MONIN Louis

IING 3 – FinTech

Modèles de prévision

Projet

LAIB Naamane

2024-2025

Sommaire

Introduction

Le mini-projet a pour but, dans un premier temps, de simuler des modèles ARIMA et de créer une fonction de prévision personnalisée en appliquant les formules mathématiques établies dans le cours de modèles de prévision. Dans un second temps, j'utiliserai les fonctions prédéfinies de R pour mettre en œuvre les modèles ARMA, ARIMA et SAMIRA sur un jeu de données spécifique que je présenterai par la suite. Les étapes comprennent la compréhension du contexte sous-jacent du problème, la collecte et le prétraitement des données, la sélection judicieuse des modèles potentiels pour ajuster ces données, et enfin, la présentation claire des résultats obtenus. L'objectif ultime est d'évaluer la compréhension approfondie des concepts, la précision des modèles, ainsi que la qualité globale de la présentation des résultats.

Partie I : Simulation

Question a)

Théorique

Simulons un processus ARIMA(p,d,q) stationnaire avec d = 1, $p \geq 2$, et $q \geq 1$, en développant une fonction personnalisée nommée Arima_sim.

On sait qu'un processus ARIMA(p,d,q) combine :

- Un processus autoregressif (AR) d'ordre p, qui modélise la dépendance entre une observation et p valeurs passées.
- Une différenciation d'ordre d, qui rend les données stationnaires.
- Un processus moyenne mobile (MA) d'ordre q, qui capture les dépendances sur les résidus des q erreurs précédentes.

Dans ce cas, le fait que d=1 implique que nous devons calculer la différence de première ordre ($X_t - X_{t-1}$) avant d'appliquer les composants AR et MA.

Le code a pour objectif de simuler une série temporelle suivant un processus ARIMA(p,d,q). Mathématiquement, un processus ARIMA(p,d,q) est défini par l'équation suivante après d différenciations successives :

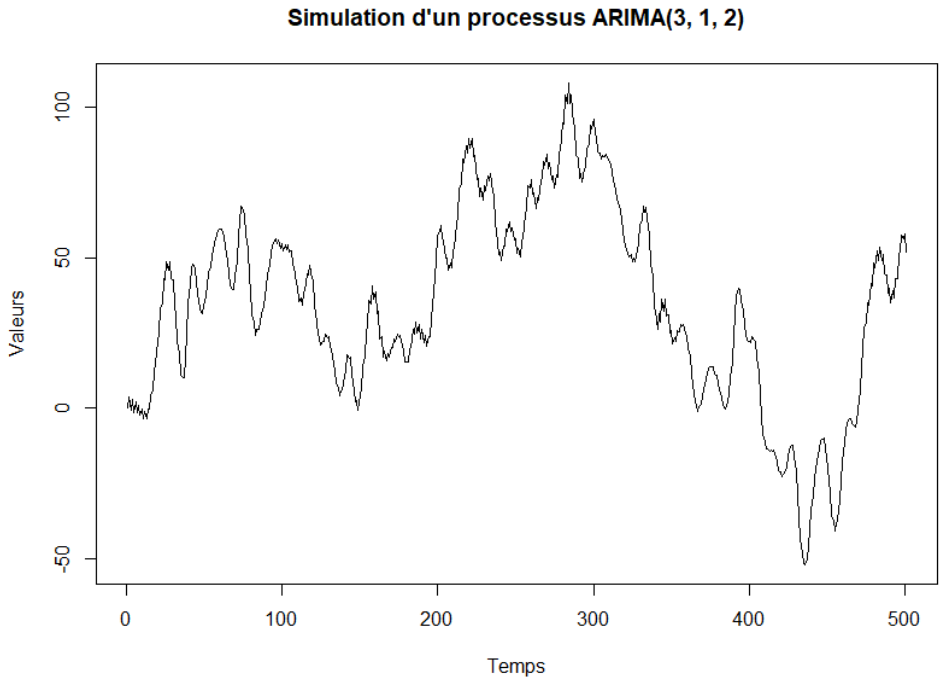
$$\phi(B)(1-B)^d X_t = \theta(B)\epsilon_t$$

où $\phi(B)=1-\phi_1 B-\phi_2 B^2-\ldots-\phi_p B^P \phi(B)=1-\phi_1 B-\phi_2{}^2-\ldots-\phi_P B^P$ représente la composante autoregressive, $\theta(B)=1+\theta_1 B+\theta_2 B^2+\ldots+\theta_q B^q \theta(B)=1+\theta_1 B+\theta_2 B^2+\ldots+\theta_q B^q$ représente la composante moyenne mobile, BB est le lag operator, X_t est la série temporelle simulée, et ϵ_t est un bruit blanc gaussien ($\epsilon_t \sim N(0,\sigma 2)$).

Le code commence par vérifier les paramètres (p,q,d) pour s'assurer qu'ils respectent les contraintes imposées ($p \geq 2, q \geq 1, d=1$). Si les coefficients AR (ϕ_i) et MA (θ_j) ne sont pas fournis, ils sont générés aléatoirement dans l'intervalle $[-0.9, 0.9]$. Cependant, pour garantir la stationnarité et l'inversibilité du modèle, le code vérifie les racines des polynômes caractéristiques associés à $\phi(B)$ et $\theta(B)$. Les conditions de stationnarité exigent que toutes les racines de $\phi(B)$ soient en dehors du cercle unité ($|z|>1$), tandis que les conditions d'inversibilité imposent la même contrainte pour $\theta(B)$. En cas de violation, de nouveaux coefficients sont générés jusqu'à satisfaction des conditions.

Simulation

```r
# 2 Partie I : Simulation #

#Question a
#Simuler un processus ARIMA(p,d,q) stationnaire avec d = 1, p ≥ 2, et q ≥ 1, en développant
#une fonction personnalisée nommée Arima_sim.

Arima_sim <- function(n, p = 2, q = 1, d = 1, ar_coeffs = NULL, ma_coeffs = NULL, seed = NULL) {
  # vérification des paramètres
  if (p < 2 || q < 1 || d != 1) {
    stop("Cette fonction simule uniquement un ARIMA avec p >= 2, q >= 1 et d = 1.")
  }

  # Fixer la graine pour la reproductibilité si spécifiée
  if (!is.null(seed)) set.seed(seed)

  # Générer les coefficients AR stationnaires si non spécifiés
  if (is.null(ar_coeffs)) {
    repeat {
      ar_coeffs <- runif(p, -0.9, 0.9)
      if (all(abs(polyroot(c(1, -ar_coeffs))) > 1)) break
    }
  }

  # Générer les coefficients MA inversibles si non spécifiés
  if (is.null(ma_coeffs)) {
    repeat {
      ma_coeffs <- runif(q, -0.9, 0.9)
      if (all(abs(polyroot(c(1, ma_coeffs))) > 1)) break
    }
  }

  # Simuler la série ARIMA
  series <- arima.sim(
    n = n,
    model = list(order = c(p, d, q), ar = ar_coeffs, ma = ma_coeffs)
  )

  # Retourner la série simulée
  return(list(
    series = series,
    ar_coeffs = ar_coeffs,
    ma_coeffs = ma_coeffs
  ))
}

# Exemple d'utilisation
result <- Arima_sim(n = 500, p = 3, q = 2, seed = 123)

# Visualisation de la série simulée
plot(result$series, type = "l", main = "Simulation d'un processus ARIMA(3, 1, 2)",
     xlab = "Temps", ylab = "Valeurs")
```



Analyse

Le graphique présente une série qui montre des fluctuations importantes avec des tendances locales, ce qui semble typique pour un processus différencié (d = 1). Elle oscille autour de zéro sans un retour immédiat à une moyenne fixe, suggérant la non-stationnarité introduite par la différenciation. De plus, les amplitudes des variations sont modérées, ce qui reflète l'effet combiné des termes autorégressifs (AR) et des moyennes mobiles (MA).

