# Introducing SAMADB: A Free Analytic Macroeconomic Database for South Africa

## + Nowcasting of GDP and Unemployment

Sebastian Krantz

Kiel Institute for the World Economy

20th April 2023

# Table of Contents

# Introduction

Modern macroeconomic resarch is increasingly based on complex computational methods and open source languages like R, Python or Julia. **This Calls for Databases and Systems that Provide:**

- Efficient Online Access (to a wide variety of data, preferably from within multiple analytical languages)
- Tidy Data (data that is organized in rows and columns, does not need to be reshaped or cleaned, and is ready for analysis)
- Highly Customizable Queries (a typical research dataset features certain series and years, pulled together across a wide variety of datasets and data sources)
- Automation, Consistency and Reproducibility of Research ($\rightarrow$ systems should be open source & preferably version controlled)

$\rightarrow$ Difficult to have a single system serving all of these objectives.

SAMADB is an Optimized Analytical Database in MySQL that serves some of these objectives particularly well. It is largely based on EconData and largely a complementary product.

## A Comparison

**EconData** (https://www.econdata.co.za/)

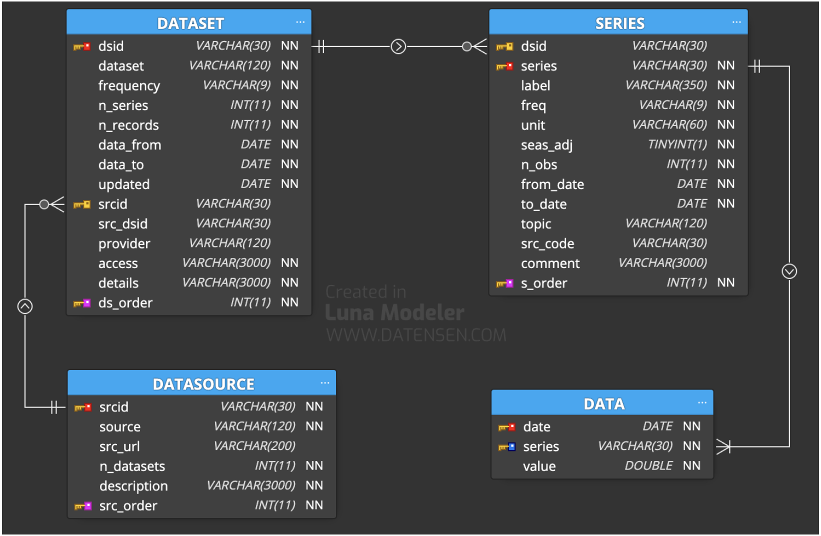
- Millions of time series (granular data) with version history
- Real time (daily) updates
- Open R and SDMX/JSON APIs (need to provide credentials)
- Moderately harmonized & tidy data (R API has tidy return, metadata and date coding differ across series and datasets)
- Decent speed (+ client-side JSON decoding and tidying)
- No cross-dataset queries of time series (to my knowledge)

**SAMADB** (South Africa Macroeconomic Database)

- ∼10,000 frequently used time series (version control only through offline vintages, no granular financial data)
- Near real time (weekly) updates through EconData
- Expressive free APIs for R, Python and Julia
- Harmonized & tidy data (uniform metadata and date coding across datasets and series, globally unique series codes, UTF8)
- Blazing fast (optimized database model, direct SQL queries, no client-side JSON parsing)
- Unlimited global queries (across datasets, series, time and frequency → generate research dataset with a single API call)
- Some data (such as the Quarterly Bulletin) not (currently) available on EconData → also updated weekly if available

# SAMADB Structure: Entity Relationship (ER) Diagram

DATASOURCE has multiple DATASET has multiple SERIES has multiple DATA




'series' is the unique primary key of the SERIES tables i.e. unique across DATASET's

# Weekly updates scheduled through GitHub Actions

```yaml
name: Update SAMADB Database

on:
  # Every Week Thursday at 1:15 AM
  schedule:
    - cron: '15 1 * * 4'

jobs:
  SAMADBUpdate:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: r-lib/actions/setup-r@v2
      - uses: r-lib/actions/setup-r-dependencies@v2
        with:
          working-directory: "r"
          cache-version: 2
          dependencies: "NA"
          extra-packages: |
            cran::fastverse
            cran::readxl
            cran::rvest
            github::SebKrantz/econdatar@hybrid_meta

      - name: Get EconData
        run: Rscript ./datasources/econdata/download_econdata.R
      - name: Get STATSSA
        run: Rscript ./datasources/statssa_sarb_other/statssa_other.R
      - name: Assemble Database
        run: Rscript ./database/assemble_database.R
      - name: Update Database
        run: Rscript ./database/upload_data.R
      # Commit all changed files back to the repository -> Generates a Vintage of the Database
      - uses: stefanzweifel/git-auto-commit-action@v4
```

