Script2

Maëlie Perier

2024-05-18

On charge les packages

```
#install.packages('tinytex')
#tinytex::install_tinytex()
library(urca)
#library(fUnitRoots)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#install.packages("tseries")
#install.packages("forecast")
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
#library(astsa)
library(ellipse)
```

```
##
## Attaching package: 'ellipse'

## The following object is masked from 'package:graphics':
##
##     pairs
```

```r
library(ggplot2)
```

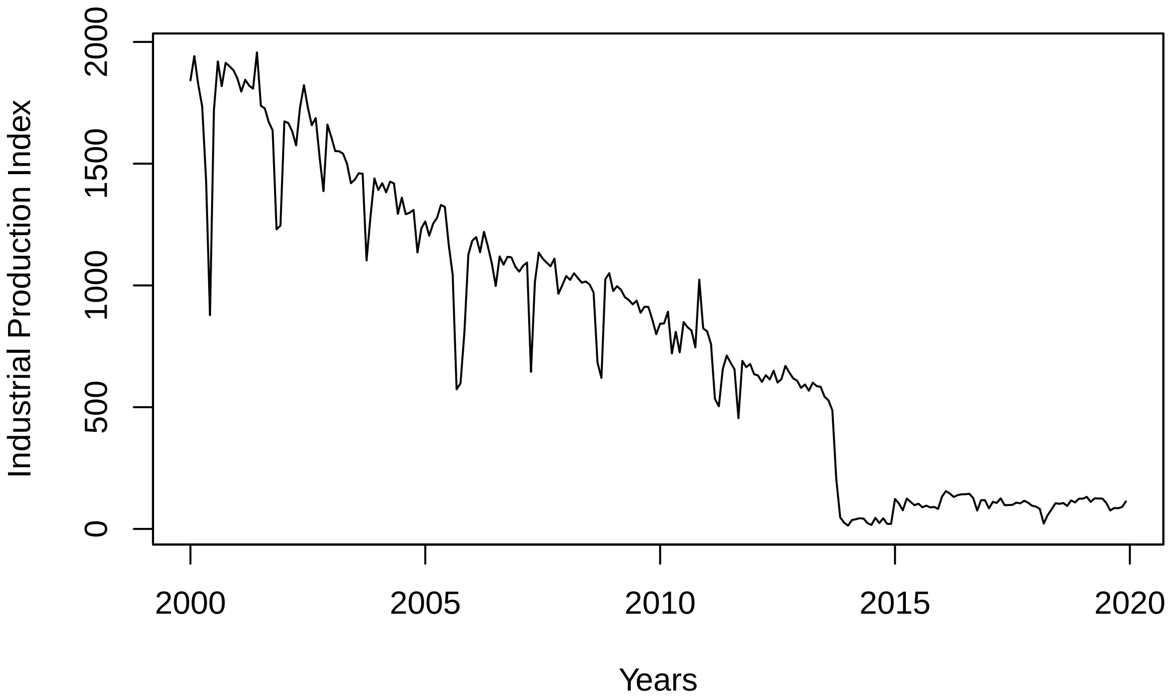
On importe les données - gaz naturel

Ensuite on prend les données de 2000 à 2019 pour éviter les chocs causés par les chocs pétroliers de 1990 ainsi que le Covid-19. On range le df par date puis on le converti en série temporelle

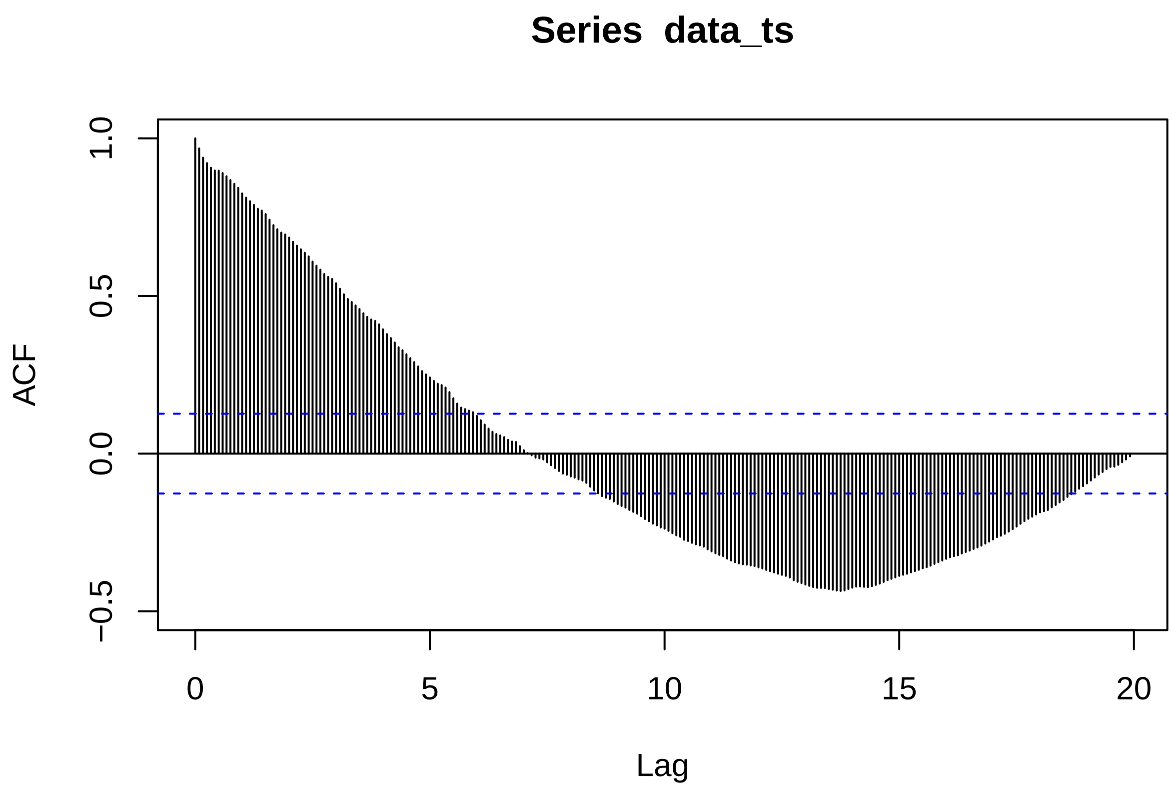
```r
datafile <- "valeurs_mensuelles.csv"
data <- read.csv(datafile, sep = ";", skip = 3, col.names = c("Date", "x1", "ignore"))

data$Date <- as.yearmon(data$Date)
data_filt <- subset(data, Date >= as.yearmon("Jan 2000") & Date <= as.yearmon("Dec 2019"))
data_filt <- data_filt[order(data_filt$Date), ]
data_ts_ <- ts(data_filt$x1, start = data_filt$Date[1], frequency = 12)

data_ts_entiere <- ts(data$x1, start = data$Date[1], frequency = 12)

data_ts <- zoo(data_ts_, order.by = index(data_ts_))
```

