BFAST change detection

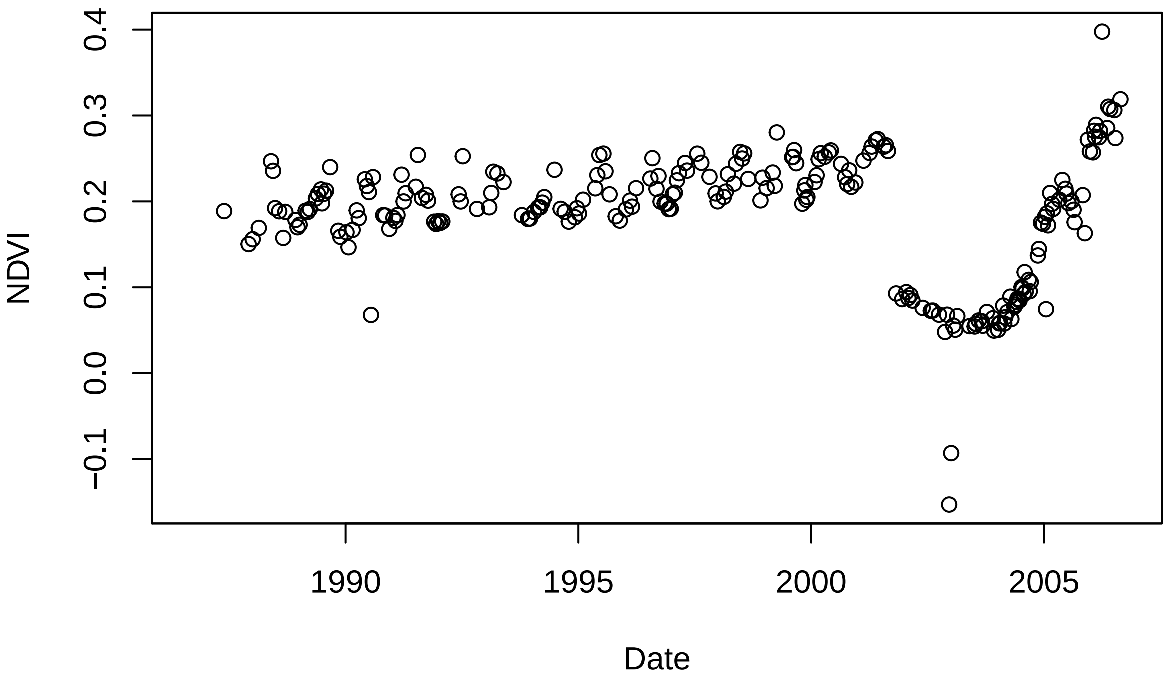
1 Standard change detection using BFAST and others

1.0.1 Load data and aggregate to monthly time series

```
path = "HackTimeSeries/"

# Load all files
ll = lapply( list.files(path,".feather",full.names = T), function(x) read_feather(x) )

# Name them
names(ll) <- file_path_sans_ext( list.files(path,".feather") )
```

