ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis BSc^2 Econometrics/Economics

Forecasting Economic Growth in Nigeria using Shrinkage and Variable Selection Methods

Name student: F.C. Neefjes
Student ID number: 429638

Supervisor: prof. dr. P.H.B.F. Franses
Second assessor: dr. A.M. Schnucker

Date final version: July 5, 2019

Abstract

Economic forecasts are of major importance to set government policies and to make investment decisions. A recent development in economic forecasting is to combine shrinkage and variable selection methods with factorisation. This research assesses the usefulness of this approach in the context of forecasting contemporary Nigerian real GDP growth. A novel type of data set including real GDP growth rates of 52 African countries and a data set containing 35 economic indicators are used to make predictions. Five shrinkage and variable selection methods are used, including least angle regression, ridge regression, elastic net regularisation, bagging, and boosting. A simulation study shows these methods are very effective when the explanatory power of many variables is low. However, their effectiveness is only limited in the application of forecasting Nigeria's economic growth, due to the low explanatory power of the data. A factor-based approach is generally preferred to using variables directly, although boosting without factors is the optimal method in terms of predictive accuracy. The best shrinkage and variable selection methods that use factors perform more than 20% better than the autoregressive benchmark, indicating these methods can be of added value in practice. Yet, the out-of-sample period of five years does not allow for strong conclusions. Therefore, more extensive data should be used in future research to verify the robustness of the results.

Contents

1 Introduction

Nigeria has the largest economy in Africa and with around 200 million inhabitants it is one of the most populated countries in the world. The country's primary source of welfare is oil, which makes economic growth subject to oil shocks (Ogundipe et al., 2014). The political environment has also considerably impacted economic developments in Nigeria. The country was led by military regimes until 1999, after which the country became a democracy. However, even in the present day corruption remains a severe problem (Nwankwo, 2014). Nevertheless, the country managed to realise relatively stable growth during the past decade. However, the economic growth was interrupted in 2016 by the first economic downturn in years. This recession was primarily caused by declining oil prices, an excessive amount of imports, and low investment rates (Agri et al., 2017). The main goal of this paper is to investigate whether such short-term economic developments in Nigeria can be predicted well by combining shrinkage and variable selection methods with factorisation.

The approach of combining shrinkage and variable selection methods is based on the methodology of Kim and Swanson (2018), who use this approach to forecast various economic variables in the United States. First, factors are generated from the data using principal component analysis (PCA) to reduce the dimensionality of the data. Then, shrinkage and variable selection methods are used to decrease the magnitude of coefficients or the number of variables in the model, which reduces the forecast variance. Various factor models are considered, including factor augmented autoregression (FAAR), ridge regression, least angle regression (LARS), elastic net regularisation, bagging, and boosting. The forecasting performance of the models is compared to autoregressive and random walk models. A simulation study is performed to investigate the differences between the methods when used in combination with factors. Thereafter, the methods are applied to make one-year-ahead out-of-sample forecasts of Nigerian real gross domestic product (GDP) growth from 2012 until 2016. The methods are compared in terms of their predictive accuracy. Moreover, it is evaluated whether a factor-based approach is preferred to the direct use of explanatory variables.

To forecast economic growth, two large data sets are considered. The first consists of real GDP growth rates of 52 African countries from 1963 until 2016, which is obtained from Franses and Vasilev (2019). This data set is used to investigate whether cross-country relations can be used to predict economic growth. The second data set consists of 35 economic indicators and is primarily retrieved from the World Bank's World Development Indicators (WDI) database. The explanatory power of both data sets is first compared by using them independently, thereafter a combination of the two is considered.

The social relevance of this paper is twofold. First, providing accurate predictions of economic growth is of vital importance for the Nigerian government, as the government budget is largely dependent on the state of the economy. Furthermore, more accurate economic growth forecasts contribute to an improved implementation of fiscal policy (Lledó and Poplawski-Ribeiro, 2013). In addition, the Central Bank of Nigeria uses economic forecasts to adjust monetary policy, which can aid to stimulate or dampen economic growth. Second, accurate information regarding economic growth in Nigeria is of interest to investors, as the value of investments is dependent on the growth outlook.

From a scientific perspective, a contribution to the literature is made by further exploring the effectiveness of shrinkage and variable selection methods in combination with factorisation in a new context. Previous research found the best method to depend both on the time period and the variable to be predicted (Kim and Swanson, 2014, 2018), hence insight is provided into which of the methods is most suitable to forecast Nigerian real GDP growth. Above that, the predictive power of a new type of data set containing regional economic growth rates is researched by using GDP growth rates of African countries as explanatory variables. Moreover, the applicability of economic indicator data to make economic forecasts in Africa has until now remained relatively unexplored, with the exception of South Africa.

In the simulation study, it is found shrinkage and variable selection methods can outperform ordinary least squares (OLS) in terms of forecasting accuracy, especially in case the explanatory power of many variables is low. The preferred method is based on a trade-off between variance and bias, as the number of variables and the size of coefficients differ among them. The effectiveness of combining a factor-based approach with shrinkage and variable selection is only limited in the context of forecasting Nigerian real GDP growth, as the benchmark models are hardly outperformed. The explanatory power of individual variables in both data sets is low, which leads to relatively better performance of factor models in general. The best method in terms of forecasting accuracy is boosting together with variables of both data sets without factorisation, however, the model's forecasts are very volatile and therefore not suitable in practice. Bagging and boosting with factors constructed using a combination of the data sets generate more stable forecasts and are able to reduce the mean squared prediction error (MSE) by more than 20% compared to the autoregressive benchmark, which indicates this approach can be useful to forecast economic growth in Nigeria.

The paper is organised as follows. Section 2 discusses relevant literature. Section 3 provides an overview of the models used in the simulation and empirical studies. The simulation study is discussed in Section 4. The data and methodology used to forecast Nigerian real GDP growth are presented with the results in Section 5. Section 6 concludes the paper.

2 Literature Review

2.1 Economic Growth in Africa and Nigeria

Economic development in Africa has been slow historically, which largely results from the pursuance of poor economic policies as well as unfavourable natural factors (Sachs and Warner, 2001). However, fundamental changes in the economy have led to substantially better economic performance during the last two decades (Rodrik, 2016). Arbache and Page (2007) study commonalities in the economic growth of African countries and find they partly follow the same business cycles. Above that, Arbache and Page (2007) find economic correlations in Sub-Saharan Africa are seemingly based on institutions rather than geography and natural resources. This paper investigates whether cross-country relations in Africa can be used to predict economic growth in Nigeria, which is done by using the real GDP growth rates of African countries as explanatory variables. The application of a data set containing regional economic growth rates is novel in a forecasting context.

Arbache and Page (2007) note the economic growth of African countries is volatile and diverse, therefore it is detrimental to know which factors influence economic growth within Nigeria. Various studies investigate the determinants of economic growth in Nigeria. For instance, the economy is found to benefit from oil shocks (Omisakin and Olusegun, 2008; Ogundipe et al., 2014). In addition, high inflation negatively impacts GDP growth in Nigeria (Chimobi, 2010). Furthermore, Akinlo (2004) finds inflows of foreign direct investment (FDI) contribute positively to economic growth, although the effects are small, dependent on the sector, and only occur after a prolonged time period. Moreover, he finds increases in exports, the labour force, and human capital augment economic growth, whereas an expansion of the money supply as a share of GDP has an adverse effect. Nurudeen and Usman (2010) find government consumption has a negative impact on economic growth overall, although the effects depend on the type of expenditure; increased expenditure on health care, communication, and transportation do exert positive influences on the economy. This study is related to the work of Nwankwo (2014), who finds that corruption significantly deteriorates growth in Nigeria.

Besides causal variables, leading indicators can be used to predict economic growth. For example, Ikoku (2010) finds the stock market is a leading indicator of Nigeria's economy. The application of economic indicators to predict the economic growth of African countries has been relatively unexplored. Existing literature primarily focuses on South Africa, which is the most developed African country. Gupta and Kabundi (2011) forecast economic growth in South Africa using a data set containing 267 variables, which includes financial data, global commodity prices, data on major trading partners, confidence indices, and business surveys. They use PCA to obtain factors, which are subsequently used as explanatory variables in the predictive model. This factor model outperforms other models when predicting short-term economic growth in South Africa (Gupta and Kabundi, 2011). In a comparable study, Cepni et al. (2019) make use of economic indicators to forecast GDP growth rates in Brazil, Indonesia, Mexico, South Africa, and Turkey. Their data set includes economic indicators relating to housing, new orders, the labour market, prices, interest rates, exchange rates, stock prices, the money supply, and real production. Above that, they include variables that capture uncertainty regarding the economy, trade, monetary policy, and migration. To forecast GDP growth in emerging markets they make use of a factor-based approach in combination with shrinkage and variable selection methods, which jointly significantly improves predictive accuracy.

This paper combines variables that cause economic growth in Nigeria with a myriad of economic indicators in a second data. By using this data set for forecasting, a contribution to the existing literature is made by combining variables known to affect or predict economic growth into a single model. Also, new insights are provided into the effectiveness of economic indicator data in making economic forecasts for a less developed African economy. Similar to the aforementioned research, factorisation is used as a dimension reduction technique. Factor models are constructed using shrinkage and variable selection methods, which are discussed next.

2.2 Shrinkage and Variable Selection Methods

Shrinkage methods, also known as regularisation methods, reduce the magnitude of coefficients in a regression model towards zero. On the other hand, variable selection methods select a subset of variables to include in a model. Therefore, variable selection can be seen as an extreme case of shrinkage, where coefficients are shrunk all the way to zero. Both methods aim to diminish the size of the coefficients, which helps to reduce the variance of the model and to avoid overfitting. These two properties make these methods particularly beneficial in the application of forecasting. However, there is a trade-off, since shrinking coefficients comes at the expense of increased bias.

Kim and Swanson (2018) combine different factor models with a variety of shrinkage and variable selection methods to forecast 11 macroeconomic variables in the United States. This is done by first generating factors from the explanatory variables and subsequently using shrinkage and variable selection methods with the estimated factors. However, they also test the performance of shrinkage and variable selection methods without the use of factors. In this paper, both approaches are used to model Nigerian GDP growth. The subset of shrinkage and variable selection methods used is in this research is briefly described below.

Least angle regression is a variable selection method introduced by Efron et al. (2004). The method resembles forward selection, which starts with an empty model and iteratively adds the predictor that is most correlated with the current residuals. This greedy algorithm does not allow for a model with highly correlated variables. As an alternative, forward stagewise regression follows a similar procedure, however, in each iteration, it only increases the coefficient of the most correlated variable with a small step. Although this method may provide a better model fit, it also more computationally intensive. LARS aims to accelerate the forward stagewise regression by increasing the step size and adjusting coefficients of multiple predictors at the same time. Similar to forward selection, LARS iteratively adds the variable that is most correlated with the current residuals to the model. Then, the coefficients of the variables in the model are increased into their joint least squares direction until some other variable becomes just as correlated with the current residuals. At that point, a new variable is added to the model and the algorithm repeats itself until all variables are included in the model.

The second method considered is ridge regression, which is a form of penalised regression where coefficients are given a squared penalty in the objective function (Hoerl and Kennard, 1970). This method has the advantage that it has a closed-form solution, however, it does not lead to a parsimonious model as coefficients are not reduced all the way to zero. To overcome this caveat, Tibshirani (1996) introduced the least absolute shrinkage and selection operator (lasso). Lasso uses an absolute penalty term instead of a squared penalty, therefore it is able to create sparse models. However, lasso only allows the number of regressors to at maximum be equal to the sample size and it tends to select one variable from a highly correlated group of variables somewhat arbitrarily (Zou and Hastie, 2005). To alleviate these problems, Zou and Hastie (2005) developed elastic net regularisation, which includes both a squared and an absolute penalty in the objective function. Although this method is seemingly different from LARS, Zou and Hastie (2005) show the solution of elastic net regularisation can be computed efficiently with an adjusted version of the LARS algorithm, which is called LARS-EN.

The final two methods considered, bagging and boosting, make use of machine learning to achieve shrinkage and variable selection. Bagging, short for bootstrap aggregation, is introduced by Breiman (1996). This method bootstraps the training sample multiple times and computes coefficients for each sample. Subsequently, with each bootstrapped sample a different prediction is made and the bagging forecast is computed as the average of the forecasts. Bühlmann and Yu (2002) show this computationally intensive method can asymptotically be seen as a form of shrinkage, where the amount of shrinkage for each coefficient is inversely related to the absolute value of the t-statistic. Next, boosting aims to combine many models with low accuracy into a single accurate model (Freund and Schapire, 1997). Numerous boosting algorithms exist, some of these are based on the aforementioned forward stagewise regression (Efron et al., 2004). With this method, the coefficient of the predictor that is most correlated with the residuals is increased with a small step, which by itself contributes little to the model's accuracy. However, by repeating this procedure a large number of times a single accurate model can be created.

In literature, there is no consent on which shrinkage or variable selection method is optimal. As an illustration, Kim and Swanson (2014, 2018) find the preferred method depends on both the macroeconomic variable to predicted and the time period considered. In this paper, the performance of shrinkage and variable selection methods is first compared in a simulation study. Afterwards, it is evaluated which method is most suitable in the context of forecasting contemporary Nigerian real GDP growth.

3 Regression Models

This section provides an overview of the regression models used in the simulation and empirical studies, commencing with the benchmark models. Subsequently, the procedure to generate and select factors is elaborated upon. Then, the implementation of shrinkage and variable selection methods is discussed. For simplicity, it is assumed all models make use of one lag of factors, which is analogous to the application of these models in the simulation study. However, the models may also be applied without the use of factors and they can easily be extended to include more lags than one. These adjustments are both made in the empirical study.

As a convention, vectors are written in bold and matrices are capitalised. Moreover, estimated variables are denoted with a hat and normalised variables with a tilde. The time span is assumed to be T periods and time is denoted by subscript t. The dependent variable is assumed to be a $T \times 1$ vector $\boldsymbol{y}$ with elements y_t for $t = 1, ..., T$. Explanatory variables are denoted by $T \times N$ matrix X, where N is the amount of explanatory variables. Individual explanatory variables are denoted by $x_{i,t}$ for $i = 1, ..., N$ and $t = 1, ..., T$. Factors are denoted by $T \times N$ matrix F, with F^r denoting the first r factors. The t^{th} row of F is denoted by $\boldsymbol{F_t}$ and the j^{th} factor by $\boldsymbol{f_j}$ with elements $f_{j,t}$ for $t = 1, ..., T$.

3.1 Autoregressive Model

An autoregressive (AR) model is used as a benchmark model, as this simple linear model is difficult to beat when forecasting economic growth (Marcellino, 2008). The AR(p) model is described by

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i y_{t-i} + \epsilon_t, \tag{1}$$

where p denotes the lag order and ϵ_t is an error term. The lag order p is chosen such that the Bayesian information criterion (BIC) is minimised. The restriction $1 \leq p \leq 5$ is imposed, as information from more than five years is unlikely to have explanatory power. In selecting the lag order, the sample size is kept the same for each p by discarding the first five observations. Therefore, the BIC is computed in this settings as

$$\text{BIC}(p) = (T-5)\ln(\hat{\sigma}_p^2) + (p+1)\ln(T-5), \tag{2}$$

where T denotes the sample size, p is the autoregressive order, and $\hat{\sigma}_p^2 = \frac{1}{T}\sum_{i=6}^{T}\hat{\epsilon}_t^2$ is the estimated variance of the residuals when p autoregressive terms are included in (1). After the optimal p is found, OLS is used to estimate (1) and to obtain coefficients. Then, one-step ahead forecasts are made as

$$\hat{y}_{t+1}^{\text{AR}} = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i y_{t+1-i}. \tag{3}$$

3.2 Random Walk Model

The second benchmark model is the random walk (RW) model, which assumes the dependent variable takes a random step at each point in time. Predictions with this model are made as

$$\hat{y}_{t+1}^{\text{RW}} = y_t, \tag{4}$$

which has the clear advantage that no parameters need to be estimated.

3.3 Factor Augmented Autoregression

In order to incorporate a high-dimensional $T \times N$ matrix X with explanatory variables into a parsimonious model, the information contained in the data is summarised by r factors with $r << N$. Factors F are constructed using standardised explanatory variables $\tilde{X}$ as $F = \tilde{X}W$, where $W = [\boldsymbol{w_1}, ..., \boldsymbol{w_N}]$ is a $N \times N$ coefficient matrix that gives weights to each of the variables. The weights, also known as loadings, are determined using PCA. This method iteratively maximises the variance contained in each factor, while restricting the weight vector to have unit length and to be orthogonal to any other weight vector. This leads to the following optimisation for $i \in 1, ..., N$:

$$\underset{\boldsymbol{w_i}}{\text{argmax}}\, \boldsymbol{w_i}^\top \tilde{X}^\top \tilde{X} \boldsymbol{w_i} : \boldsymbol{w_i} \cdot \boldsymbol{w_j} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j < i. \end{cases} \tag{5}$$

The solution to (5) is given by each $\boldsymbol{w_i}$ being equal to an eigenvector of the covariance matrix of $\tilde{X}$, where the columns of W are ordered such that the first eigenvector corresponds to the largest eigenvalue[1]. After obtaining weights W using PCA, factors are estimated as $\hat{F} = \tilde{X}W$.

[1] Matrix $\tilde{X}$ is a standardised version of X, therefore the same results can be obtained by applying PCA to the correlation matrix of X.

To reduce the dimension of the explanatory variables, a subset of factors from $\hat{F}$ needs to be selected. The factors are ordered by the variance of $\tilde{X}$ captured, therefore it is natural to select the first r factors, as they likely contain most information. Following Kim and Swanson (2018), an adjusted BIC developed by Bai and Ng (2002) is used to select the number of factors in the empirical study. This information criterion makes an adjustment to the standard BIC to better handle panel data and is computed as

$$\text{BIC}_{\text{panel}}(r) = V(r, \hat{F}^r) + rV(r_{\max}, \hat{F}^{r_{\max}}) \left(\frac{(N+T-r)\ln(NT)}{NT} \right), \tag{6}$$

where $r_{\max}$ is the maximum amount of factors considered and $V(r, \hat{F}^r)$ is the mean of squared residuals obtained after regressing $\tilde{X}$ on the first r estimated factors denoted by $\hat{F}^r$. More formally,

$$V(r, \hat{F}^r) = \min_{\Lambda} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} (\tilde{x}_{i,t} - \hat{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}} \boldsymbol{\lambda}_{\boldsymbol{i}}^{\boldsymbol{r}})^2, \tag{7}$$

where the $r \times 1$ vector $\boldsymbol{\lambda}_{\boldsymbol{i}}^{\boldsymbol{k}}$ is estimated using OLS and $\hat{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}}$ is a vector corresponding to the first r elements in row t of $\hat{F}$. The number of factors r with the lowest information criterion is selected using $r_{\max} = 20$.

The $T \times r$ matrix $\hat{F}^r$ is used in all shrinkage and variable selection methods. Above that, factors are used in the FAAR model, which extends (1) by including r lagged factors. Ordinary least squares is used to estimate

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i y_{t-i} + \hat{\boldsymbol{F}}_{\boldsymbol{t-1}}^{\boldsymbol{r}} \boldsymbol{\gamma} + \epsilon_t, \tag{8}$$

where $\boldsymbol{\gamma}$ is a $r \times 1$ vector of coefficients. Estimated coefficients from (8) are used to make one-step-ahead predictions as

$$\hat{y}_{t+1}^{\text{FAAR}} = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i y_{t+1-i} + \hat{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}} \hat{\boldsymbol{\gamma}}. \tag{9}$$

The performance of the FAAR model indicates the usefulness of a factor approach when compared to the AR(p) model, but simultaneously serves as a benchmark for shrinkage and variable selection methods that also use factors as explanatory variables.

3.4 Least Angle Regression

Least angle regression is a variable selection method introduced by Efron et al. (2004). The implementation of LARS is akin to that of Kim and Swanson (2018). Before using the LARS algorithm, estimated residuals $\hat{\boldsymbol{z}}$ are obtained from the AR(p) model. The first r standardised factors are denoted by $\tilde{F}^r$, with $\tilde{\boldsymbol{f}}_{\boldsymbol{j}}$ corresponding to j^{th} standardised factor. Let $\mathcal{G}^i$ denote the set of factors included in the model at iteration i, with $\tilde{\boldsymbol{F}}_{\boldsymbol{\mathcal{G}^i}}$ denoting a $T \times i$ matrix of standardised factors in the active set. Furthermore, let $\hat{\boldsymbol{\mu}}^{\boldsymbol{i}} = \tilde{F}^r \hat{\boldsymbol{\beta}}^{\boldsymbol{i}}$ be the explained part of the model. Here, $\hat{\boldsymbol{\beta}}^{\boldsymbol{i}}$ is a $r \times 1$ coefficient vector with elements $\hat{\beta}_j^i$ for $j = 1, ..., r$. The set $\mathcal{G}^0$ is empty and $\hat{\boldsymbol{\beta}}^{\mathbf{0}} = \mathbf{0}$ such that $\hat{\boldsymbol{\mu}}^{\mathbf{0}} = \mathbf{0}$. The LARS algorithm proceeds as follows for $i = 1, ..., r$:

1. Compute the correlations between the factors and the current residuals as $\hat{\boldsymbol{c}} = \tilde{\boldsymbol{F}}^{\boldsymbol{r}\top}(\hat{\boldsymbol{z}} - \hat{\boldsymbol{\mu}}^{\boldsymbol{i-1}})$, with elements $\hat{c}_j$ corresponding to each of the j factors. Add factor j with the highest correlation $|\hat{c}_j|$ that is not in $\mathcal{G}^{i-1}$ to the active set $\mathcal{G}^i$.
2. Let $\mathcal{F}_{\mathcal{G}_i} = [..., \tilde{\boldsymbol{f}}_{\boldsymbol{j}} s_j, ...]_{j \in \mathcal{G}_i}$ be a $T \times i$ matrix of factors in the active set multiplied with the the sign of the correlation $s_j = \text{sign}\{\hat{c}_j\}$. Moreover, let $A_{\mathcal{G}_i} = \left(\mathbf{1}_{\boldsymbol{i}}^\top \left(\mathcal{F}_{\mathcal{G}_i}^\top \mathcal{F}_{\mathcal{G}_i}\right)^{-1} \mathbf{1}_{\boldsymbol{i}}\right)^{-1/2}$, where $\mathbf{1}_{\boldsymbol{i}}$ is a $i \times 1$ vector of ones. Then, compute the weighting vector $\boldsymbol{w}_{\boldsymbol{i}} = A_{\mathcal{G}_i} \times \left(\mathcal{F}_{\mathcal{G}_i}^\top \mathcal{F}_{\mathcal{G}_i}\right)^{-1} \mathbf{1}_{\boldsymbol{i}}$ with i elements $w_{i,j}$ for $j \in \mathcal{G}_i$.
3. Compute the equiangular vector as $\boldsymbol{u}_{\boldsymbol{i}} = \tilde{\boldsymbol{F}}_{\boldsymbol{\mathcal{G}}^{\boldsymbol{i}}} \boldsymbol{w}_{\boldsymbol{i}}$.
4. Update $\hat{\mu}^{i-1} = \hat{\mu}^{i-1} + \hat{\gamma} \boldsymbol{u}_{\boldsymbol{i}}$, in which the step length is determined as

$$\hat{\gamma} = \min_{j \notin \mathcal{G}_i}{}^{+} \left\{ \frac{c^* - \hat{c}_j}{A_{\mathcal{G}_i} - a_m}, \frac{c^* + \hat{c}_j}{A_{\mathcal{G}_i} + a_j} \right\}, \tag{10}$$

   where $\min^+$ means the lowest positive element is selected, $c^* = \max\{|\hat{c}_j|\}$, and a_j is the j^{th} column of $\tilde{F}^r \boldsymbol{u}_{\boldsymbol{i}}$. Intuitively, this means that the coefficients of factors in the active set are altered in such a way that they move towards their joint squares solution, until some other factor becomes as correlated as the factors currently included in the model.
5. For each $j \in \mathcal{G}_i$, update the coefficient as $\hat{\beta}_j^{i-1} = \hat{\beta}_j^i + w_{i,j}$. Otherwise, set $\hat{\beta}_j^i = \hat{\beta}_j^{i-1}$.

Each iteration of the LARS algorithm produces a different coefficient vector, where the number of non-zero elements is equal to the iteration count. In the last iteration, the coefficient estimates are equal to OLS estimates. Tenfold cross-validation is used to find the optimal coefficient vector. This method splits the sample into ten sub-samples of equal size, each containing an equal number of consecutive years. The reason for using cross-validation is that with a limited sample size all observations are used to validate which coefficient vector is optimal. In addition, it enhances sub-sample stability, as each time period is predicted once. Cross-validation applies the following procedure for $i = 1, ..., 10$:

1. Apply the LARS algorithm using all sub-samples excluding i, providing $r + 1$ different coefficient $\hat{\boldsymbol{\beta}}^{\mathbf{0}}, ..., \hat{\boldsymbol{\beta}}^{\boldsymbol{r}}$.
2. Use linear interpolation to acquire 1,000 coefficients vectors $\hat{\boldsymbol{b}}_{\mathbf{1}}, ..., \hat{\boldsymbol{b}}_{\boldsymbol{k}}, ..., \hat{\boldsymbol{b}}_{\mathbf{1000}}$ from the $r + 1$ coefficient vectors.
3. Make out-of-sample forecasts for the observations in sub-sample i using each of the 1,000 coefficient vectors and store the sum of squared residuals (SSR).

After applying this procedure, the optimal index k_{opt} (out of 1,000) is chosen to be the smallest index with the average SSR within one standard deviation of the lowest average SSR. In this way, a coefficient vector from an earlier iteration is selected, which implies the model becomes more sparse. After obtaining the optimal index, the LARS algorithm is applied on the entire sample and again linear interpolation is used to acquire 1,000 coefficient vectors $\hat{\boldsymbol{b}}_{\mathbf{1}}, ..., \hat{\boldsymbol{b}}_{\boldsymbol{k}}, ..., \hat{\boldsymbol{b}}_{\mathbf{1000}}$. Then, the LARS coefficients are selected as $\hat{\boldsymbol{\beta}}^{\text{LARS}} = \hat{\boldsymbol{b}}_{\boldsymbol{k}_{\text{opt}}}$. Finally, forecasts are made with standardised factors $\tilde{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}}$ as

$$\hat{y}_{t+1}^{\text{LARS}} = \hat{y}_{t+1}^{\text{AR}} + \tilde{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}} \hat{\boldsymbol{\beta}}^{\text{LARS}}. \tag{11}$$

3.5 Ridge Regression

Ridge regression is a form of penalised regression introduced by Hoerl and Kennard (1970). This shrinkage method adds a squared penalty term for coefficients to the standard ordinary least squares equation. Following Kim and Swanson (2018), estimated residuals $\hat{\boldsymbol{z}}$ from the AR(p) model are regressed on r standardised factors, such that the optimisation problem becomes

$$\hat{\boldsymbol{\beta}}^{\text{Ridge}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\, ||\hat{\boldsymbol{z}} - \tilde{F}^r \boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_2^2, \tag{12}$$

where λ is the ridge parameter and $||\cdot||_p$ denotes the L_p norm. As shown by Hoerl and Kennard (1970), the solution to (12) is given by

$$\hat{\boldsymbol{\beta}}^{\text{Ridge}} = \left[I_r + \lambda \left(\tilde{F}^{r\top} \tilde{F}^r \right)^{-1} \right]^{-1} \hat{\boldsymbol{\beta}}^{\text{OLS}} \tag{13}$$

where I_r is a $r \times r$ identity matrix, and $\hat{\boldsymbol{\beta}}^{\text{OLS}}$ are ordinary least squares estimates obtained by regressing $\hat{F}^r$ on $\hat{\boldsymbol{z}}$. Given that the factors are orthogonal, it holds that $\tilde{F}^{r\top}\tilde{F}^r = I_r$, such that (13) can be simplified to

$$\hat{\boldsymbol{\beta}}^{\text{Ridge}} = (1+\lambda)^{-1} \hat{\boldsymbol{\beta}}^{\text{OLS}}, \tag{14}$$

thereby directly showing the relation between the ridge parameter and OLS estimates. As $\lambda \to 0$ the ridge regression produces the same estimates as OLS, whereas as $\lambda \to \infty$ the coefficients become zero. A downside of the ridge estimator is that coefficients are shrunk towards zero, but they do not exactly become exactly zero.

