Économétrie des Séries Temporelles

Fiche TD R #2

Processus ARMA stationnaires

Packages

```r
library(readr)
library(zoo)
library(astsa)
library(forecast)
library(stats)
library(tseries)
library(aTSA)
#install.packages("aTSA")
```

Données (identiques au TP1)

Nice : https://github.com/bilelsanhaji/EdSTM1/blob/main/Data/SH_MIN006088001.csv

Paris : https://github.com/bilelsanhaji/EdSTM1/blob/main/Data/SH_MIN175114001.csv

```r
urlSHnice <- "https://raw.githubusercontent.com/bilelsanhaji/EdSTM1/refs/heads/main/Data/SH_MIN00608800
InsoNice <- read_delim(urlSHnice,
                       delim = ";",
                       escape_double = FALSE,
                       col_types = cols(YYYYMM = col_date(format = "%Y%m")),
                       comment = "#", trim_ws = TRUE)

urlSHparis <- "https://raw.githubusercontent.com/bilelsanhaji/EdSTM1/refs/heads/main/Data/SH_MIN17511400
InsoParis <- read_delim(urlSHparis,
                        delim = ";",
                        escape_double = FALSE,
                        col_types = cols(YYYYMM = col_date(format = "%Y%m")),
                        comment = "#", trim_ws = TRUE)
```

```r
Nice_ts = ts(InsoNice$VALEUR)
Paris_ts = ts(InsoParis$VALEUR)
```

Exercice 1

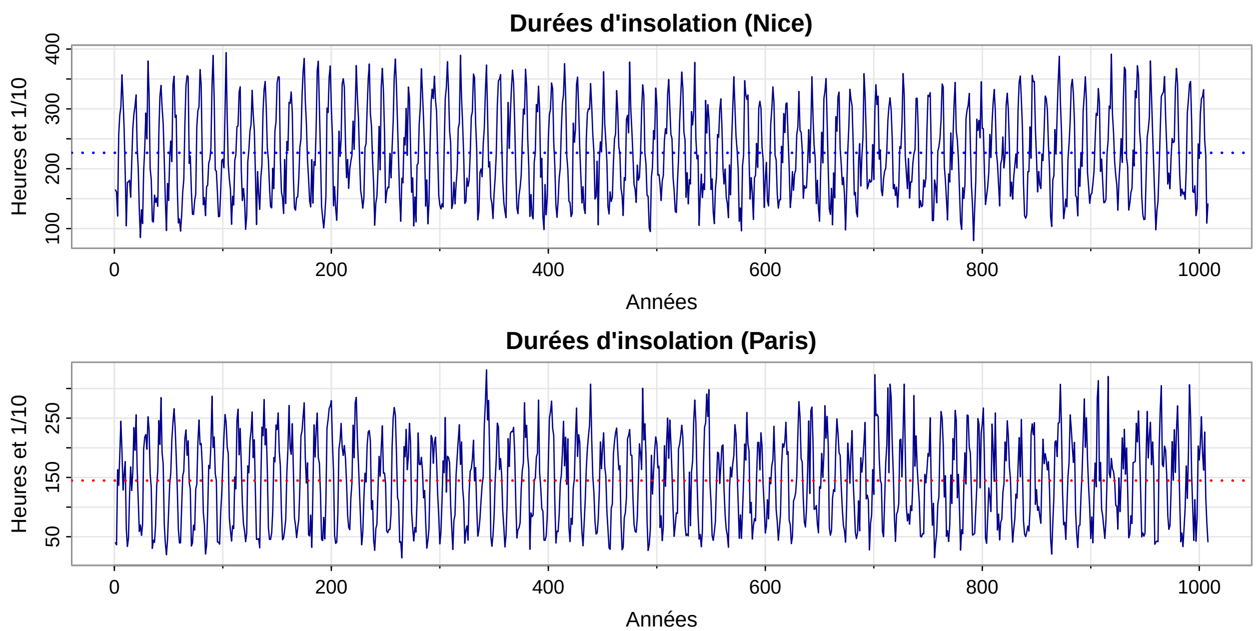
À partir des données d'insolation de Nice et Paris, utilisez les séries pour