## In Particular

- The download scripts fetch, do basic tidying, and save the raw data from different sources (EconData, STATSSA)
- The assembly script pulls all data together, enforces uniformity, creates the DB tables and validates them e.g.
  - Process strings: capitalize series codes and replace "." → "_". Remove white spaces in descriptions and enforce UTF8 characters.
  - Harmonize dates by data frequency: Monthly → "YYYY-MM-01", Quarterly → "YYYY-03/06/09/12-01", Annual → "YYYY-12-01". This ensures sensible output from mixed-frequency queries.
  - Compute content and time coverage statistics for all series and datasets. This helps the user quickly find suitable time series.
  - Check the primary and foreign key constraints of the database.
- The update script replaces old data in the database and rebuilds the search indexes for optimal query performance

# Database APIs

There is (currently) no way to access SAMADB from the web. But SAMADB has expressive APIs for R, Python and Julia. Download them by clicking **here**. In the near future the APIs will be available on official software repositories. All APIs have these functions:

```
Datasets providing information about the available data

datasources() - Data sources

datasets() - Datasets

series() - Series (can be queried by dataset)

Retrieve the data from the database

data() - By default retrieves all data

Functions to reshape data and add temporal identifiers

pivot_wider() - Wrapper around DataFrames.unstack()

pivot_longer() - Wrapper around DataFrames.stack()

expand_date() - Create year, quarter, month and day columns from a date

Helper functions to convert inputs to date strings

as_date() - E.g. "2011M01" -> "2011-01-01"
```

