SLPP

```
library(slpp)
#> Loading required package: data.table
```

This vignette presents some of the basic uses of slpp through Arezki, Ramey, and Sheng (QJE 2017). First, load in the data (downloaded from the replication page) and transform it into a panel.

```
ars = haven::read_dta("/Users/jacksonmejia/Downloads/ARS_Empirics/newspanel2016.dta")
# Filter out observations which do not have an oil news shock (sizerealistic) and which do not have an
ars <- ars[!is.na(ars$sizerealistic) & !is.na(ars$ca),]
# Set panel identifiers
ars <- fixest::panel(ars, panel.id = c("ifscode", "year"))
```

The panel is unbalanced:

```
#Number of countries
length(unique(ars$ifscode))
#> [1] 180
#Number of time periods
length(unique(ars$year))
#> [1] 53
```

Next, we define the regression formula using fixest notation. Our desired IRF comes from

$${\frac{\text{Current Account}}{GDP}}_{i,t+h} = \alpha_i + T_t + \beta_h \times \text{Oil News Shock}_{i,t} + \varepsilon_{i,t+h},$$

where α_i is a country fixed effect and T_t is a time fixed effect. Say that the maximum horizon is $H = 20$. Then we define the regression formula as

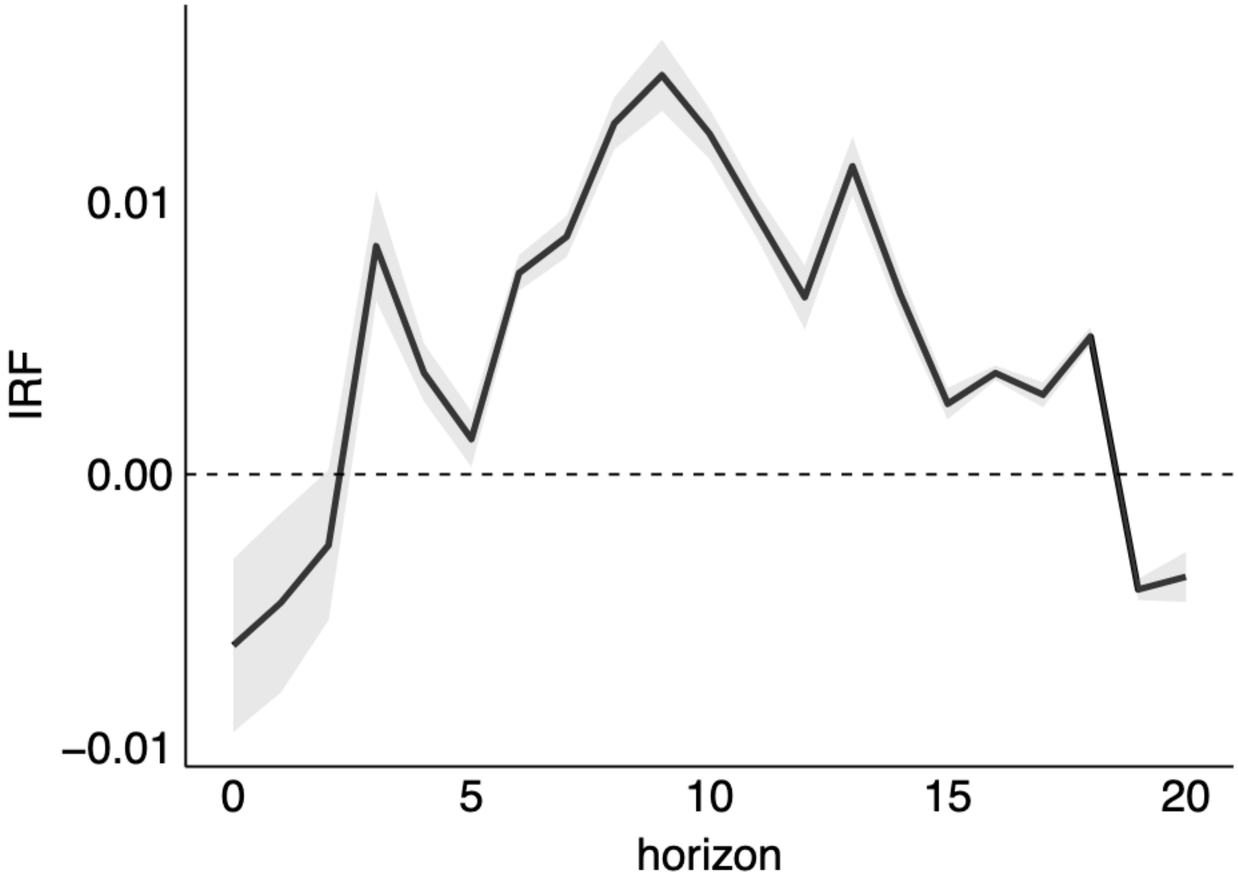
```
reg_formula <- f(ca,0:20)  ~ sizerealistic | ifscode + year
```