(a) donner une représentation graphique et tester statistiquement :

1. la stationarité

```r
moyenne_Nice <- mean(Nice_ts)
moyenne_Paris <- mean(Paris_ts)

par(mfrow=c(2,1))
#Nice
tsplot(Nice_ts,
     main = "Durées d'insolation (Nice)",
     xlab = "Années",
     ylab = "Heures et 1/10",
     col = "darkblue")
abline(h = moyenne_Nice, col = "blue", lty = 3, lwd = 2)
#Paris
tsplot(Paris_ts,
       main = "Durées d'insolation (Paris)",
       xlab = "Années",
       ylab = "Heures et 1/10",
       col = "darkblue")
abline(h = moyenne_Paris, col = "red", lty = 3, lwd = 2)
```




```r
tseries::adf.test(Nice_ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Nice_ts
## Dickey-Fuller = -9.551, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

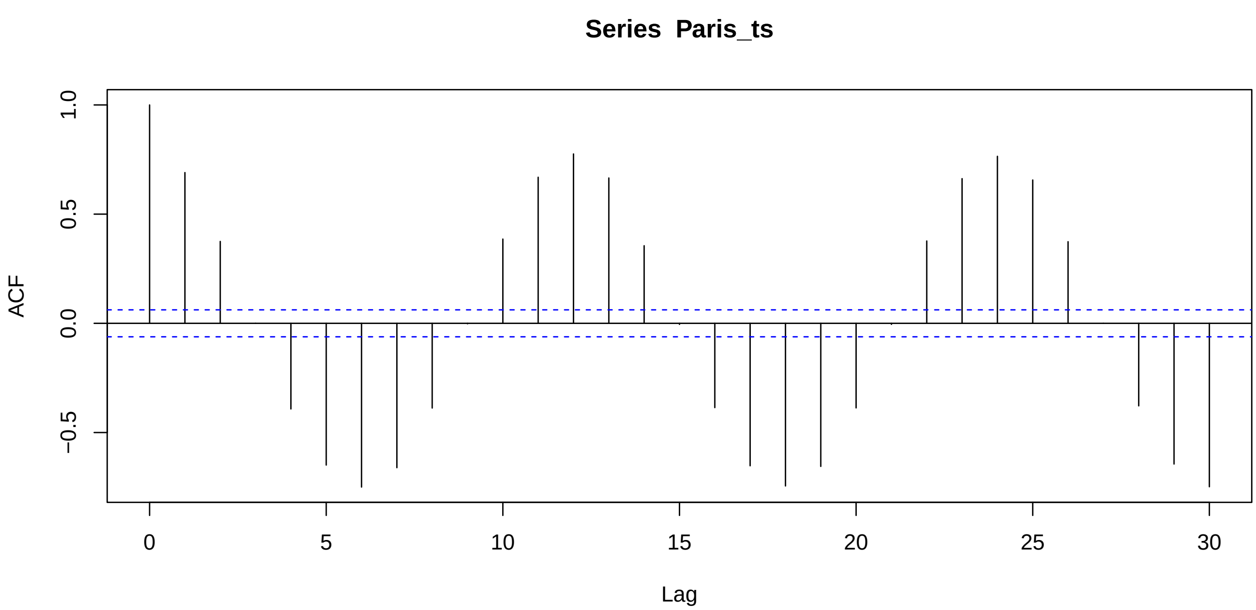
```r
aTSA::adf.test(Nice_ts)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
```

```
## [1,]   0 -3.71  0.0100
## [2,]   1 -4.18  0.0100
## [3,]   2 -4.79  0.0100
## [4,]   3 -4.52  0.0100
## [5,]   4 -3.62  0.0100
## [6,]   5 -2.72  0.0100
## [7,]   6 -2.04  0.0418
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -12.6    0.01
## [2,]   1 -15.9    0.01
## [3,]   2 -21.9    0.01
## [4,]   3 -26.1    0.01
## [5,]   4 -27.4    0.01
## [6,]   5 -27.1    0.01
## [7,]   6 -25.8    0.01
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -12.6    0.01
## [2,]   1 -15.9    0.01
## [3,]   2 -21.9    0.01
## [4,]   3 -26.1    0.01
## [5,]   4 -27.4    0.01
## [6,]   5 -27.2    0.01
## [7,]   6 -25.9    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

