MONASH University

MONASH BUSINESS SCHOOL

ISSN 1440-771X

Department of Econometrics and Business Statistics

http://monash.edu/business/ebs/research/publications

Fast forecast reconciliation using linear models

Mahsa Ashouri, Rob J Hyndman, Galit Shmueli

April 2021

Working Paper 29/19

Fast forecast reconciliation using linear models

Mahsa Ashouri
Institute of Service Science, National Tsing Hua University, Taiwan
Email: mahsa.ashouri@iss.nthu.edu.tw
Corresponding author

Rob J Hyndman
Monash University, Clayton VIC 3800, Australia
Email: rob.hyndman@monash.edu

Galit Shmueli
Institute of Service Science, National Tsing Hua University, Taiwan
Email: galit.shmueli@iss.nthu.edu.tw

19 April 2021

JEL classification: C10,C14,C22

Fast forecast reconciliation using linear models

Abstract

Forecasting hierarchical or grouped time series using a reconciliation approach involves two steps: computing base forecasts and reconciling the forecasts. Base forecasts can be computed by popular time series forecasting methods such as Exponential Smoothing (ETS) and Autoregressive Integrated Moving Average (ARIMA) models. The reconciliation step is a linear process that adjusts the base forecasts to ensure they are coherent. However using ETS or ARIMA for base forecasts can be computationally challenging when there are a large number of series to forecast, as each model must be numerically optimized for each series. We propose a linear model that avoids this computational problem and handles the forecasting and reconciliation in a single step. The proposed method is very flexible in incorporating external data. We illustrate our approach using a dataset on monthly Australian domestic tourism, as well as a simulated dataset. We compare our approach to reconciliation using ETS and ARIMA, and show that our approach is much faster while providing similar levels of forecast accuracy.

Keywords: hierarchical forecasting, grouped forecasting, reconciling forecast, linear regression

1 Introduction

Modern data collection tools have dramatically increased the amount of available time series data (Januschowski et al. 2013). For example, the internet of things (IoT) and point-of-sale (POS) scanning produce huge volumes of time series in a short period of time. Naturally, there is an interest in forecasting these time series, yet forecasting large collections of time series is computationally challenging. Although computers have become faster and cheaper, computational intensity remains an important issue for many businesses and non-for-profit organizations that employ forecasting (Makridakis, Spiliotis & Assimakopoulos 2018).

In many cases, time series can be structured and disaggregated based on hierarchies or groups such as geographic location, product type, gender, etc. Such hierarchical or grouped time series arise in a variety of applications where forecasts at different levels lead to different types of actions. For example, electricity demand and smart grid management can benefit from forecasts

at the level of individual households, transformers, feeders, substations, balancing areas, cities, regions, and countries (Taieb, Taylor & Hyndman 2020). In tourism, forecasting at the national and various sub-national levels are fundamental to managers, government, and policy-makers for effective planning (Makoni, Chikobvu & Sigauke 2021).

An example of a hierarchical time series is the monthly number of Australian domestic tourists, which can be disaggregated into different states and then into different zones. Figure 1 shows a schematic of such a hierarchical time series structure with three levels. The top level is the total series, formed by aggregating all the bottom-level series. In the middle level, series are aggregations of their own child series; for instance, series A is the aggregation of AW and AX. Finally, the bottom-level series includes the most disaggregated series. In our example, A might represent the Northern Territory (NT) state, which can be disaggregated into northern coast NT and central NT.

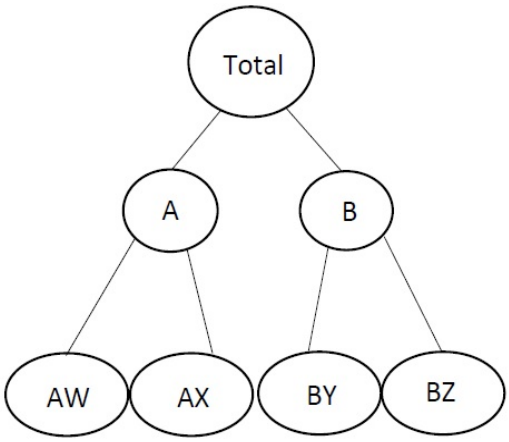

Figure 1: *An example of a two level hierarchical structure.*

Grouped time series involve more complicated aggregation structures compared to strictly hierarchical time series. To continue our Australian tourism example, suppose we have two grouping factors for each tourist that are not nested: purpose of travel (Business/Holiday) and sex (Male/Female). The disaggregated series for each combination of purpose of travel and sex can be combined to form purpose of travel subtotals, or sex subtotals. These subtotals can be combined to give the overall total. Both subtotals are of interest.

We can think of such structures as hierarchical time series without a unique hierarchy. A schematic of this grouped time series structure is shown in Figure 2 with two grouping factors, each of two levels (A/B and C/D). The series in this structure can be split first into groups A and B and then subdivided further into C and D (left side), or split first into C and D and then

subdivided into A and B (right side). The final disaggregation is identical in both cases, but the middle level aggregates are different.

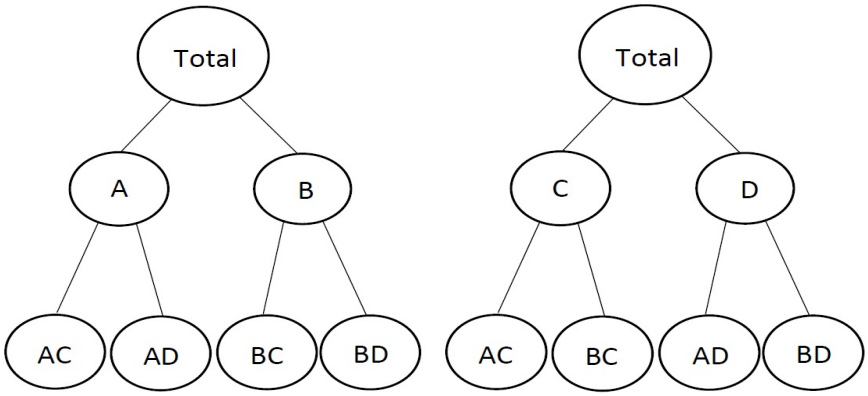

Figure 2: *An example of a two level grouped structure.*

Hierarchy and grouping structures offer useful information for improving forecasts of individual series within those structures, due to the correlation between different hierarchy/grouping levels. Useful information may be extracted from the aggregate series that would otherwise be potentially lost at the individual series (Syntetos et al. 2016). Moreover, if we just forecast each series individually, we are ignoring the hierarchical or grouping structure, and the forecasts will not be 'coherent'. That is, forecasts for lower level series will not necessarily add up to forecasts for higher level series. This means forecasts will not add up in a way that is consistent with the aggregation structure of the time series collection (Hyndman & Athanasopoulos 2018).

There are several available methods that consider the hierarchical structure information when forecasting time series. These include the top-down (Gross & Sohl 1990; Fliedner 2001), bottom-up (Kahn 1998), middle-out and optimal combination (Hyndman et al. 2011) approaches. In the top-down approach, we first forecast the total series and then disaggregate the forecast to form lower level series forecasts based on a set of historical or forecasted proportions (for details see Athanasopoulos, Ahmed & Hyndman 2009). In the bottom-up approach, the forecasts in each level of the hierarchy can be computed by aggregating the bottom-level series forecasts. However, we may not get good upper-level forecasts because the most disaggregated series can be noisy and so their forecasts are often inaccurate. In the middle-out approach, the process can be started from one of the middle levels and other forecasts can be computed using aggregation for upper levels and disaggregation for lower levels. Finally, optimal combination uses all the forecasts for all of the series in the entire structure, and then uses an optimization process to reconcile the resulting forecasts. The advantage of the optimal combination method, compared

with the other methods, is that it considers all information in the hierarchy, including any correlations among the series.

We use the same notation for both hierarchical and grouped time series, (following Hyndman & Athanasopoulos 2018) for both hierarchical and grouped time series. We denote the total series at time t by y_t, and the series at node Z (subaggregation level Z) and time t by $y_{Z,t}$. For describing the relationships between series, we use an $N \times M$ matrix, called the 'summing matrix', denoted by $\boldsymbol{S}$, in which N is the overall number of nodes and M is the number of bottom-level nodes. For example, in Figure 1, $N = 7$ and $M = 4$, while in Figure 2, $N = 9$ and $M = 4$. Then we can write $\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$, where $\boldsymbol{y}_t$ is a vector of all the level nodes at time t and $\boldsymbol{b}_t$ is the vector of all the bottom-level nodes at time t. For the example shown in Figure 2, the equation can be written as follows:

$$\begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{D,t} \\ y_{AC,t} \\ y_{AD,t} \\ y_{BC,t} \\ y_{BD,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AC,t} \\ y_{AD,t} \\ y_{BC,t} \\ y_{BD,t} \end{pmatrix}.$$

In the optimal combination method, reconciled forecasts can be computed using the following generalized least squares equation (Wickramasuriya, Athanasopoulos & Hyndman 2019)

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_{t+h}, \tag{1}$$

where $\hat{\boldsymbol{y}}_{t+h}$ represents a vector of h-step-ahead base forecasts for all levels of the hierarchy, and $\boldsymbol{W}_h$ is the covariance matrix of forecast errors for the h-step-ahead base forecasts.

Several possible methods for estimating $\boldsymbol{W}_h$ are available. Wickramasuriya, Athanasopoulos & Hyndman (2019) discuss a simple 'structural scaling' approximation whereby $\boldsymbol{W}_h = k_h\boldsymbol{\Lambda}$ with k_h being a positive constant, $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{S}\boldsymbol{1})$, and $\boldsymbol{1}$ being a unit vector of dimension M (the number of bottom-level series). Note that $\boldsymbol{\Lambda}$ simply contains the row sums of the summing matrix $\boldsymbol{S}$, and

that k_h will cancel out in (1).[1] Thus

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\hat{\boldsymbol{y}}_{t+h}. \tag{2}$$

The most computationally challenging part of the optimal combination method is producing all the base forecasts that make up $\hat{\boldsymbol{y}}_{t+h}$. In many applications, there may be thousands or even millions of individual series, and each of them must be forecast independently. The most popular time series forecasting methods, such as ETS and ARIMA models (Hyndman & Athanasopoulos 2018) , involve non-linear optimization routines to estimate the parameters via maximum likelihood estimation. Usually, multiple models are fitted for each series, and the best is selected by minimizing Akaike's Information Citerion (Akaike 1998). This computational challenges increases with the number of lower-level series as well as in the number of aggregations of interest.

We therefore propose a new approach to compute the base forecasts that is both computationally fast while maintaining an acceptable forecasting accuracy level. Our proposed approach avoids the computational challenge of using ETS or ARIMA that require numerical optimization for each series. It is useful in terms of incorporating external data, handling missing values, and model selection. And finally, our approach handles the forecasting and reconciliation in a single step.

2 Proposed approach: Linear model

Our proposed approach is based on using linear regression models for computing base forecasts. Suppose we have a linear model that we use for forecasting, and we wish to apply it to N different series which have some aggregation constraints. We have observations $y_{t,i}$ from times $t = 1, \dots, T$ and series $i = 1, \dots, N$. Then

$$y_{t,i} = \boldsymbol{\beta}_i' \boldsymbol{x}_{t,i} + \varepsilon_{t,i}$$

where $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ is a $(p+1)$-vector of regression variables, $\boldsymbol{\beta}_i = (\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \dots, \beta_{p,i})$ is a $(p+1)$-vector of coefficients, and $\varepsilon_{t,i}$ is the error[2]. This equation

[1] For simplicity of exposition, we used the structural scaling (wls_struct) summing matrix for reconciliation in all the results. In Appendix A.1, we compare two alternatives: Tables 15 and 16 display the results for estimates of $\boldsymbol{W}_h$ using a shrinkage estimator (`mint_shrink`) and variance scaling (`wls_var`) for the Australian tourism example.

[2] If different predictors are chosen for different series, we can still choose a constant p that includes all the predictors, and then set the relevant coefficients in the X matrix to 0.

for all the observations in matrix form can be written as follows:

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \vdots \\ \boldsymbol{y}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{X}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{X}_N \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_3 \\ \vdots \\ \boldsymbol{\beta}_N \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \boldsymbol{\varepsilon}_3 \\ \vdots \\ \boldsymbol{\varepsilon}_N \end{pmatrix} \tag{3}$$

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E},$$

where $\boldsymbol{y}_i = (y_{1,i}, y_{2,i}, \dots, y_{T,i})$ is a T-vector, $\boldsymbol{\beta}_i = (\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \dots, \beta_{p,i})$ is a $(p+1)$-vector, $\boldsymbol{\varepsilon}_i = (\varepsilon_{1,i}, \varepsilon_{2,i}, \dots, \varepsilon_{T,i})$ is a T-vector, and $\boldsymbol{X}_i$ is the $T \times (p+1)$-matrix

$$\boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}.$$

Equation (3) can be written as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}$, with parameter estimates given by $\hat{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$. Then the base forecasts are obtained using

$$\hat{\boldsymbol{y}}_{t+h} = \boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}, \tag{4}$$

where $\hat{\boldsymbol{y}}_{t+h}$ is an N-vector of forecasts, $\hat{\boldsymbol{B}}$ comprises N stacked $(p+1)$-vectors of estimated coefficients, and $\boldsymbol{X}^*_{t+h}$ is the $N \times N(p+1)$ matrix

$$\boldsymbol{X}^*_{t+h} = \begin{pmatrix} \boldsymbol{x}'_{t+h,1} & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{x}'_{t+h,2} & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{x}'_{t+h,3} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{x}'_{t+h,N} \end{pmatrix}.$$

Note that we use $\boldsymbol{X}^*_t$ to distinguish this matrix, which combines $\boldsymbol{x}_{t,i}$ across all series for one time, from $\boldsymbol{X}_i$ which combines $\boldsymbol{x}_{t,i}$ across all times for one series.

