tuto_week3

Rodolphe Lajugie

2026-01-21

Contents

Setup

```r
packages <- c(
  "quantmod",   # download time-series
  "tidyverse",  # data manipulation and visualization
  "stargazer",  # publication-ready tables
  "conflicted", # management of function name conflicts across packages
  "moments",    # statistical moments, like skewness and kurtosis
  "lubridate",  # manipulation of data
  "knitr",      # integrate R code with text (e.g. LaTeX, HTML, Markdown)
  "sn",         # simulation of skewed distributions
  "patchwork",  # composition of graphs
  "latex2exp"  # latex
)

# --- from the ones needed extract the ones not installed
to_install <- packages[!packages %in% installed.packages()[, "Package"]]

# --- install the packages
if (length(to_install) > 0) {
  install.packages(to_install)
}

# --- load the packages in our session
invisible(lapply(packages, library, character.only = TRUE))

# --- defining `lag()` from the dplyr package as the preference
conflict_prefer("lag", "dplyr")
```

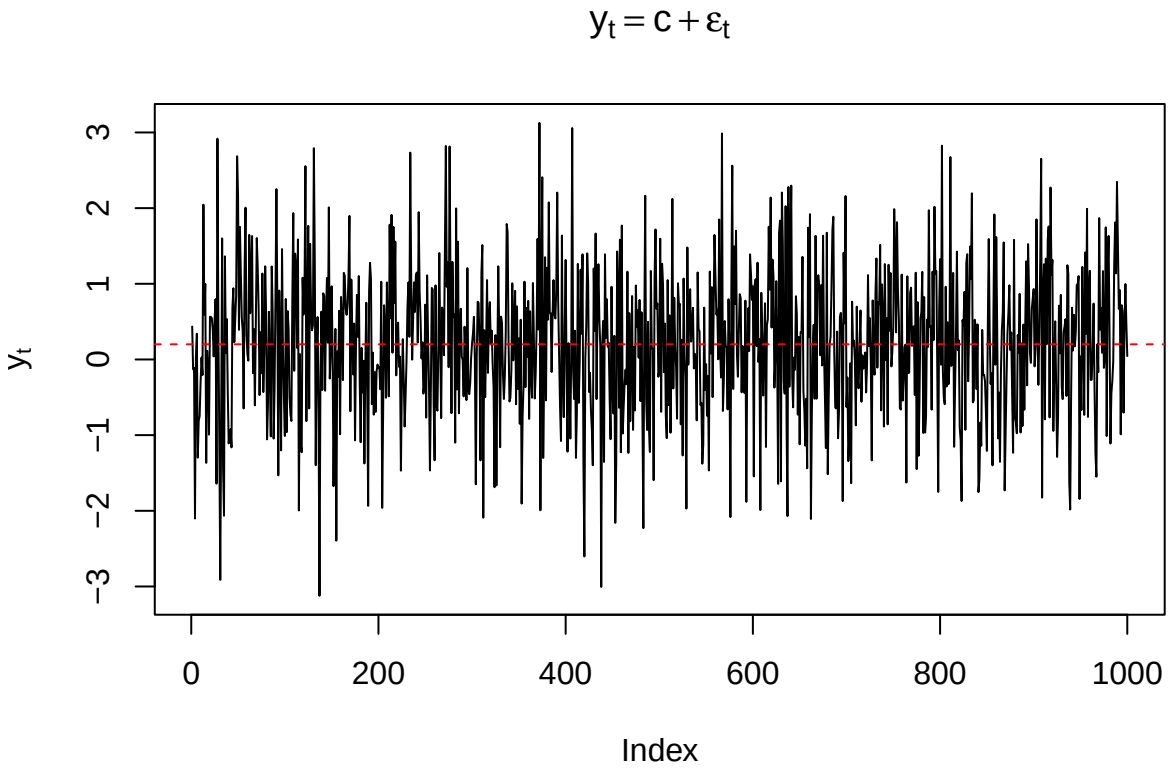
Excercice 1

```r
set.seed(111)

T <- 1000 #nb of obs
c <- 0.2
beta <- 0.1
delta <- 0.05
sigma <- 1 # std normal law
phi1 <- 0.4
phi2 <- 0.3
phi3 <- 0.2
theta1 <- 0.4
theta2 <- 0.3
```

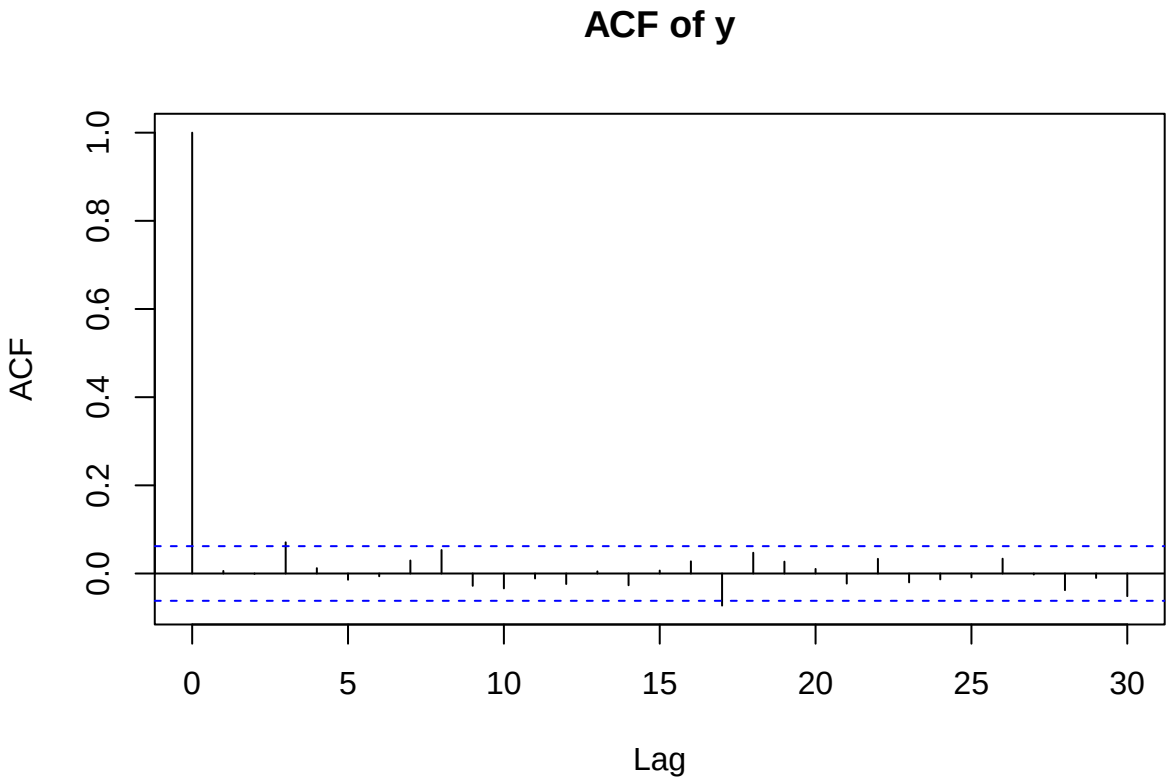
```r
epsilon <- rnorm(T, mean = 0, sd = sigma)
t <- 1:T
```

$y_t = c + \epsilon_t$

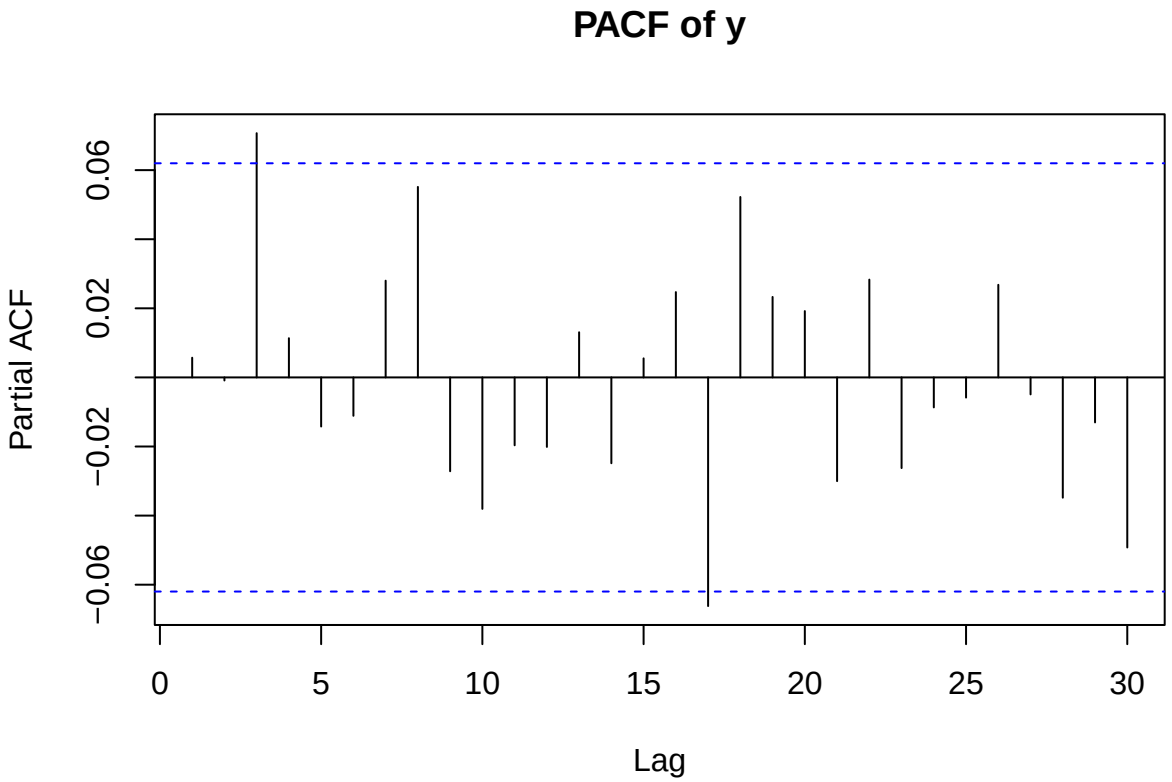
```r
y1 <- c + epsilon

plot(y1, type = "l", main = TeX("$y_t = c + \\epsilon_t$"), ylab=TeX("$y_t$"))
abline(h = c, col = "red", lty = 2)
```




```r
acf(y1, main = "ACF of y")
```

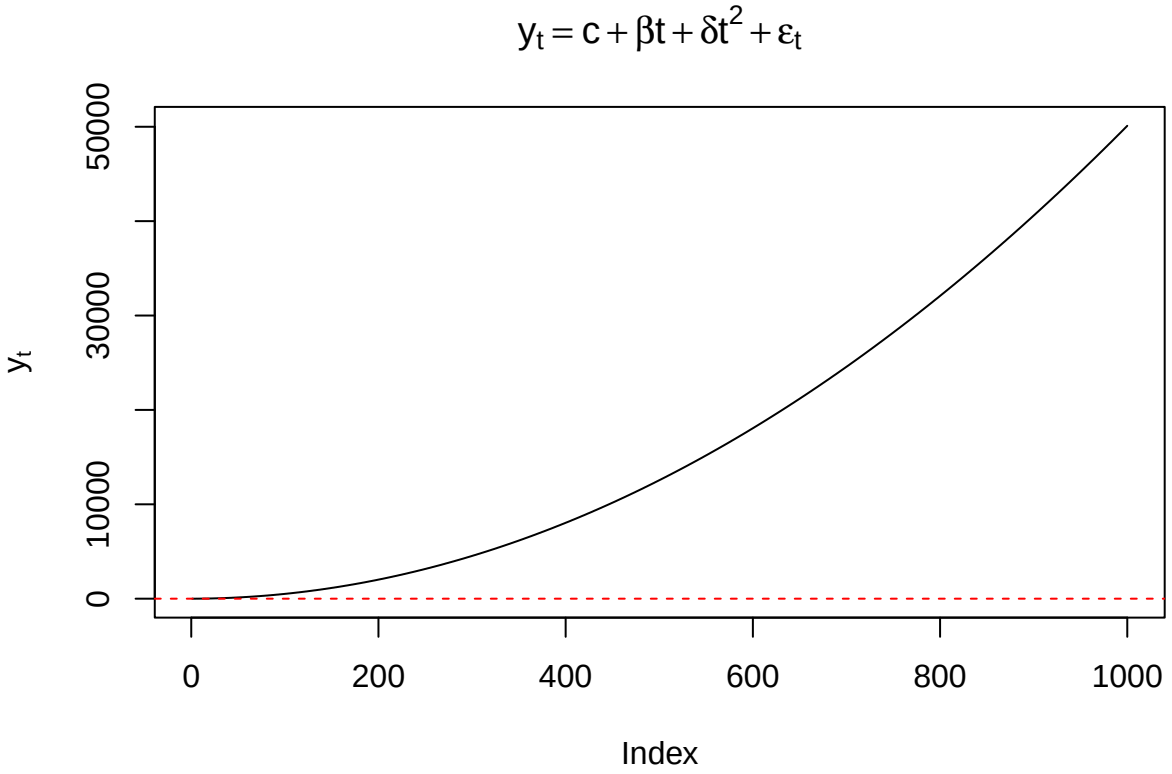



```r
pacf(y1, main = "PACF of y")
```