Including Plots

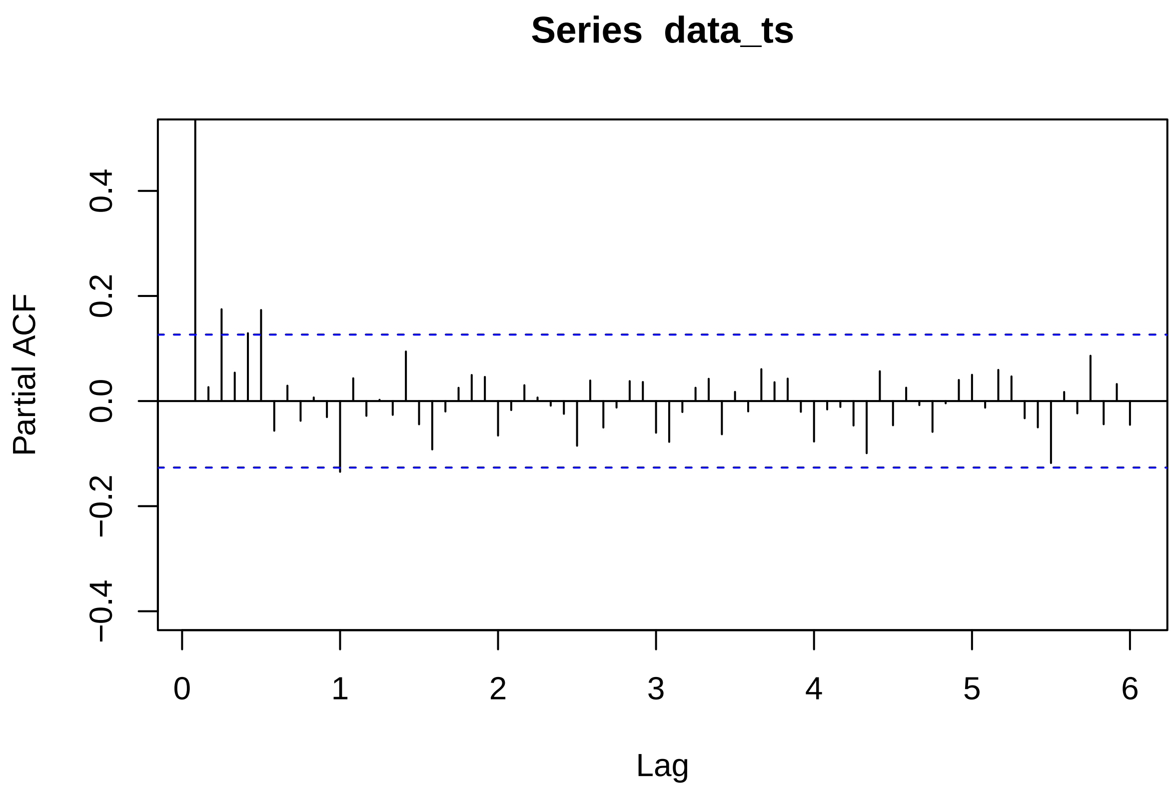
```r
plot(data_ts, xlab="Years", ylab = "Industrial Production Index")
```



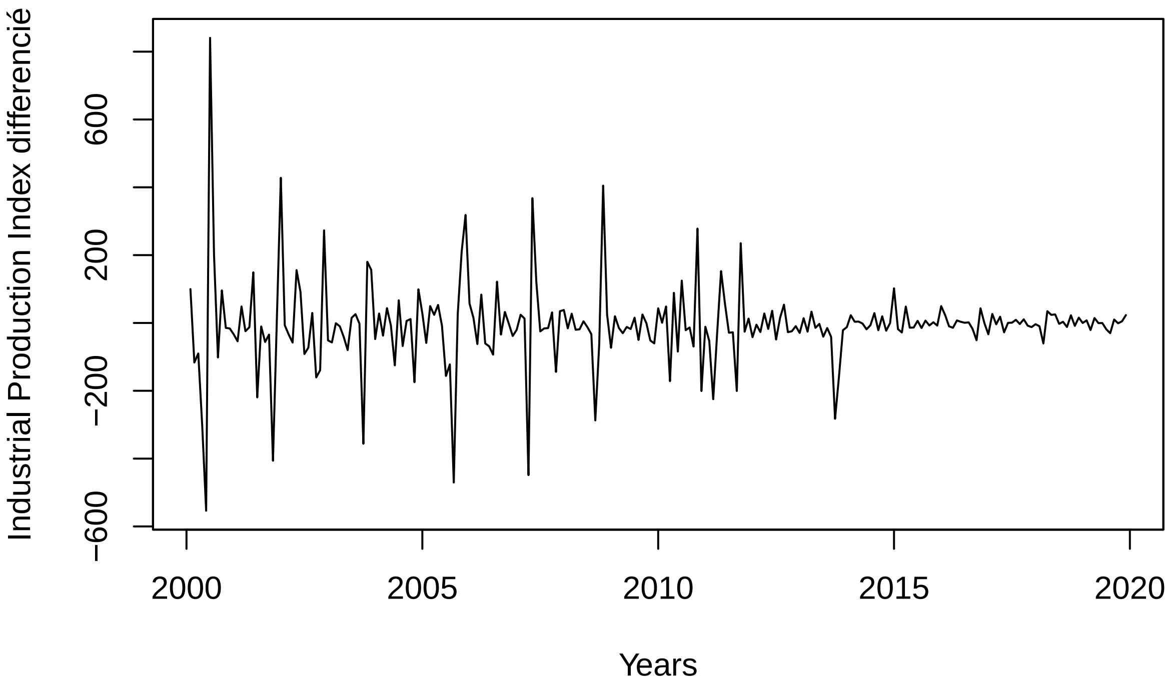

```
acf(data_ts, 20*12, ylim=c(-0.5, 1))
```