# A Walkthrough of the R API with Examples

```r
library(fastverse)
fastverse_extend(xts, tsbox, seasonal, ggplot2, dfms, samadb, install = TRUE)

# This just gets the full database overview tables
DATASOURCE = sm_datasources() %T>% View()
DATASET = sm_datasets() %T>% View()
SERIES = sm_series() %T>% View()

# Get a small dataset on business cycles
BC = sm_data("BUSINESS_CYCLES")
# Show series codes and labels, and summarize
namlab(BC)
qsu(BC, vlabels = TRUE)
# Plot the monthly series after 2010
BC %>% gvr("date|_M_") %>% sbt(date >= sm_as_date(2010)) %>% as.xts() %>% plot(legend.loc = "topleft")

# Get daily data from the Quarterly Bulletin for financial year 2020/21
QB_D = sm_data("QB", freq = "D", from = "2020Q2", to = "2021Q1")
# Compute growth rates and summarize
QB_D %>% G(t = ~ date) %>% replace_Inf() %>% qsu()

# See what series we have on electricity
sm_series("ELECTRICITY")
# Query seasonally adjusted values and volumes of electricity production
sm_data(series = .c(ELE001_S_S, ELE002_I_S)) %>% as.xts() %>% STD() %>% plot(legend.loc = "topleft")

# We could merge this with BC, or simply let the database do it for us
BC_ELC = sm_data("BUSINESS_CYCLES", series = "ELE001_S_S", from = 2010)
BC_ELC %>% gvr("_Q_", invert = TRUE) %>% as.xts() %>% STD() %>% plot(legend.loc = "topleft")
# Some more API functions. Could also specify expand.date = TRUE and wide = FALSE in sm_data()
BC_ELC %>% gvr("CPI|PPI", invert = TRUE) %>% sm_expand_date() %>%   # Additional identifiers
  collap( ~ year + quarter, fmean, flast) %>%                      # Aggregate to quarterly frequency
  sm_pivot_longer() %>% fmutate(value = STD(value, series)) %>%     # Reshape and standardize
  ggplot(aes(x = date, y = value, colour = label)) + geom_line() +  # Plot
  guides(colour = guide_legend(ncol = 1)) + theme(legend.position = "bottom")
# Only in R: Transposition of data and saving to excel
BC_ELC %>% sm_transpose(date.format = "%m/%Y") %>% sm_write_excel("BC_ELC.xlsx")
```

```r
# Now let's get a more interesting dataset
ind <- .c(
  KBP7091N, KBP7090N, # Coincident and leading business cycle indicators
  MAN001_S_S, MIN001_S_S, # Total manufacturing and mining production
  MTS003_S, RET008_S_S, WHO001_S_S, # Motortrade, wholesale and retail trade
  KBP7082T, KBP7195M, KBP7202M, KBP7203M, KBP7204M, # Manufacturing orders, sales and inventories
  KBP7196M, # Cargo at South African Ports
  CPI60001, PPI001, # Consumer and producer prices
  KBP1260M, KBP1261M, # Value and volume of credit card purchses
  KBP1368M, KBP1347M, # Total credit and credit to the private sector
  KBP1474M, KBP1478M, # New mortgages and mortgages paid out
  MIG001_A_N0_TA, MIG001_A_A0_TA, TOU036_S, TOU011_S, # Tourism
  CURX600_M, CURM600_M, # Exports and imports
  NGFC020_M, NGFC040_M, # Cash flow revenue and expenditure
  KBP5393M, KBP5395M, # Nominal and real effective exchange rates
  ELE001_S_S, ELE002_I_S # Electricity generation
)
# Get metdata of requested series: in this order (default is a fixed order)
series <- sm_series(series = ind)[match(ind, series)]
# Get data (default is to maintain the requested order if only series are requested)
data <- sm_data(series = ind, from = max(series$from_date))
# Basic exploration
qsu(data, vlabels = TRUE)
data %>% as.xts() %>% STD() %>% plot(lwd = 1)
# Getting series not seasonally adjusted and adjusting them with X13
sadj_ind <- series %$% series[!seas_adj]
get_vars(data, sadj_ind) <- data %>% get_vars(c("date", sadj_ind)) %>%
  as.xts() %>% ts_ts() %>% seas() %>% final() %>% mctl()
# Computing growth rates
data_growth <- data %>% G(t = ~ as.yearmon(date), stub = FALSE) %>% replace_Inf()
data_growth %>% as.xts() %>% STD() %>% plot(lwd = 1)
# Compute and plot greatest correlations with electricity generation
oldpar <- par(mai = c(.5,4,.5,.5))
data_growth[year(date) != 2020] %>% num_vars() %>% pwcor() %>%
  ss(rownames(.) %!in% .c(ELE001_S_S, ELE002_I_S), .c(ELE001_S_S, ELE002_I_S)) %>%
  rowMeans() %>% sort() %>% barplot(horiz = TRUE, las = 1,
    names.arg = paste0(names(.), ": ", series[match(names(.), series), label] %>% substr(1, 30)))
par(oldpar)
# Estimate a dynamic factor coincident index
dfm_mod <- DFM(num_vars(data_growth), 1, 3, idio.ar1 = TRUE)
plot(dfm_mod, method = "all")
```

# Dynamic Factor Nowcasting Model

I estimate a dynamic factor model following Bańbura & Modugno (2014) and (Bok et al., 2018) (New York Fed Nowcasting Model)

$$\mathbf{x}_t = \mathbf{C}_0\mathbf{f}_t + \mathbf{e}_t, \qquad \mathbf{e}_t \sim N(\mathbf{0}, \mathbf{R}_0) \tag{1}$$

$$\mathbf{f}_t = \sum_{j=1}^{p} \mathbf{A}_j\mathbf{f}_{t-j} + \mathbf{u}_t, \qquad \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{Q}_0). \tag{2}$$

The model is estimated using a Kalman Filter and Smoother and the Expectation Maximization (EM) algorithm, after transforming the model equations to State Space (stacked, VAR(1)) form

$$\mathbf{x}_t = \mathbf{C}\mathbf{F}_t + \mathbf{e}_t, \qquad \mathbf{e}_t \sim N(\mathbf{0}, \mathbf{R}_a) \tag{3}$$

$$\mathbf{F}_t = \mathbf{A}\mathbf{F}_{t-1} + \mathbf{u}_t, \qquad \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{Q}). \tag{4}$$

Observation errors $\mathbf{e}_t$ evolve according to autoregressive AR(1) processes to allow for unmodelled idiosynchratic dynamics

$$\mathbf{e}_t = \mathbf{\Phi}\mathbf{e}_{t-1} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R}). \tag{5}$$