The optimal ridge parameter is found using cross-validation, where a grid of λ ranging from zero to 100 with step size 0.01 is used. The cross-validation procedure is similar to that described in section 3.4. However, the 10,001 different coefficient vectors are compared directly, such that there is no need for interpolation. Moreover, the optimal λ is simply selected to be the one that yields the lowest average SSR, because the number of variables with ridge regression cannot be reduced by choosing a different parameter. Using the optimal coefficient vector selected using cross-validation, forecasts are made as

$$\hat{y}_{t+1}^{\text{Ridge}} = \hat{y}_{t+1}^{\text{AR}} + \tilde{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}} \hat{\boldsymbol{\beta}}^{\text{Ridge}}. \tag{15}$$

3.6 Elastic Net Regularisation

Elastic net (EN) regularisation adds a L_1 norm penalty to optimisation problem (12) of ridge regression, which yields

$$\hat{\boldsymbol{\beta}}^{\text{NEN}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\, ||\hat{\boldsymbol{z}} - \tilde{F}^r \boldsymbol{\beta}||_2^2 + \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||_2^2, \tag{16}$$

where λ_1 and λ_2 are two tuning parameters. The cases $\lambda_1 = 0$ and $\lambda_2 = 0$ lead to ridge regression and lasso, respectively. If both parameters are non-zero, then double shrinkage is applied. Therefore, Zou and Hastie (2005) refer to this method as the naive elastic net (NEN) and suggest scaling the coefficients as follows:

$$\hat{\boldsymbol{\beta}}^{\text{EN}} = (1 + \lambda_2)\hat{\boldsymbol{\beta}}^{\text{NEN}}. \tag{17}$$

As shown by Zou and Hastie (2005), for a fixed value of λ_2 the elastic net problem can be solved with the same method as lasso regularisation by making a straightforward data adjustment. In turn, the solution path of lasso can be computed with the LARS algorithm by restricting the sign of the coefficients to agree with the current correlation and allowing factors to be removed if this restriction is violated (Efron et al., 2004). Combining these findings, elastic net regularisation can compute coefficients efficiently using a slightly adjusted version of the LARS algorithm named LARS-EN. As done by Kim and Swanson (2018), LARS-EN is used to compute the entire solution path for $\lambda_2 \in \{0, 0.01, 0.1, 1, 10, 100\}$, which yields six different series of coefficient vectors. First, the optimal index for each λ_2 is found using the cross-validation procedure described in Section 3.4. Afterwards, the optimal λ_2^* is selected to be the parameter for which the optimal index provides the lowest average SSR. Then, LARS-EN is applied to the full sample with λ_2^* and linear interpolation is used to get 1,000 coefficient vectors. Subsequently, $\hat{\boldsymbol{\beta}}^{EN}$ is selected using the optimal index belonging to λ_2^* and forecasts are made as

$$\hat{y}_{t+1}^{\text{EN}} = \hat{y}_{t+1}^{\text{AR}} + \tilde{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}} \hat{\boldsymbol{\beta}}^{\text{EN}}. \tag{18}$$

3.7 Bagging

In line with the methodology Kim and Swanson (2018), bagging is performed the shrinkage way using the following procedure of Bühlmann and Yu (2002):

1. Estimate FAAR model (8) using OLS to obtain estimated coefficients $\hat{\beta}_0, ..., \hat{\beta}_p$ for the autoregressive part, estimated coefficients $\hat{\boldsymbol{\gamma}} = [\hat{\gamma}_1, ..., \hat{\gamma}_r]$ for the factors, and estimated residuals $\hat{\boldsymbol{\epsilon}}$.
2. Compute t-statistics for the estimated factors using Newey-West standard errors, which are denoted by t_j for $j = 1, ..., r$.
3. If $t_j < c$, set the shrinkage coefficient $\psi_j(t_j)$ of factor j to zero, where c is a constant critical value. Otherwise,

   $$\psi_j(t_j) = 1 - \Phi(t_j + c) + \Phi(t_j - c) + t_j^{-1}(\phi(t_j - c) - \phi(t_j + c)), \tag{19}$$

   where ϕ is the standard normal probability density function and Φ is the standard normal cumulative density function. The function $\psi_j(t_j)$ is a monotonically increasing function that converges to one as t_j increases.

Bagging is implemented with $c = 1.96$, which corresponds to the standard normal critical value for a two-tailed test at the 5% level. In this way, variables that are insignificant are set to zero, whereas variables with t-statistics just above the critical value are shrunk. Forecasts are made with unadjusted autoregressive coefficients and shrunk coefficients of $\hat{\boldsymbol{\gamma}}$ using

$$\hat{y}_{t+1}^{\text{Bagging}} = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i y_{t+1-i} + \sum_{j=1}^{r} \psi_j(t_j)\hat{\gamma}_j \hat{f}_{j,t}, \tag{20}$$

where $\hat{f}_{j,t}$ denotes the estimated value of factor j at time t.

3.8 Boosting

Boosting aims to improve the model's accuracy by repeatedly making use of weak learning algorithms, which are algorithms that only slightly improve upon the current model specification. Analogous to Kim and Swanson (2018), L_2Boosting with componentwise linear least of Bühlmann and Hothorn (2007) is implemented. In each iteration, the coefficient of the variable that reduces the squared forecast error the most is increased.

Residuals $\hat{\boldsymbol{z}}$ from the AR(p) model are used as the dependent variable and estimated factors $\hat{F}^r$ are used as explanatory variables. The algorithm is initialised with iteration limit m and $r \times 1$ coefficient vector $\hat{\boldsymbol{\beta}}^{\mathbf{0}} = \mathbf{0}$. Moreover, $T \times T$ matrix $D^0 = 0_{T\times T}$ is used to capture the model complexity and $T \times 1$ vector $\boldsymbol{\mu}^{\mathbf{0}}$ is the explained part of the model, which is set equal to the mean of the dependent variable[2]. Then, the following procedure is applied for $i = 1, ..., m$:

1. For $j = 1, ..., r$, regress $\hat{\boldsymbol{f}}_{\boldsymbol{j}}$ on the current residuals $\hat{\boldsymbol{z}} - \boldsymbol{\mu}^{\boldsymbol{i-1}}$ to obtain estimated residuals $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{j}}$. Then, compute the sum of squared residuals as $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{j}}^{\top}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{j}}$ and denote the index of the factor with the lowest SSR as j_i^*.
2. Let $\hat{b}$ be the estimated coefficient obtained by regressing $\hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*}$ on $\hat{\boldsymbol{z}} - \boldsymbol{\mu}^{\boldsymbol{i-1}}$. Update $\boldsymbol{\mu}^i = \boldsymbol{\mu}^{i-1} + \nu\hat{b}\hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*}$, where $0 < \nu \leq 1$ is a step length parameter. Update $\hat{\beta}^i = \hat{\beta}^{i-1} + \nu\hat{b}\boldsymbol{e}_{\boldsymbol{j_i^*}}$, where $\boldsymbol{e}_{\boldsymbol{j_i^*}}$ is a unit vector with element j_i^* equal to one.
3. Compute the projection matrix of the selected factor as $P^i = \hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*}(\hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*}^{\top}\hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*})^{-1}\hat{\boldsymbol{f}}_{\boldsymbol{j_i}^*}$. Then, update $D^i = D^{i-1} + \nu P^i(I_T - D^{i-1})$, where I_T is an identity matrix with rank T. Finally, an information criterion suggested by Bai and Ng (2009) is computed as

$$\text{IC}_i = \ln\Big((\hat{\boldsymbol{z}} - \boldsymbol{\mu}^{\boldsymbol{i}})^{\top}(\hat{\boldsymbol{z}} - \boldsymbol{\mu}^{\boldsymbol{i}})\Big) + \frac{\ln(T)\cdot \text{df}^i}{T}, \tag{21}$$

where $\text{df}^i = \text{tr}(D^i)$ is the estimated degrees of freedom. This information criterion aims to select the best coefficient vector based on model fit and model complexity, which are captured by the squared residuals and the trace of $\boldsymbol{d}^{\boldsymbol{i}}$, respectively.

Following Kim and Swanson (2018), boosting is applied with step length parameter $\nu = 0.5$. The iteration limit is set to $m = 100$, which preliminary analysis showed to be sufficient to ensure the minimum IC is found. Let $i^* = \text{argmin}_i\, \text{IC}_i$, then set $\hat{\boldsymbol{\beta}}^{\text{Boosting}} = \hat{\boldsymbol{\beta}}^{\boldsymbol{i^*}}$ to make forecasts as

$$\hat{y}_{t+1}^{\text{Boosting}} = \hat{y}_{t+1}^{\text{AR}} + \hat{\boldsymbol{F}}_{\boldsymbol{t}}^{\boldsymbol{r}}\hat{\boldsymbol{\beta}}^{\text{Boosting}}. \tag{22}$$

[2] This is zero in this application, due to the inclusion of a constant in the AR(p) model.

4 Simulation Study

The simulation study serves to compare the functioning and the forecasting performance of the different shrinkage and variable selection methods when factors are used as explanatory variables. The methodology of the simulation study is discussed in Section 4.1 and the results are presented in Section 4.2.

4.1 Methodology

Assume there are N explanatory variables, T observations in the estimation sample, and v observations in the forecasting sample. Then, data is generated as follows:

1. Generate a random $N \times N$ matrix A with elements $a_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ for $i = 1, ..., N$ and $j = 1, ..., N$. Let $\Sigma = A^\top A$ be a random semi-positive definite $N \times N$ covariance matrix.
2. Generate $T + v$ observations of explanatory variables X from a multivariate normal distribution, with zero mean and covariance matrix Σ.
3. Compute the mean and standard deviation of the N variables in X using only the first T observations (estimation sample). Then, normalise the full matrix X using the mean and standard deviation obtained from estimation sample to get normalised matrix $\tilde{X}$.
4. Apply PCA to the covariance matrix of $\tilde{X}$ in the estimation sample to obtain a $N \times N$ matrix W of factor loadings. Thereafter, a $(T + v) \times N$ matrix of factors is obtained as $F = W\tilde{X}$.
5. Generate the dependent variable $y_t = \boldsymbol{F^{20}_{t-1}}\boldsymbol{\gamma} + \epsilon_t$ for $t = 2, ..., T + v$, where $\boldsymbol{F^{20}_{t-1}}$ is a 1×20 vector containing the first 20 factors at time $t - 1$, $\boldsymbol{\gamma}$ is a 20×1 coefficient vector, and $\epsilon_t \sim \mathcal{N}(9, 0)$.

To generate the data, $N = 50$ explanatory variables are used and the forecast sample contains $v = 100$ observations. Two different coefficient vectors are considered and the size of the estimation sample T varies to include 50, 100, and 1,000 observations. The first data generating process (DGP) uses $\boldsymbol{\gamma} = [3, 1.5, 0, 0, 2, 0, ..., 0]^\top$, such that only the first, second, and fifth factor are used to generate the dependent variable. The second DGP uses $\boldsymbol{\gamma} = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, ..., 0]^\top$, which corresponds to a coefficient of two for the first 10 factors and zero for the others.

After generating the data, the estimated factors are set equal to the generated factors $\hat{F} = F$. This is done as the goal of the simulation study is to compare the performance of shrinkage and variable selection methods and not to assess the performance of PCA. The standardised matrix $\tilde{F}$ is obtained by normalising matrix F using the mean and standard deviation of the first T observations. Afterwards, the models described in Section 3 are used to estimate coefficients with $r = 20$, which corresponds to the number of factors with coefficients in the DGP. Thereafter, each model makes one-step ahead forecasts for observations $T + 1, ..., T + v$, which are used to compute the mean squared prediction error (MSE) is computed as

$$\text{MSE} = \frac{1}{v} \sum_{t=T+1}^{T+v} (y_t - \hat{y}_t)^2. \tag{23}$$

For each configuration, the procedure outlined above is repeated 100 times. Given that the estimated and generated factors are equal, the coefficients from the regression models can directly be compared to the coefficients in the DGP. However, LARS, ridge regression, and elastic regularisation use standardised factors in the models, therefore coefficient estimates in terms of the original factors can be retrieved for by dividing the estimated coefficients by the standard deviation of the factors. The differences between coefficient estimates are investigated and the sensitivity to changes in parameters are explored.

The methods are compared using five performance measures. First, the average amount of non-zero coefficients in the model is computed. Second, the fraction of instances where the number of non-zero coefficients in the estimated model is exactly equal to the number of factors used in the DGP is measured. Third, the fraction of cases where factors that are in the DGP have non-zero coefficients is computed. Lastly, the forecasting accuracy in terms of MSE is examined, which is reported as a ratio to the benchmark AR(p) model.

The simulation study is performed using `MATLAB 2019a` and the code used can be found in Appendix A.

4.2 Results

Before discussing the results of the simulation, the functioning of the different factor models is illustrated using one iteration of the simulation with DGP 1 and $T = 100$. In this case, the dependent variable is generated using the first, second, and fifth factor with coefficients of three, one and a half, and two, respectively.

The least squares estimates of FAAR in Figure 1 are found to be fairly close around the true values. However, the coefficients of factors that are not included in the DGP are not exactly zero, which leads to noise when using this model for forecasting. The goal of the shrinkage and variable selection models is to reduce this noise by shrinking these coefficients towards zero or by excluding them from the model. As seen in Figure 2, bagging estimates are exactly the same as FAAR estimates for $c = 0$, as no coefficients and shrunk. The bagging estimator improves as c increases by excluding more factors that are not in the DGP. With the critical value set at 1.96, the bagging estimate includes one factor too many (factor 16), however, this factor has

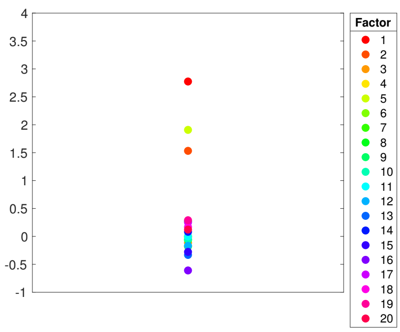


Figure 1: Estimated FAAR coefficients with true parameters values $[3, 1.5, 0, 0, 2, 0, ..., 0]$.

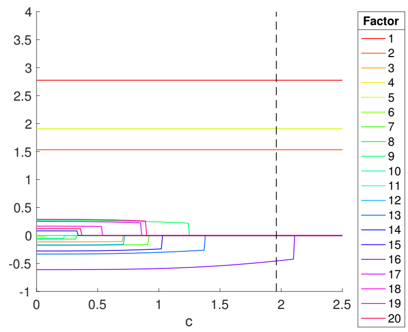
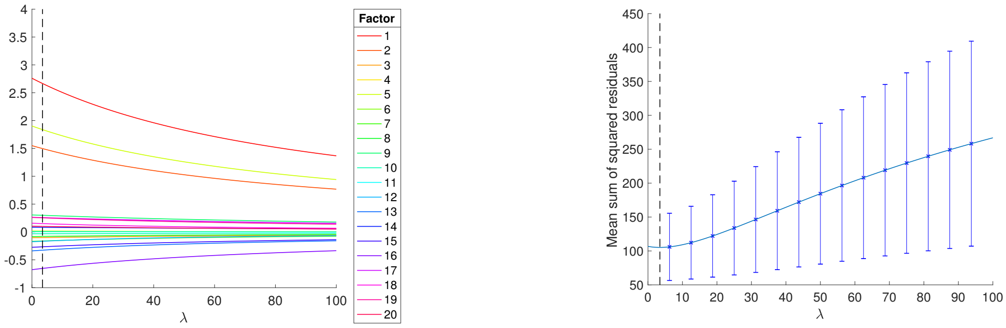


Figure 2: Estimated bagging coefficients as a function of c with true parameters values $[3, 1.5, 0, 0, 2, 0, ..., 0]$. The dashed black line indicates $c = 1.96$.

been shrunk slightly. The model's estimates could be improved by increasing the parameter c more in this case.

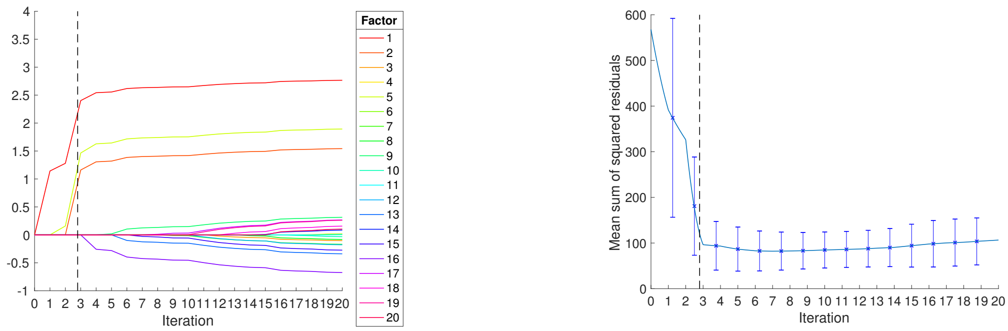
The coefficients obtained using ridge regression are shown in Figure 3a as a function of parameter λ. As the ridge parameter increases, all coefficients are gradually shrunk towards zero. This is a weakness of ridge regression provided that coefficients of factors that are included in the DGP also become smaller than their true values. In addition, the coefficients of factors that are not in the DGP only approximate zero. The optimal parameter value is $\lambda = 3.46$ as seen in Figure 3b, which leads to a small amount of shrinkage and coefficient estimates close to that of FAAR.



(a) Coefficients **(b)** Cross-validation

Figure 3: Estimated ridge coefficients in (a) as a function of λ with true parameters values $[3, 1.5, 0, 0, 2, 0, ..., 0]$. Average sum of squared residuals from tenfold cross-validation in (b) with standard deviations. The dashed black line indicates the optimal coefficients in (a) with the corresponding cross-validation error in (b).

The coefficient estimates of LARS are displayed in Figure 4a as a function of the number of iterations. As expected, the LARS algorithm iteratively includes the factors with the highest coefficient; the first factor is included in the first iteration, the fifth in the second, and the second in the third. At each iteration, the coefficients are jointly increased into their least squares direction. In the final iteration, the estimates resemble those of FAAR. As shown in Figure 4b, the lowest cross-validation error is found at around seven iterations. However, the optimal coefficient vector is selected to be that which takes the fewest iterations and is within

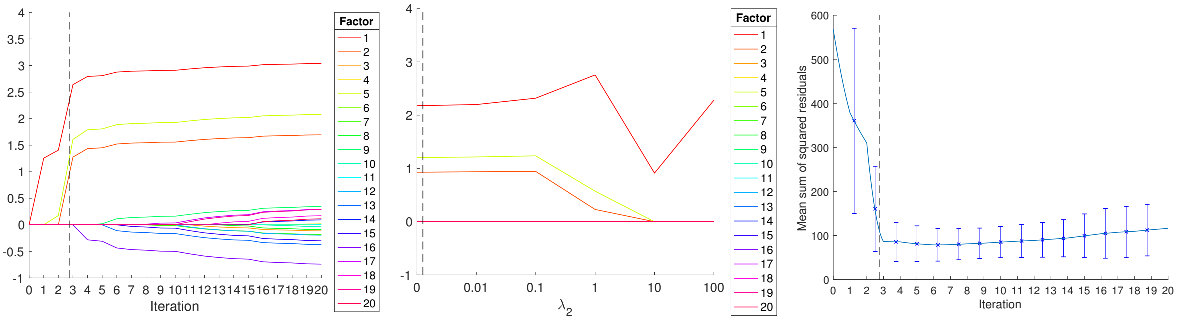


(a) Algorithm **(b)** Cross-validation

Figure 4: Estimated LARS coefficients in (a) as a function of the number of iterations with true parameters values: $[3, 1.5, 0, 0, 2, 0, ..., 0]$. Average sum of squared residuals from tenfold cross validation (b) with standard deviations. The dashed black line indicates the optimal coefficients in (a) with the corresponding cross-validation error in (b).

one standard deviation of the lowest average cross-validation error. In effect, the algorithm stops around the correct value of three iterations indicated by the dashed black line in Figure 4.

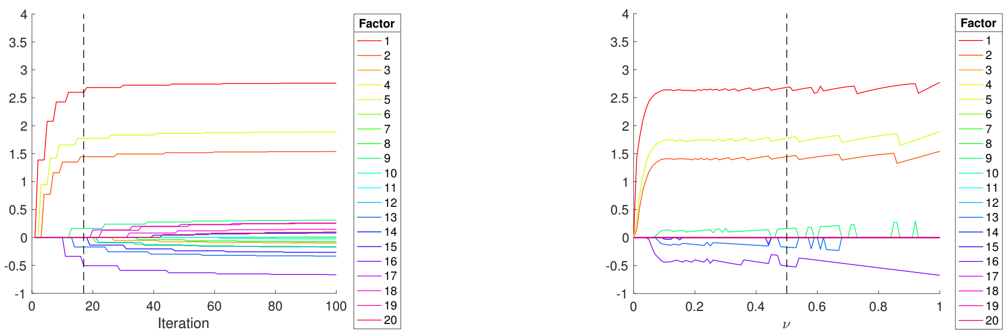
Estimated coefficients of elastic net regularisation are shown in Figure 5a and bear close resemblance to the LARS estimates displayed Figure 4a. An explanation for this is found in Figure 5b, where the optimal value of λ_2 is extremely close to zero. Therefore, elastic net regularisation practically reduces to lasso. However, as shown by Efron et al. (2004), lasso is a restricted version of LARS, hence elastic net regularisation produces almost identical estimates as LARS when λ_2 is close to zero. Moreover, it can be concluded from Figure 5b that using a small grid of λ_2 suffices, given that the coefficient estimates are stable for small values of this parameter.



(a) Algorithm **(b)** Coefficients **(c)** Cross-validation

Figure 5: Estimated elastic net regularisation coefficients in (a) as a function of the number of iterations with optimal the λ_2 and in (b) as a function of λ_2 with true parameters values $[3, 1.5, 0, 0, 2, 0, ..., 0]$. Average sum of squared residuals from tenfold cross-validation in (c) with standard deviations. The dashed black line indicates the optimal coefficients in (a), optimal λ_2 in (b), and the corresponding cross-validation error in (c).

Figure 6a shows coefficient estimates of boosting with $\nu = 0.5$. This figure shows great similarities with LARS and elastic net regularisation in terms of model construction. However, the key difference is that boosting increases only one coefficient at the time and makes smaller increments in the coefficients. Therefore, boosting requires more iterations and computation time. Figure 6a displays that as the algorithm proceeds variables are starting to be included with very small coefficients, which results from using a small step size. As shown in Figure



(a) Algorithm **(b)** Coefficients

Figure 6: Estimated boosting coefficients in (a) as a function of the number of iterations with the selected ν and in (b) as a function of ν with true parameters values: $[3, 1.5, 0, 0, 2, 0, ..., 0]$. The dashed black line indicates the optimal coefficients in (a) and $\nu = 0.5$ in (b).

6b, changes in parameter ν do not have major consequences as long as it is above 0.1. This is because the optimal coefficient vector for each value of ν is selected with an information criterion that depends on the size of coefficients and not on the number of iterations.

As we are now acquainted with the functioning of the different methods, we proceed by looking at general results obtained after 100 iterations with different DGPs and sample sizes, which are displayed in Table 1. First, Panel A shows LARS and elastic net regularisation are the best methods in terms of variable selection. For these methods, the average number of factors selected is close to the correct value of three and the fraction of instances where only the factors from the DGP are included is closest to one. Also, these two methods always include factors one, two, and five in the model when the sample size is at least $T = 100$. Both FAAR and ridge regression have non-zero coefficients for all factors, whereas bagging and boosting include too many (more than three) factors on average. This could be improved upon by using a larger c for bagging or increasing the penalty for model complexity in the information criterion used by boosting. However, it turns out that these two models perform best in terms of forecasting accuracy, with bagging having the lowest MSE overall. LARS and elastic net regularisation perform worst in terms of predictive accuracy, possibly because the cross-validation procedure is aimed at improving variable selection rather than forecasting accuracy. As expected, variable selection and forecasting accuracy improve greatly when the sample size increases.

Table 1: Performance of variable selection and forecasting with simulated data.

Panel A: DGP 1

	$T = 50$				$T = 100$				$T = 1000$			
	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio
AR(1)				1.00				1.00				1.00
Rand. Walk				1.92				1.98				1.99
FAAR	20.00	0.00	1.00	0.27	20.00	0.00	1.00	0.21	20.00	0.00	1.00	0.16
Ridge	20.00	0.00	1.00	0.29	20.00	0.00	1.00	0.23	20.00	0.00	1.00	0.16
LARS	3.13	**0.86**	0.96	0.32	3.07	0.95	1.00	0.26	3.00	1.00	1.00	0.17
Elastic Net	**3.12**	0.83	0.96	0.31	**3.04**	**0.97**	1.00	0.25	3.00	1.00	1.00	0.17
Bagging	4.19	0.36	1.00	**0.21**	3.95	0.38	1.00	**0.19**	4.01	0.38	1.00	**0.16**
Boosting	6.17	0.07	1.00	0.24	6.03	0.06	1.00	0.20	4.32	0.25	1.00	0.16

Panel B: DGP 2

	$T = 50$				$T = 100$				$T = 1000$			
	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio	Avg. # factors	Corr. # factors	Corr. incl.	MSE ratio
AR(1)				1.00				1.00				1.00
Rand. Walk				1.95				1.95				2.01
FAAR	20.00	0.00	1.00	0.13	20.00	0.00	1.00	0.10	20.00	0.00	1.00	0.08
Ridge	20.00	0.00	1.00	0.16	20.00	0.00	1.00	0.11	20.00	0.00	1.00	0.08
LARS	11.98	0.06	0.96	0.19	10.76	0.52	1.00	0.13	10.03	0.97	1.00	0.09
Elastic Net	11.90	0.08	0.96	0.18	10.44	**0.72**	1.00	0.13	**10.00**	**1.00**	1.00	0.09
Bagging	**10.54**	**0.61**	1.00	**0.11**	**10.42**	0.68	1.00	**0.09**	10.52	0.56	1.00	**0.08**
Boosting	13.47	0.03	1.00	0.15	12.47	0.08	1.00	0.11	11.87	0.18	1.00	0.08

Data is generated as $y_t = \boldsymbol{F}_{t-1}^{20}\boldsymbol{\gamma} + \epsilon_t$, with different coefficient vectors and $\epsilon_t \sim \mathcal{N}(0, 9)$. Twenty factors $\boldsymbol{F}_t^{20}$ are obtained by applying PCA to 50 normalised variables X with zero mean and covariance matrix $\Sigma = A^\top A$, where A is a random matrix will standard normally distributed elements. DGP 1 uses $\boldsymbol{\gamma} = [3, 1.5, 0, 0, 2, \overbrace{0, ..., 0}^{15}]^\top$ and DGP 2 uses $\boldsymbol{\gamma} = [\overbrace{1, ..., 1}^{10}, \overbrace{0, ..., 0}^{10}]^\top$. An estimation sample of size T is used to estimate coefficients. The autoregressive order of the AR(p) model is determined using BIC selection with $1 \leq p \leq 5$. The table shows the average amount of factors included in the model, the fraction of instances in which the number of factors included in the model is equal to the number of factors included in the DGP, and the fraction of instances in which the factors in the DGP are included in the model. Estimated coefficients are used to make 100 one-step ahead forecasts, providing MSE estimates. Best results per sample size and DGP are indicated in bold.

Panel B of Table 1 shows simulation results when the dependent variable is generated by the first 10 factors with coefficients of two. By including more factors in the DGP, the accuracy of variable selection has decreased for all methods with the exception of bagging. Bagging performs particularly well in this case as all coefficients are relatively large in magnitude, such that they are clearly significant based on t-statistics. However, as T increases LARS and elastic net regularisation again prove to be the best variable selection techniques, as the average amount of factors included is close to ten and the fraction in which only ten factors are included approaches to one. Bagging performs best in terms of forecast accuracy, closely followed by FAAR. FAAR performs better in this setting, because in the first DGP 17 out of 20 factors in the model only added noise compared to 10 out 20 for the second DGP. For the same reason, ridge regression performs relatively better in this simulation. Again, the forecasting accuracy increases and converges across methods as the sample size increases.

It can be concluded from the simulation study that each method has its strengths and weaknesses. First, FAAR always includes all variables, implying it performs well when many of the variables considered have explanatory power. On the other hand, this leads to noisy estimates if this is not the case. For ridge regression largely the same holds as for FAAR, however, it can reduce forecast variance by shrinking coefficients at the cost of increased bias. Both LARS and elastic net regularisation fare well in terms of variable selection, however, the coefficients are generally (too) small. Bagging can effectively include variables that significantly impact the dependent variable, yet it could disregard variables with small t-statistics. This could be the case if explanatory variables either have large variance or only a minor effect on the dependent variable. Boosting seems to work well generally, however, it is prone to overfitting and by incorporating many variables in the model the forecast variance becomes large.

5 Forecasting Nigerian GDP growth

The simulations showed shrinkage and variable selection methods may effectively enhance forecasting accuracy, although there is no clear method of preference. In this section, it is investigated whether the same methods can be successfully applied in the context of forecasting real Nigerian GDP growth. The data used for this purpose is examined in Section 5.1, the methodology is explained in Section 5.2, and the results are discussed in Section 5.3.

5.1 Data

Two data sets are used to forecast the economic growth in Nigeria. Section 5.1.1 describes the first data set consisting of real annual GDP growth rates in Africa and examines the growth pattern of Nigeria. Thereafter, a second data set that includes numerous economic indicators is discussed in Section 5.1.2.

5.1.1 African GDP Growth

The first data set includes real GDP growth data of 52 African countries from 1963 until 2016, which is provided by Franses and Vasilev (2019). These countries include all African countries, except for eSwatini (former Swaziland) and South Sudan. The entire time series is retrieved from the WDI database of the World Bank for 34 countries. For the remaining 18 countries, economic growth rates are predominantly missing in the first two decades of the sample. These missing growth rates are imputed by PCA. More specifically, demographic, production, and financial data is used to construct factors for each category. Subsequently, real GDP growth rates are regressed on the categorical factors and the estimated coefficients are used to impute real GDP growth. A full list of countries with summary statistics can be found in Table 6 in Appendix B.

The data is used to forecast Nigerian GDP growth, therefore a metric of interest is the correlation between the economic growth of African countries and the economic growth of Nigeria in the subsequent year. As shown in Figure 7, this correlation is predominantly positive, albeit small with an average correlation of 0.04. Uganda and Sierra Leone are most positively correlated with coefficients of 0.32 and 0.30, respectively. On the other hand, Eritrea and Zimbabwe are most negatively correlated with coefficients of -0.22 and -0.19. In line with the findings of Arbache and Page (2007), neither other large oil-producing countries nor countries around Nigeria have higher correlations than other African countries. Instead, the highest correlations are found in southeastern Africa.

