ProjetSeriesTemporelles

Description des données

Les données dont on dispose sont celles de la consommation de l'électricité depuis 1985 jusqu'à 2018 avec une période mensuelle .

ce dataset contient deux colonnes , la première est celle de la date et l'autre contient le pourcentage de consommation de l'eléctricité relevé mensuellement de 1985 à 2018 .

Les résultats obtenus

Récuperation des données

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.5
```

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

```r
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.5
```

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

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

```r
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.0.5
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

```r
data <- read.csv("C:/Electric_Production.csv", header=TRUE)
head(data,24)
```

```
##          DATE   Value
## 1  01-01-1985 72.5052
## 2  02-01-1985 70.6720
## 3  03-01-1985 62.4502
## 4  04-01-1985 57.4714
## 5  05-01-1985 55.3151
## 6  06-01-1985 58.0904
## 7  07-01-1985 62.6202
## 8  08-01-1985 63.2485
## 9  09-01-1985 60.5846
## 10 10-01-1985 56.3154
## 11 11-01-1985 58.0005
## 12 12-01-1985 68.7145
## 13 01-01-1986 73.3057
## 14 02-01-1986 67.9869
## 15 03-01-1986 62.2221
## 16 04-01-1986 57.0329
## 17 05-01-1986 55.8137
## 18 06-01-1986 59.9005
## 19 07-01-1986 65.7655
## 20 08-01-1986 64.4816
## 21 09-01-1986 61.0005
## 22 10-01-1986 57.5322
## 23 11-01-1986 59.3417
## 24 12-01-1986 68.1354
```

```r
summary(data)
```

```
##      DATE               Value       
##  Length:397         Min.   : 55.32  
##  Class :character   1st Qu.: 77.11  
##  Mode  :character   Median : 89.78  
##                     Mean   : 88.85  
##                     3rd Qu.:100.52  
##                     Max.   :129.40  
```

Vérification de l'existence des valeurs manquantes

```r
sum(is.na(data))
```

```
## [1] 0
```

Construction d'une série chronologique à la base des données dont on dispose

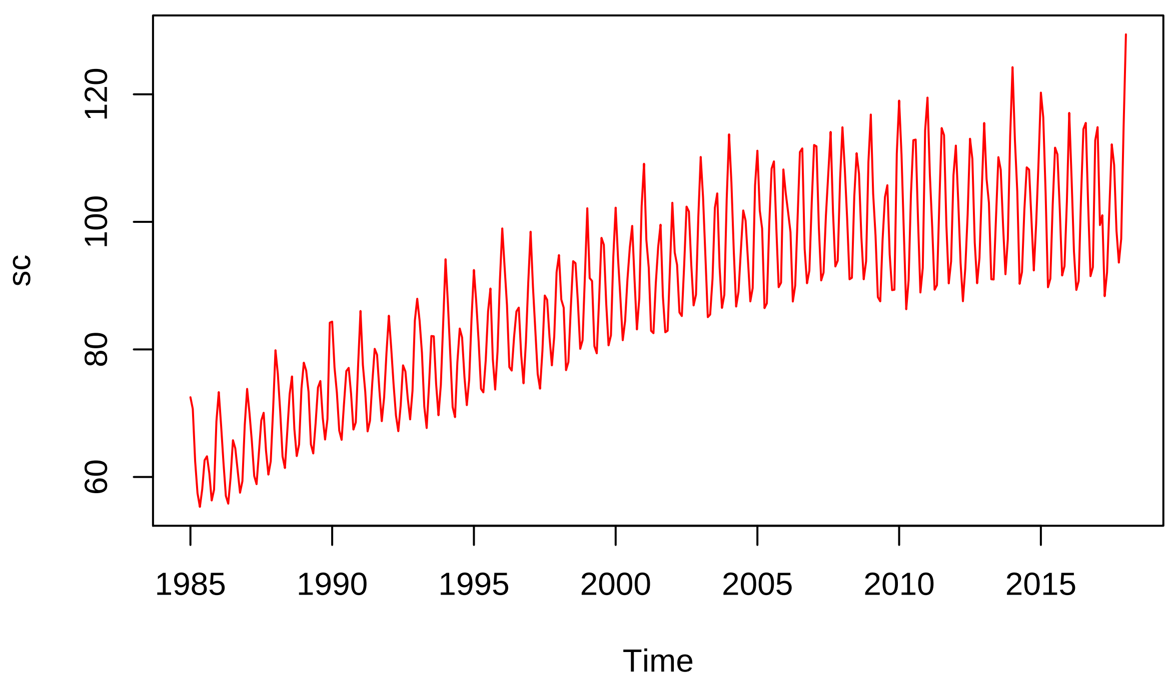
```r
sc <- ts(data$Value, start=c(1985, 1), end=c(2018, 1), frequency=12)
sc
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1985  72.5052  70.6720  62.4502  57.4714  55.3151  58.0904  62.6202  63.2485
## 1986  73.3057  67.9869  62.2221  57.0329  55.8137  59.9005  65.7655  64.4816
## 1987  73.8152  70.0620  65.6100  60.1586  58.8734  63.8918  68.8694  70.0669
## 1988  79.8703  76.1622  70.2928  63.2384  61.4065  67.1097  72.9816  75.7655
## 1989  77.9188  76.6822  73.3523  65.1081  63.6892  68.4722  74.0301  75.0448
## 1990  84.3598  77.1726  73.1964  67.2781  65.8218  71.4654  76.6140  77.1052
## 1991  86.0214  77.5573  73.3650  67.1500  68.8162  74.8448  80.0928  79.1606
## 1992  85.2855  80.1643  74.5275  69.6441  67.1784  71.2078  77.5081  76.5374
## 1993  87.9464  84.5561  79.4747  71.0578  67.6762  74.3297  82.1048  82.0605
## 1994  94.1386  87.1607  79.2456  70.9749  69.3844  77.9831  83.2770  81.8872
## 1995  92.4532  87.4033  81.2661  73.8167  73.2682  78.3026  85.9841  89.5467
## 1996  98.9732  92.8883  86.9356  77.2214  76.6826  81.9306  85.9606  86.5562
## 1997  98.4613  89.7795  83.0125  76.1476  73.8471  79.7645  88.4519  87.7828
## 1998  94.7920  87.8200  86.5549  76.7521  78.0303  86.4579  93.8379  93.5310
## 1999 102.1348  91.1829  90.7381  80.5176  79.3887  87.8431  97.4903  96.4157
## 2000 102.2301  94.2989  88.0927  81.4425  84.4552  91.0406  95.9957  99.3704
## 2001 109.1081  97.1717  92.8283  82.9150  82.5465  90.3955  96.0740  99.5534
## 2002 102.9955  95.2075  93.2556  85.7950  85.2351  93.1896 102.3930 101.6293
## 2003 110.1807 103.8413  94.5532  85.0620  85.4653  91.0761 102.2200 104.4682
## 2004 113.7226 106.1590  95.4029  86.7233  89.0302  95.5045 101.7948 100.2025
## 2005 111.1614 101.7795  98.9565  86.4776  87.2234  99.5076 108.3501 109.4862
## 2006 104.4724 101.5196  98.4017  87.5093  90.0222 100.5244 110.9503 111.5192
## 2007 112.0576 111.8399  99.1925  90.8177  92.0587 100.9676 107.5686 114.1036
## 2008 114.8331 108.2353 100.4386  90.9944  91.2348 103.9581 110.7631 107.5665
## 2009 116.8316 104.4202  97.8529  88.1973  87.5366  97.2387 103.9086 105.7486
## 2010 119.0166 110.5330  98.2672  86.3000  90.8364 104.3538 112.8066 112.9014
## 2011 119.4880 107.3753  99.1028  89.3583  90.0698 102.8204 114.7068 113.5958
## 2012 111.9646 103.3679  93.5772  87.5566  92.7603 101.1400 113.0357 109.8601
## 2013 115.5010 106.7340 102.9948  91.0092  90.9634 100.6957 110.1480 108.1756
## 2014 124.2549 112.8811 104.7631  90.2867  92.1340 101.8780 108.5497 108.1940
## 2015 120.2696 116.3788 104.4706  89.7461  91.0930 102.6495 111.6354 110.5925
## 2016 117.0837 106.6688  95.3548  89.3254  90.7369 104.0375 114.5397 115.5159
## 2017 114.8505  99.4901 101.0396  88.3530  92.0805 102.1532 112.1538 108.9312
## 2018 129.4048
##           Sep      Oct      Nov      Dec
## 1985  60.5846  56.3154  58.0005  68.7145
## 1986  61.0005  57.5322  59.3417  68.1354
## 1987  64.1151  60.3789  62.4643  70.5777
## 1988  67.5152  63.2832  65.1078  73.8631
## 1989  69.3053  65.8735  69.0706  84.1949
## 1990  73.0610  67.4365  68.5665  77.6839
## 1991  73.5743  68.7538  72.5166  79.4894
## 1992  72.3541  69.0286  73.4992  84.5159
## 1993  74.6031  69.6810  74.4292  84.2284
## 1994  75.6826  71.2661  75.2458  84.8147
## 1995  78.5035  73.7066  79.6543  90.8251
## 1996  79.1919  74.6891  81.0740  90.4855
## 1997  81.9386  77.5027  82.0448  92.1010
## 1998  87.5414  80.0924  81.4349  91.6841
```

```
## 1999  87.2248  80.6409  82.2025  94.5113
## 2000  90.9178  83.1408  88.0410 102.4558
## 2001  88.2810  82.6860  82.9319  93.0381
## 2002  93.3089  86.9002  88.5749 100.8003
## 2003  92.9135  86.5047  88.5735 103.5428
## 2004  94.0240  87.5262  89.6144 105.7263
## 2005  99.1155  89.7567  90.4587 108.2257
## 2006  95.7632  90.3738  92.3566 103.0660
## 2007 101.5316  93.0068  93.9126 106.7528
## 2008  97.7183  90.9979  93.8057 109.4221
## 2009  94.8823  89.2977  89.3585 110.6844
## 2010 100.1209  88.9251  92.7750 114.3266
## 2011  99.4712  90.3566  93.8095 107.3312
## 2012  96.7431  90.3805  94.3417 105.2722
## 2013  99.2809  91.7871  97.2853 113.4732
## 2014 100.4172  92.3837  99.7033 109.3477
## 2015 101.9204  91.5959  93.0628 103.2203
## 2016 102.7637  91.4867  92.8900 112.7694
## 2017  98.6154  93.6137  97.3359 114.7212
## 2018
```