Quarterly series of Real GDP, Nominal GDP, and Unemployment are modelled by an unobserved monthly counterpart following Mariano & Murasawa (2003). Bańbura & Modugno (2014) model quarterly variable as product of monthly counterpart: $X_t^q = \tilde{X}_t^m \tilde{X}_{t-1}^m \tilde{X}_{t-2}^m$. Taking quarterly log-differences yields

$$\begin{aligned}\log(X_t^q) - \log(X_{t-3}^q) = \log(\tilde{X}_t^m) - \log(\tilde{X}_{t-3}^m) + \log(\tilde{X}_{t-1}^m) - \\ \log(\tilde{X}_{t-4}^m) + \log(\tilde{X}_{t-2}^m) - \log(\tilde{X}_{t-5}^m).\end{aligned} \tag{6}$$

Adding and subtracting lags on the RHS, and denoting the growth rates as $x_t^q = \log(X_t^q) - \log(X_{t-3}^q)$ and $\tilde{x}_t^m = \log(\tilde{X}_t^m) - \log(\tilde{X}_{t-1}^m)$ yields

$$x_t^q = \tilde{x}_t^m + 2\tilde{x}_{t-1}^m + 3\tilde{x}_{t-2}^m + 2\tilde{x}_{t-3}^m + \tilde{x}_{t-4}^m. \tag{7}$$

Assuming that $\tilde{\mathbf{x}}_t^m$ admits the same DFM representation as the observed monthly variables, i.e. $\tilde{\mathbf{x}}_t^m = \mathbf{C}_0^q \mathbf{f}_t + \mathbf{e}_t^q$, we write a DFM representation for $\mathbf{x}_t^q$

$$\mathbf{x}_t^q = \mathbf{C}_0^q\mathbf{f}_t + \mathbf{e}_t^q + 2(\mathbf{C}_0^q\mathbf{f}_{t-1} + \mathbf{e}_{t-1}^q) + 3(\mathbf{C}_0^q\mathbf{f}_{t-2} + \mathbf{e}_{t-2}^q) + 2(\mathbf{C}_0^q\mathbf{f}_{t-3} + \mathbf{e}_{t-3}^q) + \mathbf{C}_0^q\mathbf{f}_{t-4} + \mathbf{e}_{t-4}^q.$$

This yields the following stacked DFM representation ($r =$ Nfactors, $p =$ Nlags)

$$\begin{aligned}\tilde{\mathbf{x}}_{t\ (n\times 1)} &= (\mathbf{x}_t^{m\prime},\ \mathbf{x}_t^{q\prime})', \text{where } \mathbf{x}_t^m \text{ is } n_M \times 1 \text{ and } \mathbf{x}_t^q \text{ is } n_Q \times 1 \\ \tilde{\mathbf{F}}_{t\ (rp+5n_Q\times 1)} &= (\mathbf{f}_t', \dots, \mathbf{f}_{t-p}', \mathbf{e}_t^{q\prime}, \dots, \mathbf{e}_{t-4}^{q\prime})', \text{where } \mathbf{e}_t^q \text{ is } n_Q \times 1 \\ \tilde{\mathbf{C}}_{(n\times rp+5n_Q)} &= \begin{pmatrix} \mathbf{C}_0^m & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_0^q & 2\mathbf{C}_0^q & 3\mathbf{C}_0^q & 2\mathbf{C}_0^q & \mathbf{C}_0^q & \mathbf{0} & \mathbf{I} & 2\mathbf{I} & 3\mathbf{I} & 2\mathbf{I} & \mathbf{I} \end{pmatrix}.\end{aligned}$$

# Computing the News

Since the DFM provides forecasts of all variables, it can be used to assess the impact of data updates on the nowcast. Let $\hat{\mathbf{x}}_{t+1}$ be the forecast of $\mathbf{x}_t$ at time $t$. In time $t+1$ we observe $\mathbf{x}_{t+1}$, thus

$$\mathbf{z}_{t+1} = \mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1} \tag{8}$$

denotes the 'news'. Its impact on the nowcast of quarterly variables $\hat{\mathbf{x}}^{q}_{\tau}$ ($\tau$ denotes the current quarter) is a function of the DFM parameters. It turns out that we can obtain a vector of weights $\mathbf{w}$ that summarizes this impact

$$\hat{\mathbf{x}}^{q,t+1}_{\tau} - \hat{\mathbf{x}}^{q,t}_{\tau} = diag(\mathbf{w})\ \mathbf{z}_{t+1}. \tag{9}$$

In other words:

$$\text{nowcast revision} = \text{model-based weight} \times \text{news}. \tag{10}$$

For computational details see Bańbura & Modugno (2014).

