Updates to fect

Rivka Lipkovitz

Updates to plotting

This document explains all of the new features that have been added to **fect**.

Quick summary of major changes:

- Created `did_wrapper` function
- Created function `fect_sens_anlys` for Rambachan & Roth style robust confidence sets
- Added plotting option for `fect_sens_anlys` (both average and period)
- Fixed bug during removing carryover period
- Replaced `vis` with an argument called `connected`
- Added deprecation message for `vis`
- Set `gridOff = TRUE` by default (makes plots look more clean)
- Set `show.points = TRUE` by default for connected plot
- Created `grayscale` and `vibrant` themes
- Added arguments to `fect` so that users can pass in their own colors
- Made color transition for placebo/carryover effects happen in between points
- Made lines thinner and vertical line dashed
- Removed outline around count bars
- Made background of status plot white by default
- Small visual QoL changes in all plots

Load Libraries

```
library(dplyr)
library(fixest)
library(did)
library(fect)
library(PanelMatch)
library(DIDmultiplegtDYN)
library(ggplot2)
library(panelView)
library(HonestDiDFEct)
```

Wrapper for New DID Methods

The outputs of `did_wrapper` will be the same as shown in the tutorial.

```
data(fect)
df <- hh2019
head(df)
```

```
##   bfs year nat_rate_ord indirect
```

```
## 1   1 1991     0.000000        0
## 2   1 1992     0.000000        0
## 3   1 1993     0.000000        0
## 4   1 1994     3.448276        0
## 5   1 1995     0.000000        0
## 6   1 1996     0.000000        0
```

```
# Main variables
Y <- "nat_rate_ord"
D <- "indirect"
index <- c("bfs", "year")
```

TWFE

```
res_twfe <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "twfe",
  se     = "default"
)
```

```
## Dropped 283 units when removing always treated units.
```

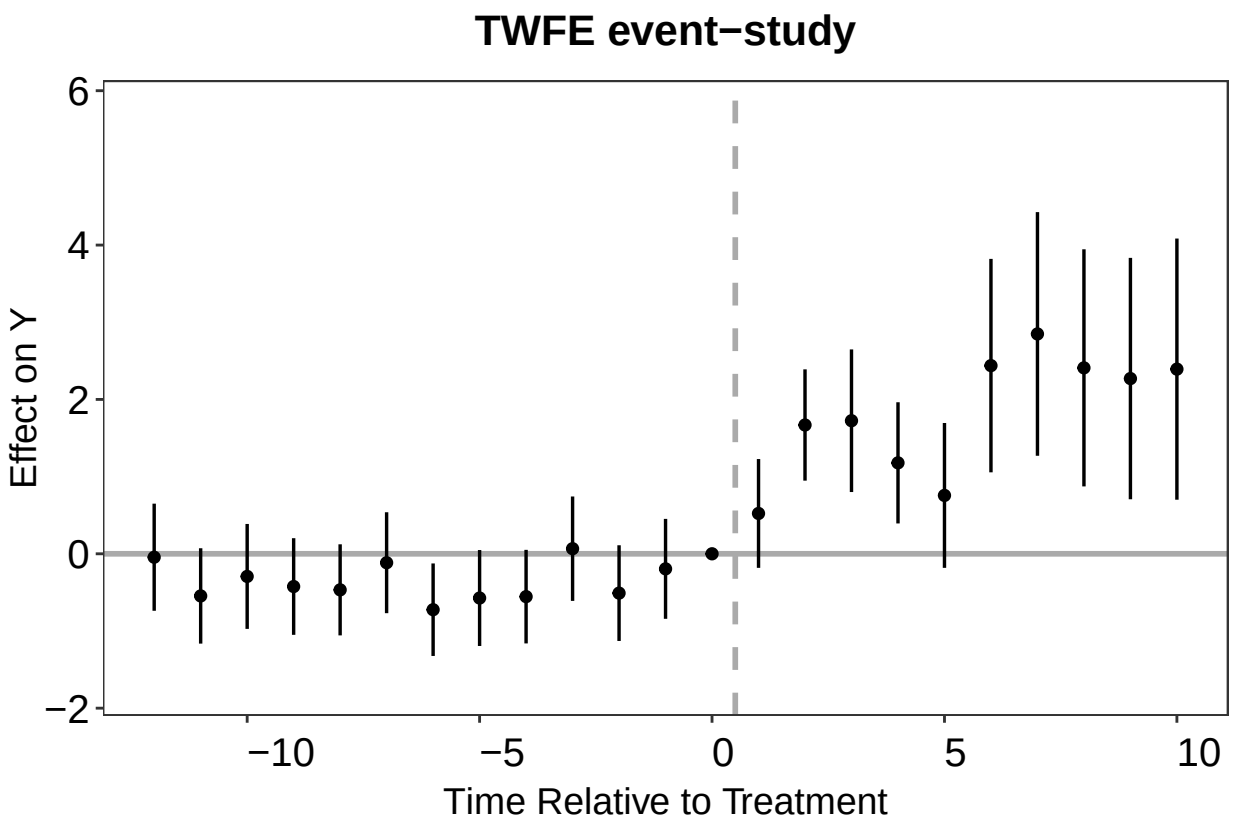
```
cat("\n>>> TWFE results:\n")
```

```
##
## >>> TWFE results:
```

```
cat("ATT:", res_twfe$ATT, "SE:", res_twfe$ATT_se,
    "CI:", res_twfe$CI_lower, "to", res_twfe$CI_upper, "\n")
```

```
## ATT: 1.608579 SE: 0.1952962 CI: 1.225798 to 1.991359
```

```
p_twfe <- esplot(data = res_twfe,
                 main = "TWFE event-study", xlim = c(-12,10))
print(p_twfe)
```



Stacked DID

```
res_st <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "st",
  se     = "default"
)
```

```
## Dropped 283 units when removing always treated units.
```

```
## The variable 'Time_to_Treatment::999999:treat' has been removed because of collinearity (see $collin
```

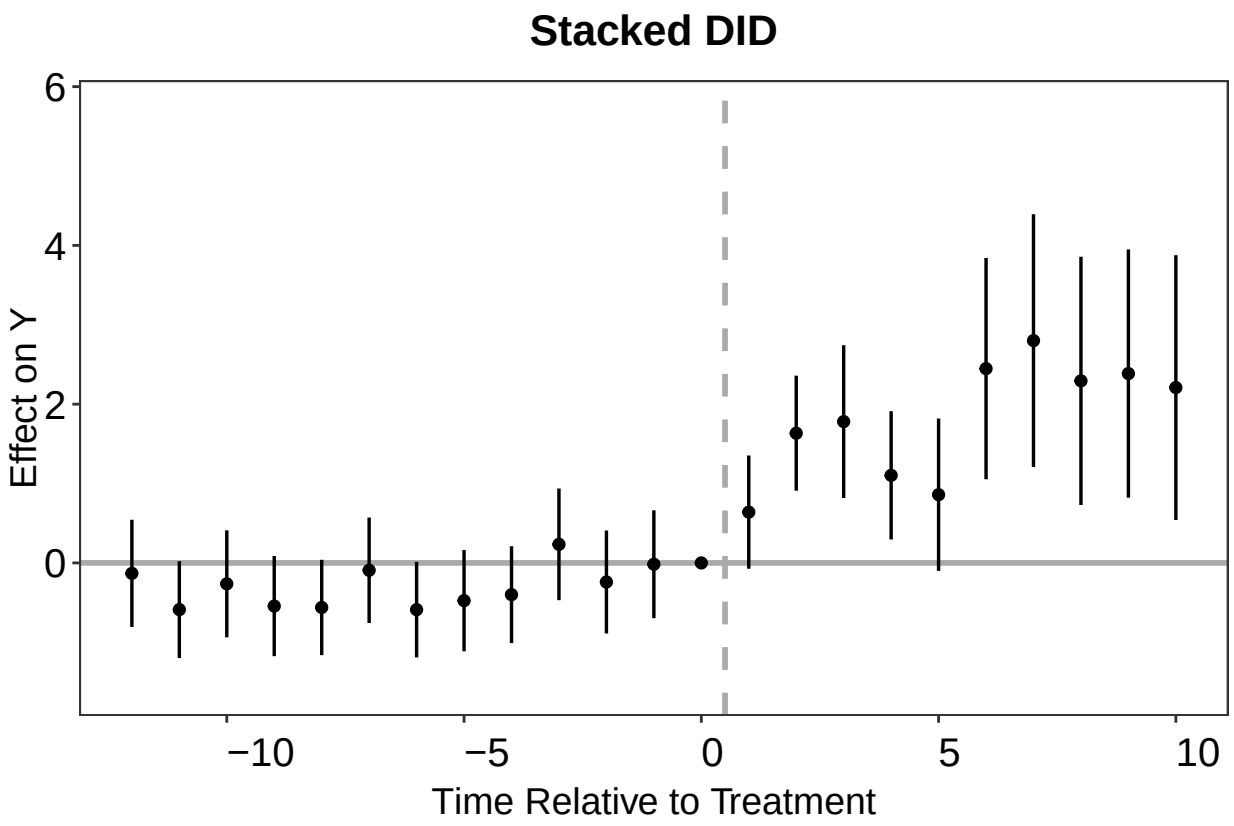
```
cat("\n>>> Stacked DID Results:\n")
```

```
##
## >>> Stacked DID Results:
```

```
cat("ATT:", res_st$ATT, "SE:", res_st$ATT_se, "\n")
```

```
## ATT: 1.668611 SE: 0.1823702
```

```
p_st <- esplot(data = res_st,
               main = "Stacked DID", xlim = c(-12,10))
print(p_st)
```



Interaction Weighted DID

```
res_iw <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "iw",
  se     = "default"
)
```

```
## Dropped 283 units when removing always treated units.
```

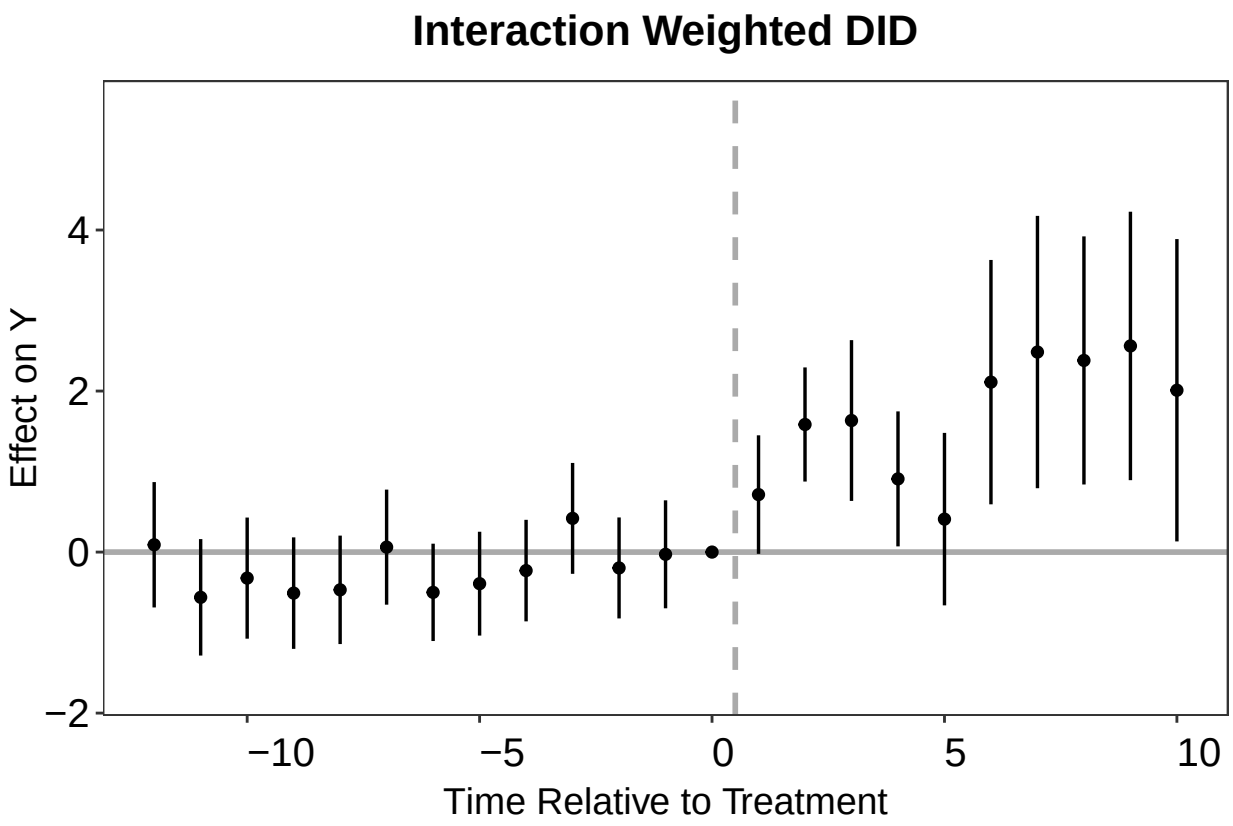
```
cat("\n>>> Interaction Weighted Results:\n")
```

```
##
## >>> Interaction Weighted Results:
```

```
cat("ATT:", res_iw$ATT, "SE:", res_iw$ATT_se, "\n")
```

```
## ATT: 1.330899 SE: 0.2879715
```

```
p_iw <- esplot(data = res_iw,
               main = "Interaction Weighted DID", xlim = c(-12,10))
print(p_iw)
```



