Supervised Macroeconomic Index for IFRS9 Forward-Looking Modeling

The R Package smi

Andrija Djurovic

Contents

1 IFRS9 Forward-Looking Modeling in Practice

IFRS9 requires that estimates of expected credit losses incorporate future reasonable and supportable information that is available without undue cost or effort. In practice, this often involves evaluating how the macroeconomic environment affects risk parameters, typically at the segment level - for example, by regressing default rates against macroeconomic indicators. Such analyses are generally referred to as forward-looking (FLI) exercises.

Most FLI implementations rely on Ordinary Least Squares (OLS) regression, with common approaches including:

- OLS regression of the target variable on macroeconomic indicators (with or without time lags);
- OLS regression that includes macroeconomic indicators and an autoregressive term of the target;
- Two-step error correction models;
- Principal Component Analysis (PCA) combined with OLS regression;
- OLS regression with various transformations of the target variable, such as logit or probit.

While this list is not exhaustive, it underscores a key question: even when multiple methods and design choices are combined, do these approaches truly capture all critical business inputs?

2 IFRS9 Forward-Looking Modeling Challenges

The limited number of observations typically available is the primary constraint in FLI exercises, giving rise to many of the challenges practitioners face. Some of the most common challenges resulting from this limitation include:

1. *Low ratio between the number of observations and available predictors*: put simply, practitioners often face a situation where there are too many potential predictors for too few observations. As a result, it is usually not statistically sound to include more than three predictors in a single model. Although this can vary in practice, typically no more than three predictors are used.

2. *Incorporation of business-guided macroeconomic indicators in the FLI model*: one of the key inputs from the macroeconomic department and business experts is often a list of potentially important predictors for the target variable. In addition to suggesting specific macroeconomic indicators, they typically provide further guidance - such as the expected relationship with the target variable, the impact of an indicator at a specific time lag, or recommendations to group certain indicators that should not appear in the same model. However, the limited number of observations, combined with other statistical constraints, presents a significant challenge for practitioners trying to incorporate these business inputs effectively.

3. *Asymmetric or opposing impacts of macroeconomic indicators under different economic conditions or when specific thresholds are reached*: in practice, practitioners often discuss this challenge alongside the previous two. When certain macroeconomic indicators are selected for this type of testing, additional analysis is typically required to determine the optimal thresholds for defining economic regimes, as well as to create new derived indicators and incorporate them into the model design. Consequently, this challenge can also be seen as stemming from the limited number of available observations, as practitioners - even without deriving new predictors - often face a relatively low ratio between the number of observations and predictors.

4. *Balancing the impact of macroeconomic indicators and other predictors*: often, the development of the target variable cannot be fully explained by macroeconomic indicators alone. However, the typical first step in FLI modeling is to explain as much of the target variable's variability as possible through the macroeconomic environment, then incorporate other important events usually not captured by these indicators. The limited number of observations poses a significant challenge in combining these inputs while controlling for the expected signs of the estimated coefficients and each variable's contribution to the model's predictions.

5. *Ensuring model stability and reliable forecasts*: when practitioners choose an FLI model that relies on only one or two macroeconomic indicators, such models are often viewed by validators and regulators as overly sensitive to those specific indicators and therefore not well-suited for the FLI exercise. This limited specification is also seen as insufficient to capture the broader and more complex influence of the macroeconomic environment. As with the previous challenge, this issue often stems from the limited number of available observations. Incorporating a larger set of macroeconomic predictors typically introduces difficulties in meeting both business expectations and fundamental statistical requirements for model robustness.

6. *Meeting statistical criteria for model selection*: in balancing business expectations with statistical rigor, practitioners typically rely on a set of statistical metrics or tests to guide the selection of the FLI model. These criteria are often sensitive to sample size and, in particular, to the ratio between the number of observations and the number of predictors. When sample sizes are small, including too many predictors can undermine model reliability. Consequently, satisfying statistical requirements is generally more feasible when the model is built on a higher observation-to-predictor ratio.

3 Supervised Macroeconomic Index

In response to the major challenges practitioners face - many of which stem from a limited number of observations - Andrija Djurovic proposes the Supervised Macroeconomic Index (SMI) approach.