Finally, we can combine the two linear equations for computing base forecasts and reconciled forecasts (Equations (2) and (4)) to obtain the reconciled forecasts with a single equation:

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}(\boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{X}^*_{t+h}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}. \tag{5}$$

2.1 Simplified formulation for a fixed set of predictors (X)

If we have the same set of predictor variables, $\boldsymbol{X}$, for all the series, we can write Equations (3) to (5) more easily using multivariate regression equations, and we can obtain all the reconciled forecasts for all the series in one equation. In that case, Equation (3) can be rearranged as follows:

$$\begin{pmatrix} y_{11} & \dots & y_{1N} \\ y_{21} & \dots & y_{2N} \\ \vdots & & \vdots \\ y_{T1} & \dots & y_{TN} \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & X_{21} & \dots & X_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & X_{T1} & \dots & X_{Tp} \end{pmatrix} \begin{pmatrix} \beta_{01} & \dots & \beta_{0N} \\ \beta_{11} & \dots & \beta_{1N} \\ \vdots & & \vdots \\ \beta_{p1} & \dots & \beta_{pN} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \dots & \varepsilon_{1N} \\ \varepsilon_{21} & \dots & \varepsilon_{2N} \\ \vdots & & \vdots \\ \varepsilon_{T1} & \dots & \varepsilon_{TN} \end{pmatrix}, \tag{6}$$

where $\boldsymbol{Y}$, $\boldsymbol{X}$, $\boldsymbol{B}$ and $\boldsymbol{E}$ are now matrices of size $T \times N$, $T \times (p+1)$, $(p+1) \times N$ and $T \times N$, respectively. Equations (4) to (5) can be written accordingly using Equation (6), and here $\boldsymbol{X}^*_{t+h,i} = \boldsymbol{X}^*_{t+h}$, where $\boldsymbol{X}^*_{t+h}$ is an $h \times (p+1)$ matrix. This simpler formulation translates into a computational advantage, as the $\boldsymbol{X}$ matrix is smaller, thereby easing matrix multiplication operations.

2.2 OLS predictors

As an example of the $\boldsymbol{X}_t$ matrix in Equation (3), we can refer to the set of predictors proposed in Ashouri, Shmueli & Sin (2018) for modeling trend, seasonality, and autocorrelation by using lagged values ($y_{t-1}, y_{t-2}, \dots$), trend variables, and seasonal dummy variables:

$$y_t = \alpha_0 + \alpha_1 t + \beta_1 s_{1,t} + \cdots + \beta_{m-1} s_{m-1,t} + \gamma_1 y_{t-1} + \cdots + \gamma_p y_{t-p} + \delta z_t + \varepsilon_t.$$

Here, $s_{j,t}$ is a dummy variable taking value 1 if time t is in season j ($j = 1, 2, \dots, m$), y_{t-k} is the kth lagged value for y_t, and z_t is some external information at time t. The seasonal period m depends on the problem; for instance, if we have daily data with day-of-week seasonality, then $m = 7$.

When a single OLS model is fitted to a collection of time series (e.g.~several bottom-level series), then trend and seasonality predictors are the same for all series and we can use the simpler multivariate regression models in Equation (6). However, this formulation is inappropriate

when including lags, which differ for each series, and/or series-specific external series, in which case we use the formulation in Equation (3).

When there are many options for choosing predictors, such as many seasonal dummy variables, lags, or high order trend terms, we can consider using model selection criteria such as Akaike's Information Criterion (AIC) or model selection approaches such as leave-one-out cross-validation (LOOCV) to select the best set of predictors in terms of forecast accuracy. In practice, LOOCV can be computationally heavy except in the special case of linear model (Christensen 2020) aand therefore using linear models provide a viable solution. Also, when the number of seasons m is large (e.g. in hourly data), Fourier terms can result in fewer predictors than dummy variables. The number of Fourier terms can also be determined using the same AIC or LOOCV approach (Hyndman & Athanasopoulos 2018).

2.3 Computational considerations

There are two ways for implementing the above equations and} computing forecasts. First, we can create the matrices $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{E}$, and then directly use the above equations to obtain the forecasts by taking advantage of sparse matrix routines (Bunch & Rose 2014), such as R's Matrix package (Maechler & Bates (2006)) to obtain the forecasts. Alternatively, we could use separate regression models to compute the coefficients for each linear model individually. Although the matrix $\boldsymbol{X}'\boldsymbol{X}$ which we need to invert is sparse and block diagonal, we find that it is still faster to use the second approach involving separate regression models.

2.4 Prediction intervals

For obtaining prediction intervals, we need to compute the variance of reconciled forecasts as follows (Wickramasuriya, Athanasopoulos & Hyndman 2019):

$$\text{Var}(\tilde{\boldsymbol{y}}_{t+h}) = \boldsymbol{S}\boldsymbol{G}\boldsymbol{\Sigma}_{t+h}\boldsymbol{G}'\boldsymbol{S}', \tag{7}$$

where $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}_{t+h}$ denotes the variance of the base forecasts given by the usual linear model formula (Hyndman & Athanasopoulos 2018)

$$\boldsymbol{\Sigma}_{t+h} = \sigma^2 \left[1 + \boldsymbol{X}^*_{t+h}(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}^*_{t+h})'\right].$$

where σ^2 is the variance of the base model residuals. Assuming normally distributed errors, we can easily obtain any required prediction intervals corresponding to elements of $\tilde{\boldsymbol{y}}_{t+h}$ using the diagonals of (7).

3 Applications

In this section we illustrate our approach using a real dataset and a simulated dataset[3]. The real data study includes forecasting monthly Australian domestic tourism. This dataset contains 304 series with both hierarchical and grouped structure with strong seasonality. In the simulation studies, we simulate series based on the monthly Australian domestic tourism data and systematically modify the forecasting horizon, noise level, hierarchy levels, and number of series. In Appendix B, we use another real dataset involving daily Wikipedia pageviews. In all these cases, we compare the forecasting accuracy of ETS, ARIMA[4] and the proposed linear OLS forecasting model, with and without the reconciliation step. In these applications, we use the weighted reconciliation approach from Equation (2). For comparing these methods, we use the average of Root Mean Square Errors (RMSEs) across all series and also display box plots for forecast errors along with the raw forecast errors. A forecast error is defined as the difference between an observed value and its forecast. To aid visibility, we suppress plotting the outliers.

We apply two methods for generating forecasts that align with two different practical forecasting scenarios. The first approach is *rolling origin* forecasting, where we generate one-step-ahead forecasts ($\tilde{\boldsymbol{y}}_{t+1}$ where t changes). This mimics the scenario where data are refreshed every time period. In the second *fixed origin* method, forecasts are generated at a fixed time t for h steps ahead: $\tilde{\boldsymbol{y}}_{t+1}, \tilde{\boldsymbol{y}}_{t+2}, \ldots, \tilde{\boldsymbol{y}}_{t+h}$ (we replace lagged values of y by their forecasts if they occur at periods after the forecast origin). See algorithms 1 and 2 for details of the rolling and fixed origin approaches).

3.1 Australian domestic tourism

This dataset has 19 years of monthly visitor nights in Australia by Australian tourists, a measure used as an indicator of tourism activity (Wickramasuriya, Athanasopoulos & Hyndman 2019). The data were collected by computer-assisted telephone interviews with 120,000 Australians aged 15 and over (Tourism Research Australia 2005). The dataset includes 304 time series, each of length 228 observations. The hierarchy and grouping structure for this dataset is made using geographic and purpose of travel information.

Table 1: *Australian geographic hierarchical structure.*

Series	Name	Label	Series	Name	Label

[3]All methods were run on a Linux server with Intel Xeon Silver 4108 (1.80GHz / 8-Cores / 11MB Cache)*2 and 8GB DDR4 2666 DIMM ECC Registered Memory. R version 3.6.1. All the displayed computation times are for the unreconciled point forecasts.

[4]For running ETS and ARIMA, we applied `ETS` and `ARIMA` functions from the `fable` package (O Hara-Wild, Hyndman & Wang 2019). The two sets of functions were run independently and not immediately one after the other.

Total			Region		
1	Australia	Total	55	Lakes	BCA
State			56	Gippsland	BCB
2	NSW	A	57	Phillip Island	BCC
3	VIC	B	58	General Murray	BDA
4	QLD	C	59	Goulburn	BDB
5	SA	D	60	High Country	BDC
6	WA	E	61	Melbourne East	BDD
7	TAS	F	62	Upper Yarra	BDE
8	NT	G	63	Murray East	BDF
Zone			64	Wimmera+Mallee	BEA
9	Metro NSW	AA	65	Western Grampians	BEB
10	Nth Coast NSW	AB	66	Bendigo Loddon	BEC
11	Sth Coast NSW	AC	67	Macedon	BED
12	Sth NSW	AD	68	Spa Country	BEE
13	Nth NSW	AE	69	Ballarat	BEF
14	ACT	AF	70	Central Highlands	BEG
15	Metro VIC	BA	71	Gold Coast	CAA
16	West Coast VIC	BB	72	Brisbane	CAB
17	East Coast VIC	BC	73	Sunshine Coast	CAC
18	Nth East VIC	BD	74	Central Queensland	CBA
19	Nth West VIC	BE	75	Bundaberg	CBB
20	Metro QLD	CA	76	Fraser Coast	CBC
21	Central Coast QLD	CB	77	Mackay	CBD
22	Nth Coast QLD	CC	78	Whitsundays	CCA
23	Inland QLD	CD	79	Northern	CCB
24	Metro SA	DA	80	Tropical North Queensland	CCC
25	Sth Coast SA	DB	81	Darling Downs	CDA
26	Inland SA	DC	82	Outback	CDB
27	West Coast SA	DD	83	Adelaide	DAA
28	West Coast WA	EA	84	Barossa	DAB
29	Nth WA	EB	85	Adelaide Hills	DAC
30	Sth WA	EC	86	Limestone Coast	DBA
31	Sth TAS	FA	87	Fleurieu Peninsula	DBB
32	Nth East TAS	FB	88	Kangaroo Island	DBC
33	Nth West TAS	FC	89	Murraylands	DCA
34	Nth Coast NT	GA	90	Riverland	DCB
35	Central NT	GB	91	Clare Valley	DCC
Region			92	Flinders Range and Outback	DCD
36	Sydney	AAA	93	Eyre Peninsula	DDA
37	Central Coast	AAB	94	Yorke Peninsula	DDB
38	Hunter	ABA	95	Australia's Coral Coast	EAA

39	North Coast NSW	ABB	96	Experience Perth	EAB
40	Northern Rivers Tropical NSW	ABC	97	Australia's SouthWest	EAC
41	South Coast	ACA	98	Australia's North West	EBA
42	Snowy Mountains	ADA	99	Australia's Golden Outback	ECA
43	Capital Country	ADB	100	Hobart and the South	FAA
44	The Murray	ADC	101	East Coast	FBA
45	Riverina	ADD	102	Launceston, Tamar and the North	FBB
46	Central NSW	AEA	103	North West	FCA
47	New England North West	AEB	104	Wilderness West	FCB
48	Outback NSW	AEC	105	Darwin	GAA
49	Blue Mountains	AED	106	Kakadu Arnhem	GAB
50	Canberra	AFA	107	Katherine Daly	GAC
51	Melbourne	BAA	108	Barkly	GBA
52	Peninsula	BAB	109	Lasseter	GBB
53	Geelong	BAC	110	Alice Springs	GBC
54	Western	BBA	111	MacDonnell	GBD

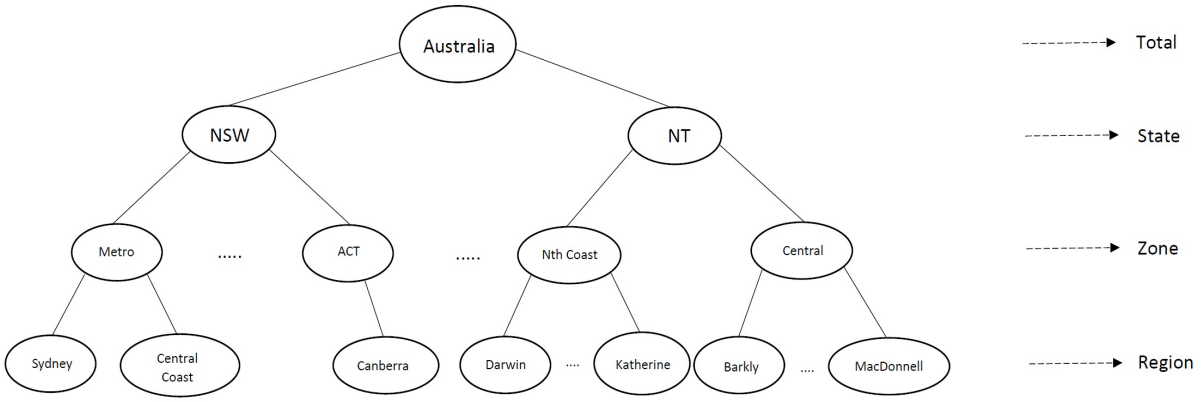

Figure 3: *Australian geographic hierarchical structure.*

This dataset uses the three levels of geographic divisions in Australia. In the first level, Australia is divided into seven 'States' including New South Wales (NSW), Victoria (VIC), Queensland (QLD), South Australia (SA), Western Australia (WA), Tasmania (TAS) and Northern Territory (NT). In the second and third levels, it is divided into 27 'Zones' and 76 'Regions'. This geographical structure is detailed in Table 1, shown schematically in Figure 3 , and as a map chart in Figure 4[5].

In addition to geography, we have four purposes of travel: Holiday (Hol), Visiting friends and relatives (Vis), Business (Bus) and Other (Oth). This results in $76 \times 4 = 304$ series at the most disaggregate level. Based on the geographic hierarchy and purpose grouping, we end up with 8 aggregation levels with 555 series in total, as shown in Table 2.