Callaway and Sant'Anna DID

```
res_csnever <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "cs_never",
  se     = "default"
)
```

5.1) cs_never

```
## Dropped 283 units when removing always treated units.
```

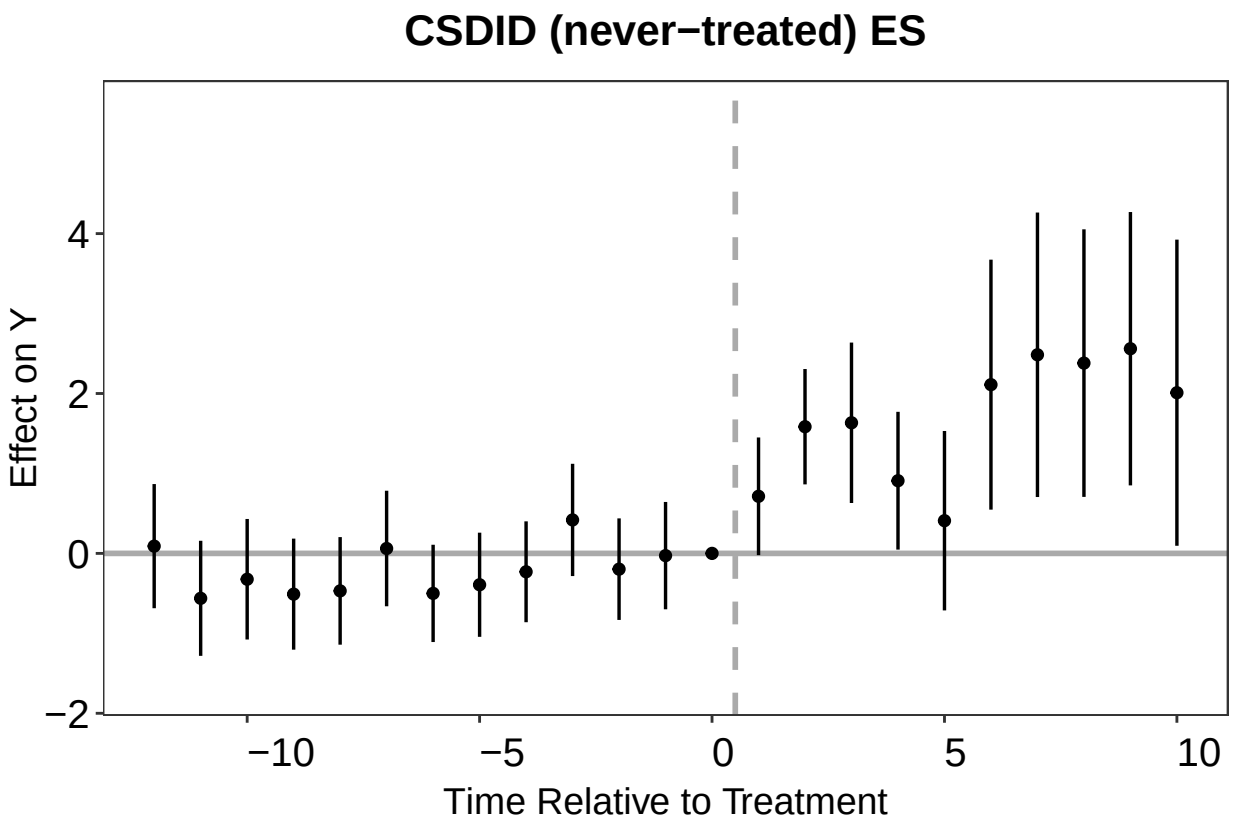
```
cat("\n>>> CSDID (never-treated) Results:\n")
```

```
##
## >>> CSDID (never-treated) Results:
```

```
cat("ATT:", res_csnever$ATT, "SE:", res_csnever$ATT_se,
    "CI:", res_csnever$CI_lower, "to", res_csnever$CI_upper, "\n")
```

```
## ATT: 1.330899 SE: 0.3018856 CI: 0.7392027 to 1.922594
```

```
p_csnever <- esplot(data = res_csnever,
                    main = "CSDID (never-treated) ES", xlim = c(-12,10))
print(p_csnever)
```




```
res_csnotyet <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "cs_notyet",
  se     = "default"
)
```

5.2) cs_notyet

```
## Dropped 283 units when removing always treated units.
```

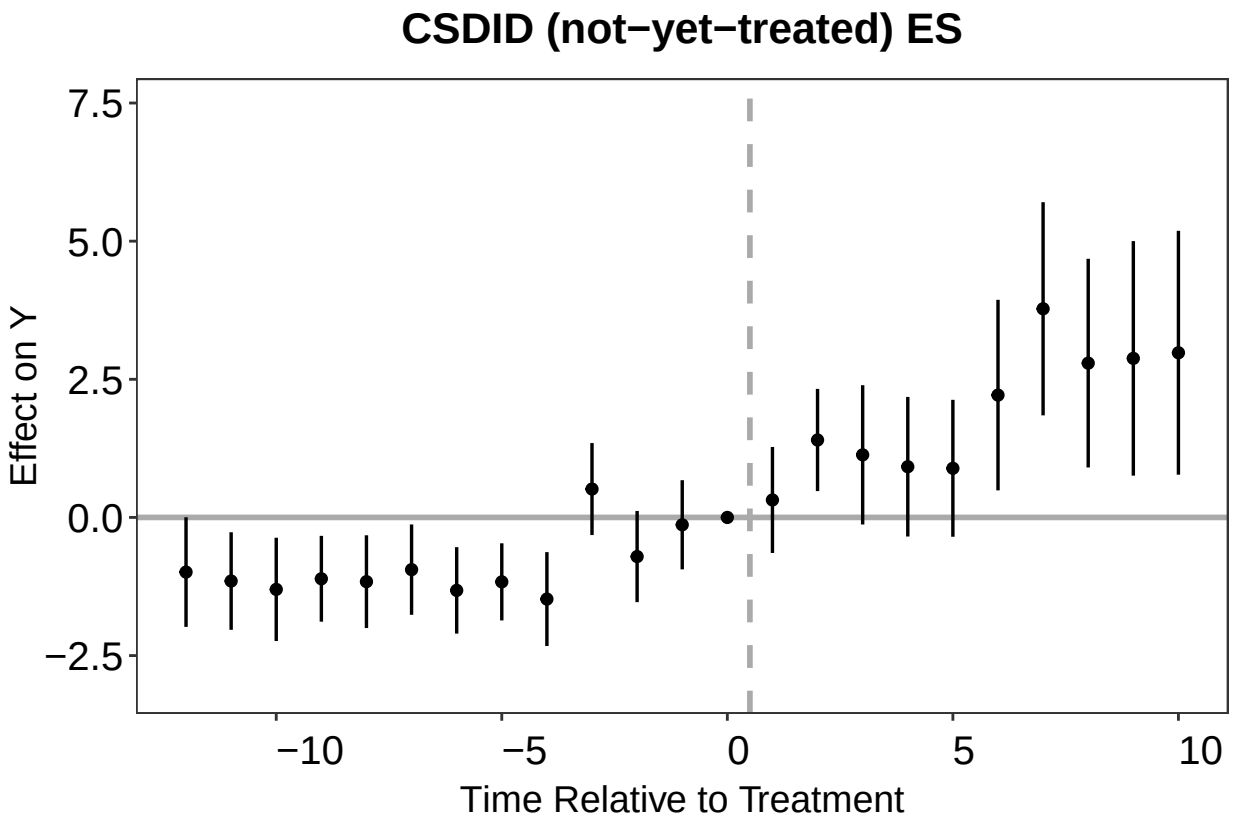
```
cat("\n>>> CSDID (not-yet-treated) Results:\n")
```

```
##
## >>> CSDID (not-yet-treated) Results:
```

```
cat("ATT:", res_csnotyet$ATT, "SE:", res_csnotyet$ATT_se,
    "CI:", res_csnotyet$CI_lower, "to", res_csnotyet$CI_upper, "\n")
```

```
## ATT: 1.29294 SE: 0.3900078 CI: 0.5285242 to 2.057355
```

```
p_csnotyet <- esplot(data = res_csnotyet,
                     main = "CSDID (not-yet-treated) ES", xlim = c(-12,10))
print(p_csnotyet)
```



DIDmultiplegtDYN ("didm") Example

```
res_didm <- did_wrapper(
  data         = df,
  Y            = Y,
  D            = D,
  index        = index,
  method       = "didm",
  didm.effects = 12,
  didm.placebo = 9,
  se           = "default"
)
```

```
## Dropped 283 units when removing always treated units.
```

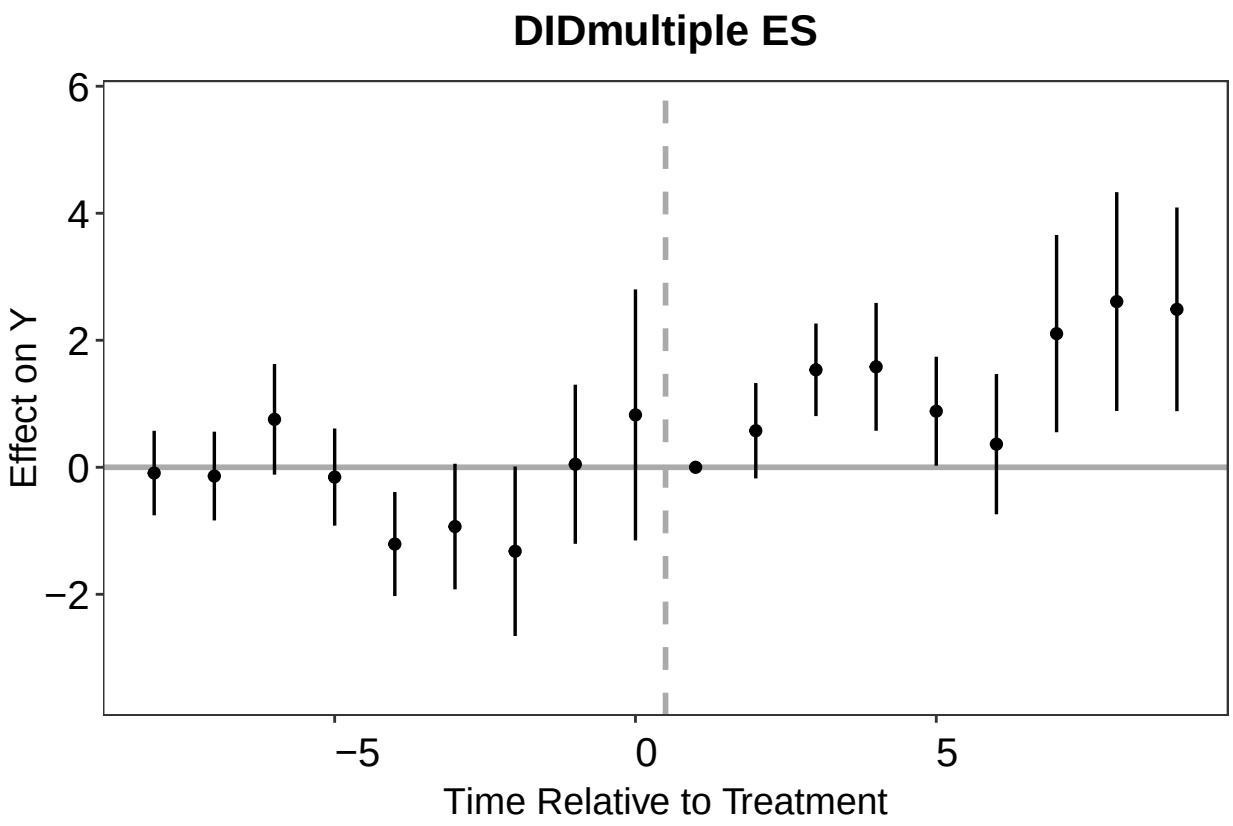
```
cat("\n>>> DIDmultiple Results:\n")
```

```
##
## >>> DIDmultiple Results:
```

```
cat("ATT:", res_didm$ATT, "SE:", res_didm$ATT_se,
    "CI:", res_didm$CI_lower, "to", res_didm$CI_upper, "\n")
```

```
## ATT: 1.30795 SE: 0.30296 CI: 0.71416 to 1.90174
```

```
p_didm <- esplot(data = res_didm,
                 main = "DIDmultiple ES", xlim = c(-12,9))
print(p_didm)
```



