

# **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](mailto:mahsa.ashouri@iss.nthu.edu.tw)

Corresponding author

**Rob J Hyndman**

Monash University, Clayton VIC 3800, Australia

Email: [rob.hyndman@monash.edu](mailto:rob.hyndman@monash.edu)

**Galit Shmueli**

Institute of Service Science, National Tsing Hua University, Taiwan

Email: [galit.shmueli@iss.nthu.edu.tw](mailto:galit.shmueli@iss.nthu.edu.tw)

9 July 2019

**JEL classification:** C10,C14,C22

# 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

---

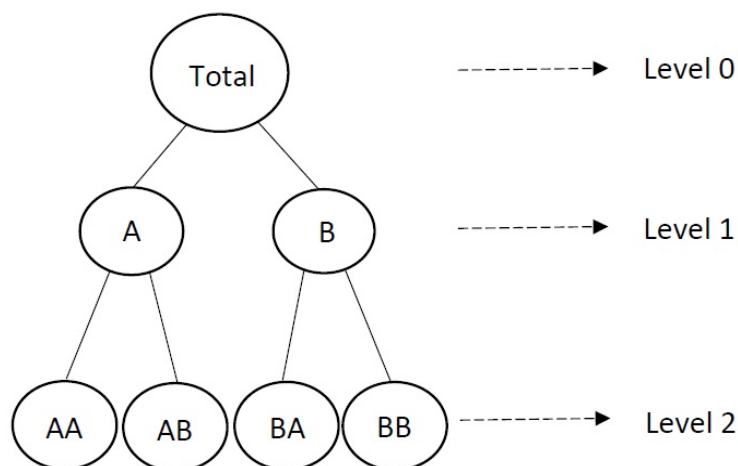
## 1 Introduction

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 AA and AB. Finally, the bottom level series (level 2), includes the most disaggregated series.



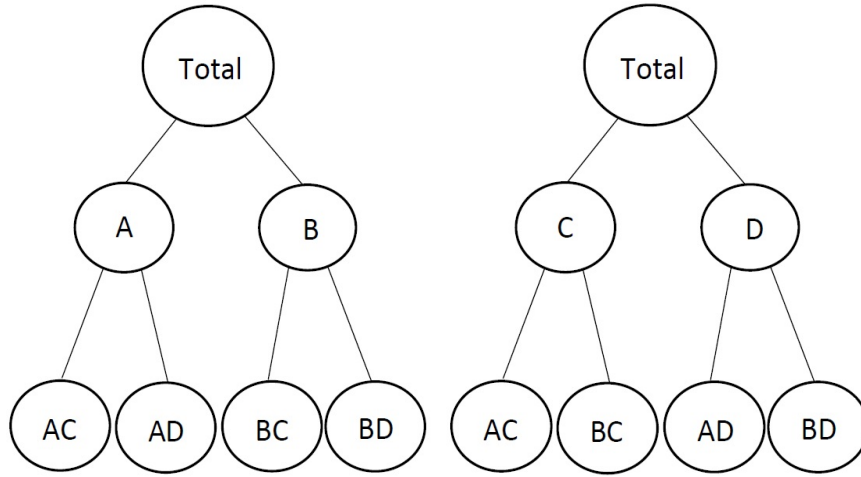
**Figure 1:** An example of a two level hierarchy structure

I suggest we label Level 2 with AW, AX, BY, BZ to avoid confusion.

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  $Z$  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  $y_t = S b_t$ , where  $y_t$  is a vector of all the level nodes at time



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

$y_t$  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 2018)

$$\tilde{y}_h = S(S'W_h^{-1}S)^{-1}S'W_h^{-1}\hat{y}_h, \quad (1)$$

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

The most difficult task is to compute  $W_h$ , but Wickramasuriya, Athanasopoulos & Hyndman (2018) and Hyndman, Lee & Wang (2016) argue that replacing it by the diagonal of  $W_1$  gives good results in practice. This is easy to obtain because it is simply the diagonal matrix comprising the residual variances from each of the base forecasts.

The most computationally challenging part of the optimal combination method is to produce all the base forecasts that make up  $\hat{y}_h$ . 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 selected by minimizing Akaike's Information Criterion (Akaike 1998). This computational challenge 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

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

$$y_{t,i} = \beta_i' x_{t,i} + \varepsilon_{t,i}$$

where  $x_{t,i} = \{1, x_{t,i,1}, \dots, x_{t,i,p}\}$  is a  $(p+1)$ -vector of regression variables, and  $\varepsilon_{t,i} \sim \text{NID}(0, \sigma_i^2)$ .

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

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} X_1 & 0 & 0 & \dots & 0 \\ 0 & X_2 & 0 & \dots & 0 \\ 0 & 0 & X_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & X_N \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_N \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_N \end{pmatrix}, \quad (2)$$

where  $y_i = \{y_{1,i}, y_{2,i}, \dots, y_{T,i}\}$  is a  $T$ -vector,  $\beta_i = \{\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \dots, \beta_{p,i}\}$  is a  $(p+1)$ -vector,  $\varepsilon_i = \{\varepsilon_{1,i}, \varepsilon_{2,i}, \dots, \varepsilon_{T,i}\}$  is a  $T$ -vector and  $X_i$  is the  $T \times (p+1)$ -matrix

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

Equation (2) can be written as

$$Y = XB + E,$$

with parameter estimates given by

$$\hat{B} = (X'X)^{-1}X'Y.$$

Then the base forecasts are obtained using

$$\hat{y}_{t+h} = X_{t+h}^* \hat{B}, \quad (3)$$

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

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

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

The reconciliation (of the WLS variety), using Equation (1) can be written as

$$\tilde{\mathbf{y}}_{t+h} = (\mathbf{S}'\mathbf{W}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}\hat{\mathbf{y}}_{t+h}, \quad (4)$$

where  $\mathbf{W}$  is an  $N \times N$  diagonal matrix with  $(i, i)$ th element  $\sigma_i^2$ , and  $\mathbf{S}$  is the summation matrix containing the aggregation constraints.

Finally, we can combine the two linear equations for computing base forecasts and reconciled forecasts (Equations (3) and (4)) to obtain the reconciled forecasts with a single equation:

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

## 2.1 Proposed approach with same set of $\mathbf{X}$

If we have same set of regression variables,  $\mathbf{X}$ , for all the series, we can write the above equations 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 (2) can be specified as follows:

$$\begin{pmatrix} y_{11} & y_{12} & \dots & y_{1N} \\ y_{21} & y_{22} & \dots & y_{2N} \\ \vdots & \vdots & & \vdots \\ y_{T1} & y_{T2} & \dots & y_{TN} \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{T1} & X_{T2} & \dots & X_{Tp} \end{pmatrix} \begin{pmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0N} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1N} \\ \vdots & \vdots & & \vdots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pN} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1N} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2N} \\ \vdots & \vdots & & \vdots \\ \varepsilon_{T1} & \varepsilon_{T2} & \dots & \varepsilon_{TN} \end{pmatrix} \quad (5)$$

We can represent the Equation (5) as follows

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}, \quad (6)$$

where  $\mathbf{Y}$ ,  $\mathbf{X}$ ,  $\mathbf{B}$  and  $\mathbf{E}$  are  $T \times N$ -matrix,  $T \times (p+1)$ -matrix,  $(p+1) \times N$ -matrix and  $T \times N$ -matrix, respectively. Using Equation (6), parameter estimates are given by

$$\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

Then the base forecasts can be written as

$$\begin{pmatrix} \hat{y}_{t+1,1} & \hat{y}_{t+1,2} & \dots & \hat{y}_{t+1,N} \\ \hat{y}_{t+2,1} & \hat{y}_{t+2,2} & \dots & \hat{y}_{t+2,N} \\ \vdots & \vdots & & \vdots \\ \hat{y}_{t+h,1} & \hat{y}_{t+h,2} & \dots & \hat{y}_{t+h,N} \end{pmatrix} = \begin{pmatrix} 1 & X_{t+1,1} & X_{t+1,2} & \dots & X_{t+1,p} \\ 1 & X_{t+2,1} & X_{t+2,2} & \dots & X_{t+2,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{t+h,1} & X_{t+h,2} & \dots & X_{t+h,p} \end{pmatrix} \begin{pmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0N} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1N} \\ \vdots & \vdots & & \vdots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pN} \end{pmatrix}$$

