MONASH University

MONASH BUSINESS SCHOOL

High-dimensional time series analysis

Rob J Hyndman

9 April 2018

Outline

1. **Visualizing many time series**
2. Finding weird time series
3. Reconciling many forecasts
4. Forecasting temporal hierarchies

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon*

INSEAD, Boulevard de Constance, 77305 Fontainebleau, France

Abstract

This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

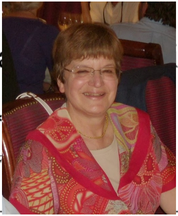
international journal of forecasting

www.elsevier.com/locate/ijforecast

etition: results, conclusions ons

Spyros Makridakis, Michèle Hibon*

EAD, Boulevard de Constance, 77305 Fontainebleau, F

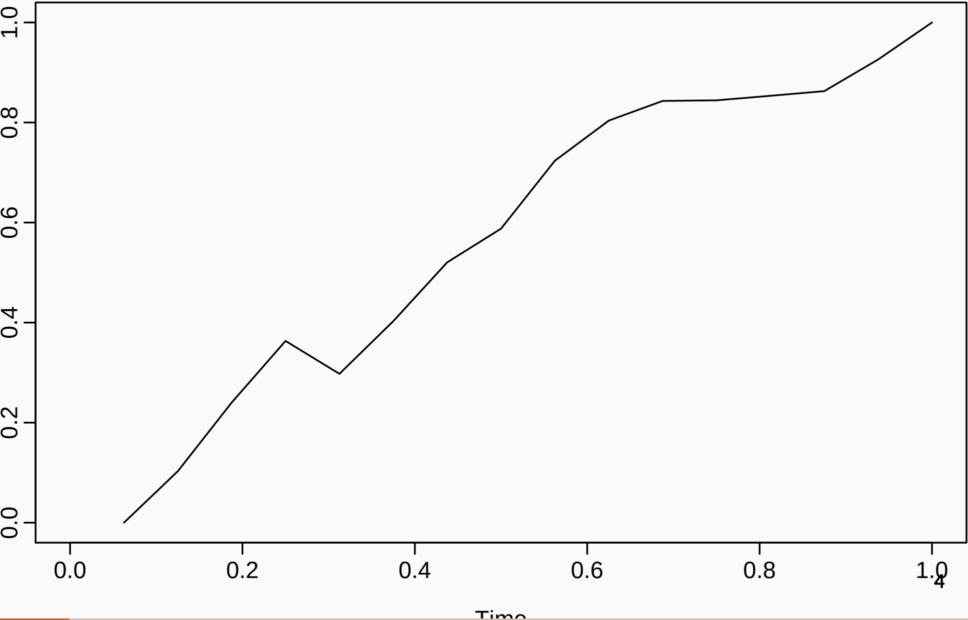


Abstract

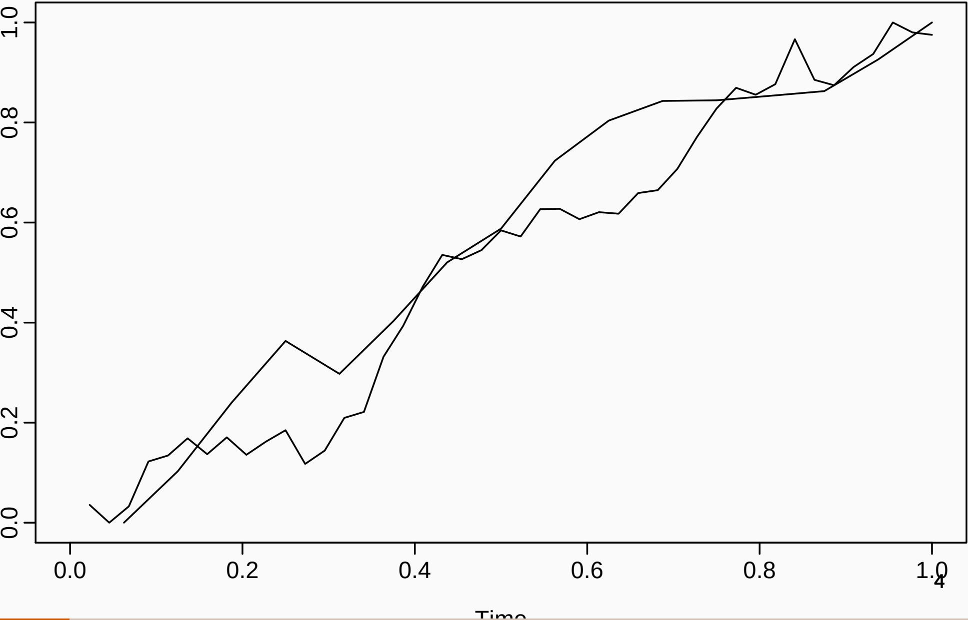
This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting

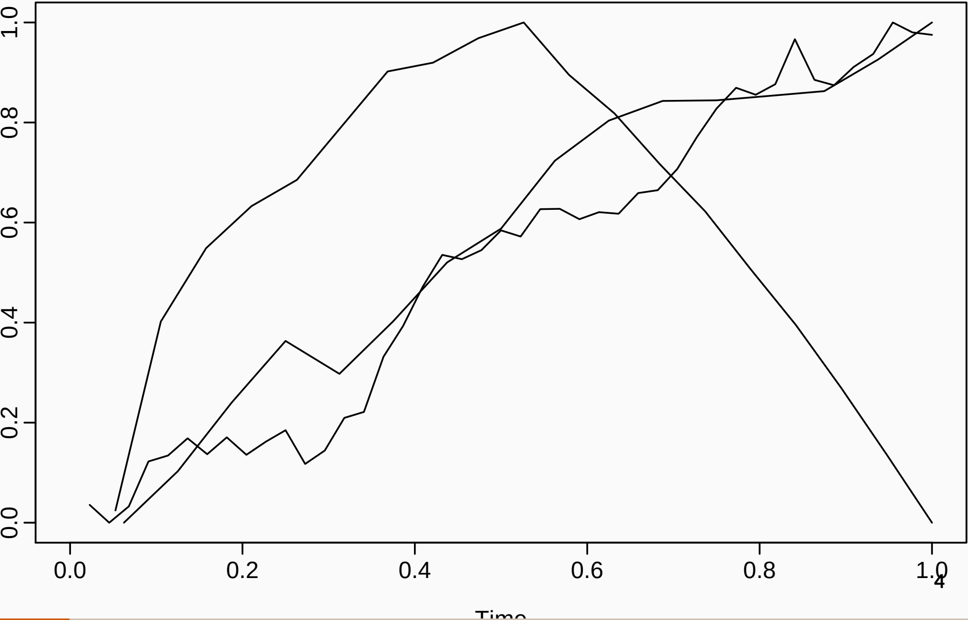
How to plot lots of time series?



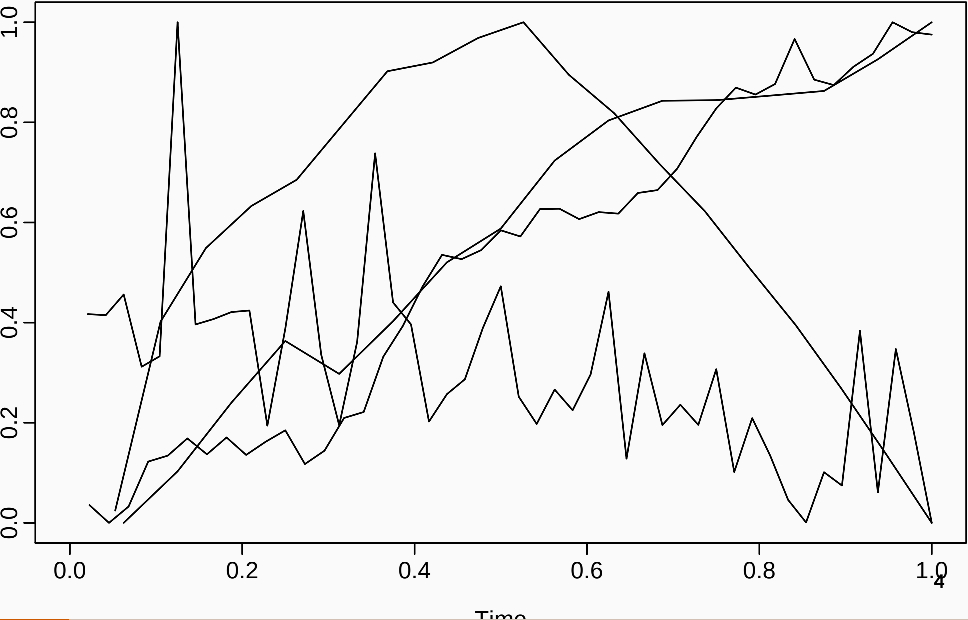
How to plot lots of time series?



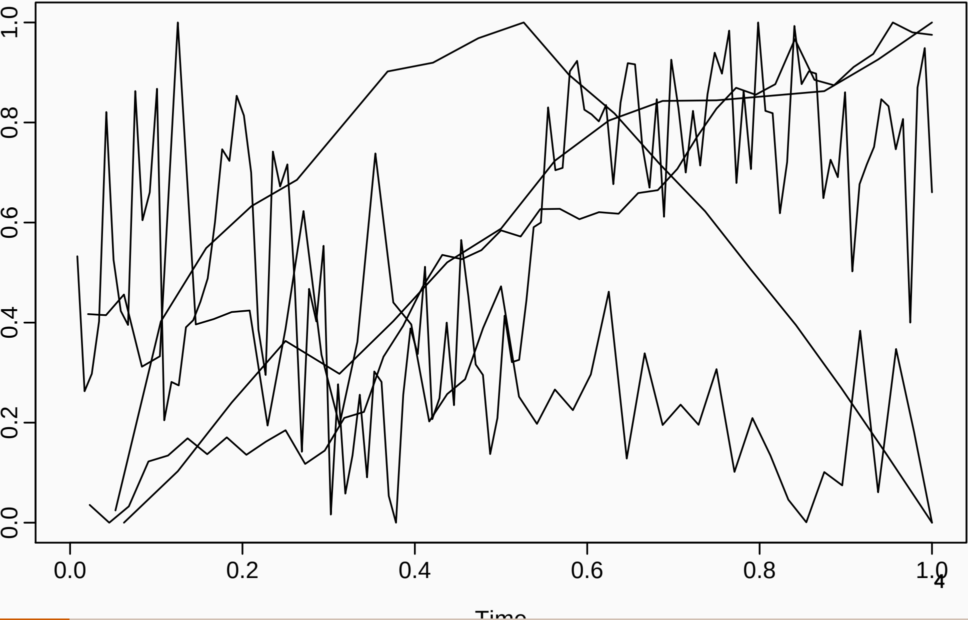
How to plot lots of time series?



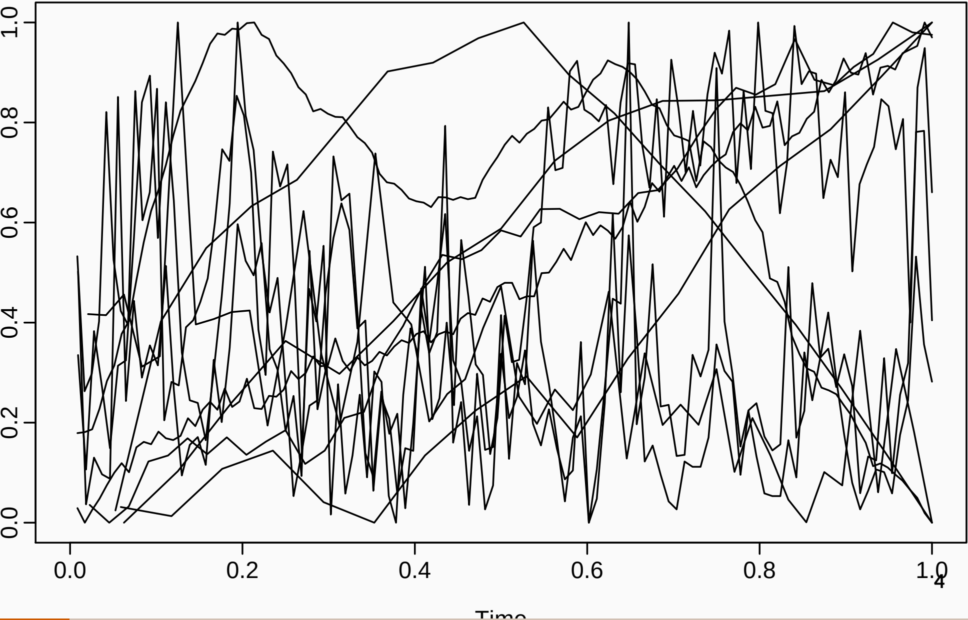
How to plot lots of time series?



