# Indicator performance validation with INDperform

Time [e.g. YEAR]

Indicator time series [IND]

Pressure time series [PRESS]


## Modelling IND trends

Each IND is modelled as a function of time using GAMs (using *mgcv*::gam).

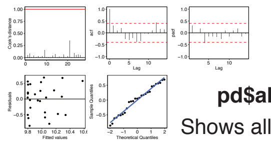
**m_trend <- model_trend(ind_tbl = IND, time = YEAR)**

Creates a tibble incl. GAM outputs, the model object, and predicted time series per IND.

### Model diagnostics

**pd <- plot_diagnostics(m_trend$model)**

Creates a tibble with individual and combined plots (ggplot2 objects).



**pd$all_plots[[1]]**

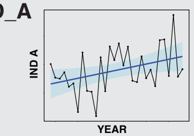
Shows all 5 plots (of 1st IND.

### Trend visualization

**pt <- plot_trend(m_trend)**

Creates a list with all IND trends from input tibble (ggplot2 objects).

**pt$IND_A**




The plot can be modified with additional ggplot2 themes.

## Modelling IND responses to pressures based on Generalized Additive Models (GAMs)

### 1. Initialization

All 3 data objects are combined into one tibble with defined training and test observations (as default 10% of last observations are kept as test data). All IND are combined with all PRESS provided as input.

**dat_init <- ind_init(ind_tbl = IND, press_tbl = PRESS, time = YEAR)**

Creates a tibble (IND x PRESS rows) including all training and test data.

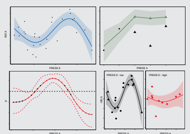
### Visualization of model results

**p <- plot_model(init_tbl = dat_init, mod_tbl = m_press)**

Creates a tibble with 4 individual and 1 combined plot (ggplot2).

#### View plots

**p$all_plots[[1]]**



Alternatively: $response_plot, $predict_plot, $deriv_plot, $thresh_plot

Generate 1 plot for multiple models:

***gridExtra*::grid.arrange(nrow=2, grobs=dat$response_plot[1:4] )**

#### Save plots

**ml <- *gridExtra*::grid.arrange( grobs = p$all_plots)**

***ggplot2*::ggsave("File.pdf", ml)**

Creates >1 page for many models:

**ml <- *gridExtra*::marrangeGrobs( grobs=p$response_plot, now=2, ncol=3)**

### 2a. Simple GAMs

A simple GAM is applied to all IND~PRESS combinations using the training observations.

**m_gam <- model_gam(init_tbl = dat_init)**

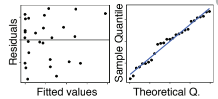
Creates a tibble incl. model outputs, diagnostics and model objects.

#### Model diagnostics

**plot_diagnostics(model_list = m_gam$model)**

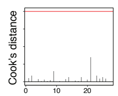
| | |
|---|---|
| **m_gam$ks_test** | P-value from Kolmogorov-Smirnoff normality test. |
| **m_gam$pres_outlier** | Index of identified outlier. |
| **m_gam$tac** | Logical vector whether temporal autocorrelation (**TAC**) is significant. |

#### Solutions



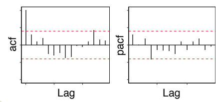

**Heterogeneity or non-normality?**

Transform original data OR change distribution: **model_gam(..., family = poisson)**




**Outlier?**

Exclude outlier: **model_gam(..., excl_outlier = m_gam$pres_outlier)**




**TAC?**

Apply simple GAMMs (step 2b)

### 2b. Simple GAMMs

Inclusion of AR and ARMA correlation structures (using *mgcv*::gamm) for all IND~PRESS GAMs or those with sign. TAC (using the filter argument).

**m_gamm <- model_gamm(init_tbl = dat_init, filter = m_gam$tac, excl_outlier = NULL)**

Creates a tibble with 6 GAMMs for each selected IND~PRESS.

#### Model diagnostics

**plot_diagnostic(model_list = m_gamm$model)**

**m_gamm$ks_test**

**m_gamm$pres_outlier**

**m_gamm$tac**

#### Selection of best corr structure

Manual or automatic selection based on the AIC using the select_model() function:

**best_gamm <- select_model(gam_tbl = m_gam, gamm_tbl = m_gamm)**

### 2c. Model merging

**m_press <- merge_models( m_gam[m_gam$tac == FALSE, ], best_gamm)**

Merges any 2 model output tibbles (can also be 2 GAM tibbles).

### 3. Derivatives of non-linear responses

Calculates for sign. GAM(M)s with edf > 1.5 (default setting) the 1st derivative of the smoothing function. The proportion of pressure range in which the IND response is significant is determined by a conditional bootstrap.

**m_press <- calc_deriv(init_tbl = dat_init, mod_tbl = m_press)**

Output is input tibble with few additional variables, incl. the mean and confidence interval of smoothing functions and derivatives from bootstrapped GAMs:

| | |
|---|---|
| **m_press$prop** | Proportion of effective pressure range, i.e. where slope of function is sign. different from zero - needed for subcriterion 10.2. |

### 4. Pressure interactions

For each significant GAM(M) a selection of pressure variables is used to test whether these modify the IND response to the original pressure variable using a threshold-GAM formulation.

