

04-08

# Les commandes TBATS, STL et NNETAR

Été 2021

Spécialisation technique en intelligence artificielle  
Algorithmes d'apprentissage non supervisé — 420-A58-SF — M. Swawola, M.Sc.

---

**NOUS ÉCLAIRONS.  
VOUS BRILLEZ.**

---

FORMATION CONTINUE  
ET SERVICES AUX ENTREPRISES



# Sommaire

1. La commande TBATS
2. La commande STL
3. La commande NNETAR
4. Références



# La commande TBATS

# La commande TBATS

- Les séries temporelles peuvent avoir des saisonnalités plus complexes que les méthodes vues précédemment ne peuvent pas modéliser
- Un exemple est **une combinaison de plusieurs saisonnalités** (hebdomadaire et mensuelle, ...)
- On parle alors de **saisonnalités multiples**
- **BATS** et **TBATS** sont deux méthodes capables de modéliser ces saisonnalités
- TBATS est une amélioration de BATS

# La commande TBATS

- **TBATS** signifie

**T**: Terme trigonométrique pour la saisonnalité

**B**: Transformation de Box et Cox pour l'hétérogénéité

**A**: Termes ARMA d'erreurs

**T**: Tendence (éventuellement amortie)

**S**: Terme saisonnier (avec éventuellement des saisons multiples)

# Le modèle TBATS (optionnel)

$$X_t^{(\omega)} = \begin{cases} (X_t^{(\omega)} - 1)/\omega & \text{si } \omega \neq 0 \\ \ln X_t & \text{si } \omega = 0 \end{cases}$$

$$X_t^{(\omega)} = l_{t-1} + \varphi b_{t-1} + \sum_{i=1}^M s_{t-m_i}^{(i)} + d_t$$

$$l_t = l_{t-1} + \varphi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \varphi)b + \varphi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t$$

$$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

# Exemple 1/2

## ■ Code R

```
h02.fit <- tbats(h02)

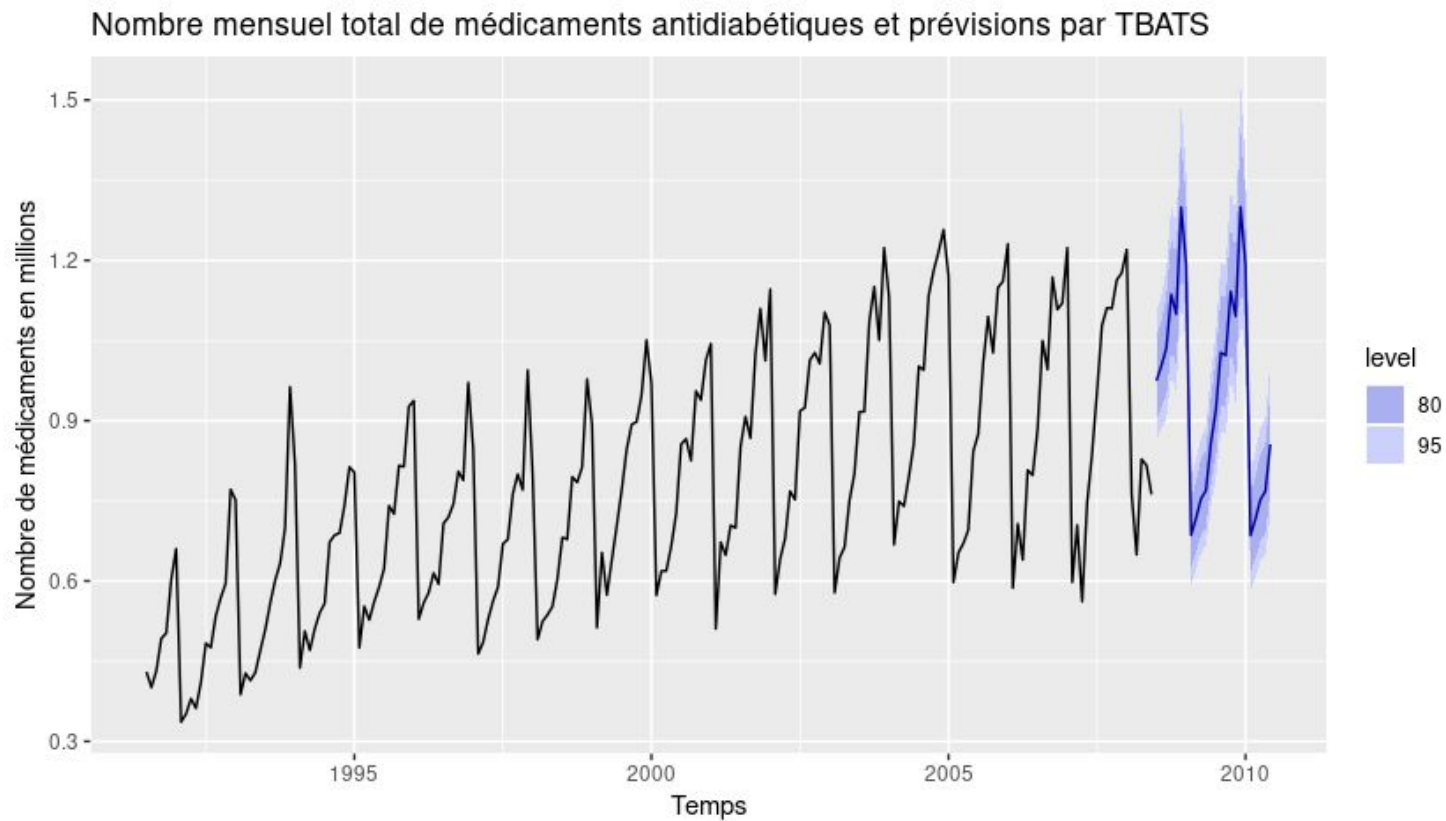
autoplot(forecast(h02.fit)) +

  ggtitle("Nombre mensuel total de médicaments antidiabétiques et prévisions par TBATS") +

  xlab("Temps") +

  ylab("Nombre de médicaments en millions")
```