The economic growth rates of Africa and Nigeria over time are summarised in Table 2. On average, economic growth in Africa was 4.0% from 1963 until 2016. Growth has accelerated recently, with the average economic growth rate from 2000 until 2016 amounting to 4.5% per

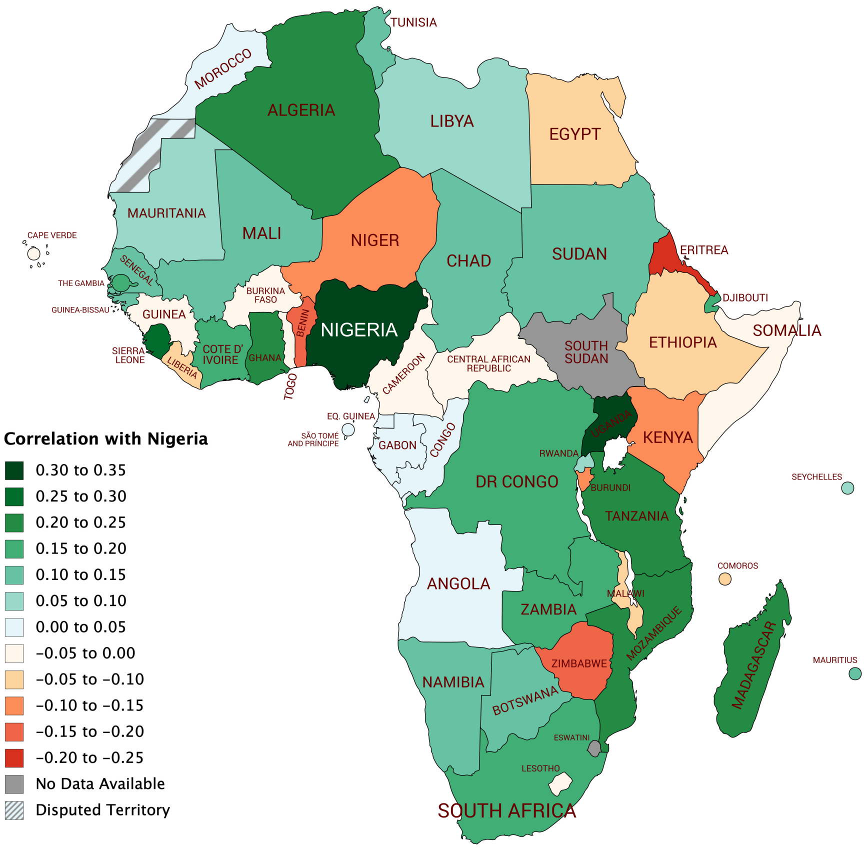


Figure 7: One year lagged correlation between economic growth of African countries and Nigeria from 1963 until 2016. Figure created with MapChart using data retrieved from Franses and Vasilev (2019).

Table 2: Annual real GDP growth rate (%) of Africa and summary statistics for Nigeria from 1963-2016.

	Africa Average		Nigeria		
Time Period	GDP Growth	Correlation	GDP Growth	Minimum	Maximum
1963 - 1981	4.31 (6.07)	0.05	4.35 (10.60)	-15.70	25.00
1982 - 1999	3.16 (6.16)	0.01	1.16 (5.83)	-10.80	12.80
2000 - 2016	4.50 (4.59)	-0.06	7.00 (7.36)	-1.60	33.70
1963 - 2016	3.98 (6.31)	0.04	4.12 (8.43)	-15.70	33.70

Standard deviations are indicated in parentheses and computed as the average of the standard deviations of individual countries. The correlation is the average one year lagged correlation of economic growth of African countries with Nigeria. Nigeria is excluded from the average correlation.

year. Moreover, fluctuations in economic growth rates have decreased, as the average standard deviation of economic growth dropped from 6.2% per year in the period 1982-1999 to 4.6% per year in the period 2000-2016. The standard deviations remain high relative to the mean, indicating economic growth is still largely unstable. Economic growth rates in Africa differ greatly across countries. For example, Equatorial Guinea realised the highest average growth rate with 12.7%, whereas Libya's economy grew on average with 0.3% per year.

Comparing Nigeria's economic growth to the African average, it can be seen in Table 2 that the average over the whole sample is similar. However, Nigeria's growth is more volatile with a large standard deviation of 8.4%. Interestingly, from 1982 until 1999 the economic growth of Nigeria is 2.5% lower than the African average, whereas it is 2.5% higher than the African average in the period from 2000 until 2016. The causes of the large variation of Nigeria's economic performance over time are discussed next.

One of the most important factors in explaining growth rates in Nigeria, especially at the start of the sample period, is the occurrence of political events. In 1966, the government was overthrown by a military coup and a counter-coup took place later that year, which likely initiated the economic decline in 1966 visible in Figure 8. Following these events, the east of Nigeria declared independence and formed the Republic of Biafra in 1967. A civil war started between Nigeria and Biafra in the same year, which lasted until 1970. The initial decline in economic growth seen in Figure 8 is largely attributable to the exclusion of economic activity on Bifrian territory from the statistics (Nafziger, 1972). However, Nigeria reconquered territory

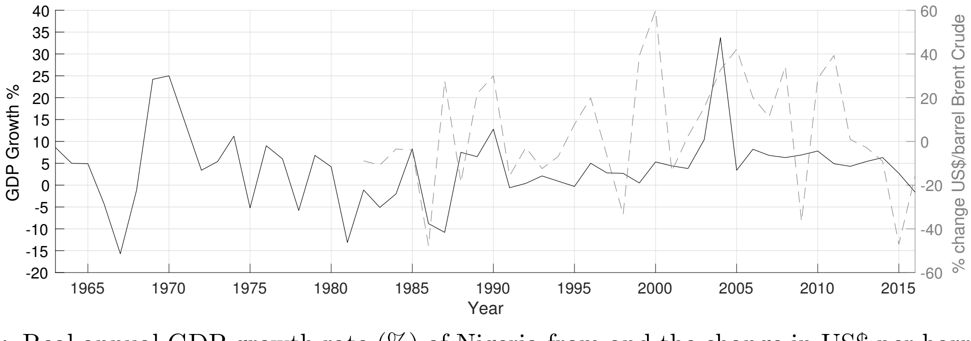


Figure 8: Real annual GDP growth rate (%) of Nigeria from and the change in US$ per barrel of Brent Crude oil (%) from 1963-2016. Data retrieved from the World Bank's WDI database and the FRED Economic Database of the Federal Reserve Bank of St. Louis.

rapidly during the war, explaining the sharp increase in GDP in 1969 and 1970. After this period, the political environment remained unstable. Two other noteworthy events include coup in 1975 and a failed coup attempt in 1976, which possibly harmed economic growth. Nigeria has only become a stable democracy in 1999, which likely contributed to more steady economic growth in the 21st century.

Besides political forces, Nigeria is also subject to oil shocks. As seen in Figure 9, crude petroleum and natural gas contributed to one-third of all economic value in 1981, but also before this period oil was the most important driver of the economy. Therefore, increased oil prices during the first oil crisis during 1973-1974 and the second oil crisis starting in 1979 enhanced economic growth during these periods. The relatively low economic growth in the sub-period from 1982 until 1999 is partly attributable to the 1980s oil glut, as oil remained the most important source of economic value. Nigeria's economy changed substantially after the turn of the century. The tertiary sector grew greatly, whereas the share of crude petroleum and natural gas production decreased. Figure 9 shows the latter sector accounted for 29% of real GDP in 1999, which starkly contrasts with 6% in 2016. Yet, this difference is overstated as oil and natural gas account for more than 90% of exports and trade activity has increased significantly. In addition, the oil price was relatively low in 2016. Therefore, a large drop in the oil price around 2015 shown in 8 still largely explains the first economic decline after prolonged economic expansion.

A noticeable observation in Figure 8 is that of 2004, where the real GDP growth rate amounts to 33.7%. Despite high growth during the last sub-sample, this magnitude of growth is implausible as no remarkable events occurred. As shown by Jerven (2016), this statistical outlier is a result of a data revision. Before the revision, real GDP growth was computed with 1990 prices and the growth rate was estimated to be 10%. The revised rates are computed with 2010 prices, which resulted in an estimated 89% increase in GDP and an excessive growth in 2004 (Jerven, 2016). This observation has a significant impact on summary statistics displayed in Table 2. Excluding this observation from the last sub-sample makes the economic growth appear a lot more stable; the annual growth rate reduces from 7.0% to 5.3% and the standard deviation decreases from 7.4% to 2.7%.

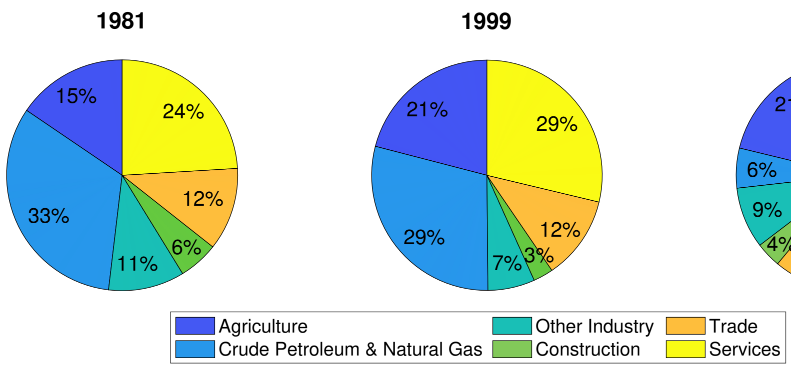


Figure 9: Sectors contributing to real GDP growth in Nigeria in 1981, 2000, and 2016. Data retrieved from the 2018 Statistical Bulletin of the Central Bank of Nigeria.

5.1.2 Economic Indicators

The second data set includes 35 economic indicators and ranges from 1982 until 2016. The data is predominantly extracted from the WDI database and from the 2018 Statistical Bulletin of the Central Bank of Nigeria. Two variables are retrieved from the Edstats Query database of the World Bank and the FRED Economic database of the Federal Reserve Bank of St. Louis. The economic indicators include official and real exchange rates, five loan and interest rates variables, three domestic saving and investment variables, four variables relating to human capital and labour, the M1 and M2 money supply, the Nigerian stock market, consumer inflation, government and household consumption expenditure, six real output variables for different sectors, FDI, four import and export variables, the global price of Brent crude oil, and the real GDP of India, the United States, and the world.

The partitioning of the data set is based on the methodology of Banerjee and Marcellino (2006), however, the number of indicators used is restricted by the data availability for Nigeria in this time span. For example, no data is included relating to economic sentiment, employment, new orders, productivity, or wages. On the other hand, several country-specific variables are included. For example, FDI is added as this variable positively influences growth in Nigeria (Akinlo, 2004). Above that, the oil price is included as it greatly affects real output. Moreover, government expenditure on health and government expenditure on education are included as separate variables as they have divergent effects on economic growth (Nurudeen and Usman, 2010). Lastly, the GDPs of Nigeria's current most important trade partners, India and the United States, are added.

Some of the time series include missing or aberrant observations, which in most cases are replaced with the mean of the previous and subsequent year. Nominal and real values, as well as indices, are transformed by computing the percentage growth, whereas the difference between consecutive years is taken for the inflation rate and any interest rate (spread) to make the time series stationary. A detailed overview of the economic indicators and the data adjustments is provided in Appendix B.

5.2 Methodology

To model economic growth in Nigeria, four different sets of variables X are considered:

1. African GDP growth data from 1963 until 2016, where X includes $N = 51$ consisting of the economic growth rates of the African countries excluding Nigeria. One year of lagged explanatory variables or factors is included in each of the models.
2. African GDP growth data from 1963 until 2016, where X includes the same $N = 51$ variables. Four years of lagged explanatory variables or factors are included in each of the models. This leads to $51 \times 4 = 254$ explanatory variables and $r \times 4$ factors to be considered, where r is the number of factors selected. This is done as more distant years may be informative in case the Nigerian economy lags behind several years on other African economies. The factors loadings are estimated as above, using one year of data.

3. Economic indicator data from 1982 until 2016, where X includes $N = 35$ variables. One year of lagged explanatory variables or factors is included in each of the models.
4. African GDP growth data and economic indicator data from 1982 until 2016, where X includes $N = 84$ variables. One year of lagged explanatory variables or factors is included in each of the models.

Based on the fitting of an AR(1) model several aberrant observations are found. These observations are treated by including indicator variables in the AR(p) model, FAAR, ridge regression, and bagging[3]. For the first and second sets of explanatory variables, dummy variables are included for the period from 1966 up to and including 1970, which covers the period of the Nigerian civil war and one year leading up to it. For all data sets an indicator variable for 2004 is used, as this observation is found to be a statistical outlier.

The procedure to handle the data is comparable to the way data is generated in the simulation study. Forecasts for a given year are made with explanatory variables X, which contains all data up to the year to be predicted. Forecasts are made as follows:

1. Compute the mean and standard deviation of the N variables in X to obtain normalised matrix $\tilde{X}$.
2. Apply PCA to the covariance matrix of $\tilde{X}$ to obtain a $N \times N$ matrix W of factor loadings. Factors are then estimated as $\hat{F} = W\tilde{X}$.
3. Find the number of factors r to be included using information criterion (6) proposed by Bai and Ng (2002).
4. Compute the mean and standard deviation of the r variables in $\hat{F}^r$ to obtain normalised matrix $\tilde{F}^r$.
5. Compute the coefficients using factors $\hat{F}^r$ and $\tilde{F}^r$ for all models as outlined in Section 3. Moreover, compute coefficients with the N explanatory variables directly using X for boosting and using $\tilde{X}$ for LARS and elastic net regularisation.
6. Use coefficient estimates to make one-step ahead forecasts with each of the methods.

The goal of this paper is to investigate the usefulness of shrinkage and variable selection methods in forecasting contemporary Nigerian real GDP growth, therefore predictions are made for the years 2012 until 2016. After forecasts have been made, the MSE is computed with the five years of forecasts using (23). This measure is used again used to assess predictive accuracy with the AR(p) model as a benchmark. Above that, it is examined which variables or factors are most important in the different models.

One should note that an expanding window is used to estimate the coefficients, which is done to make use of all available information. This implies that factor loadings W may change over time as they are re-estimated for each forecast. Furthermore, explanatory variables are also used directly to make predictions, which helps to assess the benefits of a factor-based approach in forecasting. Only LARS, elastic net regularisation, and boosting are used without factors, because they can, in contrast to the other methods, include more variables than observations

[3]The other methods make use of estimated AR(p) residuals, therefore they are indirectly treated.

in the model. In total eleven different models are considered: AR(p) and random walk as benchmarks; LARS, elastic net regularisation, and boosting with explanatory variables; FAAR, ridge regression, LARS, elastic net regularisation, bagging, and boosting with factors.

The forecasts are made using `MATLAB 2019a` and the code is provided in Appendix A.

5.3 Results

In this section, the performance of shrinkage and variable selection methods is discussed with the application of forecasting Nigeria's economic growth. To construct forecasts, African GDP growth data is used in Section 5.3.1, economic indicator data is used in Section 5.3.2, and the two data sets are jointly used in Section 5.3.3. This section only reports key results; details on estimated coefficients and factors loadings can be found in Appendix C.

5.3.1 Forecasting using African GDP Growth Data

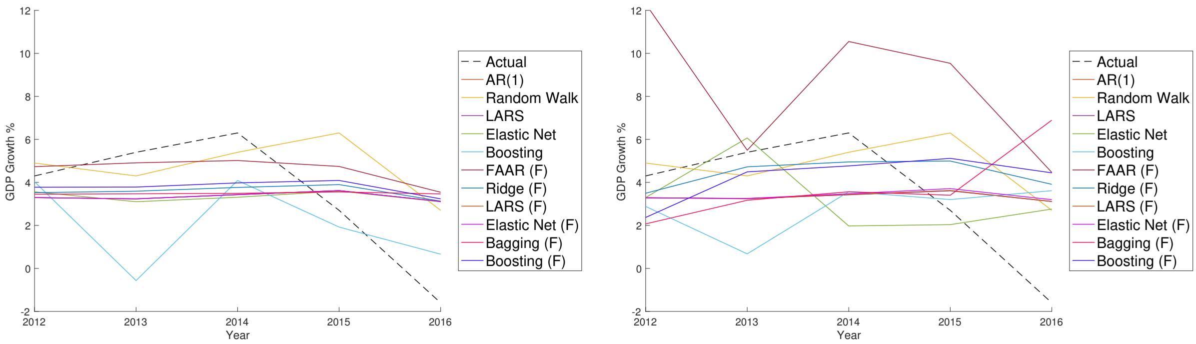
Table 3 shows the performance of the different methods when forecasting Nigerian real GDP growth using African real GDP growth data. The benchmark autoregressive model includes one lag with an autoregressive coefficient of 0.13 and a constant term of 2.75. The random walk model performs better, achieving a 9% lower MSE.

First, we examine the left-hand side of Table 3, in which only the previous year of data is used for forecasting. In this setting, variable selection methods with explanatory variables themselves do not forecast better than the benchmark models. Both LARS and elastic net regularisation include Uganda in the model, with coefficients of 0.00 and 0.20, respectively. As seen in Figure 10a, the low coefficients provide forecasts that are extremely close to that of the AR(1) model.

Table 3: Performance of models forecasting Nigerian GDP growth from 2012-2016 using African GDP growth data from 1963-2016.

		One lag			**Four lags**		
		# variables/ factors incl.	MSE	MSE ratio	# variables/ factors incl.	MSE	MSE ratio
Benchmark	AR(1)		7.40	1.00		7.40	1.00
	Random Walk		6.77	0.91		**6.77**	**0.91**
Variable	LARS	1.00	7.40	1.00	1.00	7.38	1.00
	Elastic Net	1.00	7.53	1.02	1.00	7.92	1.07
	Boosting	15.60	9.26	1.25	48.60	11.85	1.60
Factor	FAAR	6.80	**6.53**	**0.88**	27.20	33.12	4.48
	Ridge	6.80	6.82	0.92	27.20	7.71	1.04
	LARS	1.00	7.39	1.00	1.00	7.40	1.00
	Elastic Net	1.00	7.39	1.00	1.00	7.54	1.02
	Bagging	0.00	7.74	1.05	0.20	18.01	2.44
	Boosting	1.00	6.72	0.91	2.00	9.86	1.33

One-step ahead forecasts are made for the years 2012 until 2016 using an expanding window. The autoregressive order of AR(p) is determined using BIC selection with $1 \leq p \leq 5$. The number of explanatory variables and factors shown is the mean from five coefficient estimates. Factors are obtained by PCA on the estimation sample consisting of 51 real GDP growth rates of African countries. The amount of factors considered is selected using an adjusted BIC with a maximum of 20. The results on the left-hand side are obtained by including one lag of variables or factors in the model, whereas the right-hand side includes four lags. The best results are indicated in bold.



(a) One lag **(b)** Four lags

Figure 10: Nigerian GDP growth forecasts from 2012-2016 made using using African GDP growth data. The models used to make forecasts include one lag of variables or factors in (a) and four lags in (b). The F in parentheses indicates the model uses factors to make forecasts, whereas the other models use explanatory variables directly.

These low coefficients are not unexpected, as the correlations between the economy of African countries and are generally low. Boosting includes 15.6 countries in the model on average, which is indicative of overfitting. Figure 10a shows boosting forecasts are far off and extremely volatile.

Factor models include eight factors for the first three years and six factors for the last two years of forecasts. Although the number of factor changes, the factor loadings of five out of six factors are almost constant. In the last year of forecasts, six factors explain 42.1% of the total variance in the data, indicating there are some commonalities in the growth rates of African countries. Factor models on the left-hand side of Table 3 perform substantially better than models that use variables directly. FAAR has the best forecasting accuracy with a MSE 12% lower than the autoregressive benchmark and 4% lower than the random walk model. The largest coefficients of FAAR are -0.70 for the first factor, -0.38 for the fifth factor, and 0.36 for the sixth factor. The interpretation of these factors is not directly clear as they include 51 loadings. The first factor mainly has large negative loadings for countries in central and south Africa and a sole large positive loading for Eritrea, whereas the other two factors are very diverse. As shown in Figure 10a, the GDP growth forecasts of FAAR are higher than that of other factor models. Ridge regression produces similar forecasts, although it has slightly smaller coefficients than FAAR. LARS and elastic net regularisation both have a minuscule coefficient for the first factor. On the other hand, bagging does not include any factors at all, wherefore the forecasts are analogous to that of the AR(1) model. Boosting has a coefficient of -0.23 for the (standardised) first factor, which results in above-average forecasting accuracy. However, Figure 10a indicates none of the factor models forecasts the actual decline in economic growth.

On the right-hand side of Table 4 the results are shown where four lags of variables or factors are included in the model. Again, variables selection methods that use explanatory variables directly are found to be ineffective. LARS and elastic net regularisation now include Uganda at the second lag instead of the first, which does not lead to better forecasting accuracy. Boosting includes even more variables than before, which again results in poor forecasting performance.

The factors used with four lags are the same as in the previous setting, as they are computed with the same data. However, the factor models now consider four times as many factors, which translates to 32 and 24 factors for the first three and last two forecasts, respectively.

Interestingly, the first factor still has the highest coefficient in all models, although this factor is included at the second lag for FAAR and ridge regression, whereas it is included at the third lag for LARS, elastic net regularisation, and boosting. However, as indicated by the MSE ratios, all models perform worse than the benchmark models. Moreover, the forecasts with one year of data in Figure 10b vary a lot more than those in Figure 10a, due to the inclusion of four times as many variables. The forecasts in 10b are generally also more off, showing the inclusion of additional years of data is ineffective.

In general, it can be concluded the predictive power of the African GDP growth data set is low, both with and without the use of factors. Yet, the factor-based approach is slightly preferred when one lag is included. A likely explanation for the poor forecasting performance of the models includes that Nigeria is leading instead of lagging other economies. Above that, Nigeria's main trade partners are located outside of Africa, such that trade is hardly affected by economic developments within Africa.

5.3.2 Forecasting using Economic Indicator Data

Table 4 shows the forecasting performance using economic indicator data. The optimal autoregressive order is again one, however, the autoregressive coefficient is higher with 0.24 and a lower constant of 2.17. This indicates economic growth in Nigeria has become more persistent in recent years, as the sample starts 19 years later compared to the African GDP growth data. Nonetheless, the predictive accuracy of the autoregressive model has deteriorated marginally, therefore, the random walk model again outperforms the AR(1) model.

Variable selection methods LARS and elastic net regularisation both select the economic activity in the construction sector in real terms as sole explanatory variables, with coefficients of 0.02 and 0.50, respectively. The use of this variable as an economic indicator is somewhat

Table 4: Performance of models forecasting Nigerian GDP growth from 2012-2016 using economic indicator data from 1982-2016.

		# vars/factors incl.	MSE	MSE ratio
Benchmark	AR(1)		7.50	1.00
	Random Walk		6.77	0.90
Variable	LARS	1.00	7.48	1.00
	Elastic Net	1.00	6.99	0.93
	Boosting	22.00	11.29	1.50
Factor	FAAR	20.00	20.65	2.75
	Ridge	20.00	**5.68**	**0.76**
	LARS	1.00	7.51	1.00
	Elastic Net	1.00	7.78	1.04
	Bagging	4.20	7.40	0.99
	Boosting	10.60	6.96	0.93

One-step ahead forecasts are made for the years 2012 until 2016 using an expanding window. The autoregressive order of AR(p) is determined using BIC selection with $1 \leq p \leq 5$. The number of explanatory variables and factors shown is the mean from five coefficient estimates. Factors are obtained by PCA on the estimation sample consisting of 35 Nigerian economic indicators. The amount of factors considered is selected using an adjusted BIC with a maximum of 20. One lag of variables or factors is included in the model. The best results are indicated in bold.

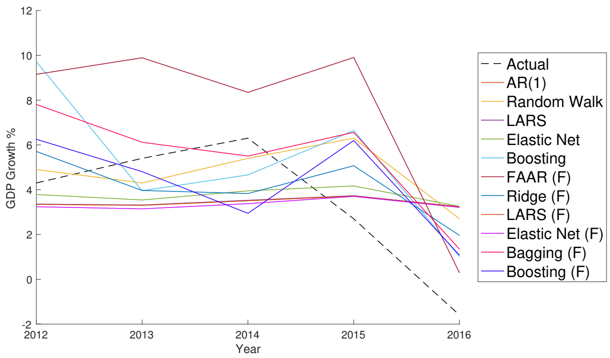


Figure 11: Nigerian GDP growth forecasts from 2012-2016 made using economic indicator data. The models used to make forecasts include one lag of variables or factors. The F in parentheses indicates the model uses factors to make forecasts, whereas the other models use explanatory variables directly.

effective, as elastic net regularisation outperforms the autoregressive model. This finding is not surprising, as particularly housing activity leads the state of the economy (Penm and Terrell, 1994). However, as indicated in Figure 11, the forecasts are fairly comparable to that of the autoregressive model. Boosting includes many variables, with the largest coefficients being 0.60 for the 3-month deposit rate, 0.51 for the interest rate spread, -0.33 for the lending interest rate, and 0.19 for construction. The positive coefficients for the deposit and interest rate are surprising, as increases in these variables are expected to affect the economy negatively. As shown in Figure 11, the forecasts of this model are inaccurate, especially in 2012.

For the factor models, the maximum of 20 factors is selected for all five forecasts. The first five factors explain more than half of the total variation in the data, which is expected as many of the economic indicators are strongly interrelated. The 20 factors together explain 96.6% of the total variance in the data. The loadings of the first few factors remain largely constant over the five forecasts, however, factors 15 until 20 change greatly. As a result, the information used by factor models changes from one year to year. The factor-based approach with economic indicator data produces mixed results. The FAAR model has large coefficients for many factors, which leads to volatile and poor estimates. On the other hand, ridge regression does better in terms of forecasting by shrinking the coefficients of FAAR. In fact, it is the only model to beat both benchmark models. The largest coefficients of ridge regression are for factors 1, 14, and 15. The first factor has high positive coefficients for variables relating to trade, the money supply, the stock market, and total government expenditure. Gross capital formation, the stock market, and India's GDP are most present in factor 14, whereas factor 15 is very diverse. As seen in Figure 11, LARS and elastic net regularisation with factors again provide similar forecasts as the AR(1) model, due to minute coefficients for the first factor. Bagging includes factors 1, 14, and 15 in the model with coefficients slightly smaller than FAAR, but larger than ridge regression. In effect, the bagging forecasts are between the forecasts made by these two models (see Figure 11). Boosting includes the same three factors, but also has sizeable coefficients for around eight other factors. The boosting model beats the autoregressive benchmark, but not the random walk model in terms of predictive accuracy.

Overall, a distinct difference between the forecasts in Figure 10 and 11 is that various models that use economic indicator data are able to predict a decrease in real GDP growth in Nigeria, whereas this was not the case with models that use African GDP growth data. This may partially be attributed to a higher autoregressive term in the AR(1) model. With economic indicator data, factor models generally perform better than models that use explanatory variables directly. Ridge regression is the only model to beat both benchmarks, which suggests the predictive power of economic indicators is limited. This is in contrast to the findings of Gupta and Kabundi (2011), who show this type of data is improves short-term forecasts of South African economic growth. The difference in results may be caused by economic indicator data being less effective for less developed countries. Moreover, the number of economic indicators used in this research could be too modest to make accurate forecasts.

5.3.3 Forecasting using Combined Data

Both the African GDP growth data and the economic indicator data are able to provide some, but limited explanatory power. In both cases a different model is found to be optimal, however, in both cases it is a factor model. We next examine whether a combination of the two data sets can provide improved forecasts of real Nigerian GDP growth. The joint data set consists of 86 variables, including GDP growth data from 51 countries and 35 economic indicators. The results with combined data are shown in Table 5. The autoregressive model is the same as in Section 5.3.2, given that the data sample also ranges from 1982 until 2016.

First, looking at the variable selection methods, LARS and elastic net regularisation include activity in the construction sector as the sole explanatory variable. This is in line with the results of Section 5.3.2, although the coefficients are smaller. Boosting includes 22 variables, with the largest coefficients being 0.44 for South Africa, -0.38 for Niger, -0.28 for Mauritius, and 0.22 for construction. Despite the reason for including these three countries being unknown, Figure 12 shows the forecasts are surprisingly accurate. Boosting performs 30% better than the autoregressive benchmark in terms of the MSE, due to good predictions from 2013 until 2016.

Table 5: Performance of models forecasting Nigerian GDP growth from 2012-2016 using African GDP growth and economic indicator data from 1982-2016.

		# vars/factors incl.	MSE	MSE ratio
Benchmark	AR(1)		7.50	1.00
	Random Walk		6.77	0.90
Variable	LARS	1.00	7.50	1.00
	Elastic Net	1.00	7.54	1.00
	Boosting	32.20	**5.22**	**0.70**
Factor	FAAR	11.80	13.67	1.82
	Ridge	11.80	7.28	0.97
	LARS	1.00	7.50	1.00
	Elastic Net	1.00	7.50	1.00
	Bagging	0.40	5.61	0.75
	Boosting	2.40	5.92	0.79

One-step ahead forecasts are made for the years 2012 until 2016 using an expanding window. The autoregressive order of AR(p) is determined using BIC selection with $1 \leq p \leq 5$. The number of explanatory variables and factors shown is the mean from five coefficient estimates. Factors are obtained by PCA on the estimation sample consisting of 51 African GDP growth rates and 35 Nigerian economic indicators. The amount of factors considered is selected using an adjusted BIC with a maximum of 20. One lag of variables or factors is included in the model. The best results are indicated in bold.

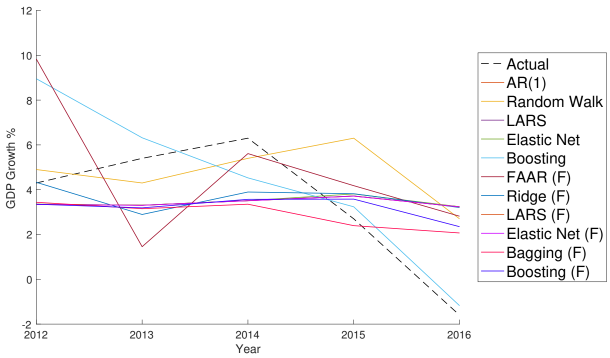


Figure 12: Nigerian GDP growth forecasts from 2012-2016 made using African GDP growth and economic indicator data. The models used to make forecasts include one lag of variables or factors. The F in parentheses indicates the model uses factors to make forecasts, whereas the other models use explanatory variables directly.

However, GDP growth in 2012 is greatly overestimated, which brings the model's robustness into doubt.