The SMI enables practitioners to incorporate critical inputs into the FLI modeling process, such as the overall impact of the macroeconomic environment through model averaging, while maintaining control over the expected relationship between macroeconomic indicators and the modeled target. Additionally, it allows for straightforward control of the contribution of specific macroeconomic indicators to the model predictor in response to correlations among predictors. The greater flexibility of the SMI helps practitioners address key limitations of another method commonly considered in FLI modeling - Principal Component Analysis (PCA).

The following steps provide a general framework for constructing and using the SMI for the purpose of the FLI exercise.

1. Define the set of candidate macroeconomic indicators that can be used as predictors for the FLI model.
2. For each individual macroeconomic indicator, define the maximum time lag to be considered in the modeling process.
3. For each individual macroeconomic indicator, define the expected relationship with the modeled target variable.
4. Based on steps 1 to 3, define groups of predictors to be combined for the model estimation process.
5. Define the maximum number of predictors to be included in the model.
6. Combine the groups of predictors from step 4, up to the maximum number defined in step 5, to generate potential regression models.
7. For each combination of predictors from step 6, estimate a constrained OLS regression of the target variable on the combined predictors. The constraints should, at a minimum, reflect the expected sign of coefficients based on the assumed relationship between macroeconomic indicators and the target variable.
8. Exclude all regression models in which at least one estimated coefficient for a macroeconomic indicator equals zero.
9. From the remaining regression candidates, construct the Supervised Macroeconomic Index (SMI) as a model average.
10. Use the SMI from step 9, either alone or in combination with additional predictors that capture significant developments affecting the target but not reflected in the macroeconomic environment, to run an integrated OLS regression of the target variable on the SMI and other predictors.
11. Evaluate model performance, address any violations of OLS assumptions, and fine-tune the model accordingly.

The above steps provide a general framework for constructing and using the SMI, but practitioners can adjust certain parts of the process. For instance, instead of running constrained OLS on all possible combinations from one to the maximum number of predictors, practitioners may choose to begin with bivariate analysis, selecting one or a few macroeconomic indicators per group. This approach can significantly reduce execution time, especially when there are many predictor candidates. Additionally, to account for correlations between macroeconomic indicators, practitioners can go beyond constraining only the expected signs by also controlling the range of coefficient values based on the results of the bivariate regressions. Furthermore, once the SMI is constructed, practitioners can test and implement various regression designs, including modeling in first differences, dynamic regressions, or the inclusion of lagged dependent variables.

Overall, the SMI offers a simple and straightforward way to address some of the key challenges practitioners currently face in the FLI exercise, while maintaining a fully explainable model design - comparable to the standard use of OLS regression for the same purpose.

4 The `R` Package `smi`

The R package `smi` provides a framework for implementing the Supervised Macroeconomic Index (SMI), which can be used for FLI modeling.

The development version is currently available on this GitHub page.

To install the smi package, run:

```
devtools::install_github(repo = "andrija-djurovic/adsfcr",
                         subdir = "model_dev_and_vld/smi_r",
                         upgrade = "never")
```

After installing the package, load it with:

```
library(smi)
```

The package includes the main functionalities needed to construct the SMI, following the steps described in the previous section. At its core, the package performs constrained OLS regressions of the form:

$$\begin{aligned} \min_{\beta} \quad & \sum_{i=1}^{T} (y_i - x_i^\top \beta)^2 \\ \text{s.t.} \quad & \beta_j \geq 0 \quad \text{for } j \in \mathcal{P}, \\ & \beta_k \leq 0 \quad \text{for } k \in \mathcal{N}. \end{aligned}$$

where:

- T is the number of observations.
- y_i is the dependent variable for observation i.
- x_i is the $n \times 1$ vector of independent variables for observation i.
- β is the $n \times 1$ vector of coefficients to be estimated.
- $\mathcal{P}$ is the set of indices of coefficients constrained to be non-negative (reflecting an expected positive relationship).
- $\mathcal{N}$ is the set of indices of coefficients constrained to be non-positive (reflecting an expected negative relationship).

The following case study explains how practitioners can use the smi package to construct the SMI. Since the examples shown represent just one possible approach, practitioners are encouraged to adapt both the process and the package's functionalities to include any additional inputs needed for their specific use case.

