# Modèles SARIMA et Prophet pour la prévision de la température en Irlande

**Une analyse comparative basée sur des données historiques**

https://github.com/Khaoula-ER/SARIMA-vs-PROPHET.git


## Khaoula Aroui

Etudiante en Master statistique pour L'évaluation et la prévision


2022-2023

# Table des matières



# Table des figures

# Introduction

L'analyse de séries chronologiques est une méthode populaire pour étudier les données temporelles. Elle est couramment utilisée pour comprendre les tendances et les modèles saisonniers dans les données, ainsi que pour faire des prévisions. Parmi les différentes méthodes d'analyse de séries chronologiques, SARIMA et Prophet sont deux des plus courantes.

SARIMA (Seasonal Autoregressive Integrated Moving Average) est une méthode statistique qui utilise des modèles autorégressifs intégrés et moyennes mobiles pour modéliser les tendances et les modèles saisonniers dans les données. Il est souvent utilisé pour prévoir des données saisonnières telles que les ventes de détail, les fluctuations du marché boursier et les données météorologiques.

Prophet, quant à lui, est une méthode plus récente développée par Facebook pour modéliser les tendances et les modèles saisonniers dans les données temporelles. Prophet utilise un modèle additif plutôt que multiplicatif, qui permet une plus grande flexibilité dans la modélisation des tendances saisonnières et non saisonnières.

Dans ce travail, nous allons examiner la variation saisonnière de la température en Irlande, en construisant un modèle à l'aide de l'approche statistique SARIMA. Nous allons ensuite comparer les résultats de prévision obtenus avec SARIMA à ceux obtenus avec Prophet.

Nous allons examiner les avantages et les inconvénients de chaque méthode pour comprendre les différences dans les résultats obtenus.

Nous allons également explorer la manière dont les prévisions sont évaluées pour chaque méthode

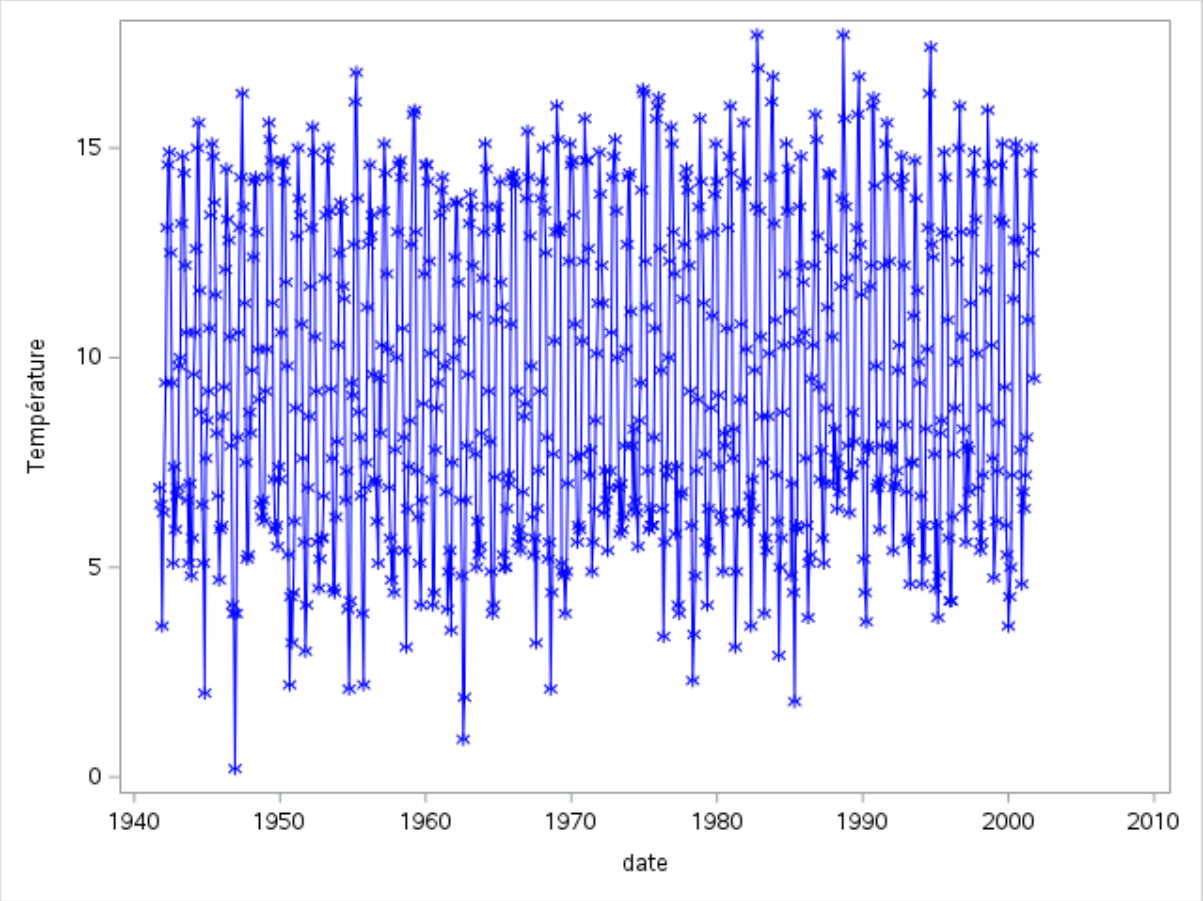
# Données

Un ensemble de données météorologiques mensuelles de Novembre 1941 à Janvier 2018 pour l'aéroport de Dublin, en Irlande, provenant du diffuseur météorologique irlandais est utilisé.

L'ensemble des données se compose de 915 observations pour 11 variables, dont la première est la date en jours et les autres sont des mesures de température. Dans notre analyse, nous nous intéressons particulièrement à la variable de température moyenne.

# Visualisation des données

*Figure 1. Température moyenne au cours du temps*




Le graphique ci-dessus met en évidence la dynamique de la température moyenne sur l'ensemble de la période considérée, ainsi que sur une base annuelle. On peut observer une récurrence des mêmes fluctuations de température, qui se reproduisent presque de manière identique de l'année 1940 à l'année 2001.

