Séries temporelles univariées

TD5 : Évaluation de modèles de prévision

HOUSSAIS Rémi, CROCHET Florian
M1 ECAP – Année 2024/2025

14 octobre 2025

Responsable d'enseignement : Benoît SÉVI
benoit.sevi@univ-nantes.fr

Sommaire

1 Chargement des librairies

```r
library(readxl)
library(tidyverse)
library(tseries)
library(forecast)
library(gridExtra)
library(lmtest)
library(sandwich)
```

```r
rend_ble <- read_excel("data/wheat_futures_returns_2006_2022.xlsx")
```

```r
rend_ble_ts <- ts(
  data = rend_ble$return,
  start = c(2006, 1, 2),
  frequency = 252
)
```

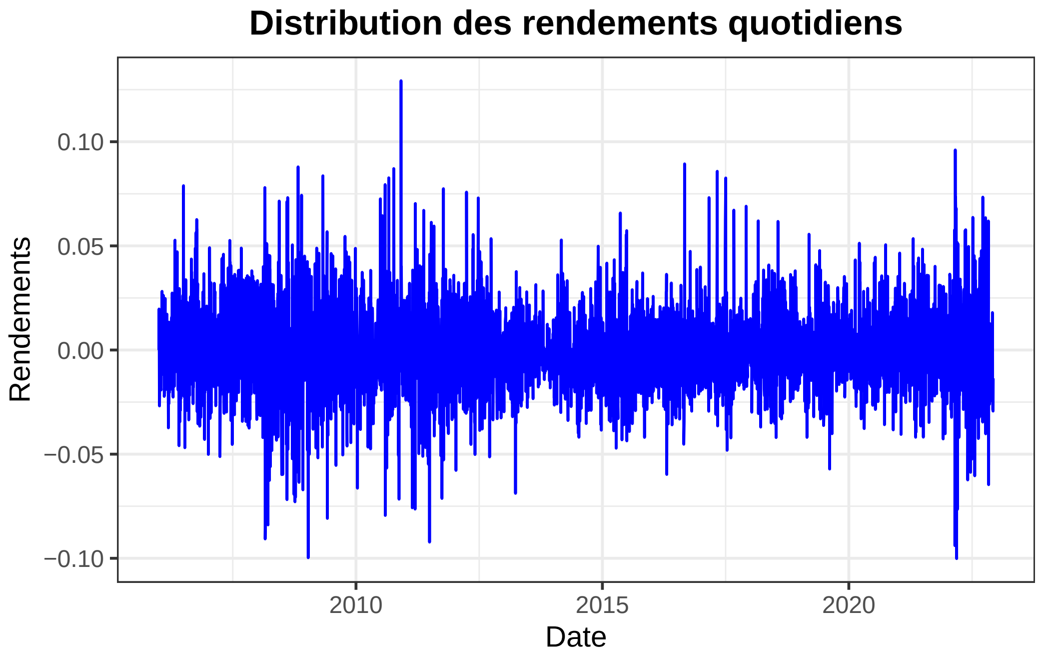
```r
# Fonction pour afficher les graphiques

fonction_graph <- function(df, variable, titre_variable, titre_graph) {
  df |>
    ggplot() +
    aes(x = date, y = !!sym(variable)) +
    geom_line(color = "blue", size = 0.5, na.rm = TRUE) +
    labs(
      title = titre_graph,
      x = "Date",
      y = titre_variable
    ) +
    theme_bw() +
    theme(
      plot.title = element_text(hjust = 0.5, face = "bold")
    )
}
```

2 Distribution des rendements quotidiens

```
graph_rend_ble <- fonction_graph(
  rend_ble,
  "return",
  "Rendements",
  "Distribution des rendements quotidiens"
)
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
graph_rend_ble
```



3 Vérification de la stationnarité

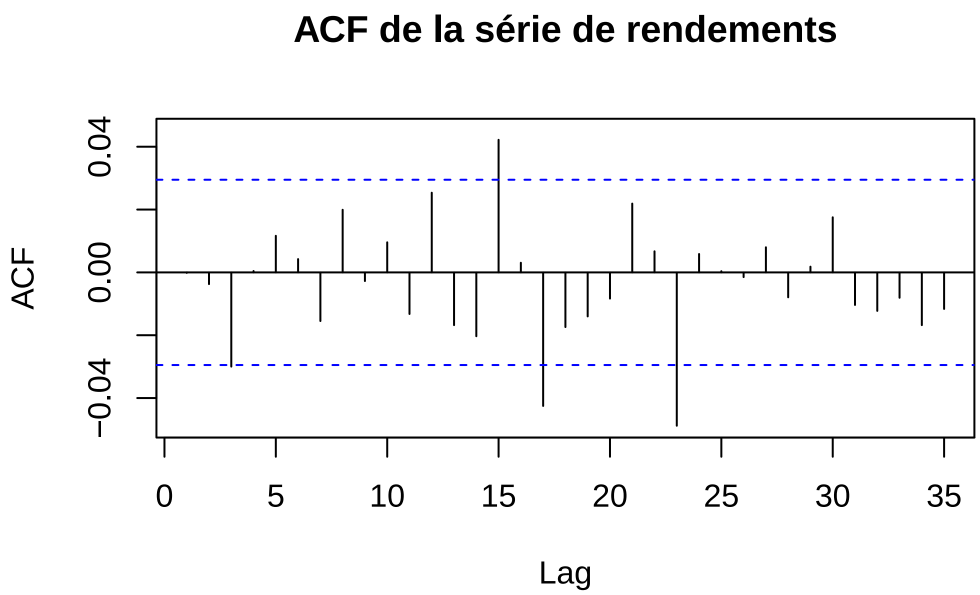
```r
adf.test(rend_ble_ts)
```

```
Warning in adf.test(rend_ble_ts): p-value smaller than printed p-value


    Augmented Dickey-Fuller Test

data:  rend_ble_ts
Dickey-Fuller = -16.304, Lag order = 16, p-value = 0.01
alternative hypothesis: stationary
```

