PROJET SERIE CHRONOLOGIQUE

NABIL MARHAR, HARON REZGUI & AYMAN YAYA

2023-04-20

Nous étudions une base de données contenant les relevés quotidiens de température de la ville de Marignane sur les 13 dernières années. Il est alors intéréssant d'analyser ce jeu de données, de telle manière à observer la tendance et la saisonnalité de cette série chronologique, et ainsi pouvoir par la suite modéliser cette série qui nous permettra alors de comprendre et prédire le comportement de cette dernière au fil du temps.

1°Analyse descriptive et quelques statistiques de notre jeu de données !

```
## 'data.frame':    4826 obs. of  2 variables:
##  $ Date     : Date, format: "2010-01-01" "2010-01-02" ...
##  $ mean.temp: num  8.77 8.32 7.45 9.37 8.42 ...
```

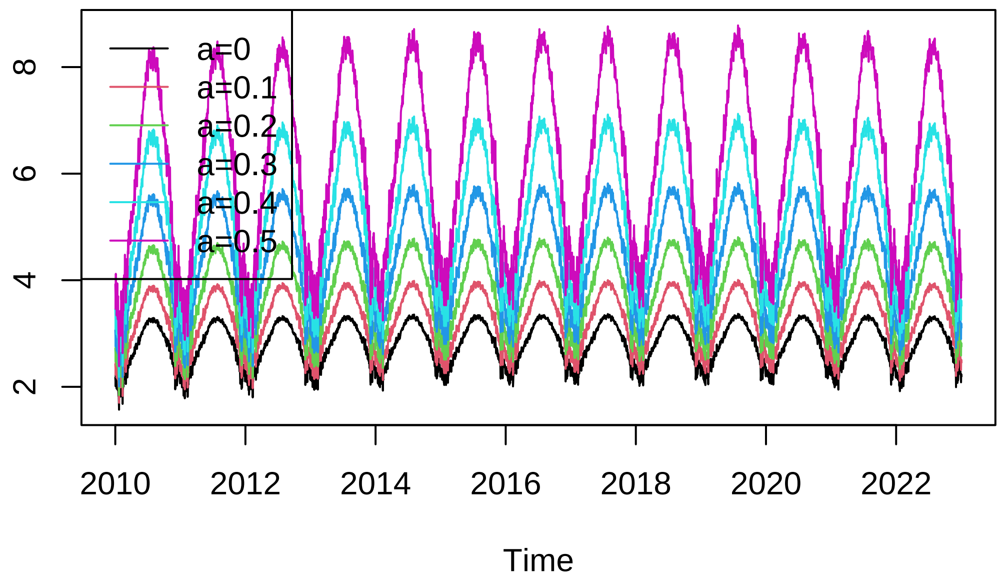
```
##         Date mean.temp
## 1 2010-01-01  8.767112
## 2 2010-01-02  8.316055
## 3 2010-01-03  7.453169
## 4 2010-01-04  9.367045
## 5 2010-01-05  8.417617
## 6 2010-01-06  8.219190
```

```
##       Date              mean.temp
##  Min.   :2010-01-01   Min.   : 4.808
##  1st Qu.:2013-04-21   1st Qu.:10.958
##  Median :2016-08-09   Median :16.976
##  Mean   :2016-08-09   Mean   :17.262
##  3rd Qu.:2019-11-28   3rd Qu.:23.623
##  Max.   :2023-03-19   Max.   :29.067
```

