ECONOMIC STATISTICS CENTRE OF EXCELLENCE

A COLLABORATION WITH

Office for National Statistics

An Evaluation Framework for Targeted Indicators Aggregates vs. Disaggregates

George Kapetanios and Fotis Papailias

ESCoE Technical Report TR-17

May 2022

ISSN 2515-4664

TECHNICAL REPORT

An Evaluation Framework for Targeted Indicators Aggregates vs. Disaggregates
George Kapetanios and Fotis Papailias
ESCoE Technical Report TR-17
May 2022

Abstract

National statistics offices and similar institutions often produce country indices which are based on the aggregation of large number of disaggregate series. In some cases these disaggregate series are also published and, therefore, are available to be used for further research. In other cases the disaggregate series are available only for in-house purposes and are still under research on whether more indices could be extracted. This report is concerned with the very specific task of comparing gains in nowcasting using a single aggregate variable/index versus the full use of all the available disaggregate indices. This approach should be viewed as part of an overall dataset assessment framework where our aim is to assist the applied statistician on whether a novel dataset of time series could be useful to economics researchers.

Keywords: nowcasting, factor models, penalised regression, neural networks, support vector regression

JEL classification: C53, E37

Fotis Papailias, King's College London
fotis.papailias@kcl.ac.uk

Published by:
Economic Statistics Centre of Excellence
National Institute of Economic and Social Research
2 Dean Trench St
London SW1P 3HE
United Kingdom
www.escoe.ac.uk

An Evaluation Framework for Targeted Indicators Aggregates vs. Disaggregates

George Kapetanios Fotis Papailias*
King's Business School, DAFM
Economic Statistics Centre of Excellence

April 06, 2022

Abstract

National statistics offices and similar institutions often produce country indices which are based on the aggregation of large number of disaggregate series. In some cases these disaggregate series are also published and, therefore, are available to be used for further research. In other cases the disaggregate series are available only for in-house purposes and are still under research on whether more indices could be extracted. This report is concerned with the very specific task of comparing gains in nowcasting using a single aggregate variable/index versus the full use of all the available disaggregate indices. This approach should be viewed as part of an overall dataset assessment framework where our aim is to assist the applied statistician on whether a novel dataset of time series could be useful to economics researchers.

Keywords: Nowcasting, Factor Models, Penalised Regression, Neural Networks, Support Vector Regression.

*Corresponding author. King's Business School, King's College London, Bush House, 30 Aldwych, London, WC2B 4BG, UK. Knot Analytics Ltd. E-mail: fotis.papailias@kcl.ac.uk

JEL Codes: C53, E37.

Contents

1 Introduction

This technical report has a very specific goal: to examine gains in nowcasting comparing a linear regression model using a single aggregate variable with models which utilise all the underlying disaggregate series. At first, this might seem like a task with limited scope. However, the applied research should view this approach as only a part of an overall assessment framework for novel datasets. Our main aim is to provide a framework which answers the following question regarding a candidate dataset of new indicators: "Should a national statistics institute invest resources in organising, editing, polishing and publishing this novel dataset of indicators and why?".

There are many ways one could provide an answer to the above question. There can be legit qualitative answers such as the importance that some indicators have in monitoring real-time social conditions (e.g., infections/deaths during the COVID-19 pandemic) which do not require further numerical support. Or, there can be quantitative answers which provide empirical evidence that a specific dataset is useful in specific research tasks. This report attempts to answer the big question on dataset usefulness taking the second stance; that of the gains in empirical exercises.

Kapetanios and Papailias (2021a), and their work in the subsequent technical report[1], already assess a wide dataset of novel indicators, the ONS Real-Time Indicators dataset, in the construction of a coincident index to monitor economic conditions during a crisis using the COVID-19 pandemic as a case study. This already provides empirical evidence that a large set of (aggregate) indicators can be useful to applied researchers and, therefore, it worths being maintained and continuously published with as less delay as possible. However, in this approach their focus is on different indicator categories (from VAT indices to the use of debit and credit cards, from online job advertisements to COVID-19 surveys, from online retail prices to traffic near ports, etc.).

Then, Kapetanios and Papailias (2021c, 2021d) provide the framework to eval-

[1]See Kapetanios and Papailias (2021b) which assesses the dataset based on gains in economic nowcasting.

uate a specific dataset of indicators comparing gains in economic nowcasting. In particular, they evaluate the predictive content of a targeted set of indicators which is VAT indices and the CHAPS-based indicator of credit and debit card purchases and whether their use improves nowcasting. Therefore, one could argue that we already have the framework to assess whether a novel set of indicators is useful to the applied researcher; that is via means of nowcasting. Of course nowcasting is not the only way to assess indicators, however it is one of the most direct ways and this is why we also adopt it in the current approach of this report.

The previous approaches either evaluate a large number of indicators across different categories (Kapetanios and Papailias, 2021a, 2021b) or focus on the use of some targeted aggregate indicators (Kapetanios and Papailias, 2021c, 2021d). In the current approach we take a step further and focus on a specific set of indicators, however this time our aim is to compare gains in nowcasting using the a linear model with the main aggregate series versus models which use all the underlying disaggregate series.

The motivation behind this approach is to shed light in all aspects of the novel dataset and not just the main index which is what -usually- most of the researchers do. In particular, there might be cases where one could employ the main index and identify trivial gains in nowcasting and -wrongly- conclude that this dataset is not really useful in economics applications. However, we must not forget that this "aggregate" index is based on a large (possible very large) universe of disaggregate series which, in turn, might prove useful when used in nowcasting and, thus, revert the conclusion.

It is important to highlight that our aim here is to provide a proof-of-concept and standardise the way a novel dataset should be considered by national statistics institutes, such as the ONS in the UK. This justifies our use of the online job advertisements and the port traffic data which are components of the ONS Real-Time Indicators dataset. Ideally, one would like to access the data in its most disaggregate level however, given data access restrictions, this report compares the main aggregates (Total UK indices - Level I) to their disaggregates (Individual Categories or Regions - Level II). Still, this allows for adequate evidence in support of the evalua-

tion framework.

The rest of this report is structured as follows. Section 2 briefly describes the dataset used in our illustrative examples. Section 3 provides the details on the econometric setup. Section 4 explains the out-of-sample nowcasting exercise design. Section 5 briefly discusses the empirical results. Finally, Section 6 offers the concluding remarks.

2 Data Description and Limitations

As discussed in the previous section, this task is heavily based on nowcasting. For this, one requires a variable which is to be used as the target (i.e. the variable to be estimated or, in terms of linear regression, the dependent variable). In what follows, we set the target variable as the "monthly index values of the gross domestic product (GDP) and the main sectors in the UK to four decimal places".[2]. Then, we switch our focus on two components of the wider ONS Real-Time Indicators dataset. These are: (i) the Online Job Advertisements[3], and (ii) the traffic in UK ports as measured by the visiting number of ships[4].

Starting with the Online Job Advertisements dataset, our aim is to evaluate gains in nowcasting comparing a linear model where the only predictor is the total online job advertisements index across all industries in the UK (Aggregate - Level I) to a large number of models which use the corresponding 43 subindices of the online job advertisements (Level II aggregates) which include the following industries & regions: Accounting/Finance, Admin/Clerical/Secretarial, Category Unknown, Catering & Hospitality, Charity/Voluntary, Construction/Trades, Creative/Design/Arts&Media, CustomerService/Support, Domestic Help, East Midlands, East of England, Education, Energy/Oil&Gas, Engineering, England, Facili-

[2]As provided on this ONS page; however, we downloaded the same data via the Macrobond data aggregator.

[3]As provided on this ONS page; however, we downloaded the same data via the Macrobond data aggregator.

[4]As provided on this ONS page; however, we downloaded the same data via the Macrobond data aggregator.

ties/Maintenance, Graduate, Healthcare& Social Care, HR & Recruitment, IT/Computing/Software, Legal, London, Management/Exec/Consulting, Manufacturing, Marketing/Advertising/PR, North East, North West, Northern Ireland, Other/General, Part-Time/Weekend, Property, Region Unknown, Sales, Scientific/QA, Scotland, South East, South West, Transport/Logistics/Warehouse, Travel/Tourism, Wales, West Midlands, Wholesale & Retail, Yorkshire & the Humber.

Continuing with the traffic in UK ports, our aim is to evaluate gains in nowcasting comparing a linear model where the only predictor is the extracted *trend* of the total visits of ships across all UK ports (Aggregate 1 - Level I) and/or the actual number of the total visits of ships across all UK (Aggregate 2 - Level I), or the combination of these two aggregates. Then, we aim to compare nowcasting based on a large number of models which use the corresponding 43 subindices of the traffic in ports which include the following categories & regions: All of UK Cargo & Tankers SA, All of UK Cargo & Tankers Unique Ships, All of UK Trend Cargo & Tankers, All of UK Unique Ships, Belfast, Belfast Cargo & Tankers, Belfast Cargo & Tankers Unique Ships, Belfast Unique Ships, Dover, Dover Unique Ships, Felixtowe, Felixtowe Cargo & Tankers Unique Ships, Felixtowe Unique Ships, Forth, Forth Cargo & Tankers Unique Ships, Forth Unique Ships, Grimsby & Immingham, Grimsby & Immingham Cargo & Tankers, Grimsby & Immingham Cargo & Tankers Unique Ships, Grimsby & Immingham Unique Ships, Liverpool, Liverpool Cargo & Tankers, Liverpool Cargo & Tankers Unique Ships, Liverpool Unique Ships, London, London Cargo & Tankers, London Cargo & Tankers Unique Ships, London Unique Ships, Milford Have Cargo & Tankers Unique Ships, Milford Haven Unique Ships, Portsmouth, Portsmouth Cargo & Tankers Unique Ships, Portsmouth Unique Ships, Southampton, Southampton Cargo & Tankers, Southampton Cargo & Tankers Unique Ships, Southampton Unique Ships, Tees & Hartlepool, Tees & Hartlepool Cargo & Tankers Unique Ships, Tees & Hartlepool Unique Ships, Tyne, Tyne Cargo & Tankers Unique Ships, Tyne Unique Ships.

Our target variable, i.e. the GDP, is expressed in the monthly frequency. On the other hand, the indicators of interest are -originally- expressed in weekly and/or daily frequency. This would allow, in principle, the use of the most recent daily

data (where available) to perform a daily nowcasting exercise for the GDP. However, something like this would be -perhaps- too noisy and not so informative. For this reason, we translate any daily data to the weekly frequency by using the weekly averages. This would allow to calculate four nowcasts per month.

The daily-turned-weekly real-time indicator datasets are available from 2019-04-07 to 2021-12-19 (i.e., 142 weekly observations). To match the same data as the real-time indicators, we use the monthly GDP from 2019-04-30 to 2021-10-31 (i.e., 31 monthly observations)[5]. In this case we face the following difficulties: (i) first, there is a mismatch in the frequencies, and (ii) second, the GDP sample size is very small to be used in any estimation. To overcome both the above issues we employ a "simple-and-standard" disaggregation of the monthly to weekly time series by smoothing in a fashion similar to Boot et al. (1967) to obtain the weekly equivalent of the target variable[6]. This allows to end up with a weekly version of the GDP, which includes 134 observations, and allows to have enough observations in-sample for the first nowcasting round. Then, once the weekly nowcasts are obtained, we can aggregate back to the monthly frequency and compare the estimate with the actual value.

Then, for the suggested framework to perform a pseudo out-of-sample cross-validation, we need to: (i) ensure stationarity of our time series, and (ii) replicate the pattern of missing values due to publication lags. Regarding stationarity, we take the period-to-period log growth of the monthly GDP and the first difference of the real-time indicators (Level I aggregates and Level II disaggregates). Next, we use a $T-2$ months as publication lags for the monthly GDP, which is equivalent to $T-8$ weeks in the weekly disaggregated version, and $T-0$ lags for the real-time indicators[7].

[5]This variable is published with a 2 months publication lag which explains the fact that values for November and December for 2021 are not observed during the time this report is being written

[6]By now, there exist other improved disaggregation methodologies, such as Santos Silva and Cardoso (2001), which , however, rely on explanatory variables. In this case, we want to keep our approach as less complicated and robust as possible, this is why we prefer a disaggregation based on smoothing rather than on other indicators.

[7]This might not be the case for the independent researcher but the ONS in-house researcher does have early access to this data.

Stationarity transformations, data descriptions and publication lags are available in the supporting .xlsx file.

3 Econometric Setup

3.1 Main Aggregates

Let y_t, $t = 1, ..., T$, be the target variable and x_t be the main Level I aggregate to be used in the modelling. We assume that there exists the linear relationship which can be described as:

$$y_t = a + \beta x_t + u_t \tag{1}$$

with $u_t \sim WN(0, \sigma_u^2)$ with $\sigma_u^2 < \infty$. It is important to notice that this is a rather simplistic modelling strategy as we totally ignore lags of y_t which could improve estimation and, thus, nowcasting. However, our purpose here is not to produce the best nowcasts for y_t. Instead, we want to compare a model which uses the Level I aggregates with models which somehow utilise the Level II disaggregates and conclude if there is any value on maintaining this dataset or not.

The latter case would involve the $x_t = (x_{1t}, ..., x_{Nt})'$ Level II disaggregated predictors. We do not need to assume a particular data generating process for y_t but simply posit the existence of a representation of the form:

$$y_t = a + f(x_{1t}, ..., x_{Nt}) + u_t, \tag{2}$$

which implies that $E(u_t|x_{1t}, ..., x_{Nt}) = 0$. While the potential nonlinearity in Equation (2) might, in principle, be worth exploring, it is extremely difficult to model nonlinearities in this context.[8] As a result, we consider an approximating linear

[8]However, in the next section we also include nowcasts based on random forests, neural networks and support vector regression which are able to capture non-linear relationships.

representation of the form:

$$y_t = a + \sum_{i=1}^{N} \beta_i x_{it} + u_t, \tag{3}$$

where u_t denotes a martingale difference process; depending on the estimation method this assumption can be relaxed to the standard assumptions in the classical linear regression framework.

3.2 Best Subset Selection

The next modelling approach is the best subset selection. Under this framework, the applied researcher aims to evaluate all possible models and choose the one which optimises a selected statistic; this statistic can be a measure of fit, such as R^2 or an information criterion, or can be an error-based statistics such as the Means Squared Error (MSE).

The main difficulty with this approach is that there are 2^N model combination to be evaluated. This means that if we have a small universe of indicators, say $N = 10$, we would have to evaluate 1,024 models which is easily doable with modern CPU power. However, in our case we have an FW set of 63 indicators which results in some millions of models to be estimated at each round in the evaluation exercise; something like this becomes almost infeasible in standard computers. This problem can be solved by adopting the "best forward stepwise" (BFW) regressions.

In the BFW approach, we start with the null model:

$$y_t = a + u_t, \tag{4}$$

which includes only a constant. Then, we also estimate the model by adding the first available variable, say x_{1t}:

$$y_t = a + \beta_1 x_{1t} + u_t, \tag{5}$$

and evaluate the chosen statistic of interest. We adopt the model which optimises the chosen statistic and then proceed with the next variable.

It is important to highlight that this approach can provide a first variable selection and indicate the relative importance of indicators, however it might miss information hidden in the cross-section of variables which are not sequential to each other.

3.3 Penalised Regressions

Penalised regression is one of the most popular ways for sparse regression in the literature which, depending on the nature of the penalty, also allows variable selection without having to consider the ordering of the variables. Various penalties have been suggested in order to effectively estimate the β_i parameters assigning zeros to the variables which should not be used in the regression (meaning that these are not part of the true model) and, consequently, in the nowcasting exercise. In what follows we denote $\beta_N = (\beta_1, ..., \beta_N)'$ and $x_N = (x_1, ..., x_N)'$.

3.3.1 Ridge Regression

Ridge Regression creates a linear regression model that is penalised with the L2-norm which is the sum of the squared coefficients. This has the effect of shrinking the coefficient values (and the complexity of the model) allowing some coefficients with minor contribution to the response to get close to zero (but not exactly equal to zero). The parameter estimators, $\widehat{\beta}^{Ridge}$, are then computed by solving the following optimisation problem:

$$\min_{\beta_N} \left\{ \sum_{t=1}^{T} \left(y_t - a - \beta_N^{'} x_{t,N} \right)^2 + \lambda \sum_{i=1}^{N} \beta_i^2 \right\}, \tag{6}$$

for given values of a and λ. λ is the penalty parameter. OLS corresponds to the no penalty case, where $\widehat{\beta}^{Ridge} \to \widehat{\beta}^{OLS}$ as $\lambda \to 0$. Also, it can be easily seen that $\widehat{\beta}^{Ridge} \to 0$ as $\lambda \to \infty$. By centering the columns of x, the intercept becomes $\widehat{\alpha} = \overline{y}$. Therefore, we typically center y, x_N and do not include the intercept term.

The variance and bias of the ridge regression estimator can be shown to be:

$$Var\left(\widehat{\beta}^{Ridge}\right) = \sigma^2 W x_N^{'} x_N W$$
$$Bias\left(\widehat{\beta}^{Ridge}\right) = -\lambda W \beta$$

where $W = \left(x_N^{'} x_N + \lambda I\right)^{-1}$. It can be shown that the total variance ($\sum_j Var\left(\widehat{\beta}_j\right)$) is a monotone decreasing sequence with respect to λ, while the total squared bias ($\sum_j Bias^2\left(\widehat{\beta}_j\right)$) is a monotone increasing sequence with respect to λ. The standard OLS assumptions are also required for Ridge regression.

Nowcasting using Ridge regression is straightforward and easy, in particular when implemented in a direct rather than iterated way (e.g., see, Marcellino et al., 2006). The algorithm can be described in three steps.

1. Replace the loss function with $\min_{\beta_{N,h}} \left\{\sum_{t=1}^{T}\left(y_t - a - \beta_{N,h}^{'} x_{t-h,N}\right)^2 + \lambda \sum_{i=1}^{N} |\beta_{i,h}|\right\}$, where h is the forecast horizon of interest, and compute $\widehat{\beta_h}^{Ridge}$ for each of a set of values of the tuning parameter λ.

2. Use a cross-validation (CV) scheme to select the preferred tuning parameter, $\widehat{\lambda}$, by minimising the cross-validated squared error risk.

3. Using the $\widehat{\beta_h}^{Ridge}$ associated with $\widehat{\lambda}$, produce the $h - step\ ahead$ forecasts (or nowcasts) as $\widehat{\beta_h}^{Ridge} x_{T,N}$ $(+\widehat{\alpha})$.

The above procedure is then recursively repeated in order to obtain the R out-of-sample nowcasts, $\widehat{y}_{T+h}, ..., \widehat{y}_{T+R+h}$.

It must be noted here that the above nowcasting implementation algorithm can be applied in all variable selection methods. Therefore, all the sparse regression methods which follow can produce nowcast estimates in the same fashion.

Since Ridge regression does not set coefficients exactly equal to zero (unless $\lambda \to \infty$, in which case they are all zero), ridge regression cannot perform variable selection and, even though it might perform well in terms of prediction accuracy, it does not offer a clear interpretation of the resulting nowcasts.

3.3.2 LASSO Regression

Least Absolute Shrinkage and Selection Operator (LASSO) creates a regression model that is penalised with the L1-norm which is the sum of the absolute coefficients. Because of the nature of this constraint, it tends to produce some coefficients that are exactly equal to 0 and, hence, gives more interpretable models. Simulation studies suggest that the LASSO enjoys some of the favourable properties of both subset selection and ridge regression. As originally noted by Tibshirani (1996), the lasso regression is better suited for predictor selection compared to the Ridge regression because the former method performs model/predictors selection keeping those variables which are more suitable for forecasting. The optimisation problem now becomes:

$$\min_{\beta_N} \left\{ \sum_{t=1}^{T} \left(y_t - a - \beta_N^{'} x_{t,N} \right)^2 + \lambda \sum_{i=1}^{N} |\beta_i| \right\}. \tag{7}$$

Although we cannot write the explicit formulas for the bias and variance of the LASSO estimator, the general trend is that the bias increases as λ increases and the variance decreases as λ increases.

Following Bühlmann and van de Geer (2011), we summarise the key properties and corresponding assumptions for the LASSO. Considering the true model in Equation (3), it is:

$$\frac{1}{T} \sum_{t=1}^{T} \left(x_{t,N} \left(\widehat{\beta}^{LASSO} - \beta \right) \right)^2 = O_P \left(\sum_{i=1}^{N} |\beta_i| \sqrt{\log(N)/T} \right), \tag{8}$$

where $O_P(\cdot)$ is with respect to $N \geq T \to \infty$. This implies that we achieve consistency of prediction if $\sum_{i=1}^{N} |\beta_i| \ll \sqrt{T/\log(N)}$.()

Faster convergence rate and estimation error bounds with respect to the L1- or

L2-norm can be achieved using the so-called oracle optimality condition:

$$\frac{1}{T}\sum_{t=1}^{T}\left(x_{t,N}\left(\widehat{\beta}^{LASSO}-\beta\right)\right)^2 = O_P\left(s_0\phi^{-2}\log\left(N\right)/T\right),$$

$$\sum_{i=1}^{N}\left|\widehat{\beta}_i^{LASSO}-\beta_i\right|^q = O_P\left(s_0^{1/q}\phi^{-2}\sqrt{\log\left(N\right)/T}\right),\ q=\{1,2\}, \quad (9)$$

where s_0 equals the true number of non-zero regression coefficients and ϕ^2 is the compatibility constant or restricted eigenvalue which is a number depending on the compatibility between the design and the L1-norm of the regression coefficient. The above rate is optimal up to the $\log\left(N\right)$ factor and the restricted eigenvalue ϕ^2.

Additionally to the oracle optimality and assuming the beta-min condition:

$$\min_{i\in S_0^c}|\beta_i| \gg \phi^{-2}\sqrt{s_0\log\left(N\right)/T},$$

we obtain the screening variable property:

$$P\left[\widehat{S}\supseteq S\right]\to 1 \quad \left(N\geq T\to\infty\right), \quad (10)$$

where $\widehat{S}=\{i;\widehat{\beta}_i^{LASSO}\neq 0, i=1,..,N\}$ and $S=\{i;\beta_i\neq 0, i=1,..,N\}$. Consistent variable selection then means

$$P\left[\widehat{S}=S\right]\to 1 \quad \left(N\geq T\to\infty\right). \quad (11)$$

3.3.3 Adaptive LASSO

Zou (2006) introduces the adaptive LASSO (A-LASSO) estimator where the L1-norms in the penalty are re-weighted. He shows that, if a reasonable initial estimator is available, under appropriate conditions, the A-LASSO correctly selects covariates with nonzero coefficients with probability converging to one, and that the estimators of nonzero coefficients have the same asymptotic distribution they would have if the zero coefficients were known in advance.

The optimisation problem now is:

$$\min_{\beta_N} \left\{ \sum_{t=1}^{T} \left(y_t - a - \beta_N' x_{t,N} \right)^2 + \lambda \sum_{i=1}^{N} \widehat{w}_i \left| \beta_i \right| \right\}, \tag{12}$$

where $\widehat{w}_i = 1/|\ \widehat{\beta}_{init,i}|^{\gamma}$, $\widehat{\beta}_{init}$ is an initial estimator and $\gamma > 0$. Usually, the initial estimator is the LASSO estimator with the constraint parameter tuned in the usual way with CV scheme as discussed earlier. Then, in the second stage CV is again used to select the λ parameter in Equation (12).

Following Haung et al. (2008) we consider the following conditions to hold for the variable selection and asymptotic normality of the A-LASSO in large samples.

1. The errors are iid.

2. The initial estimators $\widehat{\beta}_{init,i}$ are r_T-consistent for the estimation of certain η_{Ti}:

$$r_T \max_{i \leq N} \left| \widehat{\beta}_{init,i} - \eta_{T,i} \right| = O_P(1),\ r_T \to \infty$$

   where η_{Ti} are unknown constants depending on β_N and satisfy

$$\max_{i \notin J_{T1}} |\eta_{T,i}| \leq M_{T2},\ \left\{ \sum_{i \in J_{T1}} \left(\frac{1}{|\eta_{Ti}|} + \frac{M_{T2}}{|\eta_{Ti}|^2} \right)^2 \right\}^{1/2} \leq M_{T1} = o(r_T).$$

3. Adaptive irrepresentable condition. For $s_{T1} = \left(|\eta_{Ti}|^{-1} sgn(\beta_i), i \in J_{T1} \right)'$ and some $\kappa < 1$

$$\frac{1}{T} \left| x_i' X_1 \sum_{T11}^{-1} s_{T1} \right| \leq \frac{\kappa}{|\eta_{Ti}|}, \forall i \notin J_{T1}.$$

4. The constants $\{k_T, m_T, \lambda_T, M_{T1}, M_{T2}, b_{T1}\}$ satisfy

$$(\log T)^{I\{d=1\}} \left\{ \frac{(\log k_T)^{1/d}}{T^{1/2} b_{T1}} + (\log m_T)^{1/d} \frac{T^{1/2}}{\lambda_T} \left(M_{T2} + \frac{1}{r_T} \right) \right\} + \frac{M_{T1} \lambda_T}{b_{T1} T} \to 0.$$

5. There exists a constant $\tau_1 > 0$ such that $\tau_{T1} \geq \tau_1$ for all T.

Following Haung et al. (2008), Condition 1 is standard for variable selection in linear regression. Condition 2 assumes that the initial $\widehat{\beta}_{init,i}$ actually estimates some proxy $\eta_{T,i}$ of β_i so that the weights are not too large for $\beta_{0i} \neq 0$ and not too small for $\beta_{0i} = 0$. The adaptive irrepresentable condition becomes the strong irrepresentable condition for the sign-consistency of the Lasso if the $|\eta_{T,i}|$ are identical for all $i \leq N$. Condition 4 restricts the numbers of covariates with zero and nonzero coefficients, the penalty parameter, and the smallest non-zero coefficient. Condition 5 assumes that the eigenvalues of Σ_{T11} are bounded away from zero, which is reasonable since the number of nonzero covariates is small in a sparse model. If the above conditions hold, then $P\left[\widehat{\beta}^{A-LASSO} = \beta\right] \to 1$.

3.3.4 Elastic Net

Elastic Net (EN) creates a regression model that is penalised with both the L1-norm and L2-norm. Introduced by Zou and Hastie (2005), the elastic net has the effect of effectively shrinking coefficients (as in ridge regression) and setting some coefficients to zero (as in LASSO). The optimisation problem now is:

$$\widehat{\beta}^{naiveEN} = \min_{\beta_N} \left\{ \sum_{t=1}^{T} \left(y_t - a - \beta_N^{'} x_{t,N} \right)^2 + \lambda_1 \sum_{i=1}^{N} |\beta_i| + \lambda_2 \sum_{i=1}^{N} \beta_i^2 \right\}. \tag{13}$$

The above is called the naive elastic net. A correction which leads to the elastic net is then:

$$\widehat{\beta}^{EN} = (1 + \lambda_2)\, \widehat{\beta}^{naiveEN}.$$

The correction factor $(1 + \lambda_2)$ is best motivated from the orthonormal design where $\frac{1}{T} x_N^{'} x_N = I$. The main advantage of the elastic net is its usefulness when the number of predictors is much bigger than the number of observations, which is usually the case in our big data context.

The reason for adding an additional squared L2-norm penalty is motivated by Zou and Hastie (2005) as follows. For strongly correlated covariates, the LASSO may select one but typically not both of them (and the non-selected variable can then be approximated as a linear function of the selected one). From the point of view of

sparsity, this is what we would like to do. However, in terms of interpretation, we may want to have two even strongly correlated variables among the selected variables: this is motivated by the idea that we do not want to miss a "true" variable due to selection of a "non-true" which is highly correlated with the true one.

3.4 Factor Extraction via PCA

Another set of methods for modelling with a large panel of indicators involves the adoption of dimension reduction via techniques which do not require or impose any iid assumptions. In what follows we discuss the Principal Components Analysis (PCA) which has dominated the literature. This method is also frequently used for creating composite indicators (see Kapetanios and Papailias, 2021, and the references therein, for a recent discussion using the same dataset).

Factor methods have been at the forefront of developments in forecasting with large data sets and in fact started this literature with the influential work of Stock and Watson (2002a). The defining characteristic of most factor methods is that relatively few summaries of the large data sets are used in forecasting equations, which thereby become standard forecasting equations as they only involve a few explanatory variables.

The main assumption is that the co-movements across the indicator variables x_t, where $x_t = (x_{1t} \cdots x_{Nt})'$ is a vector of dimension $N \times 1$, can be captured by a $r \times 1$ vector of unobserved factors $F_t = (F_{1t} \cdots F_{rt})'$:

$$\tilde{x}_t = \Lambda' F_t + e_t \tag{14}$$

where $\tilde{x}_t$ may be equal to x_t or may involve other variables, such as lags, leads or products of the elements of x_t, and Λ is an $r \times N$ matrix of parameters describing how the individual indicator variables relate to each of the r factors, which we denote with the terms 'loadings'. In Equation (14) e_t is a zero-mean $I(0)$ vector of errors that represent, for each indicator variable, the fraction of dynamics unexplained by F_t, the 'idiosyncratic components'. The number of factors is assumed to be finite. So, implicitly, in Equation (3) $\alpha' = \tilde{\alpha}' \Lambda \tilde{x}_t$, where $F_t = \Lambda \tilde{x}_t$, which means that a

small, r, number of linear combinations of $\tilde{x_t}$ represent the factors and act as the predictors for y_t, the target variable. The main difference between different factor methods relates to how Λ and the factors are estimated.

The use of PCA for the estimation of factor models is, by far, the most popular factor extraction method. It has been popularised by Stock and Watson (2002a,b), in the context of large data sets, although the idea had been well established in the traditional multivariate statistical literature. The method of principal components is simple. Estimates of Λ and the factors F_t are obtained by solving:

$$V(r) = \min_{\Lambda,F} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} (\tilde{x}_{it} - \lambda_i' F_t)^2, \tag{15}$$

where λ_i is an $r \times 1$ vector of loadings that represent the N columns of $\Lambda = (\lambda_1 \cdots \lambda_N)$. One, non-unique, solution of Equation (15) can be found by taking the eigenvectors corresponding to the r largest eigenvalues of the second moment matrix $X'X$, which then are assumed to represent the rows in Λ, and the resulting estimate of Λ provides the forecaster with an estimate of the r factors $\hat{F}_t = \hat{\Lambda}\tilde{x}_t$. To identify the factors up to a rotation, the data are usually normalised to have zero mean and unit variance prior to the application of principal components; see Stock and Watson (2002a) and Bai (2003). We note that factor estimates obtained via PCA estimation are $\min(\sqrt{N}, T)$-consistent. Further, if $\sqrt{T}/N = o(1)$, using estimated factors rather than true factors in predictive regressions produces negligible estimation errors. PCA estimation of the factor structure is essentially a static exercise as no lags or leads of x_t are considered.

3.5 Random Forests

In the above sections, we have reviewed methods based on the specification of a parametric model, typically a linear regression, which links the target variable y with a, possibly big, number of explanatory variables x. Regression trees are based on a partition of the space of the dependent variable y into M subsets R_m, with y allocated to each subset according to a given rule and modelled as a different constant c_m in each subset. This is a powerful idea, since it can fit various functional

relationship between y and a set of explanatory variables x, say $y = f(x)$, without imposing linearity or additivity, which are commonly assumed in standard linear regression models. Let

$$y = f(x) = \sum_{m=1}^{M} c_m \mathbf{1}\left(x \in R_m\right),$$

where **1** denotes the indicator variable taking value 1 if the condition is satisfied, 0 otherwise. Then, given a partition, minimising:

$$\|y - f(y)\|_2 = \sum_{i=1}^{N} \left(y_i - f(y_i)\right)^2, \tag{16}$$

with respect to c_m yields $\hat{c}_m = \overline{y}_m$, where $\overline{y}_m$ denotes the sample mean of y over each region R_m.

A much more difficult problem is to find the best partition in terms of minimum sum of squares. Even in the two dimensional case, i.e when $N = 2$ so that $X = [x_1, x_2]$, finding the best binary partition to minimise the sum of squares is not computationally feasible. Instead, greedy algorithms are commonly used. The idea is to do one split at a time. Consider a splitting variable j (where $j = 1, .., k$) and a splitpoint s such that a region $R_1(j, s)$ is defined as

$$R_1(j, s) = \{X | X_j \leq s\} \text{ and } R_2(j, s) = \{X | X_j > s\}$$

Then, the sum of squares) is minimised with respect to j and s. For each splitting variable, the split point s can be found and, hence, by scanning through all of the variables X_j, determination of the best pair (j, s) is feasible. Having found the best split, the data are partitioned into **two** resulting regions and the same splitting exercise is repeated on each of the two regions. Then this process is repeated on all of the resulting regions and so on. How many rounds of the algorithm are done determines how deep the resulting tree is. On one hand, shallow trees might fail to capture the structure of the data. On the other hand, however, deeper trees might

overfit the data and, hence, do poorly in prediction.

There are ways to further improve the performance of regression trees. Bootstrap requires choosing with replacement a subsample and re-estimating the tree in order to get a sampling distribution of various statistics. Bagging is a general method that generates multiple versions of a predictor and uses these to get an aggregated predictor. Breiman (1996) offers an overview. In the context of regression trees, bagging averages across trees is estimated with different bootstrapped samples.

Random forests were introduced by Breiman (2001). The idea is exactly as bagging applied on regression trees: to grow a large collection of de-correlated trees (hence, the name forest) and then average them. This is achieved by bootstrapping a random sample at each node of every tree. In order to induce "decorrelation" of trees, when growing trees, before each split, select a subset of the input variables at random as candidates for splitting. This prevents the "strong" predictors imposing too much structure on the trunk of the tree.

3.6 Neural Networks

We continue with machine learning-based nonlinear methodologies and include two artificial neural network (ANN)-based approaches: (i) the Multilayer Perceptron (MLP) neural networks, and the (ii) Extreme learning machines (ELM) neural networks. Both MLP and ELM are feedforward artificial neural network with -at least- three layers of nodes: an input layer, a hidden layer (or multi-layers) and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. Both MLP and ELM are supervised machine learning approaches. In the following subsections, we provide some introduction to these approaches discussing their advantages and disadvantages. However, as it is beyond the scope of this task, we refer the reader to Ord et al. (2017) for more theoretical details on neural networks.

Figure 1 and Figure 2 provide two examples of the discovered layers using MPL and EML. Figure 1 is the estimation on the first in-sample date (2020-01-26), whereas Figure 2 is the estimation on the last in-sample date (2021-10-24). Both MPL and

EML use the sigmoid functions in the neurons. Learning occurs in both methods by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning and is usually carried out through backpropagation. However, in the two neural networks incarnations we apply here we use the lasso approach instead of backpropagation.

3.6.1 Multilayer Perceptron

MLP is one of different kinds of ANNs. The term MLP can be used to loosely describe any feedforward ANN, sometimes strictly referring to networks composed of multiple layers of perceptrons (with threshold activation). MLPs with a single hidden layer are usually called "vanilla" neural networks; this is also the case in our empirical results later.

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model. In MLPs we can have some neurons to use a non-linear activation function.

The most commonly used form of ANNs for forecasting is the feedforward multilayer perceptron. The one-step ahead forecast $\hat{y}_{t+1}$ is computed using inputs that are lagged observations of the time series or other explanatory variables. I denotes the number of inputs p_i of the ANN. Its functional form can be described as:

$$\hat{y}_{t+1} = \alpha + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=1}^{I} \gamma_{hi} p_i\right). \tag{17}$$

In the equation above, $\boldsymbol{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ are the network weights with $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_H]$ and $\gamma = [\gamma_{11}, \ldots, \gamma_{HI}]$ being the output and the hidden layers respectively. α and γ_{0i} are the biases of each neuron, which for each neuron act similarly to the intercept in a regression. H is the number of hidden nodes in the network and $g(\cdot)$ is a non-linear transfer function which is usually either the sigmoid logistic or the hyperbolic

tangent function. ANNs can model interactions between inputs, if any. The outputs of the hidden nodes are connected to an output node that produces the forecast.

In the time series forecasting context, neural networks can be perceived as equivalent to nonlinear autoregressive models. Lags of the time series, potentially together with lagged observations of explanatory variables, are used as inputs to the network. During training, pairs of input vectors and targets are presented to the network. The network output is compared to the target and the resulting error is used to update the network weights. ANN training is a complex nonlinear optimisation problem and the network can often get trapped in local minima of the error surface. In order to avoid poor quality results, training should be initialised several times with different random starting weights and biases to explore the error surface more fully.

Note that the objective of training is not to identify the global optimum. This would result in the model over-fitting to the training sample and would then generalise poorly to unseen data, in particular given their powerful approximation capabilities. Furthermore, as new data becomes available, the prior global optimum may no longer be an optimum. In general, as the fitting sample changes, with the availability of new information, so do the final weights of the trained networks, even if the initial values of the network weights were kept constant. This sampling-induced uncertainty can again be countered by using ensembles of models, following the concept of bagging.

Advantages

1. MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

2. MLPs are universal function approximators as shown by Cybenko's theorem, so they can be used to create mathematical models by regression analysis. As classification is a particular case of regression when the response variable is categorical, MLPs make good classifier algorithms.

3. MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from much simpler (and related) support vector machines.

Disadvantages

1. Computations are difficult and often time consuming.
2. The proper functioning of the model depends on the quality of the training.

3.6.2 Extreme Learning Machines

ELMs are feedforward neural networks for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes need not be tuned. These hidden nodes can be randomly assigned and never updated, i.e. they are random projections but with nonlinear transforms, or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model.

Given a training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \ldots, N\}$, activation function $g(x)$, and hidden node number $\tilde{N}$, we can describe the ELM generic algorithm as follows.

Step 1 : Randomly assign input weight $\mathbf{w}_i$ and bias b_i, $i = 1, \ldots, \tilde{N}$,

Step 2 : Calculate the hidden layer output matrix $\mathbf{H}$,

Step 3 : Calculate the output weight β defined as:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}, \tag{18}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}}$. In principle, this algorithm works for any infinitely differential activation function $g(x)$. Such activation functions include the sigmoidal

functions as well as the radial basis, sine, cosine, exponential, and many non-regular functions. The upper bound of the required number of hidden nodes is the number of distinct training samples, that is $\tilde{N} \leqslant N$.

Advantages & Disadvantages. The black-box character of neural networks, in general, and ELMs in particular is one of the major concerns that repels engineers from application in unsafe automation tasks. This particular issue was approached by means of several different techniques. One approach is to reduce the dependence on the random input. Another approach focuses on the incorporation of continuous constraints into the learning process of ELMs which are derived from prior knowledge about the specific task. This is reasonable, because machine learning solutions have to guarantee a safe operation in many application domains. The mentioned studies revealed that the special form of ELMs, with its functional separation and the linear read-out weights, is particularly well suited for the efficient incorporation of continuous constraints in predefined regions of the input space.

3.7 Support Vector Regression

The final methodology we include in this report the Support Vector Regression (SVR). It can be argued that SVR is the adapted form of Support Vector Machines when the dependent variable is numerical rather than categorical. A major benefit of using SVR is that it is a non-parametric technique. Unlike the classical linear regression which depends on the Gauss-Markov theorem, the output model from SVR does not depend on distributions of the underlying dependent and independent variables. Instead the SVR technique depends on the kernel function which is employed to captured the underlying possibly non-linear relationships.

The success of the SVR is based on the fit of the kernel functions in the data. The most popular kernel functions include: (i) the linear, (ii) the polynomial, (iii) the sigmoid, and (iv) the radial basis function. It is important to highlight that different kernel functions result to different models and, therefore, different estimations and subsequent nowcasts. However, it is beyond the scope of the current report to provide

an extensive discussion on the advantages and disadvantages of each kernel function. In what follows, we stick to the radial basis function.

The kernel allows the SVR to find a fit and then the data is mapped to the original space. In our setup, SVR works as simple non-linear model described as:

$$y_t = WK(x_t, x) + b, \tag{19}$$

where W is the product of the coefficients and the resulting support vectors and b is the negative intercept. We refer the reader to Chang and Lin (2021) Python implementation and references therein.

— The final methodology we include in this report is the Support Vector Regression (SVR). It can be argued that SVR is the adapted form of Support Vector Machines when the dependent variable is numerical rather than categorical. A major benefit of using SVR is that it is a non-parametric technique. Unlike the classical linear regression which depends on the Gauss-Markov theorem, the output model from SVR does not depend on distributions of the underlying dependent and independent variables. Another advantage of SVR is that it permits for construction of a non-linear model without changing the explanatory variables, helping in better interpretation of the resultant model.

The basic idea behind SVR is not to care about the prediction as long as the error (ϵ) is less than certain value. This is known as the principle of maximal margin. This idea of maximal margin allows viewing SVR as a convex optimization problem. The regression can also be penalized using a cost parameter, which becomes handy to avoid over-fit. SVR is a useful technique provides the user with high flexibility in terms of distribution of underlying variables, relationship between independent and dependent variables and the control on the penalty term.

SVR gives the flexibility to define how much error is acceptable in a model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data. In contrast to OLS, the objective function of SVR is to minimise the coefficients — more specifically, the l_2-norm of the coefficient vector — not the squared error. The error term is instead handled in the constraints, where we set the absolute error less

than or equal to a specified margin, called the maximum error, ϵ. ϵ can be tuned to gain the desired accuracy in a model.

The SVR technique depends on the kernel function which is employed to captured the underlying possibly non-linear relationships. The success of the SVR is based on the fit of the kernel functions in the data. The most popular kernel functions include: (i) the linear, (ii) the polynomial, (iii) the sigmoid, and (iv) the radial basis function (RBF). It is important to highlight that different kernel functions result to different models and, therefore, different estimations and subsequent nowcasts. However, it is beyond the scope of the current report to provide an extensive discussion on the advantages and disadvantages of each kernel function. In what follows, we stick to the radial basis function.

The kernel allows the SVR to find a fit and then the data is mapped to the original space. In our setup, SVR works as simple non-linear model described as:

$$y_t = WK(x_t, x) + b, \tag{20}$$

where W is the product of the coefficients and the resulting support vectors and b is the negative intercept. We refer the reader to Chang and Lin (2021) Python implementation and references therein. In machine learning, the radial basis function kernel is commonly used in support vector machine classification.

Following Vert et al. (2004), the RBF kernel on two samples $\mathbf{x}$ and $\mathbf{x}'$, represented as feature vectors in some input space, is defined as:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{x}'\right\|^2}{2\sigma^2}\right) \tag{21}$$

where $\left\|\mathbf{x} - \mathbf{x}'\right\|^2$ may be recognized as the squared Euclidean distance between the two feature vectors. σ is a free parameter. An equivalent definition involves a parameter $\gamma = \frac{1}{2\sigma^2}$:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}'\right\|^2\right). \tag{22}$$

Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when $\mathbf{x} = \mathbf{x}'$), it has a ready interpretation as a similarity measure. The feature space of the kernel has an infinite number of dimensions; for $\sigma = 1$, its expansion is:

$$\begin{aligned}
\exp\left(-\frac{1}{2}\left\|\mathbf{x}-\mathbf{x}'\right\|^2\right) &= \exp\left(\frac{2}{2}\mathbf{x}^\top\mathbf{x}' - \frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\left\|\mathbf{x}'\right\|^2\right) \\
&= \exp\left(\mathbf{x}^\top\mathbf{x}'\right)\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\exp\left(-\frac{1}{2}\left\|\mathbf{x}'\right\|^2\right) \\
&= \sum_{j=0}^{\infty}\frac{\left(\mathbf{x}^\top\mathbf{x}'\right)^j}{j!}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\exp\left(-\frac{1}{2}\left\|\mathbf{x}'\right\|^2\right) \\
&= \sum_{j=0}^{\infty}\sum_{\sum n_i=j}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\frac{x_1^{n_1}\cdots x_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\exp\left(-\frac{1}{2}\left\|\mathbf{x}'\right\|^2\right)\frac{x_1'^{n_1}\cdots x_k'^{n_k}}{\sqrt{n_1!\cdots n_k!}}.
\end{aligned} \tag{23}$$

Because support vector machines and other models employing the kernel trick do not scale well to large numbers of training samples or large numbers of features in the input space, several approximations to the RBF kernel (and similar kernels) have been introduced. Typically, these take the form of a function z that maps a single vector to a vector of higher dimensionality, approximating the kernel:

$$\left\langle z(\mathbf{x}), z\left(\mathbf{x}'\right)\right\rangle \approx \left\langle \varphi(\mathbf{x}), \varphi\left(\mathbf{x}'\right)\right\rangle = K\left(\mathbf{x}, \mathbf{x}'\right) \tag{24}$$

where φ is the implicit mapping embedded in the RBF kernel.

One way to construct such a z is to randomly sample from the Fourier transformation of the kernel as noted in Rahimi et al. (2007). Another approach uses the Nyström method to approximate the eigendecomposition of the Gram matrix K, using only a random sample of the training set as given in Williams and Seeger (2001).

4 Nowcasting Setup

In this section, we describe the setup of the out-of-sample pseudo cross-validation nowcasting exercise. In what follows, x_t denotes the vector of all available predictors.

4.1 Algorithm

Our nowcasting algorithm is structured in the following way.

1. Set the out-of-sample nowcasting dates, i.e. the weekly dates that nowcasting estimates for y_t will be produced. In our exercise, we have 92 weekly nowcast dates spanning from 2021-03-22 to 2021-12-19.

2. For each given out-of-sample date, t, collect the available data for y_t and x_t and mimic the pattern of availability by imposing missing values according to the corresponding .xlsx file. It is important to notice here that we use the weekly disaggregated version of y_t to match the weekly frequency of all x_t.

3. Once the pseudo-available data is ready, fix any potential seasonalities.

4. Transform all variables to stationarity.

5. Remove outliers.

6. Impute missing values (where applicable).

7. Assuming that y_t has k missing values at the bottom, which we actually need to nowcast, estimate the linear regression model of the generic type:

$$y_{t-k} = \alpha + \beta x_{t-k} + \varepsilon_t, \tag{25}$$

   obtain $\widehat{\alpha}$ and $\widehat{\beta}$ and produce the nowcast estimate as: $\widehat{y}_t = \widehat{\alpha} + \widehat{\beta} x_t$.

8. Repeat Steps 2 to 8 for all out-of-sample dates recursively (i.e. increasing the sample size). By the end of the nowcasting exercise, we obtain estimates for the

61out-of-sample weeks with models coming from the methodologies discussed in the previous section.

It is important to notice that the above nowcasting takes place in the weekly frequency using the disaggregated version of y_t. Nowcast errors can be computed at the weekly frequency or, more appropriately, weekly nowcasts can be aggregated to monthly and nowcast error can be computed using the monthly nowcasts and the *actual* monthly observations for y_t rather than the weekly approximates. In this report, we produce nowcast error statistics appropriately in the monthly frequency and end up with 20 out-of-sample dates from 2020-03-31 to 2021-10-31.

