Paulsen, Kenneth Sæterhagen; Fastbø, Tuva Marie; Ingebrigtsen, Tobias

Working Paper

Aggregate density forecast of models using disaggregate data - A copula approach

Working Paper, No. 5/2022

Provided in Cooperation with:
Norges Bank, Oslo

Suggested Citation: Paulsen, Kenneth Sæterhagen; Fastbø, Tuva Marie; Ingebrigtsen, Tobias (2022) : Aggregate density forecast of models using disaggregate data - A copula approach, Working Paper, No. 5/2022, ISBN 978-82-8379-234-8, Norges Bank, Oslo, https://hdl.handle.net/11250/2997500

This Version is available at:
https://hdl.handle.net/10419/264947

WORKING PAPER

Aggregate density forecast of models using disaggregate data - A copula approach

NORGES BANK RESEARCH

5 | 2022

KENNETH SÆTERHAGEN PAULSEN,
TUVA MARIE FASTBØ AND
TOBIAS INGEBRIGTSEN

NORGES BANK

Working papers fra Norges Bank, fra 1992/1 til 2009/2 kan bestilles over e-post:
111facility@norges-bank.no

Fra 1999 og senere er publikasjonene tilgjengelige på www.norges-bank.no

Working papers inneholder forskningsarbeider og utredninger som vanligvis ikke har fått sin endelige form. Hensikten er blant annet at forfatteren kan motta kommentarer fra kolleger og andre interesserte. Synspunkter og konklusjoner i arbeidene står for forfatternes regning.

Working papers from Norges Bank, from 1992/1 to 2009/2 can be ordered by e-mai
FacilityServices@norges-bank.no

Working papers from 1999 onwards are available on www.norges-bank.no

Norges Bank's working papers present research projects and reports (not usually in their final form) and are intended inter alia to enable the author to benefit from the comments of colleagues and other interested parties. Views and conclusions expressed in working papers are the responsibility of the authors alone.

ISSN 1502-8190 (online)
ISBN 978-82-8379-234-8 (online)

Aggregate density forecast of models using disaggregate data - A copula approach*

Kenneth Sæterhagen Paulsen, Tuva Marie Fastbø and Tobias Ingebrigtsen

Norges Bank,

April, 2022

Abstract

We propose a novel copula approach to producing density forecasts of economic aggregates combining models using disaggregate data. Our copula approach is more flexible compared to existing techniques, because it is applicable to any econometric model that produces density forecasts. We construct a set of Monte Carlo studies to investigate the properties of the suggested approach. In our empirical application, we use the Norwegian index for goods consumption (VKI) and the Norwegian consumer price index for underlying inflation (CPI-ATE). We find that the copula approach compares well to alternative methods using recursive out-of-sample estimation.

Keywords: Aggregate forecast, disaggregates, density forecast, copula

JEL classification: C53, E27

*This working paper should not be reported as representing the views of Norges Bank. The views expressed are those of the authors and do not necessarily reflect those of Norges Bank. We have received useful comments from colleagues at Norges Bank, especially Francesco Ravazzolo, Knut Are Aastveit, Karsten R. Gerdrup and Leif Anders Thorsrud.

1 Introduction

Policymakers rely on well-calibrated forecasts of economic aggregates to form good economic decisions. In order to assess forecast uncertainty, density or interval forecasts can provide useful information (see Granger and Pesaran (2000) and G. Elliot and Timmermann (2006)). Several central banks, including the Bank of England and the Riksbanken, regularly publish probabilistic forecasts of economic aggregates in their monetary policy reports. Using disaggregate data when forecasting aggregate series has proven to improve performance of density forecasts (see Ravazzolo and Vahey (2014) and Mazur (2015)).

However, previous studies that explore techniques to produce density forecasts of economic aggregates based on disaggregate information, only consider a restrictive subset of the now available econometric models. Mazur (2015) applies a restricted vector autoregression (VAR) approach to produce probability forecasts, building on the methodology from Hubrich (2005) and Hendry and Hubrich (2011). Ravazzolo and Vahey (2014) produce forecast densities for economic aggregates from disaggregate ensembles by exploiting the connection between the ensemble approach, as put forward in Jore *et al.* (2010), and the bottom-up approach, as surveyed in Lütkepohl (2009).

In this paper, we extend the methodology from Ravazzolo and Vahey (2014) and apply copula methodology to produce density forecasts of economic aggregates using disaggregate models. Our copula approach allows us to construct a joint probability distribution, even when the disaggregates are forecasted with different econometric methods. It also compares well with alternative methods using recursive out-of-sample estimation.

More specifically, we parameterize a Gaussian copula with an estimate of an auto-cross-correlation matrix between the disaggregate time series. By using the copula in combination with the different marginal distributions, we are able to construct an estimate of the desired joint distribution for the density forecast. This enables us to draw consistent forecast paths of the disaggregate time series over the entire forecast horizon. In contrast to Ravazzolo and Vahey (2014), we make use of the historical autocorrelation found between the disaggregated series to produce density forecasts of the aggregate series.

Our copula approach relates to Smith and Vahey (2016). They use a Gaussian copula model to take into account cross-sectional and serial dependence in time series. The copula model is set up using a correlation matrix that is parameterized using a latent stationary Markov vector autoregression (MVAR) model and marginal distributions that are estimated using a kernel density estimator or a skew t distribution. Smith and Vahey (2016) document that their model compares well, in terms of out-of-sample real time forecast, with Bayesian vector autoregression models that assume symmetric marginal distribution of the data. We could model the disaggregated series using their approach and construct a density forecast of the aggregate using the joint density produced by this model. Nevertheless, this would not make us able to combine density forecasts from a set of models for each disaggregate series and then combine these afterwards, which is the case for our approach.

We explore different weighting schemes for aggregating the disaggregate forecasts. Hendry and Hubrich (2011) argue that the bottom-up approach, where the weights are fixed and known ex ante, cannot approximate a true multivariate model. Previous studies on time-varying weights for density forecasting give support to this view (see for instance Waggoner and Zha (2012), Billio M. and Dijk (2013), Del Negro and Schorfheide (2016) and Aastveit *et al.* (2018)). Using simulations from the joint distribution with our copula approach, we apply the summation formula with fixed weights as well as dynamically selected weights based on forecast performance to calculate the density forecast of the aggregate series.

To apply the copula approach empirically, we use data on the Norwegian index for goods consumption (VKI) and the Norwegian consumer price index adjusted for tax changes and excluding energy products (CPI-ATE). We model each of the subgroups as univariate AR(1) models and generate recursive out-of-sample forecasts. We utilise our copula approach to construct recursive out-of-sample forecasts on the aggregate index. To compare the method against alternative approaches, we estimate an AR(1) model of the aggregate series and a VAR with 2 lags of the disaggregate series. We evaluate the forecasts using mean square errors (RMSE) and mean log scores (MLS). We also provide a Monte Carlo simulation where we use two different approaches (copula and VAR) for producing recursive out-of-sample density forecasts of the aggregate series. We find that our copula based method compares well with the alternative approaches. More importantly, our approach gives more flexibility to practitioners in central banks in terms of methods for forecasting the disaggregate time series.

The paper is organized as follows. In section 2, we describe the problem in general terms. In section 3, we describe in detail the assumptions we need and the methodology we use. In section 4, we perform a Monte Carlo based application to test the performance of the algorithm presented in this paper, while in section 5 we give an empirical application. In the final section we conclude.

2 General problem

Assume we start out with a set of known marginal distributions. For clarity, we restrict it to two marginal distributions. We abbreviate the two marginal distributions with $b(x)$ and $g(y)$. We are interested in the distribution of $z = F(x, y)$, where $F(x, y)$ is a function in the random variables X and Y. This means that we must find the joint distribution $p(x, y)$ such that

$$b(x) = \int_y p(x, y)dy \tag{1}$$

and

$$g(y) = \int_x p(x,y)dx. \tag{2}$$

One possible solution to this problem is to use a copula. First, let us assume that $B(x)$, $G(y)$ and $P(x,y)$ represent the cumulative density functions of the corresponding probability density functions. By Sklar's theorem we then have

$$P(x,y) = C(B(x), G(y)), \tag{3}$$

i.e. there exists a copula $C(u_x, u_y)$, where $u_i \sim U(0,1)$ for $i \in \{x,y\}$, and a set of marginal distributions which can equally represent the joint distribution $P(x,y)$. As we do not know $P(x,y)$, we cannot find $C(u_x, u_y)$ either. However, we can postulate the shape of $P(x,y)$ or $C(u_x, u_y)$. As an example, if $P(x,y)$ is the multivariate normal distribution with correlation matrix Ω and $B(x)$ and $G(y)$ are normally distributed, then the Gaussian copula $C_\Omega(u_x, u_y)$ will be the correct copula to use.[1] If we are able to find $p(x,y)$, we can then simulate x and y from this distribution and use $z = F(x,y)$ to simulation from the distribution of z.

3 Aggregating density forecast

When we have an aggregate time series that is a weighted sum of a set of disaggregate series, the copula approach is able to use information from the disaggregate in order to improve forecast accuracy on the aggregate.

Since we are generally interested in more than a one-step-ahead forecast, we have to adjust the general problem from the previous section. The joint distribution needs to be formulated for all forecast horizons, as the macroeconometric time series are autocorrelated.