TWFE with Cluster Bootstrap Example

```
res_twfe_boot <- did_wrapper(
  data   = df,
  Y      = Y,
  D      = D,
  index  = index,
  method = "twfe",
  se     = "boot",
  nboots = 50
)
```

```
## Dropped 283 units when removing always treated units.
```

```
cat("\n>>> TWFE With Cluster Bootstrap:\n")
```

```
##
## >>> TWFE With Cluster Bootstrap:
```

```
cat("ATT:", res_twfe_boot$ATT, "\n")
```

```
## ATT: 1.608579
```

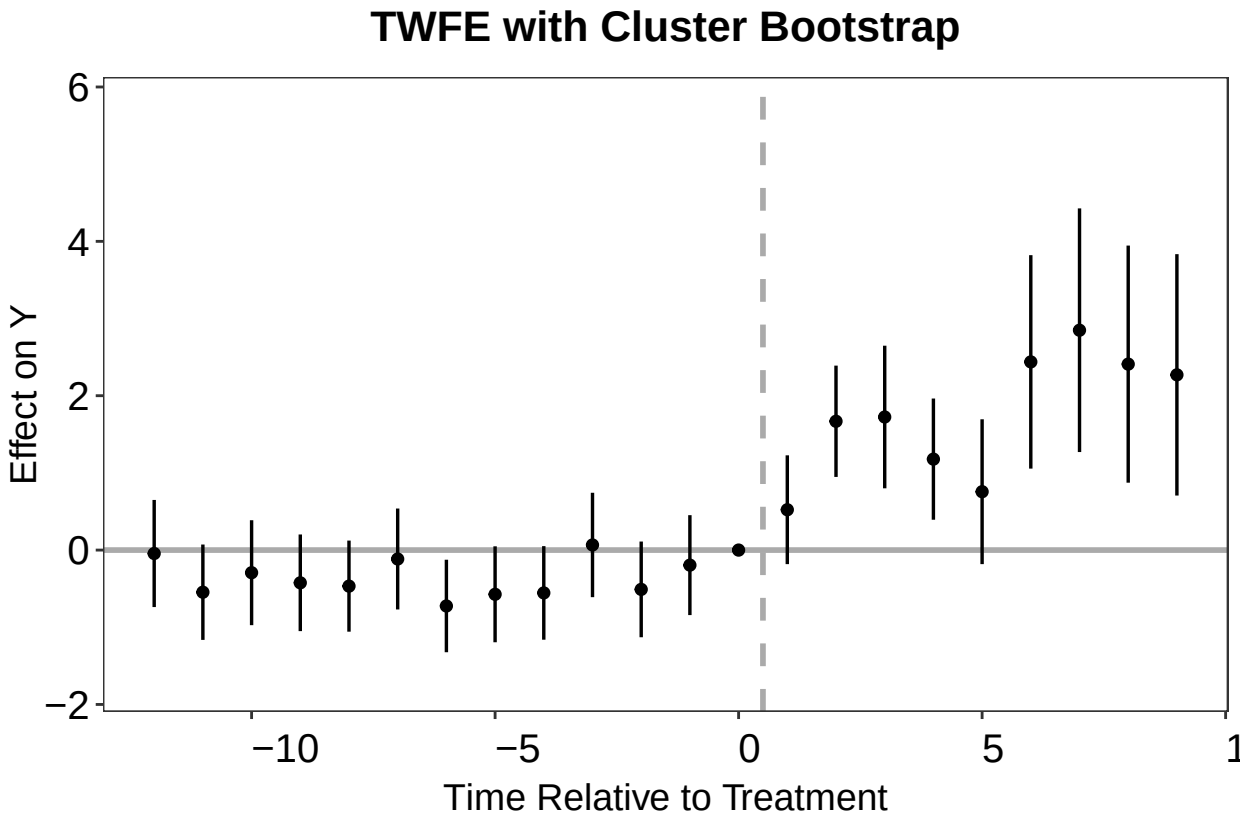
```
cat("SE (boot):", res_twfe_boot$ATT_se, "\n")
```

```
## SE (boot): 0.1550489
```

```
cat("CI:", res_twfe_boot$CI_lower, "to", res_twfe_boot$CI_upper, "\n")
```

```
## CI: 1.304683 to 1.912475
```

```
p_res_twfe_boot <- esplot(data = res_twfe_boot,
                  main = "TWFE with Cluster Bootstrap", xlim = c(-12,9))
print(p_res_twfe_boot)
```



Rambachan and Roth Sensitivity Analysis

Fit fect with `placeboTest = TRUE` and run the sensitivity analysis

```
out.fect.placebo <- fect(
  nat_rate_ord ~ indirect,
  data  = hh2019,
  index = c("bfs", "year"),
  method        = 'fe',
  se            = TRUE,
  placeboTest   = TRUE,
  placebo.period = c(-2, 0)
)
```

```
## For identification purposes, units whose number of untreated periods <1 are dropped automatically.

## Parallel computing ...

## Bootstrapping for uncertainties ...

## 200 runs
```

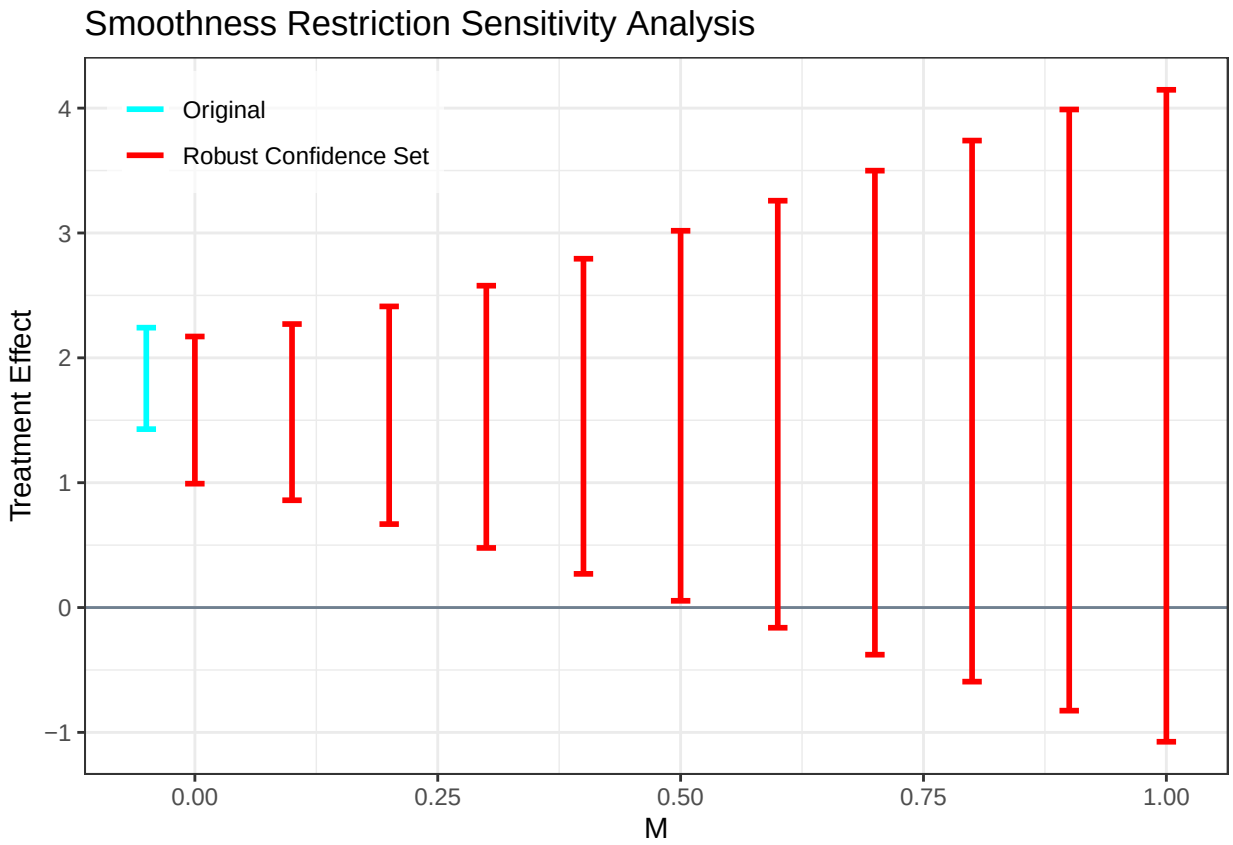
```
out.fect.placebo <- fect_sens_anlys(
  fect.out     = out.fect.placebo,
  post.periods = 1:10,
  Mbarvec      = seq(0,1, by=0.1),
```

```
  periodMbarvec      = seq(0,1, by=0.5),
  Mvec         = seq(0,0.25,0.05),
  periodMvec         = seq(0,0.25,0.25)
)
```

Plotting

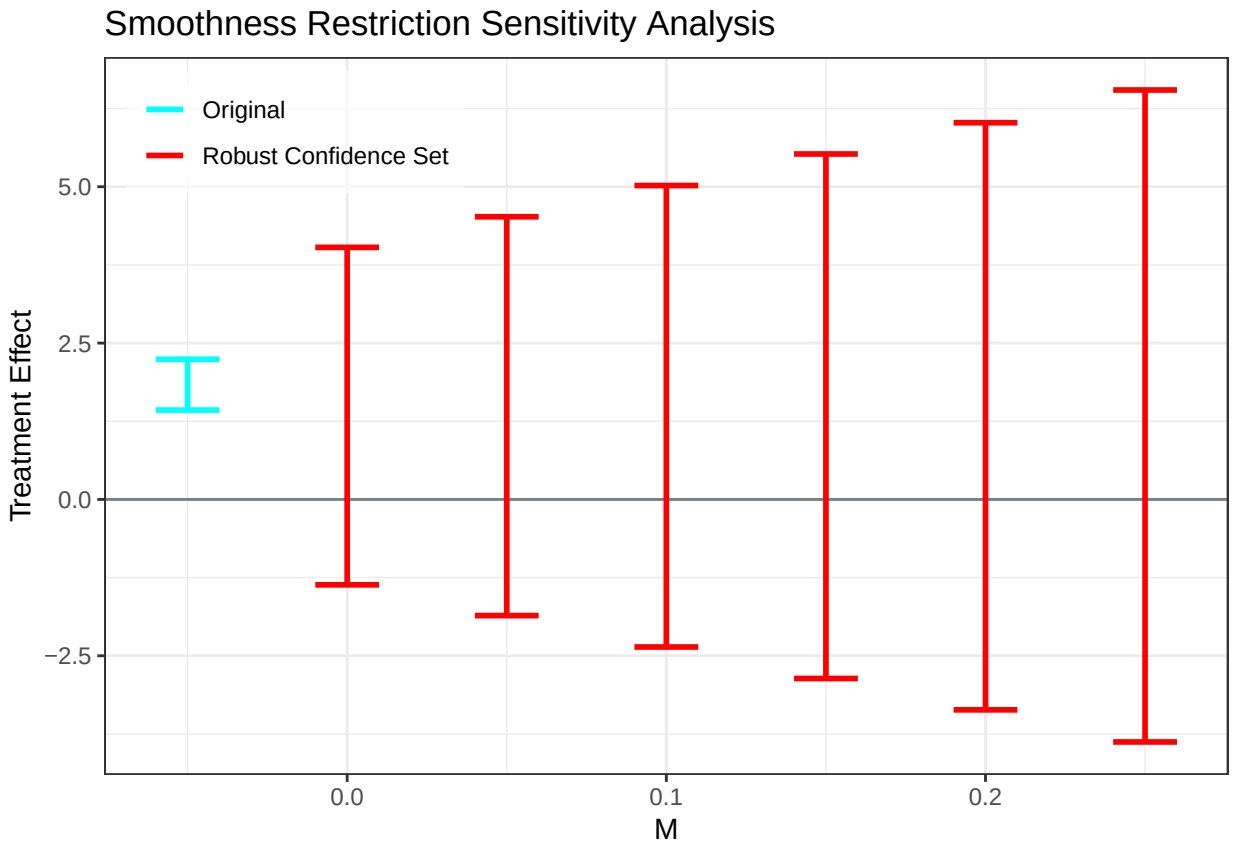
Relative Magnitude Restriction

```
plot(out.fect.placebo, type = "sens_rm", main = "Relative Magnitude Restriction")
```



