

Heladas

Alexander Quevedo y Jean Francois Mas

22/11/2019

Debido a la repetitividad de la toma de imágenes, la percepción remota es ampliamente aplicada al seguimiento de fenómenos dinámicos y a la detección de cambio. En Febrero de 2011, las heladas danaron grandes superficies de cultivos en Sinaloa (Figura 5.1). En este ejercicio, se utilizaran imágenes Landsat tomadas antes y después de la helada e imágenes de alta resolución temporal MODIS para detectar las áreas de cultivo dañadas y seguir la recuperación de la vegetación .

##Objetivo Analizar imágenes Landsat y MODIS para monitorear la fenología de la vegetación durante y después un evento de helada extrema.

```
if (!require(raster)) install.packages('raster',dependencies=T)
library(raster)
if (!require(rasterVis)) install.packages('rasterVis',dependencies=T)
library(rasterVis)
if (!require(mapview)) install.packages('mapview',dependencies=T)
library(mapview)
if (!require(mapedit)) install.packages('mapedit',dependencies=T)
library(mapedit)
if (!require(RStoolbox)) install.packages('RStoolbox',dependencies=T)
library(RStoolbox)
if (!require(zoo)) install.packages('zoo',dependencies=T)
library(zoo)
if (!require(remote)) install.packages('remote',dependencies=T)
library(remote)
if (!require(lubridate)) install.packages('lubridate',dependencies=T)
library(lubridate)
if (!require(tidyverse)) install.packages('tidyverse',dependencies=T)
library(tidyverse)
if (!require(snow)) install.packages('snow',dependencies=T)
library(snow)
if (!require(bfast)) install.packages('bfast',dependencies=T)
library(bfast)
if (!require(sf)) install.packages('sf',dependencies=T)
library(sf)
if (!require(fasterize)) install.packages('fasterize',dependencies=T)
library(fasterize)
```

##Análisis de las imágenes Landsat

```
Landsat1<-stack("Data/Landsat_sinaloa/L20110124.tif")
Landsat2<-stack("Data/Landsat_sinaloa/L20110225.tif")
plotRGB(Landsat1,r=4,g=2,b=3,maxpixels=5500000,stretch="hist",main="Fecha 1")
```

La heladas provocan el desastre agrícola en el granero mexicano

Las temperaturas bajo cero del noreste dañaron las cosechas de Sonora y Sinaloa, uno de los principales productores de grano

Por Nadia Sanders

Martes, 22 de febrero de 2011 a las 09:32

La heladas, el desastre agrícola

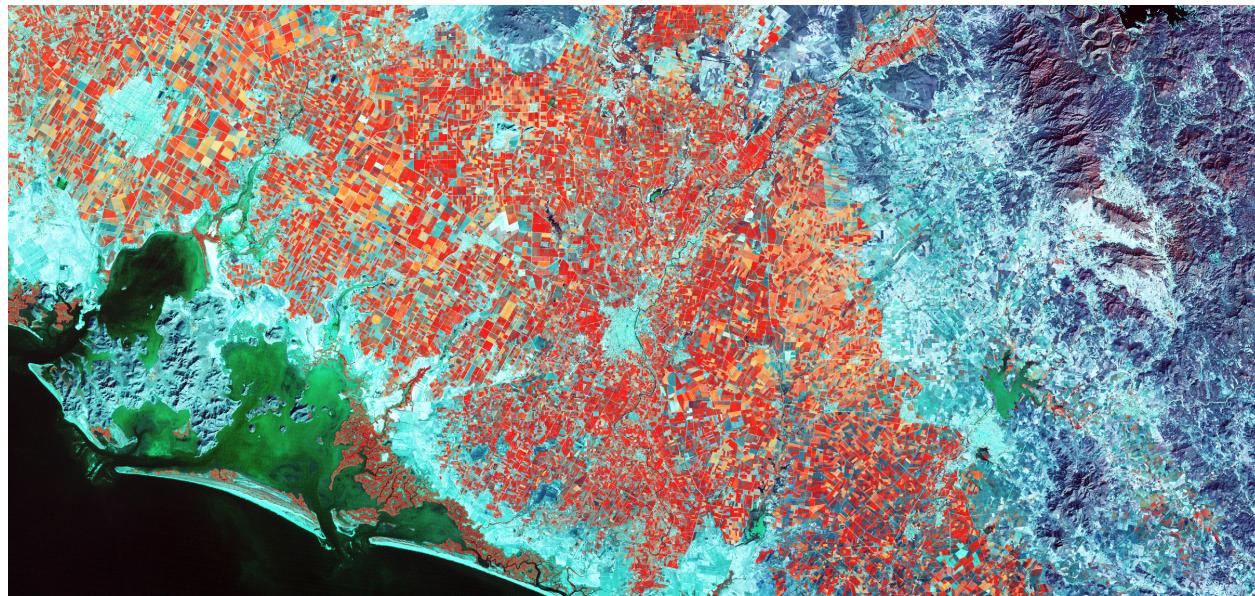


Las heladas que se registraron en el noreste de México afectaron a los principales productores de alimentos: **Sinaloa y Sonora**, considerados estratégicos como "graneros" de México.