Représentation graphique de la série chronologique

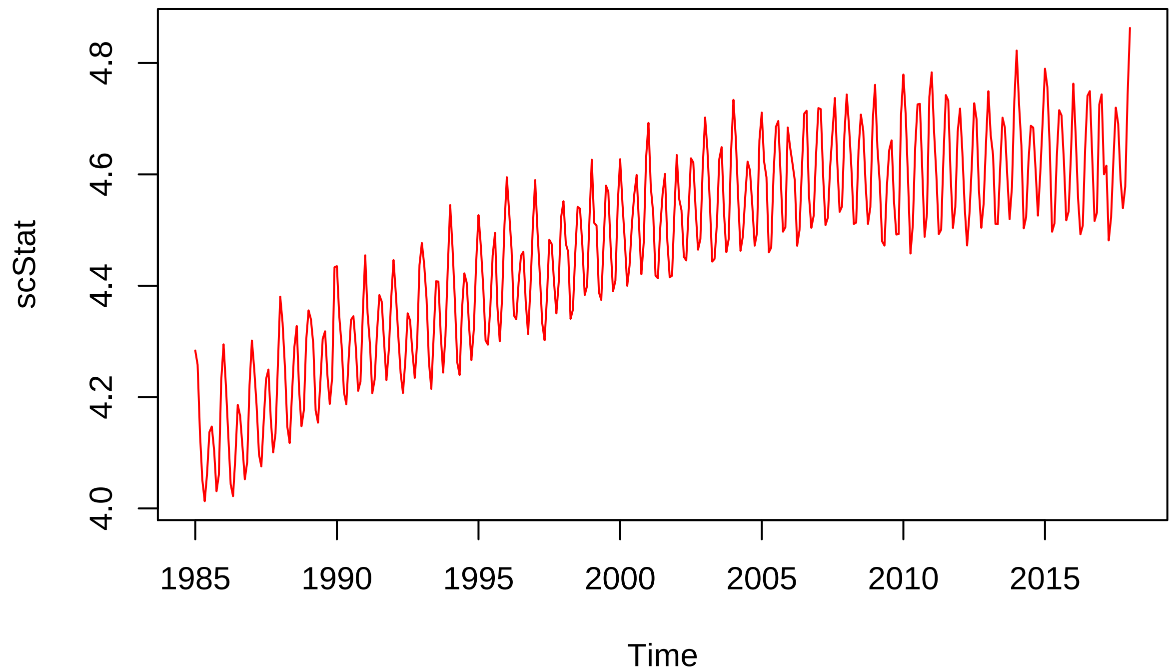
```r
plot(sc, type='l',col='red')
```



il est bien evident qu'on a un modèle multiplicatif et qu'il existe une tendance et une saisonnalité.

