Package 'anova.reg'

June 9, 2016

Type Package

Title Implementation des cas complexes de calcul de puissance (Anova et regressions).

Version 1.0

Date 2016-06-09

Author Bonjean Gregoire, Crepin Baptiste et Lair Thomas

Maintainer Lair Thomas <thomas.lair@ensimag.grenoble-inp.fr>

Description Ce package permet le calcul de la puissance d'un test d'hypothese (presence d'un effet sous la forme d'un ecart a la moyenne) dans le cas d'un de k echantillons (Anova), de la regression simple et de la regression multiple. Il comprends des fonctions qui permettent de generer des pilotes tests, ainsi que des fonctions pour le Bootstrap et la methode de Monte-Carlo.

License x

R topics documented:

anova.reg-package . . . 2
between_group_variance . . . 3
calcul_n . . . 3
F . . . 4
MC_an . . . 5
MC_rm . . . 6
MC_ru . . . 7
N . . . 8
pilote_anova . . . 8
pilote_multiple_reg . . . 9
plot_mod_mr . . . 11
plot_mod_ur . . . 12
Puissance_an . . . 13
Puissance_mr . . . 14
Puissance_ur . . . 15
sigma_m . . . 16
stat_b_0_mr . . . 17

stat_b_0_un 17
stat_b_1_mr 18
stat_b_1_un 18
stat_b_2_mr 19
stat_noise_s_mr 19
stat_noise_s_un 20
test_univariate 20
ttest_anova 21
within_group_variance 23

Index **25**

`anova.reg-package` *Calcul de puissance de cas complexes de tests d'hypothese.*

Description

Ce package permet le calcul de la puissance d'un test d'hypothese (presence d'un effet sous la forme d'un ecart a la moyenne) dans le cas d'un de k echantillons (Anova), de la regression simple et de la regression multiple. Il comprends des fonctions qui permettent de generer des pilotes tests, ainsi que des fonctions pour le Bootstrap et la methode de Monte-Carlo.

Details

Package:	anova.reg
Type:	Package
Version:	1.0
Date:	2016-06-09
License:	x

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Maintainer: Lair Thomas <thomas.lair@ensimag.grenoble-inp.fr>

References

Tests statistiques parametriques : Puissance, taille d'effet et taille d'echantillon (sous R). – Stéphane CHAMPELY, Universite Lyon 1, France.

between_group_variance *Variance entre les groupes.*

Usage

```
between_group_variance(sample_sizes, means, y)
```

Arguments

```
sample_sizes
means
y
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (sample_sizes, means, y)
{
    bgv = 0
    for (j in 1:length(sample_sizes)) {
        bgv = bgv + sample_sizes[j] * (means[j] - mean(y))^2
    }
    return(bgv)
  }
```

calcul_n *Calcul de taille d'echantillon.*

Description

Cette fonction permet le calcul par approximation affine de la taille d'echantillon necessaire à obtenir la puissance visee.

Usage

```
calcul_n(npoints, puissance, puissances, tailles)
```

Arguments

npoints	Nombre de points du graphe puissance = f(taille d'echantillon).
puissance	Puissance visee pour le test d'hypothese.
puissances	Vecteur des puissances calcules en les npoints points du graphe.
tailles	Vecteur des tailles d'echantillons (simulé par Monte-Carlo).

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (npoints, puissance, puissances, tailles)
{
    if (is.null(puissance)) {
        return(-1)
    }
    else {
        i = 0
        p_cour = 0
        while ((i < npoints + 1) && (p_cour < puissance)) {
            i = i + 1
            p_cour = puissances[i]
        }
        if (i == npoints + 1) {
            return(0)
        }
        else {
            n2 = tailles[i]
            p2 = puissances[i]
            if (i == 1) {
                n1 = 0
                p1 = 0
            }
            else {
                n1 = tailles[i - 1]
                p1 = puissances[i - 1]
            }
            pente = (p2 - p1)/(n2 - n1)
            if (pente == 0) {
                n_calc = (n2 - n1)/2
            }
            else {
                n_calc = (puissance - p1)/pente + n1
            }
            n_calc = ceiling(n_calc)
            return(n_calc)
        }
    }
  }
```

F *F*

Usage