2. l'autocorrélation

```
acf(Nice_ts)
acf(Paris_ts)
```




```
Box.test(Nice_ts, lag = 6, type = "Ljung-Box")
```

```
##
```

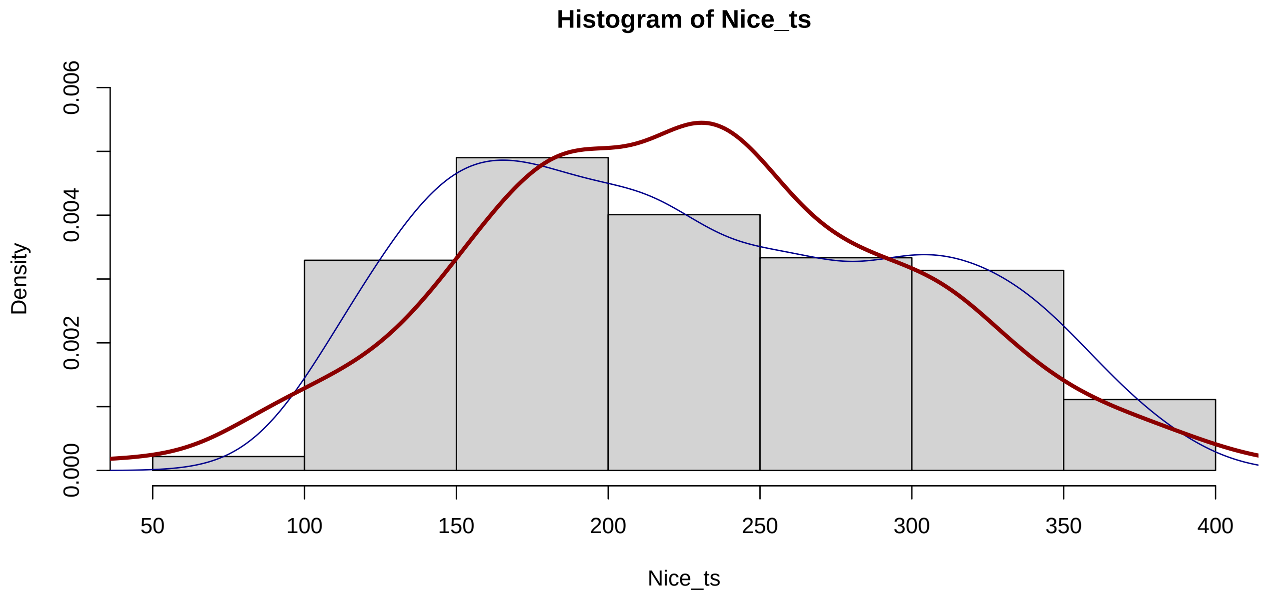
```
##  Box-Ljung test
##
## data:  Nice_ts
## X-squared = 1910.9, df = 6, p-value < 2.2e-16
```

```r
Box.test(Paris_ts, lag = 6, type = "Ljung-Box")
```