We can write the above equation in shorter term as

$$\hat{\mathbf{Y}} = \mathbf{X}^*\hat{\mathbf{B}} \quad (7)$$

Then the reconciliation for all the forecasted points,  $h$ , and all the series,  $N$ , can be written as

$$\tilde{\mathbf{Y}} = (\mathbf{S}'\mathbf{W}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}\hat{\mathbf{Y}} \quad (8)$$



Where again  $W$  is an  $N \times N$  diagonal matrix with  $(i, i)$ th element  $\sigma_i^2$ , and  $S$  is the summation matrix containing the aggregation constraints. Also in the above equation,  $\tilde{Y}$  includes all the reconciled forecasts for all the bottom level series.

At the end, we can combine two linear equations for computing base forecasts and reconciliation (Equations (7) and (8)) to obtain the reconciled forecasts with a single equation

$$\tilde{Y} = (S'WS)^{-1}S'W(X^*\hat{B}) = (S'WS)^{-1}S'W(X^*(X'X)^{-1}X'Y)$$

## 2.2 OLS predictors

As an example of the  $X_t$  matrix in Equation (2), 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 as a set of predictors in the linear model. Equation (9) shows a linear equation of this type which models linear trend, additive seasonality with  $m$  seasons, autocorrelation and external data. Here,  $t$  is the running index ( $t = 1, 2, \dots$ ), and  $s_{jt}$  is a dummy variable taking value 1 if time  $t$  ( $j = 1, 2, \dots, m$ ) is in season  $j$ ,  $y_{t-k}$  is the  $k$ th lagged value for  $y_t$  and  $z_t$  is the external data. For instance, if we have daily data with day of week seasonality,  $m$  would be 7 (6 seasonal dummies and 7 the violation of lags).

This could cause problems with seasonality modelled with both dummies and an AR process. Perhaps use fewer lags than  $m$ ?

$$y_t = \alpha_0 + \alpha_1 t + \beta_1 s_{1,t} + \beta_2 s_{2,t} + \dots + \beta_{m-1} s_{m-1,t} + \gamma_1 y_{t-1} + \gamma_2 y_{t-2} + \dots + \gamma_m y_{t-m} + \delta z_t + \varepsilon_t. \quad (9)$$

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.

## 3 Applications

In this section we illustrate our approach two examples, forecasting monthly Australian domestic tourism and forecasting daily Wikipedia pageviews. We compare the forecasting accuracy

levels 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 Error (RMSE) across all series and also display box and density plots for forecast errors along with the raw forecast errors.

Since we are using time series lags  $(1, \dots, m)$  in the linear forecasting model, we can't forecast multiple steps ahead. We therefore apply two methods for generating  $h$ -step-ahead forecast: In the first model we use 1-step-ahead forecasts and for forecasting the following periods  $(t + 2, t + 3, \dots)$  we replace the previous periods with the actual values. This value is known to us because it is in the test set. In our applications, we call this approach '1-step-ahead'. In the second method, we again use 1-step-ahead forecasts but for forecasting the following periods we use the earlier forecasted values. In our applications, we call this approach ' $h$ -step-ahead forecast'. We also show the computation challenges in all the methods.

### 3.1 Australian domestic tourism

This dataset has 19 years of monthly visitor nights in Australia by Australian tourists. This measure is used as an indicator of tourism activity (Wickramasuriya, Athanasopoulos & Hyndman 2018). This data were collected by computer-assisted telephone interviews with 120000, Australians aged 15 and up (Research tourism Australia 2005). In total this dataset includes 304 time series with length 228 each. The hierarchy and grouping structure for this dataset is made using geographic and purpose of travel information.

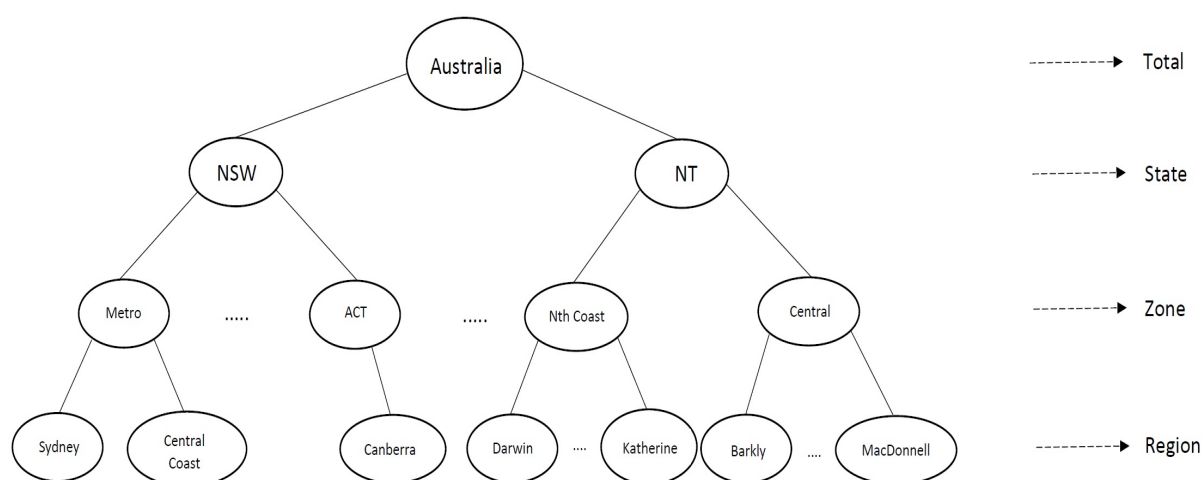
In this dataset we have three levels of geographic divisions in Australia. In the first level, Australia was divided into seven 'States' including New South Wales (NSW), Victoria (VIC), Queensland (QLD), South Australia (SA), Western Australia (WA), Tasmania (TAS) and Northern Territory (NT). In the 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). For 'Purpose' we have four groups: Holiday (Hol), Visiting (Vis), Business (Bis) and Others (Oth). Based on geographic hierarchy and purpose grouping, we end up with 8 levels of hierarchy with 555 series in total. The hierarchy structure which is used in this example includes the following levels:

- Level 0 = Total series
- Level 1 = State
- Level 2 = Zone
- Level 3 = Region
- Level 4 = Purpose
- Level 5 = State  $\times$  Purpose

- Level 6 = Zone  $\times$  Purpose
- Level 7 = bottom level series

We report the forecast results for all these hierarchy levels, as well as the average RMSE across all the levels of hierarchy.

In the predictor matrix, for the OLS forecasting model we apply linear trend, 11 dummy variables, and 12 time series lags<sup>1</sup>. This is intended to capture the monthly seasonality. In addition, before running the model, we partition the data into two parts, training and test sets. We keep the last 24 months periods (2 years) as our test set to forecast and we use the rest as our training set.