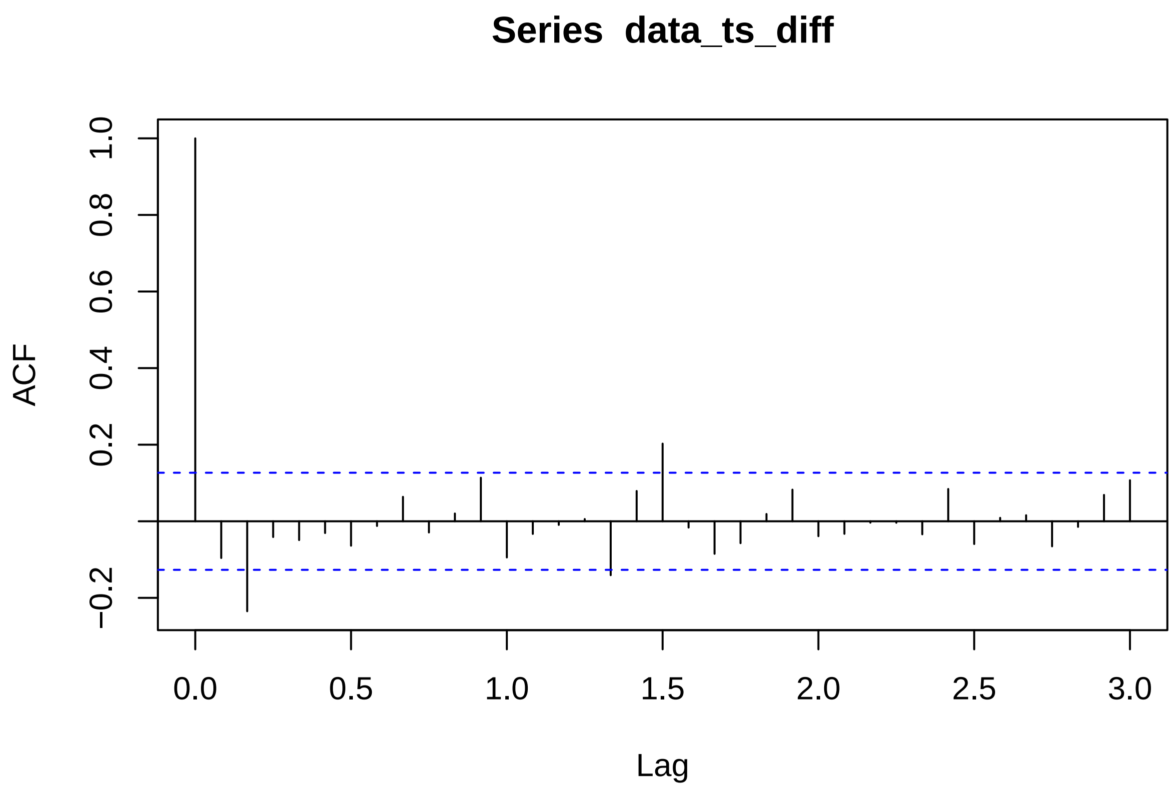

```
pacf(data_ts, 6*12, ylim=c(-0.4, 0.5))
```



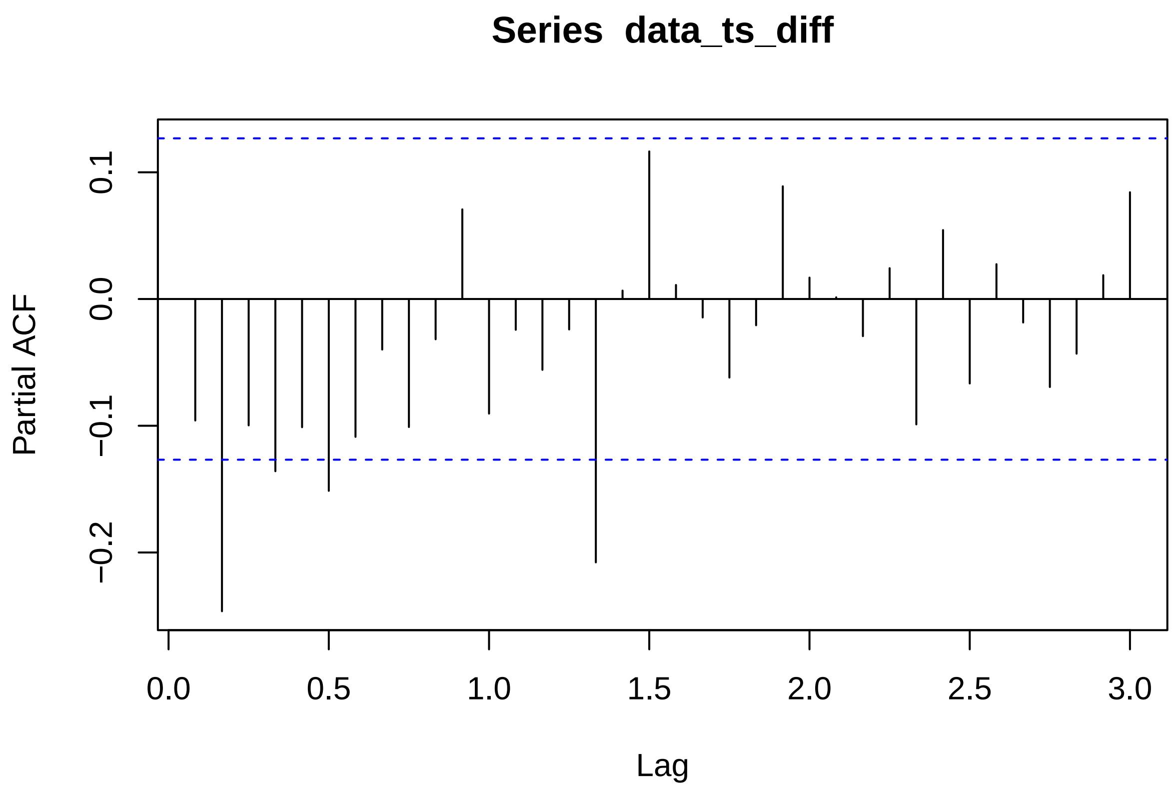

```r
data_ts_diff <- diff(data_ts)
plot(data_ts_diff, xlab="Years", ylab = "Industrial Production Index differencié")
```




```
acf(data_ts_diff, 3*12)
```




```
pacf(data_ts_diff, 3*12)
```




```r
qmax <- 2
pmax <- 2
```