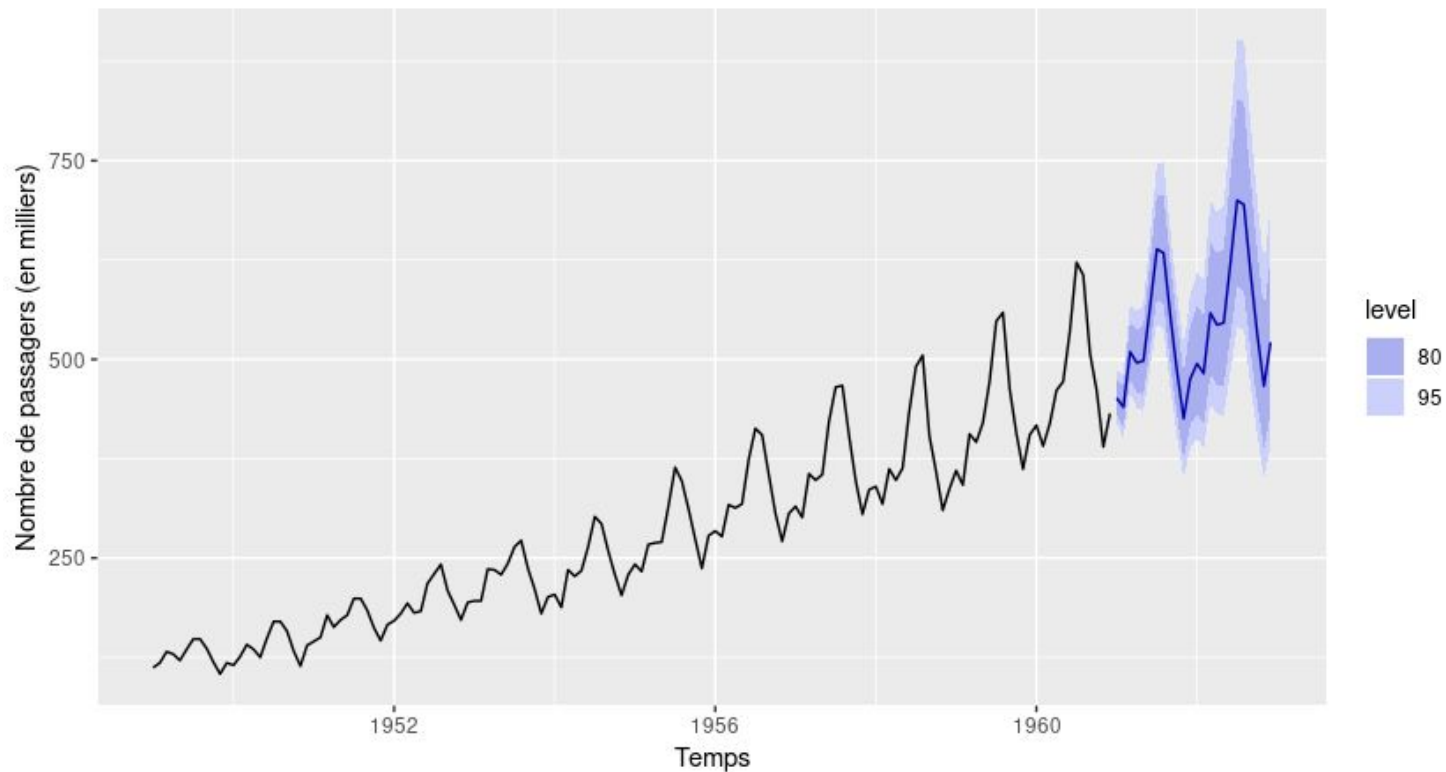
# Exemple 1/2

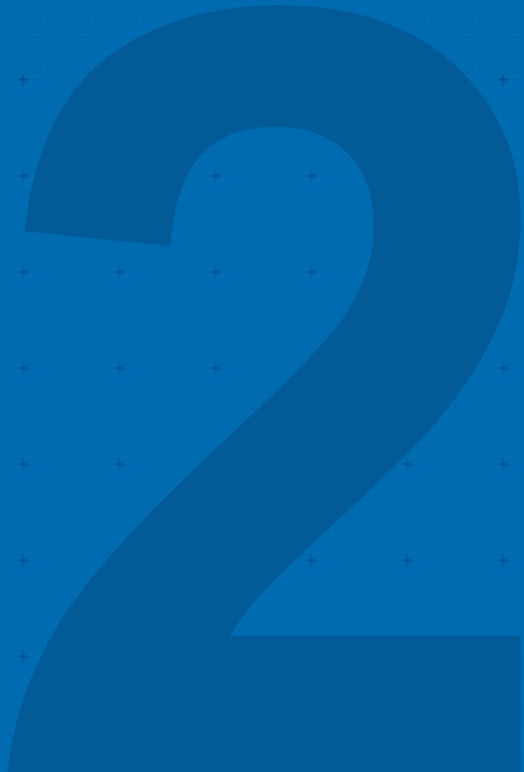




## Exemple 2/2

Nombre mensuel de passagers sur les vols aériens internationaux et prévisions TBATS





# La commande STL

# La commande STL

- **STL** (Seasonal and Trend decomposition using Loess) est une méthode polyvalente et robuste pour la décomposition de séries temporelles
- **Loess** est une méthode efficace d'estimation de relations non linéaires