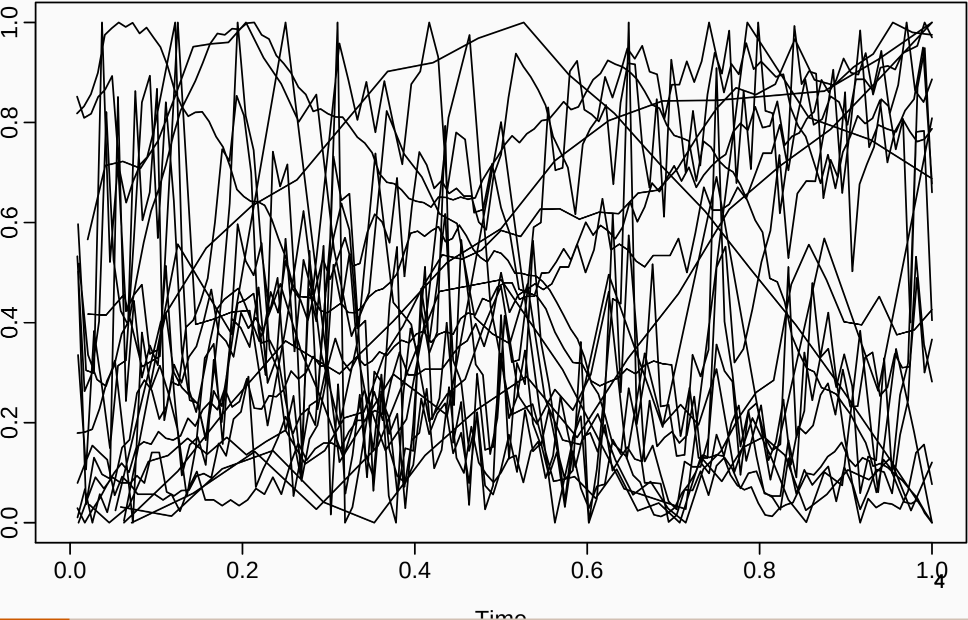
How to plot lots of time series?



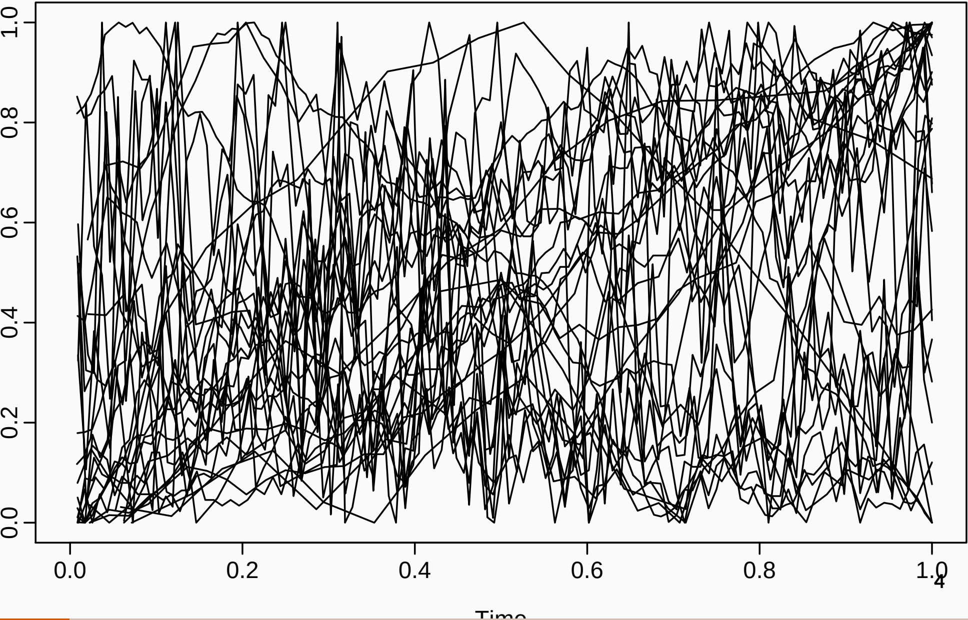
How to plot lots of time series?



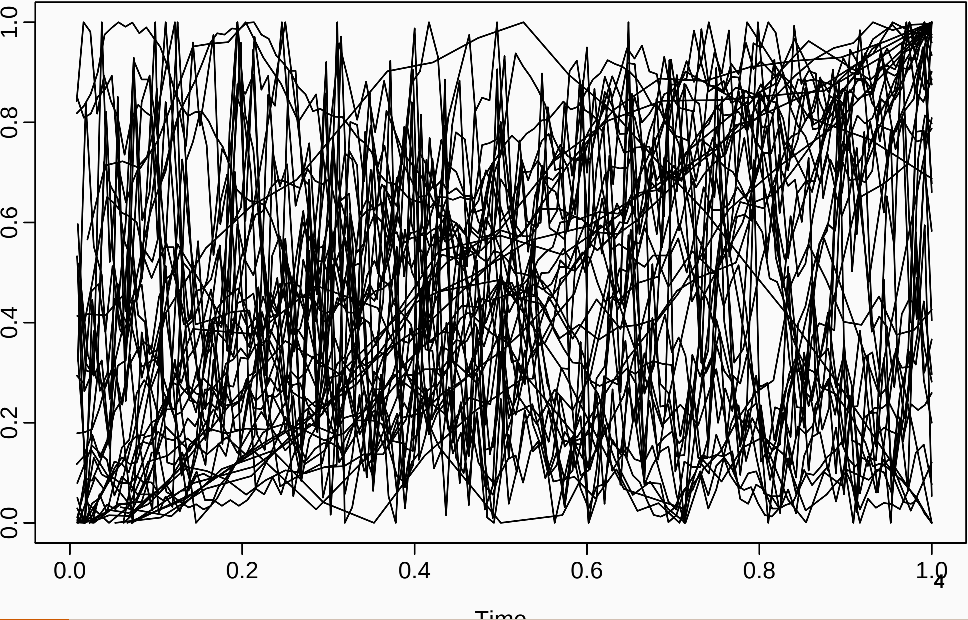
How to plot lots of time series?



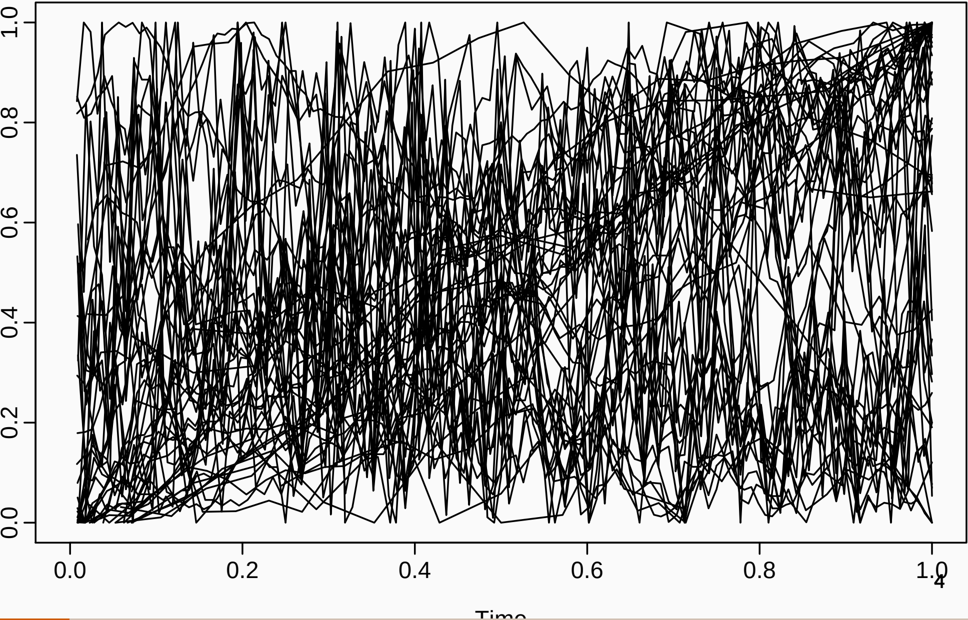
How to plot lots of time series?



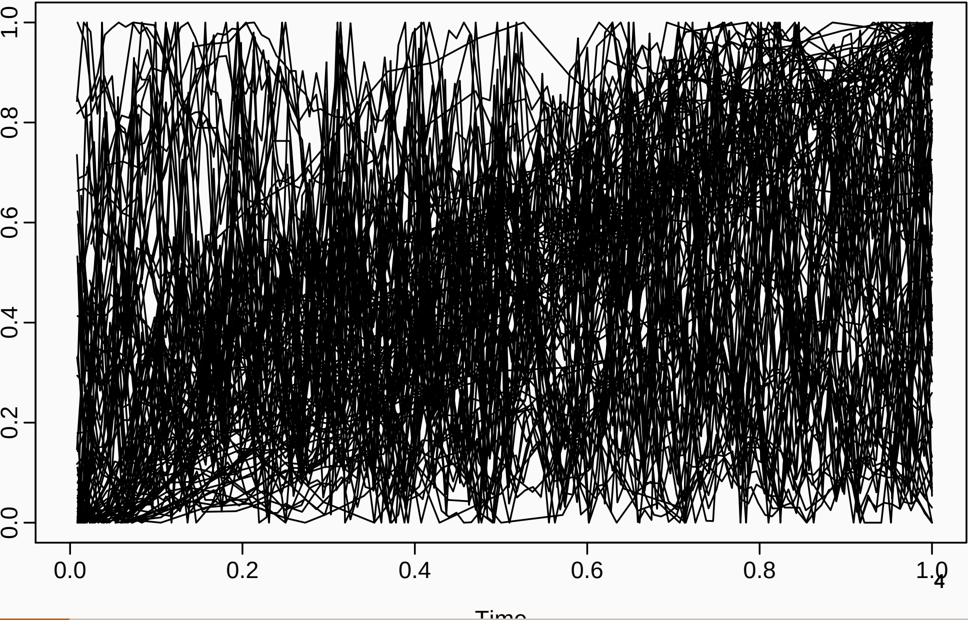
How to plot lots of time series?



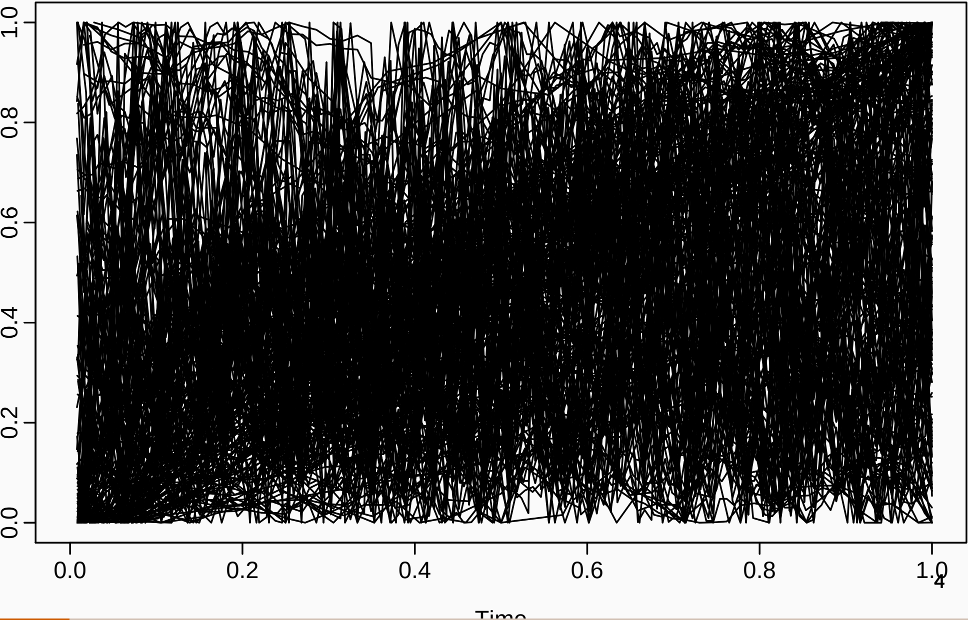
How to plot lots of time series?



