National Research University
Higher School of Economics
Faculty of Economic Sciences

Term paper

Using MIDAS Models on Russian Data

Author:
Polina Detkova,
BEC132

Supervisor:
Boris Demeshev

Moscow
2016

Contents

1 Introduction

Operation of many entities such as central banks, governments, firms, etc. depends of current values of various macroeconomic variables and their forecasts, and therefore forecasting methods are in constant development. However, all new methods have to prove their efficiency before being used in practice. This term work considers MIDAS regression analysis, a relatively new approach to the usage of high frequency data in predicting low frequency variables.

The main reason behind using higher frequency data for macroeconomic forecasting is actual dependence of some low frequency macroeconomic variables on data collected more frequently and that is why such data might be a good predictor. There are studies proving considerable implication of, for example, oil prices or exchange rates of national currency on countries' macroeconomic variables, including Russian (see, for example, Rautava 2004). Furthermore, late collection of most macroeconomic data makes this data unavailable within both current period and a considerable part of the next, while it is obvious that the higher the frequency of collection, the more up-to-date data is available for now, so that is also a part of motivation for high frequency data usage.

There is an obvious problem risen by implementing high frequency data: as this data has more observations, there are usually more coefficients to be estimated rather than after including a low frequency regressor. Sometimes this can result in shortage of observations of predicted variable to estimate the coefficients. That is especially important to Russian data, because comparatively late beginning of macroeconomic data collection; thus, it was only 1994 when information on quarterly GDP becomes available. A trade-off occurs: on the one hand, the more information is used, the higher the accuracy, on the other hand, the overfitting problem can make forecasting quite inaccurate, let alone the number of observations insufficiency for mere estimation of the coefficients.

MIDAS approach is only one of means of partially resolving the problem. It has been compared with other methods since its first introduction in 2002 (early publication of Ghysels, Santa-Clara, and Valkanov 2004) mostly by testing it on various (financial, macroeconomics, etc.) datasets, and this way of estimating relative efficiency is still found to be the best.

In this paper MIDAS is compared with two models which use only dependent series lags ($AR(p)$ and $ARIMA(p, d, q)$) and with two models using aggregates high frequency variables, so-called time-averaging and step-weighting. The results show the great dependence of effectiveness on data set.

The paper is structured as follows. Section 2 describes theoretical framework of MIDAS regression models. Section 3 contains literature review. Section 4 gives the models used in empirical study. Section 5 describes the data. Section 6 presents empirical results and Section

7 concludes.

2 Theory

The main idea behind using MIDAS models is an assumption about functional dependence between high frequency lags while predicting low frequency series (using similar approach on the latter is optional). Suppose there is a variable y_t observed, for example, quarterly[1]. Another variable, $x_t^{(m)}$, is available m times more frequently, say monthly ($m = 3$). The simplest MIDAS model is

$$y_t = \beta_0 + \beta_1 B\left(L^{\frac{1}{m}};\theta\right) x_t^{(m)} + \varepsilon_t^{(m)} \tag{1}$$

where $B\left(L^{\frac{1}{m}};\theta\right) = \sum_{k=0}^{K} b(k,\theta)L^{\frac{k}{m}}$, $L^{\frac{1}{m}}$ is high frequency lag operator such that $L^{\frac{k}{m}}x_t^m = x_{t-\frac{k}{m}}^m$, K is number of high frequency lags used for forecast and $b(k,\theta)$ is polynomial, in which k is number of lag and θ is a vector of parameters. Thus, if low frequency data is collected annually, $m = 12$ and $K = 18$, then high frequency data is available on monthly basis and a year and a half of the data (15 observations) is used for estimation.

Possible polynomial specifications are worth considering first. In conformity with the goal of their usage, function forms should depend on very few parameters and be rather flexible at the same time so as to make results as independent of chosen specification as possible. A comment on polynomial specifications is briefly given in literature review. Two popular approaches, used in this paper the following:

- The exponential Almon lag (number of parameters Q is usually chosen to be 2):

$$b(k;\theta) = \frac{exp\left(\theta_1 k + \cdots + \theta_Q k^Q\right)}{\sum_{j=0}^{K} exp\left(\theta_1 j + \cdots + \theta_Q j^Q\right)} \tag{2}$$

- The beta lag with two parameters ($\theta = (\theta_1, \theta_2)^T$):

$$\begin{aligned} b(k;\theta_1,\theta_2) &= \frac{f\left(\frac{k}{m},\theta_1,\theta_2\right)}{\sum\limits_{j=1}^{m} f\left(\frac{j}{m},\theta_1,\theta_2\right)} \\ f\left(i,\theta_1,\theta_2\right) &= \frac{i^{\theta_1-1}(1-i)^{\theta_2-1}\Gamma(\theta_1+\theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)} \end{aligned} \tag{3}$$

Both the exponential Almon and the beta lag specifications provide positive coefficients, which might be important for volatility estimation.

[1] Theoretical framework is given on the basis of Ghysels, Sinko, and Valkanov 2007 and Clements and Galvo 2009.

There was no autoregressive part in initially proposed MIDAS regression model (see Ghysels, Santa-Clara, and Valkanov 2004), but it has been commonly included in follow-up works. As shown in, for example, Ghysels, Sinko, and Valkanov 2007 on p. 60, there are at least two ways to implement autoregressive augmentation, and here it is introduced in simple form (similar to the one in Armesto, Engemann, and Owyang 2010):

$$y_t = \beta_0 + \sum_{d=1}^{D} \beta_{1d} L^d y_t + \beta_2 B\left(L^{\frac{1}{m}}; \theta\right) x_t + \varepsilon_t \tag{4}$$

As can be seen, there is no functional dependence between lags of low frequency variable or other constrains yet, so h-steps-ahead forecast of quarterly distributed low frequency variable using monthly distributed high frequency variable is given by the equation:

$$y_t = \beta_0 + \sum_{d=1}^{D} \beta_{1d} L^d y_{t-h} + \beta_2 B\left(L^{h+\frac{1}{m}}; \theta\right) x_t + \varepsilon_t, \tag{5}$$

where $B\left(L^{h+\frac{1}{m}}; \theta\right) = \sum_{k=0}^{K} b(k, \theta) L^{h+\frac{k}{m}}$. In this paper functional constrains on predicted time series are also imposed, and therefore for such cases h-steps-ahead forecast is:

$$y_t = \beta_0 + \sum_{d=1}^{D} a(d, \alpha) L^d y_{t-h} + \beta_2 B\left(L^{h+\frac{1}{m}}; \theta\right) x_t + \varepsilon_t, \tag{6}$$

where a and α are a polynomial and a vector of parameters respectively, similar to b and θ.