La série semble stationnaire, la moyenne semble nulle même si la variance évolue entre avant et après 2013. Nous souhaitons désormais vérifier notre hypothèse de stationarité à l'aide de qqs tests.

```r
adf_test_result <- adf.test(data_ts_diff)
```

```
## Warning in adf.test(data_ts_diff): p-value smaller than printed p-value
```

```r
print(adf_test_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts_diff
## Dickey-Fuller = -8.7955, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
pp_result <- pp.test(data_ts_diff)
```

```
## Warning in pp.test(data_ts_diff): p-value smaller than printed p-value
```

```r
print(pp_result)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  data_ts_diff
## Dickey-Fuller Z(alpha) = -214, Truncation lag parameter = 4, p-value =
## 0.01
## alternative hypothesis: stationary
```

```r
kpss_result <- kpss.test(data_ts_diff)
```

```
## Warning in kpss.test(data_ts_diff): p-value greater than printed p-value
```

```r
print(kpss_result)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  data_ts_diff
## KPSS Level = 0.033042, Truncation lag parameter = 4, p-value = 0.1
```

Les tests confirment qu'on rejete l'hypothèse de stationarité à tous les seuils En regardant les graphiques précédents, il semble qu'on ait pmax = 2 et qmax = 2

On regarde si le modèle ARIMA(2,0,2) est bien ajusté et valide (respectivement que ses coeffs sont significatifs et que ces résidus ne sont pas autocorrélés)

```r
arima202 <- arima(data_ts_diff,c(2,0,2))

signif <- function(estim){
coef <- estim$coef
se <- sqrt(diag(estim$var.coef))
t <- coef/se
pval <- ((1-pnorm(abs(t)))*2)
return(rbind(coef,se,pval))
}
signif(arima202)
```

```
##            ar1       ar2        ma1        ma2     intercept
## coef 0.1386717 0.1754210 -0.3604408 -0.4721126 -7.2614925430
## se   0.3381002 0.2450284  0.3226700  0.3041114  1.8800310694
## pval 0.6816962 0.4740402  0.2639699  0.1205587  0.0001122637
```

```r
Box.test(arima202$residuals,lag=15, type="Ljung-Box", fitdf=4)
```

```
##
##  Box-Ljung test
##
## data:  arima202$residuals
## X-squared = 6.8085, df = 11, p-value = 0.8144
```

```r
Qtests <- function(series, k, fitdf=0) {
pvals <- apply(matrix(1:k), 1, FUN=function(l) {
pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung-Box", fitdf=fitdf)$p.value
return(c("lag"=l,"pval"=pval))
})
return(t(pvals))
}
Qtests(arima202$residuals, 15, 4)
```

```
##       lag      pval
##  [1,]   1        NA
##  [2,]   2        NA
##  [3,]   3        NA
##  [4,]   4        NA
##  [5,]   5 0.7513196
##  [6,]   6 0.7639416
##  [7,]   7 0.9059159
##  [8,]   8 0.8498827
##  [9,]   9 0.9262199
## [10,]  10 0.9630673
## [11,]  11 0.7739638
## [12,]  12 0.6218093
## [13,]  13 0.7085559
## [14,]  14 0.7471666
## [15,]  15 0.8143721
```

L'absence d'autocorrélation entre les résidus n'est jamais rejetée à 95% jusqu'à 15 retards. Le modèle semble donc valide. Toutefois, il n'est pas bien ajusté car les coefficients MA(q) avec $q > 4$ ne sont pas significatifs. On fait une fonction pour tester tous les modèles ARIMA(p,d,q) avec $p < pmax$ et $q < qmax$ et voir s'ils sont 1) bien ajustés et 2) valides. Une fois cela fait, on choisira entre les différents modèles selectionnés.

```r
signif <- function(estim){# test des significations individuelles des coefficients
coef <- estim$coef
se <- sqrt(diag(estim$var.coef))
t <- coef/se
pval <- (1-pnorm(abs(t)))*2
return(rbind(coef,pval))
}
for (p in 0:2) {
  for (q in 0:2) {
    print(c(p,q))
    print(signif(arima(data_ts, c(p,0,q), include.mean = FALSE))) }
}
```

```
## [1] 0 0
##
## coef
## pval
## [1] 0 1
##            ma1
## coef 0.9661089
## pval 0.0000000
```

```
## [1] 0 2
##           ma1       ma2
## coef 1.516332 0.7845318
## pval 0.000000 0.0000000
## [1] 1 0
##            ar1
## coef 0.9938347
## pval 0.0000000
## [1] 1 1
##            ar1         ma1
## coef 0.9961665 -0.17850372
## pval 0.0000000  0.05638633
## [1] 1 2
##            ar1           ma1           ma2
## coef 0.9990732 -0.2124017346 -3.457175e-01
## pval 0.0000000  0.0005444728  5.114833e-08
## [1] 2 0
##            ar1        ar2
## coef 0.9063376 0.08830753
## pval 0.0000000 0.17087180
## [1] 2 1
##           ar1           ar2        ma1
## coef 1.572686 -5.729770e-01 -0.8386871
## pval 0.000000  1.342704e-12  0.0000000
## [1] 2 2
##               ar1         ar2         ma1          ma2
## coef 1.292618e+00 -0.29313727 -0.47658019 -0.249652719
## pval 2.442491e-15  0.07233333  0.00336098  0.008798877
```