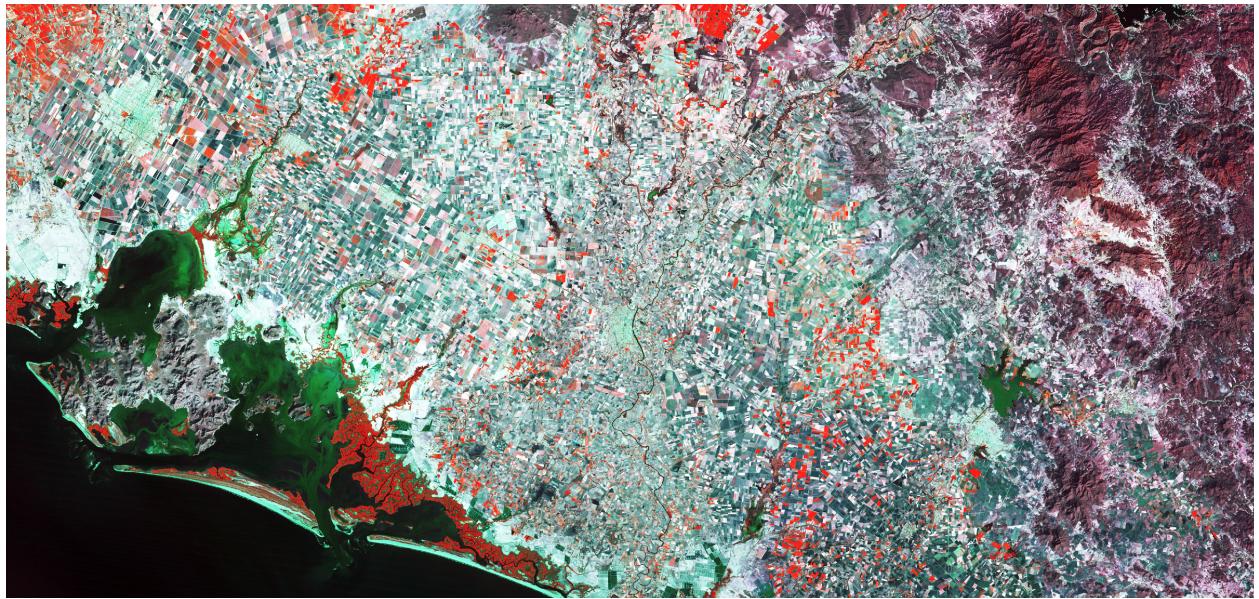
El estado de Sinaloa es el principal proveedor de maíz blanco para la producción de **tortilla**, alimento básico en la dieta mexicana. La entidad cuenta con una producción anual de entre cuatro y cinco millones de toneladas anuales y los efectos de la helada del 3, 4 y 5 de febrero podría comprobarse en el mes de agosto, fecha prevista para la distribución del maíz en el resto del país.

El noreste no había visto una helada así desde 1956. Temperaturas de entre **menos dos y menos seis grados Celsius** fueron suficientes para congelar las plantas de maíz. Las heladas arrasaron 720,000 hectáreas de éste y otros granos, además de hortalizas.

Figure 1: Nota periodística sobre las heladas en el norte de México



```
plotRGB(Landsat2,r=4,g=2,b=3,maxpixels=5500000,stretch="hist",main="Fecha 2")
```



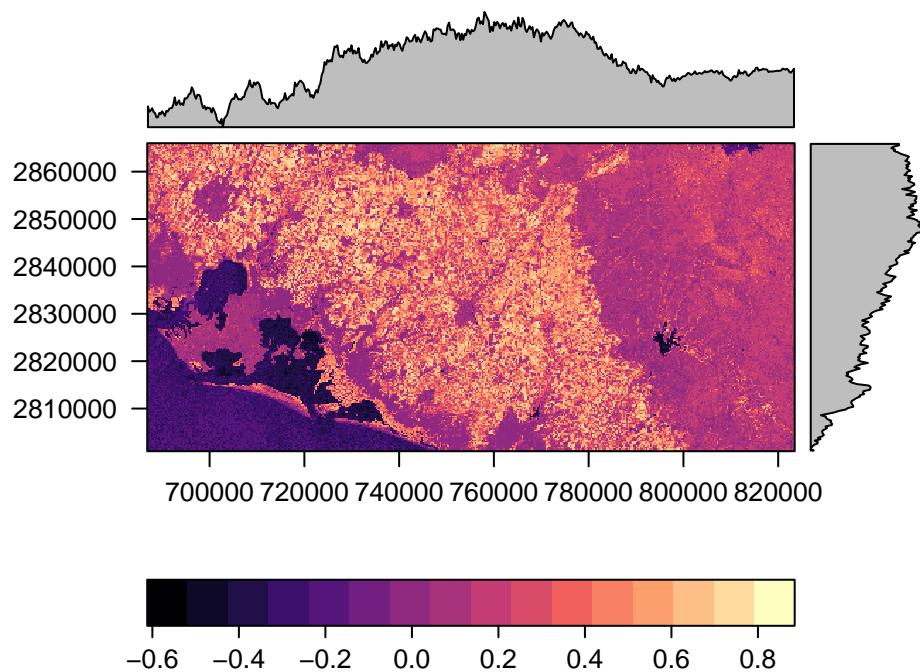
###Calculo de NDVI

Para el cálculo de NDVI emplearemos la función `spectralIndices()` del paquete `RStoolbox`.

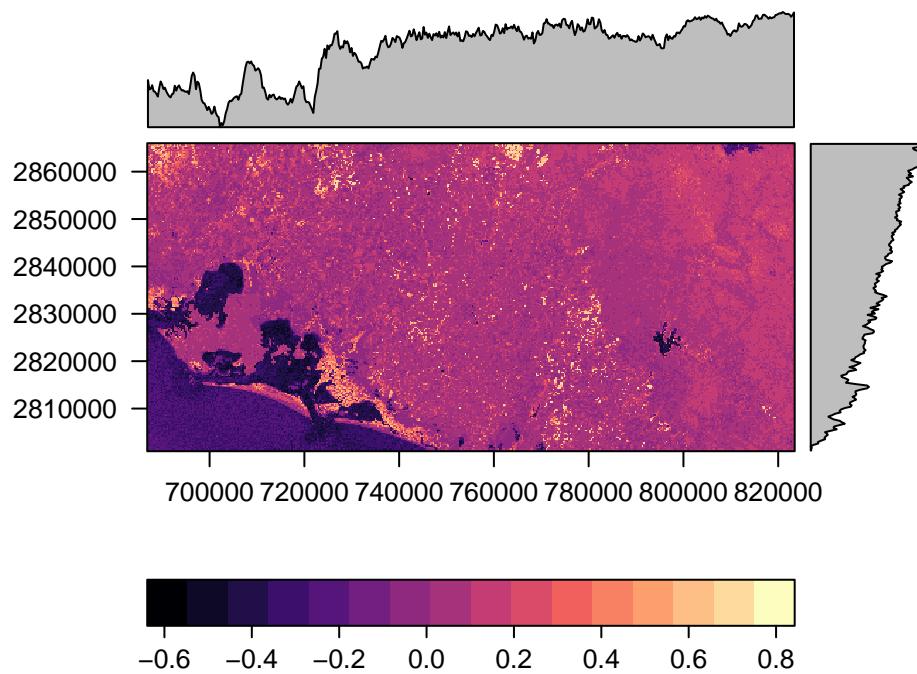
```
ndvi1 <- spectralIndices(Landsat1, red = "L20110124.3", nir = "L20110124.4", indices = "NDVI")
ndvi2 <- spectralIndices(Landsat2, red = "L20110225.3", nir = "L20110225.4", indices = "NDVI")
```

