

Compte rendu

Séries temporelles

Pierre Monnin

19 mai 2025

Table des matières

1	Analyse descriptive	3
1.1	Faire un graphique des 100 premières observations de la série temporelle $(X_t)_{1 \leq t \leq 1000}$	3
1.2	Tracer la fonction d'autocorrélation de $(X_t)_{1 \leq t \leq 1000}$	4
2	Stationnarisation par estimation et modélisation	6
2.1	Calculer des estimations $\hat{m}(t)$ et $\hat{s}(t)$ de l'éventuelle tendance m_t et de l'éventuelle saisonnalité $s(t)$	6
2.2	Estimer un modèle ARMA(p, q) du processus $(Y_t)_{1 \leq t \leq 1000}$. On détaillera notamment la méthode utilisée pour choisir les ordres p et q de l'ARMA.	9
2.3	Estimer les performances de ce modèle sur les 100 prochaines observations $(X_t)_{1001 \leq t \leq 1100}$	11
3	Modélisation d'un SARIMA	13
3.1	Etudier le processus $(Z_t)_{1 \leq t \leq 1000}$ défini par : $(Z_t) := (\Delta^d X_t)_{d \leq t \leq 1000} = (I - B)^d X_t$ avec $d = 12$	13
3.2	Estimer un modèle SARIMA pour la série $(Z_t)_{1 \leq t \leq 1000}$	15
3.3	Estimer les performances de ce modèle sur les 100 prochaines observations $(X_t)_{1001 \leq t \leq 1100}$	17
4	Conclusions	19

1 Analyse descriptive

L'objectif de ce projet est d'étudier une série de donnée afin d'en générer des prédictions en modélisant à l'aide des séries temporelles.

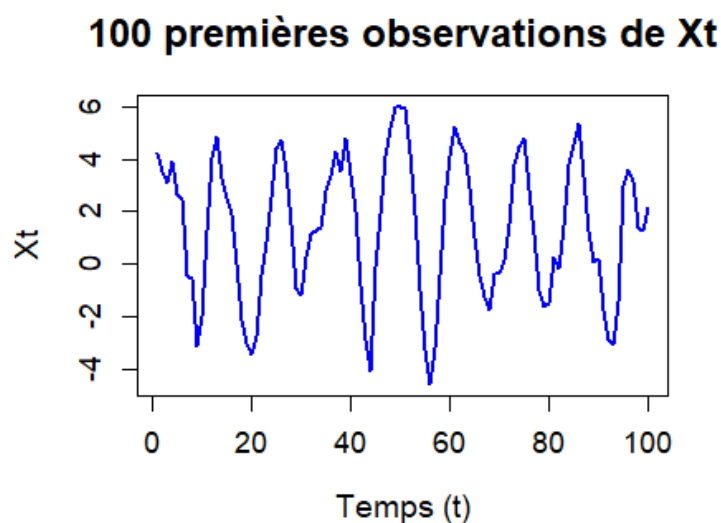
Ce projet a été réalisé sur R Studio, nous présenterons donc les différents codes utilisés dans ce compte rendu ainsi que leur sorties.

Notre logique sera détaillée et expliquée dans les commentaires de nos codes.

1.1 Faire un graphique des 100 premières observations de la série temporelle $(X_t)_{1 \leq t \leq 1000}$

Dans l'ensemble du code les `#test` sont des tests que l'on a généré afin de s'assurer du bon fonctionnement de notre code, nous n'afficherons pas les sorties liées à ces commandes.

```
1  # On importe nos données et on initialise notre série :
2  data <- read.table("chemin/du/fichier/serie", header = FALSE)
3  serie <- data[, 2]
4  #test :
5  head(serie)
6  length(serie)
7  #Notre série comporte 1000 observations
8
9  # On trace la courbe des 100 première observations afin d'avoir un premier aperçu du
10 #comportement de notre série et avoir une première idée sur sa stationnarité
11
12 plot(serie[1:100], type = "l",
13       main = "100 premières observations de Xt",
14       xlab = "Temps (t)", ylab = "Xt",
15       col = "blue",
16       lwd = 2,
17       cex.main = 1.5,
18       cex.lab = 1.2,
19       cex.axis = 1.1)
```



On peut directement émettre l'hypothèse de l'existence d'une saisonnalité au vu de la périodicité de notre graphique.

La présence d'une tendance bien que beaucoup moins flagrante pourrait s'illustrer par une légère

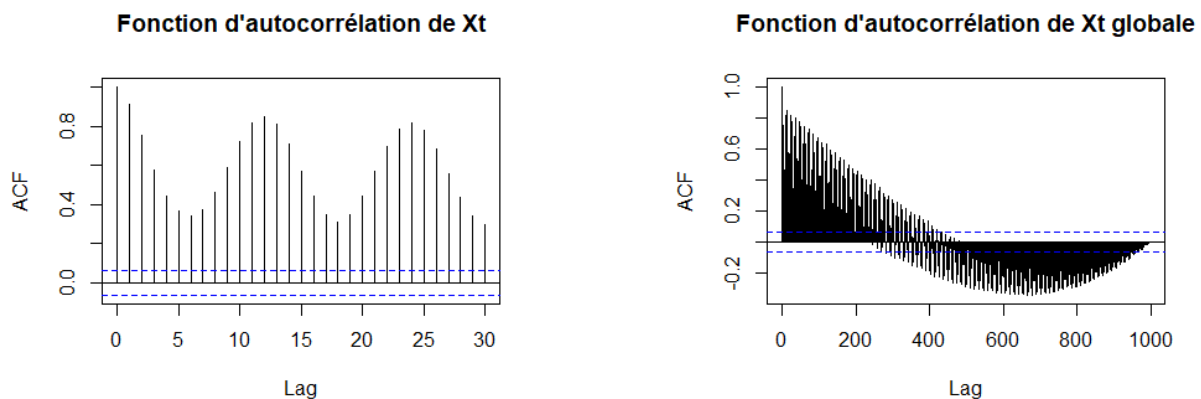
hausse de notre courbe dans le temps.