On retient les modèles ARIMA suivant qui sont ajustés : ARIMA(0,0,1) ARIMA(0,0,2) ARIMA(1,0,0) ARIMA(1,0,2) ARIMA(2,0,1)

```r
arima_coefs <- list(
  c(0, 0, 1), c(0, 0, 2),
  c(1, 0, 0), c(1, 0, 2),
  c(2, 0, 1)
)

# Loop through the list of coefficients
valid_arima <- list()
for (coefs in arima_coefs) {
  cat("\nFitting ARIMA(", coefs[1], ",", coefs[2], ",", coefs[3], ") model:\n", sep = "")
  tryCatch({
    model <- arima(data_ts, order = coefs, include.mean = FALSE)
    residuals <- residuals(model)

    # Perform Box-Ljung test on residuals
    box_test <- Box.test(residuals, lag = 15, type = "Ljung-Box", fitdf = sum(coefs))
    cat("\nBox-Ljung Test:\n")
    print(box_test)

    if (box_test$p.value > 0.05) {
      valid_arima <- c(valid_arima, list(coefs))
    }
```

```r
  })
}
```

```
##
## Fitting ARIMA(0,0,1) model:
##
## Box-Ljung Test:
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 1754.3, df = 14, p-value < 2.2e-16
##
##
## Fitting ARIMA(0,0,2) model:
##
## Box-Ljung Test:
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 748.44, df = 13, p-value < 2.2e-16
##
##
## Fitting ARIMA(1,0,0) model:
##
## Box-Ljung Test:
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 25.064, df = 14, p-value = 0.03394
##
##
## Fitting ARIMA(1,0,2) model:
##
## Box-Ljung Test:
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 15.191, df = 12, p-value = 0.2312
##
##
## Fitting ARIMA(2,0,1) model:
##
## Box-Ljung Test:
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 14.664, df = 12, p-value = 0.2604
```

```r
cat("\nList of ARIMA models with Box-Ljung test p-value > 0.05:\n")
```

```
##
## List of ARIMA models with Box-Ljung test p-value > 0.05:
```

```r
for (model in valid_arima) {
  cat("ARIMA(", model[[1]], ",", model[[2]], ",", model[[3]], ")\n", sep = "")
}
```

```
## ARIMA(1,0,2)
## ARIMA(2,0,1)
```

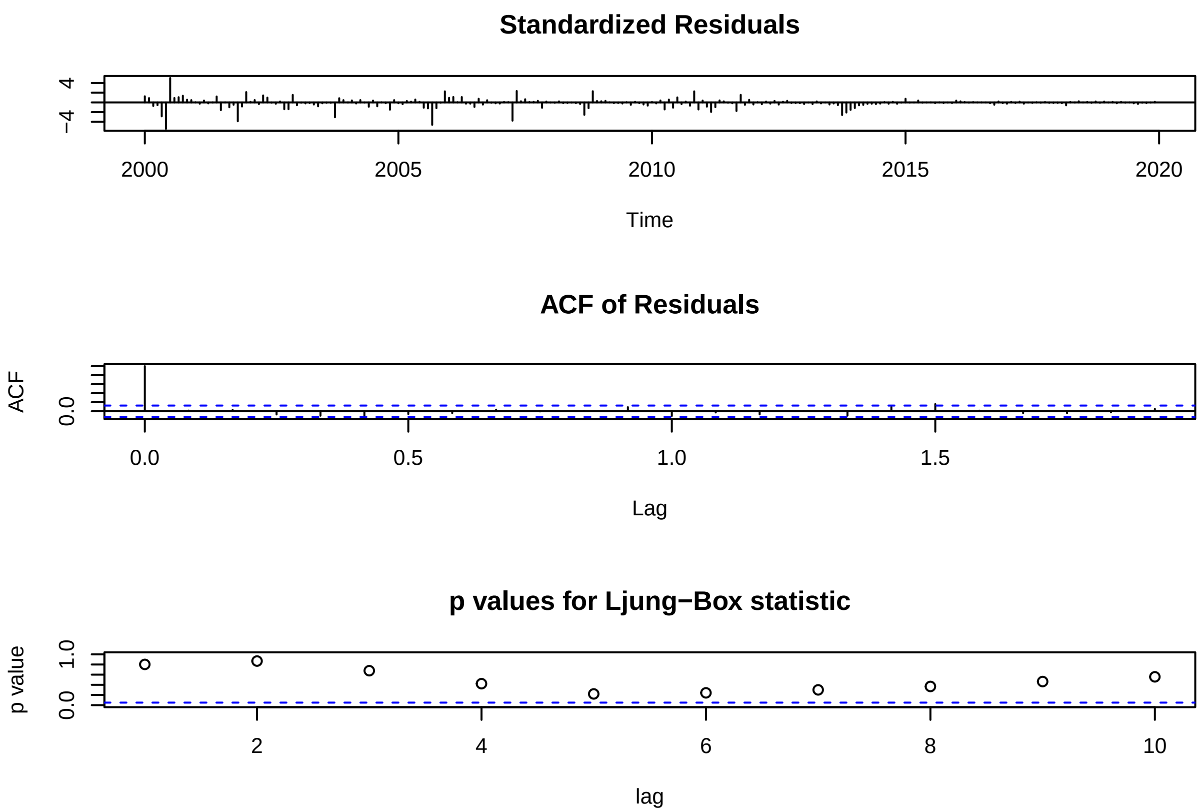
Il nous reste donc plus que deux modèles qui sont à la fois valides et avec des coefficients significatifs : ARIMA(1,0,2) et ARIMA(2,0,1) Arbitrons entre les deux à l'aide des AIC/BIC

```r
aic_12 <- AIC(arima(data_ts, order=c(1,0,2)))
bic_12 <- BIC(arima(data_ts, order=c(1,0,2)))
aic_21 <- AIC(arima(data_ts, order=c(2,0,1)))
bic_21 <- BIC(arima(data_ts, order=c(2,0,1)))

cat("AIC et BIC de l'ARMA(1,2)", aic_12, "/", bic_12,"\nAIC et BIC de l'ARMA(2,1)", aic_21, "/", bic_21
```

```
## AIC et BIC de l'ARMA(1,2) 2984.384 / 3001.788
## AIC et BIC de l'ARMA(2,1) 2987.243 / 3004.647
```

Les AIC et BIC de l'ARMA(1,2) sont légèrement plus faibles, on opte donc pour ce modèle là.

