Package 'regression'

May 26, 2016

Type Package

Title Creation de regression multilineaire et de jeu d'observations

Version 1.0

Date 2016-05-24

Author Lair Thomas, Bonjean Gregoire et Crepin Baptiste

Maintainer Lair Thomas <lairt.ensimag@gmail.com>

Description Voir commentaires en en-tete des fonctions

License x

R topics documented:

regression-package 1
interaction_regression 2
mixed_effect_regression 3
modele 4
multiple_regression 4
observations 5
regression 6
regression_blind 6
univariate_regression 7

Index **9**

`regression-package` *Creation de regression multilineaire et de jeu d'observations*

Description

Voir commentaires en en-tete des fonctions

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Lair Thomas, Bonjean Gregoire et Crepin Baptiste

Maintainer: Lair Thomas <lairt.ensimag@gmail.com>

`interaction_regression` *Interaction Regression*

Description

Regression Multiple avec Interaction. Valeurs indicatives : * n ~ 100 * b_0 ~ 1 * b_1 ~ 0.5 * b_2 ~ 0.4 * b_3 ~ 0.3 * noise_s ~ 0.2

Usage

```
interaction_regression(n, b_0, b_1, b_2, b_3, noise_s)
```

Arguments

`n`

`b_0`

`b_1`

`b_2`

`b_3`

`noise_s`

Examples

```
function (n, b_0, b_1, b_2, b_3, noise_s)
{
    x = cbind(rep(1, n), runif(n), runif(n))
    x = cbind(x, x[, 3] * x[, 2])
    b = c(b_0, b_1, b_2, b_3)
    noise_sd = noise_s
    y = x %*% b + rnorm(n, 0, noise_sd)
    mod = lm(y ~ x[, -1])
    return(list(Y = y, x = x, sum = summary(mod)))
  }
```

`mixed_effect_regression` *Mixed Effect Regression*

Description

Regression avec effets mixtes. Valeurs indicatives : * n ~ 100 * b_0 ~ 1 * b_1_f ~ 0.4 * b_2 ~ 0.3 * noise_s ~ 0.2

Usage

```
mixed_effect_regression(n, b_0, b_1_f, b_2, noise_s)
```

Arguments

`n`

`b_0`

`b_1_f`

`b_2`

`noise_s`

Examples

```
function (n, b_0, b_1_f, b_2, noise_s)
{
    npart = 10
    part = sort(rep(c(1:npart), length.out = n))
    x = runif(n)
    b0 = b_0
    b1_f = b_1_f
    b1 = b1_f + b_2 * rnorm(npart)
    noise_sd = noise_s
    y = b0 + b1[part] * x + rnorm(n, 0, noise_sd)
    library(lme4)
    mod = lmer(y ~ x + (0 + x | part))
    df1 = data.frame(Estimate = fixef(mod), Parameter = rbind(b0,
        b1_f))
    df2 = data.frame(Estimate = fixef(mod)["x"] + ranef(mod)$part$x,
        Parameter = b1)
    return(list(Y = y, x = x, sum = summary(mod), df1 = df1,
        df2 = df2))
  }
```

`modele` *Modele*

Description

Modele genere les Y en partant des observations x. A utiliser avec observations.

Usage

```
modele(alpha, beta, obs)
```

Arguments

`alpha`

`beta`

`obs`

Examples

```
function (alpha, beta, obs)
{
    n_param <- length(beta)
    n_obs <- length(obs$x)/n_param
    e <- rnorm(n_obs, mean = 0, sd = 1)
    Y <- 1:n_obs
    for (i in 1:n_obs) {
        Y[i] <- alpha + e[i]
        for (j in 1:n_param) {
            Y[i] <- Y[i] + beta[j] * obs$x[((j - 1) * n_obs) +
                i]
        }
    }
    return(list(Y = Y, e = e))
  }
```

`multiple_regression` *Multiple Regression*

Description

Regression Multiple. Valeurs Indicatives : * n ~ 100 * b_0 ~ 1 * b_1 ~ 0.5 * b_2 ~ 0.4 * noise_s ~ 0.2

Usage

```
multiple_regression(n, b_0, b_1, b_2, noise_s)
```

Arguments

n

b_0

b_1

b_2

noise_s

Examples

```
function (n, b_0, b_1, b_2, noise_s)
{
    x = cbind(rep(1, n), runif(n), runif(n))
    b = c(b_0, b_1, b_2)
    noise_sd = noise_s
    y = x %*% b + rnorm(n, 0, noise_sd)
    mod = lm(y ~ x[, -1])
    return(list(Y = y, x = x, sum = summary(mod)))
  }
```

---

observations *Observations*

Description

Observations genere les observations x de la regression multilineaire. A utiliser en deux temps avec modele.