Notre série n'est donc sûrement pas stationnaire.

Nous allons valider cette hypothèse avec les résultats de l'étape suivante :

1.2 Tracer la fonction d'autocorrélation de $(X_t)_{1 \leq t \leq 1000}$

```
1 # On trace l'acf de notre série:
2 acf(serie[1:1000], main = "Fonction d'autocorrélation de Xt") #zoom sur le début
3 acf(serie[1:1000], lag.max=1000, main = "Fonction d'autocorrélation de Xt globale")
4 #vision globale
```



On observe immédiatement de part l'amplitude de nos barres (en dehors de l'intervalle de confiance situé entre les lignes de pointillés bleus) que notre série n'est pas stationnaire, sinon dès les premiers lag celle-ci aurait alterné autour de 0.

De plus les motifs répétés tout les 12 lags viennent nous confirmer une saisonnalité de période 12.

On peut confirmer nos résultats avec différents tests :

```
1 #test adf
2
3 #H0: la série est non stationnaire
4
5 adf.test(serie)
6
7
8 #p value < 0.01, on devrait donc rejeter H0
9 #donc confirmer la stationnarité,
10 #or, nos graphiques nous disant l'inverse, on va réaliser un autre test
11
12 #Test KPSS afin de trancher, plus stable en cas de saisonnalité
13
14 library(urca)
15
16 #H0: La série est stationnaire
17
18 kpss_test <- ur.kpss(ts(serie), type = "mu") # type = "mu" pour drift ; "tau" pour tendance
19 summary(kpss_test)
20 #Notre statistique de test est supérieur à toutes les valeurs critiques du test,
21 #donc notre p value est en dessous du seuil critique et on rejette l'hypothèse de stationnarité
22 #on devrait donc accepter la non stationnarité
```

```

#####
# KPSS Unit Root Test #
#####

Test is of type: mu with 7 lags.

value of test-statistic is: 11.2085

Critical value for a significance level of:
      10pct  5pct 2.5pct  1pct
critical values 0.347 0.463 0.574 0.739

Augmented Dickey-Fuller Test

data: serie
Dickey-Fuller = -11.504, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

Message d'avis :
Dans adf.test(serie) : p-value smaller than printed p-value

```

Ici, on a un conflit au niveau de nos tests statistiques pouvant être lié à plusieurs raisons. Mais nos graphiques vont dans le sens de la non stationnarité et le résultat d'un test sur deux la confirme

On peut donc conclure de l'ensemble de nos observations que notre série n'est pas stationnaire.

2 Stationnarisation par estimation et modélisation

On sait à l'avance que :

- S'il y a une saisonnalité, celle-ci est déterministe en fonction du temps t : $s(t)$, et de période $d = 12$, donc $s(t + 12) = s(t)$.
- De plus, si cette série a une tendance, celle-ci est aussi déterministe en fonction de t : $m(t)$, et elle est forcément linéaire : $m(t) = a + bt$.

2.1 Calculer des estimations $\hat{m}(t)$ et $\hat{s}(t)$ de l'éventuelle tendance m_t et de l'éventuelle saisonnalité $s(t)$

On commence par estimer l'éventuelle tendance de notre série.

On rappelle que d'après nos observations nous nous attendons à une tendance faible et positive.

```
1  #De part la forme de notre tendance (affine) et d'après le cours,
2  #on sait qu'on peut l'estimer par régression linéaire par moindres carrés:
3
4  t <- 1:length(serie)
5  tendance_model <- lm(serie ~ t)
6  summary(trend_model)
7  m_chapeau <- tendance_model$fitted.values
8
9  #test :
10
11 plot(m_chapeau)
12
13 #m_chapeau(t)=0.01t+0.82
14 # donc tendance à la hausse très légère (coeff directeur de 0.01),
15 #cohérent avec nos précédentes observations
16 # le summary de notre régression nous indique que nos coefficients et notre estimation globale
17 #sont hautement significatives au vu de l'ordre de grandeur très faible de leur p-value
18
19 serie_détendialisée<- serie - m_chapeau
20
21 #test :
22
23 head(serie_détendialisée)
24 length(serie_détendialisée)
25
26 #visualisation du résultat:
27
28 plot(serie_détendialisée[1:100], type = "l",
29      main = "100 premières observations de notre série sans tendance",
30      xlab = "Temps (t)", ylab = "Xt",
31      col = "red",
32      lwd = 2,
33      cex.main = 1.5,
34      cex.lab = 1.2,
35      cex.axis = 1.1)
36
37 acf(serie_détendialisée[1:1000], main = "Fonction d'autocorrélation de série sans tendance")
38 acf(serie_détendialisée[1:1000], lag.max=1000, main = "Fonction d'autocorrélation de série sans tendance globale")
```

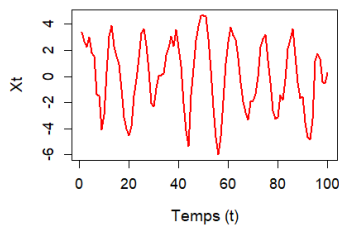
D'après nos graphiques, notre série n'est toujours pas stationnaire et l'éventuelle saisonnalité observée se dégage encore plus clairement.

