Économétrie – TD 3 : Séries temporelles

Fadli Aaron - Université de Tours — Master 1 (2025-2026)

Introduction

Ce document constitue les réponses au **TD 3 – ARIMA & GNP US** basé sur l'énoncé fourni.

Exercice 1 : Modèles ARIMA

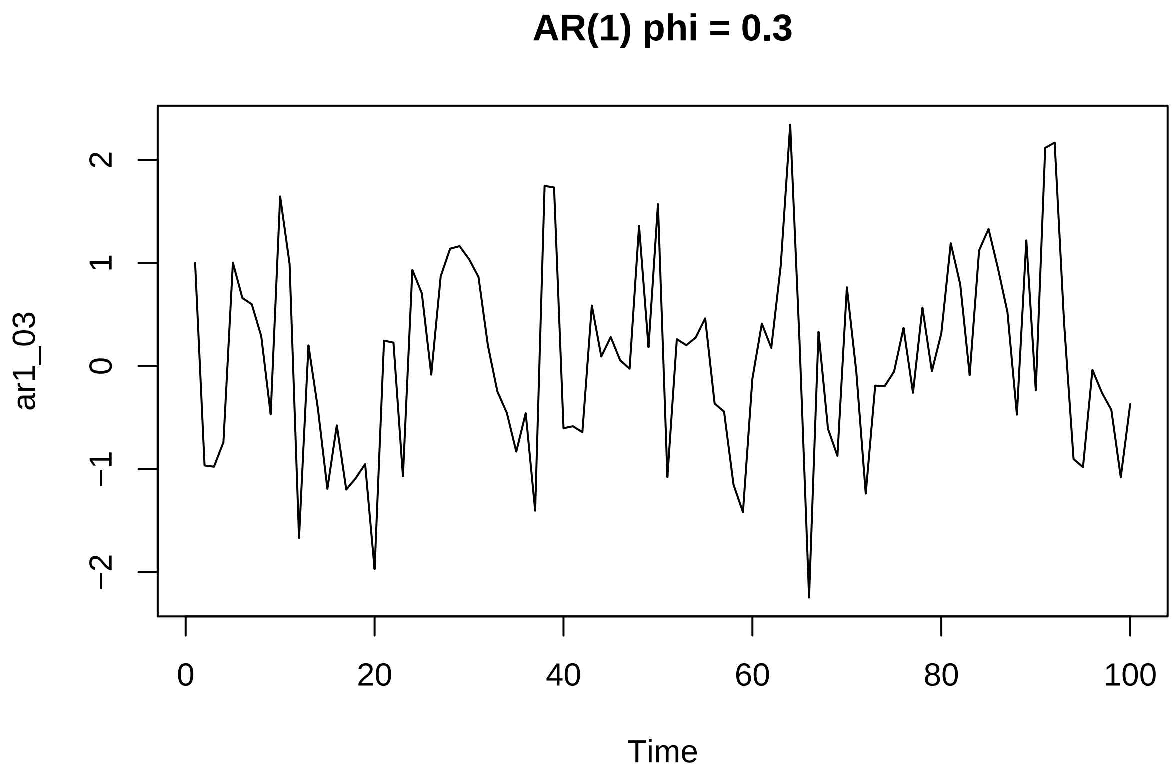
Une série temporelle X_t est ARIMA(p,d,q) si elle vérifie :

$$(1 - \phi_1 B - ... - \phi_p B^p)(1 - B)^d X_t = (1 + \theta_1 B + ... + \theta_q B^q)\varepsilon_t$$

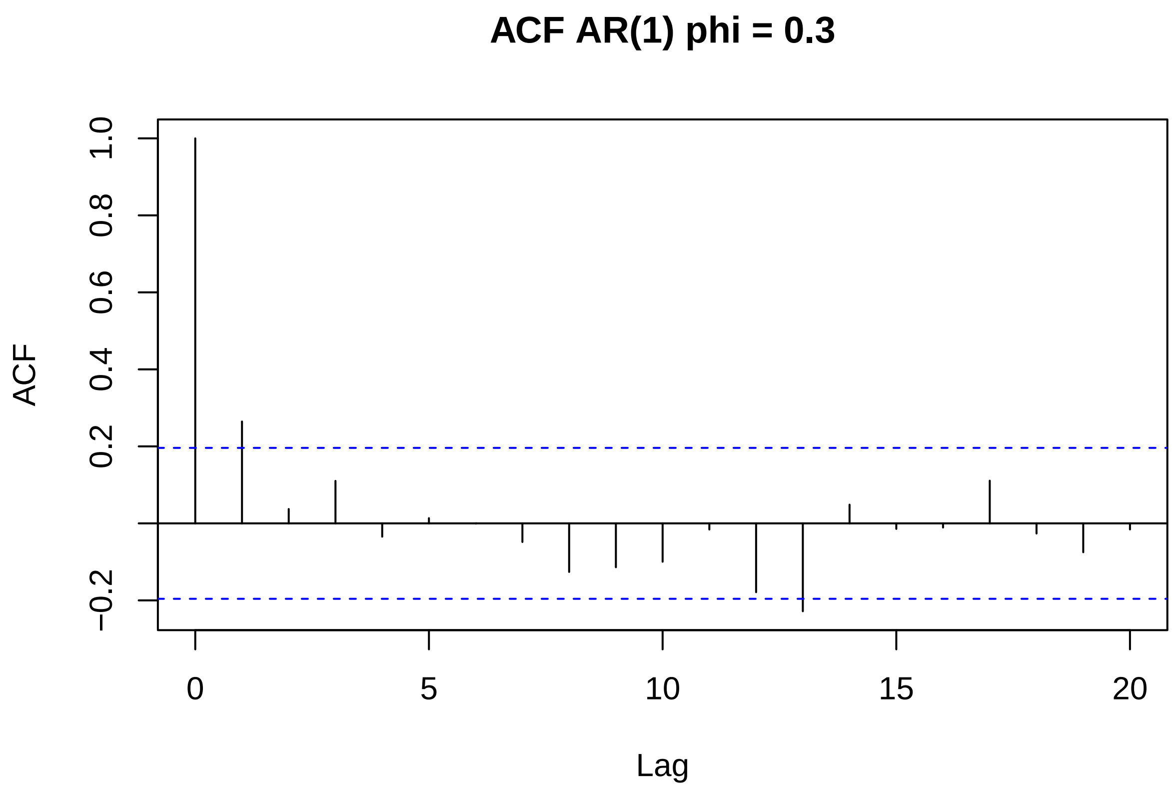
1. Simulation de processus AR(1)

On simule trois processus AR(1) avec paramètres $\phi = 0.3$, $\phi = 0.8$, $\phi = -0.8$.

