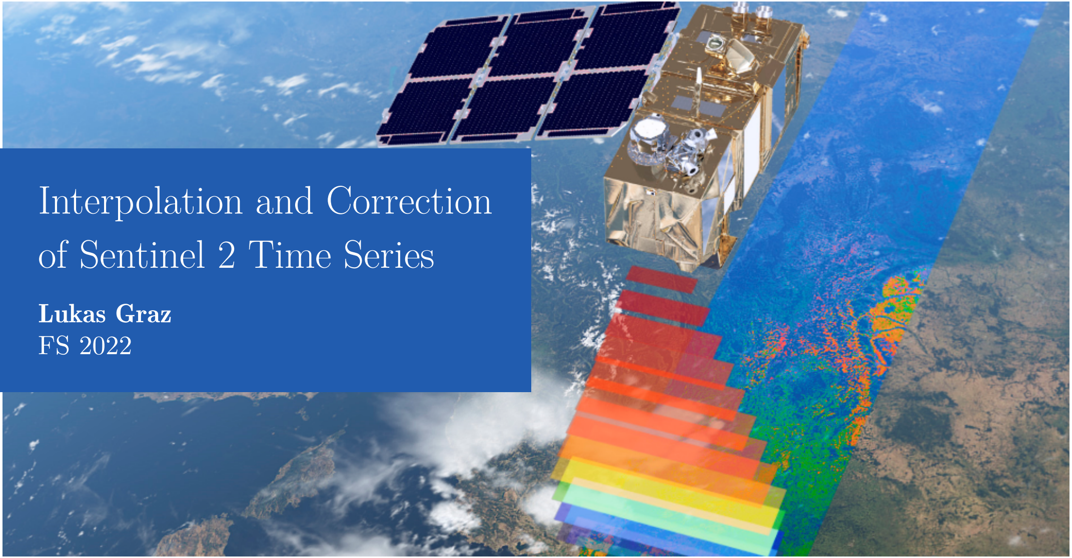
D-MATH — Seminar for Statistics
D-USYS — Crop Science

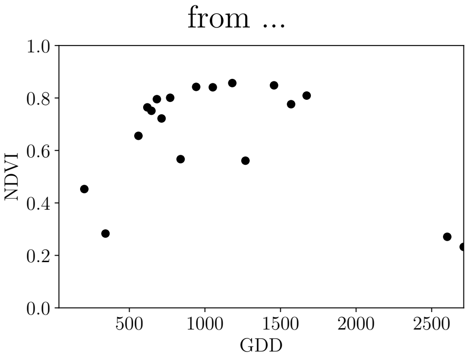


Interpolation and Correction of Sentinel 2 Time Series

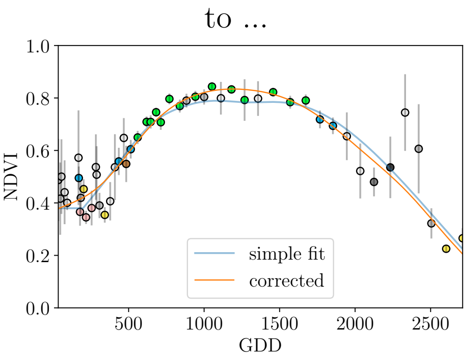
Lukas Graz

FS 2022

How to get ...



from ...
NDVI
GDD
1.0
0.8
0.6
0.4
0.2
0.0
500
1000
1500
2000
2500

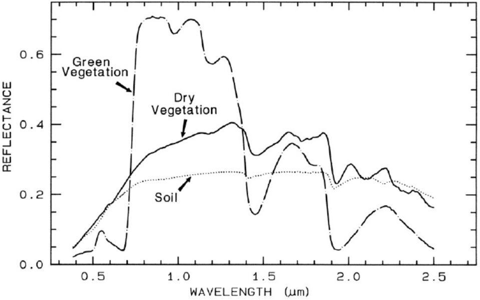


to ...
NDVI
GDD
simple fit
corrected
1.0
0.8
0.6
0.4
0.2
0.0
500
1000
1500
2000
2500

Scene Classification Layer (SCL)

Color	No.	Class
	0:	Missing Data
	1:	Saturated or defective pixel
	2:	Dark features / Shadows
	3:	Cloud shadows
	4:	Vegetation
	5:	Bare soils
	6:	Water
	7:	Cloud low probability
	8:	Cloud medium probability
	9:	Cloud high probability
	10:	Thin cirrus cloud
	11:	Snow or ice