On observe aussi que nos cycles périodiques sont plus stables dans le temps ce qui est un bon signe du retranchement de notre tendance et donc d'une bonne estimation de celle-ci.

Le résultat de notre estimation était donc bien en cohérence avec nos premières observations.

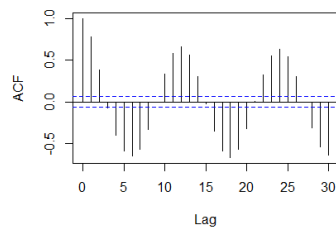
Désormais nous allons estimer notre saisonnalité qui devrait être de période 12 d'après les infor-

100 premières observations de notre série sans



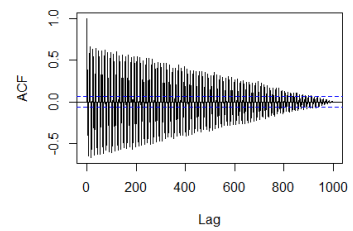
(a) 100 premières observations de notre série sans tendance

Fonction d'autocorrélation de série sans tendan



(b) Fonction d'autocorrélation de série sans tendance

Fonction d'autocorrélation de série sans tendance



(c) Fonction d'autocorrélation de série sans tendance globale

mations données ainsi que nos observations

Pour cela, on va construire $\hat{s}(t)$ en prenant pour estimateur un vecteur de 12 valeurs associant à chacune la moyenne résiduelle de nos observations sur les périodes 1 modulo 12 pour la première, 2 modulo 12 pour la deuxième etc

```
1 periode <- 12
2 seasonal_effect <- tapply(serie_détendialisé, (t - 1) %% periode + 1, mean)
3 s_chapeau <- rep(seasonal_effect, length.out = length(serie))
4
5 #test:
6 print(seasonal_effect)
7 #On affiche désormais notre série Y
8
9 Y<- serie_détendialisé - s_chapeau
10 plot(Y[1:100], type = "l",
11       main = "100 premières observations de notre série sans tendance et sans saison",
12       xlab = "Temps (t)", ylab = "Xt",
13       col = "green",
14       lwd = 2,
15       cex.main = 1.5,
16       cex.lab = 1.2,
17       cex.axis = 1.1)
18
```

100 premières observations de notre série sans tendan

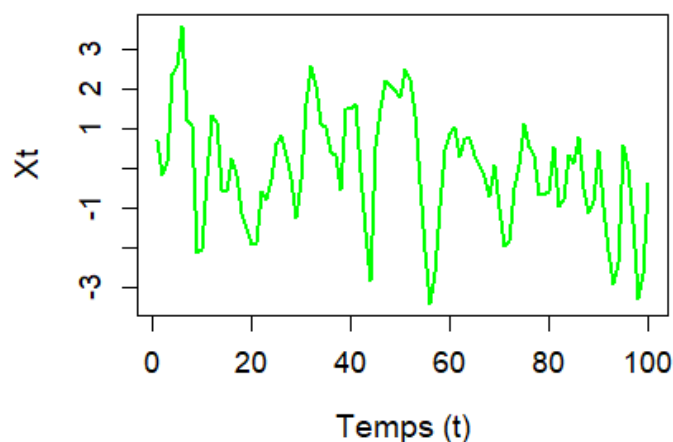
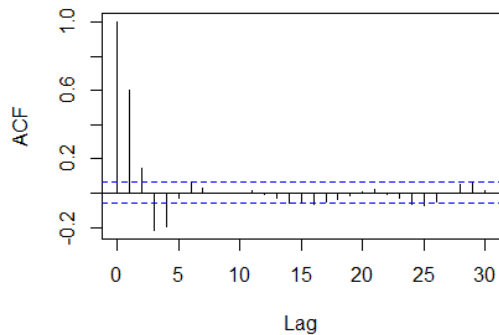


FIGURE 4 – 100 premières observations de notre série sans tendance et sans saison

On observe ici que nos valeurs fluctuent aléatoirement dans un intervalle beaucoup plus restreint $([-3;3])$ signe de stationnarité

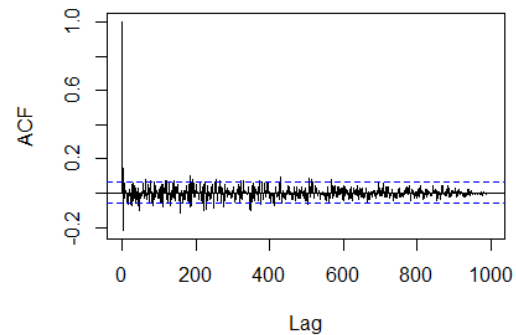
```
1 acf(Y[1:1000], main = "Fonction d'autocorrélation de série sans tendance et sans saison")
2 acf(Y[1:1000], lag.max=1000, main = "Fonction d'autocorrélation de série sans tendance et sans saison globale")
```

on d'autocorrélation de série sans tendance et s



(a) Fonction d'autocorrélation de série sans tendance et sans saison

'autocorrélation de série sans tendance et sans



(b) Fonction d'autocorrélation de série sans tendance et sans saison globale

On observe bien sur nos deux graphes que dès le 3ème lag nos barres sont toutes ou presque dans l'intervalle de confiance

On en conclue que nos estimations ont été bonnes et que notre série Y_t est stationnaire