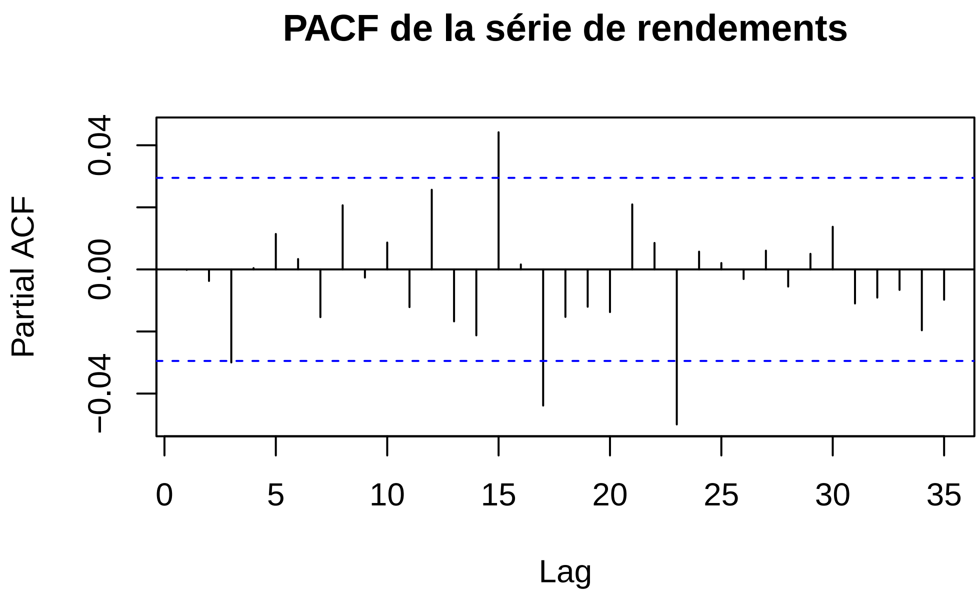
Le test de Dickey-Fuller augmenté (ADF) permet de vérifier si une série temporelle est stationnaire. Les hypothèses du test sont les suivantes : H0 stipule que la série a une racine unitaire, donc qu'elle est non stationnaire, tandis que H1 indique que la série est stationnaire. Les résultats du test montrent une statistique de test de -16.304 et une p-value inférieure à 0,01, ce qui est très faible. Par conséquent, on rejette l'hypothèse nulle (H0) au niveau de 1%, concluant que la série des rendements est stationnaire.

4 Spécification de type ARMA

```
Acf(
  rend_ble_ts,
  main = "ACF de la série de rendements",
  lag.max = 35
)
```




```
Pacf(
  rend_ble_ts,
  main = "PACF de la série de rendements",
  lag.max = 35
)
```

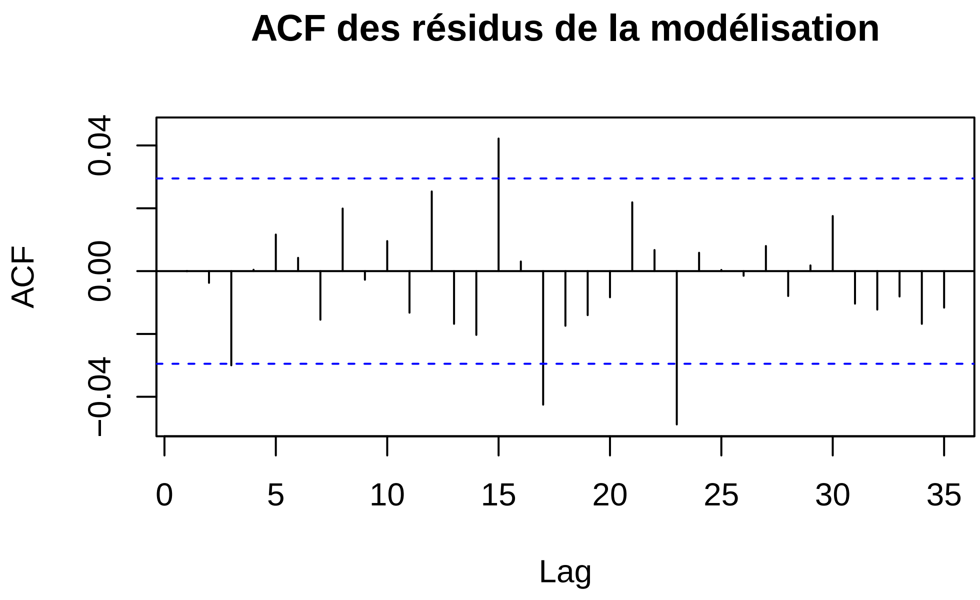


L'analyse des fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF) révèle l'absence d'autocorrélations significatives aux premiers retards. Nous choisissons donc de modéliser la série à l'aide d'un modèle autorégressif d'ordre 1 (AR(1)).

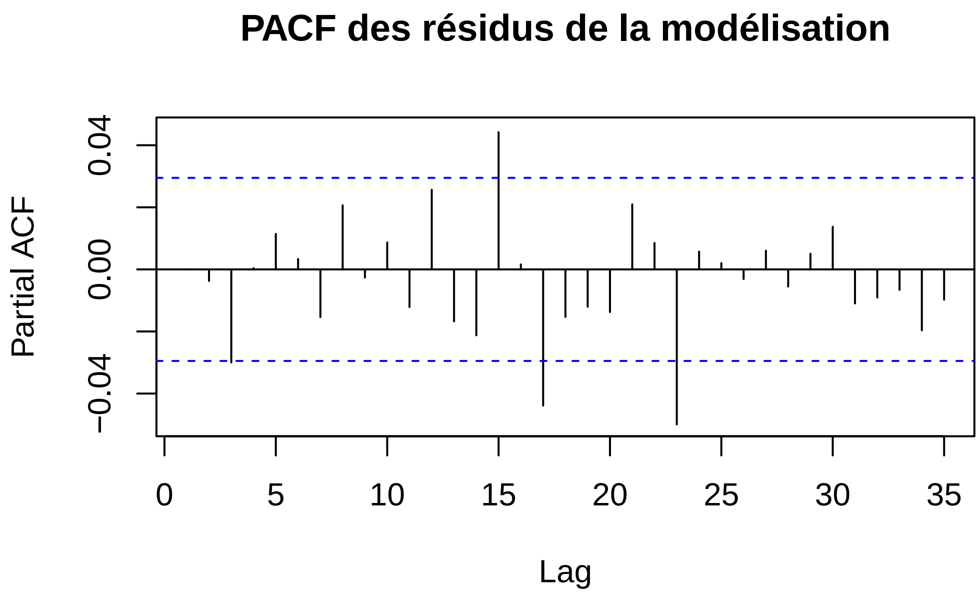
```r
modele_ar1 <- Arima(rend_ble_ts, order = c(1, 0, 0))
summary(modele_ar1)
```

```
Series: rend_ble_ts
ARIMA(1,0,0) with non-zero mean

Coefficients:
          ar1   mean
      -0.0002  2e-04
s.e.   0.0150  3e-04

sigma^2 = 0.0004297:  log likelihood = 10854.96
AIC=-21703.92   AICc=-21703.92   BIC=-21684.74

Training set error measures:
                        ME       RMSE        MAE  MPE MAPE      MASE
Training set -3.70248e-10 0.02072364 0.01522359 -Inf  Inf 0.6912772
                      ACF1
Training set -6.348745e-07
```

```r
Acf(
  modele_ar1$residuals,
  main = "ACF des résidus de la modélisation",
  lag.max = 35
)
```




```r
Pacf(
  modele_ar1$residuals,
  main = "PACF des résidus de la modélisation",
  lag.max = 35
)
```