# Corrigée en variation saisonnière

La saisonnalité est une caractéristique particulièrement endémique des données météorologiques - ce qui explique pourquoi de nombreuses régions du monde ont quatre saisons.

Lorsque la saisonnalité n'est pas prise en compte, on risque de faire des prévisions erronées des données. Alors que l'on peut prévoir une valeur moyenne pour une série temporelle particulière, les pics et les creux autour de cette moyenne affectent considérablement les prévisions pour cette série temporelle.

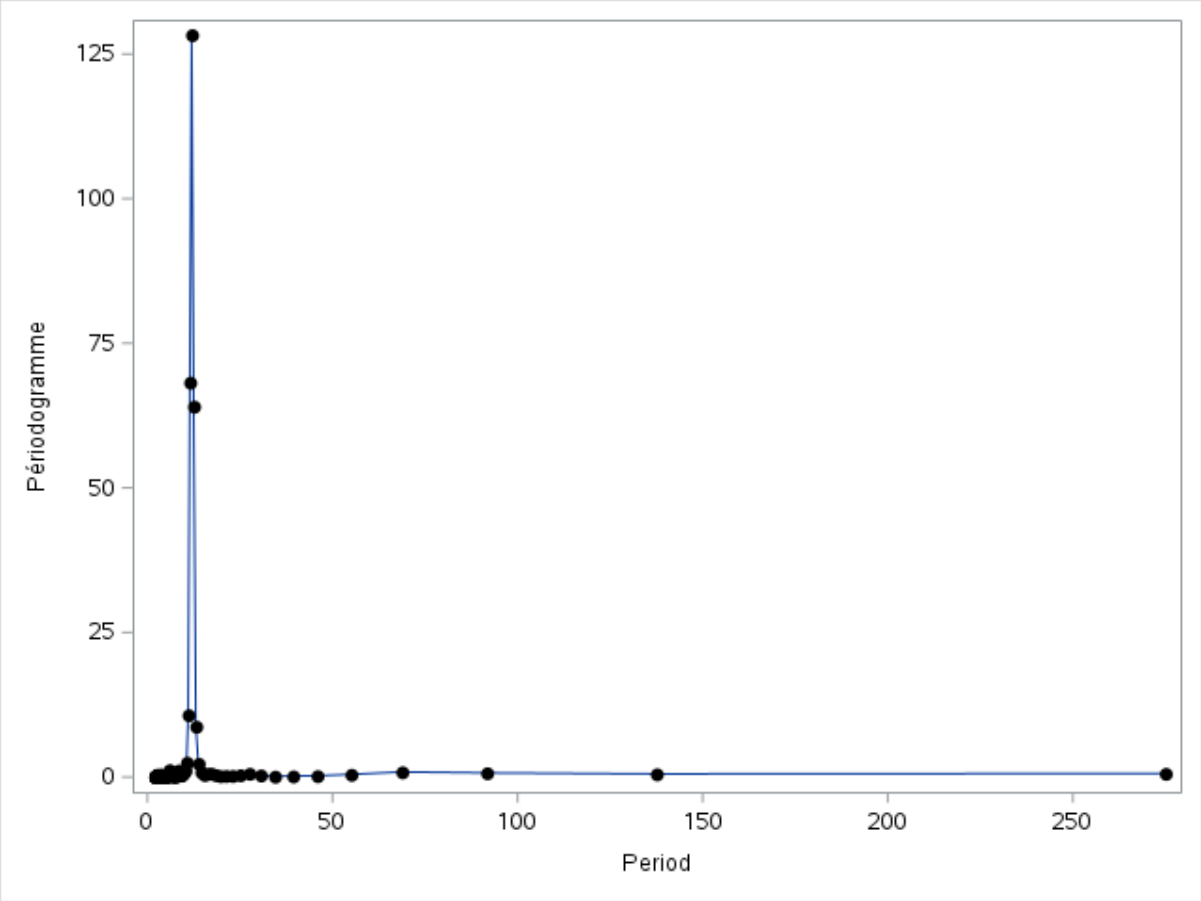
La saisonnalité est une préoccupation importante lorsqu'il s'agit de modéliser des séries temporelles.

En premier lieu, on voulait investiguer la saisonnalité de nos données et puis après corriger cette composante saisonnière.

Comme première étape on voulait confirmer l'existence d'une saisonnalité et la définir.

Une première méthode à adopter est de faire sortir le périodogramme de la densité spectrale.

*Figure 2. Périodogramme de la densité spectrale des températures moyennes – périodes*