NDVI From S2 Images



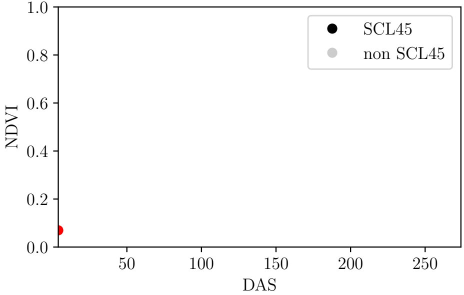
Normalized **D**ifference **V**egetation **I**ndex:

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

[0] Spectral Reflectances https://www.researchgate.net/figure/Reflectance-spectra-of-photosynthetic-green-vegetation-non-photosynthetic-dry__fig4__236677371

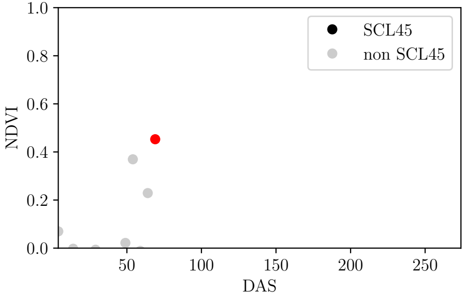
Sentinel 2 Image + NDVI Time Series

→ Cloud high probability



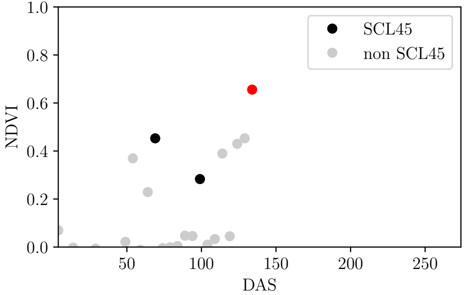
Sentinel 2 Image + NDVI Time Series

→ Bare soils



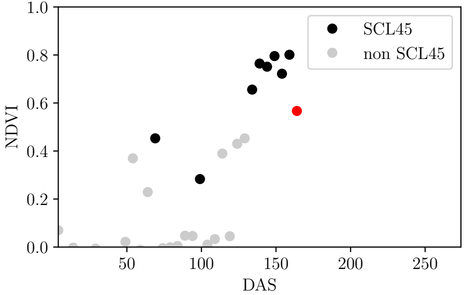
Sentinel 2 Image + NDVI Time Series

→ Vegetation



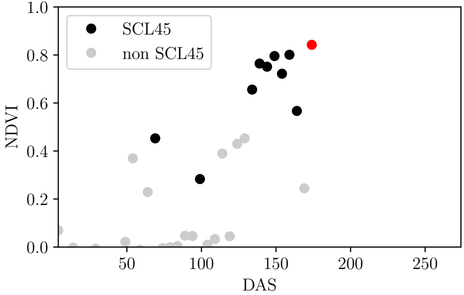
Sentinel 2 Image + NDVI Time Series

→ Vegetation