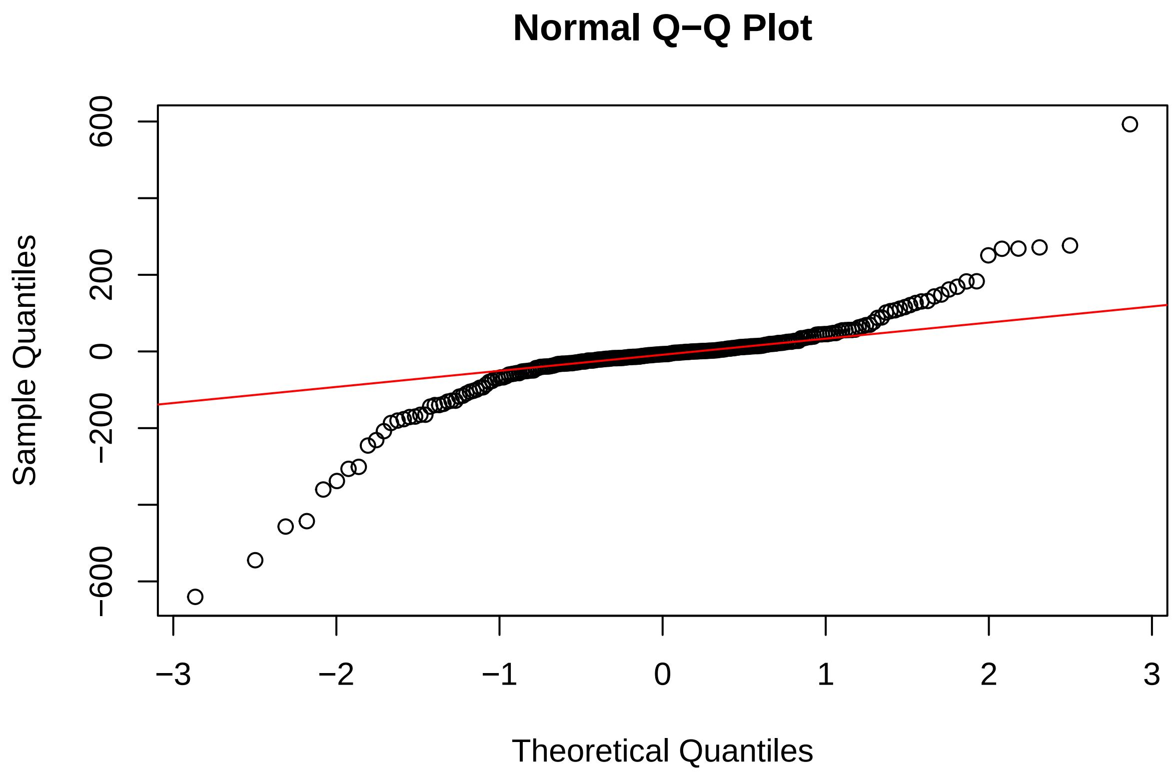
```r
## Question 7: Diagnostic du modèle et vérification de l'hypothèse de normalité des résidus

# Ajustement du modèle ARIMA(1,0,2)
fit <- arima(data_ts, order=c(1,0,2))

# Diagnostic du modèle ARIMA(1,0,2)
tsdiag(fit)
```



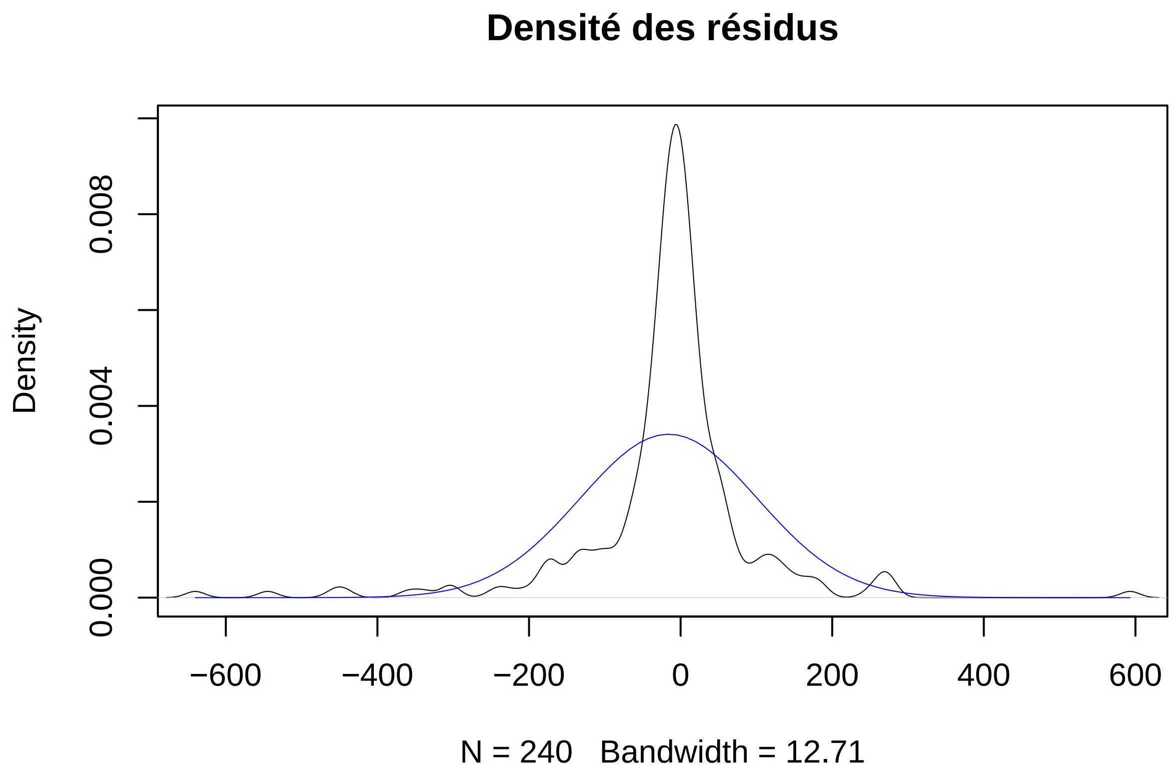

```r
# Test de Jarque-Bera pour la normalité des résidus
jarque.bera.test(fit$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  fit$residuals
## X-squared = 832.38, df = 2, p-value < 2.2e-16
```

```r
# Q-Q plot des résidus
qqnorm(fit$residuals)
qqline(fit$residuals, col = "red")
```




```r
# Densité des résidus
plot(density(fit$residuals), lwd=0.5, xlim=range(fit$residuals), main="Densité des résidus")
mu <- mean(fit$residuals)
sigma <- sd(fit$residuals)
x <- seq(min(fit$residuals), max(fit$residuals), length.out=100)
y <- dnorm(x, mean=mu, sd=sigma)
lines(x, y, lwd=0.5, col="blue")
```




```r
## Question 8: Vérification des racines du modèle

# Extraction des coefficients du modèle ARIMA(1,0,2)
phi_1 <- as.numeric(fit$coef[1])
theta_1 <- as.numeric(fit$coef["ma1"])
theta_2 <- as.numeric(fit$coef["ma2"])
sigma2 <- as.numeric(fit$sigma2)

# Affichage des coefficients
phi_1
```

```
## [1] 0.9975756
```

```r
theta_1
```

```
## [1] -0.2106072
```

```r
theta_2
```

```
## [1] -0.3438213
```

```r
sigma2
```

```
## [1] 13882.39
```

```r
# Vérification des racines
ar_coefs <- c(phi_1)
ma_coefs <- c(theta_1, theta_2)

# Vérification si les racines sont en dehors du cercle unité
ar_roots <- polyroot(c(1, -ar_coefs))
ma_roots <- polyroot(c(1, ma_coefs))

abs(ar_roots)
```

```
## [1] 1.00243
```

```r
abs(ma_roots)
```

```
## [1] 1.426438 2.038987
```

```r
all(abs(ar_roots) > 1)
```

```
## [1] TRUE
```

```r
all(abs(ma_roots) > 1)
```

```
## [1] TRUE
```