Passage d'un modèle multiplicatif à un modèle additif

```
scStat <- log(sc)
plot(scStat, type='l',col='red')
```

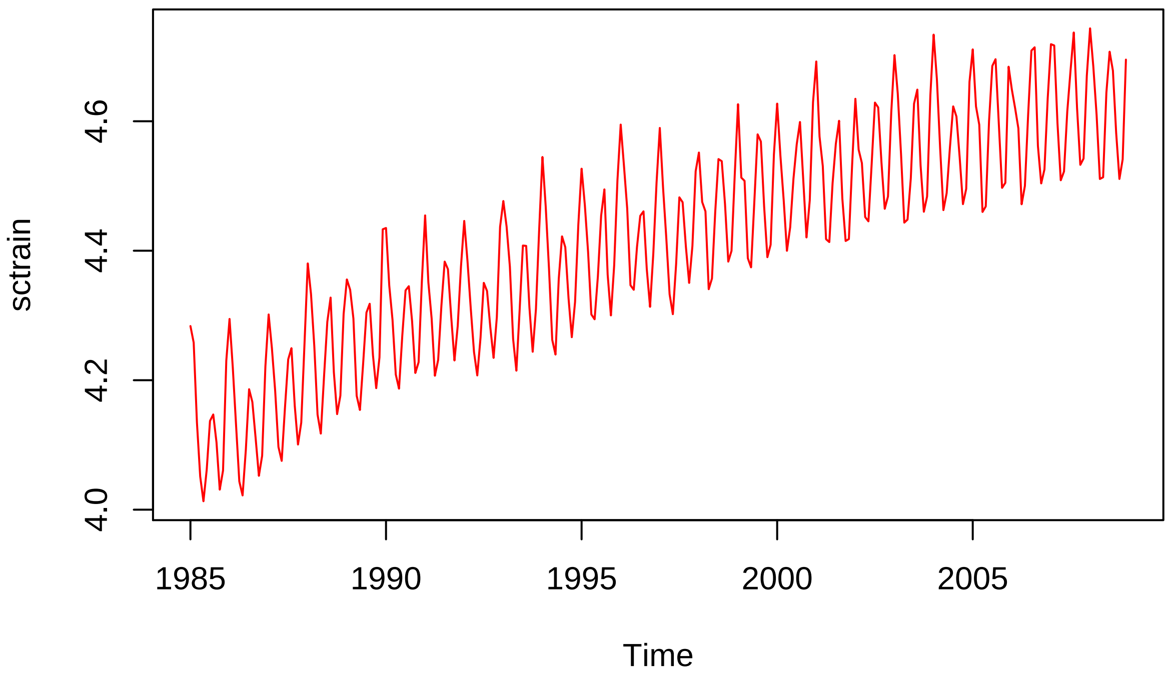


Division des données en un échantillon d'apprentissage et un échantillon de validation

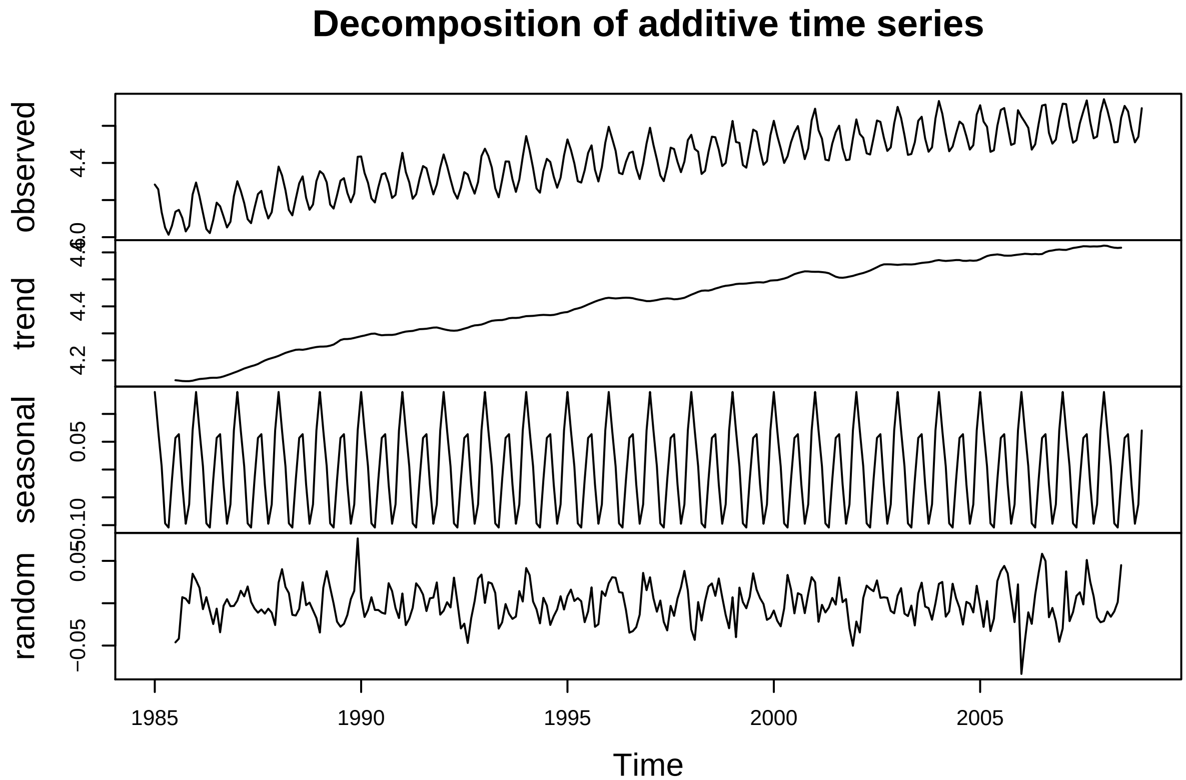
```
sctrain <- window(scStat,end=c(2008,12))
sctest <- window(scStat,start=c(2009,1))
```

Détection de la tendance et de la saisonnalité

```
plot(sctrain, type='l',col='red')
```




```
plot(decompose(sctrain))
```




```r
decompose(sctrain)$seasonal
```

```
##              Jan          Feb          Mar          Apr          May
## 1985  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1986  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1987  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1988  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1989  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1990  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1991  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1992  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1993  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1994  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1995  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1996  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1997  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1998  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 1999  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2000  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2001  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2002  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2003  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2004  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2005  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2006  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
```