L'analyse de l'ACF et de la PACF des résidus montre que les autocorrélations ne sont pas significatives, ou sont proches de ne pas l'être, pour les retards considérés. Les résidus semblent donc se comporter comme un bruit blanc. Par conséquent, nous conservons le modèle autorégressif d'ordre 1.

5 Tableau comparatif et graphiques pour les prévisions à 1 et 5 jours

5.1 Nombre de jours

```
# Estimation sur 10 ans

date_debut_est_10 <- rend_ble[[1, 1]]
date_fin_est <- date_debut_est_10 + years(10)

nb_est_10 <- rend_ble |>
  filter(date >= date_debut_est_10 & date <= date_fin_est) |>
  nrow()
nb_est_10
```

```
[1] 2610
```

```
# Estimation sur 3 ans

date_debut_est_3 <- date_fin_est - years(3)
# date_fin_est

nb_est_3 <- rend_ble |>
  filter(date >= date_debut_est_3 & date <= date_fin_est) |>
  nrow()
nb_est_3
```

```
[1] 783
```

```
# Prévision

date_debut_prev <- date_fin_est + days(1)
date_fin_prev <- rend_ble[[nrow(rend_ble), 1]]

nb_prev <- rend_ble |>
  filter(date >= date_debut_prev & date <= date_fin_prev) |>
  nrow()
nb_prev
```

```
[1] 1807
```

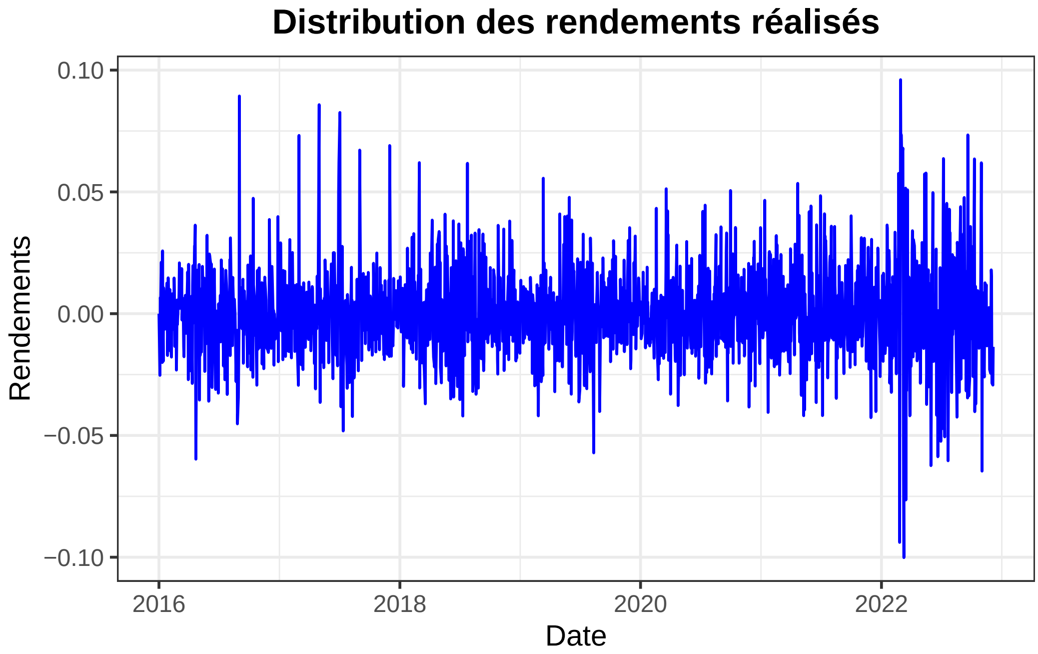
5.2 Tableau comparatif

```r
set.seed(246)

T <- nb_prev + 1

# Initialisation du tableau de résultats

tableau_comparatif <- tibble(
  date = rep(NA_Date_, T),
  realisation = rep(NA_real_, T),
  marche_aleatoire = rep(NA_real_, T),
  phi1_10 = rep(NA_real_, T),
  phi1_3 = rep(NA_real_, T),
  A10_1 = rep(NA_real_, T),
  A3_1 = rep(NA_real_, T),
  A10_5 = rep(NA_real_, T),
  A3_5 = rep(NA_real_, T)
)


for (t in 1:T) {
  derniere_est <- nb_est_10 + t - 1
  date_t <- rend_ble$date[derniere_est]
  realisation_t <- rend_ble$return[derniere_est]

  # Valeurs par défaut (NA)
  marche_aleatoire <-
    A10_1 <-
    A3_1 <-
    A10_5 <-
    A3_5 <-
    phi1_10 <-
    phi1_3 <-
    NA

  # Yt utilisé pour les prévisions à 1 jour
  if (t >= 2) {
    realisation_1j <- rend_ble$return[derniere_est - 1]

    epsilon <- rnorm(1, mean = 0, sd = 1)

    marche_aleatoire <- realisation_1j + epsilon

    # Coefficients AR(1)
    phi1_10 <- coef(
      Arima(
        rend_ble$return[t:derniere_est],
```