4.2 Evaluation

The purpose of this framework is to utilise nowcasting regressions and compare linear models using the main (Level I) aggregates versus models which are based on the disaggregate variables (Level II disaggregates). To compare the nowcasting performance we report standard nowcasting error statistics, Mean Absolute Nowcast Error (MAE), Mean Squared Nowcast Error (MSE) and the Root Mean Squared Error (RMSE) evaluated for Model M across the out-of-sample nowcasts as:

$$MAE_M = \frac{1}{T_{out}} \sum_{l=1}^{T_{out}} |e_M| \,, \tag{26}$$

$$MSE_M = \frac{1}{T_{out}} \sum_{l=1}^{T_{out}} e_M^2, \tag{27}$$

$$RMSE_M = \sqrt{MSE_M}, \tag{28}$$

where e_M is the vector of T_{out} nowcast errors for Model M. We report both the above statistics relative to a standard AR(1) benchmark, i.e. values smaller than 1 indicate predictive gains of the given model against the benchmark. This also allows the cross-comparison across models.

4.3 Models

In the out-of-sample nowcasting exercise, we consider the following models.

- Univariate time series benchmark models: **AR(1)** and **AR(P)** where $\widehat{P}$ is chosen minimising AIC.
- Linear models using the Level I **aggregate** in each case appropriately.
- **BFW** selecting variables from the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **Ridge** regression using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **Lasso** regression using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **EN** regression using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates; the mixing parameter is set to $\alpha = 0.5$ for illustration, but more choices could be considered.
- **Ad. Lasso, V1**: Adaptive Lasso regressions using the Ridge weights in the penalty (Ad. Lasso, V1).
- **Ad. Lasso, V2**: Adaptive Lasso regressions using the Lasso weights in the penalty.
- Linear regression using **PCA(1)** factor from the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **PCA(A1)**: Linear regression using PCA($\widehat{k}_1$) factor using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates; $\widehat{k}_1$ is chosen via cross-validation.

- **PCA(A2)**Linear regression using PCA($\widehat{k}_2$) factor using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates; $\widehat{k}_2$ is chosen in a fashion similar to Bai (2003).
- **Random Forests** using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **MLP** using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **ELM** using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.
- **SVR** using the Online Job Advertisements disaggregates, the Traffic in Ports disaggregates and across all disaggregates.

It is important to highlight that our main purpose is not to identify the best methodology or model suitable for nowcasting. Instead, the suggested framework utilises various models to provide robust evidence on the predictive content of the specific set of indicators considered and conclude if the detailed series of each dataset are more useful than the main aggregate.

5 Empirical Results

5.1 Limitations

This section is concerned with a brief discussion of the main results. Table 1 to Table 3 are concerned with the MAE and RMSE results for the models discussed in the previous sections across two periods: (i) the full sample which spans from 2020-03-31 to 2021-10-31 (20 obs.), and (ii) a subsample which spans from 2020-11-30 to 2021-10-31 (12 obs.). Individual figures with the nowcasts from each model are included in the Appendix.

First, it is important to highlight that the traffic in ports dataset is extremely short with the first data being published in 2019-04-01. This really limits the whole

nowcasting exercise which needs some sufficient data to be available for the first estimation and, at the same time, some sufficient data to be kept separately for the out-of-sample cross-validation. Even in the case of weekly frequency, a dataset which spans from 2019-04-01 to 2021-10-31 allows for about 124 weeks. Some part of this dataset needs to be used for the first estimation and the remaining to be available in the cross-validation exercise. One could use the first 50 weekly observations for the first estimation. This allows for about 74 weeks in the out-of-sample cross-validation. Since our target variable is monthly, this translates to about 18 months in the out-of-sample. Part of this problem is solved with the recursive estimation which leads to more improved estimates but only for the latter part of the sample.

Second, the limited out-of-sample dates also include the COVID-19 outbreak which is a shock to the economic system making the target variable to be locally nonstationary and many models to misbehave. Ideally, in case of ample data available, one could run a proper nowcasting exercise for a number of years prior to the COVID-19 pandemic and evaluate the nowcasting power of various models. Then, she could repeat the exercise including the COVID-19 outbreak and measure the shock to the system using the nowcast error. However, since we do not have data available for many years prior to the COVID-19 pandemic, we also report the statistics for a subsample which spans from one year, from 2020-11-30 to 2021-10-31 (12 obs.), to offer some insights on how different models perform in a period which does not have an economic shock; even though this period is still included in the estimation.

Third, as already discussed in the Introduction, this report is concerned with a very specific task: to provide a head-to-head comparison of a (linear) model which uses the total aggregate (Level I) of a dataset versus various models (linear and nonlinear) which attempt to exploit the information based on the disaggregate series (Level II) of the same dataset. Ideally, to allow models to "learn" and take advantage of possibly repeating patterns in the time and cross-sectional dimension, the researcher would like to have access to the most disaggregated version of the dataset possible. For example, if the underlying dataset is the Business Insights and Impact

on the UK Economy (BICS)[9] dataset, one would ideally need access to all individual UK businesses which have been interviewed across all waves. Assuming that about half of the 38,000 UK businesses have been interviewed consistently across time and have answered to about 50 questions[10], this gives about 700,000 disaggregate series to be evaluated. Due to data availability issues, we do not have access to extremely big datasets and, therefore, this report is based on two sets from the ONS Real-Time Indicators data comparing the total aggregates to their first level disaggregates. This is not close to what we describe above, however it can be used as a proof-of-concept. It is also important to say that the models based on the machine learning methodologies we discuss here can also be used in cases with extremely large datasets.

5.2 Online Job Advertisements

Starting with the case of the Online Job Advertisements, we report the nowcasting results in Table 1. This table is divided in two panels: (i) the left panel is concerned with the results evaluated in the full sample which includes the COVID-19 outbreak during the months of March to May 2020, whereas the (ii) right panel is concerned with the results evaluated after November 2020 which excludes the COVID-19 outbreak.

Starting with the full sample results, we see that a linear model using the Level I aggregate, that is the Online Job Advertisements Index across All UK Industries, already improves against the simple univariate AR(1) and AR(P) benchmarks. In particular, the Aggregate linear model has a relative MAE of 0.864 and RMSE of 0.913; this already provides evidence that this dataset, proxied via its total index, is useful in economic nowcasting. The question is to identify if there is more information in the Level II disaggregates and further justify the publication of the whole dataset.

We see that all the penalised regression variants have MAE and RMSE smaller than the univariate benchmarks and also slightly smaller than the Aggregate model. For example, we see that using Ad. Lasso V2 we obtain a relative MAE of 0.834

[9] See this ONS page.

[10] This is just a working example. Visiting the link in the previous footnote, the reader can find that BICS dataset includes more than 100 questions.

and RMSE of 0.904 which corresponds to a 3.4% and 1% improvement compared to the Aggregate model. Then, PCA(A1) provides a reduction in the nowcast error of about 3.2% in MAE and 1.2% in RMSE compared to the Aggregate model. In the non-linear machine learning models, we see that Random Forests model provides a reduction in the nowcast error of about 3.7% in MAE and 1.2% in RMSE compared to the Aggregate model.

Moving to the subsample results which excludes the COVID-19 outbreak period, we see that the nowcast error in the Aggregate model increases, however all penalised regressions, PCA(1), PCA(A1) and SVR have improved performance when compared to their performance in the full sample case. On the other hand, Random Forests, MLP and ELM have an increased nowcast error. This is evidence that in the post-COVID-19 outbreak period, the relationship becomes linear which is best captured by most of the linear models. Ad. Lasso, V2 has the best average performance with a relative MAE and RMSE of 0.752 and 0.776 which corresponds to about 18.25% reduction compared to the nowcast error of the Aggregate model.

The above results highlight that, in stable periods, exploiting the information hidden in the disaggregates leads to larger improved in nowcast error. This argument still holds when crisis periods are included in the evaluation sample, however the magnitude of the reduction in the error is much smaller.

5.3 Traffic in Ports

Then, we move to Table 2 which reports the results based on the weekly shipping indicators which measure the traffic in UK ports. As above, the table is divided in two panels: (i) the left panel is concerned with the results evaluated in the full sample which includes the COVID-19 outbreak, whereas the (ii) right panel is concerned with the results evaluated after November 2020 which excludes the COVID-19 outbreak. In this case, we include two aggregates: (i) Aggregate1 which is the extracted trend of the traffic in ports, and (ii) the seasonally adjusted number of visits by ships across all UK ports. The results indicate minor differences in favour of the second aggregate which makes most sense as it reflects all the information. Therefore, in the following

discussion we adopt Aggregate2 as our main aggregate model benchmark.

Looking at the full sample results, we again see that Aggregate2 model improves against the standard univariate benchmarks with a relative MAE and RMSE of 0.897 and 0.921 respectively. As in the previous case, this already indicates that this type of dataset has economic value. Now, do the disaggregates in this case really help in nowcasting?

At first, we see that all models using the Level II disaggregates have a MAE and RMSE smaller that unity which indicates that they all improve against the univariate benchmarks. Looking at the penalised regressions, we observe minor improvements against the Aggregate2 model. Ad. Lasso, V1 has a MAE and RMSE of 0.890 and 0.920 respectively which correspond to 0.29% and 0% improvements against the Aggregate2 model. PCA-based models also have similar performance. Random Forests model and SVR seem to provide a slightly larger reduction in the nowcast error offering a MAE of 0.888 and 0.872 and RMSE of 0.917 and 0.915 respectively. This is evidence of possibly non-linear underlying relationships which cannot be effectively captured by the previously-mentioned linear models. SVR returns a 2.79% reduction in MAE and 0.71% reduction in RMSE.

Turning to the subsample case, we see that almost all models have an increased error as both MAE and RMSE increases compared to the full sample case. This could indicate that this dataset has information which can be more useful in times of crisis.

5.4 Online Job Advertisements & Traffic in Ports

Finally, in Table 3 we report the results using both the Online Job Advertisements and the Shipping indicators from the ONS Real-Time Indicators dataset. The aim of this exercise is to have a larger number of disaggregate indicators available which could offer more insights and improve the nowcasting exercise. As in the cases described above, the table is divided in two panels: (i) the left panel is concerned with the results evaluated in the full sample which includes the COVID-19 outbreak, whereas the (ii) right panel is concerned with the results evaluated after November

2020 which excludes the COVID-19 outbreak.

The All Aggregates model is a linear model with the three Level I aggregate predictors: (i) the Online Job Advertisements Index across All UK Industries, (ii) the extracted trend of the traffic in ports, and (iii) the seasonally adjusted number of visits by ships across all UK ports. As we see in Table 3, the linear model with All Aggregates improves compared to the univariate benchmarks which, together with the individual results described in the previous two subsections, highlights that both this datasets provide gains in nowcasting even if we simply use their total aggregates.

The BFW model is the best model in the full sample case with 0.831 MAE and 0.900 RMSE which improves the aggregates benchmark model by 4.38% and 1.44% respectively. All other models, apart from MPL and ELM, improve over the aggregates model with improvements ranging from 0.22% to 3.32% in terms of MAE and 0.13% to 0.97% in terms of RMSE.

Turning to the subsample case, we see that the error in the aggregates model increases. This is also the case with most PCA and non-linear models. Penalised regressions, on the other hand side, provide improved results with BFW being the best model with 0.766 MAE and 0.788 RMSE which corresponds to 17.93% reduction in the error in terms of MAE and 18.91% reduction in terms of RMSE. This is followed by Ad. Lasso, V1 which has a 15.24% and 15.74% reduction of error in terms of MAE and RMSE respectively.

The above combined case of both datasets again illustrates that the disaggregates need to be considered together with the main aggregates and ONS should continue publishing the datasets in this detail (if not in more disaggregate level).

6 Concluding Remarks

This report has a very specific task: to examine gains in nowcasting comparing a linear regression model using a single aggregate variable with models which utilise all the underlying disaggregate series. At first, this might seem like a task with a limited scope. However, the applied research should view this approach as only a part of an overall assessment framework for novel datasets. Our main aim is to provide a

framework which answers the following question regarding a candidate dataset of new indicators: "Should a national statistics institute invest resources in organising, editing, polishing and publishing this novel dataset of indicators and why?".

Using two datasets from the ONS Real-Time Indicators we attempt to answer this question empirically via means of economic nowcasting. In particular, we consider a linear model which uses the main aggregates and compare its nowcasting performance with various, mainly machine learning-based, models which utilise all the underlying disaggregate series. In this report linear (penalised regressions and actor-based regressions) as well as non-linear (random forests, neural networks and support vector regressions) models are included.

Our findings provide empirical evidence in favour of the "big data" principle; i.e. in today's world, national statistics institutes should publish data to some, if not the highest possible, level of disaggregation as most modern econometric techniques can handle these datasets and exploit their gains in economic applications such as nowcasting or forecasting. As expected, our results show that during crises, such as the COVID-19 outbreak, non-linear models tend to perform better than linear ones, however this reverts in periods of economic stability.

7 References

1. Bai, J., 2003. Inferential Theory for Factor Models of Large Dimension. Econometrica, 71, 135-173.

2. Boot, J.C.G., Feibes, W., Lisman, J.H., 1967. Further Methods of Derivation of Quarterly Figures from Annual Data. Applied Statistics, 16(1), 65-75.

3. Breiman, L., 1996. Bagging Predictors. Machine Learning, 24(2), 123–140.

4. Breiman, L., 2001. Random Forests. Machine Learning, 45(1), 5-32.

5. Bühlmann, P., van de Geer, S., 2011. Statistics for High-Dimensional Data: Methods, Theory and Applications. Springer.

6. Chang, C.-C., Lin, C.-J., 2021. LIBSVM – A Library for Support Vector Machines. Version 3.25, released on April 14, 2021.

7. Estrella, A., Hardouvelis, G.A., 1991. The Term Structure as a Predictor of Real Economic Activity. The Journal of Finance, 46(2), 555-576.

8. Haung, J., Ma, S., Zhang, C.H., 2008. Adaptive LASSO for Sparse High Dimensional Regression Models. Statistica Sinica, 18, 1603-1618.

9. Kapetanios G., Papailias F., 2021a. UK Economic Conditions during the Pandemic: Assessing the Economy using ONS Faster Indicators. ESCoE Working Paper, DP 2021-10. Available here.

10. Kapetanios G., Papailias F., 2021b. Faster Indicators during the COVID-19 Pandemic: Individual Predictors & Selection. ESCoE Technical Report.

11. Kapetanios G., Papailias F., 2021c. An Evaluation Framework for Targeted Indicators, Part I: Monthly VAT. ESCoE Technical Report.

12. Kapetanios G., Papailias F., 2021d. An Evaluation Framework for Targeted Indicators, Part II: Weekly CHAPS. ESCoE Technical Report.

13. Marcellino, M., Stock, J.H., Watson, M.W., 2006. A Comparison of Direct and Iterated AR Methods for Forecasting Macroeconomic Series h-Steps Ahead. Journal of Econometrics, 135, 499-526.

14. Ord, K., Fildes, R., Kourentzes, N. 2017. Principles of Business Forecasting, 2e. Wessex Press Publishing Co., Chapter 10.

15. Rahimi, A., Recht, B., 2007. Random Features for Large-Scale Kernel Machines. NIPS '07: Proceedings of the 20th International Conference on Neural Information Processing Systems. December, 2007, 1177–1184.

16. Santos Silva, J.M.C., Cardoso, F.N., 2001. The Chow-Lin method using dynamic models. Economomic Modelling, 18, 269-280.

17. Stock, J.H., Watson, M.W., 2002a. Forecasting using Principal Components from a Large Number of Predictors. Journal of the American Statistical Association, 97, 147-162.

18. Stock, J.H., Watson, M.W., 2002b. Macroeconomic Forecasting using Diffusion Indexes. Journal of Business and Economic Statistics, 20, 147-162.

19. Tibshirani, R., 1996. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society: Series B, 58(1), 267-288.

20. Vert, J.P., Tsuda, K., Scholkopf, B., 2004. Kernel Methods in Computational Biology. MIT Press.

21. Williams, C.K.I., Seeger, M., 2001. Using the Nyström Method to Speed Up Kernel Machines. Advances in Neural Information Processing Systems 13 (NIPS 2000), MIT Press, 682-688.

22. Wetshoreck, F., 2020. RIP correlation: Introducing the Predictive Power Score. Towards Data Science, Apr. 23, 2020. `https://bit.ly/3cTi1w6`.

23. Zou, H., 2006. The Adaptive Lasso and Its Oracle Properties. Journal of the American Statistical Association, 101(746), 1418-1429.

24. Zou, H., Hastie, T., 2005. Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society: Series B, 67(2), 301-320.

Tables

	Full Sample		Subsample	
	MAE	**RMSE**	**MAE**	**RMSE**
AR(1)	1	1	1	1
AR(P)	0.999	1	0.998	1
Aggregate	0.864	0.913	0.920	0.961
BFW	0.835	0.902	0.778	0.797
Ridge	0.843	0.908	0.822	0.866
Lasso	0.841	0.906	0.788	0.823
EN	0.842	0.906	0.796	0.831
Ad. Lasso, V1	0.837	0.904	0.770	0.800
Ad. Lasso, V2	0.834	0.904	0.752	0.786
PCA(1)	0.860	0.911	0.834	0.909
PCA(A1)	0.836	0.902	0.916	0.957
PCA(A2)	0.858	0.908	0.815	0.867
Random Forests	0.831	0.902	0.932	0.982
MLP	1.068	1.018	1.221	1.165
ELM	0.862	0.909	0.877	0.915
SVR	0.839	0.905	0.835	0.889

Table 1: Nowcast MAE and RMSE using the Online Job Ads. Values correspond to statistics relative to the AR(1) benchmark. The Aggegate is Online Job Ads Index across all industries. The Full Sample case spans from 2020-03-31 to 2021-10-31 (20 obs.) and includes the COVID-19 outbreak (March to May, 2020) in the evaluation. The Subsample case spans from 2020-11-30 to 2021-10-31 (12 obs.).

	Full Sample		Subsample	
	MAE	**RMSE**	**MAE**	**RMSE**
AR(1)	1	1	1	1
AR(P)	0.999	1	0.998	1
Aggregate1	0.899	0.921	1.000	1.004
Aggregate2	0.897	0.921	0.991	0.994
BFW	0.890	0.922	0.972	0.984
Ridge	0.894	0.921	0.979	0.985
Lasso	0.894	0.921	0.978	0.984
EN	0.893	0.921	0.978	0.984
Ad. Lasso, V1	0.890	0.920	0.968	0.979
Ad. Lasso, V2	0.894	0.921	0.977	0.984
PCA(1)	0.895	0.920	0.979	1.005
PCA(A1)	0.894	0.922	0.991	0.993
PCA(A2)	0.895	0.921	0.996	0.996
Random Forests	0.888	0.917	1.001	1.002
MLP	0.972	0.984	0.968	0.982
ELM	0.910	0.926	0.929	0.943
SVR	0.872	0.915	0.949	1.003

Table 2: Nowcast MAE and RMSE using the traffic at UK ports. Values correspond to statistics relative to the AR(1) benchmark. Aggregate1 refers to the extracted trend of the total number of ships visiting UK ports. Aggregate2 refers to the total number of ships visiting UK ports index (SA). The Full Sample case spans from 2020-03-31 to 2021-10-31 (20 obs.) and includes the COVID-19 outbreak (March to May, 2020) in the evaluation. The Subsample case spans from 2020-11-30 to 2021-10-31 (12 obs.).

	Full Sample		Subsample	
	MAE	**RMSE**	**MAE**	**RMSE**
AR(1)	1	1	1	1
AR(P)	0.999	1	0.998	1
All Aggregates	0.869	0.913	0.933	0.972
BFW	0.831	0.900	0.766	0.788
Ridge	0.867	0.914	0.911	0.942
Lasso	0.852	0.909	0.836	0.873
EN	0.854	0.910	0.848	0.885
Ad. Lasso, V1	0.840	0.905	0.791	0.819
Ad. Lasso, V2	0.840	0.904	0.810	0.834
PCA(1)	0.863	0.912	0.852	0.926
PCA(A1)	0.860	0.911	0.923	0.962
PCA(A2)	0.864	0.912	0.930	0.962
Random Forests	0.843	0.905	0.927	0.964
MLP	1.043	1.029	1.211	1.237
ELM	0.876	0.921	0.981	0.990
SVR	0.849	0.908	0.893	0.948

Table 3: Nowcast MAE and RMSE using the online job ads and the traffic at UK ports real-time indicators. Values correspond to statistics relative to the AR(1) benchmark. All Aggregates refers to the online job index across all industries, the extracted trend of the total number of ships visiting UK ports and the number of ships visiting UK ports index (SA). The Full Sample case spans from 2020-03-31 to 2021-10-31 (20 obs.) and includes the COVID-19 outbreak (March to May, 2020) in the evaluation. The Subsample case spans from 2020-11-30 to 2021-10-31 (12 obs.).

Figures

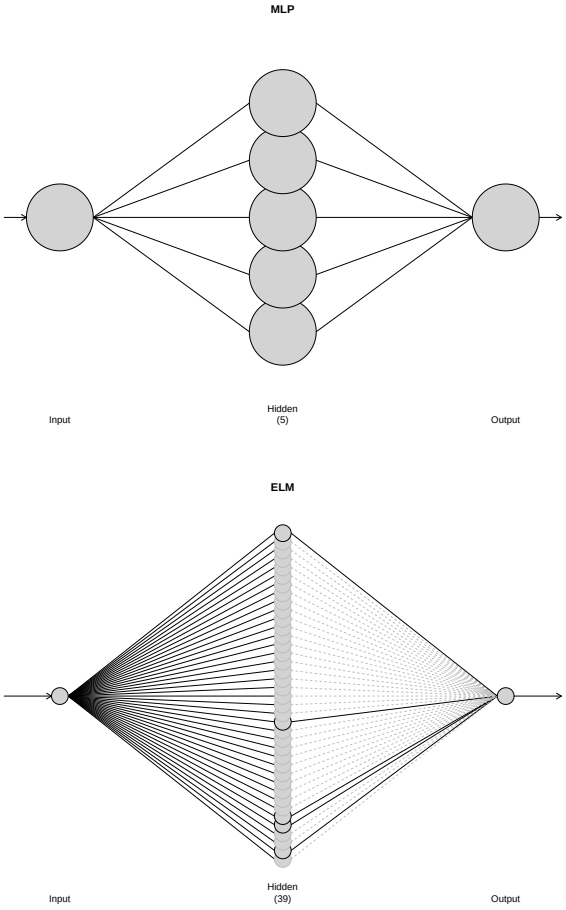


Figure 1: MLP and EML layers based on both Job Ads and port traffic 86 disaggregates; estimation as of 2020-01-26.

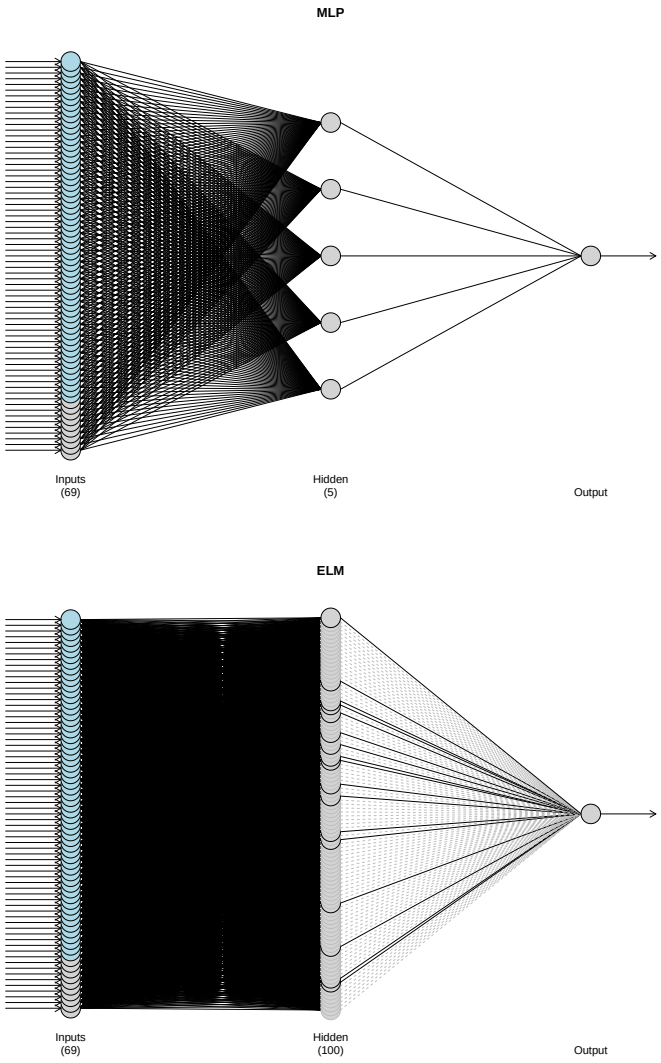
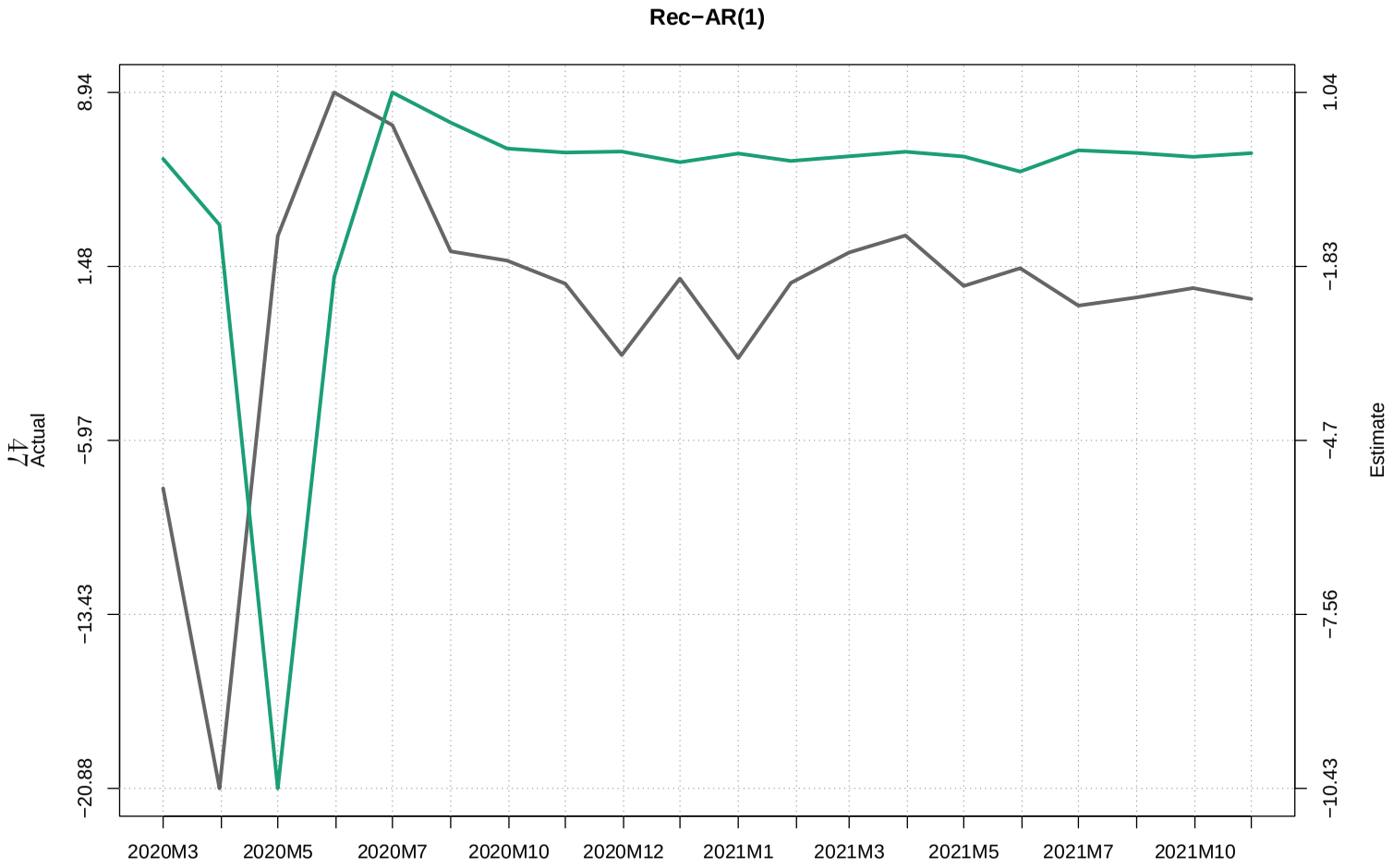
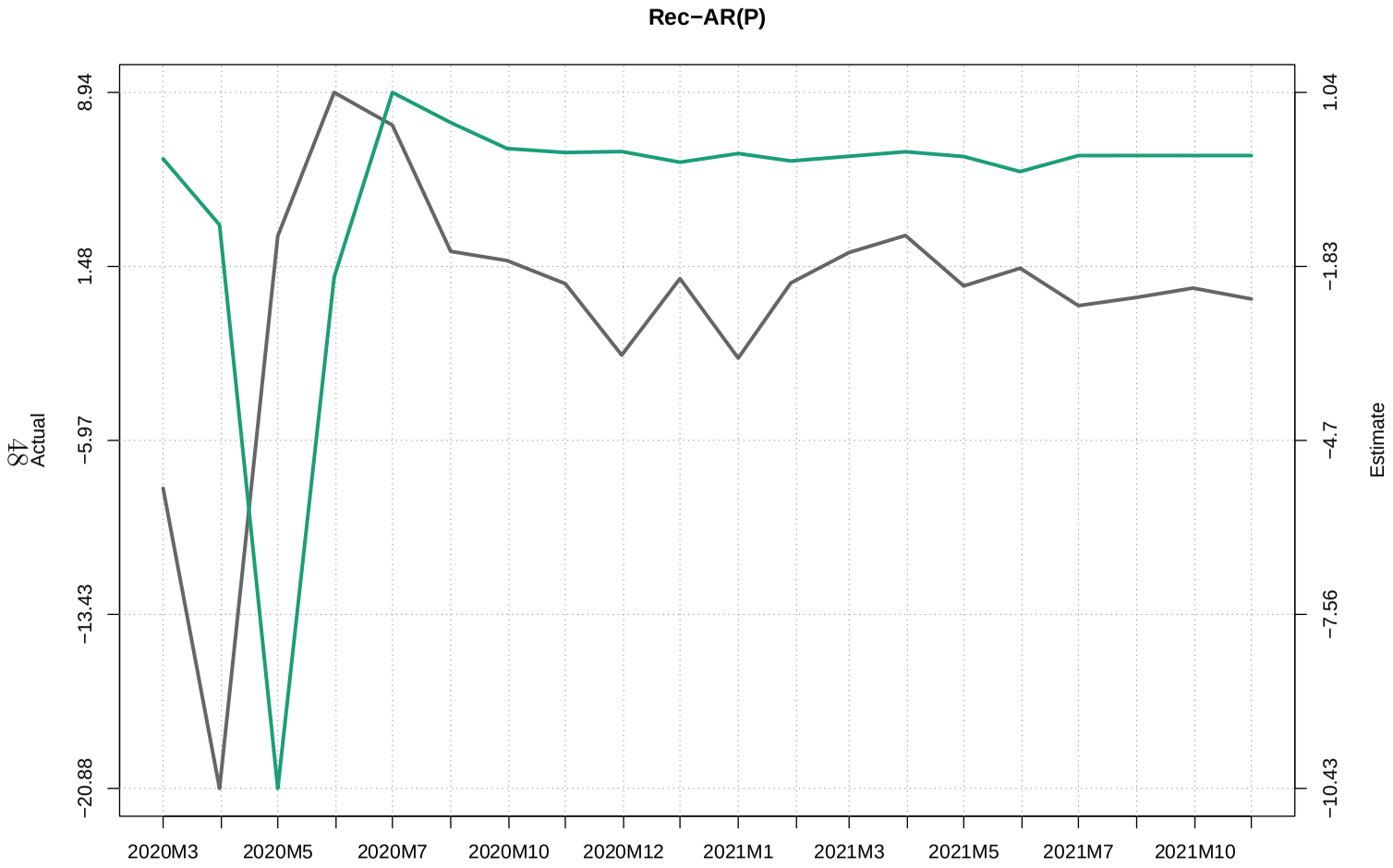


Figure 2: MLP and EML layers based on both Job Ads and port traffic 86 disaggregates; estimation as of 2021-10-24.

Appendix: All Nowcasts

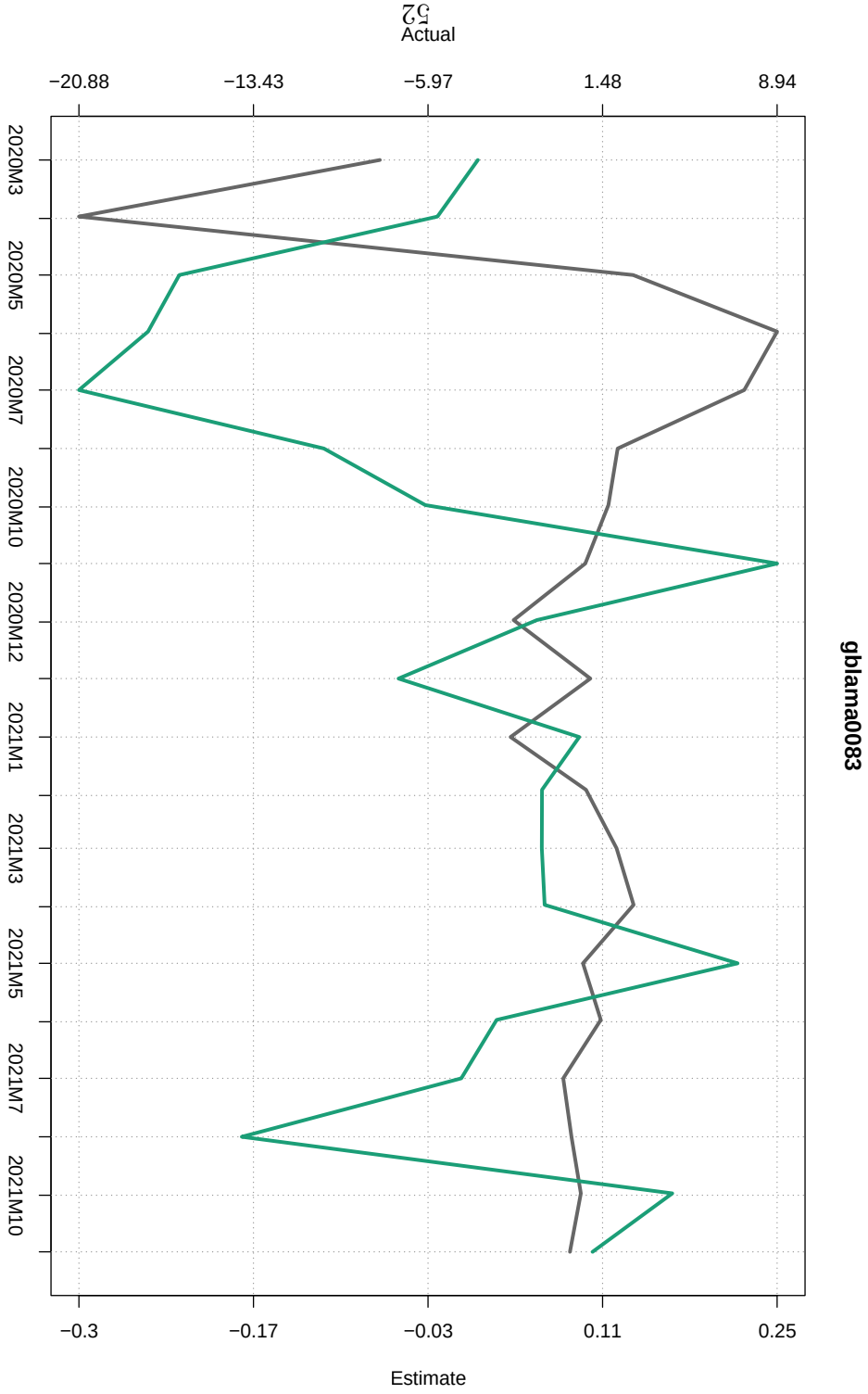

Rec−AR(1)
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.04
−1.83
−4.7
−7.56
−10.43
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

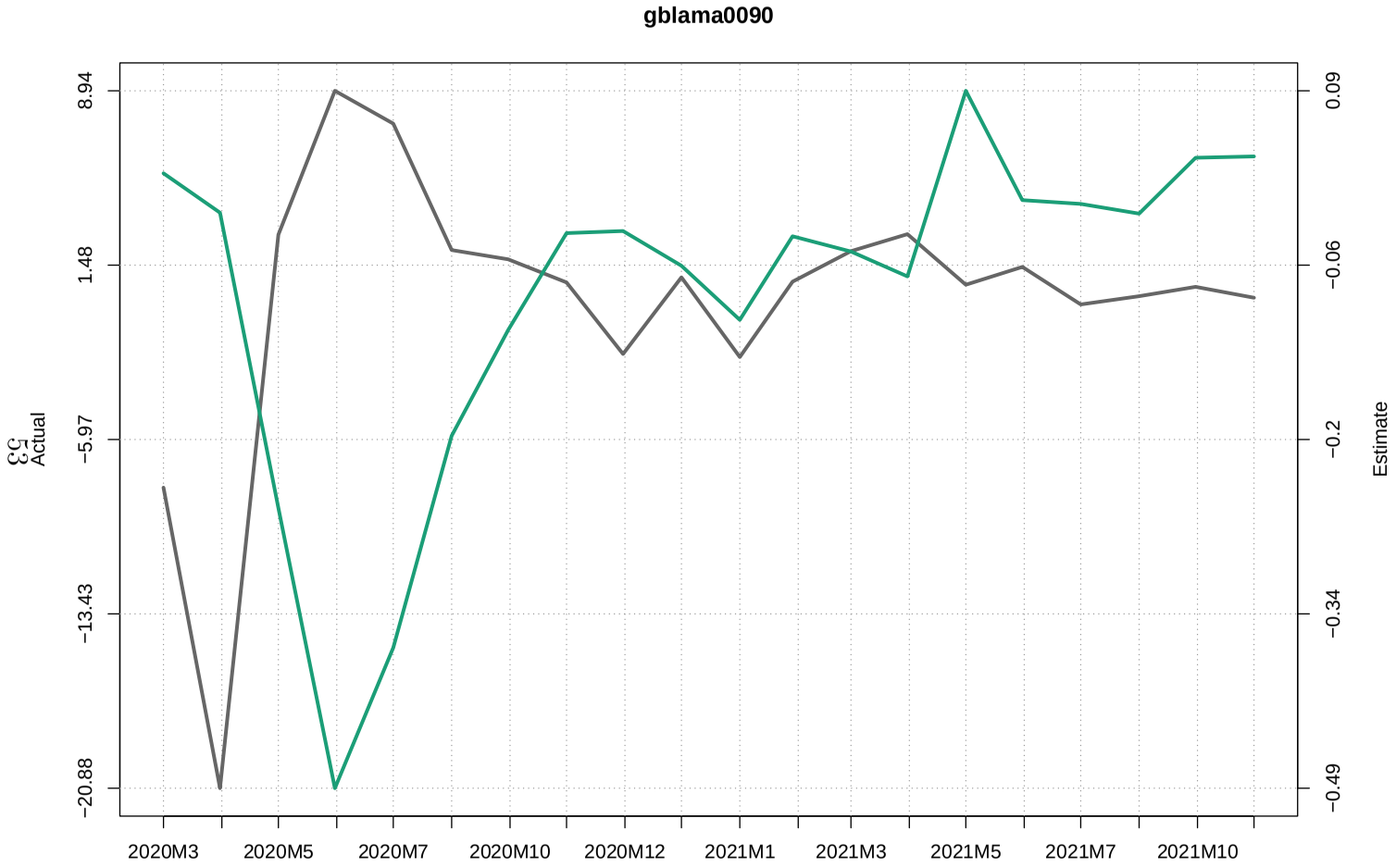

Rec−AR(P)
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.04
−1.83
−4.7
−7.56
−10.43
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

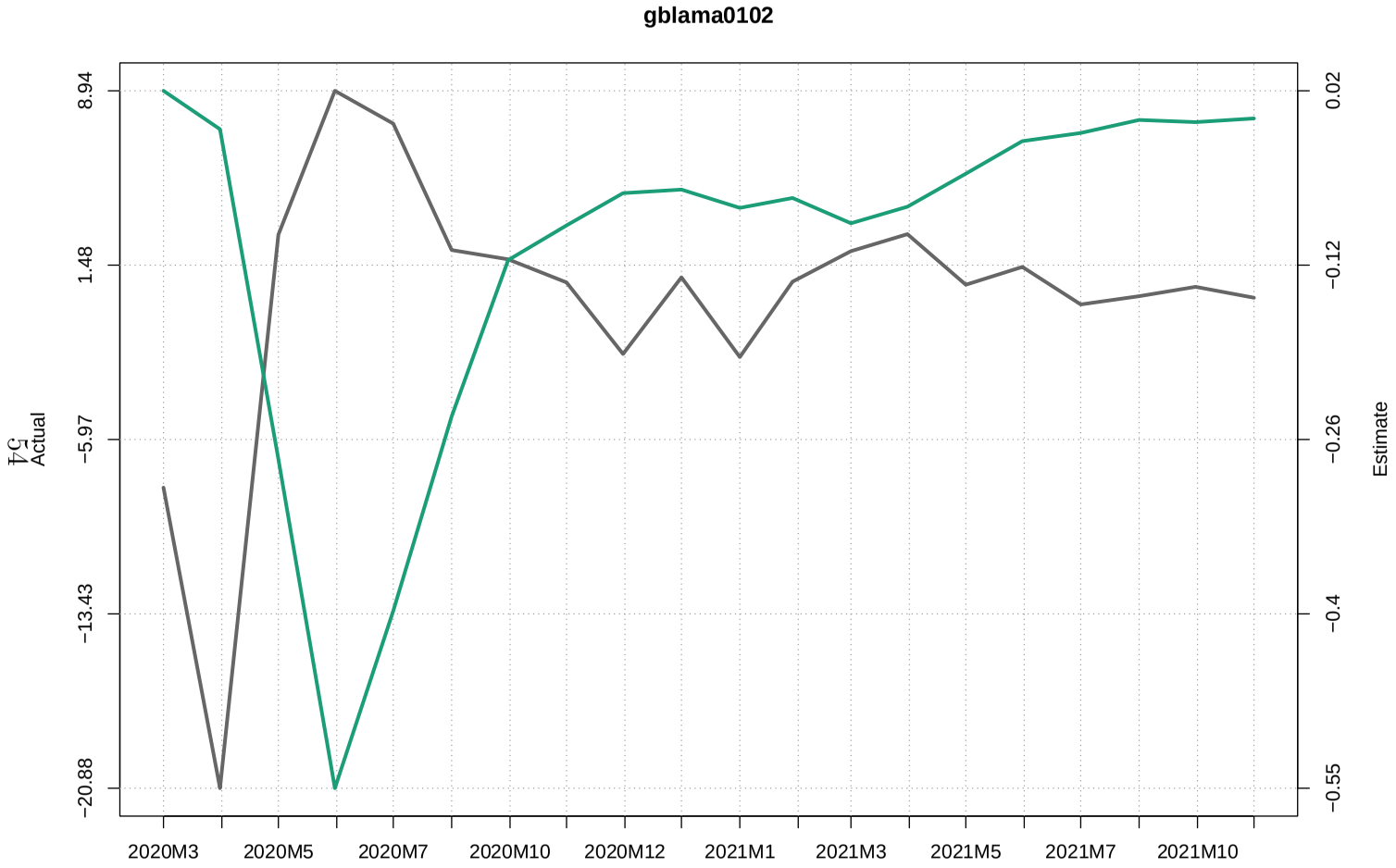
gblama0103
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.27
−0.11
0.05
0.21
0.37
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

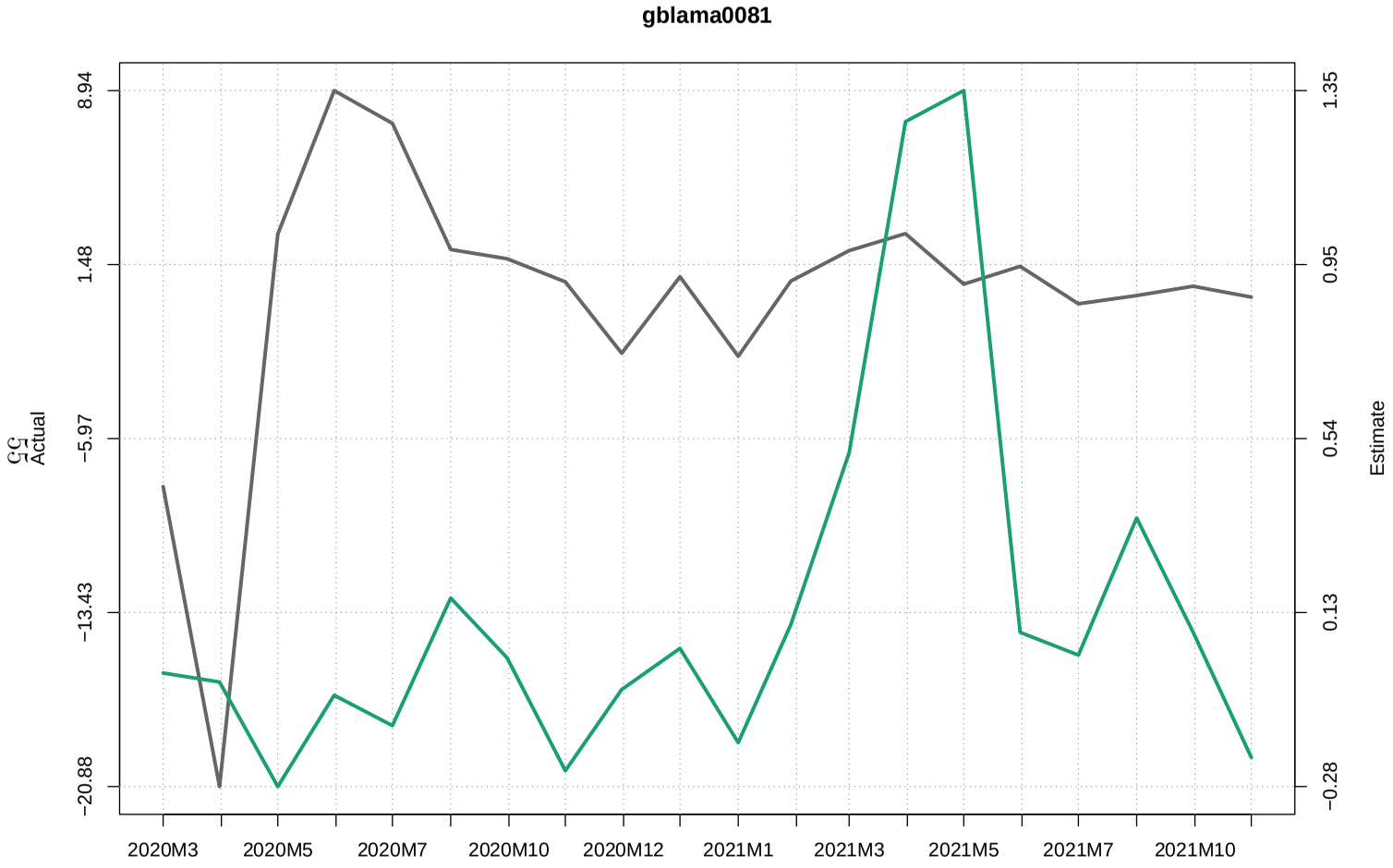
gbtran0354
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.39
−0.28
−0.16
−0.04
0.08
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gbtran0356
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.1
−0.22
−0.33
−0.45
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0083
−0.3
−0.17
−0.03
0.11
0.25
Estimate