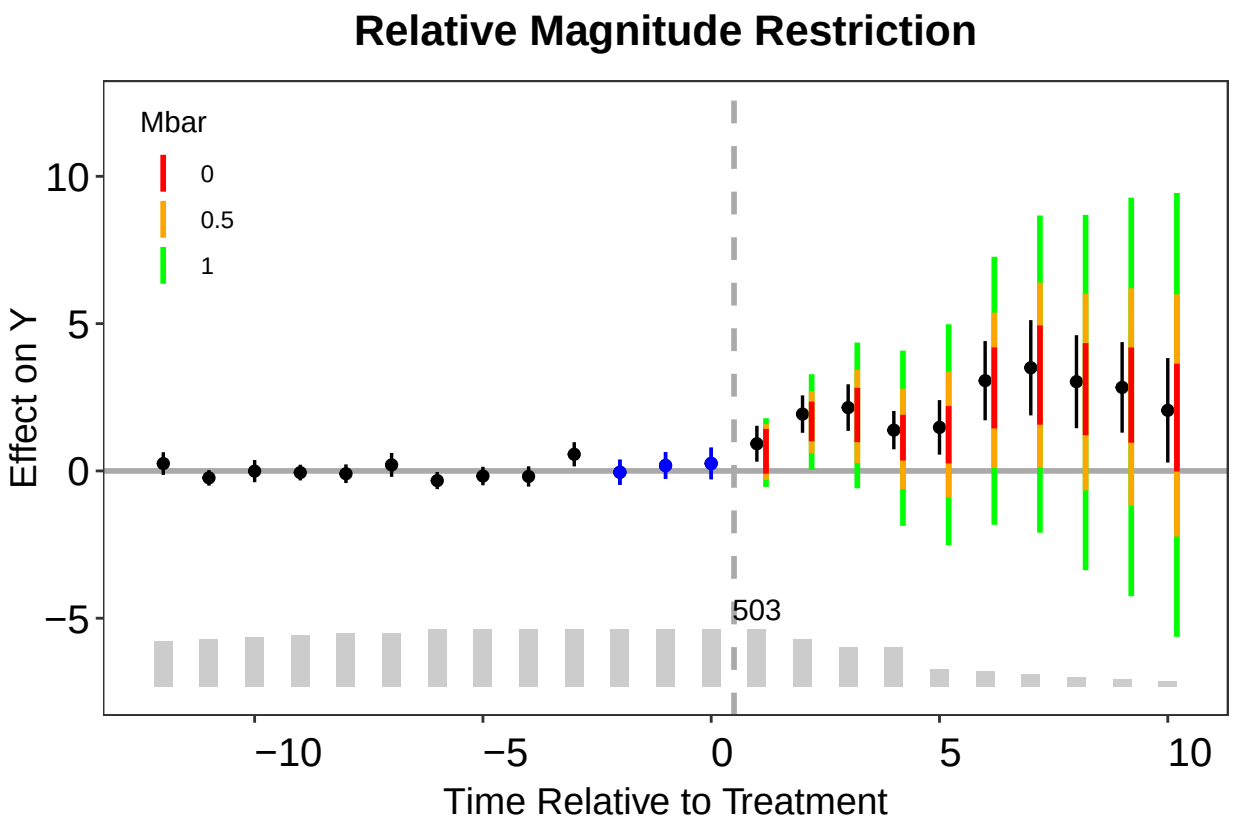
Smoothness Restriction

```
plot(out.fect.placebo, type = "sens_smooth", main = "Smoothness Restriction")
```

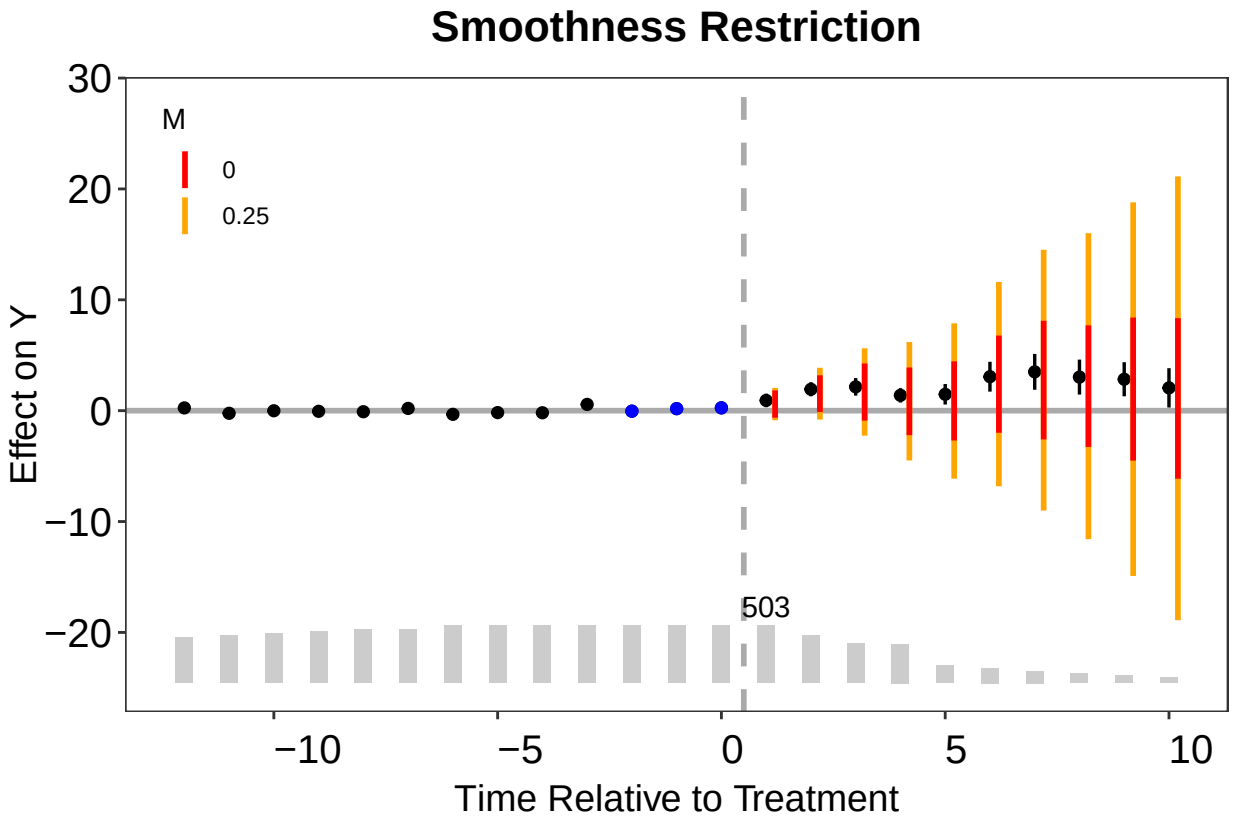


Relative Magnitude Restriction Gaps Plot

```
plot(out.fect.placebo, xlim = c(-12,10), type = "sens_rm_gaps", main = "Relative Magnitude Restriction")
```

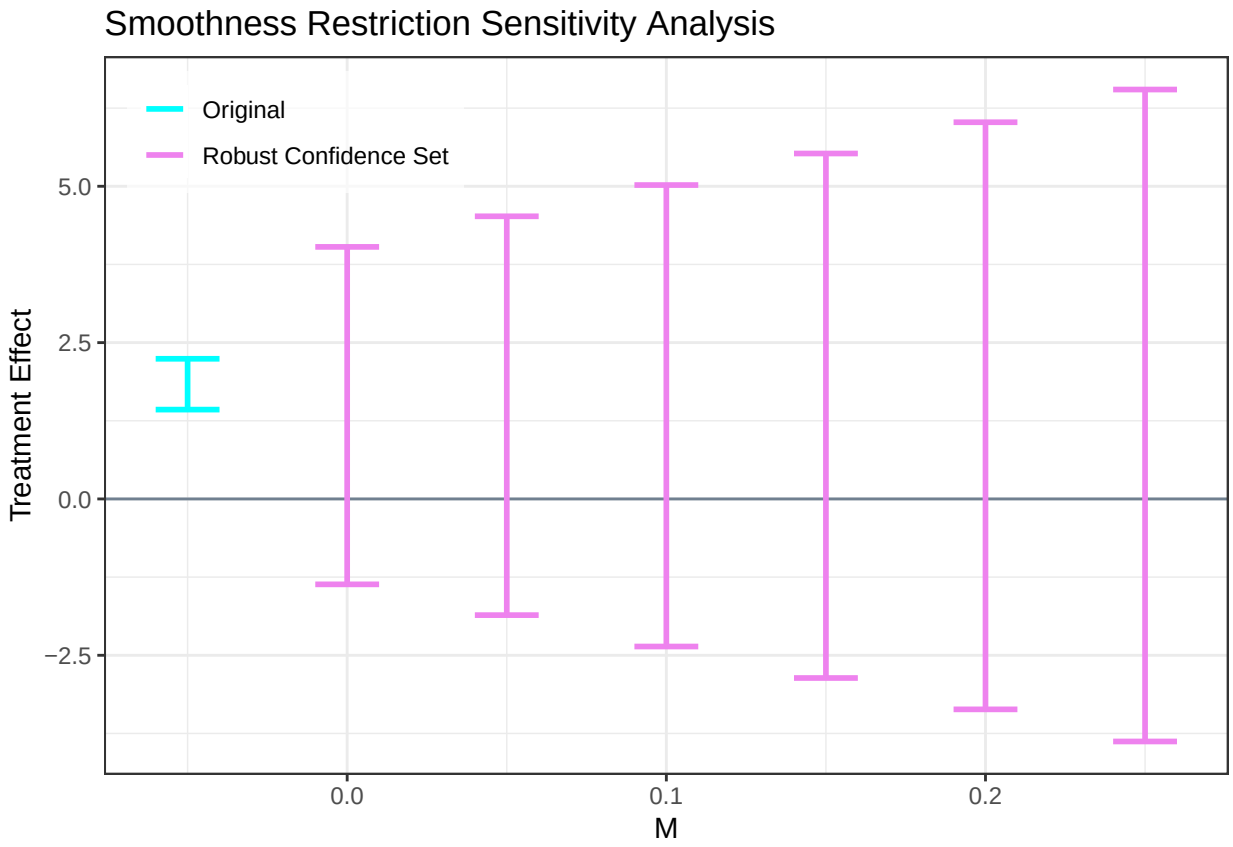



```
plot(out.fect.placebo,  xlim = c(-12,10),type = "sens_smooth_gaps", main = "Smoothness Restriction")
```

