Fiche de Mission : Epidémiologie

Leo Jean UNITE; Diego CASAS BARCENAS

2024-12-17

Fiche de Mission : Epidémiologie

Description

Le réseau Sentinelles est un réseau de recherche et de veille en soins de premier recours (médecine générale et pédiatrie) en France métropolitaine.

Créé en 1984, il est coordonné par l'équipe "Surveillance et Modélisation des maladies trans missibles" (SUMO) de l'Institut Pierre Louis d'Epidémiologie et de Santé Publique (Iplesp). L'Iplesp est une unité mixte de recherche en Santé (UMR-S 1136, anciennement UMR-S 707, 444 et 263) sous la tutelle conjointe de l'Institut national de la santé et de la recherche médicale (Inserm) et de Sorbonne Université. Le réseau Sentinelles est développé en collaboration avec l'agence nationale de Santé publique, Santé publique France. Il a obtenu une autorisation pour ses travaux scientifiques de la part du CPP et de la CNIL.

Les objectifs principaux du réseau Sentinelles sont :

- la constitution de grandes bases de données en médecine générale et en pédiatrie, à des fins de veille sanitaire et de recherche ;

- le développement d'outils de détection et de prévision épidémique ;

- la mise en place d'études cliniques et épidémiologiques.

Parmi les bases de données de surveillance continue proposées par le réseau figurent : la varicelle (depuis 1991), la diarrhée aiguë (depuis 1991) et les syndromes grippaux (depuis 1985). Une attention particulière doit être accordée à cette dernière base de données, car, à partir de la semaine 2020s12, les incidences de cet indicateur sont estimées secondairement à partir de l'indicateur des infections respiratoires aiguës (IRA, également fourni). La comparaison des estimations des syndromes grippaux entre les périodes pré-Covid-19 (1985 - février 2020) et post-Covid-19 (depuis mars 2020) doit donc être effectuée avec prudence.

Mission

En tant que membre de l'équipe de statisticiens chargée d'analyser ces données, un indicateur vous sera attribué. Vous vous concentrerez sur le taux d'incidence pour 100 000 habitants associé à cet indicateur (colonne inc100 du tableau de données correspondant).

Vous devrez :

- Rappeler le calcul de inc100.
- Traiter vos données : compléter vos données en cas de valeurs manquantes, appliquer d'éventuelles transformations, réaliser des graphiques, etc.
- Montrer la tendance de la série, en utilisant un filtre de moyennes mobiles.
- Calculer et tracer les coefficients saisonniers ; fournir la décomposition de la série.
- Tracer et commenter la série désaisonnalisée (série corrigée des variations saisonnières).
- Tracer et commenter les boxplots des résidus.
- Réaliser une prévision du taux d'incidence de l'indicateur pour le reste de cet automne et pour le prochain hiver en utilisant trois méthodes différentes.
- Evaluer la qualité des prévisions obtenues avec les trois méthodes précédentes. Pour cela, pour chacune des méthodes utilisées dans la partie précédente, utiliser les données jusqu'en 2023 comme données d'entraînement et produire une prévision pour les 46 semaines de l'année 2024. Étant donné que les données des 46 semaines de 2024 (données de test) sont observées, calculer l'erreur quadratique moyenne des prévisions pour chacune des trois méthodes.

Vous devrez produire un rapport synthétique (environ 10 à 12 pages) comprenant une introduction et une conclusion. Juste après la conclusion, inclure une section décrivant en quelques lignes les tâches réalisées par chaque membre du binôme. Toutefois, les deux membres doivent maîtriser l'intégralité du contenu du document, car ils pourront être convoqués à un entretien de 15 minutes après la remise du rapport. Tous les graphiques et indicateurs doivent être commentés.

Le rapport devra être rédigé en français, avec un résumé en anglais à la fin du document.

```
# Lecture des données
data <- read.csv("syndrome_grippaux.csv")

# Vérification de la structure des données
str(data)
```

```
## 'data.frame':    2090 obs. of  10 variables:
##  $ week      : int  202446 202445 202444 202443 202442 202441 202440 202439 202438 202437 ...
##  $ indicator : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ inc       : chr  "59293" "47482" "36039" "46572" ...
##  $ inc_low   : int  50983 40921 30122 39928 60009 71386 76555 82937 82903 49319 ...
##  $ inc_up    : int  67603 54043 41956 53216 75561 87484 93375 100383 100669 63601 ...
##  $ inc100    : chr  "89" "71" "54" "70" ...
##  $ inc100_low: int  77 61 45 60 90 107 114 124 125 74 ...
##  $ inc100_up : int  101 81 63 80 114 131 140 150 151 96 ...
##  $ geo_insee : chr  "FR" "FR" "FR" "FR" ...
##  $ geo_name  : chr  "France" "France" "France" "France" ...
```

```
# Conversion des colonnes 'inc' et 'inc100' en numériques
data$inc <- as.numeric(data$inc)  # Conversion de 'inc' en numérique
```

```
## Warning: NAs introduced by coercion
```

```
data$inc100 <- as.numeric(data$inc100)  # Conversion de 'inc100' en numérique
```

```
## Warning: NAs introduced by coercion
```

```
# Vérification de la conversion
str(data)
```

```
## 'data.frame':    2090 obs. of  10 variables:
##  $ week      : int  202446 202445 202444 202443 202442 202441 202440 202439 202438 202437 ...
##  $ indicator : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ inc       : num  59293 47482 36039 46572 67785 ...
##  $ inc_low   : int  50983 40921 30122 39928 60009 71386 76555 82937 82903 49319 ...
##  $ inc_up    : int  67603 54043 41956 53216 75561 87484 93375 100383 100669 63601 ...
##  $ inc100    : num  89 71 54 70 102 119 127 137 138 85 ...
##  $ inc100_low: int  77 61 45 60 90 107 114 124 125 74 ...
##  $ inc100_up : int  101 81 63 80 114 131 140 150 151 96 ...
##  $ geo_insee : chr  "FR" "FR" "FR" "FR" ...
##  $ geo_name  : chr  "France" "France" "France" "France" ...
```