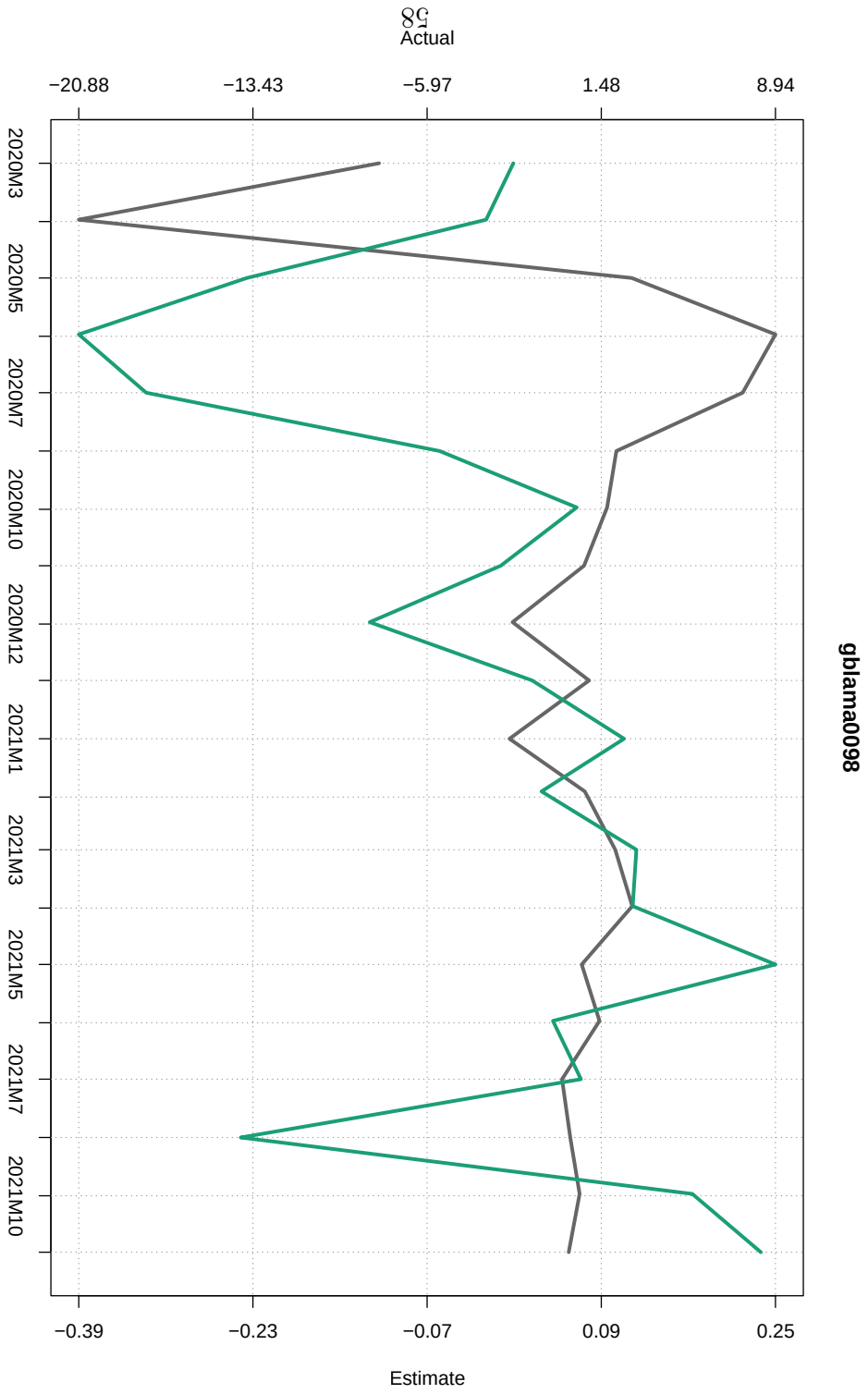

gblama0090
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.09
−0.06
−0.2
−0.34
−0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

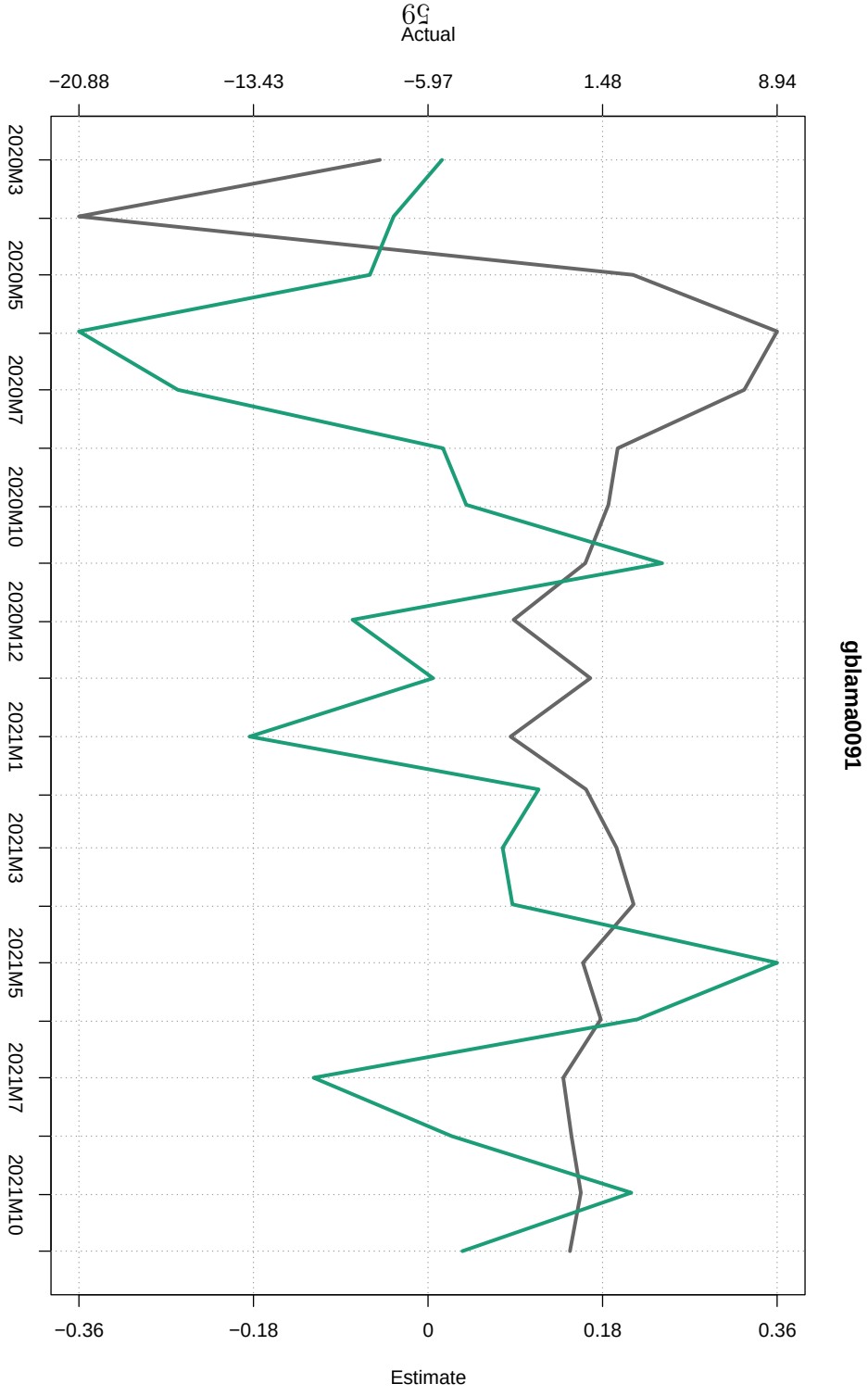

gblama0102
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.12
−0.26
−0.4
−0.55
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

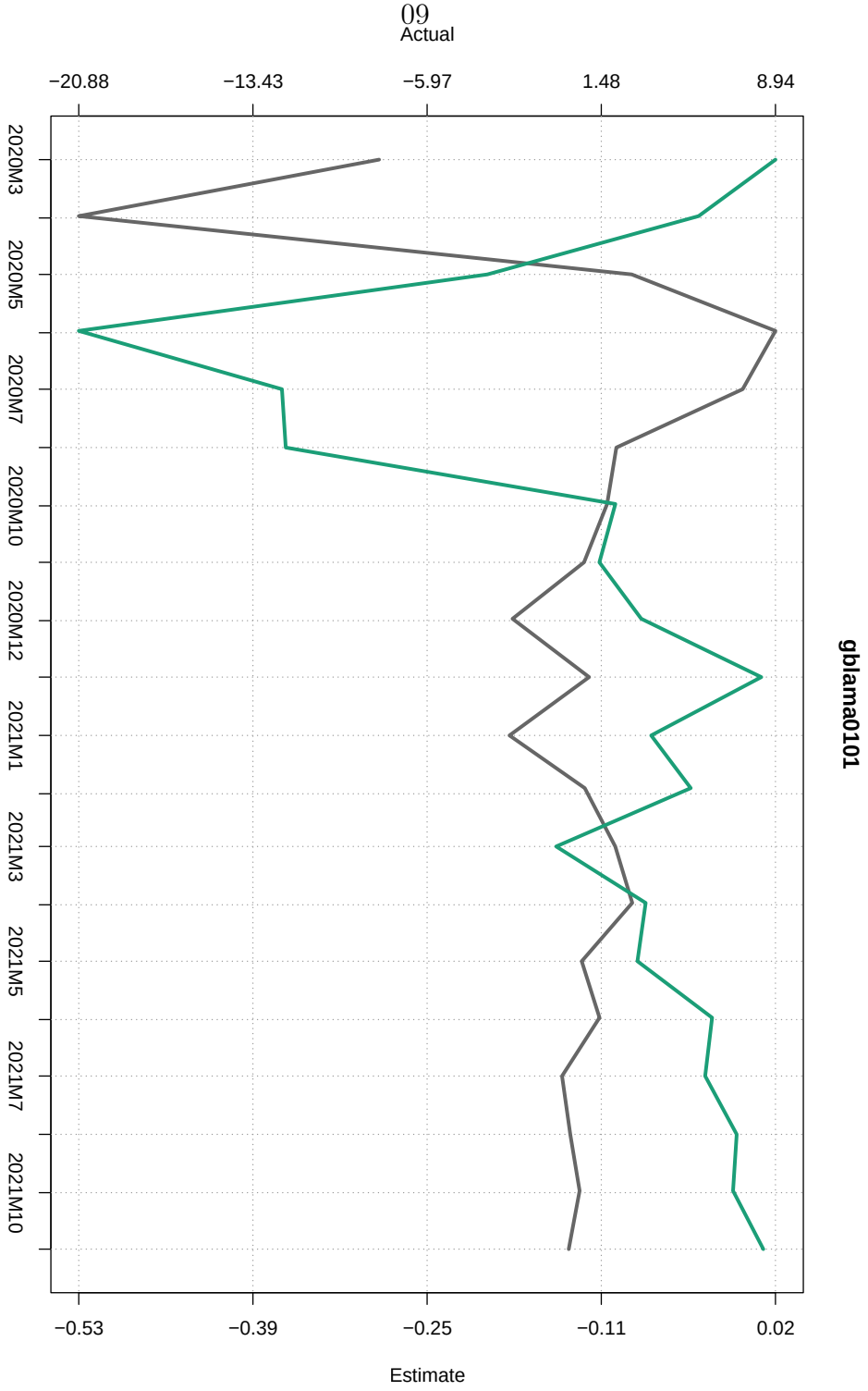

gblama0081
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.35
0.95
0.54
0.13
−0.28
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

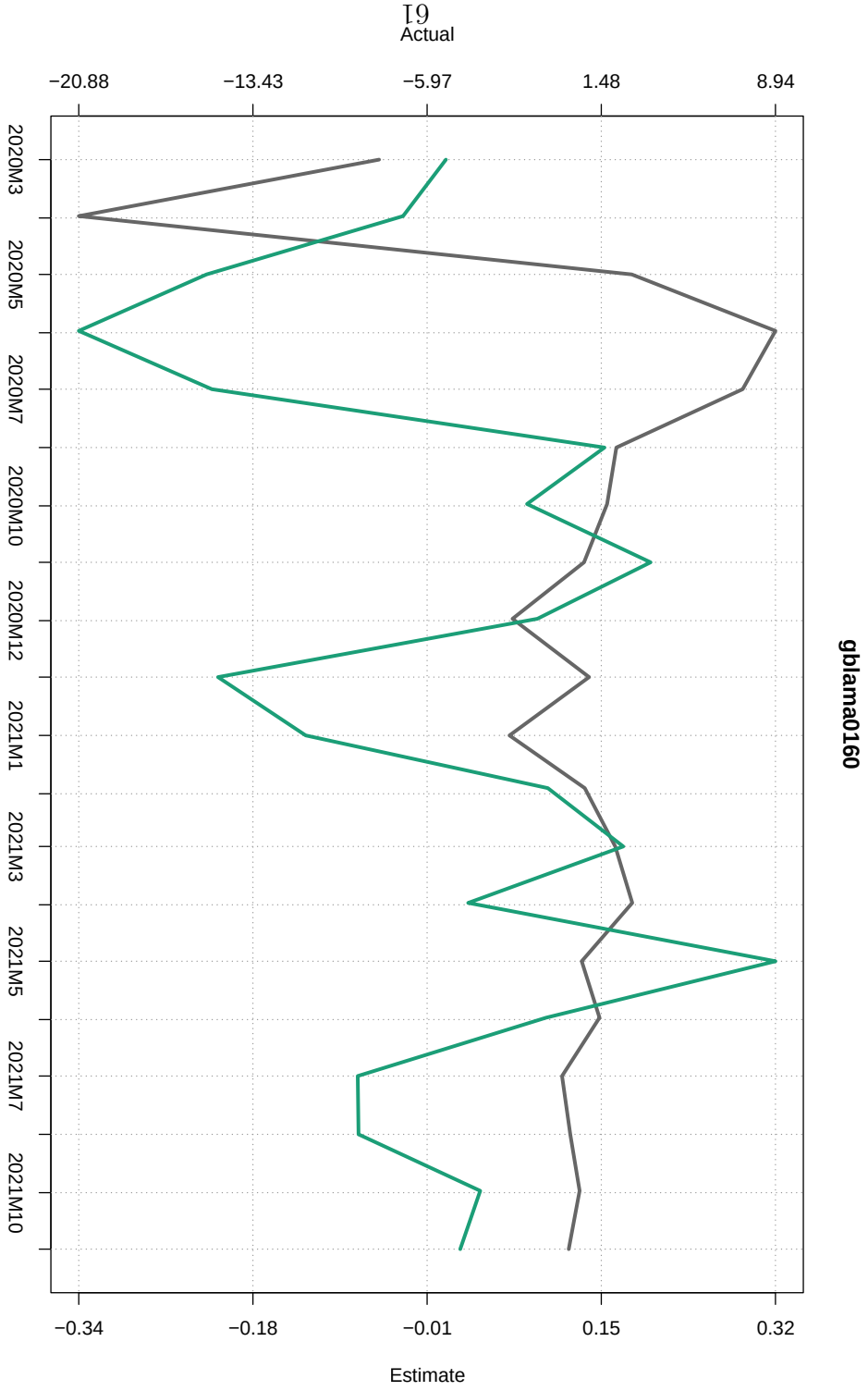
gblama0096
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.47
−0.34
−0.21
−0.08
0.05
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gblama0076
Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
Estimate
−0.33
−0.15
0.03
0.21
0.39

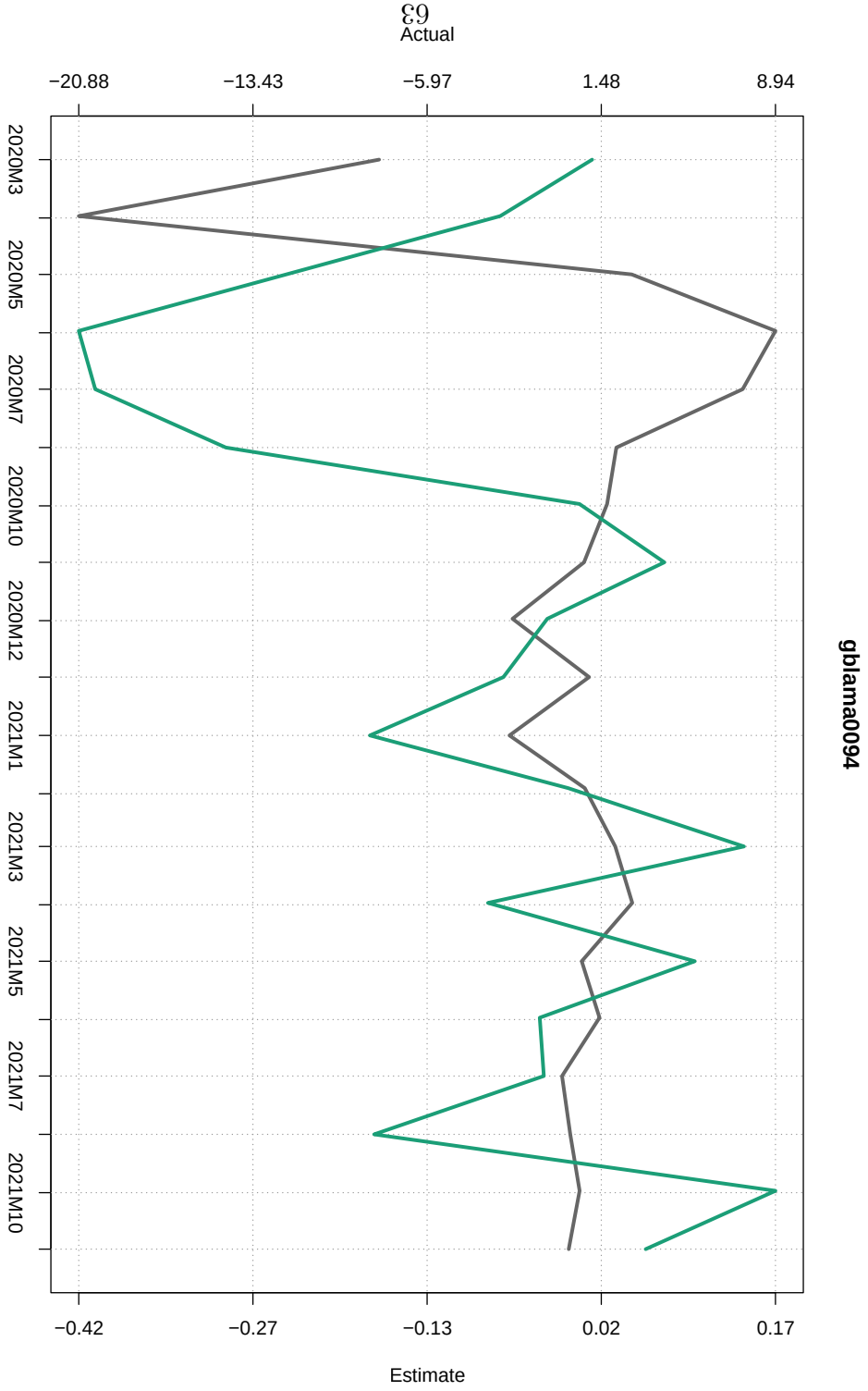

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0098
−0.39
−0.23
−0.07
0.09
0.25
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0091
−0.36
−0.18
0
0.18
0.36
Estimate

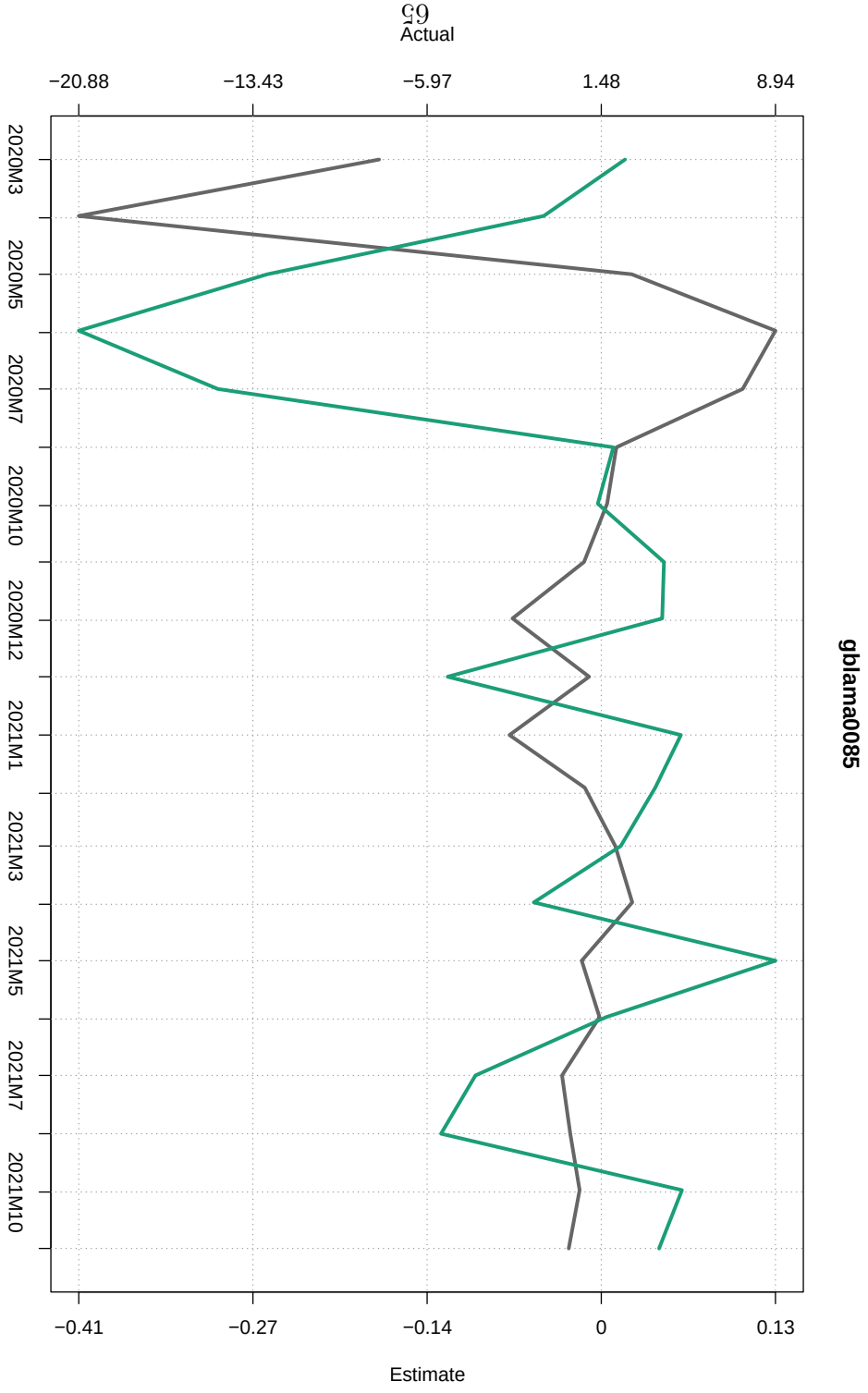

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0101
−0.53
−0.39
−0.25
−0.11
0.02
Estimate

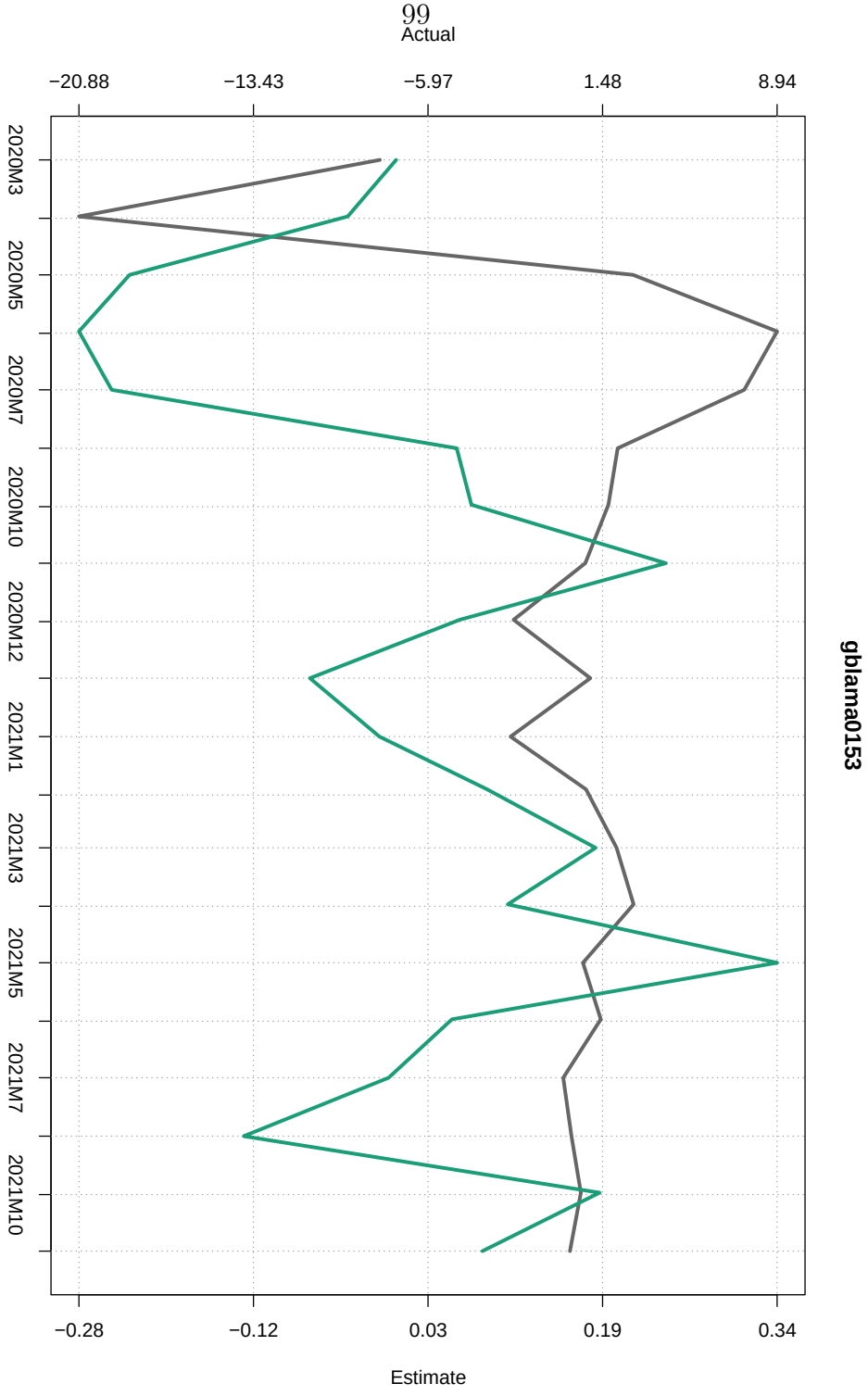

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0160
−0.34
−0.18
−0.01
0.15
0.32
Estimate

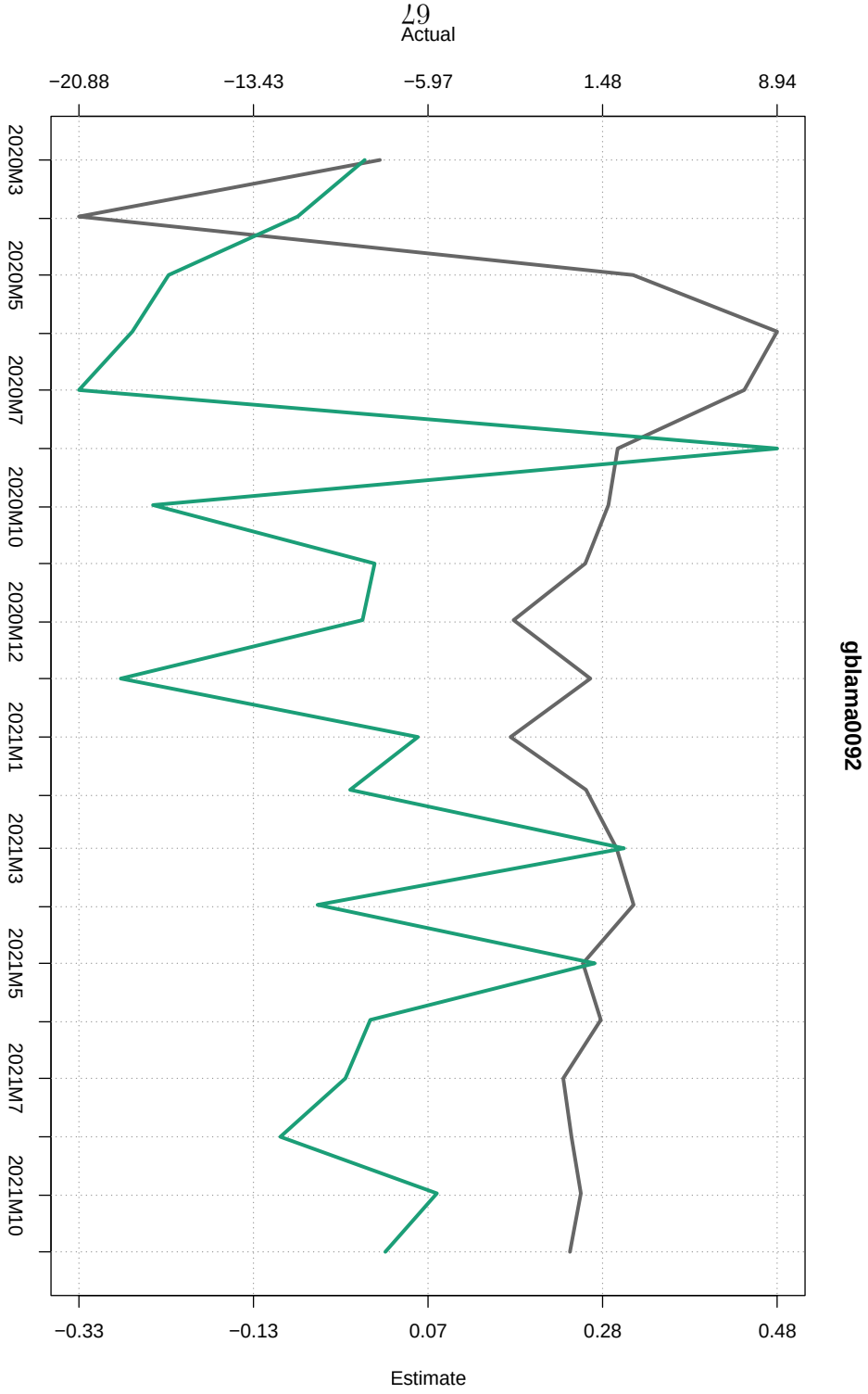
gblama0162
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.43
−0.23
−0.04
0.15
0.35
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

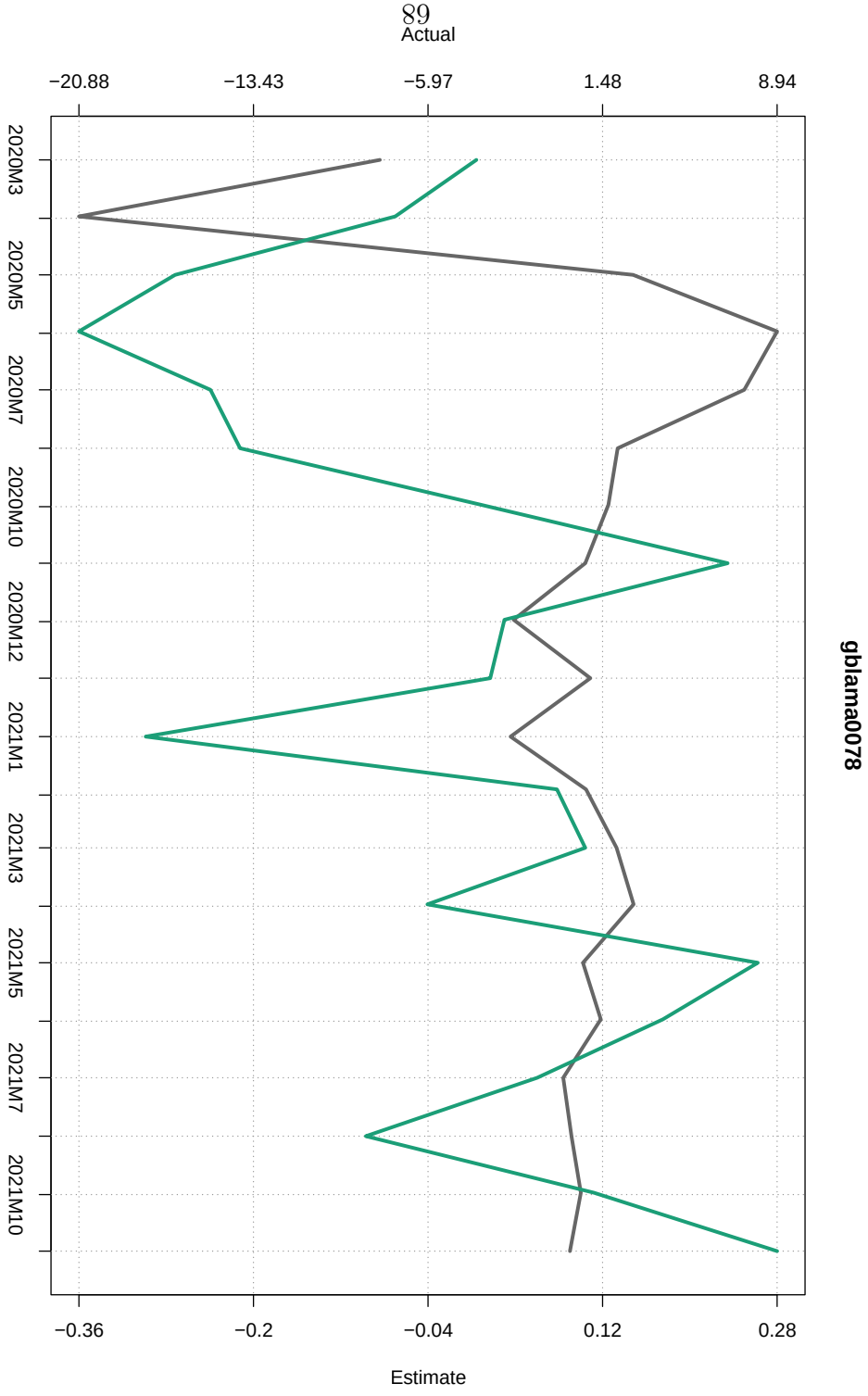

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0094
−0.42
−0.27
−0.13
0.02
0.17
Estimate

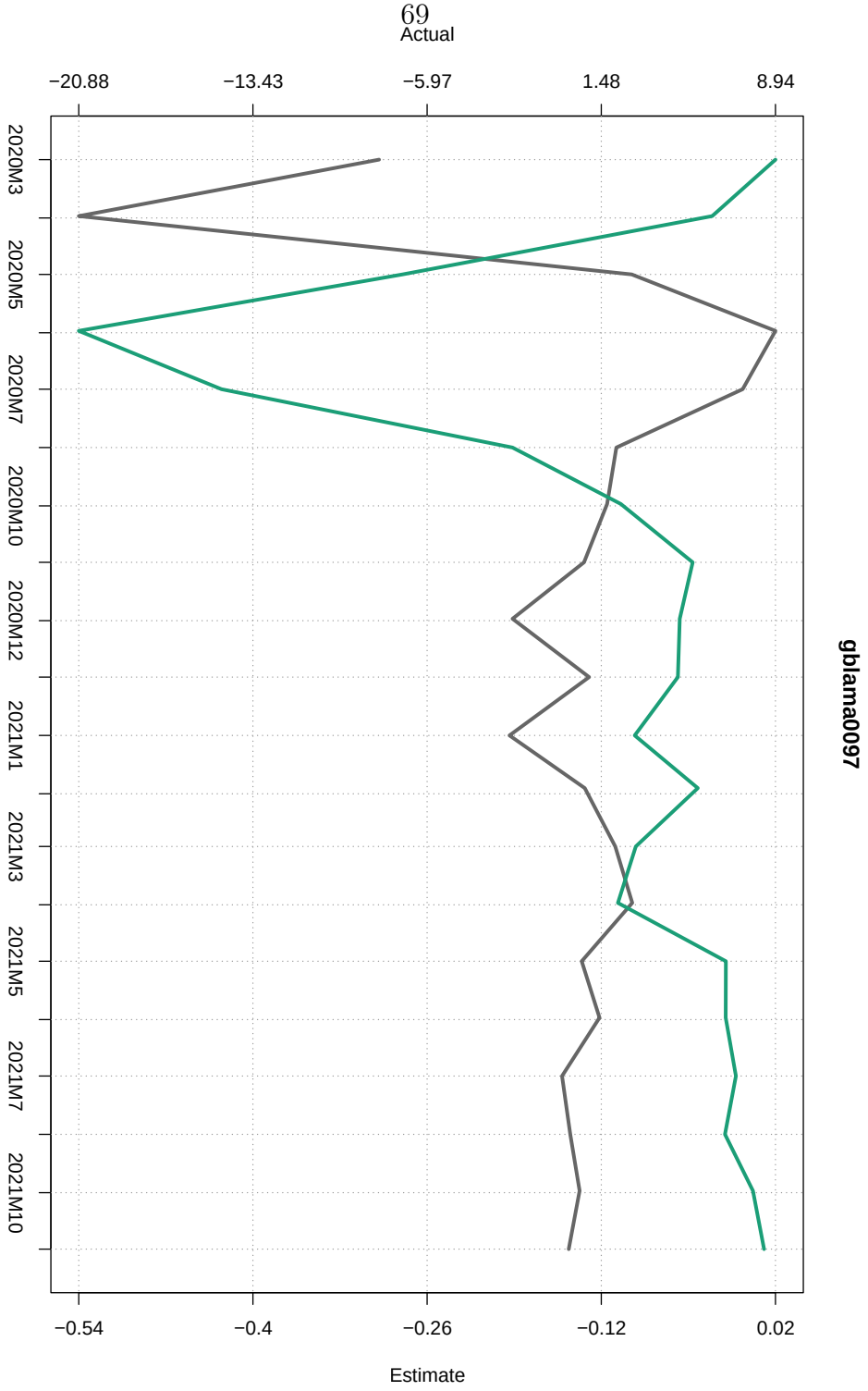
gblama0075
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.48
−0.33
−0.17
−0.02
0.14
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

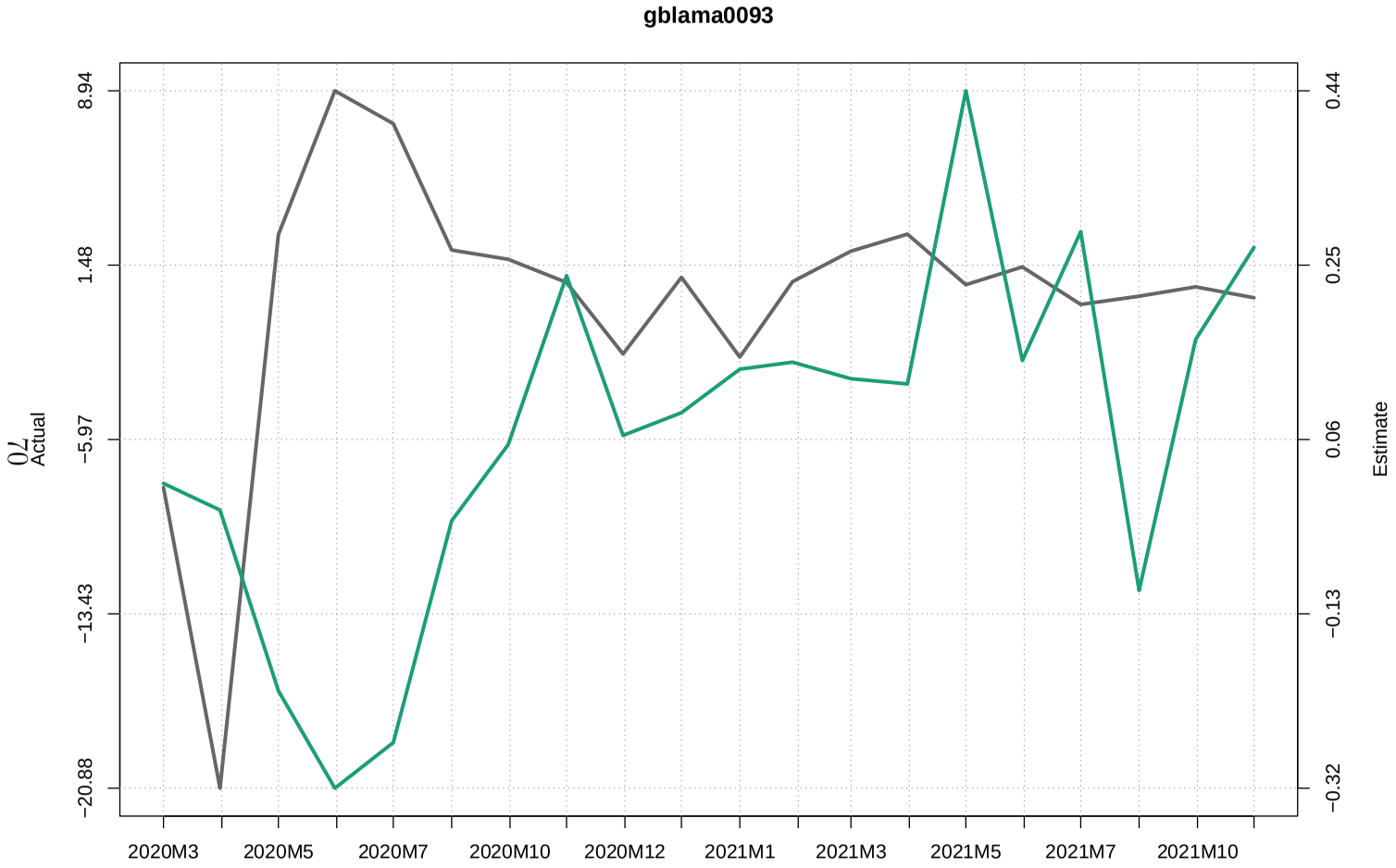

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0085
−0.41
−0.27
−0.14
0
0.13
Estimate

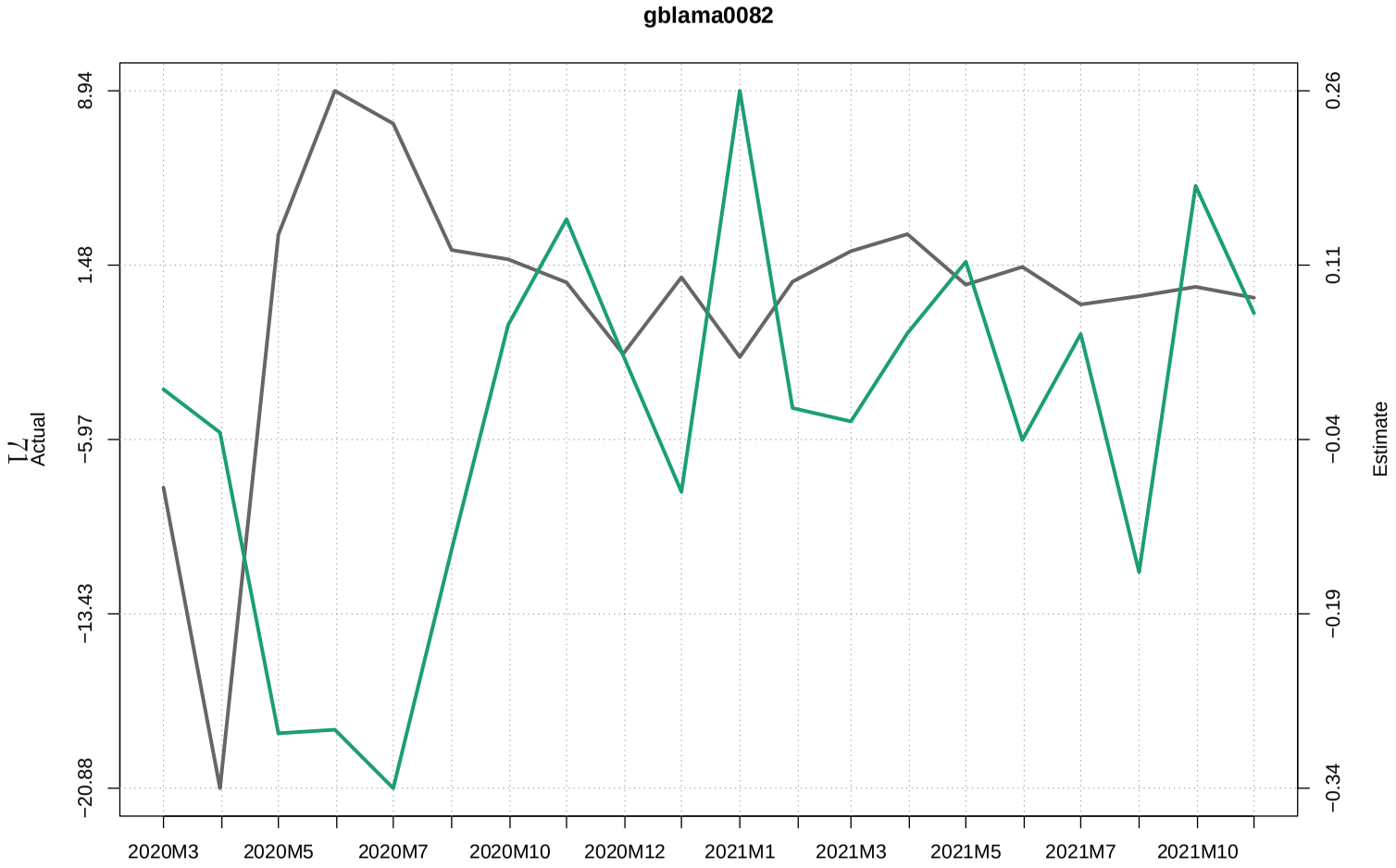

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0153
−0.28
−0.12
0.03
0.19
0.34
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0092
−0.33
−0.13
0.07
0.28
0.48
Estimate