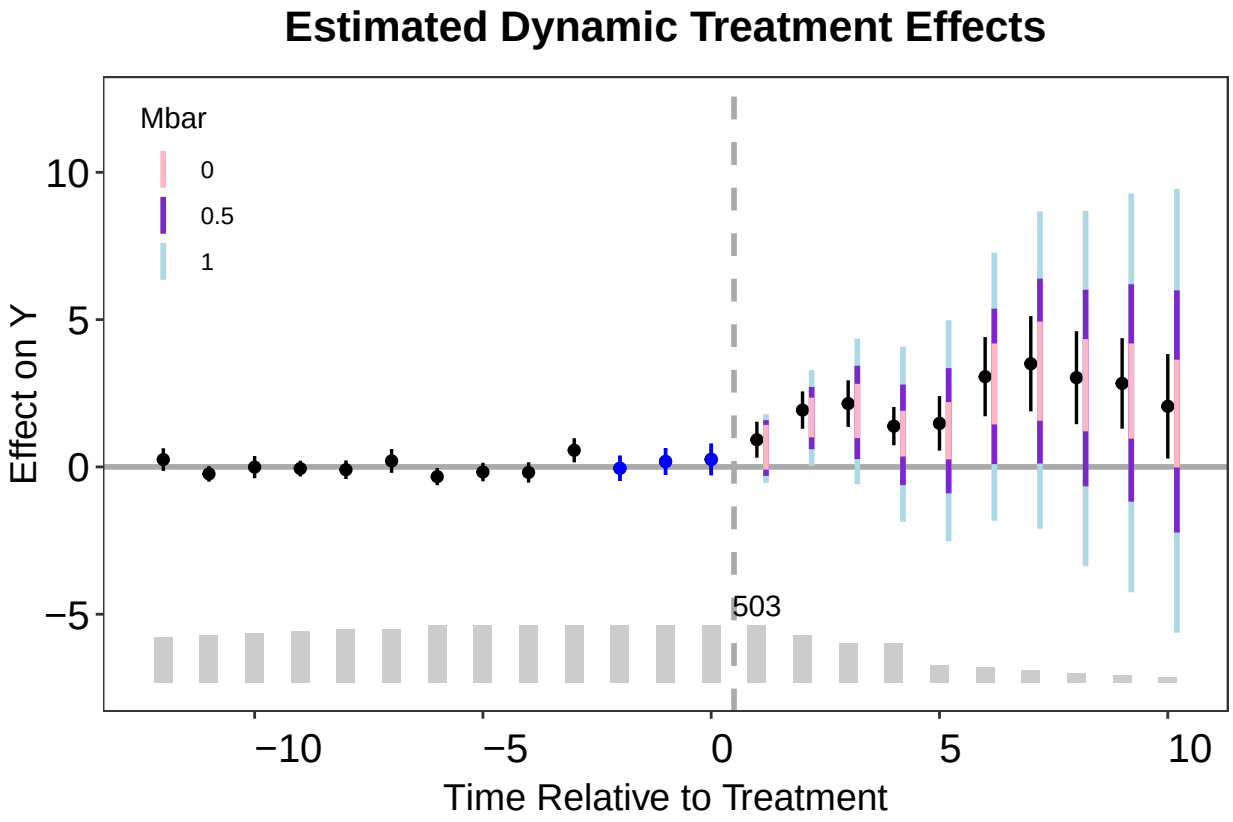


Changing color

```
plot(out.fect.placebo, type = "sens_smooth",sens.robust.colors = c("violet"))
```




```
plot(out.fect.placebo, xlim = c(-12,10), type = "sens_rm_gaps",sens.robust.colors = c("pink1","purple3"
```



Improved plotting

Creating fect object

```
out.fect <- fect(Y ~ D + X1 + X2, data = simdata, index = c("id", "time"),
                         force = "two-way", method = "ife", r = 2, CV = 0,
                         parallel = TRUE, se = TRUE, nboots = 200)
```

```
## Parallel computing ...
```

```
## Bootstrapping for uncertainties ...
```

```
## 200 runs
```

```
out.fect.p <- fect(Y ~ D + X1 + X2, data = simdata, index = c("id", "time"),
                         force = "two-way", method = "ife", r = 2, CV = 0,
                         parallel = TRUE, se = TRUE,
                         nboots = 200, placeboTest = TRUE, placebo.period = c(-2, 0))
```

```
## Parallel computing ...
```

```
## Bootstrapping for uncertainties ...
```

```
## 200 runs
```

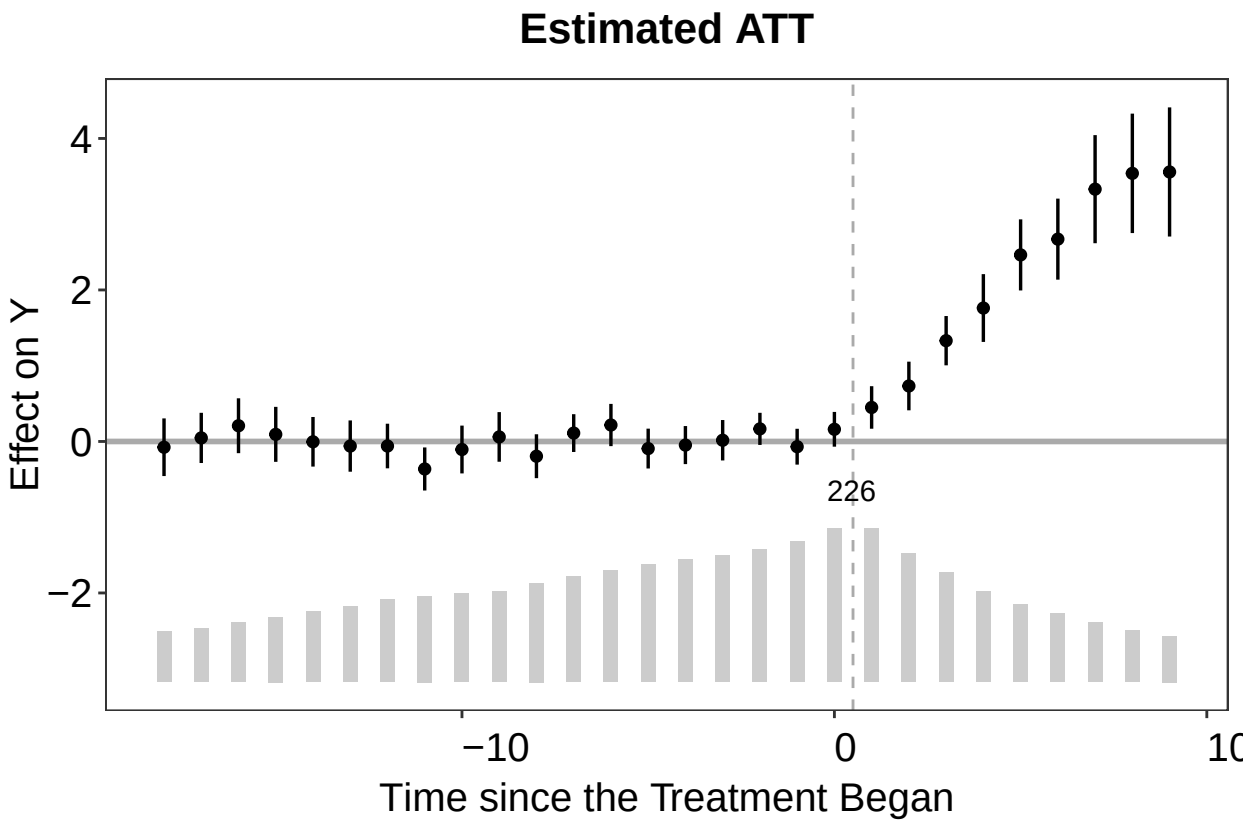
```
out.fect.c <- fect(Y ~ D + X1 + X2, data = simdata, index = c("id", "time"),
                         force = "two-way", method = "ife", r = 2, CV = 0,
                         parallel = TRUE, se = TRUE,  carryover.rm = 3,
                         nboots = 200, carryoverTest = TRUE, carryover.period = c(1, 3))
```

```
## Parallel computing ...
```

```
## Bootstrapping for uncertainties ...
## 200 runs
```

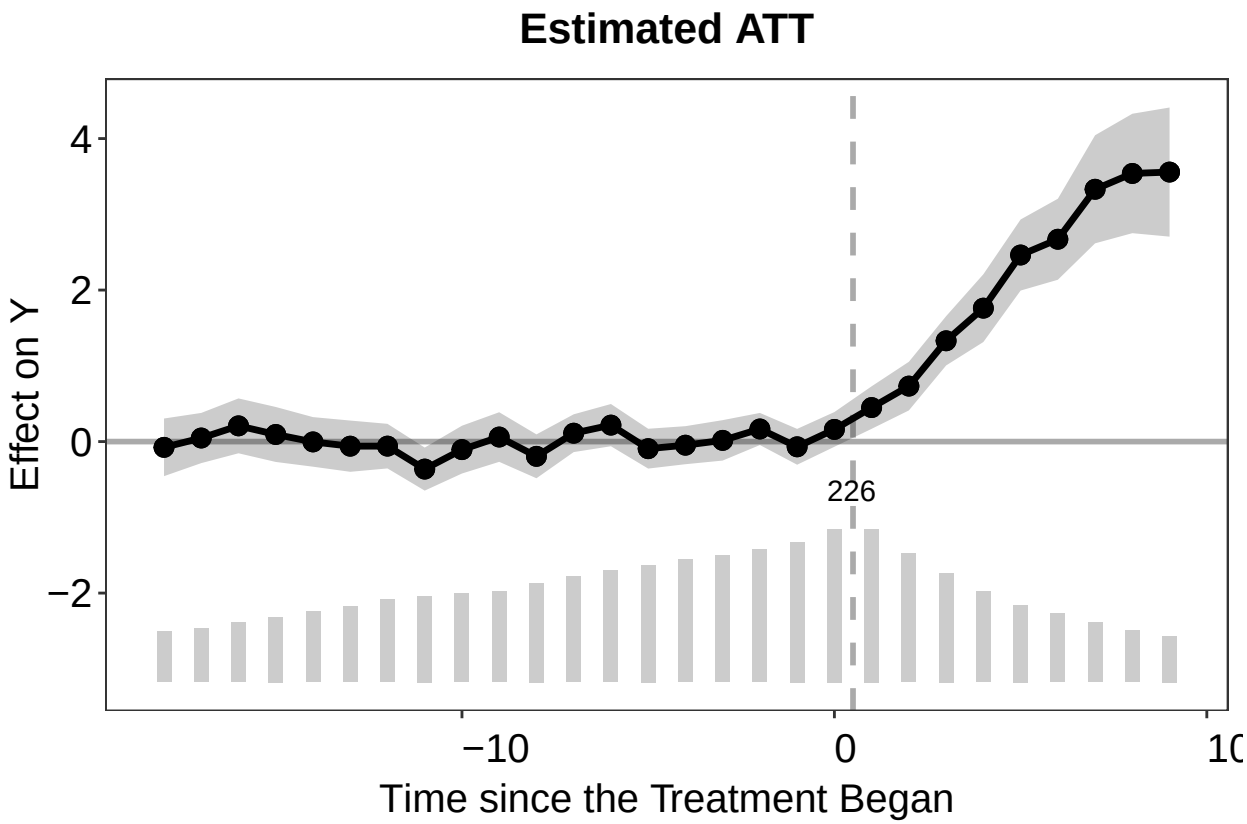
###Gaps plot Disconnected points

```
plot(out.fect)
```



Connected points

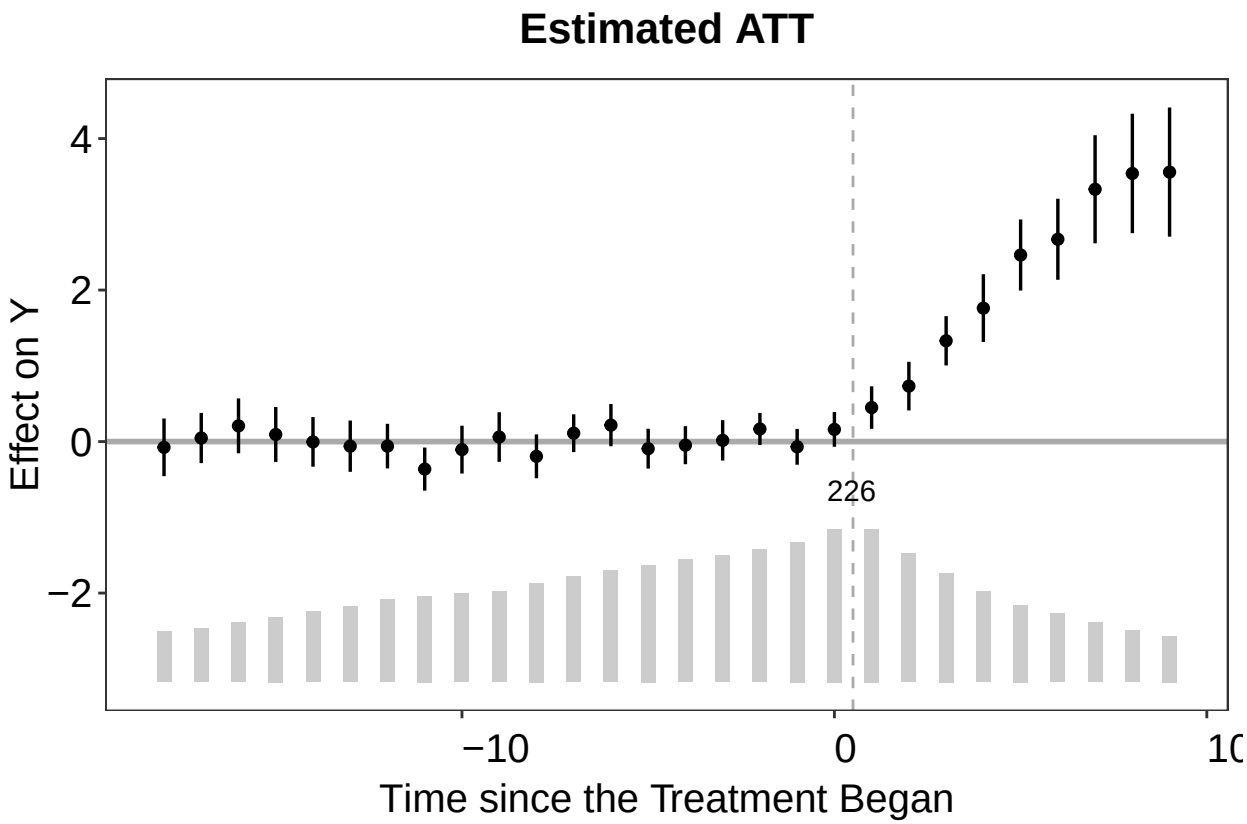
```
plot(out.fect, connected = TRUE)
```