On va tout de même vérifier nos observations avec les deux mêmes tests :

```
1 adf.test(Y)
2
3 #p value extrêmement petite donc < 0.05
4 #d'après le test adf on rejette H0
5 #la série est stationnaire
6
7 kpss_test <- ur.kpss(ts(Y), type = "mu") # type = "mu" pour drift ; "tau" pour tendance
8 summary(kpss_test)
9 #Ici notre statistique de test est inférieur à toute nos valeurs critiques,
10 #donc notre p value est supérieur au seuil critique,
11 #on ne rejette pas l'hypothèse de stationnarité,
12 #la non stationnarité est donc peu probable
13
```

```
#####
# KPSS Unit Root Test #
#####

Test is of type: mu with 7 lags.

value of test-statistic is: 0.0207

critical value for a significance level of:
      10pct  5pct  2.5pct  1pct
critical values 0.347 0.463 0.574 0.739

Augmented Dickey-Fuller Test

data: Y
Dickey-Fuller = -10.129, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

Message d'avis :
Dans adf.test(Y) : p-value smaller than printed p-value
```

On peut donc affirmer de part l'ensemble de nos observations ainsi que par les résultats de nos tests statistiques que notre série est stationnaire.

Notre série Y_t est donc un bruit que l'on va désormais pouvoir estimer avec la méthodologie ci-dessous.

2.2 Estimer un modèle $\text{ARMA}(p, q)$ du processus $(Y_t)_{1 \leq t \leq 1000}$.

On détaillera notamment la méthode utilisée pour choisir les ordres p et q de l'ARMA.

Dans un premier temps nous allons tenter de déterminer les coefficients optimaux de notre processus ARMA.

Pour cela, nous allons tracer l'ACF et le PACF de notre série Y afin de vérifier visuellement la cohérence des résultats qui nous seront données par la fonction R, `auto.arima`.

Cette fonction nous donne les meilleurs coefficients de notre processus ARMA en fonction du critère choisi. Ici nous ferons en fonction du BIC puis du AIC.

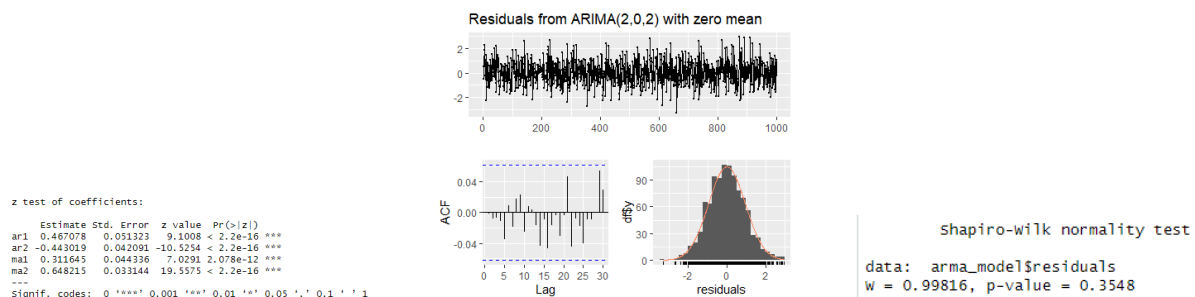
```
1 library(forecast)
2 pacf(Y[1:1000], main = "Fonction d'autocorrélation partielle de série sans tendance et sans saison")
3 # besoin juste de zoom car ce sont seulement les premières valeurs qui nous intéressent
4 acf(Y[1:1000], main = "Fonction d'autocorrélation de série sans tendance et sans saison")
5 #En fonction du critère aic:
6
7 arma_model <- auto.arima(Y, seasonal = FALSE, ic = "aic")
8 summary(arma_model)
9
10 #Le meilleur modèle selon le critère AIC est un ARMA(2;2)
11
12 #En fonction du critère bic:
13
14 arma_model <- auto.arima(Y, seasonal = FALSE, ic = "bic")
15 summary(arma_model)
```

Le calcul des paramètres par maximisation de la log vraisemblance n'est ici pas utile car le critère BIC et AIC représente justement un score illustrant un équilibre au niveau du compromis maximum de vraisemblance et complexité du modèle.

On peut donc conclure que notre modèle optimal pour estimer notre processus Y_t est un $\text{ARMA}(2;2)$

Avant de passer aux prédictions nous allons tout de même tester la validité de notre modèle en testant la significativité des coefficients, l'indépendance et la normalité des résidus

```
1 #Significativité des coefficients:
2
3 library(lmtest)
4 coeftest(arma_model)
```



Au vu des p-values nos coefficients sont bien tous significatifs (< 0.05 , résultat du test)

```
1 #Indépendance et normalité des résidus:
2 checkresiduals(arma_model)
3 shapiro.test(arma_model$residuals)
```

Comme vu précédemment nos résidus ont bien une acf comprise dans l'intervalle de confiance, signe d'indépendance des résidus

L'histogramme et les graphiques des résidus nous montre bien une répartition centré en 0 symétrique, sans structure défini et normale.

Les sorties du test de Ljung-Box nous donne qu'on ne rejette pas l'hypothèse de non corrélation jusqu'au lag 10 ($p\text{-values} > 0.05$)

On peut déduire du test de Ljung-Box ainsi que de l'allure du graphique et de l'histogramme des résidus que nos résidus sont indépendants

Les sorties du test de Shapiro Wilk nous donne qu'on ne rejette pas l'hypothèse de normalité des résidus ($p\text{-values} > 0.05$)

On peut déduire du test de shapiro wilk ainsi que de la répartition de nos résidus que ceci sont gaussiens

Notre modèle valide donc l'ensemble des hypothèses, on peut passer à l'étape de prédiction