Let Z^h be the forecast of the aggregate series at horizon $h \in \{1, ..., H\}$, where H is the maximum forecast horizon. In the same way, we define X^h and Y^h to be the forecast of the disaggregate series at horizon h. In this case, we are interested in the marginal distribution of Z^h for each horizon h. As the disaggregate time series exhibit autocorrelation, we cannot find an estimate of these marginal distributions without finding an estimate of the distribution $P(x^1, \ldots, x^H, y^1, \ldots, y^H)$, i.e. the joint distribution of the forecast of all disaggregate series at each horizon. Based on this distribution we can draw consistent paths of the disaggregate series, both over time and states. As the aggregate series are a weighted sum of the disaggregate series, we have that $Z = F(X,Y) = w_x X + w_y Y$, where w_x and w_y are diagonal matrices with size $h \cdot h$, Z is the stacked forecast of the aggregate series over all horizons, while X and Y are the stacked forecast of the disaggregate series. This can of course be generalized to the case that a series consists of more than two disaggregate series.

Seeing that we were able to apply our problem to the general form, we now have two

[1] See section A for the definition of a Gaussian copula.

options. Either find a model that provides us with the joint distribution directly, or make a model for each disaggregate series that provides us with the marginal distributions and then estimate a copula. In the latter case the marginal distributions and the copula are enough to provide us with an estimate for the joint distribution.

One way to provide the joint distribution directly, is to model the disaggregate series jointly in a VAR, as in Hubrich (2005) and Hendry and Hubrich (2011). We will use this as our benchmark in the empirical applications in section 5, and as the true model in our monte carlo exercise in section 4.

We model each disaggregate series separately. We then estimate the auto-cross-correlation matrix of the disaggregate series to estimate a Gaussian copula. To be able to do so we need some assumptions, these are provided in section 3.1, while the methodology is described in detail in section 3.2.

3.1 Assumptions

Let $\{X_{n,t}\}$ represent N stochastic processes, for $n \in \{1, ..., N\}$, with known weights, $w_{n,t}$, that are time-varying or fixed and $\sum_{n=1}^{N} w_{n,t} = 1$. Let $\{Y_t\}$ be a multivariate stochastic process consisting of the N stochastic processes $\{X_{n,t}\}$.

Assumption 1. The stochastic process $\{Y_t\}$ is assumed in this paper to satisfy

1. Strict stationarity
2. Ergodicity
3. $\mathbb{E}\left[Y_t^2\right] < \infty$.

3.2 Methodology

Let Y be the series of historical observations of all the disaggregate series with size $N \times T$, where T is the number of historical observations and N is the number of disaggregate series. Let $\theta_{n,t+h}$ be the marginal distribution (CDF) of the density forecast for variable n at time t and horizon $h \in \{1, ..., H\}$, where H is the maximum forecast horizon. We assume that these density forecasts can come from any econometric time series model or survey. We are interested in constructing a density forecast of the aggregate series by drawing paths from the $n \cdot h$ marginal distributions $\theta_{n,t+h}$, given the estimated auto-cross-correlation between the disaggregate series. We do this by constructing a Gaussian copula. This copula is the marginal distribution of the density forecasts that are produced by the models in this paper, which are either normal or close to normal.[2]

[2] A more general class of copulas may be found in Demarta and McNeil (2005), i.e. T-copulas with extensions. This class of copulas may perform better if the marginal distributions exhibits fat tails or strong skewness.

Given the assumptions 1, we can consistently estimate the auto-cross-covariance matrix $\gamma_{k,t}$ for all $k \in \{1, ..., H\}$ up until time t by

$$\overline{\gamma_{k,t}} = \frac{1}{t-k} \sum_{s=1+k}^{t} (Y_s - \overline{y})' \otimes (Y_{s-k} - \overline{y}), \tag{4}$$

where $\otimes$ is the Kronecker product operator and $\overline{y}$ is the mean of the process. By stacking the auto-cross-covariance matrices $\gamma_{k,t}$ for the different k's we get the covariance matrix of our copula.

$$\Gamma_t = \begin{bmatrix} \gamma_{0,t} & \gamma_{1,t} & \cdots & \cdots & \cdots & \gamma_{h-1,t} \\ \gamma_{1,t} & \ddots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \gamma_{0,t} & \gamma_{1,t} & \cdots & \vdots \\ \vdots & \vdots & \gamma_{1,t} & \gamma_{0,t} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{h-1,t} & \cdots & \cdots & \cdots & \cdots & \gamma_{0,t} \end{bmatrix}. \tag{5}$$

Constructing Γ_t in this way leads to estimation of $\frac{N(N+1)}{2}h$ covariance coefficients. To scale down the number of estimated covariance coefficients and to prevent over-fitting we may set all $\gamma_{k,t} = 0$ for $k > \tau$. We can get the correlation matrix by using the following relationship

$$\Omega_t = \Gamma_t \oslash \gamma_t. \tag{6}$$

where $\oslash$ is the Hadamard division operator and

$$\gamma_t = \sqrt{diag(\Gamma_t)}\sqrt{diag(\Gamma_t)}'. \tag{7}$$

Ω_t is the auto-cross-correlation matrix we need to make consistent draws from the density forecasts from the disaggregate series. We can construct a Gaussian copula over the H step ahead forecasts as[3]

$$C_{\Omega_t}|\mathbb{I} = \Phi_{\Omega_t}\left(\Phi^{-1}(u_{1,1}), ..., \Phi^{-1}(u_{N,H})\right), \tag{8}$$

where $\mathbb{I}$ is the information set up until the time of the forecast. The full multivariate distribution is then given by

$$G_{\Omega_t}|\mathbb{I} = \Phi_{\Omega_t}\left(\Phi^{-1}(\theta_{1,1}(y_{1,1})), ..., \Phi^{-1}(\theta_{N,h}(y_{N,H}))\right), \tag{9}$$

where $y_{n,t+h}$ is the forecast for variable n at time t and horizon h, which is a random variable. By using the algorithm presented in appendix B, we can make Q draws from the multivariate

[3]See appendix A for more on the concept of a Gaussian copula.

distribution $G_{\Omega_t}|\mathbb{I}$, i.e. draw realizations of the random variables $y_{n,t+h}$. Abbreviate these draws by $y_{n,t+h,q}$, for all $q \in \{1, ..., Q\}$. Finally, to make Q draws from the density forecast of the aggregate series X we use

$$x_{t+h,q} = \sum_{n=1}^{N} w_{n,t+h} y_{n,t+h,q}, \tag{10}$$

where $w_{n,t+h}$ is the weight of each variable n in the aggregate series and $x_{t+h,q}$ is the qth realization from the distribution of the aggregate density forecast at time t and horizon h.

4 Monte Carlo experiment

We simulate three disaggregate series to compare the forecasting capabilities of our algorithm. We use a vector autoregressive model that is estimated on the true data generating model which reports forecasts using the actual weights of each component as our benchmark model.

4.1 Simulation of data

For each simulation iteration s, we draw two 4×4 coefficient matrices, ϕ_1^s and ϕ_2^s, and then we run j simulations of four time series, where $j \in \{1, ..., H\}$, by using a vector autoregressive model of the form

$$Y_{t,j}^s = \sum_{i=1}^{2} \phi_i^s Y_{t-i,j}^s + \varepsilon_{t,j}^s, \tag{11}$$

where

$$Y_{t,j}^s = \begin{bmatrix} y_{1,t,j}^s \\ y_{2,t,j}^s \\ y_{3,t,j}^s \\ y_{4,t,j}^s \end{bmatrix}, \tag{12}$$

$\varepsilon \sim \mathcal{N}(0, 1)$, $t \in \{-9, ..., 100\}$ (we use a burn-in period of 10 observations), and $Y_{t,j}^s = 0$ for $t \in \{-9, -8\}$. In this experiment we draw 100 pairs of coefficient matrices and scale all 100 pairs by 1, 1.5, and 2.5[4]. For every draw s we simulate j paths of the three disaggregate series resulting in a total of 9000 paths. The simulated series all have a length of 100 observations. The aggregate series X_j^s is then constructed by

$$x_{t,j}^s = \sum_{n=1}^{3} w_n y_{n,t,j}^s, \tag{13}$$

[4]We make sure that all eigenvalues lies within the unit circle, resulting in a stationary time series.

where $s \in \{1, ..., 300\}$ is the draw index, $t \in \{1, ..., 100\}$ is the observation index, $j \in \{1, ..., 30\}$ is the simulation path index, and $n \in \{1, ..., 4\}$ is the series index.

Since we are generating data from a VAR model, we are also interested in how the properties of the coefficient matrices, ϕ_1^s and ϕ_2^s, affect our results. For each draw of the coefficient matrices, we find the eigenvalues and store them in a vector, $\mathbf{v}^s$. We then take the norm of the vector to determine the magnitude of the eigenvalues

$$norm^s = \|\mathbf{v}^s\|.$$

We use $norm^s$ to see how our results are affected by different magnitudes of the eigenvalues.

4.2 Model estimation and forecasting

We use two different approaches (copula and VAR) for producing recursive out-of-sample density forecasts of the aggregate series X_j^s for the periods $t \in \{40, ..., 100\}$ at horizons $h \in \{1, ..., 6\}$.