Now, suppose we want to look at smoothed IRFs for polynomials of order 1 and 2 with 1000 bootstraps each. We set r = c(1,2) in the function call. Note that this isn't important and we could have simply set r = 2. Then we could run the following code.

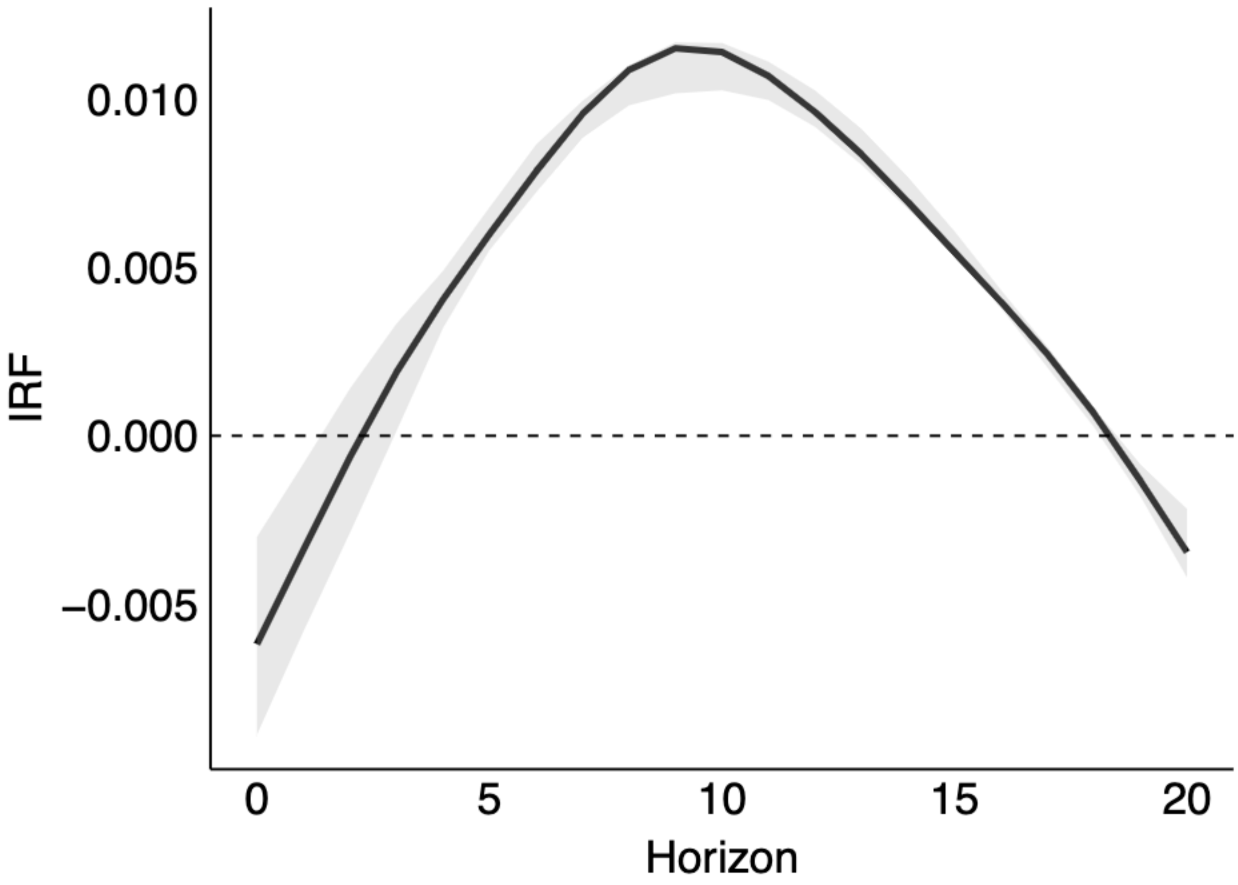
```
t1 <- Sys.time()
slpp_obj <- feols_slpp(reg_formula, r = c(1,2),
                 paneldf = ars, type = "smooth", targetvar = "sizerealistic",boots = 1000)
t2 <- Sys.time()
```