[5]Reproduced from www.tra.gov.au/tra/2016/Tourism_Region_Profiles/Region_profiles/index.html

Figure 4: *Australia map, showing the eight states.*

Table 2: *Number of Australian domestic tourism series at each aggregation level.*

Aggregation_Level	Series
Australia	1
State	7
Zone	27
Region	76
Purpose	4
State x Purpose	28
Zone x Purpose	108
Region x Purpose	304
Total	555

Algorithm 1 Hierarchical and grouped time series rolling origin OLS forecast reconciliation

$data \leftarrow matrix(y_{1:T,1:N}, x_{1:T,1:N,1:p})$
for $i \in \{1, \dots, N\}$ **do**
  for $k \in \{1, \dots, h\}$ **do**
    $data \leftarrow data_{(1:(T-h)+(k-1)),i}$
    $newdata \leftarrow data_{((T-h)+k),i}$
    $fit \leftarrow lm(y_{t,i} \sim x_{t,i,1} + x_{t,i,2} + \cdots + x_{t,i,p}, data = data)$
    $\hat{y}_{t+k,i} \leftarrow predict(fit, newdata = newdata)$
  end for
end for
$\boldsymbol{S} \leftarrow summing\ matrix(groups_{data})$
$\Lambda \leftarrow diag(\boldsymbol{S}\mathbf{1})$
$adjust \leftarrow \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}\Lambda$
for $i \in 1, \dots, h$ **do**
  $\tilde{y}_{(T-h)+i,1:N} \leftarrow adjust \times \hat{y}_{(T-h)+i,1:N}$
end for
return $\tilde{y}_{((T-h):T,1:N}$

Algorithm 2 Hierarchical and grouped time series fixed origin OLS forecast reconciliation

$data \leftarrow matrix(y_{1:T,1:N}, x_{1:T,1:N,1:p})$
for $i \in \{1, \dots, N\}$ **do**
  $data \leftarrow data_{(1:(T-h),i}$
  $fit \leftarrow lm(y_{t,i} \sim x_{t,i,1} + x_{t,i,2} + \cdots + x_{t,i,p}, data = data)$
  $newdata \leftarrow data_{((T-h)+1),i}$
  for $k \in \{1, \dots, h\}$ **do**
    $\hat{y}_{t+k,i} \leftarrow predict(fit, newdata = newdata)$
    $newdata \leftarrow \hat{data}_{((T-h)+k),i}$
  end for
end for
$\boldsymbol{S} \leftarrow summing\ matrix(groups_{data})$
$\Lambda \leftarrow diag(\boldsymbol{S}\mathbf{1})$
$adjust \leftarrow \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}\Lambda$
for $i \in 1, \dots, h$ **do**
  $\tilde{y}_{(T-h)+i,1:N} \leftarrow adjust \times \hat{y}_{(T-h)+i,1:N}$
end for
return $\tilde{y}_{((T-h):T,1:N}$

We compare the forecast error distributions for all these aggregation levels (Figures 5 and 6, as well as the average RMSE (Tables 3 and 4). We use the same predictors in the OLS predictor matrix for the rolling and fixed origin approaches: a quadratic trend, 11 dummy variables, and lags 1 and 12. This set is intended to capture trend and monthly seasonality. In addition, before running the model, we partition the data into training and test sets, with the last 24 months (2 years) as our test set, and the rest as our training set.

In Figures 5 and 6 we display the forecast error box plots for reconciled and unreconciled forecasts using all three methods, for the rolling origin and fixed origin scenarios, respectively.

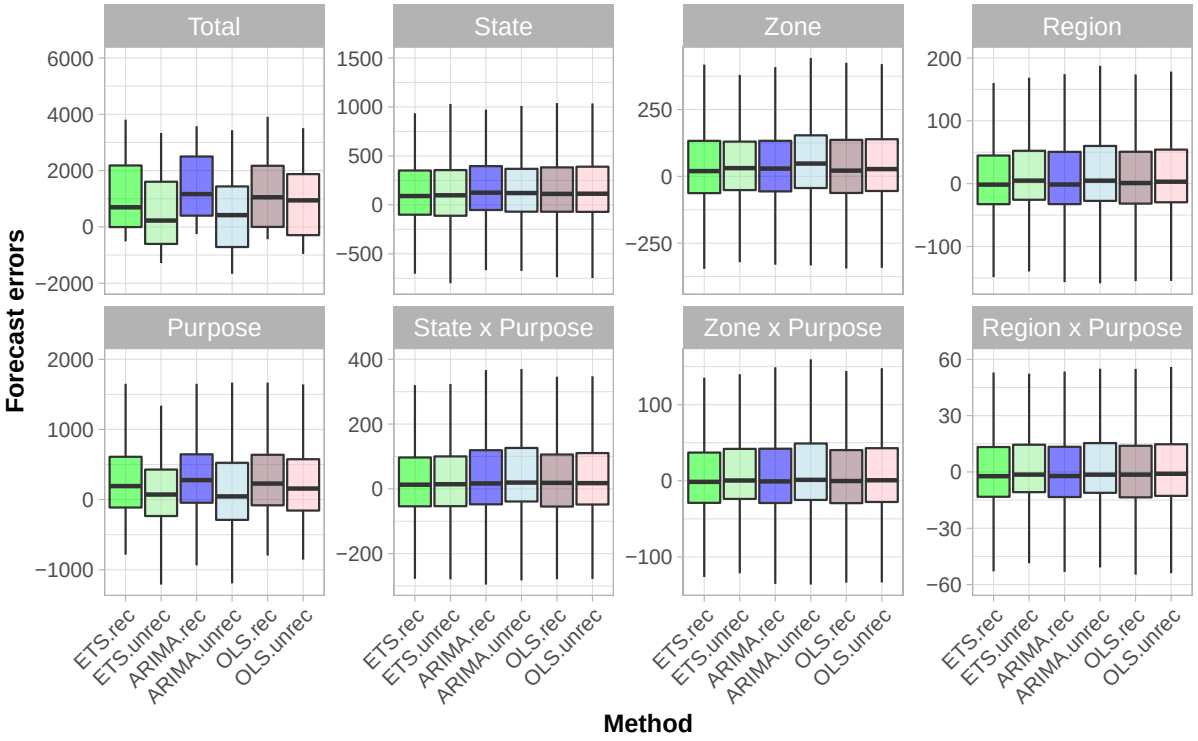

Figure 5: *Box plots of rolling origin forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level.*

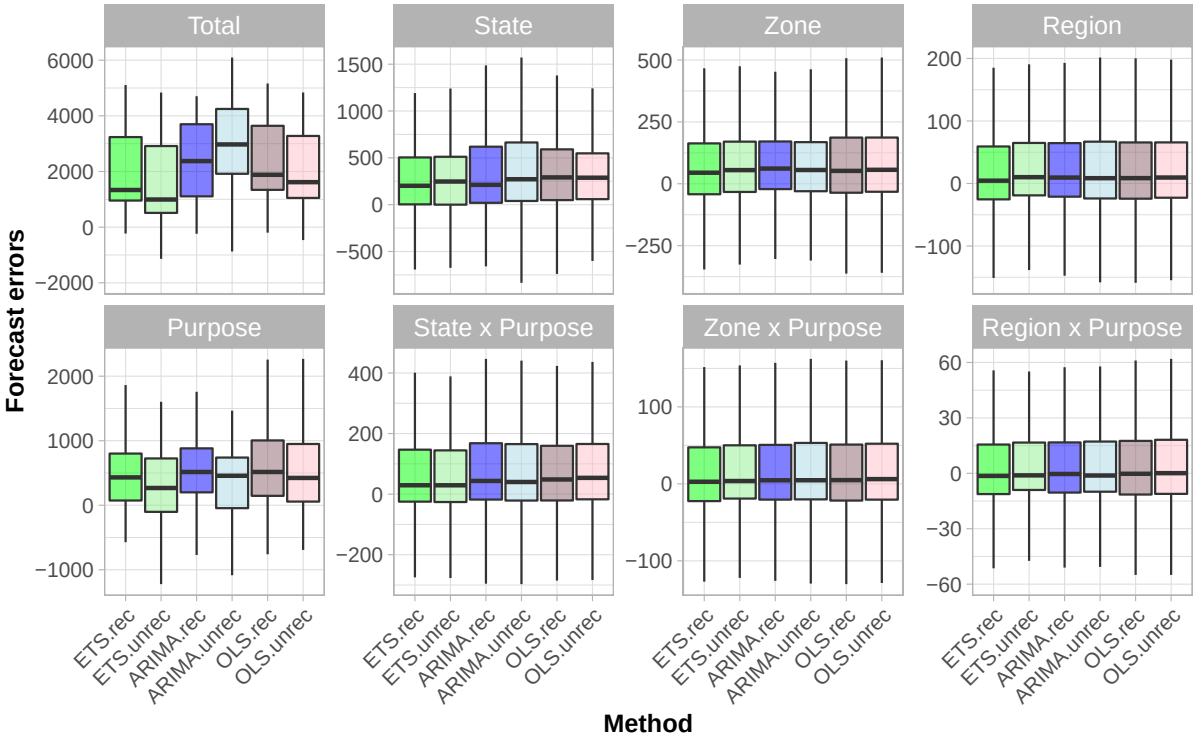

Figure 6: *Box plots of fixed origin forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level.*

Table 3: *Mean(RMSE) on 24 months test set for ETS, ARIMA and OLS with and without reconciliation - Rolling origin.*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	1516	1504	1634	1733	1840	1864
State	511	501	498	497	482	509
Zone	215	217	213	211	210	213
Region	123	125	117	118	120	117
Purpose	676	674	682	673	713	713
State x Purpose	213	217	213	208	209	213
Zone x Purpose	98	103	98	96	99	97
Region x Purpose	56	58	56	56	57	56

Table 4: *Mean(RMSE) on 24 months test set for ETS, ARIMA and OLS with and without reconciliation - Fixed origin.*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	2239	3433	2529	2492	2744	2819
State	594	610	597	573	583	612
Zone	240	230	243	237	234	243
Region	133	132	127	127	127	126
Purpose	767	829	876	822	889	921
State x Purpose	227	233	237	222	226	236
Zone x Purpose	103	106	105	102	102	104
Region x Purpose	59	59	59	58	58	58

The box plots allow comparing the error distributions across the different models. We see that all methods behave similarly, and without bias, for all levels except Total and Purpose. For Total and Purpose, all methods are biased in the same direction (under-forecasting). In both cases, for OLS reconciliation has the least effect.

Note that because higher-level series have higher counts, their errors are larger in magnitude[6], Finally, we see that (as expected) for rolling origin 1-step-ahead forecasts, the error densities are closer and more tightly distributed around zero than the fixed origin multi-step-ahead forecasts.

We also compare forecast accuracy by examining time plots. Figures 7 and 8 show the rolling and fixed origin forecast results for the total series and one of the bottom-level series, 'AAAVis' (Sydney - Business). Comparing the reconciled (dashed lines) and unreconciled (dotted lines) forecasts, we see that the reconciliation step improves the forecasts in this series. We also see the OLS model's forecast accuracy is similar to that of the other methods.

[6] Appendix A.2 shows the box plots with scaled errors, to better compare errors across all the hierarchy levels. Scaled errors are computed by subtracting the mean and dividing by the standard deviation.

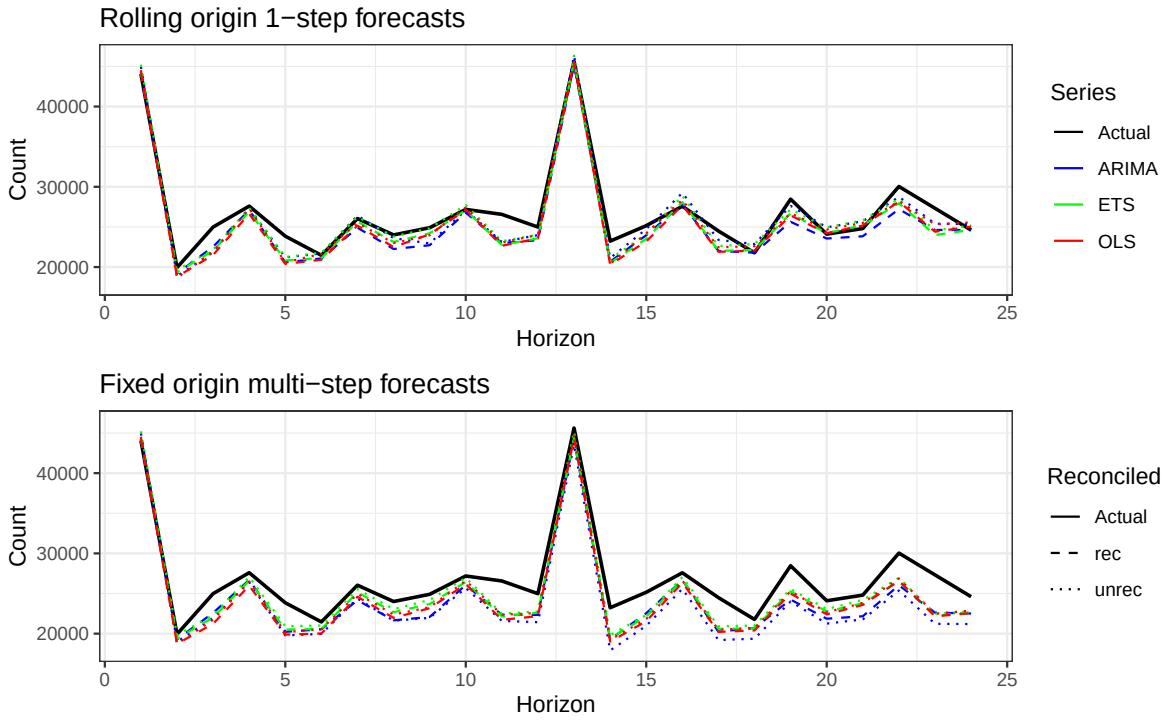

Figure 7: *Comparing ETS, ARIMA and OLS forecasts (reconciled and unreconciled) for 'Total' series. (Top: rolling origin, Bottom: fixed origin).*

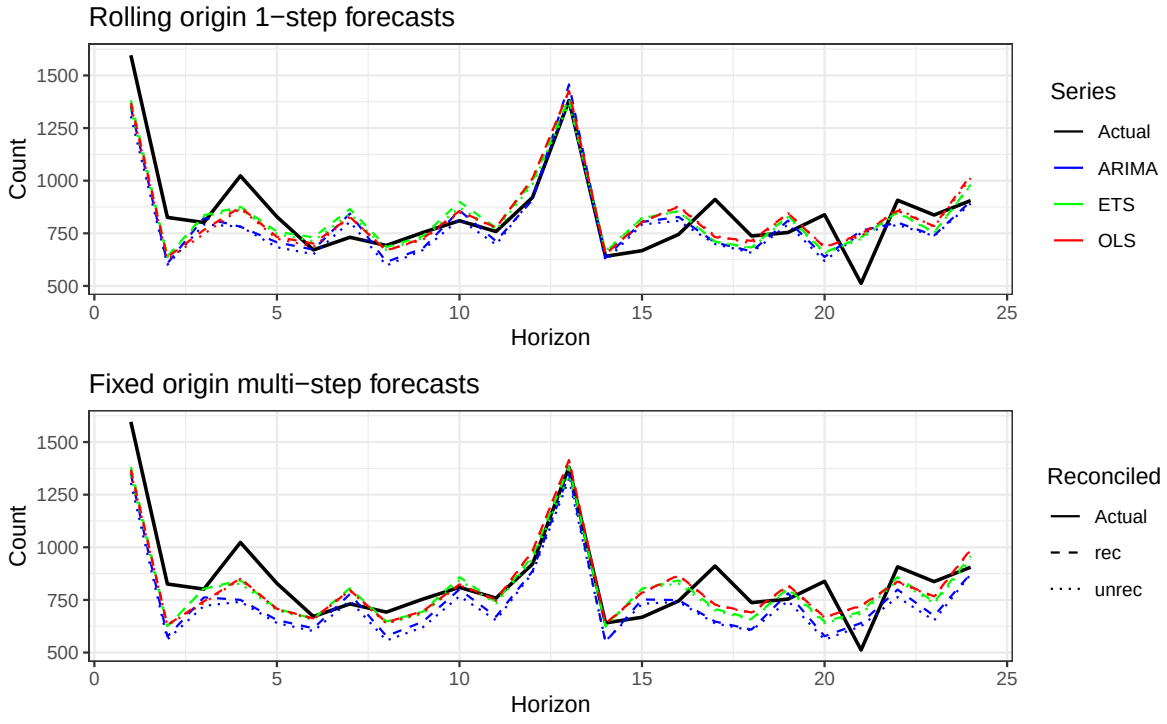

Figure 8: *Comparing ETS, ARIMA and OLS forecasts (reconciled and unreconciled) for 'AAAVis' bottom-level series. (Top: rolling origin, Bottom: fixed origin).*

Next, we expand the comparison to include prediction intervals. We display the prediction intervals (based on reconciled forecasts) for the total series (Figure 9) and for the bottom-level series 'AAAVis' (Sydney - Visiting) (Figure 10).We again see that OLS is similar to the two other methods[7].