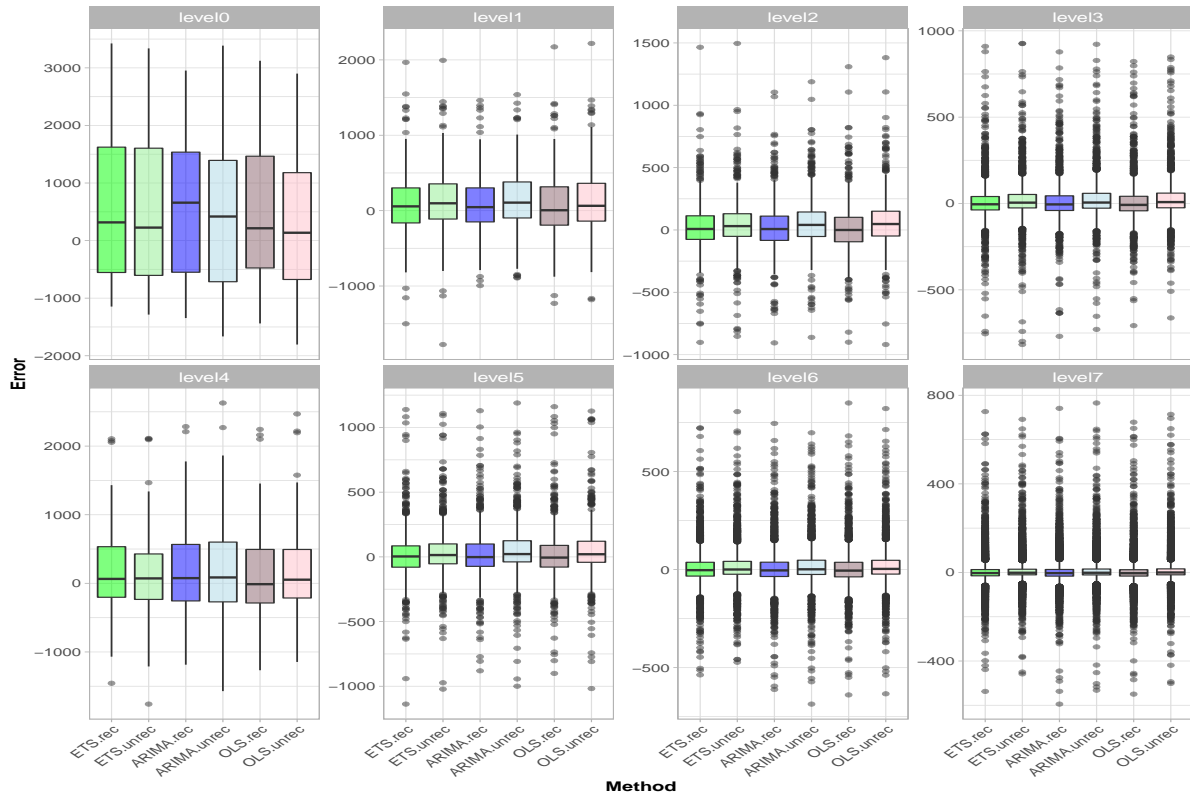
**Figure 3:** *Australia geographic hierarchy structure*

In Tables 3, 4 and 5, we have the average RMSE and computation time for the 24-month forecast period. Methods include ETS, ARIMA and our proposed OLS forecasting model. In Table 3 we forecast 24 periods by computing 1-step-ahead forecasts and rolling forward month by month. In Table 4 we generated 24-step-ahead forecasts. In these tables we have two parts related to the forecast, with and without reconciliation.

The results in Table 3 and 4 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 hierarchy levels.

<sup>1</sup>Since the forecasting results are better with the less number of lags, we use a linear trend and dummy seasonality variables and four lags in our linear model for 24-step-ahead model.

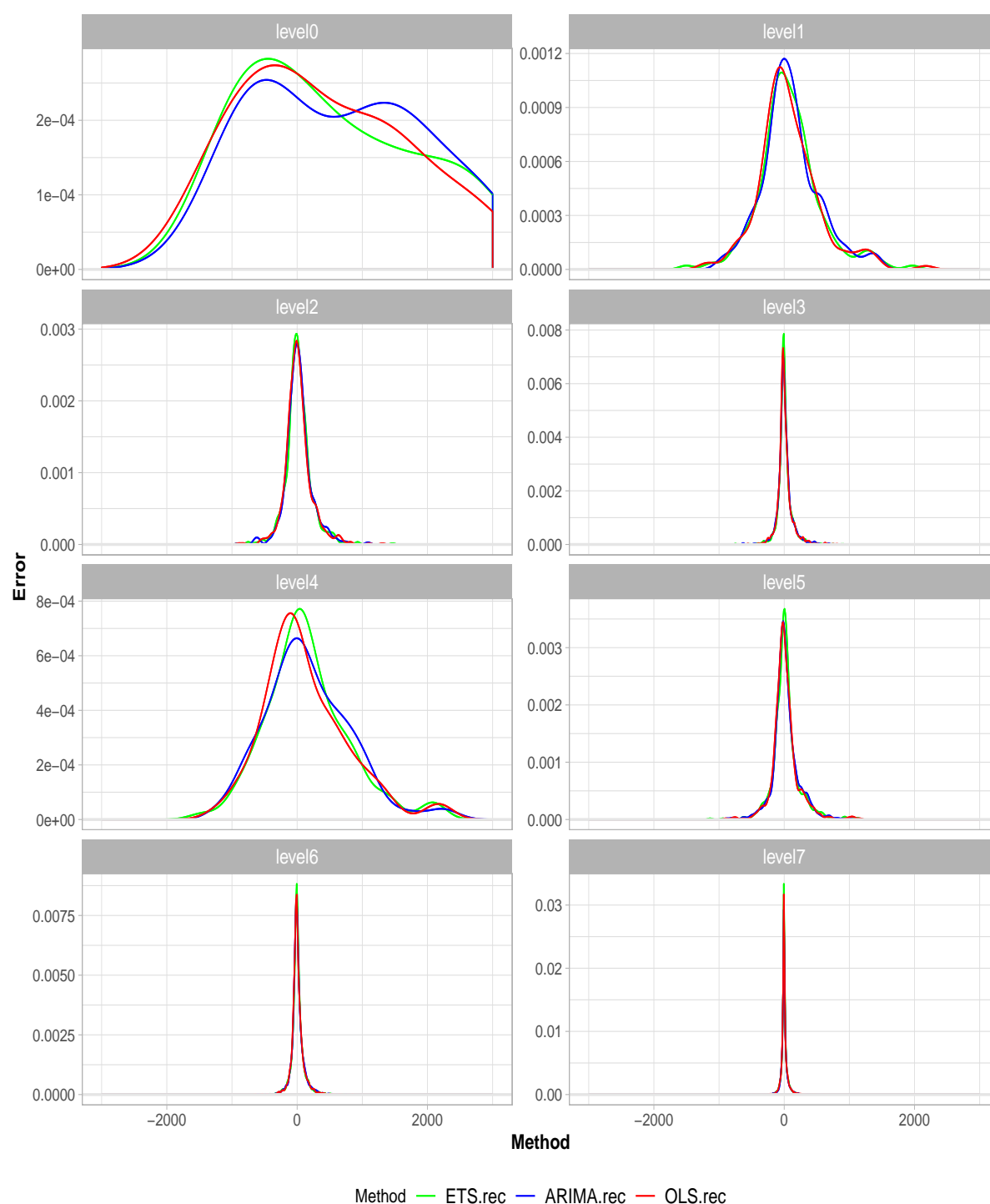
In Figures 4, 6, 5 and 7 we display the error box plots for both reconciled and unreconciled forecasts, and error density plots for reconciled forecasts using all three methods, for 1-step-ahead and 24-step-ahead. In all these figures we see the error distribution similarity across all the models, as well as usefulness of the reconciliation step in improving the forecasts. By comparing density plots 5 and 7, as could be expected, we see that by applying 1-step-ahead forecasts, the error densities are closer and more distributed around zero.



