Economics
The Open-Access, Open-Assessment E-Journal

Discussion Paper
No. 2020-5 | February 04, 2020 | http://www.economics-ejournal.org/economics/discussionpapers/2020-5

Please cite the corresponding Journal Article at
http://www.economics-ejournal.org/economics/journalarticles/2020-14

A replication of "A quasi-maximum likelihood approach for large, approximate dynamic factor models" (Review of Economics and Statistics, 2012)

Riccardo Lucchetti and Ioannis A. Venetis

Abstract
The authors replicate and extend the Monte Carlo experiment presented in Doz et al. (2012) on alternative (time-domain based) methods for extracting dynamic factors from large datasets; they employ open source software and consider a larger number of replications and a wider set of scenarios. Their narrow sense replication exercise fully confirms the results in the original article. As for their extended replication experiment, the authors examine the relative performance of competing estimators under a wider array of cases, including richer dynamics, and find that maximum likelihood (ML) is often the dominant method; moreover, the persistence characteristics of the observable series play a crucial role and correct specification of the underlying dynamics is of paramount importance.

(Replication Study)

JEL C15 C32 C55 C87
Keywords Dynamic factor models; EM algorithm; Kalman filter; principal components

Authors
Riccardo Lucchetti, DISES, Universit`a Politecnica delle Marche, Ancona, Italy
Ioannis A. Venetis, ✉ University of Patras, Greece, ivenetis@upatras.gr

Citation Riccardo Lucchetti and Ioannis A. Venetis (2020). A replication of "A quasi-maximum likelihood approach for large, approximate dynamic factor models" (Review of Economics and Statistics, 2012). Economics Discussion Papers, No 2020-5, Kiel Institute for the World Economy. http://www.economics-ejournal.org/economics/discussionpapers/2020-5

Received November 12, 2019 **Accepted as Economics Discussion Paper** January 27, 2020
Published February 4, 2020

1 Introduction

A central empirical finding from the use of dynamic factor models in applied macroeconomics is that a few factors can explain a large fraction of the total variance of many macroeconomic series. The factor structure is typically used for signal extraction and forecasting, although in some cases maximum likelihood-based methods can be used to test hypotheses of economic interest. Extensive surveys of the dynamic factor model literature can be found in Bai and Ng (2008), Stock and Watson (2011) and Stock and Watson (2016).

Maximum likelihood estimation for dynamic factor models with a high-dimensional panel of time series becomes rapidly infeasible given the large number of parameters. However, in Doz et al. (2011, 2012) a solution was put forward to overcome this problem via either (a) a two-step hybrid approach that links the simplicity and speed of principal components to the efficiency of the Kalman smoother or (b) quasi-maximum likelihood where factor estimates are computed iteratively using the Kalman smoother on the state-space representation via the expectation–maximization (EM) algorithm.

Both approaches are robust to cross-sectional misspecification, time-series correlation of the idiosyncratic components, and non-Gaussianity. Since then, state-space based maximum likelihood methods have been increasingly adopted in the applied literature, e.g. Luciani (2015) and Scotti (2016), among others.

In this paper, we replicate and extend the Monte Carlo experiment presented in Doz et al. (2012) on the suitability of ML for extracting dynamic factors from large datasets, by using different software, a larger number of replications and a wider set of scenarios. We find that most of the statements in the original article are supported, but the relative quality of full ML estimation, compared to the alternative procedures proposed in the same paper, could be substantial in some cases, whereas in other settings it may not be so decisive to outweigh the extra computational cost.

For our replication exercise, we used the `DFM` package for gretl (see Lucchetti and Venetis (2019)), which implements three estimators for dynamic factor models, namely the benchmark principal components estimator (PC), the two-step Kalman smoother based estimator (TS) put forward by Doz et al. (2011) and the quasi-maximum likelihood EM based estimator (ML) by Doz et al. (2012).

The paper is organised as follows: section 2 describes the data generating process, section 3 offers a narrow sense replication of the Monte Carlo study in Doz et al. (2012) and section 4 extends (wide sense replication) the experiment, in a direction which should address common concerns of practitioners. Section 5 briefly concludes the paper by summarising our findings.