This returns a list with the regular local projection, info about the specification, and a list of smooth local projections. We can call the smoothed IRFs for each from, for example, the following code.

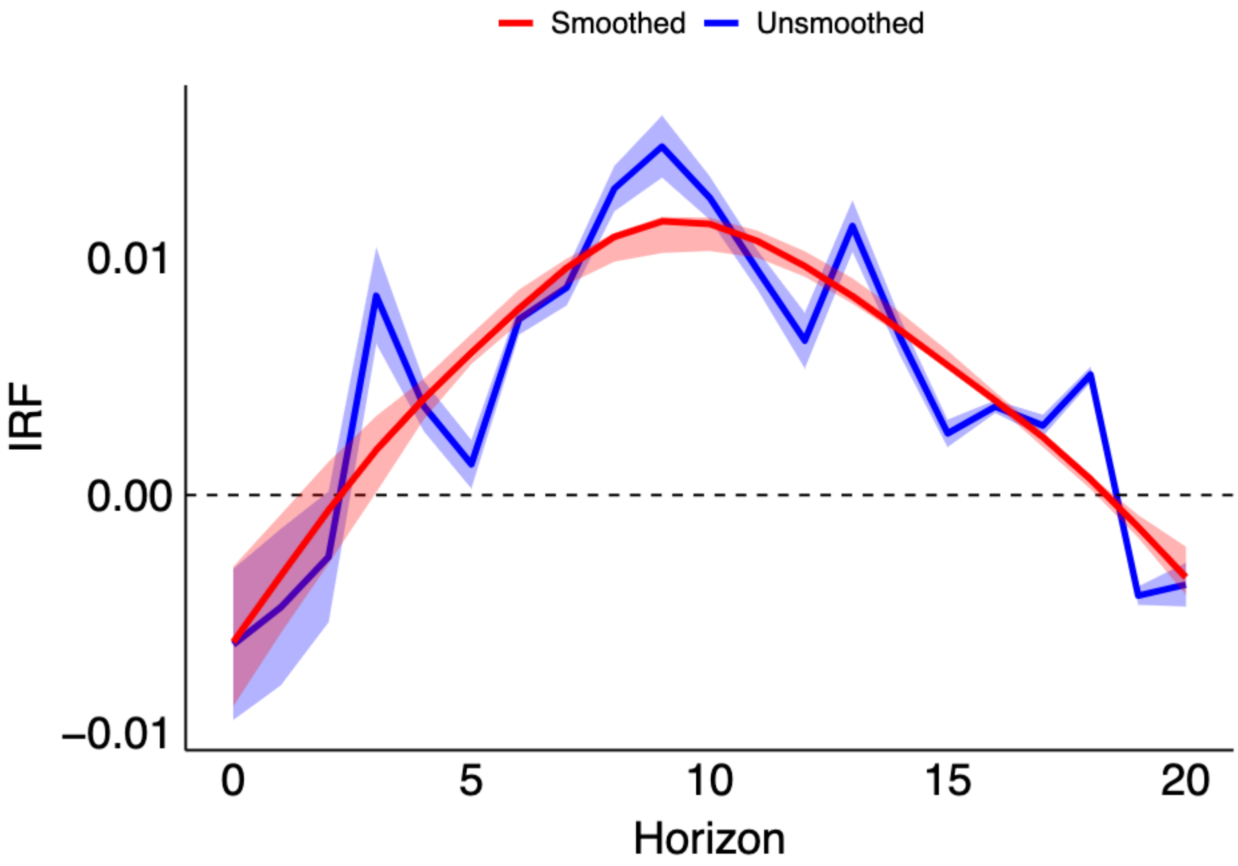
```
## Plot of PLP estimator
slpp_obj$plp$regplot
```



```
## Plot of order 1 SLPP estimator. To get the second order, we would call slpp_obj$slpp$order_1$plot
slpp_obj$slpp$order_1$plot
```



```
## Plot of both on top of each other.
slpp_obj$slpp$order_1$comp_plot
```