# Exemple

## ■ Code R

```
fit.airline <- stlf(airline, lambda=0)

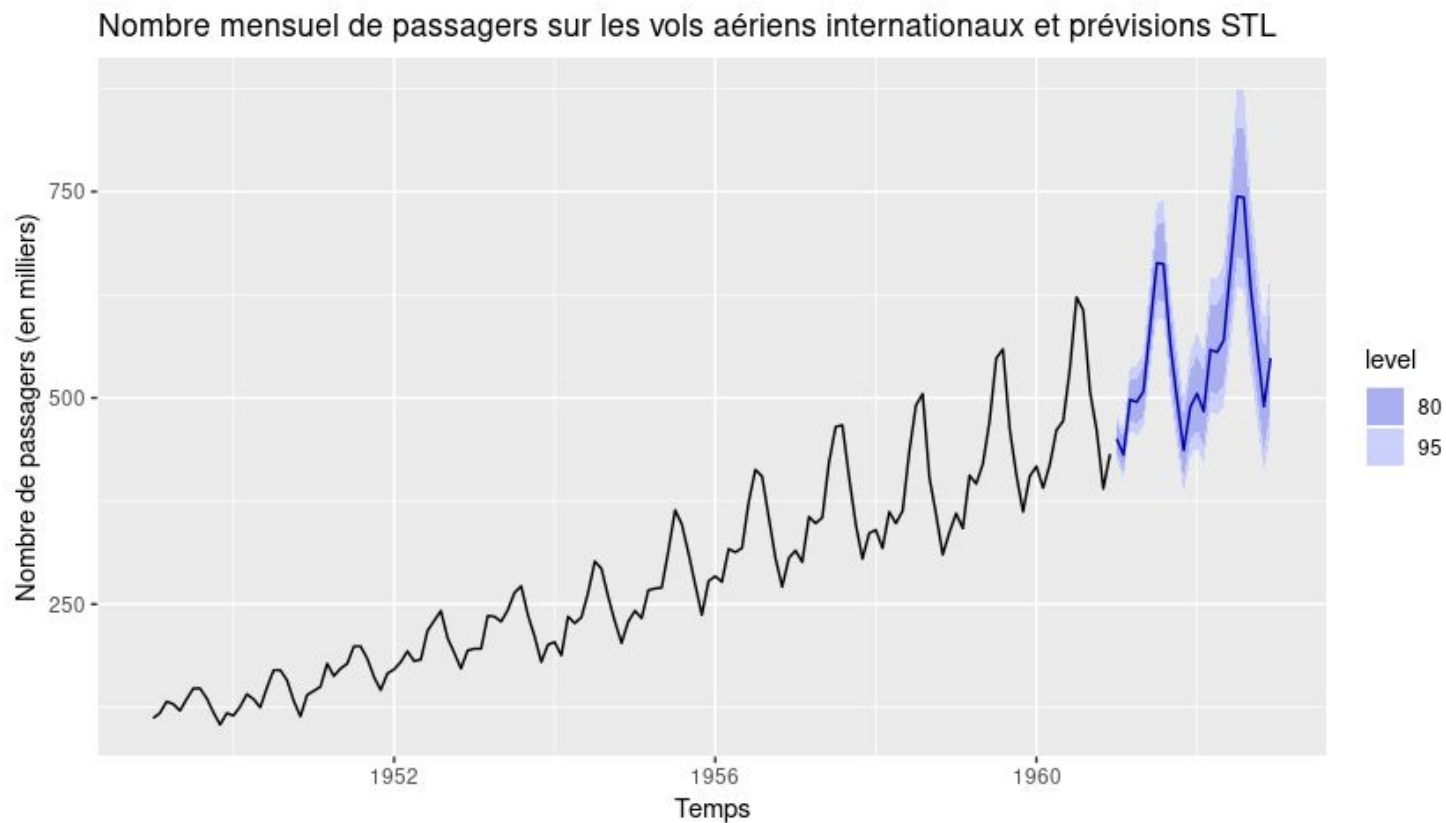
autoplot(fit.airline) +

  ggtitle("Nombre mensuel de passagers sur les vols aériens internationaux et prévisions STL") +

  xlab("Temps") +

  ylab("Nombre de passagers (en milliers)")
```