# Data

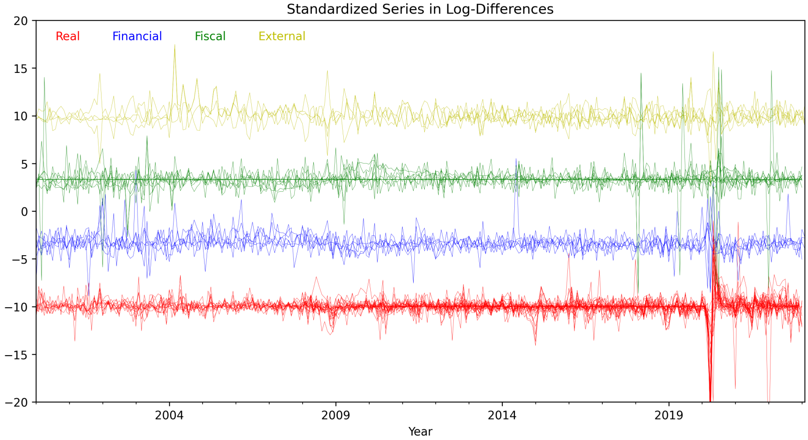
Parsimonious dataset of 57 series (54 monthly, 3 quarterly) updated through EconData, grouped into 4 sectors and topics:

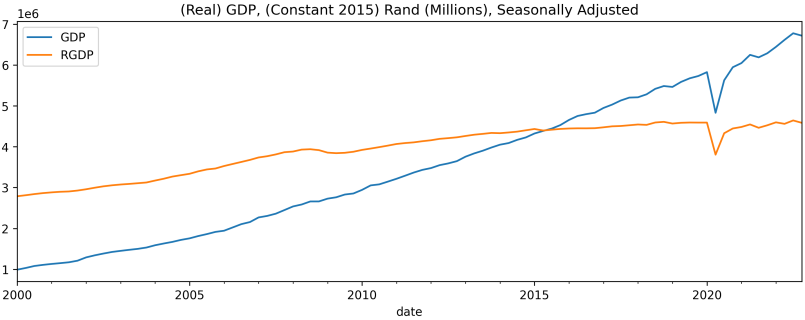
Table: Summary of Time Series for Nowcasting Model

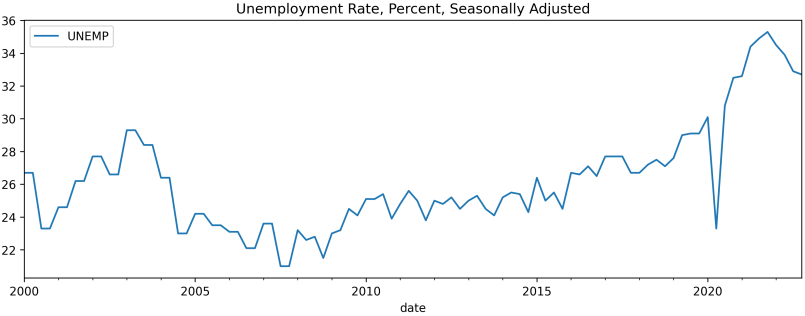
| Sector | Topic | N | Description |
|---|---|---|---|
| Real | Production | 8 | GDP, Manufacturing, Mining and Electricity |
| Real | Sales | 6 | Motor Trade, Retail and Wholesale |
| Real | Prices | 6 | Consumer and Producer Prices |
| Real | Tourism | 7 | Migration, Tourism and Accommodation |
| Real | Other Real | 6 | Land Transport and Unemployment |
| Financial | Money and Credit | 4 | Monetary Aggregates and Credit Claims |
| Financial | Other Fiancial | 2 | Net Foreign Assets and Insolvencies |
| External | Trade | 2 | Exports and Imports |
| External | Exchange Rates | 2 | USD Exchange Rate and NEER |
| External | Reserves | 2 | Official Reserve Assets and FX Reserves |
| Fiscal | Cash Flow | 3 | Cash-Flow Revenue, Expense and Balance |
| Fiscal | Financing | 5 | Bonds, T-Bill and Foreign Financing |
| Fiscal | Debt | 4 | Total, Domestic and Foreign Debt |

*Notes:* Production and Sales include value and volume indicators.