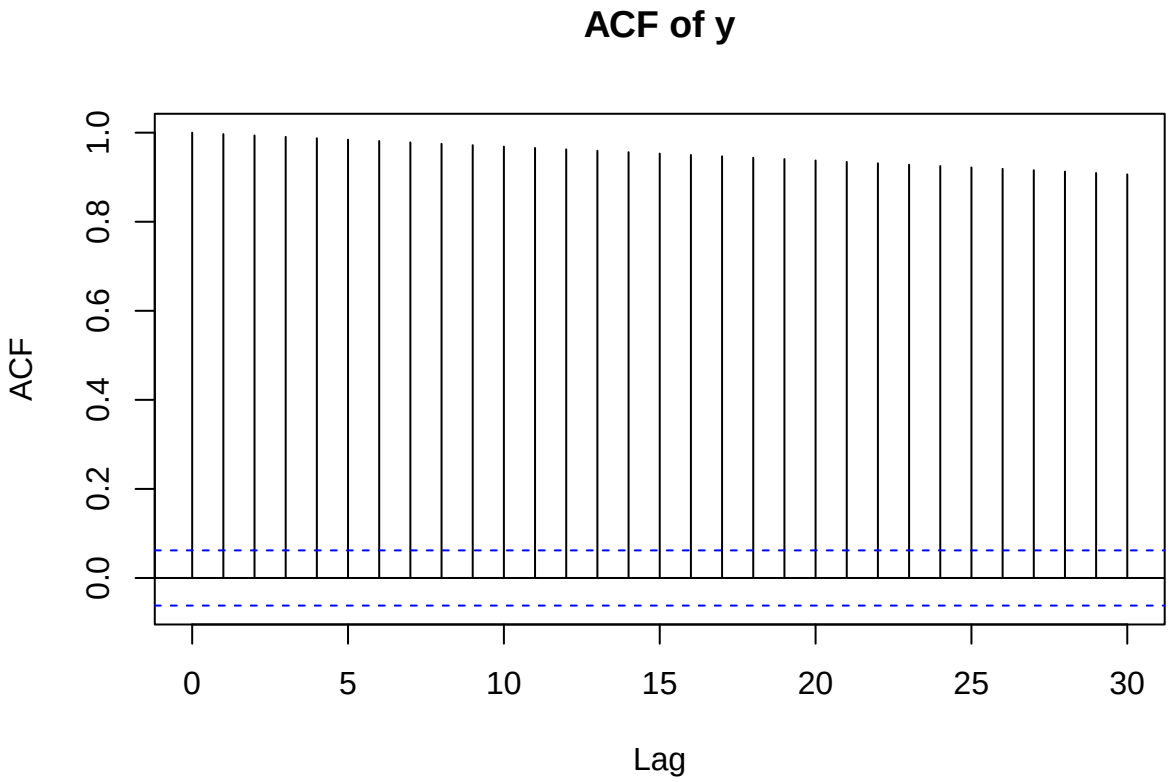


$y_t = c + \beta t + \delta t^2 + \epsilon_t$

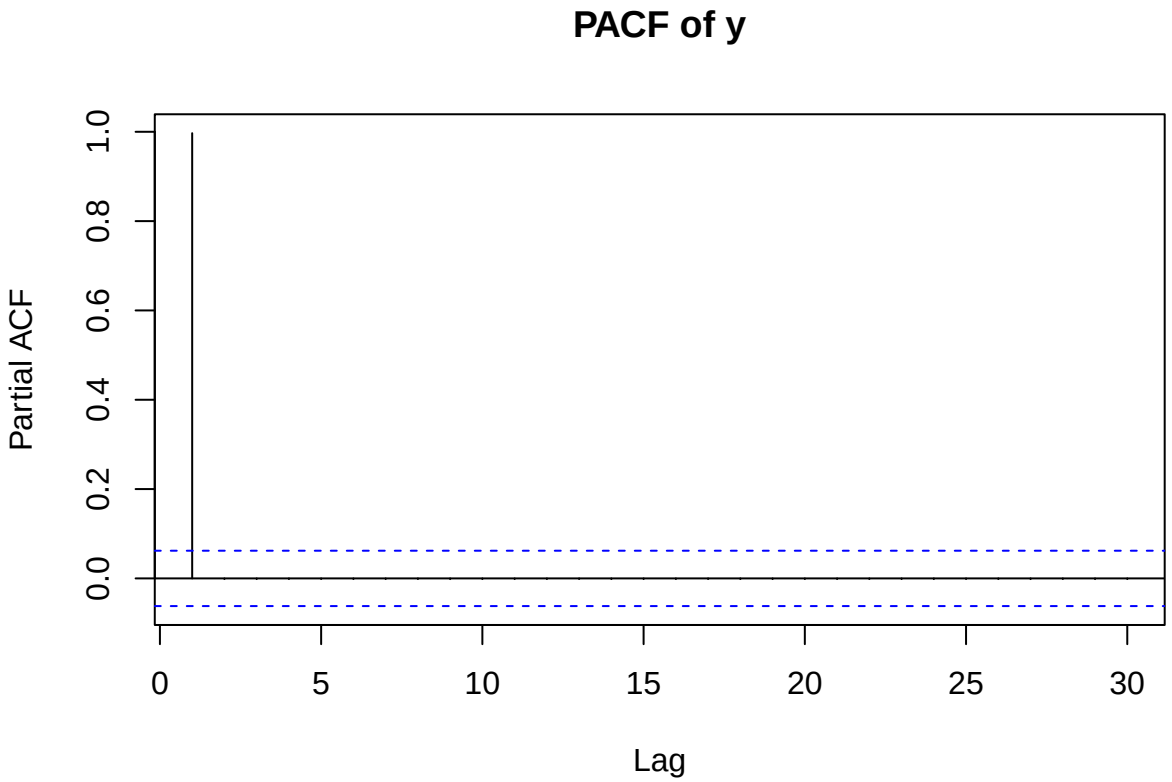
```
y2 <- c + beta*t + delta*t^2 + epsilon
plot(y2, type = "l", main = TeX("$y_t = c + \\beta{t} + \\delta{t^2} + \\epsilon_t$"), ylab=TeX("$y_t$"))
abline(h = c, col = "red", lty = 2)
```




```
acf(y2, main = "ACF of y")
```

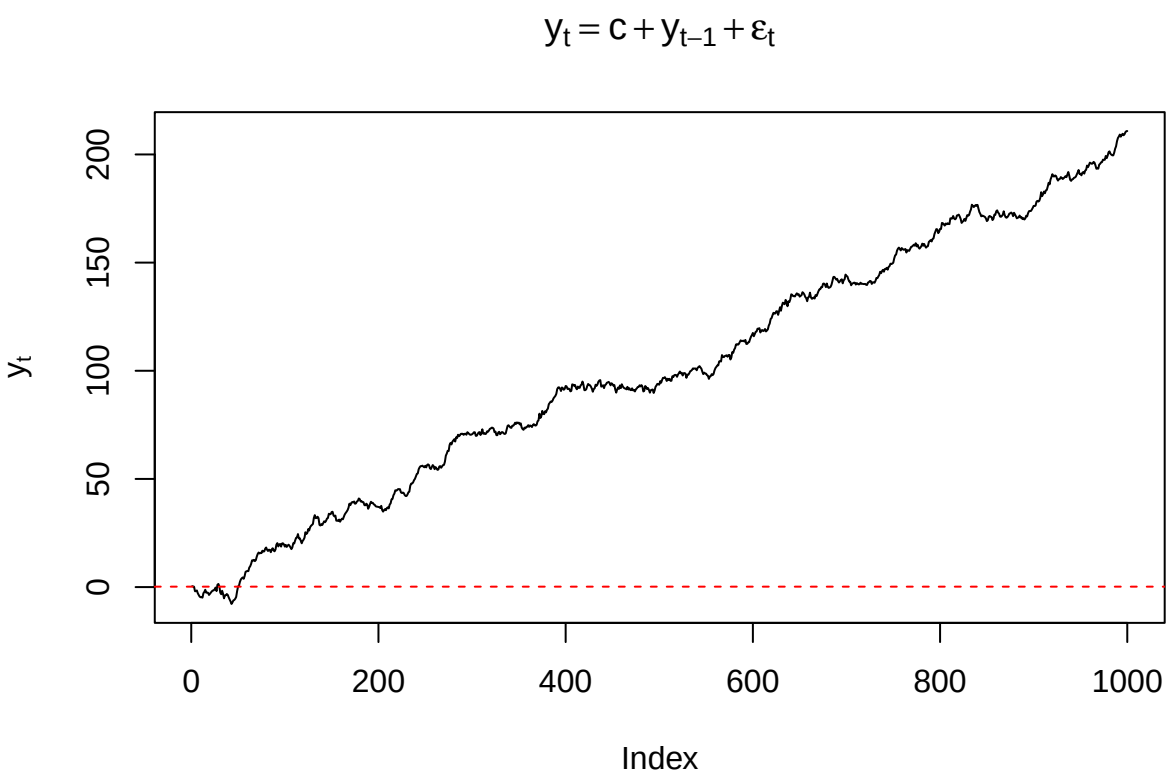



```r
pacf(y2, main = "PACF of y")
```