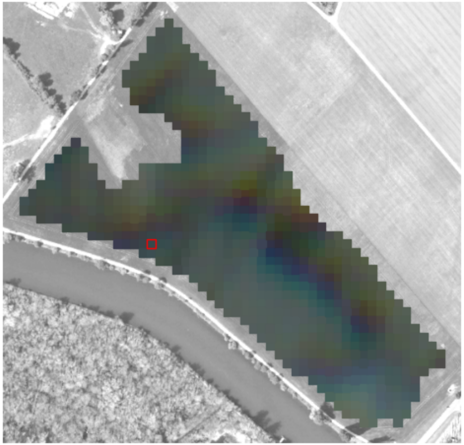


Sentinel 2 Image + NDVI Time Series

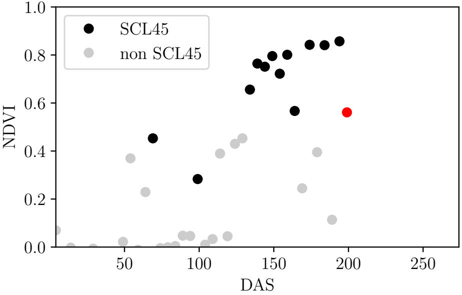
→ Vegetation



Sentinel 2 Image + NDVI Time Series

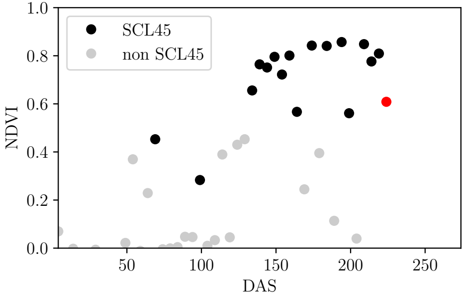


→ Vegetation



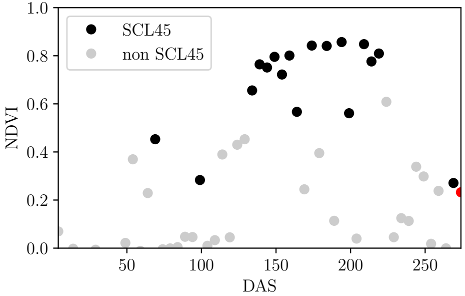
Sentinel 2 Image + NDVI Time Series

→ Thin cirrus cloud



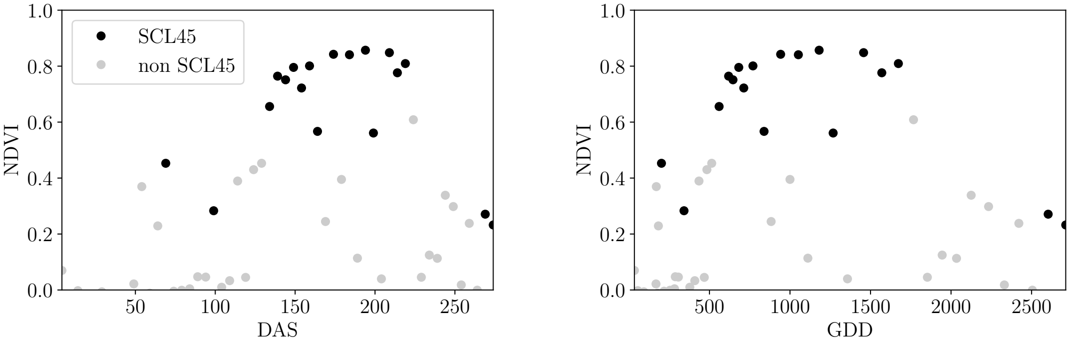
Sentinel 2 Image + NDVI Time Series

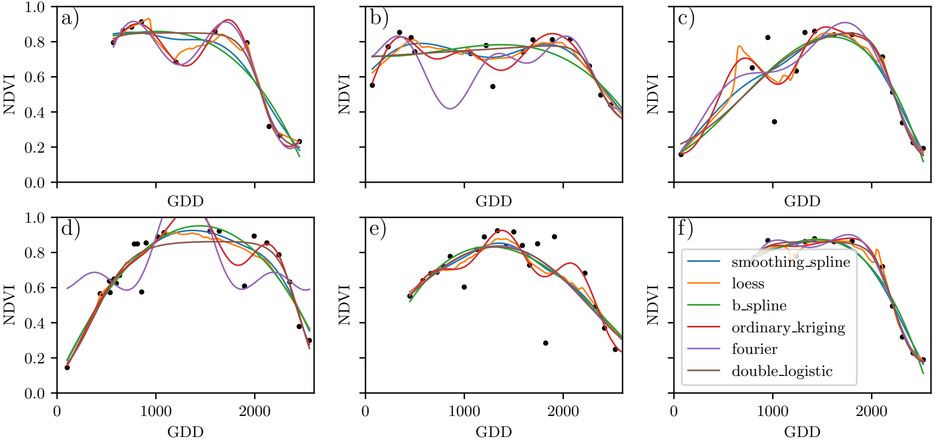
→ Bare soils



Timescale Transformation: DAS vs GDD

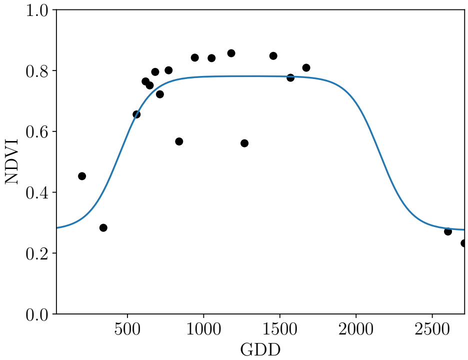
$$GDD_n := \sum_{i=0}^{n} \max(T_i - T_{base}, 0).$$




a)
b)
c)
d)
e)
f)
NDVI
GDD
1.0
0.8
0.6
0.4
0.2
0.0
0
1000
2000
smoothing_spline
loess
b_spline
ordinary_kriging
fourier
double_logistic

