MONASH University

MONASH BUSINESS SCHOOL

Progress Review Research Report

Tools for forecasting large collections of time series

Mitchell O'Hara-Wild

Supervised by Rob Hyndman and George Athanasopoulos

24 February 2025

Background and motivation

Large collections of data are collected across all industries, and with the growing use of IoT sensors and other scalable data collection processes more time series data is available than ever. The scale of this data collection is increasing both in the frequency of observations, and the number of things being measured. Making sense of this data can be challenging for a multitude of reasons, and widely used time series analysis software is unsuitable for the task. Measuring data at a finer temporal and cross-sectional granularity exposes more nuanced patterns that require more flexible models for forecasting. More complex cross-sectional relationships between time series are emerging, necessitating new approaches for encoding the coherence structure of the collection. A complete hierarchy of time series data with many disaggregating attributes can be computationally expensive to forecast, and since the majority of time series contain little forecast-able information the forecast accuracy for series of interest can worsen. Another complication in modern time series analysis is the collation and analysis of data from multiple sources which are often measured at different temporal granularities.

My research aims to ease these difficulties by developing new tools and methodology for flexibly forecasting these series across all temporal and cross-sectional granularities.

Thesis overview

My proposed research consolidates many aspects of time series analysis and forecasting into a cohesive and unified framework. Bringing together many disparate concepts allows researchers and practitioners to use these methods in new ways that works best for their needs. This work involves finding the common themes in time series analysis and research to design simple interfaces that work well together in combination to provide flexible analysis and modelling workflows. A focused theme of the thesis is forecast reconciliation, however much of the contributions are foundational with applications that reaching beyond coherent forecasting. A summary of how the thesis topics outlined below relate are as follows:

- Topic 1: *Cross-sectional coherency constraints*

  Graphs flexibly describe cross-sectional relationships between time series.

- Topic 2: *Overcoming too many series in a collection*

  Pruning graphs to remove uninformative time series can both improve forecasting accuracy and computation time.

- Topic 3: *Representing probabilistic forecasts*

  Vectorised distributions for use in a tidy forecasting workflow to adequately describe forecast uncertainty.

- Topic 4: *Temporal coherency constraints*

  Representing time with varied temporal granularities in a tidy time series data structure.

- Topic 5: *Grammar of temporal graphics*

  Extends the grammar of graphics to support calendar-based temporal visualisation.

- Topic 6: *Tidy forecasting framework*

  This software contribution combines the foundational tools described above to support a tidy forecasting workflow. The tool is capable of producing probabilistic cross-temporally coherent forecasts for large collections of time series.

A significant output of this work is the translation of research into statistical software for broader impact and practical applications. The design of this software empowers time series practitioners with the flexibility to accurately represent their data with models, and researchers with a framework to rapidly implement and evaluate new methodologies against existing techniques.

Topic 1: Reconciliation of structured time series forecasts with graphs

Accurate forecasts of large collections of time series are critically important to decision makers for the efficient operation of an organisation. These collections of time series are often intrinsically structured for aggregation. Collections of time series are typically related in hierarchical or grouped structures (Hyndman & Athanasopoulos 2021), however more flexibly structured relationships between time series are possible. Forecasting the most aggregated series in the structure is useful for organisational strategy and planning, while the disaggregated forecasts are important for managing local operations. Forecasts of each series from independent models will typically not align with the aggregation structure of the data, and this inconsistency presents an inherit forecast error. Correcting for this structural error presents an opportunity to leverage additional information from other series to produce more accurate and coherent forecasts.

The process of adjusting forecasts to satisfy these aggregation constraints was first introduced by Hyndman et al. (2011). This technique of forecast reconciliation has since been extended to include temporal aggregation (Athanasopoulos et al. 2017b), cross-temporal aggregation (Kourentzes & Athanasopoulos 2019), and improved minimum trace based reconciliation weights (Wickramasuriya, Athanasopoulos & Hyndman 2019). Girolimetto & Di Fonzo (2023) generalise these aggregation constraints beyond interactions of hierarchical, grouped and temporal to include any linear relationship between series. These general linear constraints allows forecast reconciliation techniques to be applied on collections of time series which don't follow the typical 'upper' and 'bottom' classification of series present in hierarchical and grouped structures.