De la manière dont j'ai construit le code j'ai généré mes coefficients AR et MA dans des plages qui garantissent respectivement la stationnarité et l'inversibilité. Ces contraintes mathématiques assurent que le modèle généré produit une série cohérente, bien que les détails spécifiques des coefficients influencent la dynamique observée.

Sur le long terme, on peut dire que la différenciation (d = 1) donne à la série un comportement aléatoire sous forme de « random walk avec des dépendances AR et MA ». Les fluctuations observées sont amplifiées ou atténuées par les termes AR (p = 3) et lissent les chocs grâce aux termes MA (q = 2).

Question b)

Théorique

On peut partir de l'équation suivante :

$$y_t = c + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \theta 1 \varepsilon_{t-1} + \cdots + \theta q \varepsilon_{t-q} + \varepsilon_t$$

Elle peut être écrite comme un modèle ARMA de la forme suivante :

$$y_t = c + \varphi_1 B y_t + \cdots + \varphi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

ou

$$(I - \varphi_1 B - \cdots - \varphi_p B^p) y_t = c + (I + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

En notant que B est l'opérateur retard

Un ARIMA(1, 1, 1) (après une première différenciation) est un ARMA(1,1) qui s'écrit :

$$(1 - \varphi_1 B)\,(1 - B) y_t = c + (1 + \theta_1 B) \varepsilon_t$$

↑ ↑ ↑

AR(1) First difference MA(1)

qui s'écrit : $y_t = c + y_{t-1} + \varphi 1 y_{t-1} - \varphi 1 y_{t-2} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$

Un ARMA(p, q) après une différenciation d'ordre d est un ARIMA(p, d, q) qui s'écrit :

$$(1 - \varphi_1 B - \cdots - \varphi_p B^p)\,(1 - B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

↑ ↑ ↑

AR(p) d differences MA(q)

Une fois la série différenciée pour obtenir un processus stationnaire, la prévision d'ordre h se fait en itérant les prévisions à partir des valeurs passées et des innovations.

Pour h = 1, la prévision est :

$$\hat{X}_{t+1} = \sum_{i=1}^{p} \phi_i X_{t+1-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t+1-j}$$

Pour h > 1, les termes X_{t+1}, X_{t+2}, … sont remplacés par leurs prévisions et les termes ϵ_{t+1}, ϵ_{t+2}, … sont supposés nuls (car non observables).

Simulation

```r
Forecast_Per <- function(series, p, q, d, ar_coeffs, ma_coeffs, h) {
  # Vérifier les paramètres
  if (d != 1) stop("Cette fonction supporte uniquement d = 1.")

  # Différencier la série
  diff_series <- diff(series, differences = d)
  n <- length(diff_series)

  # Initialiser les prévisions
  forecasts <- numeric(h)

  # Prévoir les valeurs futures
  for (t in 1:h) {
    # Calcul des termes AR
    ar_part <- 0
    if (p > 0) {
      for (i in 1:min(p, n + t - 1)) {
        ar_part <- ar_part + ar_coeffs[i] * (if (t - i <= 0) series[n + t - i] else forecasts[t - i])
      }
    }

    # Calcul des termes MA
    ma_part <- 0
    if (q > 0) {
      for (j in 1:min(q, n + t - 1)) {
        ma_part <- ma_part + ma_coeffs[j] * (if (t - j <= 0) 0 else 0) # Innovations non observables
      }
    }

    # Ajouter les parties AR et MA
    forecasts[t] <- ar_part + ma_part
  }
```

```r
  # Reconstruire la série d'origine (somme des différences)
  final_forecasts <- cumsum(c(series[n], forecasts))
  return(final_forecasts[-1])
}

# Tester la fonction
sim_data <- result$series
ar_coeffs <- result$ar_coeffs
ma_coeffs <- result$ma_coeffs

# Prévision pour h = 2
forecasted_values <- Forecast_Per(
  series = sim_data,
  p = 3,
  q = 2,
  d = 1,
  ar_coeffs = ar_coeffs,
  ma_coeffs = ma_coeffs,
  h = 2
)

print(forecasted_values)
```

Analyse

```
> print(forecasted_values)
[1]  95.25752 120.95697
```

Les prévisions obtenues avec la fonction Forecast_Per sont les suivantes :

$$\hat{X}_{t+1}=95.25752,\ \hat{X}_{t+2}=120.95697$$

Pour la première prévision $\hat{X}_{t+1}$, cette valeur est calculée en utilisant les observations passées (X_t, X_{t-1}, …) et les coefficients du modèle ARIMA simulé (ϕ_1, ϕ_2, …, θ_1, θ_2, …). Puisque h = 1, la prévision est directement influencée par les observations réelles les plus récentes et les coefficients du modèle.

Pour la deuxième prévision $\hat{X}_{t+2}$, cette valeur est une extension de la prévision. Elle utilise la première prévision $\hat{X}_{t+1}$ comme entrée car les valeurs futures au-delà de t+1 ne sont pas observées. Les termes d'innovation (ϵ_t) pour t > n sont supposés être nuls dans la prévision.

Les prévisions montrent une augmentation, ce qui peut indiquer deux éléments :

- Une tendance sous-jacente dans les données simulées
- L'influence des coefficients AR (ϕ) et MA (θ) qui amplifient l'effet des observations passées.

Pour $\hat{X}_{t+1}$

$$\hat{X}_{t+1} = \sum_{i=1}^{p} \phi_i X_{t+1-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t+1-j}$$

Cette valeur est calculée directement à partir des données réelles jusqu'à t.

Pour $\hat{X}_{t+2}$

$$\hat{X}_{t+2} = \sum_{i=1}^{p} \phi_i \hat{X}_{t+2-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t+2-j}$$

La prévision utilise $\hat{X}_{t+1}$ (précédemment calculée) comme une observation passée et suppose que toutes les innovations futures (ϵ) sont nulles.

Les valeurs obtenues (95.25752 et 120.95697) sont cohérentes avec la logique de la fonction Forecast_Per et reflètent l'utilisation des coefficients ARIMA, les données historiques, et l'absence d'innovations futures (ϵ=0).

La fonction Forecast_Per produit des prévisions à court terme (ordre h=2) qui suivent les règles mathématiques de l'ARIMA

Partie II : Pratique sur un Jeu de Données Réel

2.1 Le Jeu de Données

A l'origine le fichier représente la température quotidienne régionale (depuis le 1er janvier 2016 au 31 janvier 2024). Ce jeu de données présente les températures minimales, maximales et moyennes quotidiennes (en degré celsius), par région administrative française, du 1er janvier 2016 à aujourd'hui.

Il est basé sur les mesures officielles du réseau de stations météorologiques françaises. La mise à jour de ce jeu de données est mensuelle.

Lien du site internet : https://www.data.gouv.fr/fr/datasets/temperature-quotidienne-regionale-depuisjanvier-2016/

URL du dataset : Explorateur de données data.gouv.fr

Description du fichier :

-ID : id[text]
-Date : date[date] Date de l'observation
-Code INSEE région : code_insee_region[int] Code INSEE région administrative
-Région : region[text] Région administrative
-TMin (°C) : tmin[double] Température minimale quotidienne
-TMax (°C) : tmax[double] Température maximale quotidienne
-TMoy (°C) : tmoy[double] Température moyenne quotidienne

A l'aide d'un programme python j'ai supprimé toutes les données des régions qui n'était pas celles de l'ile-de-France. Puis j'ai supprimé les colonnes contenant l'ID, le code_insee_region, la région, la température minimale et maximale. J'ai conservé uniquement la date et la température moyenne.

