Package 'TestsFaciles'

March 26, 2007

Type Package

Title Facilite le calcul d'intervalles de confiance et de tests de comparaison avec prise en compte du plan d'échantillonnage.

Version 1.0

Date 2007-03-26

Author Joseph LARMARANGE

Maintainer Joseph LARMARANGE <joseph@larmarange.net>

Depends survey

Description Ce mini-package fournit deux fonctions (int.conf et compare) pour faciliter le calcul d'intervalles de confiance et de tests statistiques à l'aide du package survey afin de prendre en compte un plan d'échantillonange complexe. De plus, il fournit une fonction importer.spss, proche de read.spss, mais permettant de renvoyer un type factor lorsqu'une étiquette de données est définie dans le fichier .sav mais non présente dans les données.

License CeCILL-C - http://www.cecill.info/

URL http://joseph.larmarange.net/Tests-statistiques-avec-prise-en.html

R topics documented:

`TestsFaciles-package` *Facilite le calcul d'intervalles de confiance et de tests de comparaison avec prise en compte du plan d'échantillonnage.*

Description

Ce mini-package fournit deux fonctions (`int.conf` et `compare`) pour faciliter le calcul d'intervalles de confiance et de tests statistiques à l'aide du package `survey` afin de prendre en compte un plan d'échantillonange complexe. De plus, il fournit une fonction `importer.spss`, proche de `read.spss`, mais permettant de renvoyer un type factor lorsqu'une étiquette de données est définie dans le fichier .sav mais non présente dans les données.

Details

Package:	TestsFaciles
Type:	Package
Version:	1.0
Date:	2007-03-26
License:	CeCILL-C - http://www.cecill.info/

Author(s)

Joseph LARMARANGE <joseph@larmarange.net>
http://joseph.larmarange.net/Tests-statistiques-avec-prise-en.html

See Also

Package `survey`

Examples

```
data(api)

# Spécification du plan d'échantillonnage
donnees <- svydesign(id=~dnum, weights=~pw, data=apiclus1)

# Intervalles de confiance
int.conf(~api99+yr.rnd, donnees)

# pour mieux visualiser les résultats
t(int.conf(~api99+yr.rnd, donnees))

# résultats par sous-groupes
t(int.conf(~api99+yr.rnd, donnees, by=~stype))

# Comparaison de 2 moyennes ou de 2 proportions

compare(~api99, donnees, by=~stype)
```

```
# on peut effectuer plusieurs comparaisons en même temps
compare(~api99+yr.rnd, donnees, by=~stype)

# facilite l'affichage des résultats
t(compare(~api99+yr.rnd, donnees, by=~stype))
```

`compare` *Test de comparaison de deux moyennes ou deux proportions.*

Description

Compare des moyennes ou des proportions deux à deux selon leur valeur dans différents sous-groupes.

Usage

```
compare(x, design, by, loi = "student")
```

Arguments

`x`	formule. Liste des variables que l'on souhaite tester.
`design`	objet de la classe `svydesign` ou tableau de données. Données.
`by`	formule. Liste des variables de type facteur définissant les groupes à comparer.
`loi`	texte. *student* ou *normale*. Spécifie la loi utilisée pour le calcul des intervalles. Voir les détails.

Details

Les tests de comparaison effectués par cette fonction considèrent que, sous l'hypthèse nulle,

$$\frac{\bar{x}_1 - \bar{x}_2}{\sqrt{SE_1^2 + SE_2^2}}$$

suit une loi normale centrée réduite (si `loi='normale'`) ou une loi t de Student à $n_1 + n_2 - 2$ degrés de liberté si `loi='student'`.

Les moyennes, proportions, écarts-types et effectifs sont calculés avec `int.conf`.

Le test bilatéral teste si $\bar{x}_1 \neq \bar{x}_2$ (hypothèse nulle : $\bar{x}_1 = \bar{x}_2$).
Le test unilatéral teste si $\bar{x}_1 > \bar{x}_2$ (hypothèse nulle : $\bar{x}_1 \leq \bar{x}_2$).

Value

Un tableau de données avec une ligne par combinaison de groupes et les colonnes suivantes pour chaque variable x testée.

`valeur1.x`	Valeur de $\bar{x}$ dans le groupe 1.
`valeur2.x`	Valeur de $\bar{x}$ dans le groupe 2.
`difference.x`	valeur1.x - valeur2.x
`p.unilateral.x`	Valeur de p au test unilatéral.
`p.bilateral.x`	valeur de p au test bilatéral.
`type.x`	Type de grandeur : moyenne ou proportion.

See Also

int.conf

Examples

```
data(api)

# Spécification du plan d'échantillonnage
donnees <- svydesign(id=~dnum, weights=~pw, data=apiclus1)

# Comparaison de 2 moyennes ou de 2 proportions

compare(~api99, donnees, by=~stype)

# on peut effectuer plusieurs comparaisons en même temps
compare(~api99+yr.rnd, donnees, by=~stype)

# facilite l'affichage des résultats
t(compare(~api99+yr.rnd, donnees, by=~stype))
```

---

importer.spss *Lire un fichier SPSS au format .sav.*

Description

Cette fonction est identique à read.spss exceptée sur un point. En effet, si dans le fichier SPSS, une variable comporte des étiquettes de valeurs non présentes dans les données, read.spss renvoie une colonne de type *atomic* tandis que importer.spss renvoie une colonne de type factor. Les options par défaut diffèrent également.