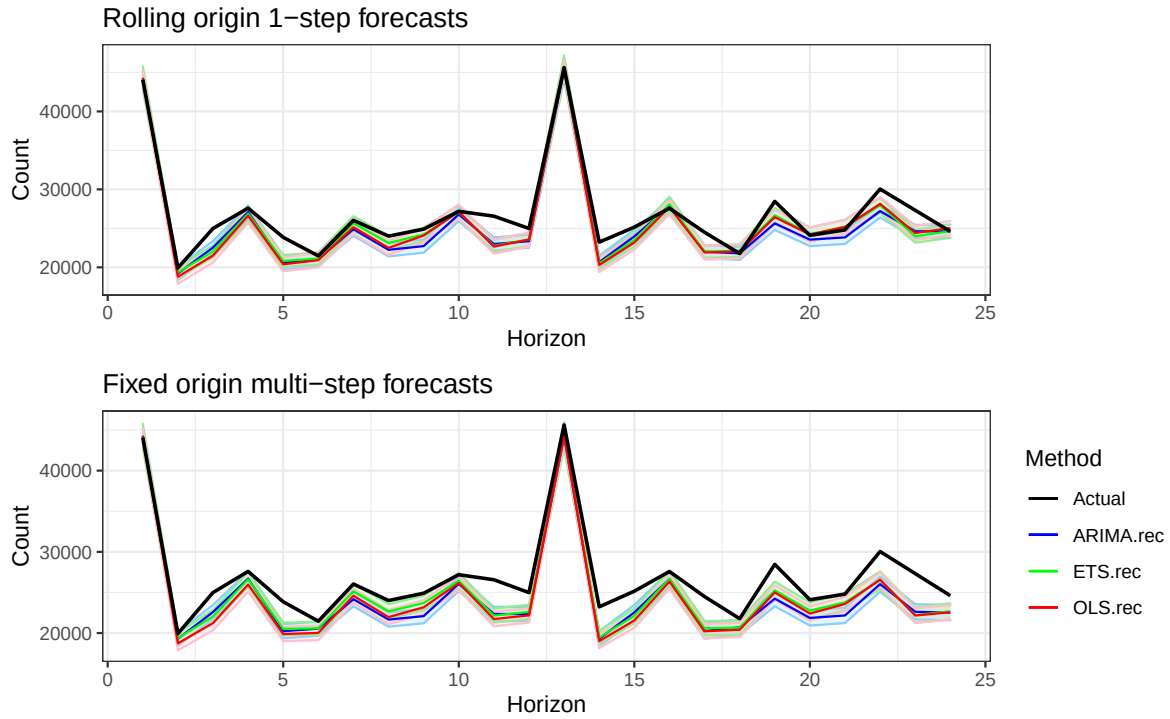

Figure 9: *Comparing ETS, ARIMA and OLS reconciled forecasts and prediction intervals for 'Total' series. (Top: rolling origin, Bottom: fixed origin).*

Table 5 compares the computation time of the three methods for rolling and fixed origin forecasting. We see that the OLS model is much faster compared to the other methods. Also, since reconciliation is a linear process, in all methods it is very fast and does not affect computation time significantly.

Since we are using a linear model, we can easily include exogenous variables which can often be helpful in improving forecast accuracy. In this application, we tried including an 'Easter' dummy variable indicating the timing of Easter. However, we found its effect on forecast accuracy was minimal, and therefore omit it in the model reported here.

Finally, Table 6 shows that, as mentioned in Section 2.3, computation is faster using separate regression models compared to the matrix approach (even using sparse matrix algebra).

In summary, for this dataset the OLS method performs similar to ETS and ARIMA in terms of forecast accuracy while being computationally much cheaper. The reason is that the OLS

[7]We compare the reconciled and unreconciled prediction intervals in Appendix A.3.

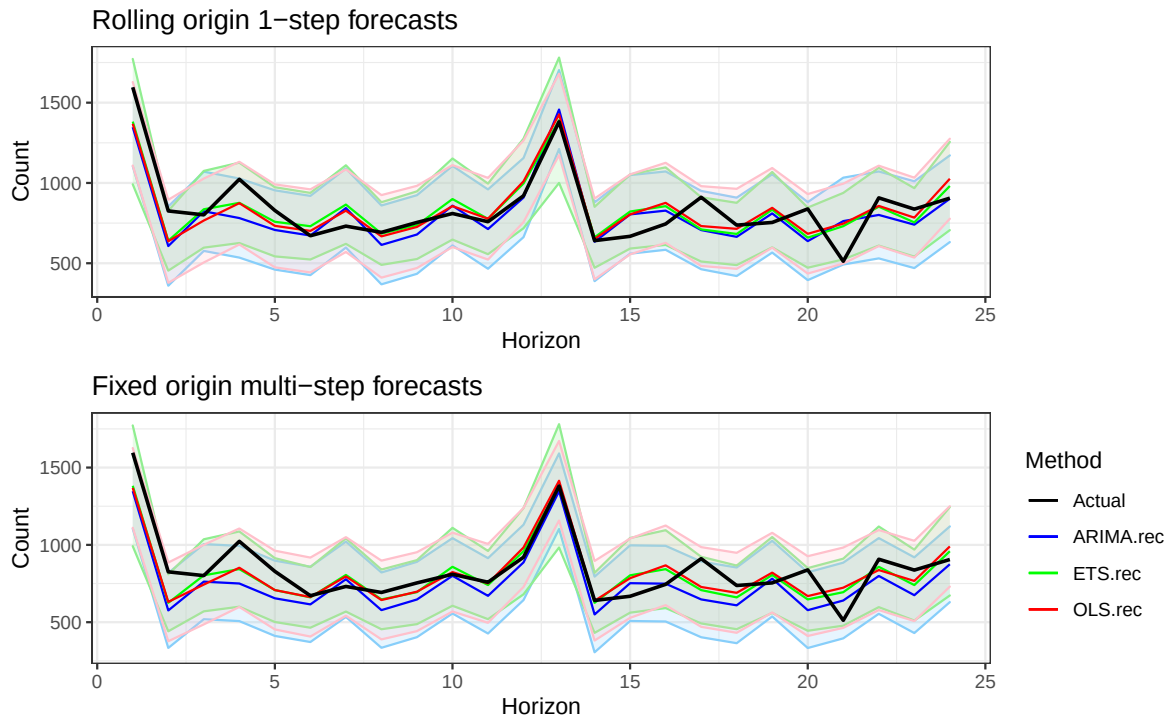

Figure 10: *Comparing ETS, ARIMA and OLS reconciled forecasts and prediction intervals for 'AAAVis' bottom-level series. (Top: rolling origin, Bottom: fixed origin).*

Table 5: *Computation time (seconds) for OLS using the matrix approach and separate regression models, with reconciliation, for rolling and fixed origin, on a 24 months test set.*

	Rolling origin	Fixed origin
ETS	14648	618
ARIMA	30346	1085
OLS	48	18

Table 6: *Computation time (seconds) for OLS using the matrix approach and separate regression models, with reconciliation, for rolling and fixed origin, on a 24 months test set.*

	Rolling origin	Fixed origin
Matrix approach	210	106
Separate models	48	18

model only involves linear operations whereas ETS and ARIMA require nonlinear optimization methods.

3.2 Australian domestic tourism simulation study

We provide results from two simulation studies based on the Australian domestic tourism dataset, to evaluate the sensitivity of our results to several factors: forecast horizon, noise level, and hierarchy/grouping structure.

3.2.0.1 Effect of Forecast Horizon and Noise Level: In the first study, we simulate bottom-level series similar to the real bottom-level series of the tourism data, with the same number of series and the same length. We then generate forecasts for four forecast horizons (12, 24, 36 and 48 months) with four different noise levels (standard deviation = 0.01, 0.1, 0.5 and 1)[8]. For the OLS approach we use the same set of predictors as in Section 3.1.

Table 7 and 8 display the average RMSE for 12 to 48 month-ahead forecasts with different noise levels. Results are shown for reconciled forecasts for both rolling and fixed origin approaches. We see that, as expected, increasing the forecast horizon and/or noise level increases the average RMSE in all the three methods. We also see that the proposed OLS approach performs similarly to ETS and ARIMA.

Table 7: *Mean RMSE of rolling origin forecasts for simulated data (304 bottom-level series and 8 levels of hierarchy), for different error levels, methods (ETS, ARIMA, OLS), forecast horizons, with reconciliation.*

Reconciliation	Error	Forecast horizon	ETS	ARIMA	OLS
rec	0.01	12	123.8	119.8	130.4
rec	0.01	24	122.6	119.4	128.8
rec	0.01	36	124.6	122.3	131.8
rec	0.01	48	122.1	120.4	129.0
rec	0.10	12	123.7	120.1	129.7
rec	0.10	24	122.9	120.3	129.2
rec	0.10	36	125.8	123.7	133.1
rec	0.10	48	123.5	122.1	130.5
rec	0.50	12	143.9	143.2	146.9
rec	0.50	24	149.9	146.3	153.2
rec	0.50	36	155.8	151.5	160.0
rec	0.50	48	154.5	150.8	159.7
rec	1.00	12	192.9	198.1	193.9
rec	1.00	24	207.1	209.0	209.3
rec	1.00	36	215.7	215.4	218.0
rec	1.00	48	218.4	217.5	220.9

This similarity between OLS and the other methods in terms of forecast accuracy can also be seen when looking at time plots of individual series.} For example, Figures 11 and 12 display one of the bottom-level series with 12 to 48 months ahead reconciled forecasts and prediction intervals, while changing the noise levels. In these two figures we see that the OLS prediction intervals are almost identical to those from ARIMA and ETS.

[8]Since the levels of the series are different, we first scale the simulated series (subtracting the mean and dividing by standard deviation), add the white noise series, and then rescale the series.

Table 8: *Mean RMSE of fixed origin forecasts for simulated data (304 bottom-level series and 8 levels of hierarchy), for different error levels, methods (ETS, ARIMA, OLS), forecast horizons, with reconciliation.*

Reconciliation	Error	Forecast horizon	ETS	ARIMA	OLS
rec	0.01	12	123.6	119.8	131.4
rec	0.01	24	126.5	124.2	137.3
rec	0.01	36	133.4	131.9	148.6
rec	0.01	48	133.5	133.9	152.3
rec	0.10	12	124.8	120.3	130.7
rec	0.10	24	128.0	124.5	137.2
rec	0.10	36	135.3	132.2	148.9
rec	0.10	48	135.5	134.3	152.6
rec	0.50	12	144.6	143.4	147.8
rec	0.50	24	154.5	151.6	158.2
rec	0.50	36	162.8	160.7	170.0
rec	0.50	48	164.1	163.5	174.3
rec	1.00	12	194.1	197.7	194.7
rec	1.00	24	212.4	213.3	213.0
rec	1.00	36	221.9	222.1	224.8
rec	1.00	48	225.8	227.1	231.0

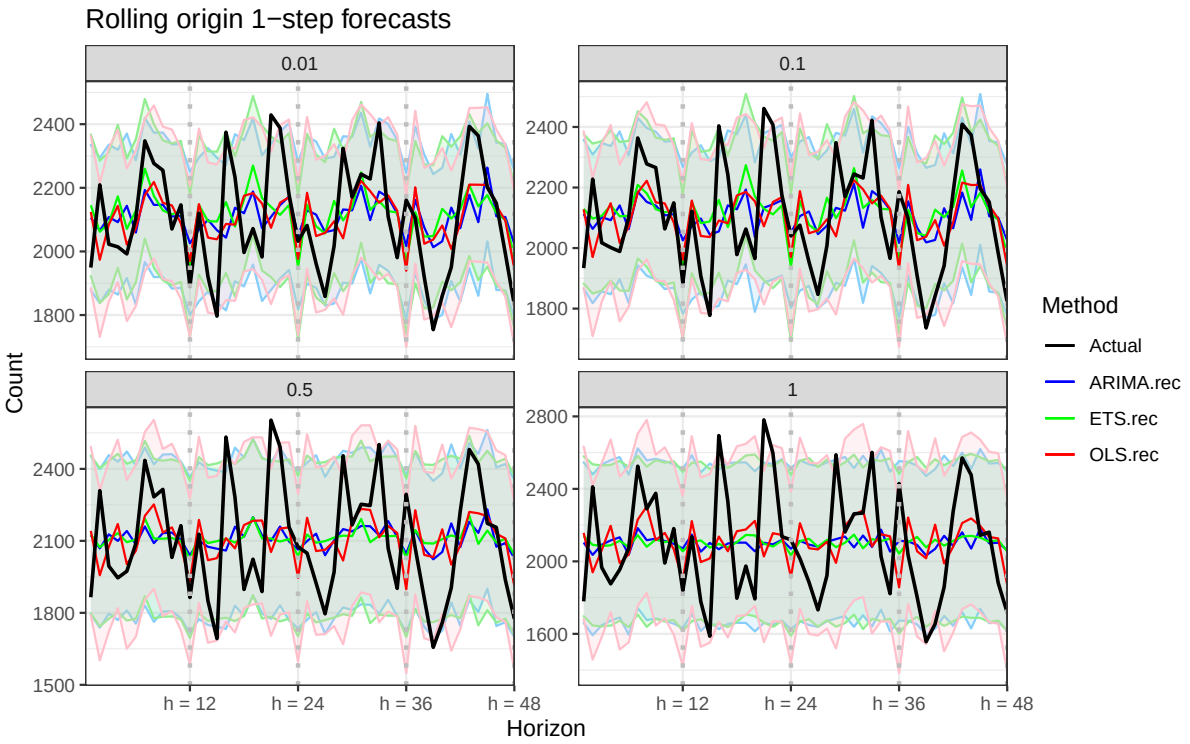

Figure 11: *Comparing reconciled 'rolling origin' forecasts and prediction intervals for a sample bottom-level series across different error levels (different panels).*