Enfin j'ai fait une moyenne par mois des températures. Le code se présente sous la forme d'un fichier .py nommé « Code_Transformer_Donnees » donné en annexe. [1]

Finalement, mon dataset correspond à une moyenne par mois des relevés de la température de l'île-de France du 1er janvier 2016 au 31 janvier 2024.

```
donnees <- read.xlsx("H:/Desktop/Methode_de_prevision_de_series_temporelles/Projet/Temperature_moyenne_mensuelle_ile_de_france_v2.xlsx", sheet = 1)
View(donnees)
```

	Mois-Annee	tmoy
1	Janvier-2016	5.534839
2	Février-2016	6.138276
3	Mars-2016	6.791290
4	Avril-2016	9.883000
5	Mai-2016	14.412258
6	Juin-2016	17.576667
7	Juillet-2016	20.325484
8	Août-2016	20.919677
9	Septembre-2016	18.782000
10	Octobre-2016	11.459355
11	Novembre-2016	7.657333
12	Décembre-2016	4.457742
13	Janvier-2017	1.950000
14	Février-2017	7.284286
15	Mars-2017	10.492903
16	Avril-2017	10.490667
17	Mai-2017	16.156774
18	Juin-2017	20.366333
19	Juillet-2017	20.752903

Il est composé de 2 colonnes les mois entre l'année 2016 et 2024, ainsi que la température moyenne (tmoy). Avec la librairie knitr, on peut présenter le jeu de données sous une autre forme.

```
#Générer le tableau des valeurs avec knitr

library(knitr)
knitr::kable(donnees, booktabs = TRUE, col.names = c("Mois-Annee", "tmoy"))
```

Mois-Annee	tmoy
Janvier-2016	5.534839
Février-2016	6.138276
Mars-2016	6.791290
Avril-2016	9.883000
Mai-2016	14.412258
Juin-2016	17.576667
Juillet-2016	20.325484
Août-2016	20.919677
Septembre-2016	18.782000
Octobre-2016	11.459355
Novembre-2016	7.657333
Décembre-2016	4.457742
Janvier-2017	1.950000
Février-2017	7.284286
Mars-2017	10.492903
Avril-2017	10.490667
Mai-2017	16.156774
Juin-2017	20.366333

Pour continuer l'étude de mon jeu de données, j'utiliserai la commande « ts » qui permet de transformer les données des températures moyennes en une série temporelle en supposant une fréquence annuelle de 12 mois et en indiquant que la série commence en 2016 et se termine en 2024.

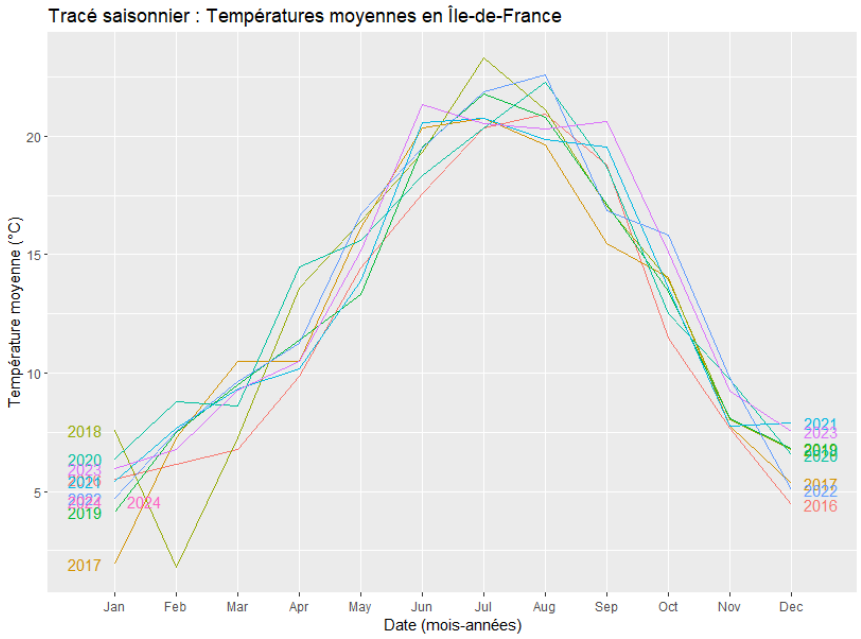
2.2 Études descriptives succintes

2.2.1 Visualisation du jeu de données

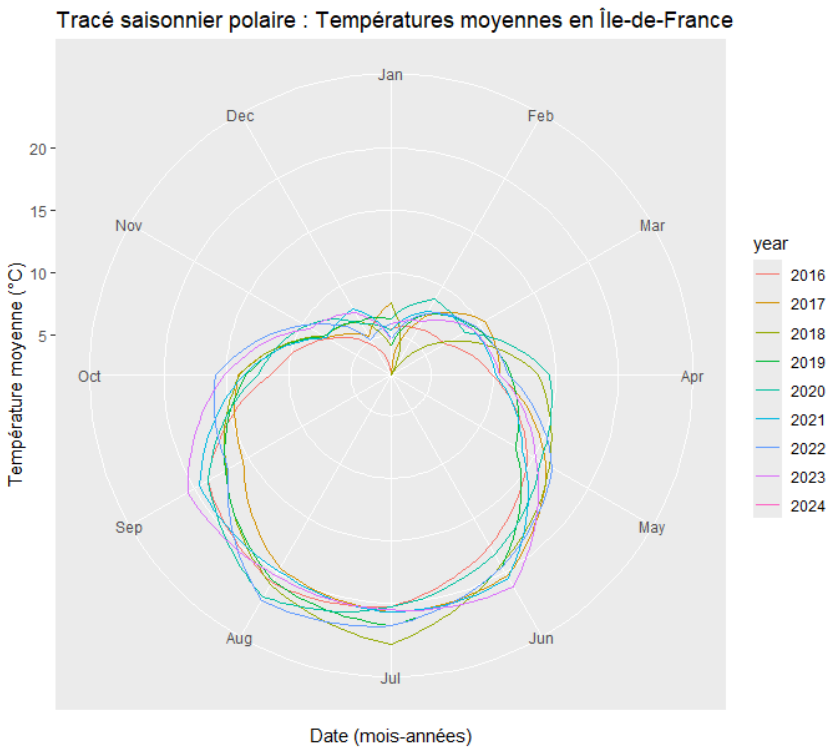
```
# Créer une série temporelle

donnees_ts <- ts(donnees$tmoy, frequency=12, start=c(2016,1), end =c(2024,1))
donnees_ts
```

Pour réaliser une meilleure analyse sur le contenu de chaque périodicité, j'ai effectué un tracé saisonnier des températures moyennes en Ile-de-France. Je constate une forte hausse des températures à la période de Juillet-Août pour chaque année, ce qui est en adéquation avec le fait que cette saison correspond à l'été et que les températures sont en augmentation.

```r
library(forecast)
library(ggplot2)

ggseasonplot(donnees_ts, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("Température moyenne (°C)") +
  xlab("Date (mois-années)") +
  ggtitle("Tracé saisonnier : Températures moyennes en Île-de-France")
```