# Exemple



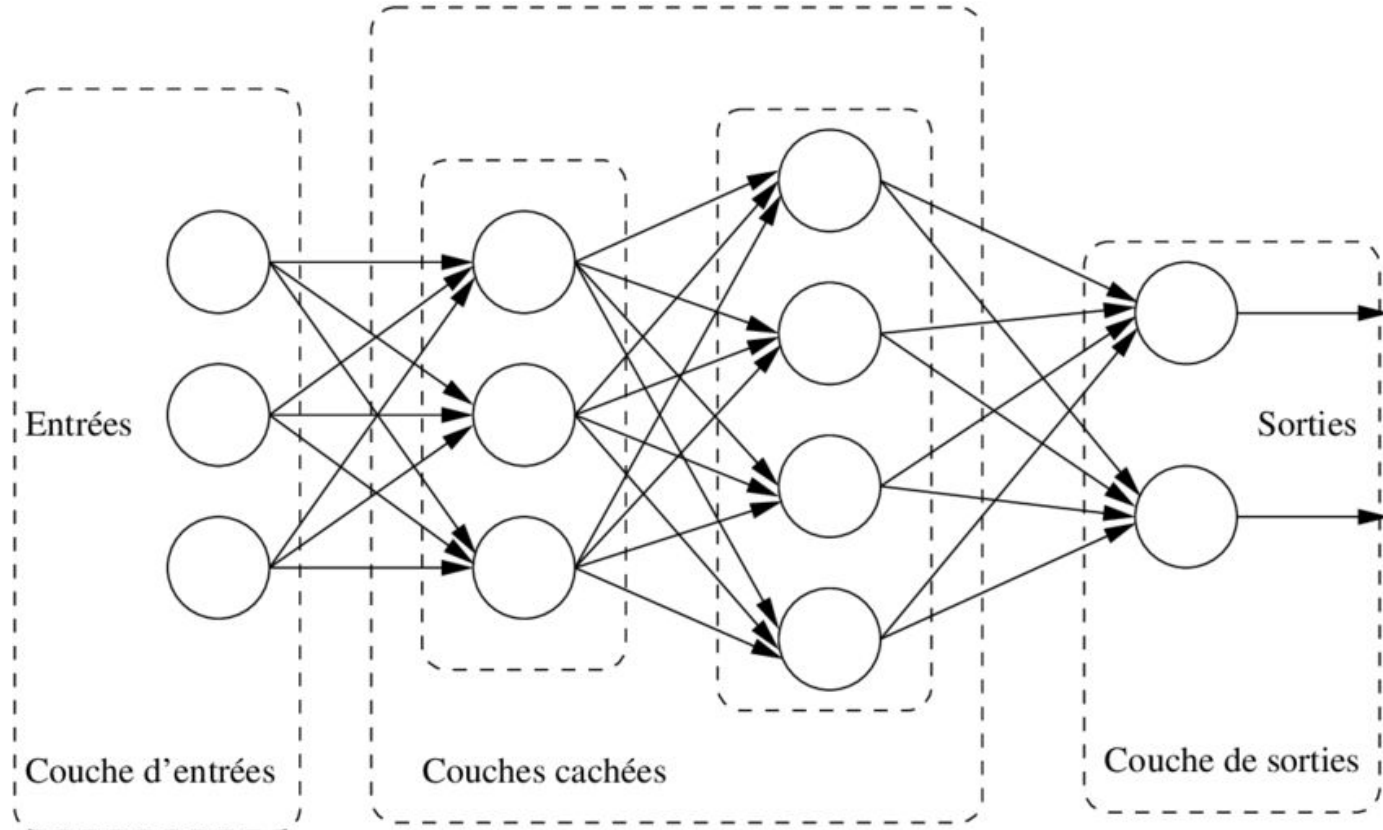
3

# La commande **NNETAR**

# Architecture neuronale

- Un **réseau de neurones** peut être considéré comme une structure de "neurones artificiels" organisée en couches
- Les entrées forment la **couche d'entrée** (input layer) et les prévisions (ou sorties) forment la **couche de sortie** (output layer)
- Il peut également y avoir des **couches intermédiaires** (hidden layer)

# Perceptron multicouche





# Perceptron multicouche

- Les valeurs retardées d'une série temporelle peuvent être utilisées comme entrées dans un réseau de neurones, tout comme on a déjà utilisé les valeurs retardées dans un modèle autorégressif
- On appelle cela un **modèle d'autorégression de réseau neuronal** ou **modèle NNAR**
- On utilise la notation **NNAR(p, k)** pour indiquer qu'il y a **p** entrées de la série retardée et **k** noeuds dans la couche cachée

# Perceptron multicouche

- Par exemple, un modèle **NNAR(9, 5)** est un réseau de neurones avec les neuf dernières observations ( $x_{t-1}, x_{t-2}, \dots, x_{t-9}$ ) utilisées comme entrées pour la prévision de la sortie  $x_t$ , et avec cinq neurones dans la couche cachée

