

Un script d'étalonnage pour débutant

```
# Charger G-Séries en R (librairie gstest)
library(gstest)

# Définir le répertoire de travail (pour les fichiers graphiques PDF)
iniwd <- getwd()
setwd("C:/Temp")

#### Exemple 1 (une série trimestrielle) ####
# Cas simple d'étalonnage d'une série trimestrielle à des valeurs annuelles

# Série indicatrice trimestrielle
ma_sc_tri <- ts(c(1.9, 2.4, 3.1, 2.2, 2.0, 2.6, 3.4, 2.4, 2.3),
               start = c(2015, 1),
               frequency = 4)
mes_ind1 <- ts_to_tsDF(ma_sc_tri)

# Étalons annuels pour données trimestrielles
ma_sc_ann <- ts(c(10.3, 10.2),
               start = 2015,
               frequency = 1)
mes_eta1 <- ts_to_bmkDF(ma_sc_ann, ind_frequency = 4)

# Étalonnage avec...
# - valeur de `rho` recommandée pour des séries trimestrielles (`rho = 0.729`)
# - modèle proportionnel (`lambda = 1`)
# - correction de la série indicatrice pour le biais avec estimation du biais
#   (`biasOption = 3`)
res_eta1 <- benchmarking(mes_ind1,
                        mes_eta1,
                        rho = 0.729,
                        lambda = 1,
                        biasOption = 3)

# Sorties
View(res_eta1$series)
View(res_eta1$benchmarks)
View(res_eta1$graphTable)

# Graphiques (ensemble par défaut)
plot_graphTable(res_eta1$graphTable,
                "Graphs_ex1.pdf")

# Avec la table des taux de croissance (« growth rates »)
```

```

plot_graphTable(res_eta1$graphTable,
  "Graphs_ex1_avec_TableTC.pdf",
  GR_table_flag = FALSE)

# Comparer avec la librairie tempdisagg qui offre l'étalonnage de Denton
# (version modifiée par Cholette)

#install.packages("tempdisagg")
library(tempdisagg)
#install.packages("dplyr")
library(dplyr)

# Denton proportionnel
denton_mult <- benchmarking(mes_ind1,
  mes_eta1,
  rho = 1,
  lambda = 1)

denton_mult_td <- td(formula = ma_sc_ann ~ 0 + ma_sc_tri,
  method = "denton-cholette",
  criterion = "proportional")

comp_mult <- ts.union(gstest = tsDF_to_ts(denton_mult$series, frequency = 4),
  tempdisagg = predict(denton_mult_td),
  dframe = TRUE) %>%
  cbind(., diff = .[, 1] - .[, 2])

# Mêmes résultats
View(comp_mult)

# Denton additif
denton_add <- benchmarking(mes_ind1,
  mes_eta1,
  rho = 1,
  lambda = 0)

denton_add_td <- td(formula = ma_sc_ann ~ 0 + ma_sc_tri,
  method = "denton-cholette",
  criterion = "additive")

comp_add <- ts.union(gstest = tsDF_to_ts(denton_add$series, frequency = 4),
  tempdisagg = predict(denton_add_td),
  dframe = TRUE) %>%
  cbind(., diff = .[, 1] - .[, 2])

# Mêmes résultats
View(comp_add)