3 Literature review

Literature review includes both the description of ideas of papers considering general MIDAS regression models characteristics and the ones focused on comparison of various approaches, including MIDAS, for macroeconomic forecasting.

Ghysels, Santa-Clara, and Valkanov 2004 is the initial work on MIDAS approach. Before this work most common approaches were either aggregating all series to the least frequency (for instance, by time-averaging approach) or disaggregating to the highest frequency sampling (for example, VAR). The analysis of asymptotic properties shows relative efficiency of MIDAS approach in comparison with the typical approaches. The authors also point to possible applications in macroeconomics and finance.

Ghysels, Sinko, and Valkanov 2007 is highly dedicated to answering researches questions about important aspects of the approach and also among first papers introducing autoregressive part. The articles considers possible polynomial specifications for MIDAS models: finite polynomials, infinite polynomials and step functions. The former are usually exponential Almon polynomial and beta polynomial (see in theory). Using infinite polynomials mostly implies including autoregressive augmentation. MIDAS regressions with step functions embrace partial sums of high frequency lags. Volatility forecasting, conducted in the article, provides results of MIDAS regressions' overcoming traditional noise-corrected schemes.

In Armesto, Engemann, and Owyang 2010 three forecasting approaches are compared: time-averaging, step-weighting and exponential Almon polynomial MIDAS (see the former two described among other methods MIDAS approach is compared with in this paper). It was one of the goals to compare accuracy due to different high and low frequency ratios: daily interest rates were used in predicting monthly inflation, monthly industrial production growth and monthly employment growth, while monthly employment growth rates were predictors for quarterly GDP growth (four models altogether, low frequency data lags also used). The authors used RMSE criteria for comparison. The results were found for both end-of-period and intra-period forecasting; the latter one is including renewals of available high frequency data within on low frequency period, while the former implies usual change of all available information only with the beginning of another low frequency period. As the authors show, in end-of-period forecasting, both at the shortest and at longer horizons, the models are materially equivalent[2]. Intra-period forecasting shows the benefit of adding intra-period information and similar results.

Kuzin, Marcellino, and Schumacher 2011 compares MIDAS regression models and mixed-frequency VAR models. The latter one uses specific aggregation schemes for implementing VAR

[2] Except for predicting GDP, in which MIDAS is better than time averaging; in addition, MIDAS comparative efficiency increases with using the term spread as predictor instead of the federal funds rate.

models on high frequency. The major difference between two approaches is the absence of any functional restriction in MF-VAR. MF-VAR also explains indicators as well as the dependent variable and more likely suffers from overfitting. The authors consider trying to compare models effectiveness within theoretical framework quite useless and therefore they test the models while forecasting quarterly euro area GDP growth using a set of monthly indicators[3]. The result of the paper is better performance of MF-VAR for longer horizons and better performance of MIDAS for shorter ones.

Andreou, Ghysels, and Kourtellos 2013 considers the question of the usage of daily data in macroeconomic forecasting. The empirical study includes building models on comparatively huge number of only daily regressors (991 daily time series, 64 available for the whole sample), let alone monthly ones. The principal component approach allows to reduce the number of daily explanatory series to five[4]; the authors also use forecast combinations so as to improve accuracy (see the whole procedure in Andreou, Ghysels, and Kourtellos 2013, pp. 243–245). The major findings are improvement of quarterly forecasts of US real GDP growth with MIDAS regression models using high frequency financial information comparatively to usage of only quarterly distributed time series

Foroni and Marcellino 2014 compares approaches while forecasting macroeconomic parameters on Euro area data. MIDAS models with autoregressive augmentation are showing good results, especially on shorter horizons. Bridge equations show good results as well, while mixed-frequency VAR are the least promising.

MIDAS models are compared with bridge equations in Schumacher 2016 while nowcasting Euro area quarterly GDP. The author introduces iterative MIDAS approach which outperforms bridge equations for the most part of the predictors (as well as traditional MIDAS models with autoregressive augmentation).

[3] sec:data of monthly indicators for the empirical part of this coursework is highly determined by this article's approach.

[4] The number was determined by informational criteria, applied to marginal contribution of adding another principal component in explaining the variation.

4 Empirical study. Methods

4.1 MIDAS models

The main purpose of empirical study is finding out whether using MIDAS regression models may improve traditional approaches' results. MSE (Mean Square Error) out of sample is chosen as the comparison criteria:

$$MSE = \frac{\sum\limits_{t=t_0+1}^{T} e_t^2}{T - t_0}$$

where $1, \dots, t_0$ are observations of train sample and $t_0+1, \dots, T$ are observations of test sample. As also mentioned in sec:data, there are 69 observations in train and 12 observations in test.

Four MIDAS polynomial specifications are used in the analysis (*nealmon* and *nbeta* are normalized exponential Almon and normalized beta lag respectively, named after functions in *midasr* package):

Table 1: Polynomial specifications

	Low frequency lags	High frequency lags
1	no restriction	nealmon
2	no restriction	nbeta
3	nealmon	nealmon
4	nealmon	nbeta

As there are 2 optional low frequency lag orders and 2 high frequency lag orders, on the whole there are 16 various MIDAS models.

Unfortunately, there is a gap in model estimation due to the of optimisation algorithm of MIDAS regression in *midasr* package, so there is restricted number of estimated MIDAS model for some function schemes:

- **GDP index** 1 model with nealmon restrictions on both own lags and high frequency lags (using 4 and 3 lags respectively, predicting current quarter GDP index from second month)
- **export**: 1 model with no restriction on equity investment lags and nealmon restriction on high frequency variables (using 4 and 6 lags respectively, predicting current quarter GDP index from third month)
- **equity investment index** 5 models with no restriction on equity investment lags and nealmon restriction on high frequency variables

4.2 Other forecasting approaches

MIDAS regression models are compared with simple time averaging, step weighting and VAR models.