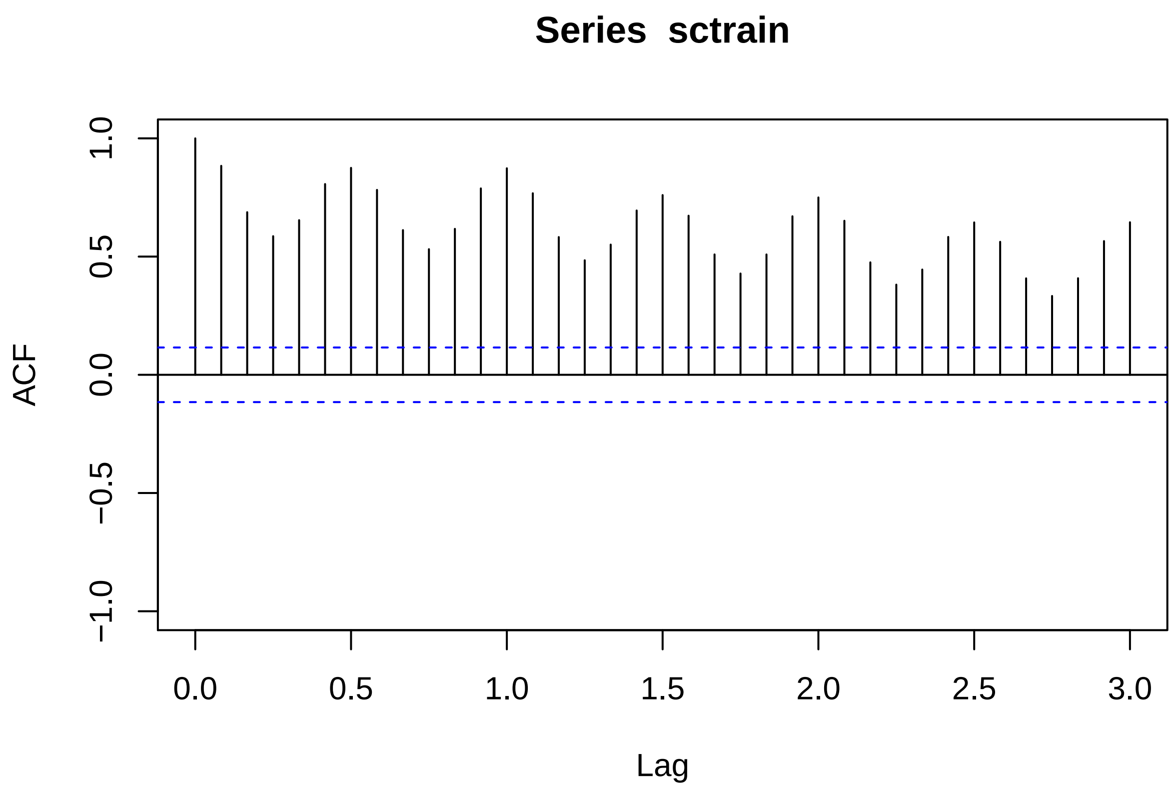
```
## 2007  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
## 2008  0.139447609  0.070280261  0.005697985 -0.096788488 -0.104302807
##               Jun          Jul          Aug          Sep          Oct
## 1985 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1986 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1987 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1988 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1989 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1990 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1991 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1992 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1993 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1994 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1995 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1996 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1997 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1998 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 1999 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2000 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2001 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2002 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2003 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2004 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2005 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2006 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2007 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
## 2008 -0.018354339  0.056868160  0.063609439 -0.026684085 -0.097467442
##               Nov          Dec
## 1985 -0.062630397  0.070324103
## 1986 -0.062630397  0.070324103
## 1987 -0.062630397  0.070324103
## 1988 -0.062630397  0.070324103
## 1989 -0.062630397  0.070324103
## 1990 -0.062630397  0.070324103
## 1991 -0.062630397  0.070324103
## 1992 -0.062630397  0.070324103
## 1993 -0.062630397  0.070324103
## 1994 -0.062630397  0.070324103
## 1995 -0.062630397  0.070324103
## 1996 -0.062630397  0.070324103
## 1997 -0.062630397  0.070324103
## 1998 -0.062630397  0.070324103
## 1999 -0.062630397  0.070324103
## 2000 -0.062630397  0.070324103
## 2001 -0.062630397  0.070324103
## 2002 -0.062630397  0.070324103
## 2003 -0.062630397  0.070324103
## 2004 -0.062630397  0.070324103
## 2005 -0.062630397  0.070324103
## 2006 -0.062630397  0.070324103
## 2007 -0.062630397  0.070324103
## 2008 -0.062630397  0.070324103
```

```r
decompose(sctrain)$trend
```

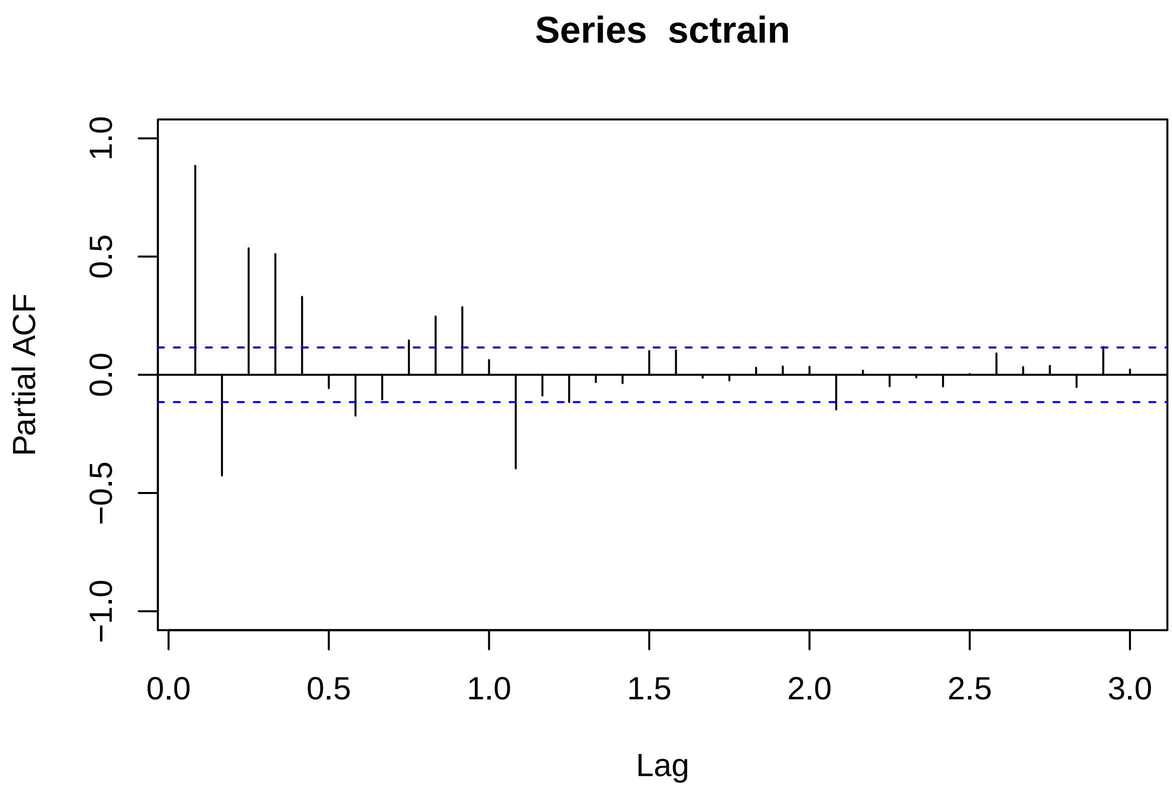
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1985       NA       NA       NA       NA       NA       NA 4.126457 4.125300
## 1986 4.128090 4.130936 4.132026 4.133202 4.135045 4.135645 4.135581 4.137122
## 1987 4.158984 4.164367 4.169903 4.173990 4.178139 4.181743 4.186496 4.193259
## 1988 4.216665 4.222340 4.227751 4.231861 4.235546 4.239169 4.240034 4.239287
## 1989 4.250860 4.251056 4.251748 4.254510 4.258643 4.266560 4.275324 4.278899
## 1990 4.289459 4.292017 4.295345 4.298521 4.299193 4.295534 4.292993 4.294013
## 1991 4.303661 4.306607 4.307995 4.309093 4.312233 4.315524 4.316123 4.317143
## 1992 4.315344 4.312573 4.310473 4.309942 4.310669 4.313785 4.317619 4.321122
## 1993 4.336966 4.342270 4.346449 4.348116 4.349032 4.349414 4.352107 4.356206
## 1994 4.363799 4.364301 4.364812 4.366348 4.367740 4.368483 4.368020 4.367383
## 1995 4.379080 4.384139 4.389390 4.392317 4.396093 4.401318 4.407010 4.412385
## 1996 4.431858 4.430431 4.429380 4.430295 4.431583 4.432163 4.431791 4.430156
## 1997 4.419541 4.421317 4.423324 4.426286 4.428323 4.429556 4.428711 4.426209
## 1998 4.443200 4.448306 4.453705 4.457830 4.458889 4.458389 4.461308 4.465983
## 1999 4.479827 4.482683 4.483798 4.483932 4.484607 4.486263 4.487568 4.489006
## 2000 4.496384 4.496998 4.499983 4.502983 4.507114 4.513336 4.519412 4.523376
## 2001 4.528019 4.528130 4.526980 4.525526 4.522806 4.516298 4.509878 4.506625
## 2002 4.516864 4.520378 4.523546 4.527925 4.532739 4.538821 4.544969 4.551396
## 2003 4.553692 4.554770 4.555741 4.555374 4.555183 4.556301 4.558738 4.560976
## 2004 4.571441 4.569530 4.568288 4.569272 4.570248 4.571605 4.571525 4.568821
## 2005 4.574190 4.580482 4.586371 4.589617 4.591056 4.592421 4.590808 4.588116
## 2006 4.592997 4.594751 4.594084 4.592936 4.594087 4.592917 4.593802 4.600756
## 2007 4.609491 4.609155 4.612547 4.616181 4.618074 4.620234 4.622718 4.622372
## 2008 4.625112 4.623873 4.619820 4.617315 4.616358 4.617340       NA       NA
##           Sep      Oct      Nov      Dec
## 1985 4.123534 4.123062 4.123117 4.124769
## 1986 4.140584 4.145016 4.149463 4.154375
## 1987 4.199610 4.204563 4.208399 4.212201
## 1988 4.241345 4.244335 4.247070 4.249428
## 1989 4.279076 4.280354 4.283092 4.286247
## 1990 4.294316 4.294333 4.296107 4.299886
## 1991 4.319176 4.321350 4.321866 4.318787
## 1992 4.326022 4.329538 4.330682 4.332778
## 1993 4.357350 4.357181 4.358171 4.361209
## 1994 4.368547 4.371232 4.375137 4.377577
## 1995 4.417731 4.422420 4.426197 4.429982
## 1996 4.426814 4.424306 4.422153 4.419467
## 1997 4.427031 4.429101 4.431727 4.437380
## 1998 4.469515 4.473477 4.476192 4.477574
## 1999 4.489174 4.488417 4.491470 4.495538
## 2000 4.526808 4.529736 4.529531 4.528282
## 2001 4.505965 4.507579 4.510337 4.512941
## 2002 4.555589 4.555807 4.555562 4.554718
## 2003 4.562268 4.563447 4.565955 4.569636
## 2004 4.568589 4.569995 4.569022 4.569879
## 2005 4.587775 4.588035 4.589845 4.591585
## 2006 4.605124 4.607003 4.609482 4.610597
## 2007 4.621528 4.622129 4.621835 4.622677
## 2008       NA       NA       NA       NA
```

les corrélogrammes simple et partiel de la série