```

```
#### Exemple 2 (plusieurs séries trimestrielles) ####
# Deux séries trimestrielles à étalonner à des valeurs annuelles,
# avec groupes-BY et coef. d'altérabilité définis par l'utilisateur.

# Données sur les ventes (mêmes ventes pour les groupes A et B; seuls les coef.
# d'alt. pour les ventes de camionnettes différent)
ventes_tri <- ts(matrix(c(# Voitures
                        1851, 2436, 3115, 2205, 1987, 2635, 3435, 2361, 2183, 2822,
                        3664, 2550, 2342, 3001, 3779, 2538, 2363, 3090, 3807, 2631,
                        2601, 3063, 3961, 2774, 2476, 3083, 3864, 2773, 2489, 3082,
                        # Camionnettes
                        1900, 2200, 3000, 2000, 1900, 2500, 3800, 2500, 2100, 3100,
                        3650, 2950, 3300, 4000, 3290, 2600, 2010, 3600, 3500, 2100,
                        2050, 3500, 4290, 2800, 2770, 3080, 3100, 2800, 3100, 2860),
                    ncol = 2),
                start = c(2011, 1),
                frequency = 4,
                names = c("voitures", "camionnettes"))

class(ventes_tri)

ventes_ann <- ts(matrix(c(# Voitures
                        10324, 10200, 10582, 11097, 11582, 11092,
                        # Camionnettes
                        12000, 10400, 11550, 11400, 14500, 16000),
                    ncol = 2),
                start = 2011,
                frequency = 1,
                names = c("voitures", "camionnettes"))

# Séries indicatrices trimestrielles (avec les coef. d'alt. par défaut pour l'instant)
mes_ind2 <- rbind(cbind(data.frame(groupe = rep("A", nrow(ventes_tri)),
                                alt_cam = rep(1, nrow(ventes_tri))),
                ts_to_tsDF(ventes_tri)),
                cbind(data.frame(groupe = rep("B", nrow(ventes_tri)),
                                alt_cam = rep(1, nrow(ventes_tri))),
                ts_to_tsDF(ventes_tri)))

# Ventes contraignantes de camionnettes (coef. d'alt. = 0) pour 2012 T1 et T2
# dans le groupe A (lignes 5 et 6)
mes_ind2$alt_cam[c(5,6)] <- 0

# Étalons annuels pour données trimestrielles (sans coef. d'alt.)
mes_eta2 <- rbind(cbind(data.frame(groupe = rep("A", nrow(ventes_ann))),
                                ts_to_bmkDF(ventes_ann, ind_frequency = 4)),
                cbind(data.frame(groupe = rep("B", nrow(ventes_ann))),
                ts_to_bmkDF(ventes_ann, ind_frequency = 4)))

# Étalonnage avec...
# - valeur de `rho` recommandée pour des séries trimestrielles (`rho = 0.729`)
# - modèle proportionnel (`lambda = 1`)
# - sans correction du biais (`biasOption = 1` et `bias` non spécifié)
# - `quiet = TRUE` afin d'éviter l'affichage de l'en-tête de la fonction
```

```

res_eta2 <- benchmarking(mes_ind2,
                        mes_eta2,
                        rho = 0.729,
                        lambda = 1,
                        biasOption = 1,
                        var = c("voitures", "camionnettes / alt_cam"),
                        with = c("voitures", "camionnettes"),
                        by = "groupe")

# Sorties
View(res_eta2$series)
View(res_eta2$benchmarks)
View(res_eta2$graphTable)

# Graphiques
plot_graphTable(res_eta2$graphTable,
                "Graphs_ex2.pdf")

#### Exemple 3 (plusieurs séries trimestrielles) ####
# Identique à l'exemple 2, mais en étalonnant les 4 séries
# en tant que groupes-BY (4 groupes-BY au lieu de 2)

mes_ind3 <- stack_tsDF(ts_to_tsDF(ts.union(A = ventes_tri, B = ventes_tri)))
mes_ind3$alter <- 1
mes_ind3$alter[mes_ind3$series == "A.camionnettes"
               & mes_ind3$year == 2012 & mes_ind3$period <= 2] <- 0
mes_eta3 <- stack_bmkDF(ts_to_bmkDF(ts.union(A = ventes_ann, B = ventes_ann),
                                     ind_frequency = 4))
res_eta3 <- benchmarking(mes_ind3,
                        mes_eta3,
                        rho = 0.729,
                        lambda = 1,
                        biasOption = 1,
                        var = "value / alter",
                        with = "value",
                        by = "series")

# Sorties
View(res_eta3$series)
View(res_eta3$benchmarks)
View(res_eta3$graphTable)

# Graphiques
plot_graphTable(res_eta3$graphTable,
                "Graphs_ex3.pdf")

# Convertir la série étalonnée en objet "mts"
mes_ind3_eta <- tsDF_to_ts(unstack_tsDF(res_eta3$series), frequency = 4)
class(mes_ind3_eta)
plot(mes_ind3_eta)