- **Time-averaging**
  The approach implies finding aggregation of high frequency data to "quarterly" data taking simple average of for the period (see Armesto, Engemann, and Owyang 2010):

$$\begin{aligned} \overline{x_t} &= \frac{1}{m}\sum_{j=1}^{m} L^{\frac{j}{m}} x_t \\ y_t &= \beta_0 + \sum_{d=1}^{D}\beta_{1d}L^d y_{t-h} + \sum_{i=1}^{K/m}\gamma_i L^i \overline{x_t} + \varepsilon_t \end{aligned} \tag{7}$$

  Here K/m should be a positive integer.

- **Step-weighting** Step weighting usually implies different coefficients for all high frequency lags and therefore is unrestricted linear model:

$$y_t = \beta_0 + \sum_{d=1}^{D}\beta_{1d}L^d y_{t-h} + \sum_{i=1}^{K}\gamma_i L^{i/m} x_t + \varepsilon_t \tag{8}$$

- **Low frequency models** These models do not require high frequency data and therefore forecasting process in them is standard for time series. As indicated above, there is a chance of a delay in low frequency data becoming available so there might be various procedures for some forecasts:

  1. *one-step forecasts* are made from all months for previous quarter and for current quarter from second and third months
  2. *two-step forecasts* two-step forecast is made for current quarter from all months and next quarter from second and third months
  3. *three-step forecasts* are made from all months for next quarter

  Two models are considered: simple $AR(p)$ process and $ARIMA(p,d,q)$[5] process (see Hyndman and Athanasopoulos 2013):

  1. $AR(p)$: the low frequency series depends only on its own lags:

$$(1-\phi_1 L - \cdots - \phi_p L^p)y_t = \phi_0 + \varepsilon_t \tag{9}$$

[5] While estimating $ARIMA$ parameters the MSE criteria criteria is going to be violated as the process without standard functions goes with difficulty. The best model will be defined by AIC criteria (see Hyndman and Khandakar 2008); however, the restriction on maximum autoregressive lags is held.

2. $ARIMA(p,d,q)$: there are lagged values of low frequency time series and its lagged errors in the model:

$$(1-\phi_1 L - \cdots - \phi_p L^p)(1-L)^d y_t = \phi_0 + (1+\theta_1 L + \cdots + \theta_q L^q)\varepsilon_t \tag{10}$$

Taking seasonal adjustment into account (quarter in this case):

$$ARIMA(p,d,q)(P,D,Q)_4 : \tag{11}$$

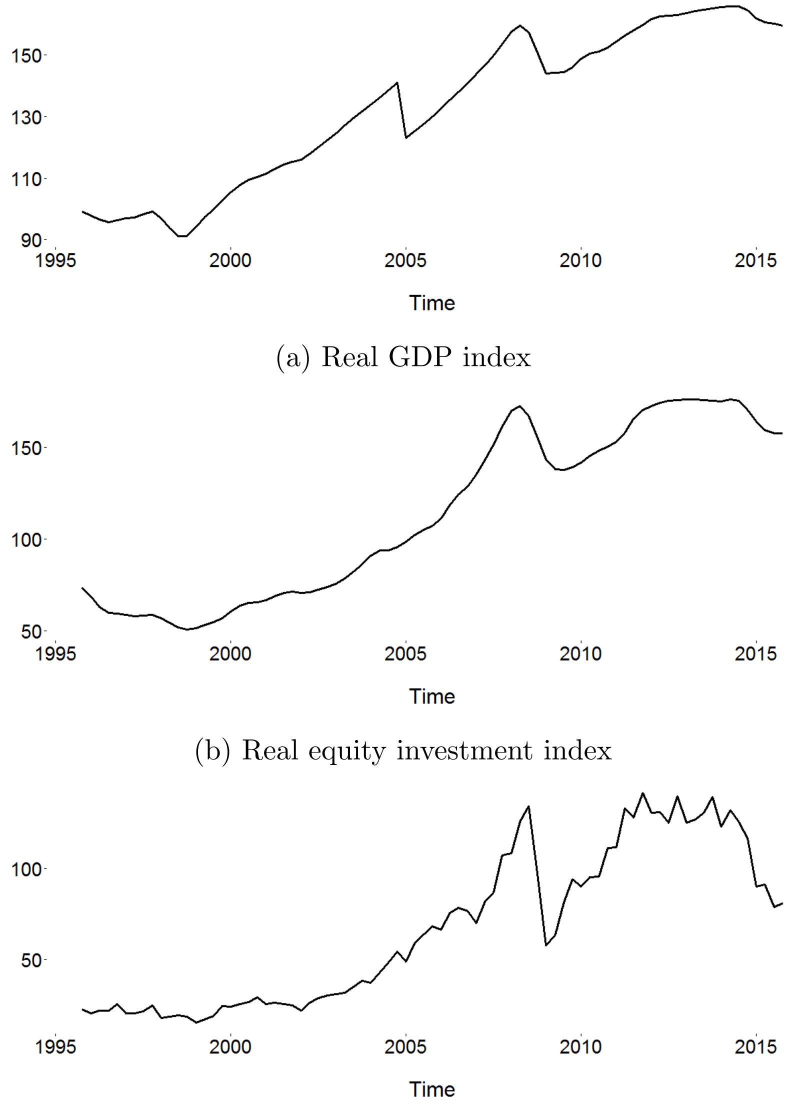
$$(1-\cdots-\phi_p L^p)(1-L)^d(1-\cdots-\Phi_p L^{4P})(1-L^4)^D y_t = \tag{12}$$

$$\phi_0 + (1+\cdots+\theta_q L^q)(1+\cdots+\Theta_Q L^{4Q})\varepsilon_t \tag{13}$$

Due to late collection of the most macroeconomic data there is a considerable time gap between end of a period and the moment its data becomes available. Considering quarterly and monthly data, the former usually is available starting from the second month of the next quarter and the latter starting from the month after the next. As there are three months in quarter, there are 9 forecasting situations in this paper: there are previous, current and next quarter low frequency data to be foretasted on the basis of information available on first, second and third month of the current quarter (for example, first month can be January, April, July or October). During the first month the last information available is quarter before previous and second month of previous quarter. During the next two months information on previous quarter usually becomes available, but due to possible delay predicting it might be also useful (for instance, the information on equity investment of the last quarter of 2015 was unavailable till April 2016).