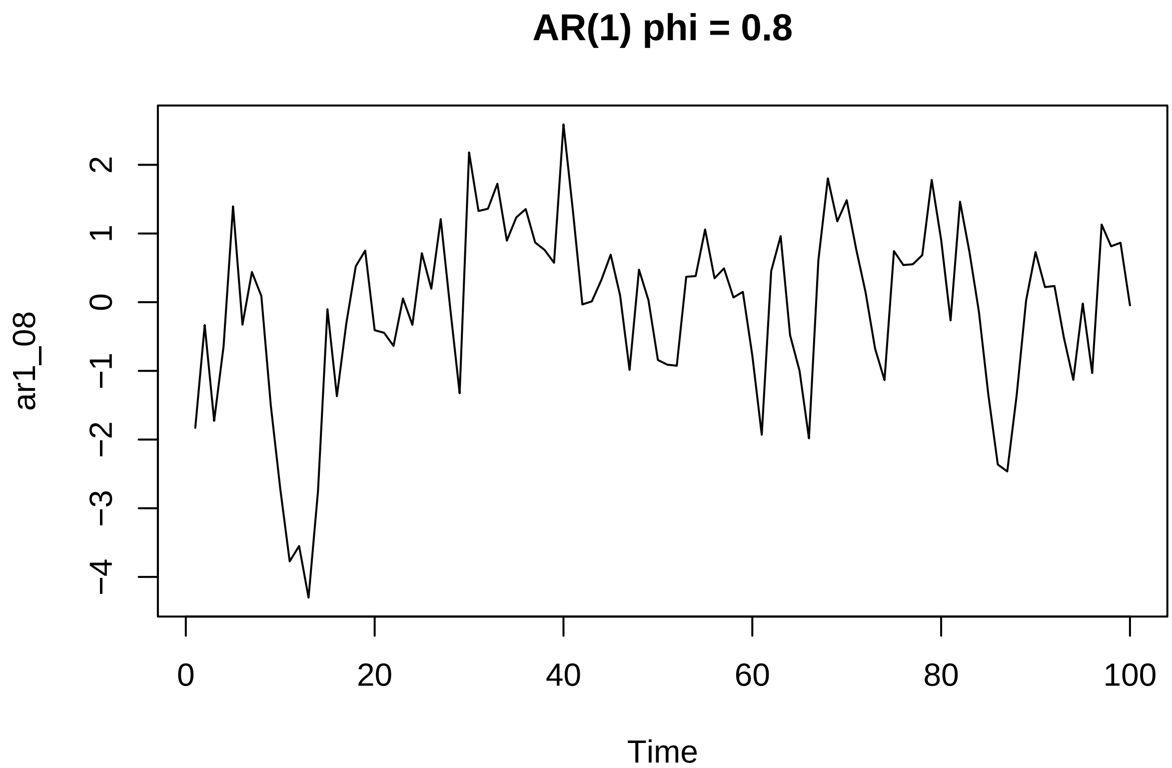
```r
set.seed(123)

# AR(1) phi = 0.3
ar1_03 <- arima.sim(model = list(ar = 0.3), n = 100)
plot(ar1_03, main = "AR(1) phi = 0.3")
```



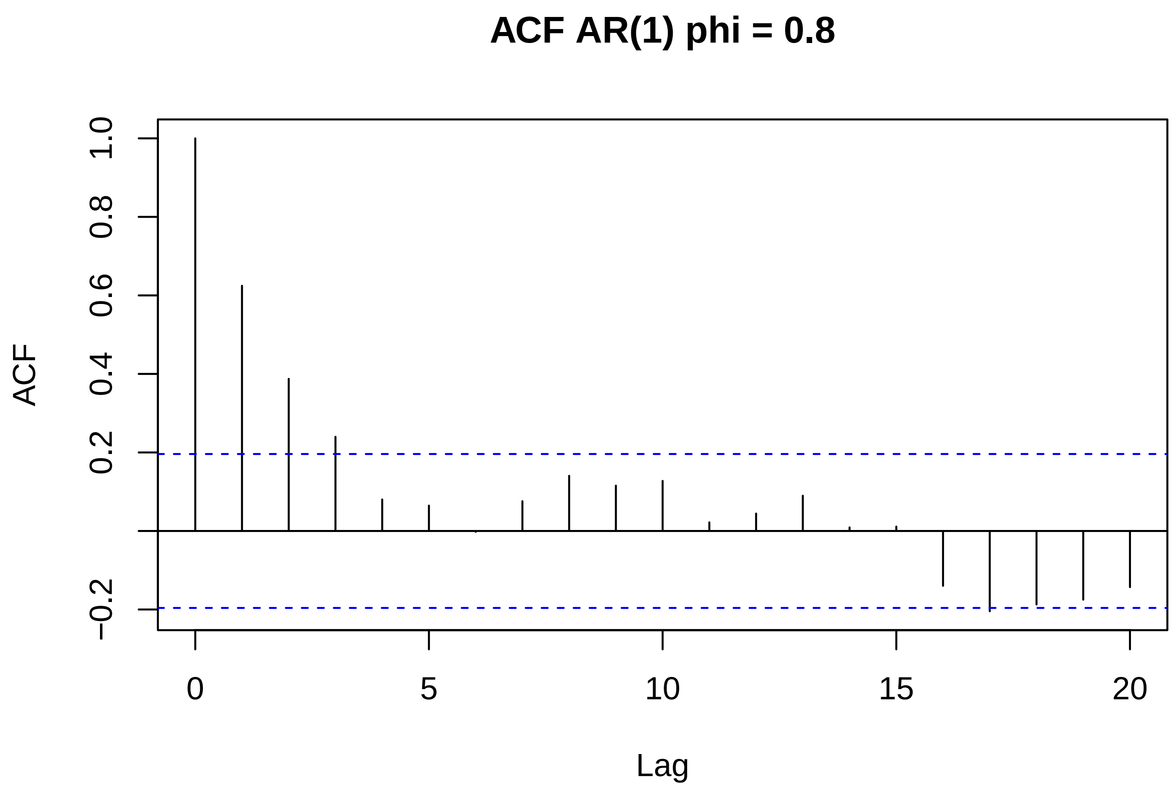

```r
acf(ar1_03, main = "ACF AR(1) phi = 0.3")
```



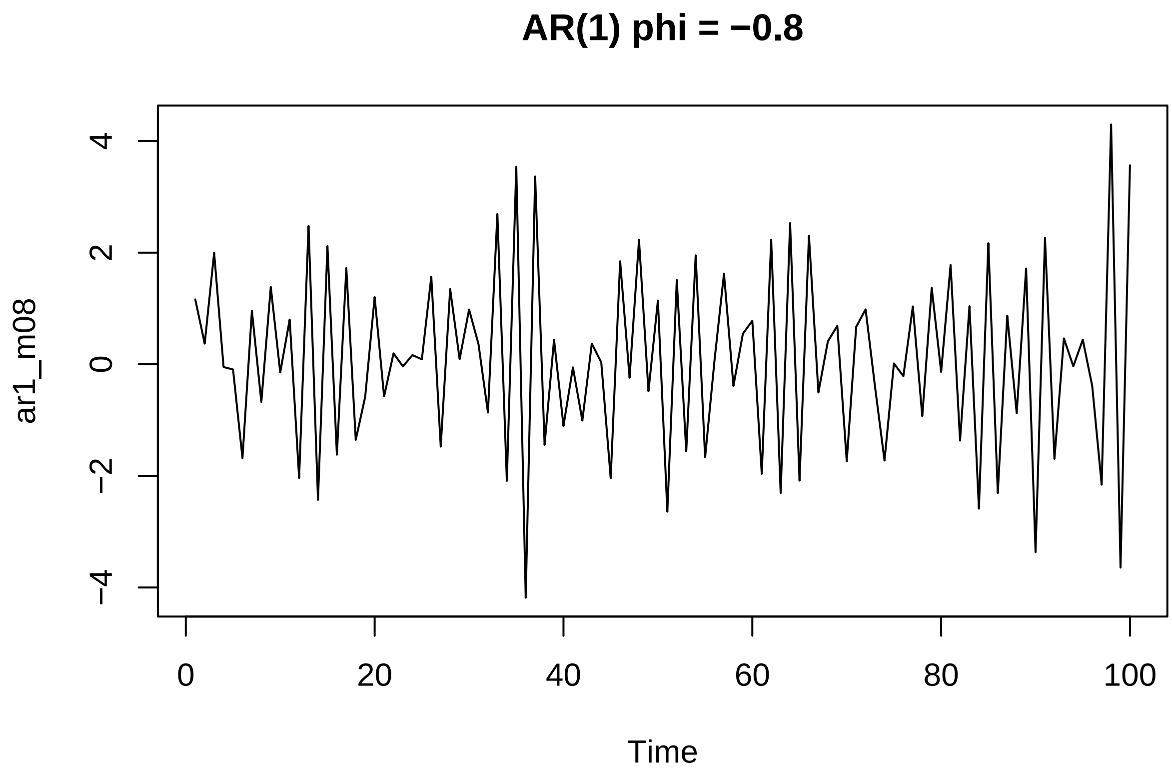

```r
# AR(1) phi = 0.8
ar1_08 <- arima.sim(model = list(ar = 0.8), n = 100)
plot(ar1_08, main = "AR(1) phi = 0.8")
```