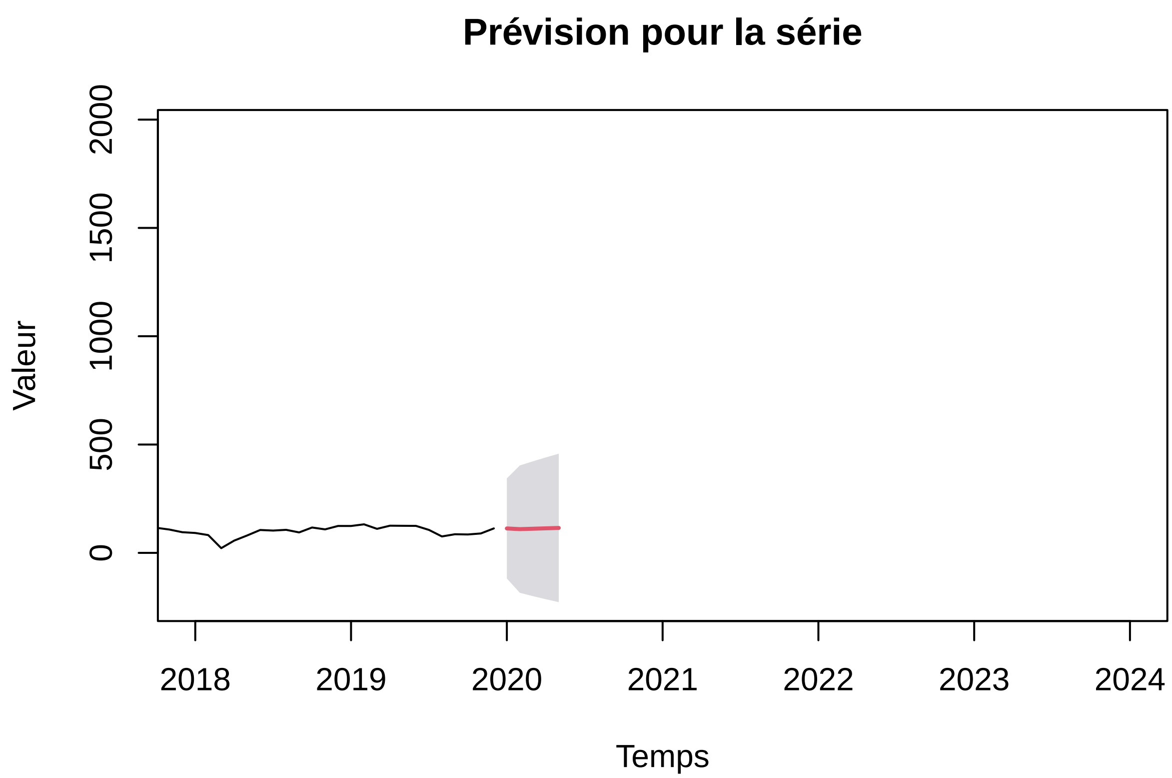
```r
## Prédiction

# Prédiction des deux prochaines valeurs
XT1 <- predict(fit, n.ahead=2)$pred[1]
XT2 <- predict(fit, n.ahead=2)$pred[2]

XT1
```

```
## [1] 112.7995
```

```r
XT2
```

```
## [1] 109.5783
```

```r
# Prédiction pour la série
fore <- forecast(fit, h=5, level=95)
par(mfrow=c(1,1))
plot(fore, xlim=c(2018,2024), col=1, fcol=2, shaded=TRUE, xlab="Temps", ylab="Valeur", main="Prévision p
```