5 Case Study

5.1 Simulation Dataset: Import and Overview

This simulation study uses a dataset that can be imported by running:

```
url.1 <- "https://raw.githubusercontent.com/andrija-djurovic/adsfcr/"
url.2 <- "main/model_dev_and_vld/smi_r/db_smi_package.csv"
url <- paste0(url.1, url.2)
db <- read.csv(file = url,
               header = TRUE)
db$QUARTER <- as.Date(x = db$QUARTER,
                      format = "%d-%m-%y")
```

The dataset contains 70 observations across eight columns, covering the period from `Q1 2011` to `Q2 2025`. Its structure can be viewed with:

```
str(db)
```

```
## 'data.frame':    70 obs. of  8 variables:
##  $ QUARTER: Date, format: "2011-03-31" "2011-06-30" ...
##  $ TYPE   : chr  "REALIZED" "REALIZED" "REALIZED" "REALIZED" ...
##  $ ODR    : num  0.0921 0.093 0.0958 0.1032 0.1102 ...
##  $ UNEMP  : num  0.0711 0.076 0.0789 0.0792 0.0837 ...
```

```
##  $ GDP    : num  0.00979 -0.0039 -0.02549 -0.06164 -0.05392 ...
##  $ WAGE   : num  0.05633 0.05625 0.02039 0.00945 -0.00199 ...
##  $ EURIBOR: num  0.0638 0.0676 0.0579 0.0389 0.0349 ...
##  $ DUMMY  : int  0 0 0 0 0 0 0 0 0 0 ...
```

where:

- `QUARTER` denotes the date of the observations;
- `TYPE` indicates whether the observations represent realized or forecast values. Realized values apply to all variables, including the target, while forecasts are available only for predictors;
- `ODR` is the observed default rate, used as the target variable in this case study;
- `UNEMP` denotes the unemployment rate;
- `GDP` is the year-on-year GDP growth;
- `WAGE` is the year-over-year wage growth;
- `EURIBOR` is the 6-month Euribor value;
- `DUMMY` is a 0/1 indicator accounting for a significant change in the bank's internal credit risk process starting in Q1 2021. This indicator is not used for SMI estimation but is integrated with the SMI in the final model.

Practitioners should keep in mind that this dataset is simulated, although it is designed to be realistic. As such, both the predictors and the evolution of the target variable should be viewed primarily as part of a technical demonstration, rather than as a fully representative example of real-world application. Nevertheless, the dataset remains close to what practitioners might expect in an actual design and development setting.

5.2 Modeling Inputs

After importing the data, the next step is typically to define the various modeling inputs. In FLI exercises, this often involves specifying the set of macroeconomic indicators to be used as potential predictors, the expected direction of their relationship with the target variable, and any potential lags that might capture delayed effects on the target. These inputs can be configured using the following code.

```
#predictors
pn <- c("UNEMP", "GDP", "WAGE", "EURIBOR")
#expected sign
ps <- c("UNEMP"   = "+",
        "GDP"     = "-",
        "WAGE"    = "-",
        "EURIBOR" = "+")
#number of lags
pl <- c("UNEMP"   = 4,
        "GDP"     = 4,
        "WAGE"    = 4,
        "EURIBOR" = 4)
```

Since the SMI estimation process requires creating lagged predictors in advance, the following code utilizes the `lv` function from the `smi` package to generate these lags and append them to the `db` dataframe.

```
db.lv <- lv(db = db,
            x = pn,
            n = pl)
db <- cbind.data.frame(db, db.lv)
```

5.3 Predictor Groups and Model Combinations

The next step in constructing the SMI is to create groups of predictors. This is accomplished using the `pg` function from the `smi` package, which accepts as its sole input a vector of predictor lags and their corresponding names. This vector has already been defined and stored in the `R` object `pl`.

```
#pl vector
pl
```

```
##   UNEMP     GDP    WAGE EURIBOR
##       4       4       4       4
```

```
#predictor groups
groups <- pg(n = pl)
#print groups
groups
```

```
## $UNEMP
## [1] "UNEMP"      "UNEMP_lag1" "UNEMP_lag2" "UNEMP_lag3" "UNEMP_lag4"
##
## $GDP
## [1] "GDP"      "GDP_lag1" "GDP_lag2" "GDP_lag3" "GDP_lag4"
##
## $WAGE
## [1] "WAGE"      "WAGE_lag1" "WAGE_lag2" "WAGE_lag3" "WAGE_lag4"
##
## $EURIBOR
## [1] "EURIBOR"      "EURIBOR_lag1" "EURIBOR_lag2" "EURIBOR_lag3" "EURIBOR_lag4"
```