5 Empirical study. Data

This empirical study concerns forecasting quarterly distributed variables using monthly distributed variables. Two low frequency variables were chosen: quarterly real GDP index, real equity investment index, both seasonally corrected, and quarterly export[6]. As can be seen from the graphs, the shape of GDP and investment indexes less dynamic than the export's and may have connection with the results.



(a) Real GDP index

(b) Real equity investment index

(c) Export

Figure 1: Quarterly distributed variables

[6] There is little practical need in forecasting quarterly export and real equity investment without taking its monthly lags in consideration, forecasting prospects using other monthly distributed variables are still interesting though.

High frequency variables for both forecasts (units are bracketed):

- Unemployment rate (%)
- Growth of the consumer price index (%)
- Average exchange rate of the dollar on MICEX Stock Exchange (roubles per dollar)
- The average RTS index (points)
- BCI[7] (points)
- Brent Crude Futures, Continuous Contract average price [8] (dollars per barrel)

Data sources: sophist.hse.ru, OECD, Quandl.com.

Data contains information from last quarter of 1995 till last quarter of 2015 (81 observations for the quarterly data and 243 for the monthly data).

[7] Business confidence index, according to OECD, "..'is based on enterprises' assessment of production, orders and stocks, as well as its current position and expectations for the immediate future; opinions compared to a "normal" state are collected and the difference between positive and negative answers provides a qualitative index on economic conditions".

[8] Quandl.com.

6 Empirical study. Results

After estimating the models, making forecasts and counting MSE all models were ranged according to MSE value. The table contains the best results of forecasting last quarter, this table describing the results for the current and the last for the next. As can be seen from the tables, time-averaging and step weighting are bad at forecasting comparing to $MIDAS$, AR and $ARIMA$. $ARIMA$ ans AR behave much better in forecasting GDP and investment, perhaps because of less dynamic changes.

Table 2: Forecasting previous quarter results

	Month:	1st		2nd		3rd	
		Model	MSE	Model	MSE	Model	MSE
	1	ARIMA	0.6663703	ARIMA	0.6663703	ARIMA	0.6663703
	2	ARIMA	1.8983943	ARIMA	1.8983943	ARIMA	1.8983943
GDP	3	MIDAS	5.023463	MIDAS	3.674422	MIDAS	3.700275
	4	AR	5.777755	MIDAS	4.799596	MIDAS	3.789321
	5	MIDAS	6.442349	MIDAS	5.439036	MIDAS	4.157034
	1	ARIMA	1.435849	ARIMA	1.435849	MIDAS	1.238367
	2	ARIMA	1.435849	ARIMA	1.435849	ARIMA	1.435849
Investment	3	MIDAS	1.693464	MIDAS	1.523517	ARIMA	1.435849
	4	MIDAS	1.966287	MIDAS	2.407763	MIDAS	1.579023
	5	MIDAS	2.563555	MIDAS	2.500556	MIDAS	1.734816
	1	ARIMA	95.73944	ARIMA	95.73944	MIDAS	26.93640
	2	ARIMA	149.96449	MIDAS	97.08501	MIDAS	39.65073
Export	3	ARIMA	157.78167	MIDAS	100.80510	MIDAS	67.27383
	4	ARIMA	157.78167	MIDAS	102.38295	MIDAS	68.84745
	5	MIDAS	189.0912	MIDAS	106.01549	ARIMA	95.73944

7 Conclusion

According to the previous research of MIDAS regression models, they have potential of improving forecasting of low frequency variables using high frequency variables. However, the results of MIDAS comparison to other approaches highly depends on the structure of data, end therefore all possible improvements are worth verifying empirically.

Applying MIDAS to Russian macroeconomic data has a benefit of estimating relatively less parameters than while using other effective methods, especially because of limited nature (regarding time) of the data. The results show that, on average, MIDAS overperforms other models in forecasting current and next quarter values, especially while forecasting export. Factually, MIDAS appeared to be the best model for export among the ones taken into consideration.

Table 3: Forecasting current quarter results

	Month:	1st		2nd		3rd	
		Model	MSE	Model	MSE	Model	MSE
GDP	1	ARIMA	0.6663703	ARIMA	0.6663703	ARIMA	0.6663703
	2	ARIMA	0.6663703	ARIMA	0.6663703	ARIMA	0.6663703
	3	ARIMA	1.8983943	ARIMA	1.8983943	ARIMA	1.8983943
	4	ARIMA	2.7825685	ARIMA	2.7825685	ARIMA	2.7825685
	5	AR	5.7777554	AR	5.7777554	AR	5.7777554
Investment	1	ARIMA	1.435849	ARIMA	1.435849	ARIMA	1.435849
	2	ARIMA	1.435849	ARIMA	1.435849	ARIMA	1.435849
	3	AR	5.473756	MIDAS	3.859356	AR	5.473756
	4	AR	5.473756	MIDAS	4.747741	AR	5.473756
	5	MIDAS	6.055852	MIDAS	7.827726	MIDAS	5.590090
Export	1	ARIMA	95.73944	ARIMA	95.73944	MIDAS	36.50947
	2	ARIMA	149.96449	MIDAS	95.12465	MIDAS	49.03189
	3	ARIMA	157.78167	MIDAS	96.56758	MIDAS	59.66968
	4	MIDAS	182.2070	MIDAS	96.73320	MIDAS	61.33924
	5	MIDAS	182.2070	MIDAS	104.09723	MIDAS	66.18667

Table 4: Forecasting next quarter results

	Month:	1st		2nd		3rd	
		Model	MSE	Model	MSE	Model	MSE
GDP	1	ARIMA	1.8983943	ARIMA	1.8983943	ARIMA	1.8983943
	2	ARIMA	2.7825685	ARIMA	2.7825685	ARIMA	2.7825685
	3	AR	7.3400541	AR	7.3400541	AR	7.3400541
	4	MIDAS	15.93931	MIDAS	11.69271	MIDAS	10.00246
	5	AR	16.1366239	MIDAS	11.75925	MIDAS	13.19644
Investment	1	MIDAS	8.837639	MIDAS	8.480522	MIDAS	6.969631
	2	MIDAS	8.949318	MIDAS	8.501347	MIDAS	7.683740
	3	MIDAS	10.011066	MIDAS	9.115095	MIDAS	8.863921
	4	MIDAS	10.095088	MIDAS	10.154123	MIDAS	9.384726
	5	MIDAS	10.785841	MIDAS	101.1368	MIDAS	11.606537
Export	1	ARIMA	157.78167	ARIMA	101.6951	MIDAS	45.17329
	2	MIDAS	184.5563	MIDAS	102.1478	MIDAS	63.23262
	3	MIDAS	203.4297	MIDAS	106.8934	MIDAS	64.95925
	4	MIDAS	216.8116	MIDAS	107.2429	MIDAS	68.21621
	5	ARIMA	223.34631	MIDAS	107.8206	MIDAS	99.15292