We can also access the IRFs as a dataframe.

```
## IRF for PLP
slpp_obj$plp$regcoefdf
#>    horizon           cil           cih          IRF
#> 1        0 -0.0094595808 -0.0031086512 -0.006284116
#> 2        1 -0.0079993224 -0.0014254130 -0.004712368
#> 3        2 -0.0053530203  0.0001573323 -0.002597844
#> 4        3  0.0063637842  0.0104015284  0.008382656
#> 5        4  0.0026783344  0.0047976920  0.003738013
#> 6        5  0.0002810964  0.0023161759  0.001298636
#> 7        6  0.0067410644  0.0080573736  0.007399219
#> 8        7  0.0079786525  0.0094872223  0.008732937
#> 9        8  0.0119326580  0.0138509475  0.012891803
#> 10       9  0.0133480271  0.0159703216  0.014659174
#> 11      10  0.0116001918  0.0134276061  0.012513899
#> 12      11  0.0086694337  0.0103074121  0.009488423
#> 13      12  0.0053305777  0.0076749189  0.006502748
#> 14      13  0.0102351627  0.0123959412  0.011315552
#> 15      14  0.0058634315  0.0074164688  0.006639950
#> 16      15  0.0020170855  0.0031689802  0.002593033
#> 17      16  0.0034530045  0.0039863489  0.003719677
#> 18      17  0.0024652166  0.0033803028  0.002922760
#> 19      18  0.0047360481  0.0053959519  0.005066000
#> 20      19 -0.0046148010 -0.0038416600 -0.004228231
#> 21      20 -0.0046806454 -0.0028526538 -0.003766650
## IRF for second order SLPP estimator
slpp_obj$slpp$order_2$irf
#> Key: <horizon>
#>     horizon           cil           cih           IRF
#>       <int>         <num>         <num>         <num>
#>  1:       0 -0.0100714617 -0.0036720235 -0.0069513149
#>  2:       1 -0.0066170424 -0.0003765646 -0.0033138489
#>  3:       2 -0.0036327420  0.0023125097 -0.0002135069
#>  4:       3 -0.0002967690  0.0039433296  0.0021923154
#>  5:       4  0.0030506356  0.0049445443  0.0039225704
#>  6:       5  0.0047875494  0.0070947905  0.0054871328
#>  7:       6  0.0063390436  0.0093875495  0.0074012813
#>  8:       7  0.0086472435  0.0107818074  0.0095972894
#>  9:       8  0.0106012675  0.0117434404  0.0114939305
#> 10:       9  0.0112827806  0.0125861072  0.0124272745
#> 11:      10  0.0111501279  0.0124750846  0.0121400918
#> 12:      11  0.0102776739  0.0115545001  0.0109371446
#> 13:      12  0.0088753749  0.0102126156  0.0093640040
#> 14:      13  0.0073008460  0.0086304224  0.0077799160
#> 15:      14  0.0059374892  0.0070832487  0.0062972669
#> 16:      15  0.0048150389  0.0056181009  0.0050334354
#> 17:      16  0.0037275851  0.0043057089  0.0040057980
#> 18:      17  0.0022823268  0.0031341066  0.0029190860
#> 19:      18  0.0005647277  0.0014603128  0.0013104476
#> 20:      19 -0.0015935390 -0.0008489397 -0.0011211325
#> 21:      20 -0.0052214779 -0.0029824546 -0.0043834931
#>     horizon           cil           cih           IRF
```

Within the list, one can also access the original estimates for each horizon of the PLP estimator as well as the matrices, penalty parameters, and so on for the SLPP estimator.

```
t2 - t1
#> Time difference of 1.084737 mins
```

