A fast and elegant method for forecast reconciliation using linear forecasting 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

7 October 2019

A fast and elegant method for forecast reconciliation using linear forecasting models

Abstract

Forecasting hierarchical or grouped time series 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 uses a single-step approach to obtain the forecasts, rather than the usual two-step approach. The proposed method is very flexible in incorporating external data and handling missing values. We illustrate our approach using two datasets: monthly Australian domestic tourism and daily Wikipedia pageviews. 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. For example, the Internet of Things and point-of-sale 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.

1.1 Hierarchical and grouped time series

In many cases, these time series can be structured and disaggregated based on hierarchies or groups such as geographic location, product type, gender, etc. An example of hierarchical time series is sales in restaurant chains, which can be disaggregated into different stores and then

different types of food or drinks. Figure 1 shows a schematic of such a hierarchical time series structure with three levels. The top level (level 0) is the total series, formed by aggregating all the bottom level series. In the middle level (level 1), series are aggregations of their own child series; for instance, series A is the aggregation of AW and AX. Finally, the bottom level series (level 2), includes the most disaggregated series.
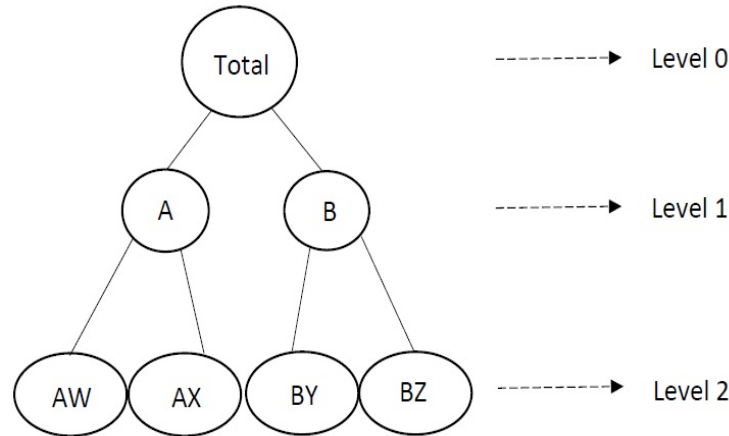

Figure 1: *An example of a two level hierarchical structure.*

Grouped time series involve more complicated aggregation structures compared to strictly hierarchical time series. To take the simplest example, suppose we have two grouping factors which are not nested: sex (Male/Female) and city (New York/San Francisco). The disaggregated series for each combination of sex and city can be combined to form city sub-totals, or sex sub-totals. These sub-totals can be combined to give the overall total. Both sub-totals 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.

We use the same notation (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 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 $y_t = Sb_t$, where y_t is a vector
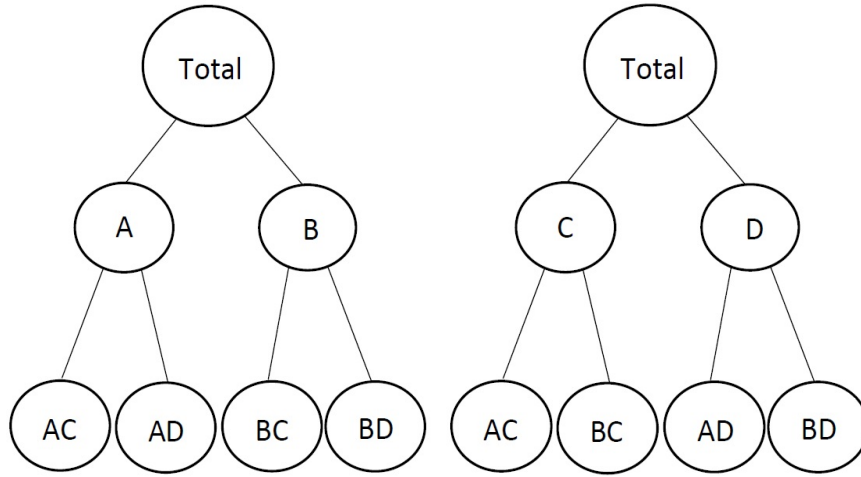
Figure 2: *An example of a two level grouped structure.*

of all the level nodes at time t and 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:

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

1.2 Forecasting hierarchical time series

If we just forecast each series individually, we are ignoring the hierarchical or grouping structure, and the forecasts will not be "coherent" (they will not add up appropriately).

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 and 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 n 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.

In the optimal combination method, reconciled forecasts can be computed using (Wickramasuriya, Athanasopoulos & Hyndman 2019)

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_h, \tag{1}$$

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

Several possibly methods for estimating $\boldsymbol{W}_h$ are available. Wickramasuriya, Athanasopoulos & Hyndman (2019) discuss a simple 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 column of 1s. Note that $\boldsymbol{\Lambda}$ simply contains the row sums of the summing matrix $\boldsymbol{S}$, and that k_h will cancel out in (1). Thus

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\hat{\boldsymbol{y}}_h. \tag{2}$$

The most computationally challenging part of the optimal combination method is to produce all the base forecasts that make up $\hat{\boldsymbol{y}}_h$. 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 select 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.

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 $t = 1, \ldots, T$ and series $i = 1, \ldots, N$. Then

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

where $\boldsymbol{x}_{t,i} = \{1, x_{t,i,1}, \ldots, x_{t,i,p}\}$ is a $(p+1)$-vector of regression variables. This equation for all the observations in matrix form can be written as follows:

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

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

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

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

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

Note that we use X^*_t to distinguish this matrix, which combines $x_{t,i}$ across all series for one time from X_i which combines $x_{t,i}$ across all time for one series.

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

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

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

If we have the same set of predictor variables, 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 Y, X, B and 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 $X^*_{t+h,i} = X^*_{t+h}$, where X^*_{t+h} is a $h \times (p+1)$ matrix.

2.2 OLS predictors

As an example of the 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:

$$
y_t = \alpha_0 + \alpha_1 t + \beta_1 s_{1,t} + \dots + \beta_{m-1}s_{m-1,t} + \gamma_1 y_{t-1} + \dots + \gamma_p y_{t-p} + \delta z_t + \varepsilon_t. \tag{7}
$$

Here, $s_{j,t}$ is a dummy variable taking value 1 if time t is in season j ($j = 1, 2, \ldots, m$), 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 $m = 7$.

In Equation (7) because of using lags and external series as predictors we do not have same set of predictors for all the series , y_t. However, if we just use trend and seasonality dummies as the predictors then the simpler equations explained in Section 2, can be written using multivariate regression models.

While OLS is popular in practice for forecasting time series, it is often frowned upon due to its independence assumption. This can cause issue for parametric inference but is less of a problem for forecasting. In fact it often performs sufficiently well for forecasting as can be seen by its popular use in practice. Further, the use of autoregressive terms in the above model should model most of the autocorrelation in the data.

2.3 Computational considerations

There are two ways for computing above forecasts. In the first way we can create matrices, Y, X and E. In this way we need to compute the inverse matrix to compute the coefficients. To avoid numerical issues due to matrix inversion, one can use 'separate regression models' to compute the coefficients for each linear model individually.

Although the matrix, $\mathbf{X}'\mathbf{X}$, for which we want to compute the inverse is sparse and block diagonal, it is still faster and more accurate to use the 'separate regression models' approach. In Appendix C we give the forecast results comparisons for Australian domestic tourism example using these two computation approaches.

3 Applications

In this section we illustrate our approach using two examples: forecasting monthly Australian domestic tourism and forecasting daily Wikipedia pageviews. We compare the forecasting accuracy of ETS, ARIMA and the proposed linear OLS forecasting model, with and without the reconciliation step. 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.

The two datasets differ in terms of structure, size and behavior. The tourism data contains 304 series with both hierarchical and grouped structure, while the Wikipedia pageviews dataset

contains 913 series with grouped structure. The tourism dataset has strong seasonality while the Wikipedia data are noiser.