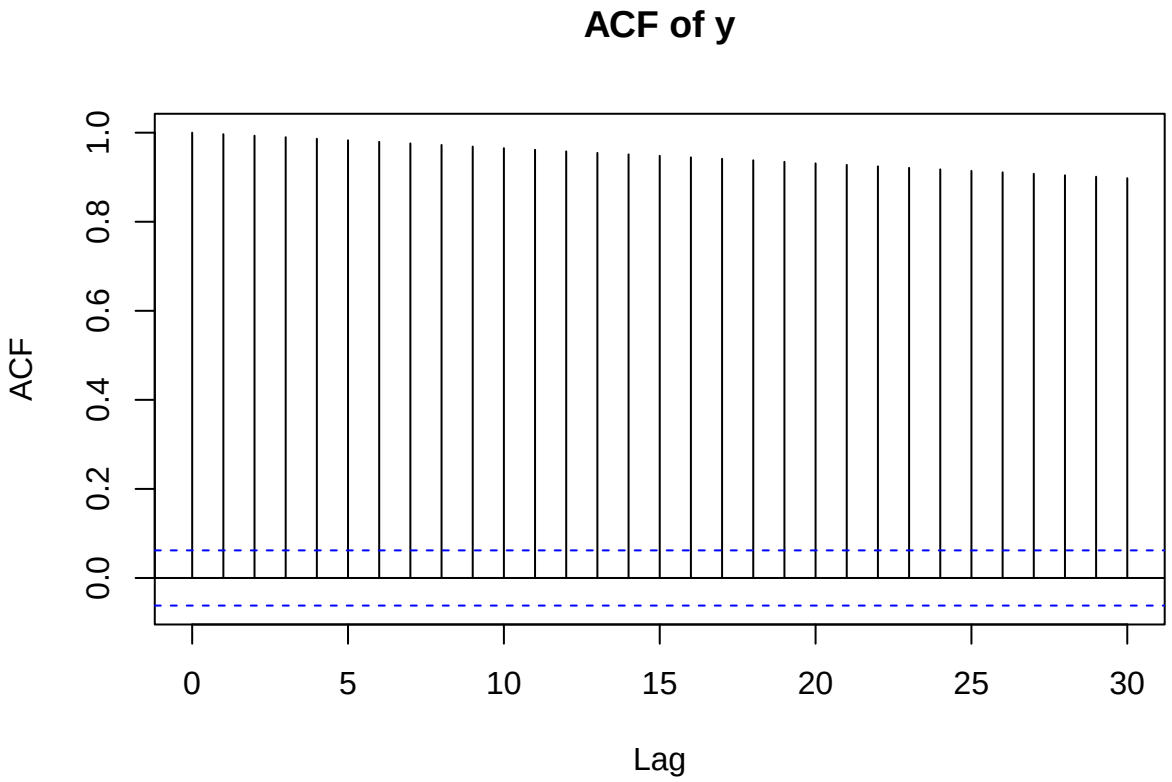


$y_t = c + y_{t-1} + \epsilon_t$

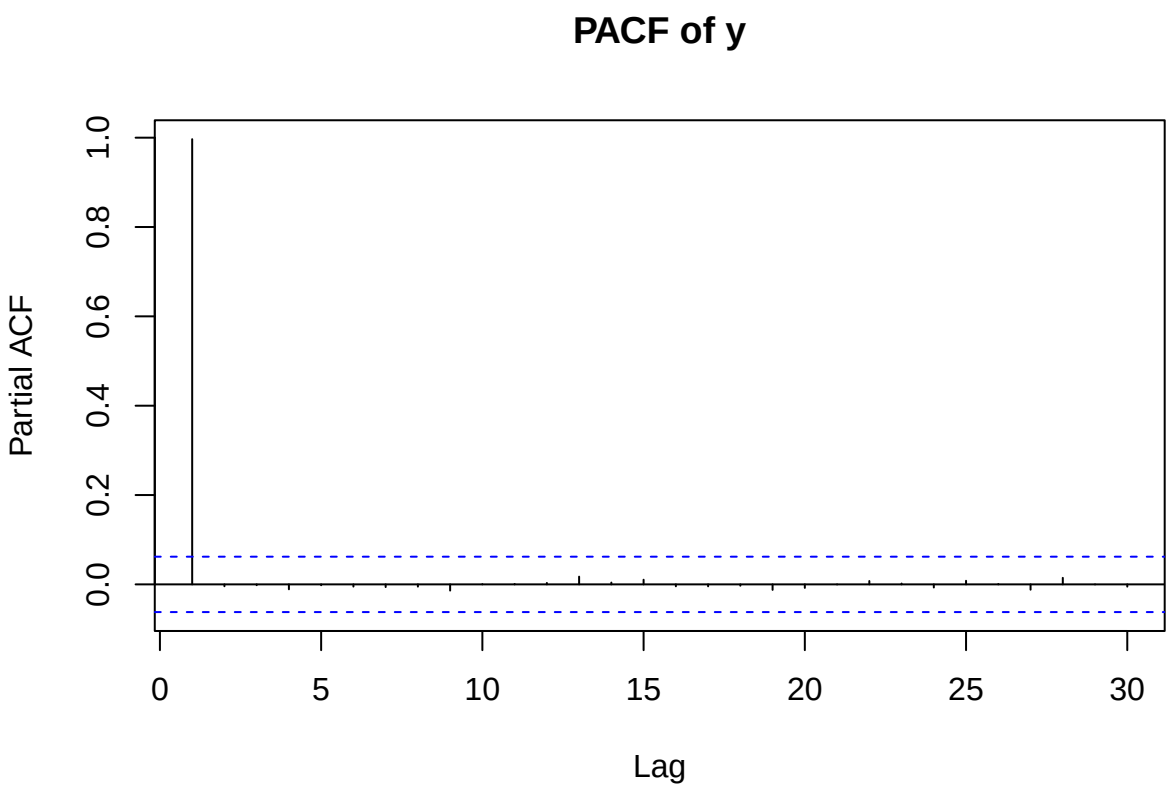
```
y3 = numeric(T)
y3[1] <- c + epsilon[1] # value init

for (i in 2:T) {
  y3[i] <- c + y3[i-1] + epsilon[i]
}
plot(y3, type = "l", main = TeX("$y_t = c + y_{t-1} + \\epsilon_t$"), ylab=TeX("$y_t$"))
abline(h = c, col = "red", lty = 2)
```




```
acf(y3, main = "ACF of y")
```

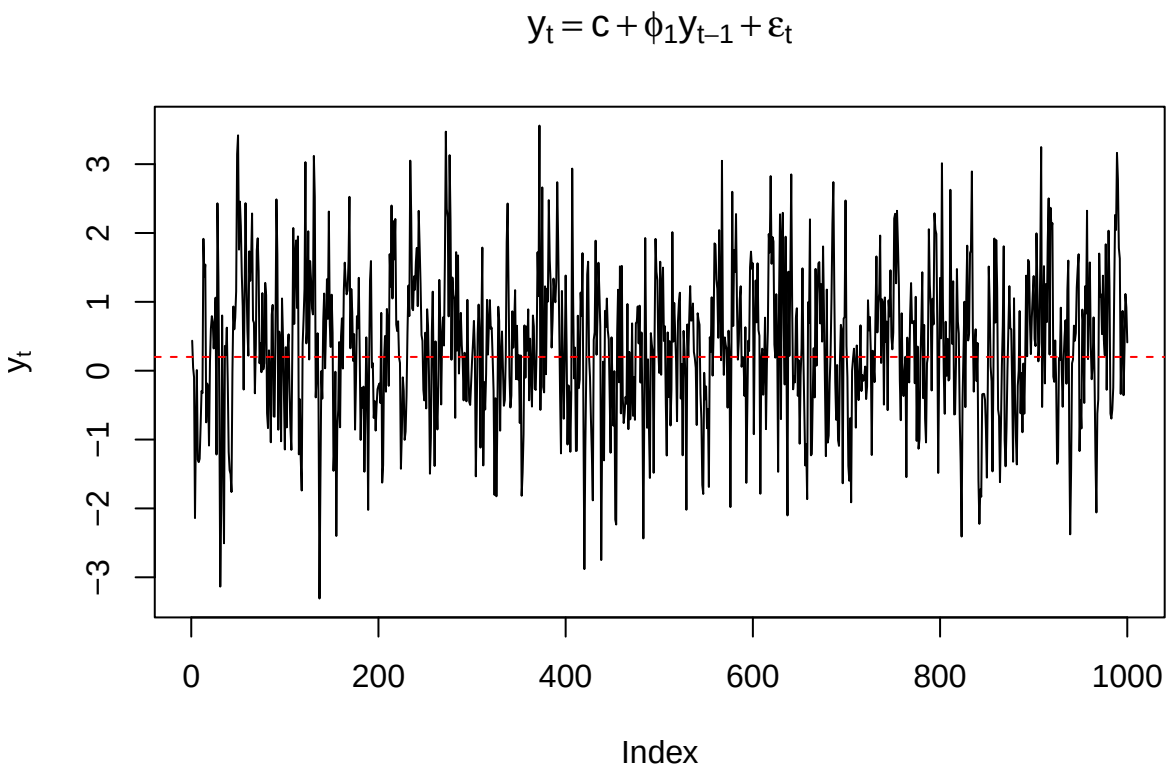



```
pacf(y3, main = "PACF of y")
```

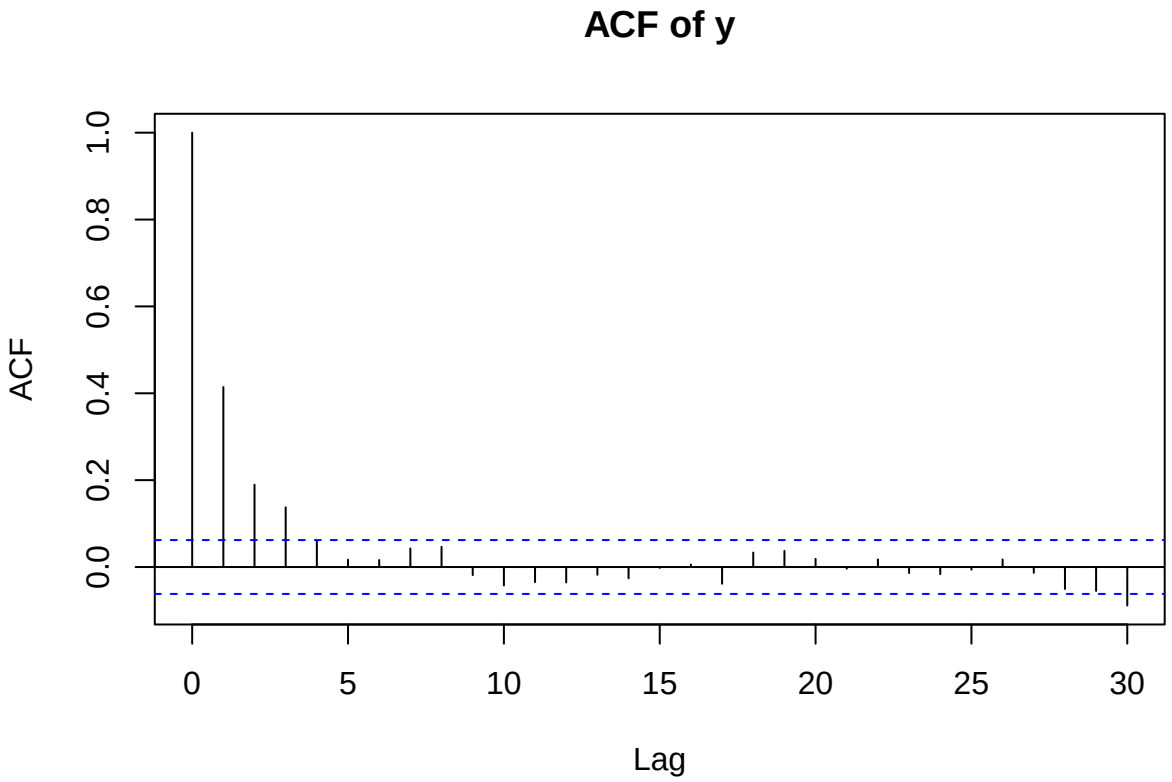


$y_t = c + \phi_1 y_{t-1} + \epsilon_t$

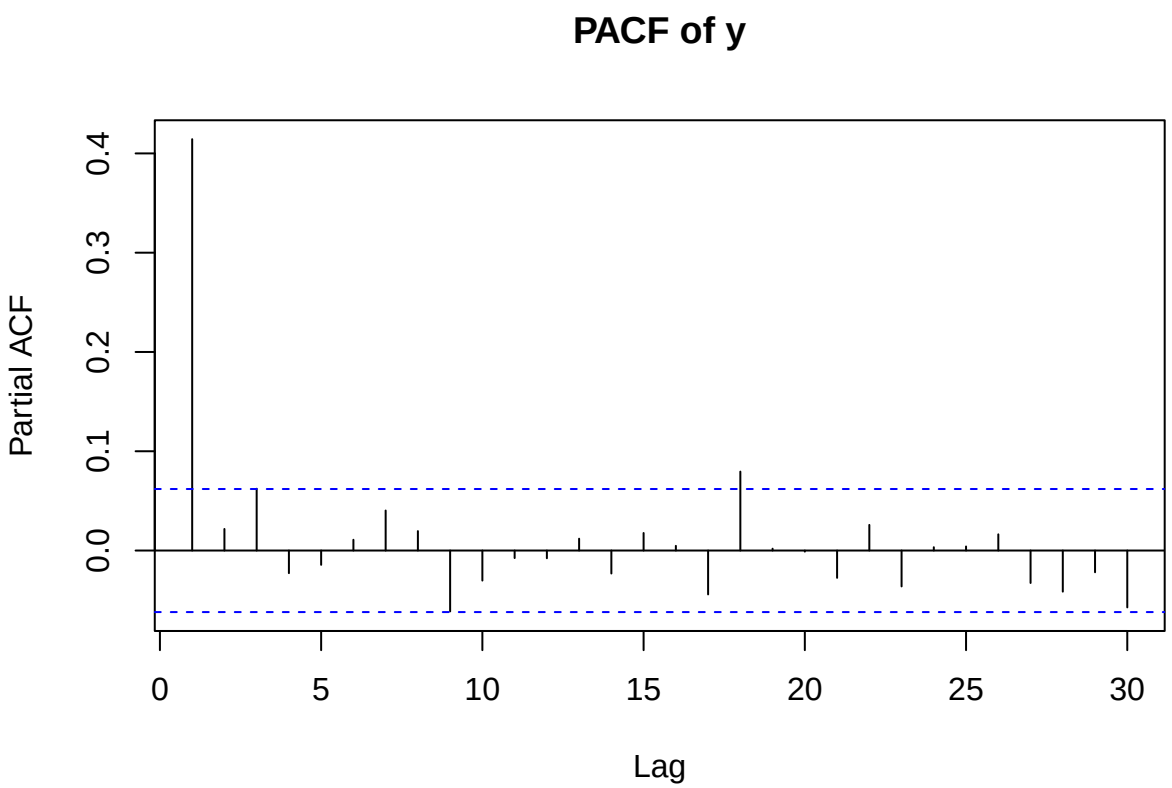
```
y4 <- numeric(T)
y4[1] <- c + epsilon[1]

for (i in 2:T){
  y4[i] <- c + phi1*y4[i-1] + epsilon[i]
}

plot(y4, type = "l", main = TeX("$y_t = c + \\phi_1y_{t-1}+\\epsilon_t$"), ylab=TeX("$y_t$"))
abline(h = c, col = "red", lty = 2)
```




```
acf(y4, main = "ACF of y")
```