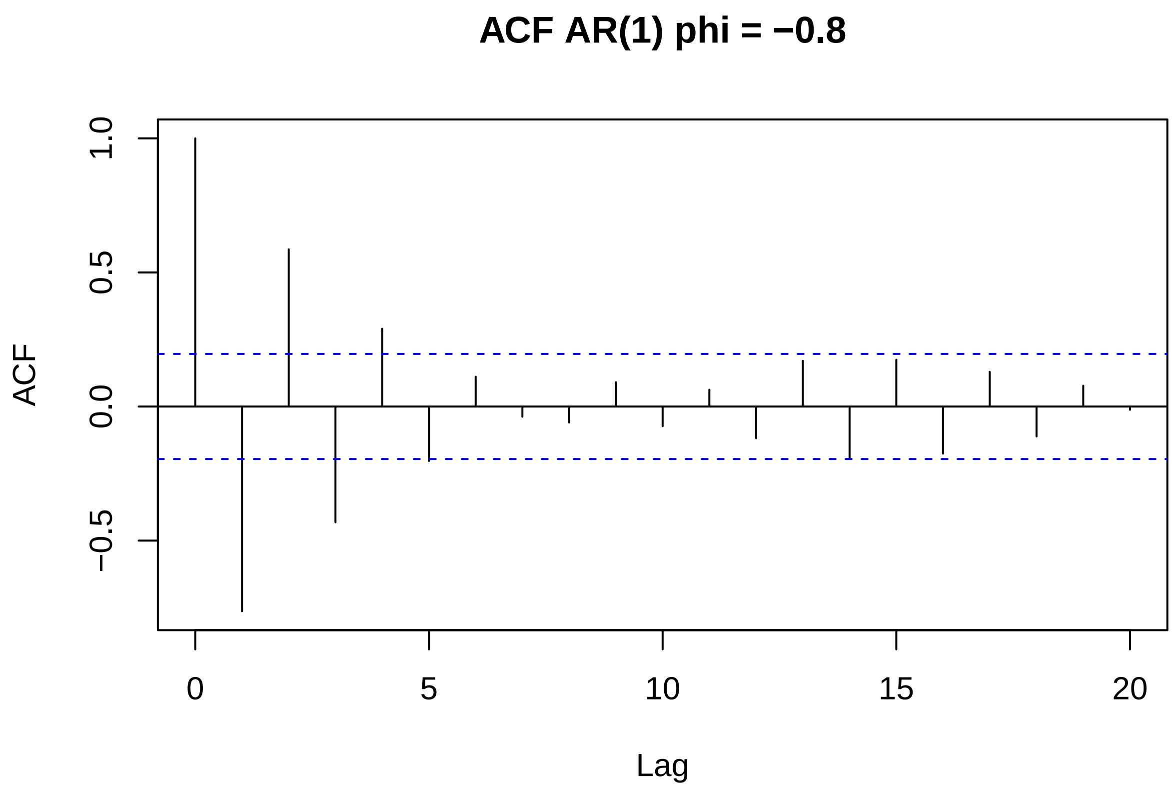

```
acf(ar1_08, main = "ACF AR(1) phi = 0.8")
```




```r
# AR(1) phi = -0.8
ar1_m08 <- arima.sim(model = list(ar = -0.8), n = 100)
plot(ar1_m08, main = "AR(1) phi = -0.8")
```




```
acf(ar1_m08, main = "ACF AR(1) phi = -0.8")
```

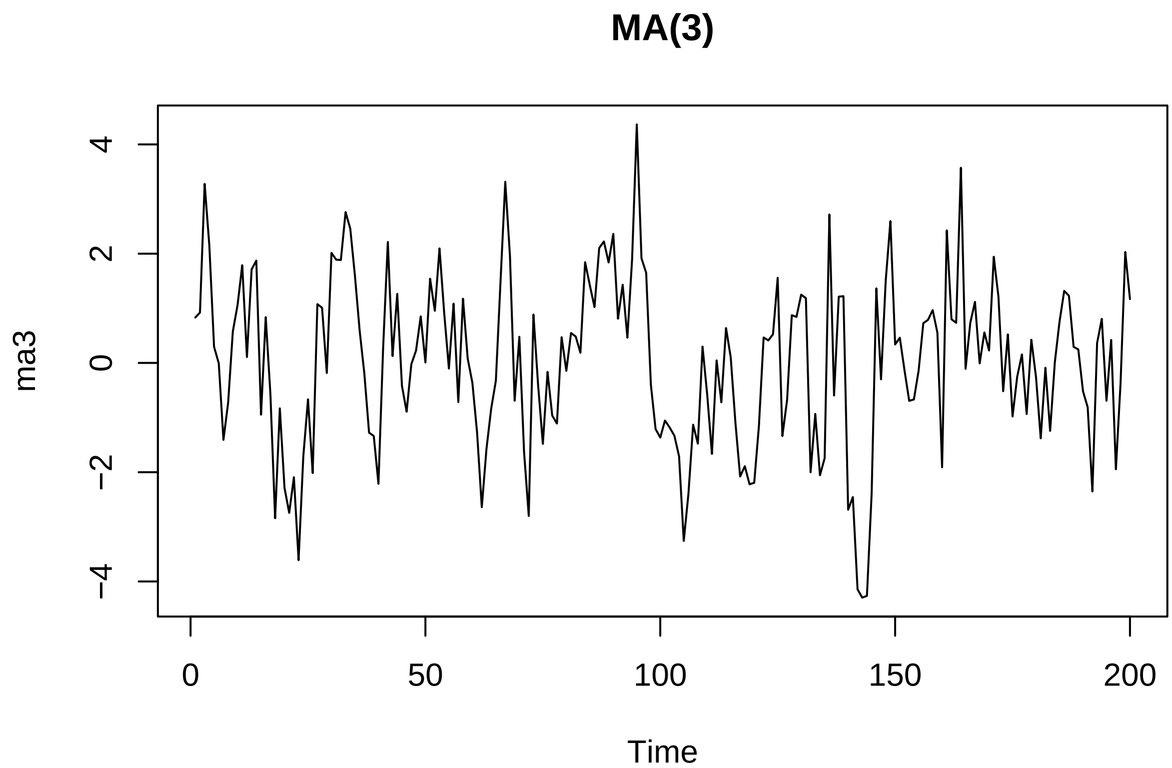


---

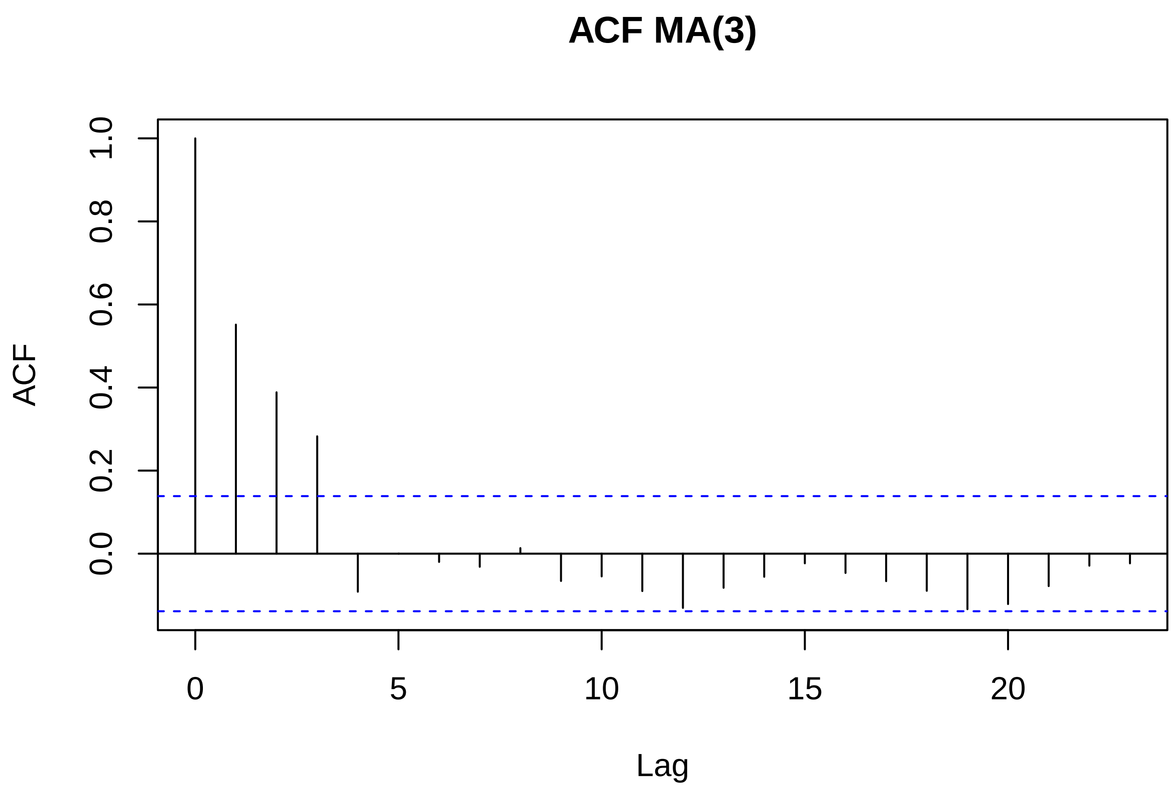
2. Simulation d'un processus MA(3)