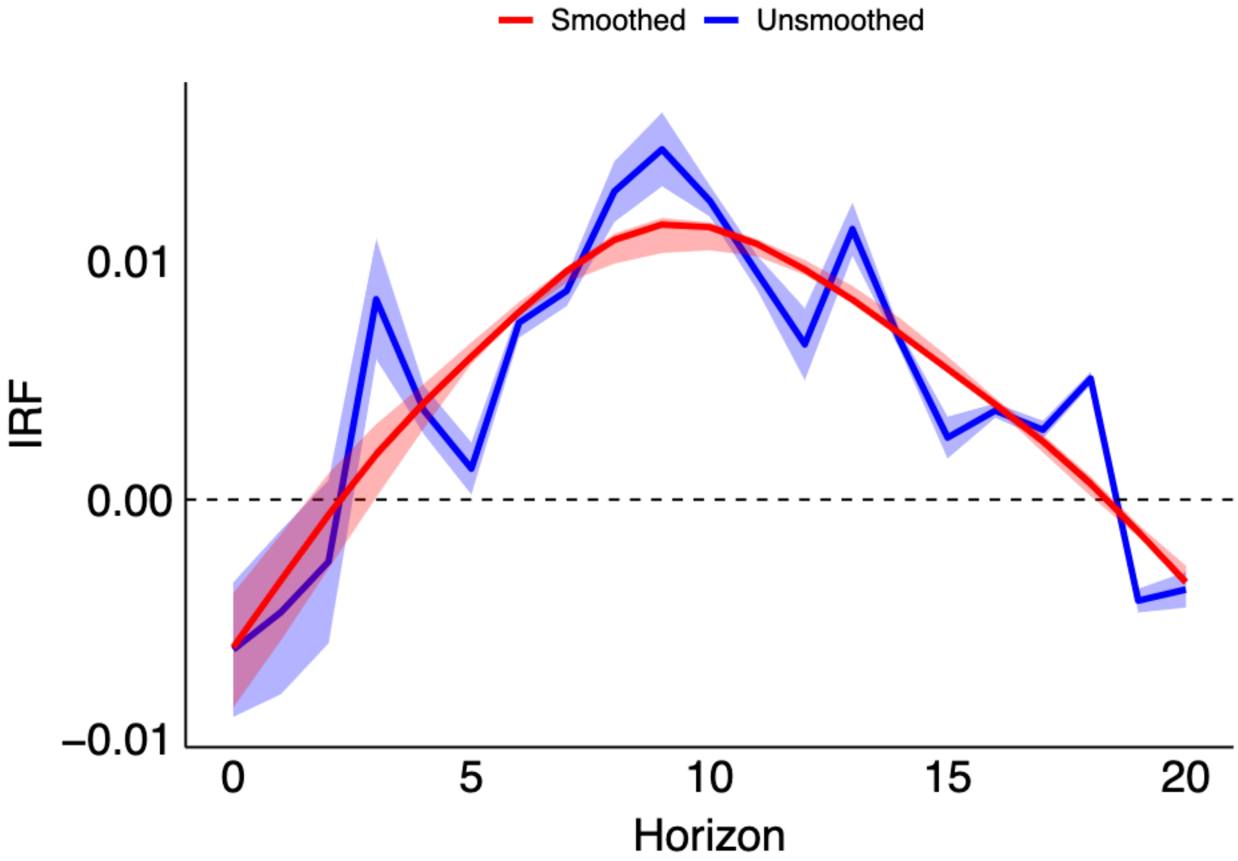
One may also be interested in a number of other different operations. Because the code is a wrapper for fixest, we can do anything permissible in that notation as long as it isn't too computationally expensive. For example, we can add regional fixed effects:

```
reg_formula2 <- f(ca,0:20)  ~ sizerealistic | ifscode + year + region_weo
```

Or we can cluster by region instead by setting clustervar = region_weo in the function call.

```
slpp_obj2 <- feols_slpp(reg_formula2, r = c(1),
                 paneldf = ars, type = "smooth", targetvar = "sizerealistic",boots = 100,clustervar = "
```

```
slpp_obj2$slpp$order_1$comp_plot
```