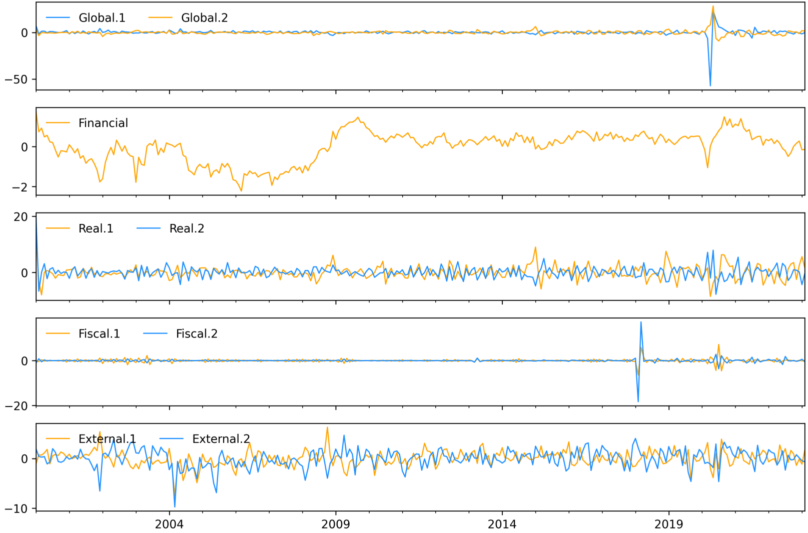
The model is estimated with all series in log-difference growth rates. Seasonal series are adjusted using X-13 ARIMA Seats (Sax & Eddelbuettel, 2018). This shows the adjusted monthly series



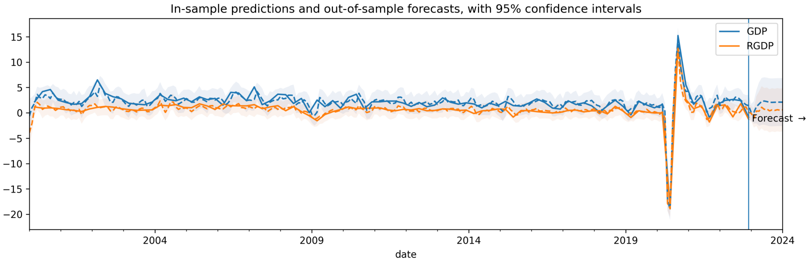

(Real) GDP, (Constant 2015) Rand (Millions), Seasonally Adjusted
1e6
GDP
RGDP
7
6
5
4
3
2
1
2000
2005
2010
2015
2020
date

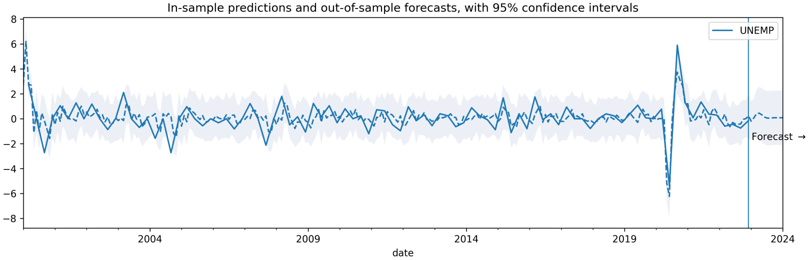

Unemployment Rate, Percent, Seasonally Adjusted
UNEMP
36
34
32
30
28
26
24
22
2000
2005
2010
2015
2020
date

To distribute factor loadings, the factor structure is blocked following (Bok et al., 2018) into 2 Global, 2 Real, 1 Financial, 2 Fiscal, and 2 External factors. Factors follow a VAR(2) process.