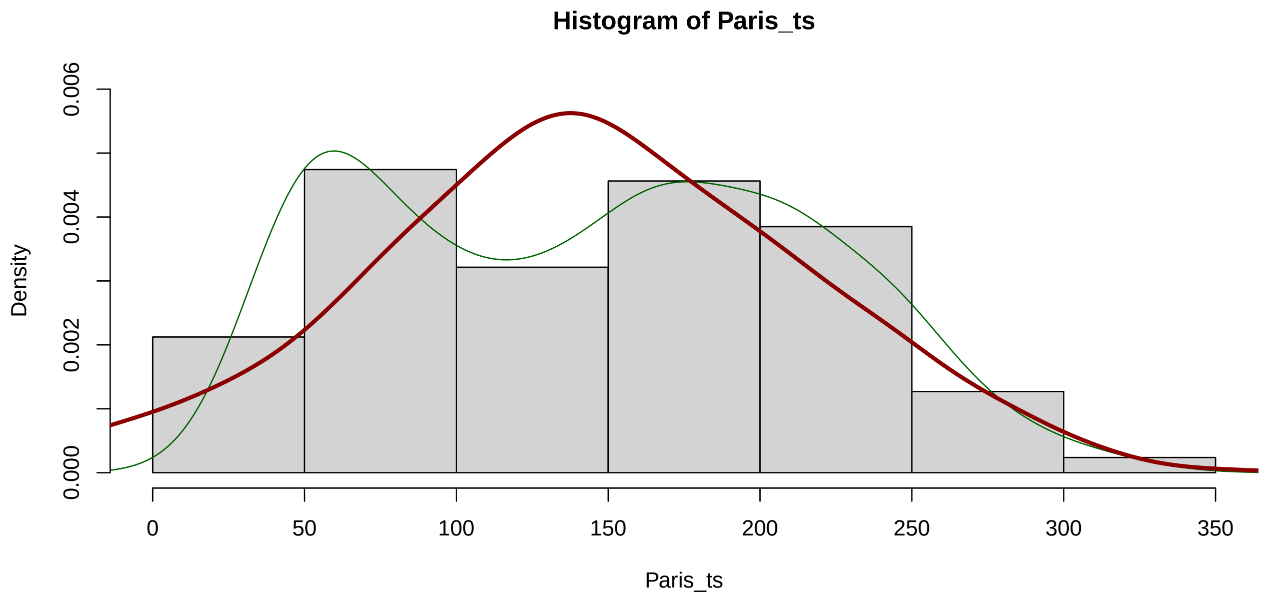
```
##
##  Box-Ljung test
##
## data:  Paris_ts
## X-squared = 1778.8, df = 6, p-value < 2.2e-16
```

3. la normalité

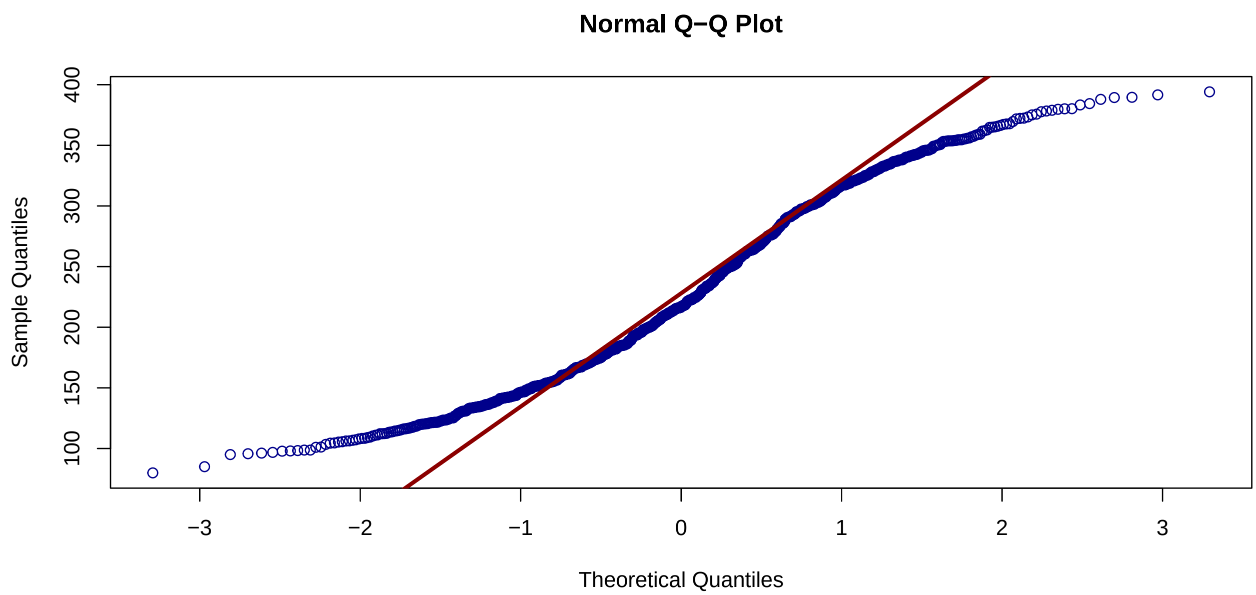
```r
hist(Nice_ts, freq = F, ylim=c(0,0.006))
lines(density(Nice_ts), col="darkblue")
lines(density(rnorm(n = length(Nice_ts), mean = mean(Nice_ts), sd = sd(Nice_ts))), col="darkred", lwd =
```