2 Setup of the experiment

The models that we consider can be written in state-space representation as

$$\underset{N\times 1}{x_t} = \Lambda_0 f_t + \Lambda_1 f_{t-1} + \cdots + \Lambda_s f_{t-s} + e_t \tag{1}$$

$$\underset{q\times 1}{f_t} = A_1 f_{t-1} + A_2 f_{t-2} + \cdots + A_p f_{t-p} + u_t \tag{2}$$

where x_t is a vector of N standardised observable variables and f_t is the q-element vector of (unobserved) common dynamic factor; the shocks to the ob-

servation equation (1), e_t, are known as the idiosyncratic component, and are assumed to be uncorrelated with f_t at all leads and lags. It is assumed that the elements of e_t are weakly correlated both cross-sectionally and serially, so that the factors f_t summarise the most relevant cross-covariance properties of the variables. Both processes f_t and e_t are assumed to be second-order stationary.

It should be noted that s, the lag order in the observation equation (1), is assumed to be finite here, in accordance with the study we are replicating. In fact, a sizeable body of literature (mainly by Forni, Hallin, Lippi and several co-authors) has analysed the so-called *generalised* dynamic factor model, in which this assumption is relaxed, and estimation is typically carried out via spectral methods. See Forni et al. (2000, 2005, 2015, 2017) and for recent applications see Barigozzi and Hallin (2015, 2017) and Barigozzi et al. (2018).

In accordance with Doz et al. (2012), we consider the following data generating process in our Monte Carlo study: the idiosyncratic shocks were generated as

$$e_t = De_{t-1} + v_t,\ v_t \sim N.i.d\left(\mathbf{0}_N, \mathcal{T}\right) \tag{3}$$

where D is a diagonal matrix. We call d_i its i-th diagonal entry. As for the matrix $\mathcal{T}$,

$$\mathcal{T}_{i,j} = \tau^{|i-j|}\left(1 - d_i d_j\right)\sqrt{\gamma_i \gamma_j} \tag{4}$$

$$\gamma_i = \frac{\beta_i}{1-\beta_i} Var\left(\chi_{i,t}\right) \tag{5}$$

where $\chi_{i,t} = \lambda'_{0,i} f_t + ... + \lambda'_{s,i} f_{t-s}$ denotes the i^{th} common component. The Toeplitz matrix $\mathcal{T}$ is the covariance matrix of the vector e_t, so the parameter τ controls for the amount of cross-correlation in the idiosyncratic errors, $Corr\left(e_{i,t}, e_{j,t}\right) = \tau^{|i-j|}$. The parameter $\beta_i = \dfrac{Var(e_{i,t})}{Var(x_{i,t})}$ denotes noise variance proportion for the $i-$th variable and is drawn from the uniform distribution $\beta_i \sim i.i.d.\ U\left([u, 1-u]\right)$. When $u = 0.5$, "observables" $x_{i,t}$ have cross-sectionally homoskedastic idiosyncratic components. The exact factor model corresponds to the case $\tau = 0$ and $D = 0$. The elements of the loadings matrices Λ_i were drawn from a standard Normal distribution. The above data generating process is considerably broader than the one adopted by Doz et al. (2012), which can be derived as a special case of equations (1)-(5). Similar experiments have also been performed by Breitung and Tenhofen (2011) and Bai and Li (2016).[1]

The aim of the experiment is to carry out a comparison between estimators on their ability to reconstruct the latent factors. As is well known, the individual factors cannot be estimated, so the comparison will be based on a statistic that measures the closeness of the space spanned by the "true" simulated factors f_t and their estimators $\hat{f}_t$. If we denote by F and $\hat{F}$ the matrices holding f_t and $\hat{f}_t$, respectively, the statistic we use is the trace R^2, that can be written as,

$$TR = \frac{tr\left[F'\hat{F}\left(\hat{F}'\hat{F}\right)^{-1}\hat{F}'F\right]}{tr\left(F'F\right)}$$

[1] In Forni et al. (2018) a comparison between alternative modelling techniques is also performed, but with a focus on forecasting.

and satisfies $0 \leq TR \leq 1$; it tends to 1 as the canonical correlation between estimated and true factors increases. Values of TR close to 0 indicate high discrepancies between the space spanned by the actual and the estimated factors.[2]

The three estimators considered are principal components (PC), the two-step estimator (TS) and the maximum likelihood estimator (ML).