Paramètres : $\theta_1 = 0.9$, $\theta_2 = 0.6$, $\theta_3 = 0.9$.

```r
set.seed(123)

ma3 <- arima.sim(model = list(ma = c(0.9, 0.6, 0.9)), n = 200)
plot(ma3, main = "MA(3)")
```




```r
acf(ma3, main = "ACF MA(3)")
```



---

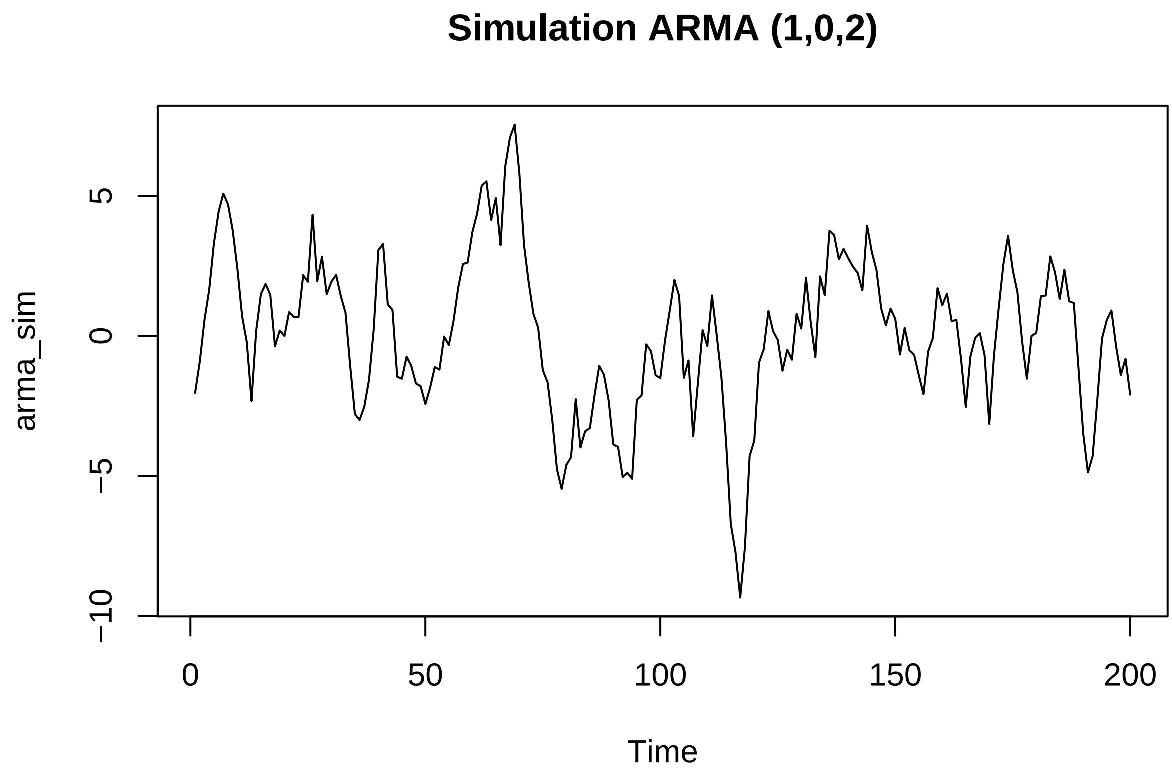
3. Simulation d'un modèle ARMA

Modèle :

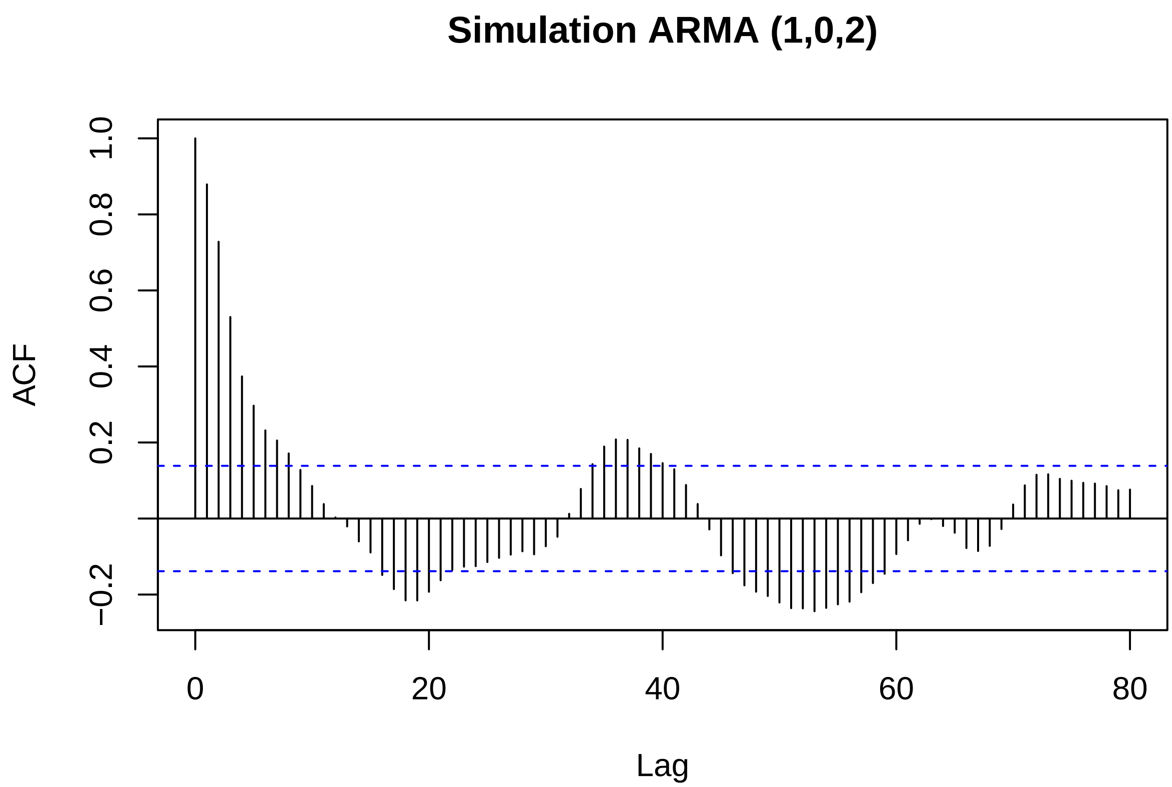
$$(1 - 0.8B)X_t = (1 + 0.3B + 0.6B^2)\varepsilon_t, \quad \varepsilon_t \sim N(0, 1.5)$$

Simulation :

```r
set.seed(123)

arma_sim <- arima.sim(
  model = list(ar = 0.8, ma = c(0.3, 0.6)),
  n = 200,
  sd = sqrt(1.5)
)

plot(arma_sim, main = "Simulation ARMA (1,0,2)")
```




```
acf(arma_sim, main = "Simulation ARMA (1,0,2)", lag.max = 80)
```