1. **Rappeler le calcul de inc100.**

```
# Calcul du taux d'incidence pour 100 000 habitants (si nécessaire)
# Ici, 'inc100' est déjà calculé, donc cette étape n'est peut-être pas nécessaire.
# Cependant, on peut vérifier que 'inc100' est cohérent avec 'inc' et 'population'.
if (!all(is.na(data$inc100))) {
  # Affichage de quelques premières lignes pour vérifier les valeurs calculées
  head(data$inc100)
}
```

```
## [1]  89  71  54  70 102 119
```

```
# Si 'inc100' n'était pas déjà calculé, voici comment faire :
# Supposons qu'il y a une colonne 'population' dans les données
# data$inc100 <- (data$inc / data$population) * 100000
```

2. **Traiter vos données : compléter vos données en cas de valeurs manquantes, appliquer d'éventuelles transformations, réaliser des graphiques, etc.**

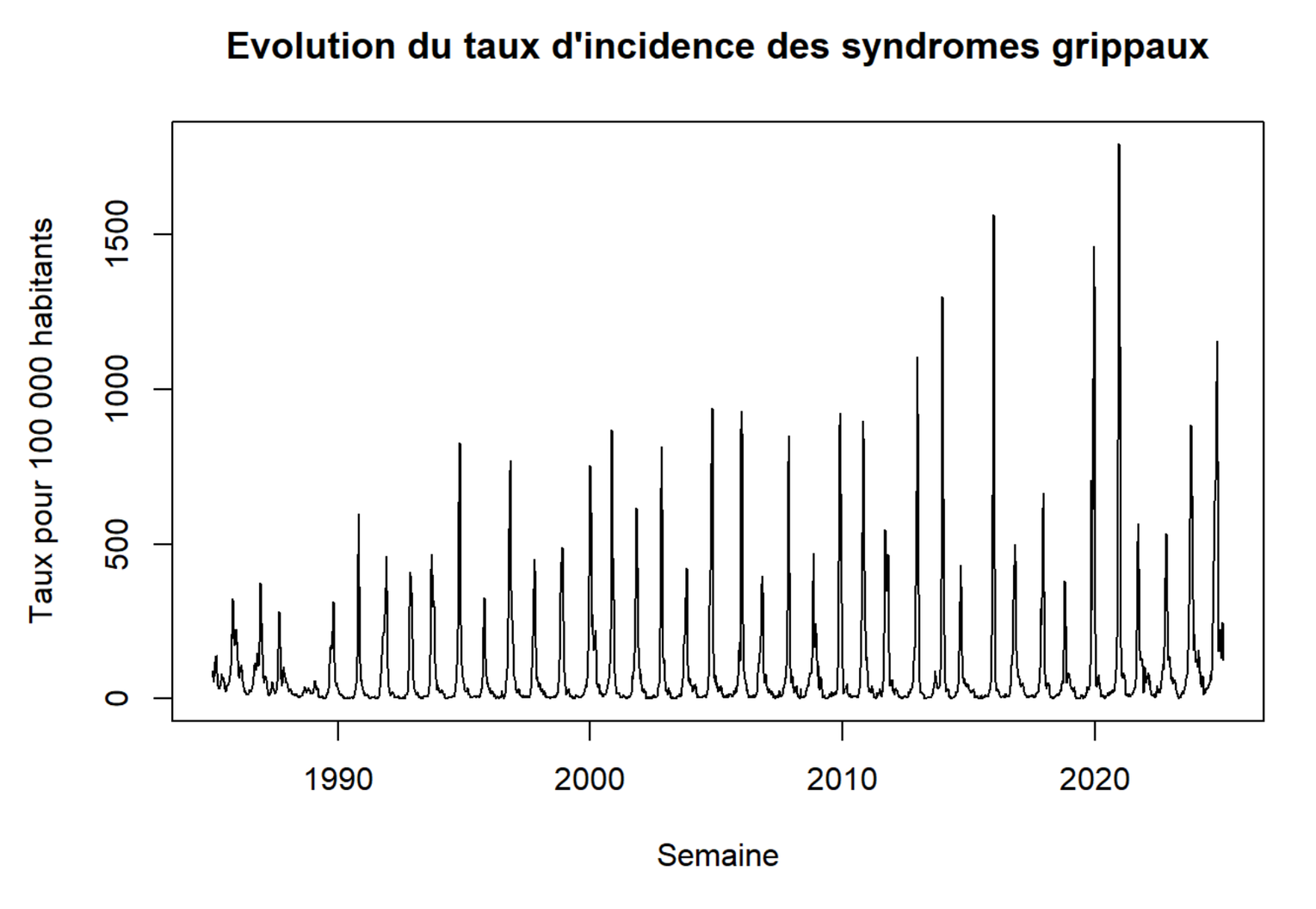
```
# Vérification des valeurs manquantes dans les données
sum(is.na(data))
```

```
## [1] 6
```