Usage

```
importer.spss(
   file,
   use.value.labels = TRUE,
   to.data.frame = TRUE,
   max.value.labels = Inf,
   trim.factor.names = TRUE)
```

Arguments

file	character. Nom du fichier *sav* à lire et transformer en data.frame.
use.value.labels	logical. Convertir les variables avec des étiquettes de valeurs en colonnes de type *factor* ?
to.data.frame	logical. Renvoyer un data.frame ? Renvoie une liste sinon.
max.value.labels	logical. Seules les variables ayant au maximum ce nombre de modalités seront converties en *factor*.
trim.factor.names	logical. Supprimer les espaces des *noms des facteurs* ?

Details

Cette fonction est identique à read.spss exceptée sur un point. En effet, si dans le fichier SPSS, une variable comporte des étiquettes de valeurs non présentes dans les données, `read.spss` renvoie une colonne de type *atomic* tandis que `importer.spss` renvoie une colonne de type *factor*.
Les options par défaut diffèrent légèrement. par défaut, `importer.spss` renvoie un tableau de données et supprime les espaces inutiles des noms de facteur.

See Also

`read.spss` pour plus de détails sur cette fonction.

Examples

```
## Not run:
donnees <- importer.spss('file.sav')
## End(Not run)
```

`int.conf` *Calcule des intervalles de confiance d'une moyenne ou d'une proportion*

Description

calcule en série des intervalles de confiance d'une moyenne (varibale numérique) ou d'une proportion (variables de type facteurs) en tenant compte du plan d'échantillonage complexe des données. Il est possible de calculer les intervalles de confiance pour des sous-groupes.

Usage

```
int.conf(x, design, by = NULL, confiance = 0.95, loi = "student")
```

Arguments

`x`	formule. Liste des variables pour lesquelles on souhaite calculer un intervalle de confiance.
`design`	objet de la classe `svydesign` ou tableau de données. Données à partir desquelles les intervalles de confiance seront calculés.
`by`	valeur nulle ou formule. liste des variables de type facteurs définissant des sous-échantillons. Si `NULL`, les intervalles de confiance sont calculés sur l'ensemble de l'échantillon.
`confiance`	numérique. Probabilité de l'intervalle de confiance.
`loi`	texte. *student* ou *normale*. Spécifie la loi utilisée pour le calcul des intervalles. Voir les détails.

Details

Si on a passé un tableau de données au paramètre design, les données seront considérées comme résultant d'un tirage aléatoire simple.

Pour les variables de type numérique, c'est la moyenne qui est calculé. Pour les variables de type facteurs, c'est la proportion de chaque facteur.

La fonction `int.conf` utilise la fonction `svymean` pour calculer les moyennes et/ou proportions ainsi que l'écart-type de l'échantillon noté SE. Les intervalles de confiance avec une probabilité α, notés IC_α sont calculés selon la formule suivante :

$$IC_\alpha = \bar{x} \pm SE \cdot l_{\frac{\alpha}{2}}$$

où $\bar{x}$ représente la moyenne ou la proportion pour laquelle l'intervalle est calculé, SE l'écrt-type de cette variable dans l'échnatillon et $l_{\frac{\alpha}{2}}$ la valeur de la loi utilisée.

Si `loi='normale'`, $l_{\frac{\alpha}{2}}$ correspondra à la valeur $z_{\frac{\alpha}{2}}$ de la loi normale centrée réduite. Si `loi='student'`, $l_{\frac{\alpha}{2}}$ correspondra à la valeur de $t_{\frac{\alpha}{2}}$ de la loi t de student avec $n-1$ degrés de liberté.

Value

La fonctin renvoie un tableau de données (data.frame) avec une ligne par groupe et pour chaque variable de `x` les colonnes suivantes

`moyenne.x ou proportion.x`
: Valeur de la moyenne ou de la proportion de la variable considérée.

`se.x`
: Écart-type sur l'échantillon de la variable.

`ic.bas.x`
: Minimum de l'intervalle de confiance.

`ic.haut.x`
: Maximum de l'intervalle de confiance.

`n.x`
: Nombres d'observations dans l'échantillon (les valeurs manquantes sont ignorées).

See Also

`svymean`, `svyby`.

Examples

```
data(api)

# Spécification du plan d'échantillonnage
donnees <- svydesign(id=~dnum, weights=~pw, data=apiclus1)

# Intervalles de confiance
int.conf(~api99+yr.rnd, donnees)

# pour mieux visualiser les résultats
t(int.conf(~api99+yr.rnd, donnees))

# résultats par sous-groupes
t(int.conf(~api99+yr.rnd, donnees, by=~stype))
```

Index