```
plot(acf(sctrain,lag.max=36,plot=FALSE),ylim=c(-1,1))
```




```
plot(pacf(sctrain,lag.max=36,plot=FALSE),ylim=c(-1,1))
```



Tests de stationnarité

```
adf.test(sctrain)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sctrain
## Dickey-Fuller = -6.1311, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(sctrain)
```

```
## Warning in kpss.test(sctrain): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  sctrain
## KPSS Level = 4.6627, Truncation lag parameter = 5, p-value = 0.01
```

```r
pp.test(sctrain)
```

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

```
##
##  Phillips-Perron Unit Root Test
##
## data:  sctrain
## Dickey-Fuller Z(alpha) = -63.996, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

```r
PP.test(sctrain)
```

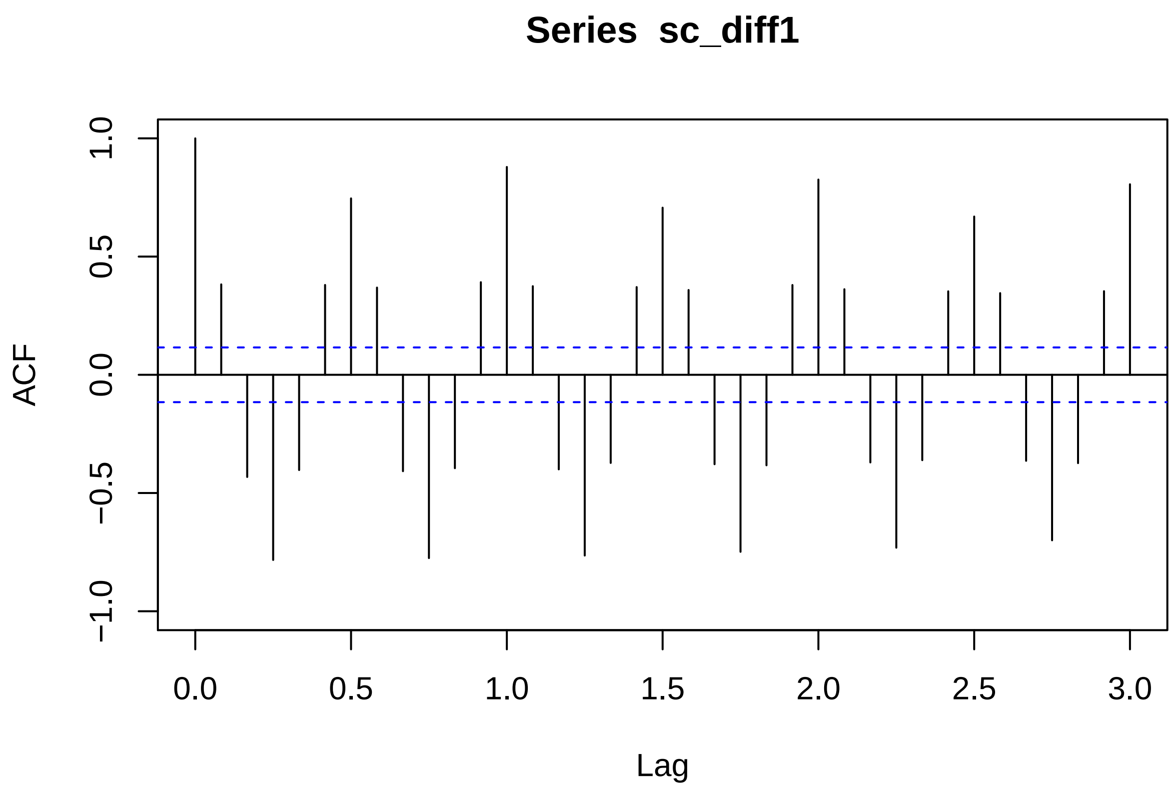
```
##
##  Phillips-Perron Unit Root Test
##
## data:  sctrain
## Dickey-Fuller = -7.7939, Truncation lag parameter = 5, p-value = 0.01
```