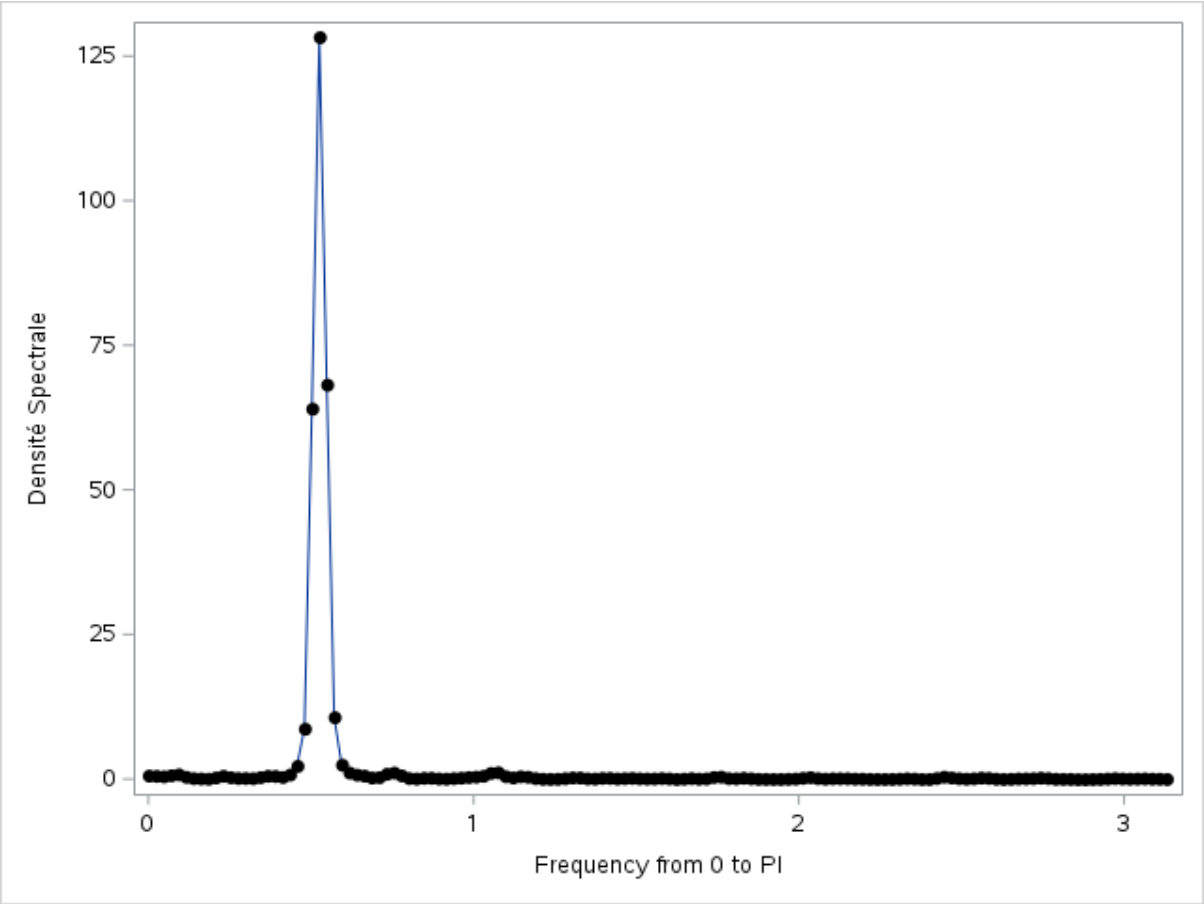

En premier lieu on a calculé le périodogramme de la série temporelle en utilisant la méthode de noyau de Parzen pour estimer la densité spectrale.

En deuxième lieu on a tracé le périodogramme obtenu. La variable "Period" en axe des abscisses correspond aux périodes de la série temporelle et en axe des ordonnées on voit les valeurs du périodogramme.

Le pic le plus élevé dans le périodogramme correspond à la période 12, cela signifie que la série des températures moyenne présente une forte variation à cette période.

Le même graphe est aussi tracé avec les fréquences en abscisses.

*Figure 3.Périodogramme de la densité spectrale des températures moyennes - Fréquences*

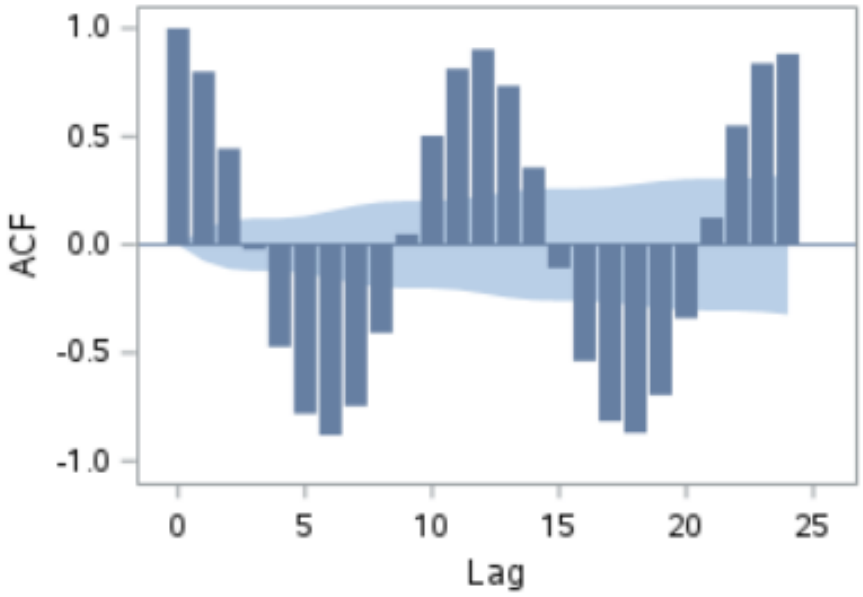



Le graphe ci-dessus montre que la fréquence qui contribue le plus à la variabilité de la série temporelle correspond à la valeur 1/12 (une période de 12 mois).

Une deuxième méthode a été appliquée pour déterminer le paramètre saisonnier à prendre en compte lors de la construction d'un modèle SARIMA. Cette méthode implique une modélisation préliminaire en utilisant la procédure ARIMA, qui permet d'analyser la fonction d'autocorrélation (ACF) et la fonction d'autocorrélation partielle (PACF).

En général, les graphiques ACF et PACF suivant fournissent des informations utiles sur la structure de la série temporelle et peuvent aider à identifier les termes saisonniers qui doivent être inclus dans le modèle SARIMA.