Double Logistic Approximation

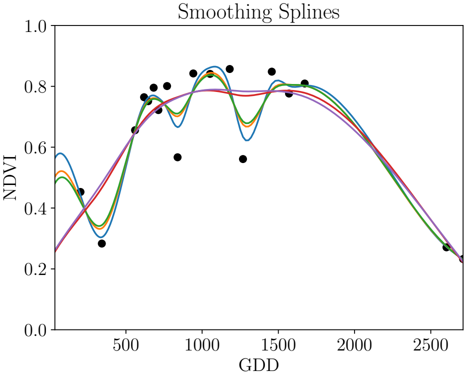
$$\mathrm{NDVI}(t) = y_{\min} + (y_{\max} - y_{\min}) \left(\frac{1}{1 + e^{-d_0(t-t_0)}} + \frac{1}{1 + e^{-d_1(t-t_1)}} - 1 \right)$$



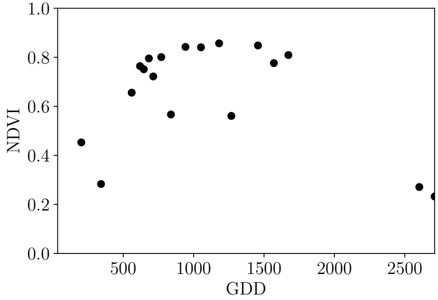
Smoothing Splines

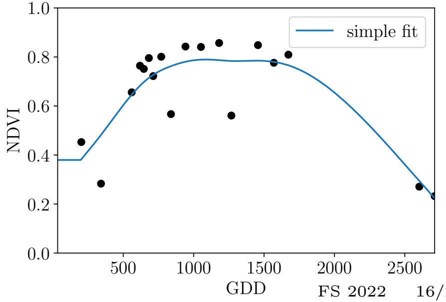
$$\widehat{NDVI} := \underset{f \in \mathcal{F}}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n} (Y_i - f(x_i))^2}_{\text{sum of squares}} + \lambda \underbrace{\int f''(x)^2 dx}_{\text{smoothness}}$$

Similar to the Whittaker (but more general)



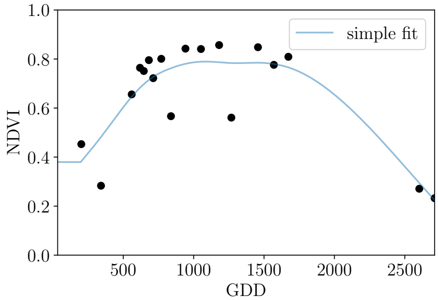
1. Interpolation

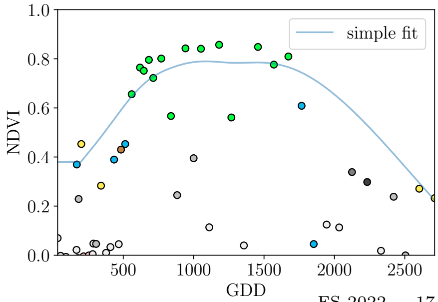




2. Other SCL-Classes

Color	No.	Class
	0:	Missing Data
	1:	Saturated or defective pixel
	2:	Dark features / Shadows
	3:	Cloud shadows
	4:	Vegetation
	5:	Bare soils
	6:	Water
	7:	Cloud low probability
	8:	Cloud medium probability
	9:	Cloud high probability
	10:	Thin cirrus cloud
	11:	Snow or ice

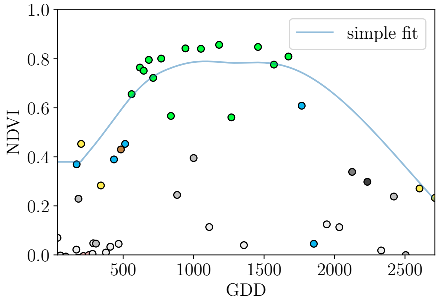


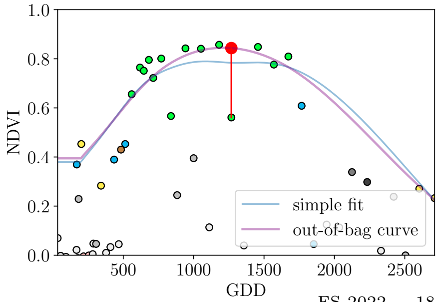


3. Correction

- get “true” NDVI
- get table:

“truth”	observed	scl-class	B2-B10
“truth”	observed	scl-class	B2-B10
...	...	...	...