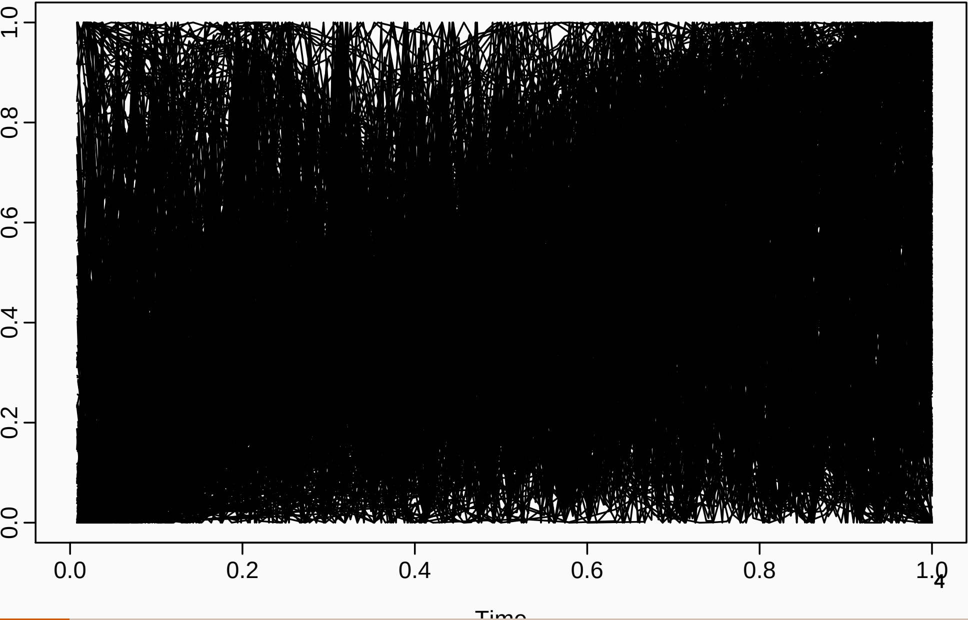
How to plot lots of time series?



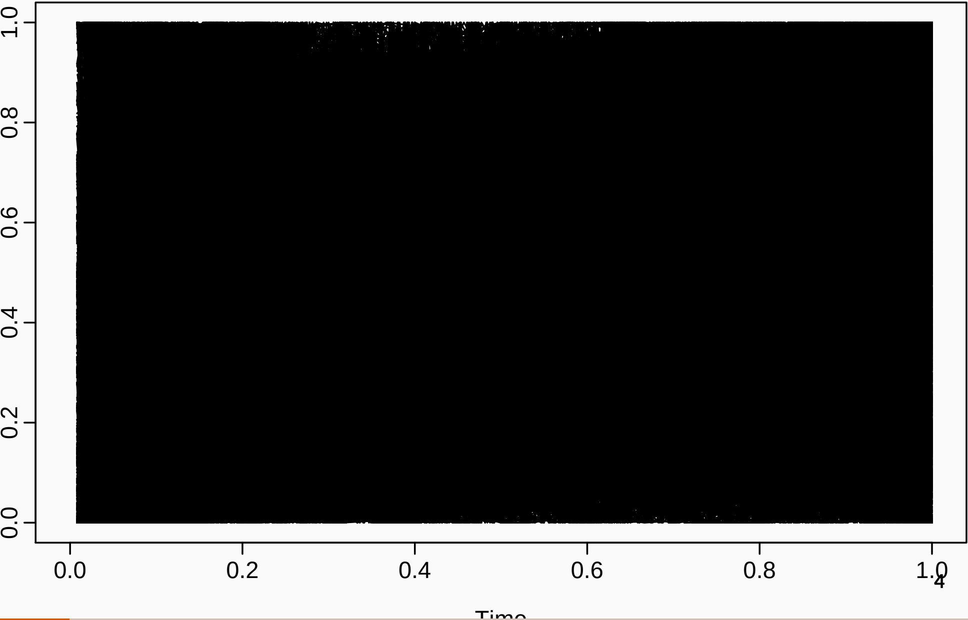
How to plot lots of time series?



How to plot lots of time series?



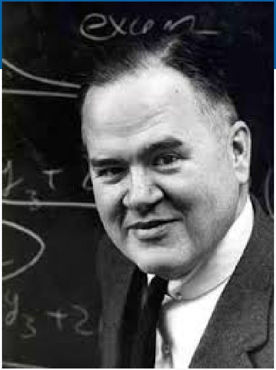
How to plot lots of time series?



Key idea

Cognostics

Computer-produced diagnostics (Tukey and Tukey, 1985).



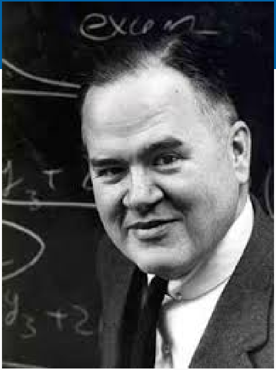
John W Tukey

Key idea

Cognostics

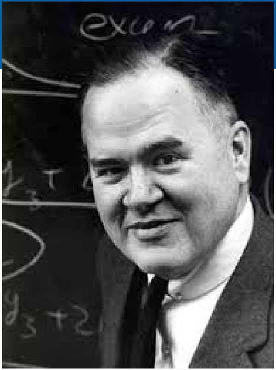
Computer-produced diagnostics (Tukey and Tukey, 1985).

- lag correlation
- size and direction of trend
- strength of seasonality
- timing of peak seasonality
- spectral entropy



John W Tukey

Key idea



John W Tukey

Cognostics

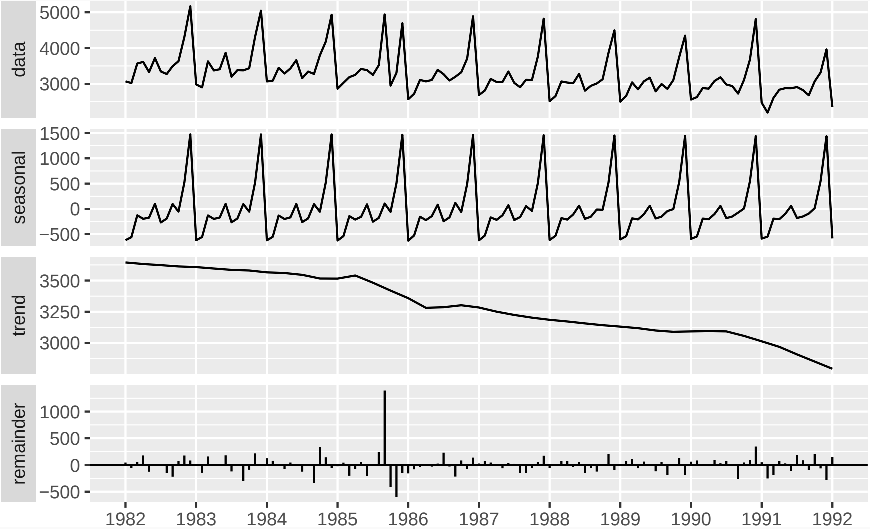
Computer-produced diagnostics (Tukey and Tukey, 1985).

- lag correlation
- size and direction of trend
- strength of seasonality
- timing of peak seasonality
- spectral entropy

Called “features” in the machine learning literature.

An STL decomposition: N2096

$Y_t = S_t + T_t + R_t$ $\qquad$ S_t is periodic with mean 0



Candidate features

STL decomposition

$$Y_t = S_t + T_t + R_t$$

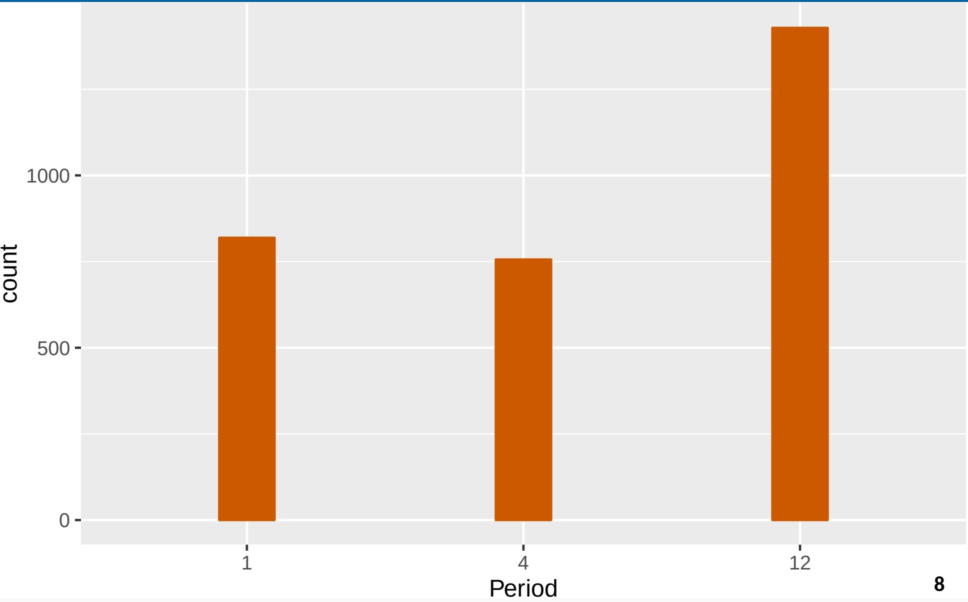
Candidate features

STL decomposition

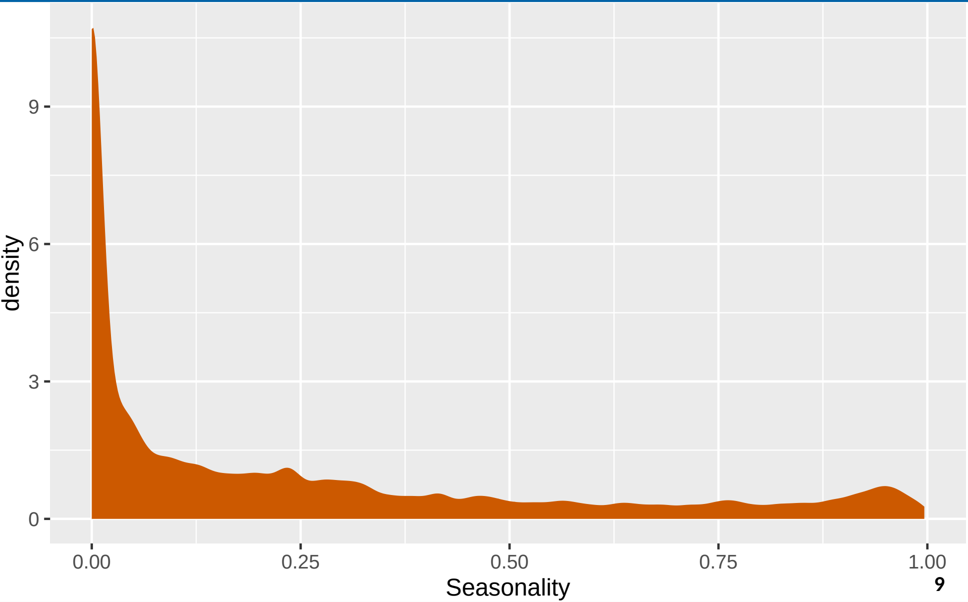
$$Y_t = S_t + T_t + R_t$$

- Seasonal period
- Autocorrelations of data $(Y_1, \dots, Y_T)$
- Autocorrelations of residuals $(R_1, \dots, R_T)$
- Strength of seasonality: $\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t+R_t)}\right)$
- Strength of trend: $\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(T_t+R_t)}\right)$
- Spectral entropy: $H = -\int_{-\pi}^{\pi} f_y(\lambda) \log f_y(\lambda) d\lambda$, where $f_y(\lambda)$ is spectral density of Y_t.
  Low values of H suggest a time series that is easier to forecast (more signal).
- Optimal Box-Cox transformation of data