3 Replication of the original results

The first exercise we perform is a narrow sense replication exercise, in which the relative performance of three different methods for factor extraction is investigated.

The original experiment assumed $s = 0$ and $p = 1$ as well as $A_1 = \alpha \cdot \mathbf{I}_q$, $D = d \cdot \mathbf{I}_N$ with α, d being scalar-valued. The parameter u is set to 0.1 in all experiments, so that the idiosyncratic components e_t can have substantially different variances and the observable series x_t exhibit different signal-to-noise ratios.

The three tables 2, 4, 6 present a reproduction of all Monte Carlo results from the original study of Doz et al. (2012). The original study's results are shown in tables 1, 3, 5 and are placed side-by-side in all tables with our replicating study's corresponding estimates. The data generation process (DGP) is exactly the same as that of the original paper, the only difference is that we ran the Monte Carlo experiment with 5000 replications, instead of 500.

The tables contain several subtables, each of which has different values of T (the sample size) for each row and of N (the number of observable variables) for each column. The first sub-table contains the average of the TR statistic for the ML estimator and the second one the average number of EM iterations needed to achieve convergence, as a rough proxy for computational speed.

The three sub-tables below display the relative performance of the PC and TS estimators compared to the ML estimator, measured by the trace ratios TR_{ML}/TR_{PC}, TR_{ML}/TR_{TS}.[3]

As can be seen, results are qualitatively identical and quantitatively very close to the original article: the maximum likelihood estimator appears to have a slight edge on the other two methods, especially so when the idiosyncratic disturbances e_t are closer to the "ideal" factor models (that is, both their cross-sectional correlation and autocorrelation are 0). It is noteworthy that the relative advantage of ML tends to vanish as either the time dimension T or the cross-sectional dimension N increase. Moreover, the extra computational cost for the extraction of factors via ML may be substantial (especially for small samples).

4 Extended replication

In this section, we extend the original Monte Carlo experiment so as to shed more light on the relative properties of the three factor extraction methods

[2] We are using this measure of the ability to reconstruct the factor space in accordance with the original article, but we would like to stress that this is a purely descriptive statistic and we are not in the position of judging if difference between methods are statistically significant.

[3] We dropped results with $TR < 0.05$ to avoid spuriously large trace ratios and we ensure that initial conditions satisfy stationarity restrictions.

Table 1: Doz et al. (2012, p. 1019) Table 1

	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}		
$T = 50$	0.52	0.68	0.74	0.75	0.76
$T = 100$	0.64	0.78	0.84	0.85	0.86
			Number of iterations		
$T = 50$	13	9	5	5	3
$T = 100$	13	7	4	4	3
			TR_{ML}/TR_{PC}		
$T = 50$	1.11	1.04	1.00	1.00	1.00
$T = 100$	1.09	1.02	1.01	1.00	1.00
			TR_{ML}/TR_{TS}		
$T = 50$	1.03	1.01	1.00	1.00	1.00
$T = 100$	1.02	1.00	1.00	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1019, Table 1).

Table 2: Replication of original Table 1 in Doz et al. (2012)

	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}		
$T = 50$	0.60	0.67	0.71	0.74	0.74
$T = 100$	0.72	0.78	0.82	0.84	0.84
			Number of iterations		
$T = 50$	10	6	4	3	3
$T = 100$	7	4	3	3	3
			TR_{ML}/TR_{PC}		
$T = 50$	1.03	1.01	1.00	1.00	1.00
$T = 100$	1.05	1.02	1.01	1.00	1.00
			TR_{ML}/TR_{TS}		
$T = 50$	0.99	0.99	1.00	1.00	1.00
$T = 100$	1.00	1.00	1.00	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 1$.

Table 3: Doz et al. (2012, p. 1019) Table 2

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}	
$T = 50$	0.48	0.59	0.65	0.67
$T = 100$	0.58	0.75	0.80	0.82
			Number of iterations	
$T = 50$	26	12	7	5
$T = 100$	20	9	5	4
			TR_{ML}/TR_{PC}	
$T = 50$	1.08	1.05	1.03	1.01
$T = 100$	1.10	1.06	1.02	1.01
			TR_{ML}/TR_{TS}	
$T = 50$	1.05	1.02	1.01	1.00
$T = 100$	1.07	1.03	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1019, Table 2).