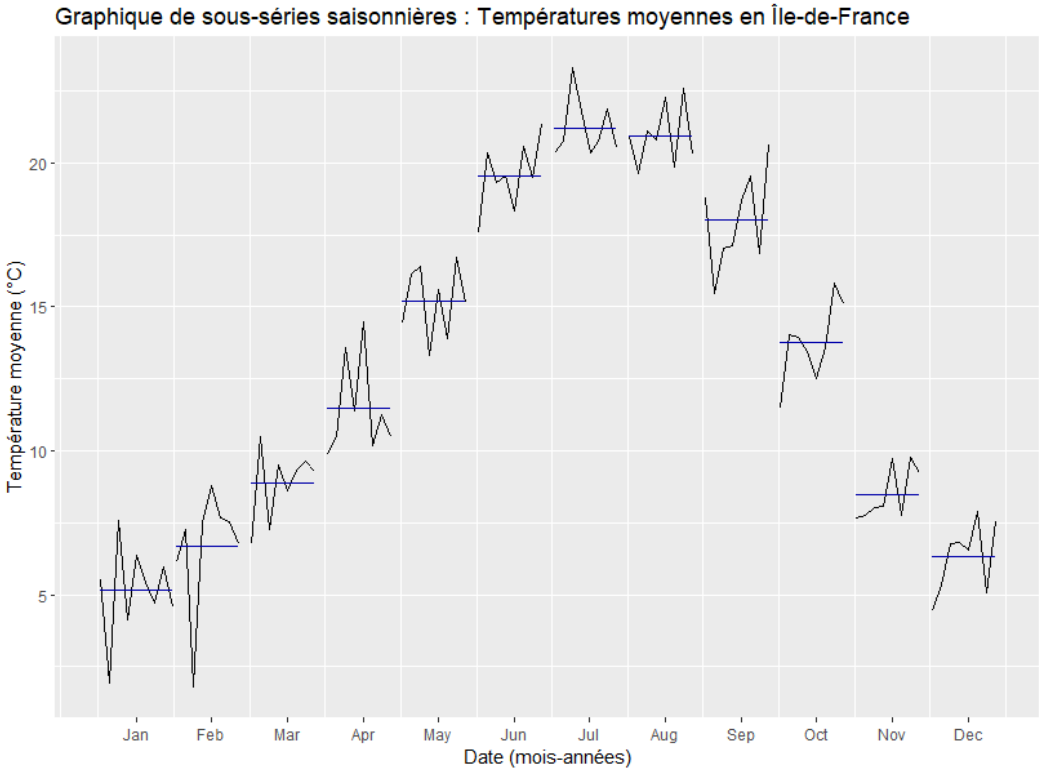



```r
# Utiliser ggseasonplot en mode polaire pour visualiser les données
ggseasonplot(donnees_ts, polar=TRUE) +
  ylab("Température moyenne (°C)") +
  xlab("Date (mois-années)") +
  ggtitle("Tracé saisonnier polaire : Températures moyennes en Île-de-France")
```



Pour poursuivre mon étude sur notre jeu de données, j'ai réalisé un graphique saisonnier alternatif. Les lignes horizontales indiquent les moyennes pour chaque mois. Cette forme de graphique permet de clairement identifier le modèle saisonnier sous-jacent et montre également les changements de saisonnalité au fil du temps. De ce graphique, j'ai pu observer que les températures augmentent et diminuent dans un motif répétitif qui correspond aux saisons. Les températures culminent en été (Juillet-Août) et sont au plus bas en hiver (Janvier-Février).

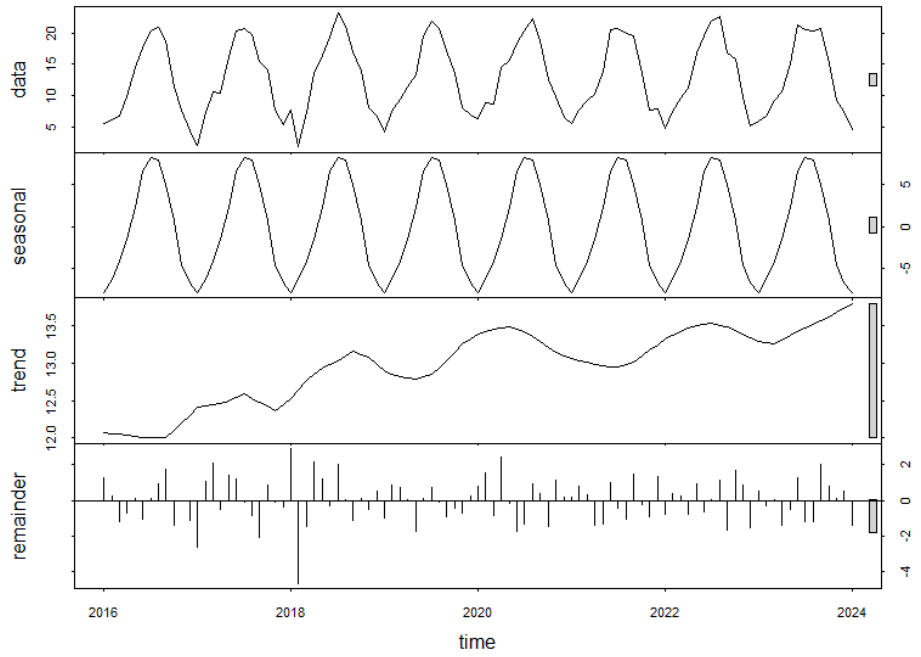
```
ggsubseriesplot(donnees_ts) +
  ylab("Température moyenne (°C)") +
  xlab("Date (mois-années)") +
  ggtitle("Graphique de sous-séries saisonnières : Températures moyennes en Île-de-France")
```



2.2.2 Analyse de la tendance et de la saisonnalité

Pour réaliser une étude de la tendance et de la saisonnalité, j'ai réalisé une étude sur R avec la méthode stl. Cette analyse m'a permis de justifier l'utilisation de ce jeu de données pour la mise en œuvre des modèles ARMA, ARIMA et SARIMA. Car j'ai du utiliser un jeu de données avec une tendance et une saisonnalité pour espérer obtenir des résultats significatifs.

```
# Analyse de la tendance et de la saisonnalité
# Décomposition STL

decomp <- stl(donnees_ts, s.window="periodic")

# Visualiser la décomposition

plot(decomp)
```



Le premier graphique montre la décomposition STL de ma série temporelle en trois composantes qui sont respectivement la saisonnalité, la tendance et le résidu. Au niveau des données saisonnières illustrées par le deuxième graphique du haut, il semble y avoir un motif saisonnier clair et cohérent qui se répète chaque année, ce qui était attendu pour des données basées sur les températures.

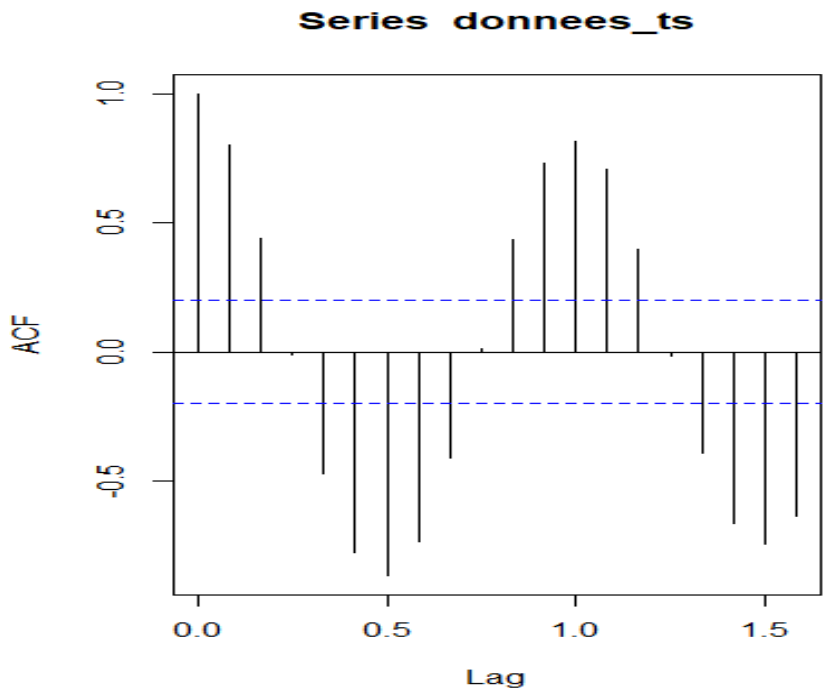
La tendance représentée sur le troisième graphique en partant du haut est relativement lisse sans fluctuations extrêmes.

Enfin, le résidu présent sur le dernier graphique semble être relativement faible et aléatoire, ce qui indique que la décomposition a capturé la plupart des comportement systématiques de la série temporelle et ce qui reste est principalement du bruit.