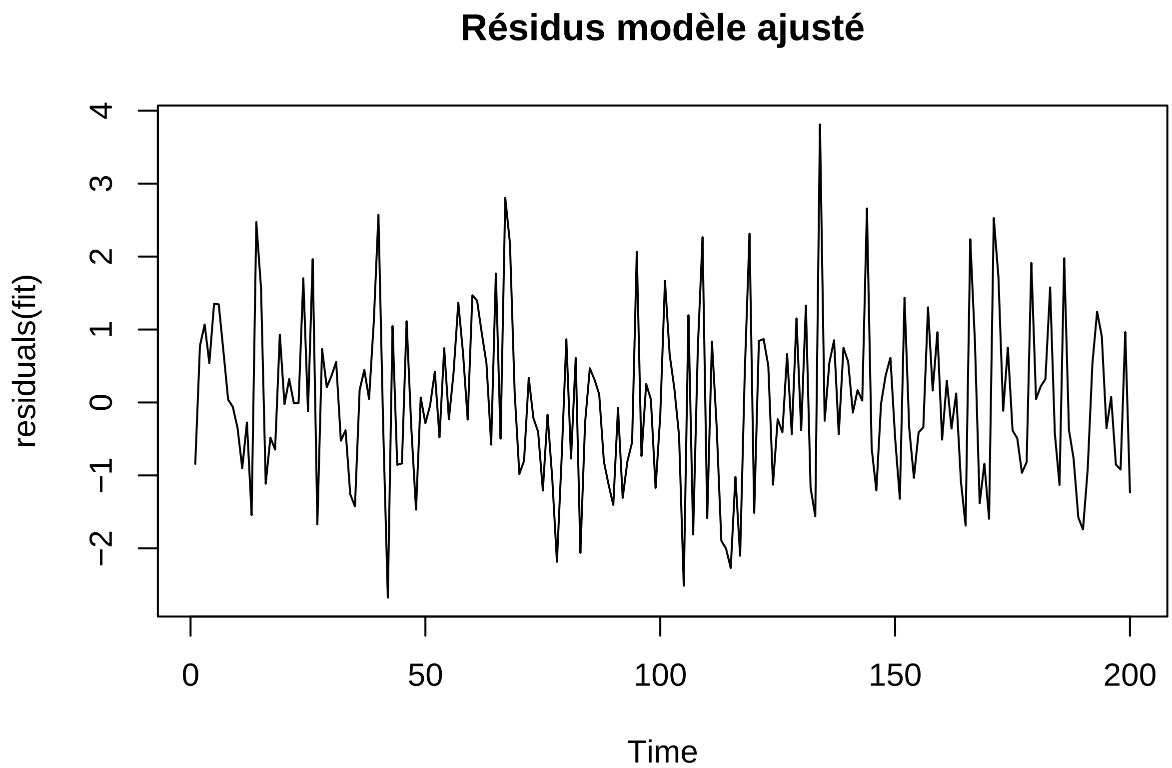


Ajustement ARIMA sur la série simulée

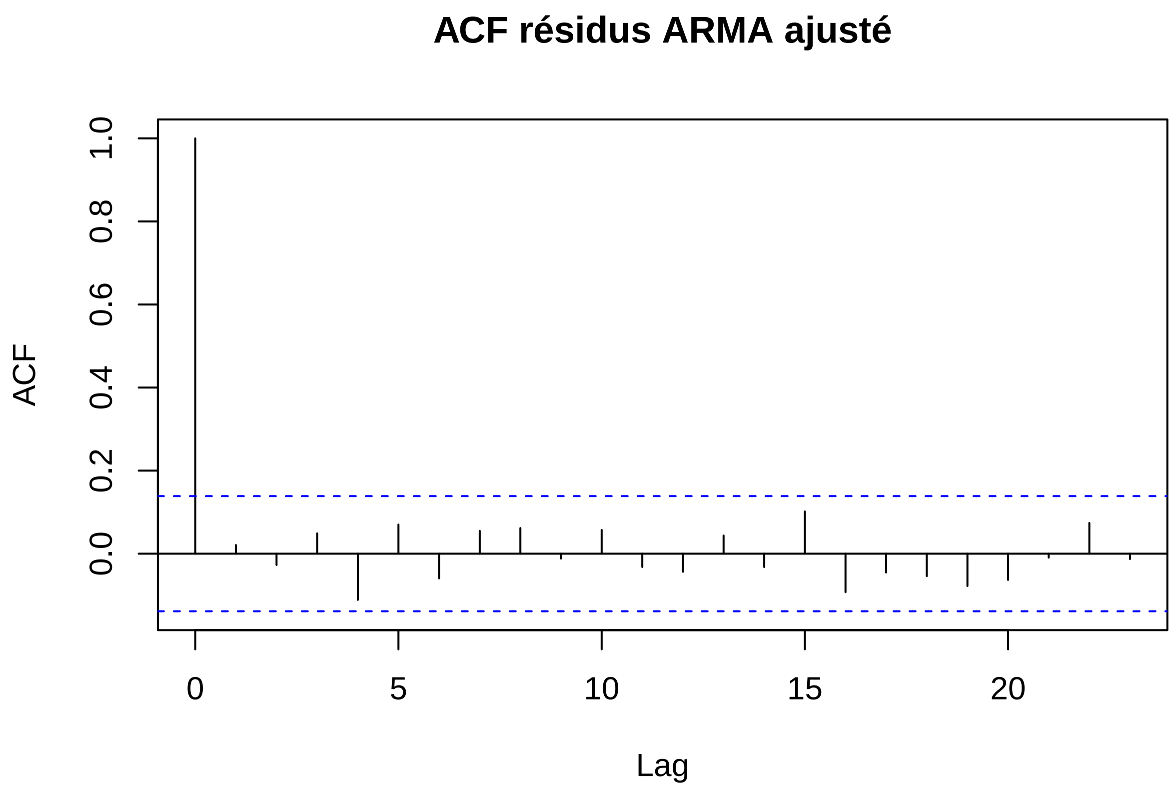
```r
fit <- arima(arma_sim, order = c(1,0,2))
fit
```

```
##
## Call:
## arima(x = arma_sim, order = c(1, 0, 2))
##
## Coefficients:
##          ar1     ma1     ma2  intercept
##       0.7275  0.2827  0.5849    -0.0226
## s.e.  0.0551  0.0635  0.0611     0.5432
##
## sigma^2 estimated as 1.298:  log likelihood = -311.09,  aic = 632.18
```

```r
plot(residuals(fit), type = "l", main = "Résidus modèle ajusté")
```




```
acf(residuals(fit), main = "ACF résidus ARMA ajusté")
```

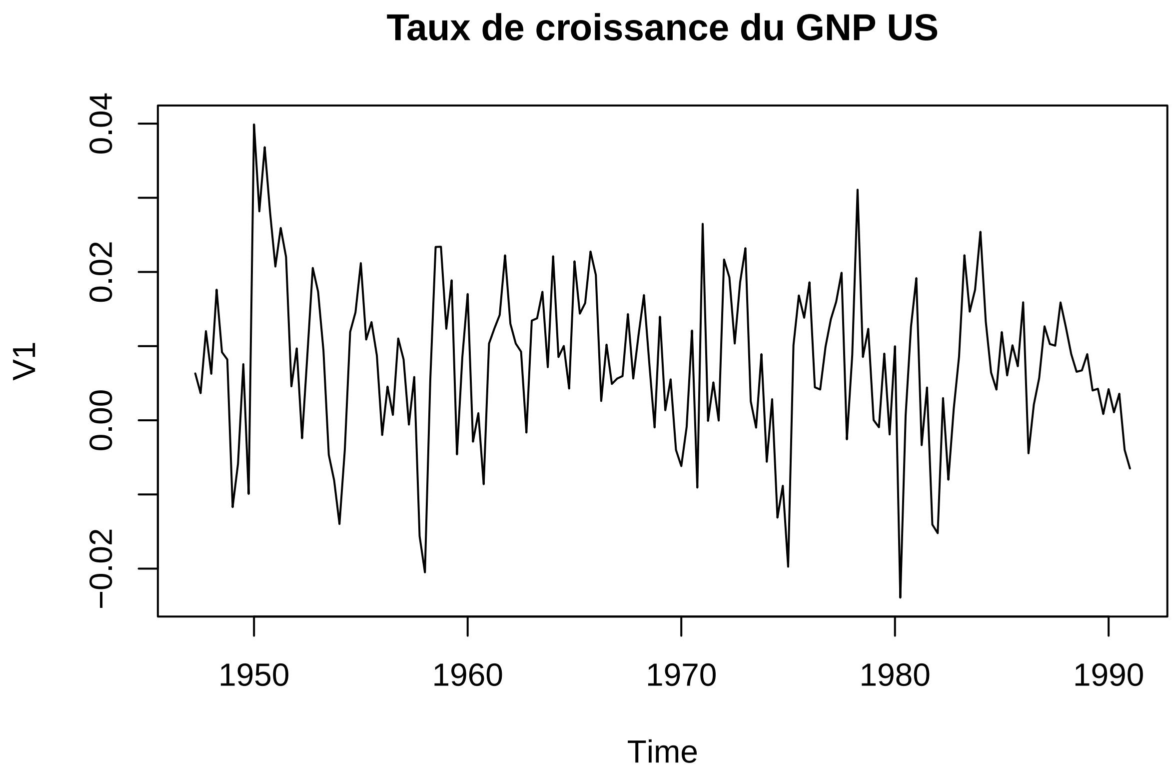


Exercice 2 : GNP US

On étudie le taux de croissance trimestriel du GNP US (1947Q2–1991Q1).