```
F(means, sample_sizes, y, k)
```

Arguments

means

sample_sizes

y

k

Examples

```
function (means, sample_sizes, y, k)
{
    num = between_group_variance(sample_sizes, means, y)
    denom = within_group_variance(sample_sizes, means, y)
    f = (num/denom) * (N(sample_sizes) - k)/(k - 1)
    return(f)
  }
```

MC_an *MC_anova*

Usage

```
MC_an(n, runs, means_empirique_pilote, s, k)
```

Arguments

n

runs

means_empirique_pilote

s

k

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

function (n, runs, means_empirique_pilote, s, k)

alpha = 0.05 sample_sizes = numeric(k) for (i in 1:k) sample_sizes[i] = n

x = c(rep(1, N(sample_sizes))) x = factor(x) fval_hand = numeric(runs) fval = numeric(runs) means_empirique = c(rep(1, k)) for (r in 1:runs) y = 1:N(sample_sizes) indice = 0 for (i in 1:k) y[(indice + 1):(indice + sample_sizes[i])] = c(rnorm(sample_sizes[i], mean = means_empirique_pilote[i], sd = s)) indice = indice + sample_sizes[i]

y1 = y[x == 1] indice = 0 for (i in 1:k) intervalle = y1[(indice + 1):(indice + sample_sizes[i])] means_empirique[i] = mean(intervalle) indice = indice + sample_sizes[i]

fval_hand[r] = F(means_empirique, sample_sizes, y1, k)

ncp = (N(sample_sizes)) * (sigma_m(means_empirique, sample_sizes)/s)^2 thr = qf(1 - alpha, df1 = k - 1, df2 = N(sample_sizes) - k) nb = sum(fval_hand > thr) p5_hand = nb/runs p5_package = pwr.anova.test(f = sigma_m(means_empirique, sample_sizes)/s, k = k, n = n)$power return(list(sd = s, p5_hand = p5_hand, p5_package = p5_package))

`MC_rm` *MC_regression_multiple*

Usage

```
MC_rm(alpha = 0.05, n, runs, b_0, b_1, b_2, noise_s)
```

Arguments

`alpha`

`n`

`runs`

`b_0`

`b_1`

`b_2`

`noise_s`

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (alpha = 0.05, n, runs, b_0, b_1, b_2, noise_s)
{
    pval0 = numeric(runs)
    pval1 = numeric(runs)
    pval2 = numeric(runs)
    tval_hand = numeric(runs)
    for (r in 1:runs) {
        reg = multiple_regression(n, b_0, b_1, b_2, noise_s)
        sum = reg$sum
        pval0[r] = sum[[4]][[10]]
        pval1[r] = sum[[4]][[11]]
        pval2[r] = sum[[4]][[12]]
    }
    p5_model_0 = sum(pval0 < alpha)/runs
    p5_model_1 = sum(pval1 < alpha)/runs
    p5_model_2 = sum(pval2 < alpha)/runs
    p5_model_12 = sum((pval1 < alpha) & (pval2 < alpha))/runs
    return(list(p5_model_0 = p5_model_0, p5_model_1 = p5_model_1,
        p5_model_2 = p5_model_2, p5_model_12 = p5_model_12))
  }
```

MC_ru *MC_univariate_regression*

Usage

```
MC_ru(alpha = 0.05, n, runs, b_0, b_1, noise_s)
```

Arguments

alpha

n

runs

b_0

b_1

noise_s

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function(alpha = 0.05, n, runs, b_0, b_1, noise_s){
    pval0 = numeric(runs)
    pval1 = numeric(runs)
    tval_hand = numeric(runs)
    for (r in 1:runs) {
        reg = univariate_regression(n, b_0, b_1, noise_s)
        sum = reg$sum
        pval0[r] = sum[[4]][[7]]
        pval1[r] = sum[[4]][[8]]
    }
    p5_model_0 = sum(pval0 < alpha)/runs
    p5_model_1 = sum(pval1 < alpha)/runs
    return(list(p5_model_0 = p5_model_0, p5_model_1 = p5_model_1))
  }
```

N *N*

Usage