In this chapter I propose an alternative graph-based representation for coherency constraints on a collection of time series. Using directed acyclical graphs to rather than constraint matrices presents several key advantages. Representing constraints with graphs simplifies their construction and enables direct visualisation of the relationship between series via graph visualisation. Using graphs to describe the structure of large collections of related time series also enables improved manipulation tools to remove irrelevant or otherwise unwanted sections of data without disrupting the coherency constraints. Graphs which constrain the parent nodes to be linear combinations of child nodes can be directly converted to general linear constraint matrices, however graph representations also enable the encoding of non-linear relationships.

Topic 2: Forecasting quality over quantity: pruning large collections of coherent time series

Large collections of related time series are commonly structured with aggregation constraints, whereby each series possesses various attributes that identify their relation to other series. These attributes typically relate to what is being measured, such as product categories or store locations for the sales of a product over time. When there exists many attributes for time series data, the number of series in the collection quickly becomes unmanageable with disproportionately many uninformative disaggregated series. This presents many problems for forecasting, since producing many forecasts can be computationally infeasible and the forecast accuracy for aggregated series of interest can worsen (Wang, Hyndman & Wickramasuriya 2024).

To overcome these problems I propose using time series features (Kang, Hyndman & Smith-Miles 2017) to identify noisy, uninformative, or otherwise unwanted series and leveraging the graph structure from topic 1 to safely remove them while preserving coherency constraints. Pruning series from the bottom of the structure would result in graph coherency constraints since a common bottom level is no longer present. Various control points are possible, including specification of features, thresholds, and coherent pruning rules to produce a reduced set of coherent series for forecasting. Pruning subgraphs of time series from the collection can substantially reduce the number of series to forecast, while retaining most of the information. This helps limit the computational complexity of forecasting, while improving forecast accuracy for aggregated series due to reduced model misspecification in more disaggregated series.

Topic 3: Statistical computing with vectorised operations on distributions

The distributional nature of model predictions are often understated, with default output of prediction methods of statistical software usually only producing point predictions (usually the mean of the distribution). Some R packages such as forecast (Hyndman & Khandakar 2008) further emphasise uncertainty by producing point forecasts and intervals by default, however the user's ability to interact with them is limited.

R is a functional programming language that provides many vectorised functions, and the included distribution functions follow this design. The statistic and shape of a distribution is characterised by the name of the function and the function's arguments parameterise the distribution. For example, the cdf/pdf/quantiles/samples from a Normal distribution are obtained using the `dnorm()/pnorm()/qnorm()/rnorm()` functions respectively. The names of these functions are brief and do not clearly describe the statistic being computed from which distribution. There have been many attempts at improving this design which typically represent the distribution as an object containing both the shape and its parameterisation. In R, the distr package (Ruckdeschel & Kohl 2014) and its extensions use S4 classes to represent many common distributions, distr6 (Sonabend & Kiraly 2022) uses R6 classes and (Hayes et al. 2022) uses S3 dispatch methods. The benefit of storing parameterised distributions as objects is that these objects can be used with common functions for regardless of the distribution's shape. These packages are generally designed to work with one distribution at a time, which is useful for teaching but not practical for working with multiple predictions from models.

Vectors of distributions solves these problems, allowing models to directly provide complete distributions for each of the predictions. This vectorised interface for distributions can be built upon the vctrs package (Wickham, Henry & Vaughan 2022), which provides tools for creating new vectorised objects that follow tidyverse design principles. Vectors usually contain objects of the same structure, but for distributions it is valuable that different shapes of distributions can co-exist within the same vector. This enables computation across different types of distributions, which is especially valuable when predicted distributions from multiple models are of a different shape within a tidy rectangular dataset. Working with vectors of distributions allows the calculation of various statistics on predictions from models in extension to the usual outputs such as cdf, pdf, quantiles and generating random numbers. This includes computing point forecasts, intervals, and HDRs (Hyndman 1996); easily evaluating prediction accuracy with continuous ranked probability scores (Matheson & Winkler 1976); and visualising these predictions with uncertainty (Kay 2022). It is also useful to modify distributions, including applying transformations, inflating values, truncating distributions and creating mixtures of distributions; this flexibility is necessary to adequately describe the structure of the data collected. A unified vector-based interface for distributions is important for the statistical software ecosystem, providing a foundation for producing forecasts with different shapes across all levels of temporal and cross-sectional disaggregation.