Continuing with factor models, forecasts for the years 2012 until 2016 are made with 14, 13, 11, 11, and 10 factors, respectively. For the final forecast, the 10 selected factors explain 61.8% of the total variance in the data. The performance of the factor models varies once more. FAAR produces poor forecasts due to the inclusion of all factors. The highest coefficients of FAAR are for the first, second, and seventh factor. The interpretation of the factors is particularly cumbersome as they are formed by a mixture of African countries and Nigerian economic indicators. However, the first factor shows great similarities to the first factor of the African growth data, although it is augmented with real GDP variables. The second factor corresponds to the difference between the first factor found with economic indicator and economic growth rates of some African countries. Factor seven contains a mixture of variables from both data sets and has no clear interpretation. LARS and elastic net regularisation both include the second factor with small coefficients, making the estimates similar to that of the AR(1) model. Bagging makings relatively good forecasts by including the second factor or no factors at all depending on the iteration. As shown in Figure 12, bagging models the downward trend better than other models, which leads to 25% reduction in the MSE compared to the autoregressive benchmark. Boosting includes the first, second, and seventh factor. The coefficients are considerably smaller than that of FAAR, which enhances the forecasting accuracy. The model performs 21% better than the autoregressive benchmark. Figure 12 shows better forecasting performance primarily results from an accurate prediction of 2016.

All in all, methods that combine both data sets perform best, as three methods are able to beat both benchmark models. However, a drawback of the factor models is that factors have no clear economic explanation, because they are constructed from two different, unrelated data sources. Moreover, the forecasting performance of the different methods remains mixed.

To summarise, it is not easy to beat the autoregressive and random walk models in the context of forecasting Nigerian GDP growth. Shrinkage selection methods clearly outperform FAAR when the number of factors (with low explanatory power) increases, which is in line with the outcome of the simulation study. There is no preferred method, as each data has a different method performing best in terms of predictive accuracy. Factor models perform slightly better than models that incorporate explanatory variables directly when the data sets are used independently. Using a combination of the two data sets leads to the best forecasting performance, with boosting directly applied to the explanatory variables making the most accurate predictions. However, machine learning methods with factors also outperform the benchmark models, which shows shrinkage and variable selection methods with factorisation can be useful.

6 Conclusion

This research has investigated the effectiveness of combining shrinkage and variable selection methods in the context of forecasting contemporary real GDP growth in Nigeria. Our main finding is that this approach only provides limited gains in forecasting accuracy compared to simple benchmark models. A simulation study showed shrinkage and variable selection methods may offer significant advantages compared to least squares, particularly when the explanatory power of many variables is low. The preferred method is largely dependent on a bias-variance trade-off. The application of real GDP growth rates of African countries as explanatory variables to forecast Nigerian economic growth is ineffective. Similarly, most shrinkage and variable selection methods cannot beat the benchmark models when using economic indicator data. Combining the two data sets leads the best models in terms of predictive accuracy. Boosting improves on the autoregressive benchmark by 30%, however, the inclusion of over 30 variables leads to volatile forecasts that are undesirable in practice. Bagging and boosting in combination with factors constructed from the combined data outperform an AR(1) model with more than 20%, while simultaneously providing more stable forecasts. This result indicates shrinkage and variable selection methods in combination with factorisation can provide added value when forecasting real GDP growth in Nigeria.

Shrinkage and variable selection methods combined with factorisation may be applied in practice to forecast economic growth in Nigeria, provided that small gains in predictive accuracy can be of great value to the Nigerian government or investors. However, the results indicate the optimal method is data-dependent, therefore this approach requires thorough testing prior to implementation. Above that, the limitations of this research should be taken into account when constructing a model based on the findings of this paper.

The limitations of the research mainly stem from the data. First, the amount of economic indicators available in Nigeria for a time period of more than 30 years is very limited, which leads to the inclusion of fewer variables than in comparable studies. Second, the data may be subject to measurement errors. The African GDP growth data set contains information starting from 1963, when economic activity was measured with a lot less precision than nowadays. Moreover, some economic indicators are found to have extremely high variance even after adjusting and transforming the data, which makes it undesirable to include these variables in a model. Lastly, only five years of data are used for out-of-sample forecasting, due to the small sample size of the economic indicator data. This period is too short to make a reliable comparison of the methods, therefore a longer forecasting period should be used to verify the robustness of results.

Given the data limitations, future research should focus on the implementation of alternative data. A sufficiently large data set may be maintained while discarding older observations by using data measured at a higher frequency. Data or factors observed at different frequencies can be combined using mixed-data sampling models. Above that, it is of interest to investigate the use of Nigerian survey data as additional economic indicators, given that this type of information is becoming more widely available. Future research may also be dedicated to the use of alternative types of factorisation to model economic growth in Nigeria. For example, Kim and Swanson (2018) consider sparse principal component analysis, which enhances the interpretability of factors by inducing sparseness in factor loadings. Lastly, it is of interest to perform a similar study using a panel of African countries in order to generalise the findings of this paper.

References

Agri, E. M., Mailafia, D., and Umejiaku, R. (2017). Impact of Economic Recession on Macroeconomic Stability and Sustainable Development in Nigeria. *Science Journal of Economics*, 2017:1–12.

Akinlo, A. E. (2004). Foreign Direct Investment and Growth in Nigeria: An Empirical Investigation. *Journal of Policy Modeling*, 26(5):627–639.

Arbache, J. S. and Page, J. (2007). *Patterns of Long Term Growth in Sub-Saharan Africa*. The World Bank.

Bai, J. and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1):191–221.

Bai, J. and Ng, S. (2009). Boosting Diffusion Indices. *Journal of Applied Econometrics*, 24(4):607–629.

Banerjee, A. and Marcellino, M. (2006). Are there any Reliable Leading Indicators for US Inflation and GDP Growth? *International Journal of Forecasting*, 22(1):137–151.

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

Bühlmann, P. and Hothorn, T. (2007). Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22(4):477–505.

Bühlmann, P. and Yu, B. (2002). Analyzing Bagging. *The Annals of Statistics*, 30(4):927–961.

Cepni, O., Guney, I., and Swanson, N. R. (2019). Forecasting and Nowcasting Emerging Market GDP Growth Rates: The Role of Latent Global Economic Policy Uncertainty and Macroeconomic Data Surprise Factors. *Journal of Forecasting*.

Chimobi, O. P. (2010). Inflation and Economic Growth in Nigeria. *Journal of Sustainable Development*, 3(2):159.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least Angle Regression. *The Annals of Statistics*, 32(2):407–499.

Franses, P. H. B. F. and Vasilev, S. (2019). Real GDP Growth in Africa, 1963-2016. *Econometric Institute Research Papers*, EI-2019-23.

Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Gupta, R. and Kabundi, A. (2011). A Large Factor Model for Forecasting Macroeconomic Variables in South Africa. *International Journal of Forecasting*, 27(4):1076–1088.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67.

Ikoku, A. E. (2010). Is the Stock Market a Leading Indicator of Economic Activity in Nigeria? *CBN Journal of Applied Statistics*, 1(1):17–38.

Jerven, M. (2016). Discrepancies: Why Do GDP Growth Rates Differ? *Review of Agrarian Studies*, 6(1):63–80.

Kim, H. H. and Swanson, N. R. (2014). Forecasting Financial and Macroeconomic Variables using Data Reduction Methods: New Empirical Evidence. *Journal of Econometrics*, 178:352–367.

Kim, H. H. and Swanson, N. R. (2018). Mining Big Data using Parsimonious Factor, Machine Learning, Variable Selection and Shrinkage Methods. *International Journal of Forecasting*, 34(2):339–354.

Lledó, V. and Poplawski-Ribeiro, M. (2013). Fiscal Policy Implementation in Sub-Saharan Africa. *World Development*, 46:79–91.

Marcellino, M. (2008). A Linear Benchmark for Forecasting GDP Growth and Inflation? *Journal of Forecasting*, 27(4):305–340.

Nafziger, E. W. (1972). The Economic Impact of the Nigerian Civil War. *The Journal of Modern African Studies*, 10(2):223–245.

Nurudeen, A. and Usman, A. (2010). Government Expenditure and Economic Growth in Nigeria, 1970-2008: A Disaggregated Analysis. *Business and Economics Journal*, 2010(4):1–11.

Nwankwo, O. (2014). Impact of Corruption on Economic Growth in Nigeria. *Mediterranean Journal of Social Cciences*, 5(6):41.

Ogundipe, O., Ojeaga, P., and Ogundipe, A. (2014). Oil Price and Exchange Rate Volatility in Nigeria. *Journal of Economics and Finance (IOSR)*, 5(4):01–09.

Omisakin, D. and Olusegun, A. (2008). Oil Price Shocks and the Nigerian Economy: A Forecast Error Variance Decomposition Analysis. *Journal of Economic Theory*, 2(4):124–130.

Penm, J. H. and Terrell, R. D. (1994). Is Housing Activity a Leading Indicator? *Economic Record*, 70(210):241–252.

Rodrik, D. (2016). An African Growth Miracle? *Journal of African Economies*, 27(1):10–27.

Sachs, J. D. and Warner, A. M. (2001). The Curse of Natural Resources. *European Economic Review*, 45(4-6):827–838.

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

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

Appendix A

This appendix includes the code that is used to perform the analyses in this paper, which is compatible with `MATLAB 2019a`. The overview includes code written by the author, in addition to several non-standard functions written by other authors. Each piece of code includes a short description and gives credit in case the work of another author is used.

General

This subsection includes code that is both used in the simulation study and to make forecasts of Nigerian real GDP growth.

Bagging

```
function [beta] = bagging(y,X_other,X_shrink,c)
% BAGGING Performs bagging the shrinkage way.
%    [beta] = bagging(y,X_other,X_shrink,c) performs bagging the shrinkage
%    way. Dependent variable y is a T x 1 vector explained by T x M matrix
%    X_other and T x N matrix X_shrink. The variables in X_other are not
%    shrunk and may be empty. The parameter c denotes the threshold t-value
%    below which all coefficients are shrunk to zero. The output is a
%    (M+N) x 1 coefficient vector.
%
%    Original function: @bagging_shrink2
%    Orginal author: Hyun Hak Kim
%    Author: Fabian Neefjes
%    Reference: 'Empirical Comparison of Forecasting Methods' by Stock and
%    Watson, 2005


[t,m] = size(X_other);
[~,n] = size(X_shrink);
X = [X_other X_shrink];             % concatenate explanatory variables

[beta,~,resid] = mvregress(X,y);    % find coefficients using OLS

sige = (resid'*resid)/(t-(m+n));    % sample covariance matrix
tmp = sige*diag(inv(X'*X));         % squared vector Newey-West std. errors
tstat = beta./(sqrt(tmp));          % compute t-statistics

t_shrink = tstat(m+1:end);  % select t-statistics of coeff. to be shrunk
t_shrink = real(t_shrink);  % convert t-statistics to real numbers

psi = zeros(n,1);                       % initialize shrinkage coefficients
for k = 1:n
    if abs(t_shrink(k))<c               % set coefficient to zero if |t|<c
        psi(k) = 0;
    else                                % compute shrinkage coefficient
        psi(k) = 1 - normcdf(t_shrink(k) + c) + ...
            normcdf(t_shrink(k) - c)+ (normpdf(t_shrink(k) - c) - ...
```

```
            normpdf(t_shrink(k) + c)) / t_shrink(k);
    end
end
beta = [beta(1:m);beta(m+1:end).*psi]; % compute coefficient vector
```

Boosting

```
function [beta_IC, beta, IC_m]= boosting(y, X, nu, M)
% BOOSTING Performs componentwise L2 boosting.
%    [beta_IC, beta, IC_m]= boosting(y,X,nu,M) boosts according to the
%    componentwise L2 boosting algorithm. Dependent variable y is a T x 1
%    vector that is explained using T x N matrix X. Parameter nu is denotes
%    the step-length taken and should be a scalar between 0 and 1. Parameter
%    M is the iteration limit. The output beta_IC is the optimal N x 1
%    vector of boosted coefficients. The optimal coefficient vector is
%    selected using an adjusted BIC from the matrix beta contiaing all
%    coefficient estimates. IC_m is the index in beta of the optimal
%    coefficient vector.
%
%    Original function: @com_boost_ic
%    Orginal author: Hyun Hak Kim
%    Author: Fabian Neefjes
%    Reference: 'Boosting Algorithms: Regularization, Prediction and Model
%    Fitting' by Buhlmann and Hothorn, 2007

[t,n]=size(X);

% Set initial function to mean of dependent variable
mean_y = mean(y);
PHI(:,1) = mean_y * ones(size(y,1),1);

% Create empty matrix for current and all coefficients
beta(:,1) = zeros(size(X,2),1);
B(:,:,1) = zeros(t,t,1);

for m = 1:M
    %% Perform Boosting
    u = y - PHI(:,m);              % subtract explained part from Y to update ...
        residuals

    SSR = zeros(n,1);
    for i = 1:n                    % loop over all explanatory variables
        z = X(:,i);                % select explanatory variable
        [~,~,E] = mvregress(z,u);  % regress residuals on explanatory variable
        SSR(i) = sum(E.^2);        % compute sum of squared residuals
    end
    [~,i_m] = min(SSR);            % find index explanatory variable with ...
        lowest SSR
    zz = X(:,i_m);                 % select best explanatory variable
    b = mvregress(zz,u);           % regress best explanatory variable on ...
```

```
        residuals
    phi = b'*zz;                      % compute explained part by new regressor
    PHI(:,m+1) = PHI(:,m) + nu * phi; % update total explained part
    %% Calculate stopping time
    pm = zz*inv(zz'*zz)*zz';          % compute projection matrix of added regressor
    B(:,:,m+1) = B(:,:,m) + nu * pm * (eye(t)- B(:,:,m)); % compute L2 ...
        boosting hat-matrix
    df = trace(B(:,:,m+1));                      % compute degress of freedom
    sig2m = (y-PHI(:,m+1))'*(y-PHI(:,m+1));      % compute sum of squared errors
    IC(m,1) = log(sig2m) + (log(t) * df) / t ;   % compute modified BIC
    loc = zeros(size(X,2),1);                    % create vector to select ...
        regressor
    loc(i_m,1) = 1;                              % select one regressor
    beta(:,m+1) = beta(:,m) + nu * b * loc;      % update beta
end
[~,IC_m] = min(IC);                              % find lowest IC
beta_IC = beta(:,IC_m+1);                        % select beta with lowest IC
```

Cross-Validation

```
function [s_opt, b_opt, res_mean, res_std] = cv(fun, K, steps, sparse, X, y, ...
    varargin)
% CV Performs K-fold cross validation on a function.
%    [S_OPT, B_OBT, RES_MEAN, RES_STD] = CROSSVALIDATE(FUN, K, STEPS,
%    SPARSE X, Y, ...) performs simple K-fold cross-validation on function
%    FUN. STEPS is the number of equidistant positions along FUN at which
%    the sum of squared residuals (SSR) is measured. Typically this is some
%    large number to ensure sufficent accuracy. SPARSE is a boolean
%    variable that selects the most sparse model within one standard
%    deviation of the best model if true. X is the data matrix used as
%    input to FUN together with the response Y.  Finally, an arbitrary
%    number of arguments may be supplied to FUN.
%
%    Returns 0 < S_OPT <= 1 that determines the optimal model position,
%    B_OPT - the optimal parameters, RES_MEAN - the mean SSR curve and
%    RES_STD - the standard deviations of the SSR curve.
%
%    Original function: @crossvalidate
%    Original author: Karl Skoglund, IMM, DTU, kas@imm.dtu.dk
%    Author: Fabian Neefjes

%% Check varargin with fun
fun = fcnchk(fun,length(varargin));

%% Perform K-fold cross-validation
[n, p] = size(X);
rp = 1:n;
kappa = floor(n/K);
step = 1/(steps - 1);
b_interpolated = zeros(steps, p);
```

```
res = zeros(K, steps);
for k = 1:K
  testidx = rp((k-1)*kappa + 1:k*kappa);     % select test data
  validx = setdiff(rp(1:K*kappa), testidx); % select validation data
  Xtest = X(testidx,:);
  ytest = y(testidx);
  Xval = X(validx, :);
  yval = y(validx);
  if isempty(yval)
    Xval = Xtest;
    yval = ytest;
  end
  b = fun(Xval, yval, varargin{:});        % copmute betas
  step2 = 1/(size(b,1)-1);                 % 1 / # possible betas
  b_interpolated = interp1((0:step2:1)', b, (0:step:1)'); % interpolate beta
  res(k, :) = sum((ytest*ones(1,steps) - Xtest*b_interpolated').^2); % ...
      compute residuals of interpolated y on interpolated beta
end

%% Find optimal index in residual vector
% Calculate mean residual curve
if size(res,1) > 1
  res_mean = mean(res);
  res_std = std(res);
else
  res_mean = res;
  res_std = zeros(size(res));
end
% Find optimal index
[res_min, idx_opt] = min(res_mean);
if sparse
limit = res_min + res_std(idx_opt);
idx_opt2 = find(res_mean < limit, 1);   % find lowest index within one st. ...
    dev. of lowest CV error
if ~isempty(idx_opt2)
  idx_opt = idx_opt2;
end
end

%% Find optimal coefficient vector
s_opt = idx_opt/steps;
b = fun(X, y, varargin{:});
b_opt = interp1q((0:step2:1)', b, s_opt);
```

Find Autoregressive Order

```
function [order] = find_ar(y,max_lag)
% FIND_AR Finds the autoregressive order of a variable.
%   [order] = find_ar(y,max_lag) returns the optimal autoregressive order
%   of a T x 1 vector y based on the BIC, where max_lag is the highest lag
```

```
%   considered.
%
%   Author: Fabian Neefjes

if max_lag > 10                          % warning if large max_lag
    warning('Large max_lag results in a loss of information.');
end

[n, ~] = size(y);
ss = n-max_lag;                          % compute sample size
bic = zeros(max_lag,1);
for p = 0:max_lag
    lag_y = lagmatrix(y, 1:p);           % make lag matrix
    lag_y = lag_y(1+max_lag:end,:);      % select lags
    exp_var = [ones(ss,1) lag_y];        % add constant term to lags
    [~,~,resid] = mvregress(exp_var, y(1+max_lag:end,:));
    bic(p+1) = (ss) * log(resid'*resid/ss) + (p+1) * log(ss); % compute BIC
end
[~, index] = min(bic);                   % find lowest BIC
order = index - 1;                       % return lag order
```

LARS

```
function beta = lars(X, y, method, stop, useGram, Gram, trace)
% LARS  The LARS algorithm for performing LAR or LASSO.
%    BETA = LARS(X, Y) performs least angle regression on the variables in
%    X to approximate the response Y. Variables X are assumed to be
%    normalized (zero mean, unit length), the response Y is assumed to be
%    centered.
%    BETA = LARS(X, Y, METHOD), where METHOD is either 'LARS' or 'LARS'
%    determines whether least angle regression or lasso regression should
%    be performed.
%    BETA = LARS(X, Y, METHOD, STOP) with nonzero STOP will perform least
%    angle or lasso regression with early stopping. If STOP is negative,
%    STOP is an integer that determines the desired number of variables. If
%    STOP is positive, it corresponds to an upper bound on the L1-norm of
%    the BETA coefficients.
%    BETA = LARS(X, Y, METHOD, STOP, USEGRAM) specifies whether the Gram
%    matrix X'X should be calculated (USEGRAM = 1) or not (USEGRAM = 0).
%    Calculation of the Gram matrix is suitable for low-dimensional
%    problems. By default, the Gram matrix is calculated.
%    BETA = LARS(X, Y, METHOD, STOP, USEGRAM, GRAM) makes it possible to
%    supply a pre-computed Gram matrix. Set USEGRAM to 1 to enable. If no
%    Gram matrix is available, exclude argument or set GRAM = [].
%    BETA = LARS(X, Y, METHOD, STOP, USEGRAM, GRAM, TRACE) with nonzero
%    TRACE will print the adding and subtracting of variables as all
%    LARS/lasso solutions are found.
%    Returns BETA where each row contains the predictor coefficients of
%    one iteration. A suitable row is chosen using e.g. cross-validation,
%    possibly including interpolation to achieve sub-iteration accuracy.
```

```
%
% Author: Karl Skoglund, IMM, DTU, kas@imm.dtu.dk
% Reference: 'Least Angle Regression' by Bradley Efron et al, 2003.

%% Input checking
% Set default values.
if nargin < 7
    trace = 0;
end
if nargin < 6
  Gram = [];
end
if nargin < 5
  useGram = 1;
end
if nargin < 4
  stop = 0;
end
if nargin < 3
  method = 'lars';
end
if strcmpi(method, 'lasso')
  lasso = 1;
else
  lasso = 0;
end

%% LARS variable setup
[n, p] = size(X);
nvars = min(n-1,p);
maxk = 8*nvars; % Maximum number of iterations

if stop == 0
  beta = zeros(2*nvars, p);
elseif stop < 0 % restrict number of variables
  beta = zeros(2*round(-stop), p);
else
  beta = zeros(100, p); % set upper bound on L1 norm
end
mu = zeros(n, 1); % current "position" as LARS travels towards lsq solution
I = 1:p; % inactive set
A = []; % active set

% Calculate Gram matrix if necessary
if isempty(Gram) && useGram
  Gram = X'*X; % Precomputation of the Gram matrix. Fast but memory consuming.
end
if ~useGram
  R = []; % Cholesky factorization R'R = X'X where R is upper triangular
end

```

```
lassocond = 0; % LASSO condition boolean
stopcond = 0; % Early stopping condition boolean
k = 0; % Iteration count
vars = 0; % Current number of variables

%if trace
%  disp(sprintf('Step\tAdded\tDropped\t\tActive set size'));
%end

%% LARS main loop
while vars < nvars && ~stopcond && k < maxk % check if not all variables are ...
    included, stopping condition not satisfied, iteration limit not reached
  k = k + 1; % increase iteration limit
  c = X'*(y - mu); % compute correlation
  [C, j] = max(abs(c(I))); % find regressor with maximum correlation from ...
      inactive set
  j = I(j); % select regressor from inactive set with maximum correlation

  if ~lassocond % if a variable has been dropped, do one iteration with this ...
      configuration (don't add new one right away)
    if ~useGram
      R = cholinsert(R,X(:,j),X(:,A));
    end
    A = [A j]; % add regressor with maximum correlation to active set
    I(I == j) = []; % remove regressor with maximum correlation from ...
        inactive set
    vars = vars + 1; % increase variable count
    %if trace
    %  disp(sprintf('%d\t\t%d\t\t\t\t\t%d', k, j, vars));
    %end
  end

  s = sign(c(A)); % get the signs of the correlations

  if useGram
    S = s*ones(1,vars);
    GA1 = pinv(Gram(A,A).*S'.*S)*ones(vars,1);
    AA = 1/sqrt(sum(GA1));
    w = AA*GA1.*s; % weights applied to each active variable to get ...
        equiangular direction
  else
    GA1 = R\(R'\s);
    AA = 1/sqrt(sum(GA1.*s));
    w = AA*GA1;
  end
  u = X(:,A)*w; % equiangular direction (unit vector)

  if vars == nvars % if all variables active, go all the way to the lsq solution
    gamma = C/AA;
  else
    a = X'*u; % correlation between each variable and equiangular vector
```

```
    temp = [(C - c(I))./(AA - a(I)); (C + c(I))./(AA + a(I))];
    gamma = min([temp(temp > 0); C/AA]); % compute step-length
  end

  % LASSO modification
  if lasso
    lassocond = 0;
    temp = -beta(k,A)./w';
    [gamma_tilde] = min([temp(temp > 0) gamma]);
    j = find(temp == gamma_tilde); % find index of regressor where ...
        correlation changes
    if gamma_tilde < gamma
      gamma = gamma_tilde; % restrict step-length
      lassocond = 1;
    end
  end

  mu = mu + gamma*u; % update function
  if size(beta,1) < k+1
    beta = [beta; zeros(size(beta,1), p)];
  end
  beta(k+1,A) = beta(k,A) + gamma*w';

  % Early stopping at specified bound on L1 norm of beta
  if stop > 0
    t2 = sum(abs(beta(k+1,:)));
    if t2 >= stop
      t1 = sum(abs(beta(k,:)));
      s = (stop - t1)/(t2 - t1); % interpolation factor 0 < s < 1
      beta(k+1,:) = beta(k,:) + s*(beta(k+1,:) - beta(k,:));
      stopcond = 1;
    end
  end

  % If LASSO condition satisfied, drop variable from active set
  if lassocond == 1
    if ~useGram
      R = choldelete(R,j);
    end
    I = [I A(j)]; % add variable to inactive set
    A(j) = []; % remove variable from active set
    vars = vars - 1; % decrease variable count
    if trace
      disp(sprintf('%d\t\t\t\t%d\t\t\t%d', k, j, vars));
    end
  end

  % Early stopping at specified number of variables
  if stop < 0
    stopcond = vars >= -stop;
  end
```

```
end

% trim beta
if size(beta,1) > k+1
  beta(k+2:end, :) = [];
end

if k == maxk
  disp('LARS warning: Forced exit. Maximum number of iteration reached.');
end

%% Fast Cholesky insert and remove functions
% Updates R in a Cholesky factorization R'R = X'X of a data matrix X. R is
% the current R matrix to be updated. x is a column vector representing the
% variable to be added and X is the data matrix containing the currently
% active variables (not including x).
function R = cholinsert(R, x, X)
diag_k = x'*x; % diagonal element k in X'X matrix
if isempty(R)
  R = sqrt(diag_k);
else
  col_k = x'*X; % elements of column k in X'X matrix
  R_k = R'\col_k'; % R'R_k = (X'X)_k, solve for R_k
  R_kk = sqrt(diag_k - R_k'*R_k); % norm(x'x) = norm(R'*R), find last ...
      element by exclusion
  R = [R R_k; [zeros(1,size(R,2)) R_kk]]; % update R
end

% Deletes a variable from the X'X matrix in a Cholesky factorisation R'R =
% X'X. Returns the downdated R. This function is just a stripped version of
% Matlab's qrdelete.
function R = choldelete(R,j)
R(:,j) = []; % remove column j
n = size(R,2);
for k = j:n
  p = k:k+1;
  [G,R(p,k)] = planerot(R(p,k)); % remove extra element in column
  if k < n
    R(p,k+1:n) = G*R(p,k+1:n); % adjust rest of row
  end
end
R(end,:) = []; % remove zero'ed out row

%% To do
%
% There is a modification that turns least angle regression into stagewise
% (epsilon) regression. This has not been implemented.
```

LARS-EN

```
function beta = larsen(X, y, lambda2, stop, trace)
% LARSEN  The LARSEN algorithm for elastic net regression.
%    BETA = LARSEN(X, Y) performs elastic net regression on the variables
%    in X to approximate the response Y.  Variables X are assumed to be
%    normalized (zero mean, unit length), the response Y is assumed to be
%    centered. The ridge term coefficient, lambda2, has a default value of
%    1e-6. This keeps the ridge influence low while making p > n possible.
%    BETA = LARSEN(X, Y, LAMBDA2) adds a user-specified LAMBDA2. LAMBDA2 =
%    0 produces the lasso solution.
%    BETA = LARSEN(X, Y, LAMBDA2, STOP) with nonzero STOP will perform
%    elastic net regression with early stopping. If STOP is negative, its
%    absolute value corresponds to the desired number of variables. If STOP
%    is positive, it corresponds to an upper bound on the L1-norm of the
%    BETA coefficients.
%    BETA = LARSEN(X, Y, LAMBDA2, STOP, TRACE) with nonzero TRACE will
%    print the adding and subtracting of variables as all elastic net
%    solutions are found.
%    Returns BETA where each row contains the predictor coefficients of
%    one iteration. A suitable row is chosen using e.g. cross-validation,
%    possibly including interpolation to achieve sub-iteration accuracy.
%
% Author: Karl Skoglund, IMM, DTU, kas@imm.dtu.dk
% Reference: 'Regularization and Variable Selection via the Elastic Net' by
% Hui Zou and Trevor Hastie, 2005.

%% Input checking
if nargin < 5
  trace = 0;
end
if nargin < 4
  stop = 0;
end
if nargin < 3
  lambda2 = 1e-6;
end

%% Elastic net variable setups
[n, p] = size(X);
maxk = 8*(n+p); % Maximum number of iterations

if lambda2 < eps
  nvars = min(n-1,p); %Pure LASSO
else
  nvars = p; % Elastic net
end
if stop > 0
  stop = stop/sqrt(1 + lambda2);
end
if stop == 0
```

```
  beta = zeros(2*nvars, p);
elseif stop < 0
  beta = zeros(2*round(-stop), p); % restrict number of variables
else
  beta = zeros(100, p); % early stopping
end
mu = zeros(n, 1); % current "position" as LARS-EN travels towards lsq solution
I = 1:p; % inactive set
A = []; % active set

R = []; % Cholesky factorization R'R = X'X where R is upper triangular

lassocond = 0; % Set to 1 if LASSO condition is met
stopcond = 0; % Set to 1 if early stopping condition is met
k = 0; % Algorithm step count
vars = 0; % Current number of variables

d1 = sqrt(lambda2); % Convenience variables d1 and d2
d2 = 1/sqrt(1 + lambda2);

%if trace
%  disp(sprintf('Step\tAdded\tDropped\t\tActive set size'));
%end

%% Elastic net main loop
while vars < nvars && ~stopcond && k < maxk
  k = k + 1;
  c = X'*(y - mu)*d2;
  [C, j] = max(abs(c(I)));
  j = I(j);

  if ~lassocond % if a variable has been dropped, do one iteration with this ...
      configuration (don't add new one right away)
    R = cholinsert(R,X(:,j),X(:,A),lambda2);
    A = [A j];
    I(I == j) = [];
    vars = vars + 1;
    %if trace
    %  disp(sprintf('%d\t\t%d\t\t\t\t\t%d', k, j, vars));
    %end
  end

  s = sign(c(A)); % get the signs of the correlations

  GA1 = R\(R'\s);
  AA = 1/sqrt(sum(GA1.*s));
  w = AA*GA1;
  u1 = X(:,A)*w*d2; % equiangular direction (unit vector) part 1
  u2 = zeros(p, 1); u2(A) = d1*d2*w; % part 2
  if vars == nvars % if all variables active, go all the way to the lsq solution
    gamma = C/AA;
```

```
    else
      a = (X'*u1 + d1*u2)*d2; % correlation between each variable and ...
          equiangular vector
      temp = [(C - c(I))./(AA - a(I)); (C + c(I))./(AA + a(I))];
      gamma = min([temp(temp > 0); C/AA]);
    end

    % LASSO modification
    lassocond = 0;
    temp = -beta(k,A)./w';
    [gamma_tilde] = min([temp(temp > 0) gamma]);
    j = find(temp == gamma_tilde);
    if gamma_tilde < gamma
      gamma = gamma_tilde;
      lassocond = 1;
    end

    mu = mu + gamma*u1;
    if size(beta,1) < k+1
      beta = [beta; zeros(size(beta,1), p)];
    end
    beta(k+1,A) = beta(k,A) + gamma*w';

    % Early stopping at specified bound on L1 norm of beta
    if stop > 0
      t2 = sum(abs(beta(k+1,:)));
      if t2 >= stop
        t1 = sum(abs(beta(k,:)));
        s = (stop - t1)/(t2 - t1); % interpolation factor 0 < s < 1
        beta(k+1,:) = beta(k,:) + s*(beta(k+1,:) - beta(k,:));
        stopcond = 1;
      end
    end

    % If LASSO condition satisfied, drop variable from active set
    if lassocond == 1
      R = choldelete(R,j);
      I = [I A(j)];
      A(j) = [];
      vars = vars - 1;
      if trace
        disp(sprintf('%d\t\t\t\t%d\t\t\t%d', k, j, vars));
      end
    end

    % Early stopping at specified number of variables
    if stop < 0
      stopcond = vars >= -stop;
    end
  end
```

```
% trim beta
if size(beta,1) > k+1
  beta(k+2:end, :) = [];
end

% divide by d2 to avoid double shrinkage
beta = beta/d2;

if k == maxk
  disp('LARS-EN warning: Forced exit. Maximum number of iteration reached.');
end


%% Fast Cholesky insert and remove functions
% Updates R in a Cholesky factorization R'R = X'X of a data matrix X. R is
% the current R matrix to be updated. x is a column vector representing the
% variable to be added and X is the data matrix containing the currently
% active variables (not including x).
function R = cholinsert(R, x, X, lambda)
diag_k = (x'*x + lambda)/(1 + lambda); % diagonal element k in X'X matrix
if isempty(R)
  R = sqrt(diag_k);
else
  col_k = x'*X/(1 + lambda); % elements of column k in X'X matrix
  R_k = R'\col_k'; % R'R_k = (X'X)_k, solve for R_k
  R_kk = sqrt(diag_k - R_k'*R_k); % norm(x'x) = norm(R'*R), find last ...
      element by exclusion
  R = [R R_k; [zeros(1,size(R,2)) R_kk]]; % update R
end

% Deletes a variable from the X'X matrix in a Cholesky factorisation R'R =
% X'X. Returns the downdated R. This function is just a stripped version of
% Matlab's qrdelete.
function R = choldelete(R,j)
R(:,j) = []; % remove column j
n = size(R,2);
for k = j:n
  p = k:k+1;
  [G,R(p,k)] = planerot(R(p,k)); % remove extra element in column
  if k < n
    R(p,k+1:n) = G*R(p,k+1:n); % adjust rest of row
  end
end
R(end,:) = []; % remove zero'ed out row
```