In the first approach (copula) we estimate an AR(1) model for each disaggregate series for every simulated path $j \in \{1, ..., 30\}$ for each coefficient draw $s \in \{1, ..., 300\}$. These recursive estimates are then used to produce recursive out-of-sample forecasts for the periods $t \in \{40, ..., 100\}$. To produce density forecasts we do 1000 draws of the residual using the estimated standard deviation of the residual, which makes a total of 1000 draws from the unknown probability distributions of each forecast.[5] This procedure is done for each series individually at all horizons $h \in \{1, ..., 6\}$. We let $Z = 60$ be the number of recursive forecasting periods. For each variable $X_{n,j}^s$, we estimate the PDF of the density forecast $\theta_{n,h,z,j}^s$ using a Gaussian kernel density estimator. These estimates are based on the 1000 draws, at horizons $h \in \{1, ..., 6\}$, simulation $s \in \{1, ..., 300\}$, and periods $z \in \{1, ..., 60\}$. We then have all we need to construct the density forecasts of the aggregate series using a copula, as described in section 3.2, as we use the known weights w_n for $n \in \{1, ..., 4\}$. In this experiment we have set $\tau = 2$.

In the second approach (VAR) we estimate a VAR(2), i.e. we estimate equation (15) on the simulated data. This will serve as the benchmark model. We use the recursive estimates to produce out-of-sample forecasts recursively for the periods $t \in \{40, ..., 100\}$. To produce density forecasts we take 1000 draws of the residuals using the estimate of the covariance matrix of the residuals of the VAR(2), which in the simulation step is assumed to be equal to the identity matrix, I. This makes a total of $Q = 1000$ draws from the unknown probability distributions of the forecasts. The procedure is done for each series individually at all horizons $h \in \{1, ..., 6\}$.

[5] Indeed, we could have use the theoretical distribution in this case, but in the general case the theoretical distribution is not available, and we want to describe an experiment that can be easily extended to more advanced modeling approaches.

To construct the density forecasts of the aggregate series in the last case we use

$$x^s_{h,j,z,q} = \sum_{n=1}^{4} w_n y^s_{n,h,j,z,q}, \tag{14}$$

where $y^s_{n,h,j,z,q}$ is the forecast of the disaggregate series n, at horizon h, for period t, coefficient draw s, simulation path j, and draw $q \in \{1, ..., 1000\}$. w_n are the known weights. We can then estimate the PDF of the aggregate forecasts constructed in this way by using a Gaussian kernel density estimator on the Q draws.

For each coefficient draw s we run $j \in \{1, ..., 30\}$ simulated paths. For each j, we get reported mean log scores. Then, we calculate the mean and variance of the reported mean log scores across j for all horizons $h \in \{1, ..., 6\}$. We then have the expected mean log scores ($\mu^s_{MLS,h}$), the corresponding variances to those estimates ($\sigma^s_{MLS,h}$), and the norm of the eigenvalues ($norm^s$) at every horizon h.

Results from experiment To evaluate our algorithm we are mainly concerned about

1. How is the preciseness of the forecasts, as measured by the mean log score, compared to our benchmark (VAR)?

2. How do the properties of the coefficient matrix used to simulate the series affect the uncertainty of our results?

To evaluate 1, we get the reported mean of the mean log scores of the density forecasts and the associated variance of the mean log scores at every horizon, h, from subsection 4.2. By assuming that the mean of the mean log scores are normally distributed, we can form a 95% confidence interval as $\mu^s_{MLS} \pm 1.96\sigma^s_{MLS}$, for each horizon h. We plot the benchmark model and the lower and upper 95% limits for each simulation s and horizon h against an increasing $norm^s$. See figures 14 to 19. From the figures, we see that $norm^s$ seems to play a role in the performance of our algorithm. We see that at the lower half of $norm_s$, the precision of our algorithm is not statistically significantly different from the benchmark (VAR) at the 95% significance level. At the upper part of $norm^s$, the precision of our algorithm decrease and becomes more unstable. The mean of the mean log score is often outside the 95% interval of the benchmark (VAR).

To evaluate 2, we look at the sensitivity of the variance of the mean log score towards increasing $norm^s$ for our algorithm (*copula*) against our benchmark (VAR). This is to see whether or not their sensitivities are statistically significantly different from each other.

First, we run a regression of the form

$$\sigma^s_{RMSE,i} = c^s + \beta_i norm^s + \varepsilon_s,$$

for i $\in \{copula, VAR\}$. Then, we do a test for differences in slopes where the test statistic is calculated as

$$t = \frac{\beta^s_{copula} - \beta^s_{VAR}}{\sqrt{S^2_{copula} + S^2_{VAR}}} \sim t(2s-4).$$

The associated p-value of our test statistic, t, is summarized in Table 4.2. With a significance level of 95%, the slopes β^s_{copula} and β^s_{VAR}, are statistically significantly different from each other at horizons 1, 2 and 3, but not at longer horizons.

Table 1: P-values

Horizon	1	2	3	4	5	6
P-value	0.0000	0.0010	0.0283	0.0576	0.0700	0.2145

To summarize, our algorithm does not perform statistically significantly differently from the benchmark. However, our algorithm is sensitive to an increase in the norm of the eigenvalues of the coefficient matrix of the model that drives the data at longer horizons.

4.3 Additional tests

To investigate the performance of the copula approach even further, we compare 4 Monte Carlo studies. They are equal to the Monte Carlo study described in section 4, except that the simulation of ϕ is drop, i.e. we set it to

$$\phi = \begin{bmatrix} 0.2063 & 0.0180 & 0.0290 & -0.0120 \\ 0.0017 & 0.2759 & -0.0115 & -0.0187 \\ -0.3209 & 0.0850 & 0.4522 & 0.4415 \\ -0.0336 & 0.0552 & 0.0080 & 0.2853 \end{bmatrix}. \tag{15}$$

Otherwise the studies depart as follows

1. Benchmark: No more departures from the Monte Carlo described in section 4.
2. Only two observed: We only observed 2 out of the 4 disaggregated variables. In this case we construct a third variable which is constructed by subtracting the contribution of the 2 observed disaggregated variables from the aggregated variable. Then we create a AR(1) model of this variable and treat it like any other disaggregated variable. The weight of this third variable will be set to 1.

3. In this case we let ϕ vary over time. We do this by adding a time-varying component (ϕ_t) that is simulated using the process

$$\phi_t = \lambda\phi_{t-1} + \epsilon_t, \tag{16}$$

where λ is a 16×16 diagonal matrix, where all coefficients along the diagonal is set to 0.95. All elements of ϵ_t are uncorrelated and are distributed $\mathcal{N}(0, 0.01)$.

4. Time-varying weights: In this case we let w_n vary over time for all $n \in \{1, ..., N\}$. We do this by adding a time-varying component ($w_{t,n}$) that is simulated using the process

$$w_{t,n} = \lambda_n w_{t-1,n} + u_{t,n}, \tag{17}$$

where $\lambda_n = 0.95$ and $u_{t,n} \sim \mathcal{N}(0, 0.01)$.

The reported mean of mean log scores are found in Table 4.3, while the mean of root mean squared errors are found in Table 4.3. The results from the Copula and VAR approaches are never significantly different, but the copula approach is marginally better at the benchmark and when only two disaggregated variables observed, otherwise the VAR approach is marginally better.

Table 2: Mean of mean log scores of Monte Carlo studies

	Horizon	Copula	VAR
Benchmark	1	-0.78	-0.82
Only two observed	1	-0.76	-0.79
Time-varying parameters	1	-0.78	-0.77
Time-varying weights	1	-0.79	-0.78
Benchmark	2	-0.86	-0.89
Only two observed	2	-0.85	-0.87
Time-varying parameters	2	-0.85	-0.85
Time-varying weights	2	-0.87	-0.87
Benchmark	3	-0.87	-0.90
Only two observed	3	-0.87	-0.88
Time-varying parameters	3	-0.86	-0.87
Time-varying weights	3	-0.88	-0.89

Table 1: Mean over 500 simulations of the mean log scores at different horizons for the 4 monte carlo studies.

Table 3: Mean of root mean squared errors of Monte Carlo studies

	Horizon	Copula	VAR
Benchmark	1	0.53	0.54
Only two observed	1	0.51	0.52
Time-varying parameters	1	0.52	0.51
Time-varying weights	1	0.53	0.52
Benchmark	2	0.57	0.58
Only two observed	2	0.56	0.57
Time-varying parameters	2	0.56	0.56
Time-varying weights	2	0.57	0.57
Benchmark	3	0.58	0.58
Only two observed	3	0.57	0.58
Time-varying parameters	3	0.57	0.57
Time-varying weights	3	0.58	0.58

Table 2: Mean over 500 simulations of root mean squared errors at different horizons for the 4 monte carlo studies.

5 Empirical application

In the empirical application of our approach, we use data on the Norwegian index of household consumption of goods (VKI) and the Norwegian consumer price index adjusted for tax changes and excluding energy products (CPI-ATE). Both VKI and CPI-ATE consist of a set of subgroups that are likely to inhabit a certain degree of correlation, thereby giving relevance to the copula approach.

5.1 Modeling approach

We model each disaggregate series as univariate AR(1) models

$$x_{n,t} = \lambda x_{n,t-1} + \epsilon_t, \tag{18}$$

where the λ is the AR coefficient. We also assume that

$$\epsilon_t = N(0, \sigma^2), \tag{19}$$

i.e. the residual is a white noise process. We estimate equation (18) using OLS. It is estimated recursively using an expanding window. These recursive estimates are then used to produce recursive out-of-sample forecasts from 2010M1 to 2019M12. To produce density forecasts from the model we simulate 1000 draws of the residual using the estimated parameters of λ and σ. This makes a total of 1000 draws from the unknown probability distributions of the forecasts. This procedure is done for each series individually at all horizons $h \in \{1, ..., H\}$.