Topic 4: Reconciling mixed temporal granularities

Time series data is collected at many different frequencies, from event data recorded with millisecond precision to annually reported data that aggregates everything from that year. Existing research and software implementations consider the temporal granularity (or resolution) of data, but are inadequate for an accurate analysis across different temporal granularities. The most common temporal granularities in software are date (ymd) and time (ymd_hms), however it is common for data to be collected less often than daily or more often than secondly. The lubridate R package (Grolemund & Wickham 2011) provides many helpful functions to work with these objects, along with time periods and intervals, but is ultimately restricted by these two granularities. Both tsibble (Wang, Cook & Hyndman 2020) and zoo (Zeileis & Grothendieck 2005) R packages provide monthly and quarterly temporal granularities, but lack the tooling for comparison between points in time of different granularities. This makes it difficult, for example, to identify if the day 2022-10-27 is before/within/after the month 2022-Oct or quarter 2022-Q1.

Mixed temporal granularities can arise for a variety of reasons. You might like to use two sources of data that are observed at different frequencies. Or perhaps the data was previously recorded once a month but is now recorded every day. Mixed temporal granularities also result from temporal aggregation, where you might start with daily data and then compute weekly aggregates from it and use both granularities for forecasting with temporal reconciliation (Athanasopoulos et al. 2017a; Di Fonzo & Girolimetto 2021). Some time series models like MIDAS regression (Andreou, Ghysels & Kourtellos 2011) are designed to forecast with data from mixed temporal granularities and would benefit from improved time classes to structure the model's data.

It is not currently possible to mix temporal granularities within the same dataset or vector, despite the need in many circumstances. As a result, it is common to either use the starting time at the finest common granularity or to aggregate up to the largest common granularity. The first approach now inaccurately represents the observations as a more exact measurement, causing issues with visualisation and modelling. The second approach throws away valuable information. Greater flexibility is needed for representing time, and this research will provide the necessary tools for improving time series visualisation, temporal reconciliation, and mixed granularity analysis.

Topic 5: Grammar of temporal graphics

Effective use of statistical graphics in exploratory time series analysis helps to uncover temporal patterns needed to accurately specify models. While several commonly used plots exist for visualizing time series, little work has been done to formalize them into a unified grammar of temporal graphics. Decomposing traditional time series graphics such as time plots and seasonal plots into modular grammatical elements provides the flexibility needed to clearly visualize multiple seasonality, cycles, and other complex patterns.

Temporal data visualization requires special handling to highlight patterns shaped by calendar systems, much like the nuances of spatial, graph, and uncertainty visualization. The proposed grammar incorporates calendrical concepts to visually align time points at different granularities and timezones, warp time to standardize irregular cyclical durations, and wrap time into hierarchical calendar layouts. Foundational to this grammar is the data layer, which leverages the calendar-based representation of time points provided by the `mixtime` R package developed to enable tidy temporal reconciliation in topic 4. The associated ggtime R package implements this grammar of temporal graphics, which supports combining modular grammatical elements into both familiar and novel visualizations of complex time series patterns.

Topic 6: Probabilistic forecasting at scale using tidy data structures

Modelling in statistical software like R typically provides tools for estimating a single model, and the code for estimating many models is left up to the analyst to implement. This makes simple tasks like comparing one model against another across multiple series cumbersome to compute. A time series dataset usually consists of multiple series, and it is common to ask similar questions about each of these series. For instance, one might wonder how the seasonality differs in each series, or wish to predict each series one year into the future. Existing implementations like the widely popular R package `forecast` (Hyndman & Khandakar 2008) are inadequate for modelling the high frequency and large scale data seen in modern forecasting projects. New methods are needed to support answering these questions across large collections of time series.

Most cross-sectional models in R share a common syntax for specifying models with a symbolic model formulae (Wilkinson & Rogers 1973; Chambers & Hastie 1993). The response variable is declared on the left, and regressors on the right of the formula separator '$\sim$'. Despite conceptual similarity with these models, time series models generally do not use this formula syntax and instead use function arguments to specify models. This obscures the model's mechanism for describing time series patterns, and makes it comparatively difficult to add regressors. Time series models in R often have inconsistent interfaces and return incompatible objects which makes performing common tasks like forecast reconciliation (Panagiotelis et al. 2022) and accuracy evaluation (Hyndman & Koehler 2006) challenging. This research aims to use symbolic model formulas to specify time series models, and standardise how models are estimated across many time series.