We apply two methods for generating h-step-ahead forecasts, which differ in how they handle unobserved lagged values as inputs. The first approach is *rolling origin*, in that it uses actual values, even when they are future to the forecast origin. This value is known to us because it is in the test set. We call these "rolling origin" forecasts. In the second method, we follow an *fixed origin* approach, and we replace lagged values of y by their forecasts if they occur at periods after the forecast origin. We call these "fixed origin" forecasts.

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: *Australia geographic hierarchical structure.*

Series	Name	Label	Series	Name	Label
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
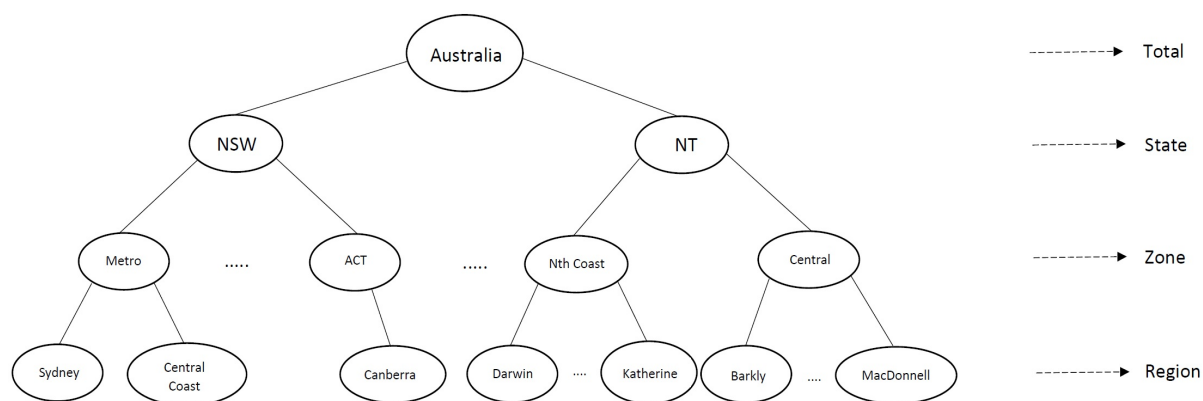
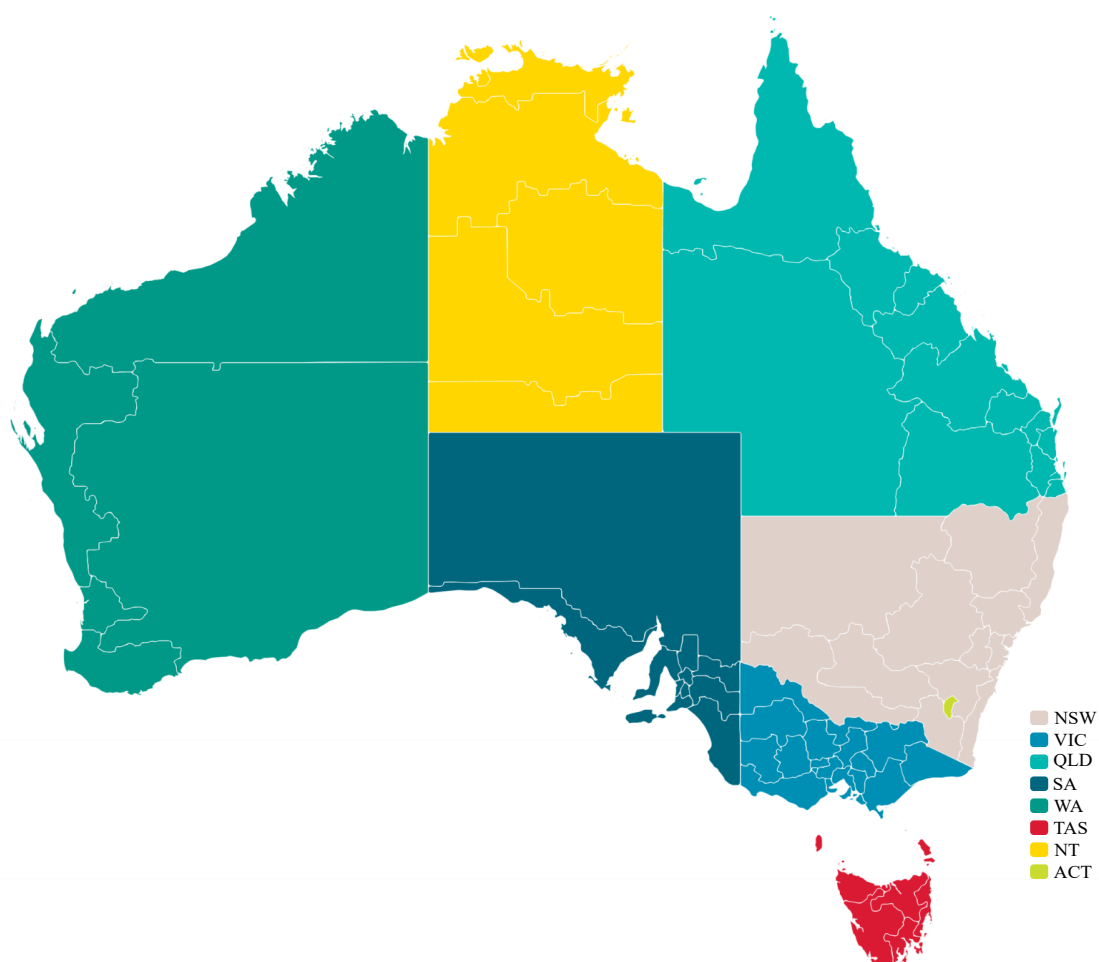**Figure 3:** *Australian geographic hierarchical structure.*

Figure 4: *Australia tourism region map - colors represent states.*

In this dataset we have 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' (for details about Australia geographic divisions see Figure 3 and Table 1 and also Figure 4 which shows Australia map divided by tourism region and colored by states[1]).

For 'Purpose' we have four groups: Holiday (Hol), Visiting friends and relatives (Vis), Business (Bus) and Other (Oth). Based on the geographic hierarchy and purpose grouping, we end up with 8 hierarchical levels with 555 series in total:

- Level 0 = Total for Australia
- Level 1 = State totals
- Level 2 = Zone totals
- Level 3 = Region totals
- Level 4 = Purpose totals
- Level 5 = State $\times$ Purpose totals
- Level 6 = Zone $\times$ Purpose totals
- Level 7 = bottom level series

Table 2: *Number of Australian domestic tourism series in each level of the hierarchy and group structure.*

Geographic division	# series (geographic division)	# series (purpose of travel)	Total
Australia	1	4	5
State	7	28	35
Zone	27	108	135
Region	76	304	380
Total	111	444	555

We report the forecast results for all these hierarchical levels, as well as the average RMSE across all the levels of the hierarchy. In the predictor matrix for the OLS forecasting model, we include a linear trend, 11 dummy variables, and 12 time series lags[2]. This is intended to capture the 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 Tables 3 and 4, we show the average RMSE for the 1-month and 24-month forecasts. Methods include ETS, ARIMA and our proposed OLS forecasting model. In Table 3 we forecast 24 periods

---

[1]For more information about the map: https://www.tra.gov.au/tra/2016/Tourism_Region_Profiles/Region_profiles/index.html

[2]Since the forecasting results are better with fewer lags, we use a linear trend and dummy seasonality variables and four lags in our linear model for the fixed origin 24-step-ahead model.