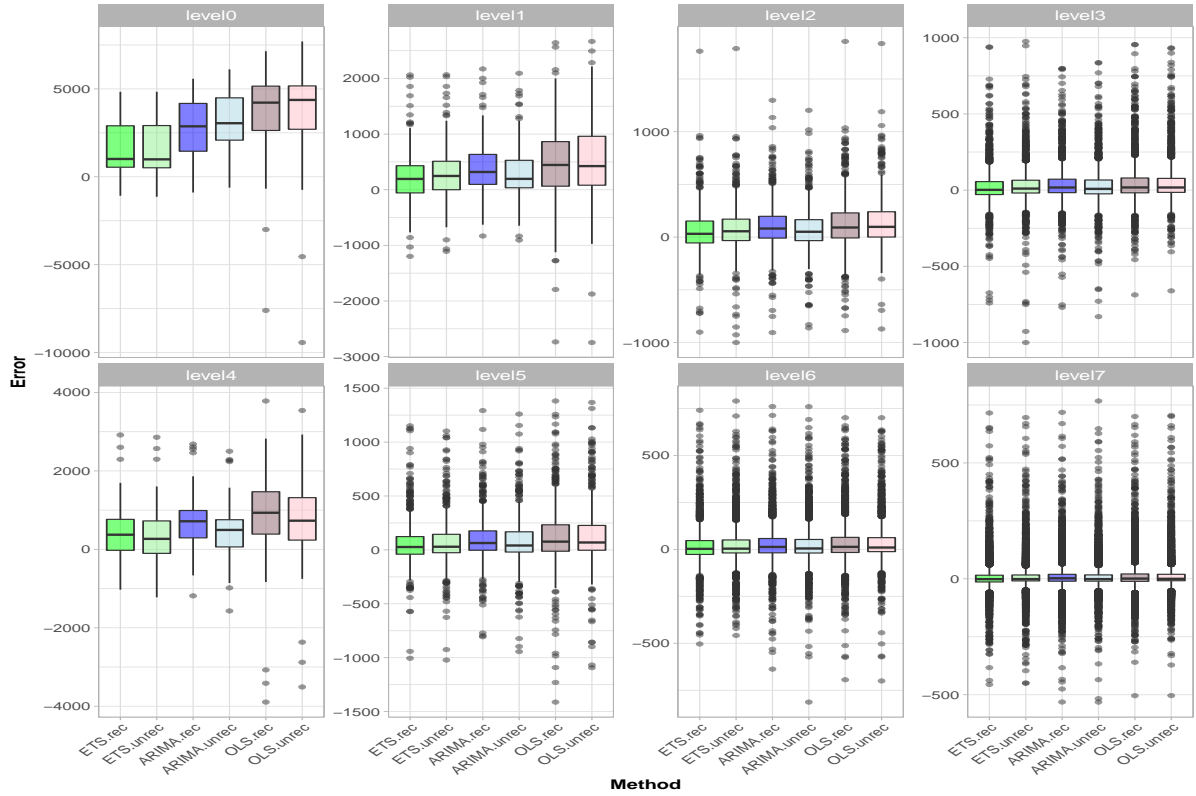
**Figure 4:** Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead tourism demand

Table 5 compares the three methods, computation time for 1-step-ahead and 24-step-ahead forecasting. We see that the OLS forecasting model is much faster compared with the other methods. Also, since reconciliation is a linear process, in all methods, it is very fast and does not affect computation time significantly.

Now since we are using linear model for forecasting, in this part we are including the ‘easter’ information to check its effect on forecasting results. We also add this information on ARIMA models and compare with OLS forecasting model. In Tables 6 and 7, we display the average RMSE of ARIMA and OLS including the easter information, ARIMAX and OLSX, across different levels with and without reconciliation. These tables are for 1-step-ahead and 24-step-ahead forecasts. Figure 8 shows the 1-step-ahead and 24-step-ahead forecast results for one of the