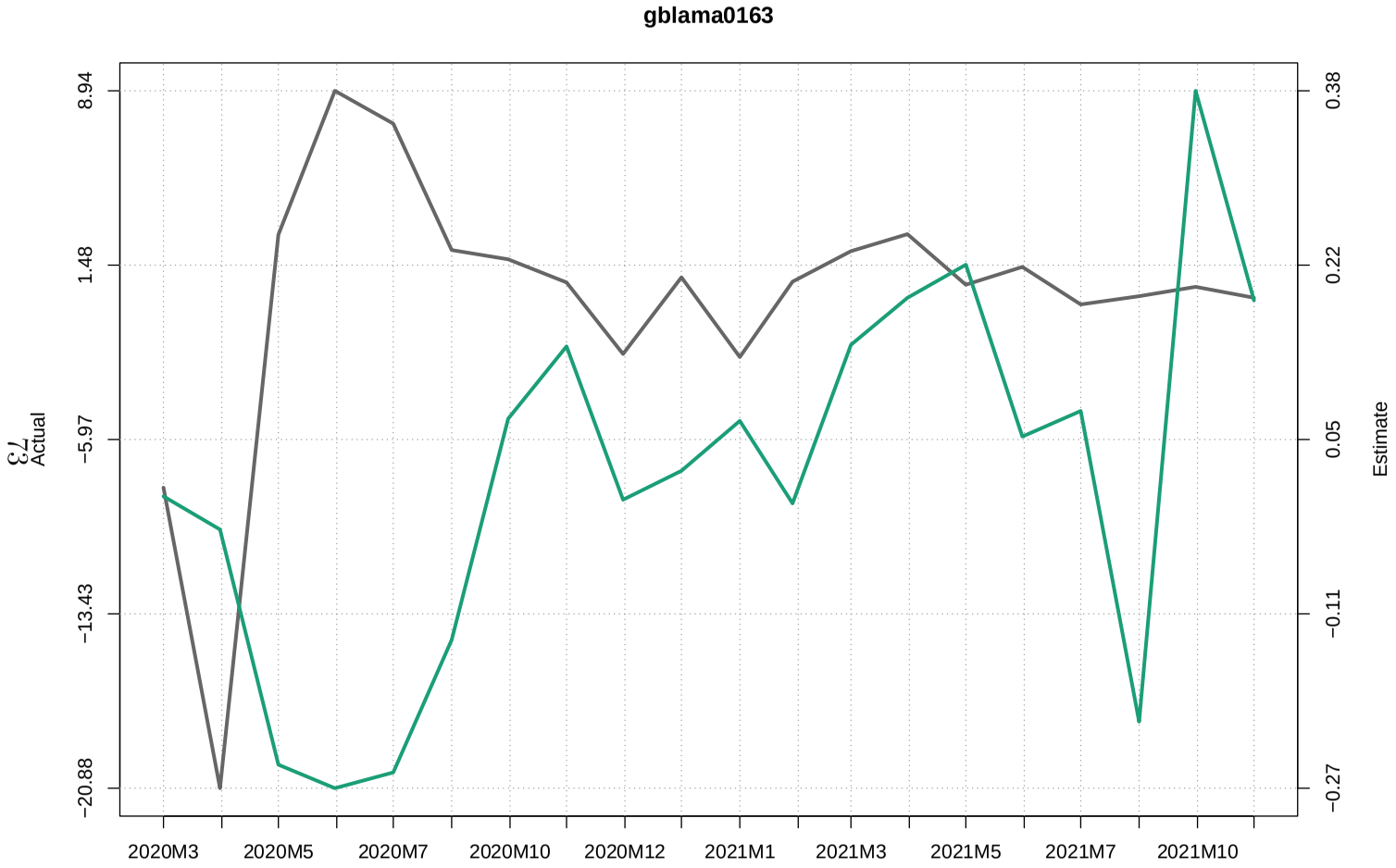

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0078
−0.36
−0.2
−0.04
0.12
0.28
Estimate

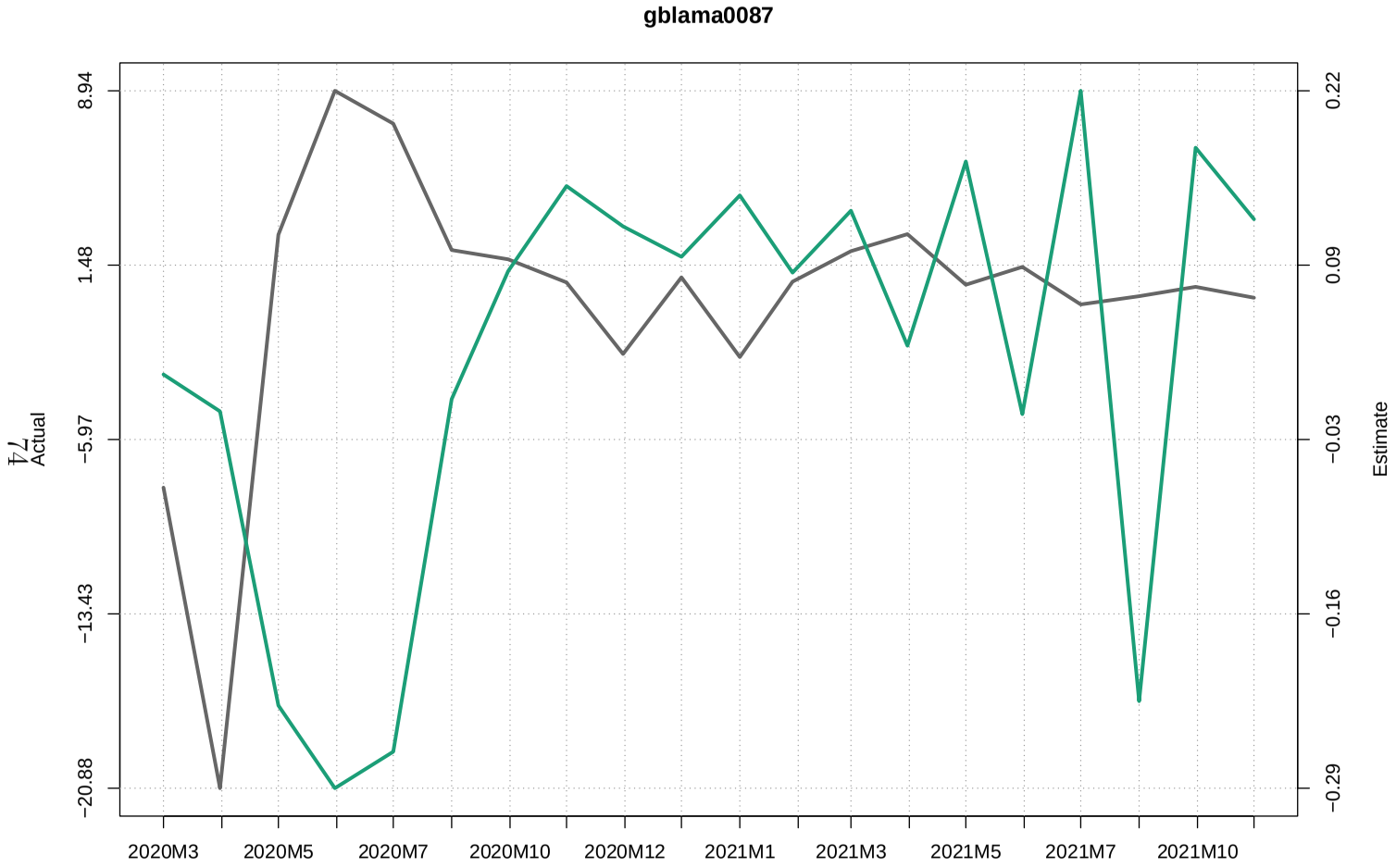

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0097
−0.54
−0.4
−0.26
−0.12
0.02
Estimate

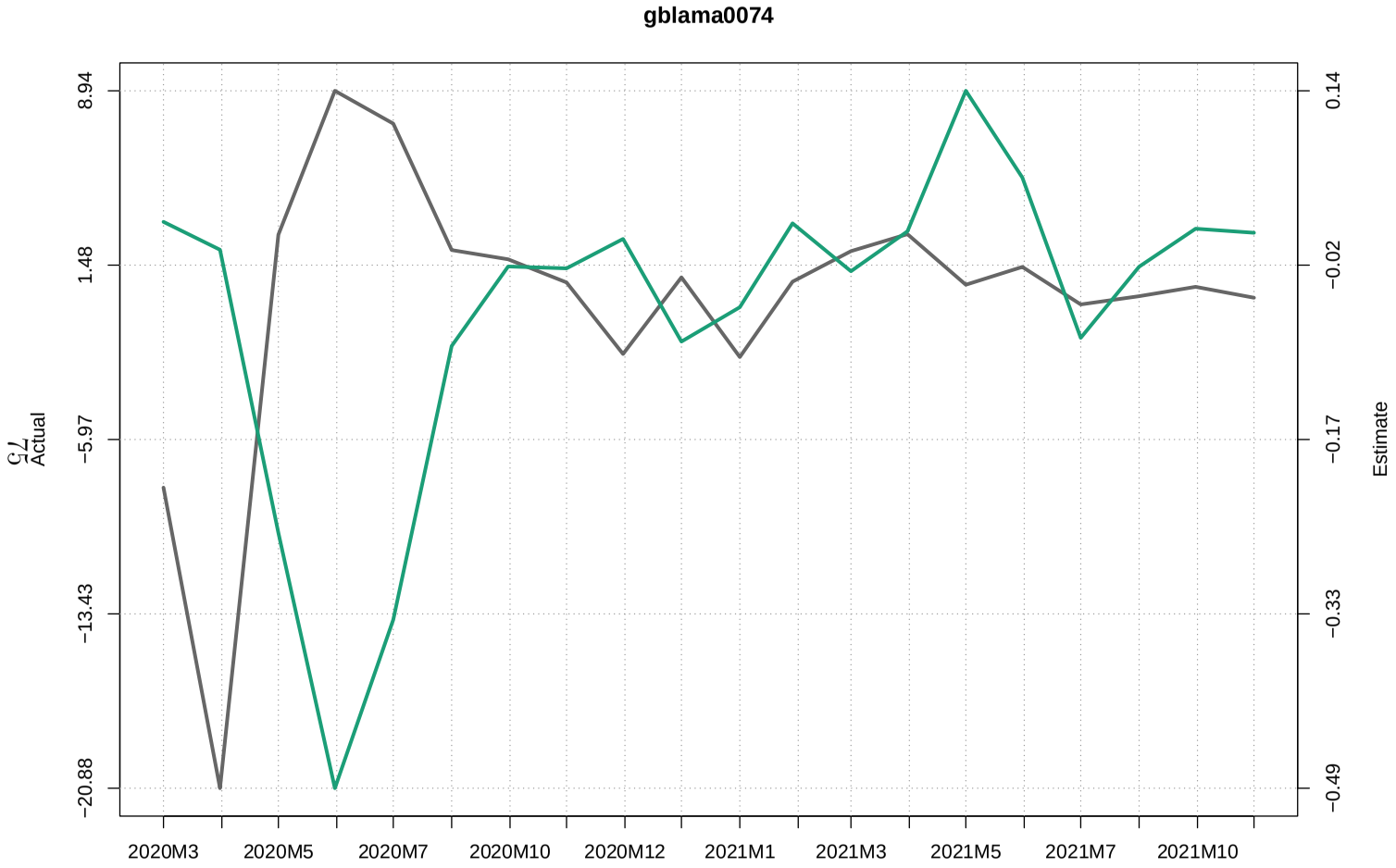

gblama0093
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.44
0.25
0.06
−0.13
−0.32
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gblama0082
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.26
0.11
−0.04
−0.19
−0.34
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

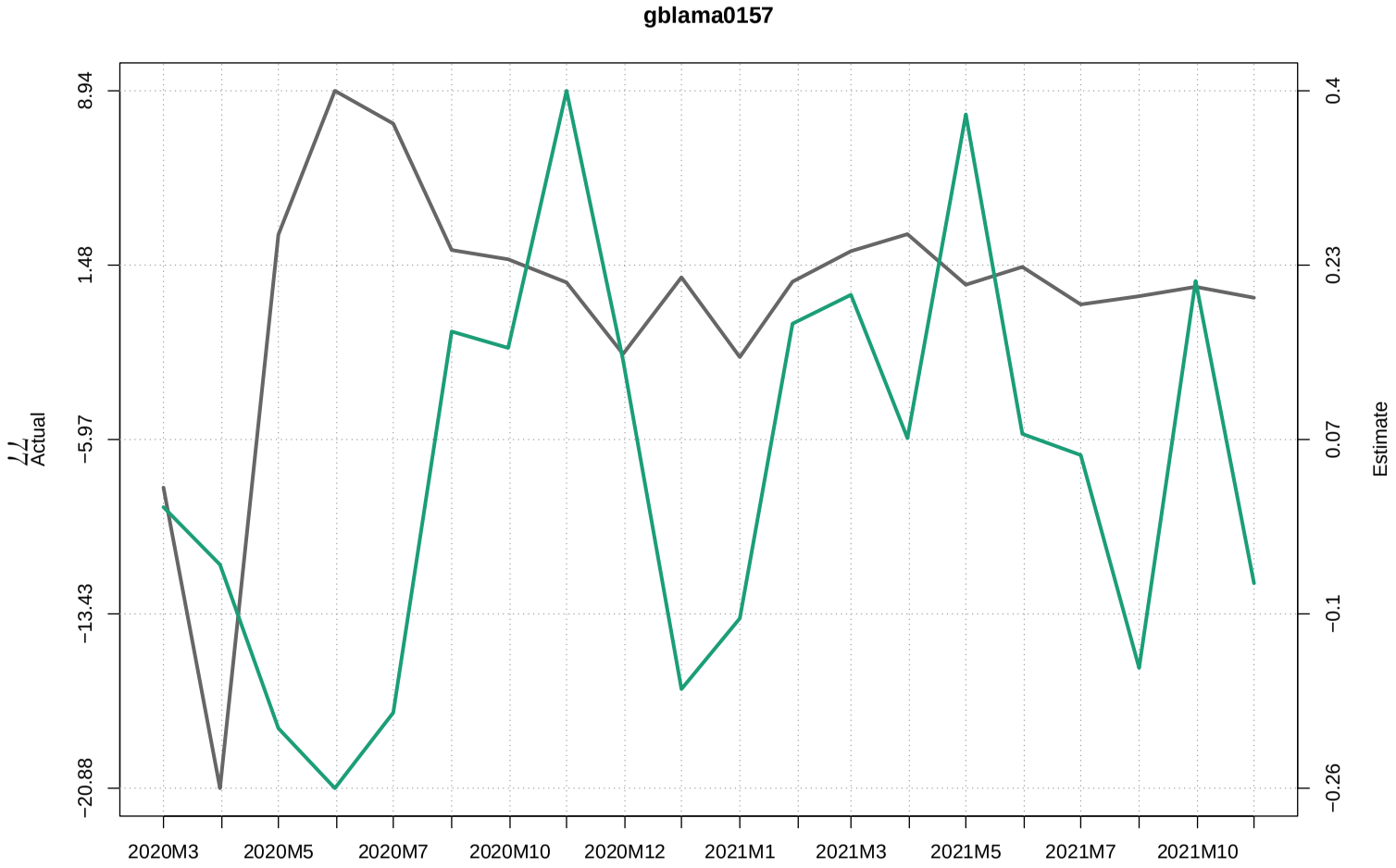
gblama0086
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.43
−0.3
−0.17
−0.04
0.09
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gblama0163
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.38
0.22
0.05
−0.11
−0.27
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gblama0087
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.22
0.09
−0.03
−0.16
−0.29
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

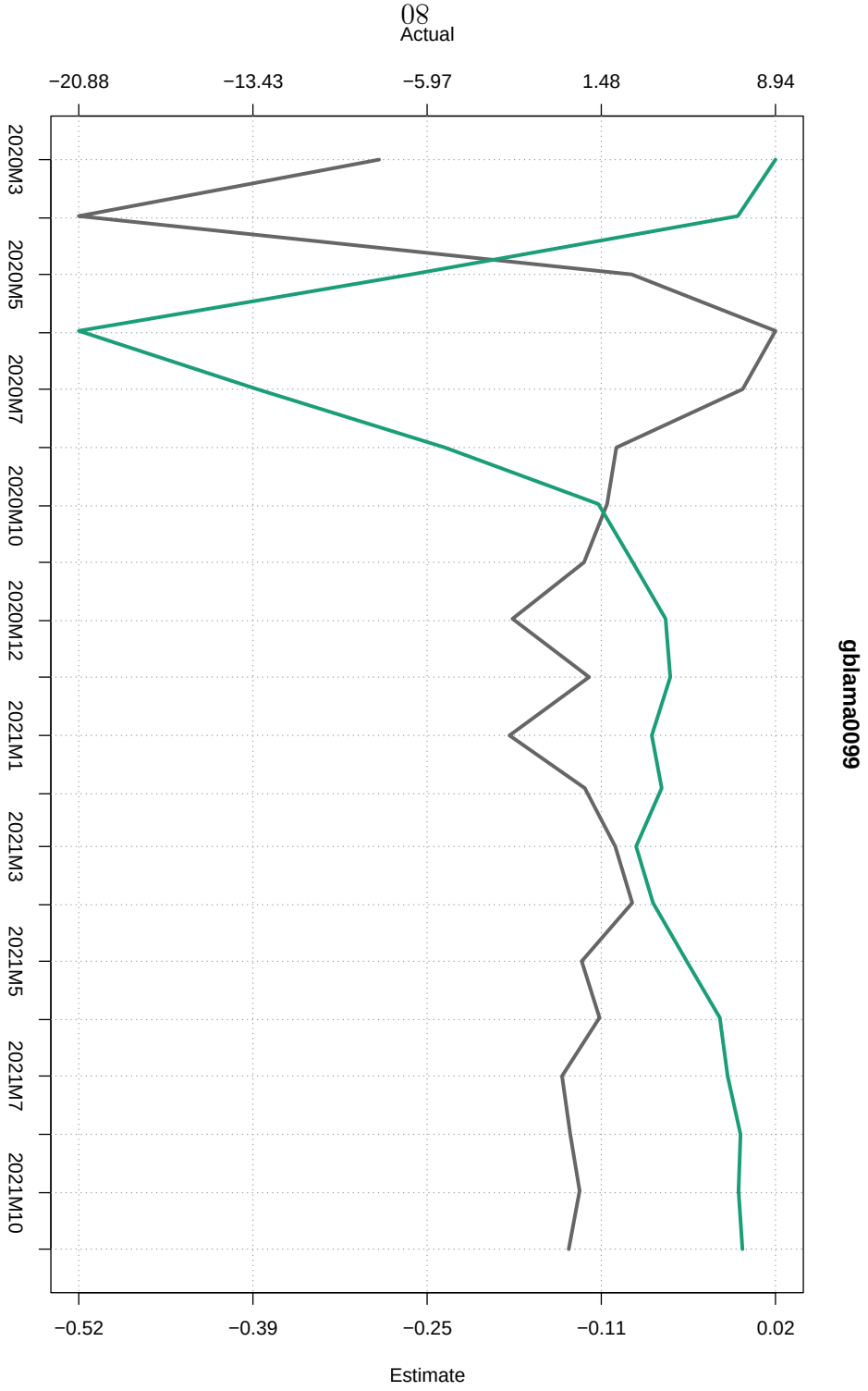

gblama0074
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.14
−0.02
−0.17
−0.33
−0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

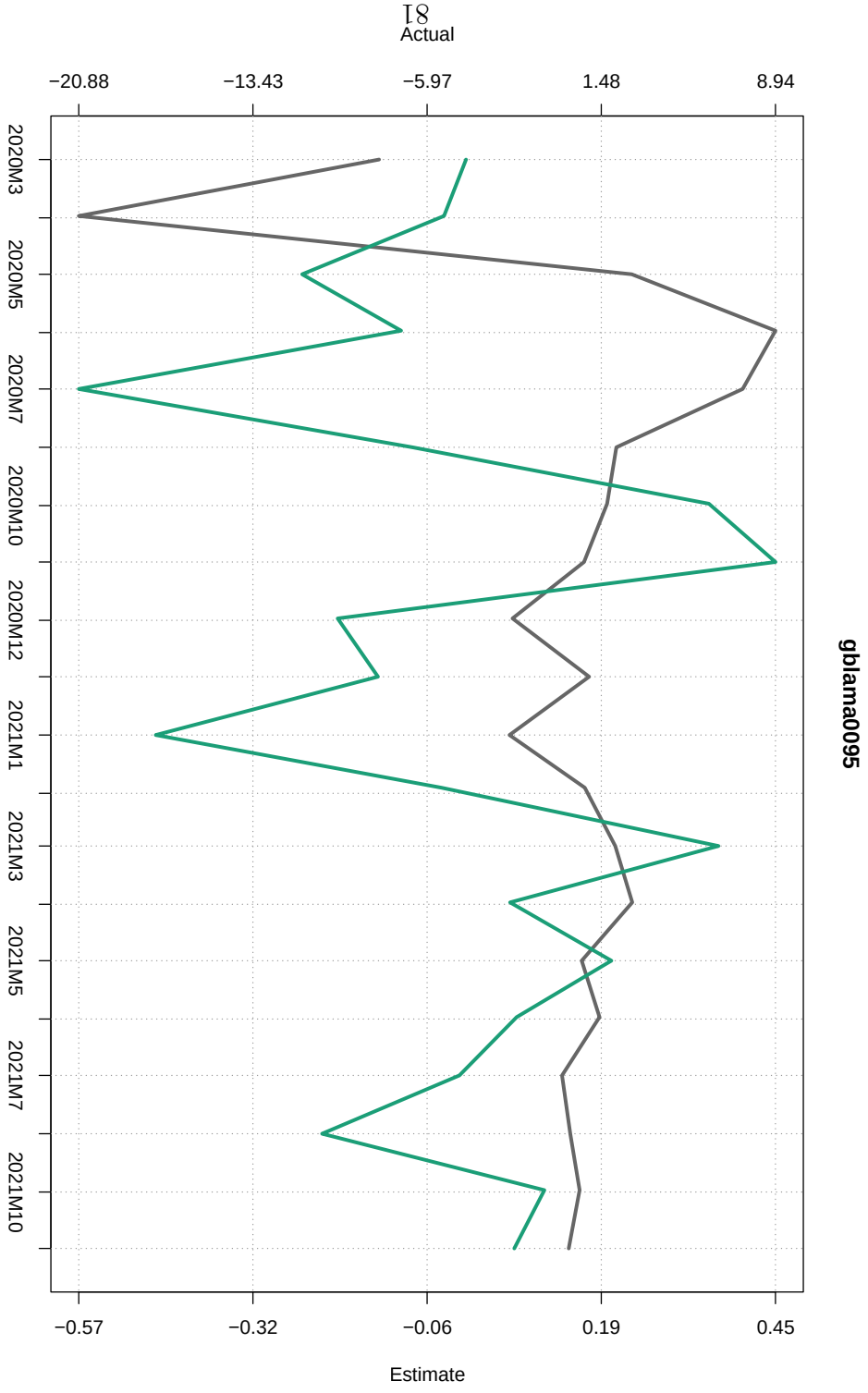
gblama0088
Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
Estimate
−0.26
−0.07
0.12
0.31
0.5


gblama0157
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.4
0.23
0.07
−0.1
−0.26
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

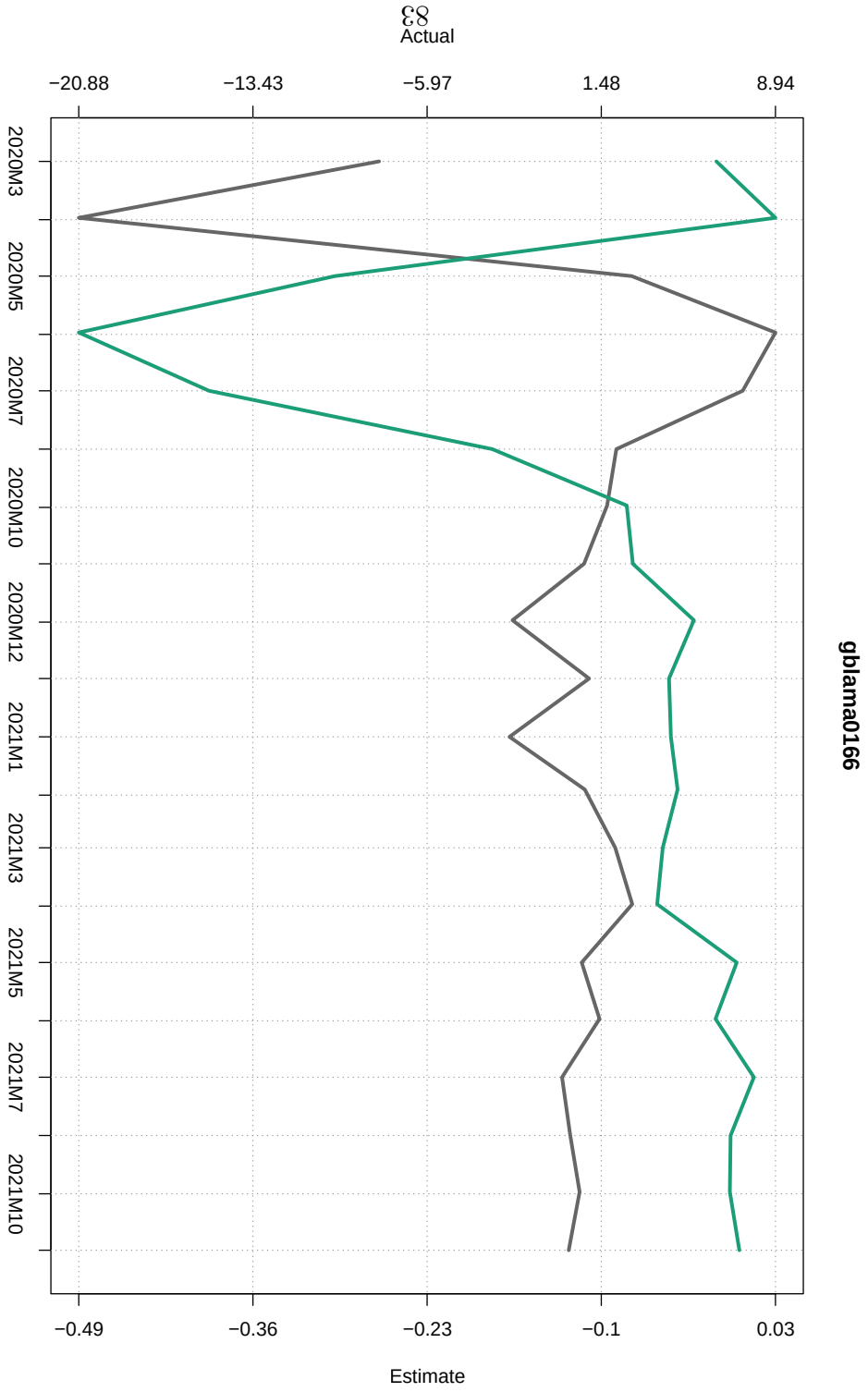
gblama0158
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.24
−0.08
0.09
0.25
0.41
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gblama0156
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.43
−0.24
−0.05
0.14
0.33
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

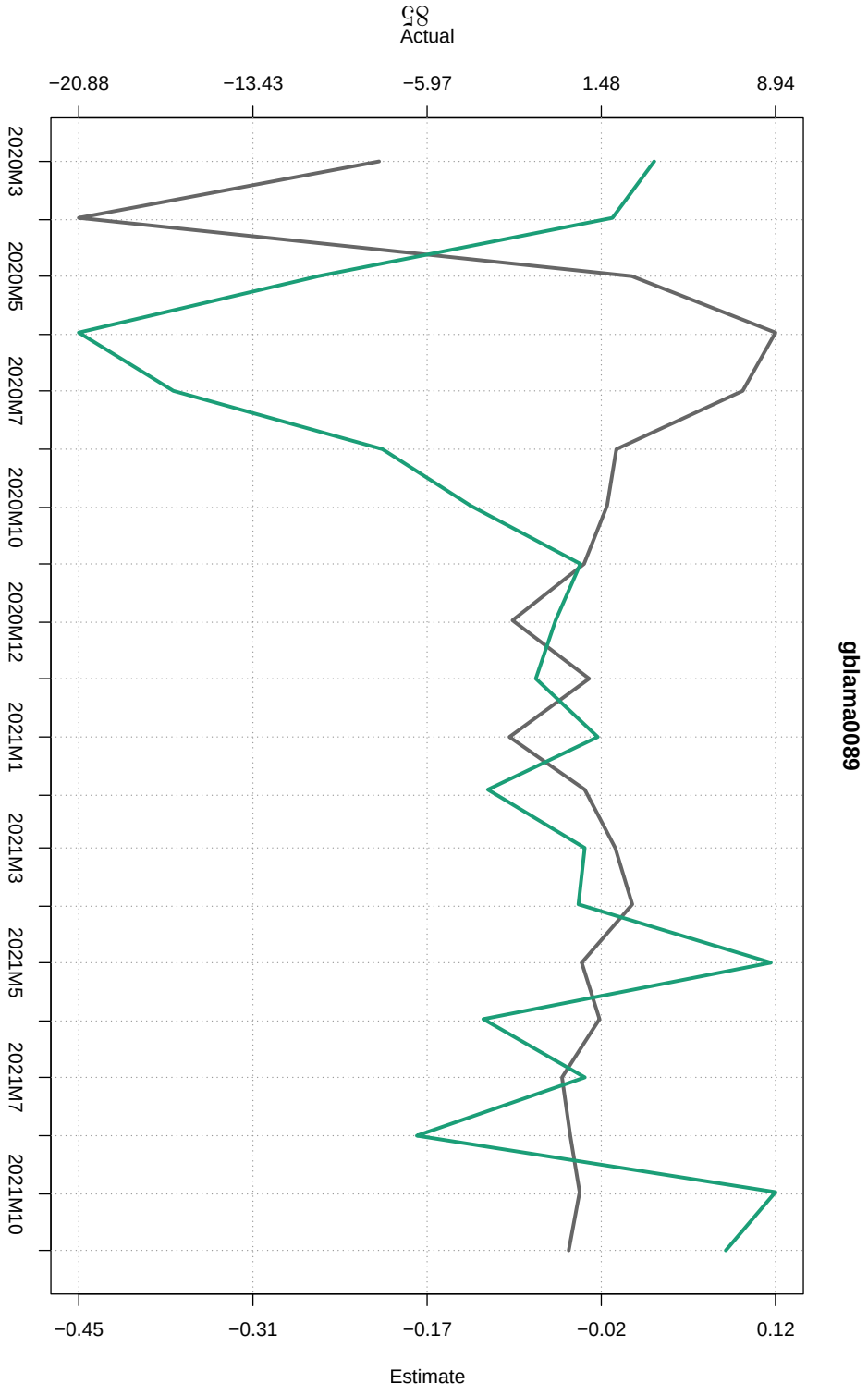

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0099
−0.52
−0.39
−0.25
−0.11
0.02
Estimate

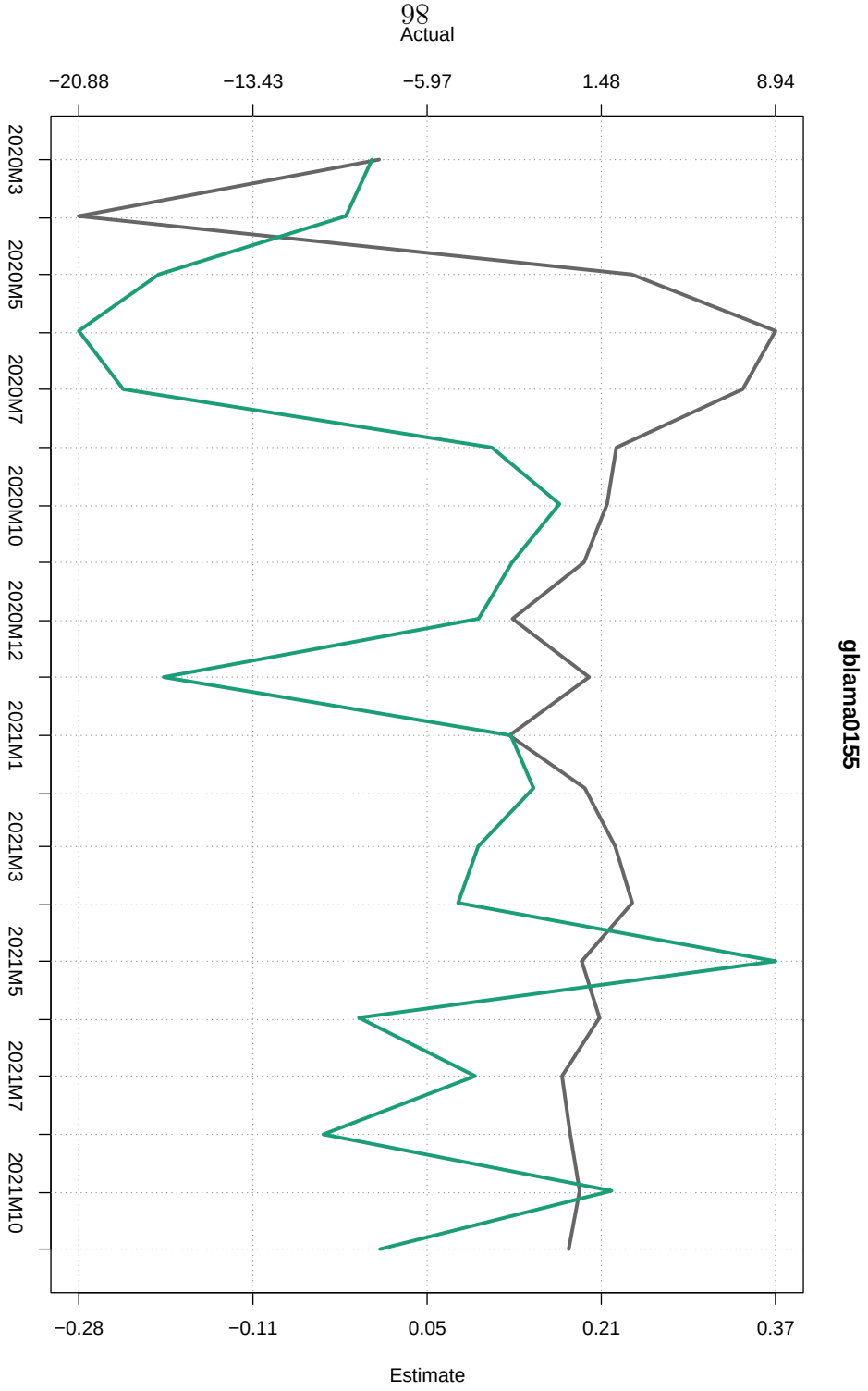

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0095
−0.57
−0.32
−0.06
0.19
0.45
Estimate

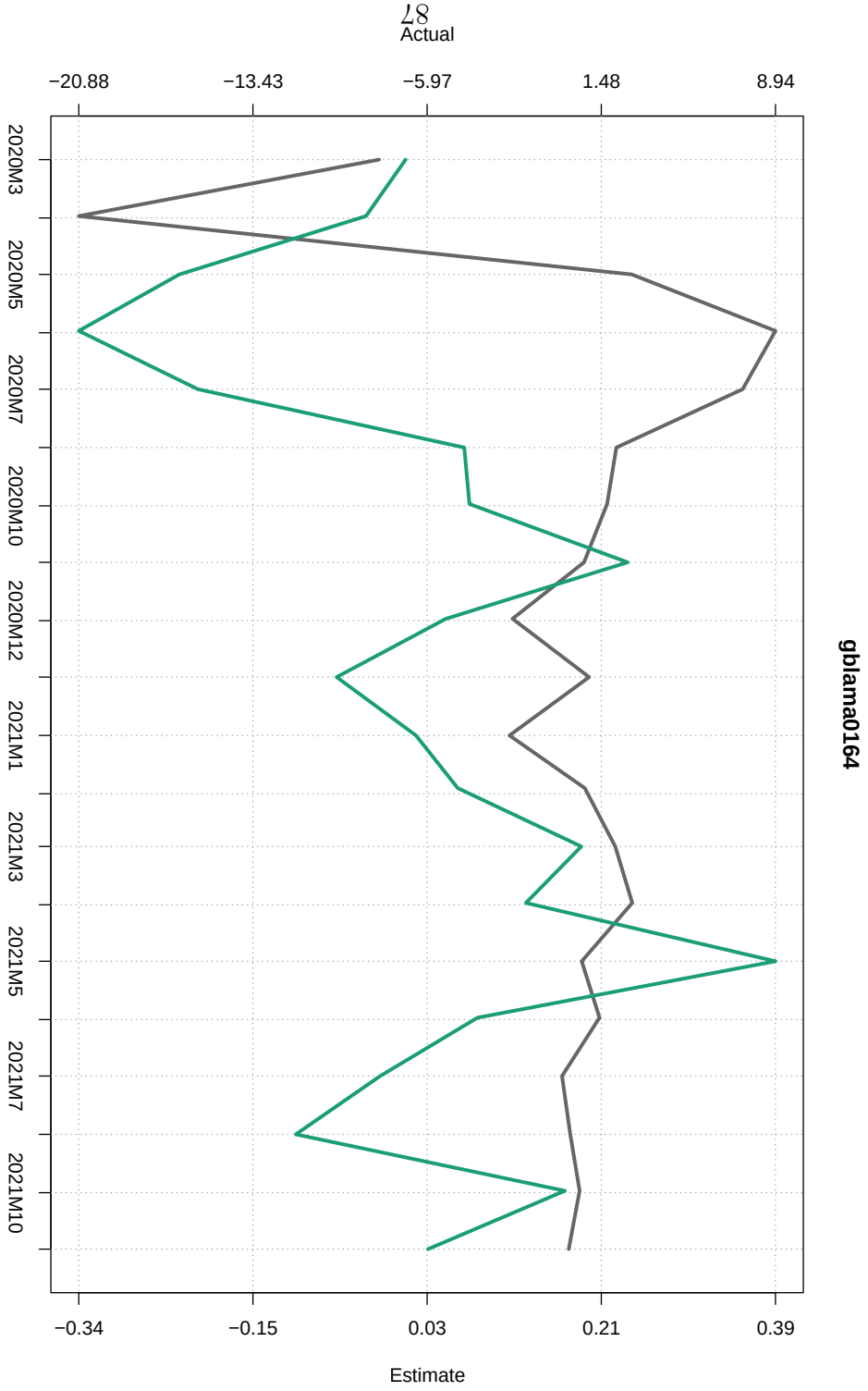
gblama0084
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.33
−0.21
−0.1
0.01
0.12
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

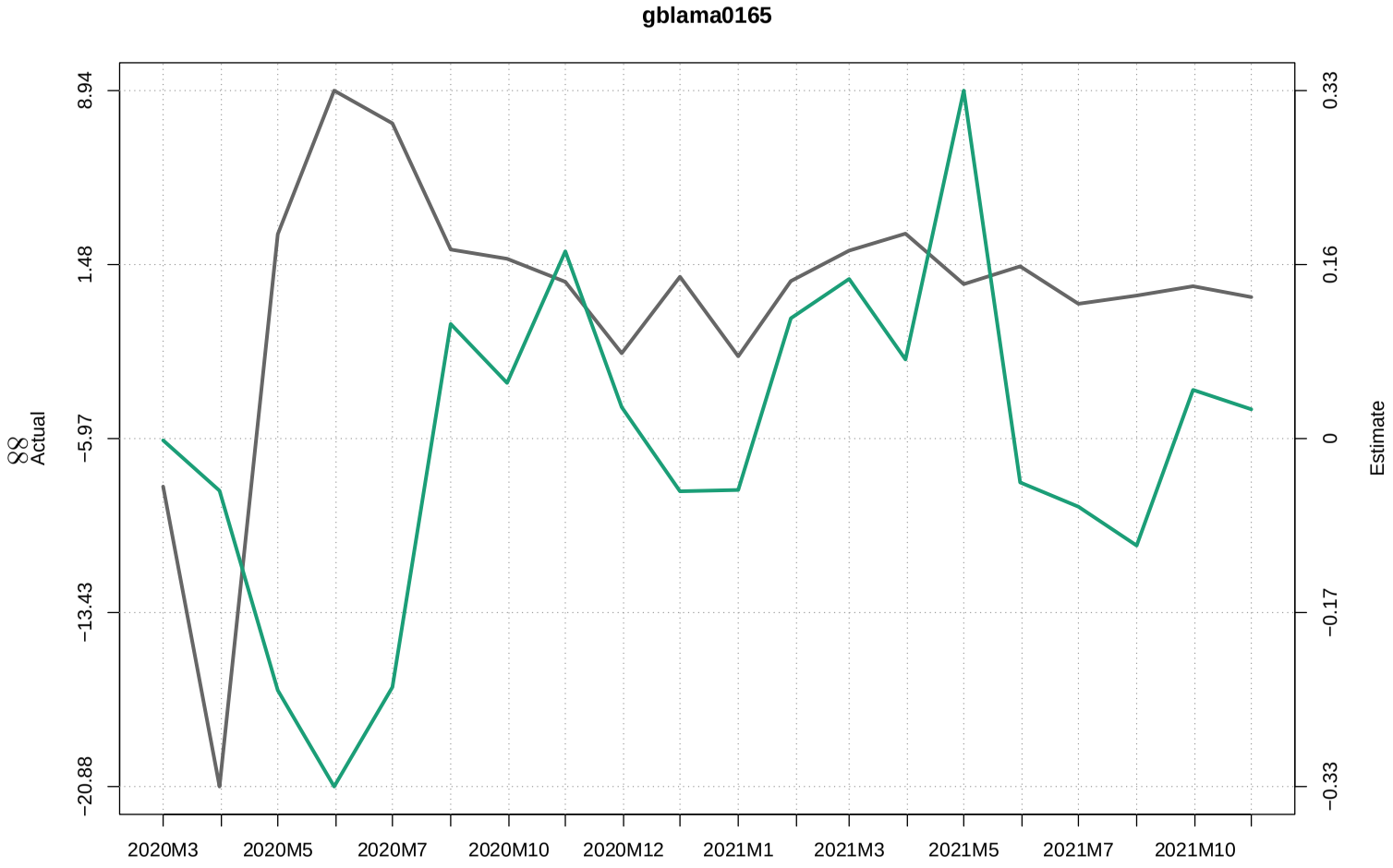

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0166
−0.49
−0.36
−0.23
−0.1
0.03
Estimate

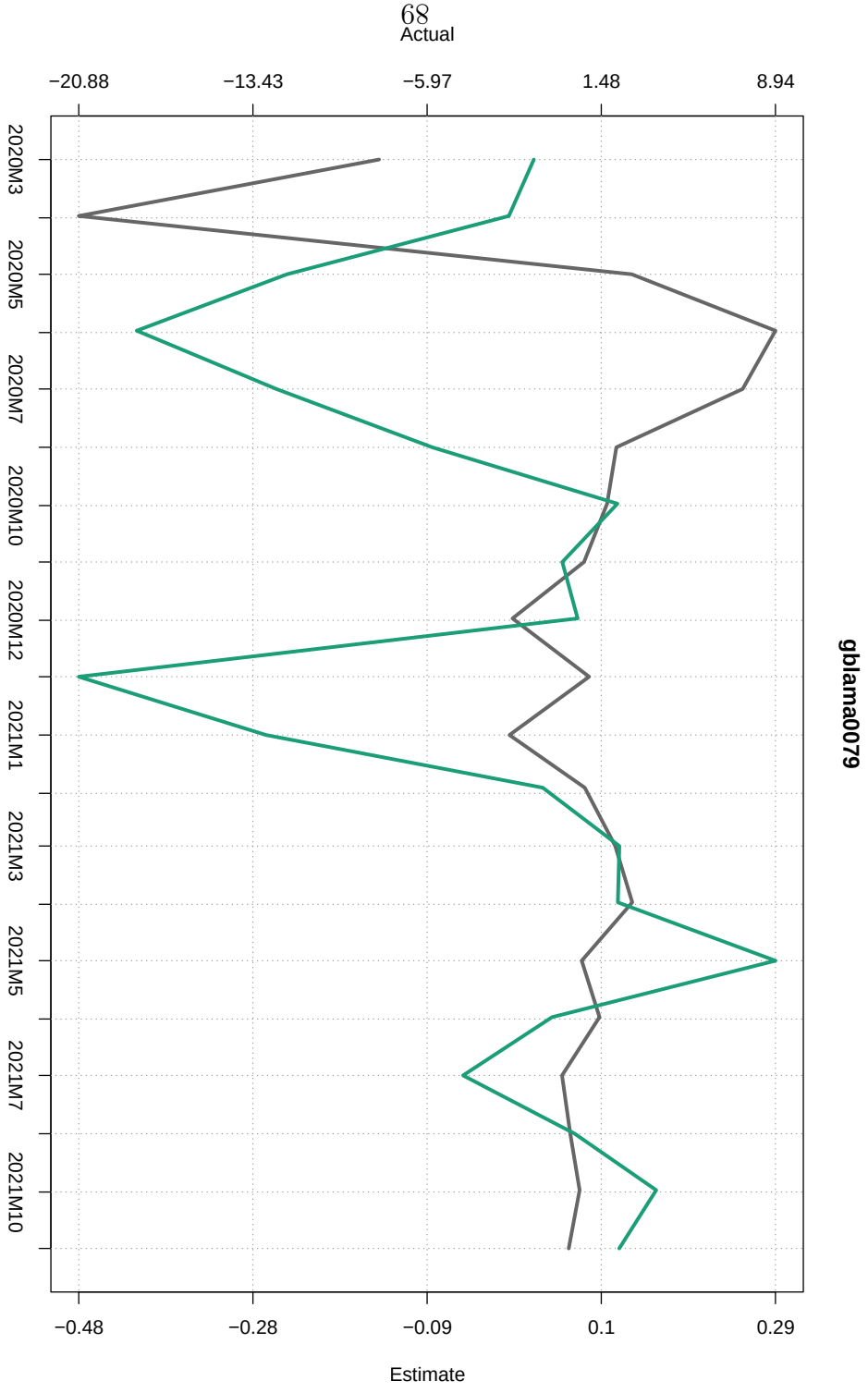
gblama0100
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.27
−0.08
0.11
0.3
0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

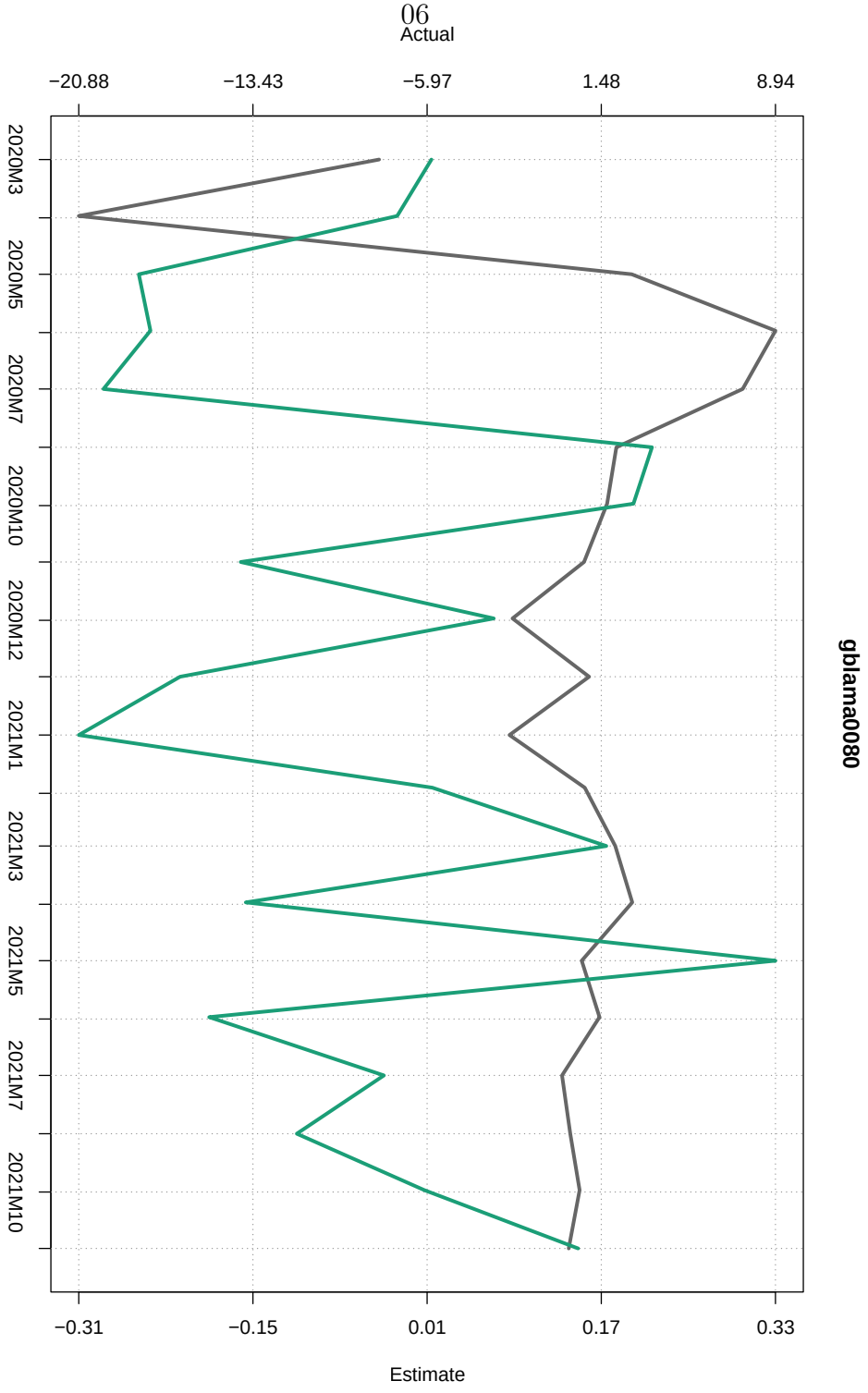

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0089
−0.45
−0.31
−0.17
−0.02
0.12
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0155
−0.28
−0.11
0.05
0.21
0.37
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0164
−0.34
−0.15
0.03
0.21
0.39
Estimate


gblama0165
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.33
0.16
0
−0.17
−0.33
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0079
−0.48
−0.28
−0.09
0.1
0.29
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gblama0080
−0.31
−0.15
0.01
0.17
0.33
Estimate