Table 3: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Rolling origin 24-step-ahead - Tourism dataset*

Level	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	1516.4	1445.5	1415.1	1533.6	1453.4	1454.4
Level 1	511.4	493.1	510.8	495.9	457.6	488.3
Level 2	214.8	219.0	224.5	209.2	207.5	212.4
Level 3	122.9	125.1	124.0	118.7	120.5	119.5
Level 4	676.0	709.2	694.5	668.3	679.7	678.5
Level 5	213.1	220.1	216.1	210.6	209.4	211.1
Level 6	97.5	102.4	101.0	96.4	99.8	98.6
Level 7	56.2	58.2	58.2	56.0	57.7	57.2

Table 4: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Fixed origin 24-step-ahead - Tourism dataset*

Level	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	2238.6	3554.0	5050.3	2250.2	3179.4	4671.1
Level 1	593.6	570.1	900.1	553.8	626.3	903.0
Level 2	239.5	229.6	278.8	234.2	242.5	291.0
Level 3	132.6	129.4	142.8	126.7	129.4	146.3
Level 4	766.8	824.0	1307.0	795.5	958.2	1434.1
Level 5	226.7	241.2	298.2	222.5	236.9	308.4
Level 6	103.0	105.4	110.9	102.0	103.9	115.4
Level 7	59.1	58.8	60.9	58.5	58.7	62.6

ahead by computing rolling origin forecasts, rolling forward month by month. In Table 4 we generate fixed origin 24-step-ahead forecasts. In these tables we show forecasts with and without reconciliation.

These results show that our proposed OLS forecasting model produces forecast accuracy similar to ETS and ARIMA, which are computationally heavy for many time series. Also they show the usefulness of the reconciliation in decreasing the average RMSE in all the three methods. Except for the total series, reconciliation can help in forecasting all the hierarchical levels. Also, because the higher level series have higher counts, the errors are larger in magnitude (see Figures 5 and 6). To better compare errors across all the hierarchy levels we scaled the errors (See Appendix A).

In Figures 5 and 6 we display the error box plots for both reconciled and unreconciled forecasts using all three methods, for 1-step ahead and 24-steps ahead. In these figures we can visualize the error distributions across all the models, as well as the usefulness of the reconciliation step

in improving the forecasts. In particular, we see that as expected by applying rolling origin 24-step-ahead forecasts, the error densities are closer and more distributed around zero than the fixed origin 24-step-ahead forecasts.