```

```

#### Données mensuelles (série de Box & Jenkins sur les passagers aériens) ####

data(AirPassengers)
mes_ind_AP <- ts_to_tsDF(AirPassengers)

# Créer des étalons annuels en changeant le niveau (5 fois plus grand), en ajoutant
# du bruit aléatoire et en laissant tomber les 2 derniers étalons.
set.seed(as.Date("2003-03-25")) # pour reproduire les résultats (date de votre choix)
#set.seed(NULL)
AP_ann <- round(jitter(aggregate.ts(AirPassengers, nfrequency = 1, FUN = sum) * 5,
                                   amount = 2500))
mes_eta_AP <- (ts_to_bmkDF(AP_ann, ind_frequency = 12))[1:10, ]

# Avec correction pour le biais (estimé avec `biasOption = 3`)
# => tout semble bien
res_eta_AP <- benchmarking(mes_ind_AP,
                           mes_eta_AP,
                           rho = 0.9,
                           lambda = 1,
                           biasOption = 3)

View(res_eta_AP$series)
View(res_eta_AP$benchmarks)
View(res_eta_AP$graphTable)

plot_graphTable(res_eta_AP$graphTable,
                "Graphs_AP.pdf")

# Sans correction pour le biais (`biasOption = 1`)
# => problèmes avec les ajustements projetés en fin de série pour les périodes
# non couvertes par un étalon
res_eta_AP_sansBiais <- benchmarking(mes_ind_AP,
                                     mes_eta_AP,
                                     rho = 0.9,
                                     lambda = 1,
                                     biasOption = 1)

View(res_eta_AP_sansBiais$series)
View(res_eta_AP_sansBiais$benchmarks)
View(res_eta_AP_sansBiais$graphTable)

plot_graphTable(res_eta_AP_sansBiais$graphTable,
                "Graphs_AP_sansBiais.pdf")

# Étalonnage de Denton (`rho = 1`, correction du biais non pertinente)
# => dernier ajustement répété (à jamais) en fin de série
# (hypothèse forte, mais pas nécessairement problématique)
res_eta_AP_denton <- benchmarking(mes_ind_AP,
                                  mes_eta_AP,

```

```

                                rho = 1,
                                lambda = 1)
View(res_eta_AP_denton$series)
View(res_eta_AP_denton$benchmarks)
View(res_eta_AP_denton$graphTable)

plot_graphTable(res_eta_AP_denton$graphTable,
                 "Graphs_AP_Denton.pdf")

# Étalonnage approximatif de Denton (`rho = 0.999`, `biasOption = 3`)
#   => approximation à l'aide du modèle de régression d'étalonnage
#   => dernier ajustement répété en fin de série (légère convergence vers le biais
#         comparativement à la méthode de Denton « pure »)
res_eta_AP_approxDenton <- benchmarking(mes_ind_AP,
                                         mes_eta_AP,
                                         rho = 0.999,
                                         lambda = 1,
                                         biasOption = 3)

View(res_eta_AP_approxDenton$series)
View(res_eta_AP_approxDenton$benchmarks)
View(res_eta_AP_approxDenton$graphTable)

plot_graphTable(res_eta_AP_approxDenton$graphTable,
                 "Graphs_AP_approxDenton.pdf")

# Prorata (`lambda = 0.5` et `rho = 0`)
#   => aucune préservation du mouvement: tous les ajustements sont regroupés en janvier
#       à chaque année (avec un impact désastreux sur certains des mouvements initiaux
#       de décembre à janvier)
#   => convergence immédiate vers le biais (estimé avec `biasOption = 3`) pour les
#       ajustements projetés en fin de série
res_eta_AP_prorata <- benchmarking(mes_ind_AP,
                                   mes_eta_AP,
                                   rho = 0,
                                   lambda = 0.5,
                                   biasOption = 3)

View(res_eta_AP_prorata$series)
View(res_eta_AP_prorata$benchmarks)
View(res_eta_AP_prorata$graphTable)

plot_graphTable(res_eta_AP_prorata$graphTable,
                 "Graphs_AP_prorata.pdf")

#### Stocks de fin d'année (« bris » dans les ajustements avec `benchmarking()`) ####

# Série de stocks trimestriels (même patron répété chaque année)
mes_ind_stock <- ts_to_tsDF(ts(rep(c(85, 95, 125, 95), 7),
                                start = c(2013, 1),