El método `levelplot()` muestra dos gráficos marginales, estos son los resúmenes de las filas y columnas del RasterLayer, calculados con la función definida por el componente FUN

```
levelplot(ndvi1, margin = list(FUN = 'mean'))
```

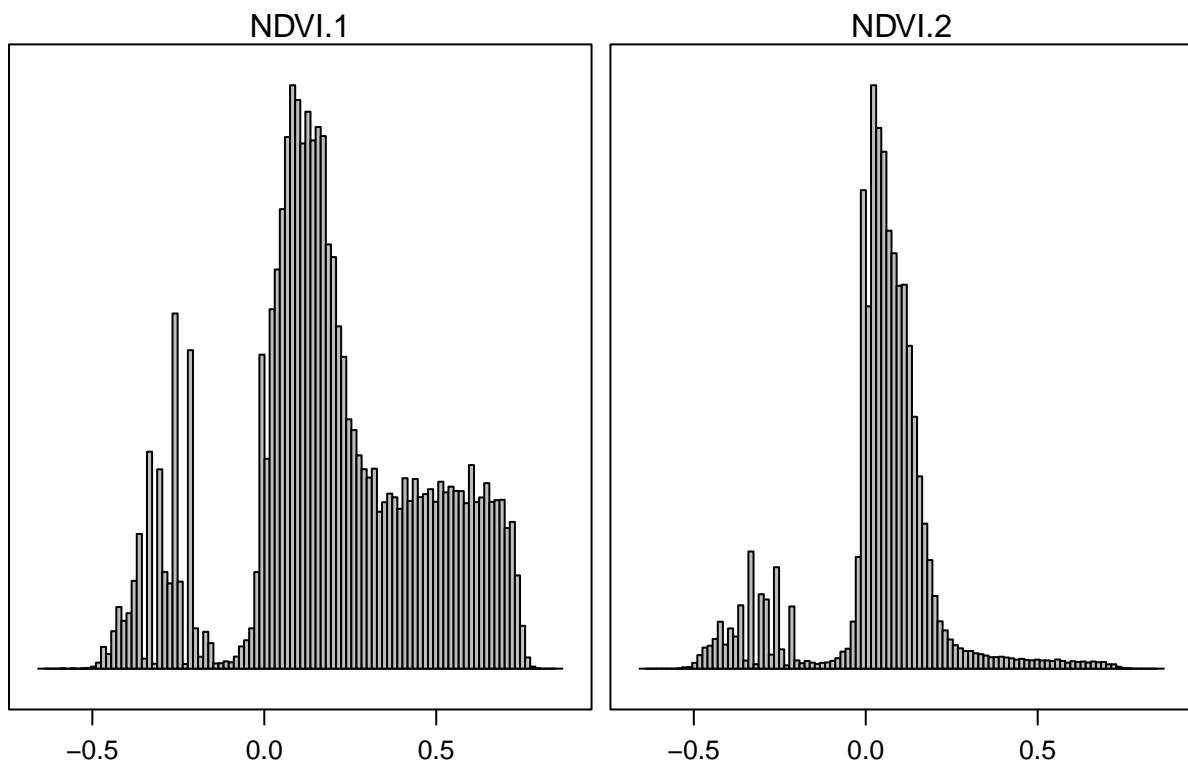


```
levelplot(ndvi2,margin = list(FUN = 'mean'))
```



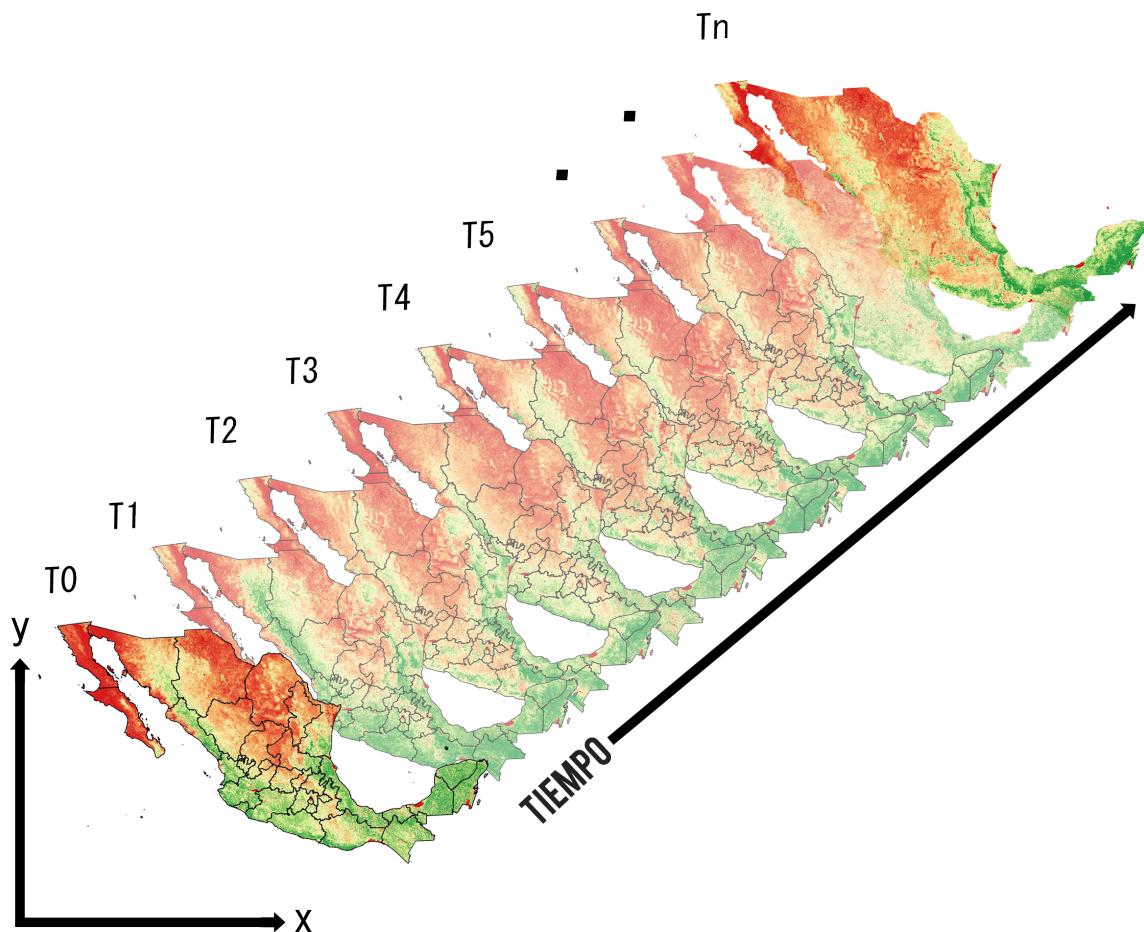
Es posible comparar los dos histogramas.

```
histogram(stack(ndvi1,ndvi2))
```