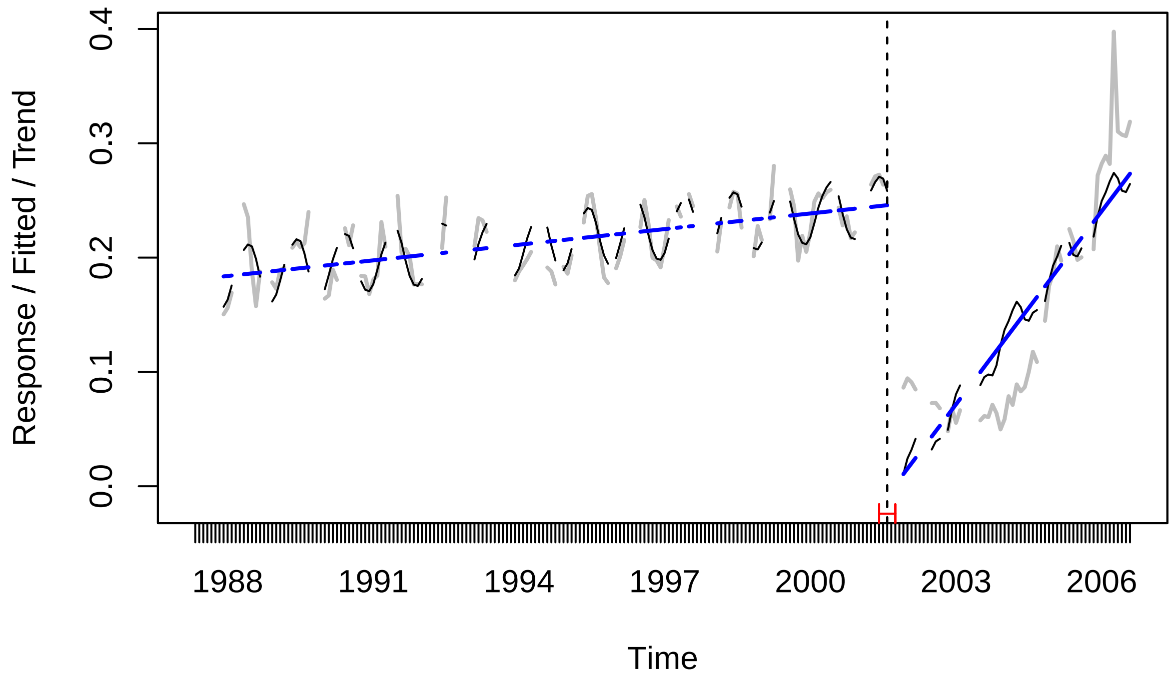
Make a test plot and show a standard bfast fit

```
sub <- ll$SE2_2014a_Craig_1__5
# Define time series
zz <- zooreg(data = sub$EVI2,order.by = sub$date)

plot(zz,xlab = "Date",ylab ="NDVI",type="p")
```




```
# Aggregate to monthly data using a max value composite
xx = suppressWarnings( aggregate(zz,as.yearmon, function(x) max(x,na.rm = T)) )
xx[which(is.infinite(xx))] <- NA

# Run bfast01 model for single breaks
rdist = 12/(length(xx)) # calculate h relative to total sample length
bf.mod <- bfast01(as.ts(xx),formula = response ~ trend+harmon, test = c("OLS-MOSUM","BIC"),aggregate=any
```

```r
plot(bf.mod)
```



Now apply it for all time series and record the date of the change. This is done in order to verify it with the some of the "dated" change estimates

```r
res <- data.frame()
# Do the loopie
for( i in names(ll)){
  sub <- ll[[i]]
  # Define time series
  zz <- suppressWarnings( zooreg(data = sub$EVI2,order.by = sub$date) )

  # Aggregate to monthly data using a max value composite
  xx = suppressWarnings( aggregate(zz,as.yearmon, function(x) max(x,na.rm = T)) )
  xx[which(is.infinite(xx))] <- NA

  # Run bfast01 model for single breaks
  rdist = 12/(length(xx)) # calculate h relative to total sample length
  bf1 = try(bfast01(as.ts(xx),formula = response ~ trend, test = c("OLS-MOSUM","BIC"),aggregate=any, ba
    bf2 = try(bfast01(as.ts(x),formula = response ~ trend + harmon, test = c("OLS-MOSUM","BIC"),aggrega
    l = list(bf1,bf2)
    # Get best model
    bf <- try( l[[which.min(lapply(l,AIC))]], silent = T )

    # Extract largest breakpoint
    if(class(bf)=="try-error")
    {
      # Check if trend only converged and use this instead
      if(class(bf1)!="try-error"){
        bf = bf1 # Reset to trend only
      } else { next() }
```

```
    }
    if(bf$breaks > 0){
    # Time of largest change
    la_year <- bf$data$time[bf$breakpoints]

    res <- rbind(res, data.frame(SSBS = i, year = la_year))
    # clean up
    rm(i,la_year)
  }
}
```