```
N(sample_sizes)
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (sample_sizes)
{
    n = 0
    for (i in 1:length(sample_sizes)) {
        n = n + sample_sizes[i]
    }
    return(n)
  }
```

pilote_anova *pilote_anova*

Usage

```
pilote_anova(k, fact, npilote, sd, runs_bs_pilote, dest_pilote)
```

Arguments

k

fact

npilote

sd

runs_bs_pilote

dest_pilote

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (k, fact, npilote, sd, runs_bs_pilote, dest_pilote)
{
    alpha = 0.05
    means = c(rep(1, k))
    sample_sizes = numeric(k)
    for (i in 1:k) {
        means[i] = 1/i
        sample_sizes[i] = npilote
    }
    means = fact * means
    means_empirique_pilote = numeric(k)
    echantillon = matrix(nrow = sample_sizes[i], ncol = k)
    conf_mean = matrix(nrow = 2, ncol = k)
    conf_sd = matrix(nrow = 2, ncol = k)
    table_sd_i = numeric(runs_bs_pilote)
    table_sd_i_sorted = numeric(runs_bs_pilote)
    for (i in 1:k) {
        echantillon[, i] = rnorm(n = sample_sizes[i], mean = means[i],
            sd = sd)
        means_empirique_pilote[i] = mean(echantillon[, i])
        conf_mean[, i] = t.test(echantillon[, i], conf.level = alpha)$conf.int
        for (j in 1:runs_bs_pilote) {
            n_bs = ceiling(0.8 * sample_sizes[i])
            table_sd_i[j] = sd((sample(echantillon[, i], n_bs,
                replace = T)))
        }
        table_sd_i_sorted = sort(table_sd_i)
        conf_sd[, i] = c(table_sd_i_sorted[floor(runs_bs_pilote *
            alpha)], table_sd_i_sorted[floor(runs_bs_pilote *
            (1 - alpha))])
    }
    numero_echantillon = matrix(nrow = sample_sizes[i], ncol = k)
    for (i in 1:k) {
        numero_echantillon[, i] = rep(i, sample_sizes[i])
    }
    jpeg(dest_pilote)
    mp <- matplot(numero_echantillon, echantillon)
    lines(1:k, means_empirique_pilote, pch = 21, col = "red")
    lines(1:k, means, pch = 21, col = "blue")
    legend("topleft", c("Moyennes empiriques", "Moyennes r<U+00E9>elles"),
        fill = c("red", "blue"), bty = "n", border = NA)
    dev.off()
    ecart_type = sd
    return(list(ecart_type = ecart_type, empiricmeans = means_empirique_pilote,
        realmeans = means, sizes = sample_sizes, conf_mean = conf_mean,
        conf_sd = conf_sd))
  }
```

---

pilote_multiple_reg *pilote_multiple_regression*

Usage

```
pilote_multiple_reg(npilote, runs_bs_pilote, b_0, b_1, b_2, noise_s, dest_pilote)
```