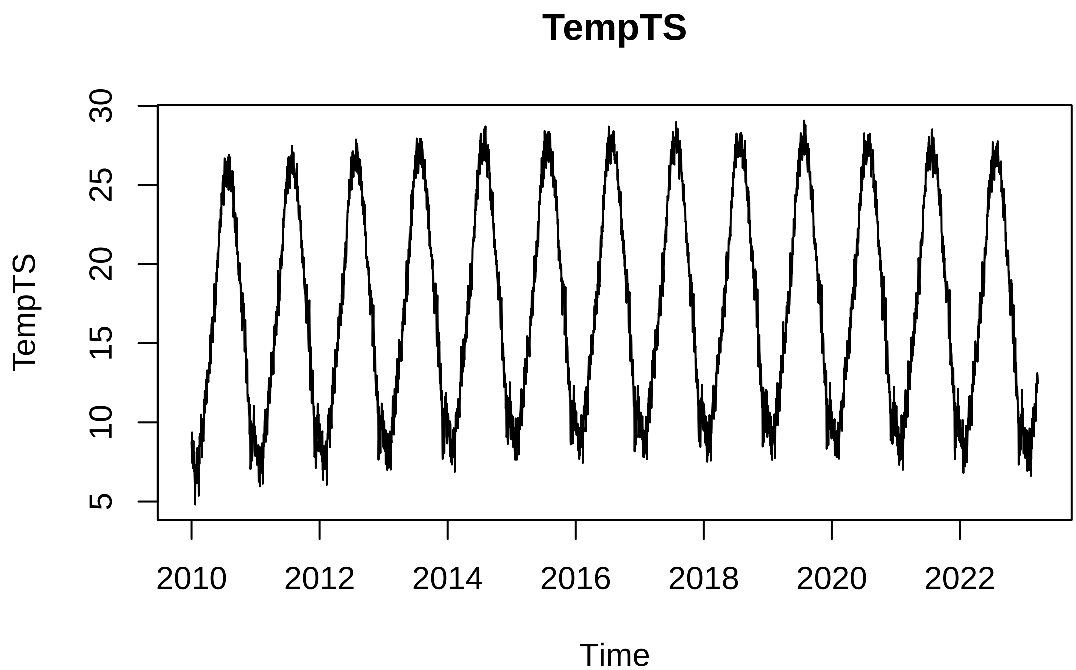
Le jeu de données présente 2 variables: la date et la température journalière moyenne, et 4826 observations correspondant à toutes les températures mesurées à Marignane depuis le 1er Janvier 2010. Sur l'ensemble de cette période, une température moyenne a été constaté de 17,3°C avec minima de 4,8° et maxima de 29,1°.

```r
# transformation modèle additif

Temp.ts=ts(Temp$mean.temp,start=c(2010,1,1),end=c(2023,3,19),frequency=365)
source("boxcox.r")

A = seq(0,0.5,0.1)
n = length(Temp.ts)
M = matrix(0,n,length(A))

for(i in (1:length(A))){
         M[,i] = boxcox(Temp.ts,A[i])
```

```
}
M = ts(M,start=c(2010,1,1),frequency=365)
ts.plot(M, col=(1:length(A)))

legend('topleft',legend=paste("a=",as.character(A),sep=""),col=(1:length(A)),lty = 1)
```

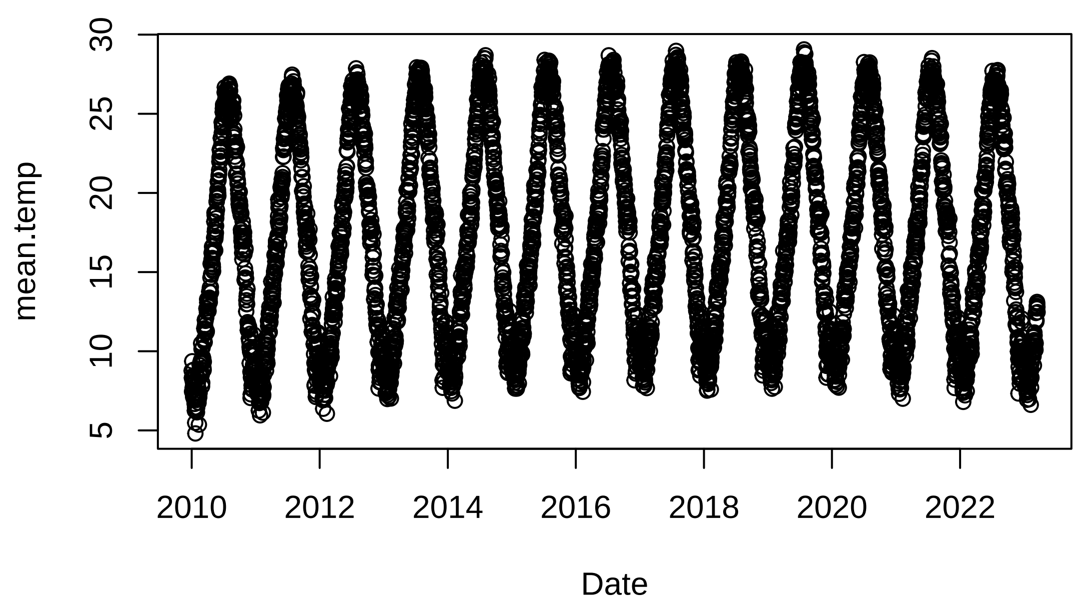



```
# Régression linéaire pour estimer la tendance
TempTS = ts(Temp$mean.temp,start=c(2010,1),frequency=365)
lm.TempTS = lm(TempTS ~ time(TempTS))
plot(TempTS, main="TempTS")
```