We let S be the number of recursive forecasting periods. For each variable X_n, we estimate the PDF of the density forecast based on the 1000 draws, at horizons $h \in \{1, ..., H\}$ and periods $s \in \{1, ..., S\}$, using a Gaussian kernel density estimator. We then have all we need to construct the density forecasts of the aggregate series, as described in section 3.2. We replicate this step for all recursive forecasting periods, as we want to evaluate the forecasting performance against a set of alternative models. We denote the density forecasts of the aggregate series constructed by this algorithm as *Copula method on disaggregate series*.

In order to evaluate the proposed copula approach, we estimate two additional models. The first reference model consists of an AR(1) model of the aggregate series. We replicate the same forecasting steps, as in the case for the disaggregate models, i.e. a simple residual sampling method is used to draw 1000 draws form the probability distributions of the forecasts. Again, these draws are used to estimate the PDF of the density forecasts, using a Gaussian kernel density estimator. We denote the density forecasts produced in this way: *AR on aggregate series*.

The second reference model is a VAR with two lags, where we include the disaggregate series. Since the series are modelled jointly, we can directly sample from the joint probability distribution of the forecasts of the disaggregate series. We apply the weights directly to each draw from this distribution to get a simulation of the probability distribution of forecast of the aggregate series. Again we use a residual sampling method to draw from the joint distribution of the forecast from the VAR. We denote the density forecasts produced in this way: *VAR model on disaggregate series*.

5.2 VKI

The Norwegian index of household consumption of goods (VKI) is a volume index that measures movements in household consumption of both durable and non-durable goods. The data set employed is obtained from Statistics Norway and includes aggregate consumption of goods as well as a decomposition of four subgroups: (i) Food, beverages and tobacco, (ii) Electricity and heating fuels, (iii) Purchase of vehicles and petrol, and (iv) Other goods. The subgroups follow definitions from the Classification of Individual Consumption by Purpose (COICOP). We apply seasonally adjusted data for the period 2000M1 to 2019M12. For all series we use 3-month log difference. We apply fixed weights, that represent the average of the actual historical weights, as well as weights based on out-of-sample forecast performance from Ravazzolo and Vahey (2014).

To evaluate the performance of the density forecasts, we look at root mean square errors (RMSE) and mean log scores (MLS). Figures 1, 2, 3, and 4 show the forecast errors scores at horizon 1 and 2. The three approaches perform relatively similar. Over the estimation period, RMSE and MLS scores are not significantly different for the three model approaches.

In order to judge the calibration of the densities, we calculate the probability integral

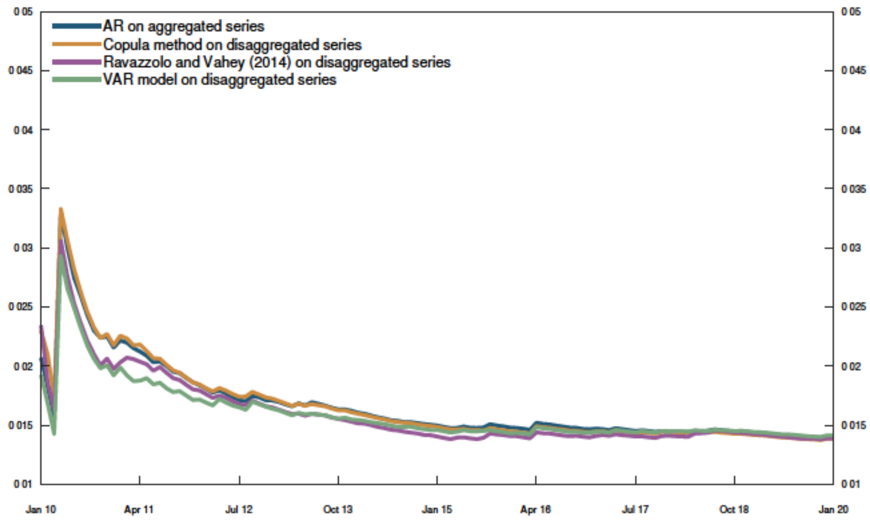

Figure 1: VKI. Mean square errors at horizon 1. Estimation period is from 2001M1 to 2019M12.

transforms (PIT) for the copula model, see figure 5. For the copula model we reject the null hypothesis of the PIT not being uniform using a Pearson chi-squared test, so we cannot reject that the model has well-calibrated density forecasts.

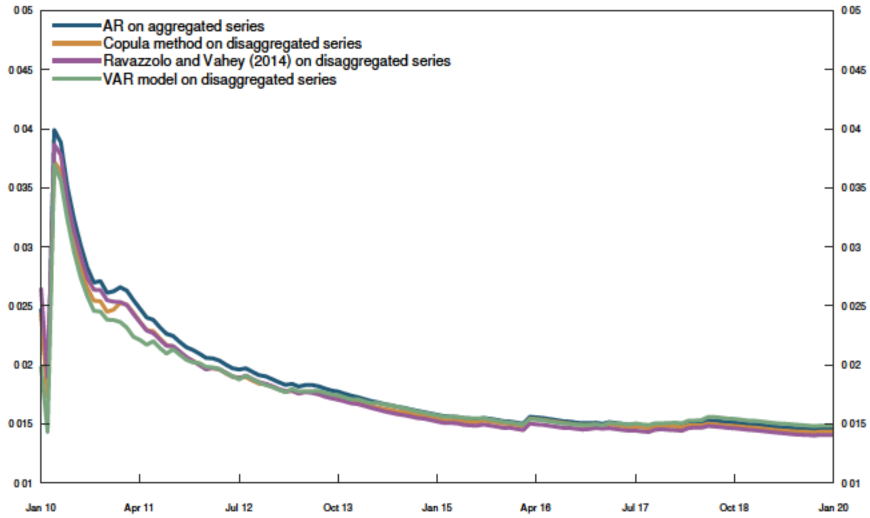

Figure 2: VKI. Mean square errors at horizon 2. Estimation period is from 2001M1 to 2019M12.

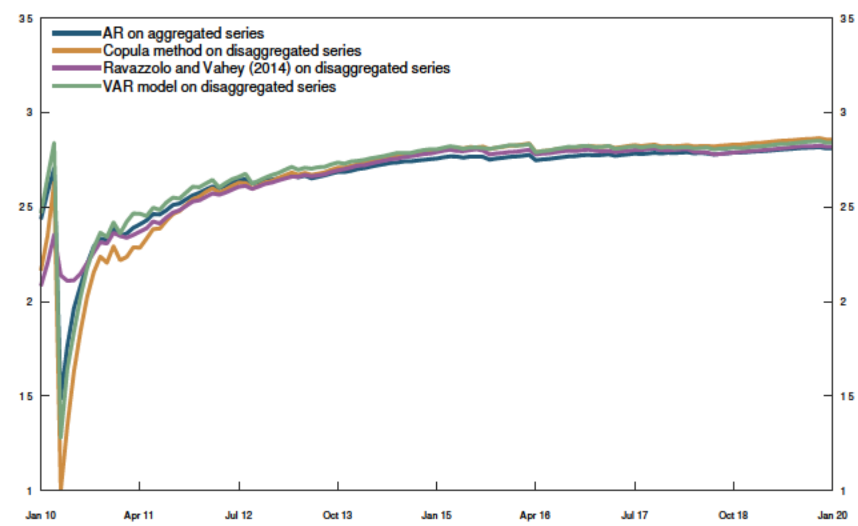

Figure 3: VKI. Mean log scores at horizon 1. Estimation period is from 2001M1 to 2019M12.

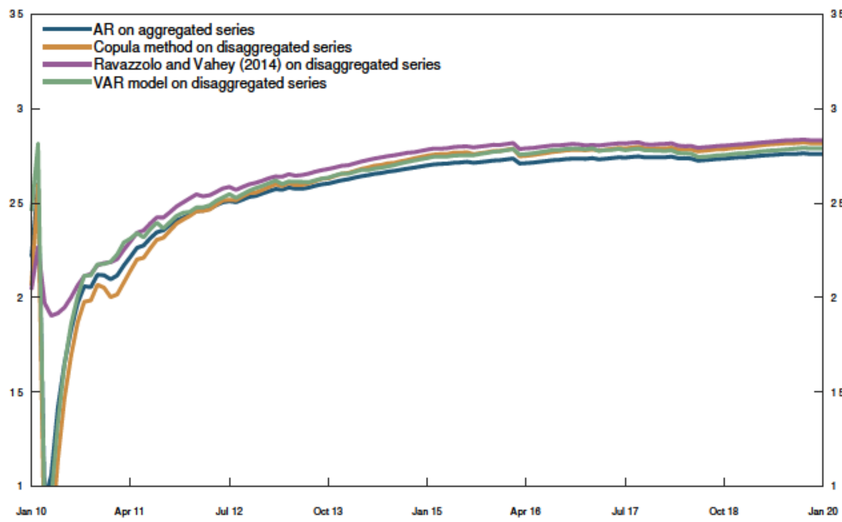

Figure 4: VKI. Mean log scores at horizon 2. Estimation period is from 2001M1 to 2019M12.