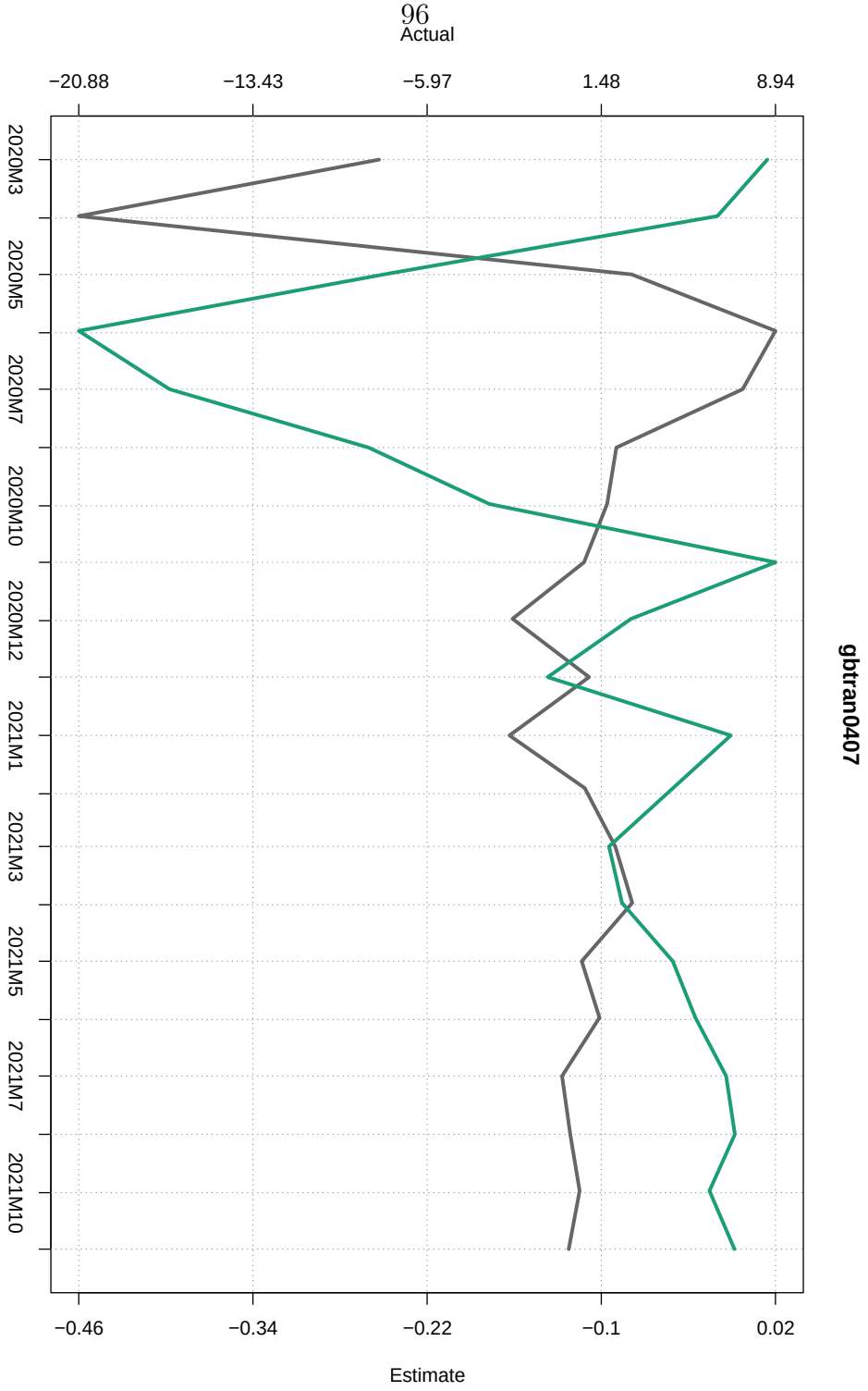
gblama0154
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.31
−0.14
0.03
0.19
0.36
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

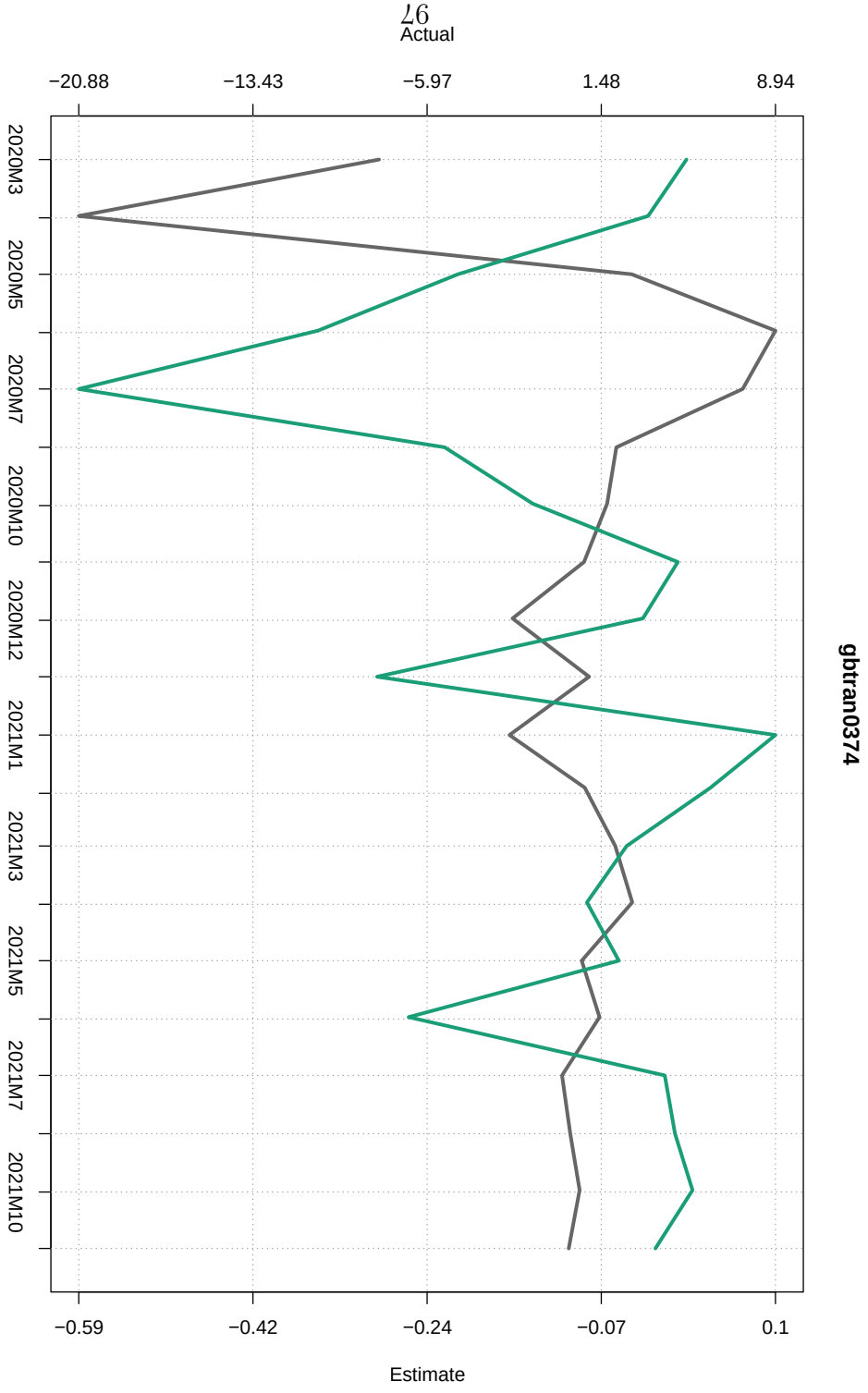
gblama0161
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.33
−0.16
0.01
0.18
0.34
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

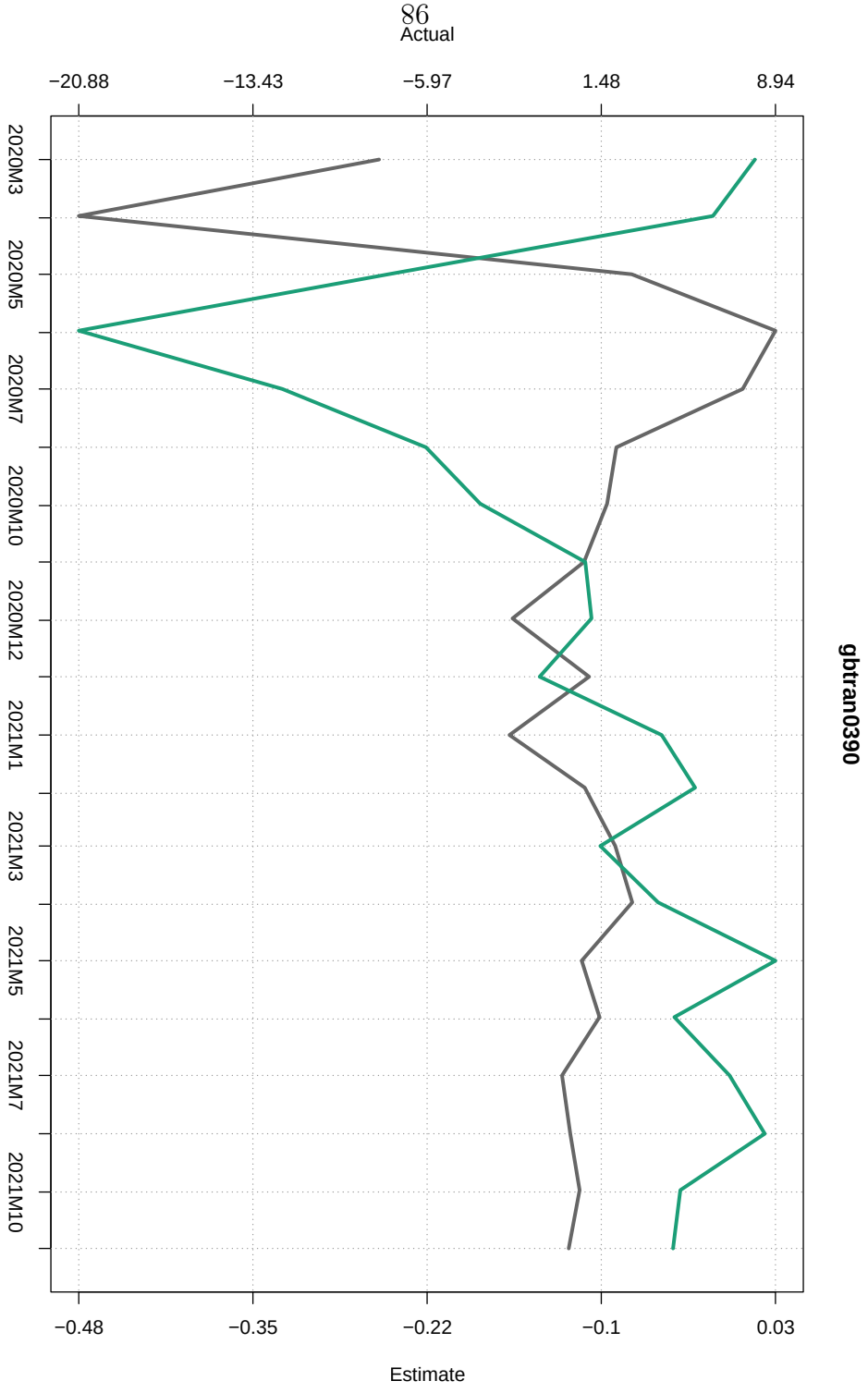
gblama0077
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.33
−0.12
0.09
0.3
0.5
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

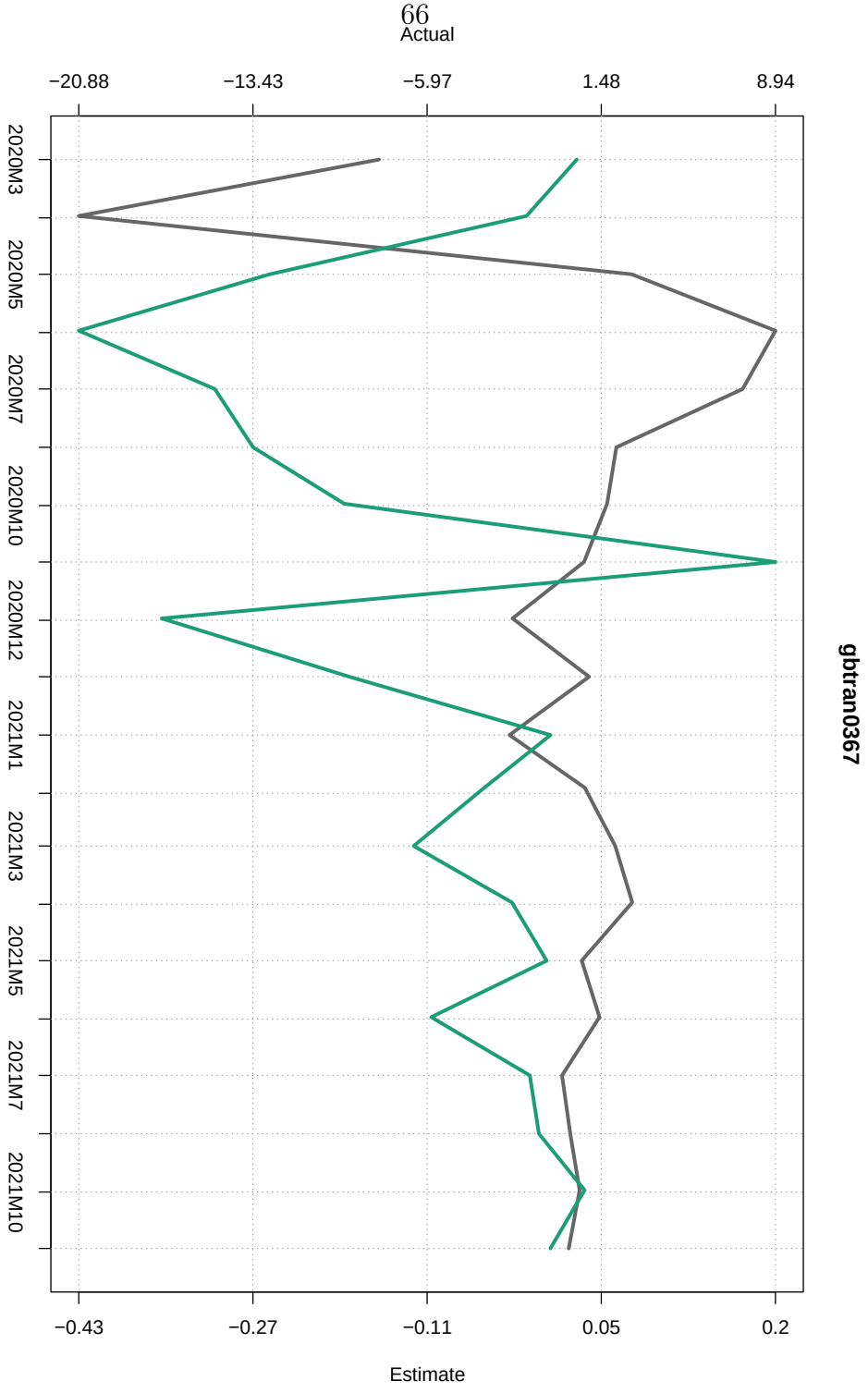
gblama0159
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.27
−0.11
0.04
0.2
0.35
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

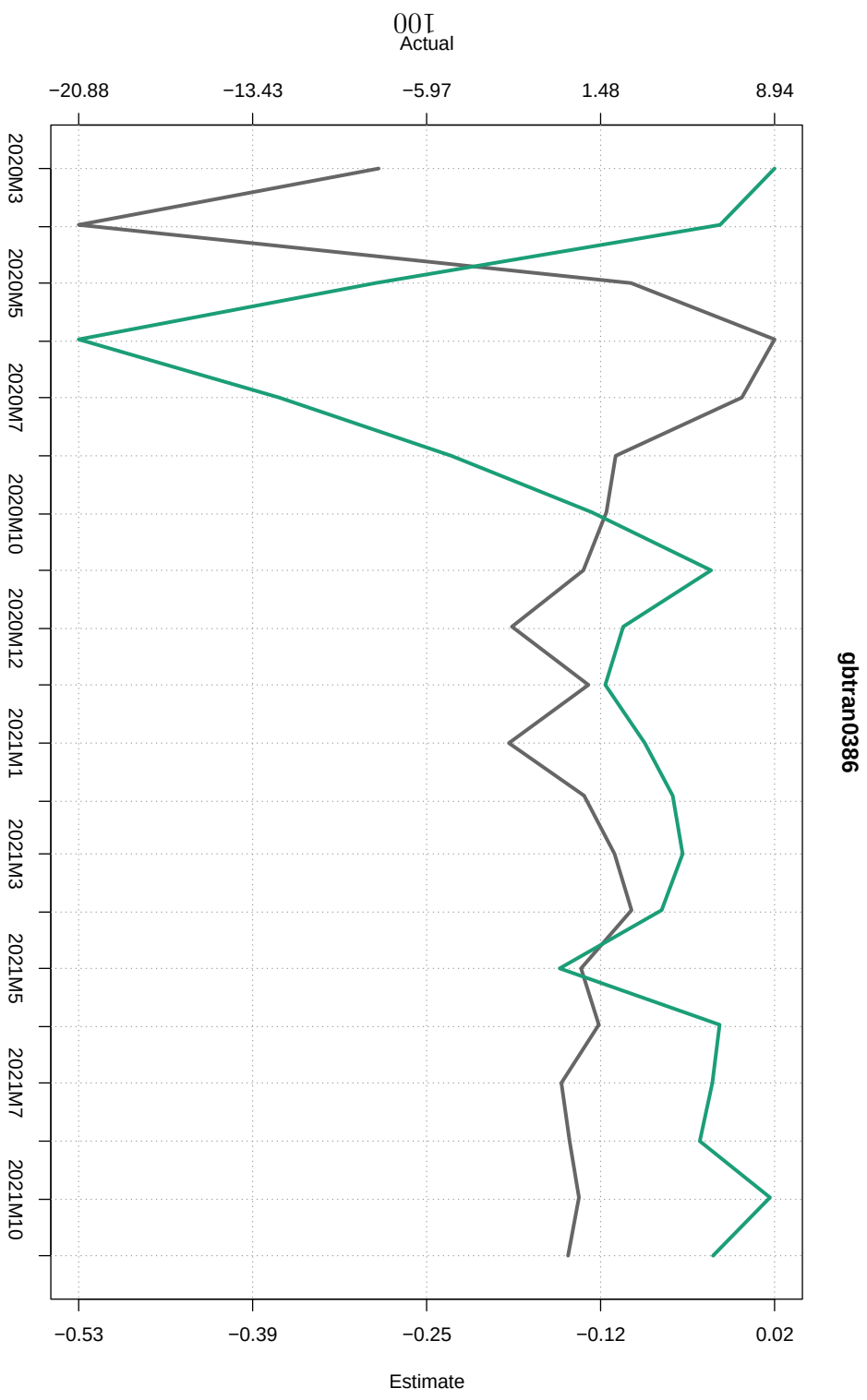
gbtran0376
Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
Estimate
−0.47
−0.35
−0.22
−0.1
0.02

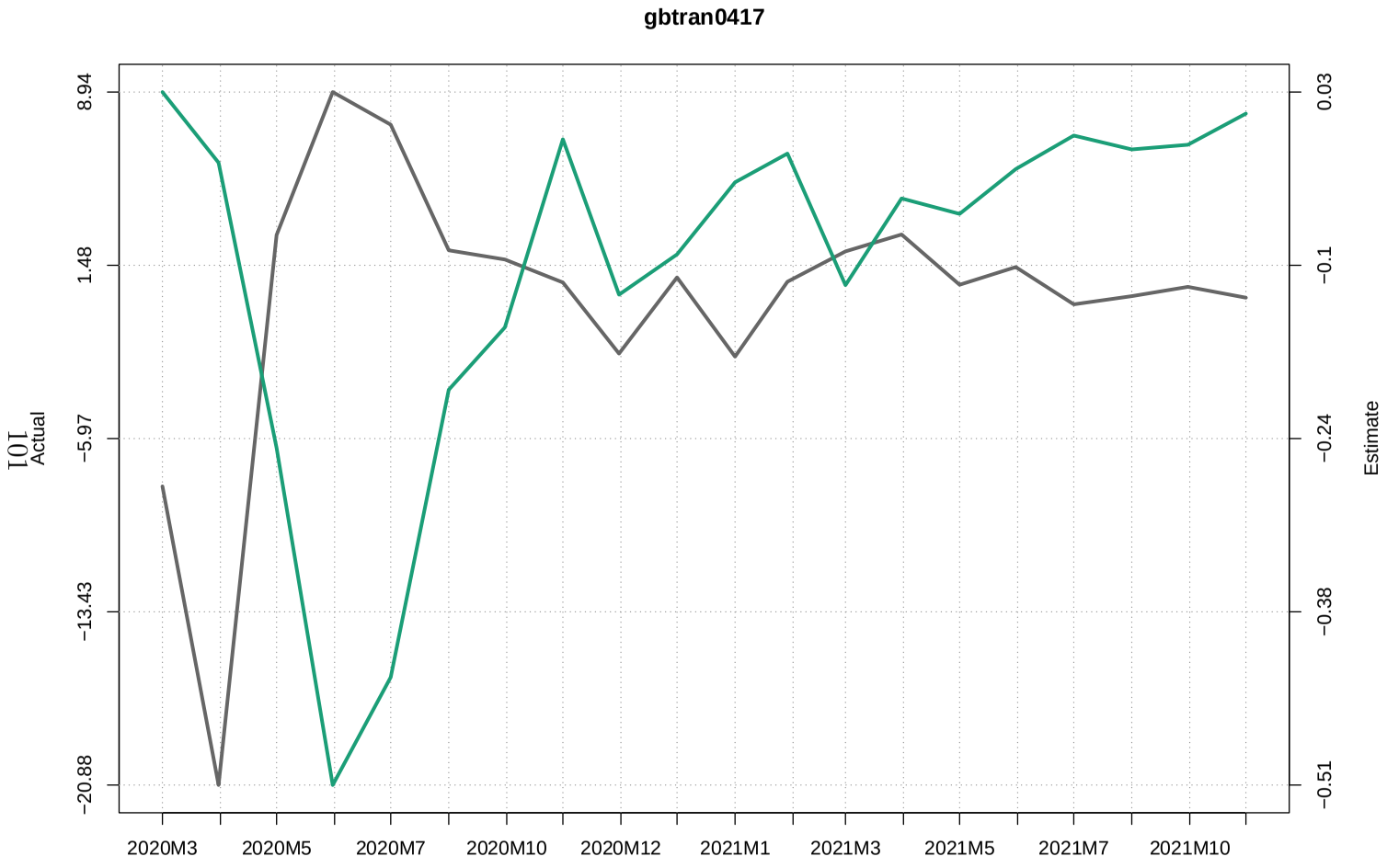

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.46
−0.34
−0.22
−0.1
0.02
Estimate
gbtran0407


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.59
−0.42
−0.24
−0.07
0.1
Estimate
gbtran0374

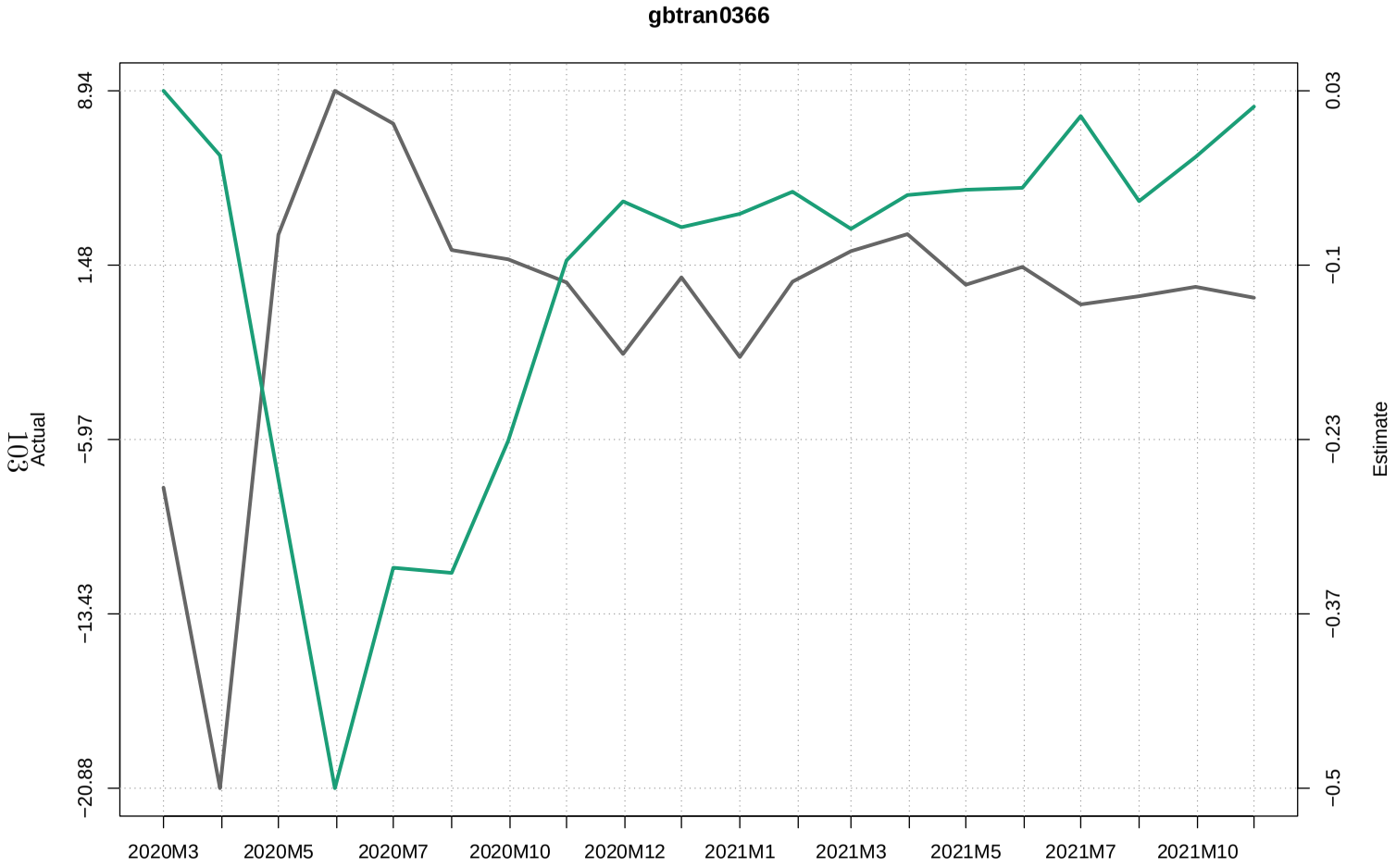

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gbtran0390
−0.48
−0.35
−0.22
−0.1
0.03
Estimate


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.43
−0.27
−0.11
0.05
0.2
Estimate
gbtran0367


100
Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
gbtran0386
−0.53
−0.39
−0.25
−0.12
0.02
Estimate

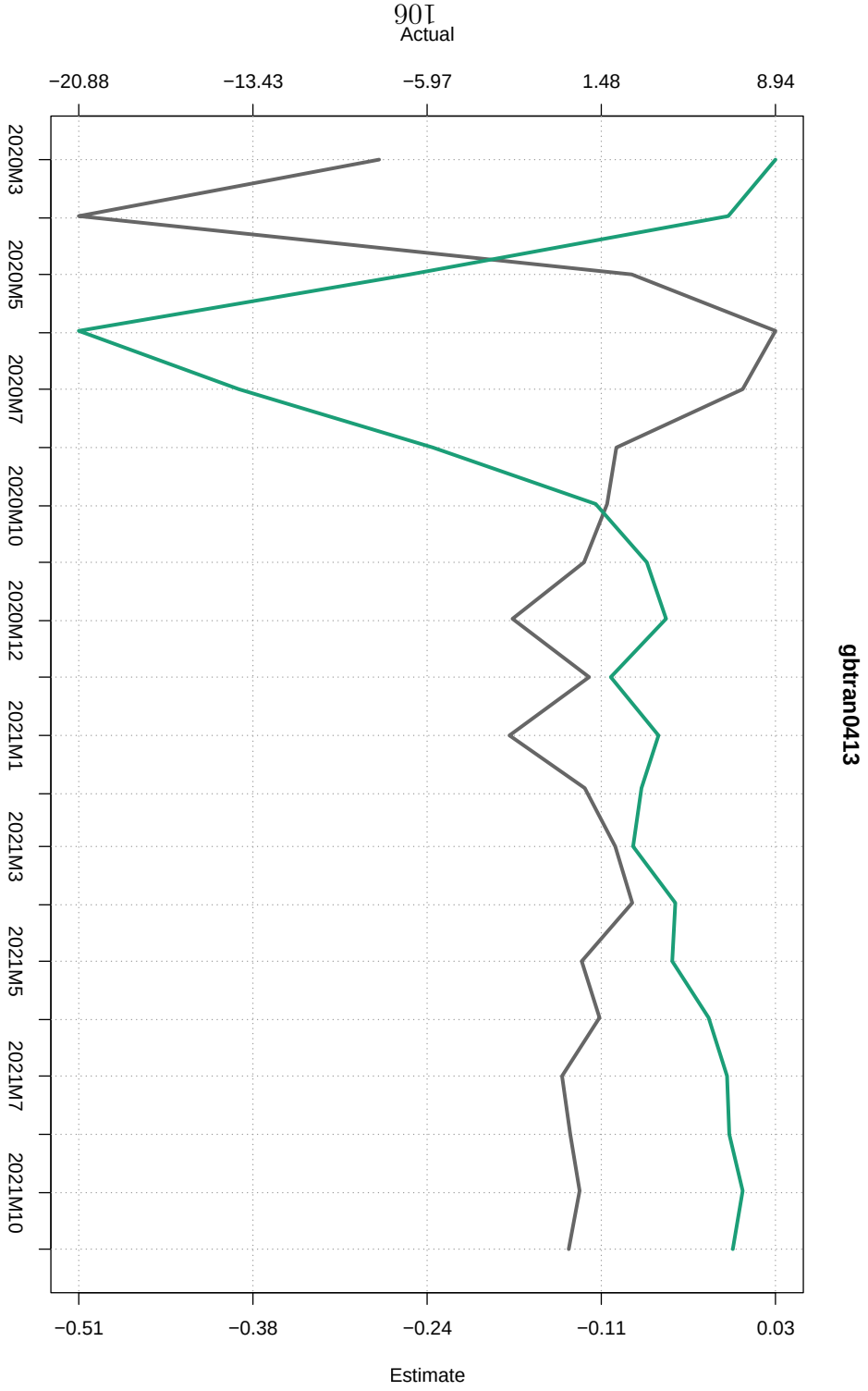

gbtran0417
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.03
−0.1
−0.24
−0.38
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

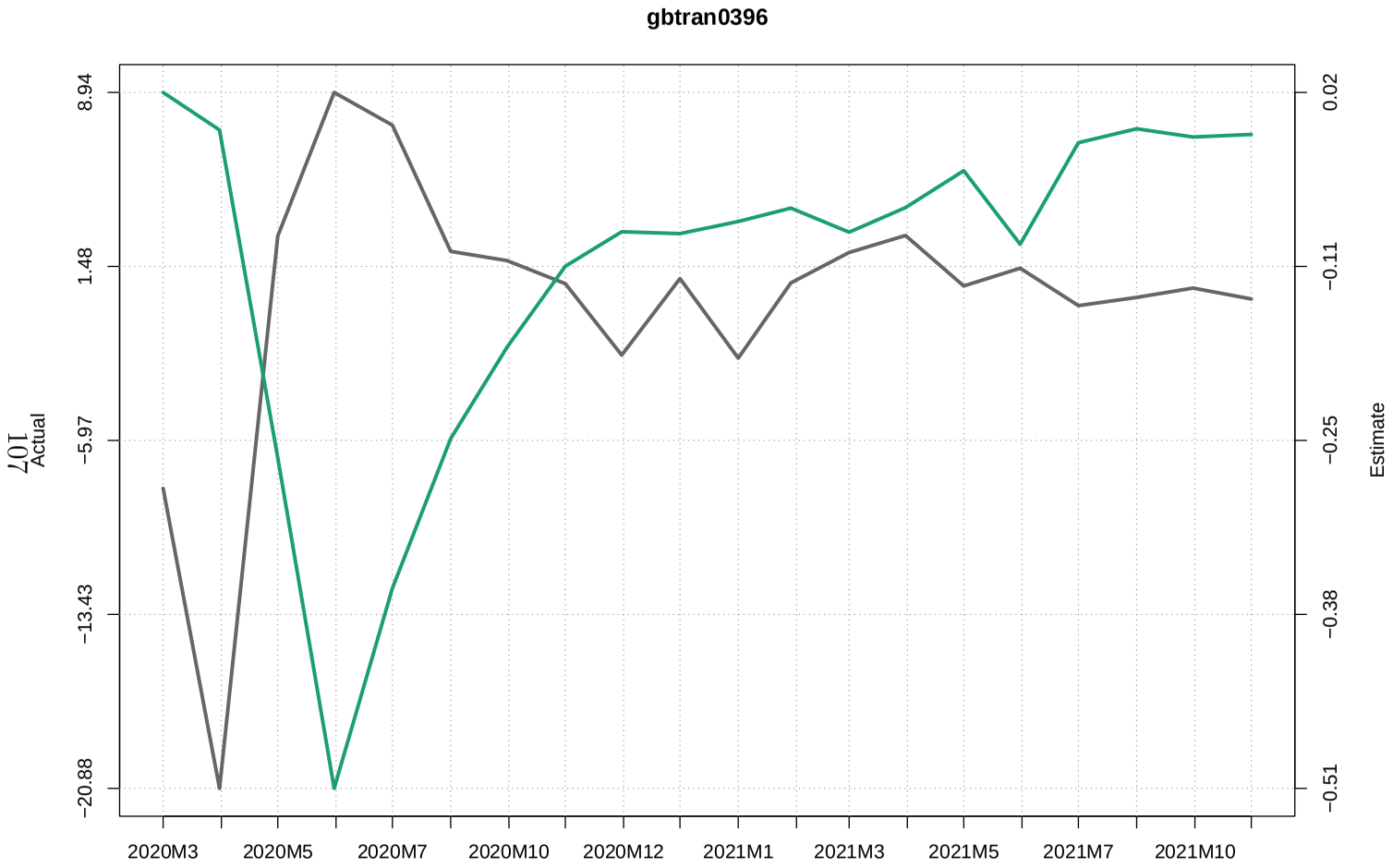
gbtran0400
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.51
−0.37
−0.24
−0.1
0.03
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

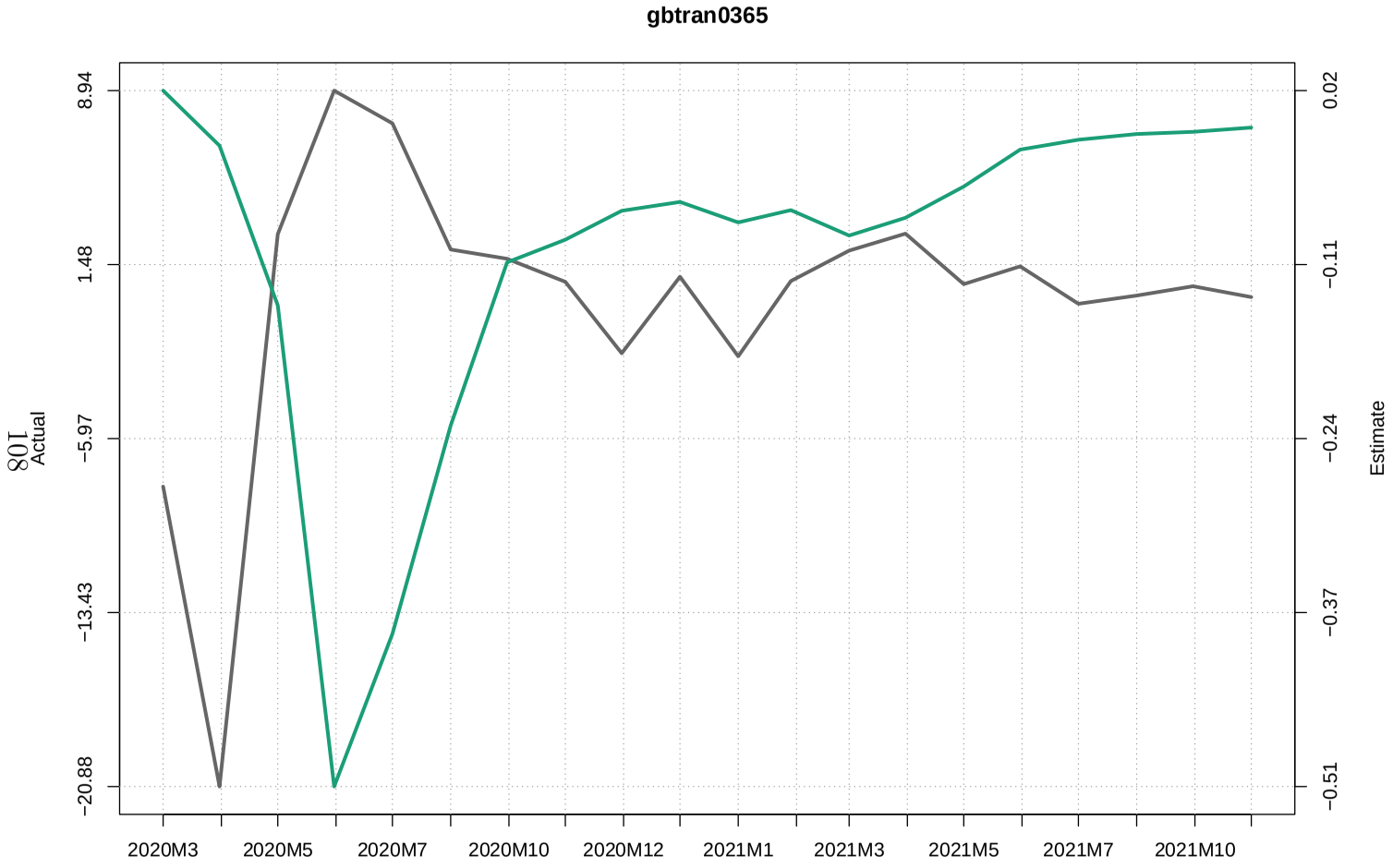

gbtran0366
Actual
103
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.03
−0.1
−0.23
−0.37
−0.5
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

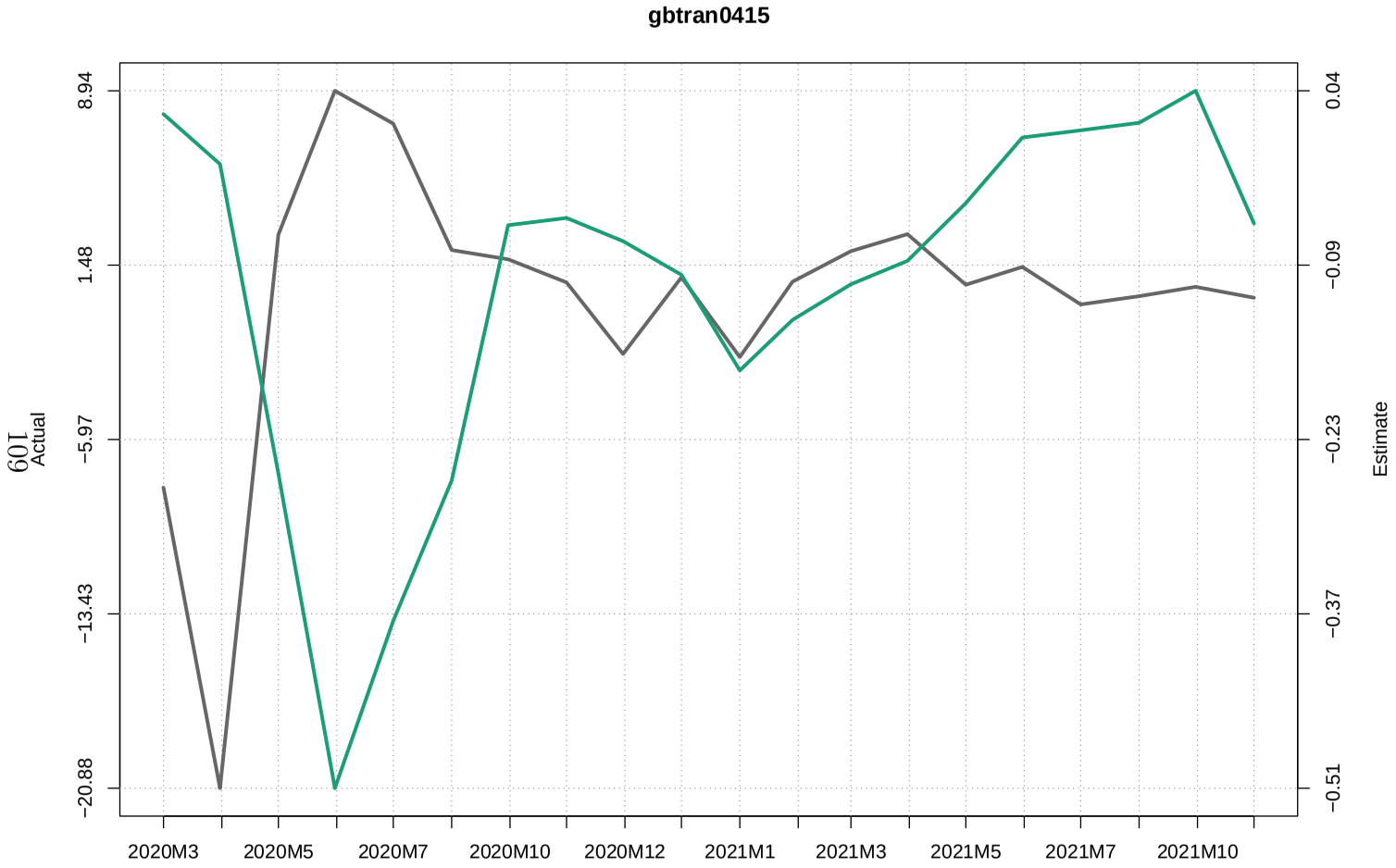
gbtran0399
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

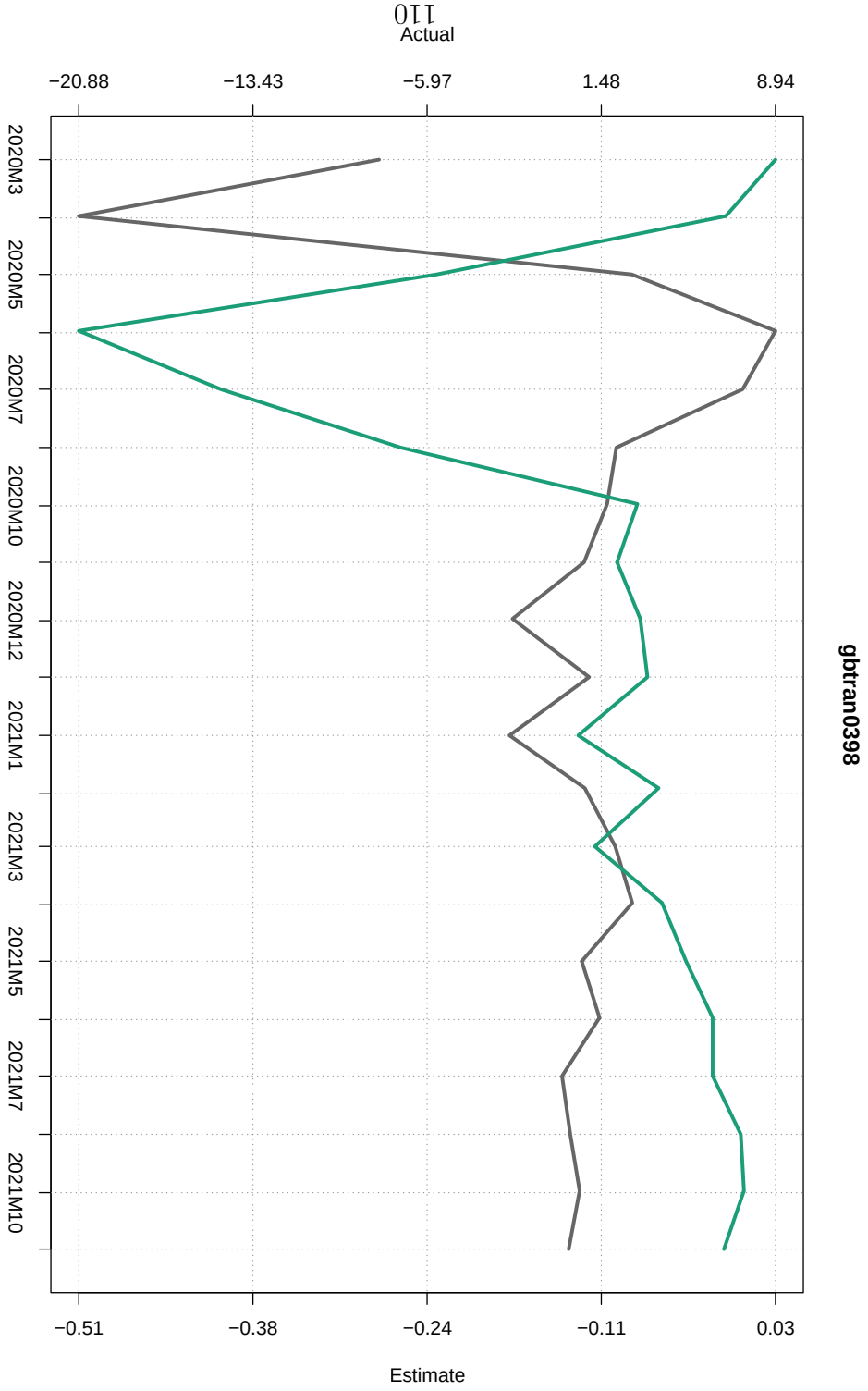
gbtran0363
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.12
−0.26
−0.4
−0.54
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