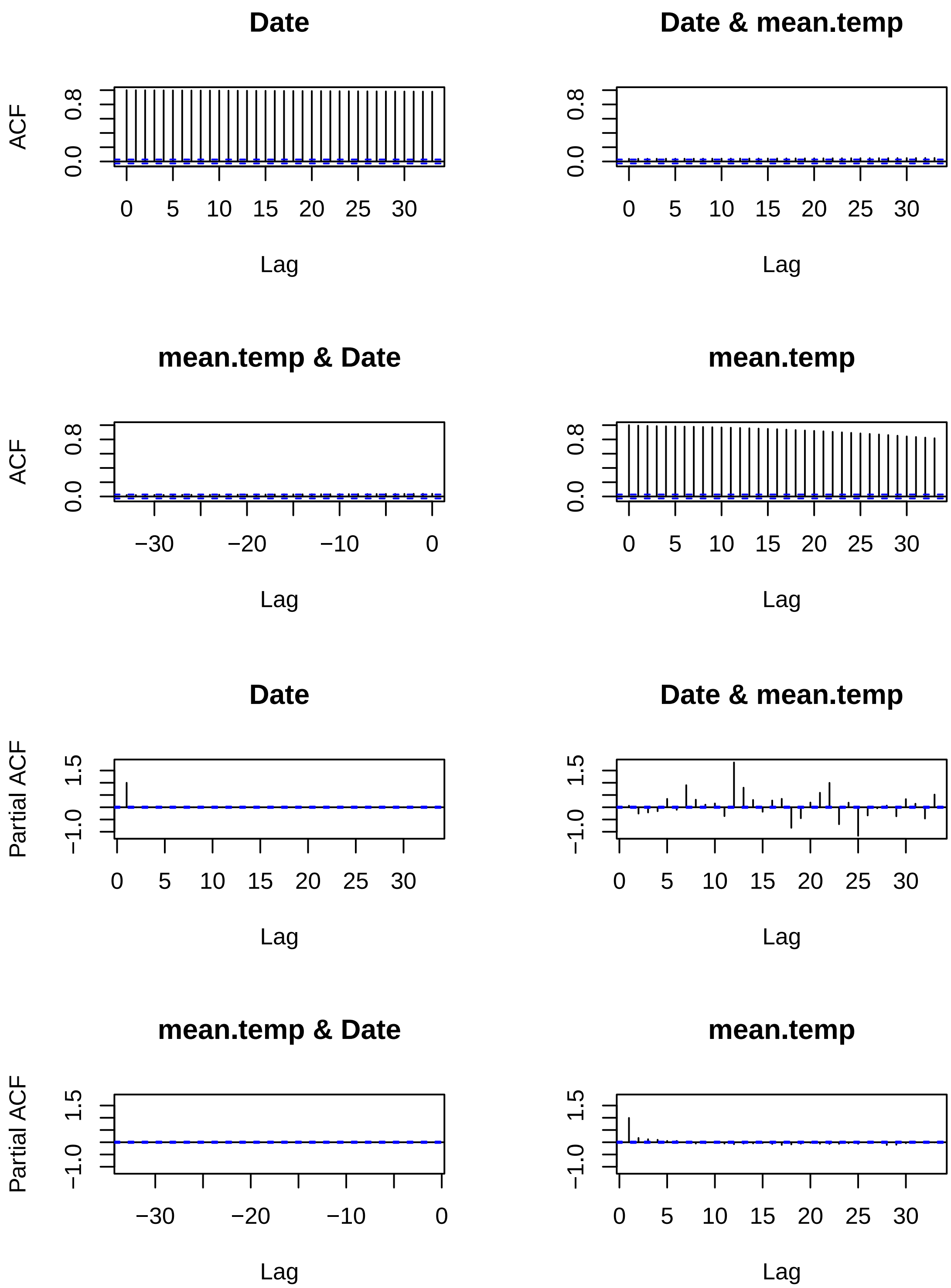


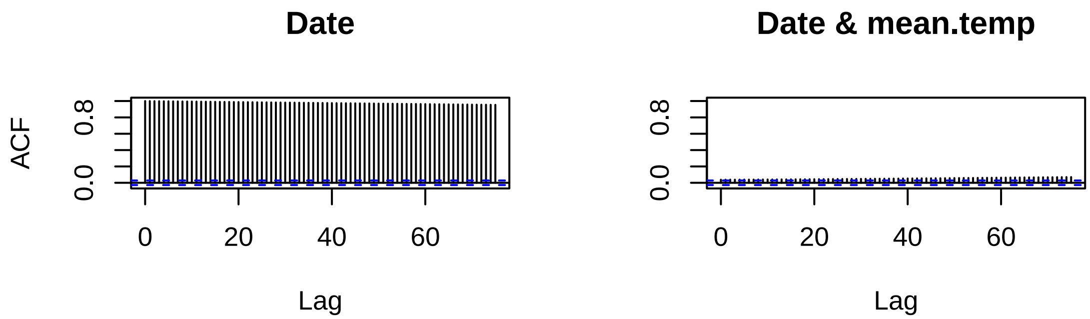

```
# lines( as.vector(time(TempTS)),lm.TempTS$fitted.values,col="blue")
```