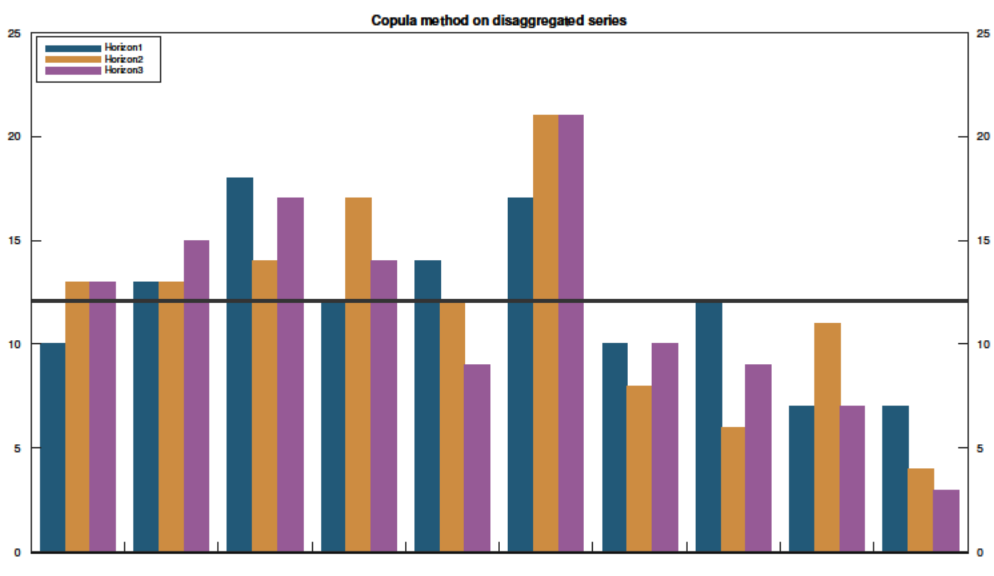

Figure 5: Probability integral transform of the forecast errors of the copula model for VKI at forecast horizons 1-3. Estimation period is from 2001M1 to 2019M12.

5.3 CPI-ATE

The Norwegian consumer price index adjusted for tax changes and excluding energy products (CPI-ATE), is the most common measure of underlying inflation in Norway. The index is published by Statistics Norway and consists of 12 subgroups.[6] We apply data for 2003M1 to 2019M12 and use dynamic weights. To prevent the models from using information that is not available at the time of the forecast, we forecast the weights recursively with a random walk. For all series we use a one-month log difference.

Figures 6, 7, 8, and 9 show the forecast errors scores at horizons 1 and 2, i.e. the copula approach has RMSE and MLS scores that are in line with the alternative approaches. Again, the scores are not significantly different.

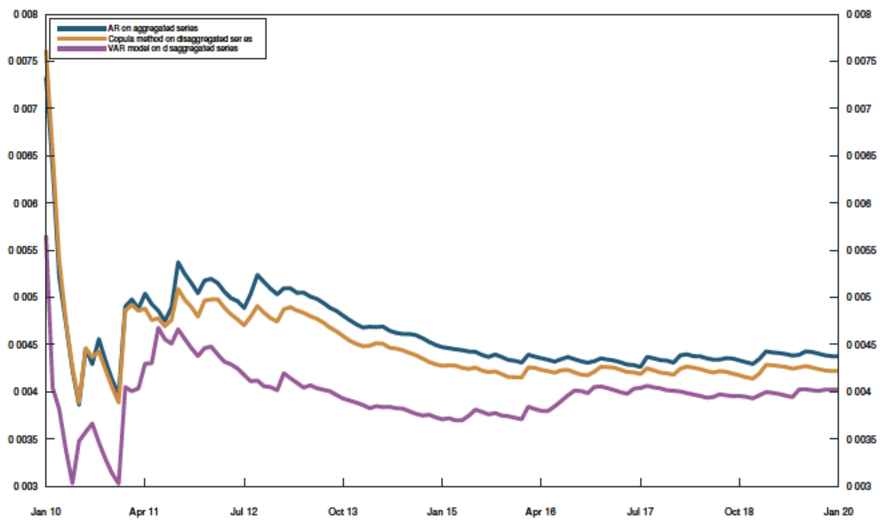

Figure 6: CPI-ATE. Mean square errors at horizon 1. Estimation period is from 2003M1 to 2019M12.

In order to judge the calibration of the densities, we calculate the probability integral transforms (PIT) for the copula model, see 10. For the copula model we reject the null hypothesis of the PIT not being uniform using a Pearson chi-squared test, so we cannot reject that the model has well-calibrated density forecasts.

[6]For one subgroup, education, we use an AR(1) model on annual data, as this series is only revised annually (in September). Then we convert the annually forecast to monthly before we aggregate the forecast with the other series. This is an illustration of how flexible our methodology is.

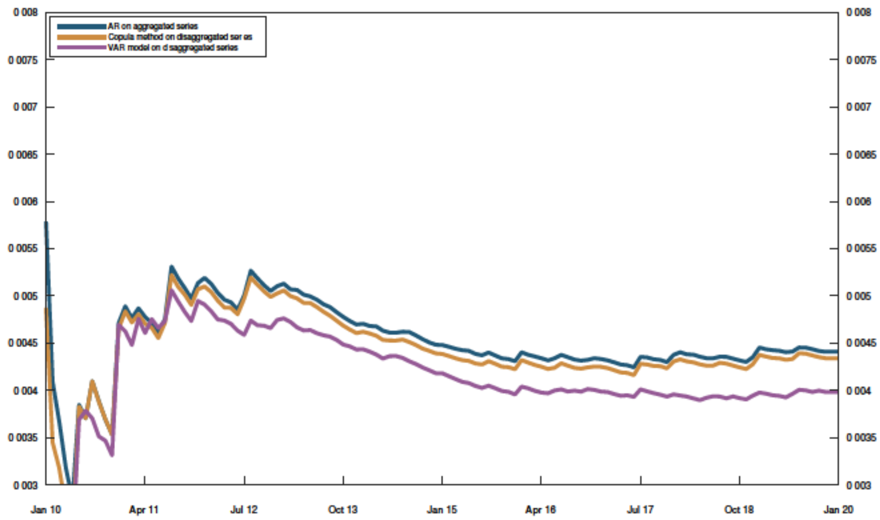

Figure 7: CPI-ATE. Mean square errors at horizon 2. Estimation period is from 2003M1 to 2019M12.

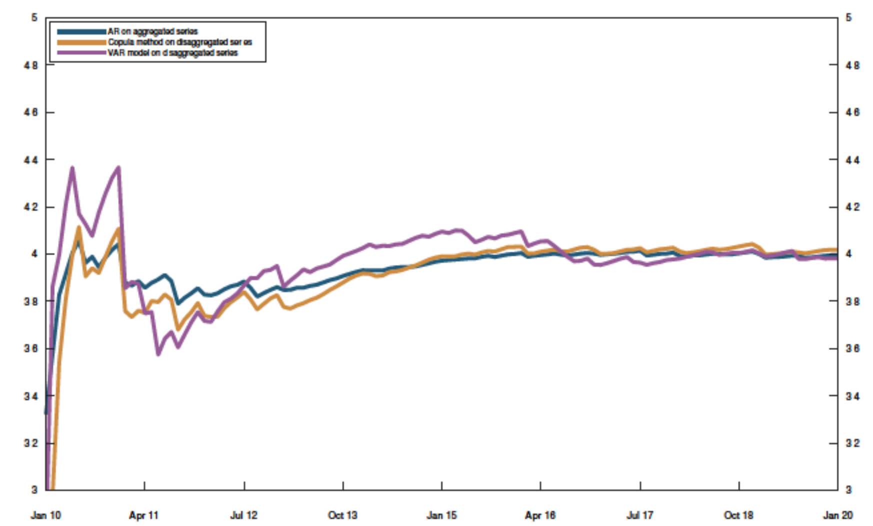

Figure 8: CPI-ATE. Mean log scores at horizon 1. Estimation period is from 2003M1 to 2019M12.

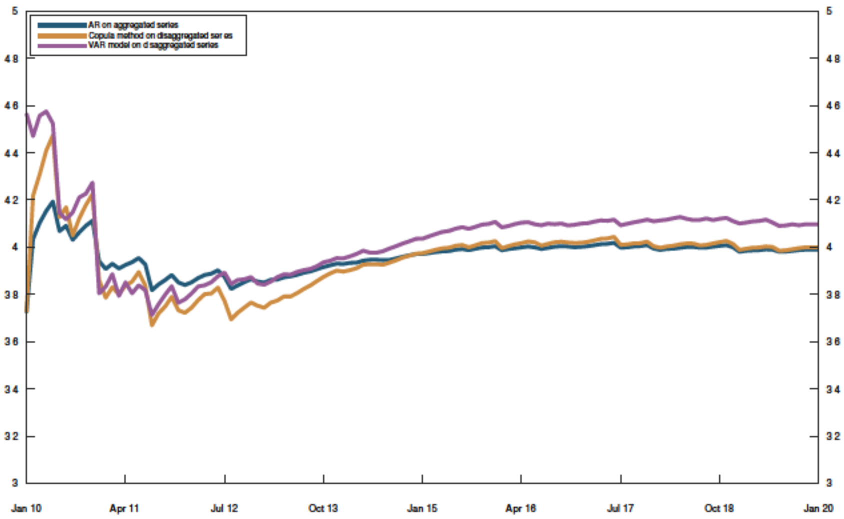

Figure 9: CPI-ATE. Mean log scores at horizon 2. Estimation period is from 2003M1 to 2019M12.