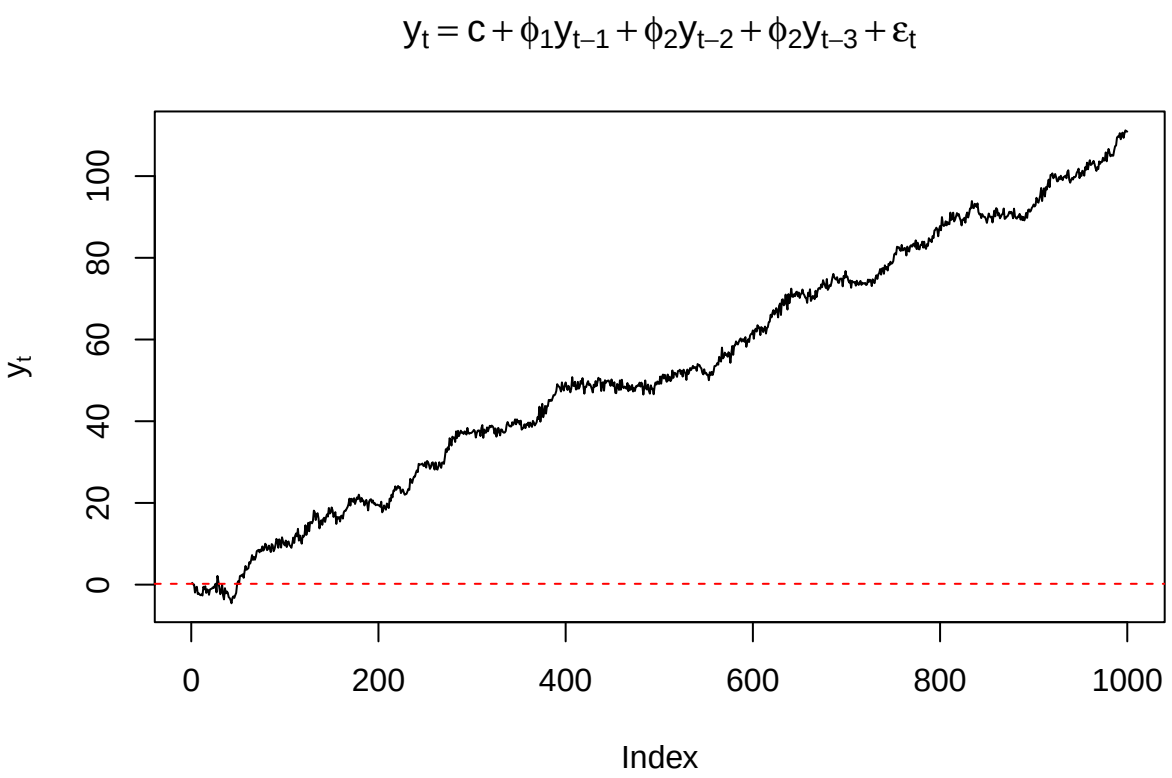



```
pacf(y4, main = "PACF of y")
```

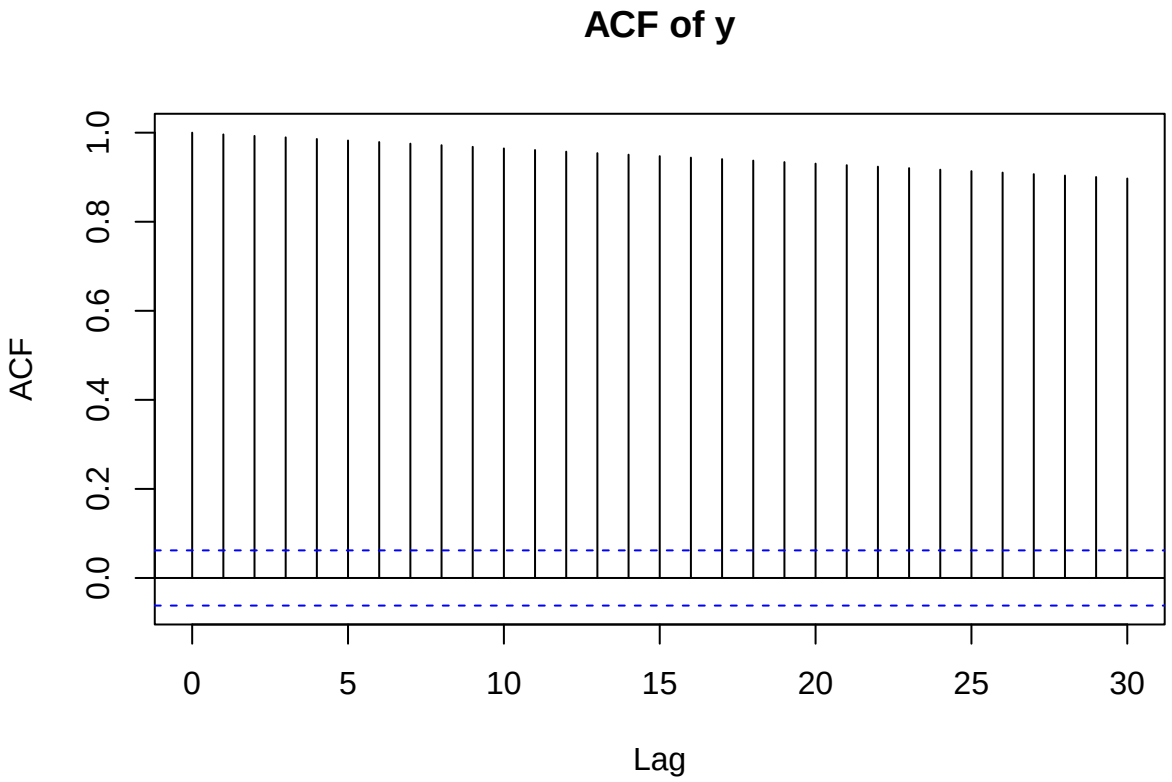


$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_2 y_{t-3} + \epsilon_t$

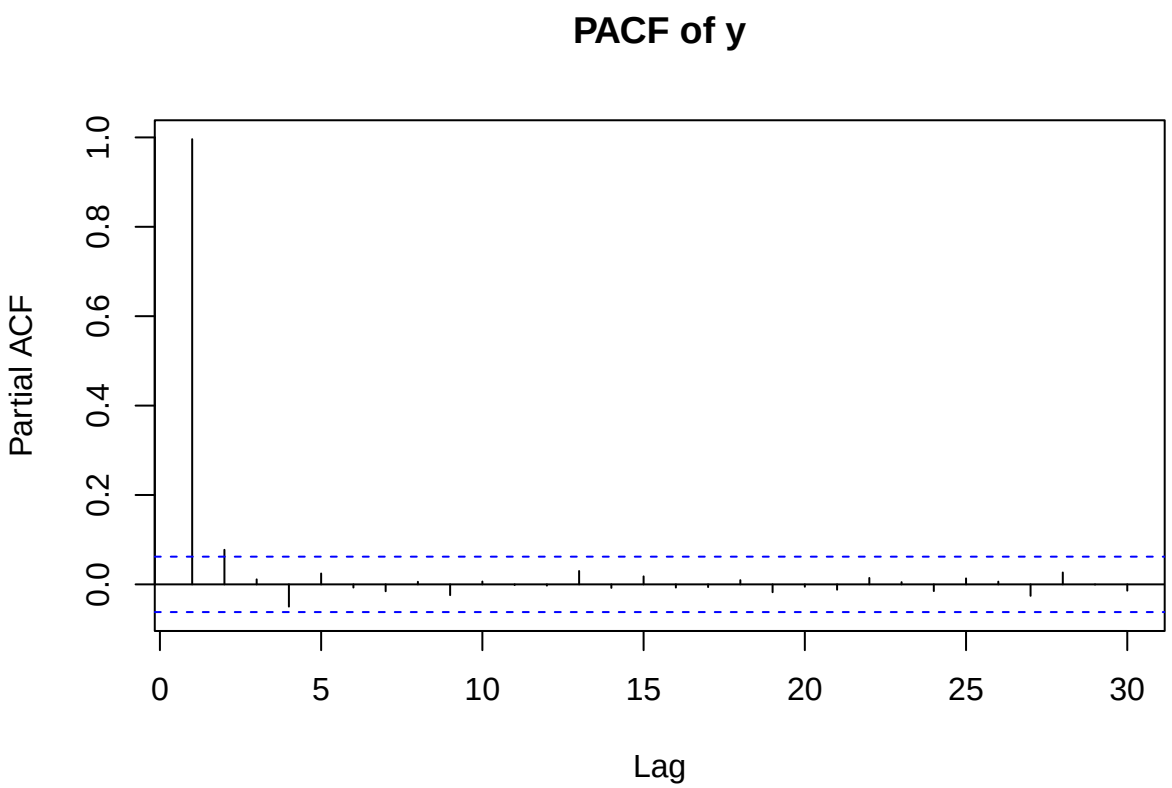
```
y5 <- numeric(T)
y5[1] <- c + epsilon[1]
y5[2] <- c + phi1*y5[1] + epsilon[2]
y5[3] <- c + phi1*y5[2] + phi2*y5[1] + epsilon[3]

for (i in 4:T){
  y5[i] <- c + phi1*y5[i-1] + phi2*y5[i-2] + phi2*y5[i-3] + epsilon[i]
}

plot(y5, type = "l", main = TeX("$y_t = c + \\phi_1y_{t-1} + \\phi_2y_{t-2} + \\phi_2y_{t-3}+\\epsilon_
abline(h = c, col = "red", lty = 2)
```




```
acf(y5, main = "ACF of y")
```