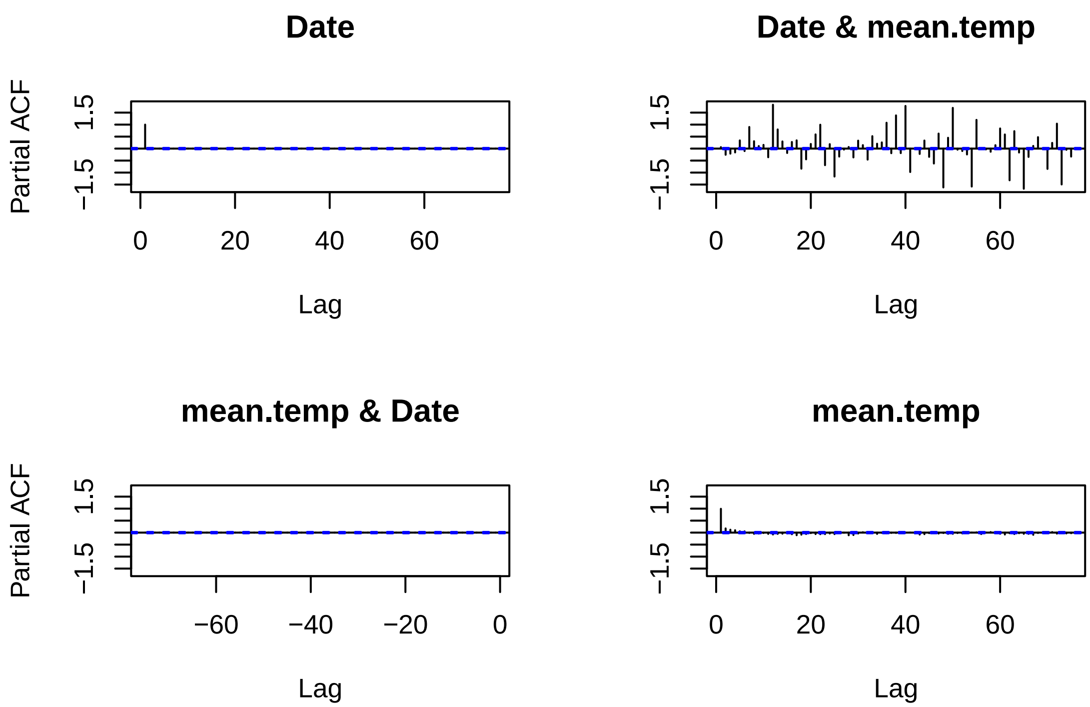
Une erreur s'affichant, nous ne nous sommes pas parvenus à tracer la droite de régression d'ordre 1. Mais au vue du graphique nous pouvons dès lors supposer une tendance croissante avec un effet saisonnier annuel, avec un pic annuel avoisinant les 28°C. Cela correspond naturellement à la période estivale.



Cidessus le nuage de points du jeu de données.







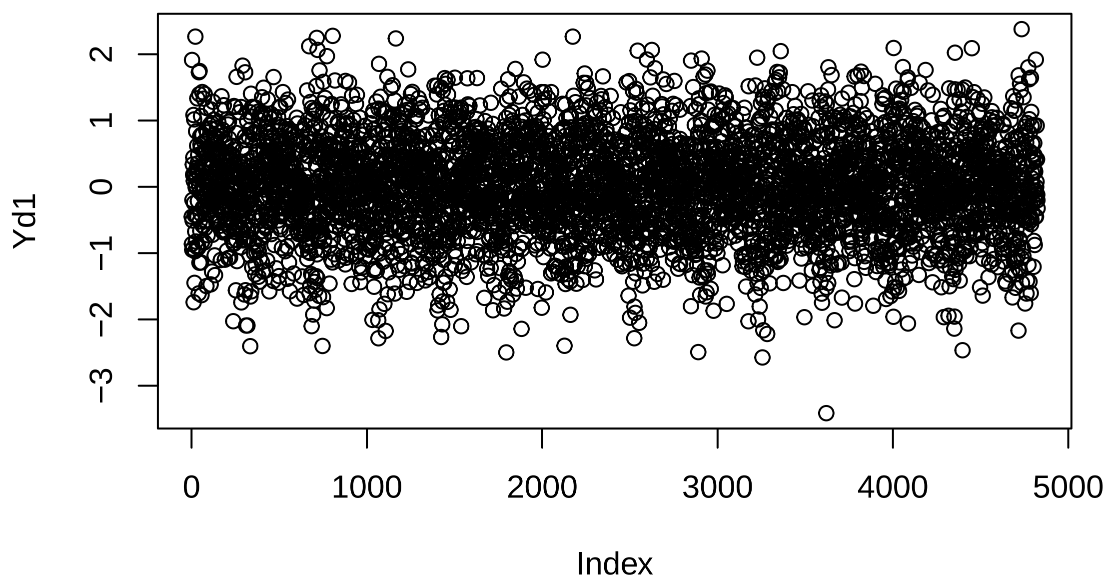
L'ACF mesure la corrélation entre une observation de la série chronologique et ses observations passées à différents retards. Ici nous avons pris retard lag=75 L'ACF montre à quelle vitesse la corrélation décroît à mesure que le retard augmente. On observe que la corrélation décroît lentement, cela suggère qu'il y a une forte dépendance temporelle dans la série. La PACF, quant à elle, mesure la corrélation entre une observation de la série chronologique et une observation passée à un retard particulier (ici lag=75), en éliminant l'effet de toutes les observations intermédiaires.

On voit alors que notre série présente une forte dépendance selon le temps.En effet, malgré des retards assez grands (resp. 30 et 75), la série présentent tout de même une corrélation importante entre ses observations.