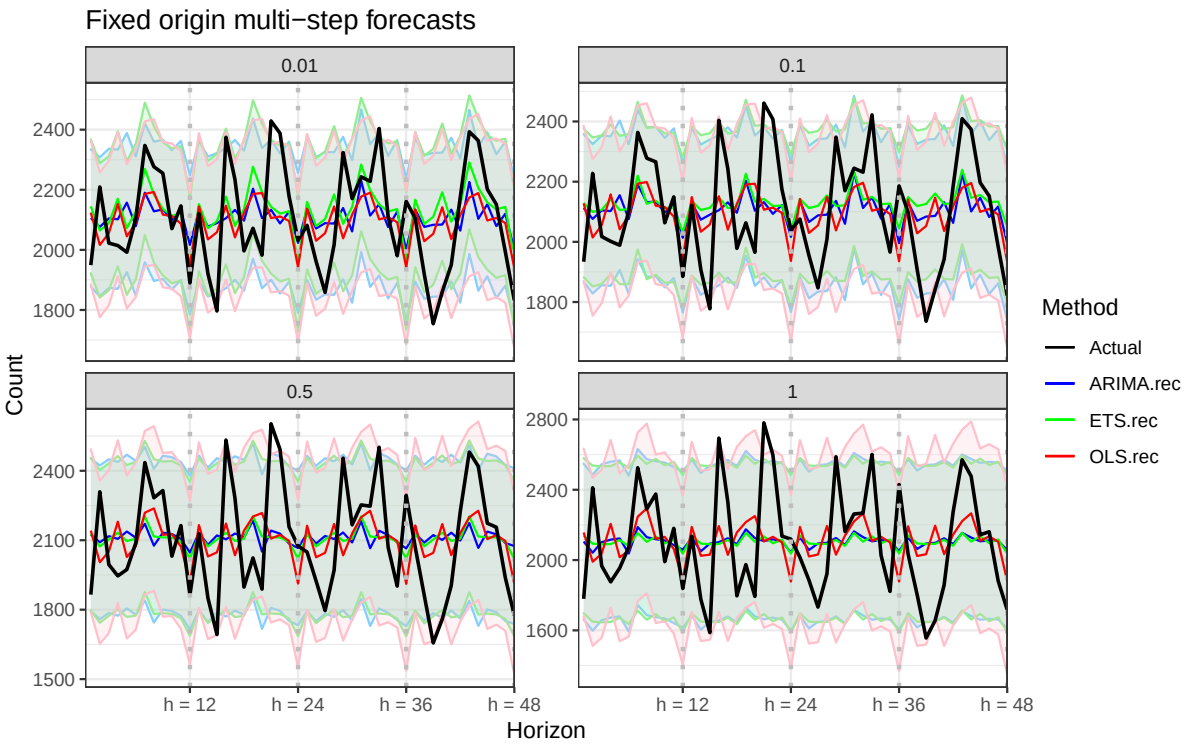

Figure 12: *Comparing reconciled 'fixed origin' forecasts and prediction intervals for a sample bottom-level series across different error levels (different panels).*

Finally, Table 9 compares the computation time (seconds) for the different methods[9]. We see that increasing the forecast horizon from 12 to 48 months increases computation time almost linearly, and that the computation time for ARIMA and ETS is significantly longer than OLS.

Table 9: *Computation time (seconds) for rolling and fixed origin reconciled forecasts using ETS, ARIMA and OLS, by horizon and for 0.5 error value, for 304 bottom-level series and 8 levels of hierarchy.*

	Rolling origin			Fixed origin		
Forecast Horizon	ETS	ARIMA	OLS	ETS	ARIMA	OLS
12	5188	3379	19	443	293	8
24	10429	7029	37	449	291	14
36	15724	10884	53	454	305	18
48	21093	14631	69	460	308	24

3.2.0.2 Effect of Hierarchy/Grouping Structure: In the second simulation study, we fix the forecast horizon at $h = 24$ and the noise level at 0.5, and then create four different aggregation levels (8, 10, 12 and 18). The aggregation levels were obtained based on the original hierarchy structure of the Australian tourism example (Table 2). The new structures include the original structure and three new structures: The original structure plus one additional three-level

[9]The table reports the computation time only for the 0.5 noise level since changing the noise level does not affect computation time; Also, since the reconciliation step takes less than a second, computation times are practically identical for reconciled and unreconciled forecasts.

hierarchy factor, the original structure plus two five-level additional hierarchy factors, and the original structure plus two hierarchy factors and one grouping factor. Table 10 displays these four structures and the number of series in each level.

Table 10: *Four simulated hierarchy/grouping structures.*

	Number of series			
Aggregation Level	Original structure	Structure 1	Structure 2	Structure 3
Australia	1	1	1	1
Level 1	-	3	3	3
Level 2	-	-	5	5
State	7	7	7	7
Zone	27	27	27	27
Region	76	76	76	76
Purpose	4	4	4	4
Group 1	-	-	-	5
Level 1 x Purpose	-	12	12	12
Level 2 x Purpose	-	-	20	20
State x Purpose	28	28	28	28
Zone x Purpose	108	108	108	108
Level 1 x Group 1	-	-	-	5
Level 2 x Group 1	-	-	-	6
State x Group 1	-	-	-	7
Zone x Group 1	-	-	-	27
Purpose x Group 1	-	-	-	20
Bottom level	304	304	304	304
Total	555	570	595	665

We also simulated four sets of bottom-level series with different numbers of series (304, 608, 1520 and 3040). In order to add series, we changed the number of 'Purpose' categories (grouping factor) in the Australian domestic tourism example. Table 11 displays the total number of series based on 304, 608, 1520 and 3040 bottom-level series with 8, 10, 12 and 18 hierarchy levels.

Table 11: *Total number of series in hierarchy structure based on the number of bottom-level series and aggregation levels.*

	No. of aggregation levels			
# Bottom-level series	8	10	12	18
304	555	570	595	665
608	999	1026	1071	1161
1520	2331	2394	2499	2649
3040	4551	4674	2643	5129

Tables 12 and 13 display the average RMSEs by number of bottom-level series and number of aggregation levels for the different forecasting approaches. We see that increasing the number

of bottom-level series and/or the number of aggregation levels increases the mean RMSE, and this similarly affects all forecasting methods, in both rolling origin and fixed origin scenarios.

We verify this further by examining time plots and prediction intervals for one of the bottom-level series. Figures 13 and 14 compare four aggregation levels (in different panels). We see that all three approaches indeed have similar prediction intervals across aggregation levels, for both the rolling origin and fixed origin forecasting scenarios.

Table 12: *Mean RMSE by number of hierarchy levels, number of bottom-level series, method, with reconciliation. Simulated series has error value 0.5. Forecasting uses rolling origin for a 24-month horizon.*

Reconciliation	Series	Levels	ETS	ARIMA	OLS
rec	304	8	149.9	146.3	153.2
rec	304	10	164.6	160.5	168.1
rec	304	12	176.6	172.3	180.2
rec	304	18	204.6	208.4	207.1
rec	608	8	786.6	806.2	822.5
rec	608	10	879.5	904.1	921.0
rec	608	12	937.6	966.8	982.5
rec	608	18	1070.4	1105.7	1122.8
rec	1520	8	1075.3	1182.3	1059.6
rec	1520	10	1202.9	1329.3	1188.2
rec	1520	12	1284.4	1422.7	1268.9
rec	1520	18	1472.9	1651.1	1459.3
rec	3040	8	1156.4	1257.7	1155.1
rec	3040	10	1296.4	1416.6	1295.4
rec	3040	12	1403.0	1518.7	1383.5
rec	3040	18	1599.8	1762.6	1601.3

Finally, we compare the computation time of the different methods across the different number of aggregation levels and number of bottom-level series (Table 14). We see that increasing the number of bottom-level series increases computation time linearly and increasing the number of aggregation levels only slightly increases computation time. For both rolling and fixed origin scenarios, we see a substantial difference in computation time between ARIMA and ETS compared with our OLS approach[10].

In summary, these two simulation studies show that our result – that OLS performs much faster but with close accuracy compared with ETS and ARIMA – is robust to changes in forecast horizon, noise levels, and hierarchy/grouping structure. We see that computation time is not affected by the noise level while it linearly increases in the forecast horizon. Also, changing the number of aggregation levels in the hierarchy/grouping structure only slightly increases the

[10]The reconciliation step computation time varies from less than one second to around three seconds, as a function of the number of series. Computation times displayed in the table are for unreconciled forecasts.

Table 13: *Mean RMSE by number of hierarchy levels, number of bottom-level series, method, with reconciliation. Simulated series has error value 0.5. Forecasting uses fixed origin for a 24-month horizon.*

Reconciliation	Series	Levels	ETS	ARIMA	OLS
rec	304	8	154.5	151.6	158.2
rec	304	10	169.6	166.4	174.1
rec	304	12	181.8	178.2	186.4
rec	304	18	209.8	205.0	213.4
rec	608	8	786.0	809.0	802.7
rec	608	10	880.5	907.5	898.6
rec	608	12	939.4	970.4	958.6
rec	608	18	1070.9	1107.3	1095.3
rec	1520	8	1077.2	1181.9	1039.1
rec	1520	10	1208.9	1327.8	1165.4
rec	1520	12	1291.0	1421.5	1244.8
rec	1520	18	1486.4	1643.6	1430.8
rec	3040	8	1148.8	1263.0	1134.8
rec	3040	10	1288.3	1420.5	1272.7
rec	3040	12	1376.2	1522.0	1359.3
rec	3040	18	1587.6	1764.4	1573.6

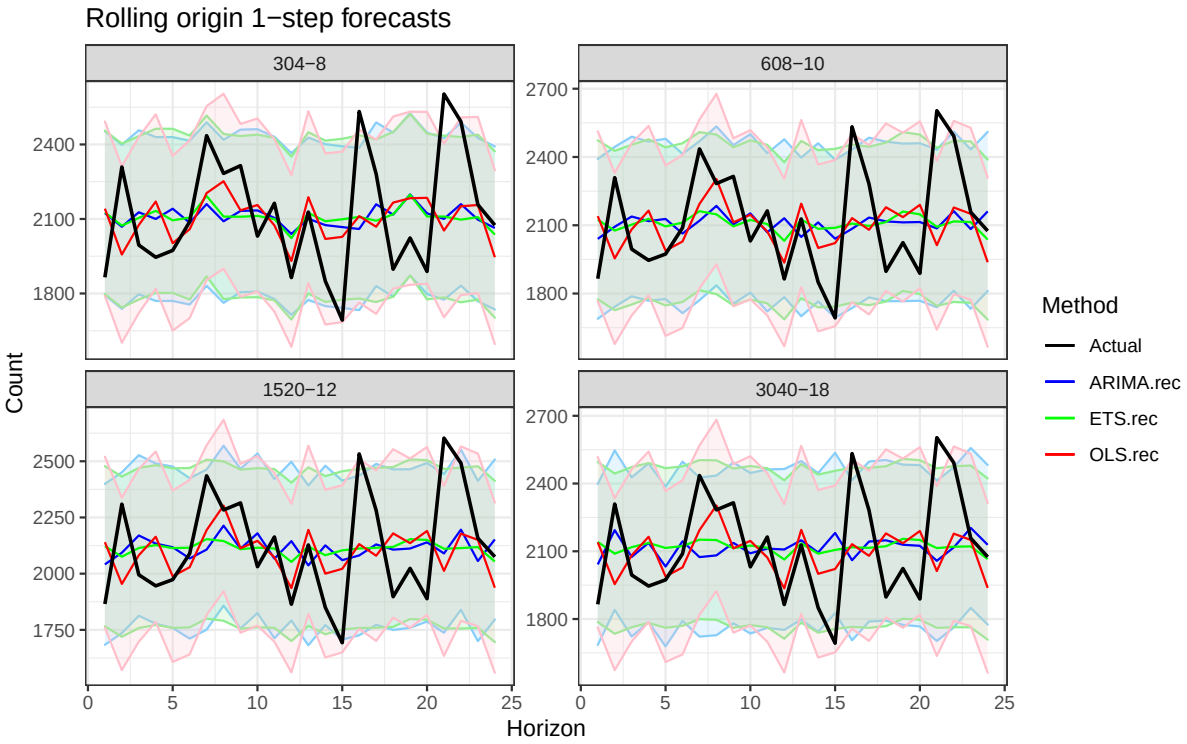

Figure 13: *Comparing reconciled rolling origin forecasts and prediction intervals for a sample bottom-level series, for different number of bottom-level series and hierarchy levels (different panels). Simulated series has error value 0.5 and 24 months test set.*