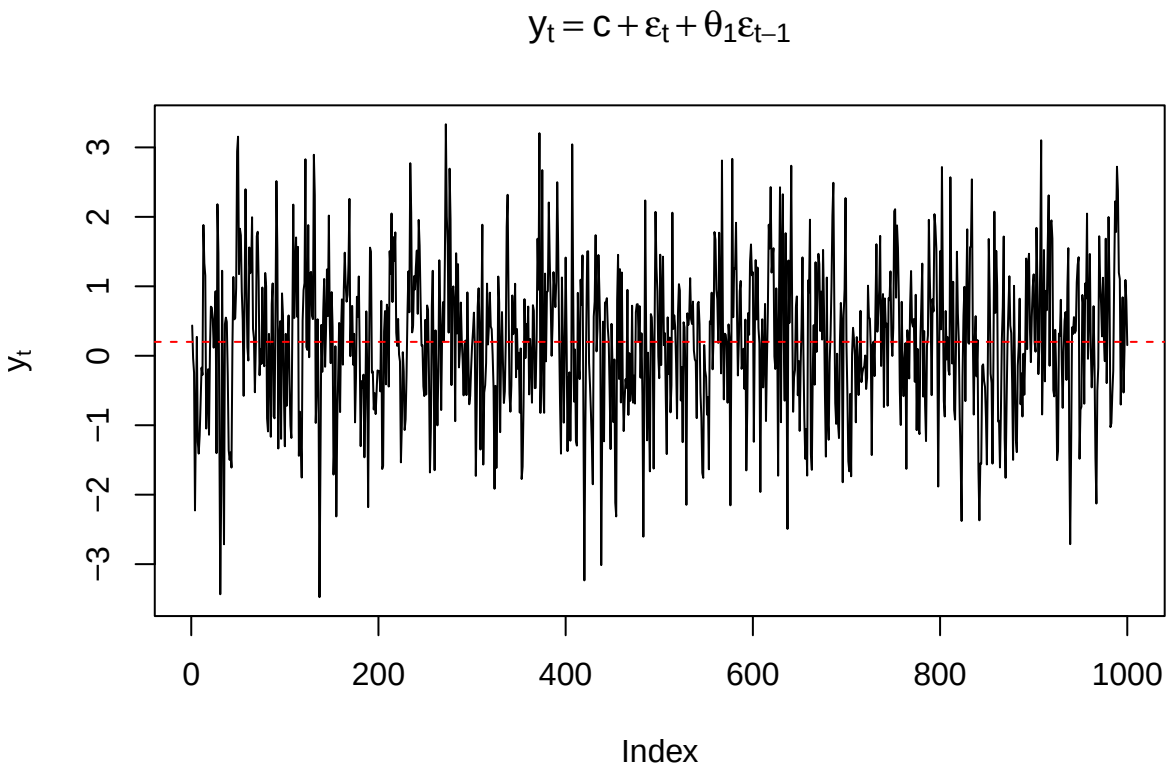



```
pacf(y5, main = "PACF of y")
```

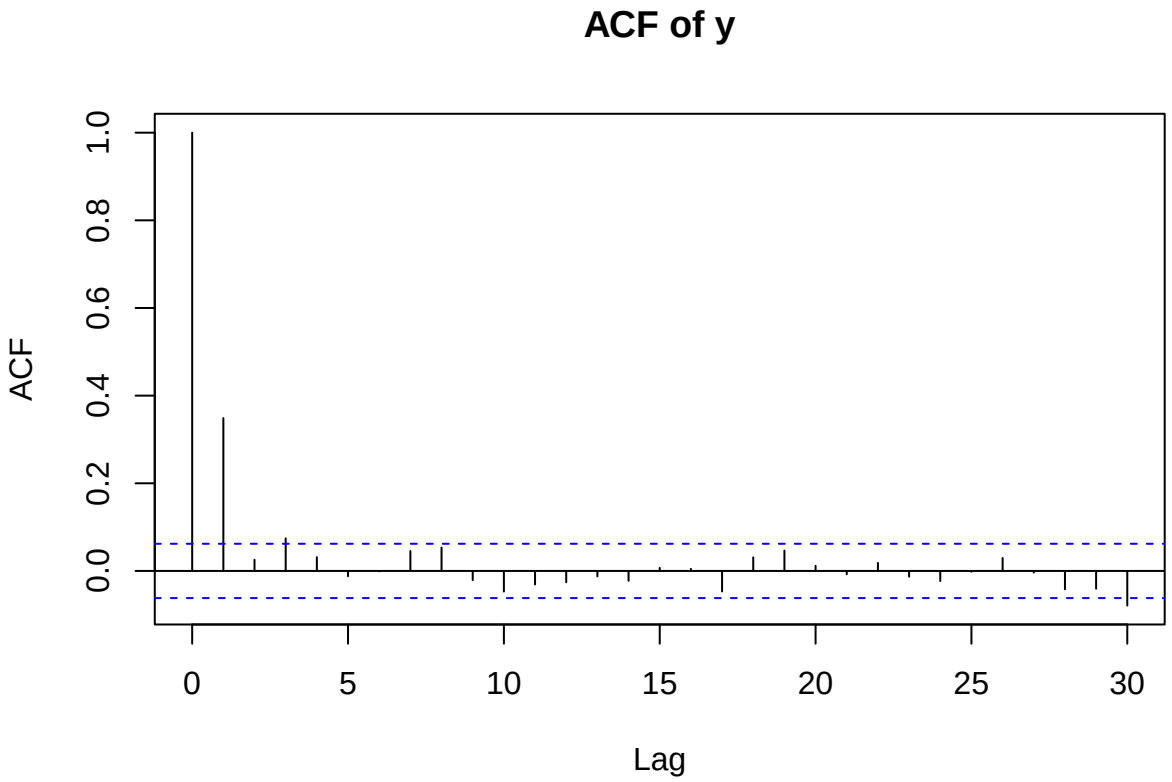


$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1}$

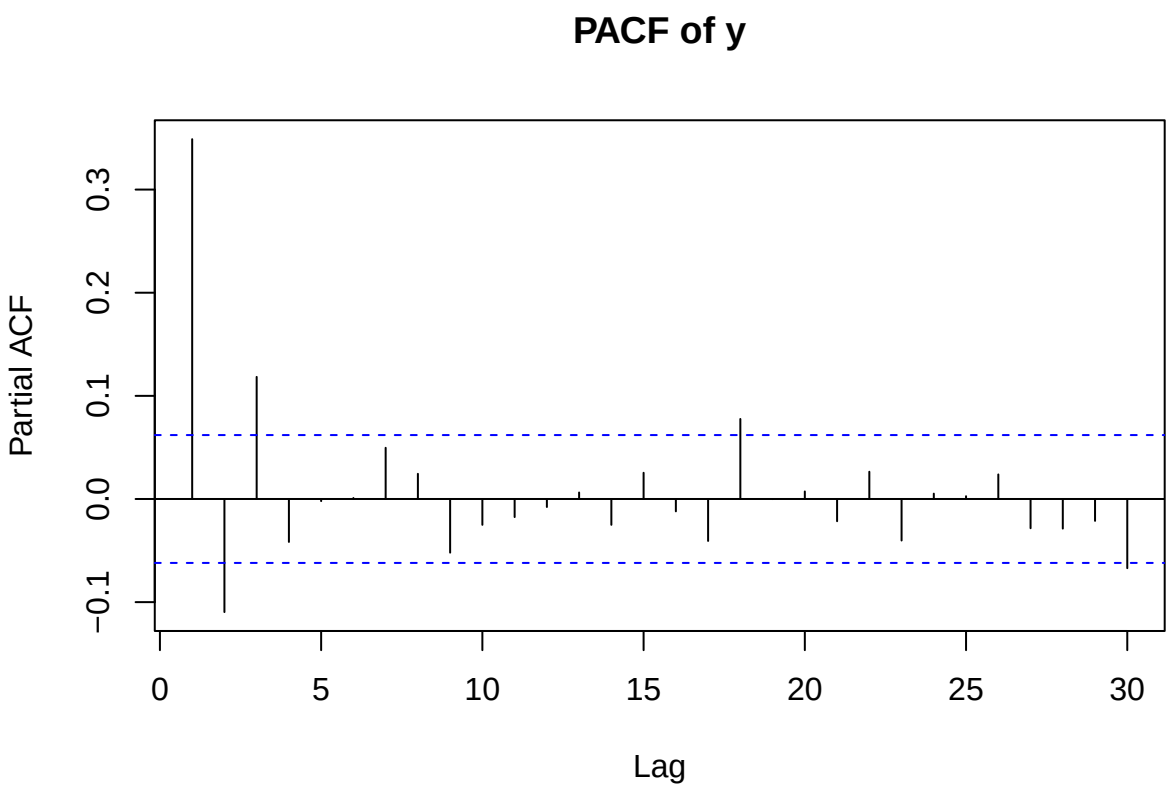
```
y6 <- numeric(T)

y6[1] <- c + epsilon[1]

for (i in 2:T){
  y6[i] <- c + epsilon[i] + theta1*epsilon[i-1]
}

plot(y6, type = "l",
     main = expression(y[t] == c + epsilon[t] + theta[1]*epsilon[t-1]),
     ylab = expression(y[t]))
abline(h = c, col = "red", lty = 2)
```




```
acf(y6, main = "ACF of y")
```