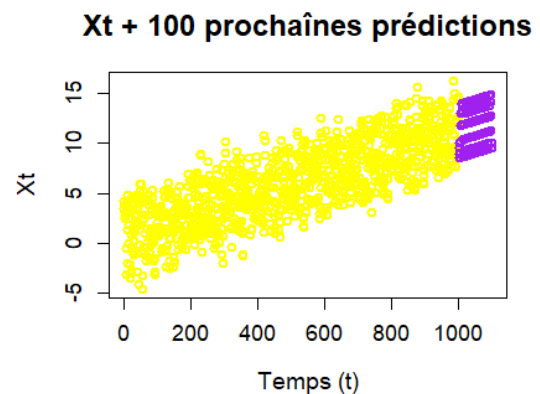
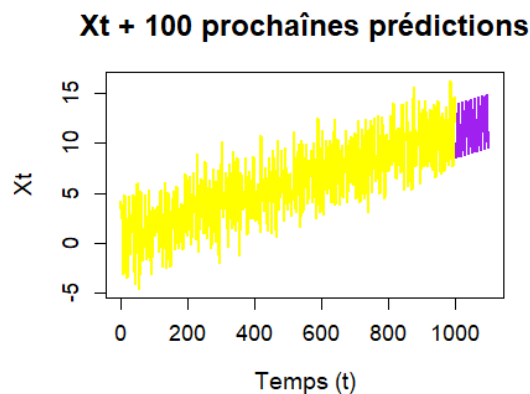
2.3 Estimer les performances de ce modèle sur les 100 prochaines observations $(X_t)_{1001 \leq t \leq 1100}$.

```
1
2 forecast_arma <- forecast(arma_model, h = 100)
3 print(forecast_arma)
4 Y_pred<- forecast_arma$mean
5
6 #Test
7
8 print(Y_pred)
9
10 #Prédiction des 100 prochaines observations de notre tendance:
11
12 h <- (length(serie) + 1):(length(serie) + 100)
13 m_prédiction <- tendance_model$coefficients[1] + tendance_model$coefficients[2] * h
14
15
16 #Test
17
18 plot(m_prédiction)
19
20 #Prédiction des 100 prochaines observation de notre saisonnalité:
21
22 s_prédiction <- rep(seasonal_effect, length.out = length(serie) + 100)[(length(serie) +1):(length(serie) +100)]
23
24 #Test
25
26 head(s_prédiction,12)
27 tail(s_chapeau,12)
28
29 #Prédiction des 100 prochaines observations de notre série temporel:
30
31 X_pred <- as.numeric(Y_pred) + m_prédiction + s_prédiction
32
33 #Test :
34
35 plot(X_pred,
36       main = "100 prochaines observations prédites zoom",
37       xlab = "Temps (t)", ylab = "Xt",
38       col = "purple",
39       lwd = 2,
40       cex.main = 1.5,
41       cex.lab = 1.2,
42       cex.axis = 1.1)
43
44 plot(X_pred, type = "l",
45       main = "100 prochaines observations prédites zoom",
46       xlab = "Temps (t)", ylab = "Xt",
47       col = "purple",
48       lwd = 2,
49       cex.main = 1.5,
50       cex.lab = 1.2,
51       cex.axis = 1.1)
52
53
54 #Visualisation:
55
56 #1:
57
58 Xt_full <- c(serie, X_pred)
59 temps <- 1:1100
60 plot(temps, Xt_full, type = "n",
61       main = "Xt + 100 prochaines prédictions",
62       xlab = "Temps (t)", ylab = "Xt",
63       cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)
64 lines(1:1000, serie, col = "yellow", lwd = 2)
65 lines(1001:1100, X_pred, col = "purple", lwd = 2)
66
67
```

```

1 #2:
2
3 plot(temps, Xt_full, type = "n",
4       main = "Xt + 100 prochaines prédictions",
5       xlab = "Temps (t)", ylab = "Xt",
6       cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)
7 points(1:1000, serie, col = "yellow", lwd = 2)
8 points(1001:1100, X_pred, col = "purple", lwd = 2)
9
10 #Préparation de l'export:
11
12 indices <- 1001:1100
13 output1 <- data.frame(index = indices, prediction = X_pred)
14
15 #Check de l'output:
16
17 print(output1)
18 write.table(output1, file = "chemin/du/fichier/serie", row.names = FALSE, col.names = FALSE, quote = FALSE)

```



Avant d'exporter nos prédictions nous avons tout de même visualiser les projections superposées à nos données d'origine.

Nos prédictions suivent bien la tendance globale de nos 1 000 premières observations

Nos résultats sont cohérents on peut donc les exporter

3 Modélisation d'un SARIMA

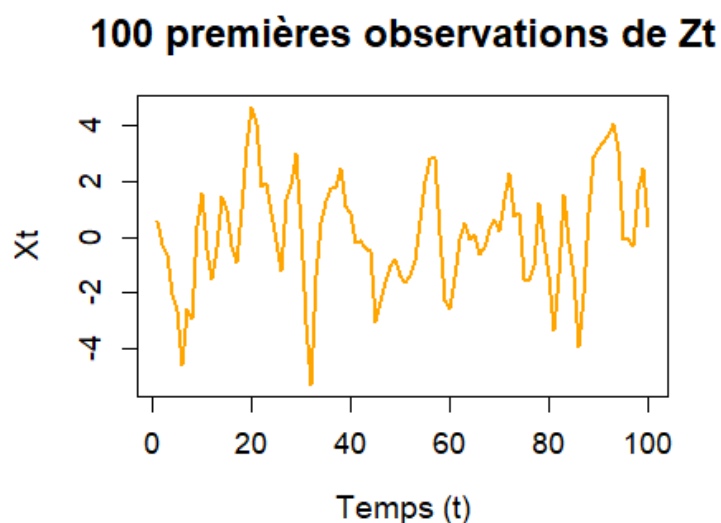
On veut désormais modéliser notre série temporelle à l'aide d'un processus SARIMA. Dans le cadre d'une mission de modélisation, il est essentiel de tester plusieurs modèles afin d'avoir une vue plus large sur les différentes productions possibles ainsi que sur le meilleur modèle à adopter. Le meilleur modèle dans un contexte ne le sera pas forcément dans un autre et parfois même avec les mêmes données, il est donc important de tester plusieurs modèles puis de choisir en fonction de la demande.