MSE

```
function [mse] = mse(y, y_hat)
% MSE Computes the mean squared prediction error.
%   [mse] = mse(y, y_hat) computes the mean squared prediction error using
```

```
%   observed T x 1 vector y and predicted T x 1 vector y_hat.
%
%   Author: Fabian Neefjes

sq_error = (y - y_hat).^2;
mse = mean(sq_error);
```

Ridge Regression

```
function b = ridge2(X,y,k,flag)
% RIDGE2 Ridge regression.
%   B1 = RIDGE2(X,y,K) returns the vector B1 of regression coefficients
%   obtained by performing ridge regression of the response vector Y
%   on the predictors X using ridge parameter K.  The matrix X should
%   not contain a column of ones.  The results are computed after
%   centering and scaling the X columns so they have mean 0 and
%   standard deviation 1.  If Y has n observations, X is an n-by-p
%   matrix, and K is a scalar, the result B1 is a row vector with p
%   elements.  If K has m elements, B1 is m-by-p.
%
%   B0 = RIDGE2(X,y,K,0) performs the regression without centering and
%   scaling.  The result B0 has p+1 coefficients, with the first being
%   the constant term.   RIDGE(X,y,K,1) is the same as RIDGE(X,y,K).
%
%   The relationship between B1 and B0 is as follows:
%
%      m = mean(X);
%      s = std(X,0,1)';
%      temp = B1./s;
%      B0 = [mean(Y)-m*temp; temp]
%
%   In general, B1 is more useful for producing ridge traces (see the
%   following example) where the coefficients are displayed on the same
%   scale.  B0 is more useful for making predictions.
%
%   Note: this function is the same as @Ridge, besides that X and y are
%   interchanged and beta is returned as a row vector to make it compatible
%   with @cv.
%   Copyright 1993-2008 The MathWorks, Inc.
%   Adjusted by: Fabian Neefjes


if nargin < 3,
    error(message('stats:ridge:TooFewInputs'));
end

if nargin<4 || isempty(flag) || isequal(flag,1)
   unscale = false;
elseif isequal(flag,0)
   unscale = true;
```

```
else
   error(message('stats:ridge:BadScalingFlag'));
end

% Check that matrix (X) and left hand side (y) have compatible dimensions
[n,p] = size(X);

[n1,collhs] = size(y);
if n~=n1,
    error(message('stats:ridge:InputSizeMismatch'));
end

if collhs ~= 1,
    error(message('stats:ridge:InvalidData'));
end

% Remove any missing values
wasnan = (isnan(y) | any(isnan(X),2));
if (any(wasnan))
   y(wasnan) = [];
   X(wasnan,:) = [];
   n = length(y);
end

% Normalize the columns of X to mean zero, and standard deviation one.
mx = mean(X);
stdx = std(X,0,1);
idx = find(abs(stdx) < sqrt(eps(class(stdx))));
if any(idx)
  stdx(idx) = 1;
end

MX = mx(ones(n,1),:);
STDX = stdx(ones(n,1),:);
Z = (X - MX) ./ STDX;
if any(idx)
  Z(:,idx) = 1;
end

% Compute the ridge coefficient estimates using the technique of
% adding pseudo observations having y=0 and X'X = k*I.
pseudo = sqrt(k(1)) * eye(p);
Zplus  = [Z;pseudo];
yplus  = [y;zeros(p,1)];

% Set up an array to hold the results
nk = numel(k);

% Compute the coefficient estimates
b = Zplus\yplus;

```

```
if nk>1
   % Fill in more entries after first expanding b.  We did not pre-
   % allocate b because we want the backslash above to determine its class.
   b(end,nk) = 0;
   for j=2:nk
      Zplus(end-p+1:end,:) = sqrt(k(j)) * eye(p);
      b(:,j) = Zplus\yplus;
   end
end

% Put on original scale if requested
if unscale
   b = b ./ repmat(stdx',1,nk);
   b = [mean(y)-mx*b; b];
end
b = b'; % change from column to row vector
```

Simulation Study

This subsection includes code that is only used in the simulation study.

Main File

```
% This program performs the simulation study and provides the performance
% of methods as output. To obtain the same results in the paper, run with
% v = 100, N=50, r=20, coeff = [[3, 1.5, 0, 0, 2, zeros(1,N-5)]',
% [2*ones(1,10), zeros(1,N-10)]'], iteration_limit = 10, t = [1000, 100,
% 50], b = [1,2], alpha = 0.
%
% Author: Fabian Neefjes

%% Configuration
clear;
clc;
rng('default')          % random number generator
v = 100;                % size forecast sample
N = 50;                 % number of explanatory variables
r = 20;                 % number of factors used to generate data
coeff = [[3, 1.5, 0, 0, 2, zeros(1,N-5)]', [2*ones(1,10), zeros(1,N-10)]']; ...
    % coefficient vector factors
iteration_limit = 100;  % number of simulations
%% Initialisation
for t = [1000, 100, 50]     % estimation sample size
for b = [1, 2]
beta = coeff(:,b);          % set beta
k_star = sum(abs(beta)>0); % number of non-zero coefficients
k_inc = abs(beta)>0;        % position of non-zero coefficients
for alpha = [0, 0.5]        % first-order autoregresive component y
p_mean = zeros(iteration_limit,1);       % autoregressive order p
k_auto = zeros(iteration_limit,r);       % # non-zero coefs AR
k_bag = zeros(iteration_limit,r);        % # non-zero coefs bagging
```

```
k_boost = zeros(iteration_limit,r);      % # non-zero coefs boosting
k_en = zeros(iteration_limit,r);         % # non-zero coefs elastic net
k_lars = zeros(iteration_limit,r);       % # non-zero coefs LARS
k_faar = zeros(iteration_limit,r);       % # non-zero coefs FAAR
k_ridge = zeros(iteration_limit,r);      % # non-zero coefs ridge
k_rw = zeros(iteration_limit,r);         % # non-zero coefs random walk
mse_auto = zeros(iteration_limit,1);     % MSE AR
mse_bag = zeros(iteration_limit,1);      % MSE bagging
mse_boost = zeros(iteration_limit,1);    % MSE boosting
mse_en = zeros(iteration_limit,1);       % MSE elastic net
mse_faar = zeros(iteration_limit,1);     % MSE FAAR
mse_lars = zeros(iteration_limit,1);     % MSE LARS
mse_ridge = zeros(iteration_limit,1);    % MSE ridge
mse_rw = zeros(iteration_limit,1);       % MSE random walk
for n=1:iteration_limit
tic;
disp(n);                                   % print iteration count
%% Load data
[train_X, forecast_X, train_y, forecast_y] = gen_data(t, v, N, beta, alpha);
y = [train_y; forecast_y];                 % concatenate training and ...
    estimation sample
X = [train_X; forecast_X];

%% Generate factors
mX = mean(train_X);                            % obtain mean of training X
sX = std(train_X);                             % obatin st. dev. of training X
train_X_norm = (train_X - mX) ./ sX;           % normalize training X
X_norm = (X - mX) ./ sX;                       % normalize all X

[pca_coeff, ~, pca_ev] = pca(train_X_norm);   % compute loadings from ...
    training X
train_factors = train_X_norm*pca_coeff;        % create factors for training X
factors = X_norm*pca_coeff;                    % create factors for X
mF = mean(train_factors);                      % obtain mean of training factors
sF = std(train_factors);                       % obtain st. dev. of training ...
    factors
train_factors_norm = (train_factors - mF) ./ sF; % standardize training factors
factors_norm = (factors - mF) ./ sF;           % standardize all factors

%% Select number of factors
r = 20;                                    % fix r = 20
train_factors = train_factors(:,1:r);      % choose first r factors
factors = factors(:,1:r);
train_factors_norm = train_factors_norm(:,1:r);
factors_norm = factors_norm(:,1:r);

%% Find autoregressive order and compute AR(p) model
p = find_ar(train_y, 5);             % find autoregressive lag
p = max(1,p);                        % set minimum p to 1
p_mean(n) = p;                       % store p
y_lag = lagmatrix(train_y, 1:p);
```

```
X_auto = [ones(t-p,1) y_lag(p+1:end,:) ];
[beta_auto,~,auto_resid] = mvregress(X_auto,train_y(p+1:end,:));
resid = auto_resid;
q = p;                                  % variable for start estimation sample

y_lag = lagmatrix(y, 1:p);
X_forecast_auto = [ones(v,1) y_lag(t+1:t+v,:)];
y_hat_auto = X_forecast_auto*beta_auto;
mse_auto(n) = mse(forecast_y,y_hat_auto);

%% Factor Augmented Autoregression
X_factor = train_factors(1+q-1:end-1,:);
X_faar = [X_auto X_factor];
[beta_faar] = mvregress(X_faar, train_y(q+1:end,:));
X_factor_forecast = factors(t:end-1,:);
X_forecast_faar = [X_forecast_auto X_factor_forecast];
y_hat_faar = X_forecast_faar*beta_faar;
mse_faar(n) = mse(forecast_y, y_hat_faar);
k_faar(n,:) = abs(beta_faar(1+p+1:end))>0;

%% Ridge regression
k = 0:0.01:100;                     % range of lambda
X_factor_norm = train_factors_norm(1+q-1:end-1,:);
[~,beta_ridge,~,~] = cv(@ridge2, 10, length(k), 0, X_factor_norm, resid, k);
beta_ridge = real(beta_ridge'); % convert to column vector
X_factor_norm_forecast = factors_norm(t:end-1,:);
y_hat_ridge = y_hat_auto + X_factor_norm_forecast*beta_ridge;
mse_ridge(n) = mse(forecast_y,y_hat_ridge);
k_ridge(n,:) = abs(beta_ridge)>0;

%% LARS
[~,beta_lars,~,~] = cv(@lars, 10, 1000, 1, X_factor_norm, resid, 'LARS', ...
    0,0,[],0);
beta_lars = real(beta_lars'); % convert to column vector
y_hat_lars = y_hat_auto + X_factor_norm_forecast*beta_lars;
mse_lars(n) = mse(forecast_y,y_hat_lars);
k_lars(n,:) = abs(beta_lars)>0;

%% Elastic net
lambda2 = [0, 0.01, 0.1, 1, 10, 100];   % range of lamda_2
best_resid = inf;
for i=1:length(lambda2)
    [s_en,beta_temp,mean_resid,~] = cv(@larsen, 10, 1000, 1, X_factor_norm, ...
        resid, lambda2(i));
    if mean_resid(s_en*length(mean_resid)+1) < best_resid
        best_resid = mean_resid(s_en*length(mean_resid)+1);  % store residuals
        beta_en = beta_temp;                                 % store ...
            coefficients
        opt_lambda2 = lambda2(i);                            % store lambda_2
        opt_index = i;                                       % store index ...
            lambda_2
```

```
    end
end
beta_en = real(beta_en'); % convert to column vector
y_hat_en = y_hat_auto + X_factor_norm_forecast*beta_en;
mse_en(n) = mse(forecast_y,y_hat_en);
k_en(n,:) = abs(beta_en)>0;

%% Bagging
c = 1.96;                    % bagging coefficient
[beta_bag] = bagging(train_y(q+1:end,:), X_auto, X_factor, c);
y_hat_bag = X_forecast_faar*beta_bag;
mse_bag(n) = mse(forecast_y, y_hat_bag);
k_bag(n,:) = abs(beta_bag(1+p+1:end))>0;

%% Boosting
M = 100;                    % max iteration count
nu = 0.5;                   % nu parameter
[beta_boost] = boosting(resid, X_factor, nu, M);
y_hat_boost = y_hat_auto + X_factor_forecast*beta_boost;
mse_boost(n) = mse(forecast_y, y_hat_boost);
k_boost(n,:) = abs(beta_boost)>0;

%% Random Walk
y_hat_rw = [train_y(end); forecast_y(1:end-1,:)];
mse_rw(n) = mse(forecast_y, y_hat_rw);
toc;
end
%% Export output
header = ["Method" "Avg. # vars" "Corr. # vars" "Corr. incl" "MSE Ratio" "MSE"];
methods = ["AR(p)" "Random Walk" "FAAR" "Ridge" "LARS" "Elastic Net" ...
    "Bagging" "Boosting"];
mse_vec = [mean(mse_auto), mean(mse_rw) mean(mse_faar), mean(mse_ridge), ...
    mean(mse_lars), mean(mse_en), mean(mse_bag), mean(mse_boost)];
mse_ratio = mse_vec./mean(mse_auto);
k_sum = [sum(k_auto,2), sum(k_rw,2), sum(k_faar,2), sum(k_ridge,2), ...
    sum(k_lars,2), sum(k_en,2), sum(k_bag,2), sum(k_boost,2)];
k_mean = mean(k_sum);
k_ratio = mean(k_sum == k_star);
inc_mat = repmat(k_inc(1:r)', [iteration_limit,1]);
k_incl = [sum(k_auto.*inc_mat,2), sum(k_rw.*inc_mat,2), ...
    sum(k_faar.*inc_mat,2), sum(k_ridge.*inc_mat,2), sum(k_lars.*inc_mat,2), ...
    sum(k_en.*inc_mat,2), sum(k_bag.*inc_mat,2), sum(k_boost.*inc_mat,2)];
k_incl = k_incl == k_star;
k_incl = mean(k_incl);
p_vec = ["Correct/Mean p", mean(p_mean==(alpha>0)), mean(p_mean), "", "", ""];
filename = join(["sim_t", string(t), "_N", string(N), "_b", string(b), ...
    "_alpha", string(alpha), "_i", string(iteration_limit), ".xls"],"");
writematrix([header; methods' k_mean' k_ratio' k_incl'  mse_ratio' mse_vec'; ...
    ["", "", "", "", "", ""]; p_vec;], filename);
end
end
```

```
end
```

Plots

```
% This program makes plots of one iteration of the simulation study.
% To obtain the same results as in the paper, set v=100, N=50, coeff =
% [[3, 1.5, 0, 0, 2, zeros(1,N-5)]',[2*ones(1,10), zeros(1,N-10)]'], t=100,
% b=1, alpha = 0.
%
% Author: Fabian Neefjes

%% Configuration
clear;
clc;
rng('default')             % random number generator
v = 100;                   % size forecast sample
N = 50;                    % number of explanatory variables
coeff = [[3, 1.5, 0, 0, 2, zeros(1,N-5)]', [2*ones(1,10), zeros(1,N-10)]']; ...
    % coefficient vector factors
%% Initialisation
t = 100;                   % estimation sample size
b = 1;                     % coefficient vector selected
beta = coeff(:,b);         % set beta
k_star = sum(abs(beta)>0); % number of non-zero coefficients
alpha = 0;                 % first-order autoregresive component y
%% Load data
[train_X, forecast_X, train_y, forecast_y] = gen_data(t, v, N, beta, alpha); ...
    % generate data
y = [train_y; forecast_y];                     % concatenate training and ...
    estimation sample
X = [train_X; forecast_X];

%% Generate factors
mX = mean(train_X);                            % obtain mean of training X
sX = std(train_X);                             % obatin st. dev. of training X
train_X_norm = (train_X - mX) ./ sX;           % normalize training X
X_norm = (X - mX) ./ sX;                       % normalize all X

[pca_coeff, ~, pca_ev] = pca(train_X_norm);    % compute loadings from ...
    training X
train_factors = train_X_norm*pca_coeff;        % create factors for training X
factors = X_norm*pca_coeff;                    % create factors for X
mF = mean(train_factors);                      % obtain mean of training factors
sF = std(train_factors);                       % obtain st. dev. of training ...
    factors
train_factors_norm = (train_factors - mF) ./ sF; % standardize training factors
factors_norm = (factors - mF) ./ sF;           % standardize all factors

%% Select number of factors
r = 20;                                    % fix r = 20
```

```
train_factors = train_factors(:,1:r);      % choose r factors
factors = factors(:,1:r);
train_factors_norm = train_factors_norm(:,1:r);
factors_norm = factors_norm(:,1:r);

%% Find autoregressive order and compute AR(p) model
p = find_ar(train_y, 5);                        % find autoregressive lag
p = max(1,p);                                   % set minimum p to 1
y_lag = lagmatrix(train_y, 1:p);
X_auto = [ones(t-p,1) y_lag(p+1:end,:)];
[beta_auto,~,auto_resid] = mvregress(X_auto,train_y(p+1:end,:));
resid = auto_resid;
q = p;                                          % variable for start ...
    estimation sample

%% Factor Augmented Autoregression
X_factor = train_factors(1+q-1:end-1,:);
X_faar = [X_auto X_factor];
[beta_faar] = mvregress(X_faar, train_y(q+1:end,:));
% Plot forecasts
fig_faar = figure;
g = gscatter(ones(1,r), beta_faar(1+p+1:end), 1:r, [],[]);
set(g, 'MarkerSize', 25);
xticks([])
xticklabels({});
xlim([0 2]);
ylim([-1 4]);
legend1 = legend('show');
set(legend1,'location','eastoutside', 'NumColumns', 1);
title(legend1,'Factor');
set(gca,'FontSize',16);
saveas(fig_faar, 'fig_faar', 'epsc');

%% Ridge regression
k = 0:0.01:100;                  % range of lambda
X_factor_norm = train_factors_norm(1+q-1:end-1,:);
[s_ridge,beta_ridge,rm_ridge,rs_ridge] = cv(@ridge2, 10, length(k), 0, ...
    X_factor_norm, resid, k);
% Plot CV error of best lambda
fig_ridge_cv = cv_plot(s_ridge,rm_ridge,rs_ridge);
xlabel('\lambda');
xticks(0:0.1:1);
xticklabels(0:10:100);
set(gca,'FontSize',16);
saveas(fig_ridge_cv, 'fig_ridge_cv', 'epsc');
% Plot coefficients as a function of lambda
beta_ridge = real(beta_ridge'); % convert to column vector
fig_ridge = figure;
hold on
col=hsv(20);
ridge_plot = zeros(r,length(k));
```

```
for i = 1:length(k)
ridge_plot(:,i) = ridge2(X_factor_norm, resid, k(i))';
end
ridge_plot = ridge_plot./[sF(1:r)'*ones(length(ridge_plot),1)'];
for i = 1:r
pr(i) = plot(k, ridge_plot(i,:), 'color', col(i,:));
end
ind_ridge = round(s_ridge*length(k));
plot([k(ind_ridge) k(ind_ridge)], [-1 4],'--', 'color', 'k', 'LineWidth', 1);
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
xlabel('\lambda');
ylim([-1 4]);
set(pr, 'LineWidth', 1);
set(gca,'FontSize',16)
set(gcf,'renderer','Painters')
saveas(fig_ridge, 'fig_ridge', 'epsc');

%% LARS
[s_lars,beta_lars,rm_lars,rs_lars] = cv(@lars, 10, 1000, 1, X_factor_norm, ...
    resid, 'LARS', 0,0,[],0);
beta_lars = real(beta_lars'); % convert to column vector
% Plot CV error
fig_lars_cv = cv_plot(s_lars,rm_lars,rs_lars);
xlabel('Iteration');
xticks([((1:r+1)-1)./r]);
xticklabels(0:r);
set(gca,'FontSize',16);
saveas(fig_lars_cv, 'fig_lars_cv', 'epsc');
% Plot coefficients as a function of iterations
fig_lars = figure;
hold on
lars_plot = lars(X_factor_norm, resid, 'LARS', 0,0,[],0)';
lars_plot = lars_plot./[sF(1:r)'*ones(length(lars_plot),1)'];
for i = 1:r
pr(i) = plot(1:length(lars_plot), lars_plot(i,:), 'color', col(i,:));
end
ind_lars = 1+(s_lars*(size(lars_plot,2)-1));
plot([ind_lars ind_lars], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
xlabel('Iteration');
ylim([-1 4]);
xlim([1 21]);
xticks(1:1:21);
xticklabels(0:1:20);
set(pr, 'LineWidth', 1);
```

```
set(gca,'FontSize',16)
saveas(fig_lars, 'fig_lars', 'epsc');

%% Elastic net
lambda2 = [0, 0.01, 0.1, 1, 10, 100]; % range of lambda_2
best_resid = inf;
s_en_opt = 0;
en_plot2 = zeros(r, length(lambda2));
for i=1:length(lambda2)
    [s_en,beta_temp,mean_resid,std_resid] = cv(@larsen, 10, 1000, 1, ...
        X_factor_norm, resid, lambda2(i));
    en_plot2(:,i) = beta_temp';
    if mean_resid(s_en*length(mean_resid)+1) < best_resid
        best_resid = mean_resid(s_en*length(mean_resid)+1);  % store residuals
        beta_en = beta_temp;                                 % store ...
            coefficients
        opt_lambda2 = lambda2(i);                            % store lambda_2
        opt_index = i;                                       % store index ...
            lambda_2
        best_resid = mean_resid(s_en*length(mean_resid)+1);  % store residuals
        rm_en = mean_resid;                                  % store mean SSR
        rs_en = std_resid;                                   % store std. SSR
        s_en_opt = s_en;                                     % index out of ...
            1,000
    end
end
beta_en = real(beta_en'); % convert to column vector
% Plot CV error with best lambda_2
fig_en_cv = cv_plot(s_en_opt,rm_en,rs_en);
xlabel('Iteration');
xticks([((1:r+1)-1)./r]);
xticklabels(0:r);
set(gca,'FontSize',14);
saveas(fig_en_cv, 'fig_en_cv', 'epsc');
% Plot coefficients as a function of iterations with best lambda_2
fig_en = figure;
hold on
en_plot = larsen(X_factor_norm, resid, opt_lambda2)';
en_plot = en_plot./[sF(1:r)'*ones(size(en_plot,2),1)'];
en_plot2 = en_plot2./[sF(1:r)'*ones(size(en_plot2,2),1)'];
for i = 1:r
pr(i) = plot(1:length(en_plot), en_plot(i,:), 'color', col(i,:));
end
ind_en = 1+(s_en_opt*(size(en_plot,2)-1));
plot([ind_en ind_en], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
ylim([-1 4]);
xlim([1 21]);
```

```
xticks(1:1:21);
xticklabels(0:1:20);
xlabel('Iteration');
set(gca,'FontSize',16);
set(pr, 'LineWidth', 1);
saveas(fig_en, 'fig_en_iter', 'epsc');
% Plot coefficients as a function of lambda_2
fig_en2 = figure;
hold on
for i = 1:r
pr(i) = plot(0:length(lambda2)-1, en_plot2(i,:), 'color', col(i,:));
end
plot([opt_lambda2 opt_lambda2], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
xlabel('\lambda_2');
xlim([0 length(lambda2)-1]);
ylim([-1 4]);
xticks(0:length(lambda2)-1);
xticklabels(lambda2);
set(gca,'FontSize',16);
set(pr, 'LineWidth', 1);
saveas(fig_en2, 'fig_en_lambda', 'epsc');
%% Bagging
c = 1.96;                   % bagging coefficient c
[beta_bag] = bagging(train_y(q+1:end,:), X_auto, X_factor, c);
% Plot coefficients as a function of c
fig_bag = figure;
hold on
c_range = 0:0.01:2.5;
bag_plot = zeros(length(beta_bag),length(c_range));
for i = 1:length(c_range)
    bag_plot(:,i) = bagging(train_y(q+1:end,:), X_auto, X_factor, c_range(i))';
end
for i = 1:r
pr(i) = plot(c_range, bag_plot(1+p+i,:), 'color', col(i,:));
end
plot([c c], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
xlabel('c');
ylim([-1 4]);
set(gca,'FontSize',16);
set(pr, 'LineWidth', 1);
saveas(fig_bag, 'fig_bag', 'epsc');

%% Boosting
```

```
M = 100;                    % max iteration count
nu = 0.5;                   % step-length parameter
[beta_boost,boost_plot,ind_boost] = boosting(resid, X_factor, nu, M);
% Plot coefficients as a function of the iterations with nu = 0.5
fig_boost = figure;
hold on
for i = 1:r
pr(i) = plot(1:size(boost_plot,2), boost_plot(i,:), 'color', col(i,:));
end
plot([ind_boost ind_boost], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
xlabel('Iteration');
ylim([-1 4]);
xlim([0 100]);
xticks(0:20:100);
set(pr, 'LineWidth', 1);
set(gca,'FontSize',16);
saveas(fig_boost, 'fig_boost', 'epsc');
% Plot coefficients as a function of nu
fig_boost2 = figure;
hold on
nu_range = 0:0.01:1;
boost_plot2 = zeros(length(beta_boost), length(nu_range));
for i=1:length(nu_range)
    boost_plot2(:,i) = boosting(resid, X_factor, nu_range(i), M);
end
for i = 1:r
pr(i) = plot(nu_range, boost_plot2(i,:), 'color', col(i,:));
end
plot([nu nu], [-1 4],'--', 'color', 'k', 'LineWidth', 1)
leg = legend(pr(1:20),'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', ...
    '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', 'location', ...
    'eastoutside');
title(leg, 'Factor');
ylim([-1 4]);
xlim([0 1]);
xticks(0:0.2:1)
xlabel('\nu');
set(gca,'FontSize',16);
set(pr, 'LineWidth', 1);
saveas(fig_boost2, 'fig_boost2', 'epsc');
```

Cross-Validation Plot

```
function [fig] = cv_plot(s_opt, res_mean, res_std)
% CV_PLOT  Simple plotting function for cross-validation results.
%    CV_PLOT(S_OPT, RES_MEAN, RES_STD) plots the mean reconstruction error
```

```
%    with error bars resulting from the function CV. The optimal model
%    choice is marked with a dashed black line.
%
%    Original function: @cvplot
%    Orginal author: Karl Skoglund, IMM, DTU, kas@imm.dtu.dk
%    Author: Fabian Neefjes

fig = figure;
hold on;
s_sub = linspace(0, 1, 17);
s_sub = s_sub(2:end-1);
t_sub = round(s_sub*length(res_mean));
errorbar(s_sub, res_mean(t_sub), res_std(t_sub), 'bx');
s = linspace(0,1,length(res_mean));
p = plot(s, res_mean);
set (p, 'LineWidth',1);
ax = axis;
line([s_opt s_opt], [ax(3) ax(4)], 'Color', 'k', 'LineStyle', '--', ...
    'LineWidth', 1);
ylabel('Mean sum of squared residuals');
```

Forecasting Real GDP Growth in Nigeria

This subsection includes code that is only used to forecast real GDP growth in Nigeria.