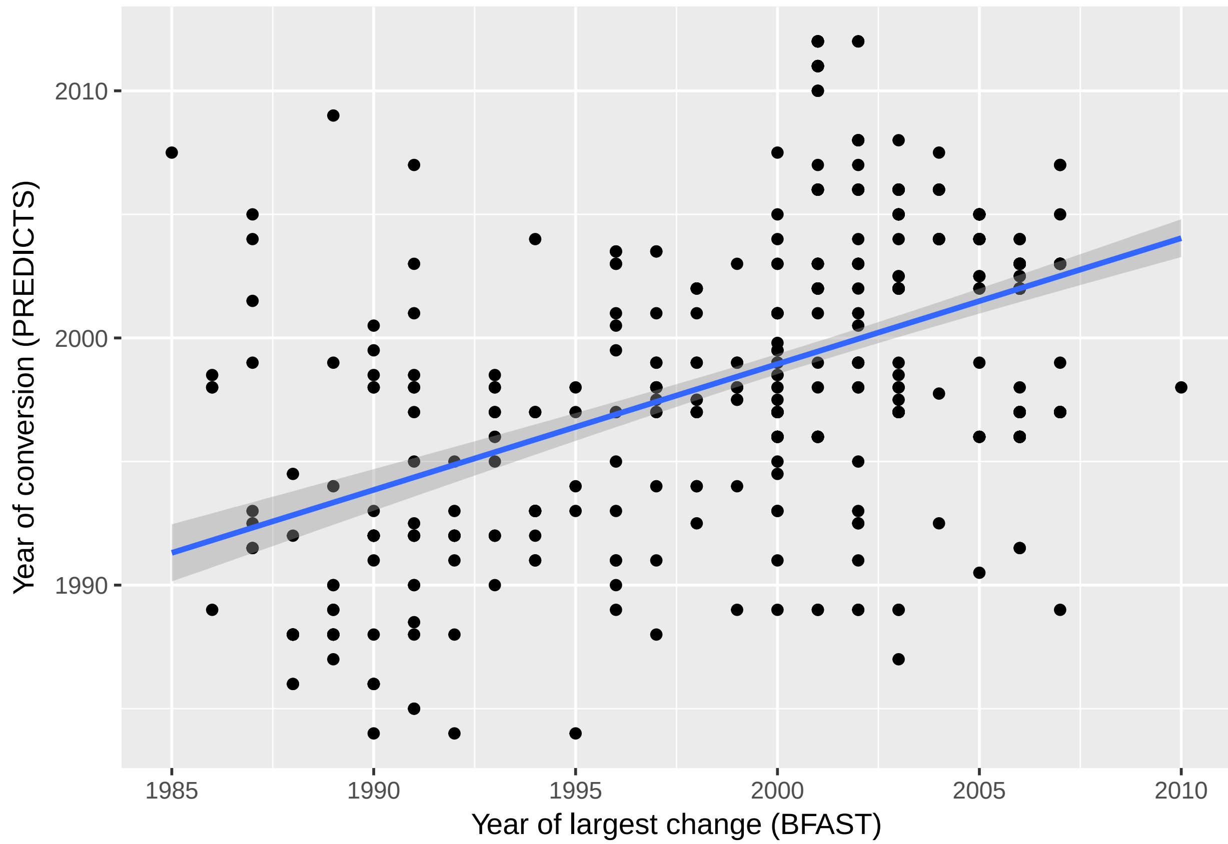
1.0.2 Now verify against the date estimate according to PREDICTS

The data contains a field that captures the approximate date of the largest change event This information might be incorrect! It could also be that the coordinates or extent are too inaccurate to capture the change

```
# Assemble full
df <- do.call("rbind",ll)
# Construct names by replacing space with underscores
df$mergeSSBS <- str_replace_all(df$SSBS,pattern = " ",replacement = "_")

# Subset to columns of interest
sub <- df %>% dplyr::mutate(SampleStart = year( Sample_start_earliest ) ) %>%
  dplyr::select(SSBS,mergeSSBS,SampleStart,Hansenlossyear, YearsOfConversion) %>% distinct_() %>%
  # Merge
  merge.data.frame(.,res,by.x = "mergeSSBS",by.y = "SSBS") %>%
  mutate(year = lubridate::year(as.yearmon(year)))
```

Now plot and assess match first for PREDICTS record

```
# Plot years and assess
qplot(sub$year,sub$YearsOfConversion,xlab = "Year of largest change (BFAST)",ylab = "Year of conversion
  geom_smooth(method="lm")
```




```
cor.test(sub$year,sub$YearsOfConversion)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sub$year and sub$YearsOfConversion
## t = 14.428, df = 613, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4420254 0.5602510
## sample estimates:
##       cor
## 0.5034913
```

```
summary(lm(sub$year~sub$YearsOfConversion))
```