# This shows in-sample predictions and forecasts, starting 2023Q1



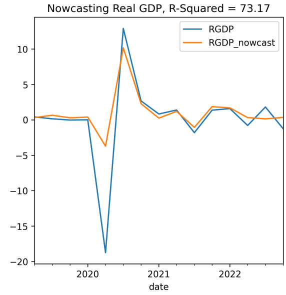



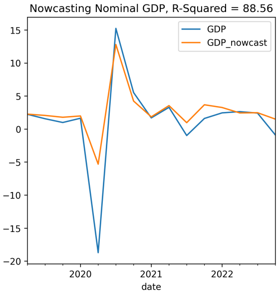
# Evaluating the Model

- Properly evaluating the model requires vintages $\rightarrow$ not available since we just started this

- I do a crude evaluation nowcasting GDP for the past 15 quarters: 2019Q2-2022Q4, using monthly data up to the end of the quarter and quarterly data up to the previous quarter

- Other candidate models are
  - A smaller DFM with only 34 out of the 54 monthly variables, which are deemed especially important, and 1 factor/block (DFM_SM)
  - A hybrid (bridge equation) model that uses a single DFM of the monthly variables, forecasts the variables and nowcasts the quarterly variabes using linear regression on the forecasted monthly ones (BE)
  - Same, but expanding monthly vars+fc into quarterly ones (blocking) and forecasting using LASSO tuned with LOO-CV (BE_LA)
  - Both bridge models with the same block structure as the DFMs (i.e. individual DFM's to estimate factors for different blocks, a global VAR(2) to forecast all factors, and then lm/LASSO) (BL suffix)



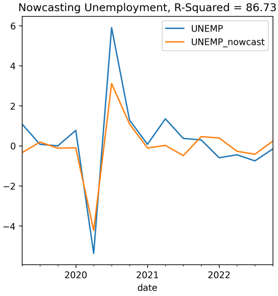




Table: Nowcast Evaluation for Real GDP: 2019Q2-2022Q4

| | Naive | DFM | DFM_SM | BE | BE_LA | BE_BL | BE_LA_BL |
|---|---|---|---|---|---|---|---|
| Bias | 0.12 | 0.98 | 0.97 | 1.25 | 0.56 | 1.21 | 0.79 |
| MAE | 5.58 | 1.73 | 1.84 | 1.77 | 1.52 | 1.52 | 2.09 |
| RMSE | 10.36 | 4.02 | 4.05 | 3.97 | 3.08 | 3.63 | 4.59 |
| R-Squared | -1.78 | 0.55 | 0.55 | 0.56 | 0.74 | 0.63 | 0.41 |
| U2 | 1.00 | 0.80 | 0.80 | 0.79 | 0.57 | 0.73 | 0.86 |
| Bias Prop. | 0.00 | 0.06 | 0.06 | 0.10 | 0.03 | 0.11 | 0.03 |
| Var. Prop. | 0.00 | 0.64 | 0.70 | 0.24 | 0.88 | 0.33 | 0.85 |
| Cov. Prop. | 1.00 | 0.30 | 0.24 | 0.66 | 0.08 | 0.56 | 0.12 |

Table: Nowcast Evaluation for Nominal GDP: 2019Q2-2022Q4

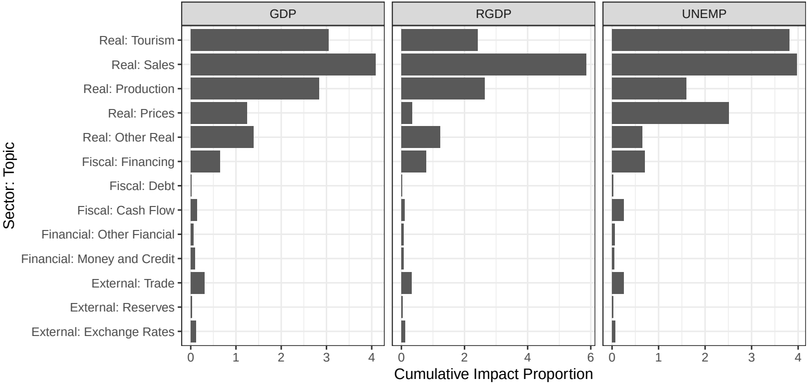
| | Naive | DFM | DFM_SM | BE | BE_LA | BE_BL | BE_LA_BL |
|---|---|---|---|---|---|---|---|
| Bias | 0.22 | 1.26 | 1.18 | 1.18 | 0.58 | 0.43 | 0.75 |
| MAE | 5.91 | 1.78 | 1.78 | 1.99 | 2.12 | 1.17 | 1.99 |
| RMSE | 11.07 | 3.68 | 3.62 | 3.71 | 3.63 | 1.45 | 3.56 |
| R-Squared | -1.72 | 0.68 | 0.69 | 0.67 | 0.69 | 0.95 | 0.70 |
| U2 | 1.00 | 0.66 | 0.64 | 0.67 | 0.57 | 0.27 | 0.58 |
| Bias Prop. | 0.00 | 0.12 | 0.11 | 0.10 | 0.03 | 0.09 | 0.04 |
| Var. Prop. | 0.00 | 0.69 | 0.73 | 0.31 | 0.87 | 0.01 | 0.86 |
| Cov. Prop. | 1.00 | 0.19 | 0.17 | 0.59 | 0.10 | 0.90 | 0.10 |