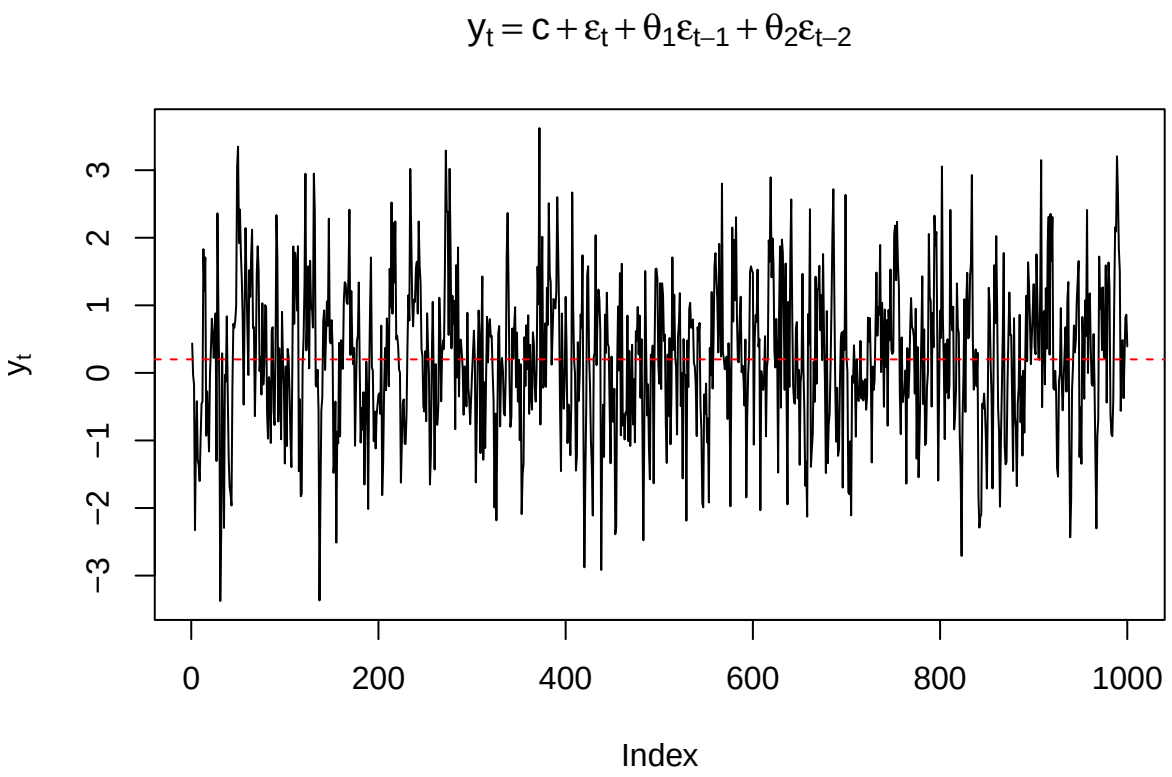



```
pacf(y6, main = "PACF of y")
```

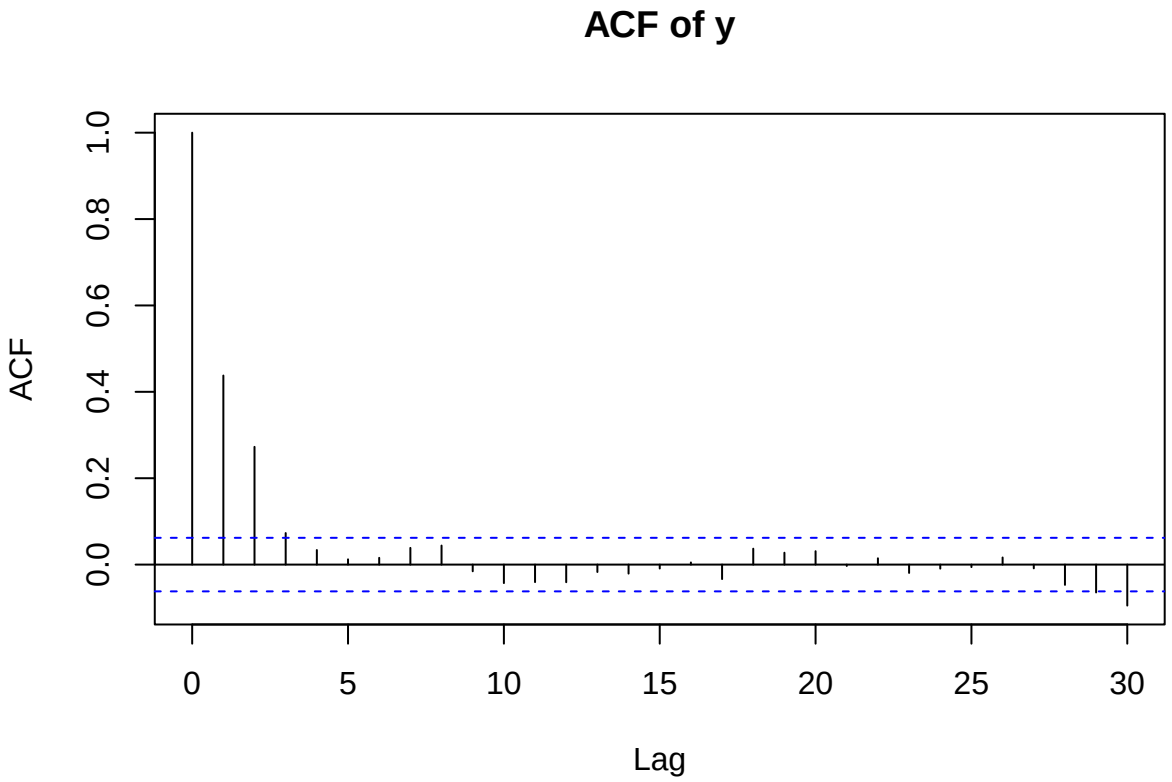


$y_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}$

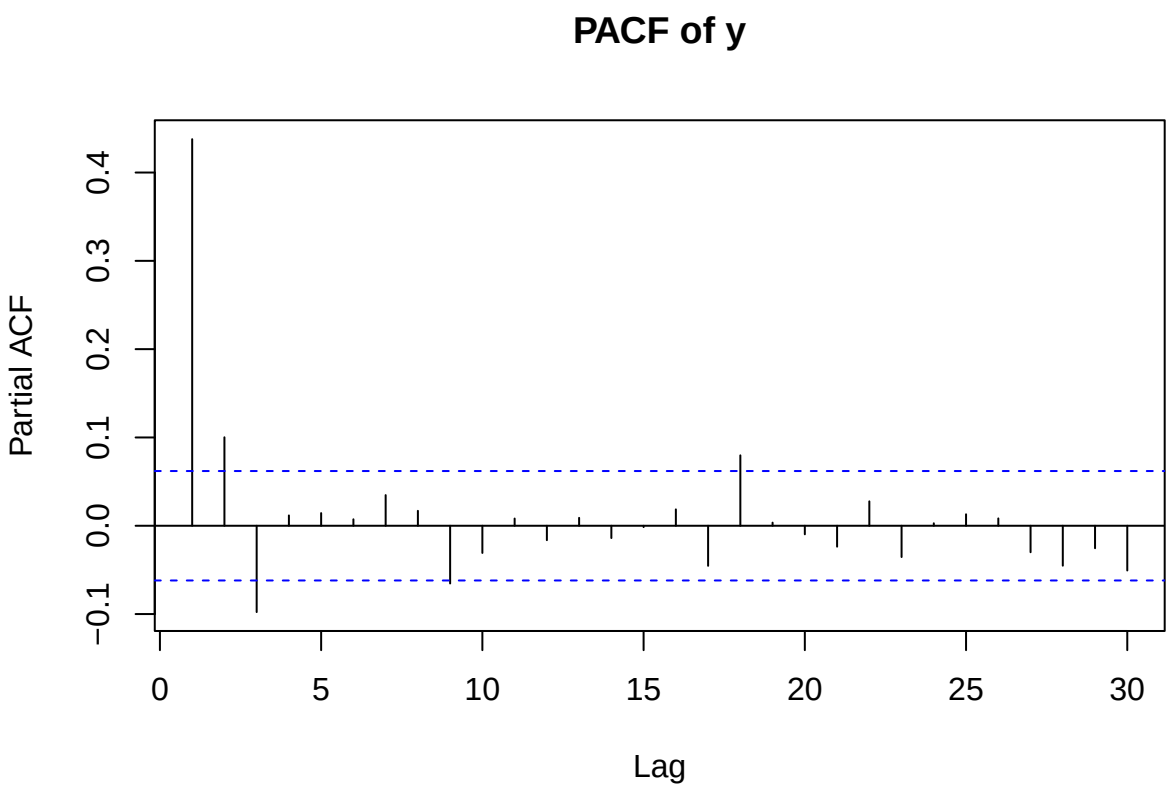
```
y7 <- numeric(T)

y7[1] <- c + epsilon[1]
y7[2] <- c + epsilon[2] + theta1*epsilon[1]

for (i in 3:T){
  y7[i] <- c + epsilon[i] + theta1*epsilon[i-1] + theta2*epsilon[i-2]
}

plot(y7, type = "l",
     main = expression(y[t] == c + epsilon[t] + theta[1]*epsilon[t-1] + theta[2]*epsilon[t-2]),
     ylab = expression(y[t]))
abline(h = c, col = "red", lty = 2)
```




```
acf(y7, main = "ACF of y")
```

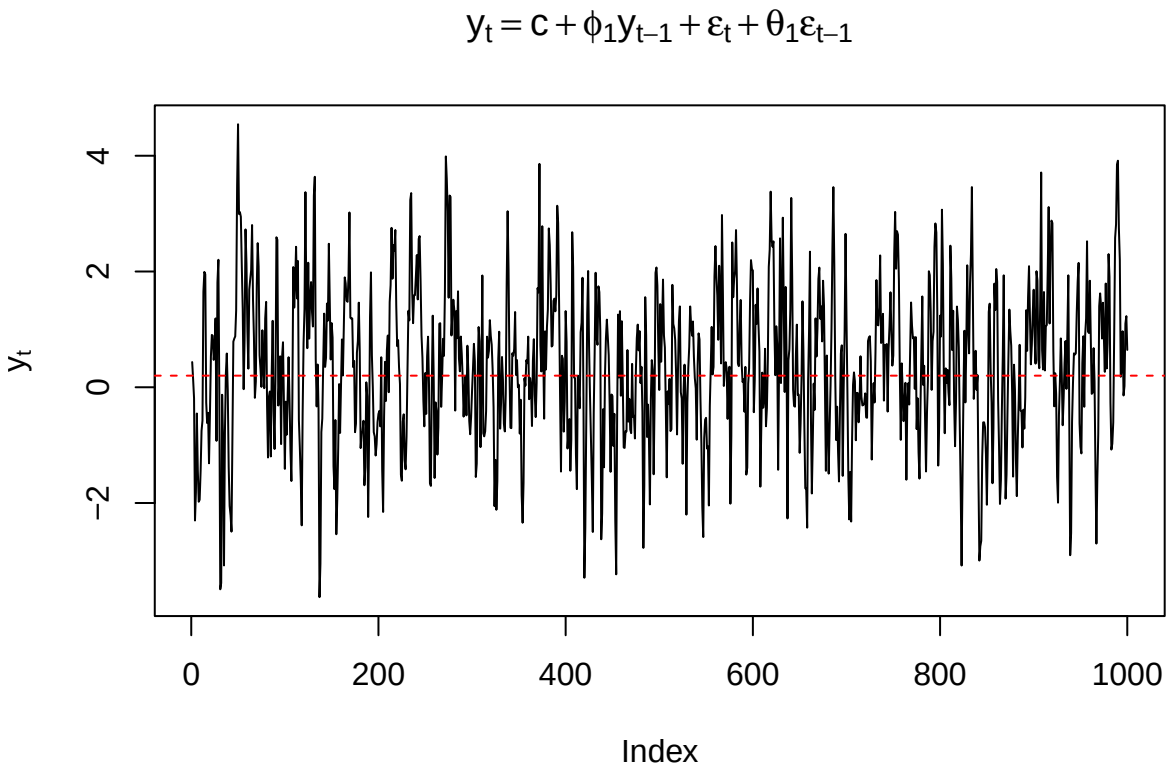



```
pacf(y7, main = "PACF of y")
```

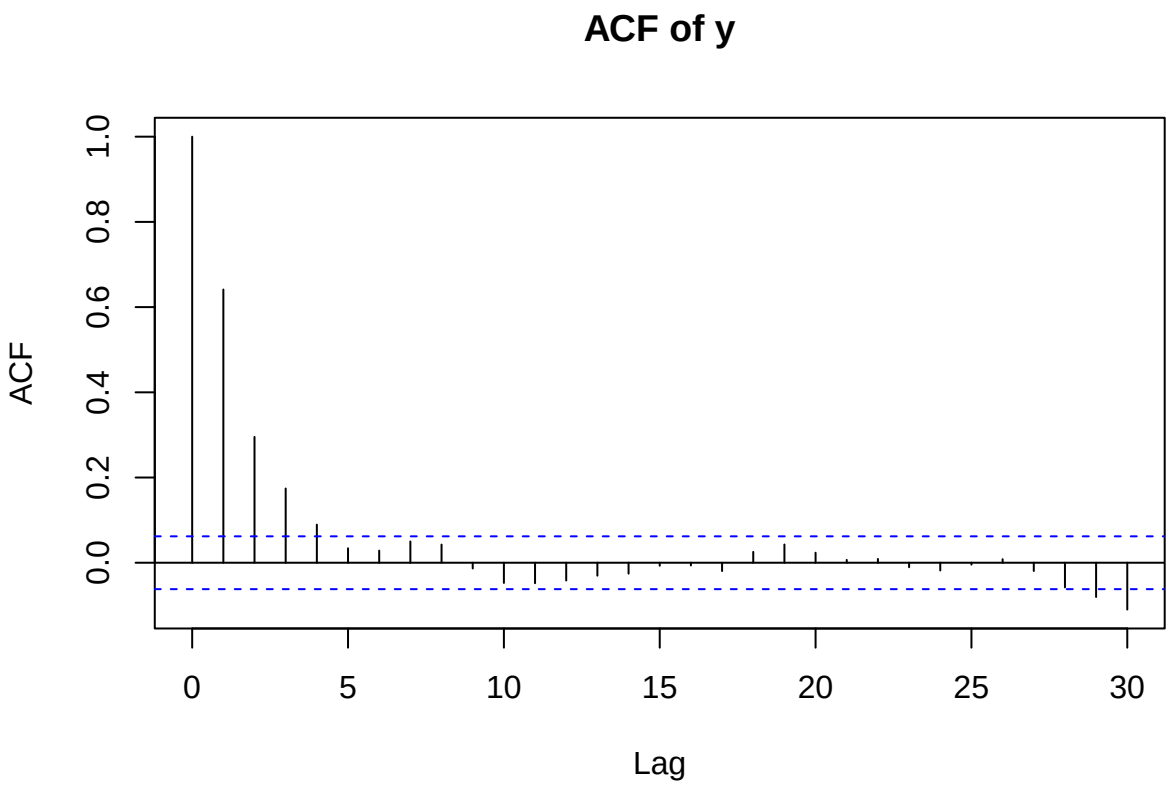


$y_t = c + \phi_1 y_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1}$

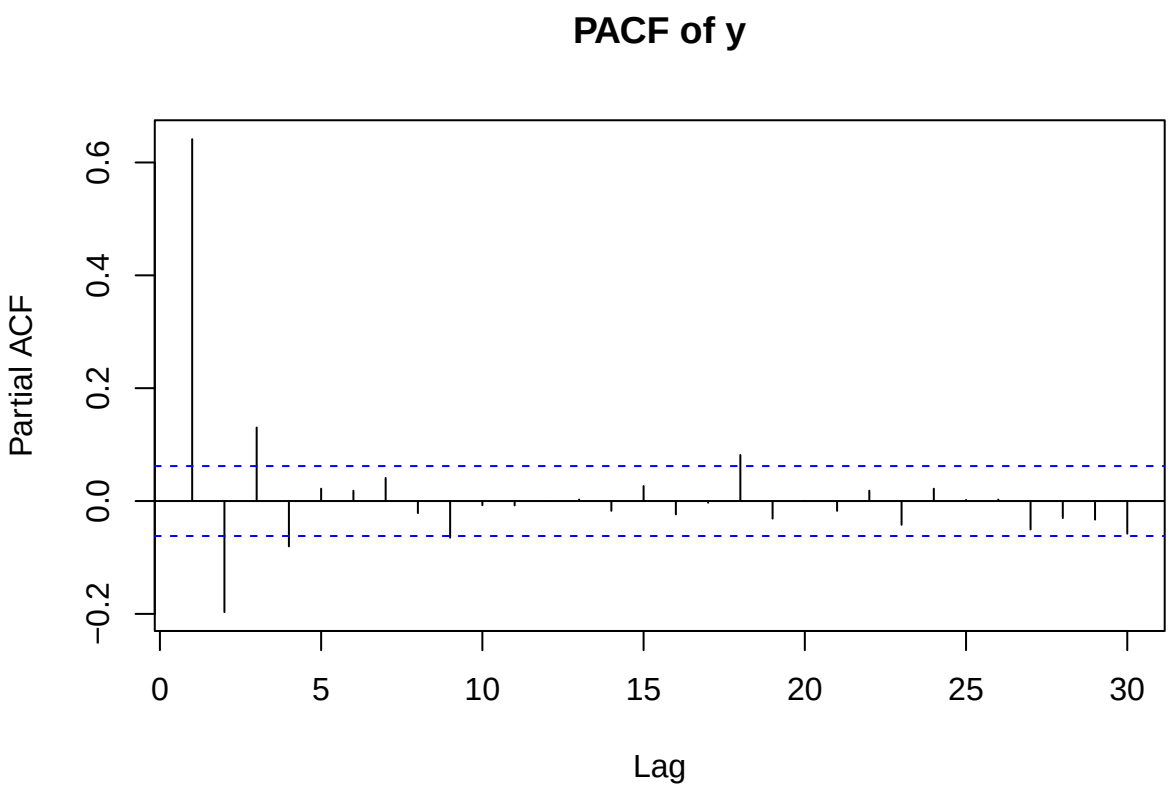
```
y8 <- numeric(T)

y8[1] <- c + epsilon[1]

for (i in 2:T){
  y8[i] <- c + phi1*y8[i-1] + epsilon[i] + theta1*epsilon[i-1]
}

plot(y8, type = "l",
     main = expression(y[t] == c + phi[1]*y[t-1]+ epsilon[t] + theta[1]*epsilon[t-1]),
     ylab = expression(y[t]))
abline(h = c, col = "red", lty = 2)
```




```
acf(y8, main = "ACF of y")
```




```
pacf(y8, main = "PACF of y")
```



Exercice 2

Loading data

S&P500

```
getSymbols(Symbols = '^GSPC', src="yahoo")
```

```
## [1] "GSPC"
```

```
sp500 <- data.frame(
  date = index(GSPC),
  SP500 = as.numeric(Ad(GSPC))
)
```