Table 4: Replication of original Table 2 in Doz et al. (2012)

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}	
$T = 50$	0.53	0.63	0.66	0.69
$T = 100$	0.64	0.76	0.80	0.82
			Number of iterations	
$T = 50$	22	9	5	4
$T = 100$	15	6	4	3
			TR_{ML}/TR_{PC}	
$T = 50$	1.08	1.03	1.01	1.00
$T = 100$	1.12	1.03	1.01	1.01
			TR_{ML}/TR_{TS}	
$T = 50$	1.03	1.00	1.00	1.00
$T = 100$	1.05	1.00	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 3$.

Table 5: Doz et al. (2012, p. 1020) Table 3

	N =10	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}	
$T = 50$	0.54	0.65	0.68	0.70
$T = 100$	0.66	0.78	0.81	0.82
			Number of iterations	
$T = 50$	21	9	6	5
$T = 100$	15	7	5	4
			TR_{ML}/TR_{PC}	
$T = 50$	1.14	1.06	1.03	1.01
$T = 100$	1.19	1.06	1.02	1.01
			TR_{ML}/TR_{TS}	
$T = 50$	1.07	1.02	1.01	1.00
$T = 100$	1.10	1.01	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1020, Table 3).

Table 6: Replication of original Table 3 in Doz et al. (2012)

	N =10	$N = 25$	$N = 50$	$N = 100$
			TR_{ML}	
$T = 50$	0.56	0.65	0.68	0.70
$T = 100$	0.70	0.78	0.80	0.82
			Number of iterations	
$T = 50$	21	8	5	4
$T = 100$	13	5	4	3
			TR_{ML}/TR_{PC}	
$T = 50$	1.16	1.07	1.03	1.01
$T = 100$	1.20	1.07	1.02	1.01
			TR_{ML}/TR_{TS}	
$T = 50$	1.08	1.02	1.00	1.00
$T = 100$	1.10	1.01	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0$, $\tau = 0$, $u = 0.1$, $q = 3$.

considered in the previous section.

The additional set of simulations we run can be motivated as follows: in most cases, the interest of the practitioner is in using the estimated factors as a by-product for a model of interest, that can be used for forecasting/nowcasting purposes, structural analysis or more.

In a time-series context (especially at higher frequencies), it is very likely that the factors may affect the observables in a lag-distributed pattern, so the proper value of s in equation (1) is larger than 0. As is well known (for example, see Bai and Ng (2007)), equation (1) can be re-written in a "static" form

$$x_t = \mathbf{\Lambda} \boldsymbol{F}_t + e_t,$$

where the vector $\boldsymbol{F}_t$ contains f_t and its lags, suitably stacked, and $\mathbf{\Lambda}$ is the horizontal concatenation of the Λ_i matrices. The dimension of the $\boldsymbol{F}_t$ vector is $q \cdot (s+1)$. The interest of the practitioner is arguably to use a method that reconstructs as closely as possible the space spanned by $\boldsymbol{F}_t$. The possibilities we investigate are

1. compute the q-vector $\hat{f}_t$ via ML and then stack its lags to form $\hat{\boldsymbol{F}}_t$;
2. compute the q-vector $\hat{f}_t$ via TS and then stack its lags to form $\hat{\boldsymbol{F}}_t$;
3. compute $\hat{\boldsymbol{F}}_t$ by extracting the first $q \cdot s$ principal components;
4. approximate $\hat{\boldsymbol{F}}_t$ via the first principal component;
5. approximate $\hat{\boldsymbol{F}}_t$ via the first principal component and its lags.

In order to initialise the algorithm, we use the method outlined[4] in Stock and Watson (2005, p. 13), Forni and Gambetti (2010, p. 206) and Lütkepohl (2014, p. 14) to retrieve the dynamic factor space from principal components. It entails running a p-order VAR on estimated principal components and then performing PC extraction on the residuals.

The last two methods are clearly sub-optimal, but we consider them to evaluate how serious the shortcoming is, in the light of the fact that in many cases it is customary for practitioners to use the first principal component as the "dominant" factor, regardless of any statistically oriented criterion, (see for example Bai and Ng (2007)).