**Figure 5:** Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead tourism demand using interval  $(-3000, 3000)$

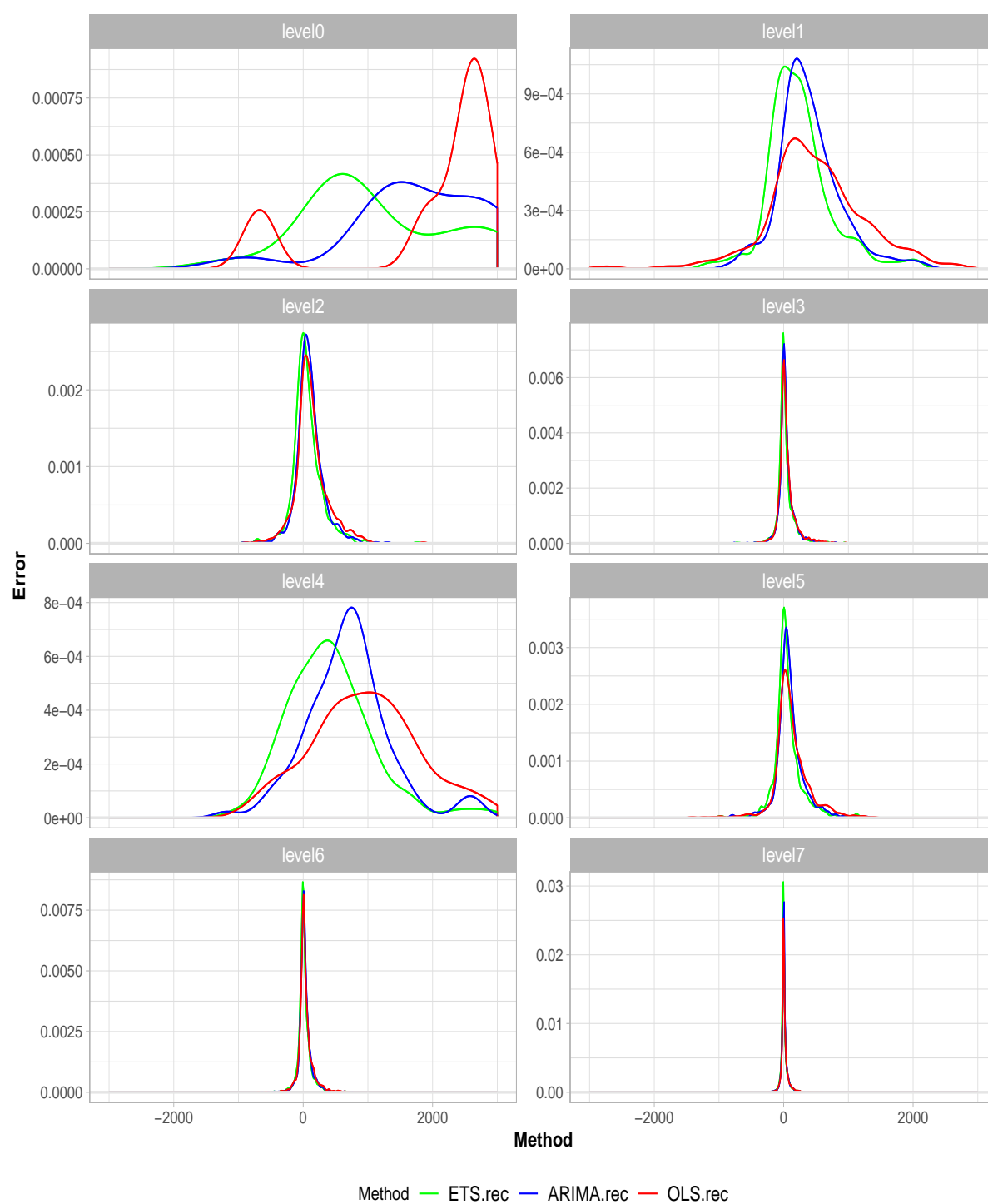


**Figure 6:** Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 24-step-ahead tourism demand

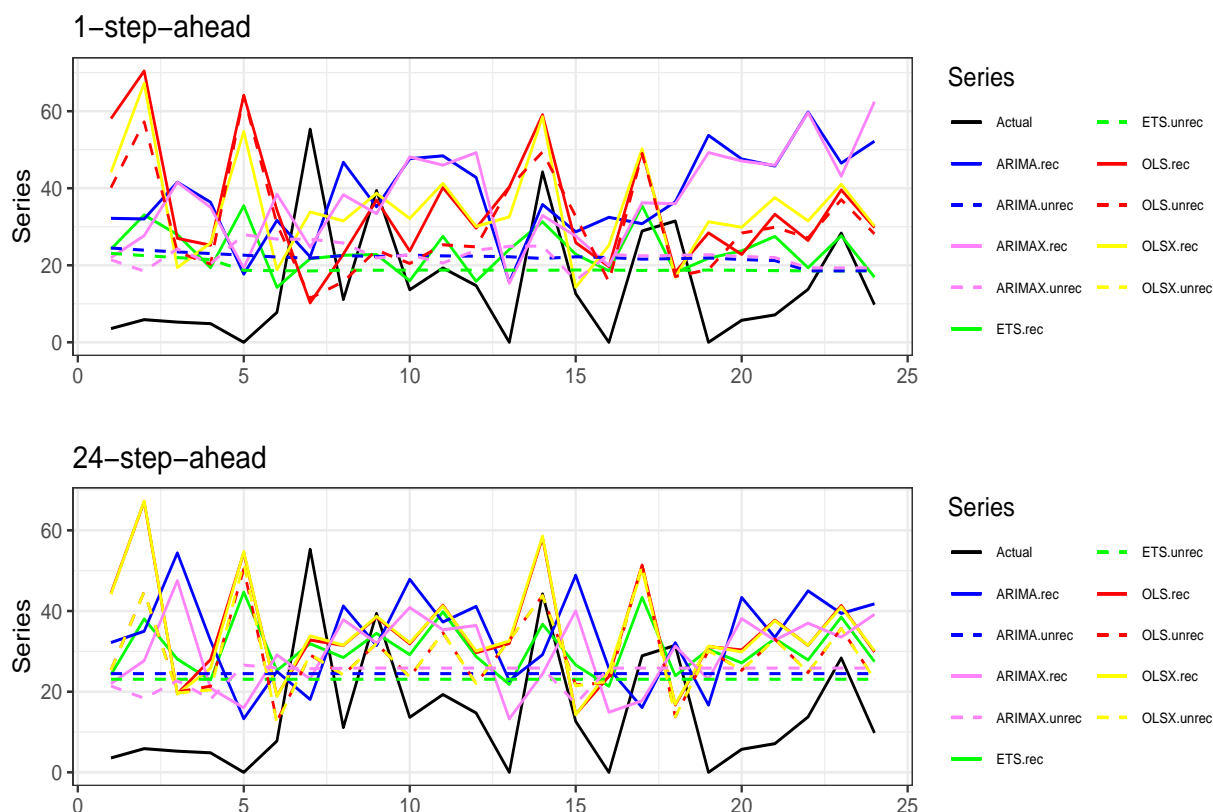
bottom level series, BACBus (Geelong - Business). In these plots we have both reconciled (solid lines) and unreconciled (dashed lines) forecasts and we see that the reconciliation step improves the forecasts in this series. We also see that the OLS model forecast accuracy is similar to the other two methods. As you can see from the results adding this external data could not change the forecasting results significantly. However in different cases, trying different external data can be helpful in improving the forecasting results.

### 3.2 Wikipedia pageviews

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 8). We display the group structure in Table 8 and Figure 9. In Figure 9 we use one possible hierarchy for this dataset, but the order of the



**Figure 7:** Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 24-step-ahead tourism demand using interval  $(-3000, 3000)$



**Figure 8:** Comparing Actual test set, Reconciled and unreconciled ETS, ARIMA and OLS for BACBus bottom level series for 1-step-ahead and 24-step-ahead tourism demand

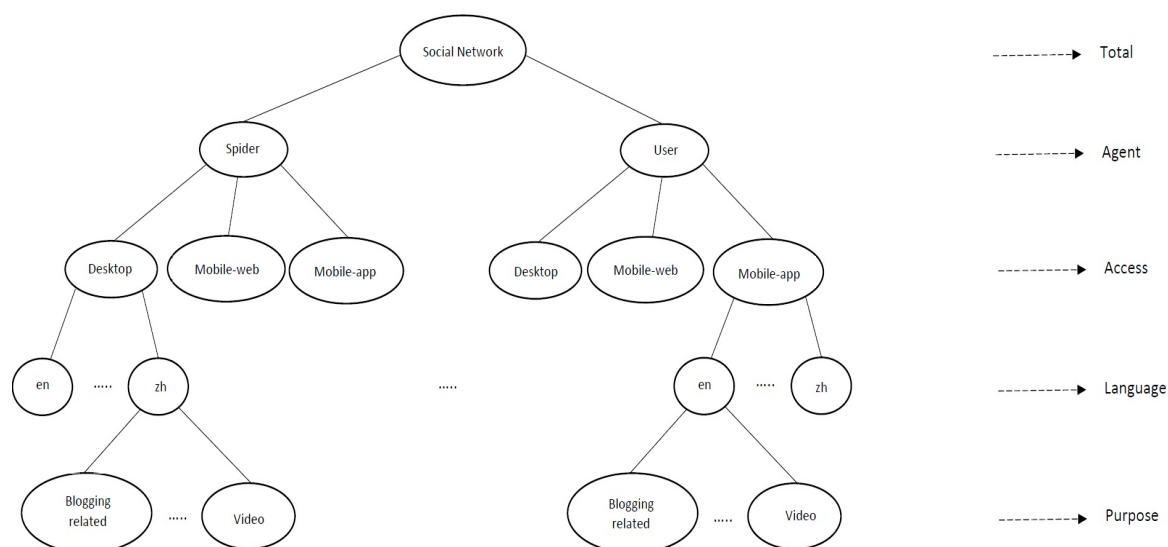
hierarchy can be switched. The final dataset includes 913 time series, each with length 394. The group structure and different levels include:

- 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 9, 10 and 11 represent the RMSE results and computation time. Although these time series are noisier, still we 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 in the forecasted errors.





