Test PMLSeg

Ninh

2024-08-09

Intro

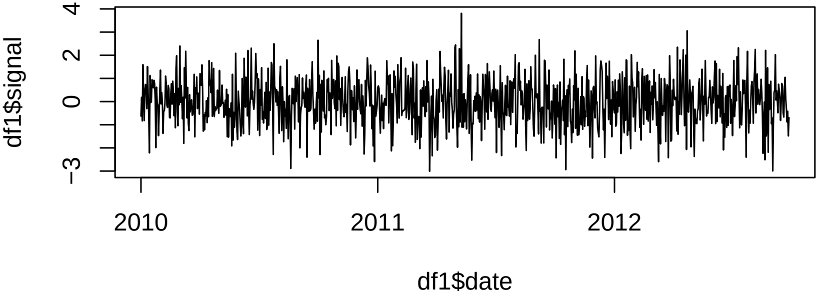
This document present result of several test of the PMLSeg package consisting:

- Test of the `Segmentation` function in case with/ without offsets for the following examples :
  - Ex1 : zero mean + IID noise
  - Ex2 : periodic mean + IID noise
  - Ex3 : periodic mean + monthly variance
- Test of other functions such as:
  - `PlotSeg` to visualize segmentation results
  - `Cluster_screening` to detect the group of close change-points (ussualy due to the outliers) and check if it is needed to keep or remove the cluster.
  - `Validation` to validate the detected changepoints with the help of metadata.

Generate example data to test

Ex1 time series

```r
set.seed(1)
length_series = 1000
df1 = data.frame(date = seq.Date(from = as.Date("2010-01-01"),
                                 to = as.Date("2010-01-01")+(length_series-1),
                                 by = "day"),
                signal = rnorm(n = length_series, mean = 0, sd = 1))

plot(df1$date, df1$signal, type = "l")
```

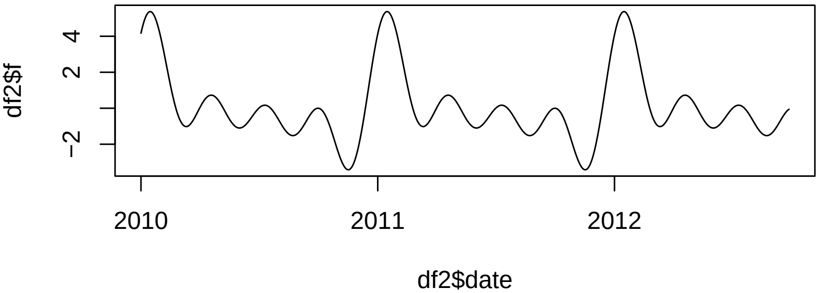


```r
head(df1, 3)
#>         date     signal
#> 1 2010-01-01 -0.6264538
#> 2 2010-01-02  0.1836433
#> 3 2010-01-03 -0.8356286
```

Generate example data to test

Ex2 time series : add the functional with 4 Fourier series with coefficient = 1

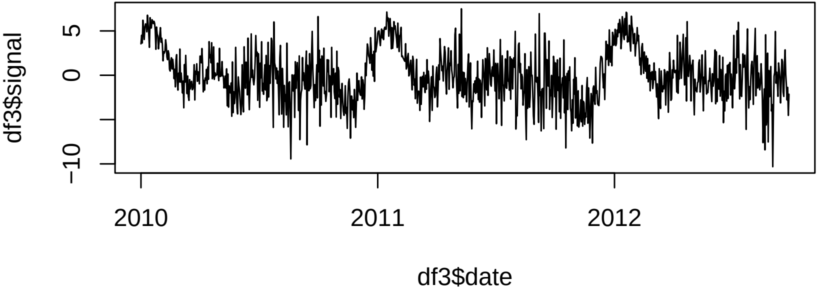
```
library(dplyr)

T <- 365.25
df2 <- df1 %>%
  mutate(t = as.numeric(date - date[1])+1,
  f = rowSums(sapply(1:4, function(i) cos(i*t*(2*pi)/T) + sin(i*t*(2*pi)/T))),
  signal = signal + f)

head(df2, 3)
#>         date   signal t        f
#> 1 2010-01-01 3.541048 1 4.167502
#> 2 2010-01-02 4.509279 2 4.325636
#> 3 2010-01-03 3.638312 3 4.473941
plot(df2$date, df2$f, type = "l")
```



Generate example data to test

Ex3 time series : add the functional with 4 Fourier series with coefficient = 1

```r
std = c(1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 2.5, 2, 1.5)

df3 <- df1 %>%
  mutate(sd = std[as.numeric(format(date, "%m"))],
         signal = signal * sd ) %>%
  mutate(signal = signal + df2$f)

head(df3, 3)
#>         date   signal sd
#> 1 2010-01-01 3.541048  1
#> 2 2010-01-02 4.509279  1
#> 3 2010-01-03 3.638312  1
plot(df3$date, df3$signal, type = "l")
```



Generate example data to test

Harmmonize format of 3 dataframes to test:

```r
df2 <- df2 %>% select(date, signal)
df3 <- df3 %>% select(date, signal)

names(df1)
#> [1] "date"   "signal"
names(df2)
#> [1] "date"   "signal"
names(df3)
#> [1] "date"   "signal"
```