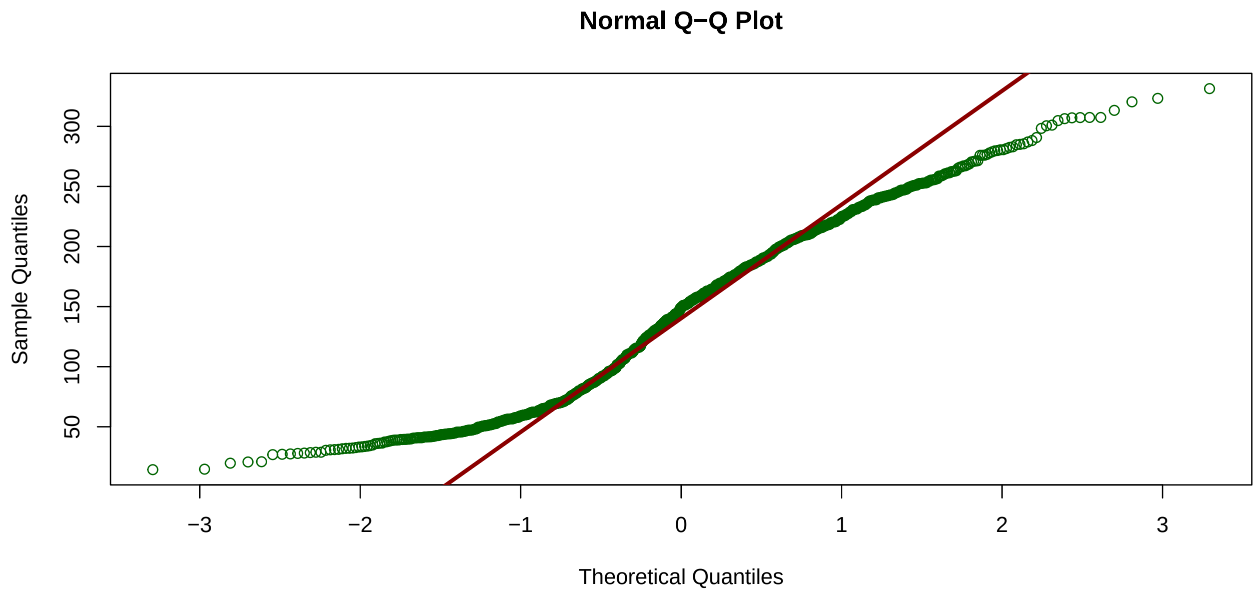

```r
hist(Paris_ts, freq = F, ylim=c(0,0.006))
lines(density(Paris_ts), col="darkgreen")
lines(density(rnorm(n = length(Paris_ts), mean = mean(Paris_ts), sd = sd(Paris_ts))), col="darkred", lwd
```




```r
qqnorm(Nice_ts, col = "darkblue")
qqline(Nice_ts, col="darkred", lwd = 3)
```




```r
qqnorm(Paris_ts, col = "darkgreen")
qqline(Paris_ts, col="darkred", lwd = 3)
```




```r
#shapiro.test(Nice_ts)
jarque.bera.test(Nice_ts)
```

```
##
##  Jarque Bera Test
##
## data:  Nice_ts
## X-squared = 53.152, df = 2, p-value = 2.872e-12
```

```r
jarque.bera.test(Paris_ts)
```

```
##
##  Jarque Bera Test
##
## data:  Paris_ts
## X-squared = 49.261, df = 2, p-value = 2.01e-11
```