1. Importer les données et créer la série temporelle

```r
gnp <- read.table("data/gnpus2.txt")
ts_gnp <- ts(gnp, start = c(1947, 2), frequency = 4)

plot(ts_gnp, main = "Taux de croissance du GNP US")
```



---

2. Test de blancheur (Ljung–Box)

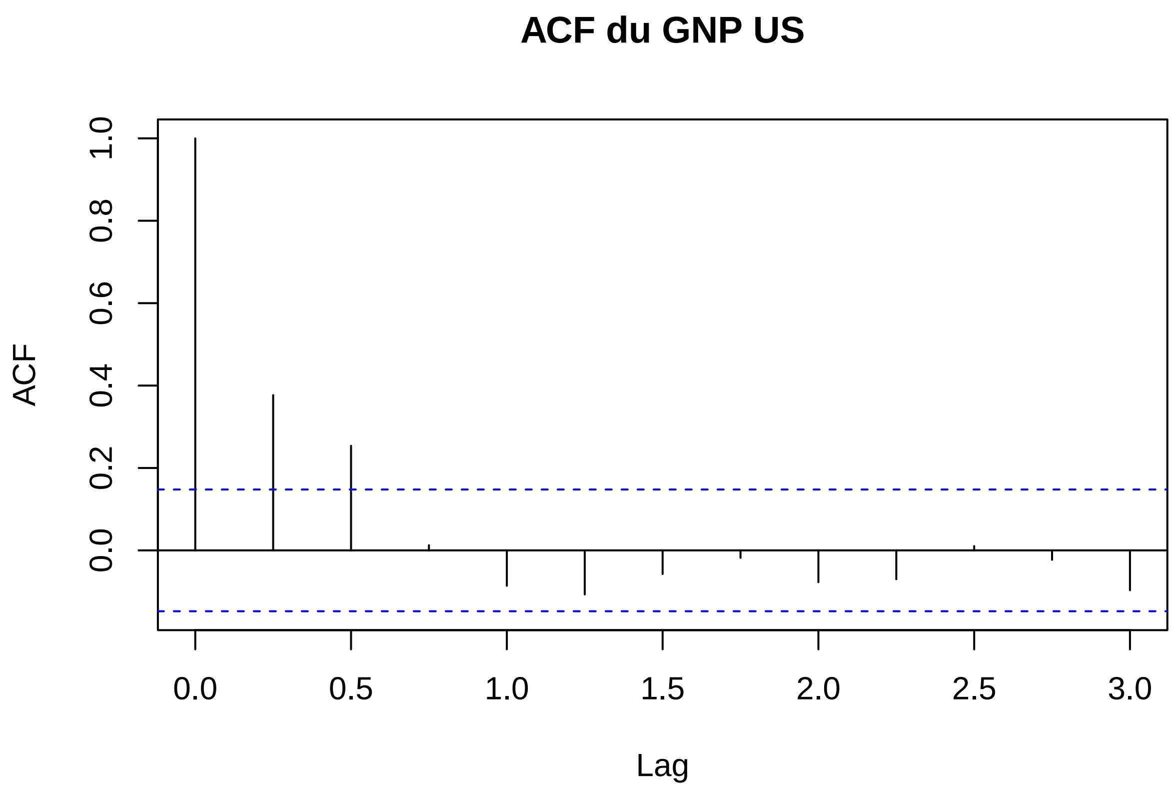
```
Box.test(ts_gnp)
```

```
##
##  Box-Pierce test
##
## data:  ts_gnp
## X-squared = 24.998, df = 1, p-value = 5.74e-07
```

---

3. ACF jusqu'à h = 12

```
acf(ts_gnp, lag.max = 12, main = "ACF du GNP US")
```



---

4. Ajuster un modèle AR

```r
fit_ar <- ar(ts_gnp)
fit_ar
```

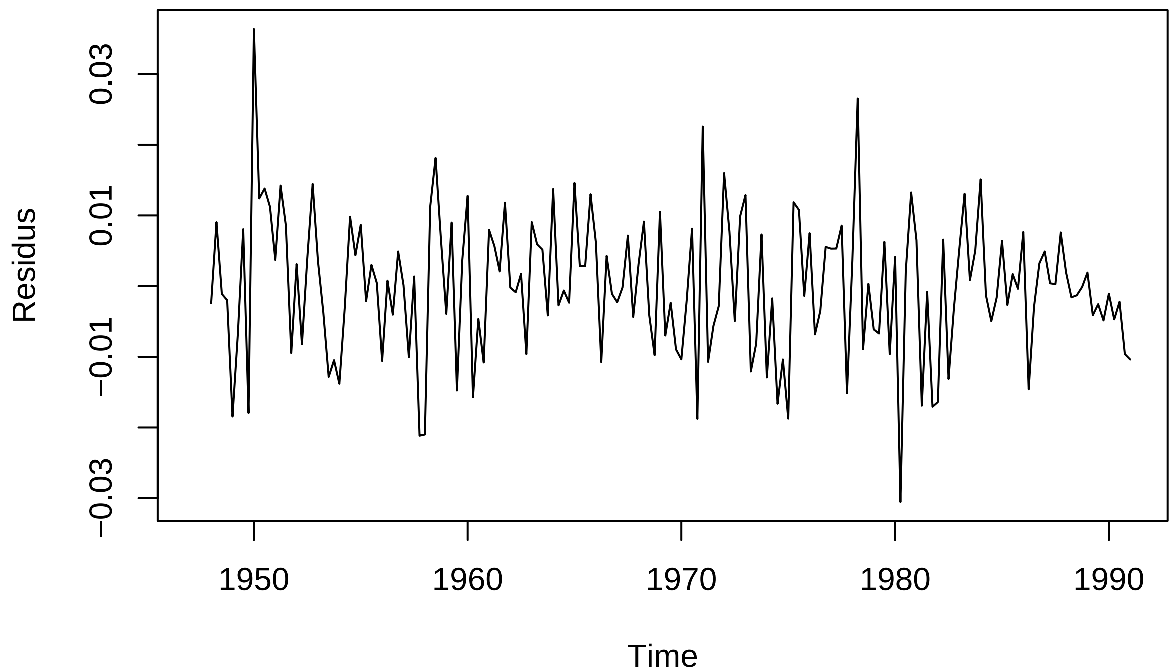
```
##
## Call:
## ar(x = ts_gnp)
##
## Coefficients:
##       1        2        3
##  0.3463   0.1770  -0.1421
##
## Order selected 3  sigma^2 estimated as  9.676e-05
```

---

5. Étude des résidus du modèle AR

```r
res_ar <- residuals(fit_ar)


fit_ar <- ar(ts_gnp)
```