GBPUSD

```
getSymbols(Symbols = 'GBPUSD=X', src="yahoo")
```

```
## Warning: GBPUSD=X contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "GBPUSD=X"
```

```
gbpusd <- data.frame(
  date = index(`GBPUSD=X`),
  GBPUSD = as.numeric(Ad(`GBPUSD=X`))
)
```

TBILL

```
getSymbols(Symbols = 'DTB3', src="FRED")
```

```
## [1] "DTB3"
```

```
dtb3 <- data.frame(
  date = index(`DTB3`),
  DTB3 = as.numeric(`DTB3`)
)
```

VIX

```
getSymbols(Symbols = '^VIX', src="yahoo")
```

```
## [1] "VIX"
```

```
vix <- data.frame(
  date = index(`VIX`),
  VIX = as.numeric(Ad(`VIX`))
)
```

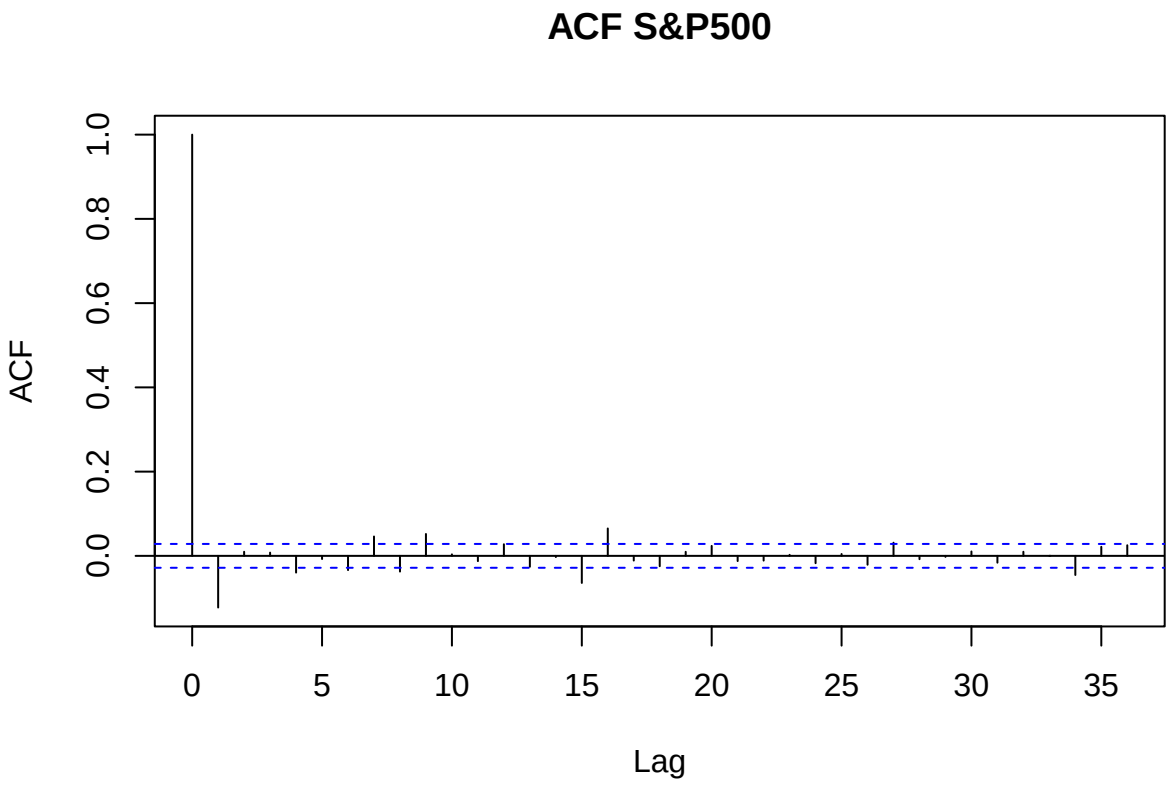
Transformation

```
sp500_d <- sp500 %>%
  mutate(
```

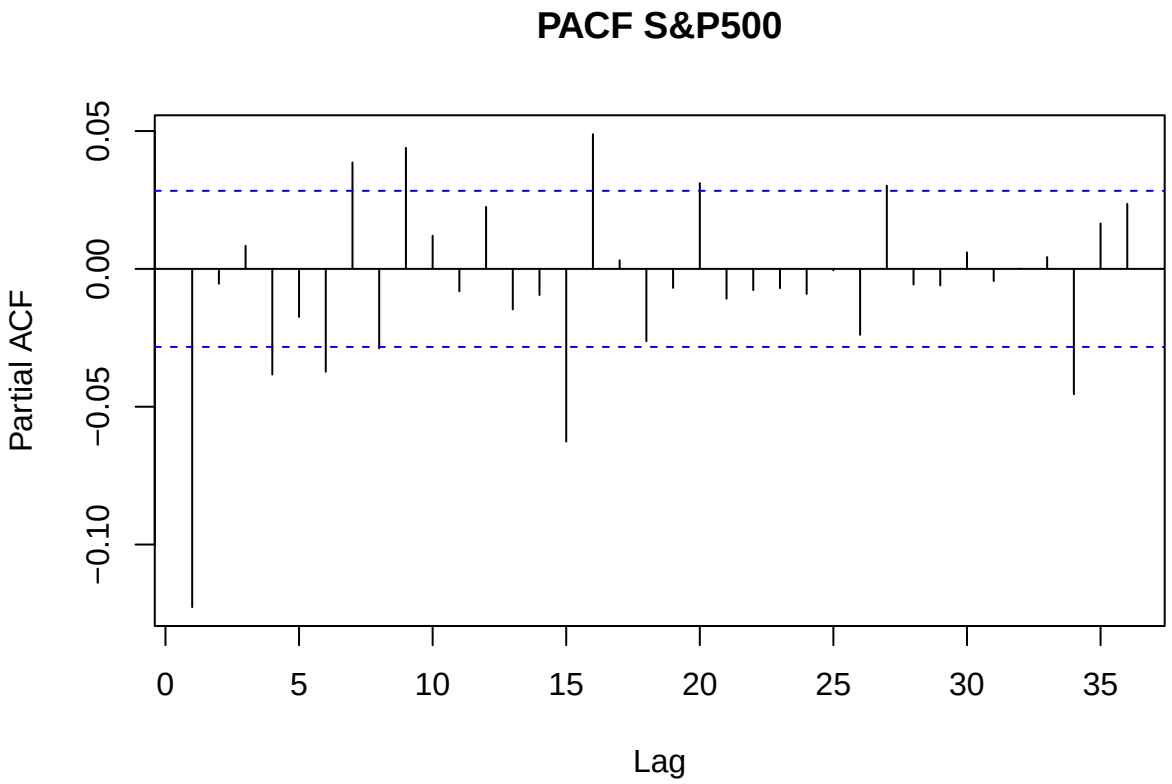
```
    lr = log(SP500) - log(lag(SP500))
  ) %>%
  na.omit()

gbpusd_d <- gbpusd %>%
  mutate(
    lr = log(GBPUSD) - log(lag(GBPUSD))
  ) %>%
  na.omit()


vix_d <- vix %>%
  mutate(
    lr = log(VIX) - log(lag(VIX))
  ) %>%
  na.omit()

dtb3_d <- dtb3 %>%
  na.omit() %>%
  mutate(
    d = DTB3 - lag(DTB3)
  ) %>%
  na.omit()
```

Plot ACF and PACF for each stock

```
acf(sp500_d$lr, main = "ACF S&P500")
```

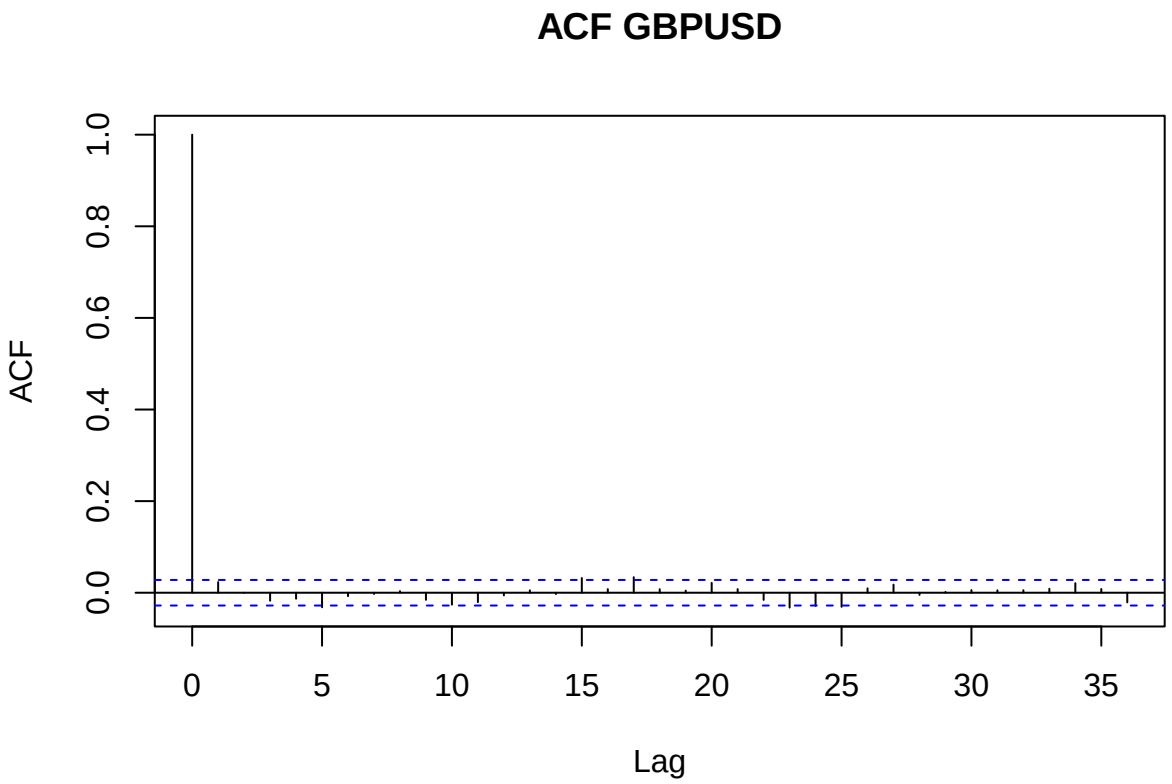


```r
pacf(sp500_d$lr, main = "PACF S&P500")
```

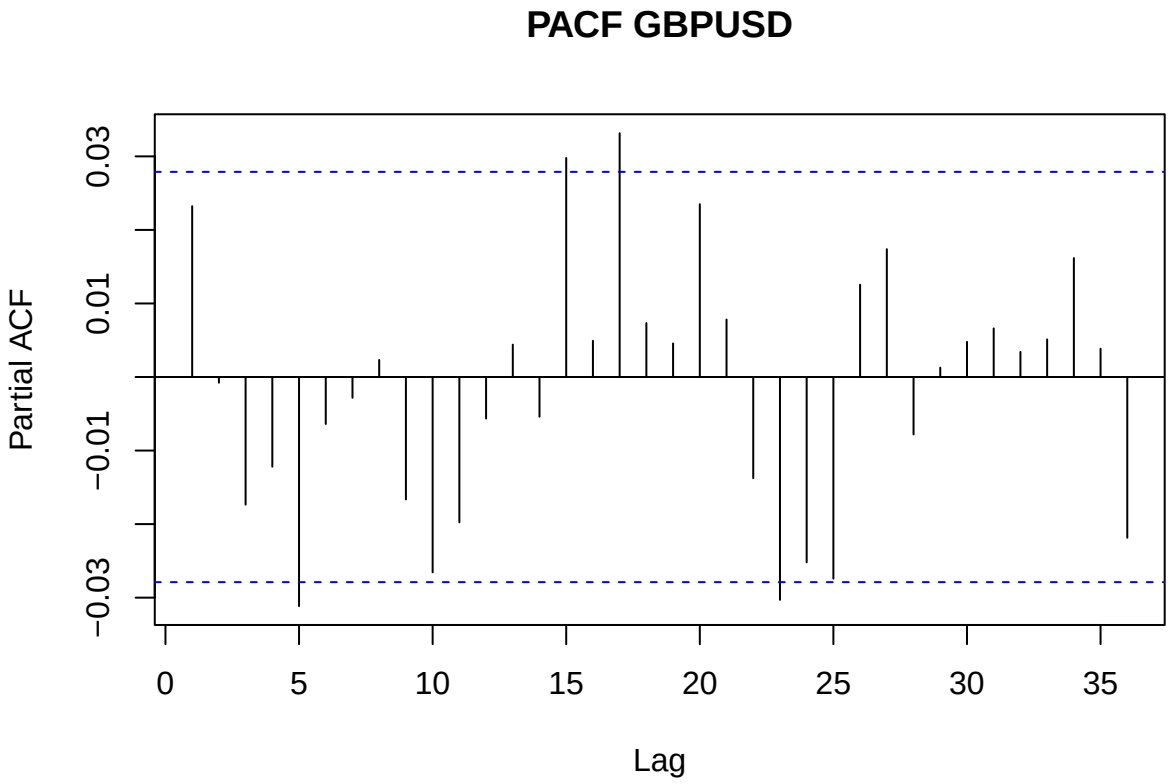


Conclusion ARMA(1) car pic (négatif) significatif au lag 1 ACF et PACF

```r
acf(gbpusd_d$lr, main = "ACF GBPUSD")
```