The forecast package (Hyndman & Khandakar 2008) is notable for emphasising forecast uncertainty by providing forecast intervals and means by default, where most other models only produce point predictions. Using the vectorised distributions described earlier, this project aims to provide forecast distributions from which intervals and point forecasts can be obtained from. The combination of modelling at scale across many series, the use of vectorised forecast distributions, and the mixed temporal granularity tools makes the design of a general interface for probabilistic cross-temporal forecast reconciliation possible.

This project builds upon the tidy temporal data structures by Wang, Cook & Hyndman (2020), offering new tidyverse compatible (Wickham et al. 2019) tools for exploring, modelling and forecasting time series at scale. The software resulting from this research aims to provide a consistent and flexible interface that is extensible to support new models and methodologies in forecasting. This work incorporates the foundational research in the prior topics in order to facilitate producing probabilistic cross-temporally coherent forecasts for large collections of time series.

Progress review presentation topic

The oral presentation component of my progress review milestone will focus on the second topic of my thesis: pruning large collections of coherent time series. In this talk I will introduce pruning coherency constraints using both time-series and graph features to improve forecasting accuracy. Core to this idea is forecasting quality over quantity, reducing the computational burden of forecasting upwards of billions of time series down to a more feasible subset of informative series (usually thousands). This bucks the current trend of forecasting at scale, where the forecasting priority is computational speed for automatic forecasting.

There exists a dimensionality problem of coherent forecasting at scale, where the number of time series grows exponentially with the depth of disaggregation. Hierarchical and grouped forecast reconciliation strategies require complete disaggregation down to a common bottom level of disaggregated time series. Complete disaggregation across many identifying dimensions results in an excess of uninformative and noisy time series which worsen forecasting accuracy and are useless for decision making. To produce a computationally feasible subset of time series for forecasting, it is common for forecasters to carefully select a subset of dimensions to completely disaggregate by. This introduces a balancing act between computational complexity and coherent forecasting completeness, which ultimately leaves useful information out of the model as a result of limitations in forecast reconciliation methodology.

Graph-based representations of of coherency constraints enable incomplete disaggregation, where a common set of bottom level time series is not required for forecast reconciliation. Graph pruning produces a coherent subset with incomplete disaggregation, where the depth of disaggregation is determined with stopping rules based on user-specified time-series and graph-based features. Time-series features offer useful indications of forecastability, while graph-based features can limit the dimensionality while retaining structurally relevant series. This enables all disaggregating dimensions to be used in forecast reconciliation, allowing reconciliation to use all information from less disaggregated series while removing the computational burden of more disaggregated noisy and uninformative series.

Additional details are available in the associated working paper.

Thesis progression

Statement of progress

The paper for graph coherency constraints (topic 1) is being rewritten with an improved paper structure which emphasises the generality and future opportunities of this reconciliation framework. The theoretical concepts having been tested and verified. The underlying software for representing graph structures in the `graphvec` R package (O'Hara-Wild 2024) has been extended to support both node-first and edge-first graph formats for more general applications. The use of graphs for reconciling linear constraints has been implemented in `fabletools` (O'Hara-Wild, Hyndman & Wang 2024).

The concepts underpinning graph pruning (topic 2) have been refined and tested, demonstrating the scalability of the solution to many practical problems. The implementation has been generalised to support both graph-based and feature-based pruning of related time series in order to produce more accurate and computationally efficient coherent forecasts of time series of interest. The work was presented at the ISF 2024 conference, and has since been further refined to better establish its broad industry applications. The attached work-in-progress paper is nearing completion, pending results from a yet to be confirmed large-scale time series dataset.

The design concepts of vectorised distributions (topic 3) has matured and been developed into the `distributional` R package (O'Hara-Wild et al. 2020). The underlying vectorisation concepts have been generalised into the `vecvec` R package, which will also underpin other vector-based innovations including the `mixtime` R package used in topic 4. The software and underlying design philosophy was presented at the UseR! 2024 conference in Salzburg, Austria. The conceptual design framework of vectorised distributional computations has been structured into a paper format targetting the R Journal. The paper highlights how elements of computer science such as vectorised computation can be applied to distributions, and proposes a consistent approach to safely recycling inputs and providing structured outputs suitable for both univariate and multivariate distributions.