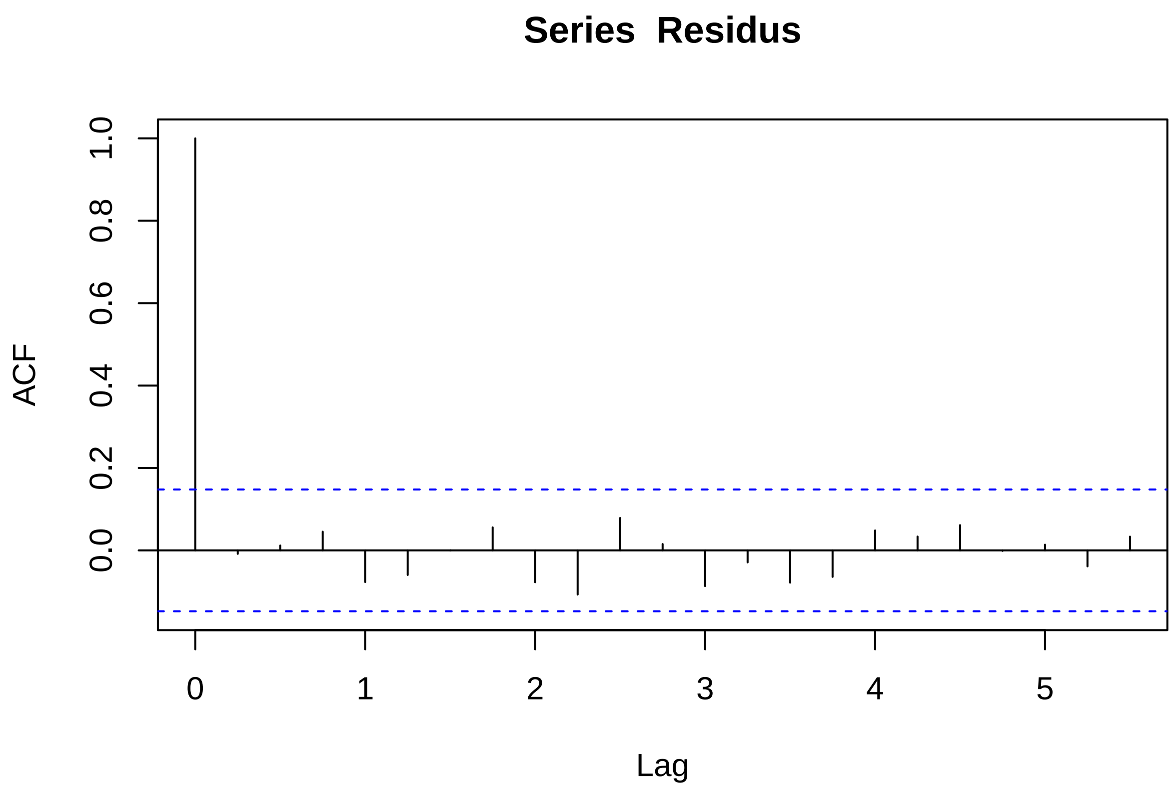
```r
Residus <- fit_ar$resid
plot(Residus)
```




```r
Box.test(Residus)
```

```
##
##  Box-Pierce test
##
## data:  Residus
## X-squared = 0.01235, df = 1, p-value = 0.9115
```

```r
acf(Residus, na.action=na.pass)
```




```
shapiro.test(Residus)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residus
## W = 0.98906, p-value = 0.2031
```

```
#plot(res_ar, type = "l", main = "Résidus (AR)")
#acf(res_ar, main = "ACF résidus AR")
#Box.test(res_ar, lag = 12, type = "Ljung")
#shapiro.test(res_ar)
```

---

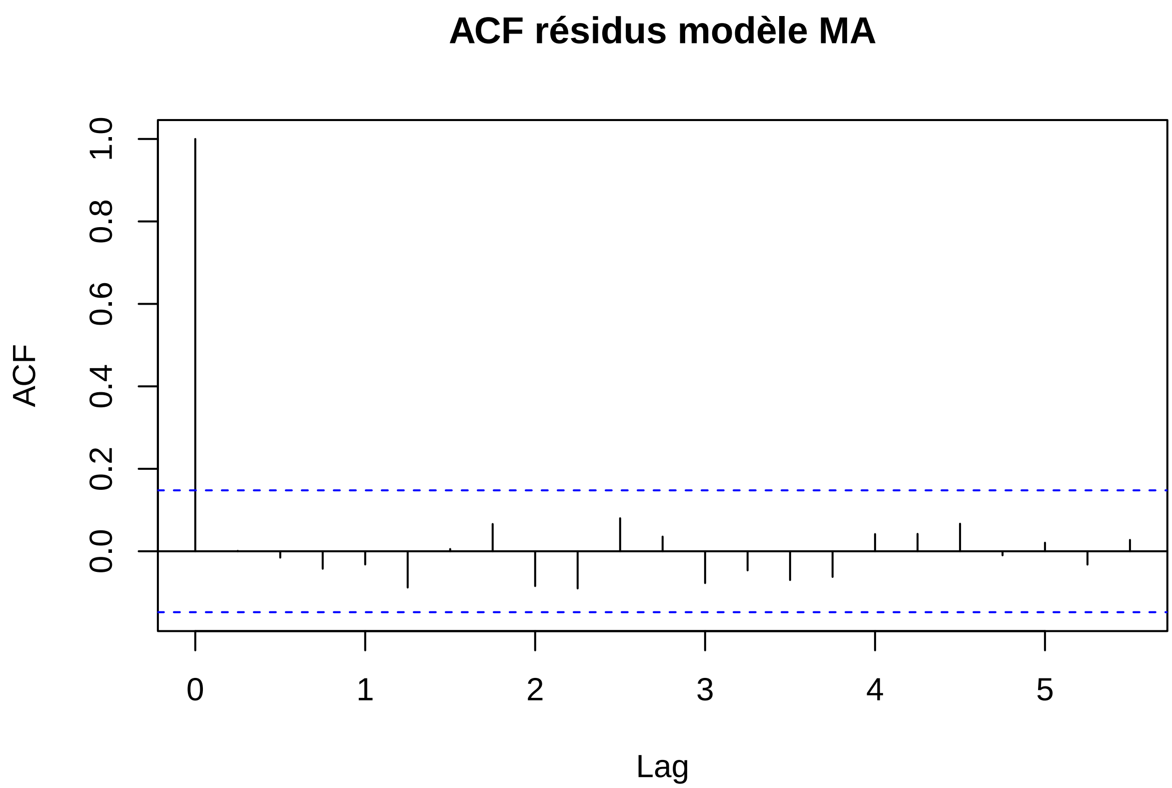
6. Ajuster un modèle MA

```
fit_ma <- arima(ts_gnp, order = c(0,0,3), include.mean = T)
fit_ma
```

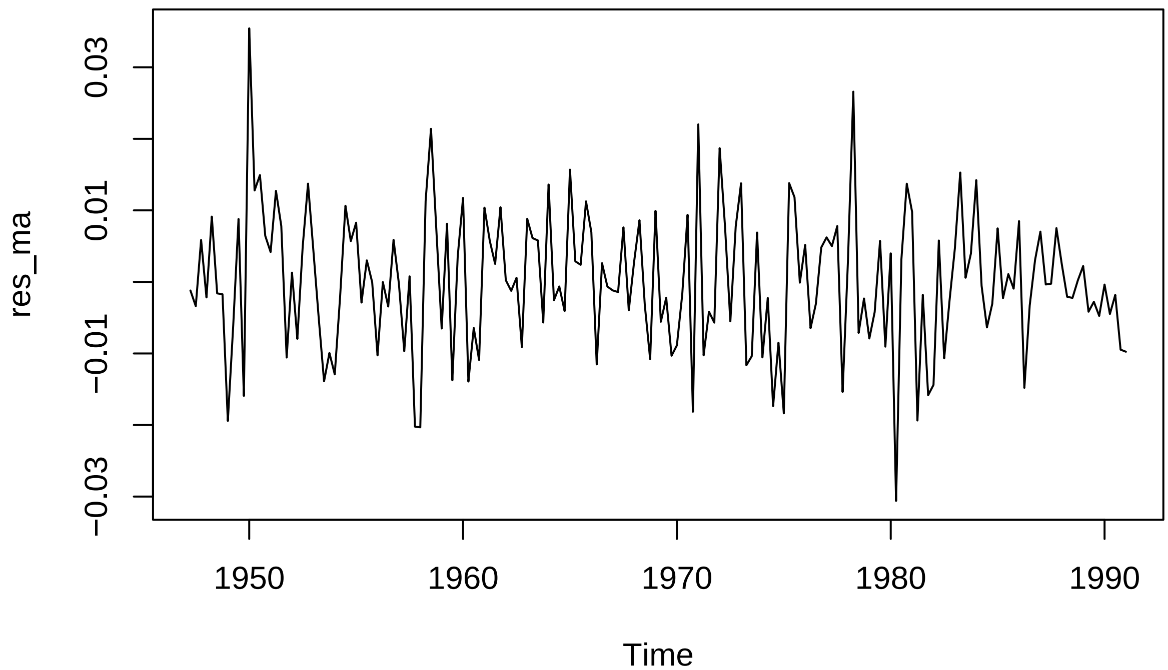
```
##
## Call:
## arima(x = ts_gnp, order = c(0, 0, 3), include.mean = T)
##
## Coefficients:
##          ma1     ma2     ma3  intercept
##       0.3378  0.3384  0.1305     0.0077
```

```
## s.e.   0.0736  0.0822  0.0863     0.0013
##
## sigma^2 estimated as 9.382e-05:  log likelihood = 566.25,  aic = -1122.49
```

```r
res_ma <- residuals(fit_ma)
acf(residuals(fit_ma), na.action=na.pass, main = "ACF résidus modèle MA")
```




```r
plot(res_ma)
```




```r
Box.test(res_ma)
```

```
##
##  Box-Pierce test
##
## data:  res_ma
## X-squared = 0.00013657, df = 1, p-value = 0.9907
```

```r
shapiro.test(res_ma)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ma
## W = 0.99106, p-value = 0.3433
```