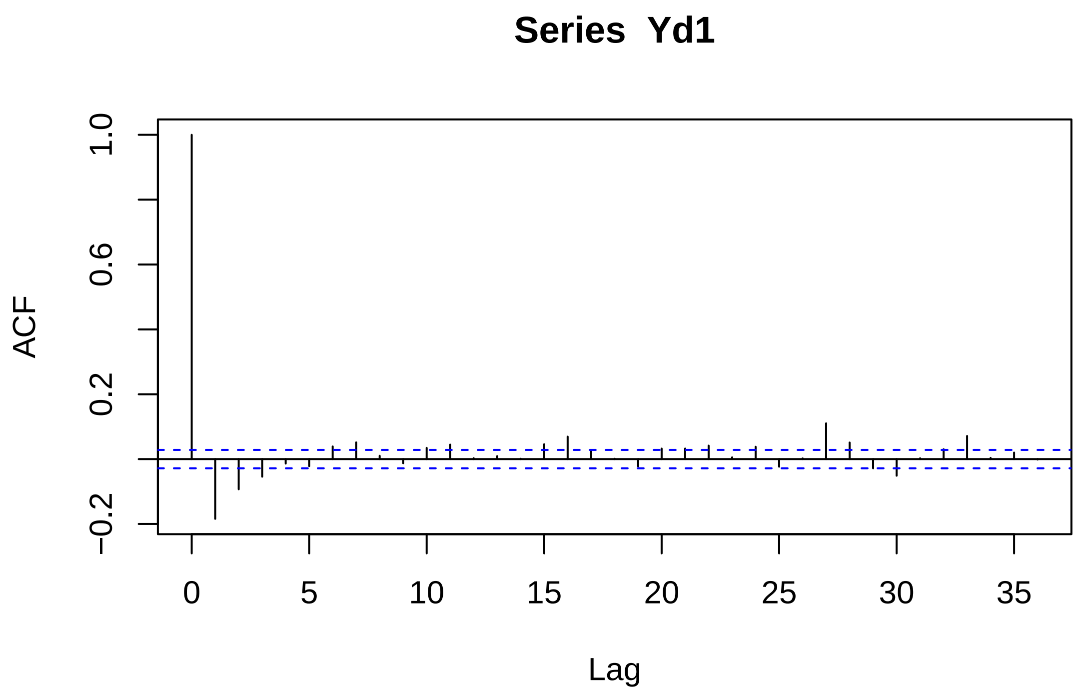
Il est alors intéréssant de stationnariser la série chronologique afin de la transformer en une série plus simple et régulière, ce qui facilitera son analyse et sa modélisation.

2° Stationnarisation de la série

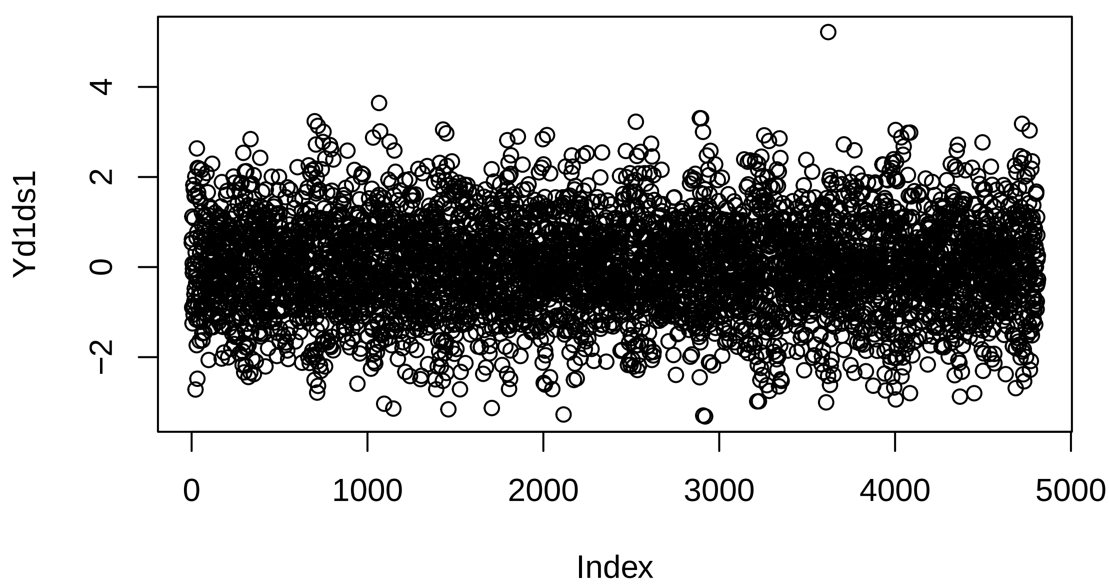
```
Yd1 = diff(Temp$mean.temp)
plot(Yd1)
```