Main File

```
%   This program forecasts Nigerian GDP growth and has excel output with
%   forecasting performance, in addition to coefficients and factor
%   loadings for the last forecast in the sample. Also makes plots of the
%   forecasts and Nigerian GDP growth.
%   To obtain the same results as in the paper, run using the following
%   configurations:
%   1. lag = 0, v = 5, data = 1:3
%   2. lag = 3, v = 5, data = 1
%
%   Author: Fabian Neefjes

%% Configuration
clear;
clc;
rng('default')           % random number generator
lag = 0;                 % additional lags included (0=F(-1), 1=F(-2), etc.)
v = 5;                   % size forecast sample
for data = 1:3           % 1 = Africa, 2 = other, 3 = both
%% Initialisation
p_hat = zeros(v,1);           % estimated number of lags
r_hat = zeros(v,1);           % estimated number of factoors included
k_auto = zeros(v,100);        % # non-zero coefs
k_bag = zeros(v,100);         % # non-zero coefs bagging with factors
k_boost = zeros(v,100);       % # non-zero coefs boosting with factors
```

```
k_boost2 = zeros(v,500);      % # non-zero coefs boosting with varibales
k_en = zeros(v,100);          % # non-zero coefs elastic net with factors
k_en2 = zeros(v,500);         % # non-zero coefs elastic net with variables
k_lars = zeros(v,100);        % # non-zero coefs LARS with factors
k_lars2 = zeros(v,500);       % # non-zero coefs LARS with variables
k_faar = zeros(v,100);        % # non-zero coefs FAAR
k_ridge = zeros(v,100);       % # non-zero coefs ridge with factors
k_rw = zeros(v,100);          % # non-zero coefs random walk
forecast_auto = zeros(v,1);   % forecasts AR
forecast_bag = zeros(v,1);    % forecasts bagging with factors
forecast_boost = zeros(v,1); % forecasts boosting with factors
forecast_boost2 = zeros(v,1);% forecats boosting with variables
forecast_en = zeros(v,1);     % forecasts elastic net with factors
forecast_en2 = zeros(v,1);    % forecasrs elastic net with variables
forecast_faar = zeros(v,1);   % forecasts FAAR
forecast_lars = zeros(v,1);   % forecasts LARS with factors
forecast_lars2 = zeros(v,1); % forecasts LARS with variables
forecast_ridge = zeros(v,1); % foreacsts ridge with factors
forecast_rw = zeros(v,1);     % forecasts random walk
%% Import data
dataset = importdata('AfricaGDP.xlsx');    % import African GDP growth data
var_africa = dataset.textdata(1,2:end)';  % extract variable names
X_africa = dataset.data(3:end-4,2:end);    % select 1963-2016
y_ind = 38;                                % index Nigeria Real GDP growth
dataset = importdata('Dataset.xlsx');      % import economic indicators
var_other = dataset.textdata.RawData(2:end,1);  % extract variable names
transform = dataset.textdata.RawData(2:end,2);  % extract transformations
raw_X = dataset.data.RawData(2:end,1:end-2);    % select 1981-2016
[vars, obs] = size(raw_X);
X_other = zeros(obs-1,vars);
for i = 1:vars                             % take difference or growth rate
    for j = 1:(obs-1)
        if transform(i) == "GR"            % compute growth rate
            X_other(j,i) = (raw_X(i,j+1) - raw_X(i,j))/ raw_X(i,j)*100;
        elseif transform(i) == "D"         % compute difference
            X_other(j,i) = raw_X(i,j+1) - raw_X(i,j);
        else
            error("Type not found.");
        end
    end
end

X_other(6,8) = mean([X_other(5,9), X_other(7,9)]);        % remove outlier ...
    change in inventories
X_other(9,8) = mean([X_other(8,9), X_other(10,9)]);       % remove outlier ...
    change in inventories
X_other(12,11) = mean([X_other(11,11),X_other(13,11)]);   % remove outlier ...
    govt exp. on edu
X_other(12,12) = mean([X_other(11,12),X_other(13,12)]);   % remove outlier ...
    govt exp. on health

```

```
if data == 1
    y = X_africa(:,y_ind);                                      % set y to ...
        Nigerian GDP
    X = [X_africa(:,1:y_ind-1) X_africa(:,y_ind+1:end)];  % set X to African ...
        GDP growth without y
    x_name = var_africa;
    x_name = [x_name(1:y_ind-1); x_name(y_ind+1:end)];    % remove y from ...
        var. names
elseif data == 2
    y = X_africa(20:end,y_ind);              % set y to Nigerian GDP from ...
        1981-2016
    X = X_other;                              % set X to economic indiactors
    x_name = var_other;
else
    y = X_africa(20:end,y_ind);              % set y to Nigerian GDP from ...
        1981-2016
    X_africa = X_africa(20:end,:);           % select African GDP growth from ...
        1981-2016
    X = [X_africa(:,1:y_ind-1) X_africa(:,y_ind+1:end) X_other]; % ...
        concatenate data sets and remove y
    x_name = [var_africa; var_other];
    x_name = [x_name(1:y_ind-1); x_name(y_ind+1:end)];          % remove y ...
        from var. names
end

y_name = var_africa(y_ind);                  % name dependent variable
t = size(X,1)-v;                             % size initial estimation sample

for n = 1:v
disp(n);                              % print iteration count (1, ..., v)
train_X = X(1:end-v+n-1,:);           % X estimation sample
train_y = y(1:end-v+n-1,:);           % y estimation sample
forecast_y = y(end-v+n,:);            % y to be predicted

if length(train_y)>44                              % check length data set
    dum = zeros(length(train_y),6);
    dum(end-44-n,1) = 1;                           % 1966 dummy
    dum(end-43-n,2) = 1;                           % 1967 dummy
    dum(end-42-n,3) = 1;                           % 1968 dummy
    dum(end-41-n,4) = 1;                           % 1969 dummy
    dum(end-40-n,5) = 1;                           % 1970 dummy
    dum(end-6-n,6) = 1;                            % 2004 dummy
else
    dum = zeros(length(train_y),1);
    dum(end-6-n,1) = 1;                            % 2004 dummy
end
%% Generate factors
mX = mean(train_X);                          % obtain mean of training X
sX = std(train_X);                           % obatin st. dev. of training X
train_X_norm = (train_X - mX) ./ sX;         % normalize training X
X_norm = (X - mX) ./ sX;                     % normalize all X
```

```
[pca_coeff, ~, pca_ev] = pca(train_X_norm);   % compute loadings from ...
    training X
train_factors = train_X_norm*pca_coeff;       % create factors for training X
factors = X_norm*pca_coeff;                   % create factors for X
mF = mean(train_factors);                     % obtain mean of training factors
sF = std(train_factors);                      % obtain st. dev. of training ...
    factors
train_factors_norm = (train_factors - mF) ./ sF; % standardize training factors
factors_norm = (factors - mF) ./ sF;          % standardize all factors

%% Select number of factors
r = number_of_factors(train_X, 20);       % use adjusted BIC to find number ...
    of factors
r_hat(n) = r;                             % store # of factors
train_factors = train_factors(:,1:r);     % select first r factors
factors = factors(:,1:r);
train_factors_norm = train_factors_norm(:,1:r);
factors_norm = factors_norm(:,1:r);

train_factors = lagmatrix(train_factors, 0:lag); % add more lags if applicable
factors = lagmatrix(factors, 0:lag);
train_factors_norm = lagmatrix(train_factors_norm, 0:lag);
factors_norm = lagmatrix(factors_norm, 0:lag);
train_X = lagmatrix(train_X, 0:lag);
X_reg = lagmatrix(X, 0:lag);
train_X_norm = lagmatrix(train_X_norm, 0:lag);
X_norm = lagmatrix(X_norm, 0:lag);

%% Find autoregressive order and compute AR(p) model
p = find_ar(train_y,5);                       % find autoregressive lag
p = max(p,1);                                 % set minimum p to 1
p_hat(n) = p;                                 % store p

y_lag = lagmatrix(train_y, 1:p);
X_auto = [ones(t+n-(p+1),1) y_lag(p+1:end,:)];
dum_reg = dum(p+1:end,any(dum(p+1:end,:)));
X_auto_dum = [X_auto dum_reg];
[beta_auto,~,auto_resid] = mvregress(X_auto_dum,train_y(p+1:end,:));
q = max(p,lag+1);                             % variable for start of ...
    estimation sample
dum_fc = dum(q+1:end,any(dum(q+1:end,:)));
X_auto = [ones(t+n-(q+1),1) y_lag(q+1:end,:) dum_fc];
resid = auto_resid(1+q-p:end,:);
y_lag = lagmatrix(y, 1:p);
X_forecast_auto = [ones(1,1) y_lag(t+n:t+n,:)];
forecast_auto(n) = X_forecast_auto*beta_auto(1:end-size(dum_reg,2));

%% Factor Augmented Autoregression
X_factor = train_factors(1+q-1:end-1,:);
X_faar = [X_auto(:,any(X_auto)) X_factor];
[beta_faar] = mvregress(X_faar, train_y(q+1:end,:));
```

```
beta_faar = [beta_faar(1:1+p); beta_faar(2+p+size(dum_fc,2):end)];
X_factor_forecast = factors(t+n-1:t+n-1,:);
X_forecast_faar = [X_forecast_auto X_factor_forecast];
forecast_faar(n) = X_forecast_faar*beta_faar;
k_faar(n) = sum(abs(beta_faar(2+p:end))>0);          % # non-zero factors

%% Ridge regression
k = 0:0.01:100;                                      % range of lambda
X_factor_norm = train_factors_norm(1+q-1:end-1,:);
[s_ridge,beta_ridge,~,~] = cv(@ridge2, 10, length(k), 0, X_factor_norm, ...
    resid, k);
beta_ridge = real(beta_ridge');                      % convert to column vector
X_factor_norm_forecast = factors_norm(t+n-1:t+n-1,:);
forecast_ridge(n) = forecast_auto(n) + X_factor_norm_forecast*beta_ridge;
k_ridge(n,1:r*(lag+1)) = abs(beta_ridge)>0;          % # non-zero factors

%% LARS
% ---Factors---
[~,beta_lars,~,~] = cv(@lars, 10, 1000, 1, X_factor_norm, resid, 'LARS', ...
    0,0,[],0);
beta_lars = real(beta_lars');                        % covert to column vector
forecast_lars(n) = forecast_auto(n) + X_factor_norm_forecast*beta_lars;
k_lars(n,1:r*(lag+1)) = abs(beta_lars)>0;            % # non-zero factors

% --Variables--
[~,beta_lars2,~,~] = cv(@lars, 10, 1000, 1, train_X_norm(1+q-1:end-1,:), ...
    resid, 'LARS', 0,0,[],0);
beta_lars2 = real(beta_lars2');                      % convert to column vector
X_norm_forecast = X_norm(t+n-1:t+n-1,:);
forecast_lars2(n) = forecast_auto(n) + X_norm_forecast*beta_lars2;
k_lars2(n,1:size(X_reg,2)) = abs(beta_lars2)>0;      % # non-zero factors

%% Elastic Net
lambda2 = [0, 0.01, 0.1, 1, 10, 100];                         % range of lambda_2
% ---Factors---
best_resid = inf;                                             % initialize to inf
for i=1:length(lambda2)
    [s_en,beta_temp,mean_resid,~] = cv(@larsen, 10, 1000, 1, X_factor_norm, ...
        resid, lambda2(i));
    if mean_resid(s_en*length(mean_resid)+1) < best_resid     % check if SSR ...
        is lower
        best_resid = mean_resid(s_en*length(mean_resid)+1);  % store residuals
        beta_en = beta_temp;                                  % store ...
            coefficients
        opt_lambda2 = lambda2(i);                             % store lambda_2
        opt_index = i;                                        % store index ...
            lambda_2
    end
end
beta_en = real(beta_en');                                     % convert to ...
    column vector
```

```
forecast_en(n) = forecast_auto(n) + X_factor_norm_forecast*beta_en;
k_en(n,1:r*(lag+1)) = abs(beta_en)>0;

% --Variables--
best_resid = inf;                                          % initalize to inf.
for i=1:length(lambda2)
    [s_en,beta_temp,mean_resid,~] = cv(@larsen, 10, 1000, 1, ...
        train_X_norm(1+q-1:end-1,:), resid, lambda2(i));
    if mean_resid(s_en*length(mean_resid)+1) < best_resid    % check if SSR ...
        is lower
        best_resid2 = mean_resid(s_en*length(mean_resid)+1); % store residuals
        beta_en2 = beta_temp;                                % store ...
            coefficients
        opt_lambda2 = lambda2(i);                            % store lambda_2
        opt_index2 = i;                                      % store index ...
            lambda_2
    end
end
beta_en2 = real(beta_en2');                                  % convert to ...
    column vector
forecast_en2(n) = forecast_auto(n) + X_norm_forecast*beta_en2;
k_en2(n,1:size(X_reg,2)) = abs(beta_en2)>0;
%% Bagging
c = 1.96;                                 % bagging parameter
[beta_bag] = bagging(train_y(q+1:end,:), X_auto, X_factor, c);
beta_bag = [beta_bag(1:1+p); beta_bag(2+p+size(dum_fc,2):end)];
forecast_bag(n) = X_forecast_faar*beta_bag;
k_bag(n,1:r*(lag+1)) = abs(beta_bag(2+p:end))>0;

%% Boosting
M = 100;                                  % max iteration count
nu = 0.5;                                 % step-length parameter

% ---Factors---
[beta_boost] = boosting(resid, X_factor, nu, M);
forecast_boost(n) = forecast_auto(n) + X_factor_forecast*beta_boost;
k_boost(n,1:r*(lag+1)) = abs(beta_boost)>0;

% --Variables---
[beta_boost2] = boosting(resid, train_X(1+q-1:end-1,:), nu, M);
forecast_boost2(n) = forecast_auto(n) + X_reg(t+n-1:t+n-1,:)*beta_boost2;
k_boost2(n,1:size(X_reg,2)) = abs(beta_boost2)>0;

%% Random Walk
forecast_rw(n) = train_y(end);

end
%% Compute MSE
mse_auto = mse(y(end-v+1:end),forecast_auto);
mse_faar = mse(y(end-v+1:end),forecast_faar);
mse_ridge = mse(y(end-v+1:end),forecast_ridge);
```

```
mse_lars = mse(y(end-v+1:end),forecast_lars);
mse_lars2 = mse(y(end-v+1:end),forecast_lars2);
mse_en = mse(y(end-v+1:end),forecast_en);
mse_en2 = mse(y(end-v+1:end),forecast_en2);
mse_bag = mse(y(end-v+1:end),forecast_bag);
mse_boost = mse(y(end-v+1:end),forecast_boost);
mse_boost2 = mse(y(end-v+1:end),forecast_boost2);
mse_rw = mse(y(end-v+1:end),forecast_rw);

%% Export output
header = ["Method" "Average k" "MSE" "MSE Ratio"];
methods = ["AR(1)" "Random Walk" "LARS" "Elastic Net" "Boosting" "FAAR (F)" ...
    "Ridge (F)" "LARS (F)" "Elastic Net (F)" "Bagging (F)" "Boosting (F)"];
mse_vec = [mse_auto, mse_rw, mse_lars2, mse_en2, mse_boost2, mse_faar, ...
    mse_ridge, mse_lars, mse_en, mse_bag, mse_boost];
mse_ratio = mse_vec./mse_auto;
k_sum = [sum(k_auto,2), sum(k_rw,2), sum(k_lars2,2), sum(k_en2,2), ...
    sum(k_boost2,2), sum(k_faar,2), sum(k_ridge,2), sum(k_lars,2), ...
    sum(k_en,2), sum(k_bag,2), sum(k_boost,2)];
k_mean = mean(k_sum);
p_vec = ["Mean p", mean(p_hat), "", ""];
r_vec = ["Mean r", mean(r_hat), "", ""];

% output forecast performance
filename_model = join(["emp_model" string(data), "L", string(lag+1), ...
    ".xls"], "");
writematrix([header; methods' k_mean' mse_vec' mse_ratio'; ["", "", "", ""]; ...
    p_vec; r_vec], filename_model);

% output factor loadings
for i=1:size(pca_coeff,2)
    factor_str(i) = join(["F", string(i)], "");
end
filename_factor = join(["emp_factor" string(data), "L", string(lag+1), ...
    ".xls"], "");
writematrix(["Variable", factor_str; x_name string(pca_coeff); "Eigenvalue" ...
    pca_ev'], filename_factor);

% output coefficients of benchmark models and models with variables
factor_str2 = factor_str(1:r);
x_name_str = x_name;
for i=1:lag
factor_str2 = [factor_str2 factor_str(1:r) + " L" + string(i+1)];
x_name_str = [x_name_str;  x_name + " L" + string(i+1)];
end
for i=1:p
    ar_str(i) = join(["AR(", string(i), ")"], "");
end
filename_coeff_X = join(["emp_coeff_X" string(data), "L", string(lag+1), ...
    ".xls"], "");
writematrix([["Coefficient"; "Constant"; ar_str; x_name_str], ...
```

```
    [methods(:,1:5); [beta_auto(1:end-size(dum_reg,2)); ...
    zeros(size(X_reg,2),1)], [0; 1; zeros(size(X_reg,2),1)], [zeros(p+1,1); ...
    beta_lars2], [zeros(p+1,1); beta_en2], [zeros(p+1,1); beta_boost2]]], ...
    filename_coeff_X);

% output coefficients of factor models
filename_coeff_F = join(["emp_coeff_F" string(data), "L", string(lag+1), ...
    ".xls"], "");
writematrix([["Coefficient"; "Constant"; ar_str; factor_str2'], ...
    [methods(:,6:end); beta_faar, [zeros(p+1,1); beta_ridge], [zeros(p+1,1); ...
    beta_lars], [zeros(p+1,1); beta_en], beta_bag, [zeros(p+1,1); ...
    beta_boost]]], filename_coeff_F);
clear ar_str factor_str factor_str2 x_name_str
%% Plot Forecasts
col=hsv(20);
plotname = join(["forecast" string(data), "L", string(lag+1)], "");
fig = figure;
hold on
x_plot = (2016-v+1:2016)';
fplot = plot(x_plot, y(end-v+1:end), '--', x_plot, forecast_auto, x_plot, ...
    forecast_rw, x_plot, forecast_lars2, x_plot, forecast_en2, x_plot, ...
    forecast_boost2, x_plot, forecast_faar,  x_plot, forecast_ridge, x_plot, ...
    forecast_lars, x_plot, forecast_en, x_plot, forecast_bag, x_plot, ...
    forecast_boost); % plot forecasts
lgd = legend(["Actual" methods], 'location', 'eastoutside');
set(lgd,'FontSize',20);
xticks(2016-v+1:2016);
ylabel('GDP Growth %');
xlabel('Year');
ylim([-2 12]);
fplot(1).Color = 'k';
fplot(10).Color = col(17,:);
fplot(11).Color = col(20,:);
fplot(12).Color = col(15,:);
set(gca,'FontSize',14);
set(fplot, 'LineWidth', 1);
set(gcf,'position',[10,10,900,500]);
saveas(fig, plotname, 'epsc');
%% Plot GDP Growth
if data == 1
    % Without oil price
    fig2 = figure;
    hold on
    yplot = plot(1963:2016, X_africa(:,y_ind), 'Color', 'k'); % plot y
    xlim([1963 2016]);
    xticks(1965:5:2015);
    yticks(-20:5:40);
    ylabel('GDP Growth %');
    xlabel('Year');
    grid on
    set(gca,'FontSize',16);
```

```
    set(yplot, 'LineWidth', 1);
    set(gcf,'position',[10,10,1000,300]);
    saveas(fig2, 'NigeraGDPgrowth','epsc');

    % With oil price
    fig3 = figure;
    set(fig3,'defaultAxesColorOrder',([0, 0, 0; 0.5, 0.5, 0.5]));
    hold on
    yyaxis left
    p1 = plot(1963:2016, X_africa(:,y_ind), 'Color', 'k'); % plot y
    xlim([1963 2016]);
    xticks(1965:5:2015);
    yticks(-20:5:40);
    ylabel('GDP Growth %');
    xlabel('Year');
    grid on
    yyaxis right
    p2 = plot(1982:2016, X_other(:,29), '--', 'Color', [.5,.5,.5]); % plot oil
    ylabel('% change US$/barrel Brent Crude');
    set(gca,'FontSize',16);
    set(p1, 'LineWidth', 1);
    set(p2, 'LineWidth', 1);
    set(gcf,'position',[10,10,1000,300]);
    saveas(fig3, 'NigeraGDPgrowth_oil','epsc');
end
end
```

Number of Factors

```
function [r] = number_of_factors(X,kmax)
% NUMBER_OF_FACTORS Computes the number of factors to be used.
%    [r] = number_of_factors(X, kmax) computes the number of factors r to be
%    used when applying PCA to T x N matrix X based on an adjusted BIC.
%    Paramter kmax denotes the maximum amount of factors to be considered.
%
%    Author: Fabian Neefjes
%    Reference: 'Determining the Number of Factors in Approximate Factor
%    Models' by Bai and Ng, 2002

[t, n] = size(X);
mX = mean(X);                       % mean of X
sX = std(X);                        % st. dev. of X
X = (X - mX) ./ sX;                 % normalize X
pca_coeff = pca(X);                 % apply PCA
factors = X*pca_coeff;              % obtain factors

if kmax>size(factors,2)
    error('The input kmax cannot exceed the number of principal components.');
end

```

```
sigma_hat = 0;
for l = 1:n                         % compute SSR of kmax
        [~, ~, k_resid] = mvregress(factors(:,1:kmax), X(:,l));
        sigma_hat = sigma_hat + sum(k_resid.^2);
end
sigma_hat = sigma_hat/(n*t);        % scale SSR of kmax
bic = zeros(kmax,1);

for i=1:kmax
    ssr = 0;
    for j = 1:n                     % compute SSR for each x
        [~, ~, resid] = mvregress(factors(:,1:i),X(:,j));
        ssr = ssr + sum(resid.^2);
    end
    bic(i) = ssr/(n*t) + i*sigma_hat*((n+t-i)*log(n*t)/(n*t)); % compute BIC
end
[~,r] = min(bic);                   % find minimum BIC and return index
```

Pie Chart Nigeria GDP Sectors

```
% This program creates a pie chart with a sectoral division of Nigeria's
% GDP over time.

%% Import data
dataset = importdata('piedata.xlsx'); % GDP data by sector
labels = dataset.textdata(2:7,1)';    % extract names
X = dataset.data(2:7,1:end);          % select 1981-2016
X81 = X(:,1);    % 1981
X99 = X(:,19);   % 1999
X16 = X(:,36);   % 2016

%% Plot Pie
fig = figure;
L = ['' '' '' '' '' ''];

% Plot 1981
ax1 = subplot(1,3,1);
H1 = pie(ax1,X81);
title(ax1,'1981');
T1 = H1(strcmpi(get(H1,'Type'),'text'));
P1 = cell2mat(get(T1,'Position'));
set(T1,{'Position'},num2cell(P1*0.6,2))
text(P1(:,1),P1(:,2),L(:))
set(gca,'FontSize',16);
set(T1, 'FontSize',16);

% Plot 1999
ax2 = subplot(1,3,2);
H2 = pie(ax2,X99);
```

```
title(ax2,'1999');
T2 = H2(strcmpi(get(H2,'Type'),'text'));
P2 = cell2mat(get(T2,'Position'));
set(T2,{'Position'},num2cell(P2*0.6,2))
text(P2(:,1),P2(:,2),L(:))
set(gca,'FontSize',16);
set(T2, 'FontSize',16);

% Plot 2016
ax3 = subplot(1,3,3);
H3 = pie(ax3,X16);
title(ax3,'2016')
T3 = H3(strcmpi(get(H3,'Type'),'text'));
P3 = cell2mat(get(T3,'Position'));
set(T3,{'Position'},num2cell(P3*0.6,2))
text(P3(:,1),P3(:,2),L(:))
set(gca,'FontSize',16);
set(T3, 'FontSize',16);

% Export output
colormap('parula');
set(gcf,'position',[10,10,1000,300])
legend(labels);
set(legend,'Position',[0.295750003586213 0.0439999801251623 0.43 0.115],...
    'NumColumns',5);
saveas(fig, 'NigeriaGDPpie','epsc');
```

Appendix B

This appendix provides an overview of the African GDP growth data set in Table 6 and the economic indicator data set in Table 7.

Table 6: Annual real GDP growth rates (%) and correlation with Nigeria of African countries from 1963-2016.

	1963-1981			**1982-1999**			**2000-2016**			**1963-2016**		
	GDP Growth	St. Dev.	Corr. Nig.	GDP Growth	St. Dev.	Corr. Nig.	GDP Growth	St. Dev.	Corr. Nig.	GDP Growth	St. Dev.	Corr. Nig.
Algeria [DZA]	7.67	9.82	0.28	2.22	2.73	0.00	3.68	1.45	0.61	4.59	6.43	0.24
Angola [AGO]	1.99	4.23	-0.42	2.24	8.05	0.13	7.09	7.78	-0.04	3.68	7.11	0.02
Benin [BEN]	3.56	3.36	-0.24	3.54	3.59	-0.18	4.24	1.64	-0.11	3.77	2.98	-0.16
Botswana [BWA]	12.33	6.37	0.31	8.19	4.61	0.25	4.27	4.43	0.11	8.41	6.13	0.13
Burkina Faso [BFA]	3.00	2.92	-0.05	4.64	4.06	-0.44	5.56	1.75	0.40	4.35	3.20	-0.05
Burundi [BDI]	4.68	6.54	-0.07	0.86	5.16	-0.27	2.68	2.78	-0.19	2.78	5.29	-0.10
Cabo Verde [CPV]	4.75	4.25	0.01	8.04	4.95	0.06	4.98	4.76	0.01	5.92	4.81	-0.05
Cameroon [CMR]	5.37	7.52	0.12	1.57	5.54	-0.41	4.31	1.25	-0.15	3.77	5.67	-0.01
Central African Republic [CAF]	1.58	2.95	0.30	1.57	5.03	-0.04	0.25	9.95	-0.18	1.16	6.43	-0.04
Chad [TCD]	-0.71	6.44	0.16	4.49	8.64	-0.13	7.42	9.20	0.18	3.58	8.67	0.12
Comoros [COM]	4.66	3.35	-0.07	2.00	3.74	-0.17	2.62	2.25	-0.06	3.13	3.35	-0.06
Congo, Dem. Rep. [COD]	1.61	4.35	0.09	-2.29	5.12	-0.01	4.79	4.06	0.01	1.31	5.30	0.17
Congo, Rep. [COG]	5.89	6.27	0.13	2.29	6.44	0.06	4.28	3.13	-0.28	4.19	5.65	0.04
Djibouti [DJI]	2.75	3.41	0.19	0.31	3.28	-0.16	4.26	1.64	-0.24	2.41	3.30	0.15
Egypt, Arab Rep. [EGY]	5.77	4.02	-0.04	4.93	2.02	-0.14	4.23	1.72	-0.11	5.00	2.85	-0.08
Equatorial Guinea [GNQ]	6.34	18.05	0.09	20.44	36.61	0.07	11.54	18.31	0.05	12.68	26.02	0.03
Eritrea [ERI]	3.27	8.36	0.00	8.49	5.27	-0.25	2.31	4.59	-0.35	4.71	6.82	-0.22
Ethiopia [ETH]	-0.33	6.28	-0.23	2.52	7.39	0.08	8.96	4.02	-0.77	3.54	7.13	-0.08
Gabon [GAB]	7.61	13.97	0.01	1.99	6.97	0.24	2.42	3.31	-0.13	4.10	9.59	0.02
Gambia, The [GMB]	4.85	3.63	0.17	3.39	2.76	0.24	3.41	3.56	0.13	3.91	3.35	0.17
Ghana [GHA]	1.31	5.29	0.17	3.64	3.62	0.37	6.09	2.75	-0.06	3.59	4.46	0.22
Guinea [GIN]	2.83	1.28	0.17	3.73	1.35	0.13	3.70	2.03	-0.31	3.40	1.60	-0.01
Guinea-Bissau [GNB]	0.92	7.49	0.34	2.22	8.31	0.21	3.15	2.53	-0.37	2.06	6.63	0.14
Ivory Coast [CIV]	7.15	6.57	0.46	1.63	3.14	-0.05	2.94	4.54	-0.34	3.99	5.46	0.17
Kenya [KEN]	6.89	6.08	-0.16	3.07	2.21	-0.01	4.56	2.37	-0.27	4.89	4.29	-0.10
Lesotho [LSO]	6.09	9.37	-0.05	4.36	2.30	0.19	3.81	1.85	0.07	4.79	5.79	-0.02
Liberia [LBR]	3.28	3.68	0.15	-1.54	33.55	-0.01	4.30	11.03	-0.74	1.99	20.22	-0.05
Libya [LBY]	-0.35	13.36	-0.03	-0.98	7.59	0.06	2.45	31.64	0.14	0.32	19.58	0.09
Madagascar [MDG]	1.42	4.39	0.18	1.60	2.54	0.29	2.93	5.03	0.18	1.95	4.08	0.23
Malawi [MWI]	5.11	5.53	-0.39	3.52	5.90	0.12	4.28	3.36	0.16	4.32	5.04	-0.06
Mali [MLI]	3.77	5.33	0.19	3.76	6.13	-0.19	4.84	3.71	0.22	4.10	5.11	0.11
Mauritania [MRT]	4.76	8.12	0.01	2.34	3.98	0.00	4.42	4.62	0.11	3.85	5.92	0.08
Mauritius [MUS]	7.33	5.87	0.34	5.51	2.15	-0.21	4.33	2.04	-0.13	5.78	4.00	0.14
Morocco [MAR]	6.28	3.62	0.24	4.37	5.27	-0.27	4.26	1.76	0.22	5.01	3.90	0.04
Mozambique [MOZ]	5.45	3.42	0.24	4.12	9.62	0.33	7.22	2.32	-0.05	5.56	6.07	0.23
Namibia [NAM]	5.34	1.36	0.25	2.48	2.73	0.03	4.73	2.85	-0.18	4.20	2.65	0.11
Niger [NER]	1.64	6.91	-0.26	1.15	6.11	-0.37	4.75	3.56	0.03	2.45	5.88	-0.15
Nigeria [NGA]	4.35	10.60	0.35	1.16	5.83	0.11	7.00	7.36	0.05	4.12	8.43	0.30
Rwanda [RWA]	4.41	7.36	0.05	1.98	16.05	0.07	7.91	2.44	-0.50	4.70	10.42	0.07
Sao Tome and Principe [STP]	7.92	8.88	-0.17	0.63	3.66	-0.10	4.70	2.32	0.23	4.48	6.48	0.02
Senegal [SEN]	2.13	4.92	0.08	2.73	3.16	0.09	4.20	1.75	0.20	2.98	3.62	0.13
Seychelles [SYC]	6.12	8.27	0.19	4.58	4.44	0.28	3.24	4.87	-0.42	4.70	6.17	0.05
Sierra Leone [SLE]	3.29	2.77	0.25	-1.26	6.14	0.00	6.24	10.04	0.34	2.70	7.39	0.27
Somalia [SOM]	3.07	10.57	0.02	3.47	10.97	-0.02	0.54	5.21	-0.07	2.41	9.30	-0.03
South Africa [ZAF]	4.55	2.15	-0.11	1.35	2.22	0.62	2.96	1.86	0.05	2.98	2.45	0.16
Sudan [SDN]	2.93	6.62	0.02	3.84	5.50	0.29	5.27	3.32	0.28	3.97	5.37	0.14
Tanzania [TZA]	3.82	1.25	0.12	3.57	1.90	-0.03	6.64	1.18	0.04	4.62	2.00	0.22
Togo [TGO]	5.72	5.97	-0.26	2.29	7.28	0.19	3.00	2.35	0.09	3.72	5.75	-0.02
Tunisia [TUN]	6.48	4.27	0.15	4.08	2.83	0.20	3.35	2.09	0.19	4.69	3.46	0.13
Uganda [UGA]	4.70	5.35	0.33	5.46	3.64	0.36	6.39	2.19	0.14	5.49	3.99	0.32
Zambia [ZMB]	3.19	5.82	-0.05	1.02	3.84	0.20	6.35	2.10	0.20	3.46	4.72	0.16
Zimbabwe [ZWE]	5.39	7.01	0.14	3.03	4.58	-0.10	0.04	9.94	-0.49	2.92	7.62	-0.19
Average	4.31	6.07	0.05	3.16	6.16	0.01	4.50	4.59	-0.06	3.98	6.31	0.04
Minimum	-0.71	1.25	-0.42	-2.29	1.35	-0.44	0.04	1.18	-0.77	0.32	1.60	-0.22
Maximum	12.33	18.05	0.46	20.44	36.61	0.62	11.54	31.64	0.61	12.68	26.02	0.32

The correlation is the one year lagged correlation of economic growth of the African country with Nigeria. Nigeria is excluded from the average correlation.