---

7. Auto–ARIMA

```r
library(forecast)
```

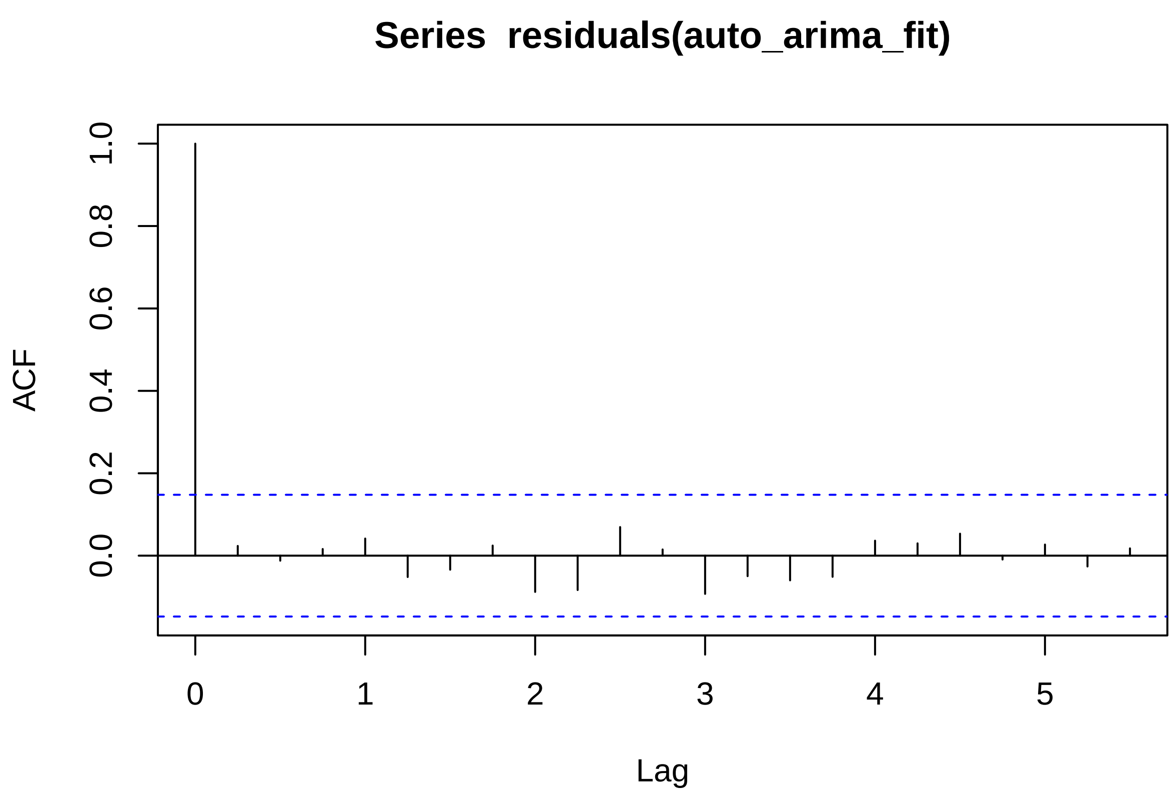
```
## Warning: le package 'forecast' a été compilé avec la version R 4.3.3
```

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

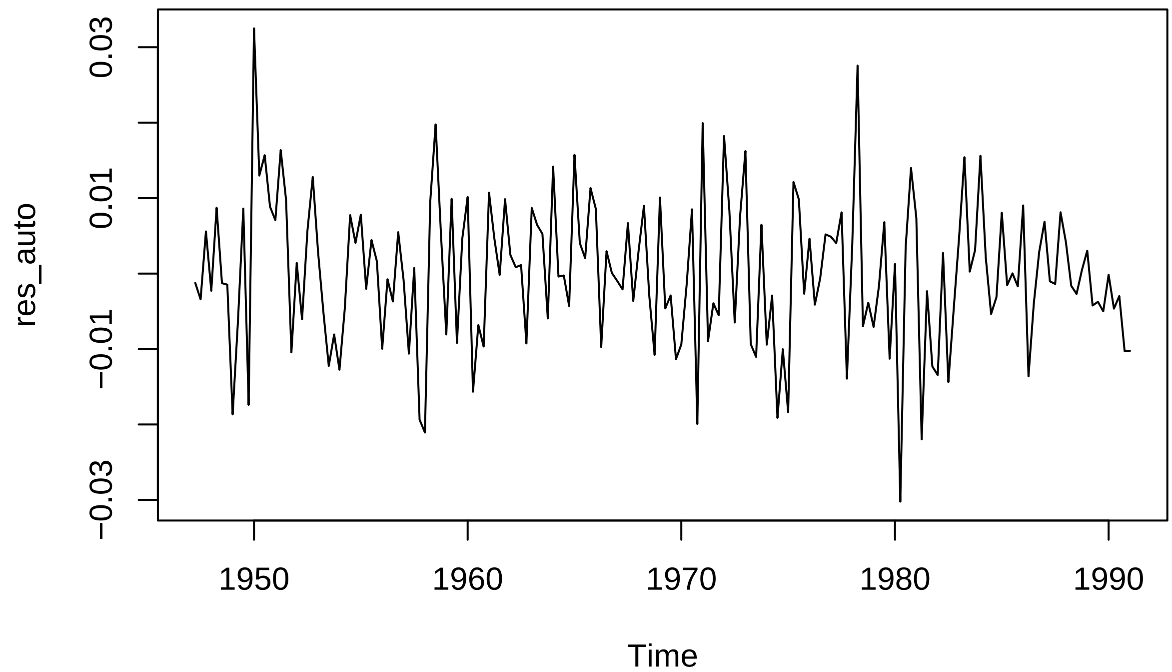
```r
auto_arima_fit <- auto.arima(ts_gnp, d=0, seasonal = F, approximation = F)
auto_arima_fit
```

```
## Series: ts_gnp
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       0.6090  -0.4541  -0.2988  0.5991  0.0077
## s.e.  0.1623   0.1684   0.1379  0.1712  0.0011
##
## sigma^2 = 9.513e-05:  log likelihood = 567.5
## AIC=-1122.99   AICc=-1122.5   BIC=-1103.97
```

```r
res_auto <- residuals(auto_arima_fit)
acf(residuals(auto_arima_fit), na.action=na.pass)
```




```r
plot(res_auto)
```




```r
Box.test(res_auto)
```

```
##
##  Box-Pierce test
##
## data:  res_auto
## X-squared = 0.098042, df = 1, p-value = 0.7542
```

```r
shapiro.test(res_auto)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_auto
## W = 0.99328, p-value = 0.5984
```

---

8. Choix du modèle

Fin du document