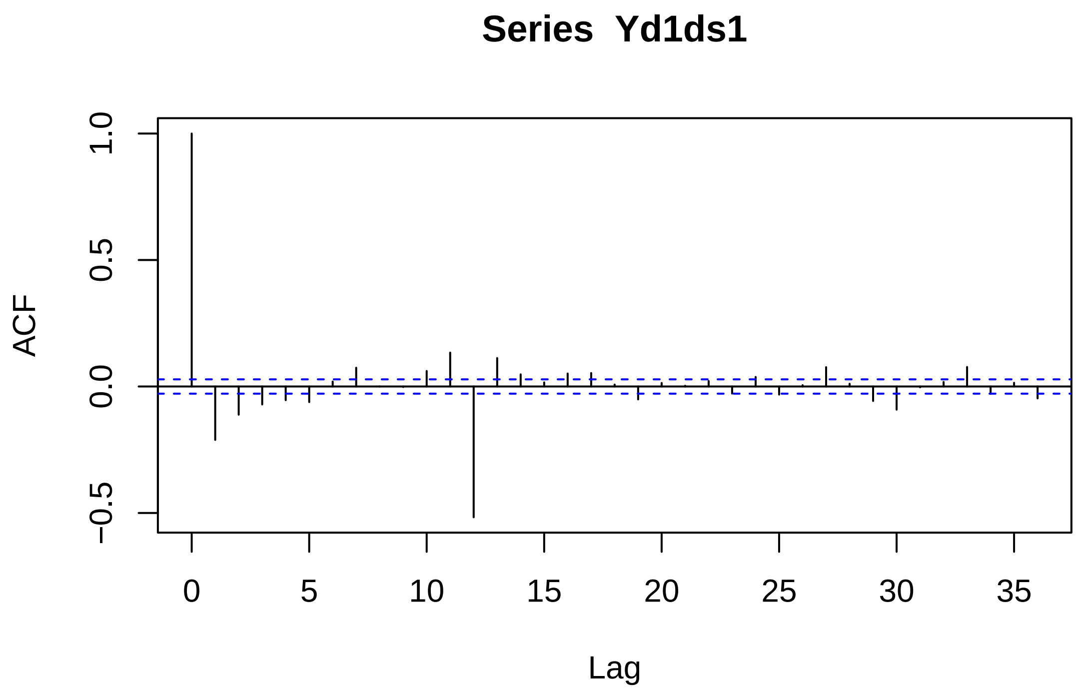

```
acf(Yd1)
```



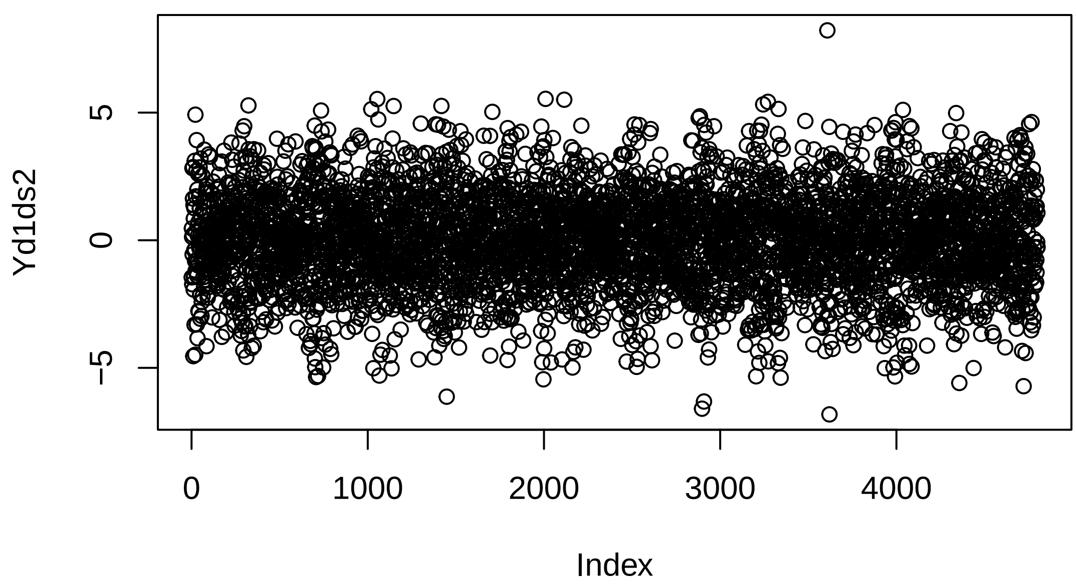

```r
Yd1ds1 =  diff(Yd1,lag=12)
plot(Yd1ds1)
```