We performed the experiment with a wide variety of alternative values for the DGP parameters; to be specific, we experimented with different values for α, d, q and s, thereby considering different scenarios on factor persistence, idiosyncratic error persistence and lag specifications in the observation equation; the full results are not tabulated here for the sake of conciseness, but are available upon request. The results for the experiment we chose to show, as representative of the whole set of results, are displayed in table 7.

As shown in table 7, ML clearly dominates TS and all the methods based on a limited set of principal components. The only way to achieve a qualitatively similar approximation to the space spanned by the true factors is by using the full set of principal components, compared to which the advantage of using ML becomes smaller, especially for larger values of T and N.

[4] In an alternative multi-step estimation setup based on principal components and least squares.

Table 7: Extended experiment

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
		TR_{ML}		
$T = 50$	0.58	0.65	0.69	0.70
$T = 100$	0.70	0.79	0.81	0.82
		Number of iterations		
$T = 50$	46	27	22	21
$T = 100$	25	15	13	12
		TR_{ML}/TR_{TS}		
$T = 50$	1.57	1.25	1.12	1.11
$T = 100$	1.70	1.22	1.16	1.08
		TR_{ML}/TR_{PC}		
$T = 50$	1.09	1.08	1.06	1.04
$T = 100$	1.13	1.09	1.05	1.03
		$TR_{ML}/TR_{PC,1}$		
$T = 50$	1.76	1.65	1.62	1.59
$T = 100$	1.86	1.76	1.69	1.66
		$TR_{ML}/TR_{PC,1,lags}$		
$T = 50$	1.59	1.54	1.52	1.50
$T = 100$	1.74	1.68	1.63	1.60

Notes: Comparing alternative dynamic factor extraction methods when $s = 1$. Data generation process input set at: $s = 1$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 2$.

The EM-based ML method exhibits a noticeably larger overhead in terms of computational complexity than in the original Monte Carlo experiment by Doz et al. (2012). This was to be expected, since the structure of the model we are considering is significantly more complex in terms of lag specification ($s = 1$ in equation (1)).

Perhaps the relative superiority of PC to TS may be considered surprising at first sight. However, the two methods use a different number of factors, since PC uses $s \cdot q$ factors and estimates consistently the factor space, whereas TS only uses q factors (plus their lags), so this result should be taken as an indication that TS may perform rather poorly in finite samples.

It is interesting to compare the three PC-based models under the viewpoint of misspecification: clearly, method 3 (PC) is a correct way to estimate the full set of static factors, whereas methods 4 and 5 (*PC,1* and *PC,1,lags*) are not, in that they are based on a specification of equation (1) where s or q are too small. It is evident - last two sub-panels of table 7 - that the effects of misspecification are quite dramatic and larger values of T and N do not mitigate the shortcomings.

Our set of extra results can be summarised as follows: the most important element to consider is persistence, either as a characteristic of latent factors or as the lag length in the observation equation (1). When persistence is high, factors are estimated more accurately with maximum likelihood than any of the other methods we considered; the only exception is principal components for the full set of static factors, which on the other hand has the disadvantage of requiring a set of $q \cdot s$ series instead of just q. Anyhow, the near-equivalence of ML and PC requires T and/or N to be rather large; in mid-sized samples ML retains a noticeable edge.

5 Conclusions and possible extensions

We use an independent software implementation to replicate the simulation results of Doz et al. (2012) regarding factor estimation in dynamic settings using the benchmark principal component method, a two step Kalman smoother based method and EM-based maximum likelihood estimation.

Our narrow sense replication exercise fully confirms the results in the original article. As for our extended replication experiment, we find that ML is the dominant method in a wide array of situations, notably when persistence is substantial. Those results go one step further than the closing remark by Doz et al. (2012) *"...Efficiency improvements are relevant when the factor extraction is difficult, that is, when there are more common factors to estimate... "*, by including a richer set of dynamic specifications and the possibility of misspecification.

The present work could be extended in several directions: for instance, the lag orders s and p and the number of dynamic factors q are assumed to be known and fixed *ex ante*; it may be conjectured that different methods could display different performance if the lag orders have to be selected or if q has to be estimated through criteria such as the one put forward in Hallin and Liška (2007). Another possible extension could be using, for the comparison, a DGP taken from some empirical exercise. These extensions, however, would broaden the present article's scope considerably and are left for future work.