*Figure 4. Tracé de la fonction d'autocorrélation*

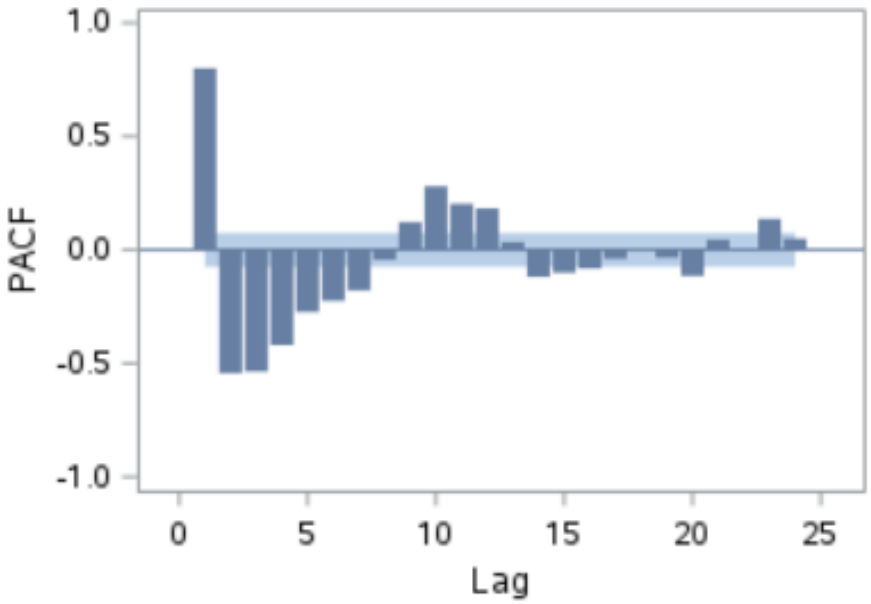



Le tracé de la fonction d'autocorrélation (ACF) permet d'analyser la corrélation entre les observations de la série temporelle à différents décalages. Dans ce cas, après avoir examiné le tracé de l'ACF, on peut observer que la corrélation la plus forte et positive se produit à un décalage de 12. Cela signifie que les observations de la série temporelle sont positivement corrélées avec celles qui ont été enregistrées 12 périodes auparavant.

De plus, on peut remarquer que les décalages précédents (4 à 8) sont négativement corrélés, ce qui suggère qu'il y a une certaine structure saisonnière dans les données. Cette structure saisonnière est attendue, car les températures peuvent varier de manière régulière sur une base annuelle, par exemple en raison des saisons.

En utilisant cette information, on peut en déduire que le paramètre saisonnier approprié pour le modèle est 12, ce qui correspond à la période annuelle de variation saisonnière dans les données de température. L'inclusion de ce paramètre saisonnier dans le modèle SARIMA permettra de capturer cette variation saisonnière et d'améliorer la précision des prévisions pour les périodes futures.

*Figure 5. Tracé de la fonction d'autocorrélation partielle*




Le tracé de la fonction d'autocorrélation partielle (PACF) permet également d'analyser la corrélation entre les observations de la série temporelle à différents décalages, mais en prenant en compte l'influence des décalages intermédiaires.

Dans ce cas, on peut observer que le tracé de la PACF présente une forte coupure au lag 1, ce qui suggère qu'il y a une relation linéaire entre les observations à ces deux décalages. Cela implique que la série temporelle suit un processus autorégressif d'ordre 1 (AR(1)), où chaque observation dépend linéairement de l'observation précédente.

De plus, on peut noter que la sortie montre clairement un phénomène saisonnier et une tendance affine en croissance. La saisonnalité peut être modélisée en utilisant le paramètre saisonnier identifié précédemment, tandis que la tendance peut être modélisée en utilisant une approche de différenciation d'ordre 1 ou d'utilisation d'une composante de tendance (par exemple, une régression linéaire sur le temps).

En utilisant ces informations, on peut conclure que la valeur appropriée pour p est 1, ce qui correspond au nombre de décalages à prendre en compte pour capturer la corrélation linéaire entre les observations.

En résumé, le modèle SARIMA approprié pour ces données de température saisonnières comporte une composante AR(1), et une période saisonnière de 12.

Déterminons maintenant le meilleur modèle SARIMA à adopter pour ces données.

# Stationnarité

On a utilisé le test de Dickey-Fuller augmenté pour évaluer la stationnarité de la série chronologique, avec un maximum de nombre de lag de 6.

Le test de Dickey-Fuller augmenté (ADF) est un test statistique utilisé pour déterminer si une série chronologique est stationnaire ou non. La nullité de ce test est que la série chronologique possède une racine unitaire, ce qui signifie que la série n'est pas stationnaire. Si la valeur-p est inférieure à un

certain niveau de signification (généralement 0,05 ou 0,01), on peut rejeter la nullité et conclure que la série est stationnaire.

Le test de Philips-Perron évalue aussi si la série est stationnaire en termes de moyenne. Plus précisément, il teste si la série a une tendance déterministe ou stochastique.

Nous avons exploité aussi ces deux tests de stationnarité pour déterminer les valeurs de d et D dans ARIMA(p,d,q)(P,D,Q)s. On a obtenu les sorties suivantes.

*Figure 6. Résultats du test de Dickey-Fuller augmenté*

| Augmented Dickey-Fuller Unit Root Tests | | | | | | | |
|---|---|---|---|---|---|---|---|
| Type | Lags | Rho | Pr < Rho | Tau | Pr < Tau | F | Pr > F |
| Zero Mean | 0 | -960.125 | 0.0001 | -40.19 | <.0001 | | |
| | 1 | -1677.26 | 0.0001 | -29.00 | <.0001 | | |
| | 2 | -3015.22 | 0.0001 | -21.42 | <.0001 | | |
| | 3 | 6263.745 | 0.9999 | -19.07 | <.0001 | | |
| | 4 | 1820.881 | 0.9999 | -16.70 | <.0001 | | |
| | 5 | 898.4728 | 0.9999 | -15.83 | <.0001 | | |
| | 6 | 546.1373 | 0.9999 | -15.59 | <.0001 | | |
| Single Mean | 0 | -960.125 | 0.0001 | -40.16 | <.0001 | 806.60 | 0.0010 |
| | 1 | -1677.22 | 0.0001 | -28.98 | <.0001 | 419.81 | 0.0010 |
| | 2 | -3015.34 | 0.0001 | -21.41 | <.0001 | 229.11 | 0.0010 |
| | 3 | 6264.688 | 0.9999 | -19.05 | <.0001 | 181.55 | 0.0010 |
| | 4 | 1821.036 | 0.9999 | -16.68 | <.0001 | 139.15 | 0.0010 |
| | 5 | 898.7623 | 0.9999 | -15.81 | <.0001 | 125.09 | 0.0010 |
| | 6 | 546.2149 | 0.9999 | -15.58 | <.0001 | 121.35 | 0.0010 |
| Trend | 0 | -960.204 | 0.0001 | -40.14 | <.0001 | 805.77 | 0.0010 |
| | 1 | -1678.13 | 0.0001 | -28.97 | <.0001 | 419.55 | 0.0010 |
| | 2 | -3024.99 | 0.0001 | -21.41 | <.0001 | 229.17 | 0.0010 |
| | 3 | 6191.827 | 0.9999 | -19.07 | <.0001 | 181.85 | 0.0010 |
| | 4 | 1813.688 | 0.9999 | -16.69 | <.0001 | 139.27 | 0.0010 |
| | 5 | 897.3027 | 0.9999 | -15.81 | <.0001 | 125.04 | 0.0010 |
| | 6 | 545.9051 | 0.9999 | -15.58 | <.0001 | 121.35 | 0.0010 |