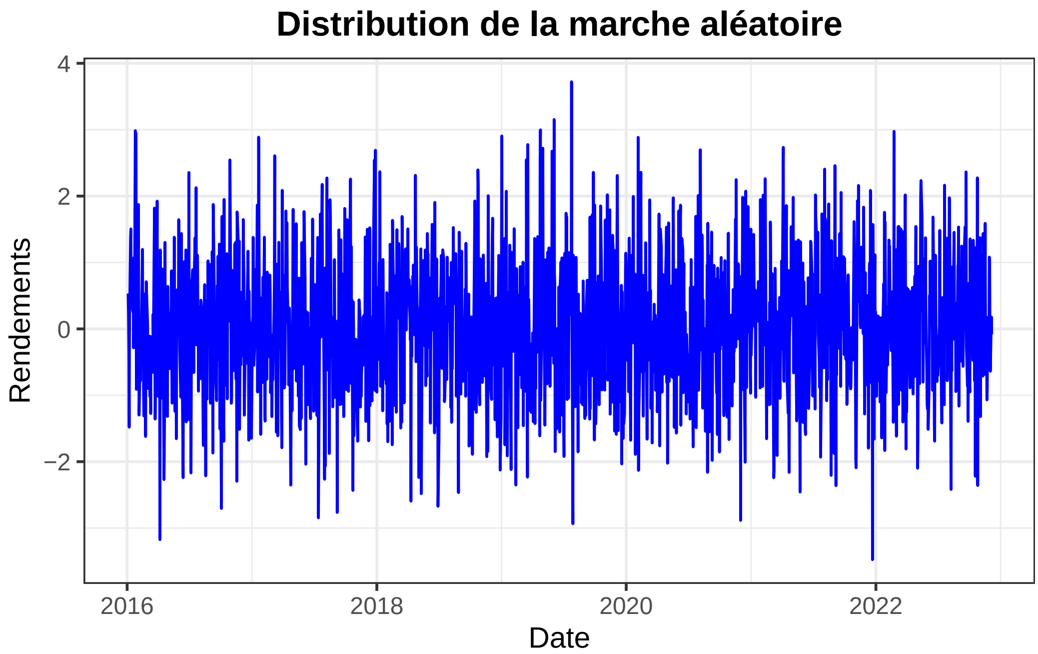
```
      order = c(1, 0, 0)
    )
  )[1]

  phi1_3 <- coef(
    Arima(
      rend_ble$return[(derniere_est - nb_est_3):derniere_est],
      order = c(1, 0, 0)
    )
  )[1]

  # Prévision à 1 jour
  A10_1 <- phi1_10 * realisation_1j
  A3_1 <- phi1_3 * realisation_1j
}


# Yt utilisé pour les prévisions à 5 jours
if (t >= 6) {
  realisation_5j <- rend_ble$return[derniere_est - 5]
  A10_5 <- phi1_10^5 * realisation_5j
  A3_5 <- phi1_3^5 * realisation_5j
}


# Remplissage du tableau
tableau_comparatif[t, ] <- list(
  date_t, realisation_t, marche_aleatoire,
  phi1_10, phi1_3, A10_1, A3_1, A10_5, A3_5
)
}
```

5.3 Représentations graphiques

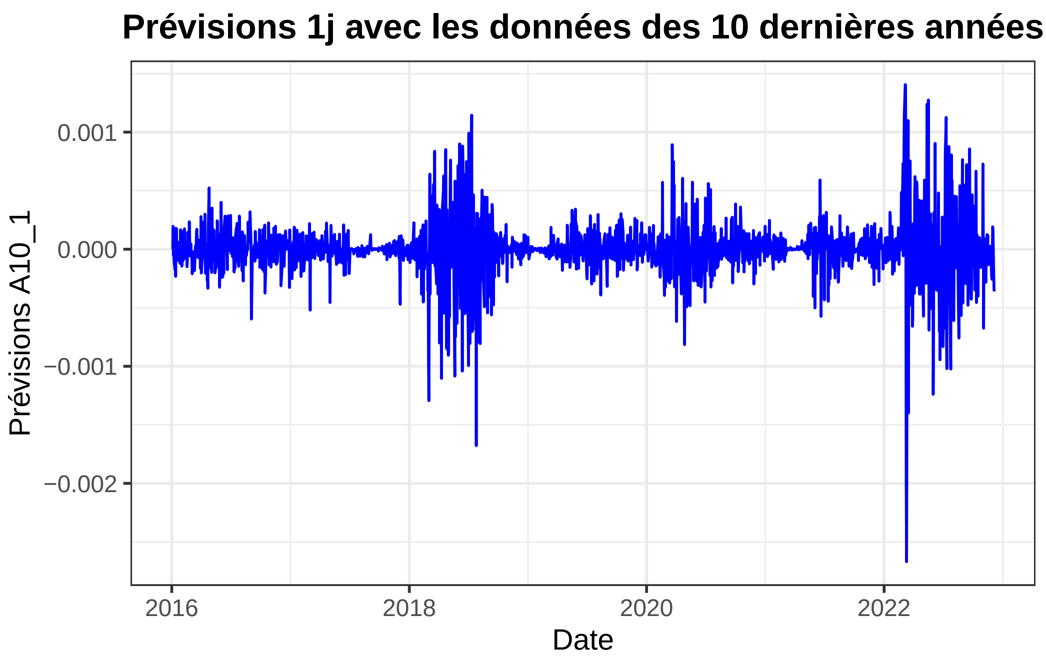
```r
# Paramètres
params <- list(
  list(
    "realisation",
    "Rendements",
    "Distribution des rendements réalisés"
  ),
  list(
    "marche_aleatoire",
    "Rendements",
    "Distribution de la marche aléatoire"
  ),
  list(
    "A10_1",
    "Prévisions A10_1",
    "Prévisions 1j avec les données des 10 dernières années"
  ),
  list(
    "A3_1",
    "Prévisions A3_1",
    "Prévisions 1j avec les données des 3 dernières années"
  ),
  list(
    "A10_5",
    "Prévisions A10_5",
    "Prévisions 5j avec les données des 10 dernières années"
  ),
  list(
    "A3_5",
    "Prévisions A3_5",
    "Prévisions 5j avec les données des 3 dernières années"
  )
)

# Affichage
walk(params, ~ {
  p <- fonction_graph(tableau_comparatif, .x[[1]], .x[[2]], .x[[3]])
  print(p)
})
```