Figures 7 and 8 show the rolling and fixed origin 24-step-ahead forecast results for the total series and one of the bottom level series, 'BACBus' (Geelong - Business). In these plots we have both reconciled (solid lines) and unreconciled (dashed lines) forecasts and we see that the reconciliation step improves the forecasts in this series. We also see that the OLS model forecast accuracy is similar to the other two methods.
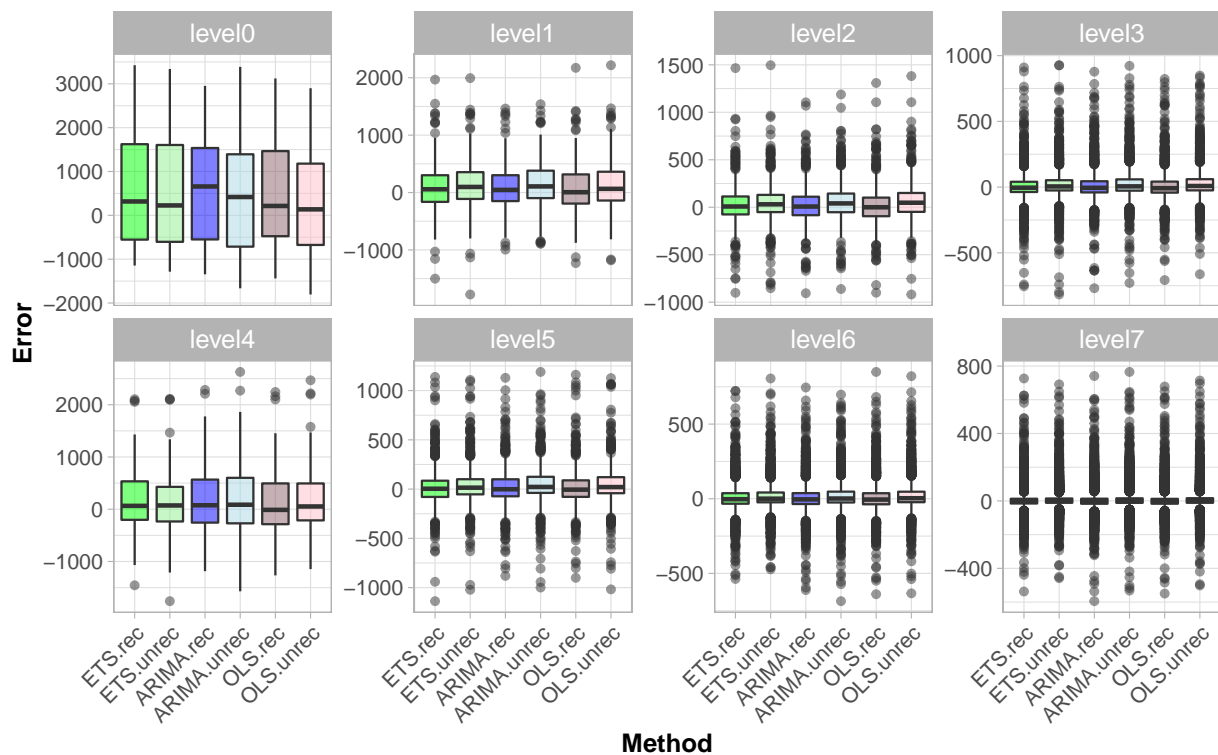
Figure 5: *Box plots of forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin 24-step-ahead tourism demand.*
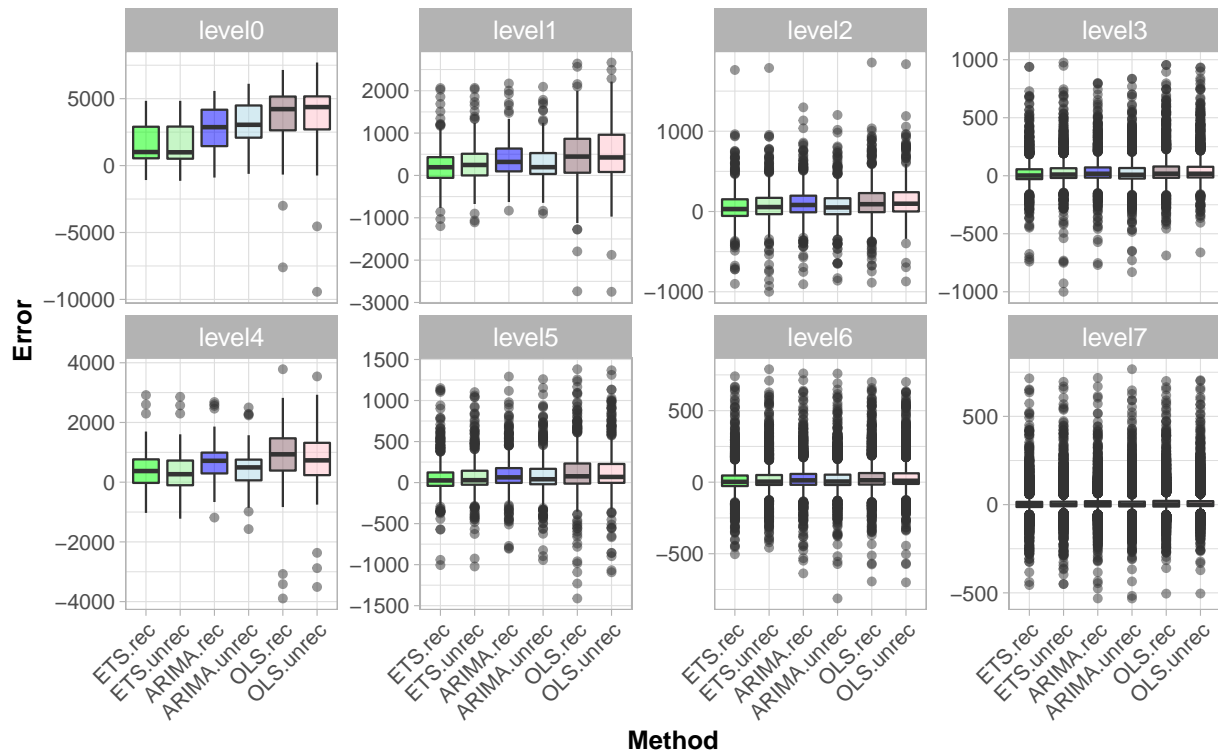
Figure 6: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for fixed origin 24-step-ahead tourism demand.*
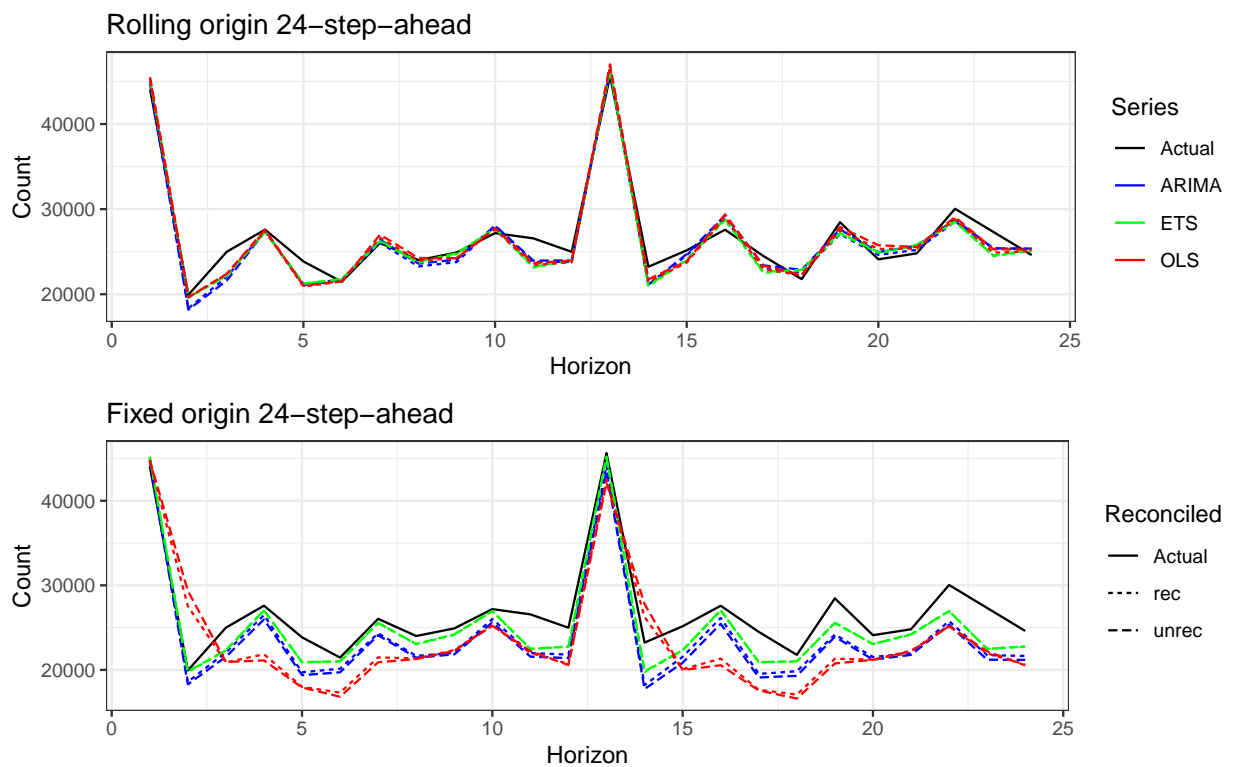
Figure 7: *The actual test set for the 'Total series' compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin 24-step-ahead tourism demand.*
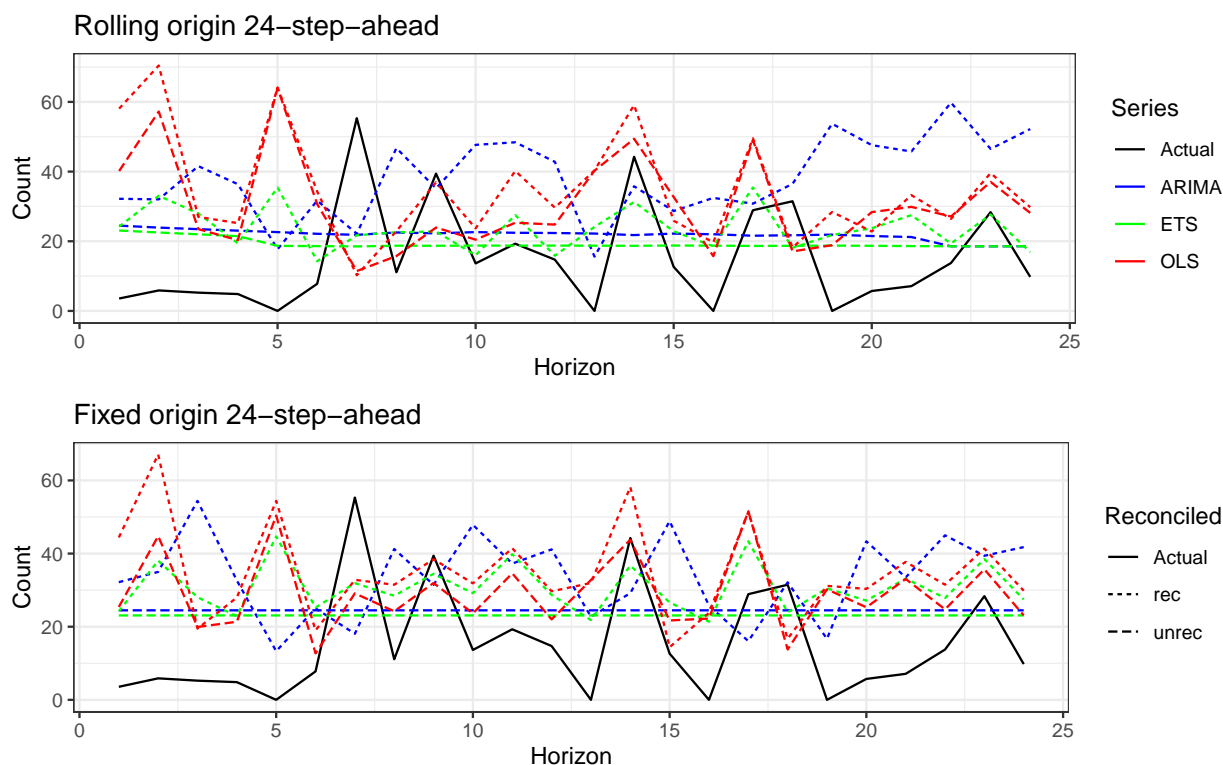
Figure 8: *The actual test set for the 'BACBus' bottom level series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin 24-step-ahead tourism demand.*

Table 5 compares the computation time of the three methods for rolling and fixed origin 24-step-ahead forecasting. We see that the OLS forecasting 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.

Table 5: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Rolling and fixed origin 24-step-ahead - Tourism dataset*

	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
ETS	10924.57	10924.60	407.10	407.15
ARIMA	31146.38	31146.52	1116.15	1116.19
OLS	48.40	48.31	16.66	16.85

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, but its affect on forecast accuracy was minimal, so it was omitted in the model reported here.

3.2 Wikipedia pageviews: Grouped structure

The second dataset consists of one year 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 compared with the Australian monthly tourism data and forecasting its series is more challenging. It has a grouped structure, with grouping attributes: '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 (check Table 6). We display the group structure in Table 6 and Figure 9. In Figure 9 we use one possible hierarchy for this dataset, but the order of the hierarchy can be switched. The final dataset includes 913 time series, each with length 394. The group structure and different levels include:

- Level 0 = Total
- Level 1 = Agent
- Level 2 = Access
- Level 3 = Language
- Level 4 = Purpose
- Level 5 = bottom level series

For this daily dataset, in the OLS forecasting model we include in the predictor matrix a linear trend, 6 seasonal dummies and 7 lags. We partitioned the data into two parts training and test sets. We used the last 28 days for our test set and the rest for the training set.