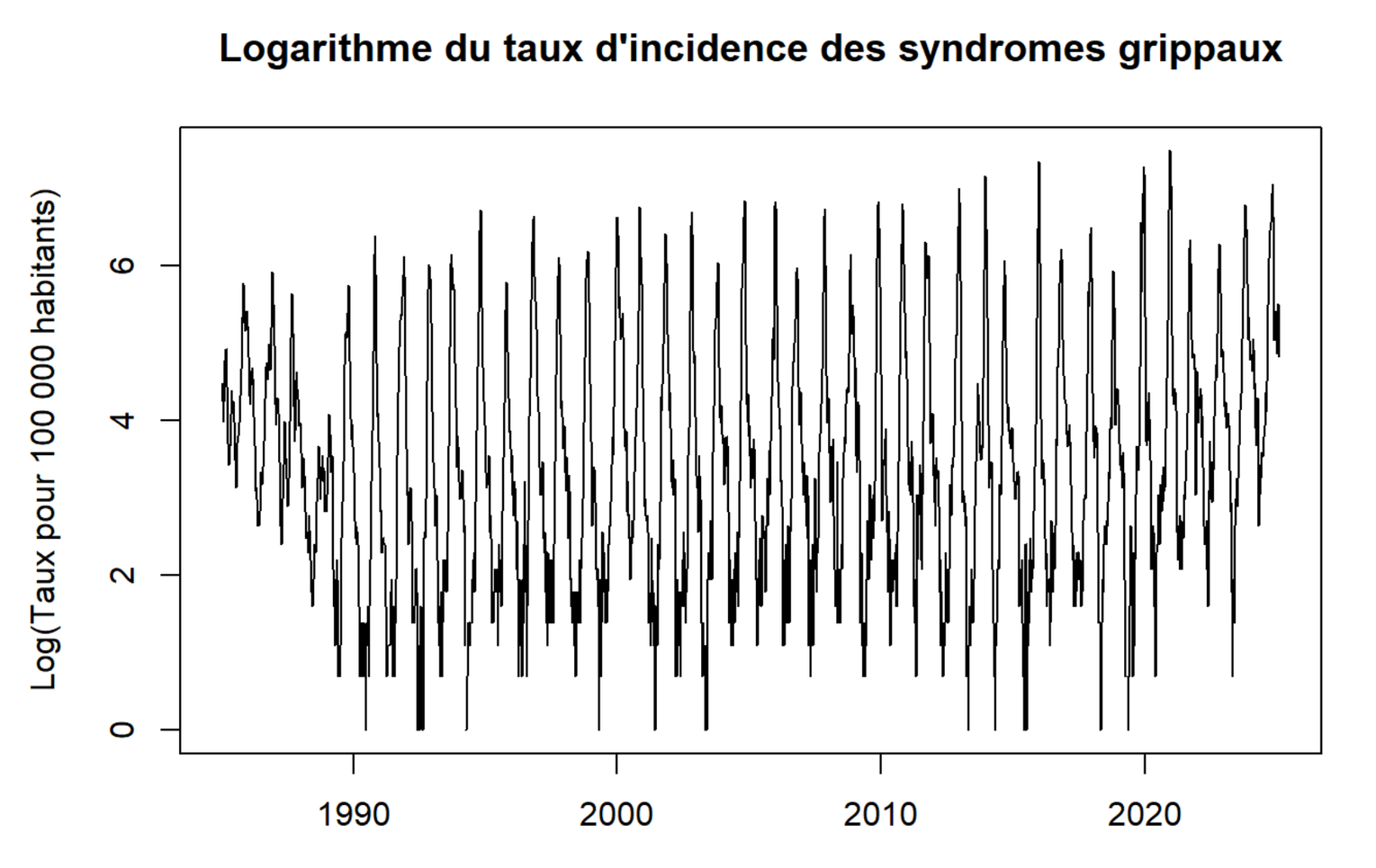
```
# Imputation ou suppression des valeurs manquantes
# Ici, nous utilisons la suppression des lignes avec des valeurs manquantes
data <- na.omit(data)

# Vérification de la structure des données après nettoyage
str(data)
```

```
## 'data.frame':    2089 obs. of  10 variables:
##  $ week      : int  202446 202445 202444 202443 202442 202441 202440 202439 202438 202437 ...
##  $ indicator : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ inc       : num  59293 47482 36039 46572 67785 ...
##  $ inc_low   : int  50983 40921 30122 39928 60009 71386 76555 82937 82903 49319 ...
##  $ inc_up    : int  67603 54043 41956 53216 75561 87484 93375 100383 100669 63601 ...
##  $ inc100    : num  89 71 54 70 102 119 127 137 138 85 ...
##  $ inc100_low: int  77 61 45 60 90 107 114 124 125 74 ...
##  $ inc100_up : int  101 81 63 80 114 131 140 150 151 96 ...
##  $ geo_insee : chr  "FR" "FR" "FR" "FR" ...
##  $ geo_name  : chr  "France" "France" "France" "France" ...
##  - attr(*, "na.action")= 'omit' Named int 1854
##   ..- attr(*, "names")= chr "1854"
```

```
# Conversion de la colonne 'week' en date (si elle n'est pas déjà dans un format approprié)
# Assurons-nous que les semaines sont interprétées correctement en tant que date
data$week <- as.Date(as.character(data$week), format = "%Y%U")  # Adaptation si nécessaire

# Création d'une série temporelle avec 'inc100'
# Nous supposons que 'week' est la variable temporelle et que 'inc100' est l'observable
inc100_ts <- ts(data$inc100, start = c(1985, 3), frequency = 52.143)  # Ajuster selon vos données

# Affichage de l'évolution du taux d'incidence
plot(inc100_ts, main="Evolution du taux d'incidence des syndromes grippaux",
     xlab="Semaine", ylab="Taux pour 100 000 habitants")
```