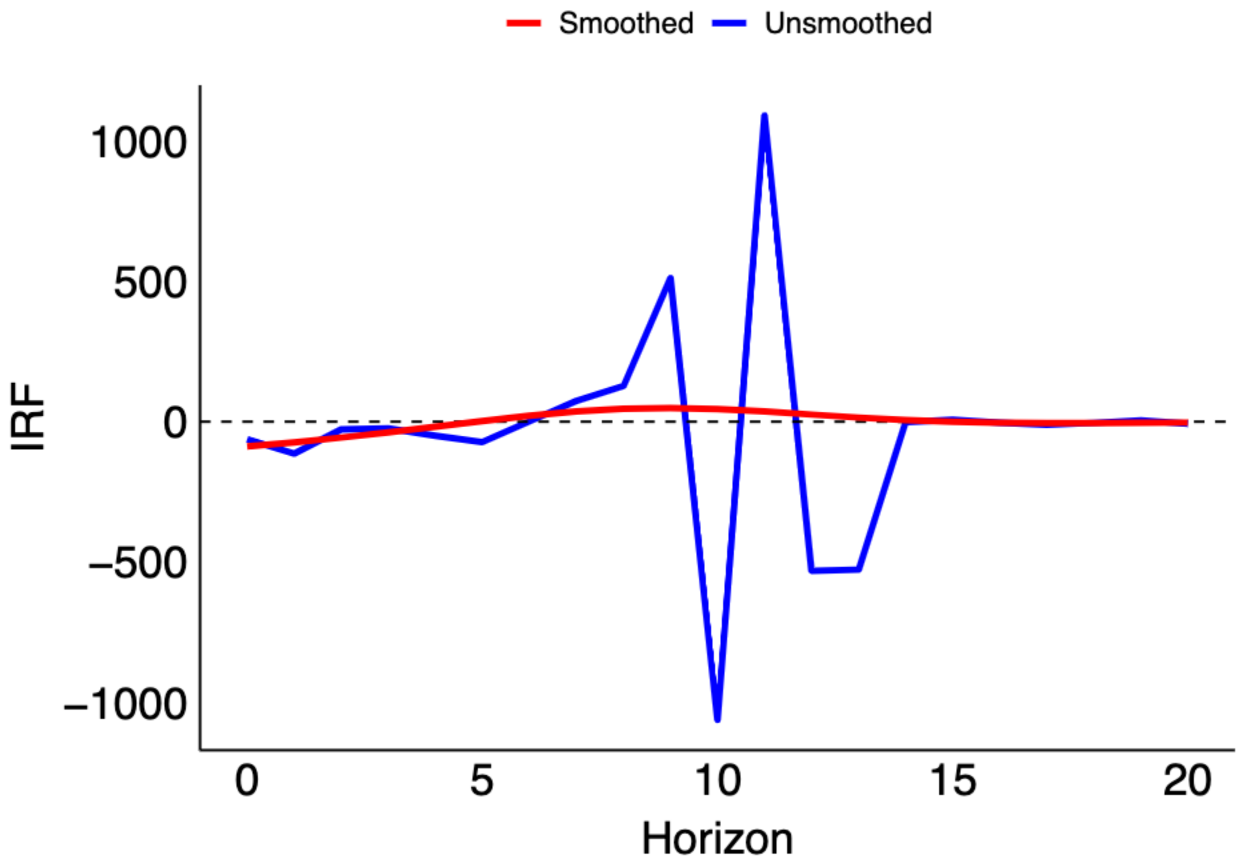


In this case, it's really not very different, but that's application-specific. We could also use instrumental variables. This isn't very theoretically sensible, but let's say we wanted to estimate the effect of wildcat drilling on the current account using the oil shock as an instrument while controlling for the exchange rate.

```
reg_formula3 <- f(ca,0:20)   ~ lreer_ins | ifscode + year | lnwildcat ~ sizerealistic

slpp_obj_iv <- feols_slpp(reg_formula3, r = 2,
                 paneldf = ars, type = "smooth", targetvar = "lnwildcat",boots = 0)
```

```
slpp_obj_iv$slpp$order_2$comp_plot
```



Super ugly, but the point is that you can do it. Next, we often want cumulative IRFs of the form