Ici la demande de prédiction est purement théorique et statistique donc nous baserons notre choix sur des tests statistiques et théoriques.

3.1 Etudier le processus $(Z_t)_{1 \leq t \leq 1000}$ défini par :

$$(Z_t) := (\Delta^d X_t)_{d \leq t \leq 1000} = (I - B)^d X_t \quad \text{avec } d = 12.$$

```
1 d <- 12
2 Z <- serie[(d+1):length(serie)] - serie[1:(length(serie) - d)]
3
4 #check:
5
6 head(Z)
7 length(Z)
8
9 #Etape a)
10
11 #On trace les 100 premières observations de notre série Z
12
13 plot(Z[1:100], type = "l",
14      main = "100 premières observations de Zt",
15      xlab = "Temps (t)", ylab = "Xt",
16      col = "orange",
17      lwd = 2,
18      cex.main = 1.5,
19      cex.lab = 1.2,
20      cex.axis = 1.1)
```

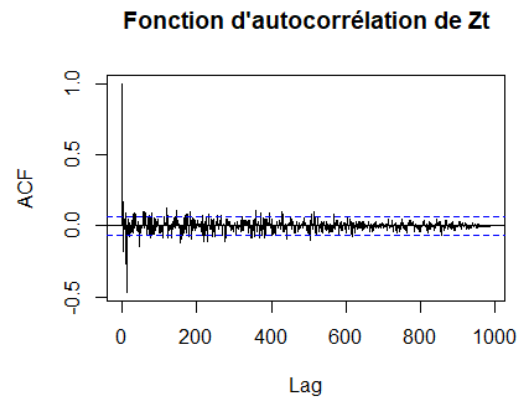
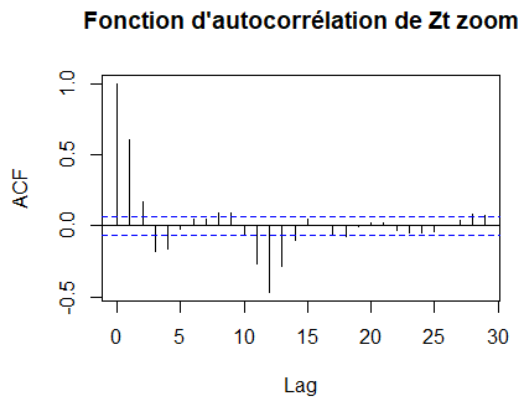


On observe sur le graphique que notre série Z_t se comporte comme une série stationnaire (varie de manière aléatoire, sans structure entre 4 et - 4)

```

1 #Etape b)
2
3 acf(Z[1:988], main = "Fonction d'autocorrélation de Zt zoom")
4 acf(Z[1:988],lag.max=1000, main = "Fonction d'autocorrélation de Zt")
5
6

```



On voit donc clairement que l'ACF chute et rentre dans la zone de confiance, on observe aussi un rebond au 12^{ème} lag (périodicité enlevé)

Notre série semble bien être stationnaire, bien que cela soit un peu moins évident que pour les résultats précédent de part quelques lag qui sortent de la zone de confiance

On peut confirmer nos résultats avec plusieurs tests statistiques :

```

1 adf.test(Z)
2
3 #La p-value étant< 0.01,
4 #on rejette l'hypothèse de non stationnarité,
5 #notre série devrait donc être stationnaire
6
7 #On peut confirmer nos résultats avec un test de Kpss:
8
9 kpss_test <- ur.kpss(ts(Z), type = "mu") # type = "mu" pour drift ; "tau" pour tendance
10 summary(kpss_test)
11
12 #Notre statistique de test est inférieure à toutes nos valeurs critiques,
13 #la p-value est donc > au seuil critique,
14 #on ne rejette donc pas l'hypothèse de stationnarité,
15 #La non stationnarité est donc peu probable

```

```

#####
# KPSS Unit Root Test #
#####

Augmented Dickey-Fuller Test

data: Z
Dickey-Fuller = -12, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

Message d'avis :
Dans adf.test(Z) : p-value smaller than printed p-value

Test is of type: mu with 7 lags.

value of test-statistic is: 0.0083

Critical value for a significance level of:
      10pct  5pct 2.5pct 1pct
critical values 0.347 0.463 0.574 0.739

```

L'ensemble de nos conclusions nous laisse à penser que notre série est bien stationnaire.

On considère donc que Z_t est un bruit, on va désormais pouvoir l'estimer à l'aide d'un processus SARIMA.