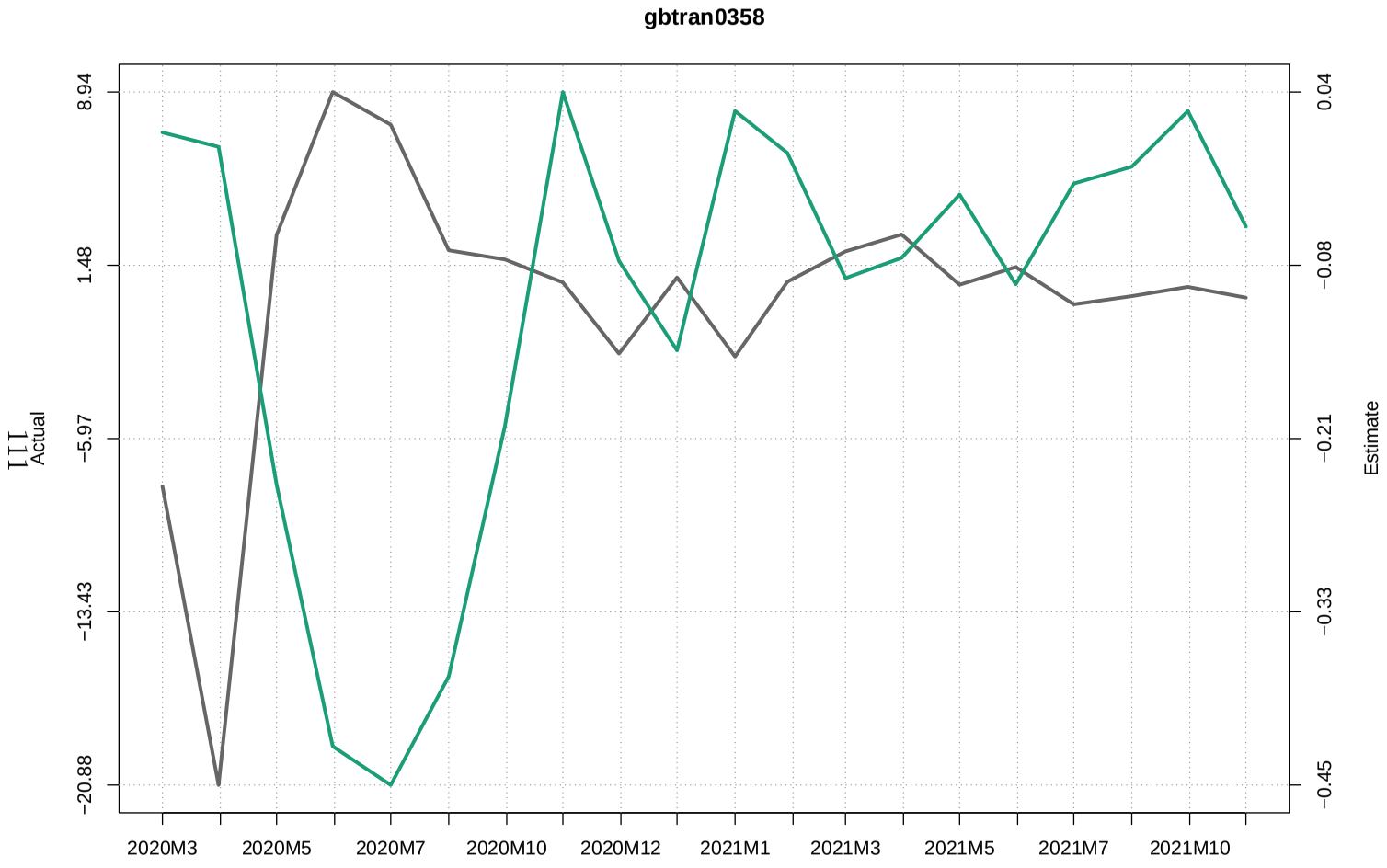

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.51
−0.38
−0.24
−0.11
0.03
Estimate
gbtran0413

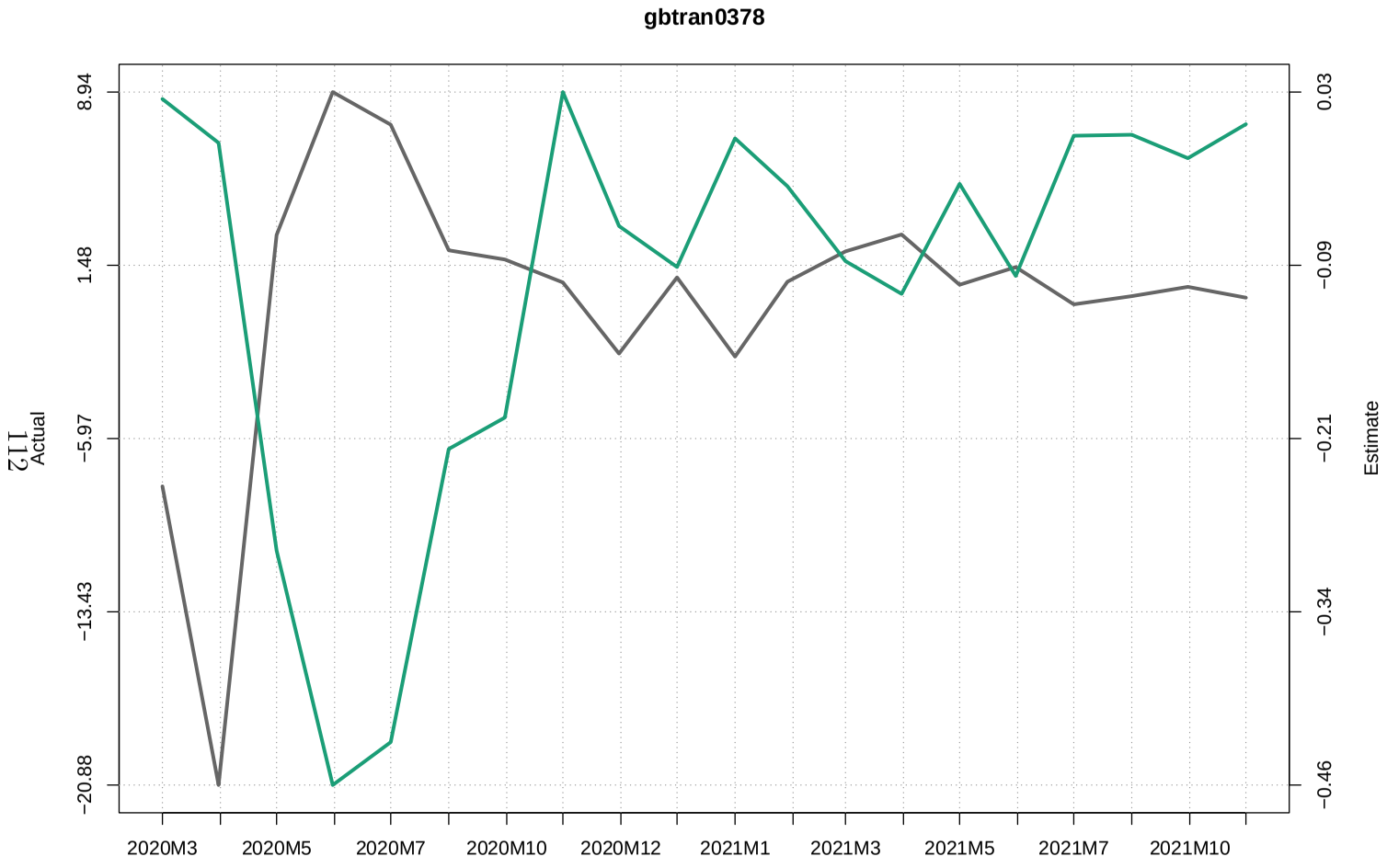

gbtran0396
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.25
−0.38
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

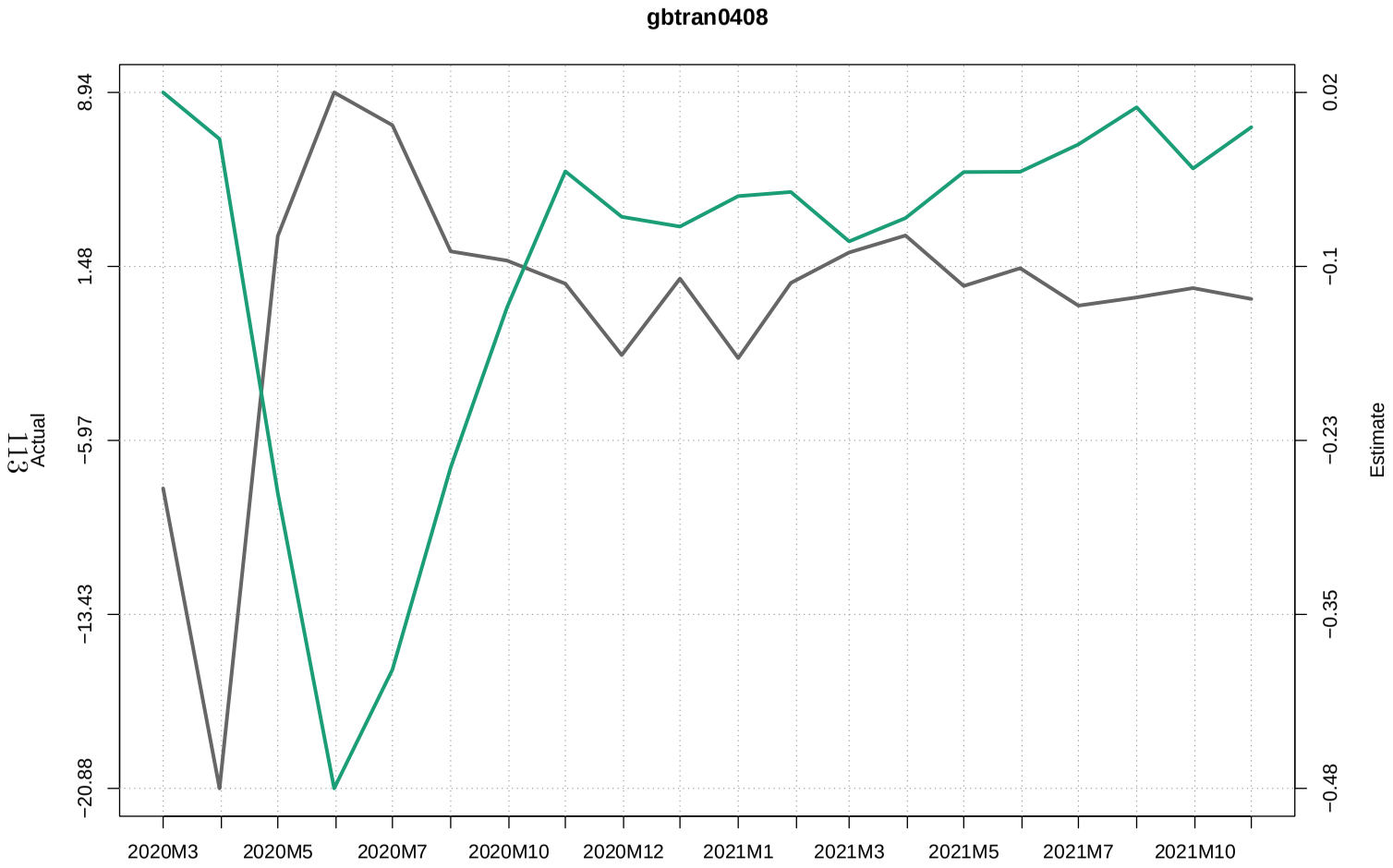

gbtran0365
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.24
−0.37
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

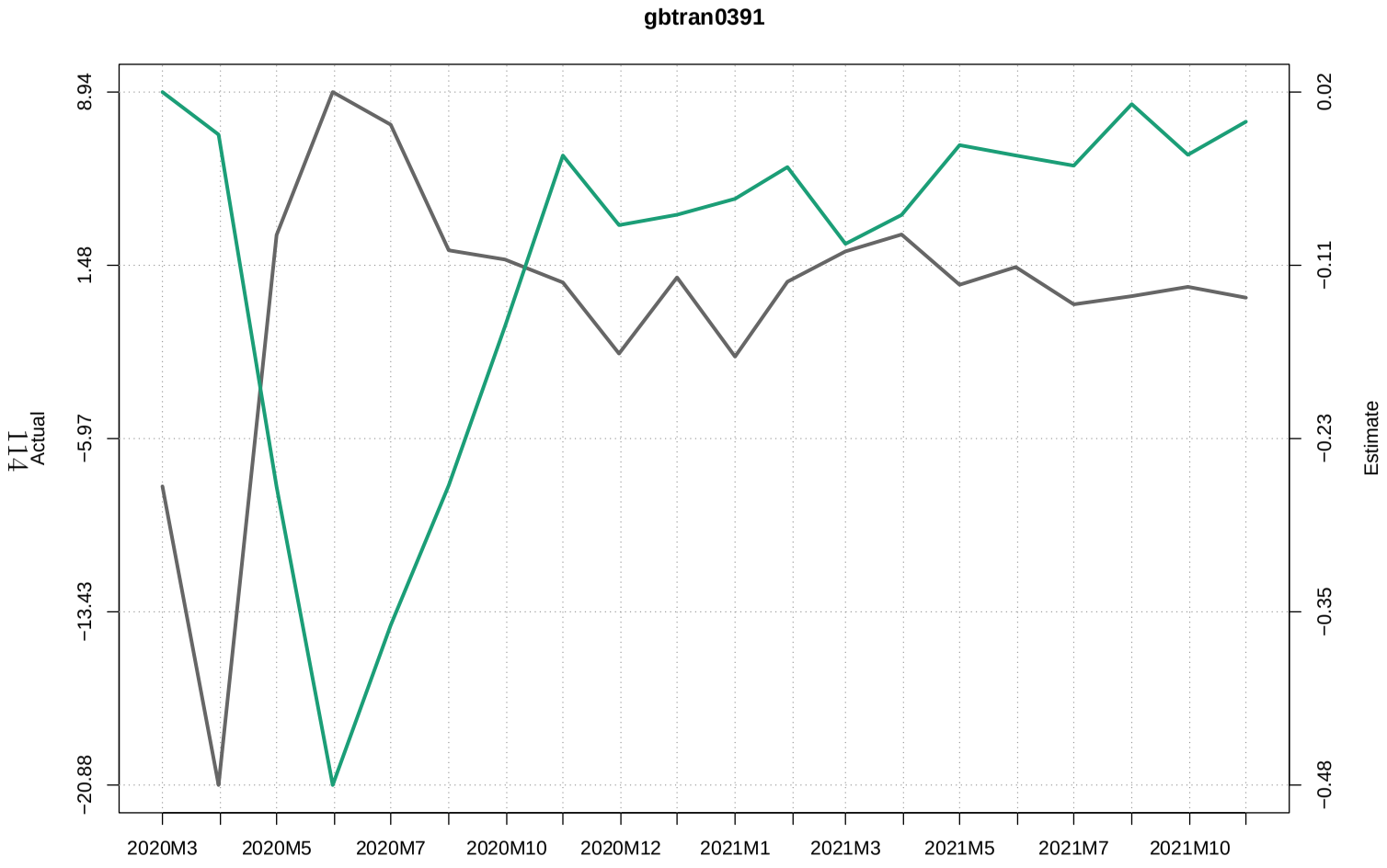

gbtran0415
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.04
−0.09
−0.23
−0.37
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.51
−0.38
−0.24
−0.11
0.03
Estimate
gbtran0398

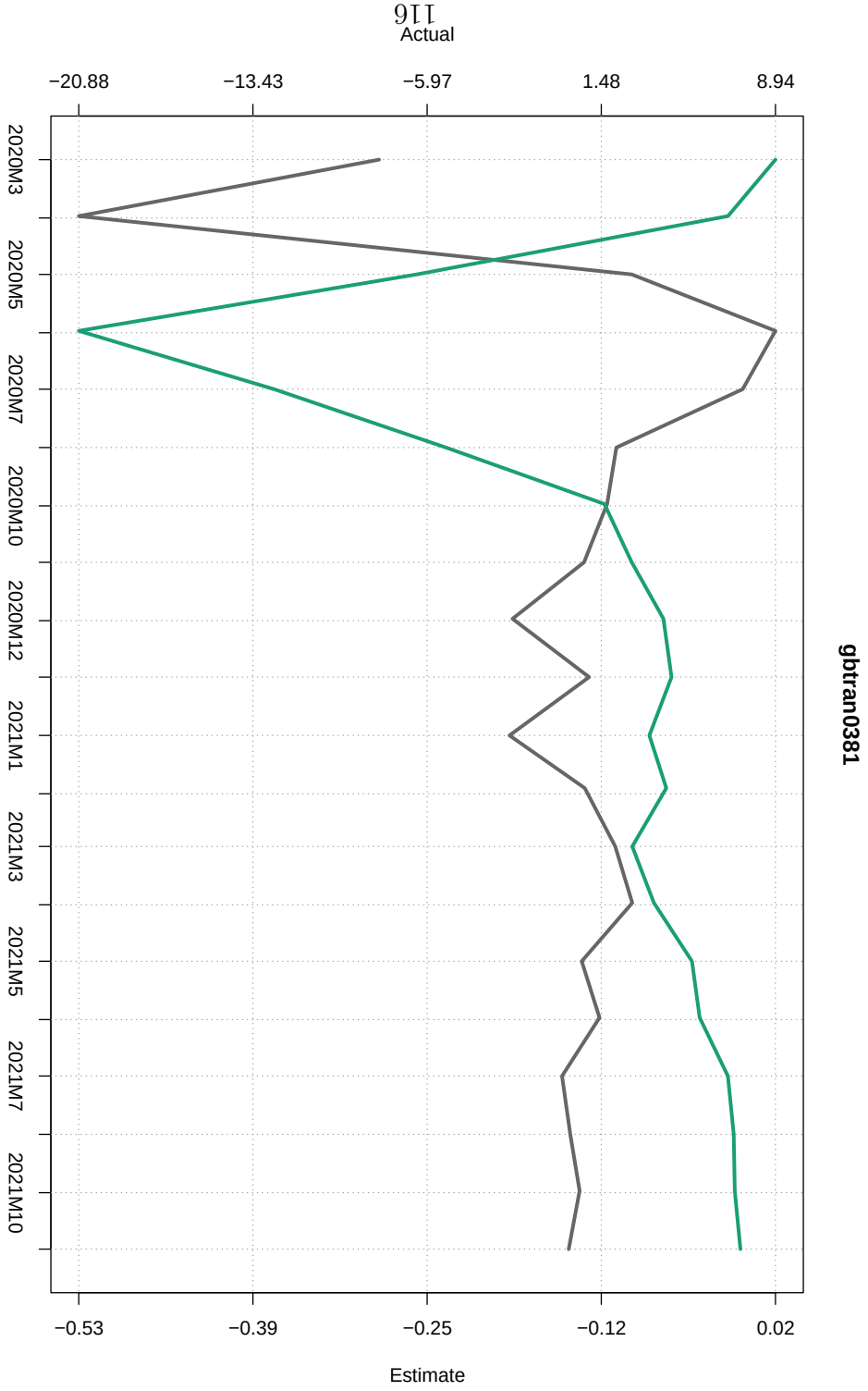

gbtran0358
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.04
−0.08
−0.21
−0.33
−0.45
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0378
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.03
−0.09
−0.21
−0.34
−0.46
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

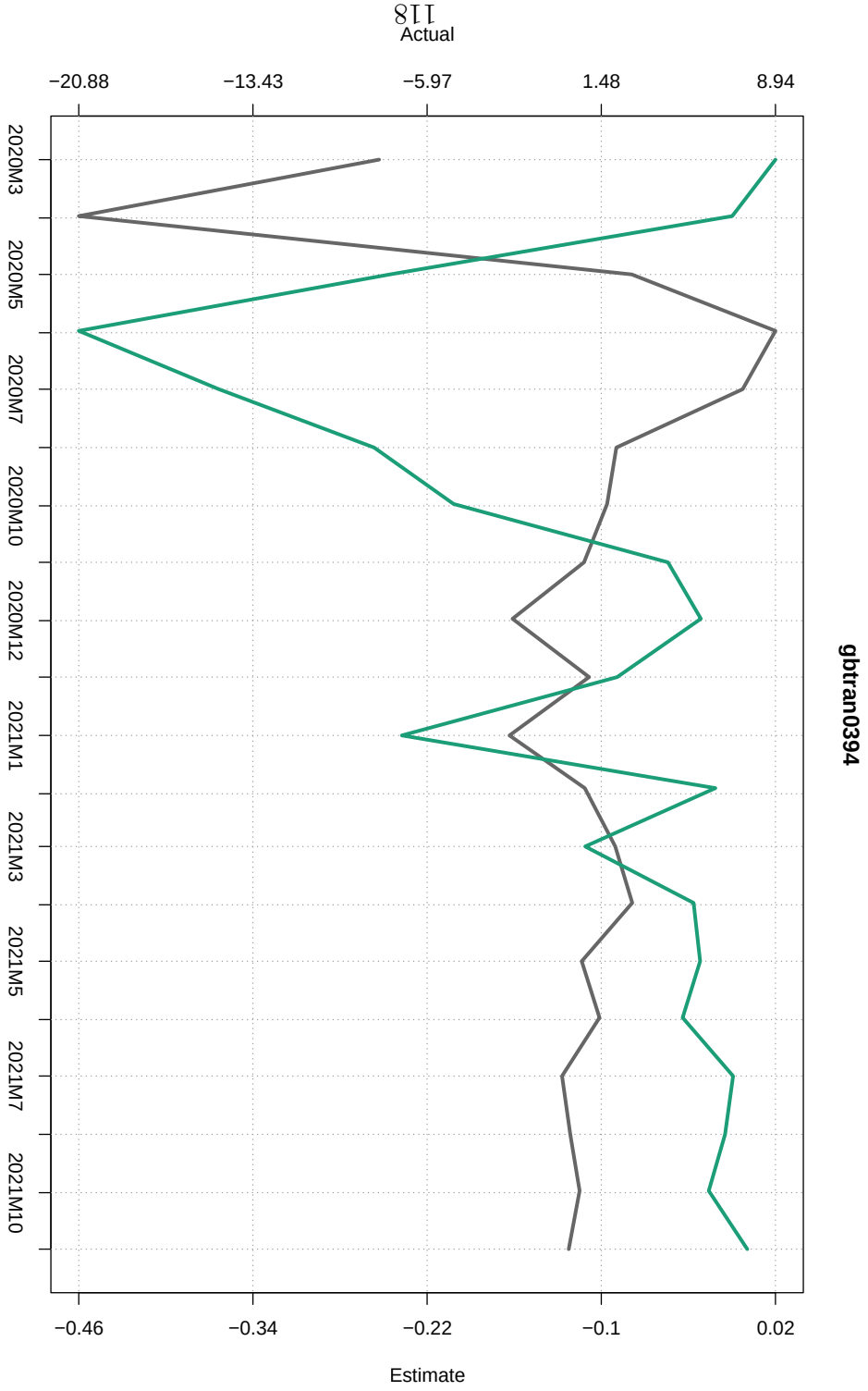

gbtran0408
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.1
−0.23
−0.35
−0.48
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0391
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.11
−0.23
−0.35
−0.48
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gbtran0361
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.52
−0.38
−0.24
−0.11
0.03
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

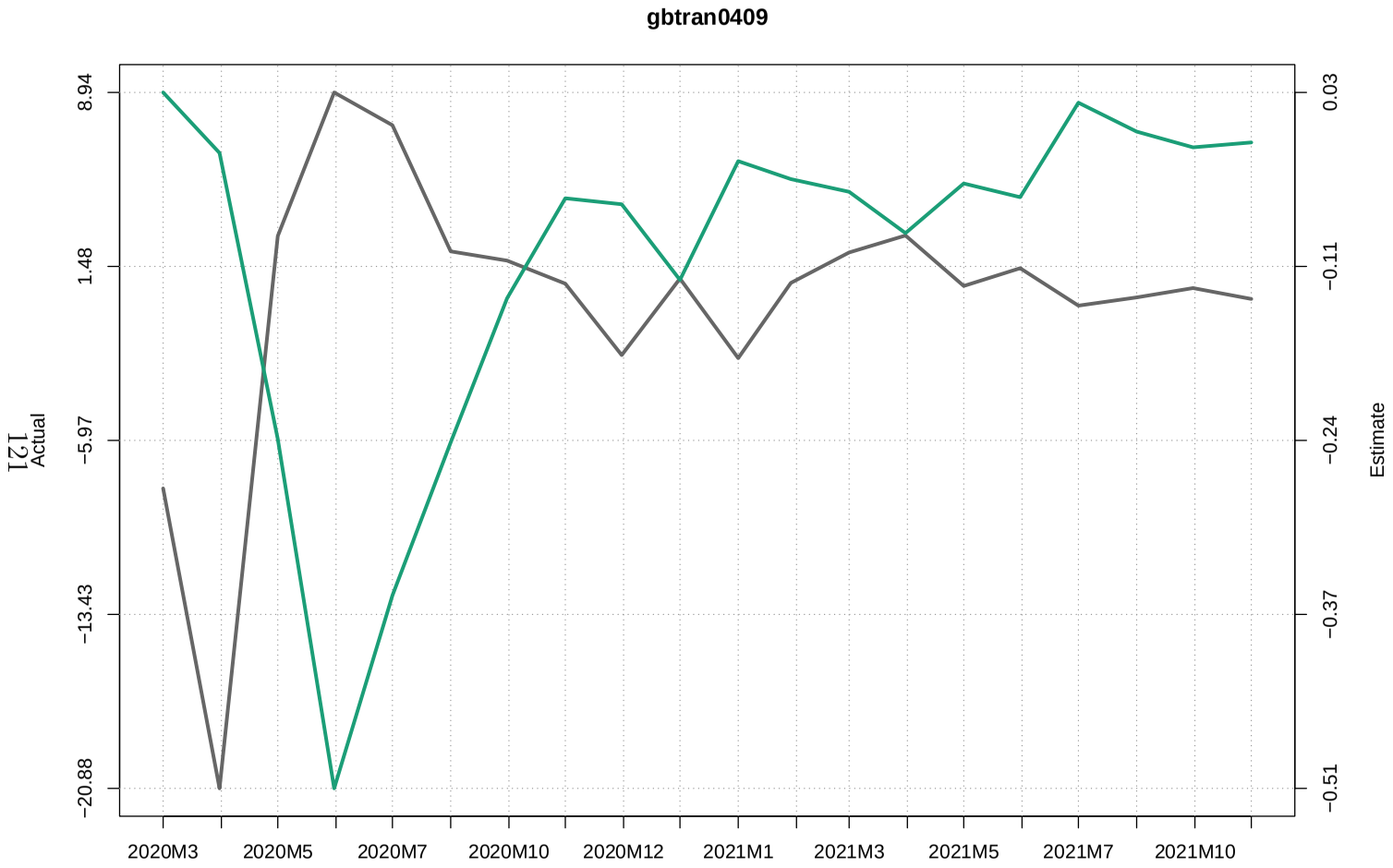

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.53
−0.39
−0.25
−0.12
0.02
Estimate
gbtran0381


gbtran0411
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.03
−0.1
−0.23
−0.36
−0.48
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

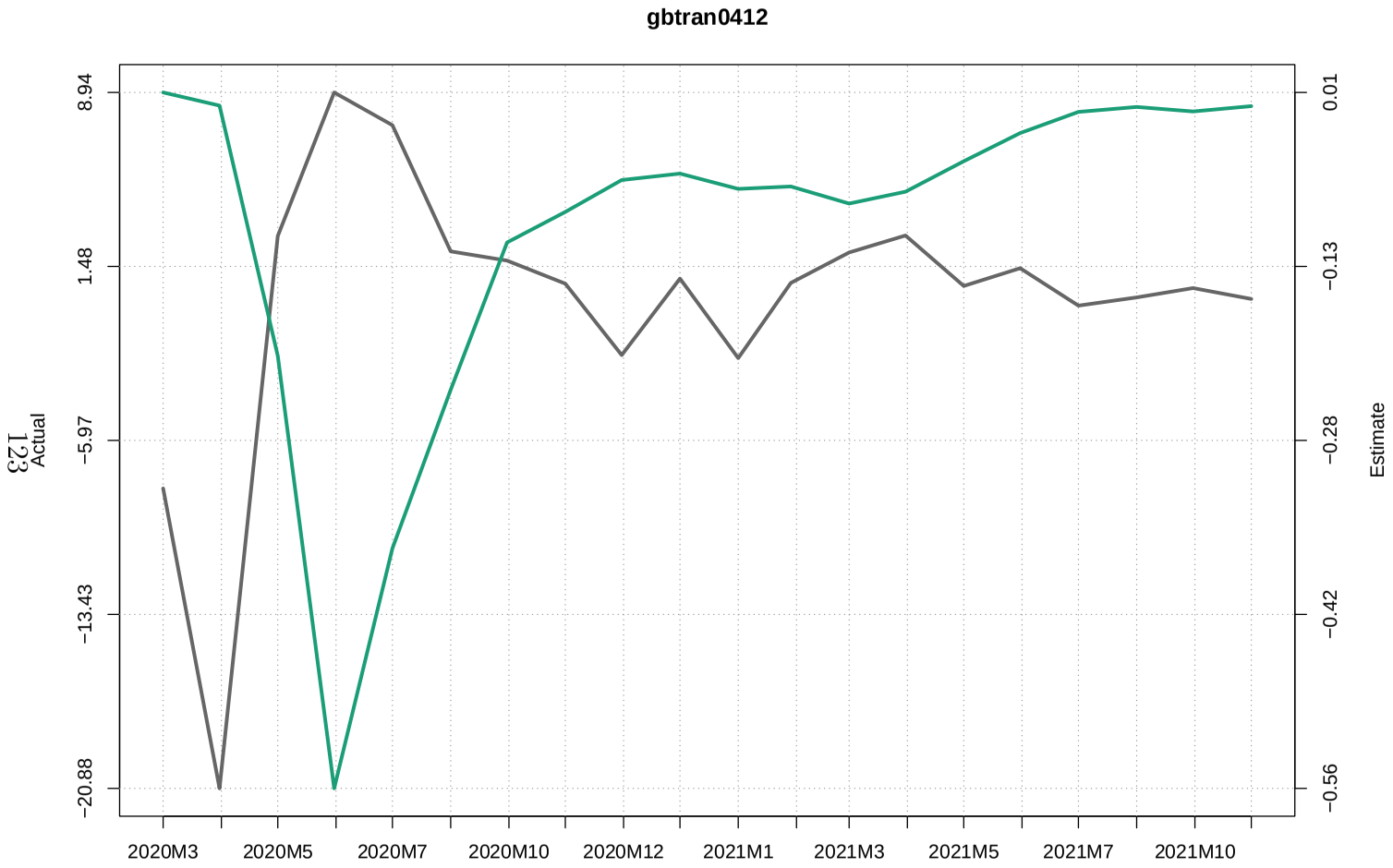

Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
−0.46
−0.34
−0.22
−0.1
0.02
Estimate
gbtran0394

gbtran0359
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.06
−0.09
−0.24
−0.4
−0.55
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gbtran0379
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.24
−0.36
−0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0409
Actual
121
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.03
−0.11
−0.24
−0.37
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

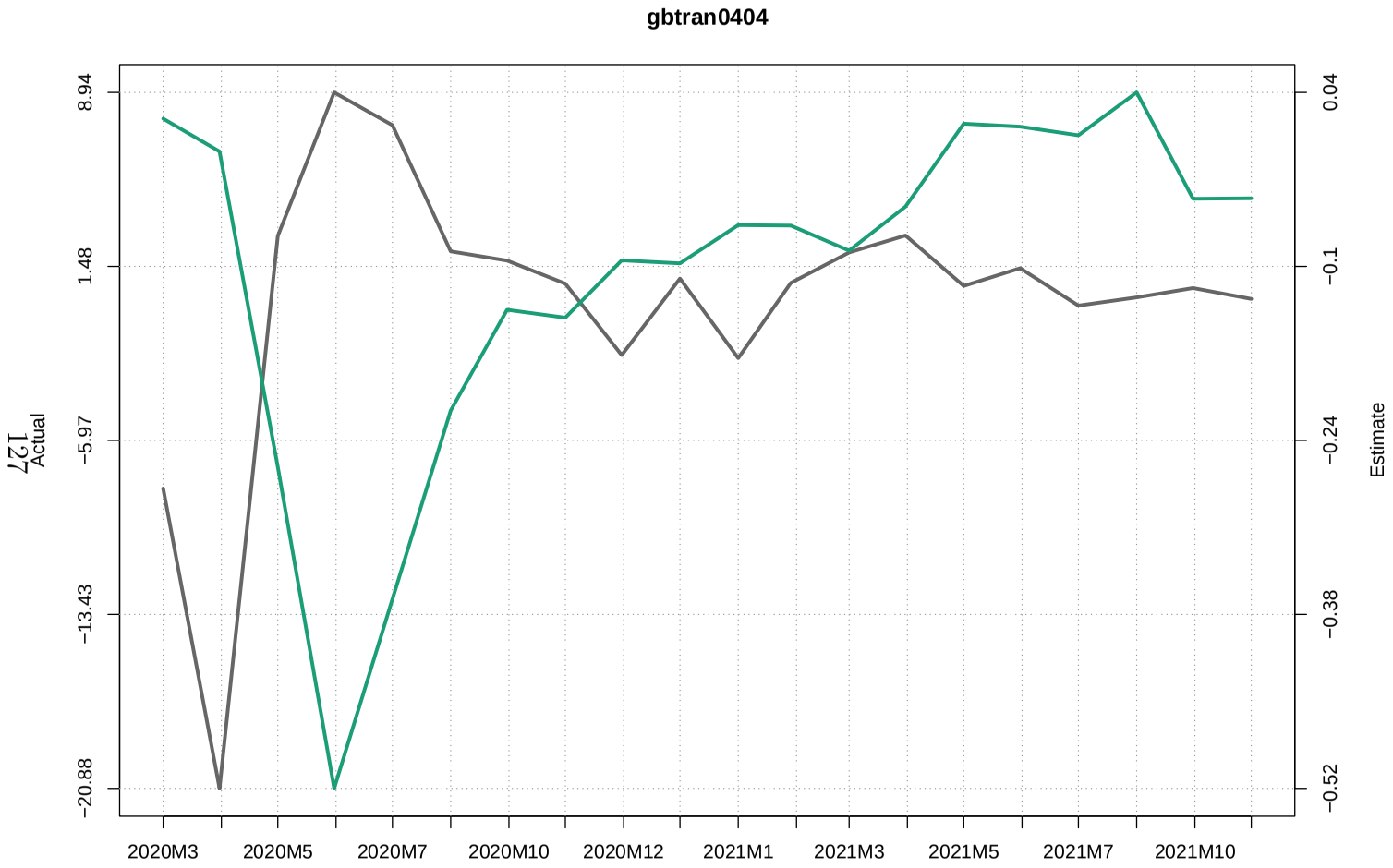
gbtran0392
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.01
−0.12
−0.25
−0.37
−0.5
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0412
Actual
123
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.01
−0.13
−0.28
−0.42
−0.56
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gbtran0395
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.12
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

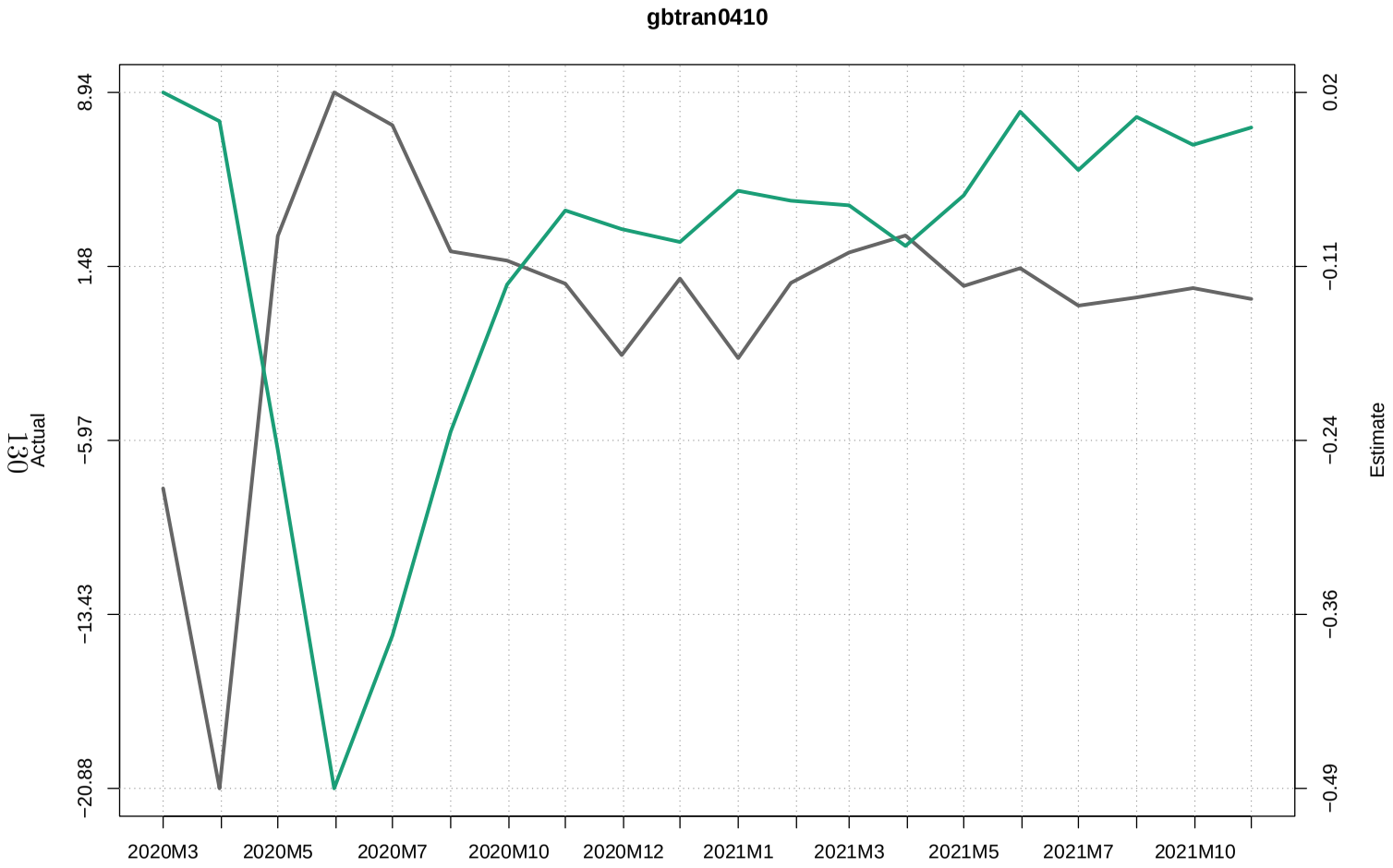
gbtran0371
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.49
−0.36
−0.24
−0.11
0.02
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

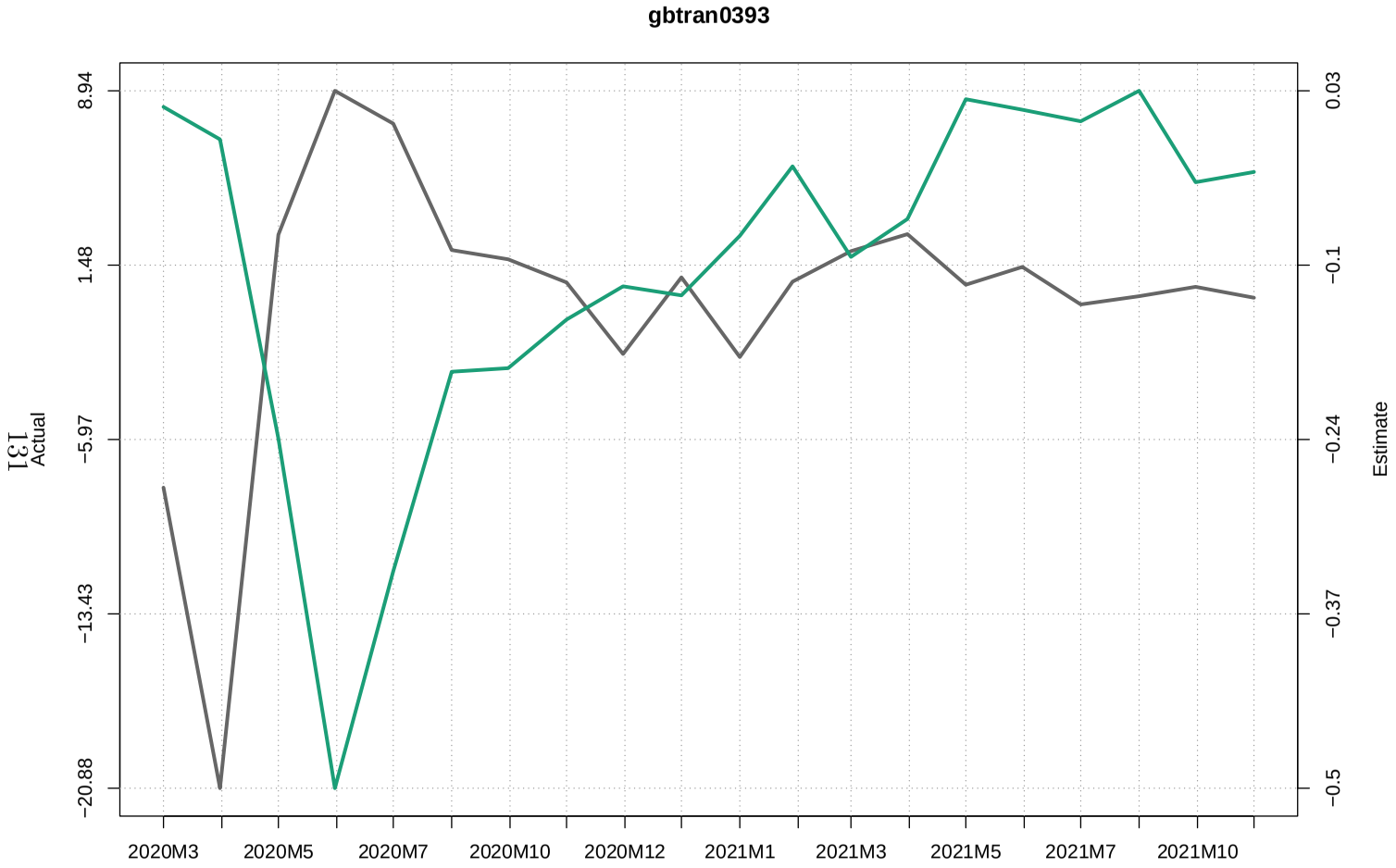
gbtran0419
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.01
−0.11
−0.24
−0.37
−0.5
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

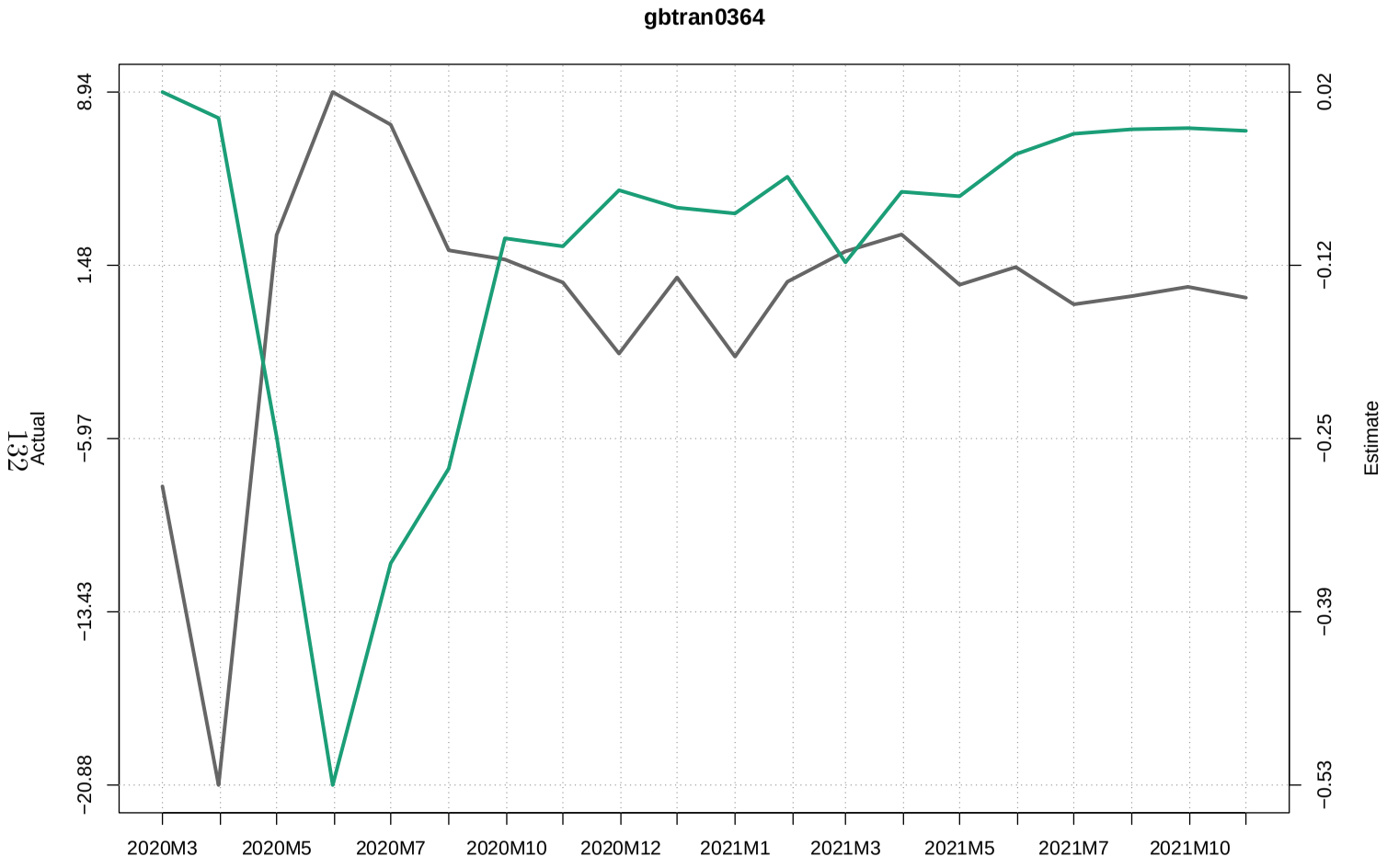

gbtran0404
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.04
−0.1
−0.24
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