The implementation of representing mixed temporal granularities (topic 4) in a single vector have been further developed, with the calendar system underpinning the special behaviour of these temporal vectors. Core concepts including time, timezones, calendars, seasons, holidays, granularity, durations, and intervals have been explored. Aggregations between temporal granularities are functional in nature, and can be used with the graph reconciliation work (topics 1-3) in order to support temporal reconciliation in full generality.

A new initiative that builds on the `mixtime` R package created for topic 4 is the `ggtime` R package (topic 5), which implements novel calendar-based temporal extensions to the grammar of graphics (Wilkinson 2011). This contribution is analogous to the `ggdist` R package (Kay 2023), which leverages the vectorised distributions (topic 3) from the `distributional` R package to implement a probabilistic grammar of graphics (Pu & Kay 2020). Development of the theoretical framework for the grammar of temporal graphics is nearing completion, and a discussion presentation about the grammar and software's design was made to the ggplot2 extension club. I have begun experimentating with nested coordinate spaces for calendar layouts and other novel computational challenges of implementing the design as an extension of the `ggplot2` R package (Wickham 2016).

All progression has been integrated into the forecasting ecosystem provided by the fabletools R package (topic 6).

A summary of thesis progress is given in the progress column of the timeline found in Table 1.

Estimated completion	Task	Progress
Topic 1: Reconciliation of structured time series forecasts with graphs		
June 2023	Theory development	100%
June 2023	ISF2023 presentation	100%
February 2024	Software development	90%*
April 2025	Paper submission	80%
Topic 2: Forecasting quality over quantity: pruning large collections of coherent time series		
May 2024	Theory development	95%
June 2024	ISF2024 presentation	100%
May 2025	Software development	70%*
July 2025	Paper submission	50%
Topic 3: Statistical computing with vectorised operations on distributions		
April 2024	Theory development	100%
July 2024	useR! presentation	100%
July 2024	Software development	100%*
September 2025	Paper submission	20%
Topic 4: Reconciling mixed temporal granularities		
May 2025	Theory development	60%
July 2025	ISF2025 presentation	10%
June 2025	Software development	20%*
January 2026	Paper submission	0%
Topic 5: Grammar of temporal graphics		
May 2025	Theory development	75%
February 2025	ggplot2-extenders presentation	100%
June 2025	ISF2025 workshop	30%
August 2025	JSM 2025 presentation	30%
August 2025	UseR! 2025 presentation	30%
June 2025	Software development	20%*
September 2025	Paper submission	10%
Topic 6: Probabilistic forecasting at scale using tidy data structures		
December 2025	Theory development	80%
March 2026	Software development	75%*

* Software is never really finished, 100% indicates that the work is ready for publication.

Table 1: *Planned timeline for completing tasks associated with each topic to form the PhD thesis.*

References

Andreou, E, E Ghysels & A Kourtellos (2011). "Forecasting with Mixed-Frequency Data". In: *The Oxford Handbook of Economic Forecasting*. Oxford University Press.

Athanasopoulos, G, RJ Hyndman, N Kourentzes & F Petropoulos (2017a). Forecasting with temporal hierarchies. *European Journal of Operational Research* **262**(1), 60–74.

Athanasopoulos, G, RJ Hyndman, N Kourentzes & F Petropoulos (2017b). Forecasting with temporal hierarchies. *European Journal of Operational Research* **262**(1), 60–74.

Chambers, JM & TJ Hastie (1993). *Statistical models in S*. Philadelphia, PA: Chapman & Hall/CRC.

Di Fonzo, T & D Girolimetto (2021). Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives. *International Journal of Forecasting*. in press.

Girolimetto, D & T Di Fonzo (2023). *Point and probabilistic forecast reconciliation for general linearly constrained multiple time series*. arXiv: 2305.05330 [stat.ME].

Grolemund, G & H Wickham (2011). Dates and Times Made Easy with lubridate. *Journal of Statistical Software* **40**(3), 1–25.

Hayes, A, R Moller-Trane, D Jordan, P Northrop, MN Lang & A Zeileis (2022). *distributions3: Probability Distributions as S3 Objects*. R package version 0.2.1. https://CRAN.R-project.org/package=distributions3.

Hyndman, RJ (1996). Computing and Graphing Highest Density Regions. *The American Statistician* **50**(2), 120–126.