*Figure 7. Résultats du test de Philips-Perron*

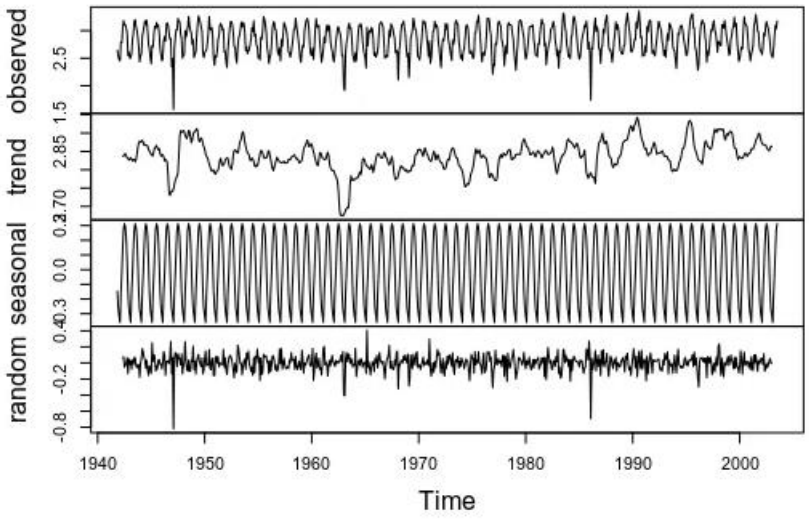
| Phillips-Perron Unit Root Tests | | | | | |
|---|---|---|---|---|---|
| Type | Lags | Rho | Pr < Rho | Tau | Pr < Tau |
| Zero Mean | 0 | -960.125 | 0.0001 | -40.19 | <.0001 |
| | 1 | -931.327 | 0.0001 | -41.12 | <.0001 |
| | 2 | -871.163 | 0.0001 | -43.97 | <.0001 |
| | 3 | -844.003 | 0.0001 | -45.90 | <.0001 |
| | 4 | -820.078 | 0.0001 | -48.14 | <.0001 |
| | 5 | -801.493 | 0.0001 | -50.39 | <.0001 |
| | 6 | -784.276 | 0.0001 | -53.06 | <.0001 |
| Single Mean | 0 | -960.125 | 0.0001 | -40.16 | <.0001 |
| | 1 | -931.326 | 0.0001 | -41.09 | <.0001 |
| | 2 | -871.162 | 0.0001 | -43.94 | <.0001 |
| | 3 | -844.002 | 0.0001 | -45.86 | <.0001 |
| | 4 | -820.077 | 0.0001 | -48.10 | <.0001 |
| | 5 | -801.492 | 0.0001 | -50.35 | <.0001 |
| | 6 | -784.274 | 0.0001 | -53.01 | <.0001 |
| Trend | 0 | -960.204 | 0.0001 | -40.14 | <.0001 |
| | 1 | -931.348 | 0.0001 | -41.07 | <.0001 |
| | 2 | -871.110 | 0.0001 | -43.92 | <.0001 |
| | 3 | -843.882 | 0.0001 | -45.86 | <.0001 |
| | 4 | -819.931 | 0.0001 | -48.10 | <.0001 |
| | 5 | -801.347 | 0.0001 | -50.36 | <.0001 |
| | 6 | -784.144 | 0.0001 | -53.03 | <.0001 |

Dans les résultats présentés le test de Philips-Perron  et de Dickey-Fuller augmenté sont effectués pour différents types de racines unitaires et pour plusieurs valeurs de décalages allant de 1 à 6. Les résultats montrent que pour tous les types de racines unitaires, la valeur de la statistique de test est très faible (proche de zéro) et la valeur p est inférieure au niveau de signification de 0,05. Cela suggère que la série est stationnaire en termes de moyenne, ce qui est une condition nécessaire pour modéliser la série avec un modèle ARIMA.

Du fait que les tests ont été fait pour d=0 et D=1 on peut confirmer que ces valeurs sont appropriées pour notre modèle à construire.

Désormais on bascule vers R pour les étapes qui suivent, en premier lieu on va décomposer notre série chronologique.

*Figure 8. Décomposition de la série chronologique des température moyenne*




De ce qui précède, on confirme qu'il existe une composante saisonnière claire dans la série chronologique. Comme l'indique également le graphique ACF, le modèle ARIMA devra comporter une composante saisonnière attachée.

# Identification et prévision

## 1. Approche classique : SARIMA

En utilisant les données susmentionnées, les procédures suivantes sont effectuées dans R :

La fonction auto.arima est utilisée pour examiner la meilleure configuration ARIMA pour les données d'entraînement (les premiers 80 % de toutes les données de température).

Les valeurs prédites sont ensuite comparées aux valeurs de test (les 20 % restants des données) pour déterminer la précision du modèle.

Enfin, le test de Ljung-Box est utilisé pour déterminer si les données sont distribuées de manière indépendante ou présentent une corrélation sérielle.