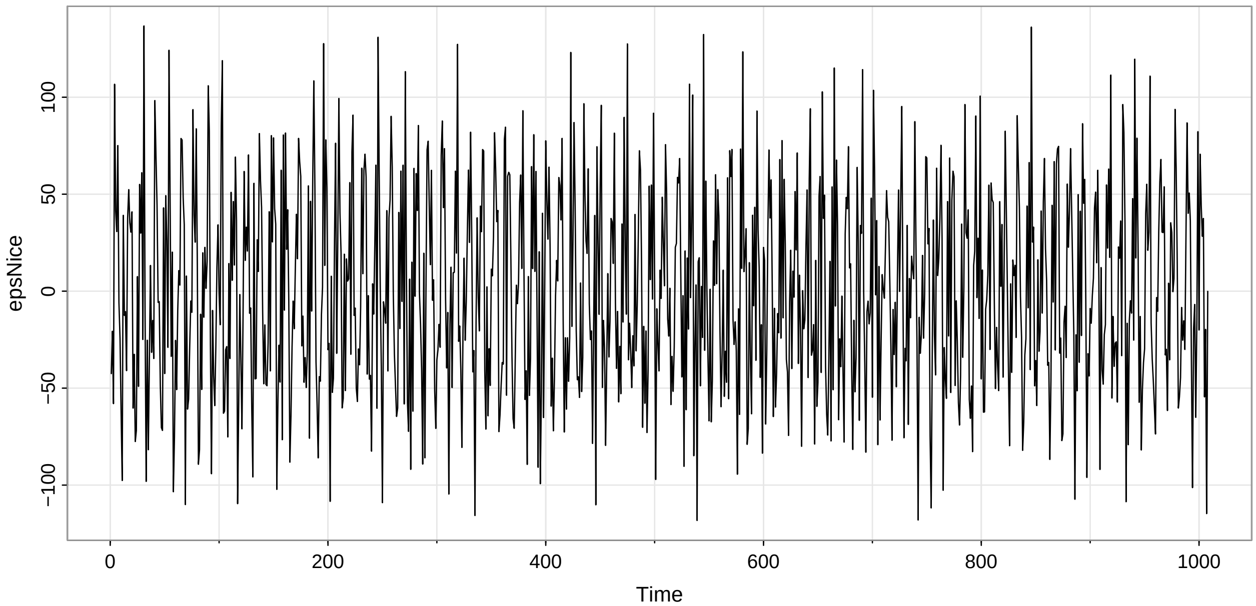
(b) estimer et interpréter un AR(1) pour chaque série, puis, sur les résidus :

```r
ar1Nice<- arima(Nice_ts, order = c(1,0,0))
summary(ar1Nice)
```

```
##
## Call:
## arima(x = Nice_ts, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.7249   226.3135
## s.e.  0.0217     5.8618
##
## sigma^2 estimated as 2634:  log likelihood = -5400.34,  aic = 10806.68
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.04839984 51.32466 42.49581 -5.990051 21.06189 0.9444972
##                    ACF1
```