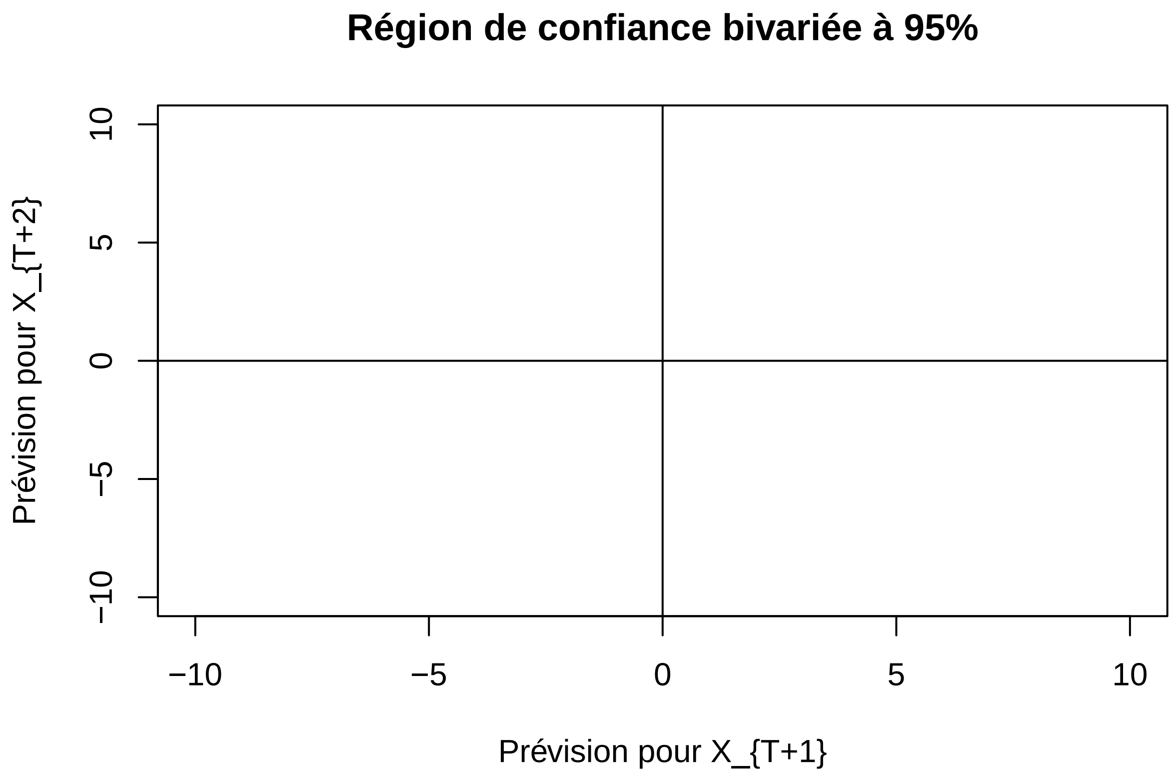

```r
# Calcul de la matrice de variance-covariance
Sigma <- matrix(c(sigma2, theta_1 * sigma2, theta_2 * sigma2,
                  theta_1 * sigma2, sigma2, 0,
                  theta_2 * sigma2, 0, sigma2), ncol=3)

# Plot de la région de confiance bivariée à 95%
library(ellipse)
plot(XT1, XT2, xlim=c(-10, 10), ylim=c(-10, 10), xlab="Prévision pour X_{T+1}", ylab="Prévision pour X_{
     main="Région de confiance bivariée à 95%")
ellipse(Sigma[1:2, 1:2], center=c(XT1, XT2), type="l", col="red", radius=1)
```

```
##                 x          y
##    [1,]  181.188336  181.188336
##    [2,]  166.592448  195.054642
##    [3,]  151.325752  208.135532
##    [4,]  135.449720  220.378333
##    [5,]  119.028279  231.733749
##    [6,]  102.127554  242.156055
##    [7,]   84.815598  251.603284
##    [8,]   67.162119  260.037395
##    [9,]   49.238201  267.424427
##   [10,]   31.116019  273.734636
##   [11,]   12.868544  278.942612
##   [12,]   -5.430749  283.027384
##   [13,]  -23.708173  285.972505
##   [14,]  -41.890134  287.766116
```

```
## [15,]  -59.903417  288.400994
## [16,]  -77.675491  287.874583
## [17,]  -95.134793  286.189003
## [18,] -112.211020  283.351040
## [19,] -128.835414  279.372123
## [20,] -144.941032  274.268273
## [21,] -160.463025  268.060042
## [22,] -175.338889  260.772427
## [23,] -189.508725  252.434774
## [24,] -202.915477  243.080655
## [25,] -215.505160  232.747736
## [26,] -227.227079  221.477624
## [27,] -238.034036  209.315700
## [28,] -247.882513  196.310935
## [29,] -256.732855  182.515696
## [30,] -264.549424  167.985530
## [31,] -271.300746  152.778946
## [32,] -276.959636  136.957175
## [33,] -281.503307  120.583926
## [34,] -284.913463  103.725128
## [35,] -287.176374   86.448665
## [36,] -288.282927   68.824104
## [37,] -288.228666   50.922413
## [38,] -287.013810   32.815675
## [39,] -284.643250   14.576800
## [40,] -281.126533   -3.720771
## [41,] -276.477819  -22.003359
## [42,] -270.715826  -40.197348
## [43,] -263.863755  -58.229476
## [44,] -255.949199  -76.027135
## [45,] -247.004025  -93.518659
## [46,] -237.064254 -110.633616
## [47,] -226.169908 -127.303091
## [48,] -214.364856 -143.459962
## [49,] -201.696632 -159.039169
## [50,] -188.216248 -173.977982
## [51,] -173.977982 -188.216248
## [52,] -159.039169 -201.696632
## [53,] -143.459962 -214.364856
## [54,] -127.303091 -226.169908
## [55,] -110.633616 -237.064254
## [56,]  -93.518659 -247.004025
## [57,]  -76.027135 -255.949199
## [58,]  -58.229476 -263.863755
## [59,]  -40.197348 -270.715826
## [60,]  -22.003359 -276.477819
## [61,]   -3.720771 -281.126533
## [62,]   14.576800 -284.643250
## [63,]   32.815675 -287.013810
## [64,]   50.922413 -288.228666
## [65,]   68.824104 -288.282927
## [66,]   86.448665 -287.176374
## [67,]  103.725128 -284.913463
## [68,]  120.583926 -281.503307
```

```
##  [69,]  136.957175 -276.959636
##  [70,]  152.778946 -271.300746
##  [71,]  167.985530 -264.549424
##  [72,]  182.515696 -256.732855
##  [73,]  196.310935 -247.882513
##  [74,]  209.315700 -238.034036
##  [75,]  221.477624 -227.227079
##  [76,]  232.747736 -215.505160
##  [77,]  243.080655 -202.915477
##  [78,]  252.434774 -189.508725
##  [79,]  260.772427 -175.338889
##  [80,]  268.060042 -160.463025
##  [81,]  274.268273 -144.941032
##  [82,]  279.372123 -128.835414
##  [83,]  283.351040 -112.211020
##  [84,]  286.189003  -95.134793
##  [85,]  287.874583  -77.675491
##  [86,]  288.400994  -59.903417
##  [87,]  287.766116  -41.890134
##  [88,]  285.972505  -23.708173
##  [89,]  283.027384   -5.430749
##  [90,]  278.942612   12.868544
##  [91,]  273.734636   31.116019
##  [92,]  267.424427   49.238201
##  [93,]  260.037395   67.162119
##  [94,]  251.603284   84.815598
##  [95,]  242.156055  102.127554
##  [96,]  231.733749  119.028279
##  [97,]  220.378333  135.449720
##  [98,]  208.135532  151.325752
##  [99,]  195.054642  166.592448
## [100,]  181.188336  181.188336
```

```r
points(XT1, XT2, col="blue")
abline(h=XT2, v=XT1, col="blue")
abline(h=0, v=0)
```


Région de confiance bivariée à 95%
10
5
0
−5
−10
Prévision pour X_{T+2}
−10
−5
0
5
10
Prévision pour X_{T+1}