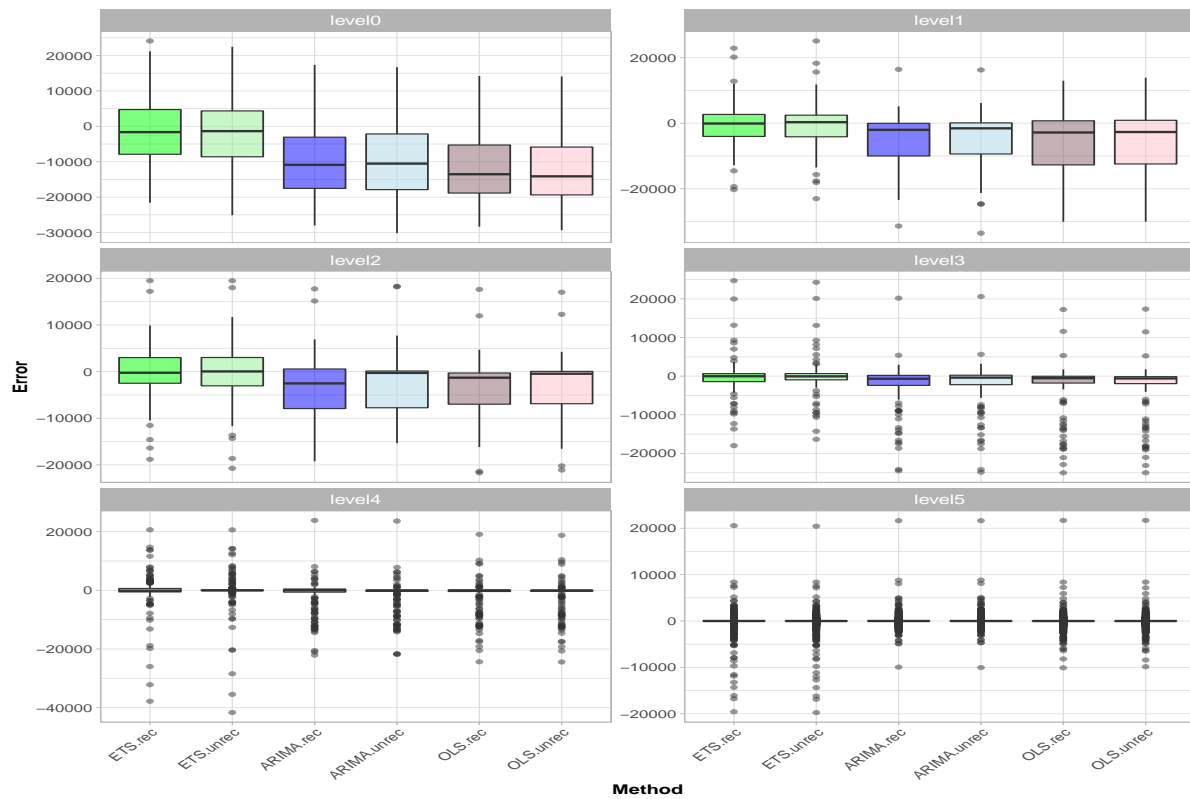
**Figure 9:** One of the possible hierarchy structures for Wikipedia pageview dataset

Figures 10 and 12 display the forecast error box plot. These plots are for 1-step-ahead and 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.

Figures 11 and 13 show the density plots for the forecast errors. For both 1-step-ahead and 28-step-ahead forecasts, we can see the density structure of the forecast errors across ETS, ARIMA and the OLS forecasting model. Except for the Total series which ETS works better, in all the other levels are the models have similar structure for the forecast errors.

In Figure 14, we display results for one of the bottom level series, desktopusenPho (desktop-user-english-photo sharing). The plot shows 1-step-ahead and 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.

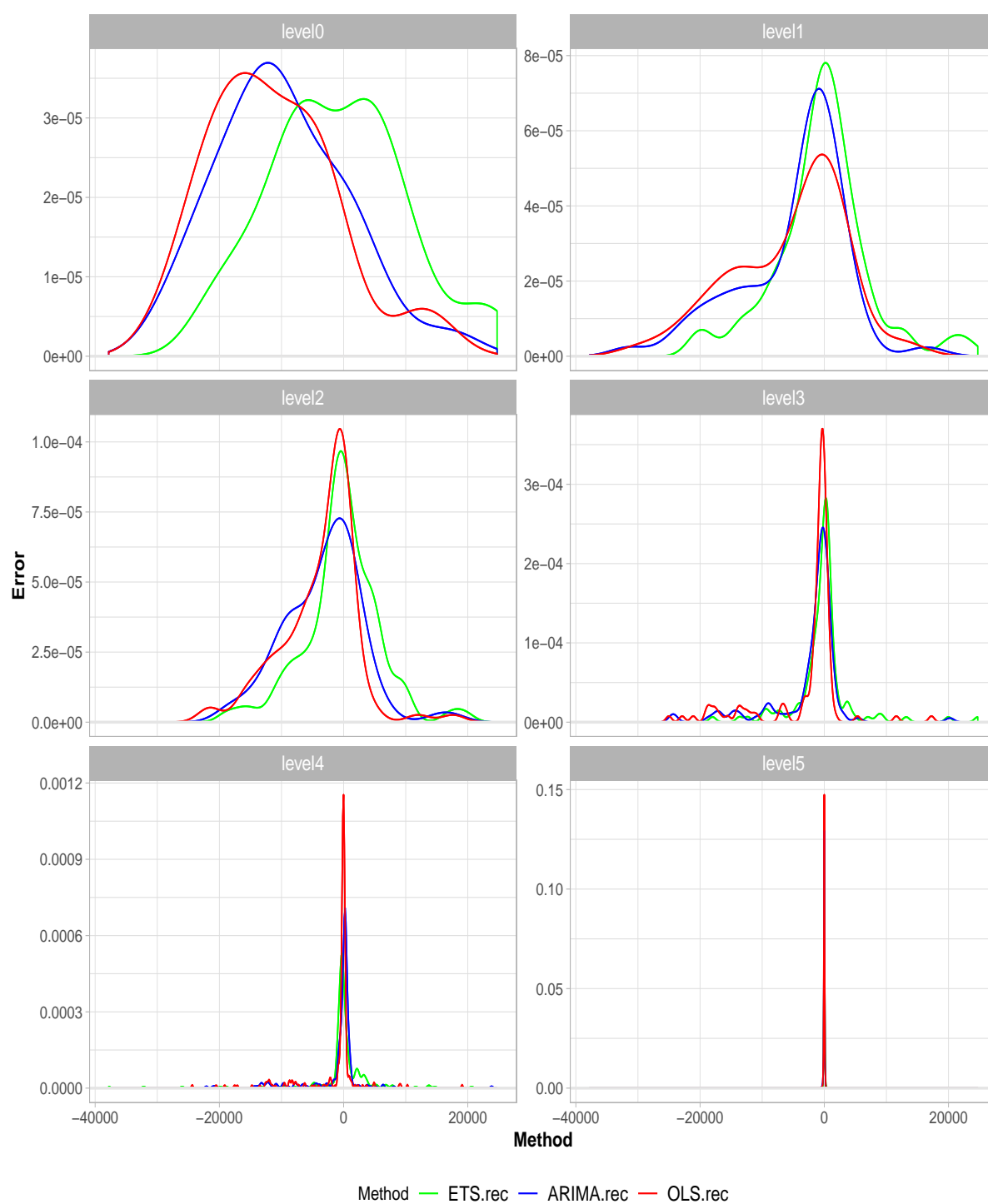
Lastly, Table 11 presents the computation times for all three methods. ETS and ARIMA are clearly much more computationally heavy compared with OLS. As in the Australian tourism dataset, running reconciliation does not have much effect on computation time.