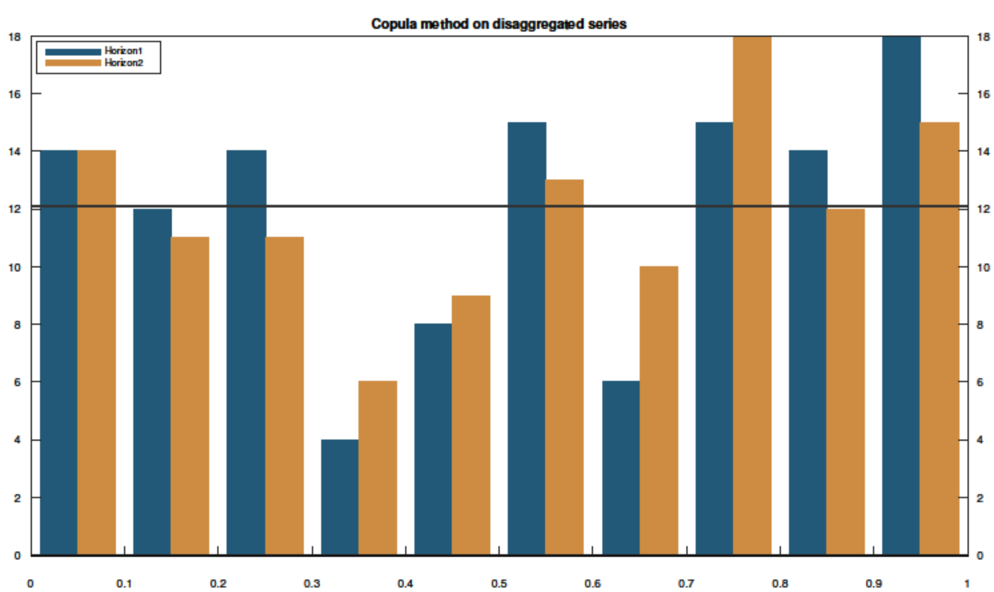

Figure 10: Probability integral transform of the forecast errors of the copula model for CPI-ATE at forecast horizons 1-3. Estimation period is from 2001M1 to 2019M12.

6 Conclusion

In this paper we have put forward a new approach to producing density forecasts of aggregate series using models on disaggregate data. Our approach applies copula methodology to simulate consistent draws across disaggregate density forecasts. Empirically, our approach compares well with a simple AR and VAR model when it comes to out-of-sample density forecast performance.

In a Monte Carlo experiment, the copula approach fares well compared with the benchmark (true) model, as long as the underlying processes of the aggregated and disaggregated series are not too close to being non-stationary.

The methodology presented does not limit itself to AR models. Any econometric model that produces density forecasts can be applied. In this respect, the copula approach is more flexible compared to other approaches suggested in previous literature. Future work should investigate how to select optimal models for forecasting disaggregate series.

References

Aastveit, K. A., Ravazzolo, F. and van Dijk, H. K. (2018) Combined density nowcasting in an uncertain economic environment, *Journal of Business and Economic Statistics*, **36(1)**, pp. 131–145.

Billio M., F. R., R. Casarin and Dijk, H. V. (2013) Time-varying combinations of predictive densities using nonlinear filtering, *Journal of Econometrics*, **192(2)**, pp. 213–232.

Del Negro, H. B. R., M. and Schorfheide, F. (2016) Dynamic prediction pools: An investigation of financial frictions and forecasting performance, *Journal of Econometrics*, **192(2)**, pp. 391–405.

Demarta, S. and McNeil, A. J. (2005) The t copula and related copulas, *International Statistical Review / Revue Internationale de Statistique*, **73**, 111–129.

G. Elliot, C., C. Granger and Timmermann, A. (2006) Forecast combination, *Handbook of Economic Forecasting*, pp. pp. 197–284.

Granger, C. and Pesaran, M. (2000) Economic and statistical measures of forecast accuracy, *Journal of Forecasting*, pp. pp. 537–560.

Hendry, D. F. and Hubrich, K. (2011) Combining disaggregate forecasts or combining disaggregate information to forecast an aggregate, *Journal of Business and Economic Statistics*, **29(2)**, pp. 216–227.

Hubrich, K. (2005) Forecasting euro area inflation: Does aggregating forecasts by hicp component improve forecast accuracy?, *International Journal of Forecasting*, **21(1)**, pp. 119–136.

Jore, A., Mitchell, J. and Vahey, S. (2010) Combining forecast densities from vars with uncertain instabilities, *Journal of Applied Econometrics*, **25(4)**, pp. 621–634.

Lütkepohl, H. (2009) Forecasting forecasting aggregated time series variables: A survey, Economics Working Papers 2009/17, European University Institute.

Mazur, B. (2015) Density forecasts based on disaggregate data: Nowcasting polish inflation, *Dynamic econometric models*, **15**, pp. 71–87.

Ravazzolo, F. and Vahey, S. P. (2014) Forecast densities for economic aggregates from disaggregate ensembles, Tech. rep.

Smith, M. S. and Vahey, S. P. (2016) Asymmetric forecast densities for u.s. macroeconomic variables from a gaussian copula model of cross-sectional and serial dependence, *Journal of Business & Economic Statistics*, **34(3)**, pp. 416–434.

Waggoner, D. F. and Zha, T. (2012) Confronting model misspecification in macroeconomics, *Journal of Econometrics*, **171**, pp. 167–184.

A Copula theory

A copula can be used to decompose a multivariate distribution into two parts: (i) The marginal distributions of each variable which describes the randomness in each variable and (ii) a copula which describes the dependence between the random variables. A copula is defined as a multivariate distribution where each marginal distribution is uniform. There are many such copulas, but in this paper we will only focus on the Gaussian copula. The cumulative distribution function (CDF) of this copula is given by

$$C_\Sigma = \Phi_\Sigma\left(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_N)\right), \tag{20}$$

where Φ_Σ is the multivariate normal CDF with correlation matrix Σ of size $N \times N$ and Φ is the univariate standard normal CDF. This means that

$$u_n \sim U(0,1) \text{ for } n \in \{1, ..., N\}, \tag{21}$$

where $U(0,1)$ is the uniform distribution on the interval $[0,1]$. Let the marginal distributions of the N variables be given by

$$x_n \sim F_n \text{ for } n \in \{1, ..., N\}, \tag{22}$$

where F_n can be any marginal CDF. By Sklar's theorem we can then represent the full multivariate CDF as

$$G_\Sigma = \Phi_\Sigma\left(\Phi^{-1}(F_1(x_1)), ..., \Phi^{-1}(F_N(x_N))\right). \tag{23}$$

The corresponding multivariate probability density function (PDF) is given by

$$c_\Sigma = \frac{1}{\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\begin{pmatrix}\Phi^{-1}(F_1(x_1)) \\ \vdots \\ \Phi^{-1}(F_N(x_N))\end{pmatrix}' \left(\Sigma^{-1} - \mathbf{I}\right)\begin{pmatrix}\Phi^{-1}(F_1(x_1)) \\ \vdots \\ \Phi^{-1}(F_N(x_N))\end{pmatrix}\right) \tag{24}$$

$$g_\Sigma = c_\Sigma \cdot f_1(x_1) \cdot \ldots \cdot f_N(x_N), \tag{25}$$

where c_Σ is the PDF of the copula, and $f_n(x_n)$ is the marginal PDFs.

A.1 Example

Let $y \sim N(2,2)$ and $x \sim GAMMA(2,2)$. Let the linear correlation between y and x be given by

$$\Sigma = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}. \tag{26}$$

Then we can construct a multivariate distribution in y and x by using a copula. The algorithm presented in section B let us draw random numbers from this distribution. In figure 11 you can see that the draws of y is coming from the $N(2,2)$ distribution, and that the draws of x is coming from the $GAMMA(2,2)$ distribution. The PDF and CDF of the multivariate distribution is plotted in figures 12 and 13.

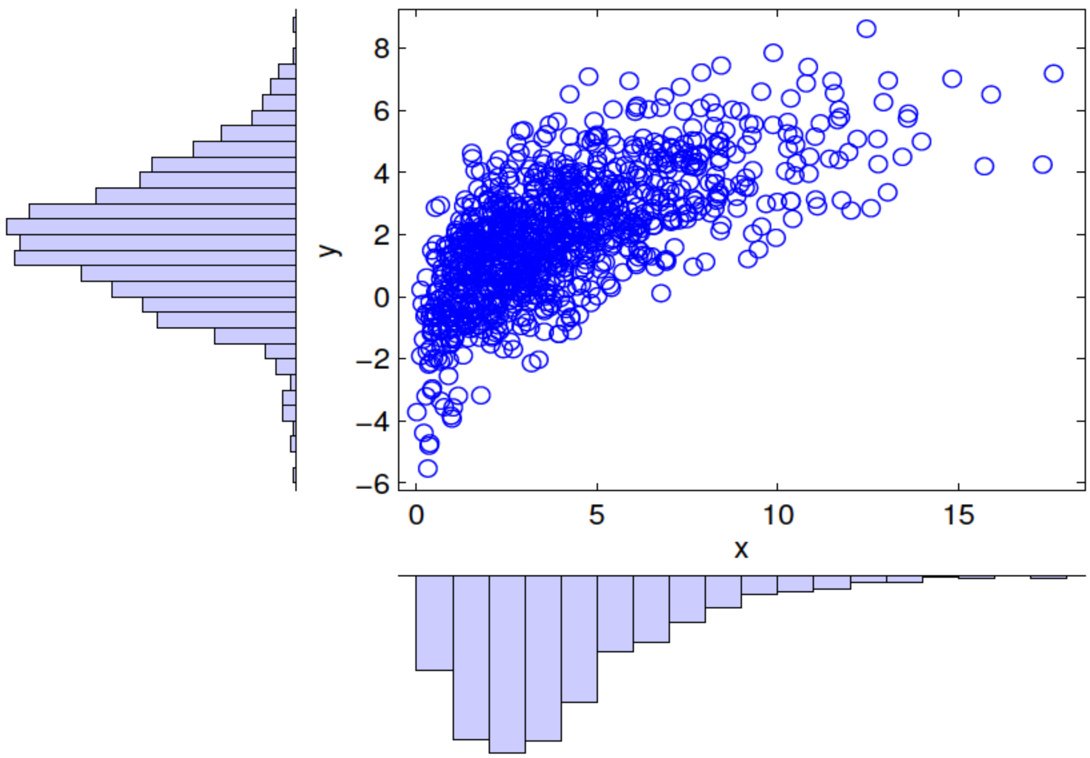

Figure 11: Drawing from a multivariate distribution using a copula.