Usage

```
observations(n_obs, mu, s)
```

Arguments

n_obs

mu

s

Examples

```
function (n_obs, mu, s)
{
    n_param <- length(mu)
    x <- 1:(n_param * n_obs)
    for (i in 1:n_param) {
        x[((i - 1) * n_obs + 1):(i * n_obs)] <- rnorm(n_obs,
            mean = mu[i], sd = s[i])
    }
    return(list(x = x))
  }
```

regression *Regression*

Description

Donne immediatement les observations x et Y. Combinaison de Observations et Modele.

Usage

```
regression(alpha, beta, n_obs, mu, s)
```

Arguments

```
alpha
beta
n_obs
mu
s
```

Examples

```
function (alpha, beta, n_obs, mu, s)
{
    n_param <- length(beta)
    x <- 1:(n_param * n_obs)
    for (i in 1:n_param) {
        x[((i - 1) * n_obs + 1):(i * n_obs)] <- rnorm(n_obs,
            mean = mu[i], sd = s[i])
    }
    e <- rnorm(n_obs, mean = 0, sd = 1)
    Y <- 1:n_obs
    for (i in 1:n_obs) {
        Y[i] <- alpha + e[i]
        for (j in 1:n_param) {
            Y[i] <- Y[i] + beta[j] * x[((j - 1) * n_obs) + i]
        }
    }
    return(list(x = x, Y = Y, e = e))
  }
```

regression_blind *Regression Aveugle*

Description

Identique à Regression, sauf qu'ici les constantes de la régression et les propriétés des normales sont déterminées aléatoirement plutôt que fournies. alpha \in [-10;10] beta \in [-5;5] mean \in [-5;5] sd \in [0;3]

Usage

```
regression_blind(n_param, n_obs)
```

Arguments

```
n_param

n_obs
```

Examples

```
function (n_param, n_obs)
{
    alpha <- 10 - sample(1:2000, 1)/100
    beta <- 1:n_param
    mu <- 1:n_param
    s <- 1:n_param
    for (i in 1:n_param) {
        beta[i] = 5 - sample(1:1000, 1)/100
        mu[i] = 5 - sample(1:1000, 1)/100
        s[i] = 3 - sample(1:300, 1)/100
    }
    x <- 1:(n_param * n_obs)
    for (i in 1:n_param) {
        x[((i - 1) * n_obs + 1):(i * n_obs)] <- rnorm(n_obs,
            mean = mu[i], sd = s[i])
    }
    e <- rnorm(n_obs, mean = 0, sd = 1)
    Y <- 1:n_obs
    for (i in 1:n_obs) {
        Y[i] <- alpha + e[i]
        for (j in 1:n_param) {
            Y[i] <- Y[i] + beta[j] * x[((j - 1) * n_obs) + i]
        }
    }
    return(list(Y = Y, x = x, alpha = alpha, beta = beta, mean = mu,
        sd = s, e = e))
  }
```

---

univariate_regression

Univariate Regression

Description

Regression Simple. Valeurs indicatives : * n ~ 100 * b_0 ~ 1 * b_1 ~ 0.5 * noise_s ~ 0.2

Usage

```
univariate_regression(n, b_0, b_1, noise_s)
```

Arguments

n

b_0

b_1

noise_s

Examples

```
function (n, b_0, b_1, noise_s)
{
    x = runif(n)
    b0 = b_0
    b1 = b_1
    noise_sd = noise_s
    y = b0 + b1 * x + rnorm(n, 0, noise_sd)
    mod = lm(y ~ x)
    return(list(Y = y, x = x, sum = summary(mod)))
  }
```

Index

∗Topic **package**
    regression-package, 1

interaction_regression, 2

mixed_effect_regression, 3
modele, 4
multiple_regression, 4

observations, 5

regression, 6
regression *(regression-package)*, 1
regression-package, 1
regression_blind, 6

univariate_regression, 7