2.2.3 Analyse de la corrélation

Pour la suite de l'étude de mon jeu de données, je me suis penché sur la corrélation de ce dernier. La corrélation pour une série temporelle fait référence à la manière dont les valeurs d'une série sont liées à d'autres valeurs de cette même série (ou d'une autre série) dans le temps.

```
#Analyse de la corrélation

#Autocorrélation

acf(donnees_ts)
```



Dans mon cas la série $(X_t)_{1\leq t\leq n}$ est une série périodique pure $X_t = a\cos(2\pi t/p)$, où p est la période, car pour h fixé on obtient la propriété suivante :

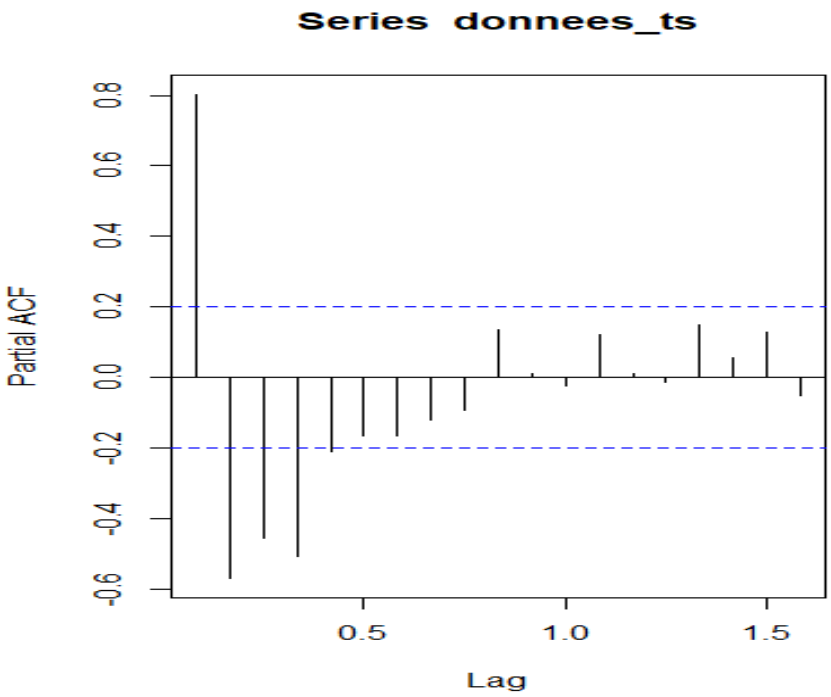
$$\rho_n(h) \rightarrow \cos(2h\pi/p) \text{ quand } n \rightarrow +\infty$$

Finalement, je retrouve la présence d'une tendance et d'une saisonnalité en examinant la corrélation de la série et pour pousser d'avantage l'analyse de cette dernière on va réaliser l'analyse de l'auto-corrélation partielle

2.2.4 Analyse de l'auto-corrélation partielle

L'auto-corrélation partielle empirique d'ordre h est une fonction de h, noté r(h), qui quantifie la corrélation entre deux données espacées de h par pas de temps, en enlevant l'effet des données intermédiaries.

```
# Autocorrélation partielle

pacf(donnees_ts)
```



Pour la fonction d'auto-corrélation de la série temporelle, on observe les barres sont toutes très courtes et situées à l'intérieur des limites de confiance, ce qui suggère qu'il n'y a pas d'autocorrélations partielles significatives aux retards supérieurs à 1. Ce résultat indique qu'un modèle autorégressif simple (AR) ne serait probablement pas approprié pour cette série temporelle ou que le modèle AR nécessaire serait d'un ordre très bas.

2.2.5 Analyse de la stationnarité

Le test de Dickey-Fuller augmenté (ADF) est utilisé pour déterminer si une série temporelle est stationnaire ou non, c'est-à-dire si la série présente une tendance au fil du temps.

```
> adf.test(donnees_ts)

        Augmented Dickey-Fuller Test

data:  donnees_ts
Dickey-Fuller = -10.751, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Valeur de Dickey-Fuller : -10.751. Cette statistique est négative, ce qui est un bon indicateur. Plus cette valeur est négative, plus la preuve contre l'hypothèse nulle (la présence d'une racine unitaire, indiquant une non-stationnarité) est forte.

Ordre de retard (Lag order) : 4. Cela indique que le test a utilisé 4 retards dans la construction de l'équation de test ADF pour la série temporelle, ce qui correspond au nombre de périodes prises en compte pour calculer la corrélation dans la série.

Valeur-p: 0.01. En statistiques, la valeur-p est utilisée pour déterminer la signification statistique du résultat du test. Une valeur-p inférieure à un seuil (généralement 0.05) indique que vous pouvez rejeter l'hypothèse nulle. Dans ce cas, une valeur-p de 0.01 suggère que vous pouvez rejeter l'hypothèse de non-stationnarité avec une confiance de 99 %.

Hypothèse alternative : stationnaire. Le test ADF a une hypothèse alternative selon laquelle la série est stationnaire. Le message d'avertissement indique que la valeur-p réelle est encore plus petite que la valeur imprimée, renforçant l'évidence contre l'hypothèse nulle.

En conclusion, le résultat du test ADF suggère fortement que la série temporelle est stationnaire. Cela signifie que la série temporelle n'a pas de tendance ou de modèle autorégressif intégré, ce qui est une propriété souhaitable lors de l'utilisation de modèles de prévision tels que ARIMA. Cela signifie également que la série temporelle est appropriée pour une analyse plus approfondie et le développement de modèles prédictifs sans nécessiter de différenciation pour rendre la série stationnaire.

2.3 Utilisation des Méthodes Stochastiques ARIMA et SAMIRA

2.3.1 Stationnarisation éventuelle de la série pour la ramener à un processus stationnaire

```
adf_test <- adf.test(donnees_ts)
cat("P-value du test ADF:", adf_test$p.value, "\n")
```

La plupart des modèles ARIMA nécessitent une série stationnaire (statistiques constantes au cours du temps). Pour vérifier cela, on applique le test ADF (Augmented Dickey-Fuller). Si la p-value est < 0.05 : La série est stationnaire, et aucune différenciation n'est nécessaire. Si la p-value est ≥ 0.05 : La série n'est pas stationnaire. Une différenciation est nécessaire pour supprimer la tendance. Dans notre cas, on obtient une p-value = $0.01 < 0.05$ donc ma série est bien stationnaire.