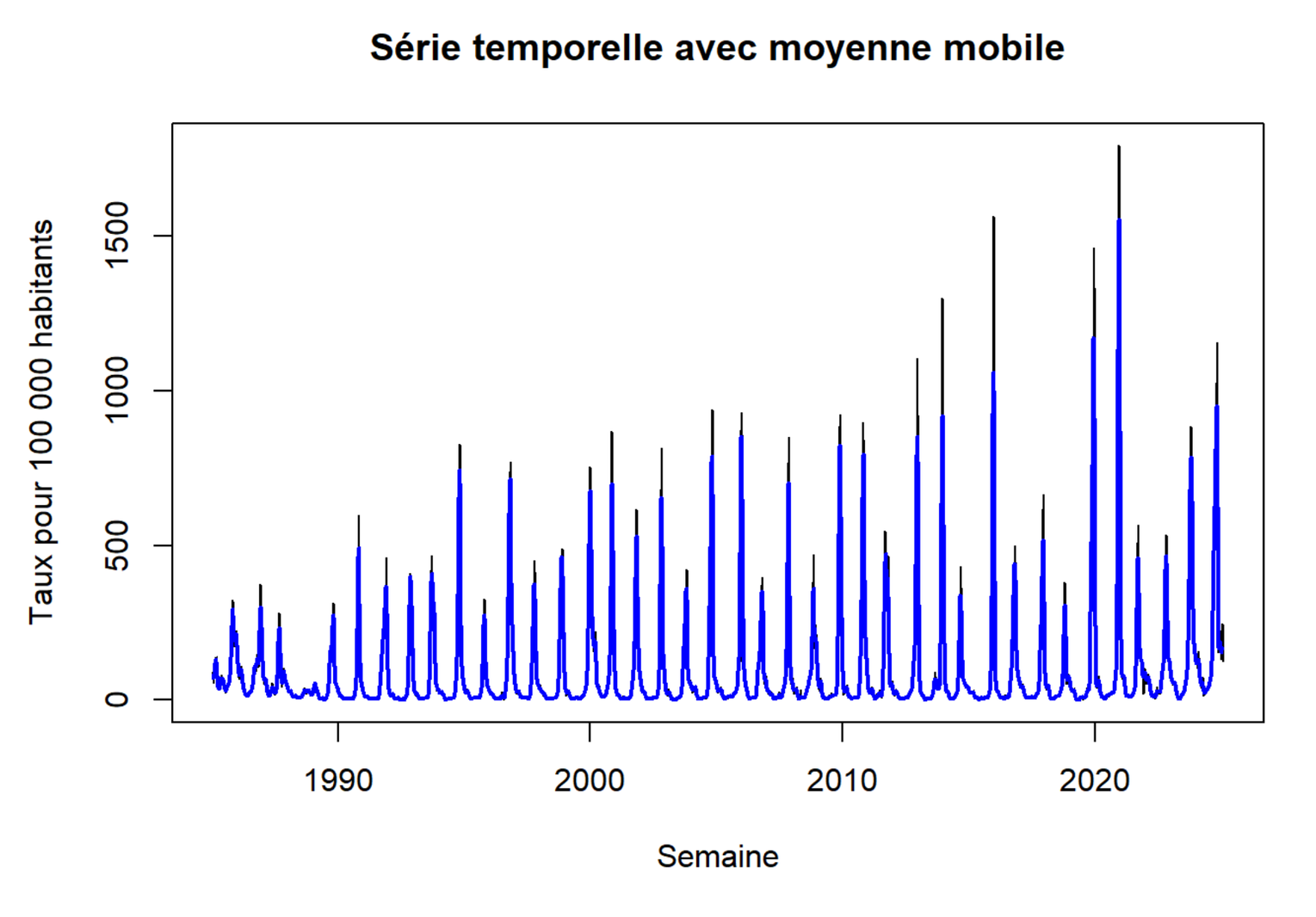



```
# Transformation logarithmique si nécessaire
log.inc100 <- log(inc100_ts)
plot(log.inc100, main="Logarithme du taux d'incidence des syndromes grippaux",
     xlab="Semaine", ylab="Log(Taux pour 100 000 habitants)")
```



