nausea: Nowcasting Under Structural Breaks

```
library("nausea")
library("nowcasting")
```

Getting data

`get_data` will download current vintage data from the NYFED and FREDMD databases up to a given date, defaulting to June 2021. These are subjected to stationarity transforms.

```
data_vignette <- get_data()
```

```
## NULL
```

```
## Joining, by = "date"
```

```
## Warning in nowcasting::Bpanel(base = fredmd_ts, trans = rep(0, ncol(fredmd_4)
## - : 3 series ruled out due to lack in observations (more than 75 % NA).
```

```
## [1] "PERMITW"       "BAAFFM"        "CES3000000008"
## NULL
```

By default, we include a version up to May 2021.

```
data(panel)
```

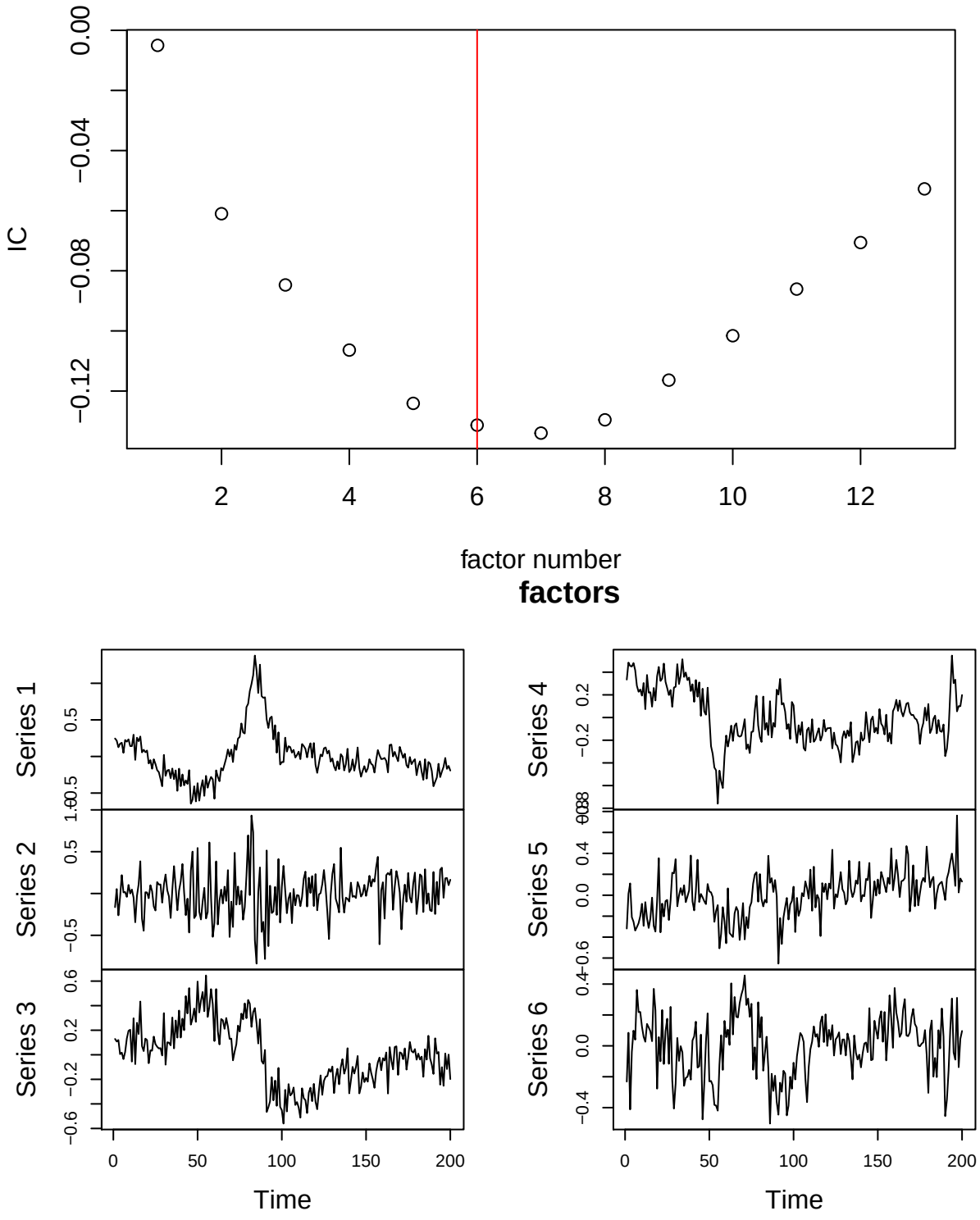
Factor modelling

`factor_model` will fit a factor model, selecting the factor number using information criterion #2 from Bai and Ng (2002). An object of class `factor_model` is returned, which has `summary`, `plot` and `predict` methods.

```
factor_model_out <- factor_model(panel$panel )
r_ <- factor_model_out$q.hat
factors <- factor_model_out$f.q#[,1:r_]
summary(factor_model_out)
```

```
## Information criterion:  -0.005012542 -0.06101498 -0.0847118 -0.1063758 -0.124104 -0.1313426 -0.13399
## Factor number:  6
## Maximum factor number:  12
## Variance explained:  0.3826887
```