Practitioners should note that this process can be adjusted in various ways. For example, not every predictor requires the same lag. Additionally, a single predictor can be split into multiple variables if experts expect a different relationship with the target based on specific thresholds or if an asymmetric impact is anticipated. However, combining predictors into the same group limits the subsequent step of predictor selection, as only one predictor from each group can be included in the SMI model estimation.

To combine predictors into different models for constructing the SMI, we use another function, `pg.c`. As noted earlier, combinations are limited to include only one predictor from each group. Additionally, the maximum number of predictors per model is set to three, following the rule of thumb that there should be at least 15 to 20 observations per predictor.

```
#predictor combinations
gr.c <- pg.c(groups = groups,
             max.pred = 3)
#sample of the model candidates
gr.c[c(1, 100, 500), ]
```

```
##            Var1 Var2         Var3
## 1         UNEMP <NA>         <NA>
## 100    GDP_lag4 WAGE         <NA>
## 500 UNEMP_lag4 WAGE EURIBOR_lag3
```

5.4 Constrained OLS Method for SMI

At its core, SMI construction relies on constrained OLS regression applied to the predictor combinations defined earlier. The current version of the `smi` package implements constraints based on the expected sign of each estimated coefficient. To perform this step, practitioners can use the `model.est` function, which takes as inputs the predictor combinations (from the previous step `gr.c`), the expected relationship between predictors and the target variable (defined in the vector `ps`), realized values of the target and predictors (from the simulation dataset `db` where `TYPE` equals `REALIZED`), the name of the target column in `db` (`ODR`), and optionally, weights for the OLS estimation. In the example below, all weights are set equally, assigning the same importance to each observation period.
Because the number of predictor groups and their combinations can be significant in real-world applications,

the estimation process may take some time to complete.

```
#constrained ols
res.a <- model.est(gr.c = gr.c,
                   ps = ps,
                   db = db[db$TYPE%in%"REALIZED", ],
                   target = "ODR",
                   weights = rep(x = 1, times = sum(db$TYPE%in%"REALIZED")))
```

The above procedure results in returning two output elements. The first one is the estimated models and another one is the within the sample predictions of each estimated model.

```
#extract models
res <- res.a[["models"]]
#sample of the estimated models
res[c(1:2, 11:12, 2487:2490), ]
```

```
##       model.id  coefficient    estimate r.squared       aic       bic
## 1      Model_1  (Intercept)  0.07255059        NA -284.7661 -280.6452
## 2      Model_1        UNEMP  0.00000000        NA -284.7661 -280.6452
## 11     Model_6  (Intercept)  0.07461722 0.1536507 -292.4418 -286.2605
## 12     Model_6          GDP -0.18381377 0.1536507 -292.4418 -286.2605
## 2487 Model_670  (Intercept)  0.04449066 0.7695440 -342.2094 -334.2534
## 2488 Model_670     GDP_lag4  0.00000000 0.7695440 -342.2094 -334.2534
## 2489 Model_670    WAGE_lag4 -0.06195123 0.7695440 -342.2094 -334.2534
## 2490 Model_670 EURIBOR_lag4  1.33832544 0.7695440 -342.2094 -334.2534
##      zero.coeff   predictor lag
## 1          TRUE (Intercept)   0
## 2          TRUE       UNEMP   0
## 11        FALSE (Intercept)   0
## 12        FALSE         GDP   0
## 2487       TRUE (Intercept)   0
## 2488       TRUE         GDP   4
## 2489       TRUE        WAGE   4
## 2490       TRUE     EURIBOR   4
```

```
#extract predictions
pred <- res.a[["pred"]]
#predictions of the model 1
pred[1]
```

```
## [[1]]
##  [1] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
##  [7] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [13] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [19] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [25] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [31] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [37] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [43] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [49] 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059 0.07255059
## [55] 0.07255059 0.07255059 0.07255059 0.07255059
```

```
#predictions of the model 670
pred[670]
```

```
## [[1]]
```

```
##  [1]         NA         NA         NA         NA 0.12632065 0.13152865
##  [7] 0.12070805 0.09596245 0.09129519 0.08526334 0.07874013 0.07676344
## [13] 0.07678716 0.07707635 0.07993763 0.08265231 0.08513708 0.08714293
## [19] 0.08877826 0.08125542 0.07848865 0.07567098 0.07235373 0.07175416
## [25] 0.06872368 0.06782865 0.06487729 0.06632221 0.06937410 0.06513395
## [31] 0.06415114 0.06371023 0.05825382 0.06136286 0.06180980 0.05851450
## [37] 0.06253335 0.05943434 0.05568268 0.05938748 0.05830096 0.05898082
## [43] 0.06215019 0.05763166 0.05769476 0.05996718 0.06095773 0.06105226
## [49] 0.05913510 0.05485411 0.05657956 0.06029311 0.06836799 0.05837128
## [55] 0.05565130 0.05617342 0.04957532 0.05737804
```