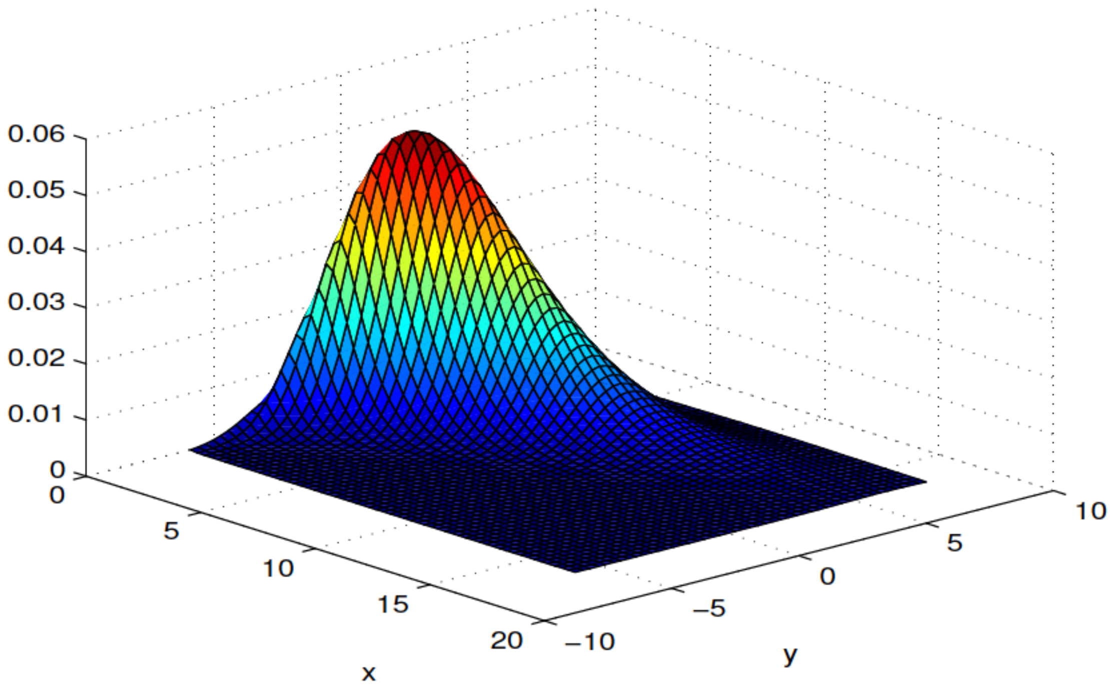

Figure 12: Multivariate PDF using a copula.

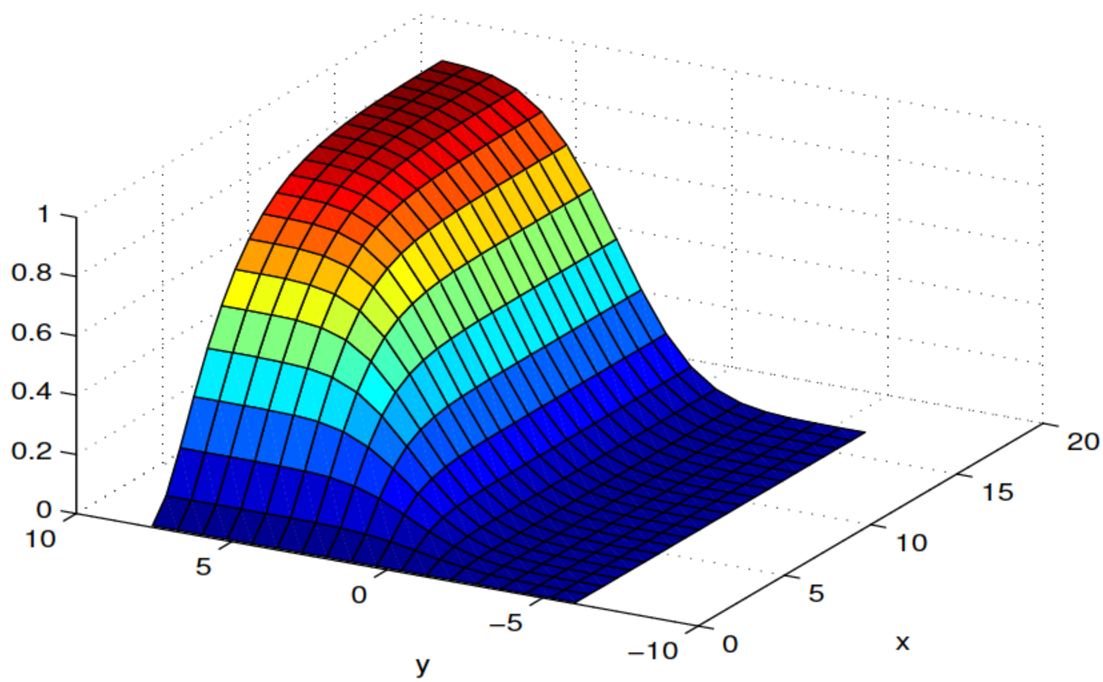

Figure 13: Multivariate CDF using a copula.

B Drawing random numbers from a Copula

To make draws from a general multivariate distribution G_Σ that is decomposed into a copula and a set of marginal distribution the following algorithm may be used

1. Draw Q number of observations from the multivariate normal distribution Φ_Σ. Abbreviate the draws from this distribution for variable $n \in [1, N]$ as y_n, which then has size $Q \times 1$.

2. For each variable $n \in \{1, ..., N\}$ map the observation found in step 1 to the interval $[0, 1]$ using $u_n = \Phi(y_n)$.

3. Map to the final draws from the marginal distribution of variable $n \in \{1, ..., N\}$ by $x_n = F_n^{-1}(u_n)$.

C Graphs and tables

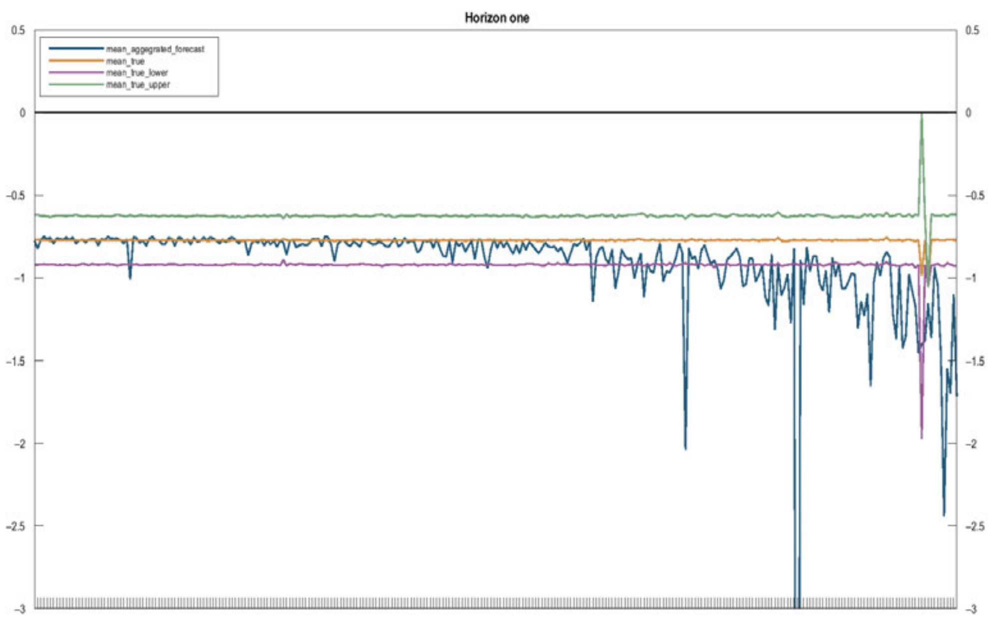

Figure 14: Monte carlo experiment at horizon 1. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model

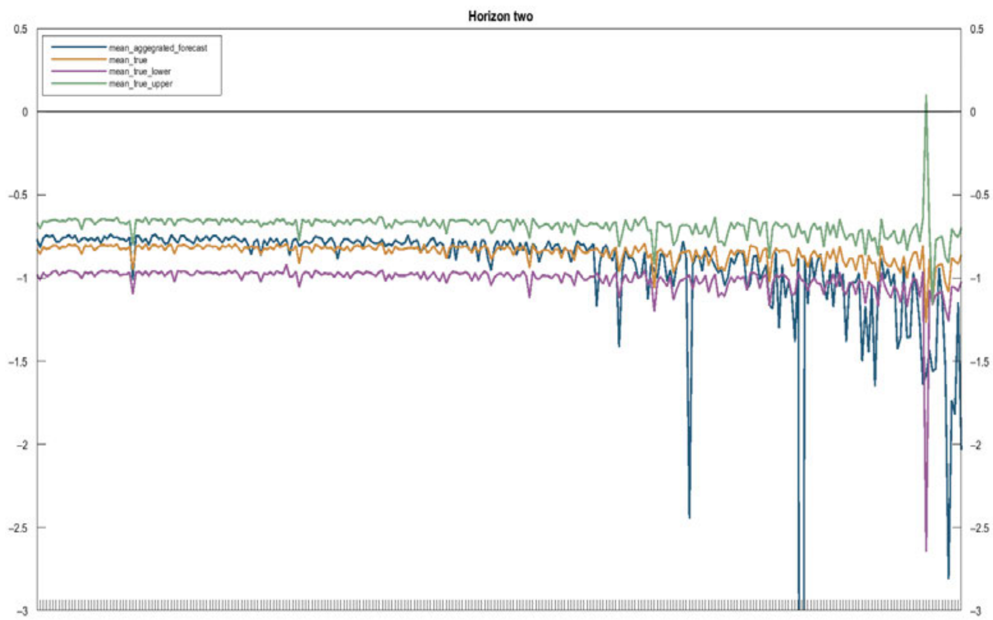

Figure 15: Monte carlo experiment at horizon 2. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model