la p-value est inferieure a 0.05 donc la serie est stationnaire

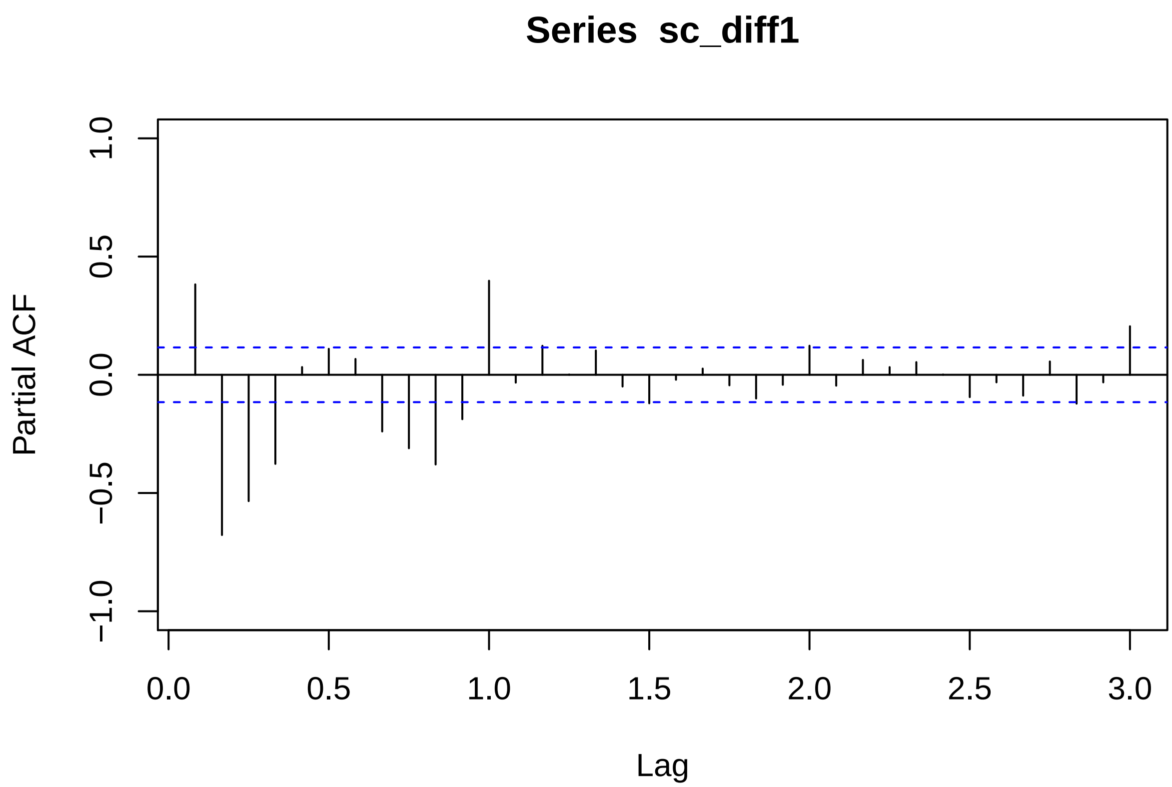
Différenciation (I-B) de la série(élimination de la tendance)

```r
sc_diff1=diff(sctrain,lag=1,differences=1)

plot(acf(sc_diff1,lag.max=36,plot=FALSE),ylim=c(-1,1))
```




```
plot(pacf(sc_diff1,lag.max=36,plot=FALSE),ylim=c(-1,1))
```



on remarque que le correlogramme partiel s'annule a partir de 3

##Tests de stationnarité ##

```r
adf.test(sc_diff1)
```

```
## Warning in adf.test(sc_diff1): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sc_diff1
## Dickey-Fuller = -7.8363, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(sc_diff1)
```

```
## Warning in kpss.test(sc_diff1): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  sc_diff1
## KPSS Level = 0.054293, Truncation lag parameter = 5, p-value = 0.1
```

```r
pp.test(sc_diff1)
```

```
## Warning in pp.test(sc_diff1): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  sc_diff1
## Dickey-Fuller Z(alpha) = -64.311, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