Regarding future work, there are still unresolved issues considering both theory of MIDAS approach and practical applications. It is still interesting to compare MIDAS to more complicated approaches than are used in this paper (MF-VAR, bridge equations, etc.). Thinking globally, as it still actual testing models on dataset which define which model performs best for forecasting, it appears to be challenging to identify some patterns. Forecasting using MIDAS in practice is highly lightened by *midasr* package in R and MIDAS Matlab Toolbox in Matlab. There are still some difficulties in estimation of models, at least using *midasr*, because of errors occurring due to wrong start values of optimizations while it has been given little consideration to their proper selection yet. Developing an explicit algorithm appears to provide a powerful tool for researchers.

References

Andreou, E., E. Ghysels, and A. Kourtellos (2013). “Should macroeconomic forecasters use daily financial data and how?” In: *Journal of Business and Economic Statistics* 31.2, pp. 240–251 (cit. on p. 6).

Armesto, M. T., K. M. Engemann, and M. T. Owyang (2010). “Forecasting with mixed frequencies”. In: *Federal Reserve Bank of St. Louis Review* 92.6, pp. 521–536 (cit. on pp. 4, 5, 8, 16).

Clements, M. P. and A. B. Galvo (2009). “Macroeconomic Forecasting with Mixed Frequency Data: Forecasting US output growth”. In: *Journal of Applied Econometrics* 24.7, pp. 1187–1206 (cit. on p. 3).

Foroni, C. and M. Marcellino (2014). “A comparison of mixed frequency approaches for nowcasting Euro area macroeconomic aggregates”. In: *International Journal of Forecasting* 30, pp. 554–568 (cit. on p. 6).

Ghysels, E., P. Santa-Clara, and R. Valkanov (2004). “The MIDAS touch: mixed data sampling regression models”. In: *Mimeo, Chapel Hill, NC.* (Cit. on pp. 2, 4, 5).

Ghysels, E., A. Sinko, and R. Valkanov (2007). “MIDAS regressions: Further results and new directions”. In: *Econometric Reviews* 26.1, pp. 53–90 (cit. on pp. 3–5).

Hyndman, R. and G. Athanasopoulos (2013). *Forecasting: principles and practice* (cit. on p. 8).

Hyndman, R. and Y. Khandakar (2008). “Automatic Time Series Forecasting: The forecast Package for R”. In: *Journal of Statistical Software* 27.3 (cit. on p. 8).

Kuzin, V., M. Marcellino, and C. Schumacher (2011). “MIDAS vs. mixed-frequency VAR: Nowcasting GDP in the Euro Area”. In: *International Journal of Forecasting* 27.2, pp. 529–542 (cit. on p. 5).

Rautava, J. (2004). “The role of oil prices and the real exchange rate in Russia’s economy–a cointegration approach”. In: *Journal of Comparative Economics* 32.2, pp. 315–327 (cit. on p. 2).

Schumacher, C. (2016). “A comparison of MIDAS and bridge equations”. In: *International Journal of Forecasting* 32, pp. 257–270 (cit. on p. 6).

8 Applications

8.1 Midasr functions in use

Armesto, Engemann, and Owyang 2010 MIDAS user guide is rather comprehensive, although a beginner might need a simple example of the usage of main functions such as provided in this application.

Russian quarterly GDP is used as low frequency variable and month average oil prices as high frequency:

```
gdp_raw <- read.csv("lf_data.csv")
oil_raw <- read.csv("hf_data199509.csv") %>% select(time, oil)
```

There are 84 observations in *gdp* and 246 observations in *oil*, so it would be useful to hold only observations of years 1996–2015 (80 and 240 observations left respectively).

```
gdp <- gdp_raw %>% slice(-c(1:4))
oil <- oil_raw %>% slice(-c(1:4, 245, 246))
nrow(gdp); nrow(oil)

## [1] 80
## [1] 240

raw_gdp <- gdp$GDP_Q_DIRI_SA
raw_oil <- oil$oil
trend <- 1:length(raw_gdp)
```

8.1.1 Data handling

Some functions of the package can be successfully applied beyond the bounds of MIDAS models. For example, for producing table of lags (which normally can be conducted with $lag()$ function) $mls()$ or $fmls()$ functions might be useful (p. 13):

```
raw_oil %>% mls(k = 0:5, m = 3) %>% tail(3)

##          X.0/m    X.1/m    X.2/m    X.3/m    X.4/m    X.5/m
## [78,] 63.75273 65.60857 61.13571 56.93864 58.79500 49.75810
## [79,] 48.53955 48.20571 56.76435 63.75273 65.60857 61.13571
## [80,] 38.90409 45.93238 49.29273 48.53955 48.20571 56.76435
```

```
tail(raw_oil)

## [1] 56.76435 48.20571 48.53955 49.29273 45.93238 38.90409
```

Argument k is for defining orders of lags in use and m is for frequency ratio (in this particular example $length(k) = 6$ lags of oil prices, observed $m = 3$ times more frequently, can be used for simple linear of other regressions). Actually, here $flms(raw\_oil, k = 5, m = 3)$ could be used as lags start from 0.

Forecasting usually implies the separation of test sample, but there are functions in *midasr* with no essential preliminary division.