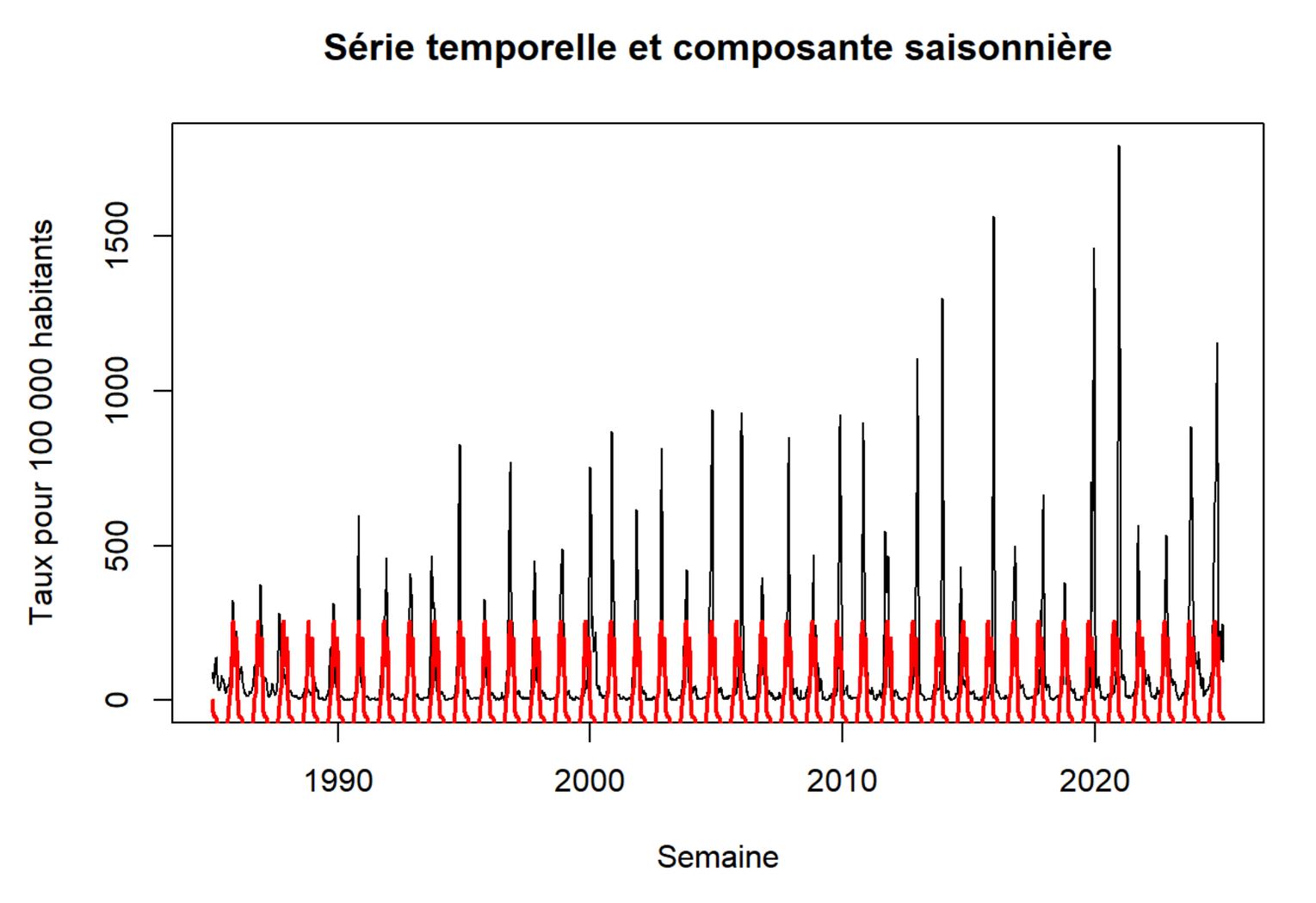
3. **Montrer la tendance de la série, en utilisant un filtre de moyennes mobiles.**

```
# Application d'une moyenne mobile pour lisser la série
# On applique une moyenne mobile sur 4 semaines pour lisser la série

inc100_ma <- stats::filter(inc100_ts, rep(1/4, 4), sides = 2)

# Affichage de la série originale et de la moyenne mobile
plot(inc100_ts, main="Série temporelle avec moyenne mobile",
     xlab="Semaine", ylab="Taux pour 100 000 habitants")
lines(inc100_ma, col="blue", lwd=2)  # Ajouter la moyenne mobile (ligne bleue)
```