*Figure 9. Détermination de la meilleure configuration ARIMA*

```
Fitting models using approximations to speed things up...

ARIMA(2,0,2)(1,1,1)[12] with drift         : Inf
ARIMA(0,0,0)(0,1,0)[12] with drift         : 2700.554
ARIMA(1,0,0)(1,1,0)[12] with drift         : 2489.563
ARIMA(0,0,1)(0,1,1)[12] with drift         : Inf
ARIMA(0,0,0)(0,1,0)[12]                    : 2693.995
ARIMA(1,0,0)(0,1,0)[12] with drift         : 2681.764
ARIMA(1,0,0)(2,1,0)[12] with drift         : 2419.911
ARIMA(1,0,0)(2,1,1)[12] with drift         : 2311.394
ARIMA(1,0,0)(1,1,1)[12] with drift         : 2288.846
ARIMA(1,0,0)(0,1,1)[12] with drift         : Inf
ARIMA(1,0,0)(1,1,2)[12] with drift         : Inf
ARIMA(1,0,0)(0,1,2)[12] with drift         : Inf
ARIMA(1,0,0)(2,1,2)[12] with drift         : Inf
ARIMA(0,0,0)(1,1,1)[12] with drift         : 2330.622
ARIMA(2,0,0)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,1)(1,1,1)[12] with drift         : 2294.973
ARIMA(0,0,1)(1,1,1)[12] with drift         : 2307.532
ARIMA(2,0,1)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,0)(1,1,1)[12]                    : 2282.273
ARIMA(1,0,0)(0,1,1)[12]                    : Inf
ARIMA(1,0,0)(1,1,0)[12]                    : 2482.985
ARIMA(1,0,0)(2,1,1)[12]                    : 2305.159
ARIMA(1,0,0)(1,1,2)[12]                    : Inf
ARIMA(1,0,0)(0,1,0)[12]                    : 2675.208
ARIMA(1,0,0)(0,1,2)[12]                    : Inf
ARIMA(1,0,0)(2,1,0)[12]                    : 2413.332
ARIMA(1,0,0)(2,1,2)[12]                    : Inf
ARIMA(0,0,0)(1,1,1)[12]                    : 2324.069
ARIMA(2,0,0)(1,1,1)[12]                    : Inf
ARIMA(1,0,1)(1,1,1)[12]                    : 2288.405
ARIMA(0,0,1)(1,1,1)[12]                    : 2300.974
ARIMA(2,0,1)(1,1,1)[12]                    : Inf
```

```
Now re-fitting the best model(s) without approximations...

ARIMA(1,0,0)(1,1,1)[12]                    : Inf
ARIMA(1,0,1)(1,1,1)[12]                    : Inf
ARIMA(1,0,0)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,1)(1,1,1)[12] with drift         : Inf
ARIMA(0,0,1)(1,1,1)[12]                    : Inf
ARIMA(1,0,0)(2,1,1)[12]                    : Inf
ARIMA(0,0,1)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,0)(2,1,1)[12] with drift         : Inf
ARIMA(0,0,0)(1,1,1)[12]                    : Inf
ARIMA(0,0,0)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,0)(2,1,0)[12]                    : 2423.105

Best model: ARIMA(1,0,0)(2,1,0)[12]
```

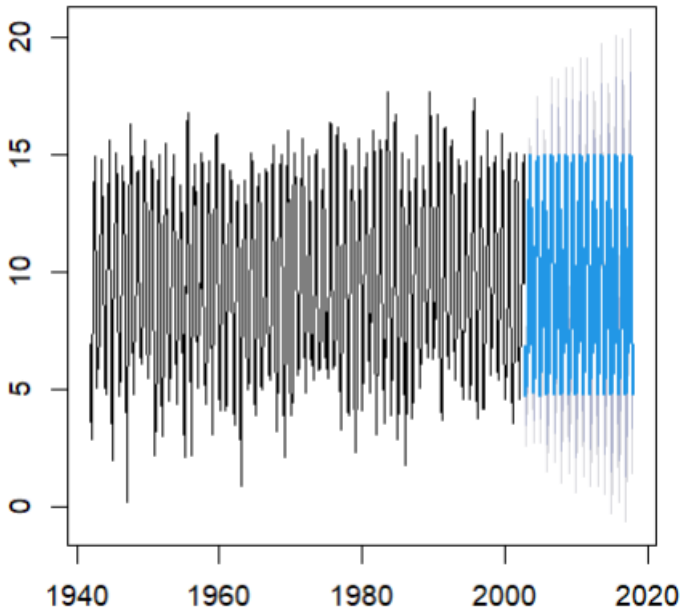
```
> fitweatherarima
Series: meant
ARIMA(1,0,0)(2,1,0)[12]

Coefficients:
         ar1     sar1     sar2
      0.2595  -0.6566  -0.3229
s.e.  0.0363   0.0356   0.0356

sigma^2 = 1.627:  log likelihood = -1198.39
AIC=2404.79   AICc=2404.84   BIC=2423.11
> confint(fitweatherarima)
           2.5 %     97.5 %
ar1   0.1883044  0.3307626
sar1 -0.7264288 -0.5868525
sar2 -0.3925918 -0.2531117
```

À partir de ce qui précède, la meilleure configuration identifiée sur la base du BIC est :

ARIMA(1,0,0)(2,1,0)[12]

Voici un graphique de la prévision :

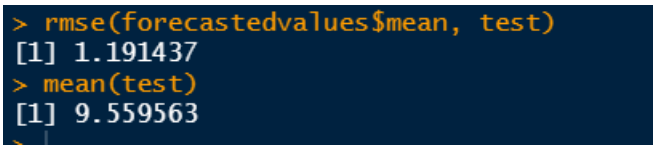
*Figure 10. Prévision d'ARIMA(1,0,0)(2,1,0)[12]*




Maintenant que la configuration a été sélectionnée, les prévisions peuvent être effectuées. Avec une taille de données de test de 183 observations, 183 prévisions sont effectuées en conséquence.

En utilisant la bibliothèque Metrics dans R, les prévisions moyennes peuvent être comparées à l'ensemble de test et évaluées sur la base de l'erreur quadratique moyenne (RMSE).