Arguments

```
npilote
runs_bs_pilote
b_0
b_1
b_2
noise_s
dest_pilote
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (npilote, runs_bs_pilote, b_0, b_1, b_2, noise_s, dest_pilote)
{
    if (is.null(b_0) | is.null(b_1) | is.null(b_2) | is.null(noise_s)) {
      print("G<U+00E9>n<U+00E9>ration d'un pilote aux param<U+00E8>tres al<U+00E9>atoires pour r<U+00E9>gressio
        reg = regression_blind(2, npilote)
        x = reg$x
        Y = reg$Y
        x = matrix(x, nrow = npilote, ncol = 2)
        model = lm(Y ~ x)
    }
    else {
        try(if (length(b_0) != 1 | length(b_1) != 1 | length(b_2) !=
            1 | length(noise_s) != 1)
            stop("b0,b1,b2 et noise_s doivent <U+00EA>tre de taille 1"))
        reg = multiple_regression(npilote, b_0, b_1, b_2, noise_s)
        x = reg$x
        Y = reg$Y
        x = x[, -1]
        model = lm(Y ~ x)
    }
    sum = summary(model)
    b0 = sum[[4]][[1]]
    b1 = sum[[4]][[2]]
    b2 = sum[[4]][[3]]
    noises = sum$sigma
    data = cbind(x, Y)
    boot_b_0 <- boot(data = data, statistic = stat_b_0_mr, R = runs_bs_pilote)
    boot_b_1 <- boot(data = data, statistic = stat_b_1_mr, R = runs_bs_pilote)
    boot_b_2 <- boot(data = data, statistic = stat_b_2_mr, R = runs_bs_pilote)
    boot_noise_s <- boot(data = data, statistic = stat_noise_s_mr,
```

```
        R = runs_bs_pilote)
    c_0 = boot.ci(boot_b_0, type = "bca")$bca
    c_1 = boot.ci(boot_b_1, type = "bca")$bca
    c_2 = boot.ci(boot_b_2, type = "bca")$bca
    c_s = boot.ci(boot_noise_s, type = "bca")$bca
    conf_b_0 = c(c_0[4], c_0[5])
    conf_b_1 = c(c_1[4], c_1[5])
    conf_b_2 = c(c_2[4], c_2[5])
    conf_noise_s = c(c_s[4], c_s[5])
    plot_mod_mr(x, Y, dest_pilote)
    return(list(b_0 = b0, b_1 = b1, b_2 = b2, noise_s = noises,
        conf_b_0 = conf_b_0, conf_b_1 = conf_b_1, conf_b_2 = conf_b_2,
        conf_noise_s = conf_noise_s))
  }
```

---

plot_mod_mr *plot_mod_multiple_regression*

Usage

```
plot_mod_mr(x, Y, dest_pilote)
```

Arguments

x

Y

dest_pilote

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (x, Y, dest_pilote)
{
    model <- lm(Y ~ x)
    jpeg(dest_pilote)
    s3d <- scatterplot3d(x[, 1], x[, 2], Y, pch = 16, highlight.3d = TRUE,
        type = "h", main = "R<U+00E9>gression Y~x")
    s3d$plane3d(model)
    dev.off()
  }
```

plot_mod_ur *plot_mod_univariate_regression*

Usage

```
plot_mod_ur(x, Y, dest_pilote, titre)
```

Arguments

x

Y

dest_pilote

titre

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (x, Y, dest_pilote, titre)
{
    model <- lm(Y ~ x)
    jpeg(dest_pilote)
    plot(x, Y, main = titre)
    abline(model)
    segments(x, fitted(model), x, Y)
    f = floor(min(x))
    c = ceiling(max(x))
    pred.frame <- data.frame(x = seq(f, c, length.out = 5))
    pc <- predict(model, interval = "confidence", newdata = pred.frame)
    pp <- predict(model, interval = "prediction", newdata = pred.frame)
    matlines(pred.frame, pc[, 2:3], lty = c(2, 2), col = "blue")
    matlines(pred.frame, pp[, 2:3], lty = c(3, 3), col = "red")
    legend("topleft", c("confiance", "prediction"), lty = c(2,
        3), col = c("blue", "red"))
    dev.off()
  }
```

Puissance_an *Puissance_anova*

Usage

```
Puissance_an(runs_bs_pilote = 1000, runs_MC = 1000, fact = 0.5, npilote = 20, sd = 1, k = 8, taille_max
```

Arguments

runs_bs_pilote

runs_MC

fact

npilote

sd

k

taille_max

dest_puissance

dest_pilote

puissance

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (runs_bs_pilote = 1000, runs_MC = 1000, fact = 0.5,
    npilote = 20, sd = 1, k = 8, taille_max = 100, dest_puissance,
    dest_pilote, puissance = NULL)
{
    library(pwr)
    library(gplots)
    pilote = pilote_anova(k, fact, npilote, sd, runs_bs_pilote,
        dest_pilote)
    tailles = seq(from = 30, to = taille_max, length.out = 15)
    longueur = length(tailles)
    puissances = numeric(longueur)
    IC_low_width = numeric(longueur)
    IC_up_width = numeric(longueur)
    for (i in 1:longueur) {
        results = ttest_anova(tailles[i], runs_MC, pilote)
        puissances[i] = results$Puissance_moy_hand
        IC_low_width[i] = puissances[i] - results$IC_Puissance_hand_inf
        IC_up_width[i] = results$IC_Puissance_hand_sup - puissances[i]
    }
    jpeg(dest_puissance)
```