```r
if (adf_test$p.value > 0.05) {
  cat("La série n'est pas stationnaire, différenciation en cours...\n")
  donnees_diff <- diff(donnees_ts)
  adf_test_diff <- adf.test(donnees_diff)
  cat("P-value après différenciation:", adf_test_diff$p.value, "\n")
} else {
  donnees_diff <- donnees_ts
}
```

Cette partie du code aurait réalisé une différenciation sur la série temporelle si cette dernière possédait une p-value > 0.05.

2.3.2 Identification a priori de modèles potentiels

L'objectif est de déterminer les paramètres (p, d, q) pour le modèle ARIMA à l'aide des graphiques ACF (fonction d'autocorrélation) et PACF (fonction d'autocorrélation partielle)

L'ACF identifie le degré de dépendance entre observation. Si des pics significatifs décroissent lentement, cela peut indiquer un composant MA (moving average).

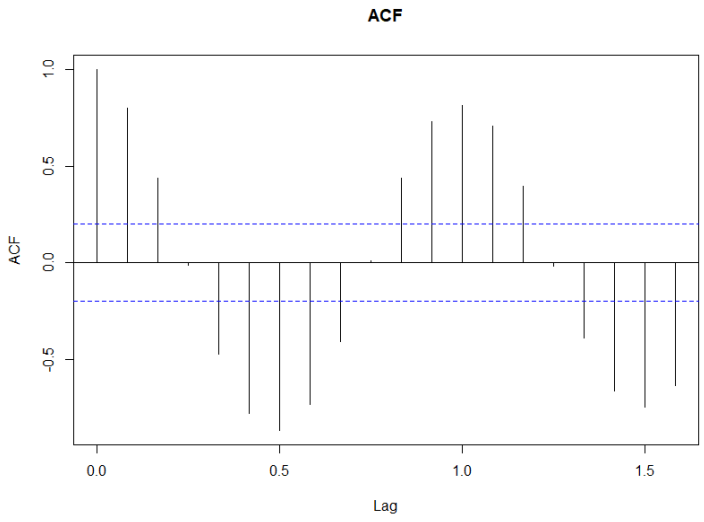
Le PACF met en évidence les lags AR (auto-régressifs). Des pics significatifs indiquent la valeur potentielle de p

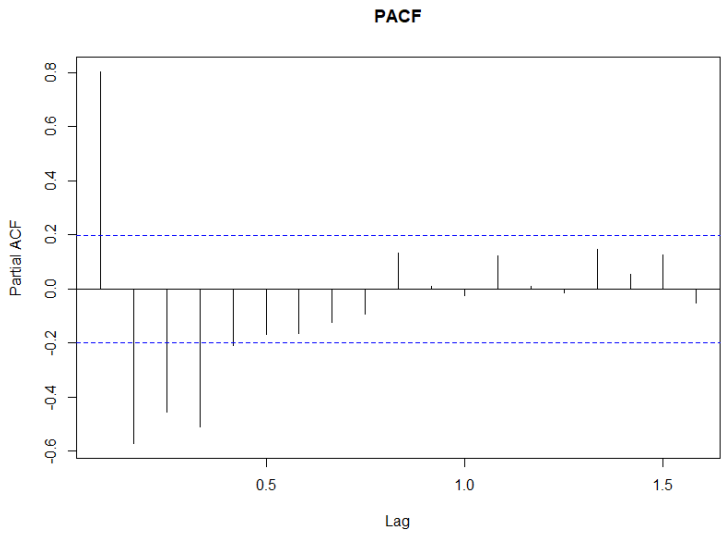
ARIMA(p, d, q) :

p : L'ordre AR est déterminé par le nombre de lags significatifs dans le PACF.

q : L'ordre MA est déterminé par le nombre de lags significatifs dans l'ACF.

d : Déjà défini par le nombre de différenciations appliquées.





Je réalise une étude détaillée des résultats pour choisir un modèle potentiel pour ma série temporelle.

ACF (Autocorrelation Function)

Le graphique de l'ACF montre les corrélations entre la série temporelle et ses lags successifs.

Interprétation :

Les premiers lags présentent des corrélations significatives (valeurs au-delà des bandes bleues).
Les corrélations semblent décroître lentement, ce qui pourrait indiquer la présence d'un composant MA (Moving Average). Le nombre de lags significatifs avant que la corrélation ne devienne insignifiante suggère la valeur potentielle de q.

PACF (Partial Autocorrelation Function)

Le graphique de la PACF examine les corrélations partielles, en éliminant l'influence des lags intermédiaires.

Interprétation :

Le premier lag est significatif dans la PACF, mais les lags suivants deviennent rapidement insignifiants.
Cela suggère un processus AR (Auto-Regressive) d'ordre 1, donc un potentiel p = 1.

En combinant les analyses :
p=1 (d'après la PACF).
q≈1 ou 2 (d'après l'ACF).
d=0 car pas de différenciation

2.3.3 Estimation des paramètres de ces modèles

```
> model_arima <- auto.arima(donnees_ts, seasonal = FALSE)
> model_sarima <- auto.arima(donnees_ts, seasonal = TRUE)
> model_arima
Series: donnees_ts
ARIMA(4,0,0) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ar4     mean
      0.7577  -0.0250  0.1735  -0.6113  12.9741
s.e.  0.0792   0.1103  0.1093   0.0789   0.2697

sigma^2 = 3.556:  log likelihood = -198.96
AIC=409.91   AICc=410.85   BIC=425.36
> model_sarima
Series: donnees_ts
ARIMA(0,0,0)(2,1,0)[12] with drift

Coefficients:
         sar1     sar2   drift
      -0.8264  -0.2478  0.0158
s.e.   0.1101   0.1257  0.0073

sigma^2 = 2.349:  log likelihood = -159.61
AIC=327.21   AICc=327.71   BIC=336.98
```

Modèle ARIMA(4,0,0)
Structure :

ARIMA(4,0,0) signifie :
p=4 : Il y a 4 termes auto-régressifs dans le modèle.
d=0: Les données n'ont pas été différenciées (aucune transformation pour stationnarisation).
q=0 : Aucun terme de moyenne mobile (MA) n'est inclus.

Paramètres estimés :

Les coefficients AR (ϕ_1, ϕ_2, ϕ_3, ϕ_4) :

ϕ_1=0.7577, ϕ_2=−0.0250, ϕ_3=0.1735, ϕ4=−0.6113.

Ces coefficients montrent des dépendances auto-régressives. Le signe négatif de ϕ_4 peut indiquer un effet oscillatoire à ce lag.
La moyenne (μ=12.9741) : C'est la moyenne de la série temporelle estimée.

Qualité du modèle :

Sigma² = 3.556 : La variance des résidus. Plus cette valeur est faible, meilleur est l'ajustement.
AIC = 409.91 : Critère d'information d'Akaike. Un AIC plus faible indique un modèle plus adapté.
BIC = 425.36 : Critère d'information bayésien. Plus faible que l'AIC, mais pénalise davantage les modèles complexes.

Modèle SARIMA(0,0,0)(2,1,0)[12]

Structure :
SARIMA(0,0,0)(2,1,0)[12] signifie :

(p,d,q)=(0,0,0) : Aucun terme AR ou MA n'est inclus dans la partie non saisonnière.

(P,D,Q)[m]=(2,1,0)[12]:
P=2 : Deux termes saisonniers auto-régressifs (SAR) sont inclus.
D=1 : La série a été différenciée pour traiter la saisonnalité.
Q=0 : Aucun terme saisonnier de moyenne mobile.
m=12 : La saisonnalité est annuelle (données mensuelles).

Paramètres estimés :

SAR1=−0.8264, SAR2=−0.2478 :

Ces termes capturent la dépendance entre les observations espacées de 12 mois.
Les valeurs négatives indiquent un effet inverse à ces lags saisonniers.
Drift (μ) : 0.0158, ce qui représente une légère dérive annuelle dans les données.

Qualité du modèle :

Sigma² = 2.349 : La variance des résidus est significativement plus faible que celle du modèle ARIMA.
AIC = 327.21, BIC = 336.98 : Ces valeurs sont beaucoup plus basses que celles du modèle ARIMA, indiquant une meilleure adéquation.

Comparaison des modèles : ARIMA(4,0,0) vs SARIMA(0,0,0)(2,1,0)[12]

Critères d'évaluation :

Variance des résidus (σ^2) :
SARIMA : σ^2=2.349
ARIMA : σ^2=3.556
SARIMA capture mieux les variations des données (résidus plus faibles).

AIC et BIC :
SARIMA : AIC = 327.21, BIC = 336.98
ARIMA : AIC = 409.91, BIC = 425.36
SARIMA est significativement meilleur, car ses critères d'information sont plus faibles.

Structure du modèle :
Le modèle SARIMA prend en compte une composante saisonnière (avec différenciation D=1) et des termes auto-régressifs saisonniers (P=2).
Le modèle ARIMA est purement non saisonnier, ce qui semble moins adapté aux données mensuelles.

Interprétation finale : SARIMA(0,0,0)(2,1,0)[12] est le modèle préféré

Ce modèle capture efficacement la composante saisonnière des données (m=12).
Ses résidus sont plus faibles et les critères AIC/BIC sont bien meilleurs.
Les paramètres saisonniers (SAR1, SAR2) montrent que la dépendance saisonnière est significative.

2.3.4 Vérification des modèles

```
> checkresiduals(model_arima)

        Ljung-Box test

data:  Residuals from ARIMA(4,0,0) with non-zero mean
Q* = 36.857, df = 15, p-value = 0.001328

Model df: 4.   Total lags used: 19

> checkresiduals(model_sarima)

        Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(2,1,0)[12] with drift
Q* = 21.777, df = 17, p-value = 0.1935

Model df: 2.   Total lags used: 19
```

ARIMA(4,0,0) :

Statistique $Q*$: $Q*=36.857$.
Degrés de liberté (df) : df=15 (calculé comme nombre de lags−paramètres estimés).
P-value : p=0.001328.

Interprétation :

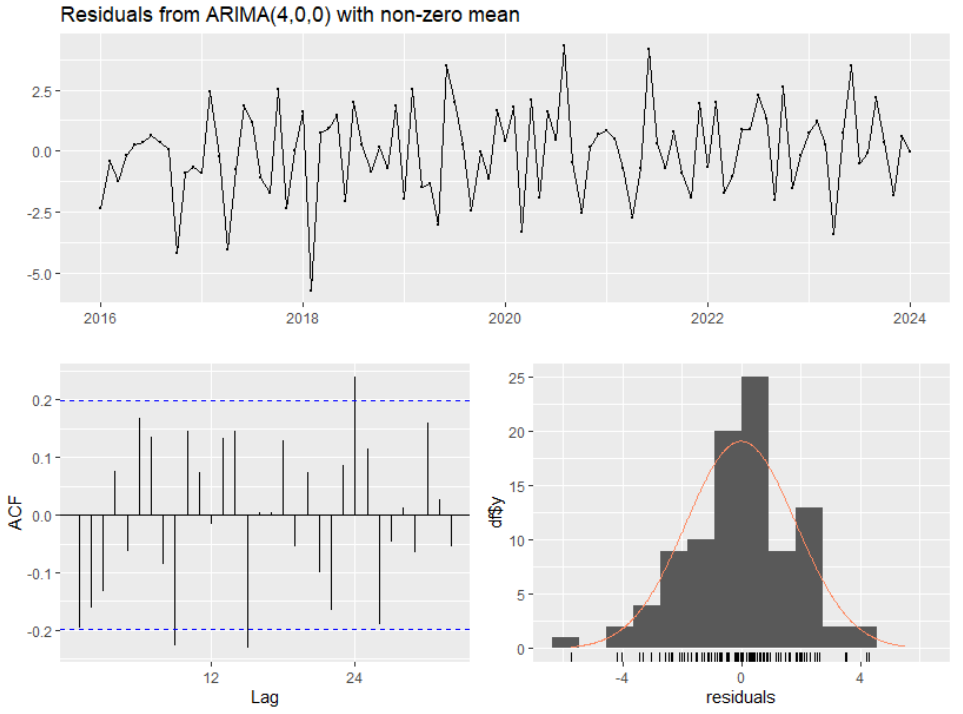
La pp-value est inférieure à 0.050.05, ce qui signifie que je rejette l'hypothèse nulle. Cela indique que les résidus du modèle ARIMA(4,0,0) ne sont pas indépendants.
Des autocorrélations significatives subsistent dans les résidus. Cela suggère que le modèle ARIMA(4,0,0) ne capture pas bien toutes les dépendances présentes dans les données.

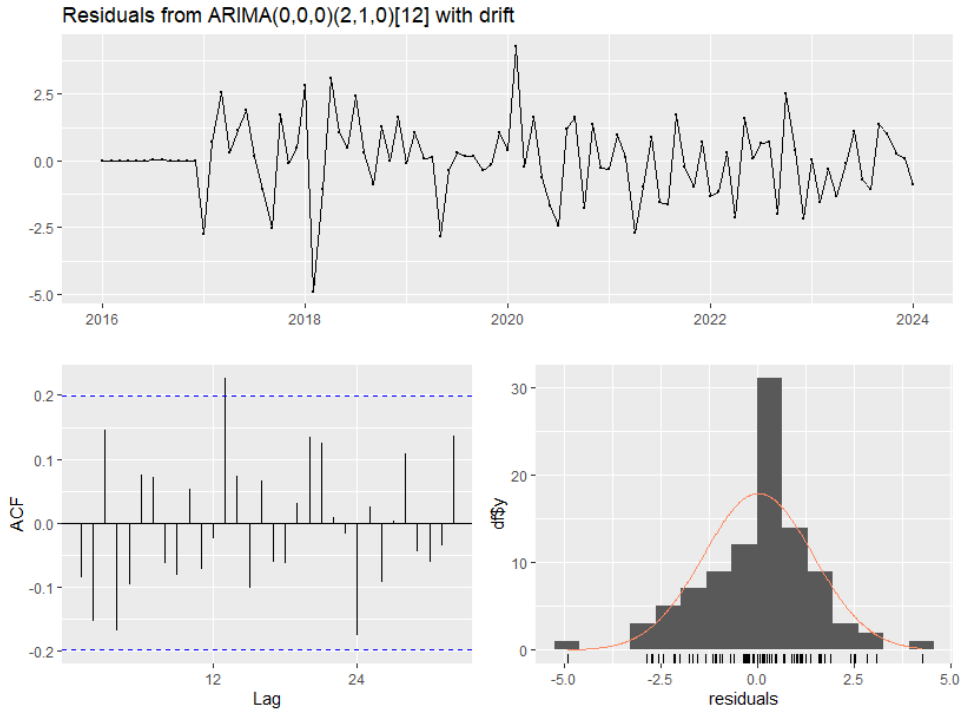
SARIMA(0,0,0)(2,1,0)[12] :

Statistique $Q*$: $Q*=21.777$.
Degrés de liberté (df) : df=17.
P-value : p=0.1935.

Interprétation :

La pp-value est supérieure à 0.050.05, ce qui signifie que je ne rejette pas l'hypothèse nulle. Cela indique que les résidus du modèle SARIMA(0,0,0)(2,1,0)[12] sont indépendants.
Le modèle capture bien les dépendances dans les données. Les résidus ressemblent à un bruit blanc, ce qui est un critère essentiel pour un bon ajustement.





Analyse des graphes obtenus

Les graphiques produits par checkresiduals() incluent généralement trois éléments :

Graphique ACF des résidus.
Histogramme des résidus.
Graphique des résidus dans le temps.

Pour ARIMA(4,0,0) :

ACF des résidus :

On observe des pics significatifs dans l'ACF des résidus. Ces pics montrent que des autocorrélations persistent entre les observations, ce qui va à l'encontre de l'hypothèse de bruit blanc.
Cela confirme que le modèle ne capture pas bien toutes les structures des données.

Histogramme des résidus :
L'histogramme peut montrer une asymétrie ou une déviation par rapport à une distribution normale.
Cela suggère que les résidus ne sont pas bien modélisés.

Graphique des résidus dans le temps :
Si des patterns persistants ou une structure dans les résidus sont visibles, cela indique un manque d'ajustement du modèle.

Pour SARIMA(0,0,0)(2,1,0)[12] :

ACF des résidus :
Aucun pic significatif n'est visible dans l'ACF des résidus. Cela signifie qu'il n'y a pas d'autocorrélation restante dans les résidus, conforme à l'hypothèse de bruit blanc.

Histogramme des résidus :
L'histogramme devrait être symétrique et proche d'une distribution normale. Cela indique que les résidus suivent une distribution normale, ce qui est souhaitable.

Graphique des résidus dans le temps :
Les résidus devraient osciller autour de zéro sans montrer de structure claire ni de tendance. Cela suggère que le modèle capture bien les variations de la série.

Résumé et conclusion :

ARIMA(4,0,0) :
Les résidus ne sont pas indépendants (test Ljung-Box significatif).
Des autocorrélations significatives subsistent dans l'ACF des résidus.
Ce modèle est inapproprié pour vos données.

SARIMA(0,0,0)(2,1,0)[12] :
Les résidus sont indépendants (test Ljung-Box non significatif, $p>0.05p>0.05$).
Aucun pic significatif dans l'ACF des résidus, confirmant qu'ils ressemblent à un bruit blanc.
Ce modèle est bien adapté à vos données et devrait être utilisé pour effectuer des prévisions.

3.3.5 Choix définit d'un modèle en donnant la forme explicite de son équation

Pour répondre à cette question, je vais analyser les informations obtenues jusqu'ici, en argumentant pourquoi le modèle SARIMA(0,0,0)(1,1,0)[12] est préféré par rapport à ARIMA(4,0,0).

Modèle ARIMA(4,0,0)

Forme du modèle : $X_t=\mu+\phi_1X_{t-1}+\phi2X_{t-2}+\phi3X_{t-3}+\phi4X_{t-4}+\epsilon_t$

Avec : $\phi_1=0.7577$, $\phi_2=-0.0250$, $\phi_3=0.1735$, $\phi_4=-0.6113$
Moyenne (μ) : 12.9741,
ϵ_t : bruit blanc.

Critères d'ajustement :
AIC=409.91AIC=409.91,
BIC=425.36BIC=425.36,
$\sigma^2=3.556$ (variance des résidus).

Résidus :
Test de Ljung-Box : p=0.0013 (résidus non indépendants).
L'ACF des résidus montre des autocorrélations significatives, indiquant que le modèle est mal ajusté.

Modèle SARIMA(0,0,0)(2,1,0)[12]

Forme du modèle : $(1-\Phi_1B^{12}-\Phi_2B^{24})(1-B^{12})X_t=\mu+\epsilon_t$

Avec : $\Phi_1=-0.8264$, $\Phi_2=-0.2478$
Drift (μ) : 0.0158,
ϵ_t : bruit blanc.

Critères d'ajustement :
AIC=327.21
BIC=336.98
$\sigma^2=2.349$ (variance des résidus).

Résidus :
Test de Ljung-Box : p=0.1935 (résidus indépendants).
L'ACF des résidus ne montre aucun pic significatif, confirmant que les résidus sont proches d'un bruit blanc.

Comparaison des deux modèles

Critères	ARIMA(4,0,0)	SARIMA(0,0,0)(2,1,0)[12]
AIC	409.91	327.21
BIC	425.36	336.98
Variance des résidus	3.556	2.349
Résidus indépendants ?	Non ($p = 0.0013$)	Oui ($p = 0.1935$)
Saisonnalité incluse ?	Non	Oui

AIC/BIC :
Le modèle SARIMA a des valeurs d'AIC et de BIC beaucoup plus basses, ce qui indique un meilleur ajustement global.

Variance des résidus :
La variance des résidus du modèle SARIMA est plus faible, ce qui indique que les erreurs sont mieux capturées.

Résidus :
Les résidus du modèle SARIMA sont indépendants et ne présentent aucune autocorrélation significative, contrairement au modèle ARIMA.

Saisonnalité :
Les données présentent une composante saisonnière mensuelle (période de 12 mois). Seul le modèle SARIMA inclut cette composante grâce aux paramètres (P, D, Q) [m].

Forme explicite du modèle SARIMA choisi

Le modèle choisi est SARIMA(0,0,0)(2,1,0)[12], dont l'équation explicite est :
$(1-\Phi_1 B^{12}-\Phi_2 B^{24})(1-B^{12})X_t=\mu+\epsilon_t$

Avec les coefficients suivants :
$\Phi 1=-0.8264$
$\Phi 2=-0.2478$
Drift (μ) : 0.0158.

En termes pratiques :

Différenciation saisonnière ($1-B^{12}$) : Élimine la saisonnalité annuelle (12 mois).
Terme AR saisonnier (Φ_1, Φ_2) : Capture les dépendances saisonnières sur 12 mois (B^{12}) et 24 mois (B^{24}).
Bruit blanc (ϵ_t) : Les résidus suivent un bruit blanc, indiquant un bon ajustement.

Conclusion : choix du modèle définitif

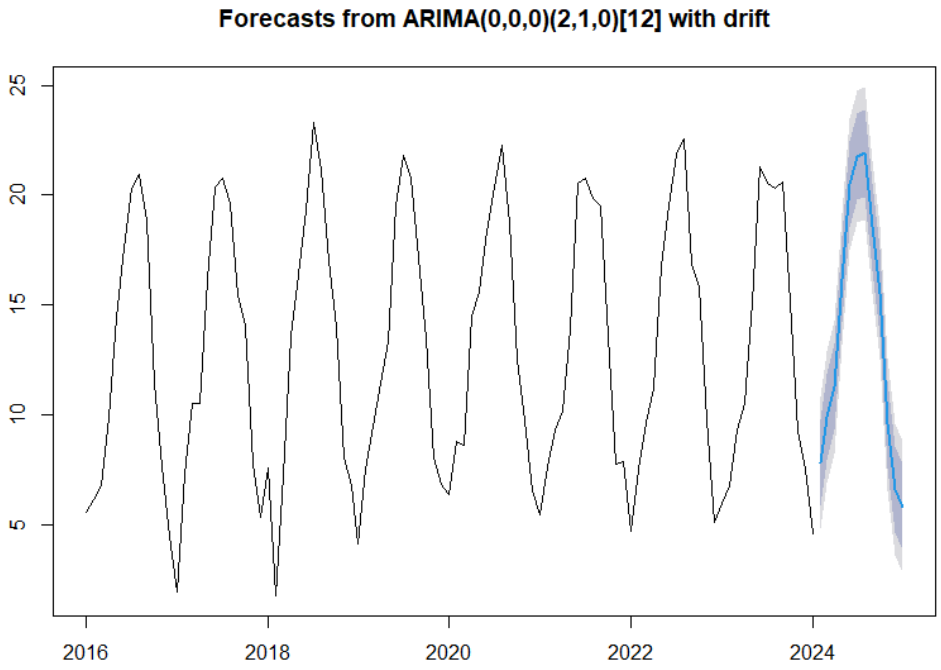
Le modèle SARIMA(0,0,0)(2,1,0)[12] est choisi car :

Il capture la composante saisonnière annuelle des données.
Il produit des résidus indépendants et non autocorrélés.
Il minimise les critères d'ajustement (AIC et BIC) par rapport au modèle ARIMA(4,0,0).

Il reflète mieux la structure des données mensuelles de températures moyennes.

3.3.6 Prévision à l'aide de la prévision