5.5 Filtering Models Based on Coefficient Constraints

Some models may have estimated coefficients equal to zero, indicating that the constrained estimation has reached a boundary where the expected sign cannot be maintained. Such models can be excluded from further steps in creating the SMI. The following code removes these models.

```r
#identify models with zero coefficients
me <- unique(res[, c("model.id", "zero.coeff")])
#exclude models with zero coefficients from the full set of estimated models
res.f <- res[!res$zero.coeff, ]
#remove predictors of models with zero coefficients from all model predictions
pred <- pred[!me$zero.coeff]
```

5.6 Constructing the SMI via Model Averaging

The final step in SMI construction involves averaging the coefficients from the selected models to obtain the final components of the SMI. This can be done using the `smi` function, which is included in the package. The current version supports only simple averaging, but practitioners can easily modify the function's source code to incorporate alternative weighting schemes if needed.
The following code first averages the model coefficients used to construct the SMI, then assigns the SMI to the `db` dataframe, and finally prints the contribution of each predictor to the SMI:

```r
#smi function
smi.r <- smi(models = res.f,
             db = db,
             weights = "average")
#add smi to db
db$smi <- smi.r$smi
#smi coefficients
smi.r$coef
```

```
##     coefficient     estimate
## 1   (Intercept)  0.035617054
## 2       EURIBOR  0.237987635
## 3  EURIBOR_lag1  0.285223483
## 4  EURIBOR_lag2  0.305118729
## 5  EURIBOR_lag3  0.250265329
## 6  EURIBOR_lag4  0.181902948
## 7           GDP -0.010650688
## 8      GDP_lag1 -0.016210407
## 9      GDP_lag2 -0.015250458
## 10     GDP_lag3 -0.011515322
## 11     GDP_lag4 -0.007205748
## 12   UNEMP_lag1  0.006069636
```

```
## 13   UNEMP_lag2  0.024892486
## 14   UNEMP_lag3  0.048206230
## 15   UNEMP_lag4  0.094342483
## 16         WAGE -0.017896242
## 17    WAGE_lag1 -0.022079269
## 18    WAGE_lag2 -0.021264857
## 19    WAGE_lag3 -0.014210180
## 20    WAGE_lag4 -0.007560304
```

As shown by the coefficients above, nearly all macroeconomic indicators at all four lags contribute to the SMI. The only exception is `UNEMP`, which does not have an immediate effect on `ODR` but shows the highest estimated coefficient at a four-quarter lag.

Once constructed, the SMI can be used as a predictor in the final FLI model.