`vis` is deprecated now

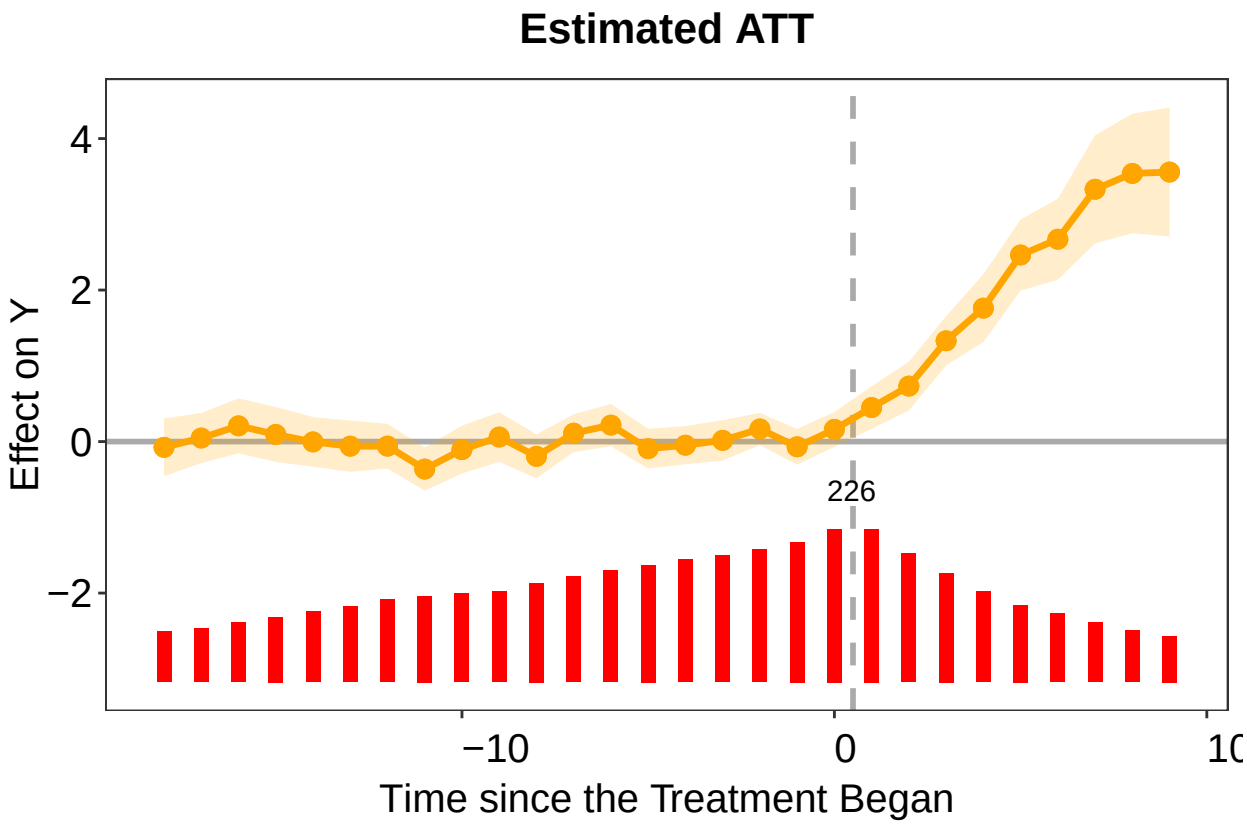
```
plot(out.fect, vis = "none")
```

```
## Warning: 'vis' is deprecated and will be removed in future versions.
```



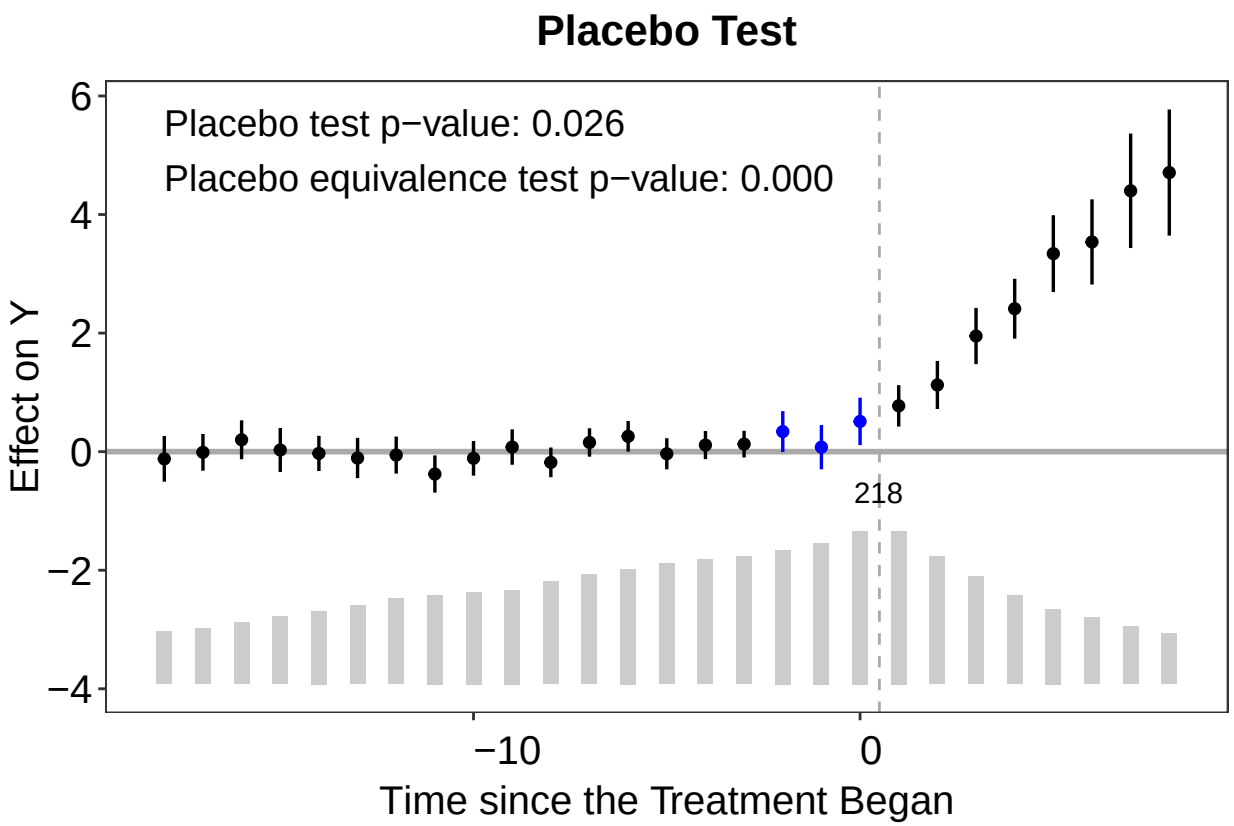
Custom colors

```
plot(out.fect, connected = TRUE, count.color = "red", color = "orange")
```



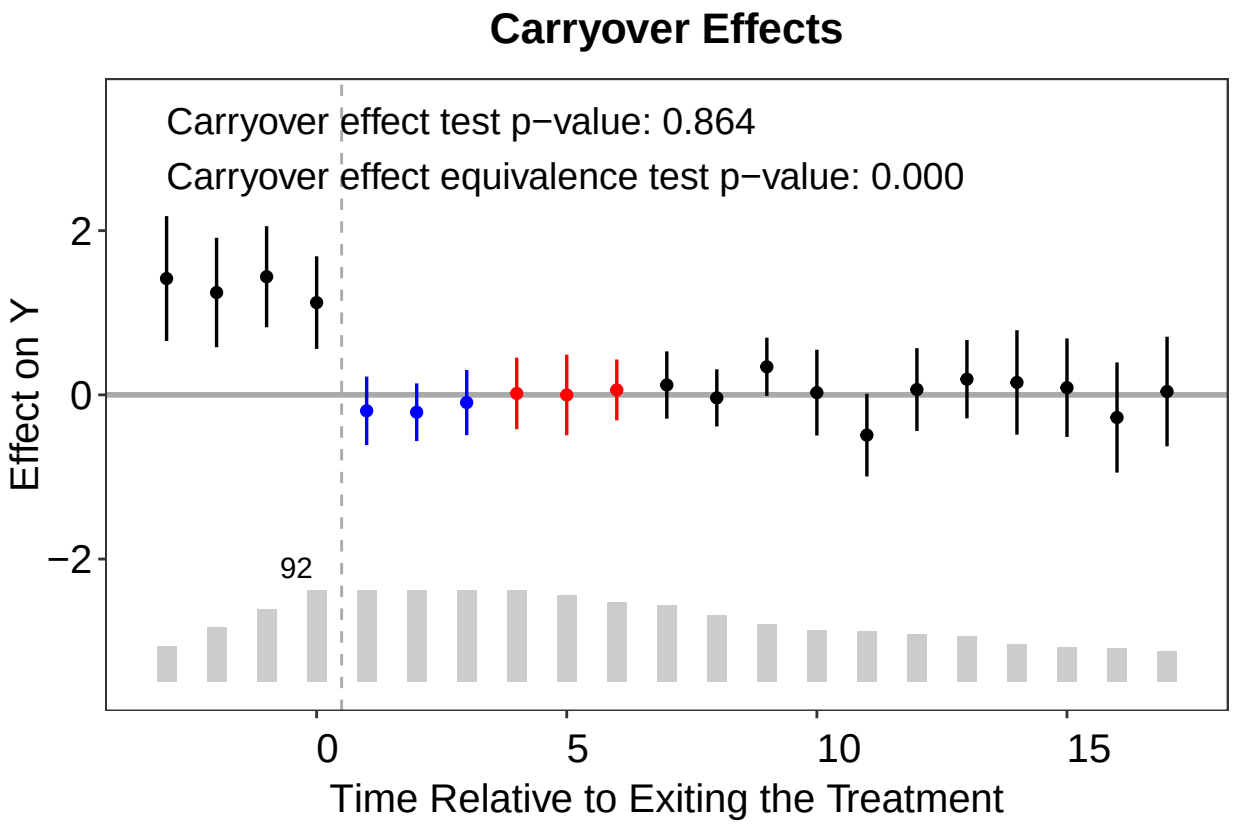
###Placebo plot

```
plot(out.fect.p)
```