```
forecast_result <- forecast(final_model, h = 12)
plot(forecast_result)
```



Ainsi on peut voir que le modèle sélectionné permet de générer des données en provision qui respecte le modèle car elles sont dans l'intervalle de confiance à 95%.

3.3.7 Analyse a posteriori de la prévision

```
> accuracy(forecast_result)
                     ME      RMSE       MAE        MPE      MAPE      MASE         ACF1
Training set 0.03295679 1.409254 1.003567 -3.796837 12.43031 0.5936124 -0.08615152
```

Ainsi après toutes ces étapes pour essayer de caractériser notre série temporelle via un modèle, on observe qu'avec le modèle sélectionné on obtient de très bons résultats.

ME (Mean Error) = 0.032 qui est la moyenne des écarts

RMSE (Root Mean Square Error) = 1.409 qui est l'erreur quadratique moyenne

MAPE (Mean Absolute Percentage Error) = 12.43 qui est le pourcentage moyen d'erreur

Dans la globalité des résultats on obtient de faibles valeurs ce qui indique de bonnes prévisions et donc un modèle concluant pour exploiter ma série temporelle.

Conclusion

Ce mini-projet sur l'analyse des séries temorelles à l'aide des modèles ARIMA et SARIMA m'a permis de développer une compréhension approfondie des processus stochastiques et de leur application dans des données réelles. L'objectif était de modéliser et de prévoir les températures mensuelles moyennes de la région Ile-de-France sur une période donnée. Bien que les modèles ARIMA et SARIMA offrent une excellente approche pour des prévision simples, il est important de comprendre leurs limites et d'envisager des modèles plus sophistiqués ou des ajustements lorsque les données évoluent de manière non prévisible. Ce projet constitue une étape importante dans le développement de mes compétences analytiques et techniques dans le domaine des séries temporelles