```
plot(factor_model_out)
```



Change point analysis with `mosumvar`

Based on the `mosumvar` package and methodology, `mosumvar_factor` will perform change point analysis on the panel assuming a static factor model with low-dimensional vector autoregression (VAR) dynamics on the factors. The order `p` is recommended to be 1. The bandwidth `G` needs to be large enough for estimation purposes, but small enough to capture the quickly changing structure of the data, which is a difficult problem

with monthly data. We recommend `G=18` or `24`. For similar reasons, we recommend the `Score` method. `nu` quantifies how close successive change points are allowed to be declared in the estimation phase; `nu=.1` is a suitably flexible choice.

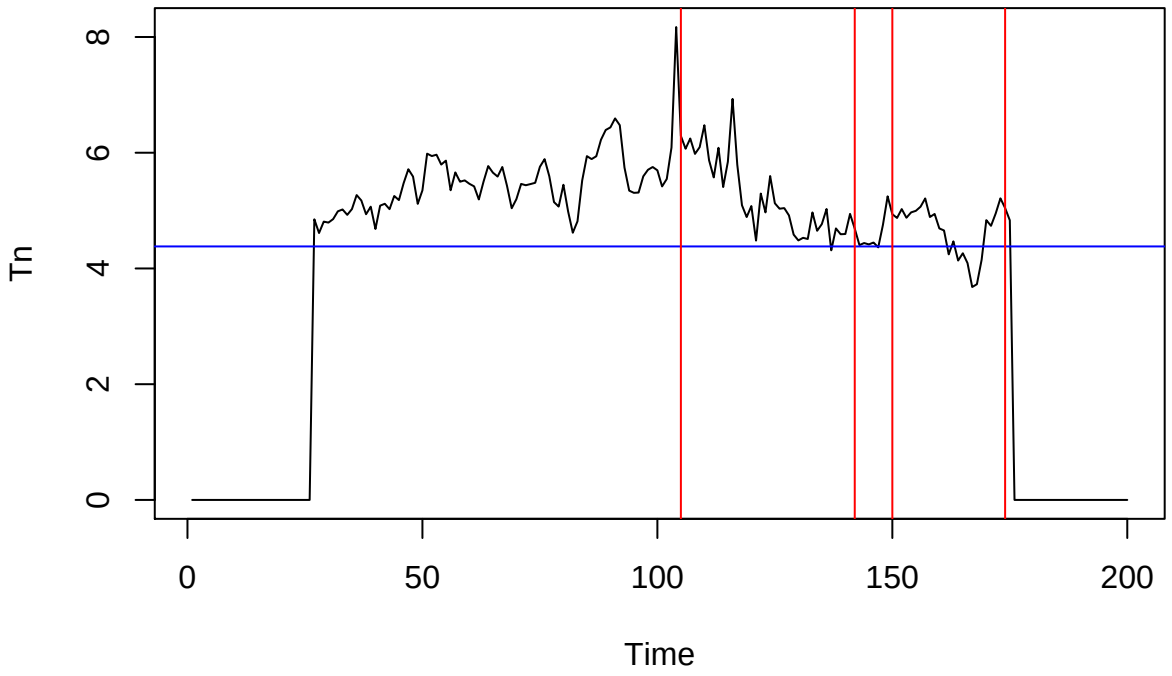
```
mosumvar_out <- mosumvar_factor(panel$panel, p = 1, G= 24, nu = .1, method = "Score")
```

```
## Warning in dim_warning(n, G, d, p, "Score"): Not enough degrees of freedom for Score method: set G >
```

```
## Warning in dim_warning(n, G, d, p, "Score"): Bandwidth too small relative to model dimensions: set G
```

```
## Warning in dim_warning(n, G, d, p, "Score"): Bandwidth small relative to sample size: consider setti
```