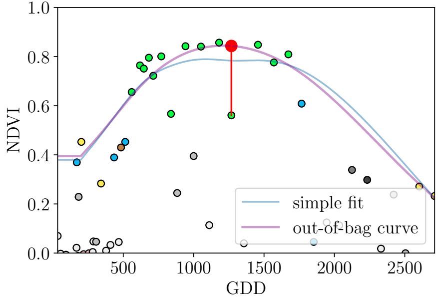


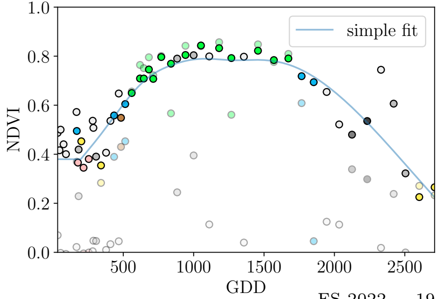
3. Correction

- get "true" NDVI
- get table:

"truth"	observed	scl-class	B2-B10
"truth"	observed	scl-class	B2-B10
...	...	...	...

- Statistical model
- predict/correct NDVI



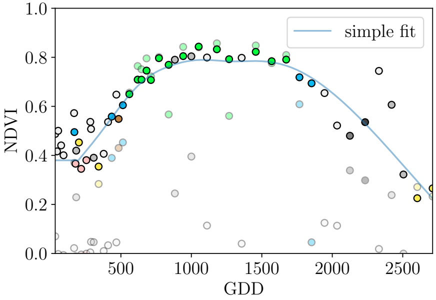


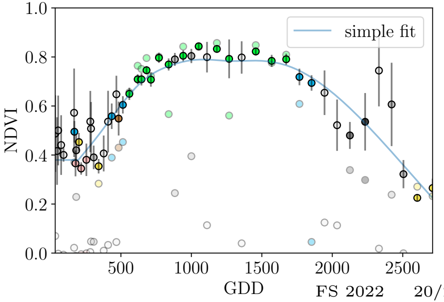
4. Uncertainty Estimation

- Table with residuals:

residuals	observed	scl-class	B2-B10
residuals	observed	scl-class	B2-B10
⋮	⋮	⋮	⋮

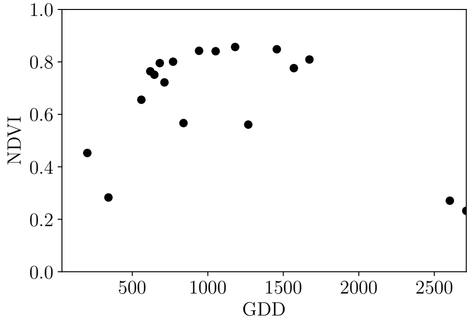
- Statistical model
- predict residuals
- $weights = \frac{1}{|residual|}$