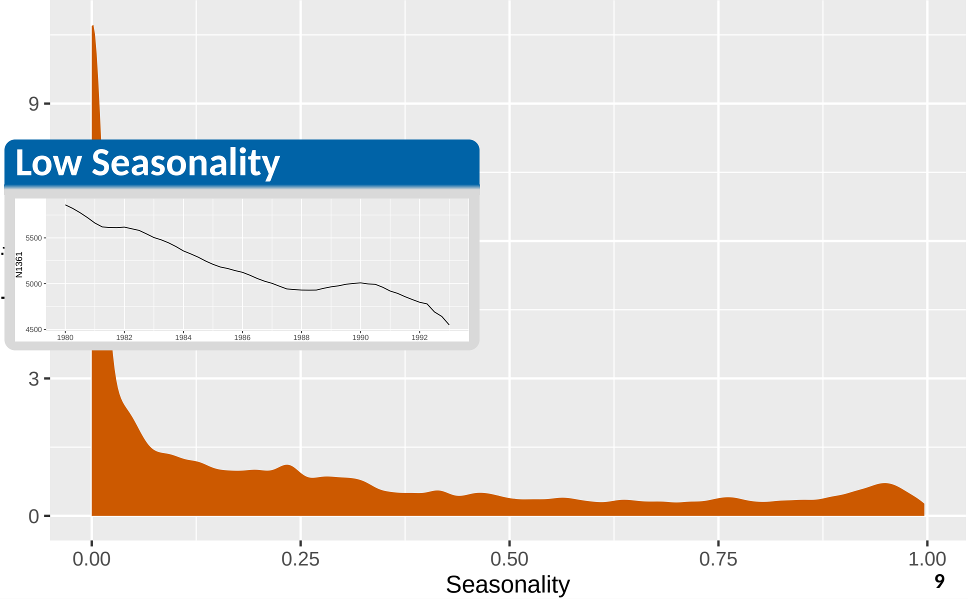
Distribution of Period for M3



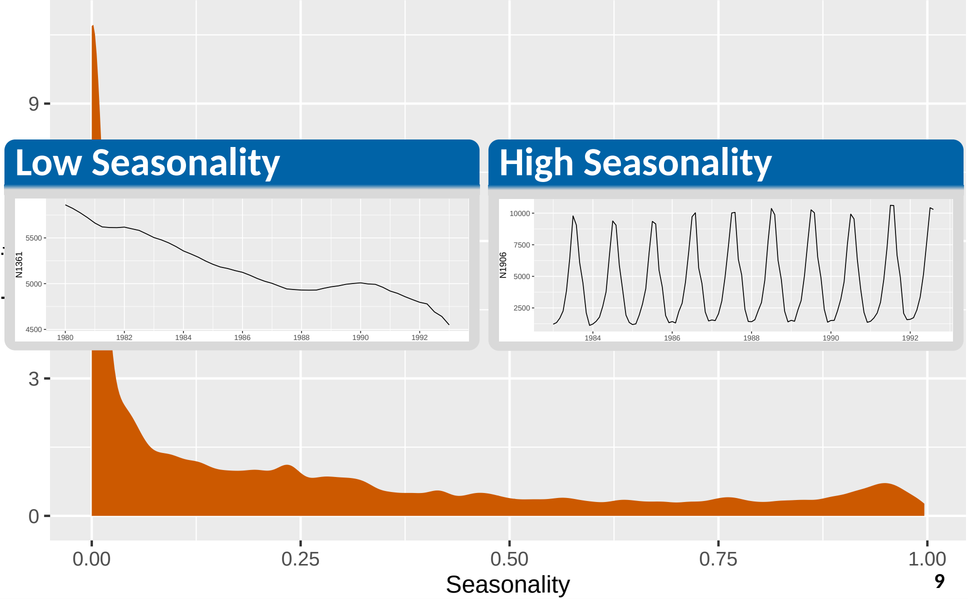
Distribution of Seasonality for M3



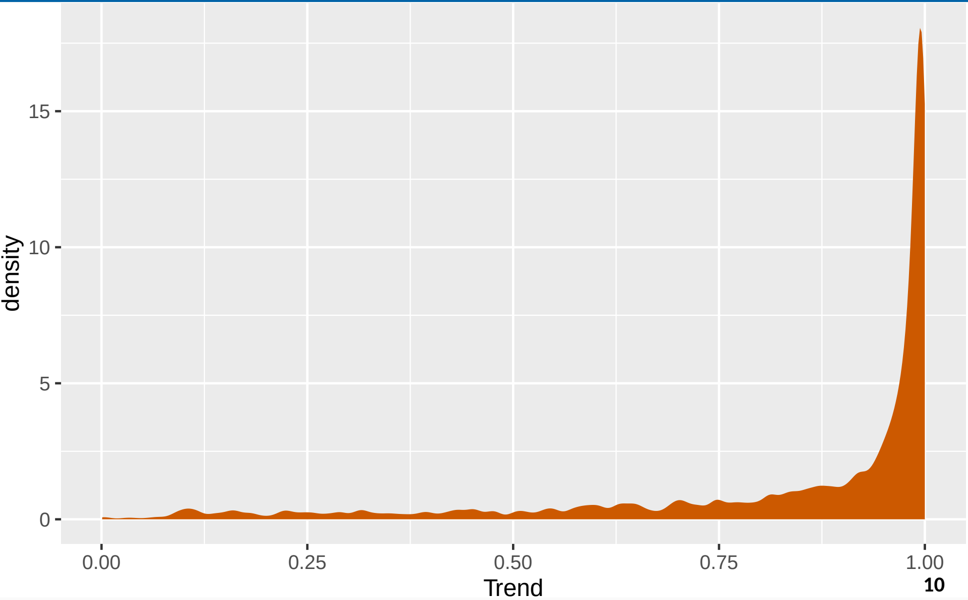
Distribution of Seasonality for M3



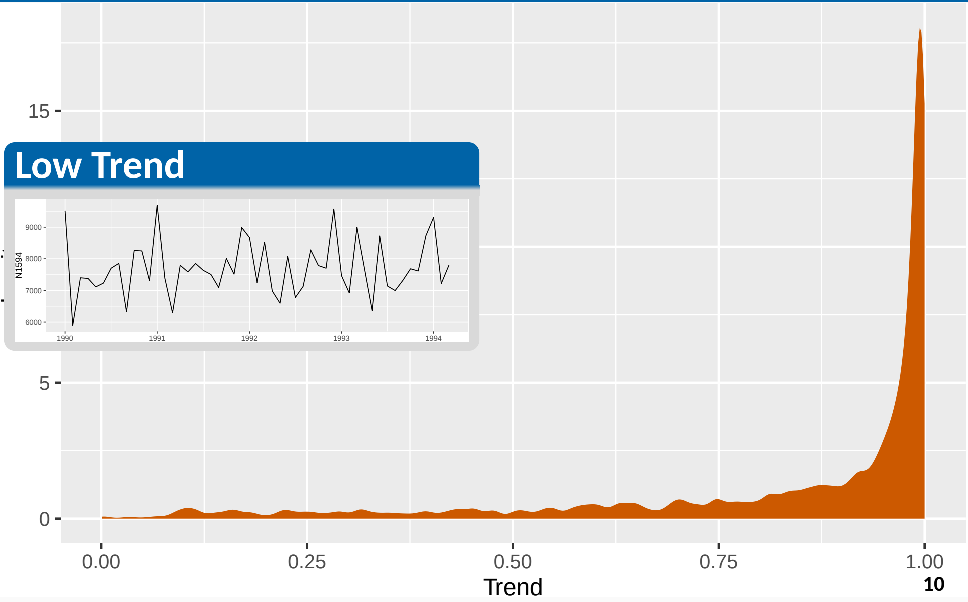
Distribution of Seasonality for M3



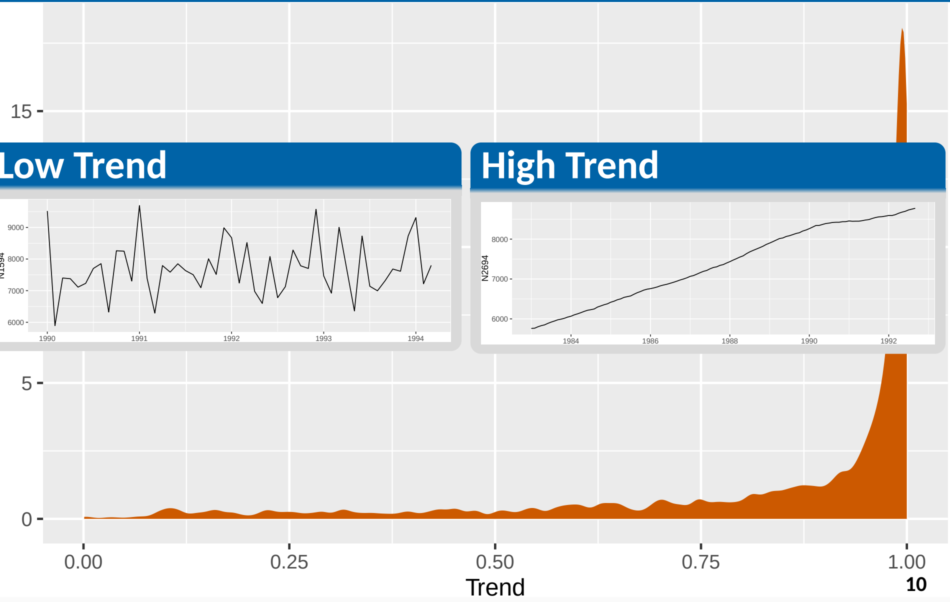
Distribution of Trend for M3



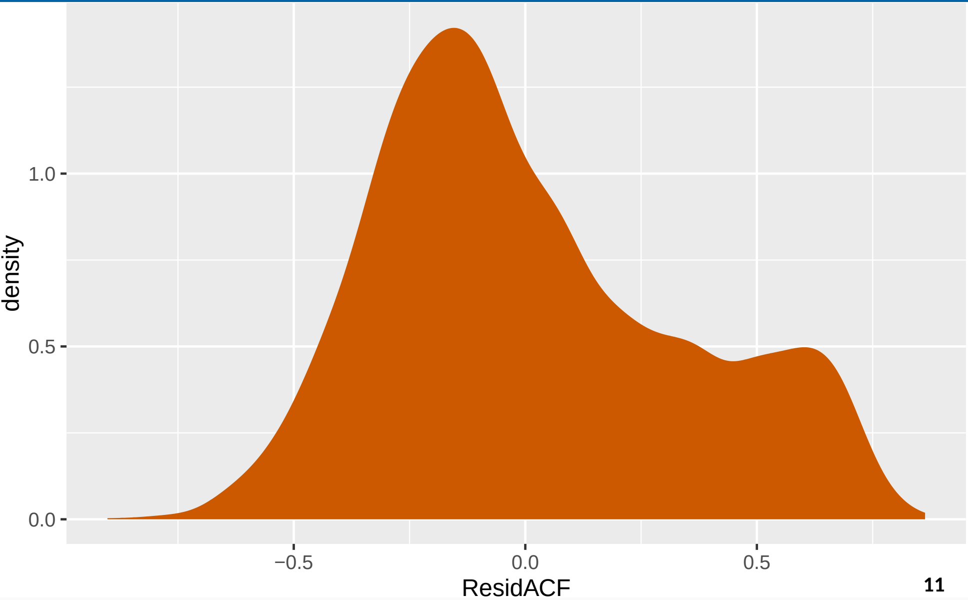
Distribution of Trend for M3



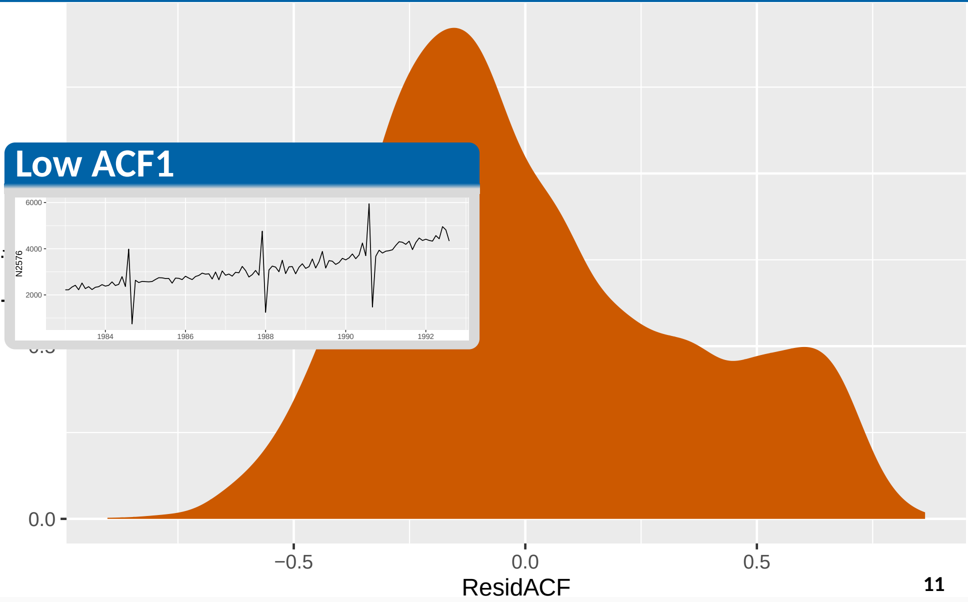
Distribution of Trend for M3



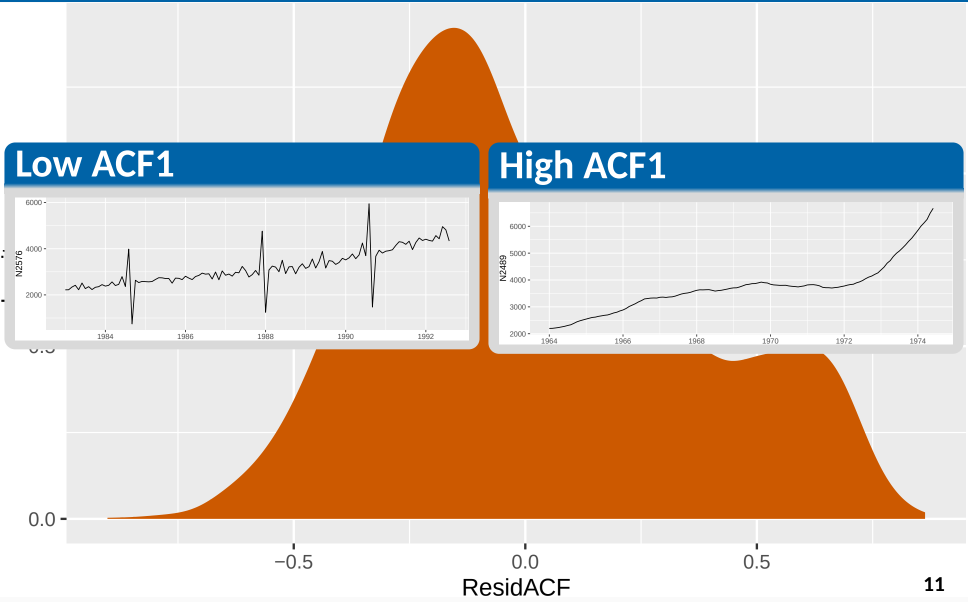
Distribution of residual ACF1 for M3



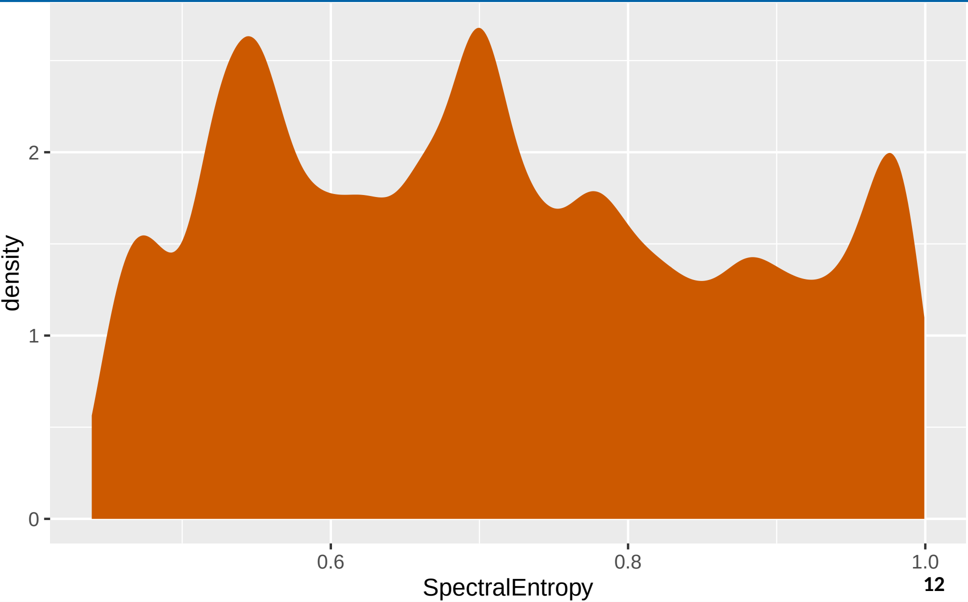
Distribution of residual ACF1 for M3



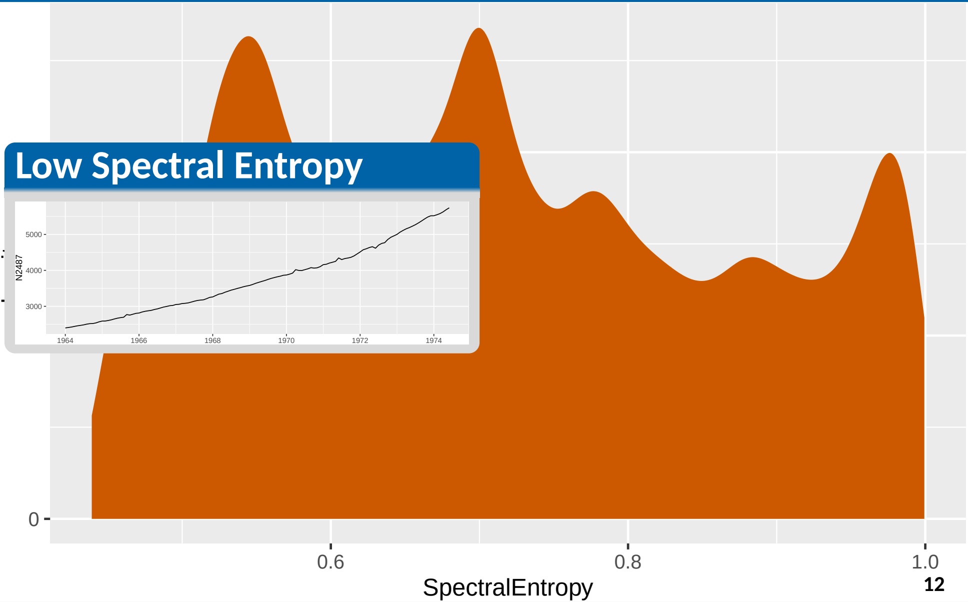
Distribution of residual ACF1 for M3