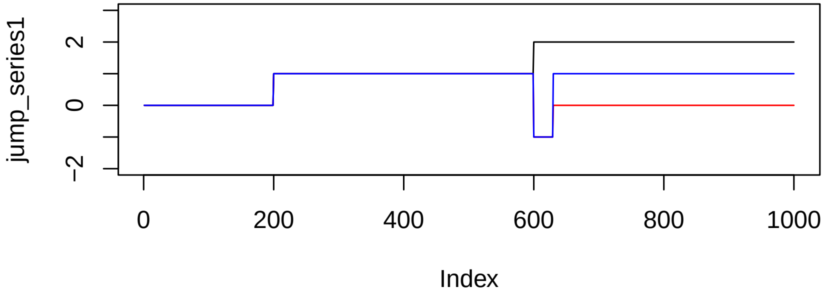
Premilinary setting

Generate different offset series to add into the original series (which is without change-point) :

```r
# Function to generate jump series
generate_jump_series <- function(jump_indices, jump_amp, length_series) {
  jump_series <- rep(0, length_series)
  jump_indices <- c(1, jump_indices, length_series + 1)

  changes <- rep(0, length_series)
  changes[jump_indices[-length(jump_indices)]] <- jump_amp

  jump_series <- cumsum(changes)

  return(jump_series)
}
# No cluster (group of close changepoint within 80 days
jump_ind1 <- c(200, 600)
jump_amp1 <- c(0, 1, 1)
# One cluster formed by the second and third changepoints, which need to be keep
jump_ind2 <- c(200, 600, 630)
jump_amp2 <- c(0, 1, -2, 1)
# One cluster formed by the second and third changepoints, which need to be remove
jump_amp3 <- c(0, 1, -2, 2)
```

Premilinary setting

```r
# Generate jump series
jump_series1 <- generate_jump_series(jump_ind1, jump_amp1, length_series)
jump_series2 <- generate_jump_series(jump_ind2, jump_amp2, length_series)
jump_series3 <- generate_jump_series(jump_ind2, jump_amp3, length_series)
plot(jump_series1, type = "l", ylim = c(-2,3))
lines(jump_series2, col = "red")
lines(jump_series3, col = "blue")
```



Test the Segmentation function

When series is homogeneous

```
library(PMLseg)
# Ex1
seg1a = Segmentation(OneSeries = df1, FunctPart = FALSE)
str(seg1a)
#> List of 5
#>  $ Tmu     :'data.frame':    1 obs. of  5 variables:
#>   ..$ begin: int 1
#>   ..$ end  : int 1000
#>   ..$ mean : num -0.00423
#>   ..$ se   : num 29.7
#>   ..$ np   : num 1000
#>  $ FitF    : logi FALSE
#>  $ CoeffF  : logi FALSE
#>  $ MonthVar: num [1:12] 1.089 0.887 1.334 1.092 1.21 ...
#>  $ SSR     : num 933
seg1a$Tmu
#>   begin  end         mean       se   np
#> 1     1 1000 -0.004229094 29.73496 1000
```

No change-point is detected show in `Tmu` dataframe, which listed all segments of the series. The mean is close to 0. Additionally, the result is a list includes not only `Tmu` dataframe but also the fitted functional part `FitF` (which is not fitted by setting FunctPart = FALSE), coefficient of functional element `CoeefF`, monthly variance `MonthVar` and the Sum Square of Residual `SSR`.

Test the Segmentation function

When series is homogeneous for example 2 and 3

```
# Ex2
seg2a = Segmentation(OneSeries = df2, FunctPart = TRUE)
seg2a$Tmu
#>   begin  end         mean       se   np
#> 1     1 1000 -0.006732759 29.62058 1000
# Ex3
seg3a = Segmentation(OneSeries = df3, FunctPart = TRUE)
seg3a$Tmu
#>   begin  end        mean       se   np
#> 1     1 1000 -0.02525641 17.75405 1000
```

No changepoint is detected neither in these two example.

Test the Segmentation function

When we add the jump series in the series

```
# Ex1
df1b <- df1 %>% mutate(signal = signal + jump_series1)
seg1b = Segmentation(OneSeries = df1b, FunctPart = FALSE)
seg1b$Tmu
#>   begin  end       mean       se  np
#> 1     1  199 0.04231954 13.32685 199
#> 2   200  598 0.99732352 18.65043 399
#> 3   599 1000 1.96897996 18.96817 402

# Ex2
df2b <- df2 %>% mutate(signal = signal + jump_series1)
seg2b = Segmentation(OneSeries = df2b, FunctPart = TRUE)
seg2b$Tmu
#>   begin  end       mean       se  np
#> 1     1  200 0.02595179 13.31313 200
#> 2   201  598 0.99381315 18.57034 398
#> 3   599 1000 1.97858505 18.89634 402

# Ex3
df3b <- df3 %>% mutate(signal = signal + jump_series1)
seg3b = Segmentation(OneSeries = df3b, FunctPart = TRUE)
seg3b$Tmu
#>   begin  end      mean        se  np
#> 1     1  233 0.2053781  9.277863 233
#> 2   234  697 1.0960332 11.191458 464
#> 3   698 1000 2.1025174 10.198203 303
```