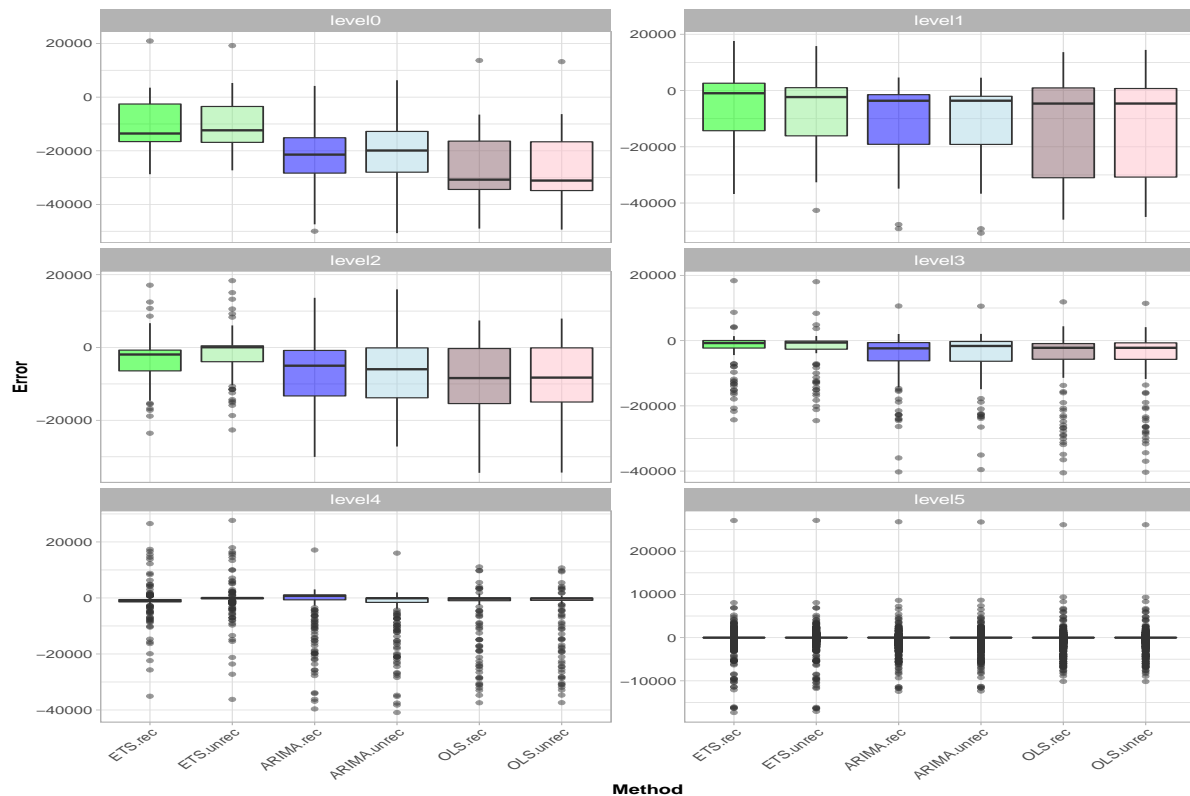
**Figure 10:** Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead Wikipedia pageviews

## 4 Conclusion

We proposed a single-step linear 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 good in terms of forecasting power and accuracy, they can be computationally heavy when facing large collections of time series in 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 ETS and ARIMA in terms of forecasting accuracy level. We also note that OLS has the additional practice feature in handling missing data while ETS and ARIMA requires imputation. One more important feature of our model is the ability to easily include external information such as holiday dummies or other external series. In addition to the computation adjustment, our proposed approach forecasts hierarchical time series in a parsimonious single step whereas other available methods all forecast in two-steps.



**Figure 11:** Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead Wikipedia pageviews



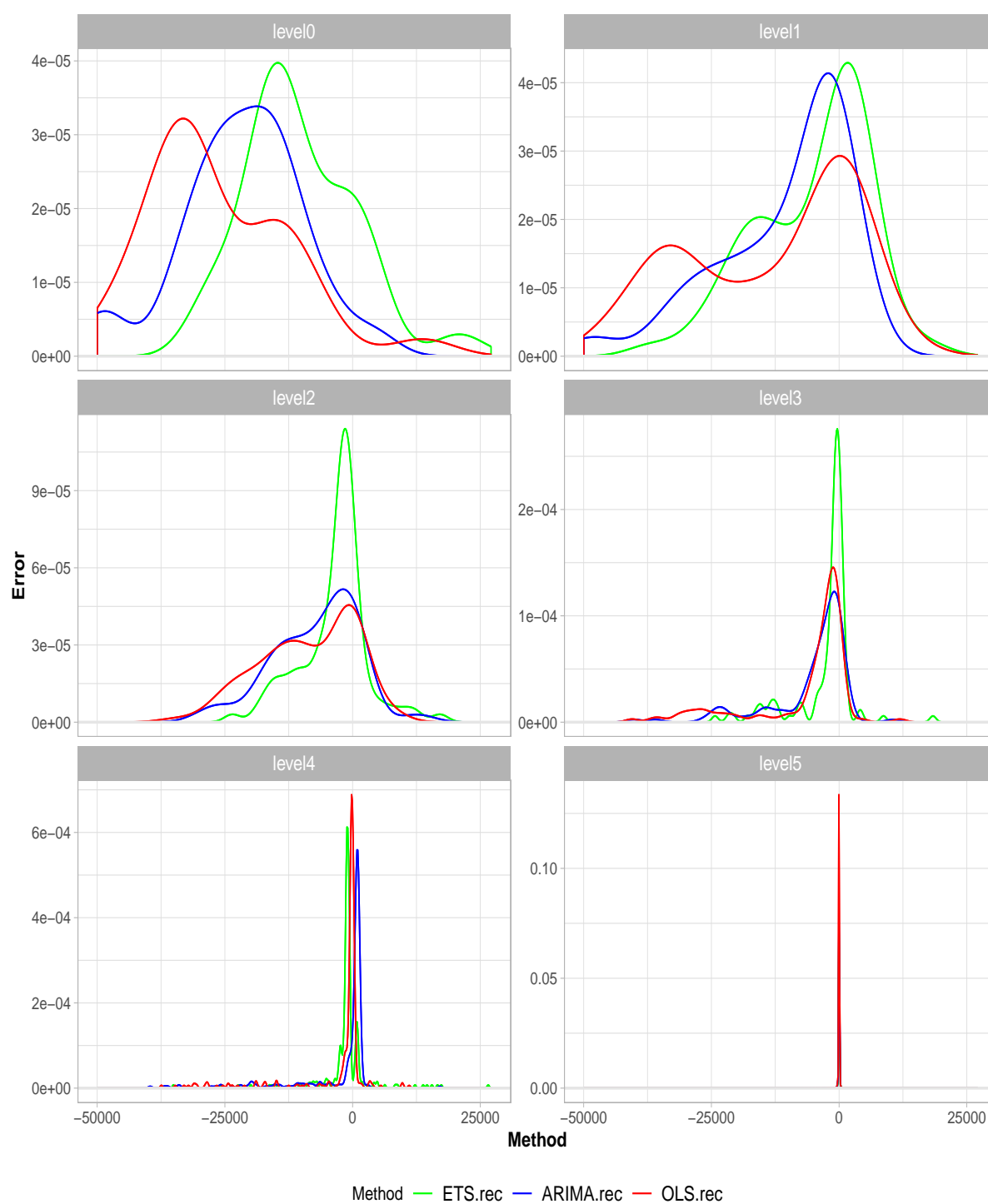
**Figure 12:** Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 28-step-ahead Wikipedia pageviews