*Figure 11. Evaluation du modèle*

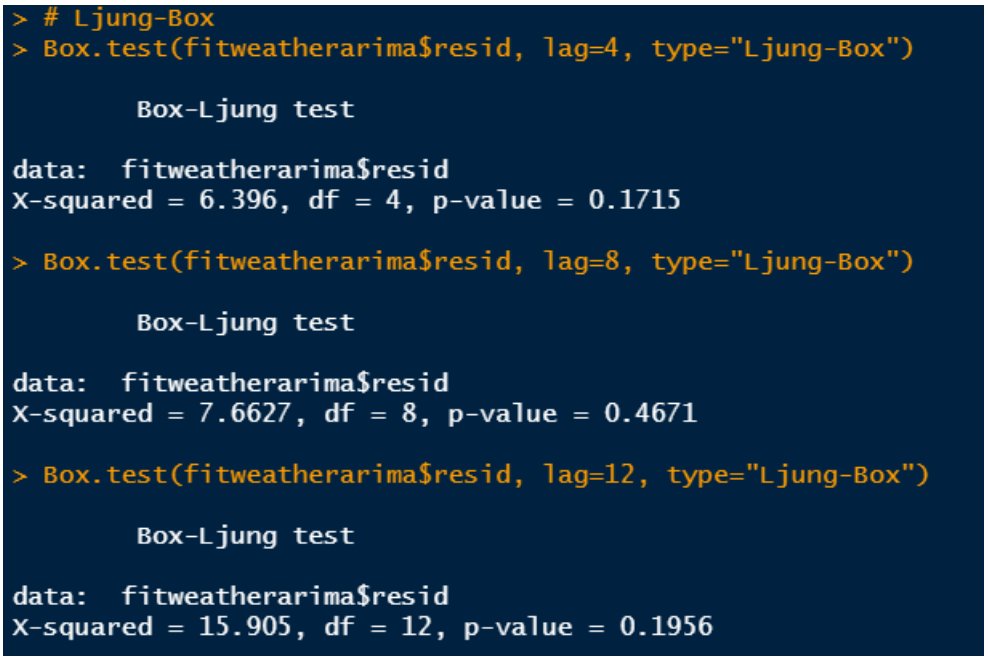



En utilisant la bibliothèque Metrics dans R, les prévisions moyennes peuvent être comparées à l'ensemble de test et évaluées sur la base de l'erreur quadratique moyenne (RMSE).

Un test de Ljung-Box est maintenant effectué. Essentiellement, le test est utilisé pour déterminer si les résidus de notre série chronologique suivent un modèle aléatoire ou s'il existe un degré significatif de non-aléatoire.

H0 : Les résidus suivent un modèle aléatoire HA : Les résidus ne suivent pas un modèle aléatoire Notez que la méthode pour choisir un nombre spécifique de décalages pour le test de Ljung-Box dépend des données en question. Étant donné que nous travaillons avec une série chronologique mensuelle, nous effectuerons le test de Ljung-Box avec les décalages 4, 8 et 12. Pour exécuter ce test dans R, nous utilisons les fonctions suivantes :

*Figure 12. Résultats du test de Ljung-Box*




Nous constatons que sur les décalages 4, 8 et 12, l'hypothèse nulle selon laquelle les décalages suivent un modèle aléatoire ne peut pas être rejetée et donc notre modèle ARIMA est exempt d'autocorrélation.

## 2. Prophet

Il s'agit d'un modèle additif composé de quatre composantes comme suit :

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t$$

Discutons de la signification de chaque composante :

g(t) : Elle représente la tendance et l'objectif est de capturer la tendance générale de la série. Par exemple, le nombre de vues publicitaires sur Facebook est susceptible d'augmenter au fil du temps à mesure que de plus en plus de personnes rejoignent le réseau. Mais quelle serait la fonction exacte de cette augmentation ?

s(t) : Il s'agit de la composante de saisonnalité. Le nombre de vues publicitaires peut également dépendre de la saison. Par exemple, dans l'hémisphère nord pendant les mois d'été, les gens sont susceptibles de passer plus de temps à l'extérieur et moins de temps devant leur ordinateur. De telles fluctuations saisonnières peuvent être très différentes pour différentes séries chronologiques commerciales. La deuxième composante est donc une fonction qui modélise les tendances saisonnières.

h(t) : La composante des vacances. Nous utilisons les informations sur les vacances qui ont un impact clair sur la plupart des séries chronologiques commerciales. Notez que les vacances varient d'une année à l'autre, d'un pays à l'autre, etc., et que les informations doivent donc être explicitement fournies au modèle.

Le terme d'erreur εt représente les fluctuations aléatoires qui ne peuvent pas être expliquées par le modèle. Comme d'habitude, on suppose que εt suit une distribution normale N(0, σ2) avec une moyenne nulle et une variance inconnue σ qui doit être déduite des données.

Le même ensemble de données utilisé précédemment est utilisé pour créer un modèle Prophet en Python afin de prévoir les données de température moyenne pour l'aéroport de Dublin. Les résultats de la prévision sont évalués par rapport à l'ensemble de test en utilisant le RMSE.

Les composantes d'apprentissage et de test sont définies comme suit :

```
[9]: train_df=dataset[:732]
     train_df
```

```
[6]: test_df=dataset[732:915]
     test_df
```

Après, les données sont formatées pour les rendre compatibles avec Prophet pour l'analyse.