```
    plotCI(tailles, puissances, uiw = IC_up_width, liw = IC_low_width,
        type = "o", barcol = "red")
    dev.off()
    results
    return(calcul_n(puissance, puissances, tailles))
  }
```

Puissance_mr *Puissance_multiple_regression*

Usage

```
Puissance_mr(npilote = 20, runs_bs_pilote = 1000, runs_MC = 1000, taille_max = 100, b_0 = NULL, b_1 = N
```

Arguments

npilote

runs_bs_pilote

runs_MC

taille_max

b_0

b_1

b_2

noise_s

dest_puissance

dest_pilote

puissance

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (npilote = 20, runs_bs_pilote = 1000, runs_MC = 1000,
    taille_max = 100, b_0 = NULL, b_1 = NULL, b_2 = NULL, noise_s = NULL,
    dest_puissance, dest_pilote, puissance = NULL)
{
    library(gplots)
    library(regression)
    library(boot)
    library(scatterplot3d)
    pilote = pilote_multiple_reg(npilote, runs_bs_pilote, b_0,
        b_1, b_2, noise_s, dest_pilote)
    tailles = seq(from = 20, to = taille_max, length.out = 15)
    longueur = length(tailles)
```

```
  puissances = rep(0, longueur)
  IC_low_width = numeric(longueur)
  IC_up_width = numeric(longueur)
  for (i in 1:longueur) {
      results = test_multiple(n = tailles[i], runs = runs_MC,
          pilote = pilote)
      puissances[i] = results$Puissance_moy_model_12
      IC_low_width[i] = puissances[i] - results$IC_Puissance_model_12_inf
      IC_up_width[i] = results$IC_Puissance_model_12_sup -
          puissances[i]
  }
  jpeg(dest_puissance)
  plotCI(tailles, puissances, uiw = IC_up_width, liw = IC_low_width,
      type = "o", barcol = "red")
  dev.off()
  results
  return(calcul_n(puissance, puissances, tailles))
}
```

Puissance_ur *Puissance_univariate_regression*

Usage

```
Puissance_ur(npilote = 20, runs_bs_pilote = 1000, runs_MC = 1000, taille_max = 400, b_0 = NULL, b_1 = N
```

Arguments

npilote
runs_bs_pilote
runs_MC
taille_max
b_0
b_1
noise_s
dest_puissance
dest_pilote
puissance

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (npilote = 20, runs_bs_pilote = 1000, runs_MC = 1000,
    taille_max = 400, b_0 = NULL, b_1 = NULL, noise_s = NULL,
    dest_puissance, dest_pilote, puissance = NULL)
{
    library(gplots)
    library(regression)
    library(boot)
    pilote = pilote_univariate_reg(npilote, runs_bs_pilote, b_0,
        b_1, noise_s, dest_pilote)
    tailles = seq(from = 10, to = taille_max, length.out = 15)
    longueur = length(tailles)
    puissances = rep(0, longueur)
    IC_low_width = numeric(longueur)
    IC_up_width = numeric(longueur)
    for (i in 1:longueur) {
        results = test_univariate(n = tailles[i], runs = runs_MC,
            pilote = pilote)
        puissances[i] = results$Puissance_moy_model_1
        IC_low_width[i] = puissances[i] - results$IC_Puissance_model_1_inf
        IC_up_width[i] = results$IC_Puissance_model_1_sup - puissances[i]
    }
    jpeg(dest_puissance)
    plotCI(tailles, puissances, uiw = IC_up_width, liw = IC_low_width,
        type = "o", barcol = "red")
    results
    dev.off()
    return(calcul_n(puissance, puissances, tailles))
  }
```

---

sigma_m *sigma_m*

Usage

```
sigma_m(means, sample_sizes)
```

Arguments

means

sample_sizes

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (means, sample_sizes)
{
    res = 0
    for (i in 1:length(sample_sizes)) {
        res = res + sample_sizes[i] * (means[i] - mean(means))^2
    }
    res = sqrt(res/N(sample_sizes))
    return(res)
  }
```