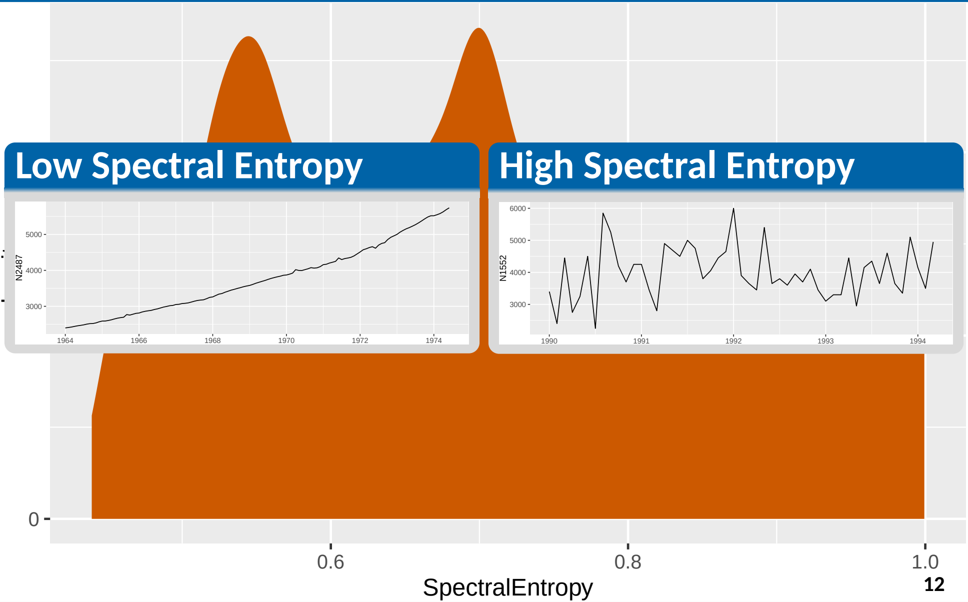
Distribution of Spectral Entropy for M3



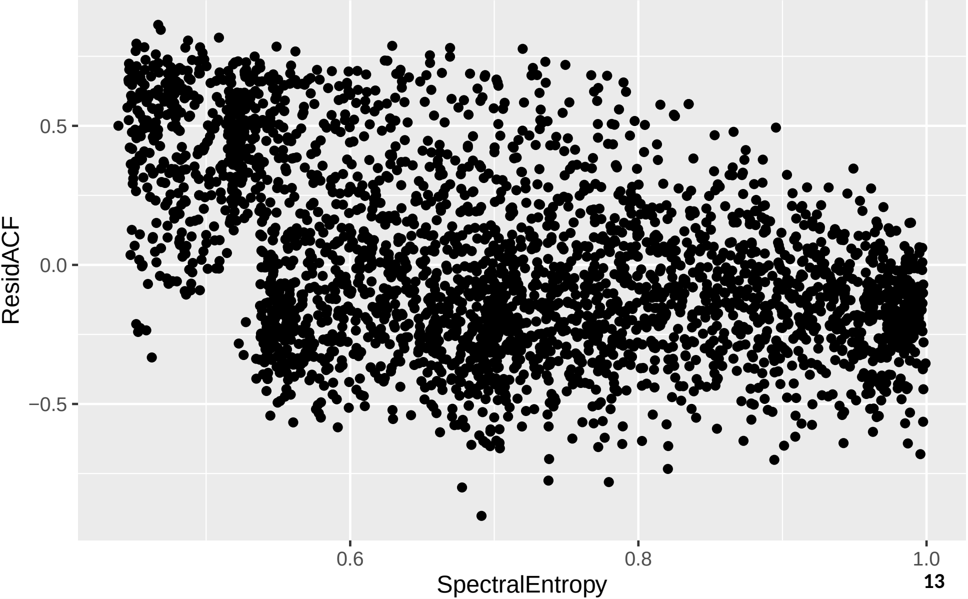
Distribution of Spectral Entropy for M3



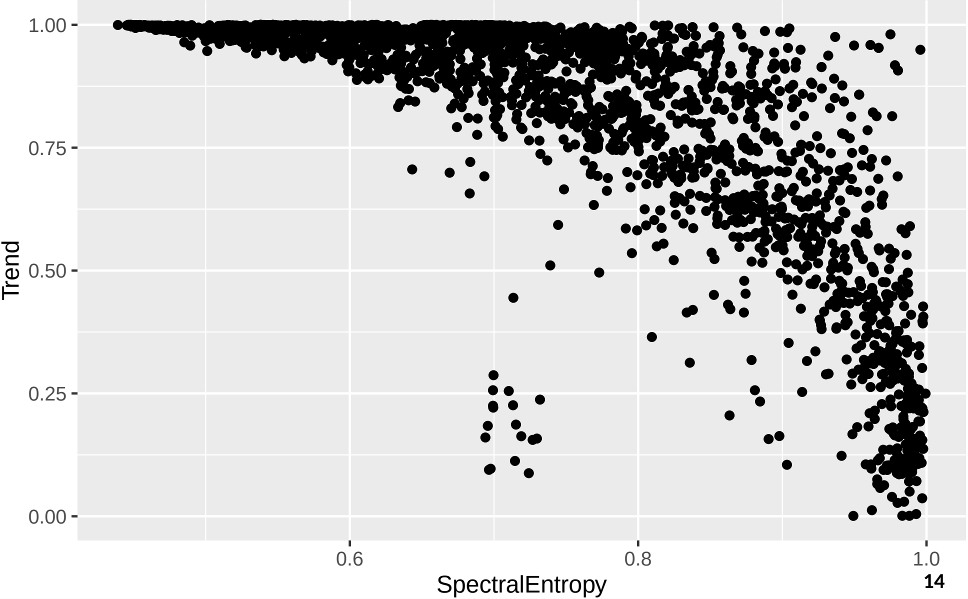
Distribution of Spectral Entropy for M3



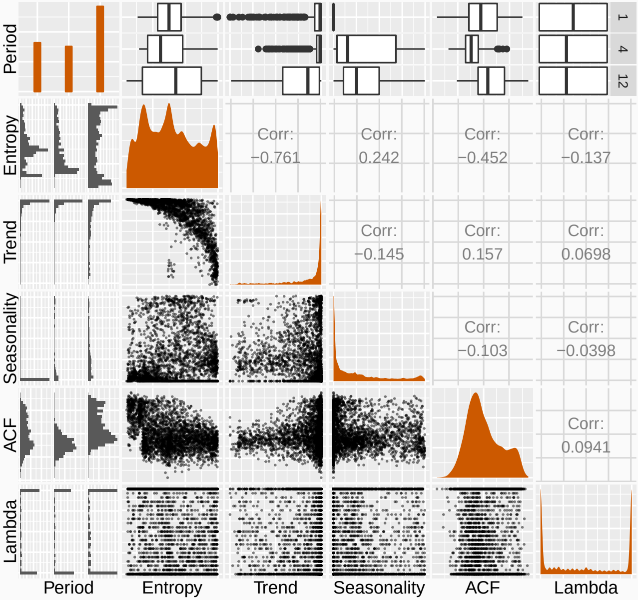
Feature distributions



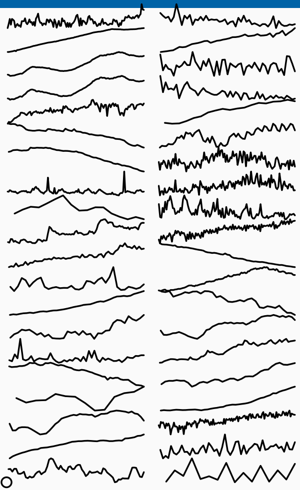
Feature distributions



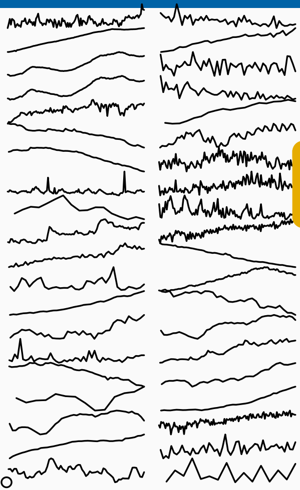
Feature distributions

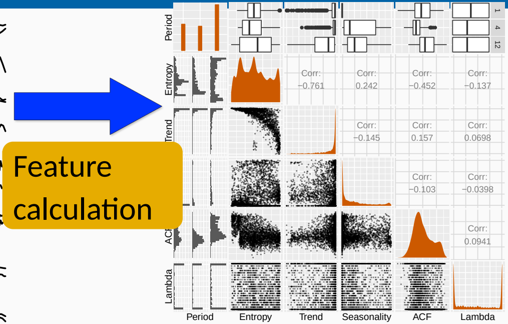


Dimension reduction for time series

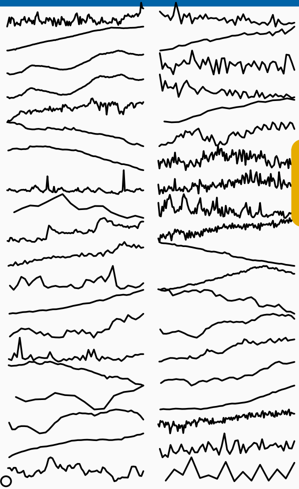


Dimension reduction for time series

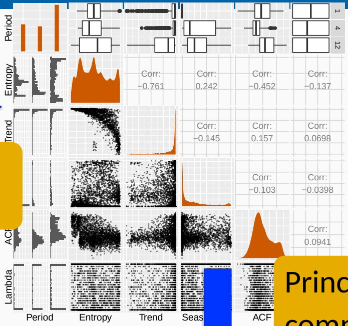




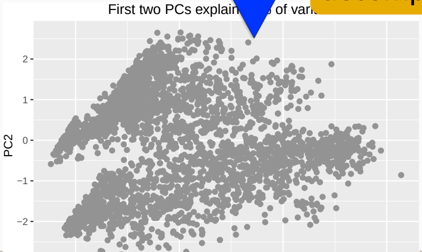
Dimension reduction for time series



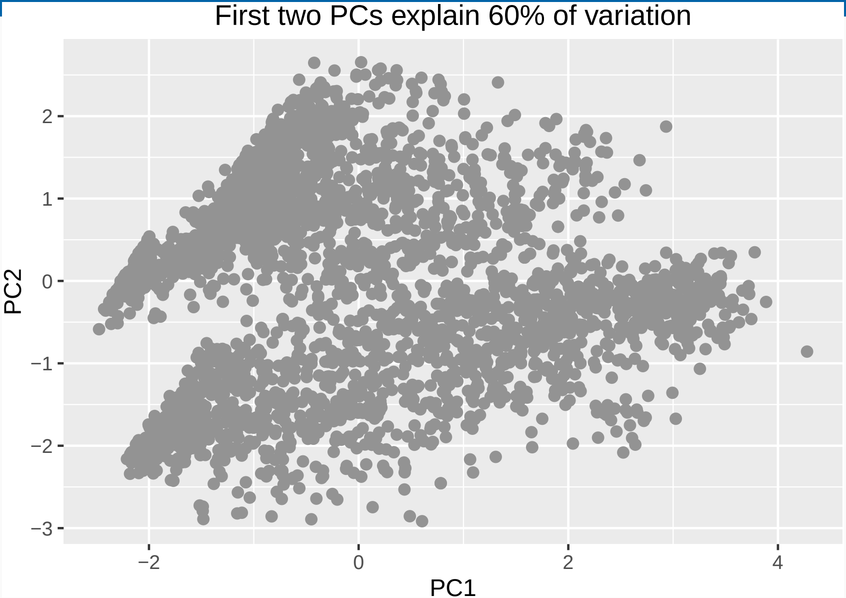
Feature calculation