Table 7: Overview of economic indicators from 1982-2016.

Series	Trans.	Type	Mean	St. Dev.	Notes	Source
Monthly average official exchange rate (₦/$US)	GR	Exchange Rate	25.98	59.59		2018 SB, CBN
Real effective exchange rate	GR	Exchange Rate	1.81	28.59	Base year 2010	WDI, World Bank
Commercial banks' loans and advances (₦ billion)	GR	Interest Rate	26.90	28.88		2018 SB, CBN
Deposit rate 3 months (%)	D	Interest Rate	0.06	3.28	Period weighted average	2018 SB, CBN
Deposit rate over 12 months (%)	D	Interest Rate	-0.03	3.93	Period weighted average	2018 SB, CBN
Interest rate spread (%)	D	Interest Rate	0.18	1.83	Lending rate minus deposit rate	WDI, World Bank
Lending interest rate (%)	D	Interest Rate	0.23	3.18		WDI, World Bank
Changes in inventories (₦)	GR	Investment	23.61	81.79	At constant local currency, outliers in 1986 and 1989 set to average growth rate of the preceding and proceeding year	WDI, World Bank
Gross fixed capital formation (₦)	GR	Investment	-0.35	13.94	At constant local currency	WDI, World Bank
Total savings (₦ billion)	GR	Investment	25.14	17.08		2018 SB, CBN
Government expenditure on education (₦ billion)	GR	Labour, Human Capital	30.71	104.26	Outlier for 1993 set to average growth rate of the preceding and proceeding year	2018 SB, CBN
Government expenditure on health (₦ billion)	GR	Labour, Human Capital	42.37	163.74	Outlier in 1993 set to average growth rate of the preceding and proceeding year	2018 SB, CBN
Gross enrolment ratio, primary, both sexes (%)	D	Labour, Human Capital	-0.52	4.97	Missing data for 1996, 1997, and 2015 set to the average level of the preceding and proceeding year	Edstats Query, World Bank
Labor force, total	GR	Labour, Human Capital	2.60	0.21	Missing data for 1981 until 1989 labour force computed assuming same growth rate as population from 15-64*	WDI, World Bank
Money supply (M1) (₦ billion)	GR	Money Supply	23.48	17.85		2018 SB, CBN
Total monetary liabilities (M2) (₦ billion)	GR	Money Supply	23.99	15.02		2018 SB, CBN
All Shares Index	GR	Other	22.04	34.45	December observations, missing data for 1984 set to January 1985, for 1981 until 1983 set to base level 100	2018 SB, CBN
Inflation, consumer prices (%)	D	Other	-0.15	15.82		WDI, World Bank
Government expenditure (₦ billion)	GR	Output	22.29	27.43		2018 SB, CBN
Households and NPISHs final consumption expenditure (₦)	GR	Output	4.67	16.17	At constant local currency	WDI, World Bank
Real GDP - agriculture (₦ billion)	GR	Output	6.06	9.41	At 2010 constant prices	2018 SB, CBN
Real GDP - construction (₦ billion)	GR	Output	3.74	10.53	At 2010 constant prices	2018 SB, CBN
Real GDP - crude petroleum & natural gas (₦ billion)	GR	Output	0.75	9.03	At 2010 constant prices	2018 SB, CBN
Real GDP - manufacturing (₦ billion)	GR	Output	4.78	11.94	At 2010 constant prices	2018 SB, CBN
Real GDP - services (₦ billion)	GR	Output	5.73	4.39	At 2010 constant prices	2018 SB, CBN
Real GDP - trade (₦ billion)	GR	Output	5.76	7.34	At 2010 constant prices	2018 SB, CBN
Current account (₦ billion)	GR	Trade	24.51	164.11	Includes estimates of informal cross-border trade	2018 SB, CBN
Direct Investment (₦ million)	GR	Trade	54.67	132.14		2018 SB, CBN
Global price of Brent crude oil (US$/barrel)	GR	Trade	3.91	25.97	Yearly average based on monthly price	FRED Economic Database, Federal Reserve Bank of St. Louis
Imports (₦ billion)	GR	Trade	31.09	71.05	Includes estimates of informal cross-border trade	2018 SB, CBN
India GDP (US$)	GR	Trade	6.11	1.99	At 2010 constant US$	WDI, World Bank
Non-oil exports (₦ billion)	GR	Trade	40.06	85.96	Includes estimates of informal cross-border trade	2018 SB, CBN
Oil exports (₦ billion)	GR	Trade	33.97	76.89	Includes estimates of informal cross-border trade	2018 SB, CBN
United States GDP (US$)	GR	Trade	2.71	1.92	At 2010 constant US$	WDI, World Bank
World GDP (US$)	GR	Trade	2.92	1.29	At 2010 constant US$	WDI, World Bank

*Population from 15-64 used to impute the labour force is retrieved from the WDI database from the World Bank.
Summary statistics are of transformed variables, where GR indicates the annual percentage growth rate is used and D indicates the difference is used. The source 2018 SB, CBN refers to the 2018 Statistical Bulletin of the Central bank of Nigeria and WDI refers to the World Development Indicators database of the World Bank.

Appendix C

This appendix includes coefficient estimates and factor loadings of the models used to forecast real Nigerian GDP growth. Tables 8, 9, and 10 correspond to Section 5.3.1, Tables 11 and 12 correspond to Section 5.3.2, and Tables 13 and 14 correspond to results of Section 5.3.3.

Table 8: Coefficient estimates used to forecast real GDP growth in Nigeria with African GDP growth data using one lag.

	Benchmark	Variable			Factor					
	AR(1)	LARS	Elastic Net	Boosting	FAAR	Ridge	LARS	Elastic Net	Bagging	Boosting
Constant	2.75				3.43				3.43	
AR(1)	0.13				0.01				0.01	
Benin [BEN]				-0.31						
Burkina Faso [BFA]				-0.14						
Eritrea [ERI]				-0.14						
Guinea-Bissau [GNB]				0.18						
Kenya [KEN]				-0.14						
Mali [MLI]				0.03						
Morocco [MAR]				-0.05						
Mozambique [MOZ]				0.11						
Niger [NER]				-0.17						
Senegal [SEN]				0.06						
Seychelles [SYC]				0.23						
Sierra Leone [SLE]				0.15						
Somalia [SOM]				0.03						
Uganda [UGA]		0.00	0.20	0.16						
Zimbabwe [ZWE]				-0.04						
Factor 1					-0.70	-0.39	0.00	0.00		-0.23
Factor 2					0.11	0.03				
Factor 3					-0.03	-0.01				
Factor 4					0.17	0.09				
Factor 5					-0.38	-0.14				
Factor 6					0.36	0.23				

Coefficients are estimated with data from 1963 until 2015 to forecast 2016. Variables and factors are normalised for ridge regression, LARS and elastic net. Variables not selected by any model are not shown. The random walk model is omitted, as no coefficients are estimated.

Table 9: Coefficient estimates used to forecast real GDP growth in Nigeria with African GDP growth data using four lags.

	Benchmark	Variable			Factor					
	AR(1)	LARS	Elastic Net	Boosting	FAAR	Ridge	LARS	Elastic Net	Bagging	Boosting
Constant	2.75				4.02				4.02	
AR(1)	0.13				-0.10				-0.10	
Benin [BEN] L1				-0.22						
Chad [TCD] L1				-0.02						
Guinea-Bissau [GNB] L1				0.16						
Ivory Coast [CIV] L1				0.03						
Kenya [KEN] L1				-0.03						
Mozambique [MOZ] L1				0.01						
Niger [NER] L1				-0.15						
Rwanda [RWA] L1				0.01						
Seychelles [SYC] L1				0.09						
Sierra Leone [SLE] L1				0.06						
Sudan [SDN] L1				0.04						
Togo [TGO] L1				-0.03						
Angola [AGO] L2				0.02						
Benin [BEN] L2				-0.06						
Burundi [BDI] L2				-0.04						
Chad [TCD] L2				0.07						
Ethiopia [ETH] L2				0.16						
Guinea-Bissau [GNB] L2				-0.07						
Liberia [LBR] L2				0.01						
Malawi [MWI] L2				-0.02						
Mali [MLI] L2				-0.03						
Niger [NER] L2				0.10						
Senegal [SEN] L2				-0.26						
Seychelles [SYC] L2				-0.05						
Somalia [SOM] L2				-0.02						
Uganda [UGA] L2		0.02	3.52							
Algeria [DZA] L3				-0.16						
Chad [TCD] L3				-0.02						
Congo, Rep. [COG] L3				-0.02						
Eritrea [ERI] L3				-0.09						
Ethiopia [ETH] L3				0.01						
Gabon [GAB] L3				-0.01						
Liberia [LBR] L3				0.01						
Madagascar [MDG] L3				0.15						
Mauritius [MUS] L3				0.16						
Mozambique [MOZ] L3				0.14						
Senegal [SEN] L3				0.02						
Sudan [SDN] L3				0.10						
Togo [TGO] L3				-0.09						
Zimbabwe [ZWE] L3				0.01						
Angola [AGO] L4				-0.02						
Central African Republic [CAF] L4				0.12						
Egypt, Arab Rep. [EGY] L4				-0.07						
Equatorial Guinea [GNQ] L4				0.00						
Eritrea [ERI] L4				-0.02						
Ethiopia [ETH] L4				-0.04						
Lesotho [LSO] L4				-0.05						
Mali [MLI] L4				0.11						
Niger [NER] L4				-0.17						
Senegal [SEN] L4				0.15						
Somalia [SOM] L4				0.02						
Sudan [SDN] L4				-0.06						
Zambia [ZMB] L4				0.08						
Zimbabwe [ZWE] L4				0.03						
Factor 1 L1					0.58	-0.10				
Factor 2 L1					1.90	0.10				
Factor 3 L1					-0.04	0.02				
Factor 4 L1					0.92	0.13				
Factor 5 L1					-1.49	-0.15				
Factor 6 L1					0.09	0.20				
Factor 1 L2					-2.25	-0.47			-1.64	-0.25
Factor 2 L2					-0.81	-0.24				
Factor 3 L2					0.83	-0.05				
Factor 4 L2					0.01	-0.15				
Factor 5 L2					0.37	0.02				
Factor 6 L2					-0.81	-0.25				
Factor 1 L3					0.57	-0.42	-0.01	-0.10		-0.39
Factor 2 L3					-1.31	-0.25				
Factor 3 L3					-0.41	-0.09				
Factor 4 L3					0.52	0.09				
Factor 5 L3					-0.55	-0.28				
Factor 6 L3					1.30	0.22				
Factor 1 L4					0.09	-0.31				
Factor 2 L4					-0.63	-0.16				
Factor 3 L4					-0.21	0.04				
Factor 4 L4					0.65	0.12				
Factor 5 L4					-1.33	-0.26				
Factor 6 L4					0.09	0.03				

Coefficients are estimated with data from 1963 until 2015 to forecast 2016. Variables and factors are normalised for ridge regression, LARS and elastic net. Variables not selected by any model are not shown. The random walk model is omitted, as no coefficients are estimated. The lag order indicated with L.

Table 10: Factor loadings of the first twenty factors generated using African GDP growth data.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
Algeria [DZA]	-0.08	0.20	-0.12	0.14	-0.01	-0.09	-0.01	-0.11	-0.02	0.28	0.22	-0.03	0.09	-0.22	0.05	0.12	0.09	0.24	-0.01	-0.20
Angola [AGO]	-0.25	-0.08	-0.01	0.02	0.23	-0.18	-0.02	0.06	-0.07	-0.09	-0.01	-0.11	0.08	0.03	0.20	0.02	0.11	-0.10	-0.18	-0.02
Benin [BEN]	-0.05	-0.03	0.04	-0.14	0.10	0.10	0.12	-0.10	-0.02	0.46	0.16	-0.05	-0.04	-0.17	-0.07	0.09	-0.02	-0.14	0.04	0.33
Botswana [BWA]	0.01	0.32	-0.04	-0.04	0.01	0.15	-0.04	0.01	-0.12	-0.17	-0.06	0.05	-0.13	0.04	0.13	0.02	0.25	0.06	-0.29	-0.08
Burkina Faso [BFA]	-0.12	-0.25	-0.04	0.17	-0.08	0.18	-0.12	0.11	-0.18	0.13	-0.04	-0.21	0.05	0.17	-0.10	-0.09	0.09	-0.19	-0.04	-0.13
Burundi [BDI]	-0.07	0.07	0.05	0.02	-0.17	-0.16	0.02	0.30	-0.11	0.12	-0.19	0.29	-0.20	-0.24	-0.10	-0.11	-0.01	0.09	-0.13	0.05
Cabo Verde [CPV]	0.03	-0.20	-0.18	-0.17	0.11	-0.14	0.30	-0.07	-0.13	-0.07	-0.10	0.04	-0.03	-0.05	0.07	-0.18	0.11	0.00	-0.17	0.09
Cameroon [CMR]	0.02	0.08	-0.01	0.07	0.32	-0.06	-0.15	0.13	-0.28	0.11	0.18	0.21	0.18	0.08	0.02	-0.22	-0.12	-0.04	0.02	0.00
Central African Republic [CAF]	-0.02	-0.03	-0.16	0.01	-0.02	0.00	0.02	-0.01	-0.27	-0.03	-0.16	-0.29	0.23	-0.09	0.04	-0.16	0.04	0.29	0.35	0.33
Chad [TCD]	-0.06	-0.17	-0.01	0.10	-0.01	-0.23	-0.25	0.08	-0.18	-0.12	0.16	-0.20	-0.08	-0.07	-0.01	-0.04	-0.02	0.17	-0.32	-0.04
Comoros [COM]	0.10	0.16	0.11	0.13	0.21	0.16	0.25	0.21	0.05	0.12	-0.18	-0.01	-0.08	-0.09	0.03	0.03	0.10	0.01	0.13	-0.10
Congo, Dem. Rep. [COD]	-0.23	0.12	0.23	-0.06	0.14	-0.11	-0.25	0.12	-0.08	-0.08	-0.03	0.01	0.09	-0.02	0.05	0.01	0.08	0.06	0.13	0.08
Congo, Rep. [COG]	0.05	0.14	0.32	0.15	0.07	0.09	0.02	-0.16	-0.25	0.09	-0.05	-0.10	0.01	0.10	-0.01	-0.19	0.02	0.00	-0.01	-0.13
Djibouti [DJI]	-0.17	0.15	0.29	0.03	0.08	-0.16	-0.17	-0.01	0.03	-0.03	-0.13	-0.16	-0.06	-0.25	0.03	-0.05	-0.05	0.11	0.04	-0.08
Egypt, Arab Rep. [EGY]	0.07	0.05	-0.13	0.22	0.10	-0.19	0.22	0.11	-0.09	0.16	-0.14	0.02	0.03	0.19	0.27	0.22	0.15	0.20	-0.13	0.09
Equatorial Guinea [GNQ]	0.01	-0.18	-0.19	-0.03	0.36	0.12	-0.01	-0.17	-0.06	-0.05	-0.02	-0.11	-0.17	-0.13	-0.17	-0.04	0.07	-0.08	0.00	-0.18
Eritrea [ERI]	0.29	-0.05	0.02	-0.05	-0.02	0.11	-0.26	-0.01	-0.16	0.00	-0.12	-0.03	-0.11	0.12	0.20	-0.05	0.07	-0.10	-0.04	0.11
Ethiopia [ETH]	-0.16	-0.20	0.20	-0.04	0.08	0.07	0.01	-0.15	0.22	0.01	-0.14	0.08	0.12	0.12	0.22	0.08	0.07	0.12	0.06	-0.09
Gabon [GAB]	-0.06	0.12	-0.01	0.23	-0.27	0.05	-0.04	-0.14	-0.13	-0.12	-0.10	-0.15	-0.14	0.03	0.14	0.16	-0.05	-0.12	0.13	0.23
Gambia, The [GMB]	0.03	0.05	-0.02	0.22	0.00	-0.24	0.12	0.00	-0.03	-0.24	0.20	0.10	-0.25	0.10	0.08	0.17	-0.34	-0.07	0.22	-0.18
Ghana [GHA]	-0.18	-0.13	0.02	-0.32	-0.07	0.11	-0.04	0.16	-0.01	0.06	0.09	-0.04	-0.12	-0.13	0.16	-0.03	-0.04	0.04	0.09	-0.12
Guinea [GIN]	-0.11	-0.14	-0.23	-0.17	-0.07	0.13	-0.02	0.05	-0.05	0.03	0.09	0.03	-0.23	0.06	0.25	0.18	0.02	0.29	-0.06	0.20
Guinea-Bissau [GNB]	0.05	-0.04	0.00	0.03	-0.04	0.19	-0.16	-0.09	-0.20	0.02	0.17	0.14	0.32	-0.17	0.39	0.11	-0.08	-0.28	-0.07	-0.17
Ivory Coast [CIV]	-0.14	0.18	-0.25	0.10	0.02	0.04	-0.03	0.05	0.02	-0.01	0.01	0.28	0.10	0.08	0.03	-0.22	-0.28	0.06	0.13	-0.04
Kenya [KEN]	-0.11	0.28	-0.04	-0.10	0.17	0.15	-0.19	0.01	0.18	-0.13	-0.06	-0.08	0.15	0.12	-0.08	-0.01	0.11	-0.03	-0.12	0.12
Lesotho [LSO]	-0.01	0.11	-0.19	-0.25	-0.12	0.01	0.12	0.31	-0.20	0.03	0.13	-0.03	0.05	0.00	0.08	0.01	0.10	-0.17	0.20	-0.18
Liberia [LBR]	-0.13	-0.06	-0.11	0.09	0.31	0.12	0.01	-0.03	-0.27	0.01	-0.17	0.06	-0.22	-0.08	-0.12	0.03	-0.05	0.05	0.08	-0.15
Libya [LBY]	-0.02	0.06	-0.08	0.09	0.01	0.02	-0.11	-0.04	0.05	0.02	0.37	-0.26	-0.28	0.26	0.07	-0.14	0.12	0.10	0.31	-0.01
Madagascar [MDG]	-0.14	-0.06	-0.05	0.05	0.17	-0.06	0.11	0.06	0.33	-0.11	-0.18	-0.20	-0.07	-0.12	0.24	-0.12	0.02	-0.10	0.17	-0.16
Malawi [MWI]	-0.13	0.06	0.04	0.03	0.04	0.28	0.10	0.14	0.02	-0.20	-0.02	-0.09	0.25	0.25	-0.18	0.30	-0.12	0.03	-0.08	0.11
Mali [MLI]	-0.06	-0.12	-0.21	0.23	-0.11	-0.01	-0.22	0.13	0.12	0.14	-0.19	-0.05	-0.02	-0.19	-0.14	0.15	-0.03	-0.12	-0.17	-0.03
Mauritania [MRT]	-0.23	-0.04	0.07	0.07	-0.14	-0.01	0.29	-0.04	0.03	0.11	-0.06	0.01	0.02	0.16	0.12	-0.27	-0.18	-0.09	-0.19	0.05
Mauritius [MUS]	-0.07	0.16	-0.19	0.10	-0.21	0.17	0.01	0.04	-0.08	-0.13	0.01	0.03	0.13	-0.28	-0.09	0.04	0.21	0.03	0.11	-0.08
Morocco [MAR]	-0.12	0.10	-0.17	0.06	-0.17	-0.25	-0.13	0.02	0.01	0.13	-0.24	-0.11	0.06	0.15	-0.02	0.05	0.22	-0.19	0.10	-0.16
Mozambique [MOZ]	-0.24	-0.12	-0.13	-0.07	-0.02	0.14	-0.12	-0.27	0.03	-0.05	-0.05	0.17	-0.13	0.02	-0.16	0.02	0.09	-0.13	0.10	-0.03
Namibia [NAM]	-0.22	0.13	-0.02	0.01	0.03	0.02	0.03	0.08	-0.01	-0.21	0.15	0.06	-0.15	-0.13	-0.17	-0.19	0.01	-0.19	-0.05	0.30
Niger [NER]	-0.15	-0.13	-0.12	0.05	0.02	0.09	-0.12	0.22	0.10	0.26	-0.02	0.04	-0.02	0.32	-0.06	-0.17	-0.02	0.13	-0.04	-0.11
Rwanda [RWA]	-0.09	-0.06	0.13	0.19	0.16	0.30	0.15	0.23	-0.16	-0.05	-0.01	-0.07	-0.05	-0.08	-0.09	0.16	-0.12	0.15	0.01	0.02
Sao Tome and Principe [STP]	-0.26	0.17	-0.14	-0.02	-0.07	-0.13	0.13	-0.05	0.05	0.04	0.04	-0.07	0.14	0.09	-0.19	0.00	-0.09	0.04	-0.06	-0.03
Senegal [SEN]	-0.08	-0.13	0.01	0.37	-0.02	0.09	-0.08	-0.11	0.12	0.10	0.03	0.10	0.11	-0.13	0.15	-0.29	-0.02	0.03	0.04	0.25
Seychelles [SYC]	-0.05	0.16	-0.18	-0.01	0.02	0.29	0.00	0.16	0.15	-0.04	-0.07	0.04	-0.12	0.01	0.29	-0.13	-0.10	-0.04	-0.09	-0.02
Sierra Leone [SLE]	-0.14	0.05	0.23	-0.04	-0.02	0.00	-0.05	0.00	-0.04	0.13	0.13	0.31	-0.15	0.23	-0.11	0.05	0.38	0.05	0.06	0.02
Somalia [SOM]	0.11	-0.06	-0.23	0.04	0.22	-0.07	-0.16	-0.16	0.07	0.01	-0.14	0.32	0.14	0.01	-0.06	0.16	0.01	0.17	0.17	0.03
South Africa [ZAF]	-0.24	0.13	0.02	-0.06	0.05	-0.06	0.17	-0.21	-0.15	0.09	-0.14	0.08	-0.03	-0.02	0.14	0.00	0.17	-0.23	0.12	0.04
Sudan [SDN]	-0.02	-0.22	-0.01	0.30	-0.10	0.00	0.11	-0.03	-0.15	-0.28	0.01	0.21	-0.08	0.08	-0.02	-0.01	0.23	-0.07	0.03	0.06
Tanzania [TZA]	-0.23	-0.14	0.12	0.00	0.06	-0.06	-0.17	0.20	0.11	-0.09	0.08	0.15	-0.01	-0.11	0.17	0.19	0.04	-0.03	0.10	0.17
Togo [TGO]	-0.17	0.09	-0.14	-0.17	0.18	-0.19	-0.05	-0.13	-0.20	0.01	-0.05	-0.08	-0.03	0.14	0.01	0.20	-0.25	-0.14	-0.10	0.12
Tunisia [TUN]	-0.02	0.22	-0.13	0.19	0.13	0.08	0.01	-0.23	0.18	0.05	0.23	-0.06	-0.13	-0.10	0.07	-0.01	0.11	0.04	-0.27	0.05
Uganda [UGA]	-0.18	-0.07	0.01	-0.15	-0.16	0.07	0.09	-0.23	-0.10	-0.24	0.01	0.06	0.16	-0.04	0.00	-0.17	0.01	0.36	-0.08	-0.22
Zambia [ZMB]	-0.27	-0.10	0.12	0.10	-0.12	0.10	0.13	-0.12	-0.08	0.17	0.11	-0.03	0.00	-0.03	0.00	0.23	-0.07	0.03	-0.04	-0.13
Zimbabwe [ZWE]	-0.01	0.15	0.05	-0.10	-0.15	0.13	-0.21	-0.19	-0.11	0.13	-0.33	0.04	-0.26	0.07	0.07	0.00	-0.29	0.13	0.00	-0.03

Factor loadings are estimated with data from 1963 until 2015 to make the 2016 forecast.

Table 11: Coefficient estimates used to forecast real GDP growth in Nigeria with economic indicator data.

	Benchmark	Variable			Factor					
	AR(1)	LARS	Elastic Net	Boosting	FAAR	Ridge	LARS	Elastic Net	Bagging	Boosting
Constant	2.91				2.17				2.17	
AR(1)	0.12				0.24				0.24	
Monthly average off. exchange rate				-0.01						
Real effective exchange rate				0.02						
Deposit rate 3 months				0.60						
Interest rate spread				0.53						
Lending interest rate				-0.33						
Changes in inventories				0.01						
Gross fixed capital formation				0.02						
Government exp. on education				0.00						
Labor force, total				-0.20						
Money supply (M1)				-0.01						
Total monetary liabilities (M2)				-0.02						
All Shares Index				-0.01						
Government expenditure				-0.04						
Househ. and NPISHs final cons. exp.				0.03						
Real GDP - construction		0.02	0.50	0.19						
Real GDP - manufacturing				-0.02						
Real GDP - trade				-0.01						
Current account				-0.01						
Direct Investment				0.01						
Global price of Brent crude oil				0.06						
Non-oil exports				0.00						
Oil exports				0.01						
World GDP				-0.03						
Factor 1					0.91	0.84	0.00	0.02	0.84	0.71
Factor 2					0.29	0.36				0.22
Factor 3					0.76	0.46				0.29
Factor 4					0.19	0.19				
Factor 5					0.15	0.28				
Factor 6					0.07	0.13				
Factor 7					-0.94	-0.39				-0.33
Factor 8					-0.24	-0.08				
Factor 9					0.27	0.25				
Factor 10					0.25	0.24				
Factor 11					-0.68	-0.30				-0.32
Factor 12					0.34	0.02				
Factor 13					1.14	0.44				0.55
Factor 14					-2.27	-0.69			-1.98	-1.35
Factor 15					2.36	0.79			2.20	1.69
Factor 16					-0.16	0.02				
Factor 17					1.47	0.34				0.55
Factor 18					0.25	0.03				
Factor 19					-1.35	-0.36				-0.67
Factor 20					-1.62	-0.55				-1.64

Coefficients are estimated with data from 1982 until 2015 to forecast 2016. Variables and factors are normalised for ridge regression, LARS and elastic net. Variables not selected by any model are not shown. The random walk model is omitted, as no coefficients are estimated.