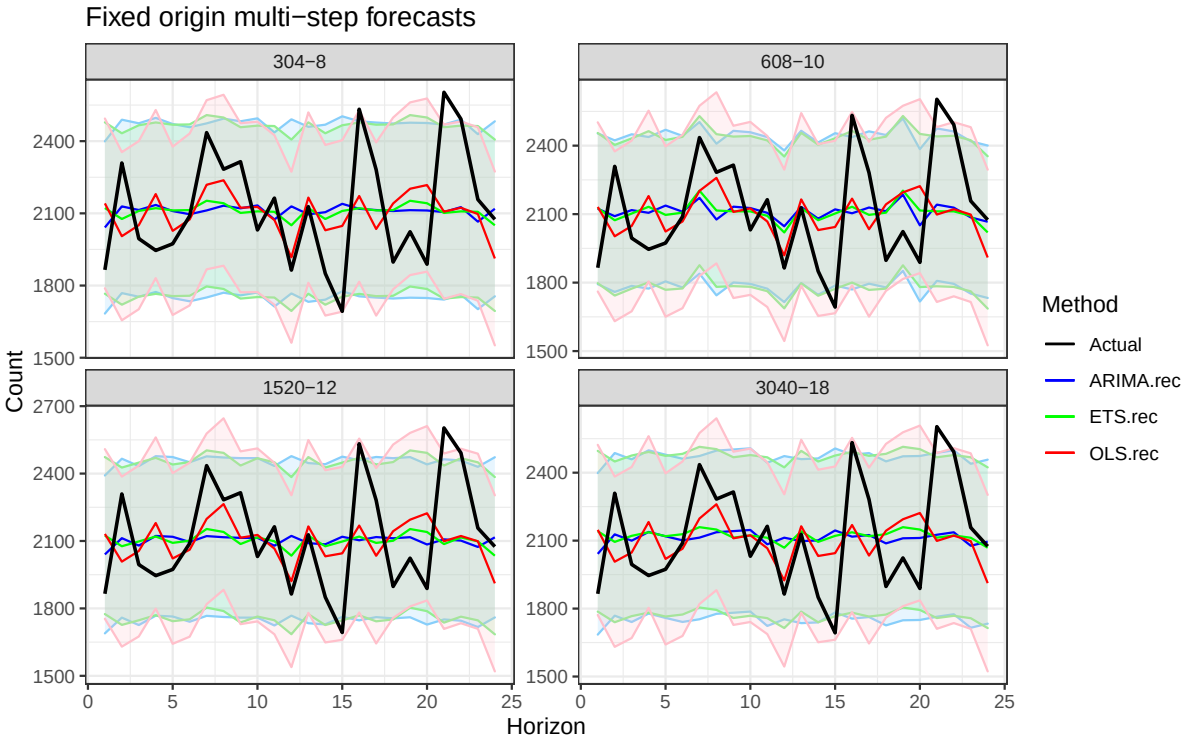

Figure 14: *Comparing reconciled fixed origin forecasts and prediction intervals for a sample bottom-level series, for different number of bottom-level series and hierarchy levels (different panels). Simulated series has error value 0.5 and 24 months test set.*

Table 14: *Computation time (seconds) for rolling and fixed origin reconciled forecasts using ETS, ARIMA and OLS, by number of bottom-level series (x-axis), and by levels of hierarchy (panels). Simulated series has error value 0.5 and 24 months test set.*

		Rolling origin			Fixed origin		
Series	Levels	ETS	ARIMA	OLS	ETS	ARIMA	OLS
304	8	10490	7079	36	19	290	19
	10	10468	7092	36	20	299	20
	12	11618	7559	38	20	323	20
	18	13737	8321	44	21	356	21
608	8	15049	10074	63	35	719	35
	10	15538	10403	64	35	760	35
	12	16322	10996	67	36	794	36
	18	17891	12133	73	41	836	41
1520	8	35323	24661	200	79	1935	79
	10	36202	25564	210	81	2037	81
	12	37584	26770	194	85	2186	85
	18	41458	28975	226	88	2414	88
3040	8	73406	54080	271	159	7691	159
	10	72505	58184	297	160	8081	160
	12	80636	61646	309	165	9247	165
	18	86112	64088	326	212	10437	212

computation time while increasing the number of bottom-level series increases computation time linearly.

4 Conclusion

We have proposed a linear model approach to fast forecasting of hierarchical or grouped time series, with accuracy that nearly matches that of forecast methods such as ETS and ARIMA. This is especially useful in large collections of time series, as is typical in hierarchical and grouped structures. Although ETS and ARIMA are advantageous in terms of forecasting power and accuracy, they can be computationally heavy when facing large collections of time series in the hierarchy.

An important feature of our proposed OLS model is its ability to easily include external information such as holiday dummies or other external series.We also note that OLS has the additional practical advantage of handling missing data while ETS and ARIMA require imputation. Another advantage of the OLS approach is its model selection flexibility, where computationally cheap criteria (e.g. AIC) and approaches (e.g. leave-one-out-cross-validation, LOOCV) can be used for choosing a best subset of predictors. We leave the study of missing data and model selection for future research.

Pennings & Dalen (2017) proposed another approach for forecasting hierarchical time series using state space models. Although their approach is flexible in handling outliers, missing data and external features, it is less flexible across different kinds of datasets and is computationally much more demanding.

Another advantage of our approach is that it can be computed in a single matrix equation (Equation (5)). This makes it extremely fast and easy to implement, and enables standard results to be derived with minimal effort (e.g., prediction intervals).

In this paper, we computed prediction intervals based on a normality assumption for all the real and simulated examples. One future direction for potentially improving prediction intervals is applying bootstrap-based methods which only assume uncorrelated forecast errors.

Acknowledgements

Ashouri and Shmueli were partially funded by Ministry of Science and Technology (MOST), Taiwan [Grant 106-2420-H-007-019]. Hyndman's research is supported by the Australian Center of Excellence in Mathematical and Statistical Frontiers.

Supplementary material

The datasets and R codes used in this paper are publicly available at github.com/robjhyndman/linear-hierarchical-forecasting. We have also created a Binder interface for our Australian tourism and Wikipedia examples which allows easily running our code in a browser. The demo folder is reachable from github.com/mahsaashouri/AUS-Wiki-Binder.

A Further analyses for tourism demand forecasting

A.1 Different reconciliation matrices

We present reconciliation results using two other reconciliation matrices introduced in (Wickramasuriya, Athanasopoulos & Hyndman 2019): shrinkage estimator (`mint_shrink`) and variance scaling (`wls_var`). These results are presented for both rolling origin (Table 15) and fixed origin forecasts (Table 16). We see that in this case the shrinkage estimator provides the best reconciled forecasts.

Table 15: *Comparing Mean(RMSE) of three different reconciliation matrices for rolling origin forecasts on a 24 months test set.*

	mint_shrink			wls_var			wls_struct		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	1600	1543	1834	1705	1853	1910	1628	1680	1753
State	483	448	500	489	478	511	504	492	503
Zone	208	202	209	210	209	213	213	214	213
Region	117	116	115	117	119	116	120	122	117
Purpose	662	645	703	675	725	720	675	694	698
State x Purpose	207	203	209	208	209	213	210	213	213
Zone x Purpose	96	97	96	97	99	97	97	101	98
Region x Purpose	56	57	55	56	57	55	56	58	56

Table 16: *Comparing Mean(RMSE) of three different reconciliation matrices for fixed origin forecasts on a 24 months test set.*

	mint_shrink			wls_var			wls_struct		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	2444	2721	2506	2541	2738	2880	2369	3108	2678
State	567	580	569	577	585	618	583	597	604
Zone	232	233	228	235	236	244	238	232	243
Region	125	126	120	125	127	126	130	129	127
Purpose	819	868	849	836	898	936	795	859	898
State x Purpose	221	224	224	223	226	238	225	229	236
Zone x Purpose	101	102	100	101	103	104	103	104	105
Region x Purpose	58	58	56	58	58	58	59	59	59

A.2 Scaled forecast errors boxplots

For further comparison of the forecast errors for the tourism example across different aggregation levels, we plot the *scaled errors* for rolling forward forecasts (Figures 15) and fixed origin forecasts (Figure 16). Scaled errors are computed by subtracting the mean and dividing by the standard deviation. We see that the scaled error magnitudes are similar at different hierarchy/grouping levels.

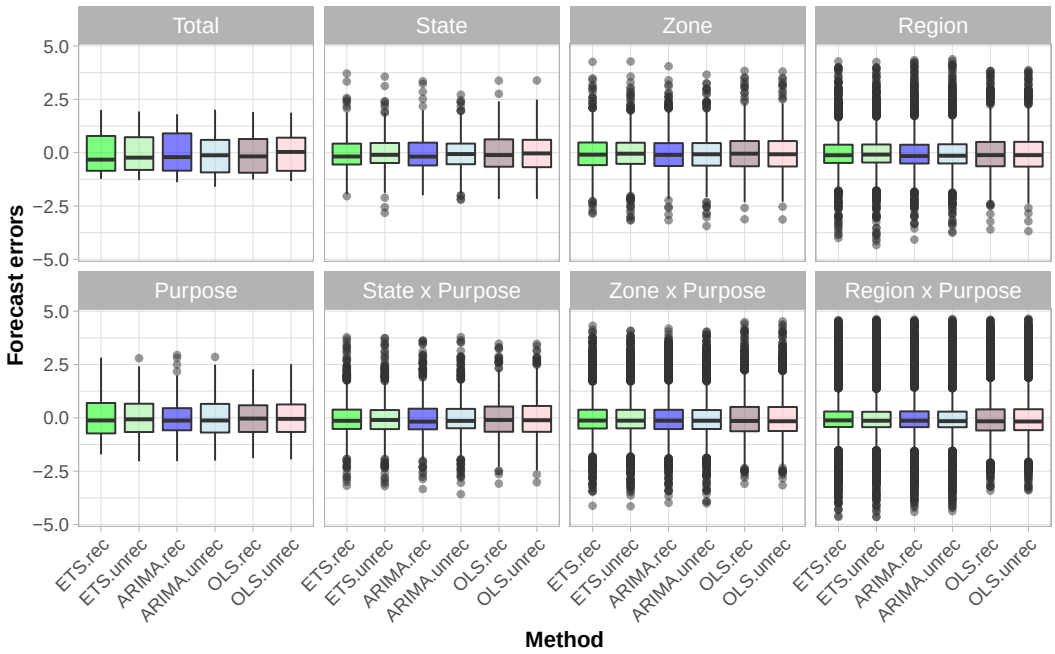

Figure 15: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA, and OLS rolling origin forecasts for tourism demand data. Panels are different hierarchy levels.*

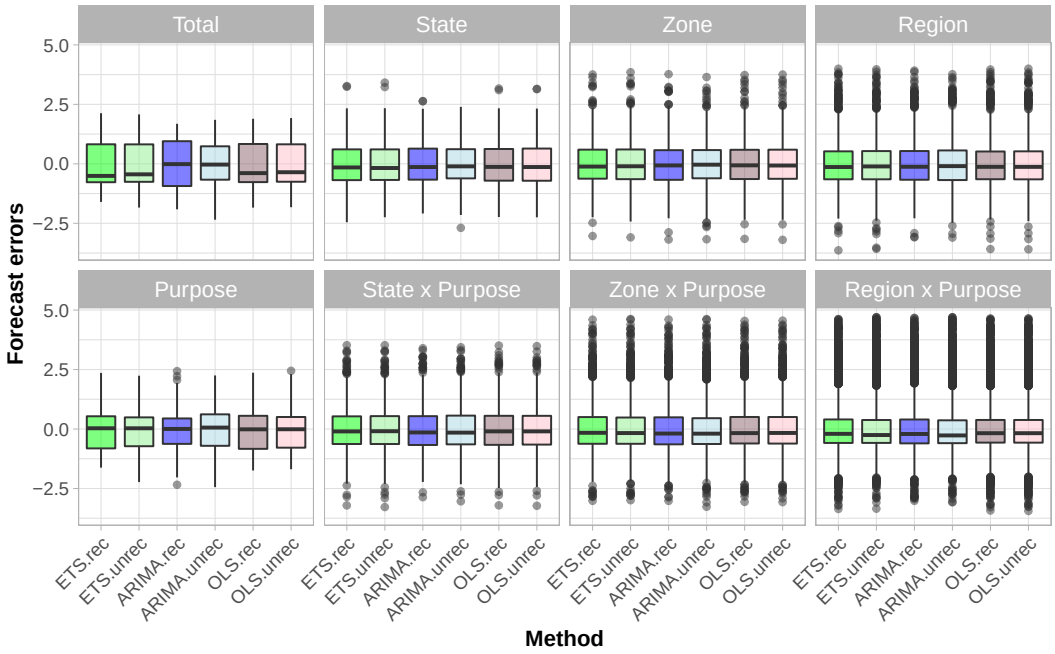

Figure 16: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA, and OLS fixed origin forecasts for tourism demand data. Panels are different hierarchy levels.*

A.3 Reconciled and unreconciled prediction intervals comparison

Figures 15 and 16 display boxplots of the scaled forecast errors for the tourism example. These plots are displayed for both rolling forward and multiple-step-ahead forecasts, respectively. We see that scaled error magnitudes are similar at different hierarchy/grouping levels.

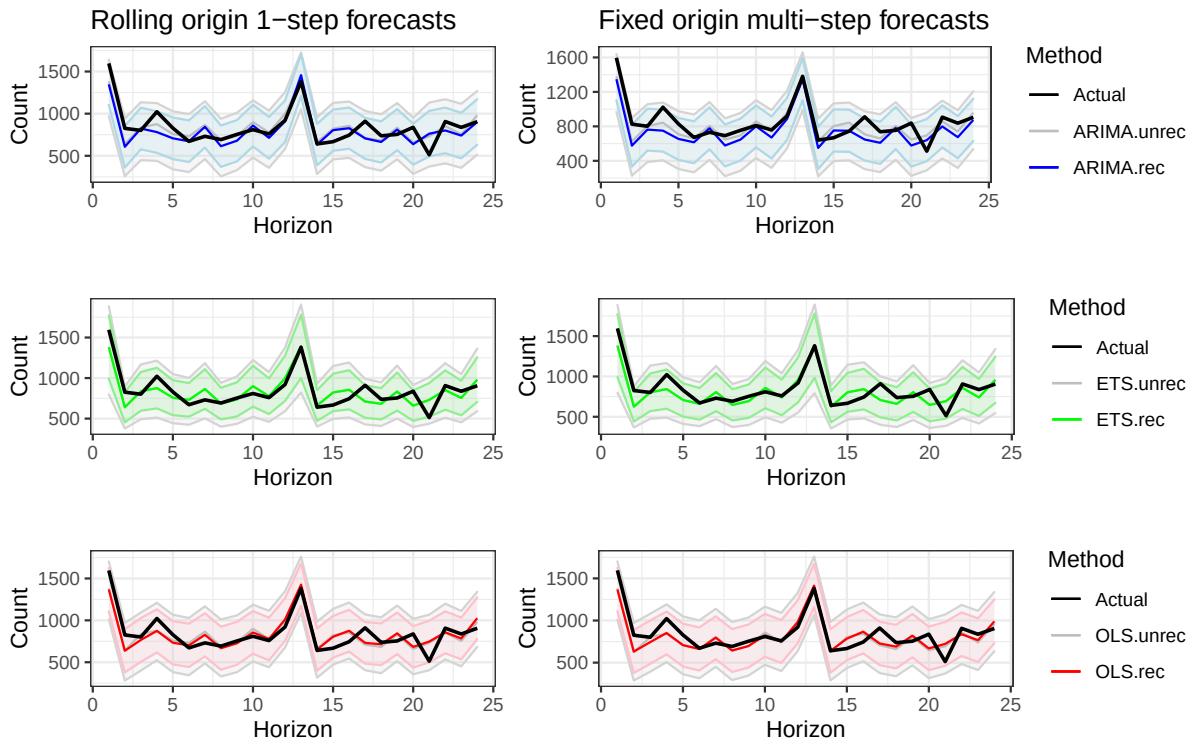

Figure 17: *Comparing forecasts and prediction intervals of 'AAAVis' bottom-level series across methods. Left: rolling origin. Right: fixed origin.*

B Wikipedia pageviews: Grouped structure

We illustrate our method on another real dataset. The Wikipedia dataset includes 12 months of daily data (2016-06-01 to 2017-06-29) on Wikipedia pageviews for the most popular social networks articles (Ashouri, Shmueli & Sin 2018). This dataset is noisier than the Australian monthly tourism data, making forecasting more challenging. The data has a grouped structure with the following attributes (see Table 17):

- *Agent*: Spider, User;
- *Access*: Desktop, Mobile app, Mobile web;
- *Language*: en (English), de (German), es (Spanish), zh (Chinese); and
- *Purpose*: Blogging related, Business, Gaming, General purpose, Life style, Photo sharing, Reunion, Travel, Video.