---

stat_b_0_mr *stat_b_0_mr*

Usage

```
stat_b_0_mr(data, indice)
```

Arguments

```
data
indice
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 3] ~ data[indice, 1:2])
    return(summary(model)[[4]][[1]])
  }
```

---

stat_b_0_un *stat_b_0_un*

Usage

```
stat_b_0_un(data, indice)
```

Arguments

```
data
indice
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 2] ~ data[indice, 1])
    return(summary(model)[[4]][[1]])
  }
```

`stat_b_1_mr` *stat_b_1_mr*

Usage

```
stat_b_1_mr(data, indice)
```

Arguments

`data`

`indice`

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 3] ~ data[indice, 1:2])
    return(summary(model)[[4]][[2]])
  }
```

`stat_b_1_un` *stat_b_1_un*

Usage

```
stat_b_1_un(data, indice)
```

Arguments

`data`

`indice`

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 2] ~ data[indice, 1])
    return(summary(model)[[4]][[2]])
  }
```

---

stat_b_2_mr *stat_b_2_mr*

Usage

```
stat_b_2_mr(data, indice)
```

Arguments

```
data
indice
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 3] ~ data[indice, 1:2])
    return(summary(model)[[4]][[3]])
  }
```

---

stat_noise_s_mr *stat_noise_s_mr*

Usage

```
stat_noise_s_mr(data, indice)
```

Arguments

```
data
indice
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 3] ~ data[indice, 1:2])
    return(summary(model)$sigma)
  }
```

---

stat_noise_s_un *stat_noise_s_un*

Usage

```
stat_noise_s_un(data, indice)
```

Arguments

```
data
```

```
indice
```

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (data, indice)
{
    model = lm(data[indice, 2] ~ data[indice, 1])
    return(summary(model)$sigma)
  }
```

---

test_univariate *test_univariate*

Usage

```
test_univariate(alpha = 0.05, n, runs, pilote)
```

Arguments

alpha

n

runs

pilote

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (alpha = 0.05, n, runs, pilote)
{
    conf_b_0 = pilote$conf_b_0
    conf_b_1 = pilote$conf_b_1
    conf_noise_s = pilote$conf_noise_s
    b_0inf = conf_b_0[1]
    b_0sup = conf_b_0[2]
    b_1inf = conf_b_1[1]
    b_1sup = conf_b_1[2]
    noise_sinf = conf_noise_s[1]
    noise_ssup = conf_noise_s[2]
    b_0 = pilote$b_0
    b_1 = pilote$b_1
    noise_s = pilote$noise_s
    MC_inf = MC_ru(n = n, runs = runs, b_0 = b_0, b_1 = b_1inf,
        noise_s = noise_s)
    MC_moy = MC_ru(n = n, runs = runs, b_0 = b_0, b_1 = b_1,
        noise_s = noise_s)
    MC_sup = MC_ru(n = n, runs = runs, b_0 = b_0, b_1 = b_1sup,
        noise_s = noise_s)
    IC_Puissance_model_0 = c(MC_inf$p5_model_0, MC_sup$p5_model_0)
    IC_Puissance_model_1 = c(MC_inf$p5_model_1, MC_sup$p5_model_1)
    Puissance_moy_model_0 = MC_moy$p5_model_0
    Puissance_moy_model_1 = MC_moy$p5_model_1
   results = data.frame(n = n, runs = runs, Puissance_moy_model_0 = Puissance_moy_model_0,
      Puissance_moy_model_1 = Puissance_moy_model_1, IC_Puissance_model_0_inf = IC_Puissance_model_0[1],
      IC_Puissance_model_0_sup = IC_Puissance_model_0[2], IC_Puissance_model_1_inf = IC_Puissance_model_1[1],
        IC_Puissance_model_1_sup = IC_Puissance_model_1[2])
    return(results)
  }
```