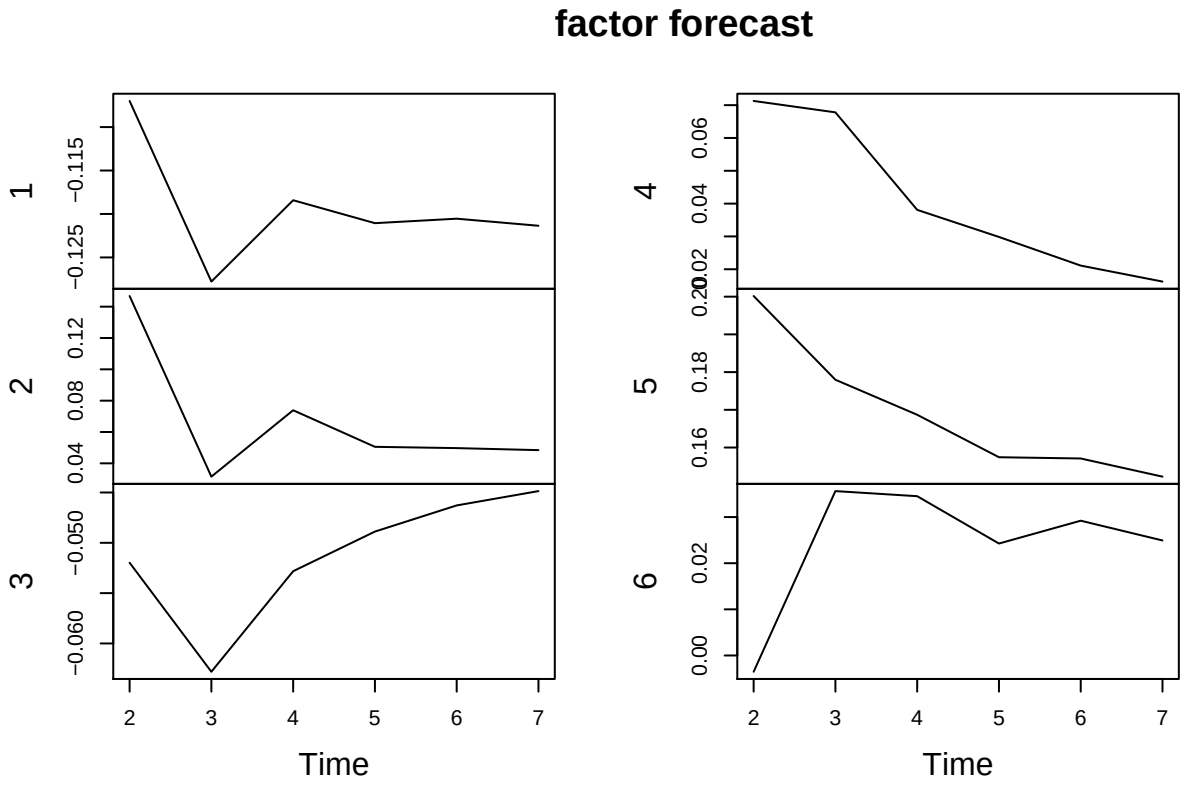
```
## Warning in dim_warning(n, G, d, p, "Score"): Large dimensions: consider `option = univariate`
```