Figure 18 hows one possible hierarchy for this dataset, but the order of the hierarchy can be switched.

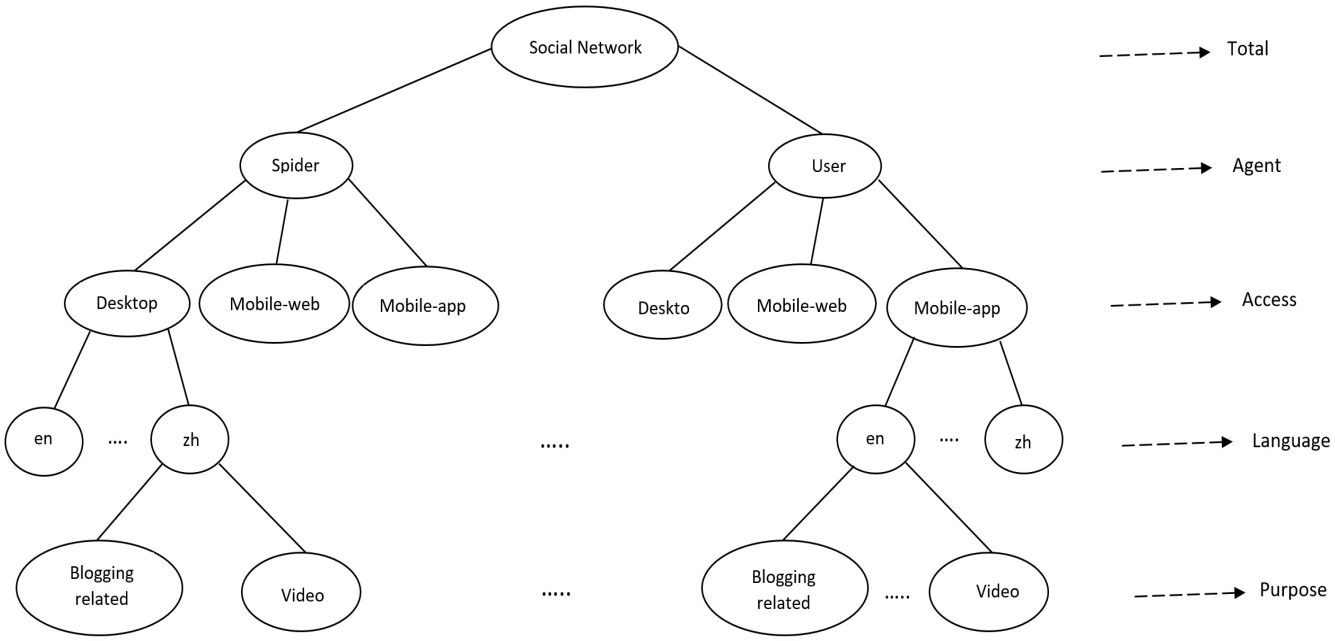

Figure 18: *One possible hierarchical structure for the Wikipedia pageviews dataset.*

Table 17: *Social networking Wikipedia article grouping structure*

Grouping	Series	Grouping	Series
Total		Language	
	1. Social Network		10. zh (Chinese)
Access		Purpose	
	2. Desktop		11. Blogging related
	3. Mobile app		12. Business
Agent			13. Gaming
	4. Mobile web		14. General purpose
	5. Spider		15. Life style
	6. User		16. Photo sharing
Language			17. Reunion
	7. en (English)		18. Travel
	8. de (German)		19. Video
	9. es (Spanish)		

We consider the main aggregation factors and their two-way combinations[11]. The final dataset includes 913 time series, each with length 394. Table 18 shows the group structure's different levels and the number of series in each level.

[11]First, while we present results for single groups, we applied all the two-way combinations in the reconciliation step. Second, there are four more 3-way aggregation combinations that we do not include: Agent × Access × Language, Agent × Access × Purpose, Agent × Language × Purpose, and Access × Language × Purpose. Including these four additional aggregations might slightly improve the results but for simplicity, we excluded them.

Table 18: *Number of Wikipedia pageviews series at each aggregation level.*

Aggregation_Level	Series
Total pageviews	1
Access	3
Agent	2
Language	4
Purpose	9
Access x Agent	5
Access x Language	12
Access x Purpose	27
Agent x Language	8
Agent x Purpose	18
Language x Purpose	33
Bottom level	913
Total	1035

For this daily dataset, in the OLS forecasting model we include in the predictor matrix the following terms: a quadratic trend, 6 seasonal dummies, and lags 1 and 7 for rolling and fixed origin models. We partitioned the data into training and test sets. We used the last 28 days for our test set and the rest for the training set.

Tables 19 and 20 show the RMSE results. Although these time series are noisier, we still get acceptable results for the OLS forecasting model compared with ETS and ARIMA. In this case, we get similar results with and without the reconciliation step.

Table 19: *Mean RMSE for ETS, ARIMA and OLS with and without reconciliation - Rolling origin - Wikipedia dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	11224	14835	12968	12745	13304	11959
Access	6532	6660	5832	6787	6818	5885
Agent	8263	10278	9372	8571	8992	8504
Language	4889	6273	5688	6059	5879	5448
Purpose	5237	4663	4107	4548	4132	3805
Bottom level	358	240	262	364	242	263

Figures 19 and 20 display the forecast error box plot. These plots are for rolling and fixed origin forecasts over 28 days in each level of grouping. We see that the error distribution is similar in all levels across the different methods. The only exception is the Total series, where ETS performs significantly better than ARIMA and OLS. We also note that the reconciliation is less effective. As in the tourism example, in higher levels series have higher counts and therefore their error magnitudes are larger.

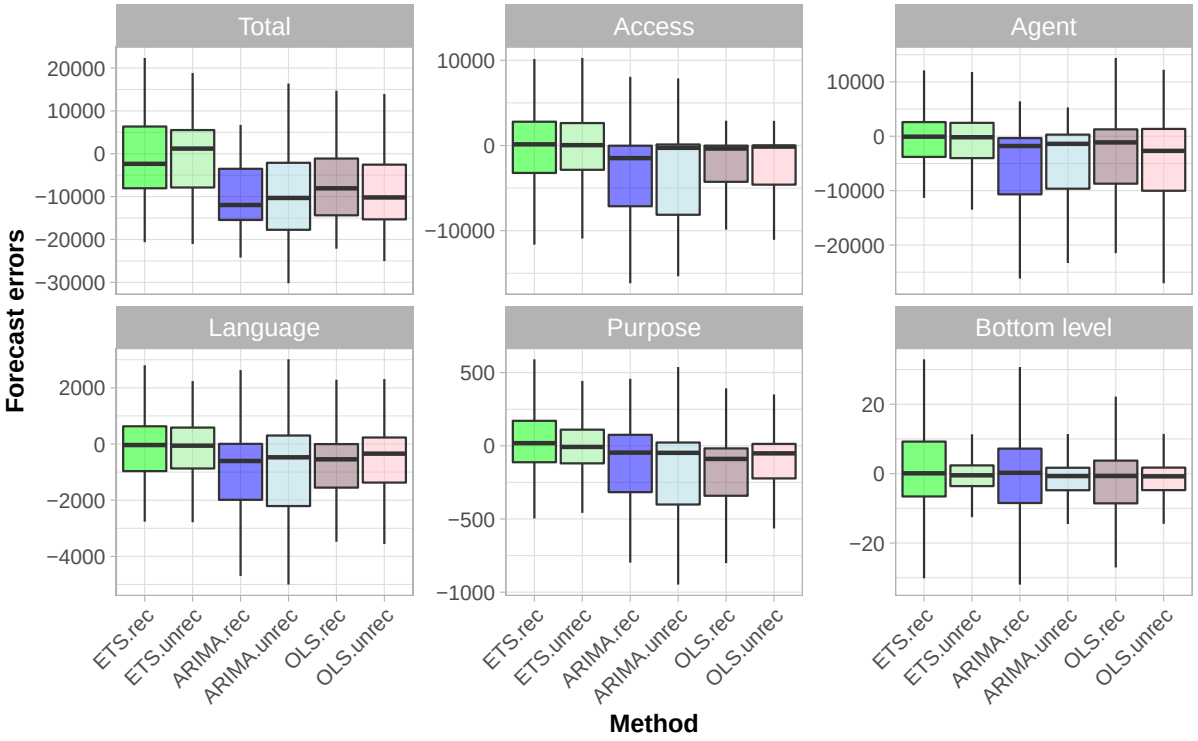

Figure 19: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchy level for rolling origin forecasts of Wikipedia pageviews.*

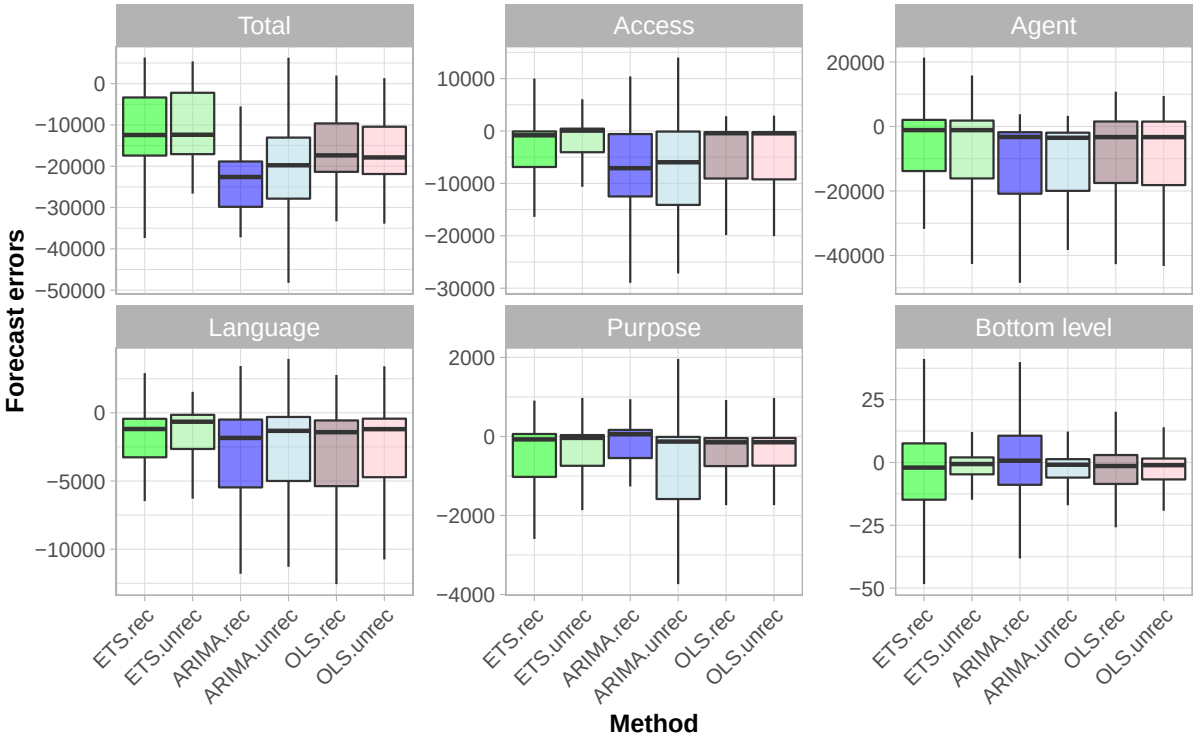

Figure 20: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchy level for fixed origin forecasts of Wikipedia pageviews.*

