ensemble des fonctions prises en charge.