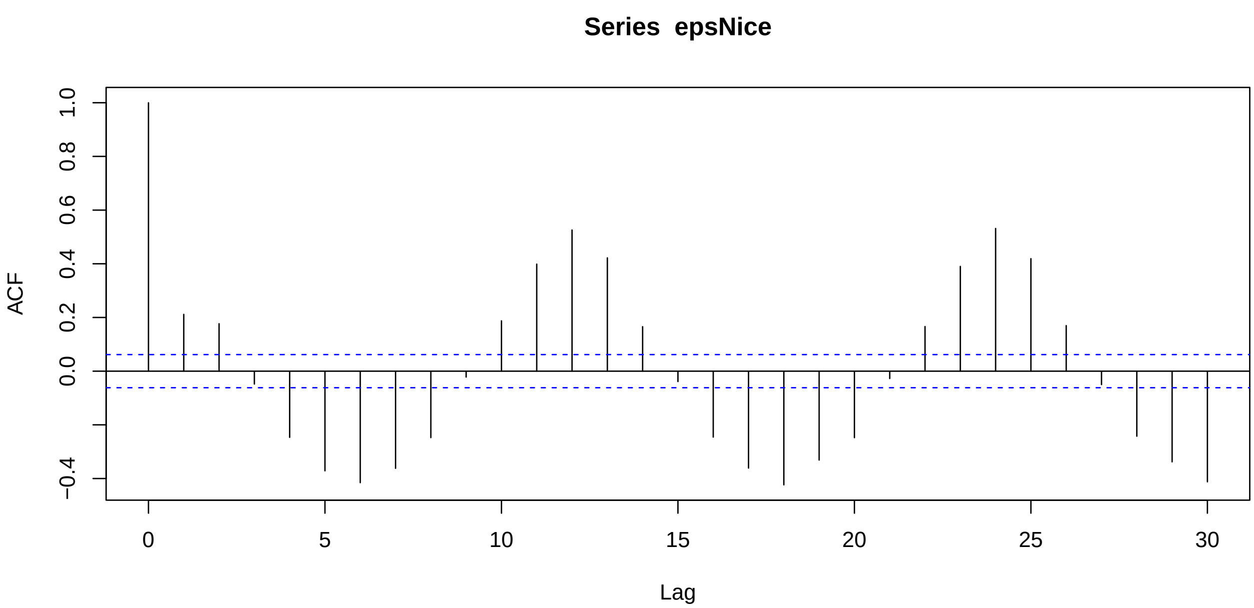
```
## Training set 0.2115974
```

```r
epsNice <- ar1Nice$residuals
#epsNice <- residuals(ar1Nice)
tsplot(epsNice)
```



1. tester autocorrelation, normalité et hétéroscédasticité

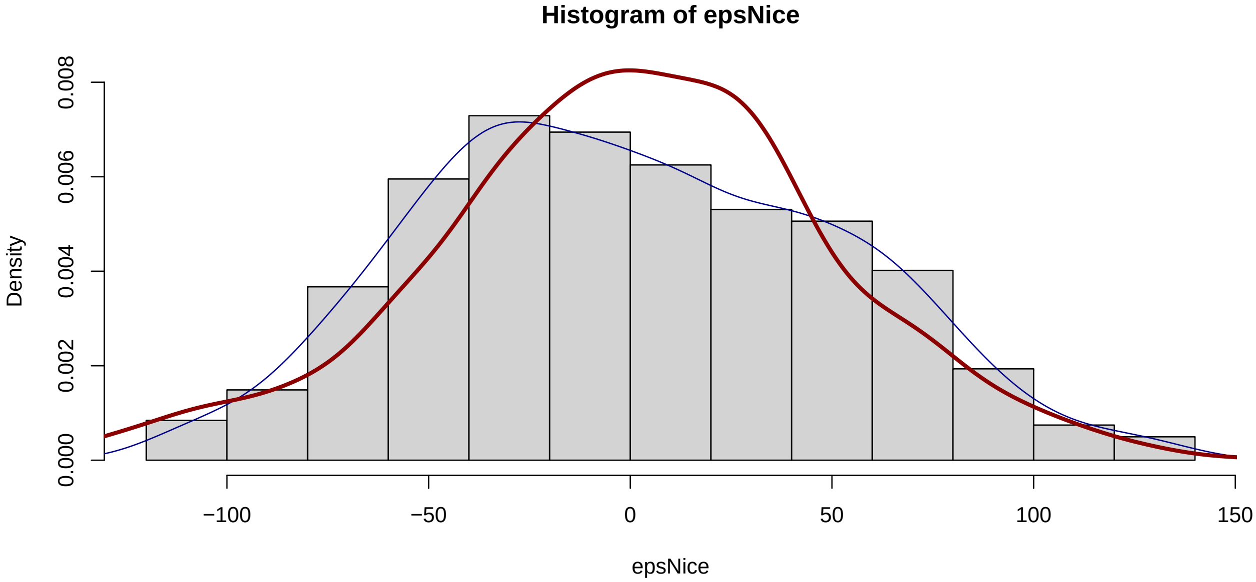
```r
acf(epsNice)
```




```r
Box.test(epsNice, lag = 6, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  epsNice
## X-squared = 456.3, df = 6, p-value < 2.2e-16
```

```r
hist(epsNice, freq = F, ylim=c(0,0.008))
lines(density(epsNice), col="darkblue")
lines(density(rnorm(n = length(epsNice), mean = mean(epsNice), sd = sd(epsNice))), col="darkred", lwd =
```