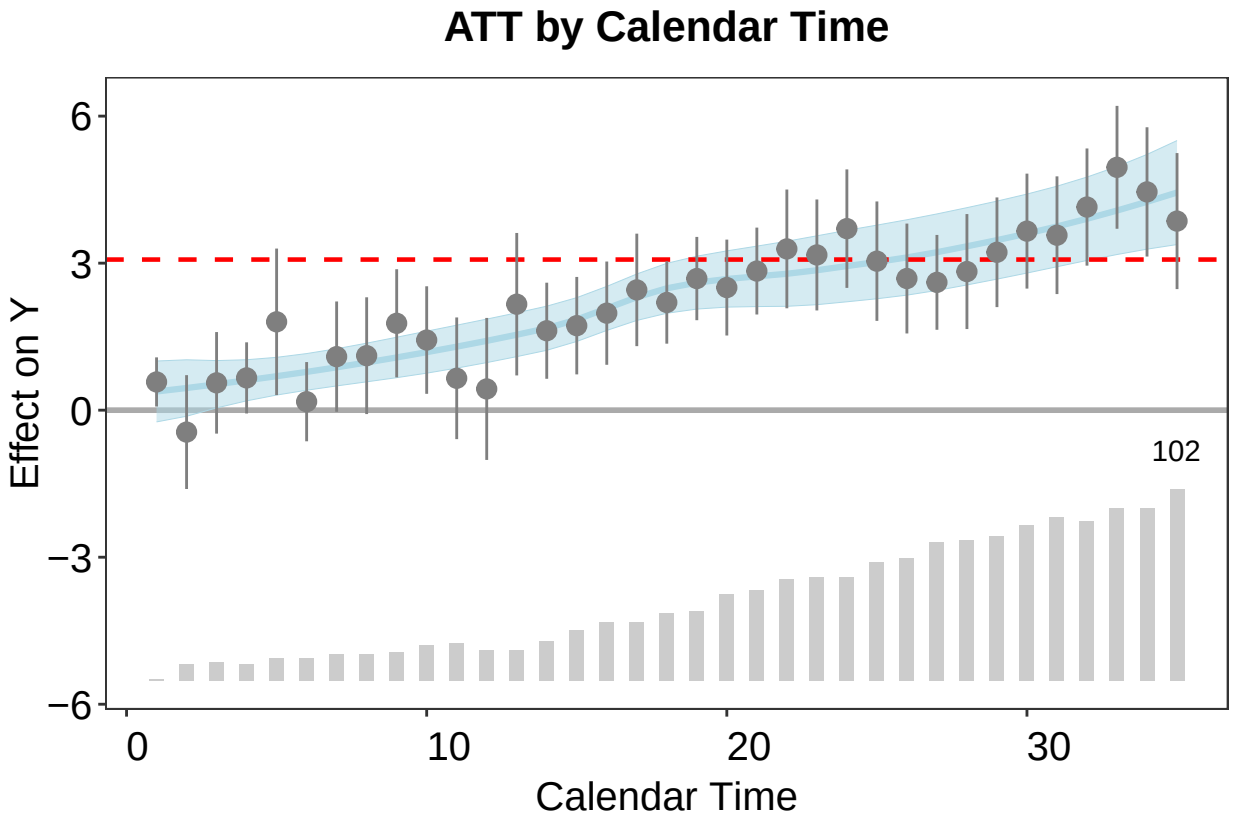
###Carryover plot

```
plot(out.fect.c,ylim = c(-3.5,3.5))
```



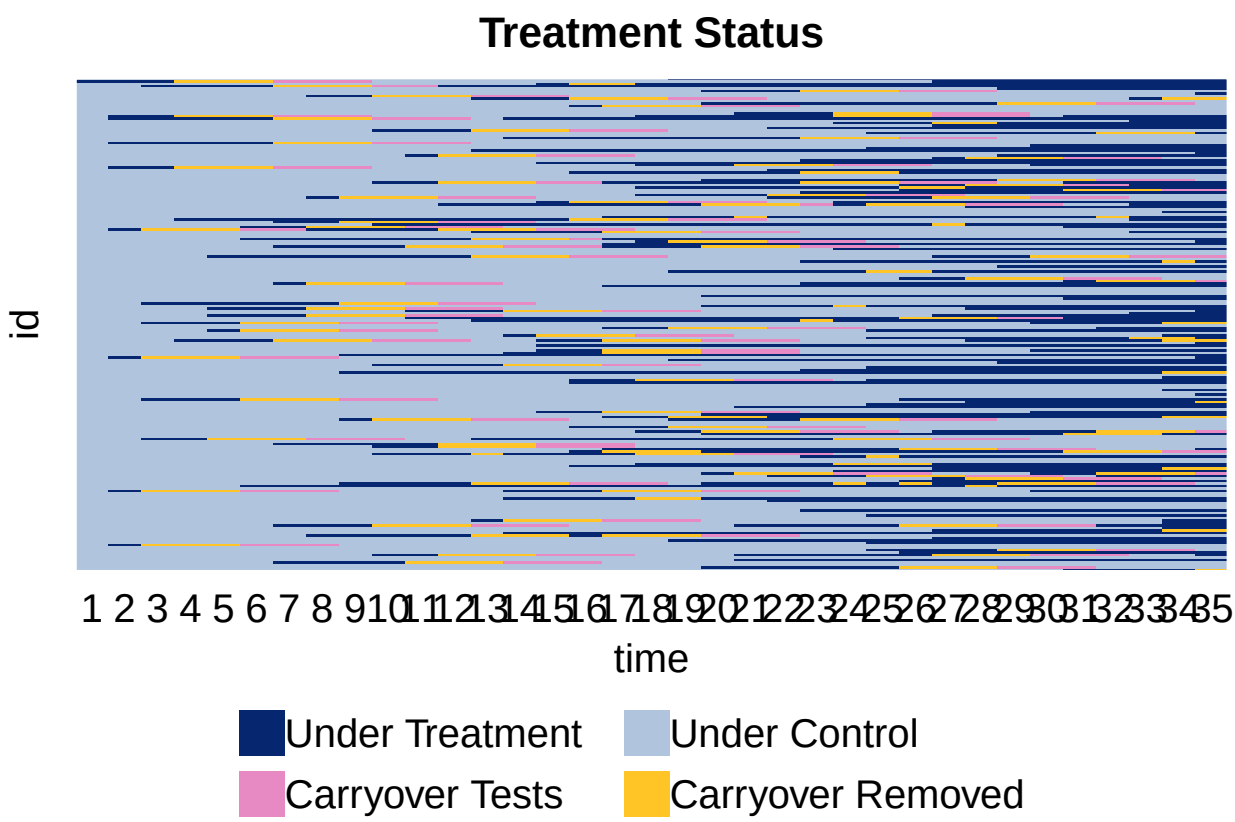
###Calendar plot

```
plot(out.fect, type = "calendar", xlim = c(1, 35))
```



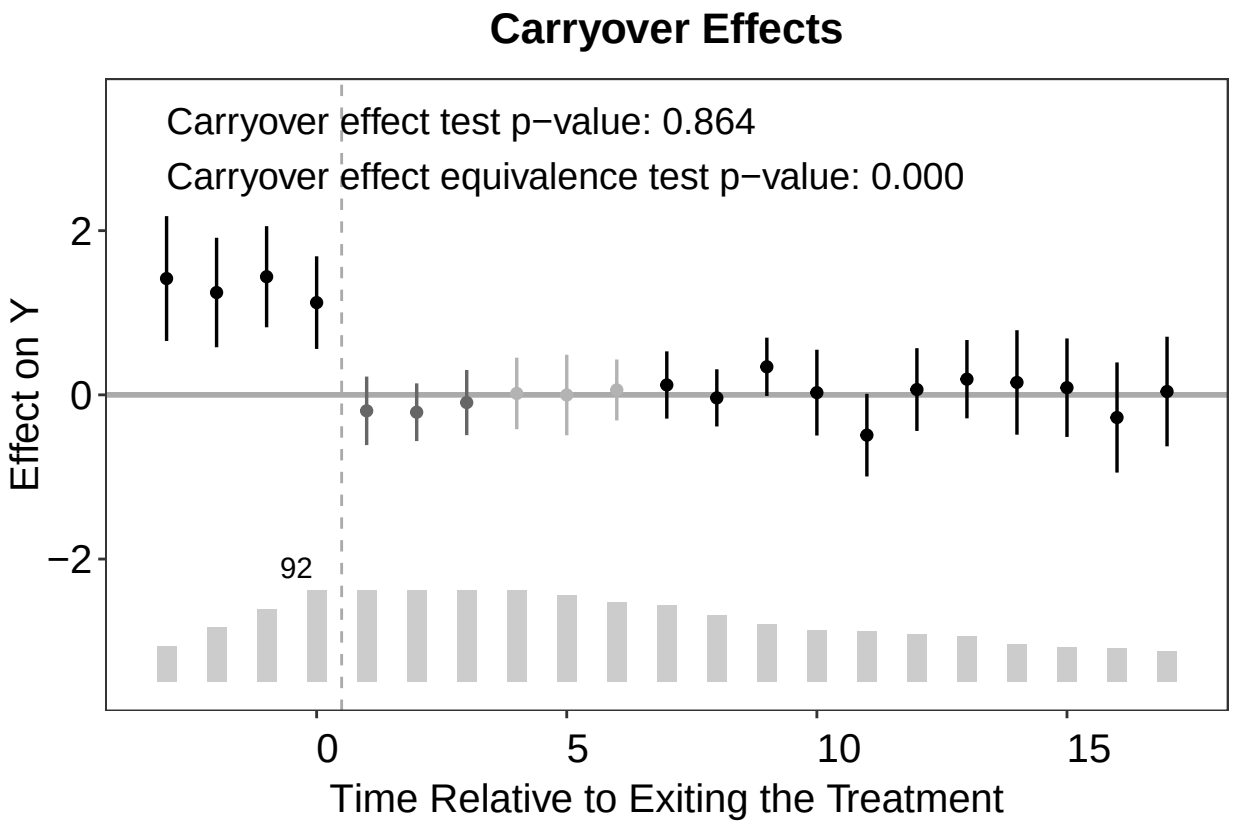
###Status plot

```
plot(out.fect.c, type="status")
```



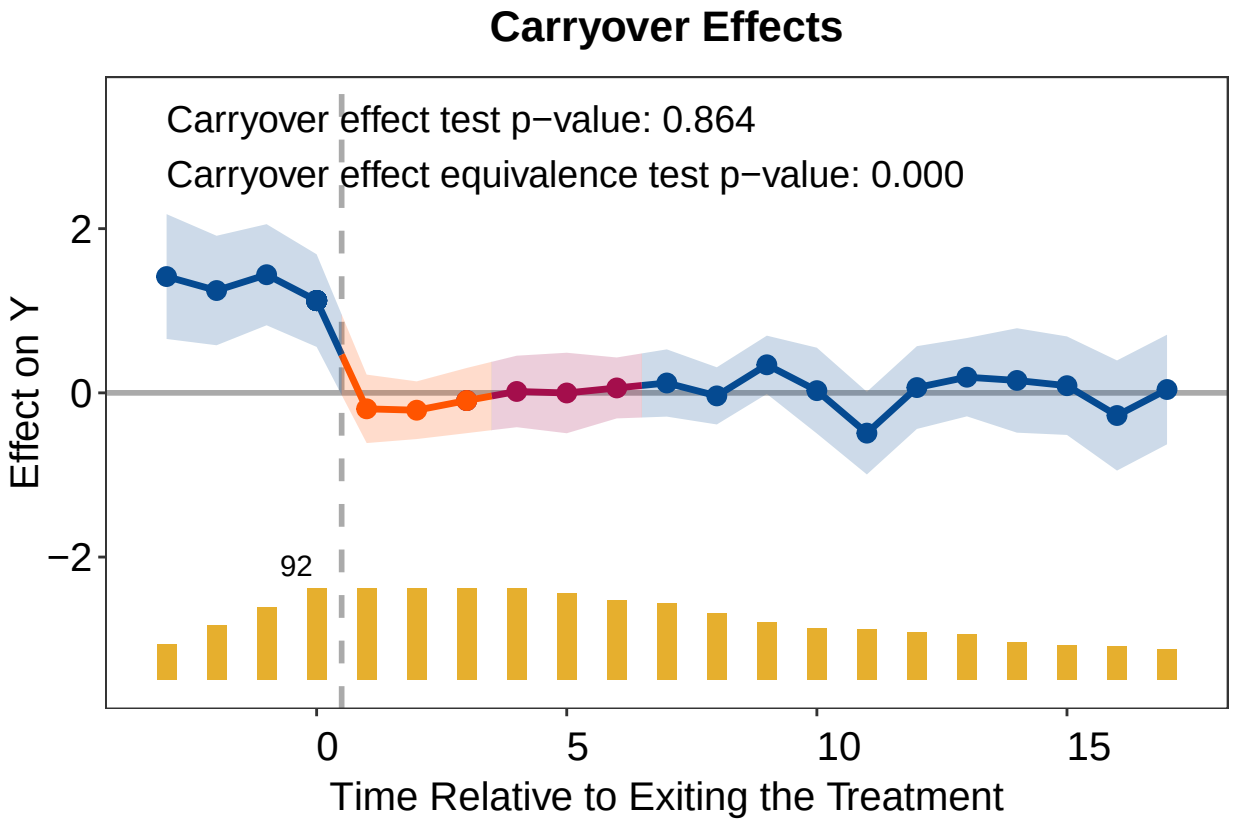
###Themes Grayscale

```
plot(out.fect.c,ylim = c(-3.5,3.5), theme = "grayscale")
```