8.1.2 Unrestricted models

Suppose 4 low frequency and 6 high frequency lags are used in forecast with simple linear model ($m = 3$):

$$gdp_t = \beta_0 + \sum_{d=1}^{5} \beta_{1d} gdp_{t-d} + \sum_{s=0}^{6} \beta_{2s} oil_{t-s/m} \tag{14}$$

Either simple $lm()$ function with usage of $mls()$ or $fmls()$ can be used of $midas_ur()$

```
unr_model <- midas_r(raw_gdp ~ trend + mls(raw_gdp, 1:4, 1) +
                       fmls(raw_oil, 5, 3), start = NULL)
```

Forecasting in this form (using $forecast()$ function) is rather complicated due to parameter choosing process as NAs are produced while using $mls()$ function. Now we use the constriction of normalized[9] exponential Almon lag polynomials on high frequency data:

```
nealmon_model <- midas_r(raw_gdp ~ trend + mls(raw_gdp, 1:4, 1) +
                           fmls(raw_oil, 5, 3, nealmon),
                         start = list(raw_oil = c(1, 1)))
```

Forecasting and comparison of forecasts is rather simple with $average_forecast()$ function: there is no need in previous definition of train and test samples, any conducted estimation of compared models is enough:

[9] The normalization constraint is coefficients summing up to unity.

```
comparison <- average_forecast(modlist = list(unr_model, nealmon_model),
                               data = list(trend = trend, raw_gdp = raw_gdp,
                                           raw_oil = raw_oil),
                               insample = 1:72, outsample = 73:80,
                               type = "rolling", measures = c("MSE", "MAPE", "MASE")
                               show_progress = FALSE)
```

As explained in package description on CRAN, parameter type defines the following: the ”fixed” forecast uses model estimated with insample data, the ”rolling” forecast reestimates model each time by increasing the in-sample by one low frequency observation and dropping the first low frequency observation and the ”recursive” forecast differs from ”rolling” that it does not drop observations from the beginning of data. Two models can be compared due to their out-of-sample accuracy measures:

```
comparison$accuracy$individual %>% select(2:4) %>% xtable
```

	MSE.out.of.sample	MAPE.out.of.sample	MASE.out.of.sample
1	5.20	1.18	2.07
2	1.76	0.74	1.30

8.1.3 Model selection

There are two functions in *midasr* for choosing both functional form and lag orders. *midas_r_ic_table*() returns information on all models so as to help a researcher make the choice according to his own criteria and *select_and_forecast*() contains almost all possible procedures which might be needed. Although choosing one of the implied information criteria might simplify the process, an analysis beyond MIDAS approach and therefore implementing original *midasr AIC* of *BIC* criteria beyond original functions can be quite challenging (see GitHub repository of Vaidotas Zemlys for factual form of AIC and BIC criteria used in the package). For *midas_r_ic_table*() function *expand_weights_lags*() should be used first[10]:

```
set_oil <- expand_weights_lags(weights = c("nealmon", "nbeta"),
                               from = 0, to = c(2), m = 3,
                               start = list(nealmon = c(1, -1),
```

[10] Function *expand_weights_lags*() does not let estimate unrestricted models so henceforth normalized exponential Almon end normalized beta lag are compared.

```
                                            nbeta = c(1, 0, 1)))
set_gdp <- expand_weights_lags(weights = c("nealmon"),
                               from = 0, to = c(2), m = 3,
                               start = list(nealmon = c(1, -1)))
```

```
table_r <- midas_r_ic_table(raw_gdp ~ trend + mls(raw_gdp, 0, 1) +
                           mls(raw_oil, 0, 3),
                 table = list(raw_gdp = set_gdp,
                              raw_oil = set_oil))
```

```
selection <- select_and_forecast(raw_gdp ~ trend + mls(raw_gdp, 0, 1) +
                            mls(raw_oil, 0, 3),
         from = list(raw_gdp = c(4, 8),
         raw_oil = c(12, 24)),
         to = list(raw_gdp = rbind(c(14, 19), c(18, 23)),
                   raw_oil = rbind(c(22, 27), c(34, 39))),
         insample = 1:72, outsample = 73:80,
         weights = list(raw_gdp = c("nealmon", "almonp"),
                        raw_oil = c("nealmon", "almonp")),
         wstart = list(nealmon = rep(1, 3), almonp = rep(1, 3)))
```

Function $average\_forecast()$ is recommended to evaluate the forecasting performance (thus one can follow his or her own criteria of comparison).

8.2 Code example

This subsection provides code for estimating models for export. The order is:

1. MIDAS models

2. time-averaging

3. step-weighting

4. AR and ARIMA

Export data is downloaded from sophist.hse.ru directly; high frequency time series data set is provided on github.