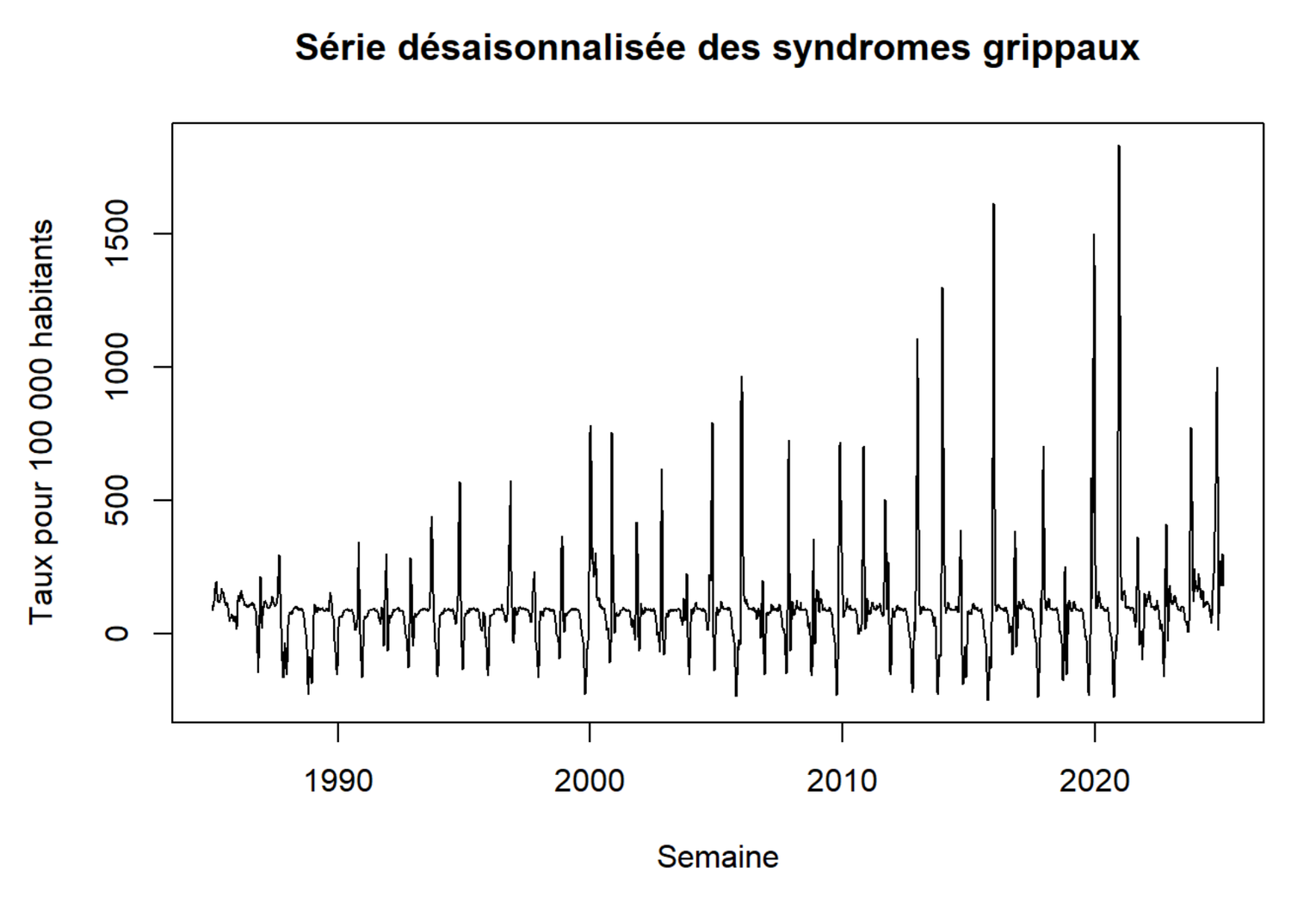
4. **Calculer et tracer les coefficients saisonniers ; fournir la décomposition de la série.**

```
# Décomposition de la série temporelle
dec <- decompose(inc100_ts)

# Composante saisonnière
hat.s <- dec$seasonal

# Tracé de la série originale avec la composante saisonnière
plot(inc100_ts, main="Série temporelle et composante saisonnière",
     xlab="Semaine", ylab="Taux pour 100 000 habitants")
lines(hat.s, col="red", lwd=2)  # Composante saisonnière en rouge
```