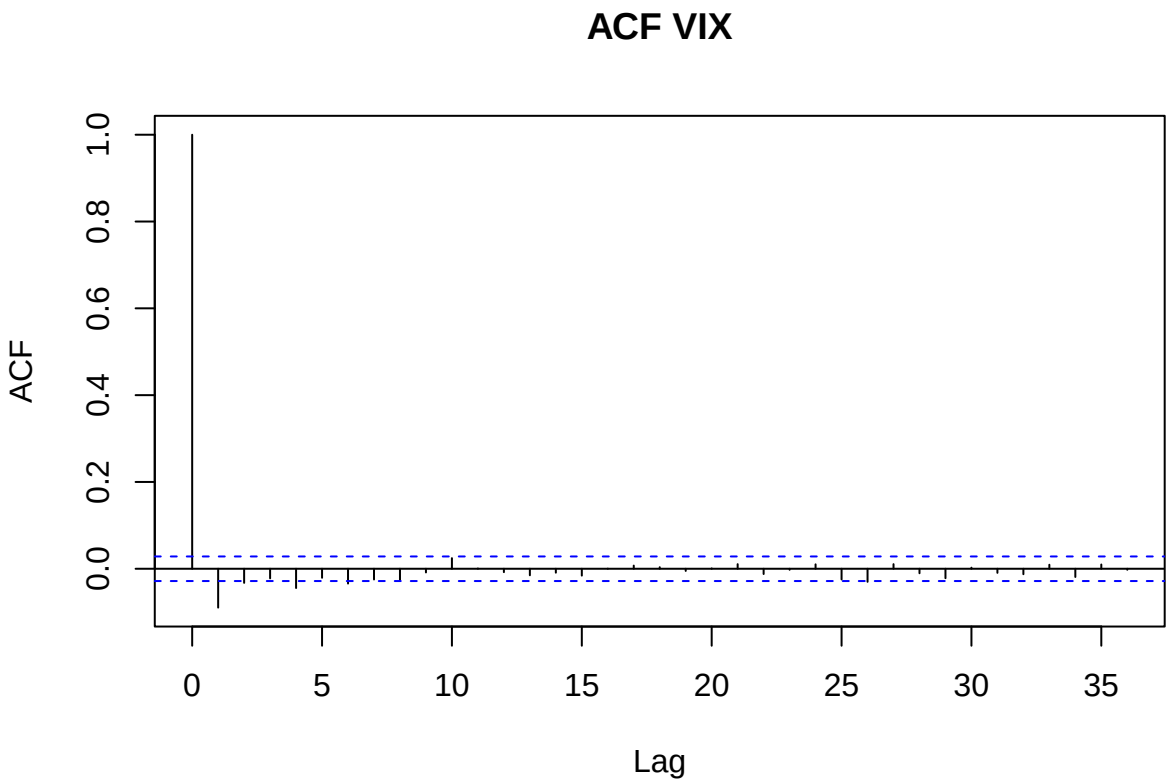



```
pacf(gbpusd_d$lr, main = "PACF GBPUSD")
```

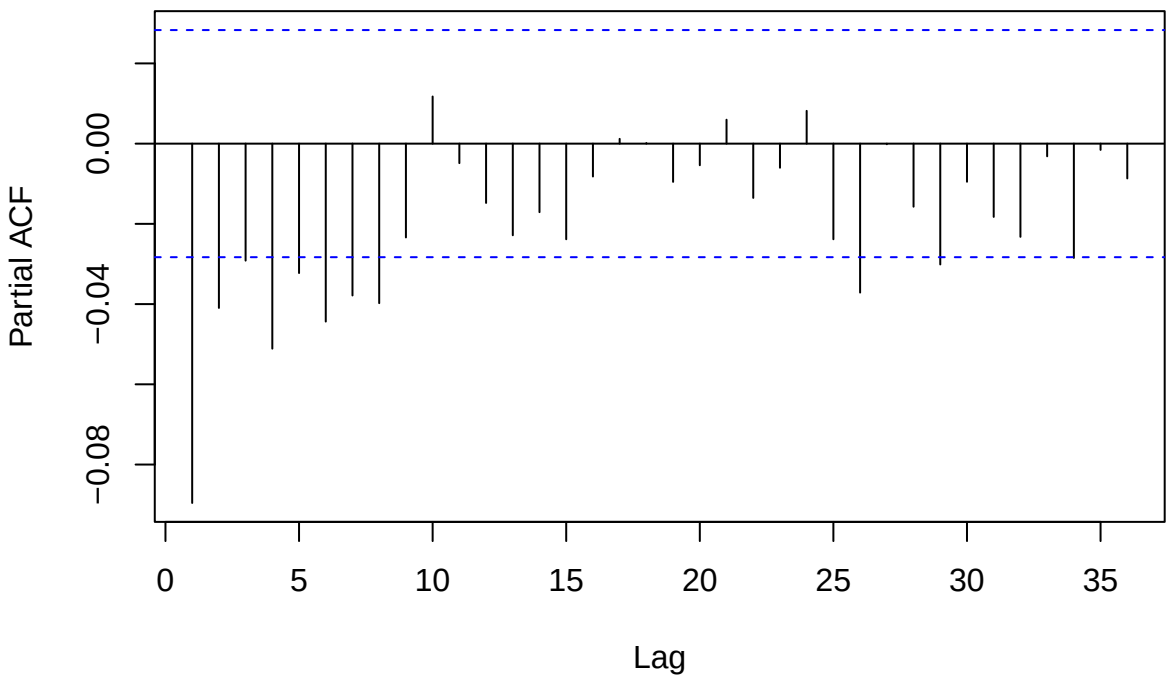


Conclusion rien de significatif, que ce soit dans le ACF ou PACF. ARMA(0,0)

```r
acf(vix_d$lr, main = "ACF VIX")
```

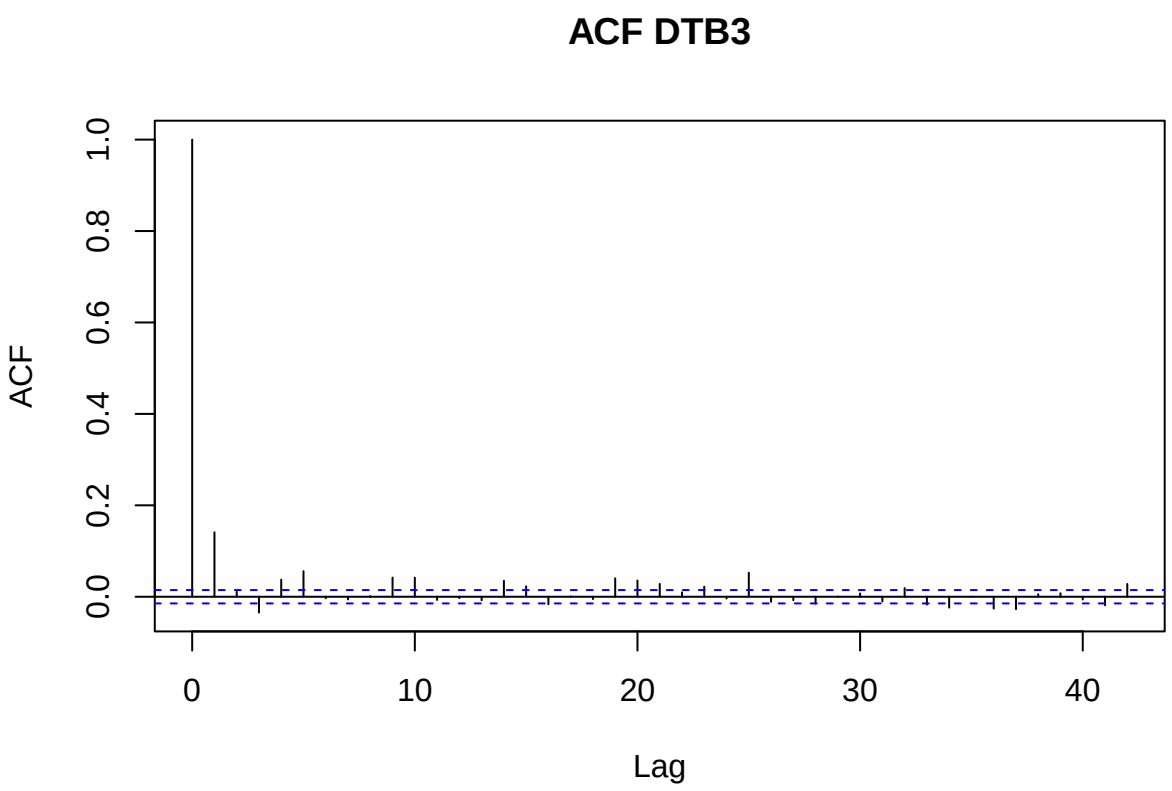



```r
pacf(vix_d$lr, main = "PACF VIX")
```

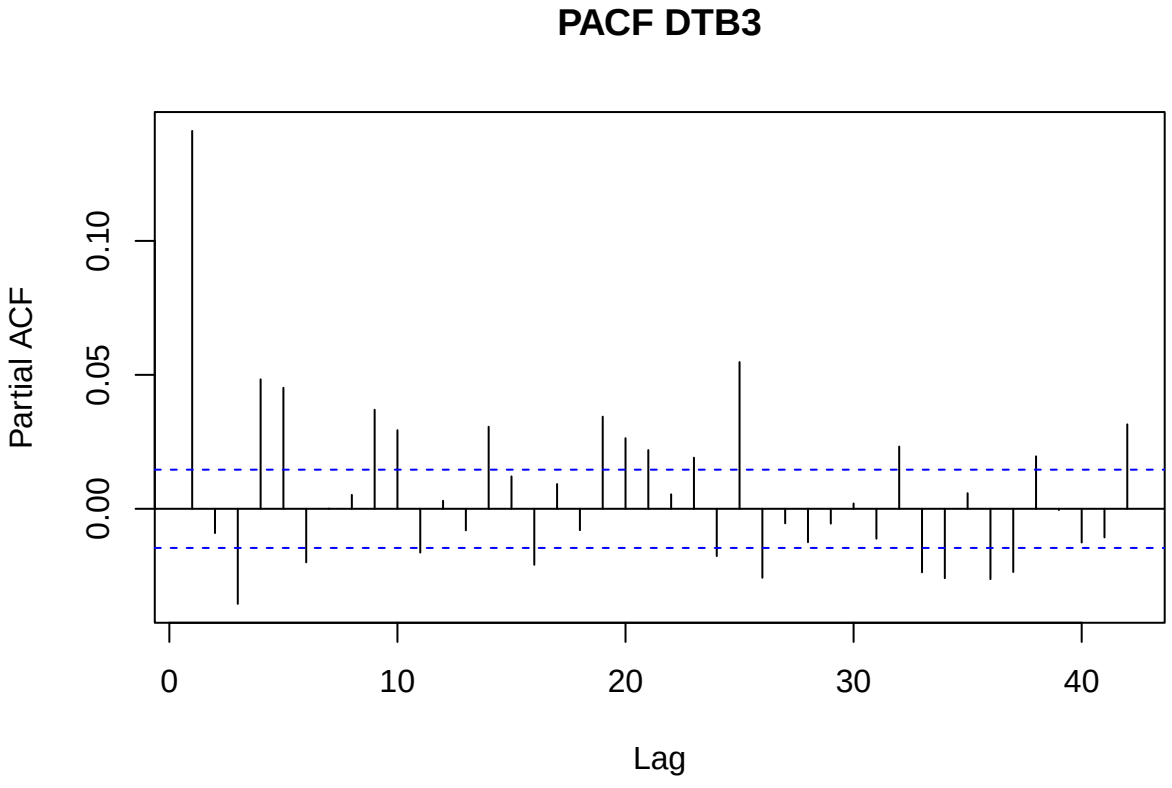


Conclusion AR(1) car pic au ACF - Décroissance lente PACF

```
acf(dtb3_d$d, main = "ACF DTB3")
```




```
pacf(dtb3_d$d, main = "PACF DTB3")
```



Conclusion