```r
PP.test(sc_diff1)
```

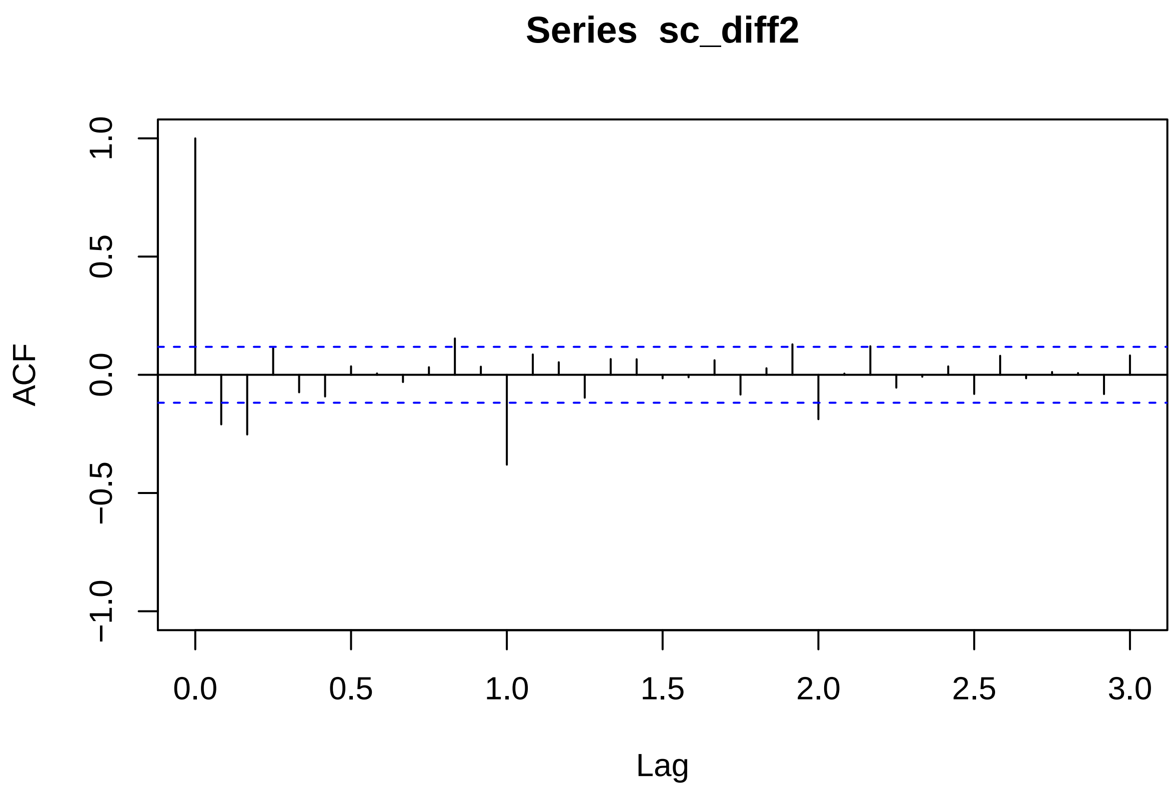
```
##
##  Phillips-Perron Unit Root Test
##
## data:  sc_diff1
## Dickey-Fuller = -12.724, Truncation lag parameter = 5, p-value = 0.01
```

la p-value est superieure a 0.05 donc la série n'est pas stationnaire.

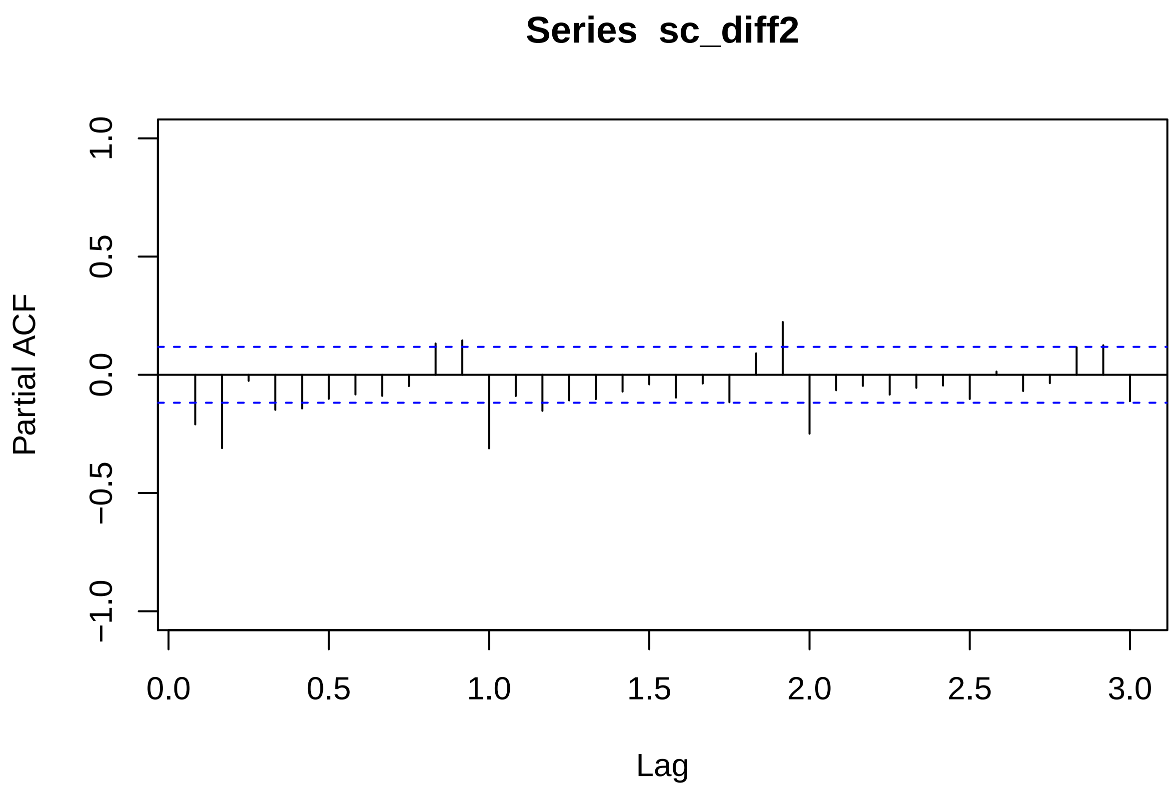
Différenciation (I-B^12) saisonniére

```r
sc_diff2=diff(sc_diff1,lag=12,differences=1)

plot(acf(sc_diff2,lag.max=36,plot=FALSE),ylim=c(-1,1))
```




```
plot(pacf(sc_diff2,lag.max=36,plot=FALSE),ylim=c(-1,1))
```



les corrélogrammes simple et partiel s'annulent tous les deux à partir de 2.

```
########## Tests de stationnarité ###########

adf.test(sc_diff2)
```

```
## Warning in adf.test(sc_diff2): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  sc_diff2
## Dickey-Fuller = -9.1131, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(sc_diff2)
```

```
## Warning in kpss.test(sc_diff2): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  sc_diff2
## KPSS Level = 0.012353, Truncation lag parameter = 5, p-value = 0.1
```

```r
pp.test(sc_diff2)
```

```
## Warning in pp.test(sc_diff2): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  sc_diff2
## Dickey-Fuller Z(alpha) = -262.38, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

```r
PP.test(sc_diff2)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  sc_diff2
## Dickey-Fuller = -23.63, Truncation lag parameter = 5, p-value = 0.01
```

la p-value est inferieure a 0.05 donc la serie est stationnaire.

Identification , estimation et validation des modèles

Modèle 1

```r
model1=arima(sctrain,order=c(3,1,2),list(order=c(1,1,2),period=12), include.mean=FALSE,method="CSS-ML")
coeftest(model1)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ar1   0.096661   0.307475  0.3144  0.75324
## ar2   0.172476   0.174413  0.9889  0.32272
## ar3   0.094602   0.068605  1.3789  0.16791
## ma1  -0.509448   0.304801 -1.6714  0.09464 .
## ma2  -0.411685   0.282958 -1.4549  0.14569
## sar1 -0.525302   0.324379 -1.6194  0.10536
## sma1 -0.196720   0.309405 -0.6358  0.52491
## sma2 -0.488371   0.235978 -2.0696  0.03849 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modèle 2

```r
model2=arima(sctrain,order=c(2,1,3),list(order=c(1,1,2),period=12),include.mean=FALSE,method="CSS-ML")
coeftest(model2)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.17146    0.36381  0.4713  0.63744
## ar2   0.26254    0.17929  1.4643  0.14311
## ma1  -0.57795    0.36160 -1.5983  0.10997
## ma2  -0.47007    0.29280 -1.6054  0.10840
## ma3   0.11965    0.10731  1.1150  0.26484
## sar1 -0.54075    0.32539 -1.6618  0.09655 .
## sma1 -0.18346    0.30950 -0.5928  0.55334
## sma2 -0.50341    0.23661 -2.1276  0.03337 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modèle 3

```r
model3=arima(sctrain,order=c(1,1,1),list(order=c(0,1,2),period=12),include.mean=FALSE,method="CSS-ML")
coeftest(model3)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.507373   0.063632   7.9736 1.541e-15 ***
## ma1  -0.934193   0.027795 -33.6106 < 2.2e-16 ***
## sma1 -0.765094   0.073102 -10.4661 < 2.2e-16 ***
## sma2 -0.045372   0.074993  -0.6050    0.5452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modèle 4

```r
model4=arima(sctrain,order=c(1,1,1),list(order=c(1,1,2),period=12),include.mean=FALSE,method="CSS-ML")
coeftest(model4)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.514299   0.062897   8.1769 2.913e-16 ***
## ma1  -0.935654   0.027006 -34.6460 < 2.2e-16 ***
## sar1 -0.584775   0.327317  -1.7866   0.07401 .
## sma1 -0.153132   0.309518  -0.4947   0.62078
## sma2 -0.545474   0.239384  -2.2787   0.02269 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Validation du Modèle4 à l’aide des méthodes Box et Jenkins

vérifier si les résidus sont un bruit blanc

```r
Box.test(model4$residuals,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  model4$residuals
## X-squared = 8.5824e-05, df = 1, p-value = 0.9926
```

```r
Box.test(model4$residuals,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  model4$residuals
## X-squared = 8.4936e-05, df = 1, p-value = 0.9926
```

```r
Box.test(model4$residuals,type="Box-Pierce")$p.value
```

```
## [1] 0.9926467
```

on remarque que la p-value est superieure à 0.05 donc les résidus sont des bruits blancs.