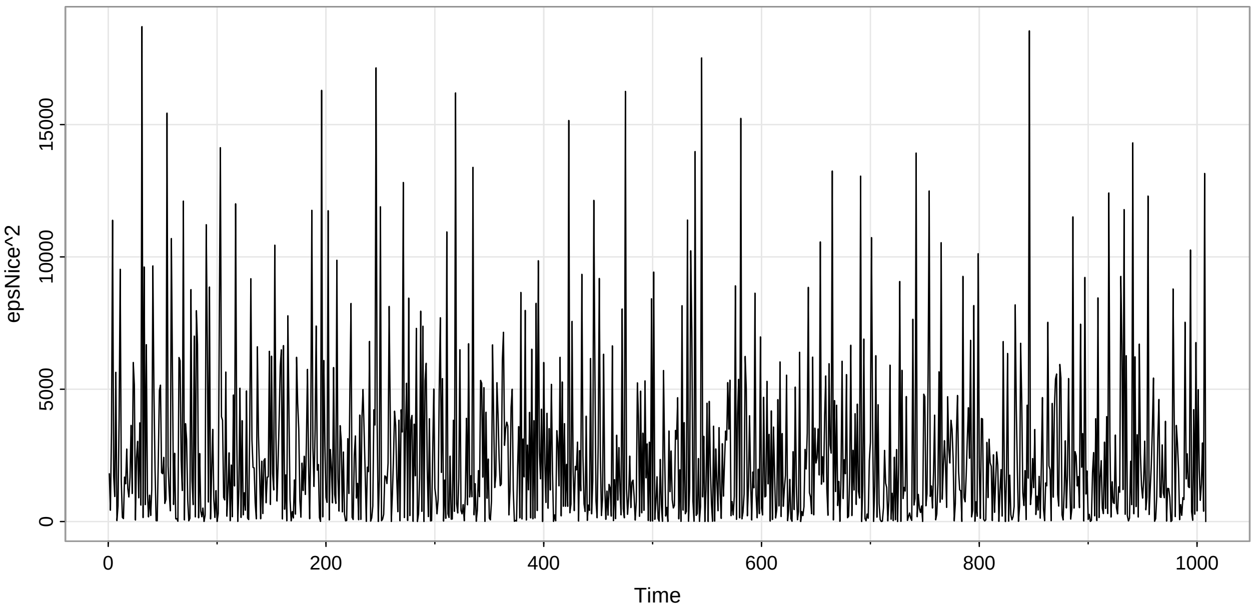

```r
jarque.bera.test(epsNice)
```

```
##
##  Jarque Bera Test
##
## data:  epsNice
## X-squared = 17.707, df = 2, p-value = 0.0001429
```

```r
aTSA::arch.test(ar1Nice)
```

```
## ARCH heteroscedasticity test for residuals
## alternative: heteroscedastic
##
## Portmanteau-Q test:
##      order    PQ  p.value
## [1,]     4  32.7 1.36e-06
## [2,]     8  39.3 4.35e-06
## [3,]    12  71.0 2.08e-10
## [4,]    16  76.0 8.57e-10
## [5,]    20  89.9 7.67e-11
## [6,]    24 118.4 1.89e-14
## Lagrange-Multiplier test:
##      order   LM  p.value
## [1,]     4 93.4 0.00e+00
## [2,]     8 45.1 1.28e-07
## [3,]    12 28.8 2.47e-03
## [4,]    16 17.6 2.84e-01
## [5,]    20 12.9 8.44e-01
## [6,]    24 10.5 9.88e-01
```

```r
tsplot(epsNice^2)
```



2. interprétez tous les résultats obtenus
3. discutez la différence qu'il y a entre les séries

Exercice 2

Simulez un processus AR(1) stationnaire avec 50 observations. Puis

(a) "testez" graphiquement et testez statistiquement :

1. la stationarité
2. l'autocorrélation
3. la normalité
4. l'hétéroscédasticité

(b) estimez la série simulée et discutez les résultats
(c) reproduire les étapes (a) et (b) avec 5000 observations