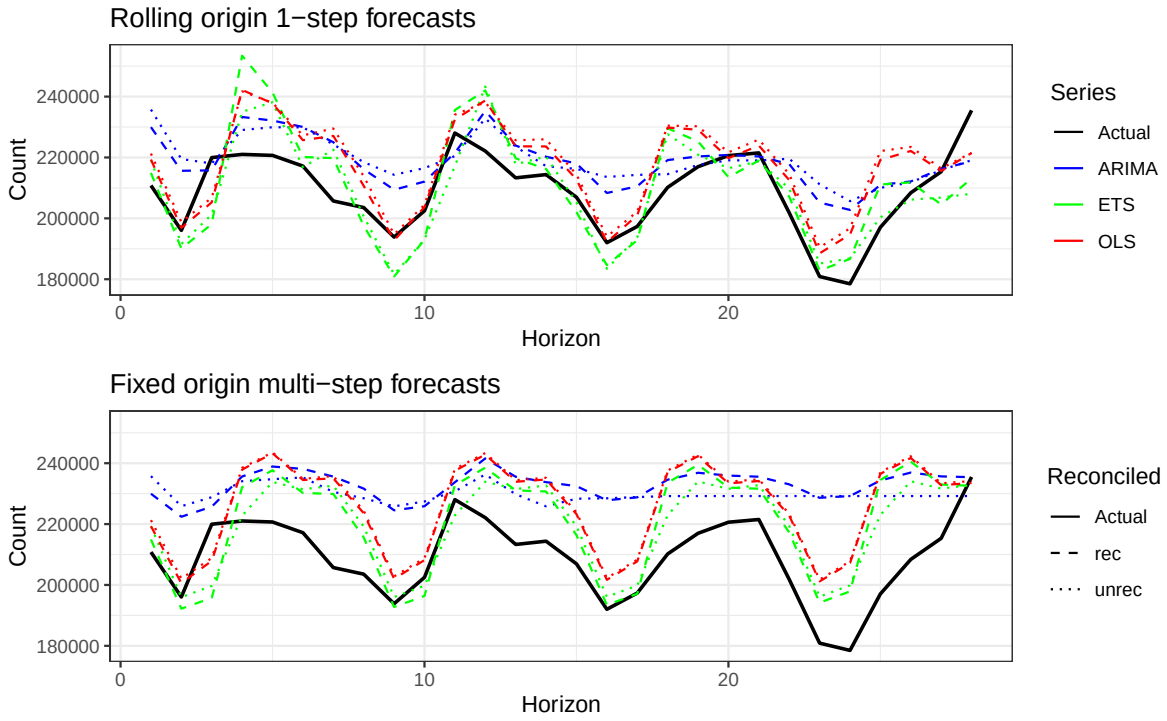

Figure 21: *Comparing reconciled and unreconciled ETS, ARIMA and OLS rolling and fixed origin forecasts for Wikipedia pageviews 'Total' Series.*

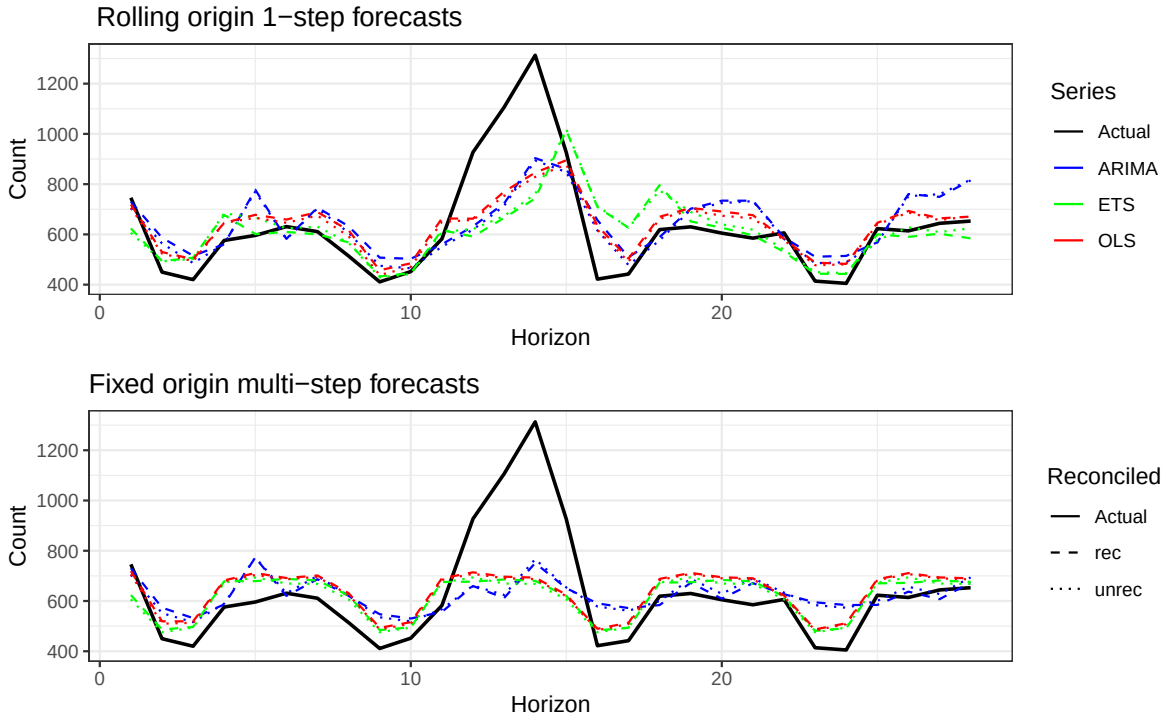

Figure 22: *Comparing reconciled and unreconciled ETS, ARIMA and OLS rolling and fixed origin forecasts for Wikipedia pageviews 'desktopusenPho21' bottom-level series*

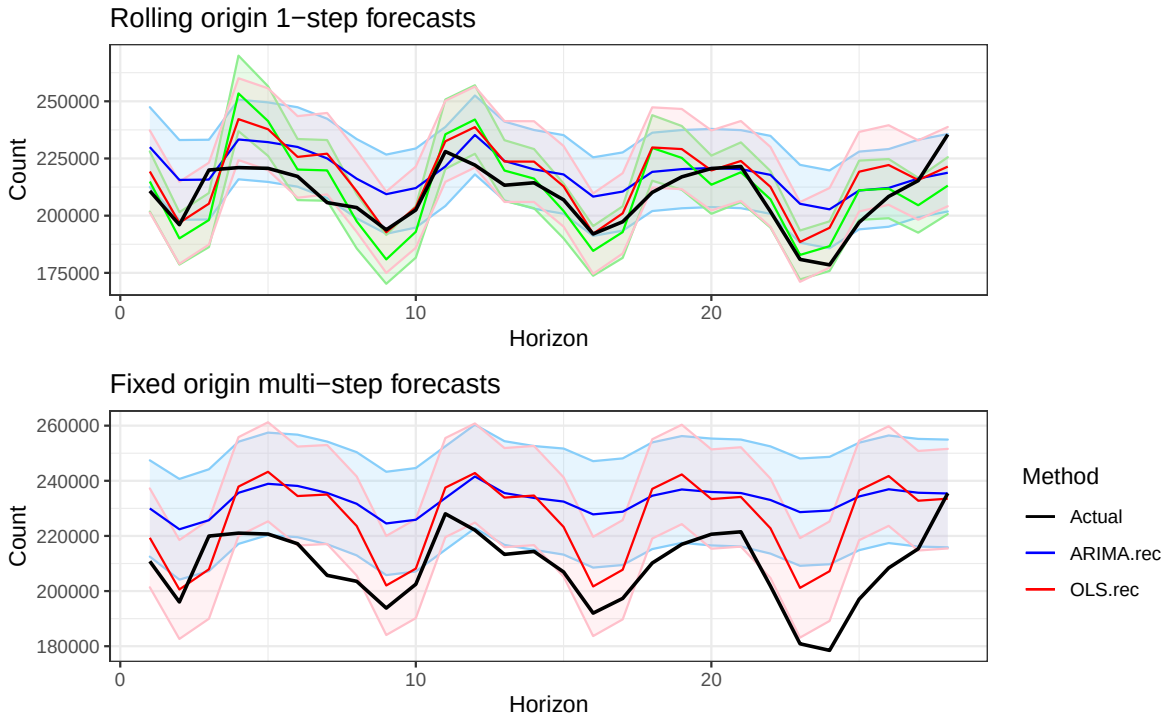

Figure 23: *The actual test set for the 'Total series' compared to the forecasts from reconciled and unreconciled OLS methods with prediction interval for rolling and fixed origin Wikipedia pageviews.*

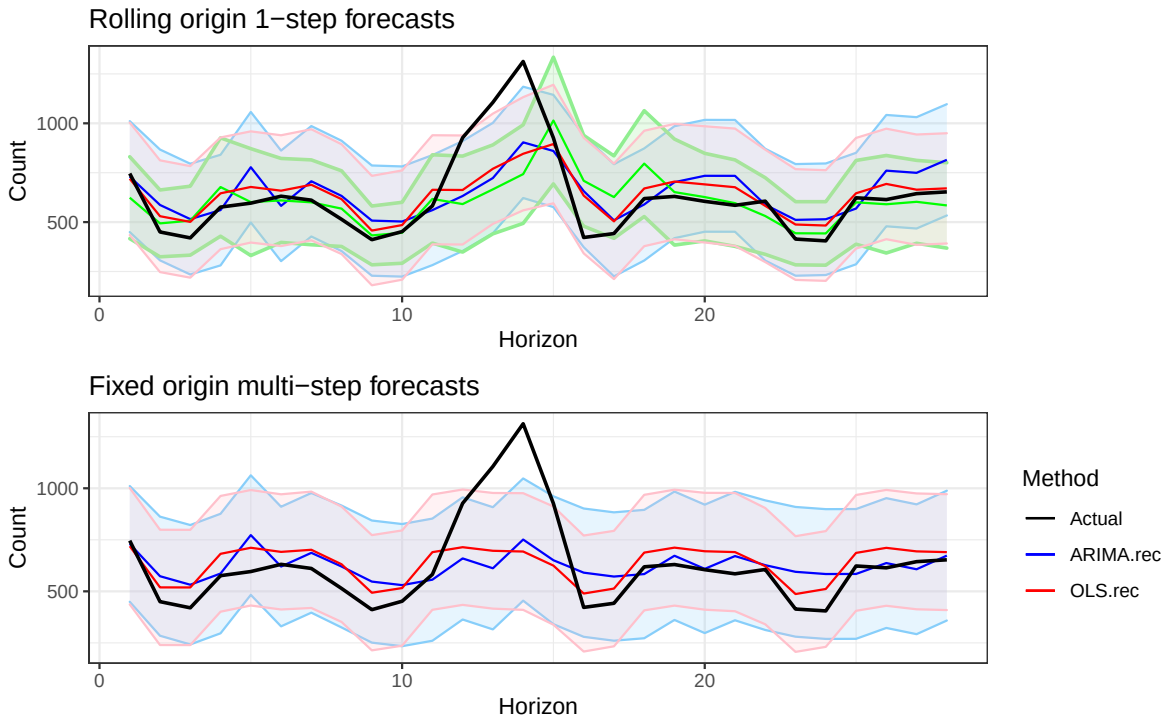

Figure 24: *The actual test set for the 'desktopusenPho21' bottom level series compared to the forecasts from reconciled and unreconciled OLS methods with prediction interval for rolling and fixed origin Wikipedia pageviews.*

Table 20: *Mean RMSE for ETS, ARIMA and OLS with and without reconciliation - Fixed origin - Wikipedia dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	14731	24206	20204	16570	26148	19691
Access	7098	10710	8497	8728	11519	8381
Agent	13634	18163	14986	12508	17656	14580
Language	6248	9505	7914	6855	10756	8022
Purpose	7442	8614	5694	5259	7508	5542
Bottom level	437	388	366	441	390	366

In Figures 21 and 22, we display results for the total and one of the bottom-level series, 'desktopusenPho21' (desktop-user-english-photo sharing). The plot shows rolling and fixed origin forecast results over the 28 day test set for ETS, ARIMA and OLS, with (dashed lines) and without (dotted lines) applying reconciliation. We see that the OLS forecasting model performs close to the other two methods, and reconciliation improves the forecasts. In Figures 23 and 24, we present the reconciled prediction intervals for the total and 'desktopusenPho21' series. In the fixed origin results, the ETS prediction intervals were much wider and hence have been omitted to allow for easier visualization of the results.

Table 21 presents the computation times for all three methods. ETS and ARIMA are clearly much more computationally heavy compared with OLS. As in the Australian tourism dataset, running reconciliation does not have much effect on computation time.

Table 21: *Computation time (seconds) for ETS, ARIMA and OLS with reconciliation - Rolling and fixed origin forecasts - Wikipedia dataset*

	Rolling origin	Fixed origin
ETS	22592	882
ARIMA	20016	1682
OLS	116	61

References

Akaike, H (1998). "Information theory and an extension of the maximum likelihood principle". In: *Selected Papers of Hirotugu Akaike*. Springer Series in Statistics (Perspectives in Statistics). Springer, pp.199–213.

Ashouri, M, G Shmueli & CY Sin (2018). Clustering time series by domain-relevant features using model-based trees. *Proceedings of the 2018 Data Science, Statistics & Visualization (DSSV)*.

Athanasopoulos, G, RA Ahmed & RJ Hyndman (2009). Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting* **25**(1), 146–166.

Bunch, JR & DJ Rose (2014). *Sparse matrix computations*. Academic Press.

Christensen, R (2020). *Plane answers to complex questions: the theory of linear models*. 5th. Springer.

Fliedner, G (2001). Hierarchical forecasting: issues and use guidelines. *Industrial Management & Data Systems* **101**(1), 5–12.

Gross, CW & JE Sohl (1990). Disaggregation methods to expedite product line forecasting. *Journal of Forecasting* **9**(3), 233–254.

Hyndman, RJ, RA Ahmed, G Athanasopoulos & HL Shang (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis* **55**(9), 2579–2589.

Hyndman, RJ & G Athanasopoulos (2018). *Forecasting: principles and practice*. Melbourne, Australia: OTexts. https://OTexts.org/fpp2.

Januschowski, T, S Kolassa, M Lorenz, C Schwarz, et al. (2013). Forecasting with in-memory technology. *Foresight: The International Journal of Applied Forecasting* **31**, 14–20.

Kahn, KB (1998). Revisiting top-down versus bottom-up forecasting. *The Journal of Business Forecasting* **17**(2), 14.

Maechler, M & D Bates (2006). 2nd Introduction to the Matrix package. *R Core Development Team. Accessed on: https://stat. ethz. ch/R-manual/R-devel/library/Matrix/doc/Intro2Matrix. pdf*.

Makoni, T, D Chikobvu & C Sigauke (2021). Hierarchical Forecasting of the Zimbabwe International Tourist Arrivals. *Statistics, Optimization & Information Computing* **9**(1), 137–156.

Makridakis, S, E Spiliotis & V Assimakopoulos (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one* **13**(3), e0194889.

O Hara-Wild, M, R Hyndman & E Wang (2019). *fable: Forecasting Models for Tidy Time Series. R package version 0.1. 0*.

Pennings, CL & J van Dalen (2017). Integrated hierarchical forecasting. *European Journal of Operational Research* **263**(2), 412–418.

Syntetos, AA, Z Babai, JE Boylan, S Kolassa & K Nikolopoulos (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research* **252**(1), 1–26.

Taieb, SB, JW Taylor & RJ Hyndman (2020). Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 1–17.

Tourism Research Australia (2005). Travel by Australians, September Quarter 2005. *Tourism Australia*.

Wickramasuriya, SL, G Athanasopoulos & RJ Hyndman (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association* **14**(526), 804–819.