Test de normalité des résidus du modèle4

```r
shapiro.test(model4$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model4$residuals
## W = 0.98727, p-value = 0.01212
```

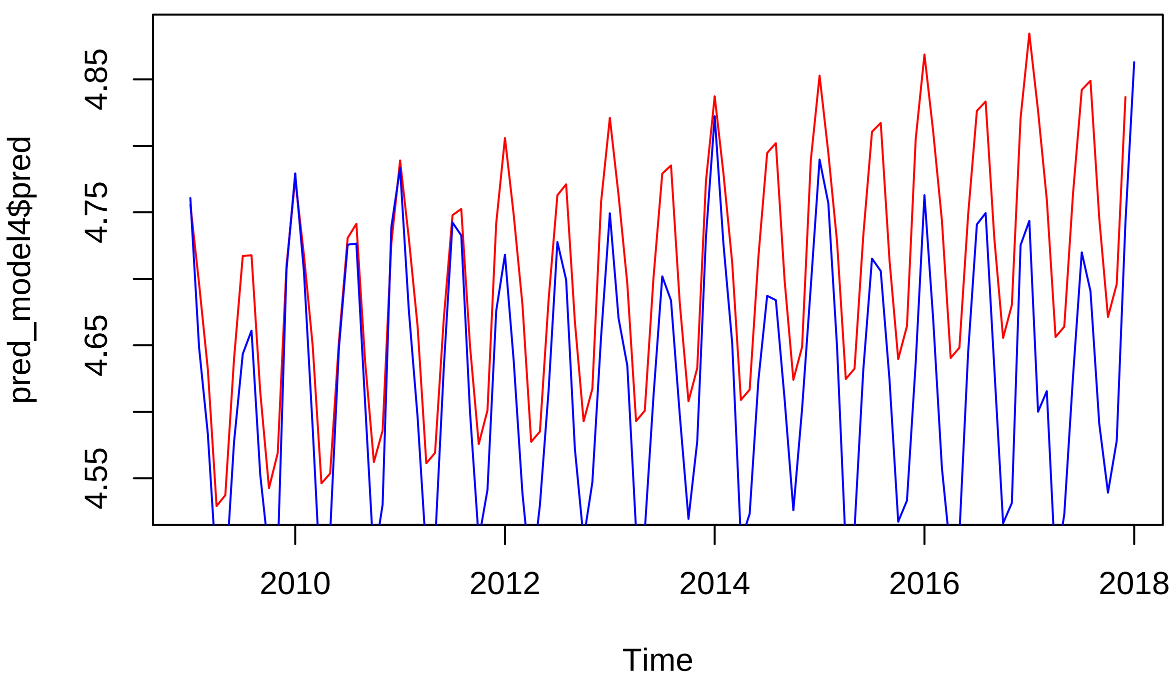
```r
jarque.bera.test(model4$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  model4$residuals
## X-squared = 25.717, df = 2, p-value = 2.604e-06
```

on a p-value pour les deux tests est inferieure à 0.05 donc les résidus ne suivent pas une loi normale.

Prévision

```r
pred_model4=predict(model4,n.ahead=108)
plot(pred_model4$pred , col="red")
lines(sctest, col="blue")
```



RMSE et MAPE

```r
rmse=sqrt(mean((sctest - pred_model4$pred)^2))
rmse
```

```
## [1] 0.09420565
```

```r
mape=mean(abs(1-pred_model4$pred/sctest))*100
mape
```

```
## [1] 1.810346
```

le rmse est suffisemment petit pour q'on puisse qualifier notre modèle4 d'efficace et Donc le modèle4 correspond bien à notre serie. Ceci est confirmé par la fonction auto.arima suivante:

```
auto.arima(sctrain, trace=TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,1,1)[12]                    : -1186.419
##  ARIMA(0,1,0)(0,1,0)[12]                    : -1016.439
##  ARIMA(1,1,0)(1,1,0)[12]                    : -1067.822
##  ARIMA(0,1,1)(0,1,1)[12]                    : -1154.772
##  ARIMA(2,1,2)(0,1,1)[12]                    : -1191.129
##  ARIMA(2,1,2)(0,1,0)[12]                    : Inf
##  ARIMA(2,1,2)(0,1,2)[12]                    : -1189.244
##  ARIMA(2,1,2)(1,1,0)[12]                    : -1124.799
##  ARIMA(2,1,2)(1,1,2)[12]                    : -1192.457
##  ARIMA(2,1,2)(2,1,2)[12]                    : -1180.513
##  ARIMA(2,1,2)(2,1,1)[12]                    : -1182.636
##  ARIMA(1,1,2)(1,1,2)[12]                    : -1191.419
##  ARIMA(2,1,1)(1,1,2)[12]                    : -1187.709
##  ARIMA(3,1,2)(1,1,2)[12]                    : -1190.539
##  ARIMA(2,1,3)(1,1,2)[12]                    : -1190.595
##  ARIMA(1,1,1)(1,1,2)[12]                    : -1193.493
##  ARIMA(1,1,1)(0,1,2)[12]                    : -1147.579
##  ARIMA(1,1,1)(1,1,1)[12]                    : -1193.035
##  ARIMA(1,1,1)(2,1,2)[12]                    : -1185.947
##  ARIMA(1,1,1)(0,1,1)[12]                    : -1149.04
##  ARIMA(1,1,1)(2,1,1)[12]                    : -1188.053
##  ARIMA(0,1,1)(1,1,2)[12]                    : -1164.104
##  ARIMA(1,1,0)(1,1,2)[12]                    : -1149.443
##  ARIMA(0,1,0)(1,1,2)[12]                    : -1135.92
##  ARIMA(0,1,2)(1,1,2)[12]                    : -1182.89
##  ARIMA(2,1,0)(1,1,2)[12]                    : -1169.306
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,1,1)(1,1,2)[12]                    : -1276.387
##
##  Best model: ARIMA(1,1,1)(1,1,2)[12]
```

```
## Series: sctrain
## ARIMA(1,1,1)(1,1,2)[12]
##
## Coefficients:
##          ar1      ma1     sar1     sma1     sma2
##       0.5143  -0.9357  -0.5848  -0.1531  -0.5455
## s.e.  0.0629   0.0270   0.3273   0.3095   0.2394
##
## sigma^2 estimated as 0.0005226:  log likelihood=644.35
## AIC=-1276.7   AICc=-1276.39   BIC=-1255
```