References

Bai J, Li K. 2016. Maximum likelihood estimation and inference for approximate factor models of high dimension. *The Review of Economics and Statistics,* **98**: 298–309.

Bai J, Ng S. 2007. Determining the number of primitive shocks in factor models. *Journal of Business & Economic Statistics,* **25**: 52–60.

Bai J, Ng S. 2008. Large dimensional factor analysis. *Foundations and Trends in Econometrics,* **3**: 89–163.

Barigozzi M, Hallin M. 2015. Generalized dynamic factor models and volatilities: recovering the market volatility shocks. *The Econometrics Journal,* **19**: C33–C60.

Barigozzi M, Hallin M. 2017. Generalized dynamic factor models and volatilities: estimation and forecasting. *Journal of Econometrics,* **201**: 307–321.

Barigozzi M, Hallin M, Soccorsi S. 2018. Identification of global and local shocks in international financial markets via general dynamic factor models. *Journal of Financial Econometrics,* **17**: 462–494.

Breitung J, Tenhofen J. 2011. GLS Estimation of Dynamic Factor Models. *Journal of the American Statistical Association,* **106**: 1150–1166.

Doz C, Giannone D, Reichlin L. 2011. A two-step estimator for large approximate dynamic factor models based on Kalman filtering. *Journal of Econometrics,* **164**: 188–205.

Doz C, Giannone D, Reichlin L. 2012. A quasi maximum likelihood approach for large approximate dynamic factor models. *Review of Economics and Statistics,* **94**: 1014–1024.

Forni M, Gambetti L. 2010. The dynamic effects of monetary policy: A structural factor model approach. *Journal of Monetary Economics,* **57**: 203–216.

Forni M, Giovannelli A, Lippi M, Soccorsi S. 2018. Dynamic factor model with infinite-dimensional factor space: Forecasting. *Journal of Applied Econometrics,* **33**: 625–642.

Forni M, Hallin M, Lippi M, Reichlin L. 2000. The generalized dynamic-factor model: Identification and estimation. *Review of Economics and statistics,* **82**: 540–554.

Forni M, Hallin M, Lippi M, Reichlin L. 2005. The generalized dynamic factor model: one-sided estimation and forecasting. *Journal of the American Statistical Association,* **100**: 830–840.

Forni M, Hallin M, Lippi M, Zaffaroni P. 2015. Dynamic factor models with infinite-dimensional factor spaces: One-sided representations. *Journal of Econometrics,* **185**: 359–371.

Forni M, Hallin M, Lippi M, Zaffaroni P. 2017. Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis. *Journal of Econometrics,* **199**: 74–92.

Hallin M, Liška R. 2007. Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association,* **102**: 603–617.

Lütkepohl H. 2014. Structural Vector Autoregressive Analysis in a Data Rich Environment: A Survey. Discussion Papers of DIW Berlin 1351, DIW Berlin, German Institute for Economic Research.
URL `https://ideas.repec.org/p/diw/diwwpp/dp1351.html`

Lucchetti R, Venetis IA. 2019. Dynamic Factor Models in gretl. The DFM package. gretl working papers 7, Università Politecnica delle Marche (I), Dipartimento di Scienze Economiche e Sociali.

Luciani M. 2015. Monetary policy and the housing market: A structural factor analysis. *Journal of Applied Econometrics,* **30**: 199–218.

Scotti C. 2016. Surprise and uncertainty indexes: Real-time aggregation of real-activity macro-surprises. *Journal of Monetary Economics,* **82**: 1 – 19.

Stock JH, Watson MW. 2005. Implications of Dynamic Factor Models for VAR Analysis. NBER working paper no. 11467, issued in July 2005.
URL `http://www.nber.org/papers/w11467`

Stock JH, Watson MW. 2011. Dynamic factor models. In Clements MJ, Hendry DF (eds.) *Oxford Handbook on Economic Forecasting*, chapter 2. Oxford: Oxford University Press, 35–39.

Stock JH, Watson MW. 2016. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In Taylor JB, Uhlig H (eds.) *Handbook of Macroeconomics*, volume 2, chapter 8. Netherlands, Amsterdam: Elsevier, 415–525.