Table 6: *Social networking Wikipedia article grouping structure*

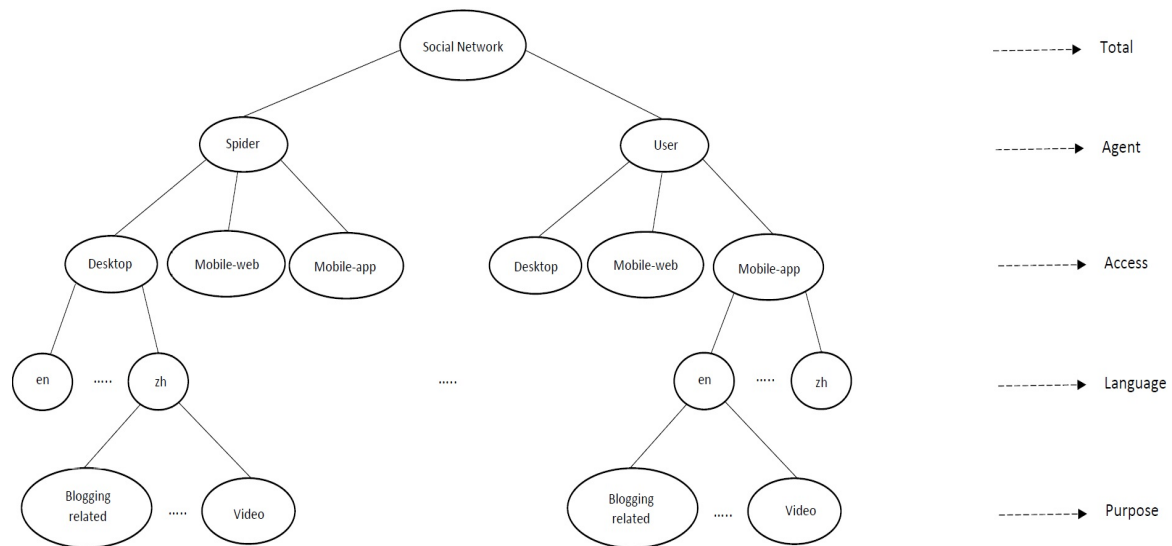| Grouping	Series	Grouping	Series
Total		Language	
	1. Social Network		10. zh (Chinese)
Agent		Purpose	
	2. Spider		11. Blogging related
	3. User		12. Business
Access			13. Gaming
	4. Desktop		14. General purpose
	5. Mobile app		15. Life style
	6. Mobile web		16. Photo sharing
Language			17. Reunion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

Figure 9: *One of the possible hierarchical structures for the Wikipedia pageview dataset.*

Table 7, 8 and 9 represent the RMSE results and computation time. 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.

Figures 10 and 11 display the forecast error box plot. These plots are for rolling and fixed origin 28-step-ahead forecasts in each level of grouping. Further, we can see that the error distribution is almost 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. In Appendix B we display box plots for the errors devided by the error range for easier comparison.

Table 7: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Rolling origin 28-step-ahead - Wikipedia dataset*

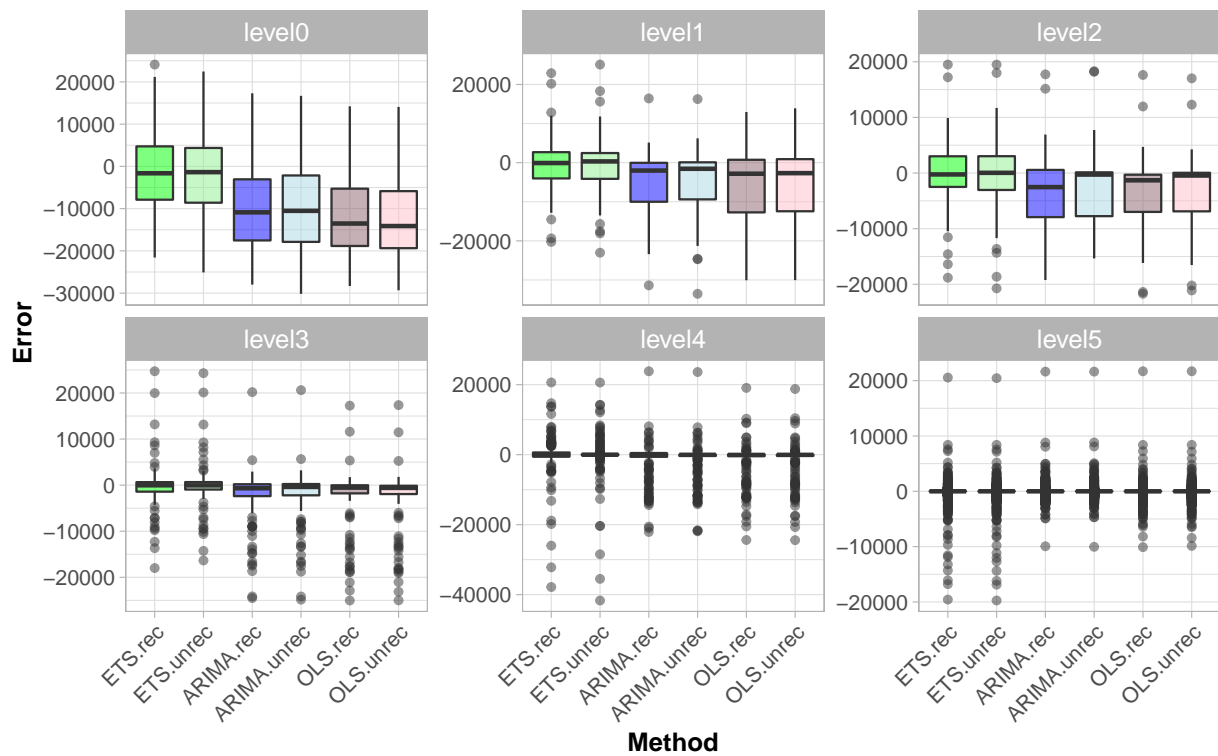|	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	10773.7	15060.7	15748.2	11014.7	14276.5	15270.2
Level 1	8272.9	10196.3	10623.9	7736.9	9904.1	10674.0
Level 2	6524.7	6705.0	6979.6	6257.4	7142.5	7286.0
Level 3	4870.1	6333.0	7150.1	4981.9	6370.0	7106.1
Level 4	5233.5	4659.5	4675.2	5001.4	4586.5	4650.3
Level 5	358.1	239.0	255.0	362.2	241.6	256.1