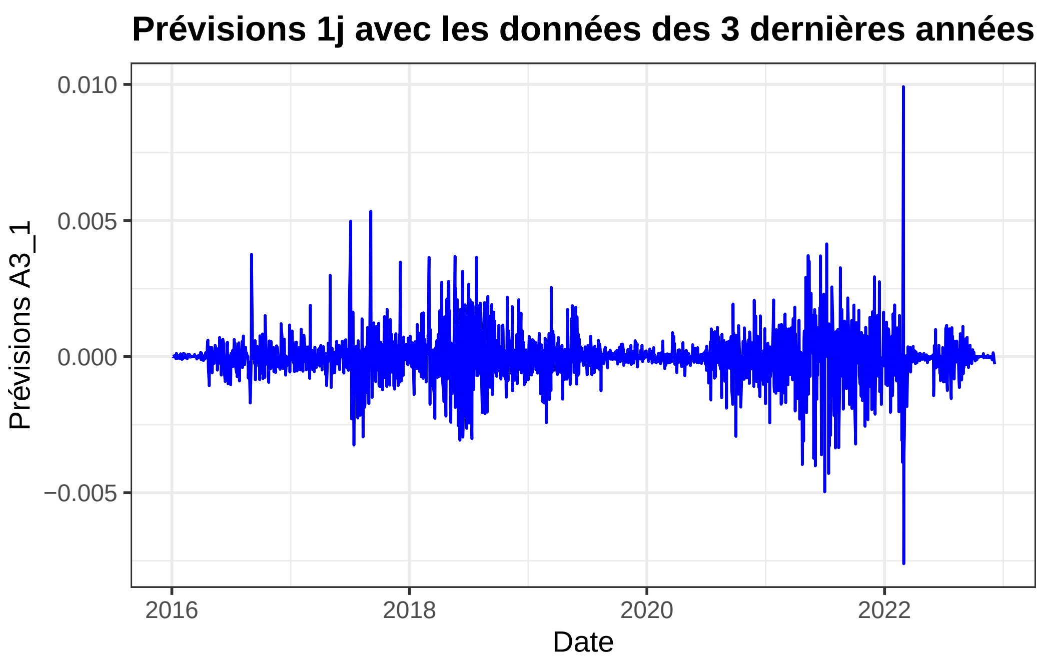


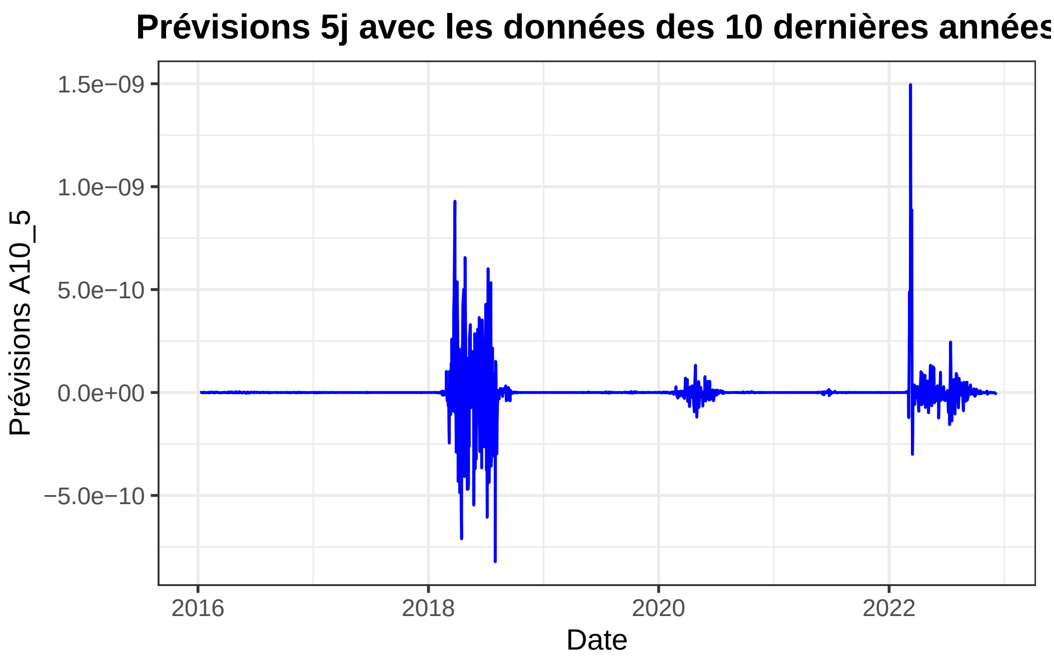
Distribution des rendements réalisés
0.10
0.05
0.00
−0.05
−0.10
Rendements
2016
2018
2020
2022
Date

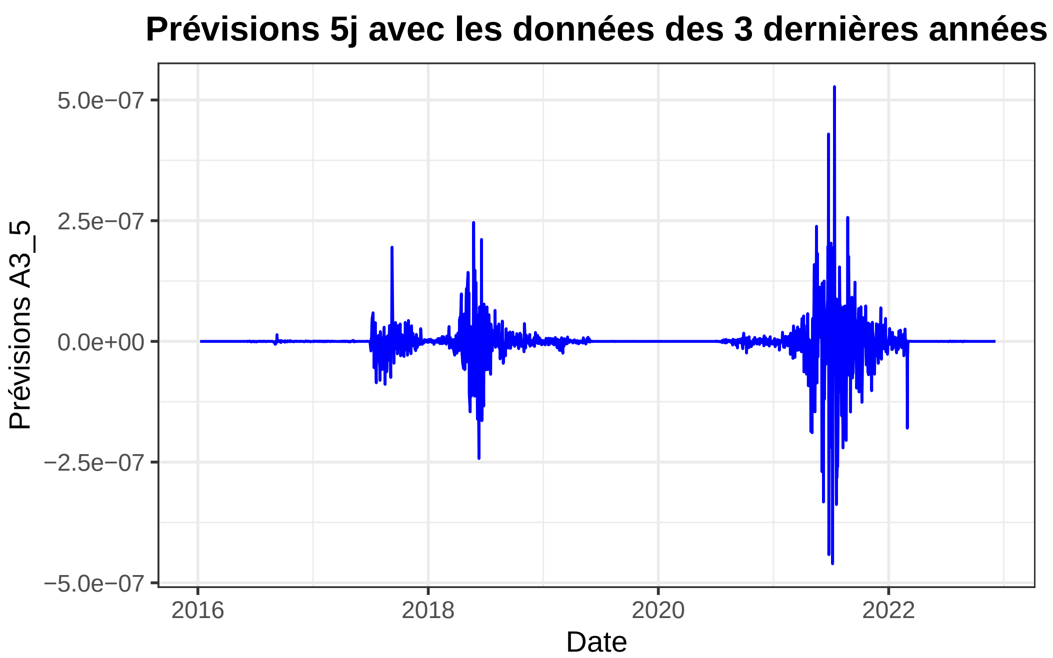


Distribution de la marche aléatoire
4
2
0
−2
Rendements
2016
2018
2020
2022
Date


Prévisions 1j avec les données des 10 dernières années
0.001
0.000
−0.001
−0.002
Prévisions A10_1
2016
2018
2020
2022
Date