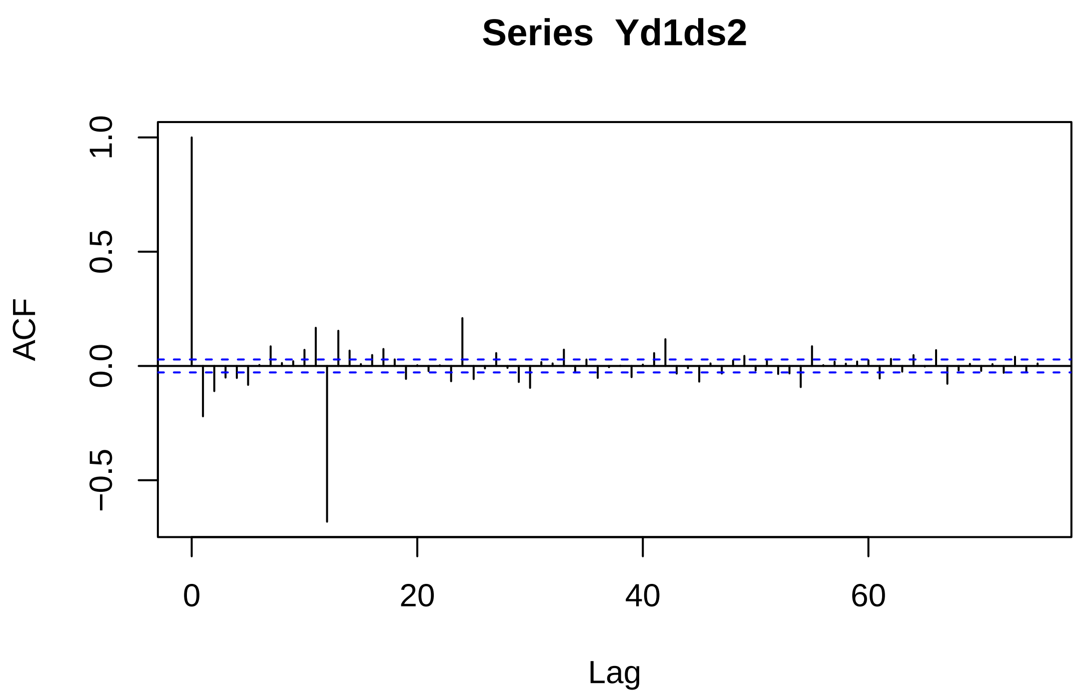
```
acf(Yd1ds1)
```



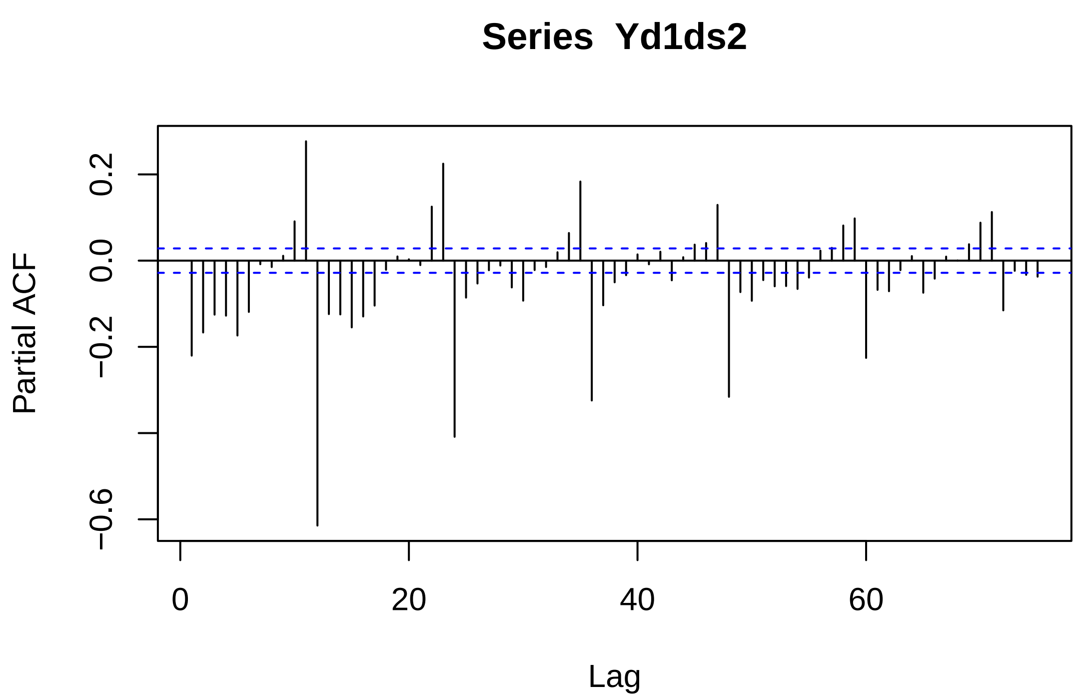

```
Yd1ds2 =  diff(Yd1ds1,lag=12)
plot(Yd1ds2)
```



```r
acf(Yd1ds2,lag=75)
```




```r
pacf(Yd1ds2,lag=75)
```