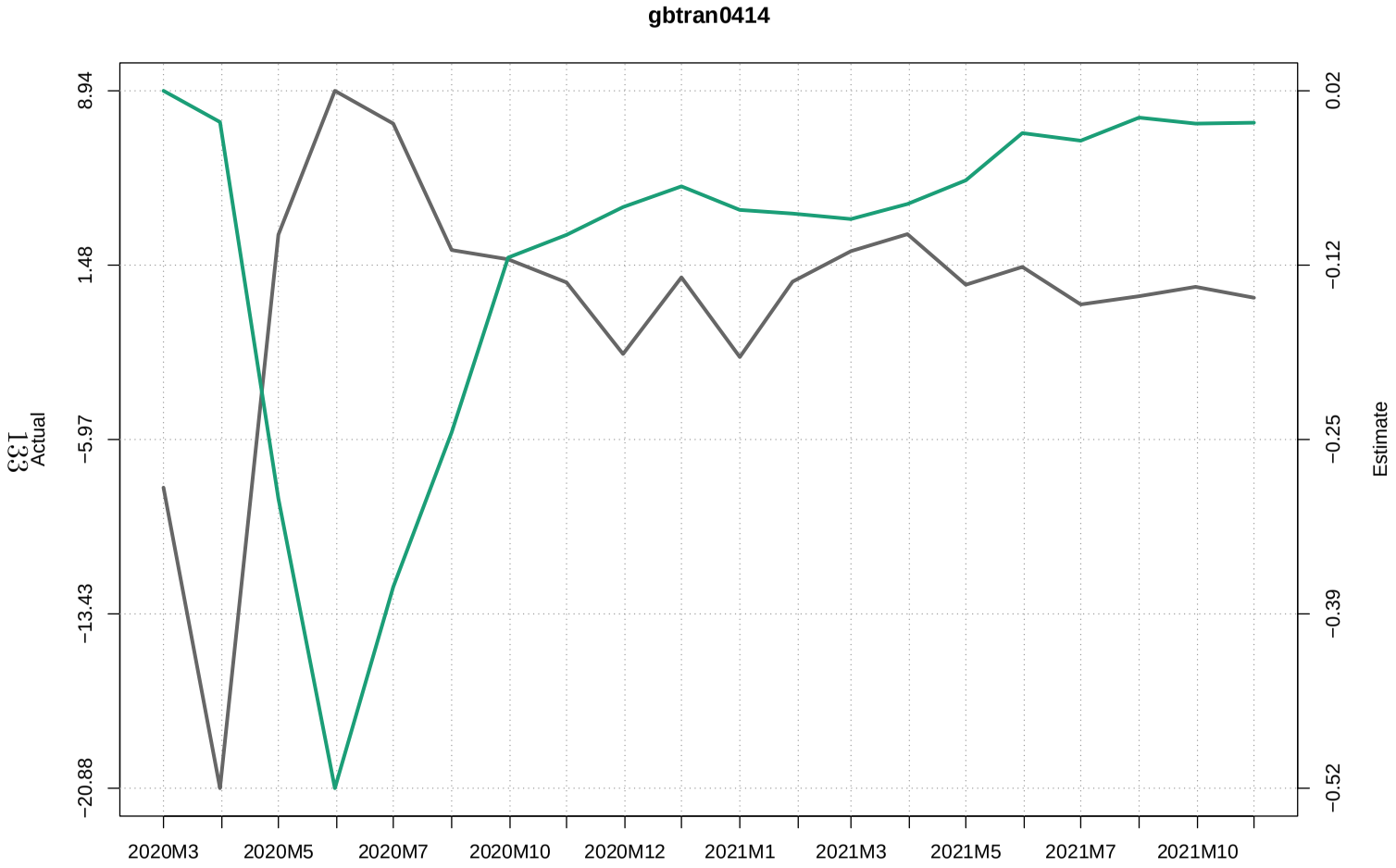
gbtran0360
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.53
−0.39
−0.25
−0.12
0.02
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

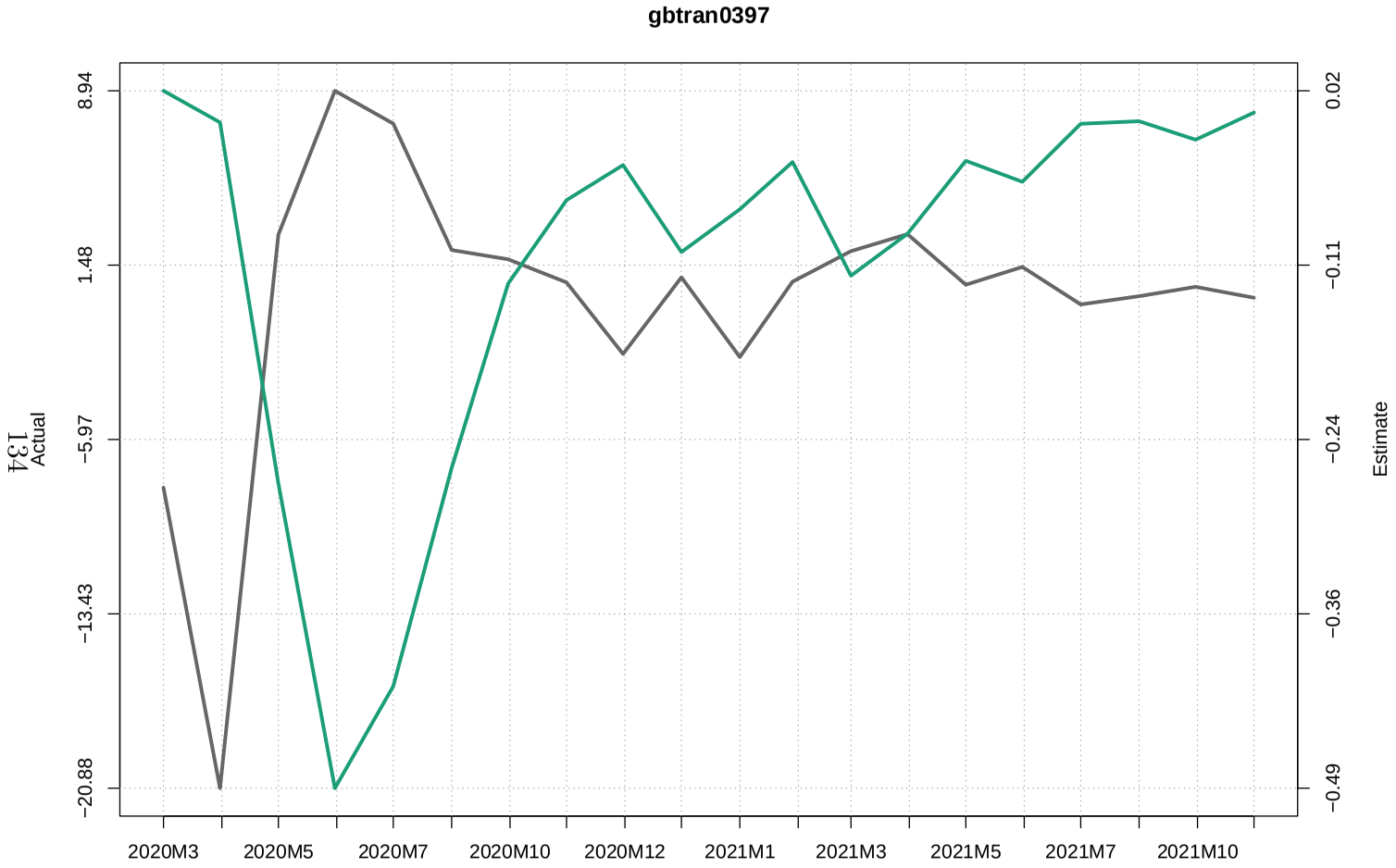
gbtran0380
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.24
−0.38
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

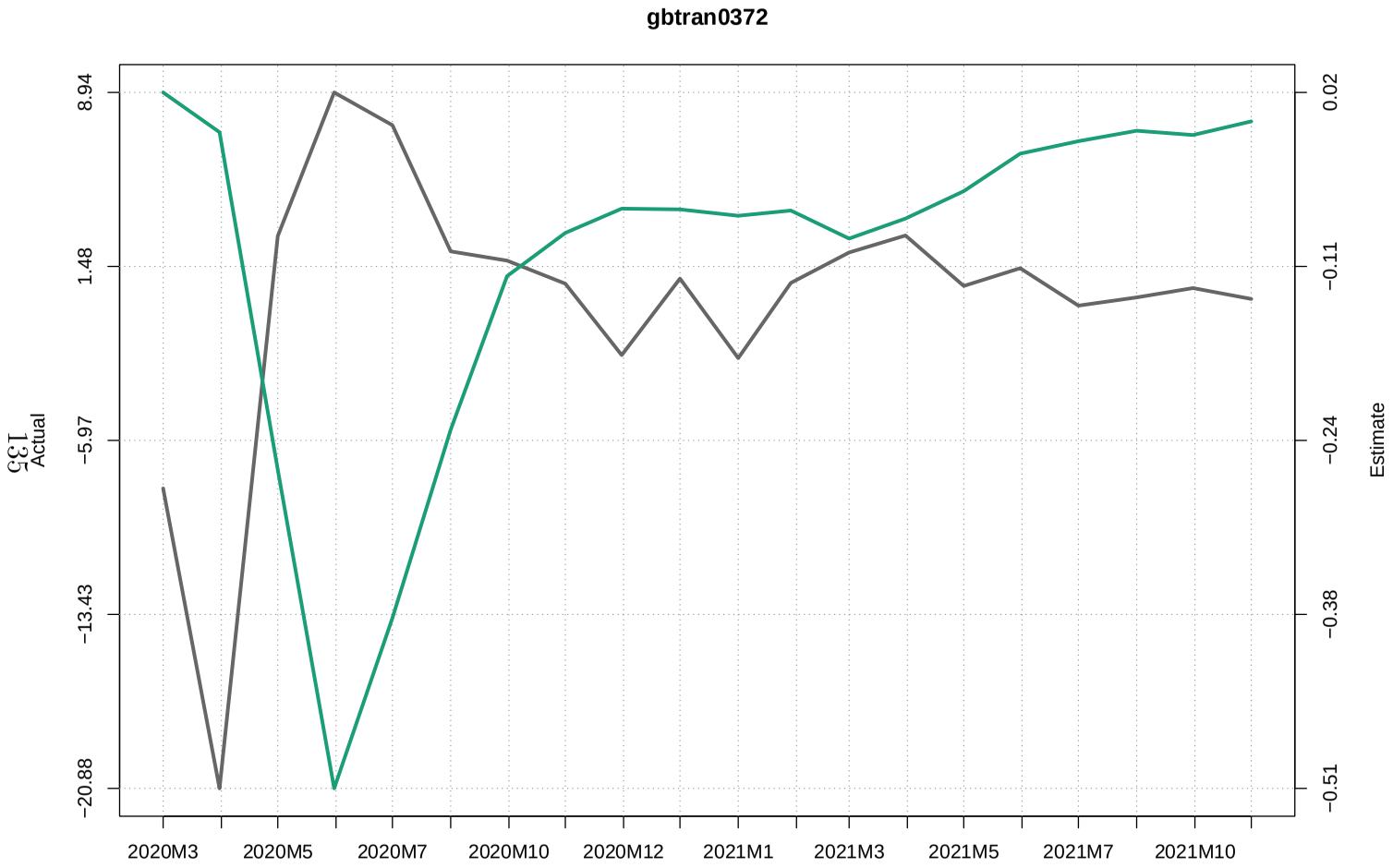

gbtran0410
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.11
−0.24
−0.36
−0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0393
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.03
−0.1
−0.24
−0.37
−0.5
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

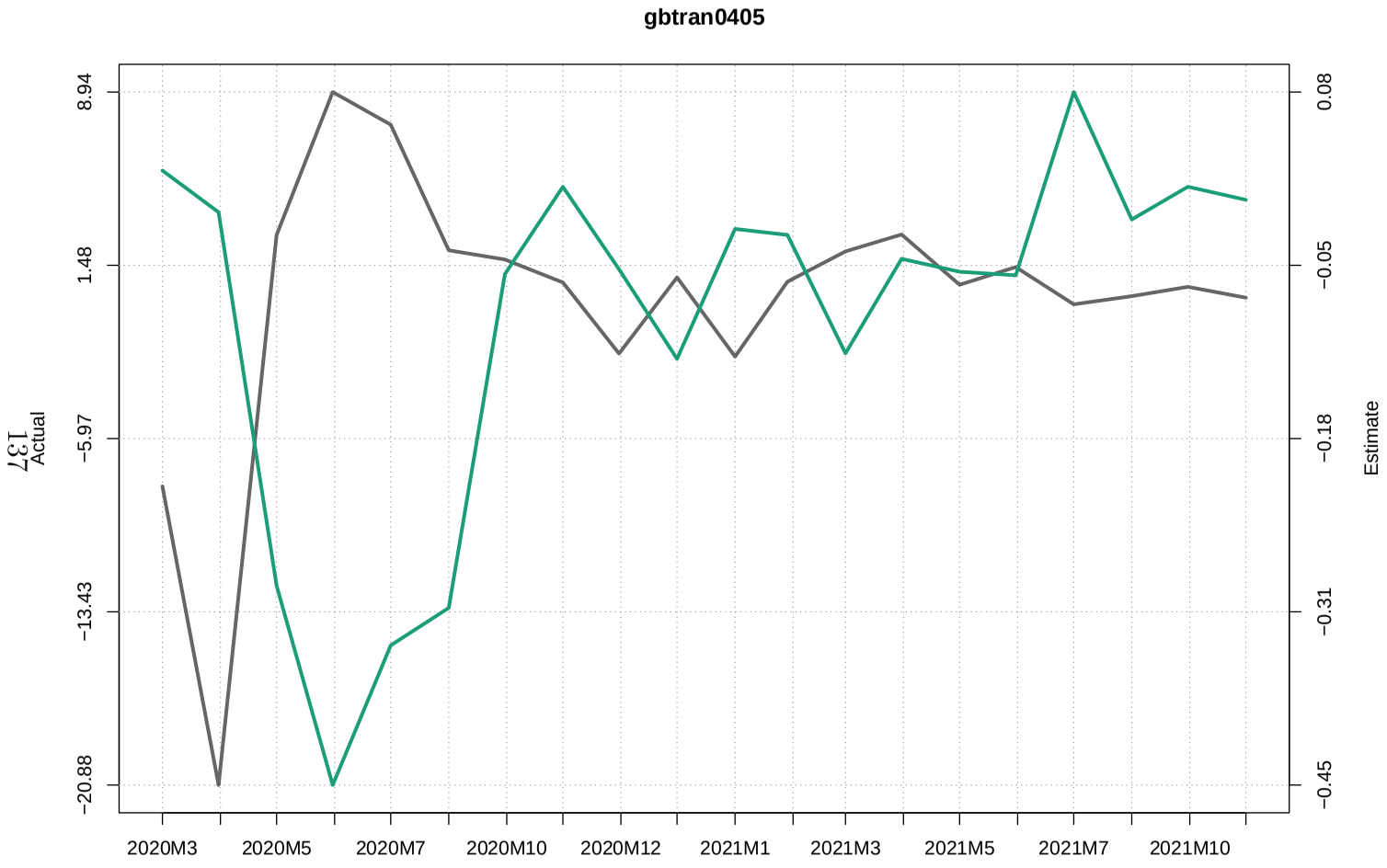

gbtran0364
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.12
−0.25
−0.39
−0.53
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0414
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.12
−0.25
−0.39
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0397
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.24
−0.36
−0.49
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


gbtran0372
Actual
135
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.24
−0.38
−0.51
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

gbtran0420
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.12
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

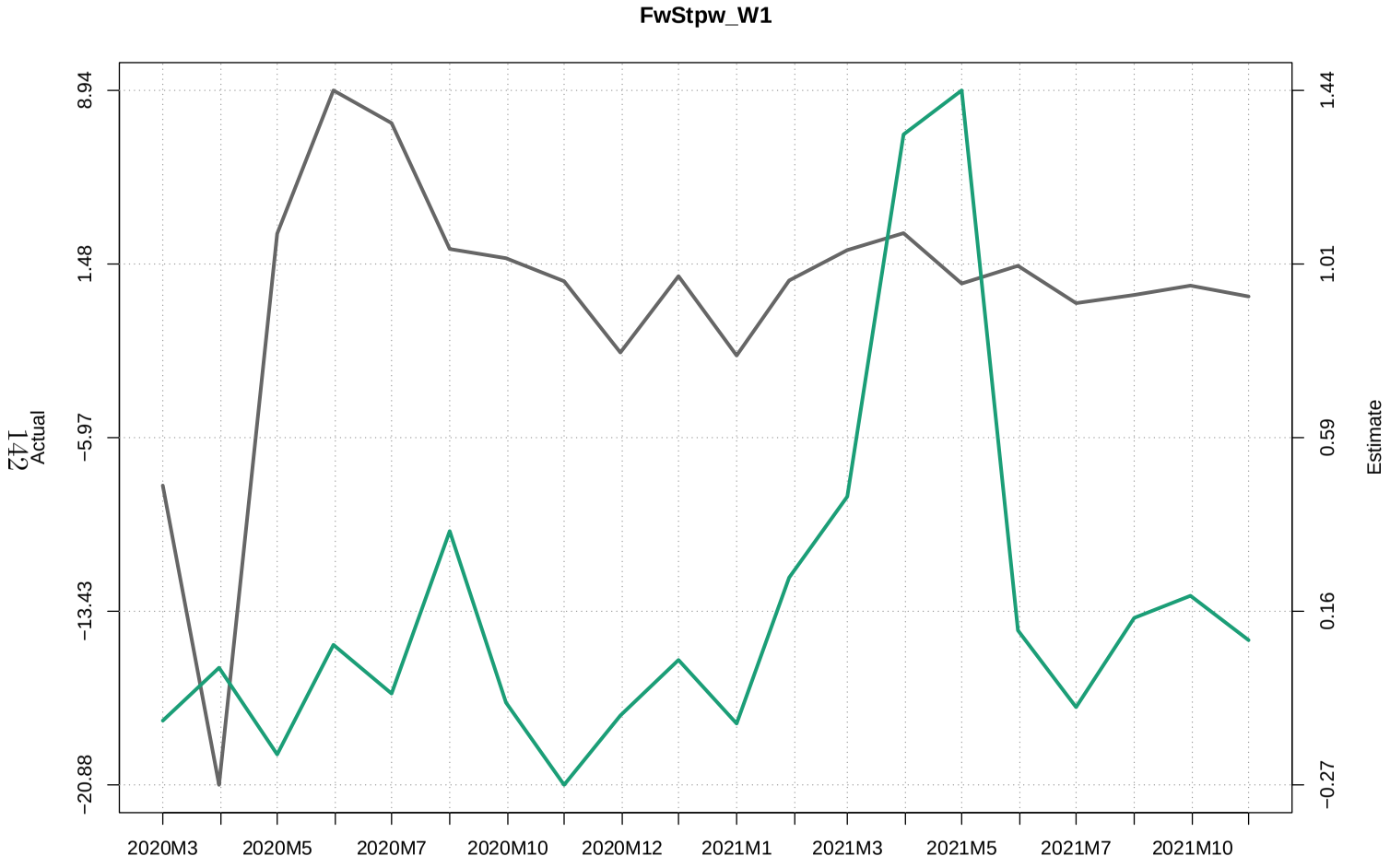

gbtran0405
137
Actual
8.94
1.48
−5.97
−13.43
−20.88
0.08
−0.05
−0.18
−0.31
−0.45
Estimate
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

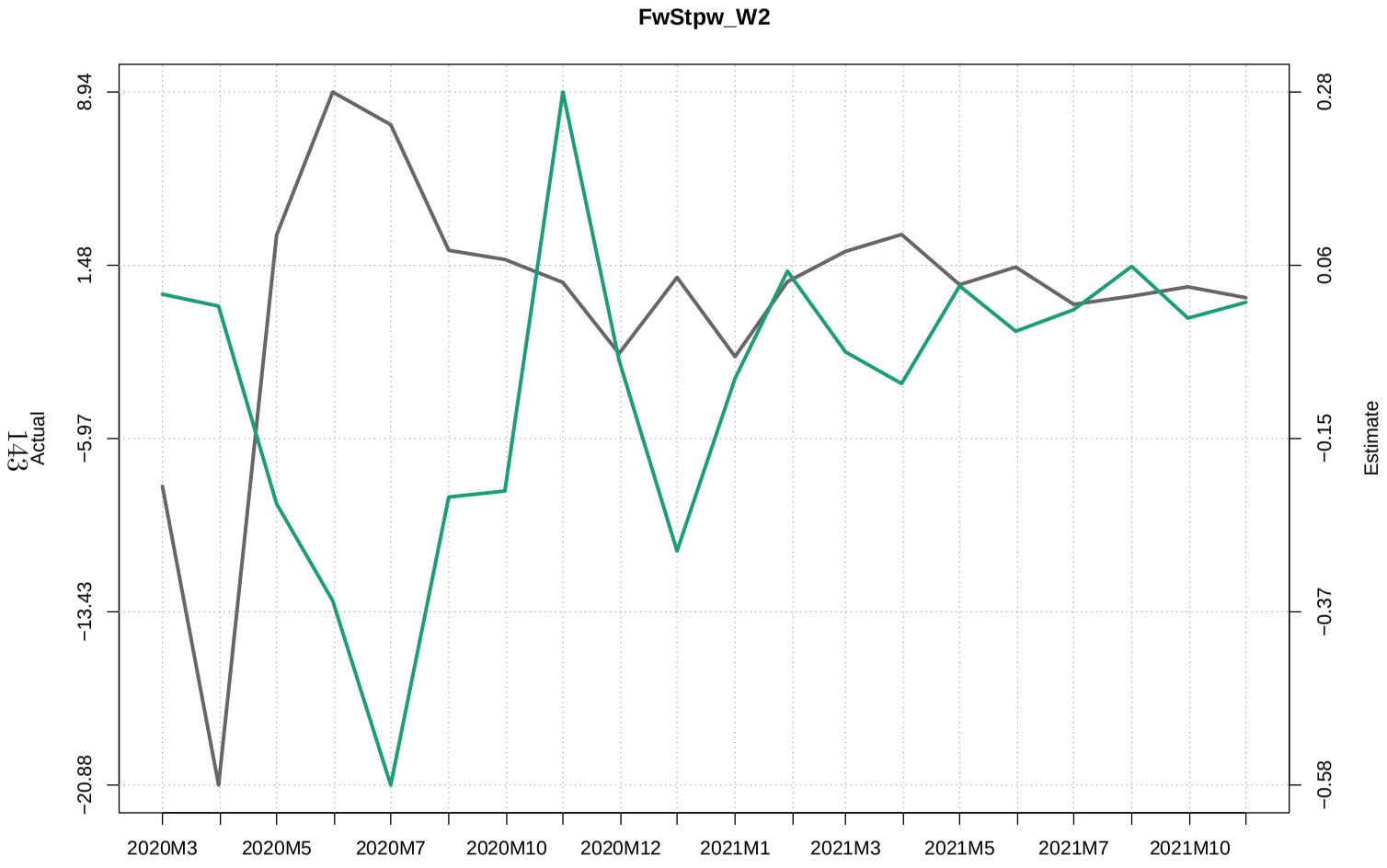
Job
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.27
−0.11
0.05
0.21
0.37

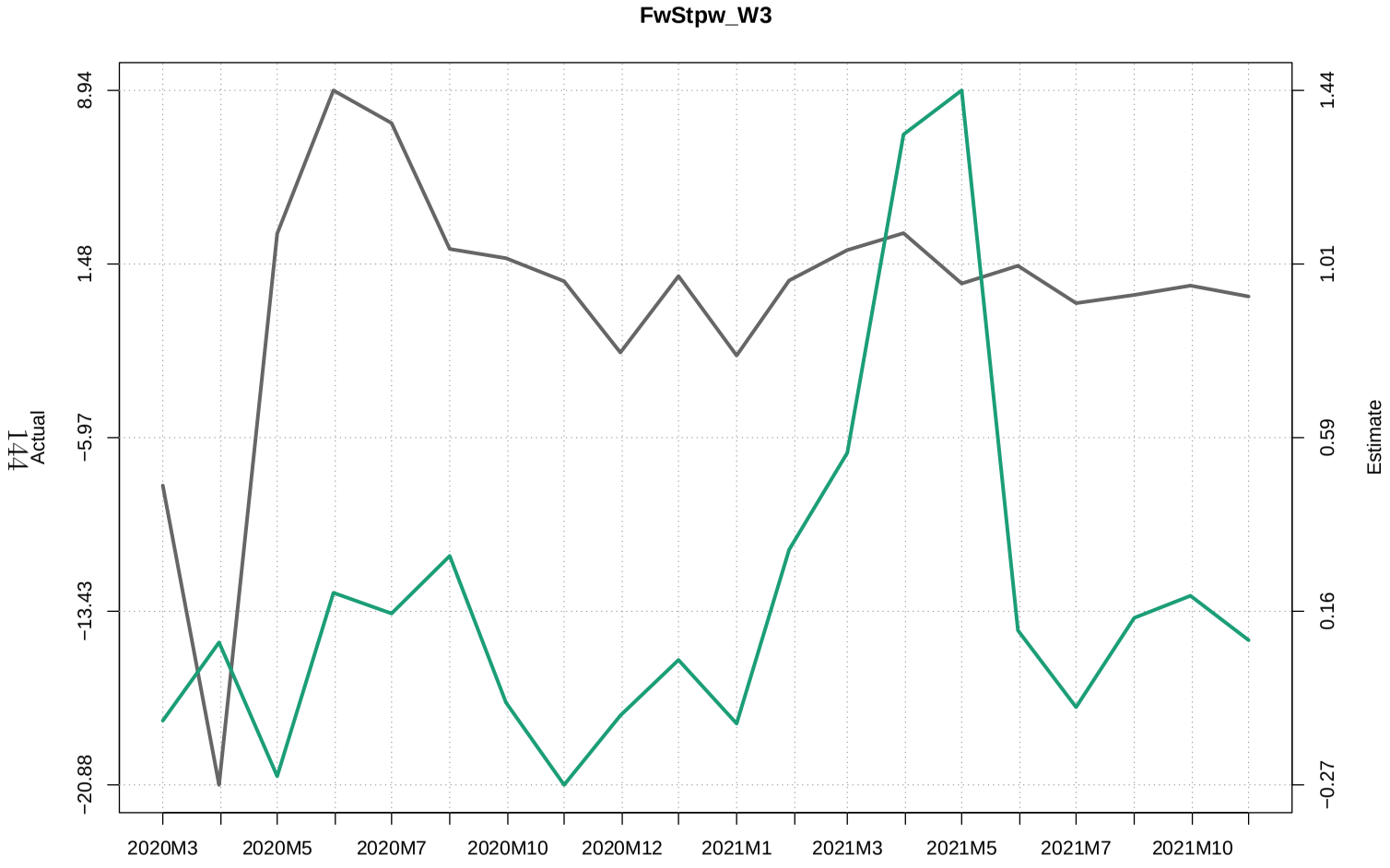

Traffic1
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.08
−0.04
−0.16
−0.28
−0.39
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Traffic2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.1
−0.22
−0.33
−0.45
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

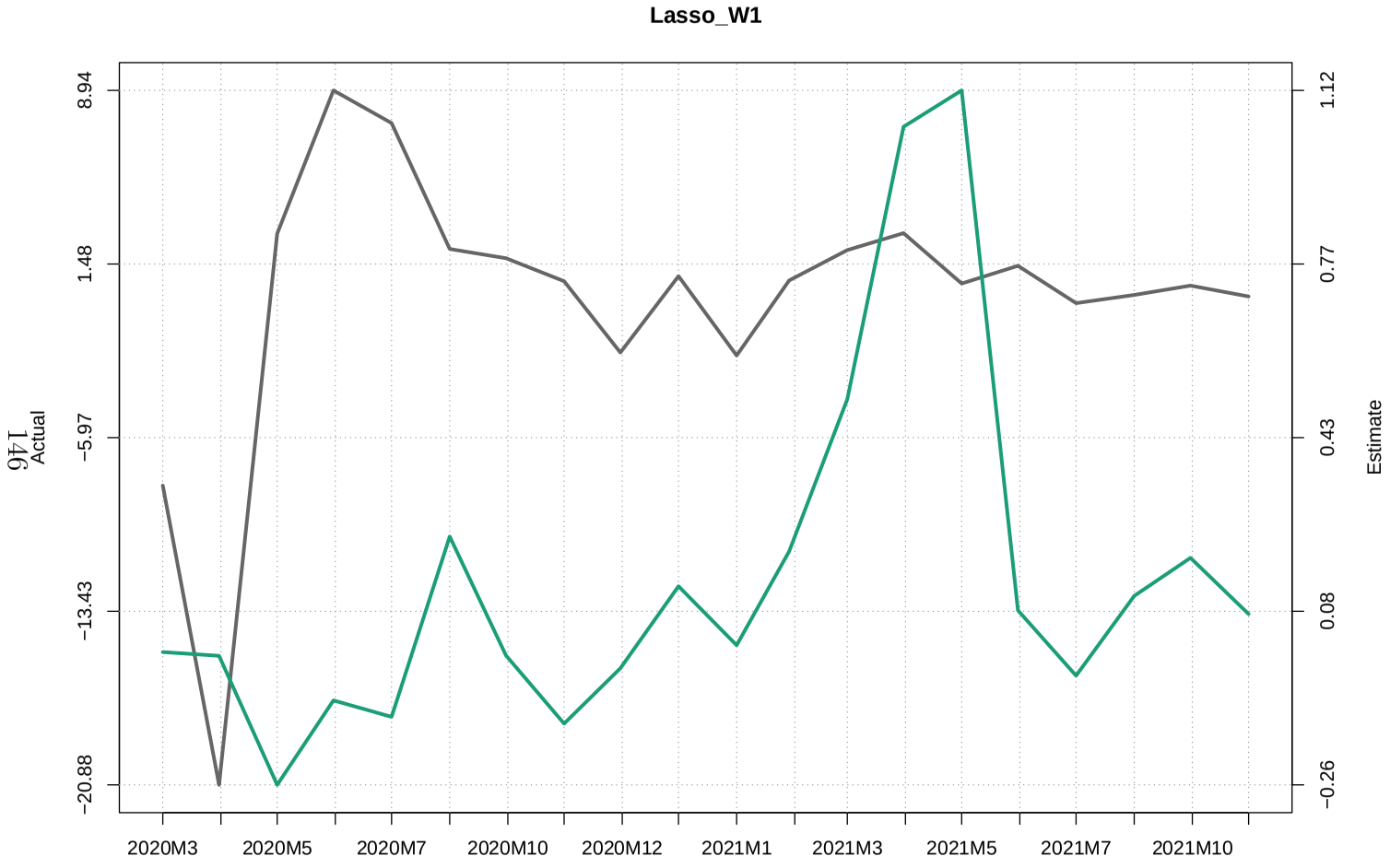

All3
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.39
0.22
0.06
−0.11
−0.28
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


FwStpw_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.44
1.01
0.59
0.16
−0.27
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


FwStpw_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.28
0.06
−0.15
−0.37
−0.58
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


FwStpw_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.44
1.01
0.59
0.16
−0.27
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Ridge_W1
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.76
0.51
0.27
0.02
−0.23
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

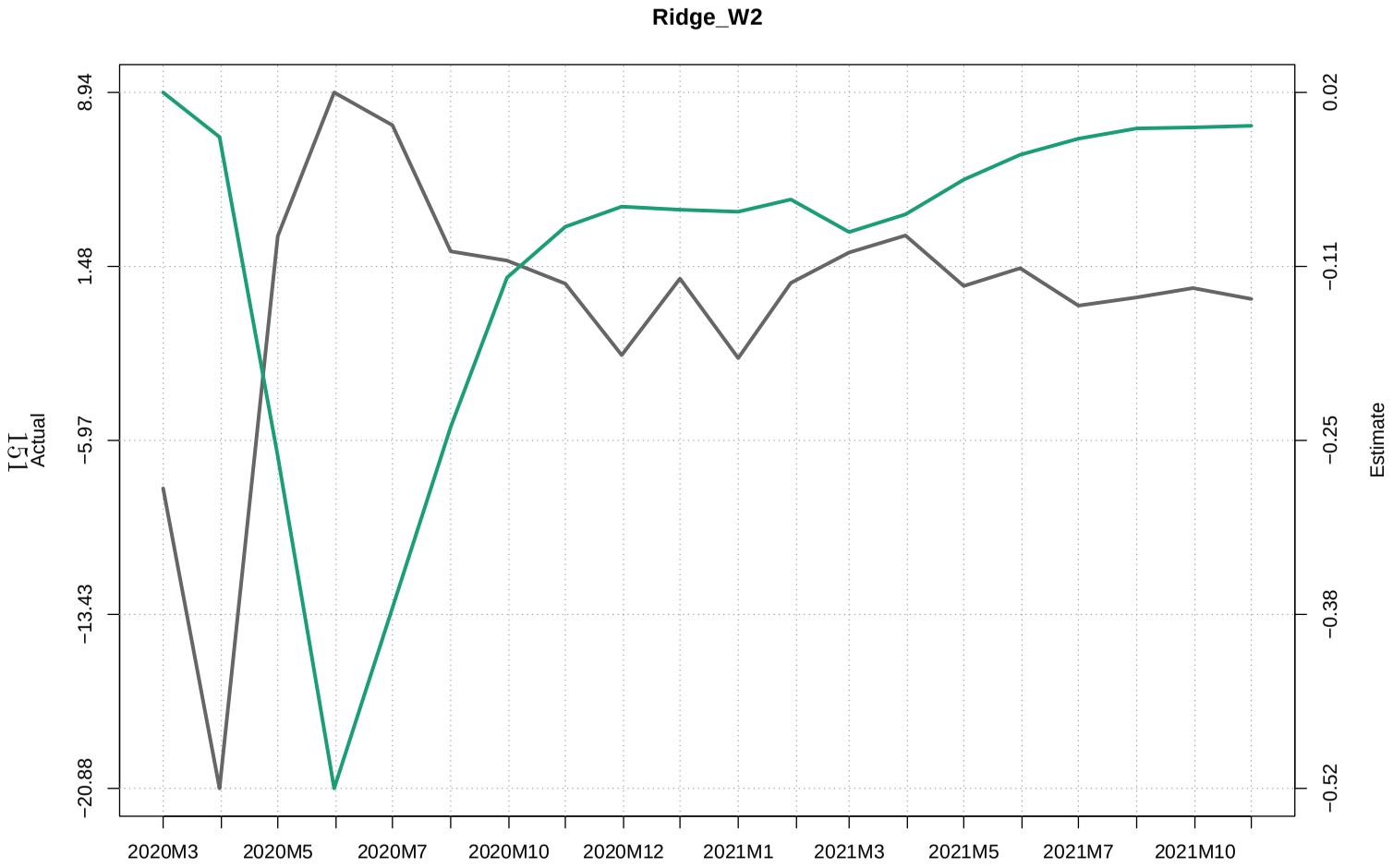

Lasso_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.12
0.77
0.43
0.08
−0.26
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

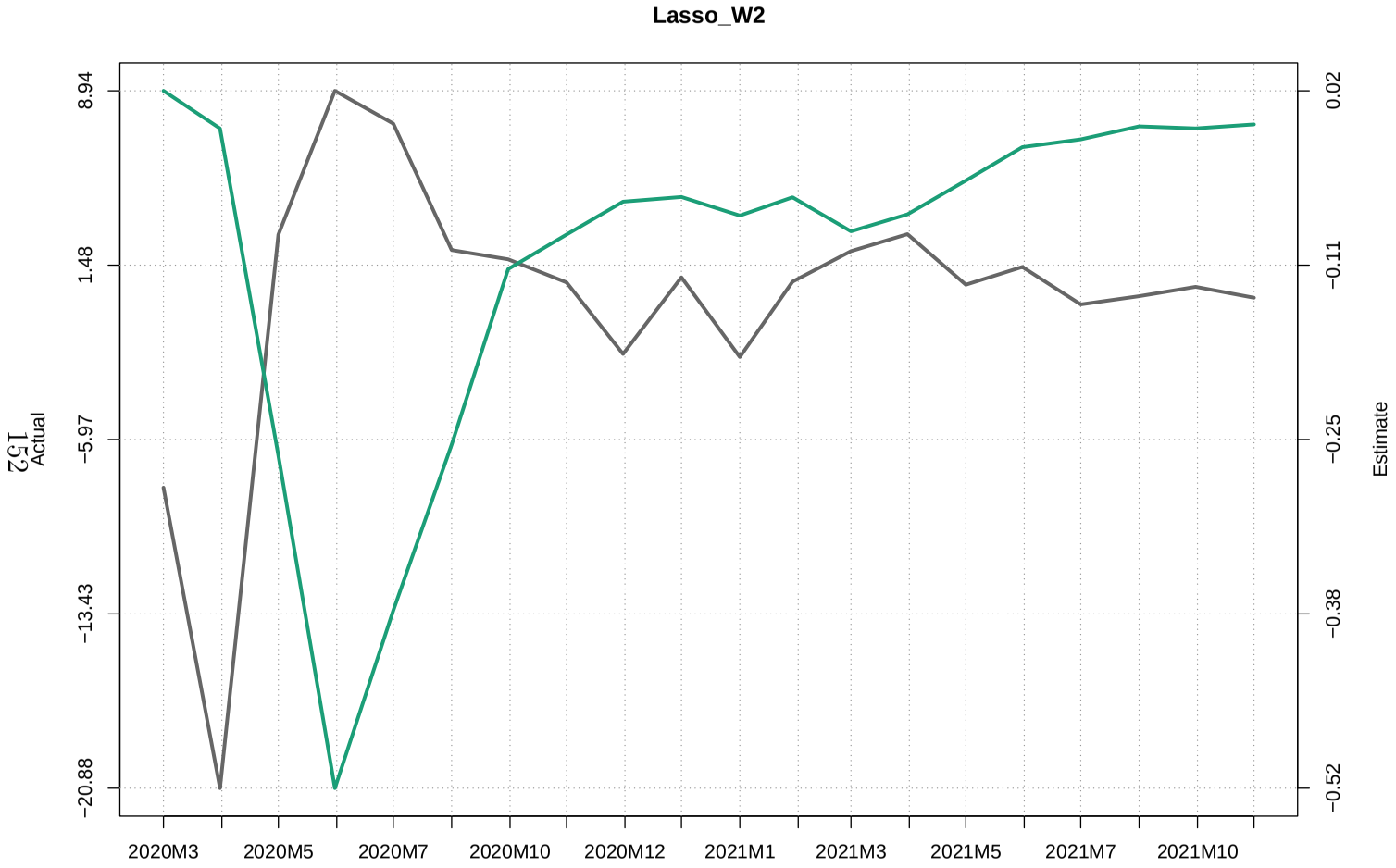
EN_W1
Actual
−20.88
−13.43
−5.97
1.48
8.94
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10
Estimate
−0.26
0.07
0.4
0.72
1.05

AdLasso1_W1
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
1.26
0.89
0.51
0.13
−0.24
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


AdLasso2_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.41
0.99
0.57
0.15
−0.28
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

RandF_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.45
0.31
0.17
0.02
−0.12
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Ridge_W2
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


Lasso_W2
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

EN_W2
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.02
−0.11
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

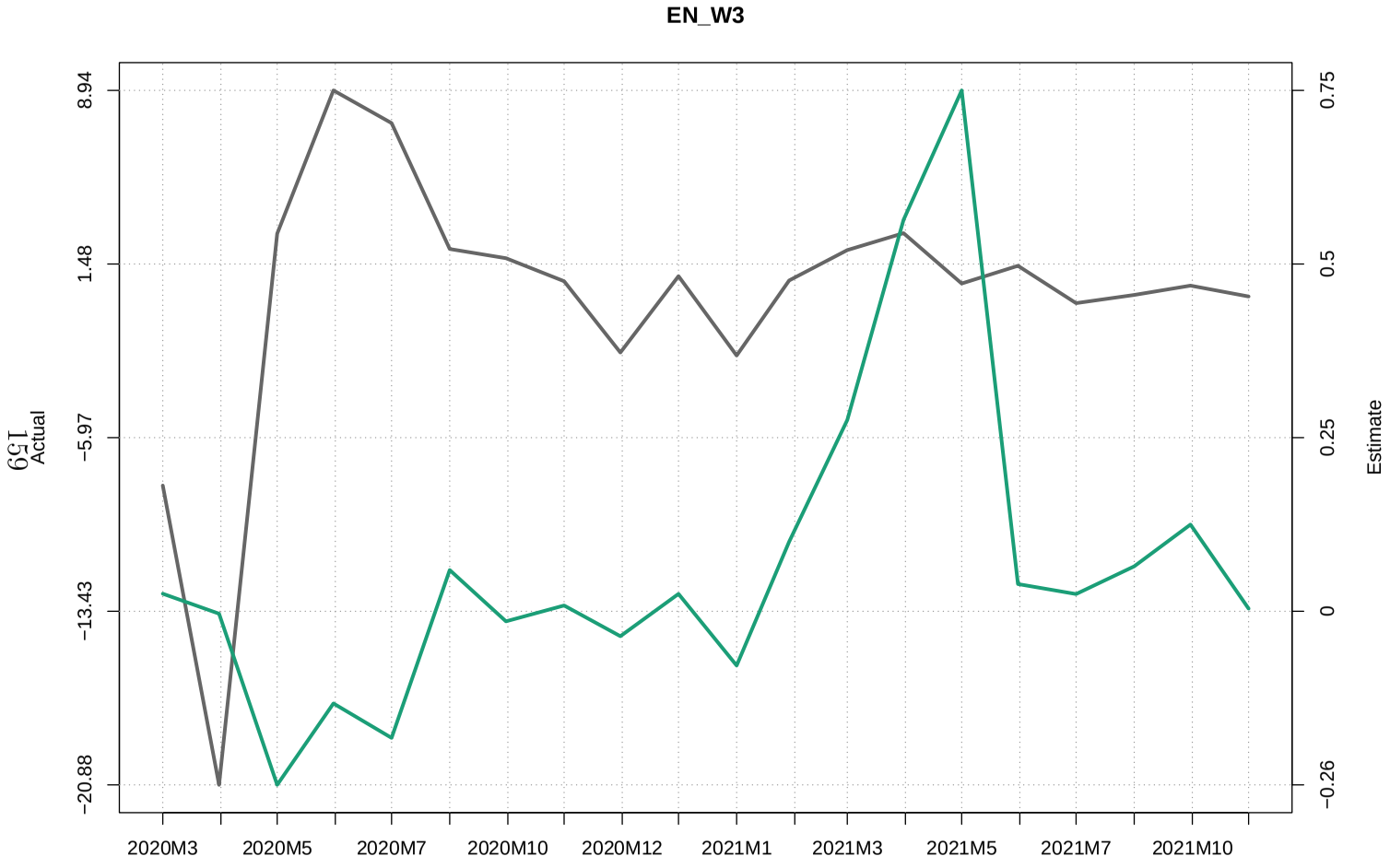

AdLasso1_W2
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.03
−0.09
−0.21
−0.33
−0.45
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


AdLasso2_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.11
−0.25
−0.38
−0.52
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

RandF_W2
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.41
−0.23
−0.05
0.13
0.31
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

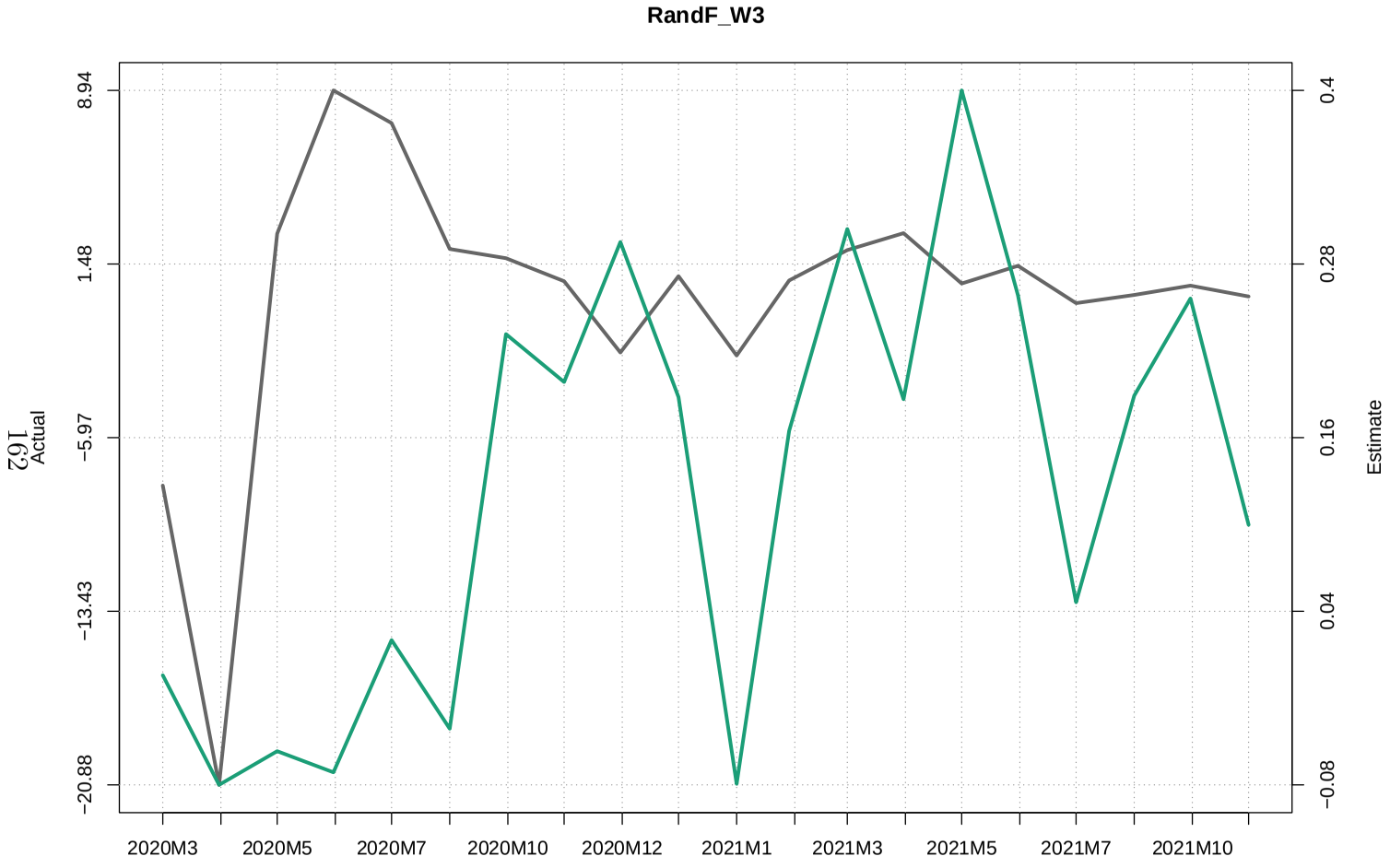
Ridge_W3
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.31
−0.13
0.04
0.21
0.38
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