Table: Nowcast Evaluation for Unemployment: 2019Q2-2022Q4

| | Naive | DFM | DFM_SM | BE | BE_LA | BE_BL | BE_LA_BL |
|---|---|---|---|---|---|---|---|
| Bias | 0.09 | -0.30 | -0.28 | 0.22 | -0.19 | 0.49 | 0.25 |
| MAE | 2.10 | 0.74 | 0.82 | 1.04 | 1.24 | 1.36 | 1.52 |
| RMSE | 3.71 | 1.03 | 1.11 | 1.88 | 2.17 | 2.37 | 3.12 |
| R-Squared | -1.81 | 0.77 | 0.73 | 0.23 | -0.02 | -0.21 | -1.11 |
| U2 | | | | | | | |
| Bias Prop. | 0.00 | 0.08 | 0.06 | 0.01 | 0.01 | 0.04 | 0.01 |
| Var. Prop. | 0.00 | 0.52 | 0.56 | 0.35 | 0.94 | 0.13 | 0.09 |
| Cov. Prop. | 1.00 | 0.39 | 0.38 | 0.64 | 0.05 | 0.83 | 0.90 |

# Summarizing the News

I use the crude backtesting for 15 quarters 2019Q2-2022Q4 to compute model-based news and summarize it here[1]




The most impactful monthly series across all nowcasts are retail sales, accommodation stay nights sold, and manufacturing production.

[1] Since I only estimate one model per quarter, the next vintage always has new obs. for GDP and unemployment. These alone explain around 40% of their nowcast revision. To summarize the news I thus remove the quarterly variables. This exercise is thus deeply flawed, as we are usually interested in the impact of revisions to monthly data on the nowcast of a quarterly indicator within the current quarter.

# Nowcast Automation

**The combined power of GitHub Actions and SAMADB:**

- Action to generate a weekly nowcasting dataset (vintage) from SAMADB every Thursday night (following SAMADB update on Wednesday night) $\rightarrow$ save to GitHub repository
- Action to run nowcasting model on this vintage and the previous one, generating an updated nowcast and computing the news $\rightarrow$ triggered by vintage generation action completion, and appends nowcast and news to CSV files in the same repo
- Create a simple web-application (using Plotly Dash) that, on startup, fetches the CSV files from the repo and displays interactive visualizations

For transparency and easy fetching of data, this repo is public at https://github.com/Stellenbosch-Econometrics/SA-Nowcast. The Dash-app still needs to be finalized and will be hosted soon.

# Conclusions

- SAMADB is a simple, powerful and broadly accessible database for macroeconomic research and automation
- It is limited in scope and always will be (the focus is on core macro data, not millions of time series)
- Mixed frequency DFMs are powerful tools for macroeconomic nowcasting and interpreting the 'news', also in South Africa
- We hope to release a public nowcasting platform very soon, and also make SAMADB APIs available on public repositories i.e. CRAN, PyPI, and the Julia Package Registry
- We are working on a paper that does more rigorous nowcast/forecast evaluation using a larger dataset generated from the QB. Preliminary evidence suggests that the DFM performs much better with more data ($\sim$ 150 series).

# References


Bańbura, M. & Modugno, M. (2014). Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Journal of applied econometrics*, *29*(1), 133–160.

Bok, B., Caratelli, D., Giannone, D., Sbordone, A. M. & Tambalotti, A. (2018). Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics*, *10*, 615–643.

Mariano, R. S. & Murasawa, Y. (2003). A new coincident index of business cycles based on monthly and quarterly series. *Journal of applied Econometrics*, *18*(4), 427–443.

Sax, C. & Eddelbuettel, D. (2018). Seasonal adjustment by x-13arima-seats in r. *Journal of Statistical Software*, *87*, 1–17.