Figure 10: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin 28-step-ahead Wikipedia pageviews.*
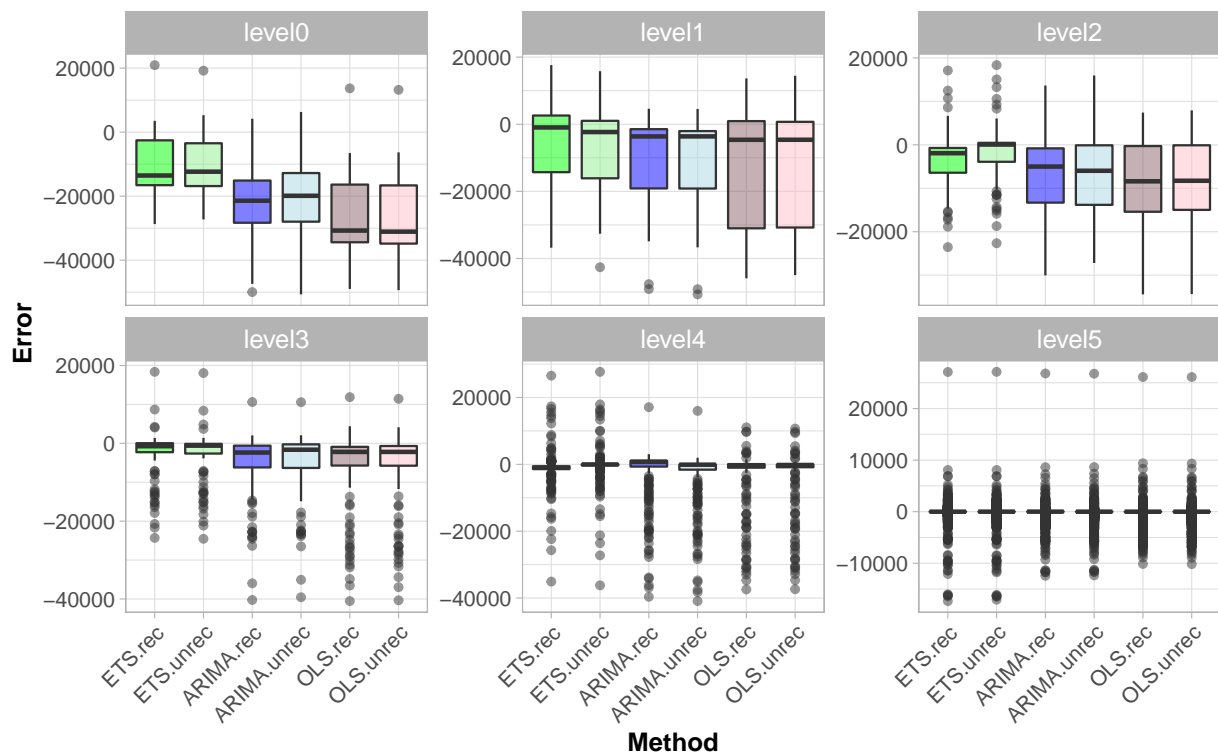

Figure 11: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for fixed origin 28-step-ahead Wikipedia pageviews.*

Table 8: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Fixed origin 28-step-ahead - Wikipedia dataset*

Level	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	14846.9	24298.8	29840.6	14999.2	24649.9	29665.7
Level 1	13608.7	17277.0	21165.3	12240.3	16810.5	21048.1
Level 2	7117.4	10732.0	12678.9	7523.4	11068.8	12811.2
Level 3	6475.9	9580.4	12056.6	6509.0	9799.1	12112.5
Level 4	5302.7	8611.3	8451.1	5307.3	8239.8	8460.3
Level 5	435.6	389.4	390.1	437.7	391.2	391.0

In Figures 12 and 13, we display results for the total and one of the bottom level series, 'desktopusenPho' (desktop-user-english-photo sharing). The plot shows rolling and fixed origin 28-step-ahead forecast results for ETS, ARIMA and OLS, with (solid lines) and without (dashed lines) applying reconciliation. We see that the OLS forecasting model performs close to the other two methods, and reconciliation improves the forecasts.
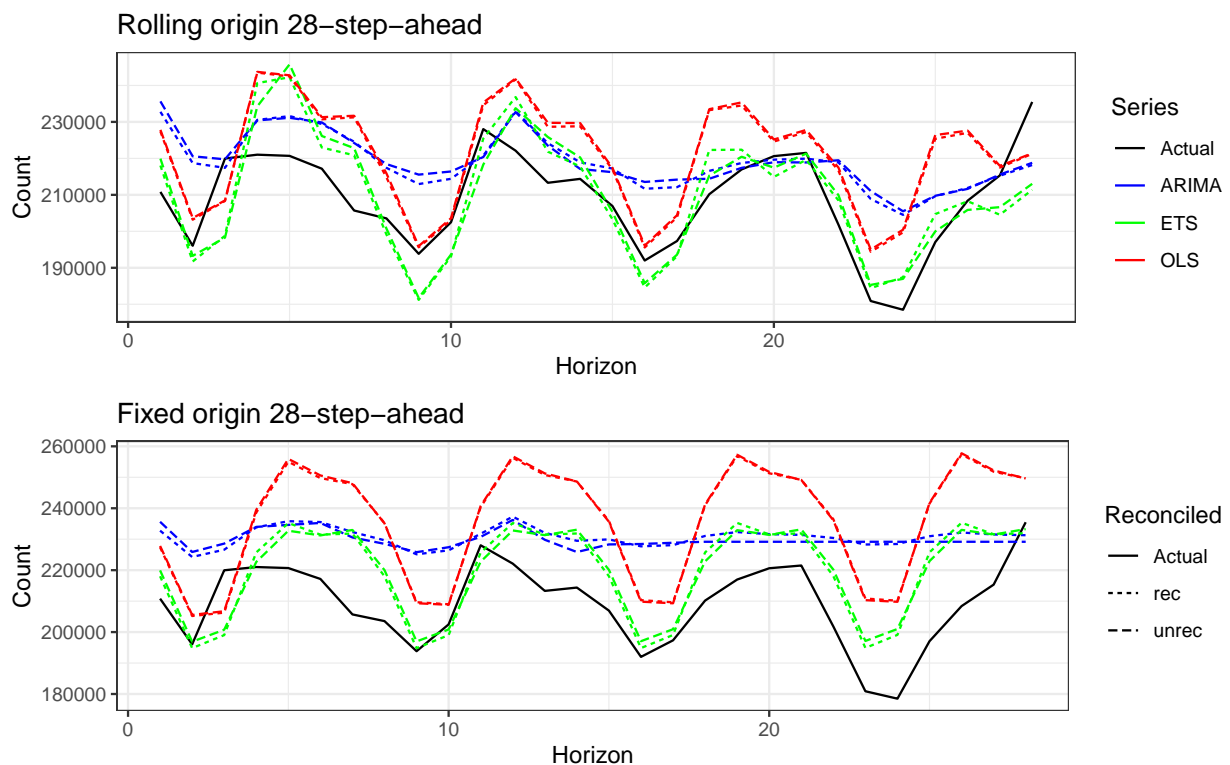

Figure 12: *The actual test set for the 'Total' series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin 28-step-ahead Wikipedia pageviews.*

Lastly, Table 9 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.
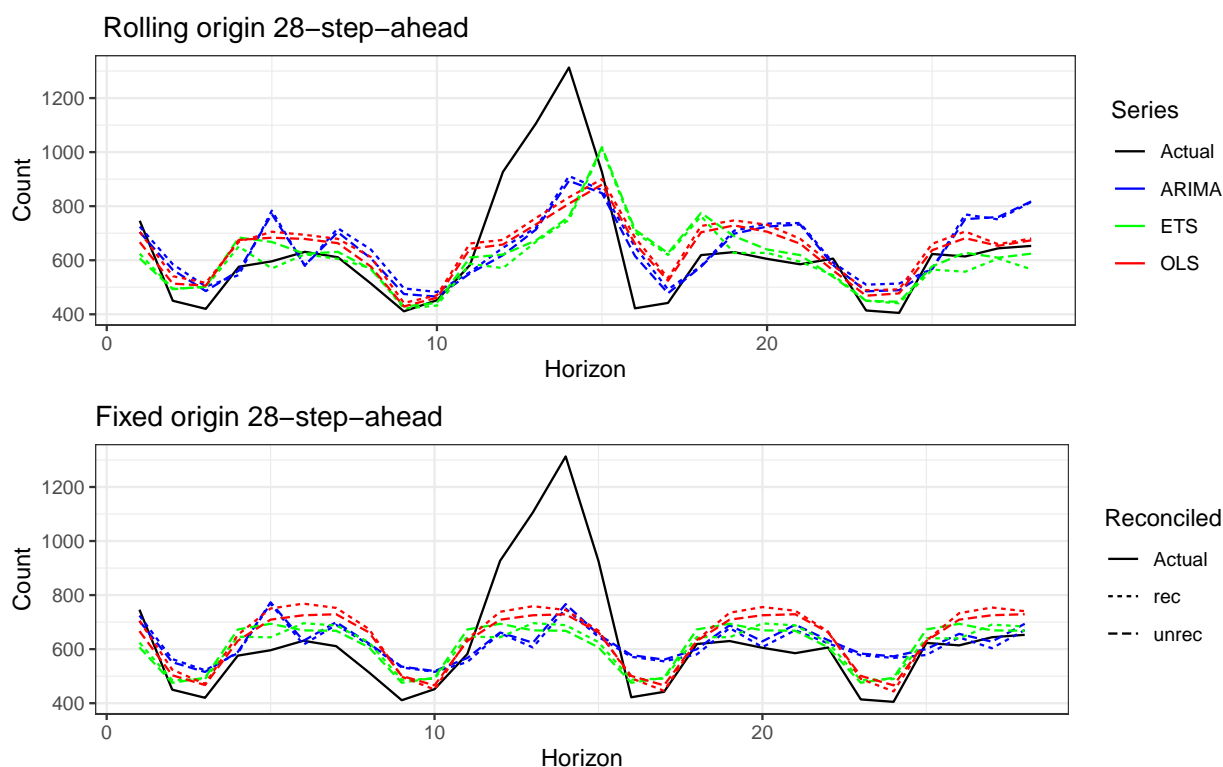
Figure 13: *The actual test set for the 'desktopusenPho21' bottom level series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin 28-step-ahead Wikipedia pageviews.*