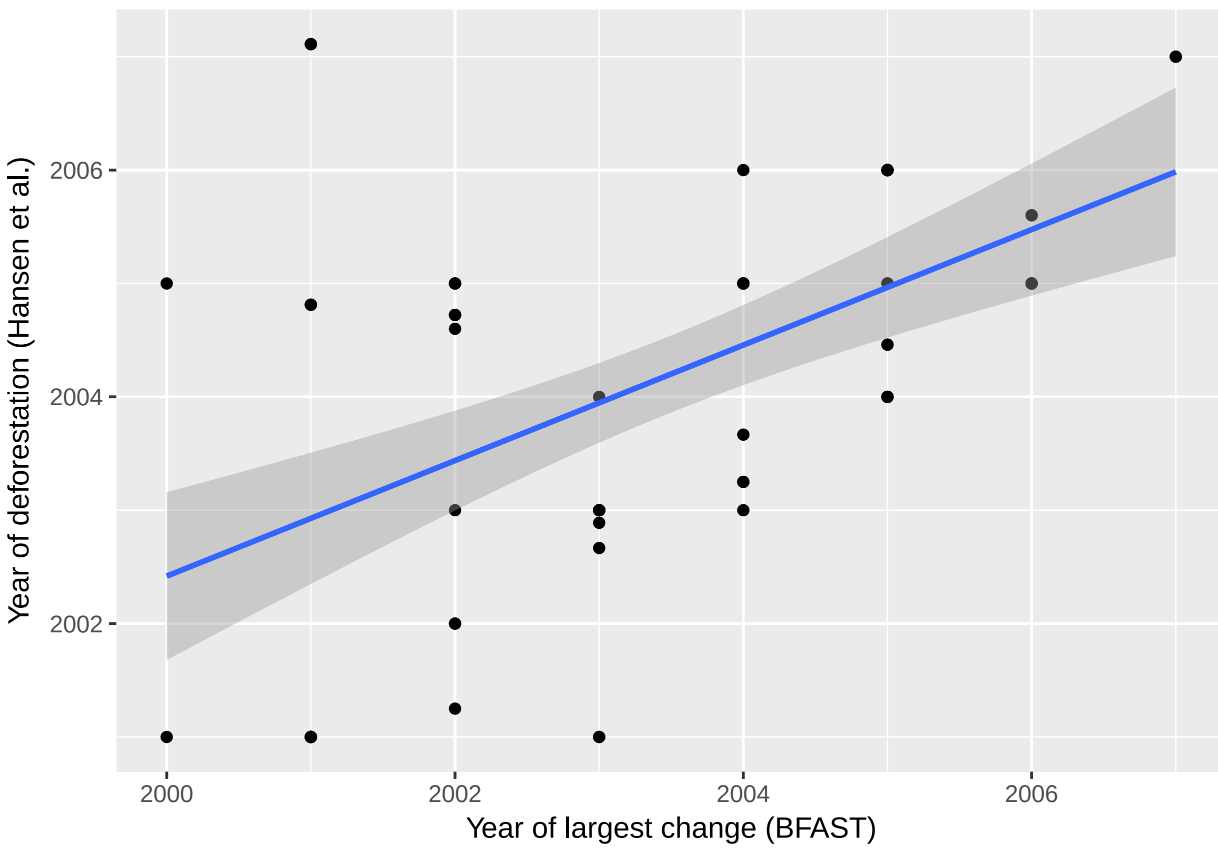
```
##
## Call:
## lm(formula = sub$year ~ sub$YearsOfConversion)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.7244  -3.5072   0.9996   3.5154  11.4837
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1005.5186    68.9704   14.58   <2e-16 ***
## sub$YearsOfConversion    0.4977     0.0345   14.43   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.05 on 613 degrees of freedom
## Multiple R-squared:  0.2535, Adjusted R-squared:  0.2523
## F-statistic: 208.2 on 1 and 613 DF,  p-value: < 2.2e-16
```

Now the same for a remote-sensing derived estimate of "deforestation" Hansen only monitors from 2000 onwards, so we subset it to estimate >= 2000 Also limit only to those sites where sampling started before hansen deforestation

"'r subs <- subset(sub, year >= 2000) %>% dplyr::filter(SampleStart >= Hansenlossyear) # Filter out sites that started after disturbance

qplot(subs*year*, *subs*Hansenlossyear,xlab = "Year of largest change (BFAST)",ylab = "Year of deforestation (Hansen et al.)") + geom_smooth(method="lm") "'



`r  cor.test(subs$year,subs$Hansenlossyear)`

```
##   ##     Pearson's product-moment correlation  ##   ## data:  subs$year and subs$Hansenlossyear
## t = 5.3802, df = 78, p-value = 7.568e-07  ## alternative hypothesis: true correlation
is not equal to 0  ## 95 percent confidence interval:  ##  0.3393204 0.6640600  ## sample
estimates:  ##       cor   ## 0.5202503
```

`r  summary(lm(subs$year~subs$YearsOfConversion))`

```
##   ## Call:  ## lm(formula = subs$year ~ subs$YearsOfConversion)  ##   ## Residuals:
##     Min      1Q  Median      3Q     Max   ## -3.5684 -1.6772  0.4316  1.5404  3.7580
##   ## Coefficients:  ##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2221.61039  129.78533  17.118   <2e-16 ***  ## subs$YearsOfConversion
```

```
-0.10880    0.06474  -1.681   0.0968 .    ## ---  ## Signif. codes:  0 '***' 0.001 '**'
0.01 '*' 0.05 '.' 0.1 ' ' 1  ##   ## Residual standard error: 1.794 on 78 degrees of
freedom  ## Multiple R-squared:  0.03495,   Adjusted R-squared:  0.02257   ## F-statistic:
2.825 on 1 and 78 DF,  p-value: 0.09683
```