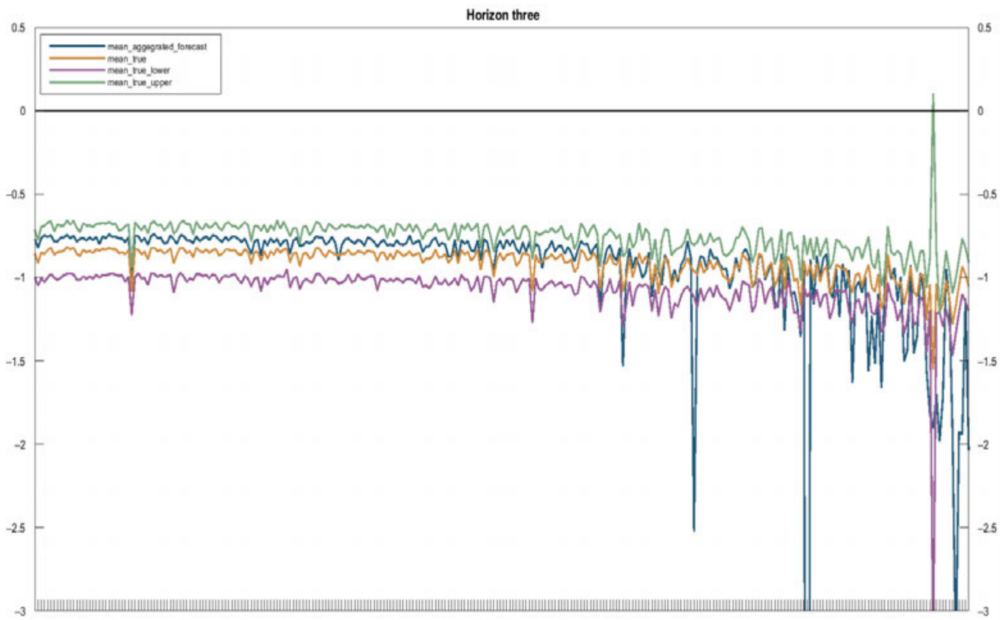

Figure 16: Monte carlo experiment at horizon 3. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model

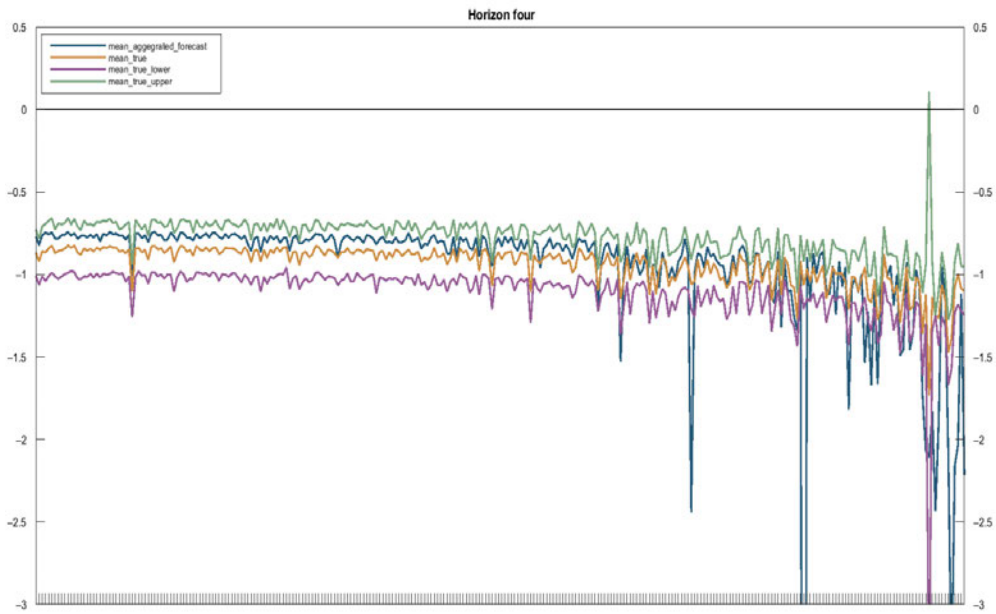

Figure 17: Monte carlo experiment at horizon 4. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model

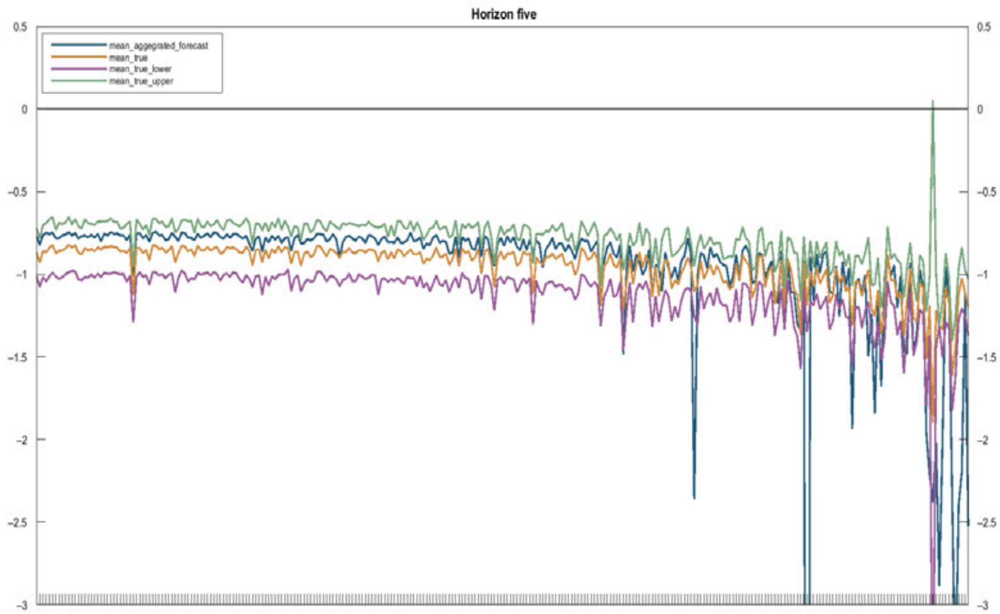

Figure 18: Monte carlo experiment at horizon 5. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model

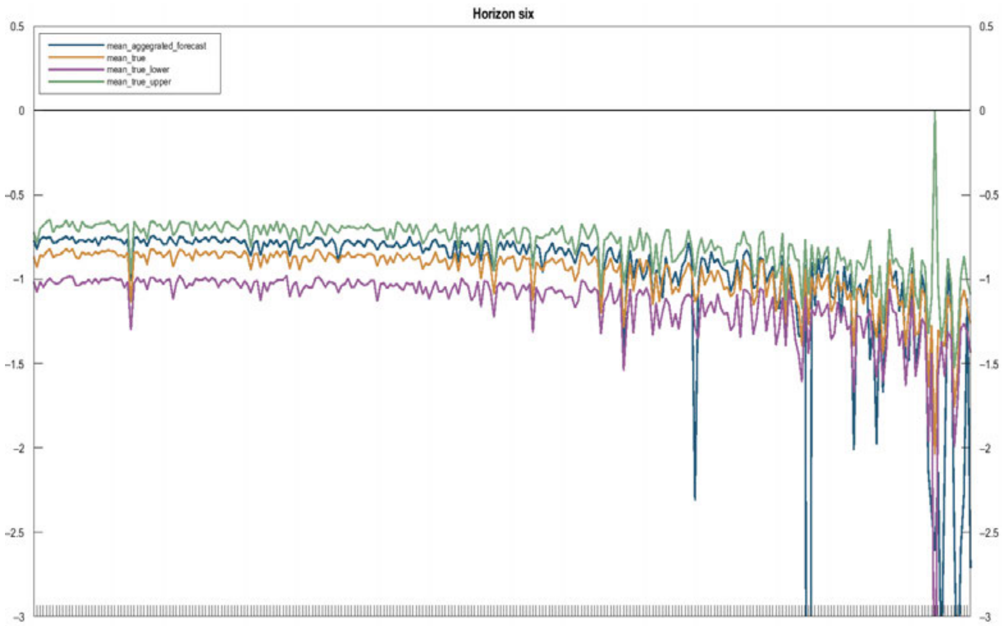

Figure 19: Monte carlo experiment at horizon 6. mean_aggregated_forecast is the mean of the mean log scores for the copula model, mean_true is the mean of the mean log scores for the true VAR model (estimated by OLS), mean_true_lower is mean_true minus 1.96 times the standard deviation of the mean log scores of the true VAR model, and mean_true_upper is mean_true plus 1.96 times the standard deviation of the mean log scores of the true VAR model