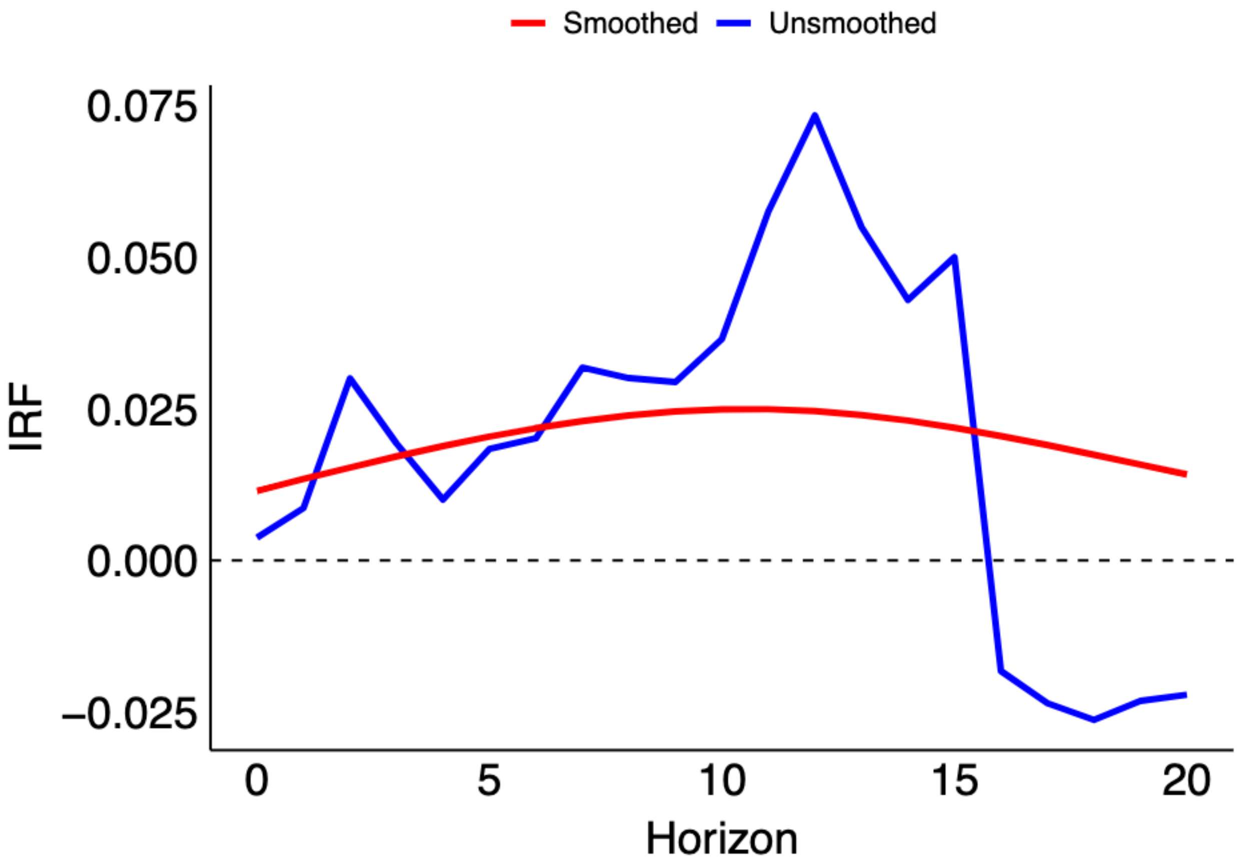
$$y_{i,t+h} - y_{i,t-1} = \alpha_i + T_t + \beta_h \times \text{Shock}_{i,t} + \varepsilon_{i,t+h}.$$

This is easy to do. Let's do it for consumption

```
reg_formula4 <- f(lnclcu,0:20) -l(lnclcu)  ~ sizerealistic | ifscode + year

slpp_obj_cum <- feols_slpp(reg_formula4, r = 1,
                 paneldf = ars, type = "smooth", targetvar = "sizerealistic",boots = 0)
```

```
slpp_obj_cum$slpp$order_1$comp_plot
```



Finally, note that the function is very fast at calling IRFs alone by simply setting boots = 0. It's also possible to manually set the penalty parameter.