Pooled forecasting

`pooled_forecast` will forecast the factor series using the most recent estimated change point, using either `robust`,`exp`, or `equal` weights on the models fit prior to the change. `predict` will predict for the panel.

```
pooled_forecast_out <- pooled_forecast(factors, cp = mosumvar_out$cps, p =1, window_size = 50, weights =
plot(pooled_forecast_out, main = "factor forecast")
```




```
panel_forecast <- predict(factor_model_out, pooled_forecast_out)
```

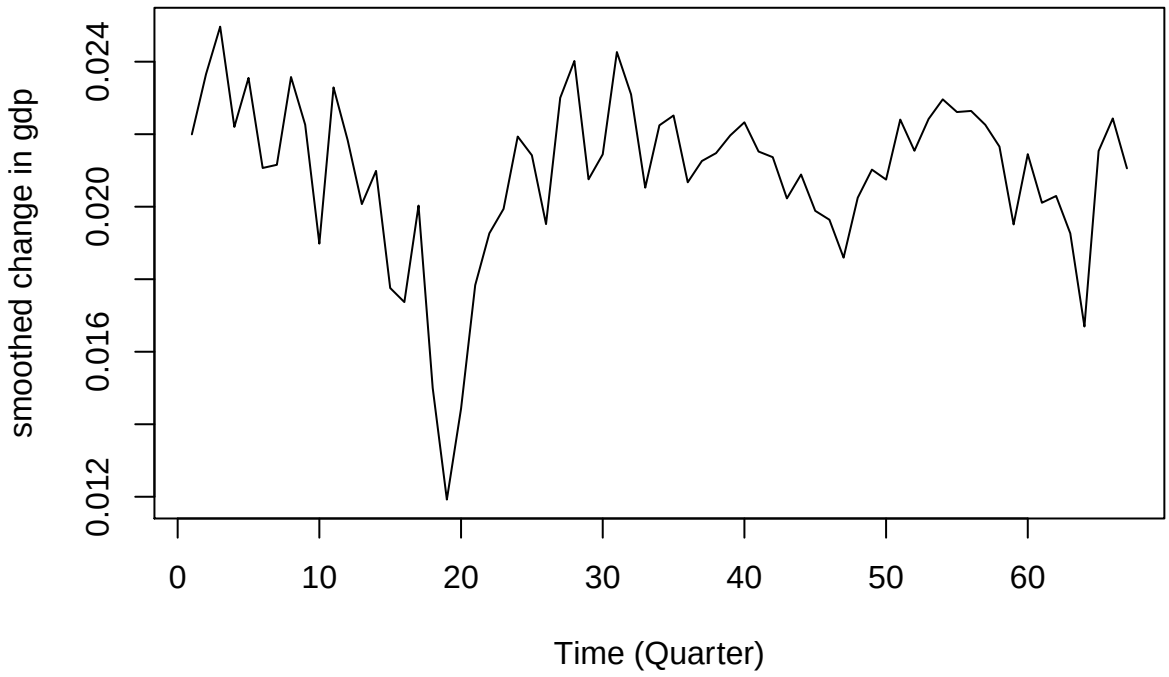
Nowcasting

Nowcasting of the GDP component can be performed with e.g. a linear regression model via `lm`, which permits use of the `predict` method.

```
lm_data <- data.frame(gdp = panel$gdp, factors)
gdp_lm <- lm(gdp ~ ., data = lm_data)
gdp_lm
```

```
##
## Call:
## lm(formula = gdp ~ ., data = lm_data)
##
## Coefficients:
## (Intercept)           X1           X2           X3           X4           X5
##    0.020871     0.002930     0.000818    -0.003607     0.004946    -0.001376
##          X6
##    0.004695
```

```
plot.ts(gdp_lm$fitted.values, xlab = "Time (Quarter)", ylab = "smoothed change in gdp")
```




```
predict(gdp_lm, newdata = data.frame(gdp = NA, pooled_forecast_out))
```

```
##          1          2          3          4          5          6
## 0.02092588 0.02100691 0.02089365 0.02077910 0.02075106 0.02070493
```