## 5 Acknowledgements

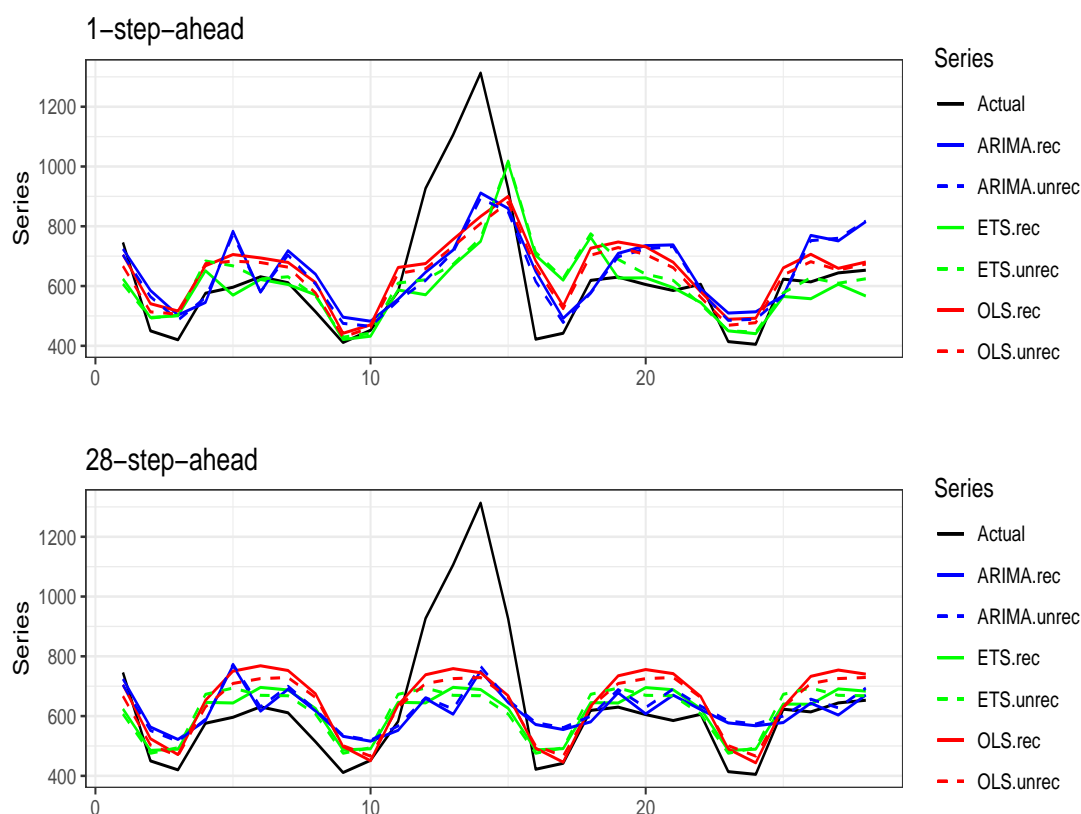
The first and third authors of this research were partially funded by Ministry of Science and Technology (MOST), Taiwan [Grant 106-2420-H-007-019].

## 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.
- Australia, TR (2005). Travel by Australians, September Quarter 2005. *Tourism Australia*.
- Fliedner, G (2001). Hierarchical forecasting: issues and use guidelines. *Industrial Management & Data Systems* **101**(1), 5–12.



**Figure 13:** Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 28-step-ahead Wikipedia pageviews



**Figure 14:** Comparing Actual test set, Reconciled and unreconciled ETS, ARIMA and OLS for desktop-user-english-photo sharing bottom level series- 1- and 28-step-ahead Wikipedia pageviews

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>.

Hyndman, RJ, AJ Lee & E Wang (2016). Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis* 97, 16–32.

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

Wickramasuriya, SL, G Athanasopoulos & RJ Hyndman (2018). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association* (just-accepted), 1–45.

**Table 1:** *Australia geographic hierarchy 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

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

geographic division	# of series (geographic division)	# of 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

**Table 3:** Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 1-step-ahead - Tourism dataset

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	1516.40	1445.49	1415.06	1533.58	1453.44	1454.39
Level 1	511.37	493.14	510.83	495.88	457.65	488.33
Level 2	214.81	219.01	224.50	209.16	207.52	212.44
Level 3	122.91	125.08	123.97	118.67	120.52	119.52
Level 4	675.99	709.22	694.50	668.26	679.74	678.54
Level 5	213.06	220.08	216.11	210.64	209.39	211.13
Level 6	97.53	102.41	101.03	96.36	99.77	98.56
Level 7	56.17	58.20	58.17	55.98	57.68	57.20

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

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	2238.58	3553.99	5050.33	2250.22	3179.39	4671.05
Level 1	593.57	570.13	900.10	553.76	626.32	902.99
Level 2	239.52	229.64	278.84	234.21	242.46	290.95
Level 3	132.58	129.40	142.78	126.74	129.40	146.32
Level 4	766.78	824.00	1306.97	795.48	958.24	1434.05
Level 5	226.74	241.18	298.24	222.48	236.94	308.39
Level 6	103.02	105.38	110.91	101.95	103.93	115.35
Level 7	59.12	58.81	60.90	58.54	58.71	62.64



**Table 5:** *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - 1- and 24-step-ahead - Tourism dataset*

	Computation time (secs)			
	1-step-ahead		24-step-ahead	
	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

**Table 6:** *Mean(RMSE) for ARIMAX and OLSX, adding easter information, with and without reconciliation - 1-step-ahead - Tourism dataset*

	Mean(RMSE)							
	Unreconciled				Reconciled			
	ARIMA	OLS	ARIMAX	OLSX	ARIMA	OLS	ARIMAX	OLSX
Level 0	1445.49	1415.06	1564.95	1444.49	1453.44	1454.39	1546.53	1487.23
Level 1	493.14	510.83	500.66	514.29	457.65	488.33	472.03	492.68
Level 2	219.01	224.50	220.91	225.82	207.52	212.44	210.47	213.48
Level 3	125.08	123.97	125.85	123.69	120.52	119.52	121.02	119.44
Level 4	709.22	694.50	702.78	680.56	679.74	678.54	682.92	662.45
Level 5	220.08	216.11	222.07	215.15	209.39	211.13	211.45	209.55
Level 6	102.41	101.03	103.04	100.93	99.77	98.56	100.53	98.49
Level 7	58.20	58.17	58.60	58.05	57.68	57.20	58.04	57.15

**Table 7:** *Mean(RMSE) for ARIMAX and OLSX, adding easter information, with and without reconciliation - 24-step-ahead - Tourism dataset*

	Mean(RMSE)							
	Unreconciled				Reconciled			
	ARIMA	OLS	ARIMAX	OLSX	ARIMA	OLS	ARIMAX	OLSX
Level 0	3553.99	5050.33	3528.13	5032.41	3179.39	4671.05	3114.40	4677.99
Level 1	570.13	900.10	509.84	910.73	626.32	902.99	603.66	907.33
Level 2	229.64	278.84	235.06	280.06	242.46	290.95	243.76	291.62
Level 3	129.40	142.78	129.39	143.66	129.40	146.32	129.63	147.19
Level 4	824.00	1306.97	811.08	1324.70	958.24	1434.05	930.72	1444.75
Level 5	241.18	298.24	232.88	304.55	236.94	308.39	229.59	313.48
Level 6	105.38	110.91	104.73	111.85	103.93	115.35	104.50	116.28
Level 7	58.81	60.90	59.22	61.41	58.71	62.64	59.02	63.12

**Table 8:** Social networking Wikipedia article grouping structure

Series	Name	Series	Name
Total		Language	
1	Social Network	10	zh (Chinese)
Agent		Purpose	
2	Spider	11	Bloggng 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)		

**Table 9:** Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 1-step-ahead - Wikipedia dataset

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	10773.66	15060.65	15748.18	11014.73	14276.47	15270.23
Level 1	8272.92	10196.34	10623.85	7736.88	9904.12	10673.98
Level 2	6524.72	6705.03	6979.58	6257.44	7142.49	7285.97
Level 3	4870.08	6333.02	7150.13	4981.91	6369.98	7106.11
Level 4	5233.50	4659.53	4675.18	5001.40	4586.53	4650.26
Level 5	358.90	238.97	254.98	362.25	241.60	256.11

**Table 10:** Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 28-step-ahead - Wikipedia dataset

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	14846.93	24298.84	29840.58	14999.18	24649.91	29665.70
Level 1	13608.73	17277.01	21165.30	12240.30	16810.45	21048.06
Level 2	7117.43	10731.97	12678.89	7523.43	11068.81	12811.18
Level 3	6475.90	9580.38	12056.62	6509.03	9799.11	12112.46
Level 4	5302.74	8611.25	8451.09	5307.34	8239.77	8460.35
Level 5	435.64	390.05	389.41	437.67	391.22	390.97

**Table 11:** *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - 1- and 28-step-ahead - Wikipedia dataset*

	Computation time (secs)			
	1-step-ahead		28-step-ahead	
	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