3.2 Estimer un modèle SARIMA pour la série $(Z_t)_{1 \leq t \leq 1000}$.

Nous allons procéder exactement de la même manière que pour notre processus ARIMA :

```
1  acf(Z[1:988], main = "Fonction d'autocorrélation de Zt")
2  pacf(Z[1:988], main = "Fonction d'autocorrélation partielle de Zt")
3
4  #Fonction d'auto corrélation partiel chute après 2 lag => p=2
5  #Fonction d'auto corrélation partiel chute après 2 lag => q=2
6  #d=0 car série Zt stationnaire
7  #D=1 car saisonnalité présente ( de périodicité 12 d'ou le modulo en fin)
8  #pas de pic a lag 12 sur fonction d'auto corrélation partiel => P=0
9  #pic à lag 12 sur fonction d'auto corrélation => Q=1
10 #SARIMA(2,0,2)(0,1,1)[12] optimal graphiquement,
11 #vérifions à l'aide de la fonction auto.arima avec saisonnalité activé
12 #Nous l'appliquerons selon le critère BIC puis selon le critère AIC
13 serie_ts <- ts(serie, frequency = 12)
14 sarima_model <- auto.arima(serie_ts,
15                             seasonal = TRUE,
16                             ic="aic",
17                             D = 1,           # car stationnaire au bout d'un écart
18                             d = 0,           # car tendance inexistante ou trop faible
19                             max.P = 2, max.Q = 2, # standard 2 suffisent, à ajuster éventuellement ensuite
20                             max.p = 5, max.q = 5, # pareil
21                             stepwise = FALSE,
22                             approximation = FALSE,
23                             trace = TRUE)
24
25 # Best model: SARIMA(2,0,2)(0,1,1)[12]
26 #On reteste aussi avec BIC:
27 sarima_model <- auto.arima(serie_ts,
28                             ic="bic",
29                             seasonal = TRUE,
30                             D = 1,           # car stationnaire au bout d'un écart
31                             d = 0,           # car tendance inexistante ou trop faible
32                             max.P = 2, max.Q = 2, # standard 2 suffisent, à ajuster éventuellement ensuite
33                             max.p = 5, max.q = 5, # pareil
34                             stepwise = FALSE,
35                             approximation = FALSE,
36                             trace = TRUE)
37
38 #Best model: SARIMA(2,0,2)(0,1,1)[12]
```

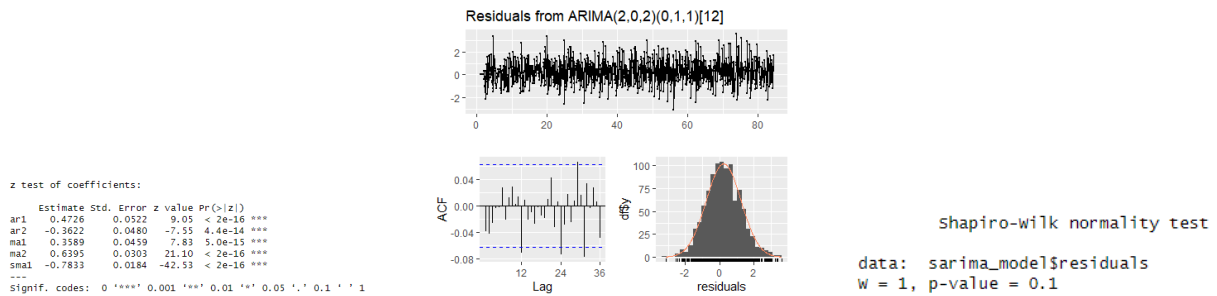
Là aussi le calcul des paramètres par la log vraisemblance n'est pas utile pour les mêmes raisons qu'expliquées plus haut

Avant de passer aux prédictions nous allons tout de même tester la validité de notre modèle en vérifiant les mêmes hypothèses que pour notre ARMA :

```

1 #Significativité des coefficients:
2
3 library (lmtest)
4 coeftest(sarima_model)
5
6 #Au vu des p-values nos coefficients sont biens tous significatifs
7
8 #Indépendance et normalité des résidus:
9
10 checkresiduals(sarima_model)
11 shapiro.test(sarima_model$residuals)

```



Comme vu précédemment nos résidus ont bien une acf comprise dans l'intervalle de confiance malgré des piques tout les 12 lags

L'histogramme et les graphiques des résidus nous montre bien une répartition centré en 0 symétrique, sans structure défini et normale

Les sorties du test de Ljung-Box nous donne qu'on ne rejette pas l'hypothèse de non corrélation jusqu'au lag 10 (p-values>0.05)

On peut déduire du test de Ljung-Box ainsi que de l'allure du graphique et de l'histogramme des résidus que nos résidus sont indépendants

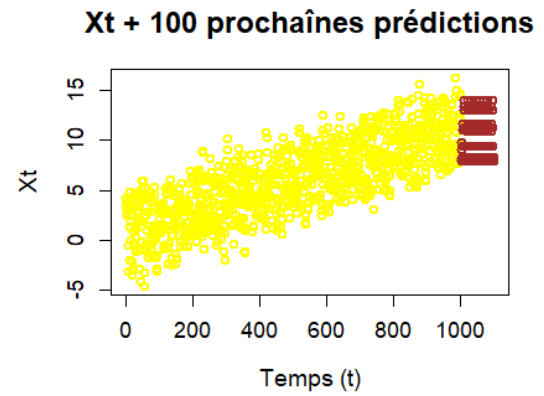
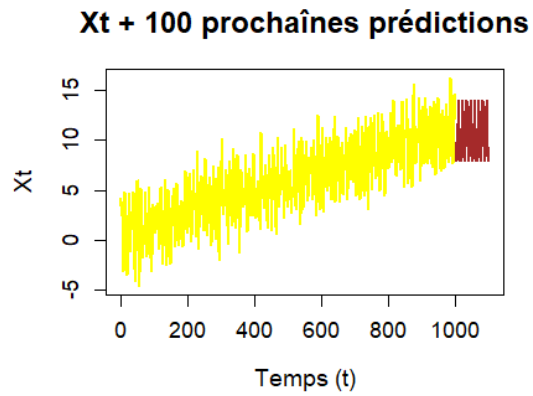
Les sorties du test de Shapiro Wilk nous donne qu'on ne rejette pas l'hypothèse de normalité des résidus (p-values>0.05)