**it <- select_interaction(mod_tbl = m_press)**

**m_press <- test_interaction(init_tbl = dat_init, mod_tbl = m_press, interactions = it)**

Output is input tibble with few additional variables:

| | |
|---|---|
| **$interaction** | Logical vector indicating whether any threshold-GAM was better than the GAM(M) |
| **$thresh_model** | List of better performing threshold-GAMs |

#### Model diagnostics

**plot_diagnostics(m_press$thresh_model)**


This cheat sheet presents the INDperform package in version 0.0.0.9000 that is based on the framework described in Otto et al. 2018, *Ecol. Ind.*, 84: 619-631)
See https://github.com/saskiaotto/INDperform/ for more details.

Updated 10/2017

# Score-based IND performance

Output from trend models

Output from pressure models

## Scoring based on model outputs

| press | press_type |
|---|---|
| | |
| | |
| | |

Additional table provided by the user that lists the pressure type of each pressure needed for criterion C11 (link to management) and the visualisation.

+

| crit | subcrit | score | weight | condition | condition_var |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |

The package provides the full criteria template ("**crit_scores_tmpl**") described in the underlying framework (Otto et al., 2018, Ecol. Ind.), which is set as default in the scoring() function. It contains the scores and weights for each (sub-)criterion, the variables from the model output tibbles on which each(sub)criterion is based on as well as the condition to determine the actual score. The user can modify the weights, scores, conditions or remove specific (sub)crits.

```
scores <- scoring(trend_tbl = m_trend, mod_tbl = m_press, press_type)
```

Remove single criteria, e.g.trend criterion C8 (no trend model output needed anymore)

```
scores_noC8 <- scoring(m_press, press_type,
                  crit_scores = crit_scores_tmpl[crit_scores_tmpl$crit_id > 1, ])
```

```
scores <- expect_resp(scores)
```

Runs a shiny app to modify the score for the subcriterion 10.1 (IND response as expected) based on the response curves (default score 1 for neutral / no expectation).

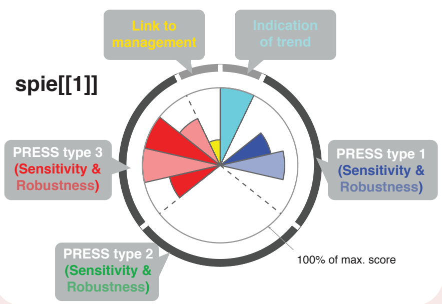
```
sum_sc <- summary_sc(scores)
```

Provides a user-friendly summary of the scoring output tibble.

## Score visualization

```
spie <- plot_spiechart(sum_sc)
```

spie[[1]]




## Cluster analysis

```
scores_dist <- dist_sc(scores)
```

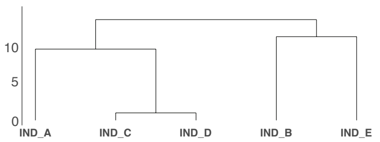
Calculates a (Euclidean) distance matrix based on all scores.

```
scores_clust <- clust_sc(
                  scores_dist )
```

Returns a hclust object and prints the Gower distance and Cophonetic correlation coefficient.

```
plot_clust_sc(scores_clust)
```

ggplot2 that can be modified with additional themes.




Selection of best performing and complementary IND suite

# State assessment based on IND suite

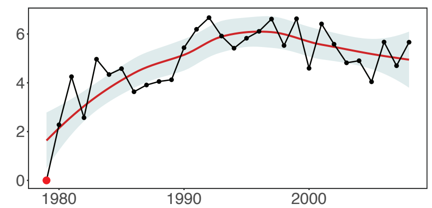
Two approaches based on trajectories in state space are implemented in the package to determine the current state of the system in comparison to an earlier reference period using the selected IND suite (state space = n-dimensional space of possible locations of IND variables).

## Euclidean distance in state space

Calculation of the Euclidean distance in state space of any dimensionality between each single year and a defined reference year.

```
ed <- statespace_ed(x = IND_sub, time = YEAR, ref_time = YEAR[1] )

plot_statespace_ed(ed)
```




## Convex hull of state space

Given the identification of a reference domain in state space, more recent observations might lie within or outside this domain. The convex hull is a multivariate measure derived from computational geometry representing the smallest convex set containing all the reference points in Euclidean plane or space. For visualization, only 2 dimensions considered (dimen-sion reduction through e.g. Principal Component Analysis suggested).

```
# State space of 2 INDs
ch <- statespace_ch(x = IND$A, y = IND$B, period_ref = 1979:1983,
                    period_current = 2004:2008)

# State space of first 2 principal components
pca <- vegan::rda(IND_sub, scale = TRUE)
pcas <- vegan::scores(pca_out, scaling = 0)
ch <- statespace_ch(x = pcas$scores[,1], y = pcas$scores[,2],
                    period_ref = 1979:1983, period_current = 2004:2008)

plot_statespace_ch(ch)
```