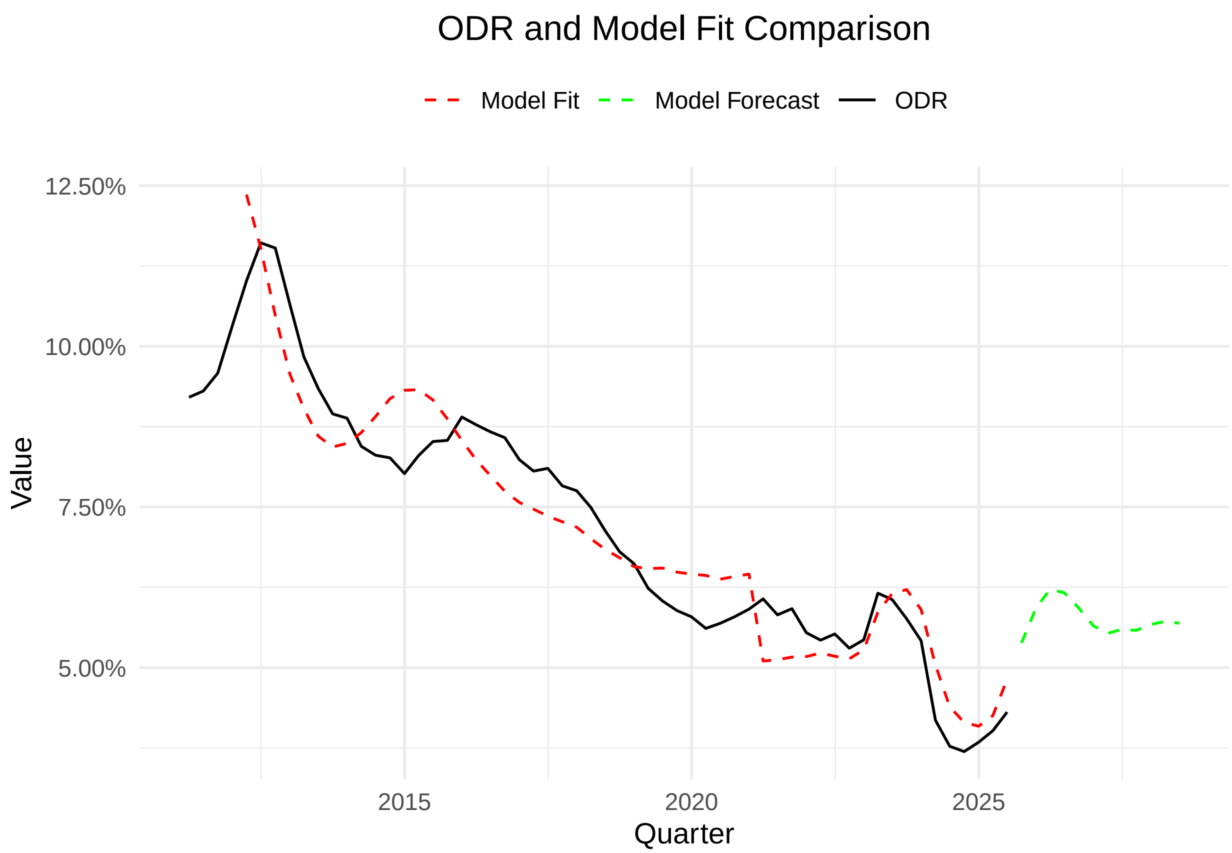
5.7 FLI Model with SMI as Predictor

After constructing the SMI, it can be used as a predictor in the final FLI model. The following example demonstrates how to do this, including the `DUMMY` variable to account for specific internal changes at the bank:

```r
#fli model
fli <- lm(formula = ODR ~ smi + DUMMY,
          data = db[db$TYPE%in%"REALIZED", ])
#model summary
summary(fli)
```

```
##
## Call:
## lm(formula = ODR ~ smi + DUMMY, data = db[db$TYPE %in% "REALIZED",
##     ])
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0134175 -0.0053692  0.0008983  0.0055565  0.0108532
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -0.003355   0.006061  -0.554                0.582
## smi          1.112579   0.079033  14.077 < 0.0000000000000002 ***
## DUMMY       -0.013110   0.002208  -5.938          0.000000257 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006524 on 51 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8963, Adjusted R-squared:  0.8922
## F-statistic: 220.4 on 2 and 51 DF,  p-value: < 0.00000000000000022
```

As a good practice, after estimating an FLI model, practitioners should visualize both the model fit and the forecasts based on the estimated model. The graph below illustrates this comparison:



Although this is not typically the final step, since OLS assumptions would usually be examined further, potential corrections applied if violations are detected, and other refinements explored, such as alternative link functions or reviewing predictors with limited contributions to the SMI, this example concludes the demonstration of the `smi` package. The main reason is the potentially large number of different designs even in the earlier steps. Nonetheless, this does not diminish the value of the example in illustrating the core idea behind the SMI. Practitioners are encouraged to adapt and extend these steps to meet their specific needs.