---

ttest_anova *ttest_anova*

Usage

```
ttest_anova(n, runs, pilote)
```

Arguments

n

runs

pilote

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (n, runs, pilote)
{
    sizes = pilote$sizes
    k = length(sizes)
    empiricmeans = pilote$empiricmeans
    conf_mean = pilote$conf_mean
    means = pilote$realmeans
    empiricmeansinf = conf_mean[1, ]
    empiricmeanssup = conf_mean[2, ]
    ICmeaninf_sorted = numeric(k)
    ICmeansup_sorted = numeric(k)
    INF = numeric(k)
    SUP = numeric(k)
    conf_mean_sorted = conf_mean[, order(conf_mean[2, ])]
    ICmeaninf_sorted = conf_mean_sorted[1, ]
    ICmeansup_sorted = conf_mean_sorted[2, ]
    p1 = ICmeaninf_sorted[1]
    pk = ICmeansup_sorted[k]
    SUP[1] = p1
    SUP[k] = pk
    l = pk - p1
    pas = l/(k - 1)
    pcour = p1
    for (i in 2:(k - 1)) {
        opti = pcour + pas
        infcour = ICmeaninf_sorted[i]
        supcour = ICmeansup_sorted[i]
        if (opti < infcour) {
            SUP[i] = infcour
        }
        else {
            if (opti > supcour) {
                SUP[i] = supcour
            }
            else {
                SUP[i] = opti
            }
        }
    }
    p1 = ICmeansup_sorted[1]
```

```
    pk = ICmeaninf_sorted[k]
    if (p1 > pk) {
        SUP = rep(p1, k)
    }
    else {
        INF[1] = p1
        INF[k] = pk
        l = pk - p1
        pas = l/(k - 1)
        pcour = p1
        for (i in 2:(k - 1)) {
            opti = pcour + pas
            infcour = ICmeaninf_sorted[i]
            supcour = ICmeansup_sorted[i]
            if (opti < infcour) {
                INF[i] = infcour
            }
            else {
                if (opti > supcour) {
                  INF[i] = supcour
                }
                else {
                  INF[i] = opti
                }
            }
        }
    }
    sd = pilote$ecart_type
    conf_sd = pilote$conf_sd
    sdinf = mean(conf_sd[1, ])
    sdsup = mean(conf_sd[2, ])
    MC_inf = MC_an(n, runs, INF, sdsup, k)
    MC_sup = MC_an(n, runs, SUP, sdinf, k)
    MC_moy = MC_an(n, runs, empiricmeans, sd, k)
    results = data.frame(runs = runs, Puissance_moy_hand = MC_moy$p5_hand,
        IC_Puissance_hand_inf = MC_inf$p5_hand, IC_Puissance_hand_sup = MC_sup$p5_hand,
      Puissance_moy_package = MC_moy$p5_package, IC_Puissance_package_inf = MC_inf$p5_package,
        IC_Puissance_package_sup = MC_sup$p5_package)
    return(results)
  }
```

`within_group_variance` *NA*

Usage

```
within_group_variance(sample_sizes, means, y)
```

Arguments

sample_sizes

means

y

Author(s)

Bonjean Gregoire, Crepin Baptiste & Lair Thomas

Examples

```
function (sample_sizes, means, y)
{
    wgv = 0
    indice = 0
    for (i in 1:length(sample_sizes)) {
        for (j in 1:sample_sizes[i]) {
            wgv = wgv + (y[indice + j] - means[i])^2
        }
        indice = indice + sample_sizes[i]
    }
    return(wgv)
  }
```

Index

∗Topic **package**
anova.reg-package, 2

anova.reg *(*anova.reg-package*)*, 2
anova.reg-package, 2

between_group_variance, 3

calcul_n, 3

F, 4

MC_an, 5
MC_rm, 6
MC_ru, 7

N, 8

pilote_anova, 8
pilote_multiple_reg, 9
plot_mod_mr, 11
plot_mod_ur, 12
Puissance_an, 13
Puissance_mr, 14
Puissance_ur, 15

sigma_m, 16
stat_b_0_mr, 17
stat_b_0_un, 17
stat_b_1_mr, 18
stat_b_1_un, 18
stat_b_2_mr, 19
stat_noise_s_mr, 19
stat_noise_s_un, 20

test_univariate, 20
ttest_anova, 21

within_group_variance, 23