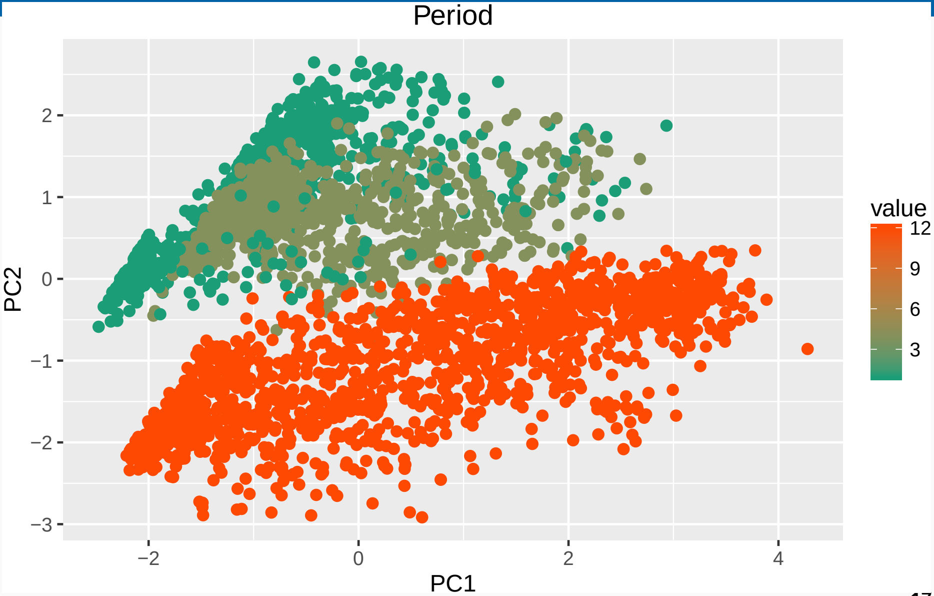
Principal component decomposition



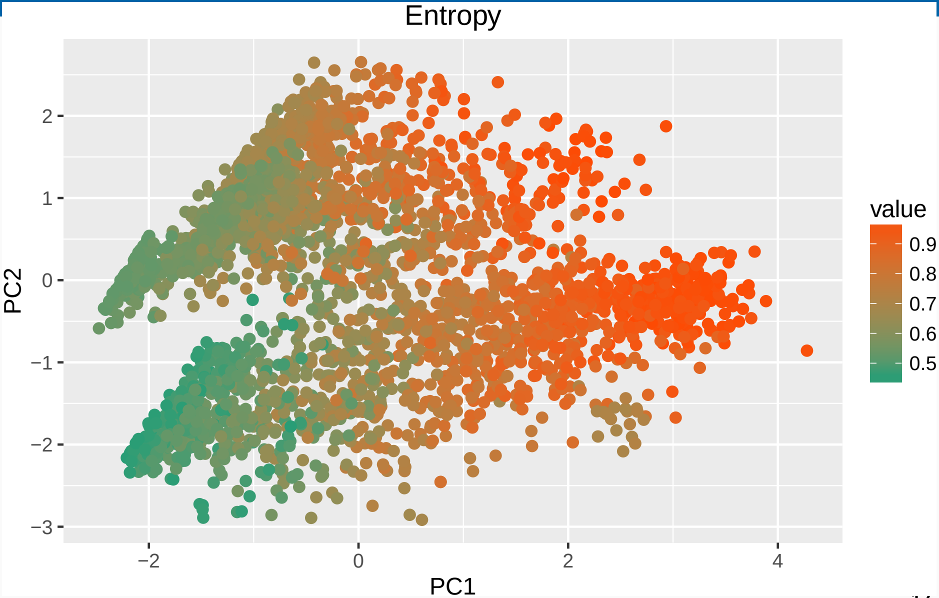
Feature space of M3 data



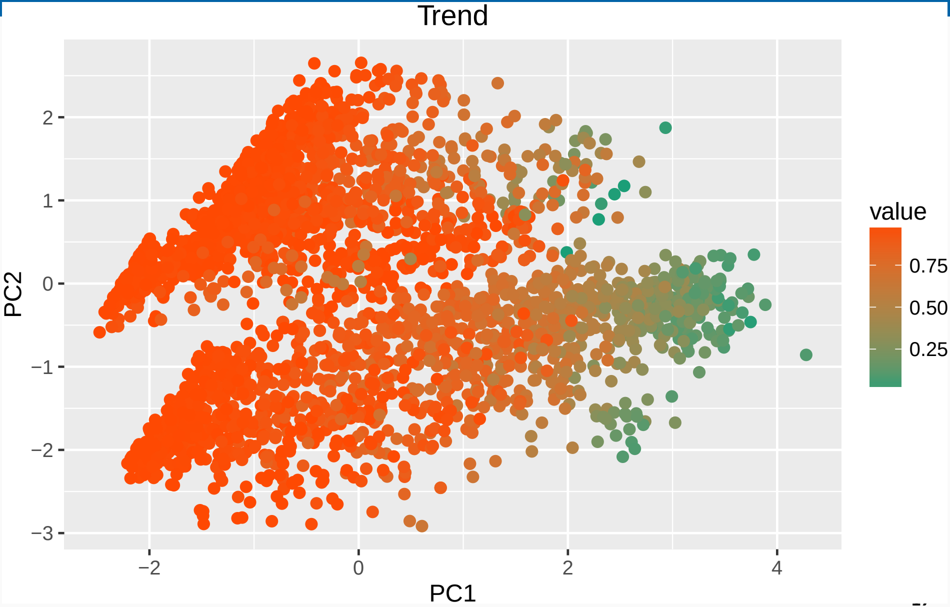
Feature space of M3 data



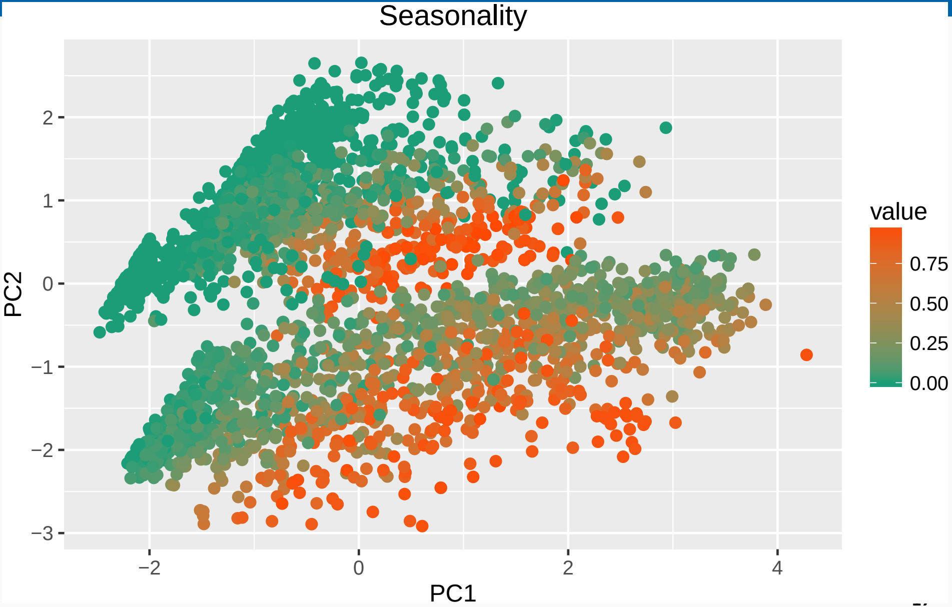
Feature space of M3 data



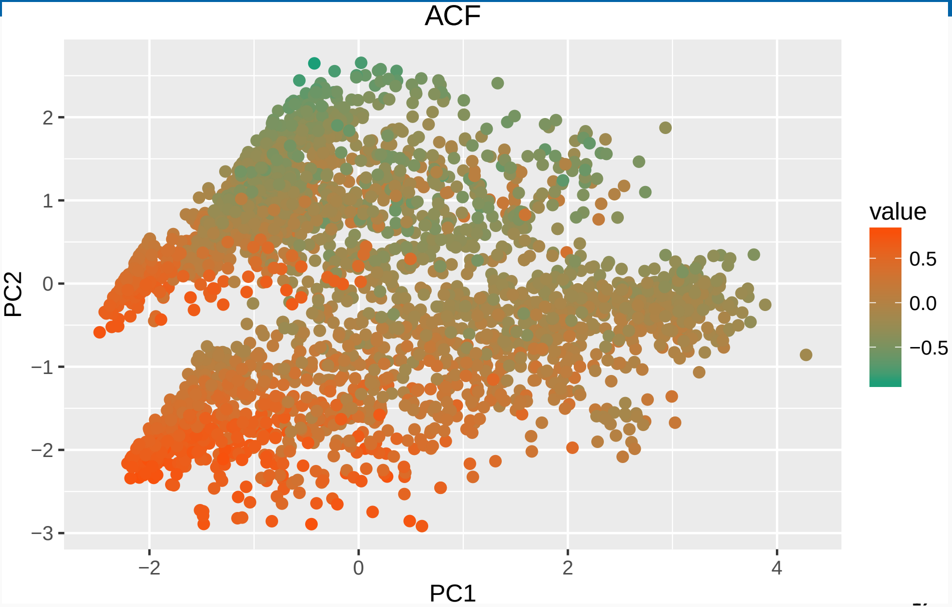
Feature space of M3 data



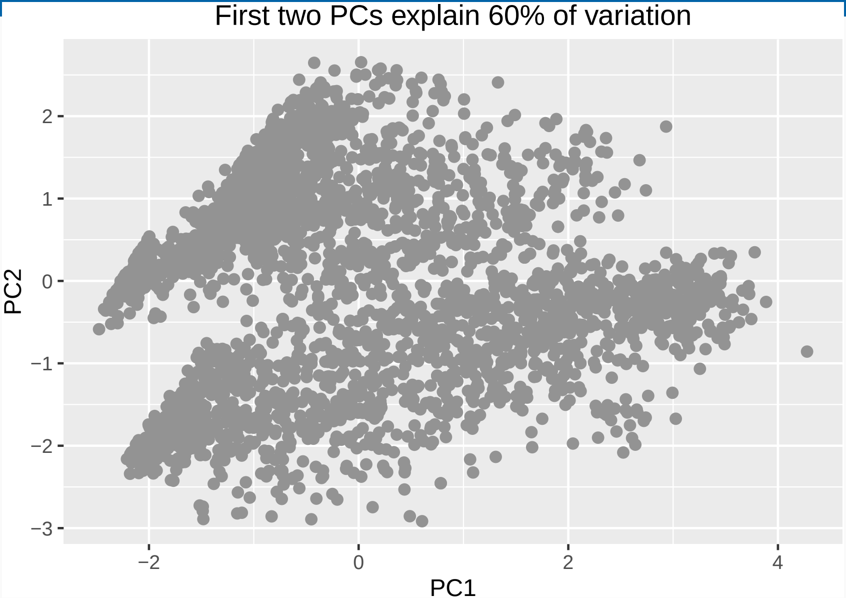
Feature space of M3 data



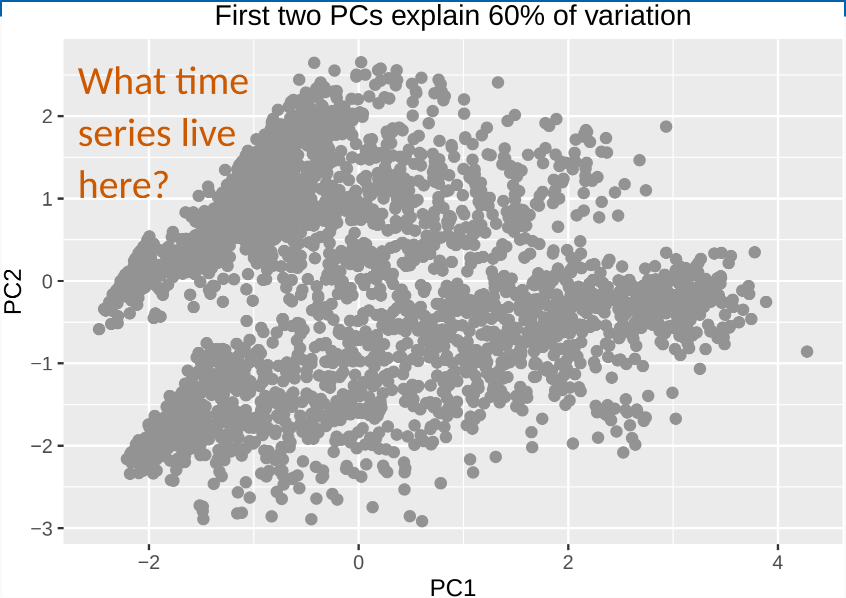
Feature space of M3 data



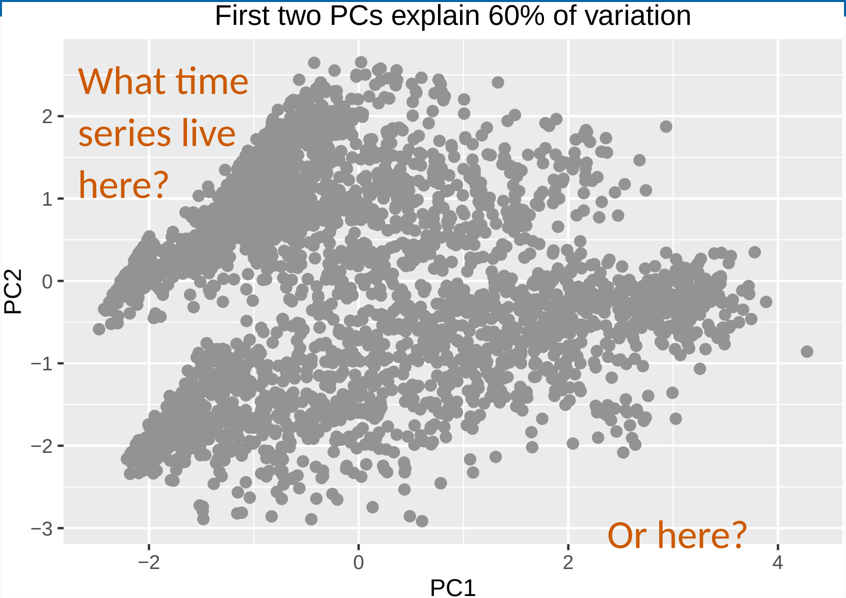
What about the holes?



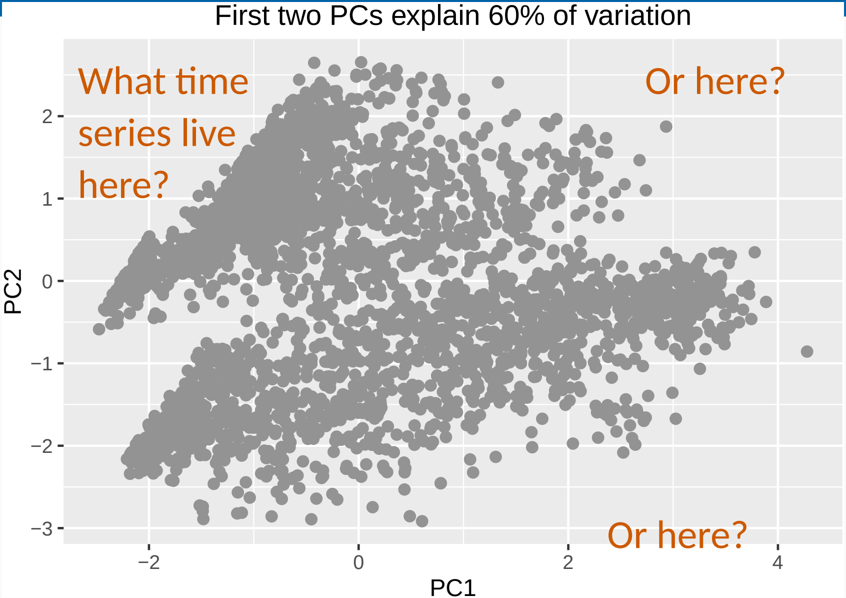
What about the holes?



What about the holes?



What about the holes?



Generating new time series

We can use the feature space to:

- Generate new time series with similar features to existing series