# La commande NNETAR

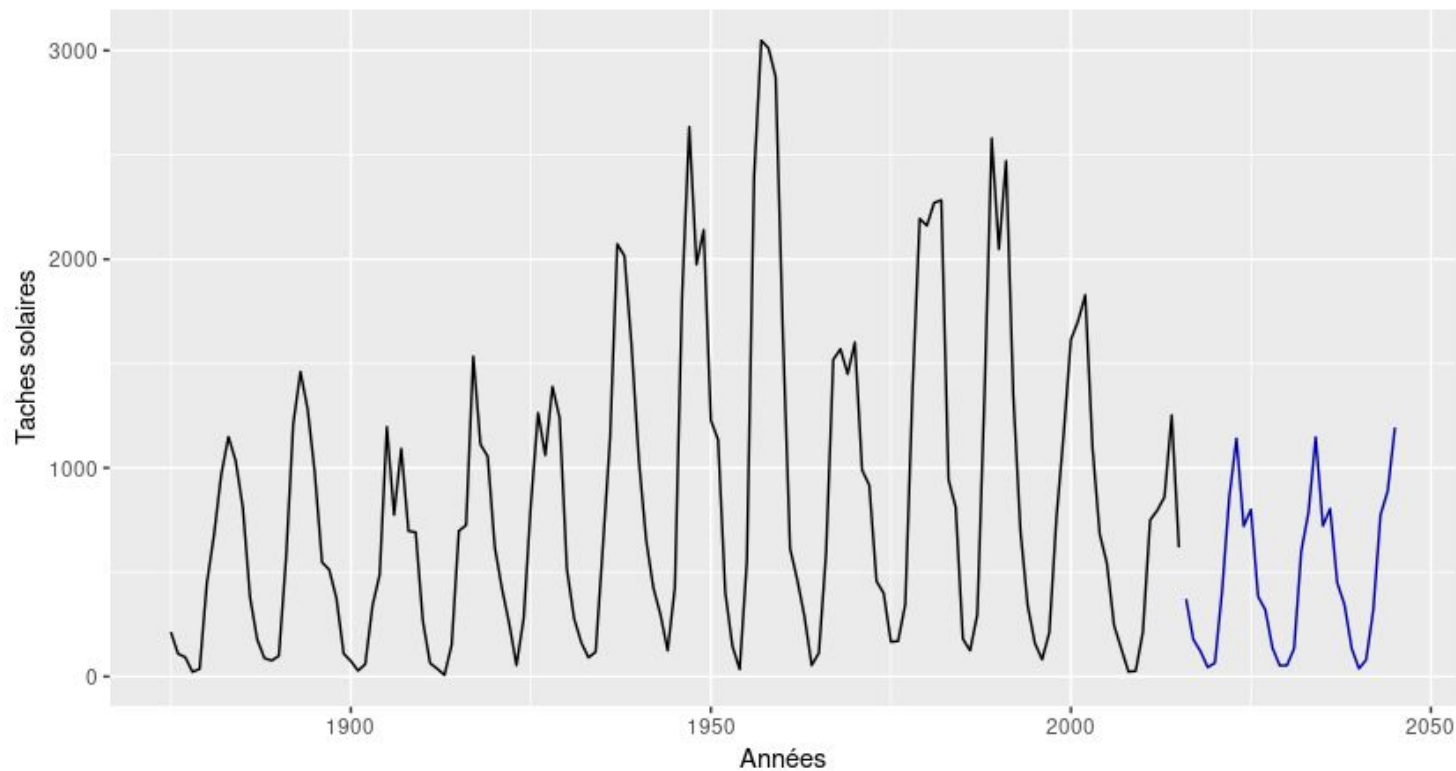
## ■ Code R

```
fit <- nnetar(sunspotarea, lambda=0)

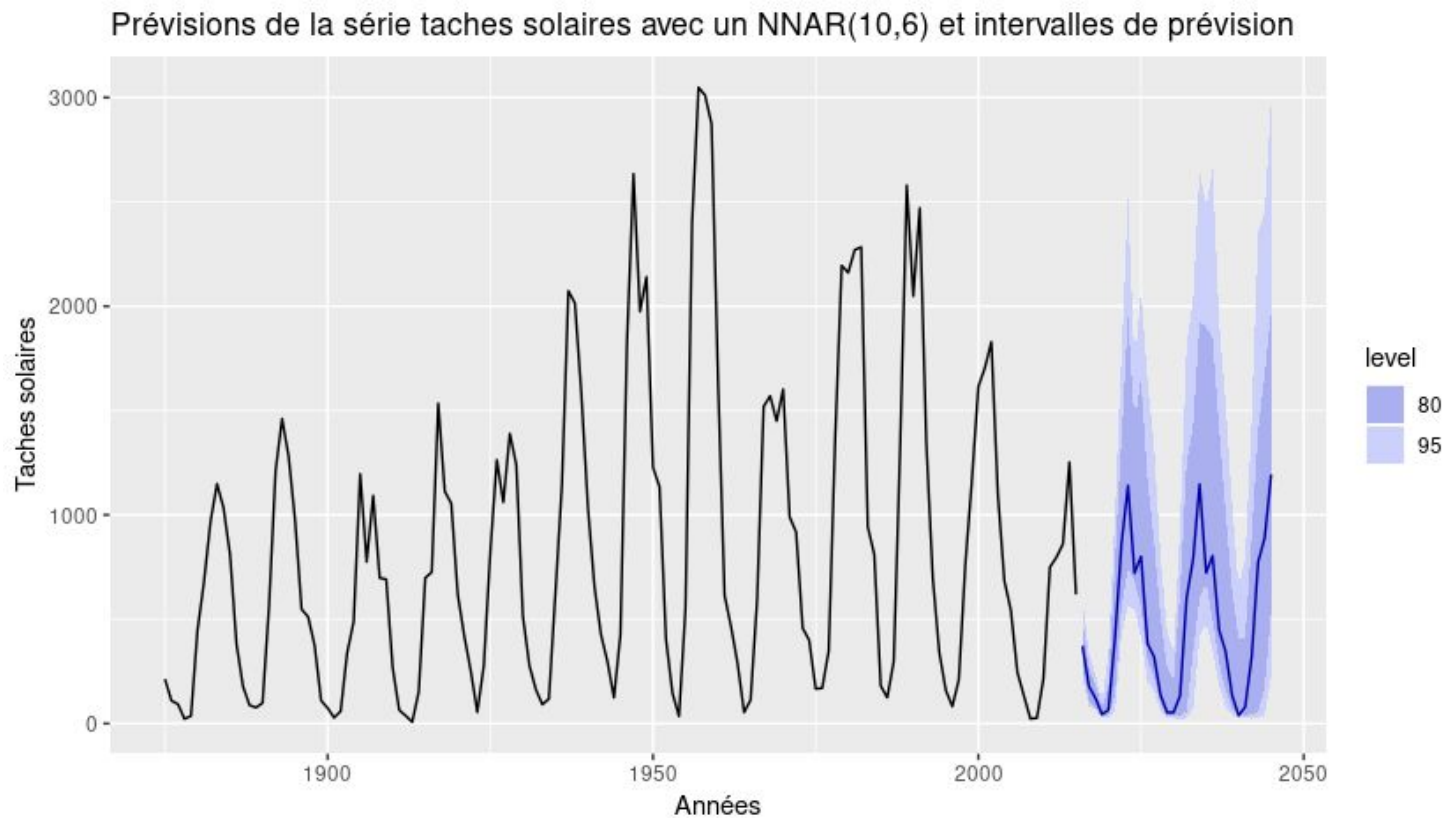
autoplot(forecast(fit, h=30)) +
  xlab("Années") +
  ylab("Taches solaires") +
  ggtitle("Prévisions de la série taches solaires avec un NNAR(10,6)")
```

# Exemple

Prévisions de la série taches solaires avec un NNAR(10,6)



# Exemple





Pull de <https://github.com/mswawola-cegep/420-a58-sf.git>

**04-08-TP**



# Références

# Références

- [1] Cours “R et la prévision de séries temporelles” de Michel Carbon - Université Laval
- [2] [Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing](#)
- [3] [Forecasting Time Series with Multiple Seasonalities using TBATS in Python](#)
- [4] [Time Series Forecasting in R](#)