Par la différenciation, nous pouvons alors étudier la stationnarisation de la série. On observe qu'au degré 2, nous ne sommes pas encore dans un état stationnaire. En observant l'AFC et le PAFC on retrouve encore des périodes dépassant les bornes stationnaires, mais les pics sont moins élevés désormais. Pour ajuster notre modèle ARMA d'ordre (p,q) où p (resp. q) sont les nombres de pics sur le graphique AFC (resp. graphique PAFC), on a donc p=3 et q=6.

```r
source("armaic.r")

A=armaic(Yd1ds2,M=9,include.mean=FALSE)
```

```
## Warning in log(s2): Production de NaN

## Warning in log(s2): Production de NaN

## Warning in log(s2): Production de NaN

## Warning in log(s2): Production de NaN
```

```r
A
```

```
## $model
##
## Call:
## arima(x = x, order = c(p, 0, q), include.mean = FALSE, optim.control = list(maxit = 600))
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3      ma4      ma5     ma6
##       0.5704  -0.0092  -1.0113  0.0140  0.0095  -0.0095  -0.0140  1.0113
## s.e.  0.0148   0.0153   0.0037  0.0062  0.0077   0.0077   0.0062  0.0039
##          ma7
##       -1.0000
## s.e.   0.0014
##
## sigma^2 estimated as 1.633:  log likelihood = -8012.16,  aic = 16044.32
##
## $AIC
##          0        1        2        3        4        5        6        7
## 0 19270.74 18900.36 18663.87 18304.27 18185.95 18138.21 18140.20 16810.41
## 1 19033.65 18146.04 18147.35 18149.19 18146.41 17665.03 18142.19 16699.20
## 2 18900.36 18147.33 18149.72 17821.90 17797.87 17427.43 16836.08 16044.32
## 3 18826.43 18149.23 17814.20 17500.48 17373.93 17331.75 16143.43       NA
## 4 18749.54 18148.12 17785.60 17474.03 17065.11 16630.83       NA       NA
## 5 18604.52 18551.34 17773.57 17450.07 17060.89       NA       NA       NA
## 6 18538.03 18539.60 18031.66 17275.68       NA       NA       NA       NA
## 7 18539.69 18541.58 17612.19       NA       NA       NA       NA       NA
## 8 18540.63 17960.46       NA       NA       NA       NA       NA       NA
## 9 18542.01       NA       NA       NA       NA       NA       NA       NA
##          8        9
## 0 16614.17 16472.29
## 1 16454.73       NA
## 2       NA       NA
## 3       NA       NA
## 4       NA       NA
## 5       NA       NA
## 6       NA       NA
## 7       NA       NA
```

```
## 8       NA       NA
## 9       NA       NA
##
## $popt
## [1] 2
##
## $qopt
## [1] 7
##
## $aic
## [1] 16044.32
```

On choisit ainsi le modèle ARMA (2,7), celui qui minimise un maximum l'AIC.

```
out.Yd1ds2=A$model
ResY=out.Yd1ds2$residuals
p= 2
q= 7
Box.test(ResY,lag=10,type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ResY
## X-squared = 254.6, df = 10, p-value < 2.2e-16
```

Ainsi, avec une p-value < 2.2e-16, on peut donc rejeter l'hypothèse H0 où le modèle serait stationnaire.

```
out.Temp = arima(Temp$mean.temp,order=c(3,1,6),seasonal=list(order=c(0,2,0),period=12),include.mean = F
H = 50
predList = predict(out.Temp,n.ahead=H)
predList
```

```
## $pred
## Time Series:
## Start = 4827
## End = 4876
## Frequency = 1
##  [1] 13.41967 15.59638 14.72644 13.88821 12.72096 13.16386 14.60179 15.09902
##  [9] 15.99680 15.56962 15.23813 14.86547 16.52039 18.71330 17.37199 15.94266
## [17] 13.62659 14.27390 16.37223 17.60410 18.82452 18.10754 17.77311 17.27472
## [25] 19.56246 21.87194 20.07110 17.95943 14.52853 15.44166 18.14163 20.08423
## [33] 21.68897 20.67274 20.28732 19.69453 22.64091 25.02774 22.76432 20.00526
## [41] 15.44520 16.60130 19.92620 22.58775 24.55296 23.24201 22.82374 22.12391
## [49] 25.71956 28.19906
##
## $se
## Time Series:
## Start = 4827
## End = 4876
## Frequency = 1
##  [1]  1.292670  1.679335  2.013426  2.095710  2.095941  2.095946  2.117242
##  [8]  2.148299  2.149354  2.156470  2.157826  2.163782  3.241566  3.869677
## [15]  4.538180  4.676391  4.679749  4.680824  4.712572  4.754283  4.756080
## [22]  4.763394  4.764653  4.772356  5.976593  6.764908  7.650121  7.835444
## [29]  7.840446  7.842115  7.883907  7.938944  7.941364  7.950641  7.952283
```

```
## [36]  7.962404  9.275918 10.191535 11.264207 11.487315 11.494196 11.496668
## [43] 11.546131 11.610793 11.613673 11.624042 11.625888 11.637770 13.061328
## [50] 14.086965
```

A partir du modèle ARMA,il nous est alors possible de prédire les 50 prochaines observations à l'aide de la fonction ARIMA.