- Generate new time series where there are “holes” in the feature space.

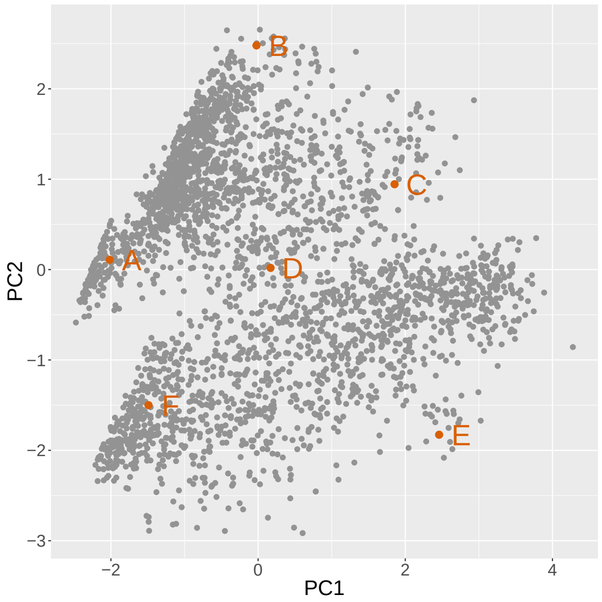
Generating new time series

We can use the feature space to:

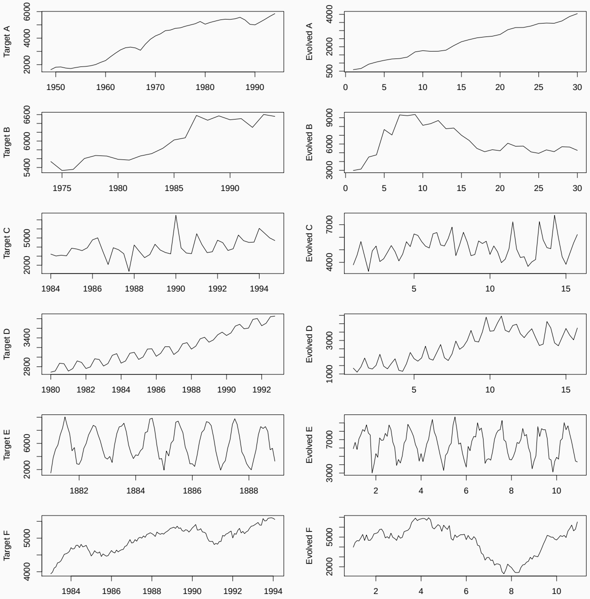
- Generate new time series with similar features to existing series
- Generate new time series where there are "holes" in the feature space.

- Let $\{\mathrm{PC}_1, \mathrm{PC}_2, \dots, \mathrm{PC}_n\}$ be a "population" of time series of specified length and period.
- Genetic algorithm uses a process of selection, crossover and mutation to evolve the population towards a target point T_i.
- Optimize: Fitness $(\mathrm{PC}_j) = -\sqrt{(|\mathrm{PC}_j - T_i|^2)}$.
- Initial population random with some series in neighbourhood of T_i.