Series de tiempo

Una serie de tiempo es una secuencia de observación de datos medidos en un intervalo de tiempo.



```
list.grid<-list.files("Data/NDVI_500/",pattern = "*.tif$", recursive = T,full.names = T)
head(list.grid)
```

```

## [1] "Data/NDVI_500//MOD13A1_NDVI_2000_049.tif"
## [2] "Data/NDVI_500//MOD13A1_NDVI_2000_065.tif"
## [3] "Data/NDVI_500//MOD13A1_NDVI_2000_081.tif"
## [4] "Data/NDVI_500//MOD13A1_NDVI_2000_097.tif"
## [5] "Data/NDVI_500//MOD13A1_NDVI_2000_113.tif"
## [6] "Data/NDVI_500//MOD13A1_NDVI_2000_129.tif"

NDVI_16d <-stack(list.grid)
NDVI_16d

## class      : RasterStack
## dimensions : 98, 220, 21560, 274  (nrow, ncol, ncell, nlayers)
## resolution : 0.006220954, 0.006220592  (x, y)
## extent     : -109.1427, -107.7741, 25.29268, 25.9023  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## names      : MOD13A1_NDVI_2000_049, MOD13A1_NDVI_2000_065, MOD13A1_NDVI_2000_081, MOD13A1_NDVI_2000_097, MOD13A1_NDVI_2000_113, MOD13A1_NDVI_2000_129
## min values  : -1923, -1972, -1914,
## max values  : 8805, 8790, 8667,
substring(list.grid[[1]], 1, 10)

## [1] "Data/NDVI_"

year<-substring(list.grid, 29, 32)
day<-as.numeric(substring(list.grid, 34, 36))

date<-as.Date(day,origin=paste(year,"-01-01",sep = ""))

```

Asignamos una fecha para cada imagen usando `setZ()`.

```

NDVI_16d<-setZ(NDVI_16d,date,name="time")

```

Ahora seleccionaremos las imágenes posteriores al 31 de diciembre de 2007.

```

NDVI_16d.3a <- NDVI_16d[[which(getZ(NDVI_16d) >= as.Date("2008-01-01"))]]

```

Un proceso común al trabajar con series de tiempo es la reducción de la serie de tiempo. En este caso calcularemos el valor promedio por mes.