On peut déduire du test de shapiro wilk ainsi que de la répartition de nos résidus que ceci sont gaussiens

Notre modèle valide donc l'ensemble des hypothèses, on peut passer à l'étape de prédiction

3.3 Estimer les performances de ce modèle sur les 100 prochaines observations $(X_t)_{1001 \leq t \leq 1100}$

```
1 forecast_sarima <- forecast(sarima_model, h = 100)
2 X_pred<- as.numeric(forecast_sarima$mean)
3
4 # Affiche les prédictions
5
6 head(X_pred)
7 length(X_pred)
8
9 # Test
10 plot(X_pred, type = "l",
11       main = "100 premières prédiction de Xt par modèle SARIMA",
12       xlab = "Temps (t)", ylab = "Xt",
13       col = "brown",
14       lwd = 2,
15       cex.main = 1.5,
16       cex.lab = 1.2,
17       cex.axis = 1.1)
18
19 plot(X_pred,
20       main = "100 premières prédiction de Xt par modèle SARIMA ZOOM",
21       xlab = "Temps (t)", ylab = "Xt",
22       col = "brown",
23       lwd = 2,
24       cex.main = 1.5,
25       cex.lab = 1.2,
26       cex.axis = 1.1)
27
28 #Comparé à notre première prédiction on notera l'absence de tendance croissante (logique)
29
30 #Visualisation:
31
32 #1:
33
34 Xt_full <- c(serie, X_pred)
35 temps <- 1:1100
36 plot(temps, Xt_full, type = "n",
37       main = "Xt + 100 prochaines prédictions",
38       xlab = "Temps (t)", ylab = "Xt",
39       cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)
40 lines(1:1000, serie, col = "yellow", lwd = 2)
41 lines(1001:1100, X_pred, col = "brown", lwd = 2)
42
43
44 #2:
45
46 plot(temps, Xt_full, type = "n",
47       main = "Xt + 100 prochaines prédictions",
48       xlab = "Temps (t)", ylab = "Xt",
49       cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)
50 points(1:1000, serie, col = "yellow", lwd = 2)
51 points(1001:1100, X_pred, col = "brown", lwd = 2)
52
```



On ne retrouve pas la tendance croissante de nos précédents résultats
Néanmoins, cela s'explique par la construction de notre modèle de prédiction, en effet, la différenciation saisonnière vient supprimer la tendance qui n'est désormais plus prise en compte, les résultats sont donc cohérents, on peut donc exporter les résultats

4 Conclusions

Notre modèle ARMA semble donc meilleur que notre modèle SARIMA et ceux pour plusieurs raisons.

Premièrement, nos sorties graphiques nous montre que nos prédictions générés par notre modèle ARMA suivent beaucoup mieux la tendance global de nos séries temporelles.

Cette écart s'explique par la non prise en compte de la légère mais existante tendance positive de notre série temporelle.

En effet par construction, la différentiation saisonnière qu'on applique en R vient bien désaisonnalisé notre série, néanmoins, notre tendance est ignoré, ce qui explique l'allure périodique mais constante de nos prédictions générés par le SARIMA au lieu de prédictions périodique légèrement croissante comme avec notre ARMA.

De plus, les hypothèses sur les résidus générées au moment de valider le modèle était plus forte pour le modèle ARMA que pour le SARIMA, on en déduit donc que notre modèle ARMA est plus fiable.

Sur l'hypothèse de significativité des coefficients, celle-ci est presque identique pour les 2, pour les deux tous les coefficients sont hautement significatifs.

Concernant l'indépendance des résidus, les pics de l'ACF du processus SARIMA sont partiellement plus hauts, de plus en refaisant un test de Ljung-Box sur nos deux processus que l'on paramètre avec le même nombre de lag pour les deux afin d'assurer plus d'équité dans nos comparaisons.

On observe une p-value de 0.9 pour le modèle ARMA contre une de 0.6 pour le SARIMA.

On peut donc conclure que l'hypothèse d'indépendance des résidus est plus forte pour notre modèle ARMA que SARIMA. Enfin concernant la normalité des résidus, bien que nos deux distributions en regardant les deux histogrammes soient bien centrés en 0, celle de notre ARMA semble plus symétrique.

Les sorties du test de shapiro wilk nous renvoie une p-value de notre modèle ARMA de 0.4 contre une p-value de 0.1 pour notre modèle SARIMA.

On en déduit donc que là encore, l'hypothèse de normalité de nos résidus est plus forte sur le processus ARMA.

Enfin, bien que nos deux processus aient une structure de modèle différente, l'un par différentiation saisonnière, l'autre par retranchement de la tendance et de la saison estimée, on peut tout de même regarder lequel minimise les critères BIC et AIC.

Les résultats bien que moins fiables, nous donnent là aussi le modèle ARMA comme optimal par rapport au modèle SARIMA.

Le modèle ARMA est donc le modèle de prédiction le plus optimal pour la prédiction de notre série.

Voici le code qui nous a permis de réaliser notre conclusion :

```
1  #Modèle 1:
2
3  checkresiduals(arma_model)
4  shapiro.test(arma_model$residuals)
5  Box.test(residuals(arma_model), lag = 20, type = "Ljung-Box", fitdf = 4)
6
7
8
9  #Modèle 2 :
10
11  checkresiduals(sarima_model)
12  shapiro.test(sarima_model$residuals)
13  Box.test(residuals(sarima_model), lag = 20, type = "Ljung-Box", fitdf = 5)
14
```