Possible low frequency lags are 4 and 6, possible high frequency lags are 3 and 6.

```
lf_data <- sophisthse("EX_Q") %>% tail(84) %>% as.data.frame()
hf_data <- read.csv("hf_data.csv")
hf <- hf_data %>% head(-2) %>% tail(-1); lf <- (lf_data %>% tail(-3))
ex <- lf$EX_T_Q; unemp <- hf$unemp; cpi <- hf$cpi; dol <- hf$dol
rts <- hf$rts; bci <- hf$bci; oil <- hf$oil

# models parameters and list of start values
ctype1 <- c("NULL", "nealmon"); ctype2 <- c("nealmon", "nbeta")
l_lag <- c(4, 8); h_lag <- c(3, 6); months <- 1:3; quat <- 1:3; MSE <- 0
midas_table <- expand.grid(
  ctype1 = ctype1, ctype2 = ctype2,
  l_lag = l_lag, h_lag = h_lag, months = months,
  quat = quat, MSE = 0
)
start <- list("nealmon" = c(c(1/2, -1/2), c(1/3, -1/2)),
              "nbeta" = c(c(0, 1, 0), c(10, 1, -1), c(1, 1, 1)))
knealmon <- (start[["nealmon"]] %>% length()) / 2
knbeta <- (start[["nbeta"]] %>% length()) / 3

# function
midas_func <- function(m_table, row_num){
  if (type1 == "nealmon") {
    if (type2 == "nealmon") {
      for (i in 1:knealmon) {
model <- try(midas_r(as.formula(paste(
      "ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1, ", 1, nealmon) +
                       mls(unemp, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, ", neal
                              mls(cpi, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(dol, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(rts, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(bci, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(oil, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                  start = list(ex = start[["nealmon"]][i:(i+1)],
                               unemp = start[["nealmon"]][i:(i+1)],
                               cpi = start[["nealmon"]][i:(i+1)],
                               dol = start[["nealmon"]][i:(i+1)],
                               rts = start[["nealmon"]][i:(i+1)],
                               bci = start[["nealmon"]][i:(i+1)],
                               oil = start[["nealmon"]][i:(i+1)])), silent = T)
        if (is.null(model) == FALSE) {break()}
      }
    } else if (type2 == "nbeta") {
      for (i in 1:knealmon) {
        for (j in 1:knbeta) {
model <- try(midas_r(as.formula(paste(
      "ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1, ", 1, nealmon) +
                     mls(unemp, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, ", nbeta
                              mls(cpi, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(dol, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(rts, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(bci, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(oil, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                   start = list(ex = start[["nealmon"]][i:(i+1)],
                                unemp = start[["nbeta"]][j:(j+2)],
                                cpi = start[["nbeta"]][j:(j+2)],
                                dol = start[["nbeta"]][j:(j+2)],
                                rts = start[["nbeta"]][j:(j+2)],
                                bci = start[["nbeta"]][j:(j+2)],
                                oil = start[["nbeta"]][j:(j+2)])), silent = T)
          if (is.null(model) == FALSE) {break()}
        }
      }
    }} else {
      if (type2 == "nealmon") {
        for (i in 1:knealmon) {
model <- try(midas_r(as.formula(paste(
      "ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1, ", 1) +
                     mls(unemp, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, ", nealmo
                              mls(cpi, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(dol, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(rts, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(bci, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(oil, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                     start = list(unemp = start[["nealmon"]][i:(i+1)],
                                  cpi = start[["nealmon"]][i:(i+1)],
                                  dol = start[["nealmon"]][i:(i+1)],
                                  rts = start[["nealmon"]][i:(i+1)],
                                  bci = start[["nealmon"]][i:(i+1)],
                                  oil = start[["nealmon"]][i:(i+1)])), silent = T)
          if (is.null(model) == FALSE) {break()}
        }
      } else if (type2 == "nbeta") {
        for (j in 1:knbeta) {
model <- try(midas_r(as.formula(paste(
      "ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1, ", 1) +
                     mls(unemp, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, ", nbeta
                              mls(cpi, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(dol, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(rts, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(bci, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                              mls(oil, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3, "
                     start = list(unemp = start[["nbeta"]][j:(j+2)],
                                  cpi = start[["nbeta"]][j:(j+2)],
                                  dol = start[["nbeta"]][j:(j+2)],
                                  rts = start[["nbeta"]][j:(j+2)],
                                  bci = start[["nbeta"]][j:(j+2)],
                                  oil = start[["nbeta"]][j:(j+2)])), silent = T)
          if (is.null(model) == FALSE) {break()}
        }
      }
    }
  ave <- average_forecast(modlist = list(model), list(trend = trend, ex = ex, unemp
                                                      cpi = cpi, dol = dol, rts = rt
                                                      bci = bci, oil = oil),
                          insample = 1:69, outsample = 70:81,
                          type = "recursive", measures = c("MSE"))
  return(ave$accuracy$individual$MSE.out.of.sample)
}
```

```r
# loop
m_table <- midas_table
trend <- 1:length(ex)
for (row_num in 1:144) {
    model <- NULL
    type1 = m_table[row_num, 1]
    type2 = m_table[row_num, 2]
    DD = m_table[row_num, 3]
    M = m_table[row_num, 4]
    mon = m_table[row_num, 5]
    QQ = m_table[row_num, 6]
    try(m_table[row_num, "MSE"] <- midas_func(m_table = midas_table, row_num = 1))
}

# time-averaging
ave_table <- expand.grid(
  ctype1 = "NULL", ctype2 = "",
  l_lag = l_lag, h_lag = h_lag, months = months,
  quat = quat, MSE = 0
)
index <- rep(1:81, 3)
hf["ave_index"] <- sort(index)
unemp <- (summarise(group_by(hf, ave_index), mean(unemp))%>% as.data.frame())[, 2]
cpi <- (summarise(group_by(hf, ave_index), mean(cpi))%>% as.data.frame())[, 2]
dol <- (summarise(group_by(hf, ave_index), mean(dol))%>% as.data.frame())[, 2]
rts <- (summarise(group_by(hf, ave_index), mean(rts))%>% as.data.frame())[, 2]