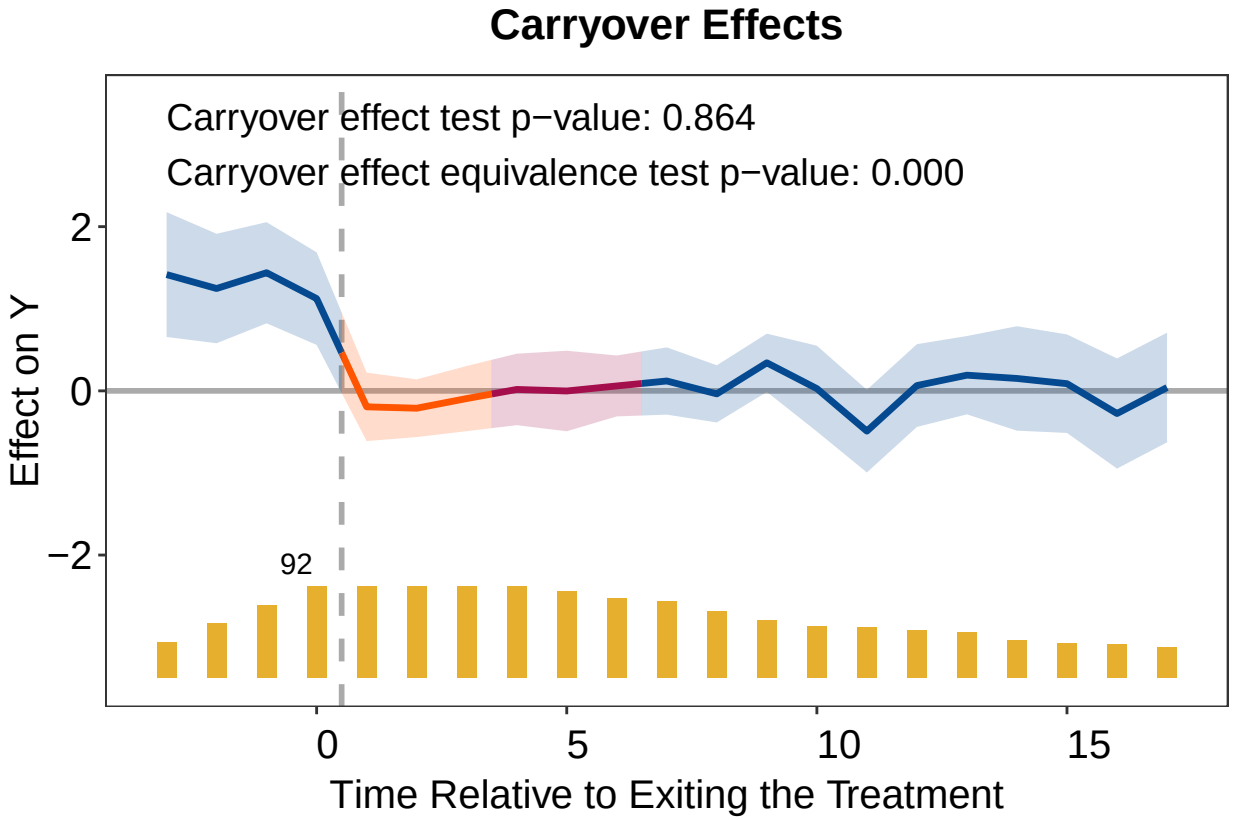
Vibrant

```
plot(out.fect.c,ylim = c(-3.5,3.5), theme = "vibrant")
```



Vibrant without points

```
plot(out.fect.c,ylim = c(-3.5,3.5), theme = "vibrant", show.points = FALSE)
```



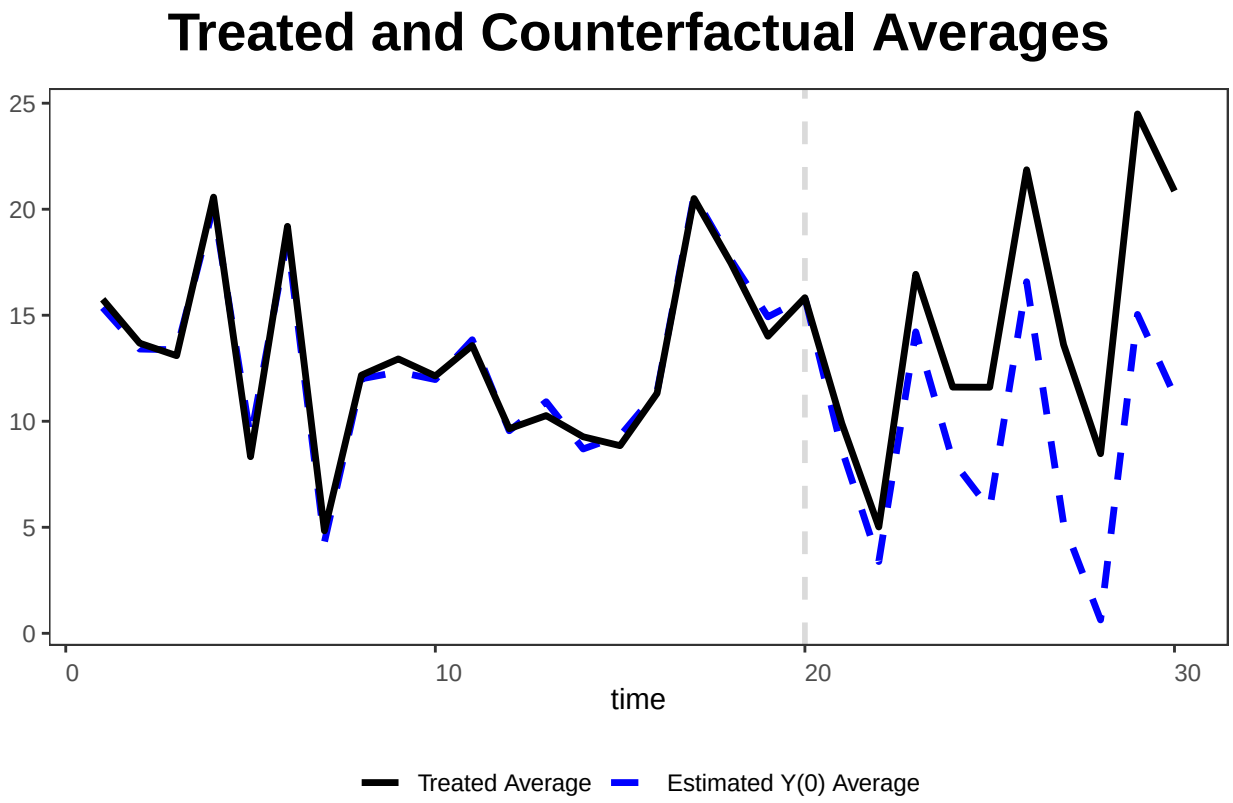
###Gsynth

```
out <- fect(Y ~ D + X1 + X2, data = simgsynth, index = c("id","time"),
            method = "gsynth", force = "two-way", CV = TRUE, r = c(0, 5),
            se = TRUE, nboots = 1000,vartype = 'parametric', parallel = TRUE, cores = 16)
```

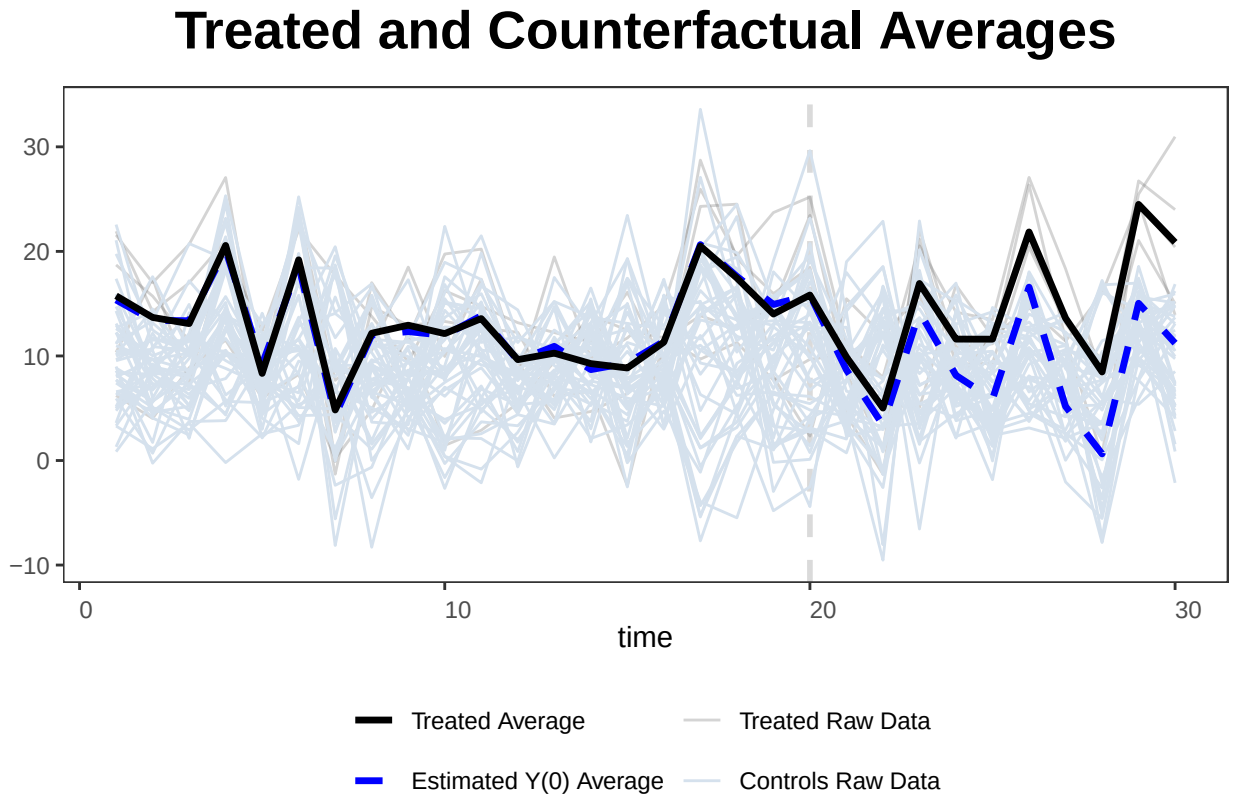
```
## Parallel computing ...
## Cross-validating ...
## Criterion: Mean Squared Prediction Error
## Interactive fixed effects model...
## Cross-validating ...
## r = 0; sigma2 = 1.84865; IC = 1.02023; PC = 1.74458; MSPE = 2.37280
## r = 1; sigma2 = 1.51541; IC = 1.20588; PC = 1.99818; MSPE = 1.71743
## r = 2; sigma2 = 0.99737; IC = 1.16130; PC = 1.69046; MSPE = 1.14540
## *
## r = 3; sigma2 = 0.94664; IC = 1.47216; PC = 1.96215; MSPE = 1.15032
## r = 4; sigma2 = 0.89411; IC = 1.76745; PC = 2.19241; MSPE = 1.21397
## r = 5; sigma2 = 0.85060; IC = 2.05928; PC = 2.40964; MSPE = 1.23876
##
##  r* = 2
##
## Bootstrapping for uncertainties ...
```

```
## Parametric Bootstrap
## Simulating errors ...
## 1000 runs
## Can't calculate the F statistic because of insufficient treated units.
```

```
plot(out, type = "counterfactual")
```




```
plot(out, type = "counterfactual", raw = "all")
```




```
plot(out, type = "counterfactual", raw = "band")
```