Prévisions 1j avec les données des 3 dernières années
0.010
0.005
0.000
−0.005
Prévisions A3_1
2016
2018
2020
2022
Date


Prévisions 5j avec les données des 10 dernières années
Prévisions A10_5
1.5e−09
1.0e−09
5.0e−10
0.0e+00
−5.0e−10
2016
2018
2020
2022
Date


Prévisions 5j avec les données des 3 dernières années
Prévisions A3_5
5.0e−07
2.5e−07
0.0e+00
−2.5e−07
−5.0e−07
2016
2018
2020
2022
Date

6 Mincer-Zarnowitz

6.1 Fonction

```
fonction_mz <- function(y, prevision, nom_prevision) {
  # ------------------------------
  #            Modèle
  # ------------------------------

  modele <- lm(y ~ prevision) # régression
  resume <- summary(modele)


  # ------------------------------
  #            alpha
  # ------------------------------

  alpha_est <- resume$coefficients["(Intercept)", "Estimate"]

  alpha_pvalue <- resume$coefficients["(Intercept)", "Pr(>|t|)"]

  ## Vérification de l'hypothèse alpha = 0
  H0_alpha <- if (alpha_pvalue < 0.05) "rejetée" else "non rejetée"


  # ------------------------------
  #            beta
  # ------------------------------

  beta_est <- resume$coefficients["prevision", "Estimate"]
  beta_sd <- resume$coefficients["prevision", "Std. Error"]

  beta_tstat <- (beta_est - 1) / beta_sd # Statistique de test pour beta = 1

  ddl <- resume$df[2] # Degrés de liberté

  beta_pvalue <- 2 * pt(-abs(beta_tstat), ddl) # Test bilatérale

  ## Vérification de l'hypothèse beta = 1
  H0_beta <- if (beta_pvalue < 0.05) "rejetée" else "non rejetée"


  # ------------------------------
  #             Test
  # ------------------------------

  ## Vérification de l'hypothèse alpha = 0 et beta = 1
```

```r
  H0 <- if (H0_alpha == "rejetée" | H0_beta == "rejetée") {
    "H0 rejetée"
  } else {
    "H0 non rejetée"
  }


  # ------------------------------
  #           Résultat
  # ------------------------------

  tableau_mz <- tibble(
    nom_prevision = nom_prevision,
    alpha = alpha_est,
    beta = beta_est,
    alpha_pvalue = alpha_pvalue,
    beta_pvalue = beta_pvalue,
    H0_alpha = H0_alpha,
    H0_beta = H0_beta,
    H0 = H0
  )

  return(tableau_mz)
}
```

6.2 Résultat

```r
# Modèle de référence : marche_aleatoire
nom_previsions <- c("marche_aleatoire", "A10_1", "A3_1", "A10_5", "A3_5")

resultat_mz <- map(nom_previsions, function(p) {
  fonction_mz(
    tableau_comparatif$realisation,
    tableau_comparatif[[p]],
    p
  )
}) |> bind_rows()

resultat_mz[, 1:5]
```

```
# A tibble: 5 x 5
  nom_prevision       alpha     beta alpha_pvalue beta_pvalue
  <chr>               <dbl>    <dbl>        <dbl>       <dbl>
1 marche_aleatoire 0.000243 -7.25e-6        0.590      0
2 A10_1            0.000251  2.64e+0        0.578      0.339
3 A3_1             0.000268  8.23e-1        0.552      0.704
4 A10_5            0.000253 -8.56e+6        0.575      0.0763
5 A3_5             0.000246  9.72e+2        0.586      0.925
```

```r
resultat_mz[, c(1, 6, 7, 8)]
```

```
# A tibble: 5 x 4
  nom_prevision    H0_alpha    H0_beta     H0
  <chr>            <chr>       <chr>       <chr>
1 marche_aleatoire non rejetée rejetée     H0 rejetée
2 A10_1            non rejetée non rejetée H0 non rejetée
3 A3_1             non rejetée non rejetée H0 non rejetée
4 A10_5            non rejetée non rejetée H0 non rejetée
5 A3_5             non rejetée non rejetée H0 non rejetée
```

6.3 Interprétation

Le test de Mincer-Zarnowitz évalue la qualité des prévisions en comparant $\hat{Y}_{t+h|t}$ aux valeurs réelles Y_{t+h} à travers un modèle linéaire. L'hypothèse nulle (H_0) stipule que $\alpha = 0$ et $\beta = 1$, tandis que l'hypothèse alternative (H_1) suggère une déviation de ces valeurs. Si H_0 est rejetée, la prévision est considérée comme n'étant pas de bonne qualité.

Pour le modèle de marche aléatoire, α est proche de zéro, mais β est très faible. La p-value associée à α (0.589) indique que H_0 n'est pas rejetée pour α, mais celle pour β est extrêmement faible (0.000), entraînant le rejet de H_0 pour β. Cela montre une forte déviation de β par rapport à 1, ce qui indique que le modèle de marche aléatoire n'est pas idéal.

Concernant la prévision "A10_1", α est proche de zéro et β a une p-value de 0.338, supérieure à 0.05, ce qui signifie que H_0 n'est pas rejeté pour β. La prévision "A10_1" est donc correcte, avec des coefficients proches des valeurs théoriques.

Pour "A3_1", α et β sont également proches des valeurs attendues. Les p-values indiquent que H_0 n'est pas rejetée pour les deux paramètres, suggérant que cette prévision est correcte.

Dans le cas de "A10_5", bien que β soit très éloigné de 1, H_0 n'est pas rejeté pour β (p-value = 0.076), ce qui signifie que la déviation n'est pas statistiquement significative. Ainsi, la prévision reste acceptable, malgré l'écart de β.