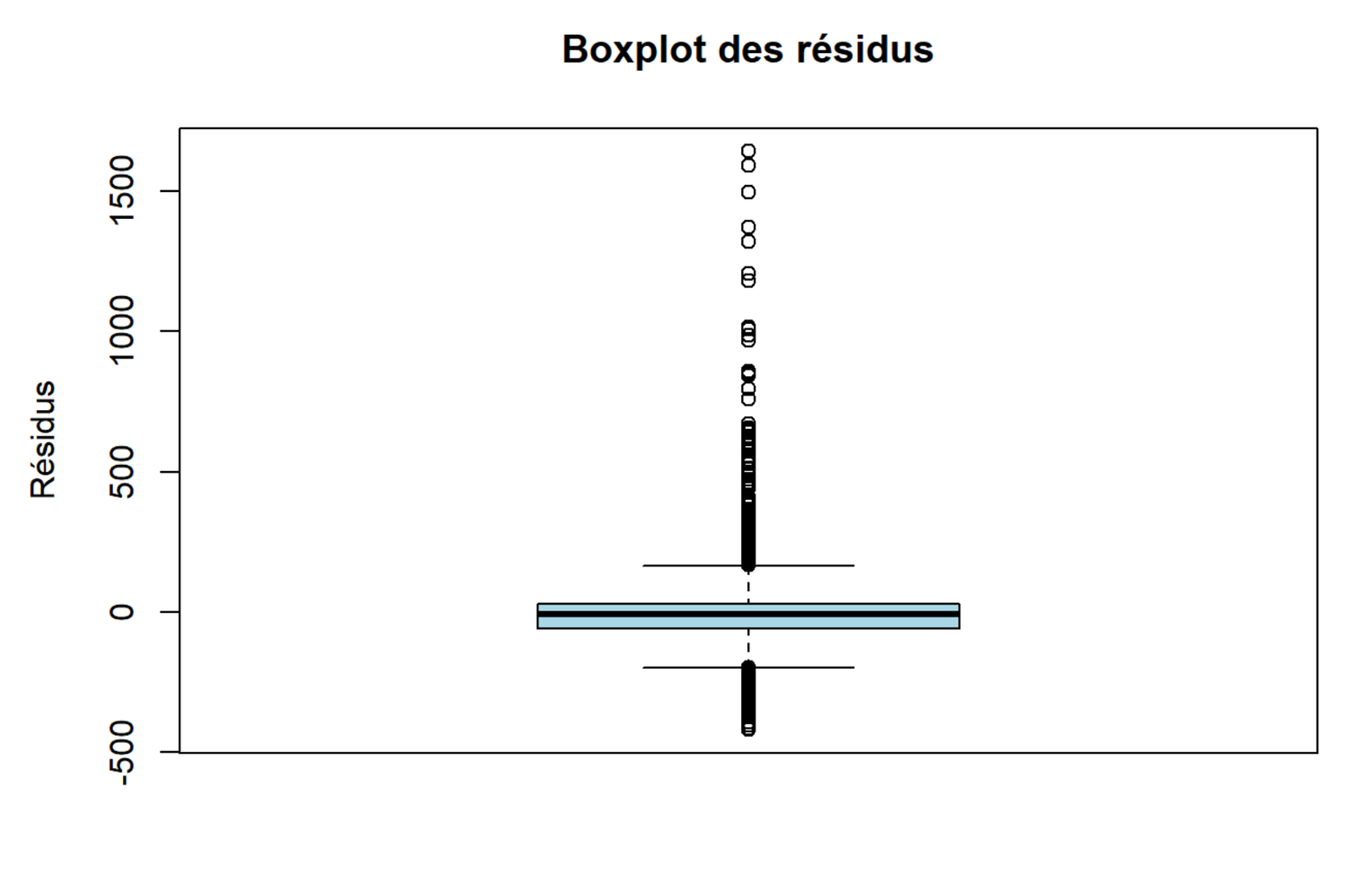
5. **Tracer et commenter la série désaisonnalisée (série corrigée des variations saisonnières).**

```
# Désaisonnalisation de la série en soustrayant la composante saisonnière
CVS <- inc100_ts - hat.s

# Tracé de la série désaisonnalisée
plot(CVS, main="Série désaisonnalisée des syndromes grippaux",
     xlab="Semaine", ylab="Taux pour 100 000 habitants")
```