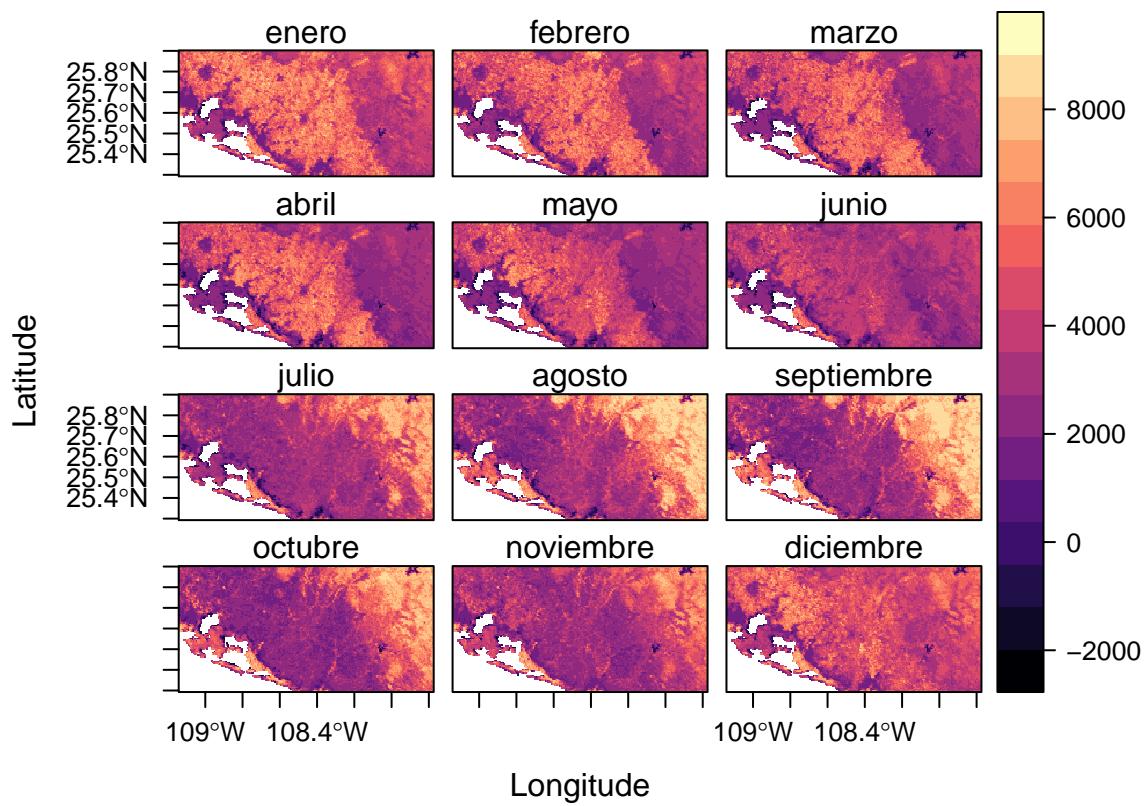
```

promedioM <- zApply(NDVI_16d.3a, by=month,fun=mean)

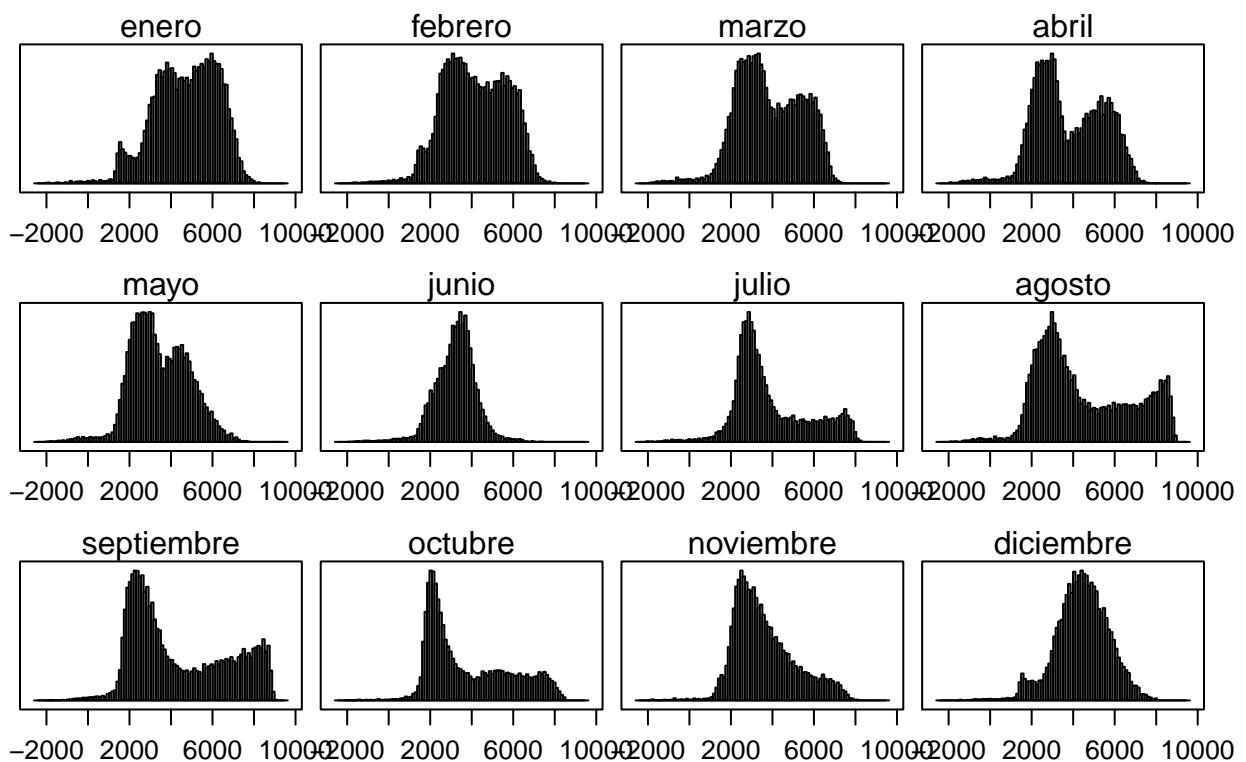
names(promedioM)<-c("enero","febrero","marzo","abril",
                      "mayo","junio","julio","agosto",
                      "septiembre","octubre","noviembre",
                      "diciembre")

levelplot(promedioM)

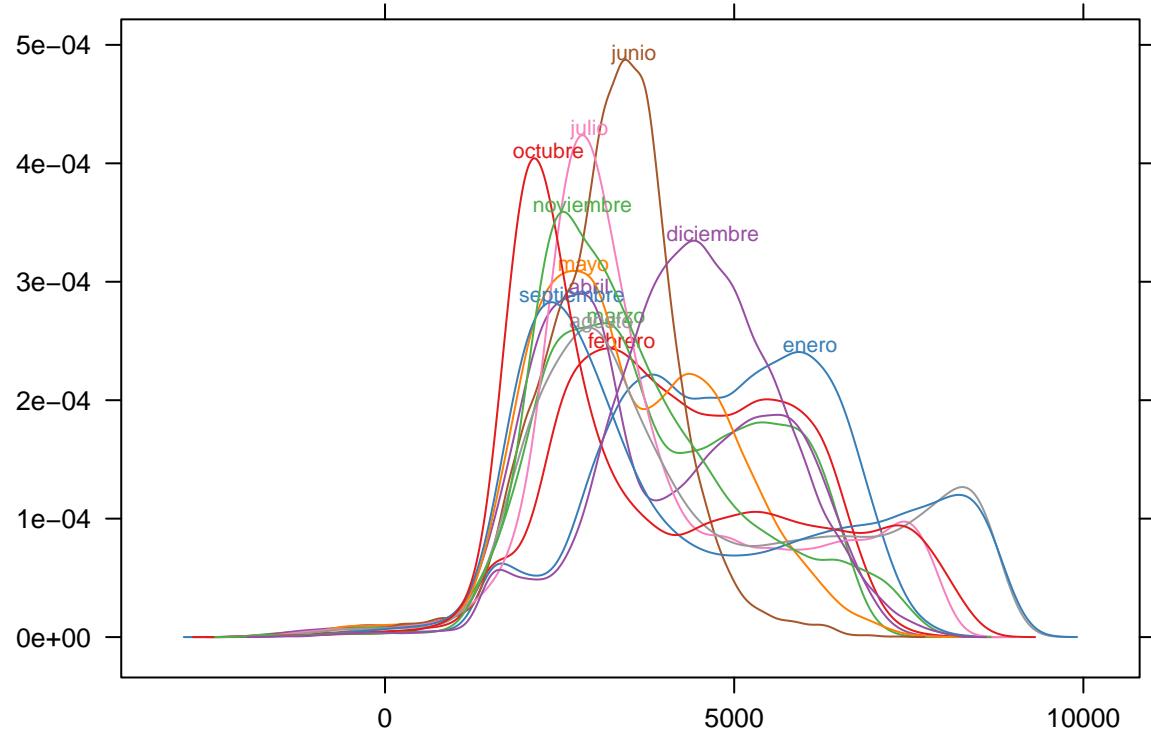
```