Hyndman, RJ & G Athanasopoulos (2021). *Forecasting: principles and practice, 3rd edition*. OTexts. https://otexts.com/fpp3/.

Hyndman, RJ, RA Ahmed, G Athanasopoulos & HL Shang (2011). Optimal Combination Forecasts for Hierarchical Time Series. *Comput. Stat. Data Anal.* **55**(9), 2579–2589.

Hyndman, RJ & Y Khandakar (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software* **27**(3), 1–22.

Hyndman, R & A Koehler (2006). Another look at measures of forecast accuracy. English. *International Journal of Forecasting* **22**(4), 679–688.

Kang, Y, RJ Hyndman & K Smith-Miles (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting* **33**(2), 345–358.

Kay, M (2022). *ggdist: Visualizations of Distributions and Uncertainty*. R package version 3.2.0. https://mjskay.github.io/ggdist/.

Kay, M (2023). ggdist: Visualizations of Distributions and Uncertainty in the Grammar of Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 1–11. (Visited on 02/24/2025).

Kourentzes, N & G Athanasopoulos (2019). Cross-temporal coherent forecasts for Australian tourism. *Annals of Tourism Research* **75**, 393–409.

Matheson, JE & RL Winkler (1976). Scoring Rules for Continuous Probability Distributions. *Management Science* **22**(10), 1087–1096.

O'Hara-Wild, M (2024). *graphvec: Vectorised graph data structures*. R package version 0.0.0.9000. http://pkg.mitchelloharawild.com/graphvec.

O'Hara-Wild, M, RJ Hyndman & E Wang (2024). *fabletools: Core Tools for Packages in the 'fable' Framework*. https://fabletools.tidyverts.org/.

O'Hara-Wild, M, M Kay, A Hayes & R Hyndman (2020). *distributional: Vectorised Probability Distributions*. en. Institution: Comprehensive R Archive Network Pages: 0.5.0. https://CRAN.R-project.org/package=distributional (visited on 02/24/2025).

Panagiotelis, A, P Gamakumara, G Athanasopoulos & RJ Hyndman (2022). Probabilistic forecast reconciliation: properties, evaluation and score optimisation. *European J Operational Research*. in press.

Pu, X & M Kay (2020). A Probabilistic Grammar of Graphics. en. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, pp.1–13. https://dl.acm.org/doi/10.1145/3313831.3376466 (visited on 02/24/2025).

Ruckdeschel, P & M Kohl (2014). General Purpose Convolution Algorithm in S4 Classes by Means of FFT. *Journal of Statistical Software* **59**(4), 1–25.

Sonabend, R & F Kiraly (2022). *distr6: The Complete R6 Probability Distributions Interface*. https://alan-turing-institute.github.io/distr6/.

Wang, E, D Cook & RJ Hyndman (2020). A new tidy data structure to support exploration and modeling of temporal data. *Journal of Computational and Graphical Statistics* **29**(3), 466–478.

Wang, X, RJ Hyndman & SL Wickramasuriya (2024). *Optimal forecast reconciliation with time series selection*. https://github.com/xqnwang/hfs.

Wickham, H (2016). *ggplot2*. en. Use R! Cham: Springer International Publishing. http://link.springer.com/10.1007/978-3-319-24277-4 (visited on 02/24/2025).

Wickham, H, M Averick, J Bryan, W Chang, LD McGowan, R François, G Grolemund, A Hayes, L Henry, J Hester, M Kuhn, TL Pedersen, E Miller, SM Bache, K Müller, J Ooms, D Robinson, DP Seidel, V Spinu, K Takahashi, D Vaughan, C Wilke, K Woo & H Yutani (2019). Welcome to the tidyverse. *Journal of Open Source Software* **4**(43), 1686.

Wickham, H, L Henry & D Vaughan (2022). *vctrs: Vector Helpers*. R package version 0.5.0. https://vctrs.r-lib.org/.

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* **114**(526), 804–819.

Wilkinson, GN & CE Rogers (1973). Symbolic Description of Factorial Models for Analysis of Variance. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **22**(3), 392–399.

Wilkinson, L (2011). "The Grammar of Graphics". In: *Handbook of computational statistics: Concepts and methods*. Statistics and Computing. Springer, pp.375–414. http://link.springer.com/10.1007/0-387-28695-0.

Zeileis, A & G Grothendieck (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software* **14**(6), 1–27.