Table 9: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Rolling and fixed origin 28-step-ahead - Wikipedia dataset*

	Computation time (secs)			
	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
ETS	13963.93	13963.96	450.89	450.92
ARIMA	10327.02	10327.15	670.40	670.44
OLS	82.55	82.62	35.39	35.43

4 Conclusion

We proposed a single-step linear model approach to forecast hierarchical or grouped time series in a much faster way, but 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. Adding another faster option for calculating base forecasts was our purpose in this research. Here we suggest a linear model, OLS, instead of ETS and ARIMA which is not computationally intensive. We also showed that OLS can compete with ETS and

ARIMA in terms of forecasting accuracy. An important feature of our 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. In addition to the computation adjustment, our proposed approach forecasts hierarchical time series in a parsimonious single step whereas other available methods all require a two-step procedure.

Acknowledgements

Appendix A

We provide boxplots of the scaled forecasted errors for the tourism example. These plots are displayed for both rolling forward and multiple-step-ahead forecasts.
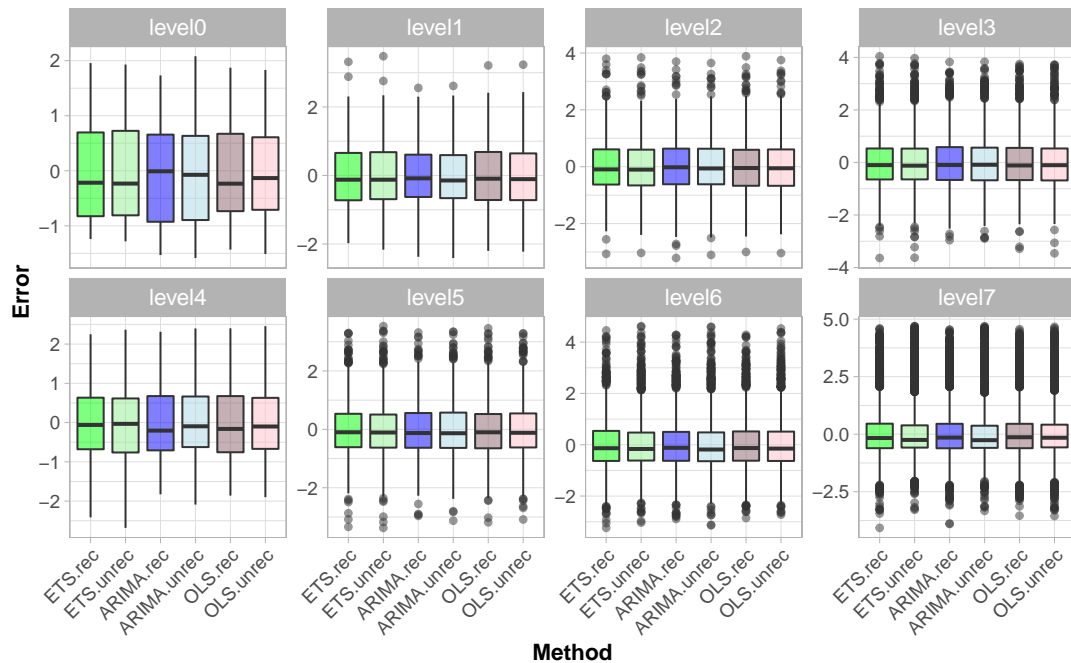
Figure 14: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin 24-step-ahead tourism demand.*
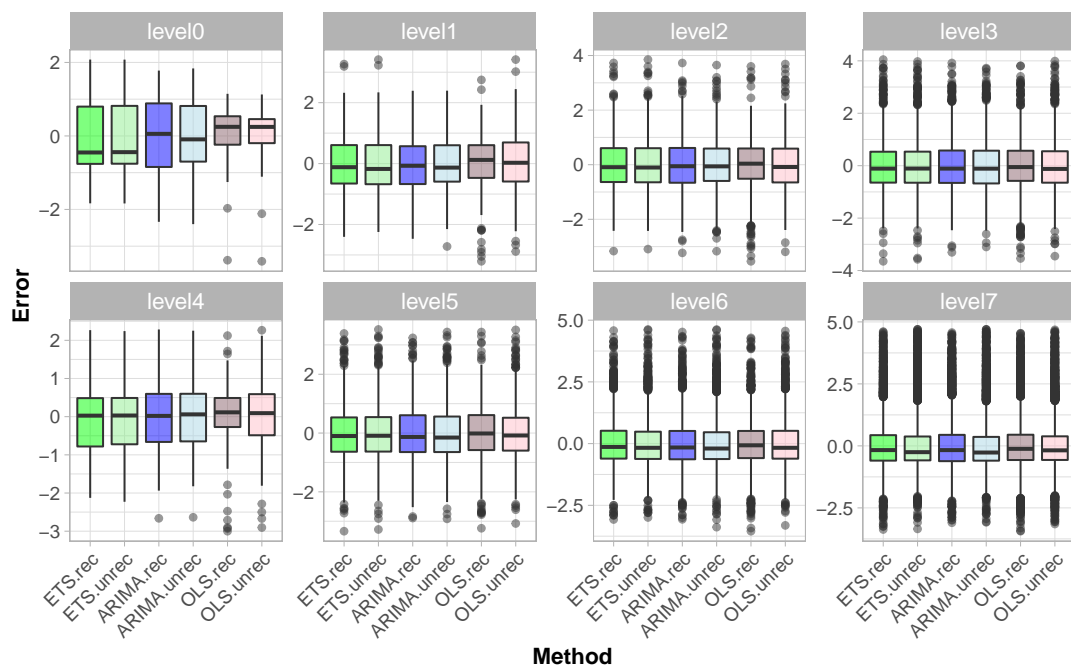

Figure 15: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for fixed origin 24-step-ahead tourism demand.*