Table 12: Factor loadings of the first twenty factors generated using economic indicator data.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
Monthly average off. exchange rate	0.03	-0.09	0.29	0.15	-0.13	0.13	-0.06	-0.17	0.00	-0.23	0.42	0.08	0.06	0.12	-0.26	0.20	0.11	0.16	-0.22	-0.21
Real effective exchange rate	0.14	0.14	-0.12	-0.10	-0.26	-0.05	-0.16	-0.03	0.35	0.13	-0.01	0.21	0.07	0.26	0.30	-0.16	0.29	-0.21	0.12	-0.03
Comm. banks' loans and advances	0.17	0.19	-0.14	-0.14	-0.14	-0.09	0.02	-0.10	-0.04	-0.35	-0.06	0.37	0.00	0.02	0.11	-0.11	-0.31	0.20	-0.12	-0.04
Deposit rate 3 months	0.06	-0.17	0.43	-0.14	0.01	0.02	0.12	-0.04	-0.01	0.12	-0.17	-0.10	0.04	0.02	0.17	0.06	0.11	0.13	-0.05	-0.02
Deposit rate over 12 months	0.08	-0.14	0.45	-0.13	0.04	0.03	0.12	-0.08	0.06	0.05	-0.09	-0.05	0.01	0.04	0.19	-0.10	-0.08	0.01	0.03	0.12
Interest rate spread	0.08	0.09	-0.01	0.21	-0.02	0.17	-0.21	0.14	0.26	-0.06	-0.32	0.45	-0.11	-0.22	-0.32	-0.03	0.06	0.05	0.03	0.07
Lending interest rate	0.14	-0.05	0.41	-0.02	-0.05	0.07	0.11	0.19	0.18	0.02	-0.21	0.05	-0.06	-0.08	-0.11	-0.13	-0.01	0.16	0.02	0.34
Changes in inventories	0.12	0.16	0.02	0.01	-0.14	-0.36	0.19	0.29	0.20	0.12	0.06	-0.06	0.25	-0.10	0.24	-0.12	-0.20	0.06	0.08	-0.25
Gross fixed capital formation	0.08	0.18	0.03	0.08	0.13	0.40	-0.18	0.08	-0.20	0.17	-0.06	-0.02	0.11	0.35	0.03	-0.10	-0.21	0.36	0.15	-0.14
Total savings	0.09	0.32	0.09	-0.14	-0.21	0.01	0.09	-0.23	-0.02	-0.17	-0.09	-0.01	-0.07	-0.05	-0.12	-0.14	0.34	0.27	0.08	-0.10
Government exp. on education	0.05	0.09	0.15	0.44	0.09	-0.20	0.01	-0.06	-0.07	0.16	0.03	0.16	-0.12	0.06	0.10	-0.18	-0.01	-0.03	-0.28	0.02
Government exp. on health	0.03	0.13	0.13	0.42	0.06	-0.29	0.00	0.04	-0.10	0.12	0.03	0.05	-0.17	0.02	0.01	-0.18	-0.06	0.00	-0.23	0.04
Gross enrol. ratio, prim., both sexes	-0.06	0.10	0.04	-0.04	-0.20	0.14	-0.04	0.12	0.21	0.55	0.22	0.20	0.15	-0.12	-0.05	0.33	-0.14	0.15	0.01	0.10
Labor force, total	0.19	0.10	0.09	0.14	0.08	0.24	-0.22	0.27	-0.28	-0.06	0.12	-0.05	0.01	-0.23	0.15	-0.01	0.23	-0.26	0.24	0.21
Money supply (M1)	0.24	0.26	-0.03	0.02	-0.25	0.03	0.12	-0.03	-0.27	0.13	-0.11	-0.07	-0.13	0.02	-0.09	0.17	-0.04	-0.32	0.02	0.07
Total monetary liabilities (M2)	0.22	0.30	0.05	-0.09	-0.24	0.00	0.14	-0.13	-0.26	0.07	-0.05	-0.15	-0.02	0.12	-0.10	0.03	0.00	-0.05	-0.04	0.15
All Shares Index	0.29	-0.11	-0.20	0.00	0.01	-0.06	0.05	0.16	-0.02	-0.08	-0.02	0.05	0.05	0.40	0.12	0.40	0.02	0.14	-0.24	0.40
Inflation, consumer prices	0.09	0.06	0.10	0.14	-0.11	-0.39	-0.26	-0.04	0.04	0.08	-0.22	-0.20	-0.16	-0.05	-0.13	0.45	0.07	0.11	0.23	-0.24
Government expenditure	0.25	-0.01	0.25	0.04	-0.09	0.04	0.09	0.06	0.12	-0.26	0.31	0.09	-0.01	0.19	-0.11	-0.05	-0.30	-0.26	0.31	-0.09
Househ. and NPISHs final cons. exp.	0.07	0.11	-0.01	-0.09	0.18	-0.17	0.32	0.43	0.02	-0.27	0.02	0.04	0.10	-0.11	-0.06	0.18	0.30	0.11	-0.10	-0.03
Real GDP - agriculture	-0.04	0.06	0.12	-0.04	0.30	-0.09	0.08	-0.08	-0.26	0.16	-0.17	0.41	0.39	0.23	-0.14	0.13	0.14	-0.14	0.03	-0.22
Real GDP - construction	0.16	0.29	0.04	0.02	0.13	0.15	-0.27	0.08	-0.05	-0.11	-0.03	-0.24	0.14	-0.22	0.27	0.04	-0.09	0.23	-0.19	-0.20
Real GDP - crude petr. & nat. gas	0.16	-0.08	-0.12	0.10	0.02	0.23	0.40	0.27	-0.02	0.12	-0.10	-0.10	-0.09	-0.04	-0.24	-0.06	-0.13	-0.15	-0.08	-0.36
Real GDP - manufacturing	-0.01	0.16	0.01	-0.02	0.36	-0.10	0.16	0.02	-0.06	0.05	0.35	0.20	-0.37	0.02	0.16	0.08	0.11	0.19	0.45	0.02
Real GDP - services	0.01	0.31	0.07	-0.14	0.34	0.05	0.01	-0.08	0.19	-0.09	-0.08	-0.02	0.24	-0.15	-0.06	0.12	-0.11	-0.24	-0.09	0.09
Real GDP - trade	0.01	0.23	0.01	-0.03	0.31	-0.03	0.07	-0.31	0.33	0.01	-0.04	-0.17	-0.14	0.06	-0.18	0.14	-0.27	-0.07	0.05	0.11
Current account	0.26	-0.13	-0.22	-0.14	0.19	0.03	0.07	0.04	0.08	0.18	-0.18	-0.06	-0.29	0.18	-0.09	-0.17	0.07	0.19	0.03	-0.12
Direct Investment	0.24	-0.22	0.11	0.00	0.10	0.14	-0.12	-0.07	0.09	-0.11	-0.19	0.16	-0.25	-0.01	0.36	0.28	-0.03	-0.24	-0.11	-0.32
Global price of Brent crude oil	0.23	0.08	-0.09	0.02	0.05	0.21	0.23	-0.32	0.13	0.24	0.23	0.04	0.01	-0.25	0.10	-0.01	0.26	0.00	-0.22	-0.09
Imports	0.35	-0.16	-0.08	-0.07	0.06	-0.17	-0.26	0.02	-0.03	0.02	0.16	-0.06	0.08	-0.15	-0.16	-0.04	-0.14	0.06	0.02	0.06
India GDP	0.03	0.11	-0.01	0.32	0.13	0.05	-0.06	0.04	0.34	-0.10	-0.01	-0.31	0.21	0.36	-0.05	-0.11	0.25	-0.07	0.03	0.02
Non-oil exports	0.26	-0.16	0.00	-0.14	0.18	-0.25	-0.15	-0.19	-0.17	0.11	-0.05	0.07	0.26	-0.11	-0.15	-0.20	0.03	-0.05	0.11	0.03
Oil exports	0.38	-0.22	-0.07	-0.01	0.05	-0.04	-0.08	-0.04	0.09	0.07	0.19	-0.08	0.07	-0.05	-0.16	-0.08	0.09	0.07	-0.05	0.00
United States GDP	0.04	-0.19	-0.09	0.34	-0.18	0.07	0.21	-0.16	-0.01	-0.09	-0.17	0.00	0.30	-0.08	0.04	0.09	0.01	0.13	0.35	-0.06
World GDP	0.11	-0.08	-0.17	0.35	0.07	0.05	0.25	-0.27	0.00	-0.09	-0.09	0.02	0.12	-0.19	0.21	0.09	-0.12	0.13	0.15	0.22

Factor loadings are estimated with data from 1982 until 2015 to make the 2016 forecast.

Table 13: Coefficient estimates used to forecast real GDP growth in Nigeria with African GDP growth and economic indicator data.

	Benchmark	Variable			Factor					
	AR(1)	LARS	Elastic Net	Boosting	FAAR	Ridge	LARS	Elastic Net	Bagging	Boosting
Constant	2.91			0.00	3.63				3.63	
AR(1)	0.12			0.00	-0.05				-0.05	
Angola [AGO]				0.02						
Burkina Faso [BFA]				-0.04						
Cameroon [CMR]				-0.03						
Central African Republic [CAF]				-0.01						
Comoros [COM]				-0.04						
Congo, Rep. [COG]				0.07						
Eritrea [ERI]				-0.08						
Ethiopia [ETH]				-0.03						
Liberia [LBR]				0.01						
Libya [LBY]				0.03						
Madagascar [MDG]				-0.10						
Mauritius [MUS]				-0.29						
Mozambique [MOZ]				0.02						
Namibia [NAM]				-0.08						
Niger [NER]				-0.38						
Sao Tome and Principe [STP]				0.07						
Seychelles [SYC]				0.01						
Sierra Leone [SLE]				0.10						
South Africa [ZAF]				0.43						
Togo [TGO]				0.12						
Tunisia [TUN]				0.05						
Comm. banks' loans and advances				0.00						
Deposit rate 3 months				0.16						
Interest rate spread				0.04						
Changes in inventories				0.01						
Government exp. on education				0.00						
All Shares Index				-0.02						
Inflation, consumer prices				0.03						
Real GDP - construction		0.00	0.04	0.22						
Real GDP - trade				-0.02						
Direct Investment				0.01						
Global price of Brent crude oil				0.01						
Factor 1					0.57	0.30				0.19
Factor 2					0.68	0.46	0.01	0.01	0.53	0.51
Factor 3					0.36	0.22				
Factor 4					-0.07	-0.10				
Factor 5					0.04	0.08				
Factor 6					0.47	0.15				
Factor 7					0.65	0.33				0.33
Factor 8					-0.05	-0.12				
Factor 9					0.14	0.04				
Factor 10					0.37	0.11				

Coefficients are estimated with data from 1982 until 2015 to forecast 2016. Variables and factors are normalised for ridge regression, LARS and elastic net. Variables not selected by any model are not shown. The random walk model is omitted, as no coefficients are estimated.

Table 14: Factor loadings of the first twenty factors generated using African GDP growth and economic indicator data.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
Algeria [DZA]	0.07	-0.14	0.14	0.19	0.10	0.04	0.01	0.01	0.17	0.06	-0.08	-0.04	-0.02	0.25	0.02	0.04	0.08	-0.09	-0.11	-0.10
Angola [AGO]	0.20	-0.01	-0.04	0.21	0.09	0.05	0.03	0.01	-0.06	-0.06	0.00	-0.02	0.08	-0.13	-0.04	0.06	-0.07	-0.11	-0.04	0.02
Benin [BEN]	0.05	0.06	-0.05	0.03	-0.11	0.01	-0.26	0.10	0.20	0.17	0.04	-0.03	-0.04	-0.01	-0.19	0.11	-0.08	0.14	0.07	0.16
Botswana [BWA]	-0.01	0.02	-0.09	-0.04	0.22	0.14	0.08	-0.04	0.09	-0.22	0.13	-0.01	0.13	-0.03	0.13	0.09	-0.06	0.03	0.11	0.01
Burkina Faso [BFA]	0.16	-0.04	-0.04	0.00	-0.11	-0.14	-0.06	-0.08	0.15	-0.17	-0.12	-0.12	0.17	-0.01	0.05	0.05	0.12	0.16	0.03	0.11
Burundi [BDI]	0.10	-0.14	0.06	-0.12	0.02	0.19	-0.16	0.14	-0.07	-0.13	0.06	0.15	0.04	-0.10	0.07	-0.04	-0.14	0.01	0.00	0.02
Cabo Verde [CPV]	-0.06	0.14	-0.11	0.24	0.00	-0.01	-0.04	-0.06	-0.04	-0.19	0.02	0.02	-0.04	0.15	0.02	0.07	-0.11	0.01	-0.03	0.06
Cameroon [CMR]	0.10	-0.22	0.05	0.24	0.06	-0.09	-0.05	-0.07	0.04	0.02	0.05	-0.01	-0.04	0.01	-0.09	0.04	-0.04	0.01	-0.01	0.03
Central African Republic [CAF]	-0.01	-0.01	-0.05	0.08	-0.02	0.01	-0.05	-0.07	0.14	-0.13	-0.01	-0.12	-0.04	-0.08	-0.12	-0.07	0.13	-0.08	-0.28	-0.06
Chad [TCD]	0.10	-0.15	-0.05	0.13	0.00	0.10	0.02	0.10	0.03	-0.11	-0.18	0.22	-0.08	0.08	0.24	-0.11	-0.03	0.07	-0.01	-0.04
Comoros [COM]	-0.07	0.03	0.15	0.11	0.02	0.19	-0.04	-0.01	0.16	0.17	0.05	0.12	0.13	0.02	-0.15	-0.05	-0.09	0.10	0.29	-0.06
Congo, Dem. Rep. [COD]	0.23	-0.15	0.05	0.03	0.12	0.03	0.11	0.01	-0.02	0.03	0.05	-0.03	-0.10	-0.07	-0.11	0.06	-0.08	-0.06	-0.01	-0.07
Congo, Rep. [COG]	0.00	-0.11	0.17	0.07	-0.05	0.04	0.19	0.05	0.26	0.03	0.00	-0.09	0.21	0.01	0.00	-0.06	0.00	0.00	0.01	-0.15
Djibouti [DJI]	0.15	-0.19	0.09	0.03	0.04	0.19	0.00	-0.01	-0.07	0.06	0.04	0.02	-0.06	-0.05	-0.09	-0.08	-0.20	-0.16	-0.09	-0.07
Egypt, Arab Rep. [EGY]	-0.05	-0.03	0.03	0.21	0.01	0.19	0.03	-0.09	0.15	-0.06	-0.02	-0.17	0.20	-0.10	0.08	-0.03	-0.18	-0.24	-0.05	0.13
Equatorial Guinea [GNQ]	0.00	0.09	-0.09	0.22	-0.12	-0.11	-0.14	-0.14	0.07	0.03	0.07	0.19	0.08	-0.03	0.13	-0.01	0.17	0.00	-0.02	-0.05
Eritrea [ERI]	-0.18	-0.12	-0.17	-0.03	0.02	-0.12	0.08	-0.04	0.04	0.02	0.09	-0.04	-0.11	-0.08	0.17	0.06	-0.11	0.15	0.00	-0.08
Ethiopia [ETH]	0.14	0.08	0.08	0.02	-0.03	-0.04	0.22	-0.14	-0.12	0.13	0.15	-0.10	-0.12	-0.17	0.02	0.12	-0.06	0.06	0.09	0.01
Gabon [GAB]	0.02	-0.04	-0.21	-0.08	-0.01	0.06	0.16	0.05	0.14	0.15	-0.08	0.15	0.05	0.08	-0.03	0.28	0.07	-0.01	-0.21	-0.01
Gambia, The [GMB]	-0.03	0.00	0.05	0.08	0.02	0.13	0.05	-0.12	-0.21	0.14	-0.23	0.14	0.06	-0.08	0.10	0.13	0.08	-0.11	0.07	0.04
Ghana [GHA]	0.16	0.07	-0.08	-0.13	0.02	0.04	-0.09	0.08	-0.06	0.10	0.10	0.01	-0.11	0.00	-0.19	-0.01	0.09	0.14	-0.19	-0.07
Guinea [GIN]	0.08	0.14	-0.17	-0.11	0.02	0.00	-0.12	-0.05	0.13	-0.10	0.14	0.06	-0.01	0.05	0.00	0.17	-0.21	0.06	-0.05	-0.13
Guinea-Bissau [GNB]	0.01	0.08	-0.10	0.03	0.04	0.00	0.13	0.11	0.12	0.14	0.16	-0.09	-0.01	-0.25	0.14	-0.20	0.00	-0.01	-0.09	0.09
Ivory Coast [CIV]	0.14	-0.03	-0.04	-0.05	0.03	-0.21	-0.07	-0.16	-0.02	0.11	-0.13	0.00	0.15	0.03	-0.03	0.12	-0.24	-0.07	0.06	-0.07
Kenya [KEN]	0.21	-0.03	-0.09	-0.11	0.18	0.02	0.08	-0.03	-0.07	0.08	-0.02	-0.08	-0.02	-0.12	0.05	-0.02	-0.07	0.07	0.08	-0.02
Lesotho [LSO]	-0.02	0.05	-0.21	-0.08	-0.04	0.12	0.10	0.20	0.05	-0.02	-0.06	-0.08	0.19	-0.13	-0.09	0.07	0.07	-0.02	-0.14	-0.14
Liberia [LBR]	0.09	-0.03	-0.04	0.17	-0.06	-0.06	-0.08	-0.25	0.05	0.00	0.21	0.22	0.19	-0.04	-0.08	-0.01	0.08	0.01	0.01	-0.01
Libya [LBY]	0.03	0.00	-0.01	0.01	-0.07	0.02	-0.05	0.03	-0.03	0.00	-0.13	0.01	0.15	0.14	0.16	0.24	-0.12	-0.06	0.06	-0.21
Madagascar [MDG]	0.09	0.10	-0.05	0.17	0.05	0.16	-0.06	-0.08	-0.12	0.11	-0.22	-0.16	0.05	-0.07	-0.08	0.05	-0.05	0.04	0.01	-0.13
Malawi [MWI]	0.10	0.08	0.10	-0.03	0.09	-0.06	0.07	0.01	0.18	0.07	-0.05	-0.01	0.11	-0.07	-0.07	0.19	0.02	0.04	-0.14	0.40
Mali [MLI]	0.11	-0.04	-0.13	-0.05	-0.07	-0.05	-0.19	-0.03	0.02	-0.07	-0.22	0.08	-0.12	-0.03	0.13	-0.07	0.06	-0.03	-0.13	0.16
Mauritania [MRT]	0.16	0.09	0.07	-0.03	0.10	0.01	0.07	-0.07	-0.07	-0.07	-0.13	-0.12	0.04	0.16	0.05	0.09	-0.06	0.24	0.01	0.13
Mauritius [MUS]	-0.02	0.09	-0.05	-0.01	0.10	0.05	-0.18	0.10	-0.09	-0.07	0.09	-0.13	0.24	-0.21	-0.08	-0.05	0.08	0.02	0.29	-0.02
Morocco [MAR]	0.06	-0.17	-0.18	-0.04	-0.10	0.06	0.01	0.06	0.00	-0.11	-0.04	-0.26	-0.01	-0.10	0.07	0.10	0.21	-0.06	0.15	-0.02
Mozambique [MOZ]	0.17	0.14	-0.08	-0.05	-0.10	-0.15	-0.08	-0.10	-0.02	0.03	0.04	-0.10	-0.06	0.03	0.15	0.00	0.18	-0.03	0.04	-0.09
Namibia [NAM]	0.17	0.03	0.03	0.03	0.02	-0.04	-0.03	0.12	-0.18	-0.04	0.03	0.14	0.07	0.05	-0.06	-0.05	0.11	0.27	-0.05	-0.01
Niger [NER]	0.20	-0.01	-0.06	-0.11	-0.12	-0.12	-0.05	-0.05	-0.02	-0.14	-0.13	-0.02	0.10	0.06	-0.03	0.06	-0.13	-0.06	0.16	-0.09
Rwanda [RWA]	0.13	0.06	0.18	0.03	0.12	-0.02	-0.04	-0.09	0.22	0.04	-0.08	0.20	0.16	-0.05	-0.10	0.08	0.10	0.00	-0.01	0.09
Sao Tome and Principe [STP]	0.21	0.01	0.00	0.01	-0.14	0.00	0.00	0.04	0.08	-0.08	-0.13	-0.08	0.07	0.16	0.05	-0.20	-0.21	0.01	-0.12	-0.01
Senegal [SEN]	0.13	0.00	0.11	0.09	0.06	-0.13	-0.11	-0.08	0.04	0.01	-0.12	-0.18	0.05	-0.04	-0.09	-0.19	0.07	0.14	-0.04	-0.26
Seychelles [SYC]	0.04	0.07	-0.11	-0.11	0.07	0.07	-0.16	-0.05	0.02	0.08	0.18	0.05	0.27	0.08	0.13	0.02	-0.17	0.07	-0.25	-0.09
Sierra Leone [SLE]	0.11	-0.04	0.13	-0.04	-0.03	-0.02	0.03	0.15	0.01	-0.01	0.28	-0.06	-0.06	0.16	0.10	0.22	0.04	0.00	0.16	-0.01
Somalia [SOM]	-0.04	0.04	-0.10	0.10	-0.04	-0.15	-0.08	-0.22	0.07	0.09	0.06	-0.14	-0.19	0.05	0.01	-0.11	-0.15	-0.20	0.23	-0.09
South Africa [ZAF]	0.17	0.12	-0.09	0.14	0.03	0.09	0.08	-0.01	0.06	-0.01	0.18	-0.10	-0.15	0.06	-0.03	0.11	0.12	0.02	-0.02	-0.17
Sudan [SDN]	0.06	0.15	0.13	0.09	-0.03	-0.14	0.07	-0.08	-0.15	-0.17	0.02	0.07	0.14	-0.08	0.11	0.00	0.17	-0.06	0.01	-0.21
Tanzania [TZA]	0.24	-0.01	0.05	-0.03	0.05	0.06	-0.03	0.05	-0.05	0.14	0.09	0.03	-0.18	-0.05	-0.04	-0.02	0.06	-0.06	0.03	0.04
Togo [TGO]	0.12	-0.04	-0.25	0.11	0.08	-0.11	0.01	-0.01	-0.01	0.04	0.08	0.03	0.01	0.01	-0.09	-0.15	-0.02	-0.21	-0.02	0.01
Tunisia [TUN]	-0.02	0.11	0.03	0.19	0.01	0.07	-0.21	0.08	-0.14	0.12	-0.05	0.03	0.02	0.04	0.20	0.06	-0.09	-0.03	-0.04	0.13
Uganda [UGA]	0.07	0.17	0.05	-0.01	-0.11	-0.03	0.28	-0.07	0.11	-0.12	0.00	-0.12	-0.01	0.10	-0.02	0.06	0.08	-0.06	-0.08	0.11
Zambia [ZMB]	0.23	0.10	0.10	-0.04	-0.09	0.08	0.03	-0.07	0.07	0.12	-0.01	0.05	-0.01	-0.08	0.05	0.05	0.08	0.06	0.02	0.07
Zimbabwe [ZWE]	-0.01	-0.09	-0.18	-0.19	0.03	-0.10	0.08	-0.07	0.04	0.17	0.08	-0.02	0.10	-0.04	0.10	-0.03	-0.13	0.09	0.02	0.03
Monthly average off. exchange rate	-0.06	0.05	0.06	-0.06	0.09	0.05	-0.13	-0.27	-0.14	-0.08	0.01	-0.11	-0.07	0.01	0.02	-0.22	0.06	0.09	0.03	0.14
Real effective exchange rate	-0.09	0.14	-0.01	0.10	-0.12	-0.06	0.29	-0.02	0.04	0.07	0.07	-0.03	-0.14	0.04	0.14	0.01	-0.20	0.12	-0.03	-0.01
Comm. banks' loans and advances	-0.01	0.18	-0.04	0.19	-0.15	-0.07	-0.03	-0.02	0.02	-0.02	0.10	0.15	0.06	-0.20	0.01	0.06	-0.07	-0.07	-0.02	-0.03
Deposit rate 3 months	-0.06	0.02	0.26	-0.14	0.10	0.07	-0.15	-0.07	0.04	-0.04	0.04	-0.09	-0.06	0.03	0.04	0.12	0.00	-0.09	-0.06	-0.20
Deposit rate over 12 months	-0.03	0.05	0.26	-0.17	0.10	0.06	-0.13	-0.11	0.08	-0.03	0.05	-0.12	-0.05	-0.03	0.09	0.16	0.01	-0.02	-0.11	-0.05
Interest rate spread	-0.04	0.08	-0.07	-0.06	0.02	0.07	0.17	-0.11	-0.01	0.16	0.11	0.25	0.17	-0.02	0.20	-0.20	0.13	-0.04	0.04	0.02
Lending interest rate	-0.08	0.10	0.20	-0.17	0.07	0.14	-0.09	-0.15	0.10	0.16	-0.01	0.01	0.04	-0.09	0.12	0.03	0.00	-0.04	0.07	-0.08
Changes in inventories	-0.02	0.10	-0.01	0.07	-0.13	0.06	0.03	-0.07	0.27	0.09	-0.14	0.00	-0.21	-0.07	-0.02	0.03	0.02	0.11	0.11	-0.25
Gross fixed capital formation	0.08	0.11	0.00	-0.13	-0.09	-0.01	0.03	0.00	-0.13	0.05	0.02	-0.07	0.20	0.30	-0.05	-0.08	0.10	-0.16	0.19	0.12
Total savings	-0.02	0.14	0.07	0.07	-0.24	0.13	-0.04	-0.02	-0.11	-0.06	0.09	-0.01	-0.03	-0.12	0.09	0.05	-0.14	0.00	-0.09	0.03
Government exp. on education	0.06	0.04	-0.14	-0.11	0.06	0.24	0.06	-0.18	0.19	-0.17	0.04	0.03	-0.03	0.03	-0.01	-0.07	0.08	-0.07	0.14	0.02
Government exp. on health	0.05	0.03	-0.16	-0.13	0.01	0.24	0.06	-0.15	0.20	-0.16	-0.06	0.08	-0.04	-0.05	-0.05	-0.04	0.09	-0.03	0.14	0.02
Gross enrol. ratio, prim., both sexes	-0.10	-0.02	0.09	0.03	-0.13	0.06	0.06	-0.02	0.09	0.09	0.08	0.03	0.11	0.33	-0.05	-0.22	0.00	0.25	0.02	-0.09
Labor force, total	0.03	0.15	-0.09	-0.23	0.00	-0.02	-0.02	-0.03	-0.04	0.15	-0.01	0.03	0.05	0.14	-0.17	-0.14	-0.17	-0.26	-0.09	-0.03
Money supply (M1)	-0.08	0.23	-0.07	0.02	-0.17	0.18	-0.02	0.11	-0.03	-0.07	-0.06	-0.04	0.00	0.03	-0.12	-0.01	-0.02	-0.08	-0.04	-0.03
Total monetary liabilities (M2)	-0.06	0.23	0.03	0.03	-0.21	0.17	-0.06	0.09	-0.05	-0.10	-0.02	-0.08	-0.02	-0.02	-0.12	0.07	-0.10	-0.05	0.01	0.07
All Shares Index	-0.02	0.17	-0.07	0.03	0.14	-0.10	0.03	0.20	0.05	0.02	-0.14	0.03	-0.01	0.08	0.07	0.18	-0.04	0.10	0.16	0.00
Inflation, consumer prices	-0.04	0.04	0.02	-0.08	0.00	0.14	0.13	-0.06	-0.03	-0.20	-0.22	0.27	-0.13	-0.07	-0.14	-0.04	-0.09	0.14	0.03	-0.21
Government expenditure	-0.03	0.22	0.03	-0.10	0.03	-0.01	-0.13	-0.18	0.08	0.02	-0.12	-0.13	-0.08	-0.05	0.14	-0.06	0.01	0.13	-0.11	0.11
Househ. and NPISHs final cons. exp.	0.06	0.03	-0.02	-0.03	-0.08	-0.02	-0.17	0.13	0.16	0.20	-0.13	0.14	-0.17	-0.13	0.19	0.01	0.03	-0.05	0.04	-0.08
Real GDP - agriculture	0.08	-0.04	0.06	-0.09	-0.07	-0.02	-0.11	0.06	0.16	-0.19	0.16	0.15	-0.15	0.23	0.10	0.01	0.08	-0.21	0.07	0.07
Real GDP - construction	0.15	0.17	0.01	-0.13	-0.13	-0.01	0.13	-0.04	-0.17	0.08	0.02	0.09	-0.01	0.03	-0.15	0.02	0.08	-0.06	0.01	-0.14
Real GDP - crude petr. & nat. gas	-0.03	0.08	-0.08	0.04	0.10	0.11	-0.12	0.20	0.07	0.25	-0.12	-0.09	-0.04	0.09	0.04	-0.09	0.25	-0.17	0.01	0.00
Real GDP - manufacturing	0.18	-0.02	-0.04	-0.11	-0.08	0.00	-0.13	0.12	0.16	-0.04	0.03	-0.06	0.04	-0.08	0.12	-0.24	-0.17	0.12	0.07	-0.05
Real GDP - services	0.19	0.04	0.13	-0.02	-0.18	0.05	0.04	0.04	-0.03	0.04	0.06	0.09	-0.06	-0.02	0.12	-0.04	-0.04	-0.11	-0.04	0.02
Real GDP - trade	0.20	0.03	0.10	0.04	-0.07	0.12	0.07	0.07	-0.09	-0.09	-0.01	0.02	-0.03	-0.09	0.16	-0.11	0.02	0.13	-0.01	0.04
Current account	0.03	0.13	0.02	0.06	0.14	-0.12	0.06	0.28	0.04	0.01	-0.09	-0.01	0.08	-0.02	0.18	-0.03	0.01	-0.08	0.20	-0.09
Direct Investment	-0.01	0.13	0.05	-0.08	0.24	-0.07	0.04	0.00	-0.04	-0.08	0.00	-0.04	0.06	0.03	0.21	-0.04	0.05	-0.15	-0.26	-0.23
Global price of Brent crude oil	0.08	0.17	0.05	0.14	0.06	0.14	0.03	0.14	0.02	0.01	0.17	-0.14	0.00	0.14	0.08	-0.15	-0.02	0.07	-0.05	0.03
Imports	0.00	0.20	0.05	0.00	0.19	-0.19	0.07	0.08	0.09	-0.08	-0.10	0.14	-0.02	-0.07	-0.16	-0.12	-0.07	0.00	-0.02	0.02
India GDP	0.13	0.02	-0.07	0.02	0.07	0.13	0.16	-0.19	-0.04	0.10	-0.20	-0.02	-0.10	0.15	0.09	-0.06	-0.10	-0.06	0.08	0.15
Non-oil exports	0.03	0.13	0.10	-0.01	0.16	-0.16	-0.01	0.10	0.10	-0.21	0.09	0.17	-0.15	-0.05	-0.08	0.01	-0.11	-0.15	0.09	0.07
Oil exports	-0.01	0.21	0.06	0.05	0.27	-0.11	0.04	0.09	0.07	-0.01	-0.08	0.01	-0.07	0.00	-0.05	-0.15	-0.07	-0.01	0.04	0.04
United States GDP	-0.05	0.03	-0.16	0.13	0.22	0.13	-0.13	-0.12	-0.09	0.03	0.06	0.06	-0.11	0.19	0.00	0.09	0.05	0.10	0.00	-0.02
World GDP	0.10	0.08	-0.18	0.12	0.23	0.17	-0.07	0.00	-0.04	-0.02	0.12	0.01	-0.02	0.07	0.01	0.05	0.03	0.11	0.03	-0.05

Factor loadings are estimated with data from 1982 until 2015 to make the 2016 forecast.