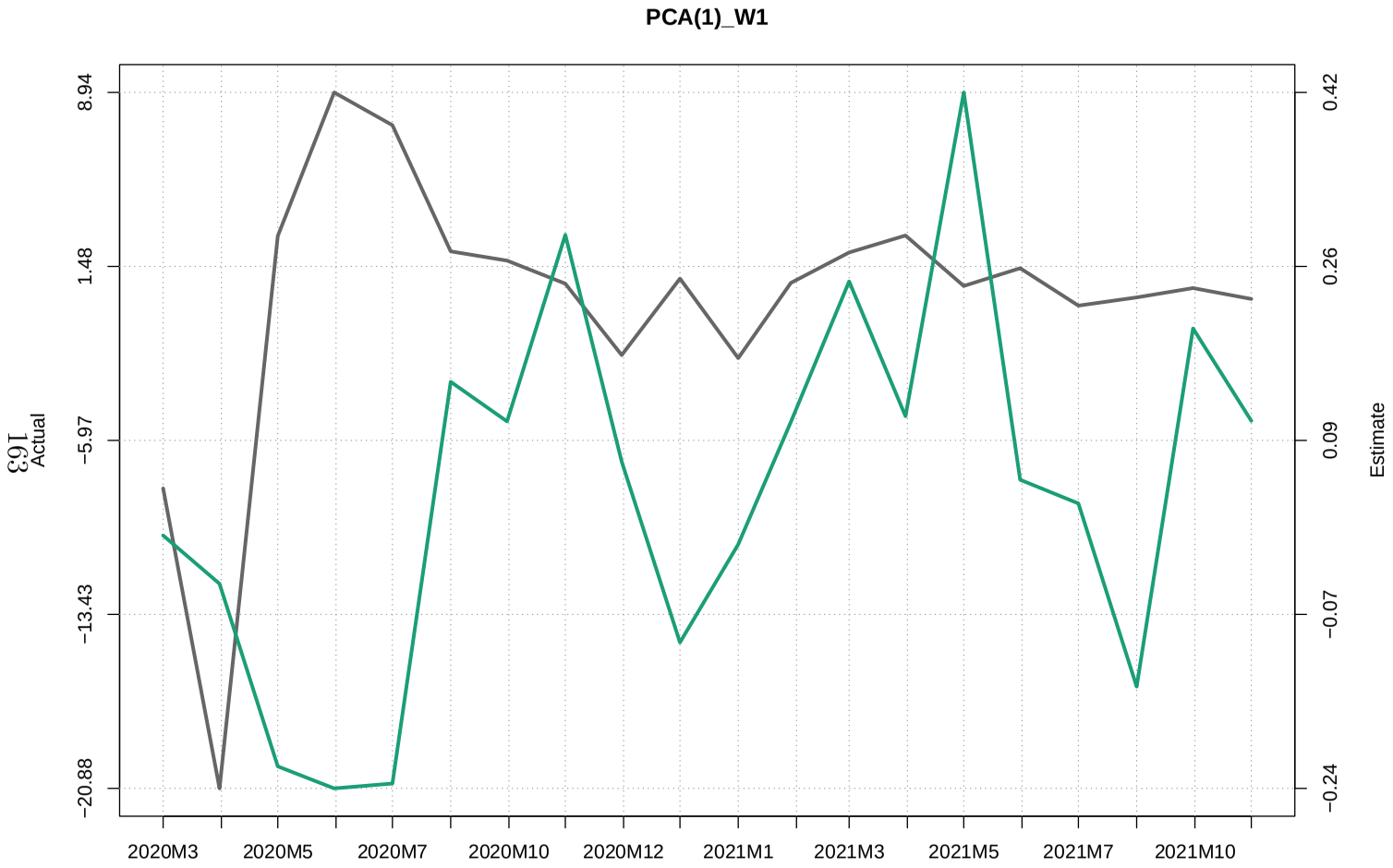
Lasso_W3
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.26
0.01
0.27
0.54
0.8
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

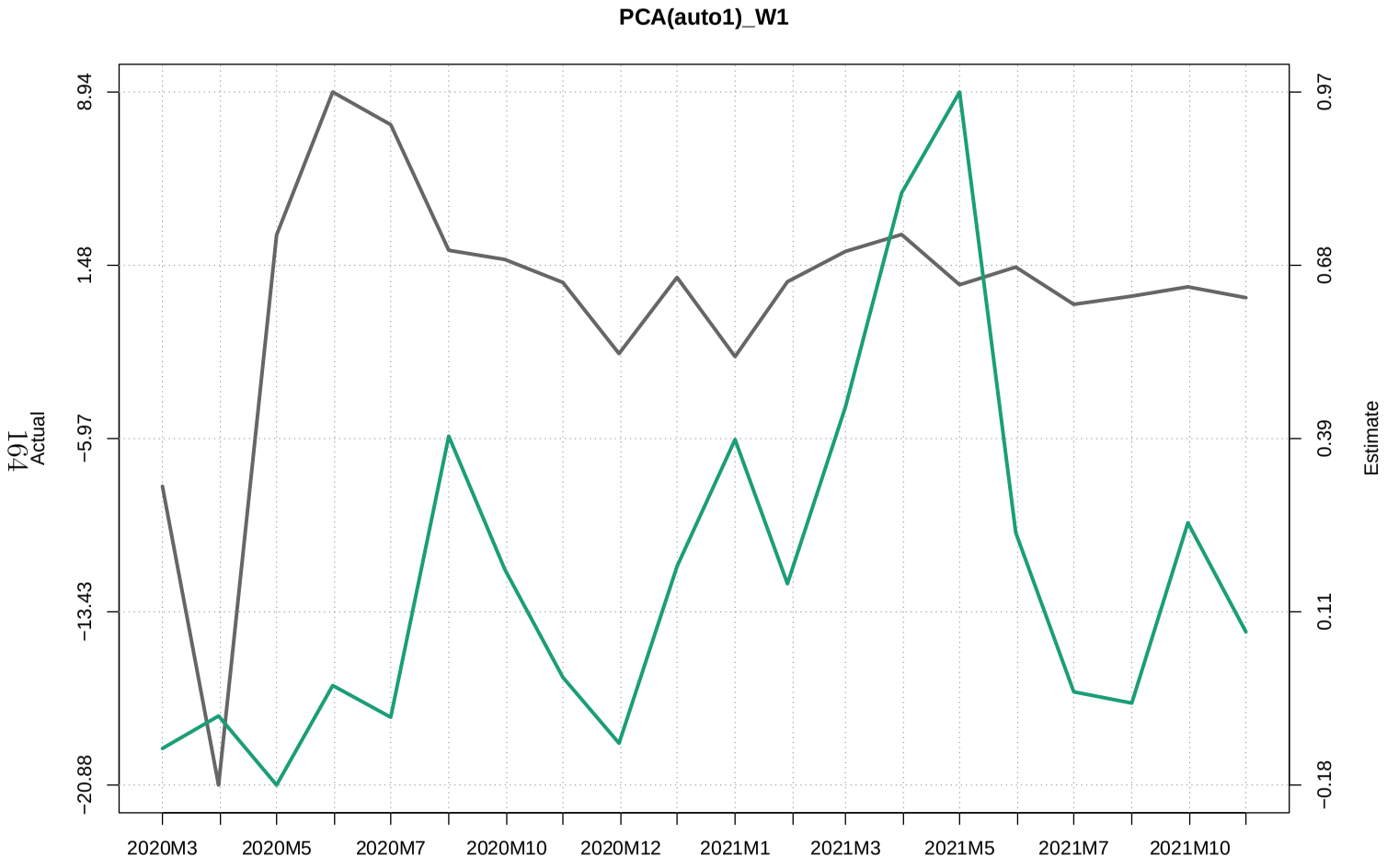

EN_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.75
0.5
0.25
0
−0.26
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

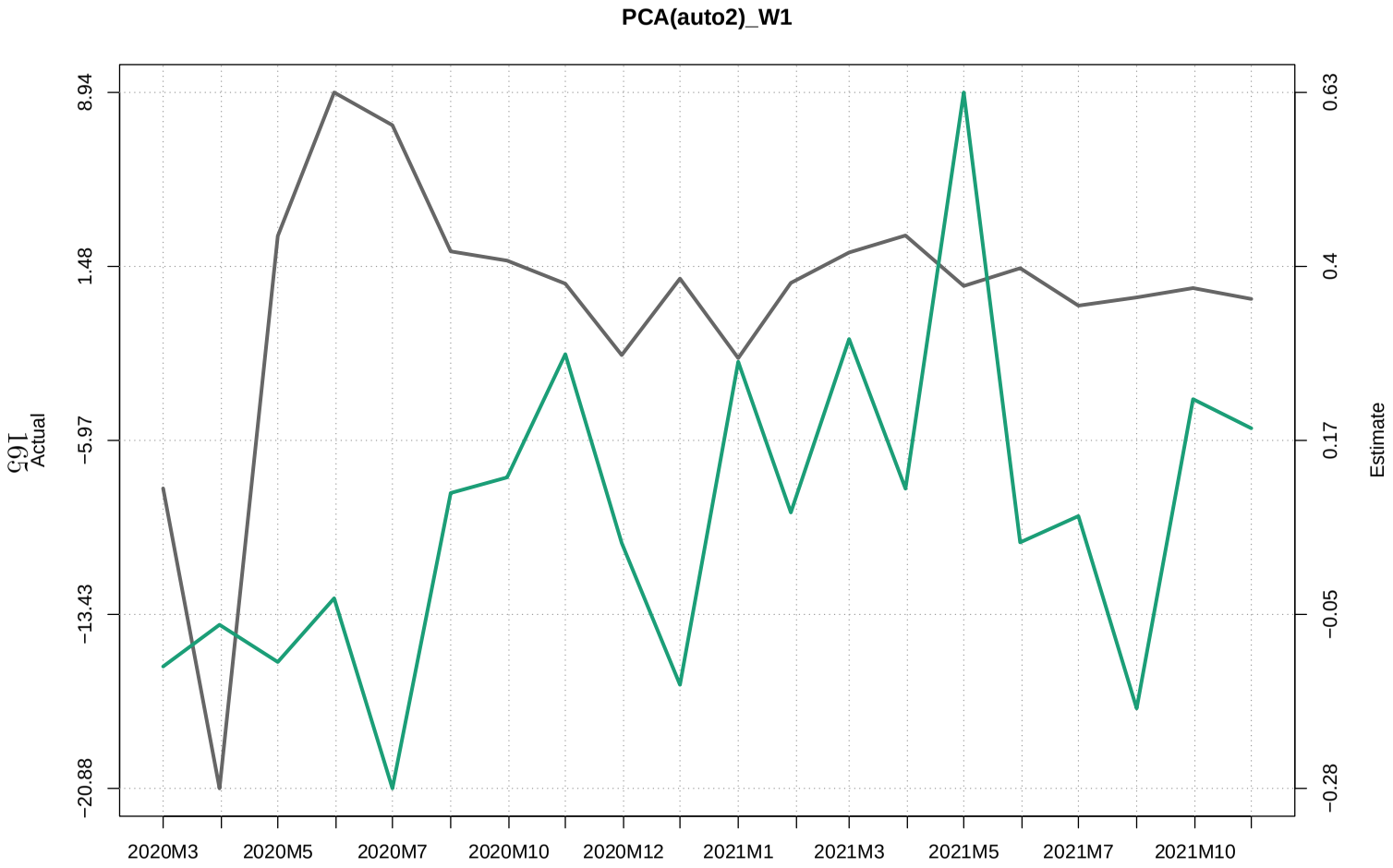
AdLasso1_W3
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
1.19
0.83
0.47
0.11
−0.26
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

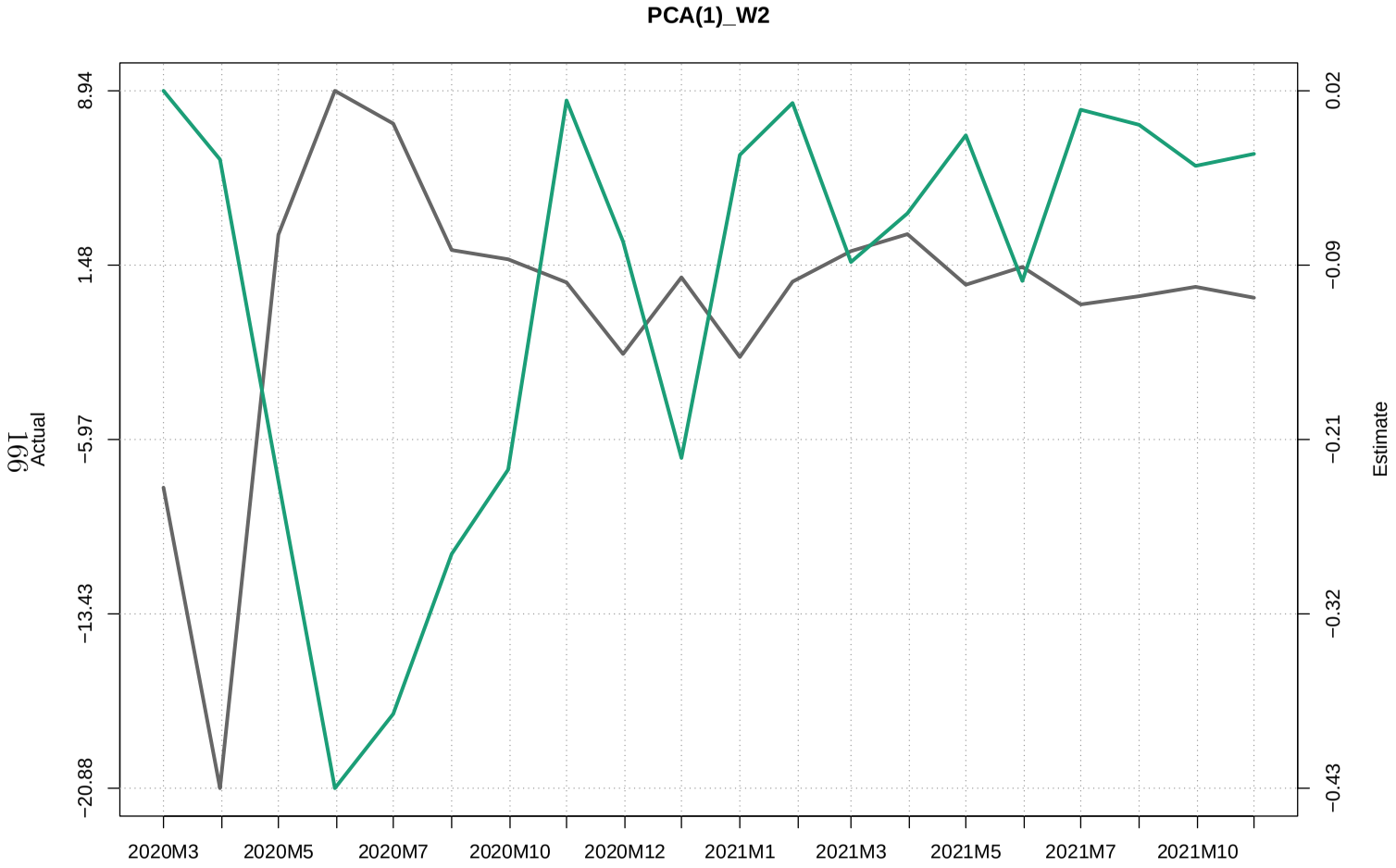

AdLasso2_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.26
0.87
0.49
0.11
−0.28
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

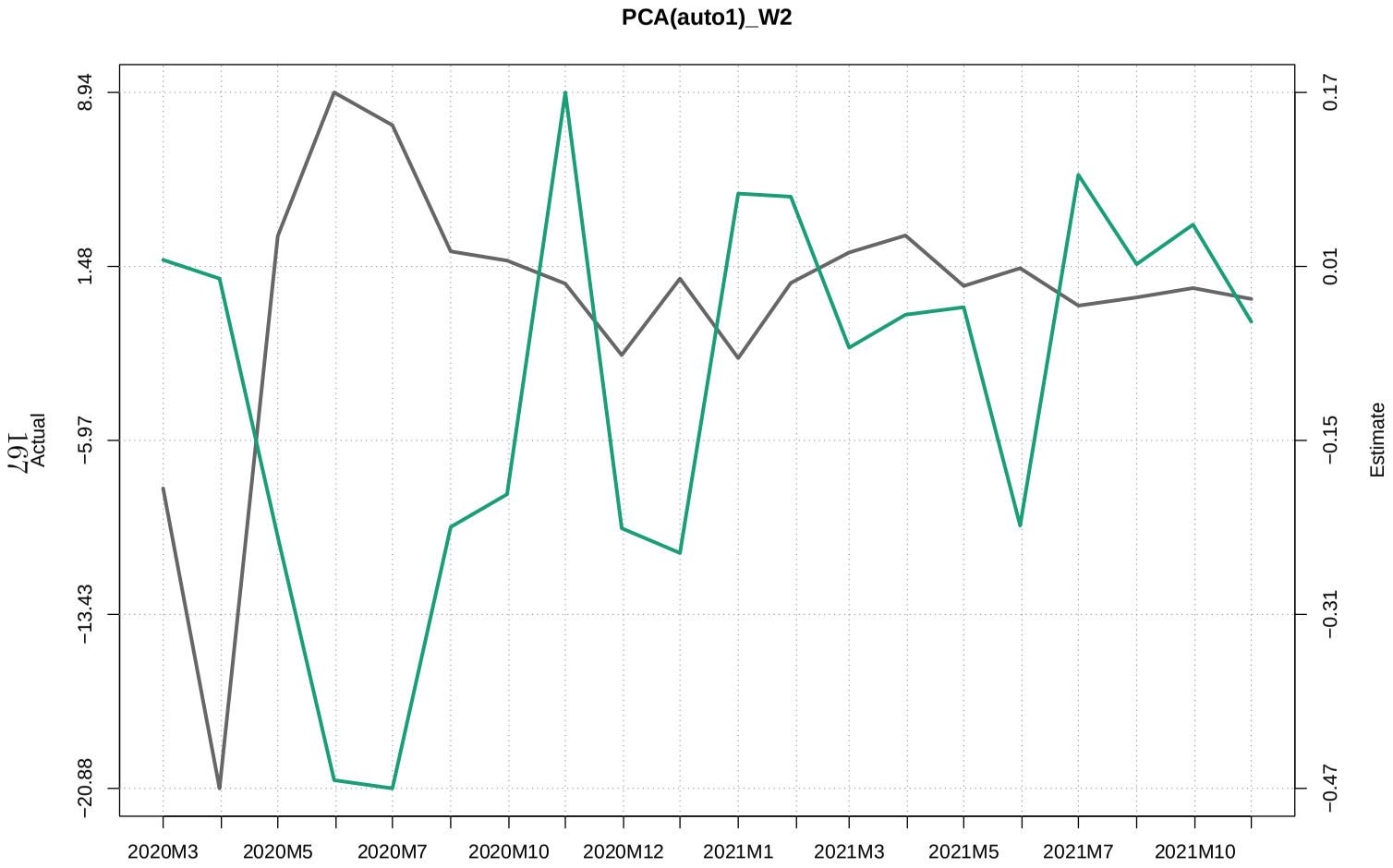

RandF_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.4
0.28
0.16
0.04
−0.08
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


PCA(1)_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.42
0.26
0.09
−0.07
−0.24
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


PCA(auto1)_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.97
0.68
0.39
0.11
−0.18
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

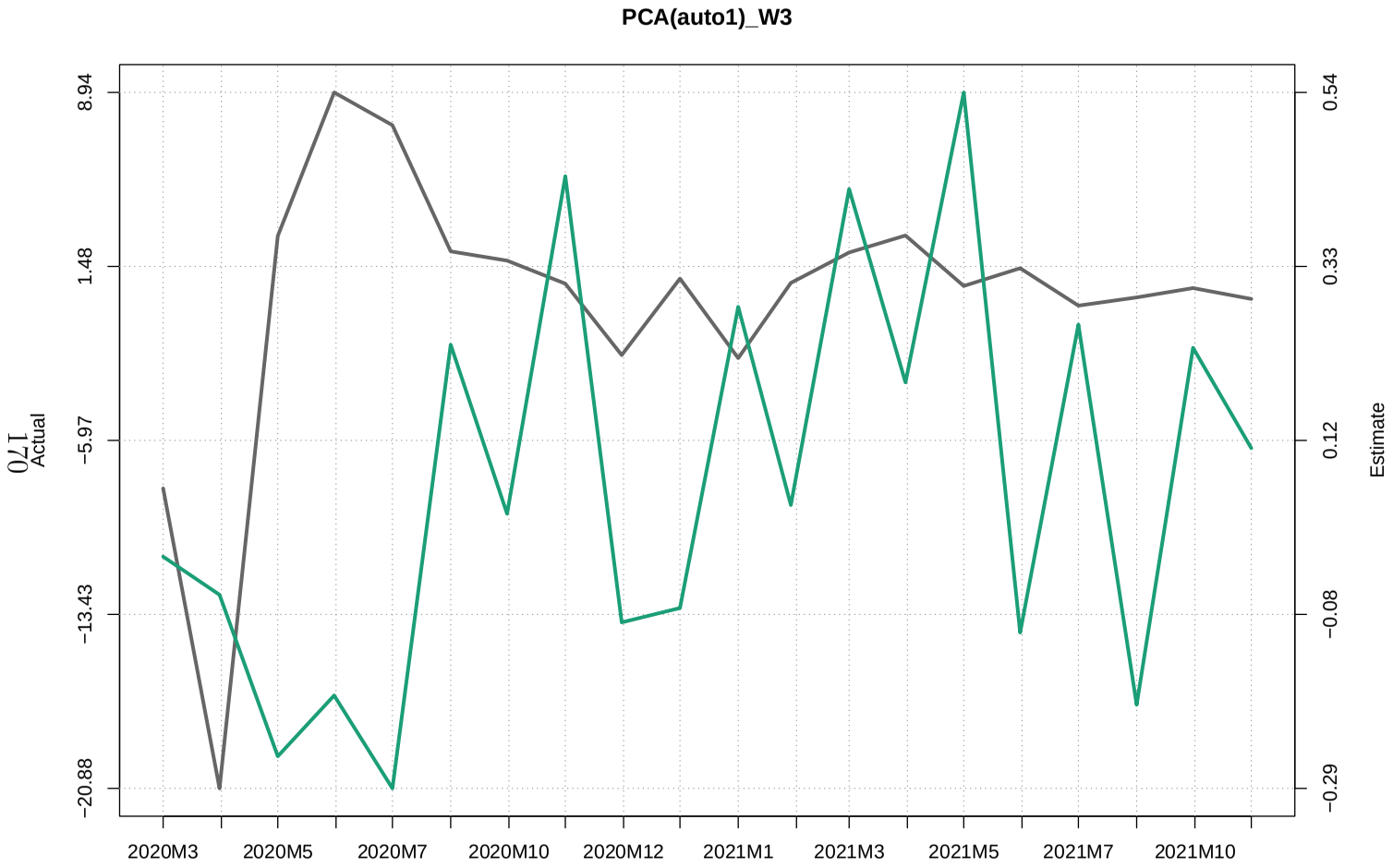

PCA(auto2)_W1
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.63
0.4
0.17
−0.05
−0.28
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

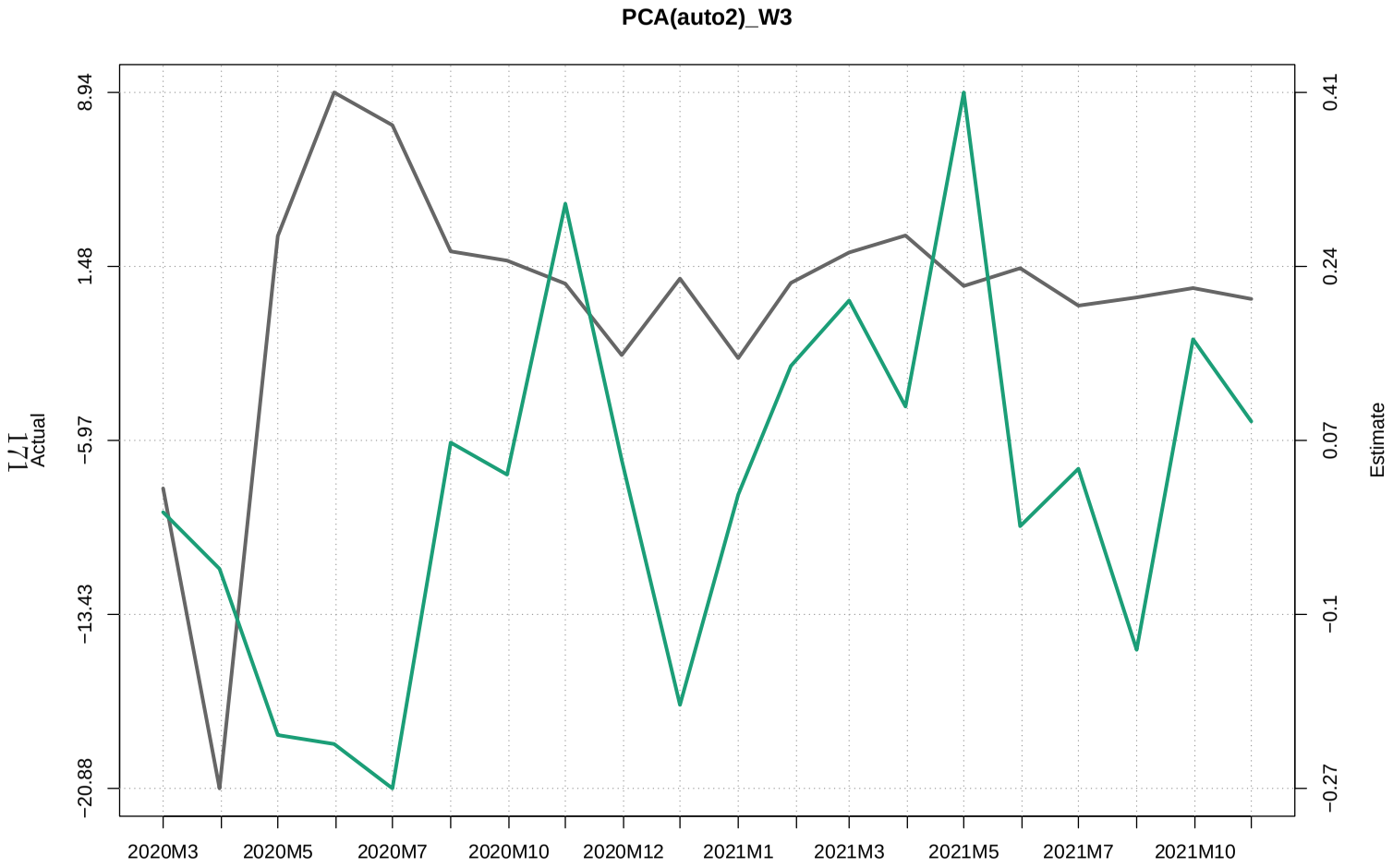

PCA(1)_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.02
−0.09
−0.21
−0.32
−0.43
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

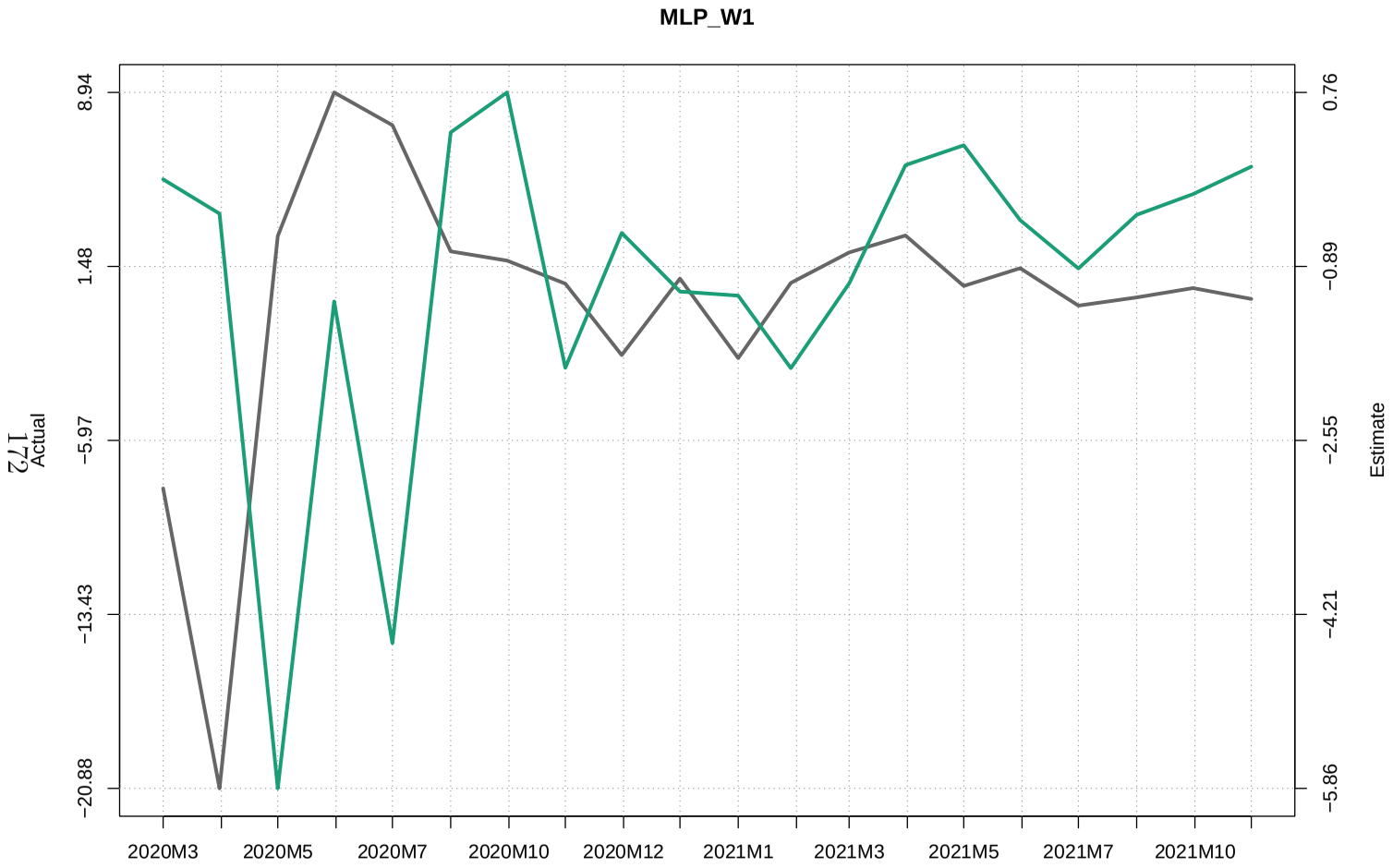

PCA(auto1)_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.17
0.01
−0.15
−0.31
−0.47
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

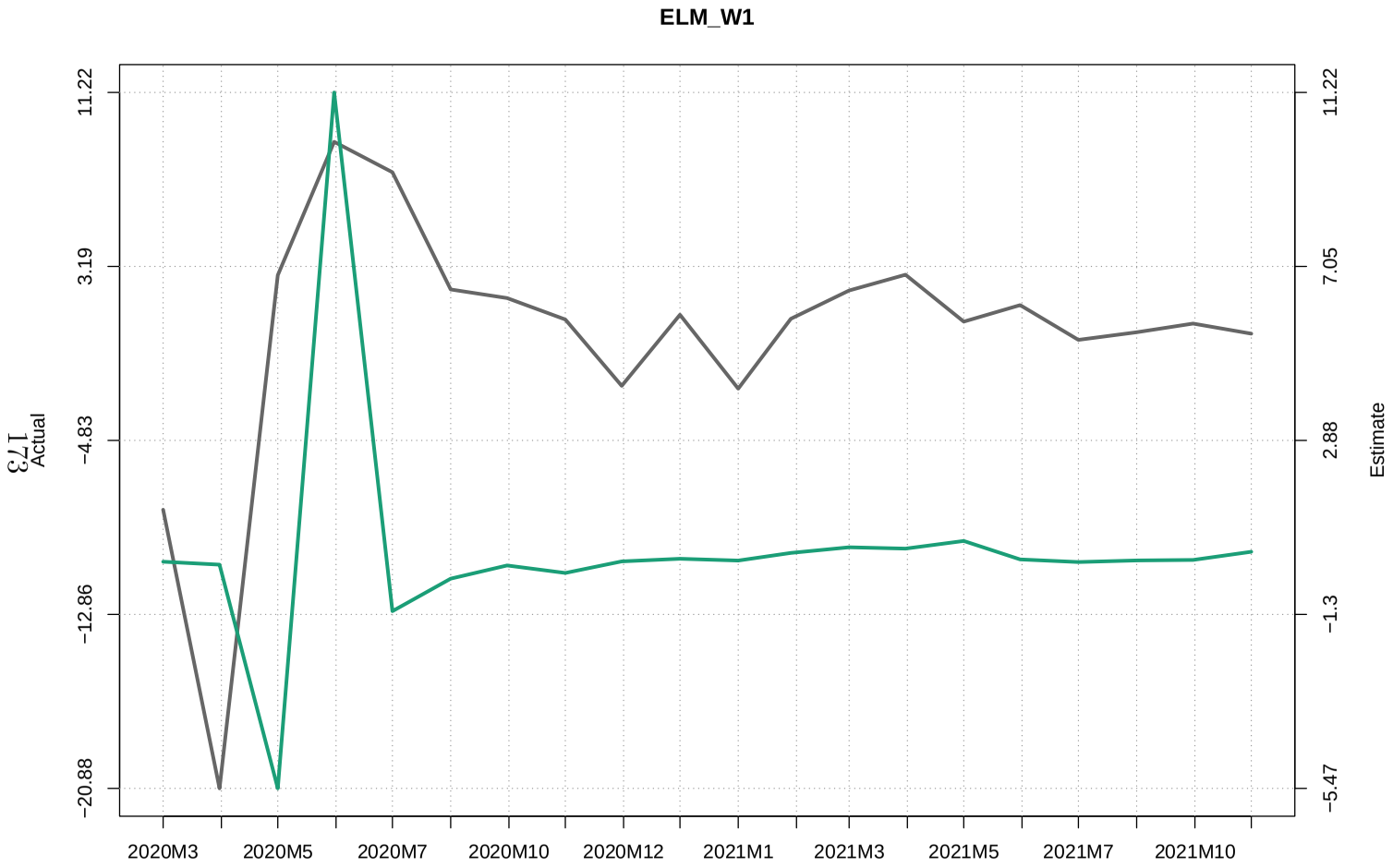

PCA(auto2)_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.16
0
−0.15
−0.3
−0.46
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

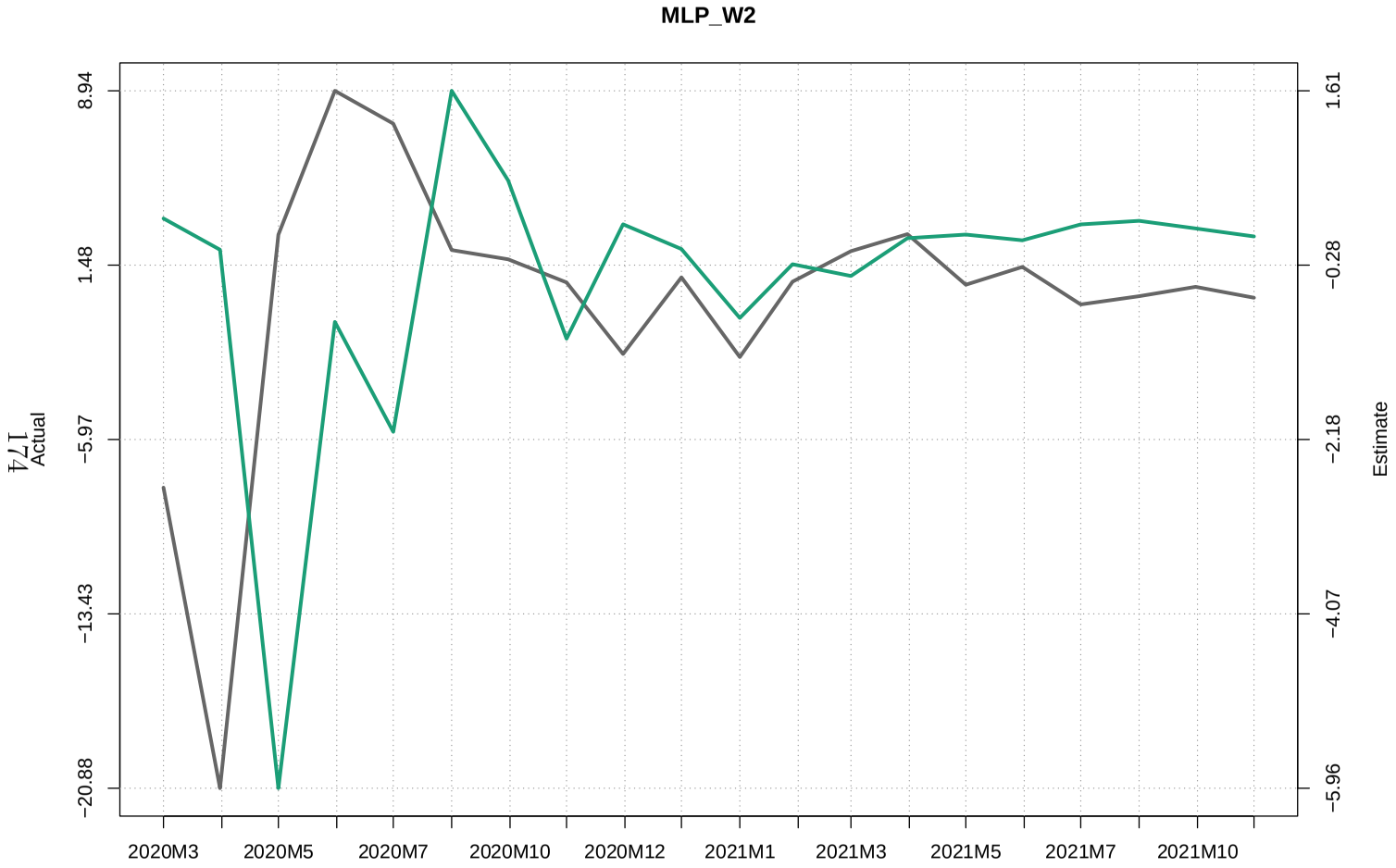
PCA(1)_W3
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.27
−0.1
0.07
0.24
0.41
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

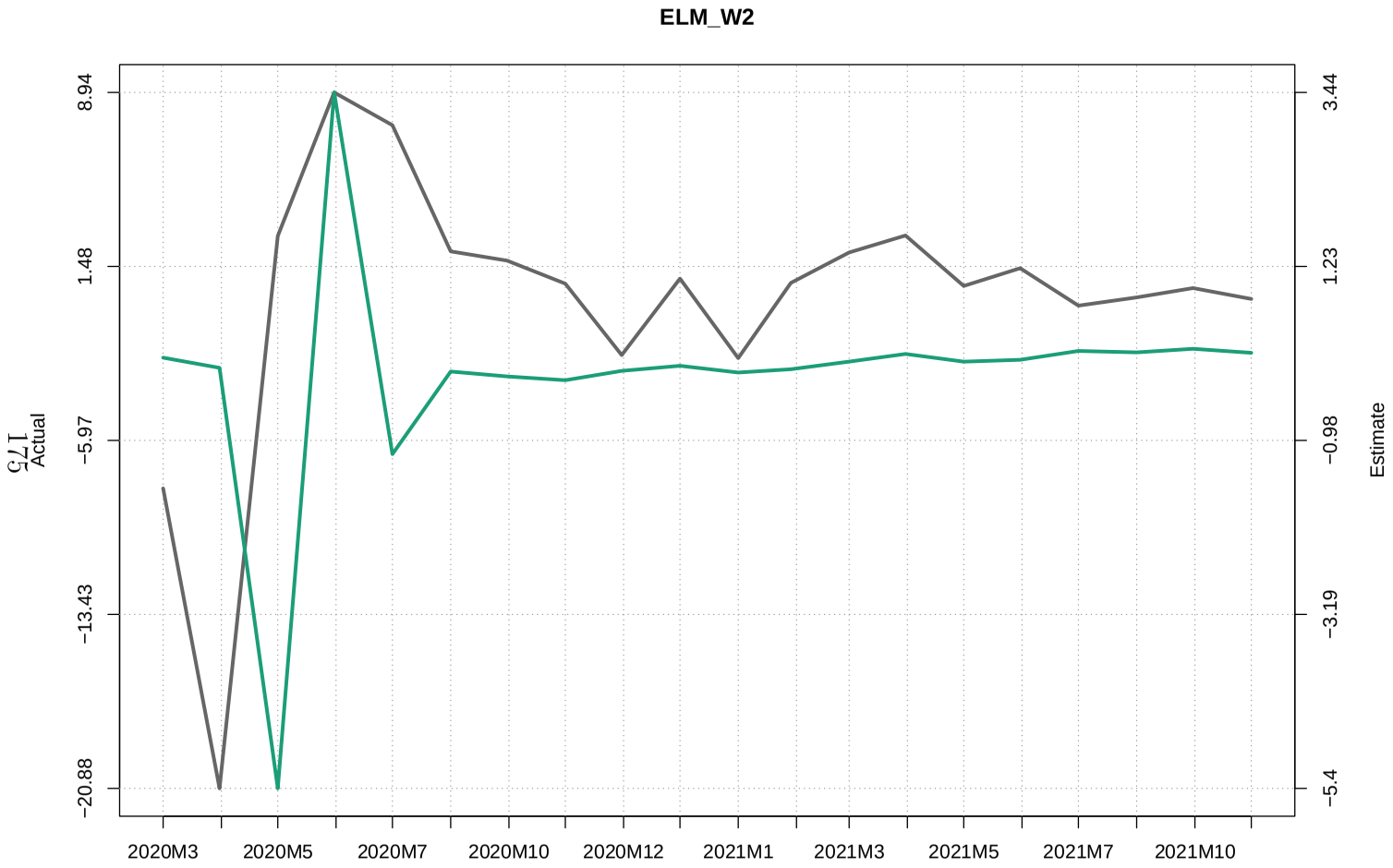

PCA(auto1)_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.54
0.33
0.12
−0.08
−0.29
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


PCA(auto2)_W3
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.41
0.24
0.07
−0.1
−0.27
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

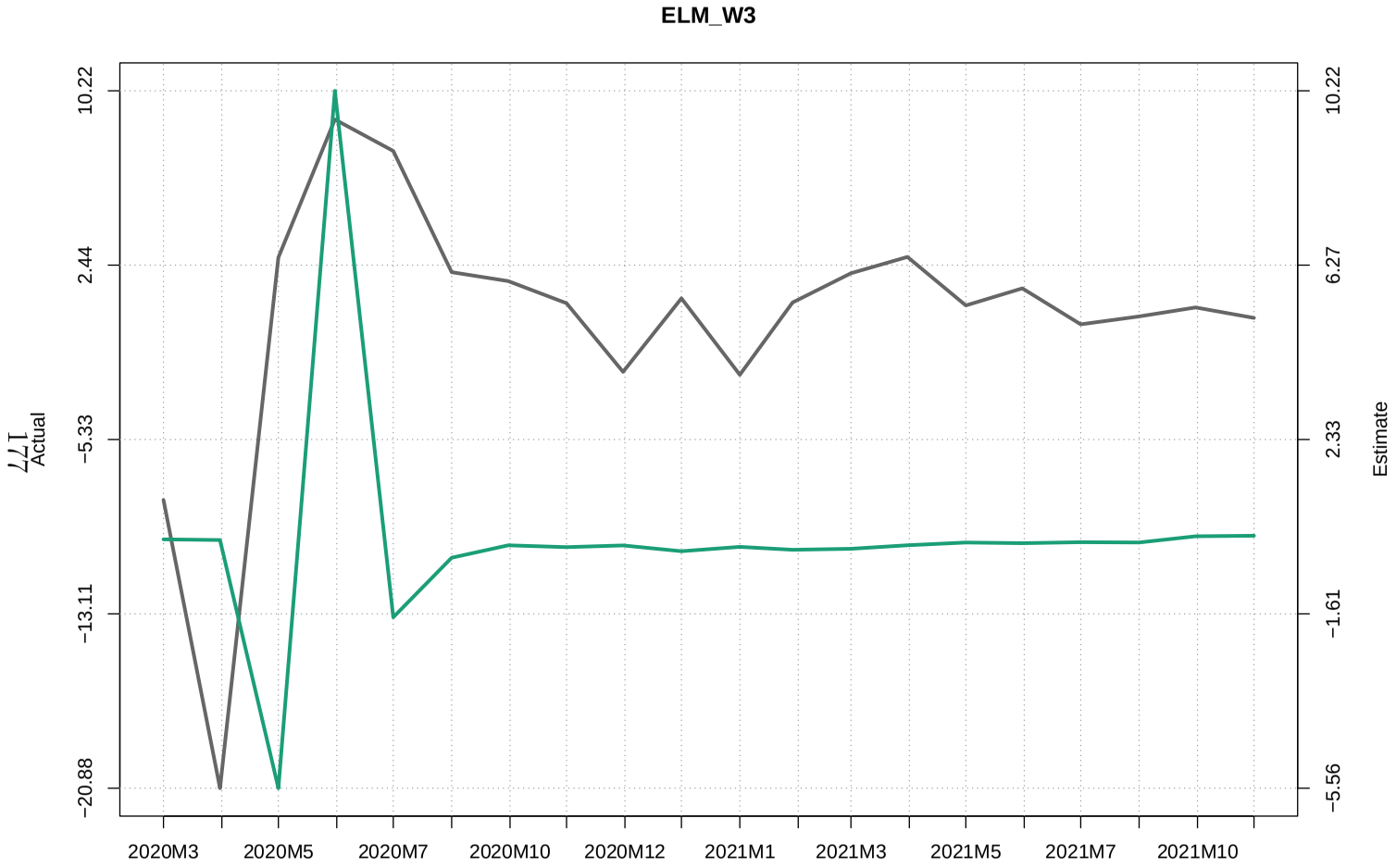

MLP_W1
Actual
T72
8.94
1.48
−5.97
−13.43
−20.88
Estimate
0.76
−0.89
−2.55
−4.21
−5.86
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


ELM_W1
Actual
Estimate
11.22
3.19
−4.83
−12.86
−20.88
11.22
7.05
2.88
−1.3
−5.47
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


MLP_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
1.61
−0.28
−2.18
−4.07
−5.96
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


ELM_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
3.44
1.23
−0.98
−3.19
−5.4
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

MLP_W3
Actual
8.94
1.48
−5.97
−13.43
−20.88
Estimate
1.35
−0.47
−2.3
−4.13
−5.96
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


ELM_W3
Actual
Estimate
10.22
2.44
−5.33
−13.11
−20.88
6.27
2.33
−1.61
−5.56
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

SVR_W1
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.23
−0.07
0.09
0.25
0.41
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10


SVR_W2
Actual
Estimate
8.94
1.48
−5.97
−13.43
−20.88
0.3
0.16
0.03
−0.1
−0.24
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10

SVR_W3
Actual
−20.88
−13.43
−5.97
1.48
8.94
Estimate
−0.17
−0.04
0.1
0.23
0.36
2020M3
2020M5
2020M7
2020M10
2020M12
2021M1
2021M3
2021M5
2021M7
2021M10