Appendix B

We provide boxplots of the scaled forecasted errors for the Wikipedia pageviews example. These plots are displayed for both rolling forward and multiple-step-ahead forecasts.
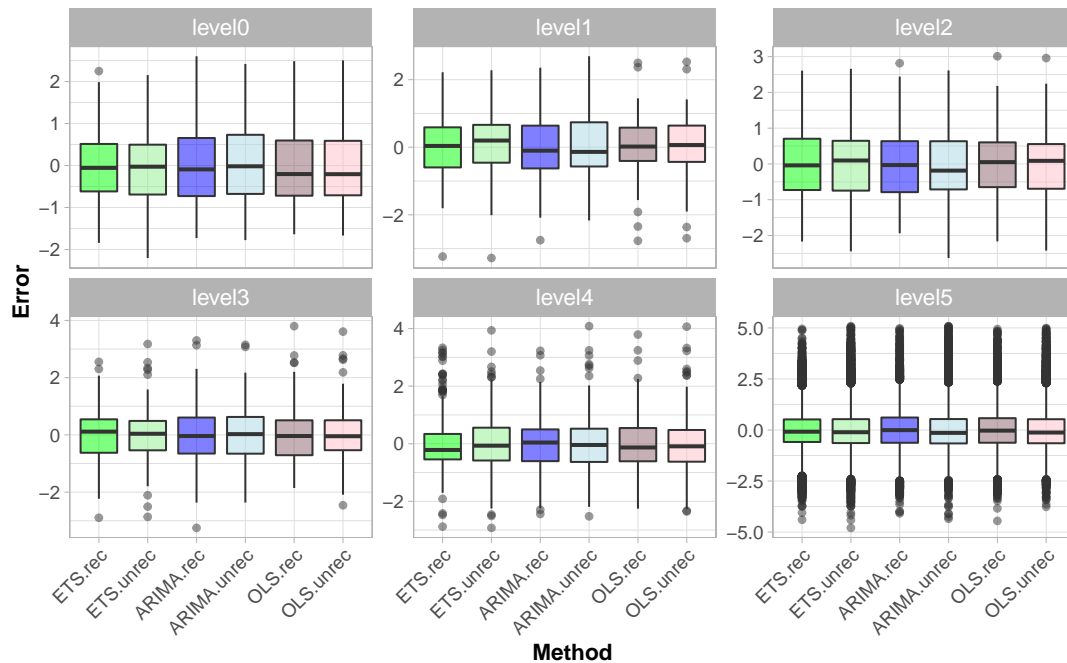

Figure 16: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin 28-step-ahead Wikipedia pageviews.*
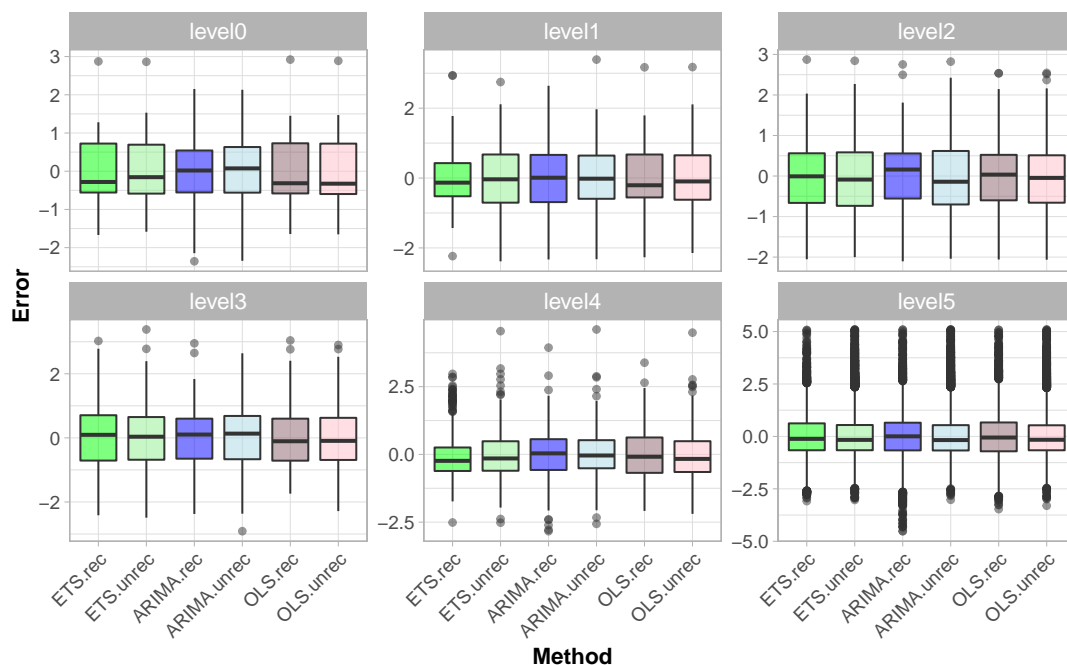

Figure 17: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS method at each hierarchical level for fixed origin 28-step-ahead Wikipedia pageviews.*

Appendix C

In Figure 18, we compare the forecasts obtained using separate regression models versus matrix computation for the Australian domestic tourism example. Although mathematically both approaches should yield the same results, the results in practice are slightly different due to numerical deviations. Table 10 shows the matrix approach is computationally heavier due to greater memory requirements.
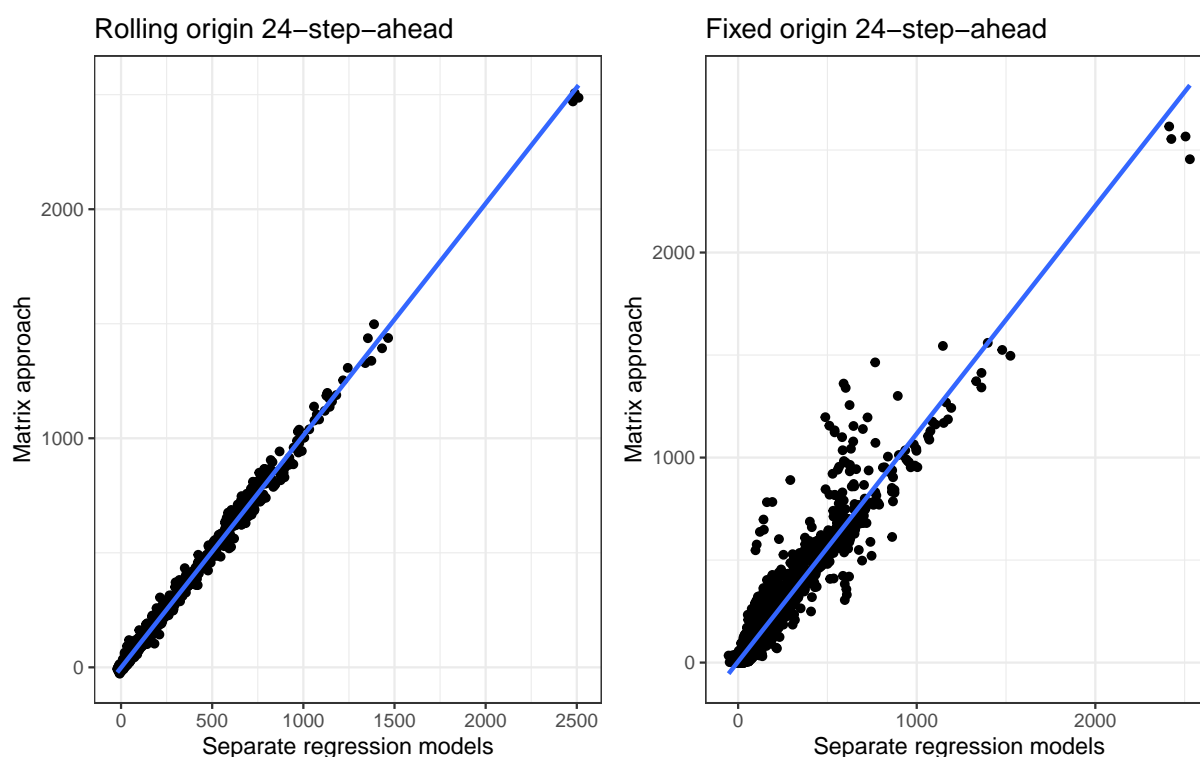

Figure 18: *Comparison of the forecasts obtained using a matrix approach and separate regression models to reconcile forecasts for rolling and fixed origin 24-step-ahead tourism demand (bottom level series only).*

Table 10: *Computation time (seconds) for OLS using the matrix approach and separate regression models, with and without reconciliation, on a rolling and fixed origin for 24 steps ahead, using the tourism dataset.*

	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
Matrix approach	202.06	209.84	87.73	105.69
Separate models	48.40	48.31	16.66	16.85

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.

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.

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

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.