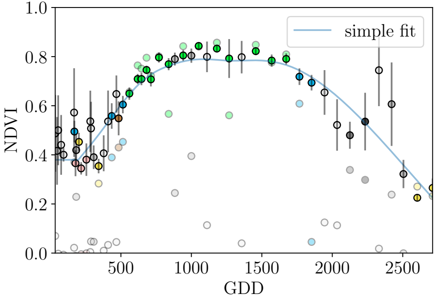


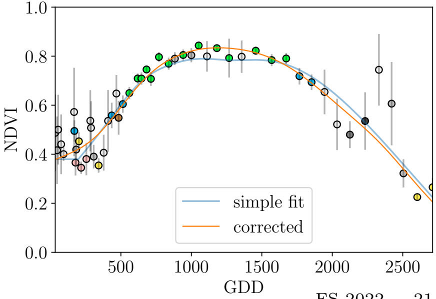


5. Robust Fit to Corrected NDVI

Reminder: Original Situation







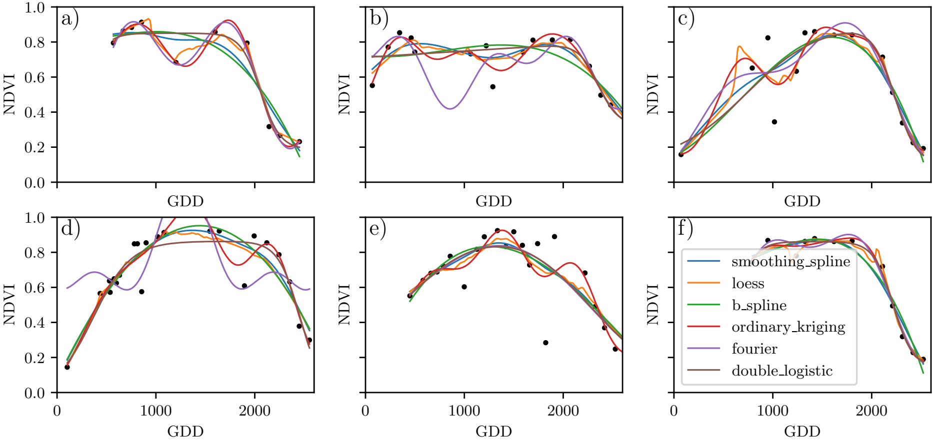
R-Package Provided

```
library(CorrectTimeSeries)
data(timeseries_list) # load NDVI-TS data

# Train RF
  # Add "true" NDVI (or generally the response), by Out-Of-Bag estimation
    timeseries_list <- lapply(timeseries_list, function(df) {
      df$oob_ndvi <- OOB_est(df$gdd, df$ndvi_observed) # gdd is the time-axis
      df})
  # Train correction model
    formula <- "oob_ndvi ~ B02+B03+B04+B05+B06+B07+B08+B8A+B11+B12+scl_class"
    RF <- train_RF_with_fromula(formula, timeseries_list, robustify=TRUE)
# ADD CORRECTION
  timeseries_list <- lapply(timeseries_list, function(df) {
    df$corrected_ndvi <- randomForest:::predict.randomForest(RF, df)
    df})

# Get interpolation for each timeseries
  lapply(timeseries_list, function(df){
    ss <- smoothing_spline(df$gdd, df$corrected_ndvi)
    predict(ss, 1:1000)$y})
```

Yield Mapping Data

Thank You !

Robustifying



Results

	RF	OLS-SCL	OLS-all	MARS	GAM	LASSO	no correction
SS	0.676	0.736	0.726	0.731	0.729	0.731	0.703
SS^{rob}	0.676	0.724	0.709	0.704	0.715	0.716	0.705
DL	0.672	0.692	0.691	0.691	0.702	0.701	0.667
DL^{rob}	0.667	0.685	0.690	0.719	0.704	0.697	0.669

APPENDIX: OLS$^{\text{SCL}}$

$$
\begin{aligned}
\text{NDVI}_{\text{corr}} = \quad & 0.711\,\text{NDVI}_{\text{observed}} + 0.215\,\mathbb{1}_{SCL=2} + 0.237\,\mathbb{1}_{SCL=3} + 0.210\,\mathbb{1}_{SCL=4} \\
& + 0.116\,\mathbb{1}_{SCL=5} + 0.162\,\mathbb{1}_{SCL=6} + 0.327\,\mathbb{1}_{SCL=7} + 0.474\,\mathbb{1}_{SCL=8} \\
& + 0.575\,\mathbb{1}_{SCL=9} + 0.306\,\mathbb{1}_{SCL=10} + 0.512\,\mathbb{1}_{SCL=11}
\end{aligned}
$$

$$
\begin{aligned}
\widehat{\text{abs}}\,(\text{NDVI}_{\text{`true'}} - \text{NDVI}_{\text{corr}}) = \quad & -0.133\,\text{NDVI}_{\text{observed}} + 0.186\,\mathbb{1}_{SCL=2} + 0.185\,\mathbb{1}_{SCL=3} \\
& + 0.146\,\mathbb{1}_{SCL=4} + 0.089\,\mathbb{1}_{SCL=5} + 0.167\,\mathbb{1}_{SCL=6} \\
& + 0.203\,\mathbb{1}_{SCL=7} + 0.181\,\mathbb{1}_{SCL=8} + 0.173\,\mathbb{1}_{SCL=9} \\
& + 0.180\,\mathbb{1}_{SCL=10} + 0.172\,\mathbb{1}_{SCL=11}
\end{aligned}
$$