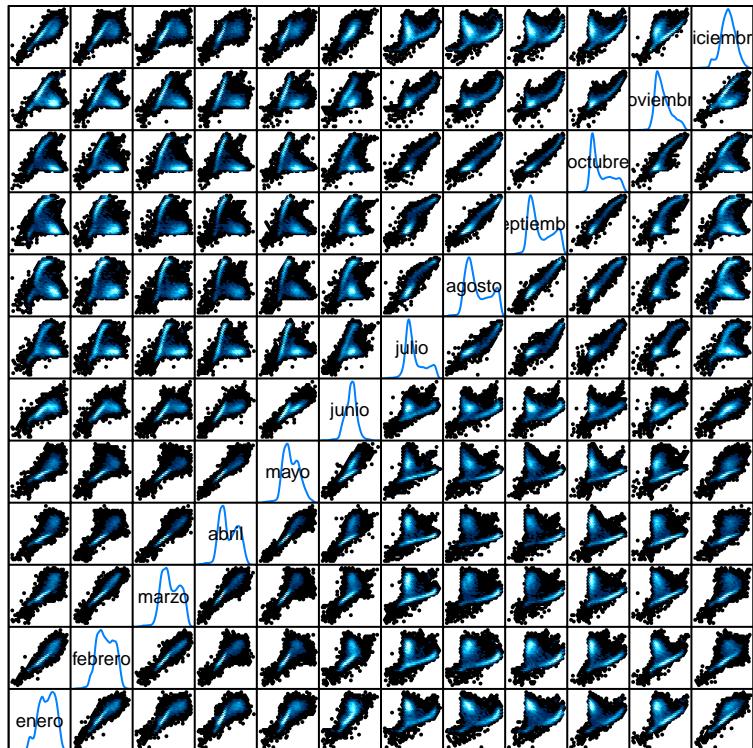
```
histogram(promedioM)
```



```
densityplot(promedioM)
```



```
splom(promedioM,maxpixels=10000)
```



Scatter Plot Matrix

Ahora crearemos un perfil de NDVI por tipos de cubierta.

```

campos <- st_read("Data/Sitios_perfil_WGS84.shp")

## Reading layer `Sitios_perfil_WGS84' from data source `/home/alequech/Data/Sitios_perfil_WGS84.shp' u
## Simple feature collection with 3 features and 2 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:            xmin: -109.0307 ymin: 25.6475 xmax: -107.9607 ymax: 25.82631
## epsg (SRID):   4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs

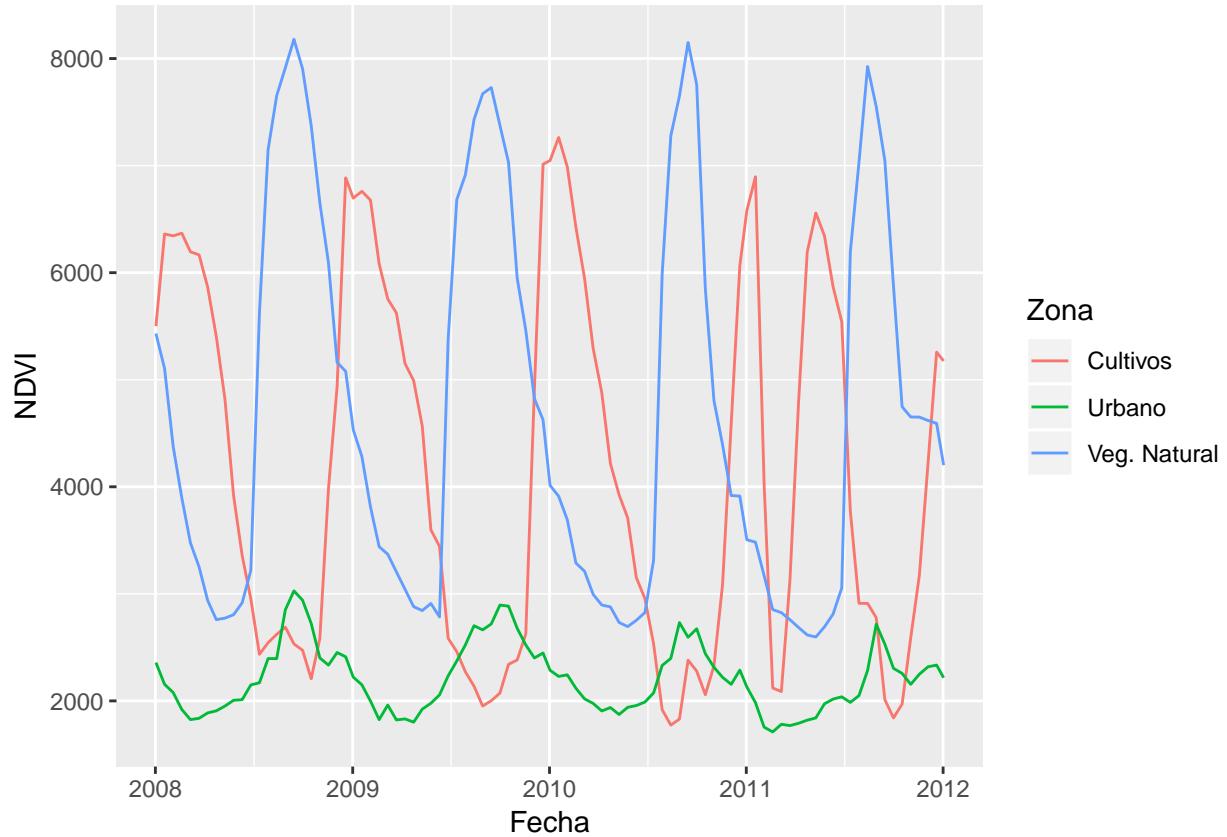
campos_r <- fasterize(campos,NDVI_16d.3a[[1]], field = "ID", background=NA)
zonal_stat <- zonal(NDVI_16d.3a, campos_r, 'mean')

fechas<-getZ(NDVI_16d.3a)
zona <- factor(zonal_stat[, 1])
v_ndvi <- zonal_stat[, -1]

p_zona <- data.frame(
  Fecha = rep(fechas, each = nrow(v_ndvi)),
  Zona = rep(zona, length(fechas)),
  NDVI = c(v_ndvi))

ggplot(p_zona,aes(x = Fecha, y = NDVI, col = Zona), labels=c("a","b","c")) +
  geom_line()+
  scale_color_discrete(labels = c("Cultivos","Urbano","Veg. Natural"))