Pour "A3_5", α est proche de zéro et β est élevé. Les p-values montrent que H_0 n'est pas rejetée pour α et β, ce qui suggère que cette prévision est correcte et acceptable.

En conclusion, les prévisions du modèle "marche_aleatoire" sont de qualité inférieure, principalement en raison de la déviation de β par rapport à 1, indiquant une mauvaise capacité prédictive. Les modèles "A10_1", "A3_1" et "A3_5" offrent des résultats plus fiables, avec des coefficients proches des valeurs théoriques attendues. Le modèle "A3_1" est le plus conforme à H_0, tandis que "A10_5" et "A3_5" sont acceptables malgré quelques écarts. Ainsi, les modèles alternatifs surpassent largement la marche aléatoire en termes de précision et de robustesse.

7 Statistique de Diebold et Mariano

7.1 Calcul des critères MSE, MAD et Quad-Quad pour chaque modèle de prévision

7.1.1 Calcul des critères

```
# MSE
fonction_mse <- function(df, y, prevision) {
  mse <- (df[[y]] - df[[prevision]])^2
  return(mse)
}

# MAD
fonction_mad <- function(df, y, prevision) {
  mad <- abs(df[[y]] - df[[prevision]])
  return(mad)
}

# Quad-Quad
fonction_quad_quad <- function(df, y, prevision, alpha1, alpha2) {
  quad_quad <- alpha1 * (df[[y]] - df[[prevision]])^2 +
    alpha2 * ifelse(
      (df[[y]] - df[[prevision]]) < 0,
      (df[[y]] - df[[prevision]])^2,
      0
    )
  return(quad_quad)
}
```

7.1.2 Calcul pour chaque modèle de prévision

```
# Liste des modèles de prévision
nom_previsions <- c("marche_aleatoire", "A10_1", "A3_1", "A10_5", "A3_5")

# Fonction
calculer_critere <- function(
    fonction_critere, previsions, alpha1 = NULL, alpha2 = NULL, nom = NULL) {
  resultat <- tableau_comparatif |>
    select(date) |>
    bind_cols(
      map(previsions, ~ {
        if (!is.null(alpha1) & !is.null(alpha2)) {
          fonction_critere(tableau_comparatif, "realisation", ., alpha1, alpha2)
        } else {
          fonction_critere(tableau_comparatif, "realisation", .)
        }
      }) |>
```

```r
        set_names(previsions)
    )

  attr(resultat, "nom") <- nom # Utilisation du nom comme attribut

  return(resultat)
}


# MSE
mse <- calculer_critere(fonction_mse, nom_previsions, nom = "mse")
mse
```

```
# A tibble: 1,808 x 6
   date       marche_aleatoire     A10_1     A3_1     A10_5      A3_5
   <date>                <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
 1 2016-01-01         NA        NA         NA         NA         NA
 2 2016-01-04          0.312     0.000641  0.000641  NA         NA
 3 2016-01-05          0.0481    0.0000401 0.0000428 NA         NA
 4 2016-01-06          0.0822    0.0000109 0.0000105 NA         NA
 5 2016-01-07          2.23      0.000153  0.000152  NA         NA
 6 2016-01-08          0.000638  0.000450  0.000445   0.000446  0.000446
 7 2016-01-11          1.92      0.000395  0.000403   0.000402  0.000402
 8 2016-01-12          2.19      0.000656  0.000663   0.000665  0.000665
 9 2016-01-13          0.213     0.0000428 0.0000450  0.0000459 0.0000459
10 2016-01-14          0.472     0.000384  0.000383   0.000382  0.000382
# i 1,798 more rows
```

```r
# MAD
mad <- calculer_critere(fonction_mad, nom_previsions, nom = "mad")
mad
```

```
# A tibble: 1,808 x 6
   date       marche_aleatoire   A10_1    A3_1   A10_5    A3_5
   <date>                <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 2016-01-01           NA      NA       NA      NA      NA
 2 2016-01-04            0.559   0.0253  0.0253  NA      NA
 3 2016-01-05            0.219   0.00633 0.00654 NA      NA
 4 2016-01-06            0.287   0.00330 0.00324 NA      NA
 5 2016-01-07            1.49    0.0124  0.0123  NA      NA
 6 2016-01-08            0.0253  0.0212  0.0211   0.0211  0.0211
 7 2016-01-11            1.39    0.0199  0.0201   0.0201  0.0201
 8 2016-01-12            1.48    0.0256  0.0257   0.0258  0.0258
 9 2016-01-13            0.461   0.00654 0.00670  0.00678 0.00678
10 2016-01-14            0.687   0.0196  0.0196   0.0195  0.0195
# i 1,798 more rows
```

```r
# Quad-Quad
quad_quad <- calculer_critere(
  fonction_quad_quad,
```

```r
  nom_previsions,
  alpha1 = 1,
  alpha2 = 1,
  "quad_quad"
)
quad_quad
```

```
# A tibble: 1,808 x 6
   date       marche_aleatoire      A10_1       A3_1      A10_5       A3_5
   <date>                <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
 1 2016-01-01         NA        NA          NA         NA          NA
 2 2016-01-04          0.624     0.00128     0.00128   NA          NA
 3 2016-01-05          0.0962    0.0000401   0.0000428 NA          NA
 4 2016-01-06          0.0822    0.0000109   0.0000105 NA          NA
 5 2016-01-07          2.23      0.000153    0.000152  NA          NA
 6 2016-01-08          0.00128   0.000450    0.000445   0.000446   0.000446
 7 2016-01-11          3.84      0.000790    0.000805   0.000804   0.000804
 8 2016-01-12          4.38      0.000656    0.000663   0.000665   0.000665
 9 2016-01-13          0.425     0.0000856   0.0000899  0.0000918  0.0000918
10 2016-01-14          0.945     0.000768    0.000766   0.000764   0.000764
# i 1,798 more rows
```