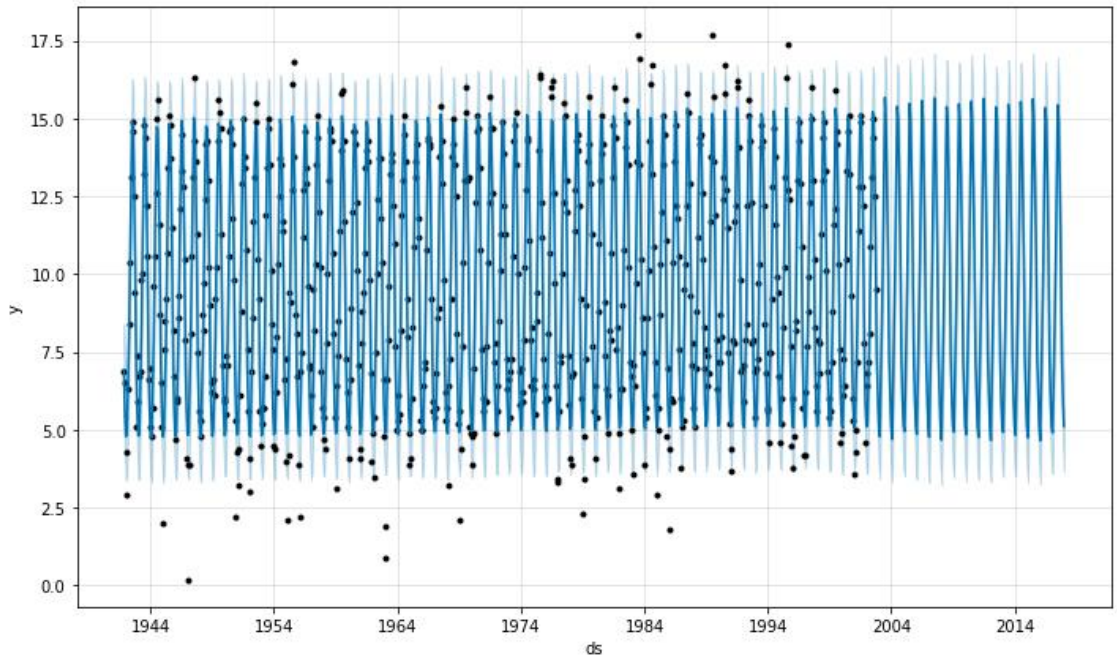
## Implémentation du modèle

Tout d'abord, Un modèle standard Prophet est défini, c'est-à-dire un modèle dans lequel la composante de saisonnalité est sélectionnée automatiquement. Ce modèle est entrainé sur l'ensemble d'entrainement.

On a ensuite fait des prévisions sur l'ensemble de test.

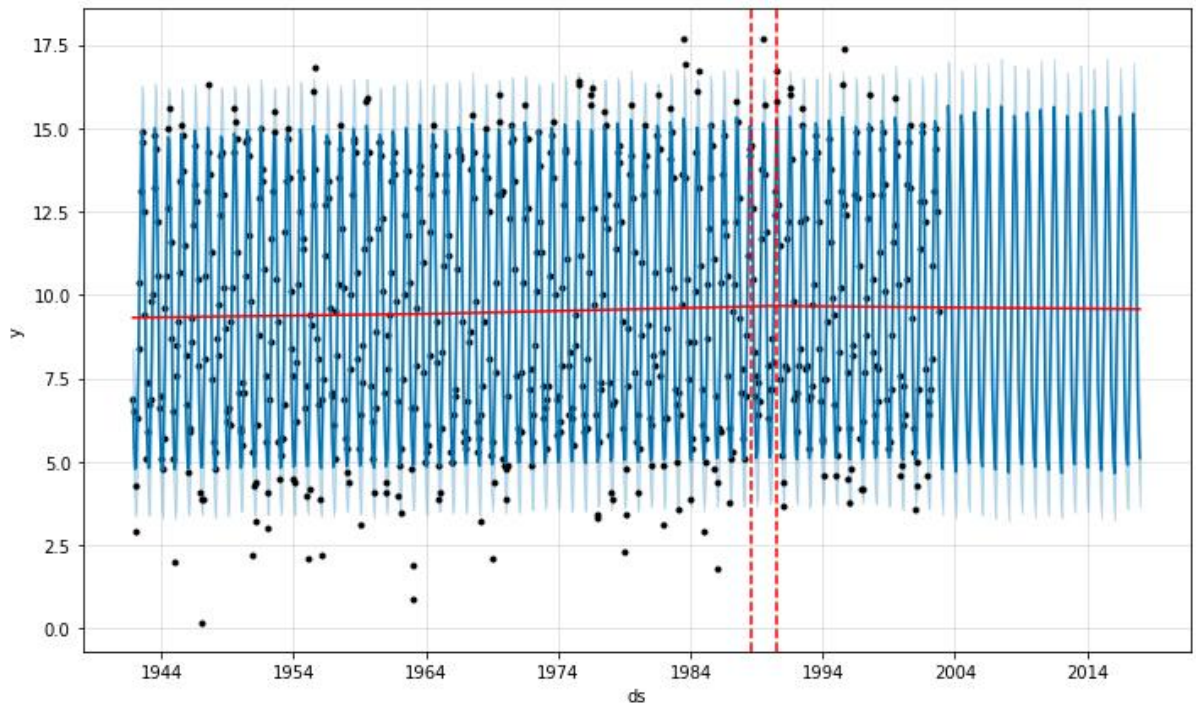
Le graphique des prévisions est comme suit :

*Figure 13. Prévisions du modèle PROPHET*




Le modèle PROPHET offre la possibilité d'identifier les change points pour ensuite les prendre en considération en vue d'améliorer le modèle.

*Figure 14. Prévisions avec identification des change points*



On a pu récupérer les dates des changes points. Après on les a pris en compte et on a élaboré un nouveau jeu de prévisions.

Finalement on évalue les performances de ce dernier modèle sur les données de test.

*Figure 15. Evaluation du modèle PROPHET*

```
from sklearn.metrics import mean_squared_error
from math import sqrt
mse = mean_squared_error(test, yhat)
rmse = sqrt(mse)
print('RMSE: %f' % rmse)
```

```
RMSE: 1.143002
```

Avec une RMSE de 1,14 , le modèle Prophet a en réalité légèrement mieux performé que le modèle ARIMA (qui a donné une RMSE de 1,91).

Le projet est disponible sous le répertoire Github https://github.com/Khaoula-ER/SARIMA-vs-PROPHET.git