```



Calculo de anomalías

Crearemos dos funciones para realizar un cálculo básico de anomalías usando promedios y desviaciones estándar.

```
mes <- function(x)format(x, '%m')
##Calcular anomalías mediante agrupaciones mensuales
anomalia <- function(x){
  ## Promedio mensual
  pm <- ave(x, mes(tiempo), FUN = mean)
  ## sd mensual
  sdm <- ave(x, mes(tiempo), FUN = sd)
  ## anomalia
  (x - pm)/sdm
}

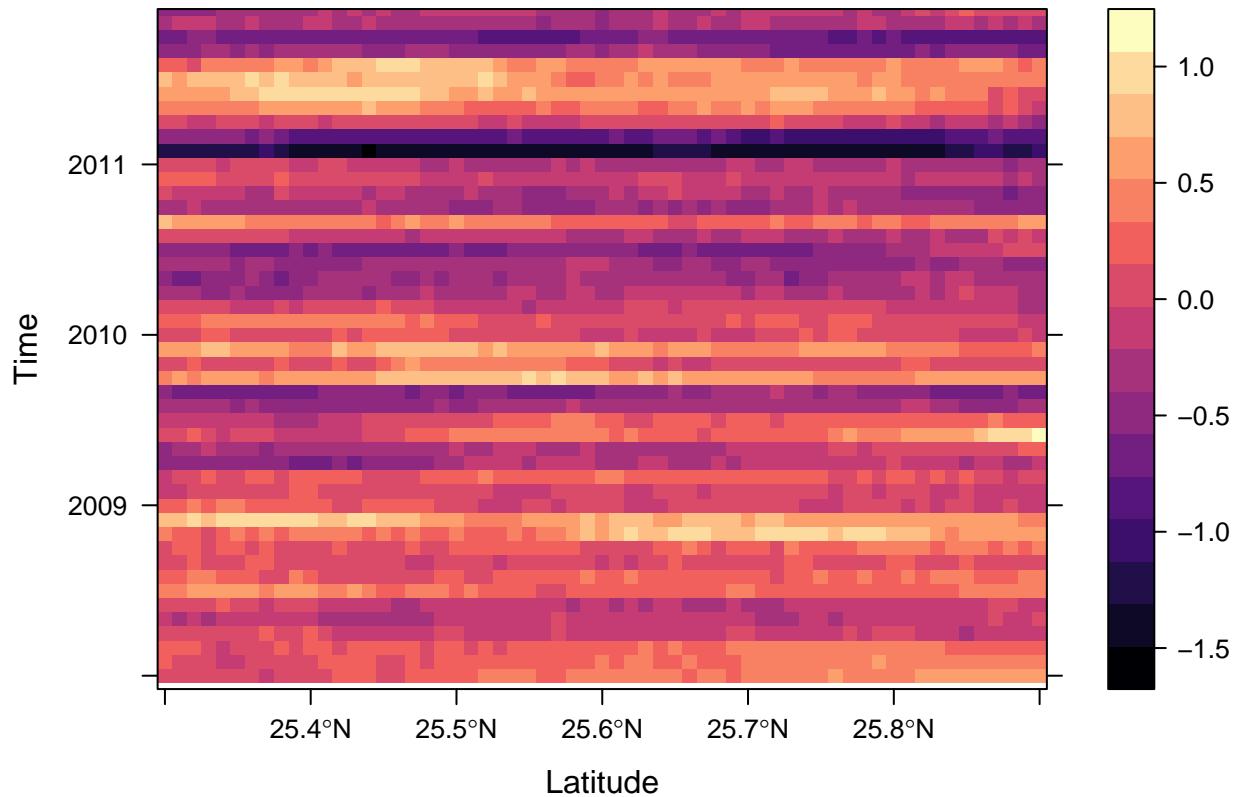
tiempo <- as.yearmon(getZ(NDVI_16d.3a))

anom_NDVI <- calc(NDVI_16d.3a,fun=anomalia)

anom_NDVI <- setZ(anom_NDVI, tiempo)
```

Para visualizar las anomalías usaremos un diagrama de Hovmöller.

```
hovmoller(anom_NDVI)
```