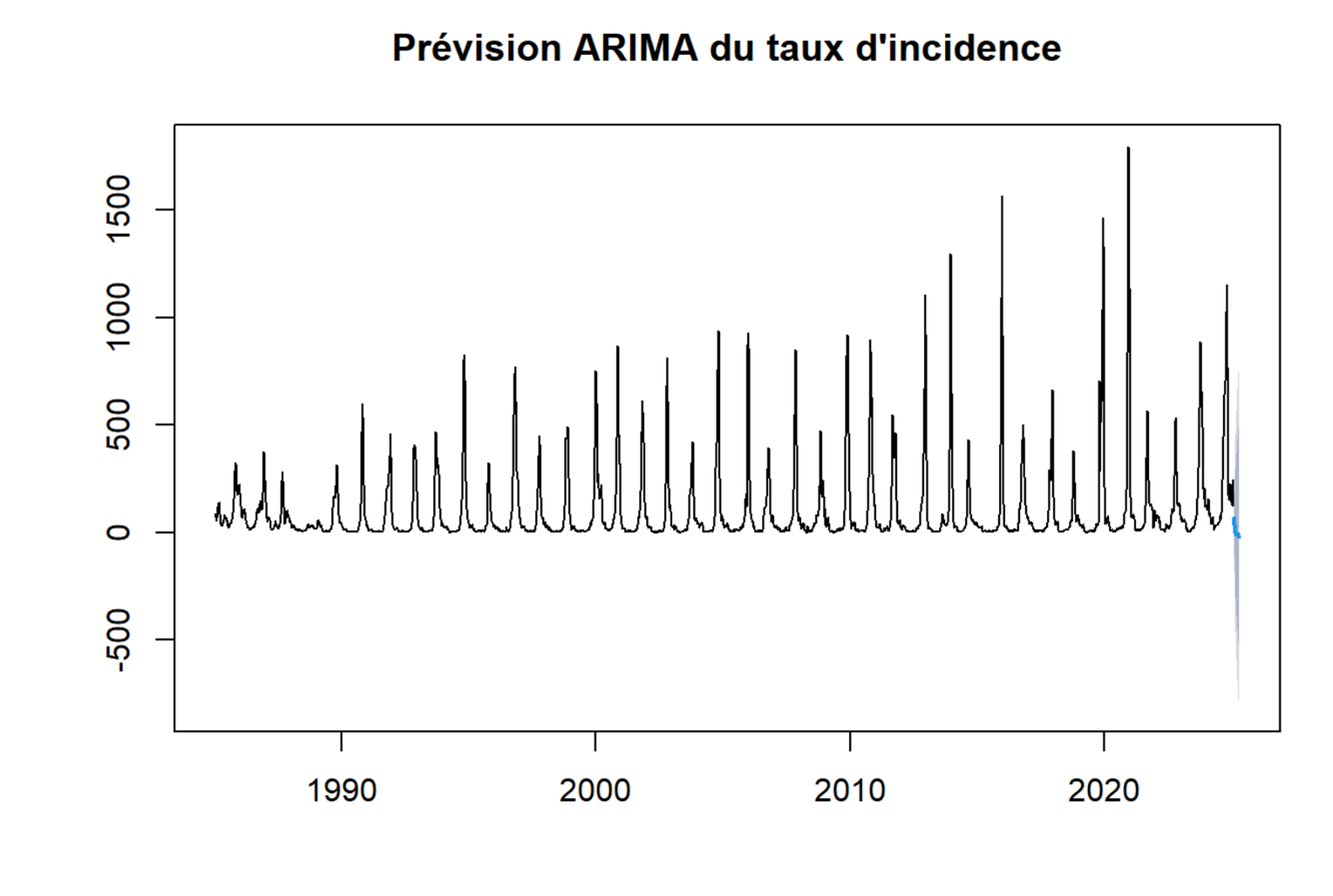
6. **Tracer et commenter les boxplots des résidus.**

```
# Calcul des résidus (différence entre la série originale et la tendance + saisonnalité)
residuals <- inc100_ts - (dec$trend + dec$seasonal)

# Tracé des boxplots des résidus
boxplot(residuals, main="Boxplot des résidus",
        ylab="Résidus", col="lightblue")
```



7. **Réaliser une prévision du taux d'incidence de l'indicateur pour le reste de cet automne et pour le prochain hiver en utilisant trois méthodes différentes.**

```
# Exemple avec ARIMA
fit_arima <- auto.arima(inc100_ts)
forecast_arima <- forecast(fit_arima, h=12)  # Prévision pour 12 semaines (automne et hiver prochains)

# Tracé de la prévision
plot(forecast_arima, main="Prévision ARIMA du taux d'incidence")
```



8. **Evaluer la qualité des prévisions obtenues avec les trois méthodes précédentes. Pour cela, pour chacune des méthodes utilisées dans la partie précédente, utiliser les données jusqu'en 2023 comme données d'entraînement et produire une prévision pour les 46 semaines de l'année 2024. Étant donné que les données des 46 semaines de 2024 (données de test) sont observées, calculer l'erreur quadratique moyenne des prévisions pour chacune des trois méthodes.**

```
# Diviser les données en ensemble d'entraînement (jusqu'en 2023) et de test (2024)
train_data <- window(inc100_ts, end=c(2023, 52))  # Données jusqu'en 2023
test_data <- window(inc100_ts, start=c(2024, 1))  # Données 2024

# Exemple avec ARIMA
fit_arima <- auto.arima(train_data)
forecast_arima <- forecast(fit_arima, h=length(test_data))

# Calcul de l'erreur quadratique moyenne (RMSE)
rmse_arima <- sqrt(mean((forecast_arima$mean - test_data)^2))
print(paste("RMSE pour ARIMA:", rmse_arima))
```

```
## [1] "RMSE pour ARIMA: 235.512291335808"
```