```

```

frequency = 4))

# Étalons annuels (stocks de fin d'année)
mes_eta_stock <- ts_to_bmkDF(ts(c(135, 125, 155, 145, 165),
                                start = 2013,
                                frequency = 1),
                             discrete_flag = TRUE,
                             alignment = "e",
                             ind_frequency = 4)

# Avec `benchmarking()` (approche « Proc Benchmarking »)
res_stock_PB <- benchmarking(mes_ind_stock,
                             mes_eta_stock,
                             rho = 0.729,
                             lambda = 1,
                             biasOption = 3)

# Avec `stock_benchmarking()` (approche « Stock Benchmarking »)
res_stock_SB <- stock_benchmarking(mes_ind_stock,
                                    mes_eta_stock,
                                    rho = 0.729,
                                    lambda = 1,
                                    biasOption = 3)

# Ajustements d'étalonnage des deux approches
plot_benchAdj(PB_graphTable = res_stock_PB$graphTable,
              SB_graphTable = res_stock_SB$graphTable)

# Avez-vous remarqué que les ajustements de `stock_benchmarking()` sont plus lisses
# que ceux de `benchmarking()` ?

# L'amélioration de la qualité des données étalonnées qui en résulte n'est pas
# nécessairement évidente dans cet exemple.
plot(res_stock_SB$graphTable$t, res_stock_SB$graphTable$benchmarked,
     type = "b", col = "red", xlab = "t", ylab = "Stocks étalonnés")
lines(res_stock_PB$graphTable$t, res_stock_PB$graphTable$benchmarked,
      type = "b", col = "blue")
legend(x = "topleft", bty = "n", inset = 0.05, lty = 1, pch = 1,
      col = c("red", "blue"), legend = c("res_stock_SB", "res_stock_PB"))
title("Stocks étalonnés")

# Graphiques PDF
plot_graphTable(res_stock_PB$graphTable, "Graphs_stock_PB.pdf")
plot_graphTable(res_stock_SB$graphTable, "Graphs_stock_SB.pdf")

# Qu'en est-il des cas où un indicateur plat (rectiligne) est utilisé, ce qui se produit
# souvent en pratique en l'absence d'un bon indicateur des mouvements infra-annuels ?
mes_ind_plat <- mes_ind_stock
mes_ind_plat$value <- 1
res_stock_PB2 <- benchmarking(mes_ind_plat,
                             mes_eta_stock,
                             rho = 0.729,
                             lambda = 1,

```

```

                                biasOption = 3)
res_stock_SB2 <- stock_benchmarking(mes_ind_plat,
                                    mes_eta_stock,
                                    rho = 0.729,
                                    lambda = 1,
                                    biasOption = 3)

plot(res_stock_SB2$graphTable$t, res_stock_SB2$graphTable$benchmarked,
     type = "b", col = "red", xlab = "t", ylab = "Stocks étalonnés")
lines(res_stock_PB2$graphTable$t, res_stock_PB2$graphTable$benchmarked,
      type = "b", col = "blue")
legend(x = "topleft", bty = "n", inset = 0.05, lty = 1, pch = 1,
      col = c("red", "blue"), legend = c("res_stock_SB2", "res_stock_PB2"))
title("Stocks étalonnés - Indicateur plat")

# L'apparence plutôt étrange des valeurs étalonnées produites par `benchmarking()` devient
# soudainement plus évidente. En effet, la série étalonnée correspond aux ajustements
# d'étalonnage lorsqu'on utilise un indicateur plat (par exemple, une série de 1 avec
# un étalonnage proportionnel) :
plot_benchAdj(PB_graphTable = res_stock_PB2$graphTable,
              SB_graphTable = res_stock_SB2$graphTable)

# Les lacunes de l'approche « Proc Benchmarking » (fonction `benchmarking()`) avec
# des stocks sont également très visibles lorsque l'on regarde les taux de croissance
# trimestriels résultants, qui sont commodément produits par `plot_graphTable()`.
# Portez une attention particulière à la transition des taux de croissance de T4 à T1
# à chaque année dans les graphiques PDF générés.
plot_graphTable(res_stock_PB2$graphTable, "Graphs_stock_PB_indicateur_plat.pdf")
plot_graphTable(res_stock_SB2$graphTable, "Graphs_stock_SB_indicateur_plat.pdf")

# Réinitialiser le répertoire de travail à son emplacement initial
setwd(iniwd)

```