bci <- (summarise(group_by(hf, ave_index), mean(bci))%>% as.data.frame())[, 2]
oil <- (summarise(group_by(hf, ave_index), mean(oil))%>% as.data.frame())[, 2]
for (row_num in 1:36) {
  DD = ave_table[row_num, 3]
  M = ave_table[row_num, 4]
  mon = ave_table[row_num, 5]
  QQ = ave_table[row_num, 6]
  model <- midas_r(as.formula(paste("ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1,
                     mls(unemp, ", QQ, ":", M/3 + QQ - 1, ", 1) +
                     mls(cpi, ", QQ, ":", M/3 + QQ - 1, ", 1) +
                     mls(dol, ", QQ, ":", M/3 + QQ - 1, ", 1) +
                     mls(rts, ", QQ, ":", M/3 + QQ - 1, ", 1) +
                     mls(bci, ", QQ, ":", M/3 + QQ - 1, ", 1) +
                     mls(oil, ", QQ, ":", M/3 + QQ - 1, ", 1)")), start = NULL)

  ave <- average_forecast(modlist = list(model), list(trend = trend, ex = ex, unemp
                                                      cpi = cpi, dol = dol, rts = rt
                                                      bci = bci, oil = oil),
                          insample = 1:69, outsample = 70:81,
                          type = "recursive", measures = c("MSE"))
  ave_table[row_num, "MSE"] <- ave$accuracy$individual$MSE.out.of.sample
}
# step-weighting
step_table <- expand.grid(
  ctype1 = "NULL", ctype2 = "NULL",
  l_lag = l_lag, h_lag = h_lag, months = months,
  quat = quat, MSE = 0
)
unemp <- hf$unemp
cpi <- hf$cpi
dol <- hf$dol
rts <- hf$rts
bci <- hf$bci
oil <- hf$oil
for (row_num in 1:36) {
  DD = step_table[row_num, 3]
  M = step_table[row_num, 4]
  mon = step_table[row_num, 5]
  QQ = step_table[row_num, 6]
  model <- midas_r(as.formula(paste("ex ~ trend + mls(ex, ", QQ, ":", DD + QQ - 1, "
                               mls(unemp, (", 3 - mon, ":(", M + 2 - mon, ")), ", 3
                                              mls(cpi, (", 3 - mon, ":(", M + 2 - mo
                                    mls(dol, (", 3 - mon, ":(", M + 2 - mon, ")), "
                                    mls(rts, (", 3 - mon, ":(", M + 2 - mon, ")), "
                                    mls(bci, (", 3 - mon, ":(", M + 2 - mon, ")), "
                                    mls(oil, (", 3 - mon, ":(", M + 2 - mon, ")), "
  step <- average_forecast(modlist = list(model), list(trend = trend, ex = ex, unemp
                                                      cpi = cpi, dol = dol, rts = rt
                                                      bci = bci, oil = oil),
                          insample = 1:69, outsample = 70:81,
                          type = "recursive", measures = c("MSE"))
  step_table[row_num, "MSE"] <- step$accuracy$individual$MSE.out.of.sample
}

# AR and ARIMA
# 1 step
ar_forecast_4_1step <- c()
ar_forecast_8_1step <- c()
arima_forecast_4_1step <- c()
arima_forecast_8_1step <- c()
for (i in 1:12) {
  ex_train <- ex[c(1:(68 + i))]
  ar_ex_4 <- ar(ex_train, order.max = 4)
  ar_forecast_4_1step_one <- predict(ar_ex_4, n.ahead = 1)$pred %>% as.vector()
  ar_forecast_4_1step <- c(ar_forecast_4_1step, ar_forecast_4_1step_one)
  ar_ex_8 <- ar(ex_train, order.max = 8)
  ar_forecast_8_1step_one <- predict(ar_ex_8, n.ahead = 1)$pred %>% as.vector()
  ar_forecast_8_1step <- c(ar_forecast_8_1step, ar_forecast_8_1step_one)
  arima_ex_4 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_4_1step_one <- predict(arima_ex_4, n.ahead = 1)$pred %>% as.vector
  arima_forecast_4_1step <- c(arima_forecast_4_1step, arima_forecast_4_1step_one)
  arima_ex_8 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_8_1step_one <- predict(arima_ex_8, n.ahead = 1)$pred %>% as.vector
  arima_forecast_8_1step <- c(arima_forecast_8_1step, arima_forecast_8_1step_one)
}
ex_test <- ex[70:81]
mse_arima_4_1step <- mse(ex_test, arima_forecast_4_1step)
mse_arima_8_1step <- mse(ex_test, arima_forecast_8_1step)
mse_ar_4_1step <- mse(ex_test, ar_forecast_4_1step)
mse_ar_8_1step <- mse(ex_test, ar_forecast_8_1step)
# 2 steps
```

```
ar_forecast_4_2step <- c()
ar_forecast_8_2step <- c()
arima_forecast_4_2step <- c()
arima_forecast_8_2step <- c()
for (i in 1:12) {
  ex_train <- ex[c(1:(67 + i))]
  ar_ex_4 <- ar(ex_train, order.max = 4)
  ar_forecast_4_2step_one <- (predict(ar_ex_4, n.ahead = 2)$pred %>% as.vector())[2]
  ar_forecast_4_2step <- c(ar_forecast_4_2step, ar_forecast_4_2step_one)
  ar_ex_8 <- ar(ex_train, order.max = 8)
  ar_forecast_8_2step_one <- (predict(ar_ex_8, n.ahead = 2)$pred %>% as.vector())[2]
  ar_forecast_8_2step <- c(ar_forecast_8_2step, ar_forecast_8_2step_one)
  arima_ex_4 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_4_2step_one <- (predict(arima_ex_4, n.ahead = 2)$pred %>% as.vector
  arima_forecast_4_2step <- c(arima_forecast_4_2step, arima_forecast_4_2step_one)
  arima_ex_8 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_8_2step_one <- (predict(arima_ex_8, n.ahead = 2)$pred %>% as.vector
  arima_forecast_8_2step <- c(arima_forecast_8_2step, arima_forecast_8_2step_one)
}
ex_test <- ex[70:81]
mse_arima_8_2step <- mse(ex_test, arima_forecast_8_2step)
mse_arima_4_2step <- mse(ex_test, arima_forecast_4_2step)
mse_ar_4_2step <- mse(ex_test, ar_forecast_4_2step)
mse_ar_8_2step <- mse(ex_test, ar_forecast_8_2step)
# 3 steps
ar_forecast_4_3step <- c()
ar_forecast_8_3step <- c()
arima_forecast_4_3step <- c()
arima_forecast_8_3step <- c()
for (i in 1:12) {
  ex_train <- ex[c(1:(66 + i))]
  ar_ex_4 <- ar(ex_train, order.max = 4)
  ar_forecast_4_3step_one <- (predict(ar_ex_4, n.ahead = 3)$pred %>% as.vector())[3]
  ar_forecast_4_3step <- c(ar_forecast_4_3step, ar_forecast_4_3step_one)
  ar_ex_8 <- ar(ex_train, order.max = 8)
  ar_forecast_8_3step_one <- (predict(ar_ex_8, n.ahead = 3)$pred %>% as.vector())[3]
  ar_forecast_8_3step <- c(ar_forecast_8_3step, ar_forecast_8_3step_one)
  arima_ex_4 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_4_3step_one <- (predict(arima_ex_4, n.ahead = 3)$pred %>% as.vector
  arima_forecast_4_3step <- c(arima_forecast_4_3step, arima_forecast_4_3step_one)
  arima_ex_8 <- auto.arima(ex_train, allowdrift = F)
  arima_forecast_8_3step_one <- (predict(arima_ex_8, n.ahead = 3)$pred %>% as.vector
  arima_forecast_8_3step <- c(arima_forecast_8_3step, arima_forecast_8_3step_one)
}
ex_test <- ex[70:81]
mse_arima_4_3step <- mse(ex_test, arima_forecast_4_3step)
mse_arima_8_3step <- mse(ex_test, arima_forecast_8_3step)
mse_ar_4_3step <- mse(ex_test, ar_forecast_4_3step)
mse_ar_8_3step <- mse(ex_test, ar_forecast_8_3step)
ar_n_arima_ex <- data.frame(mse_arima_4_1step, mse_arima_4_2step, mse_arima_4_3step,
                            mse_arima_8_1step, mse_arima_8_2step, mse_arima_8_3step,
                            mse_ar_4_1step, mse_ar_4_2step, mse_ar_4_3step,
                            mse_ar_8_1step, mse_ar_8_2step, mse_ar_8_3step)
```