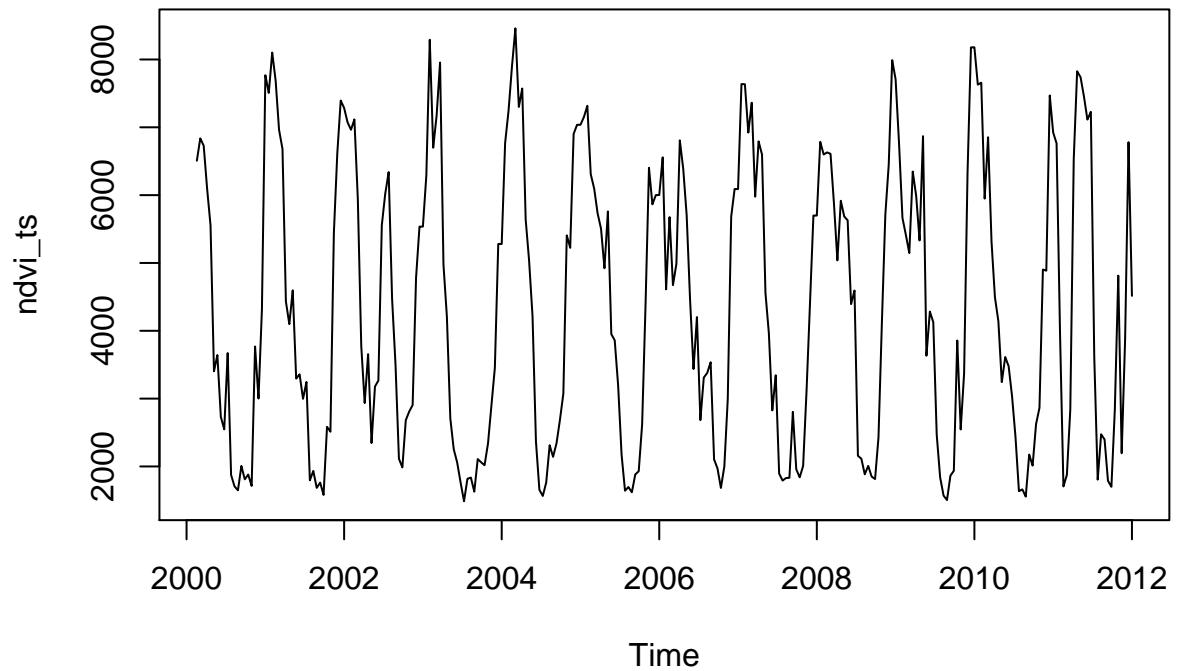


Breaks For Additive Season and Trend - Bfast

```
xy <- as.data.frame(cbind(-108.7863, 25.77303))
punto <- SpatialPoints(xy)
projection(punto)<-projection(NDVI_16d[[1]])

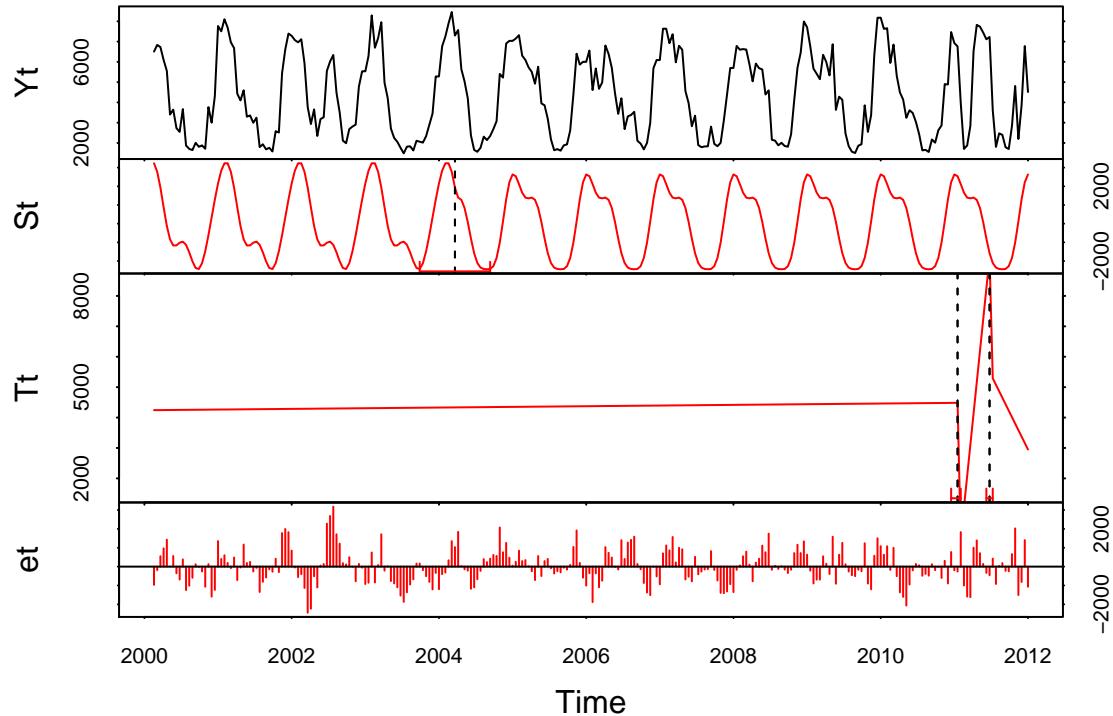
beginCluster(n=4)
  ts_pto<- raster::extract(NDVI_16d,punto)
endCluster()

ndvi_ts<-bfastts(as.vector(ts_pto), date, type = c("16-day"))
plot(ndvi_ts)
```



```
h <- 10/length(ndvi_ts)
bfas00<-bfast(ndvi_ts,h = h,
               season ="harmonic", max.iter=100, breaks=2)
plot(bfas00)
```

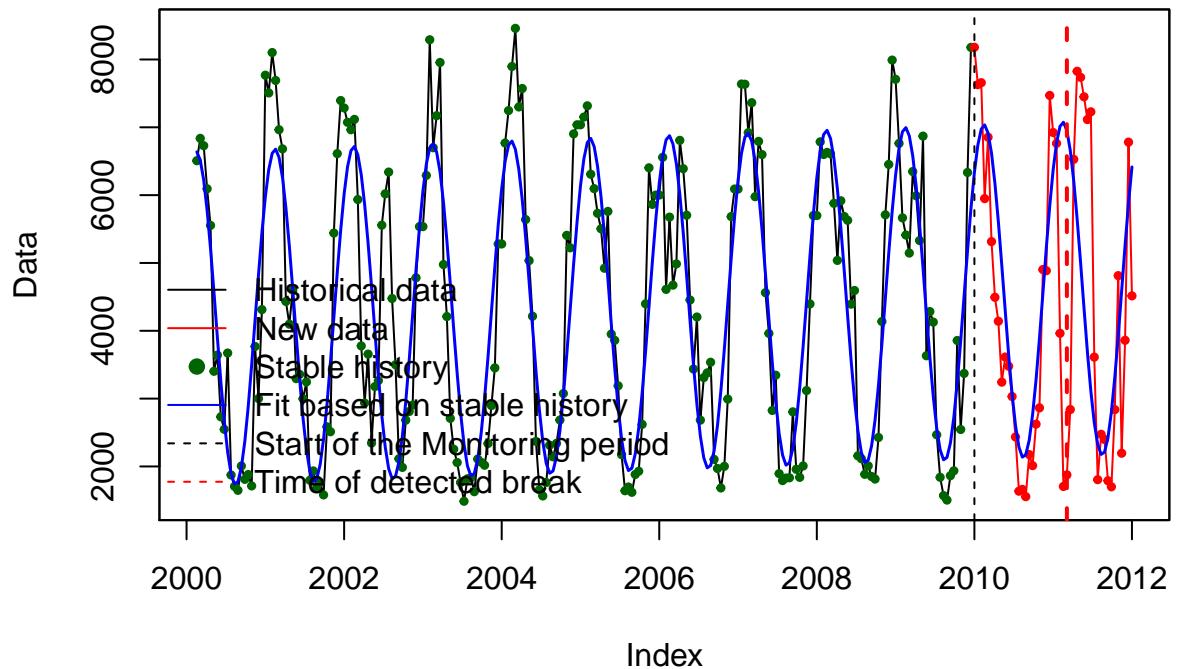
no. iterations to estimate breakpoints: 2



```
bfm = bfastmonitor(ndvi_ts, start=c(2010, 1),
  formula = response ~ trend + harmon,
  order = 1, history=c("ROC"), type="OLS-CUSUM")

plot(bfm)
```

Break detected at: 2011(5)



Exploremos otro punto usando `mapedit()`