7.2 Algorithme du test de Diebold-Mariano

```
fonction_dm <- function(critere, modele_A, modele_B) {
  # Extraction des pertes
  perte_A <- critere[[modele_A]]
  perte_B <- critere[[modele_B]]

  # Calcul de delta (différence des pertes sans valeurs manquantes)
  delta <- na.omit(perte_A - perte_B)

  # Régression pour obtenir la statistique DM
  regression <- lm(delta ~ 1)

  # Estimation de la covariance des erreurs avec la procédure Newey-West
  lag_delta <- sqrt(length(delta))
  covariance_residus <- NeweyWest(regression, lag = lag_delta)

  # Calcul de la statistique de test DM
  sigma_carre_NW <- covariance_residus[1, 1]
  DM_stat <- mean(delta) / sqrt(sigma_carre_NW / length(delta))

  # Décision de rejet en fonction du test DM
  alpha <- 0.05
  valeur_critique <- qnorm(1 - alpha / 2)

  if (abs(DM_stat) > valeur_critique) {
    if (DM_stat < 0) {
      decision <- "H0 rejetée"
      meilleur_modele <- modele_A
    } else {
      decision <- "H0 rejetée"
      meilleur_modele <- modele_B
    }
  } else {
    decision <- "H0 non rejetée"
    meilleur_modele <- NA
  }


  # Résultat
  resultat <- tibble(
    critere = attr(critere, "nom"),
    modele_A = modele_A,
    modele_B = modele_B,
    DM_stat = DM_stat,
    decision = decision,
    meilleur_modele = meilleur_modele
  )
```

```
  return(resultat)
}
```

7.3 Résultat

```r
# Paramètres
criteres <- list(mse, mad, quad_quad)
modeles <- c("marche_aleatoire", "A10_1", "A3_1", "A10_5", "A3_5")

# fonction_dm à chaque combinaison de criteres et modeles
resultat_dm <- map(modeles, function(i) {
  map(modeles[modeles > i], function(j) {
    map(criteres, function(criteres) {
      fonction_dm(criteres, i, j)
    }) |> bind_rows()
  })
}) |> bind_rows()

print(resultat_dm, n = 30)
```

```
# A tibble: 30 x 6
   critere   modele_A modele_B         DM_stat decision   meilleur_modele
   <chr>     <chr>    <chr>              <dbl> <chr>      <chr>
 1 mse       A10_1    marche_aleatoire -1608.  H0 rejetée A10_1
 2 mad       A10_1    marche_aleatoire -2974.  H0 rejetée A10_1
 3 quad_quad A10_1    marche_aleatoire -1352.  H0 rejetée A10_1
 4 mse       A10_1    A3_1                12.0 H0 rejetée A3_1
 5 mad       A10_1    A3_1               -27.9 H0 rejetée A10_1
 6 quad_quad A10_1    A3_1                41.3 H0 rejetée A3_1
 7 mse       A10_1    A10_5              -29.7 H0 rejetée A10_1
 8 mad       A10_1    A10_5               54.5 H0 rejetée A10_5
 9 quad_quad A10_1    A10_5              -26.8 H0 rejetée A10_1
10 mse       A10_1    A3_5               -29.6 H0 rejetée A10_1
11 mad       A10_1    A3_5                54.5 H0 rejetée A3_5
12 quad_quad A10_1    A3_5               -26.8 H0 rejetée A10_1
13 mse       A3_1     marche_aleatoire -1608.  H0 rejetée A3_1
14 mad       A3_1     marche_aleatoire -2974.  H0 rejetée A3_1
15 quad_quad A3_1     marche_aleatoire -1352.  H0 rejetée A3_1
16 mse       A3_1     A3_5               -26.2 H0 rejetée A3_1
17 mad       A3_1     A3_5                42.2 H0 rejetée A3_5
18 quad_quad A3_1     A3_5               -50.4 H0 rejetée A3_1
19 mse       A10_5    marche_aleatoire -1600.  H0 rejetée A10_5
20 mad       A10_5    marche_aleatoire -2968.  H0 rejetée A10_5
21 quad_quad A10_5    marche_aleatoire -1340.  H0 rejetée A10_5
22 mse       A10_5    A3_1                26.2 H0 rejetée A3_1
23 mad       A10_5    A3_1               -42.2 H0 rejetée A10_5
24 quad_quad A10_5    A3_1                50.4 H0 rejetée A3_1
25 mse       A10_5    A3_5                4.32 H0 rejetée A3_5
26 mad       A10_5    A3_5               -26.4 H0 rejetée A10_5
27 quad_quad A10_5    A3_5                16.8 H0 rejetée A3_5
28 mse       A3_5     marche_aleatoire -1600.  H0 rejetée A3_5
```

```
29 mad       A3_5     marche_aleatoire -2968.   H0 rejetée A3_5
30 quad_quad A3_5     marche_aleatoire -1340.   H0 rejetée A3_5
```

```
compte_meilleur_modele <- tibble(
  modele = modeles,
  nb_apparitions = map_int(modeles, ~ sum(resultat_dm$meilleur_modele == .x))
)

compte_meilleur_modele
```

```
# A tibble: 5 x 2
  modele           nb_apparitions
  <chr>                     <int>
1 marche_aleatoire              0
2 A10_1                         8
3 A3_1                          9
4 A10_5                         6
5 A3_5                          7
```

7.4 Interprétation

Les résultats du test de Diebold et Mariano montrent que la marche aléatoire n'est jamais le modèle le plus performant, l'hypothèse nulle étant systématiquement rejetée. Cela suggère que les modèles A10_1, A3_1, A10_5, et A3_5 offrent de meilleures prévisions. Parmi eux, A10_1 et A3_1 se distinguent fréquemment comme les meilleurs, avec A3_1 en tête, sélectionné 9 fois contre 8 pour A10_1. Les autres modèles apparaissent moins souvent comme les meilleurs, indiquant une performance légèrement inférieure. En conclusion, bien que A3_1 soit légèrement plus performant, A10_1 constitue également une alternative solide, et ensemble, ces deux modèles surpassent largement la marche aléatoire, prouvant leur efficacité pour les prévisions.

8 Conclusion

En conclusion, les tests de Mincer-Zarnowitz et de Diebold et Mariano montrent que la marche aléatoire est un modèle peu fiable en raison de la déviation significative de ses paramètres. Les prévisions des modèles A10_1, A3_1, A10_5 et A3_5 sont globalement plus précises, avec A3_1 et A10_1 se distinguant particulièrement. Ces deux modèles surpassent largement la marche aléatoire en termes de qualité et de fiabilité des prévisions, A3_1 étant légèrement plus performant. Ces résultats confirment l'efficacité de ces modèles alternatifs pour les prévisions.