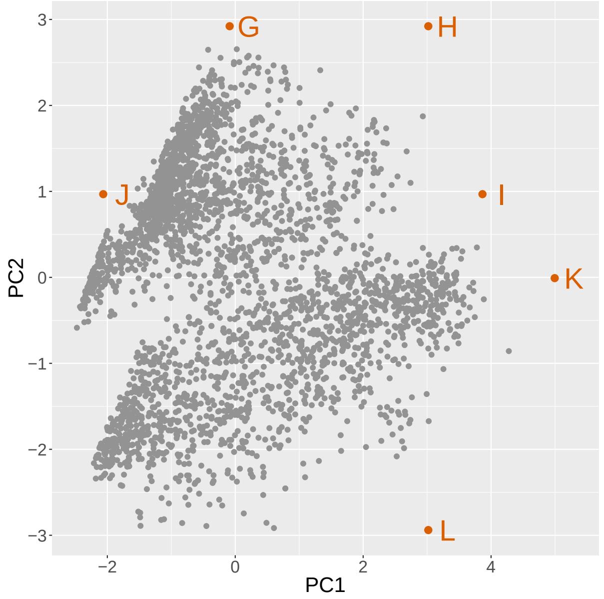
Evolving new time series



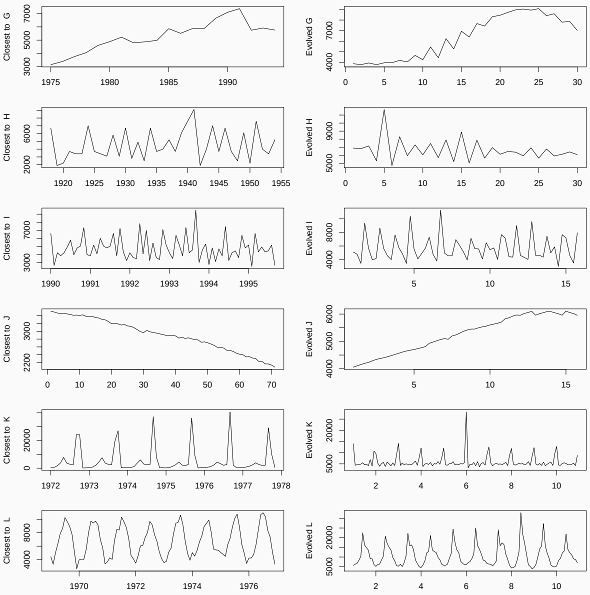
Evolving new time series



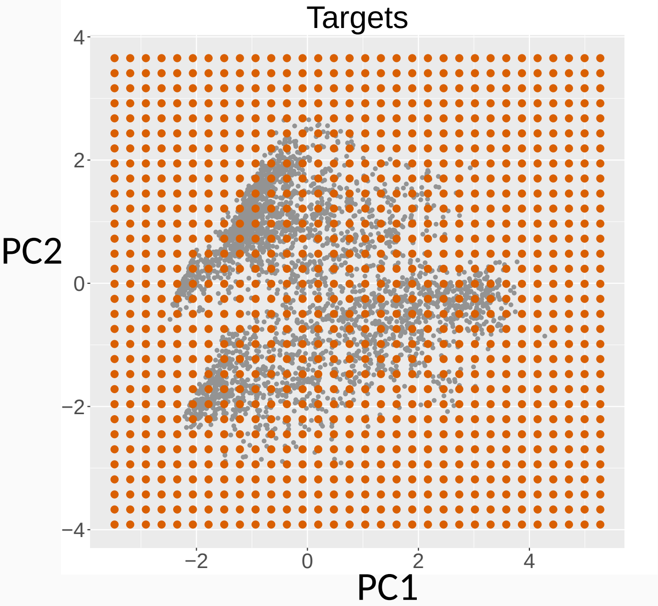
Evolving new time series



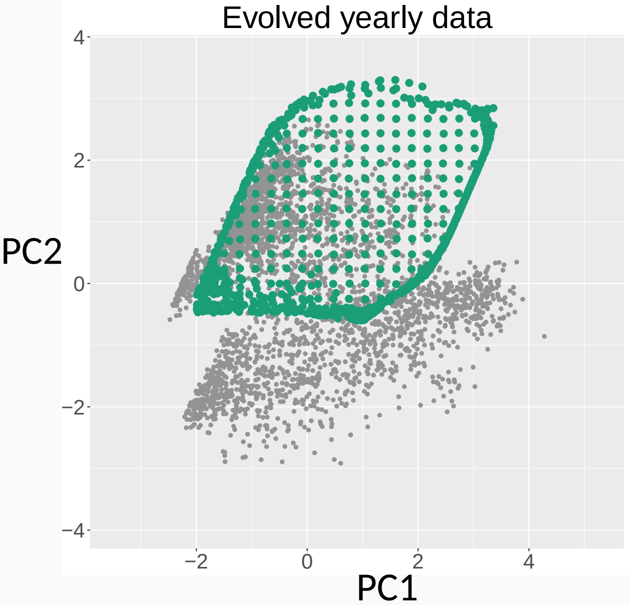
Evolving new time series



Evolving new time series

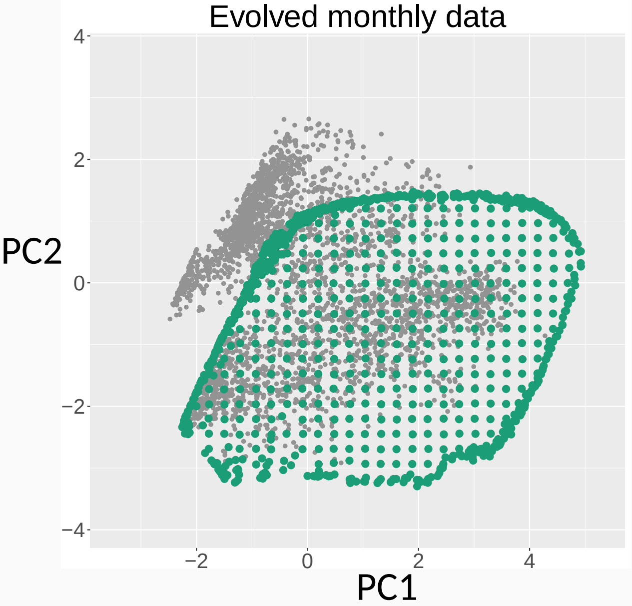


Evolving new time series

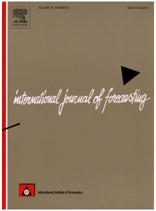


Evolving new time series

Evolving new time series



Papers and packages



Kang, Hyndman, & Smith-Miles, K. (2017) Visualising forecasting algorithm performance using time series instance spaces. *IJF*, **33**(2) 345–358.

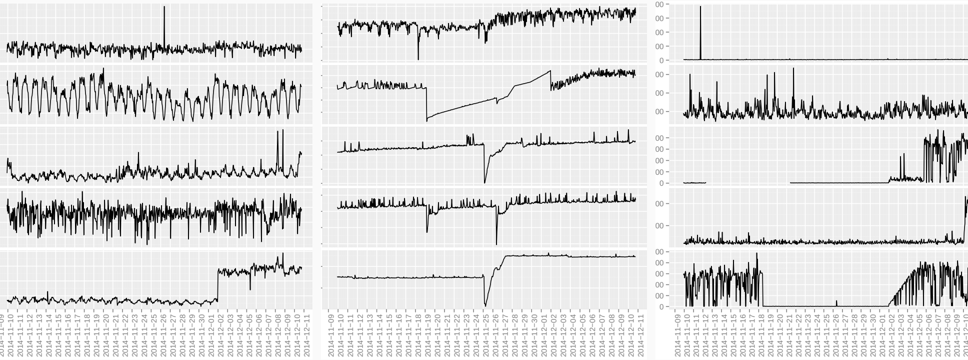
Hyndman, Wang, Kang, Talagala & Montero-Manso (2018). **tsfeatures**: Time Series Feature Extraction. `github.com/robjhyndman/tsfeatures/`

Outline

Yahoo web-traffic

- Tens of thousands of time series collected at one-hour intervals over one month.
- Consisting of several server metrics (e.g. CPU usage and paging views) from many server farms globally.
- Aim: find unusual (anomalous) time series.

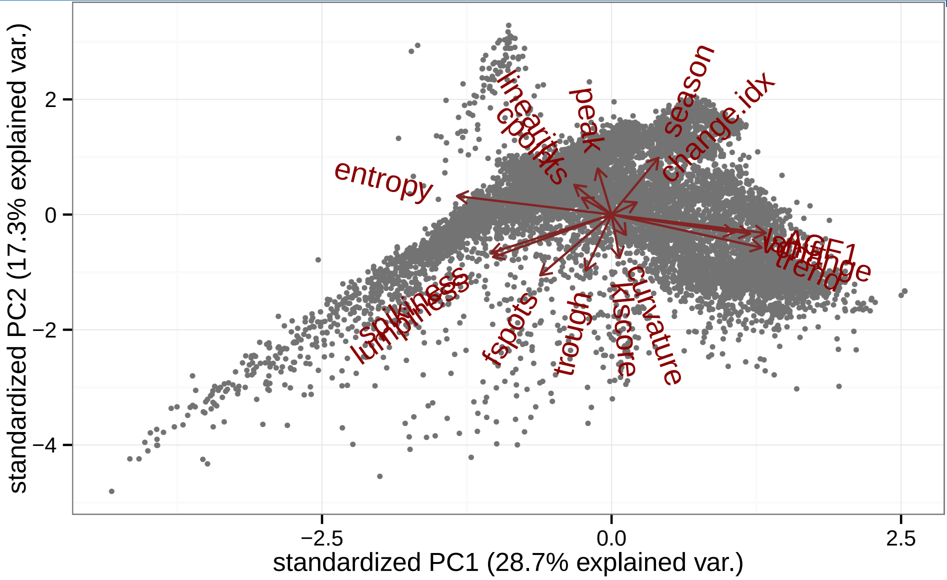
Yahoo web-traffic



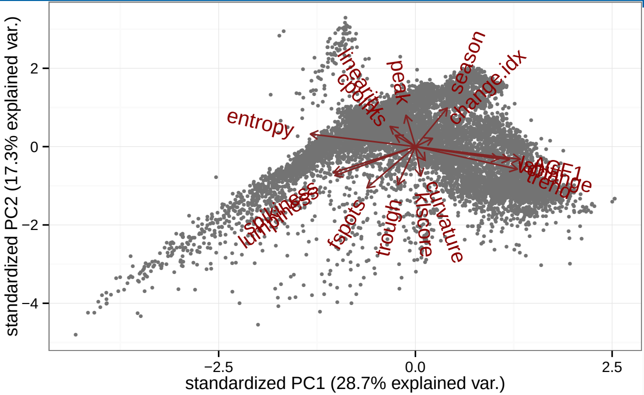
Feature space

- **ACF1**: first order autocorrelation = $\text{Corr}(Y_t, Y_{t-1})$
- Strength of **trend** and **seasonality** based on STL
- Trend **linearity** and **curvature**
- Size of seasonal **peak** and **trough**
- Spectral **entropy**
- **Lumpiness**: variance of block variances (block size 24).
- **Spikiness**: variances of leave-one-out variances of STL remainders.
- **Level shift**: Maximum difference in trimmed means of consecutive moving windows of size 24.
- **Variance change**: Max difference in variances of consecutive moving windows of size 24.
- **Flat spots**: Discretize sample space into 10 equal-sized intervals. Find max run length in any interval.
- Number of **crossing points** of mean line.
- **Kullback-Leibler score**: Maximum of $D_{KL}(P\|Q) = \int P(x) \ln P(x)/Q(x) dx$ where P and Q are estimated by kernel density estimators applied to consecutive windows of size 48.
- **Change index**: Time of maximum KL score

Principal component analysis

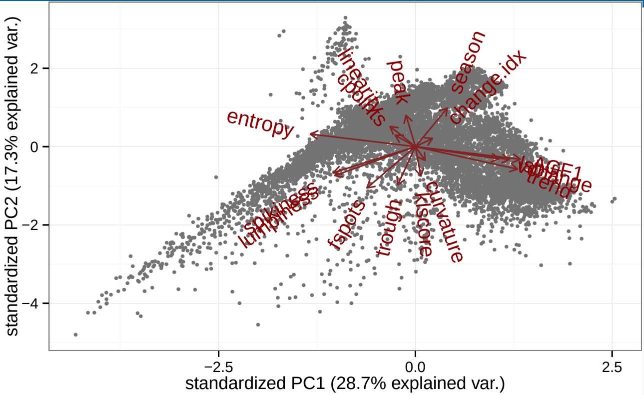


What is “anomalous”?



We need a measure of the “anomalousness” of a time series.

What is “anomalous”?



We need a measure of the “anomalousness” of a time series.

Idea: We rank points based on their local density using a bivariate kernel density estimate.

Bivariate kernel density

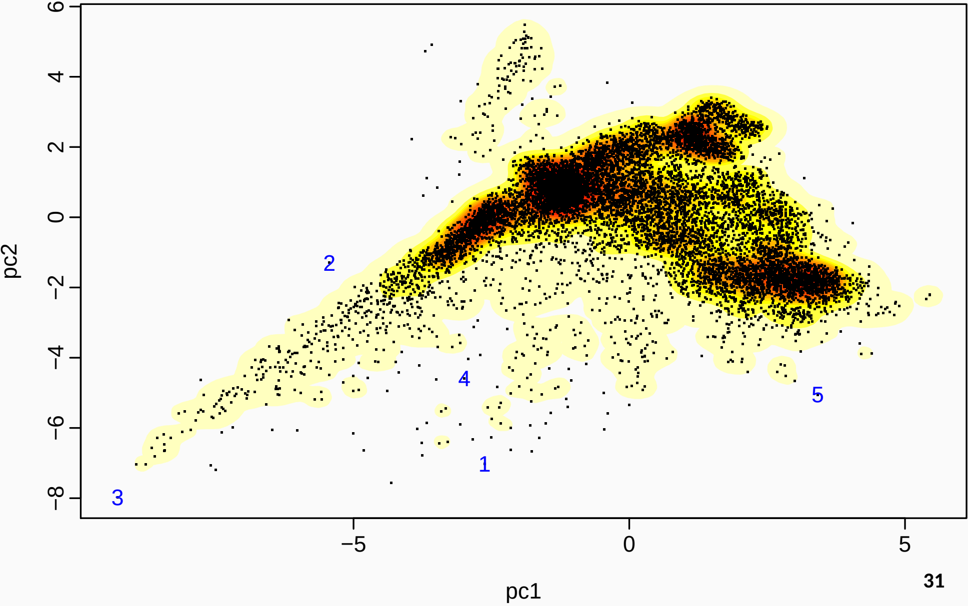
$$\hat{f}(\mathbf{x};\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{X}_i)$$

Bivariate kernel density

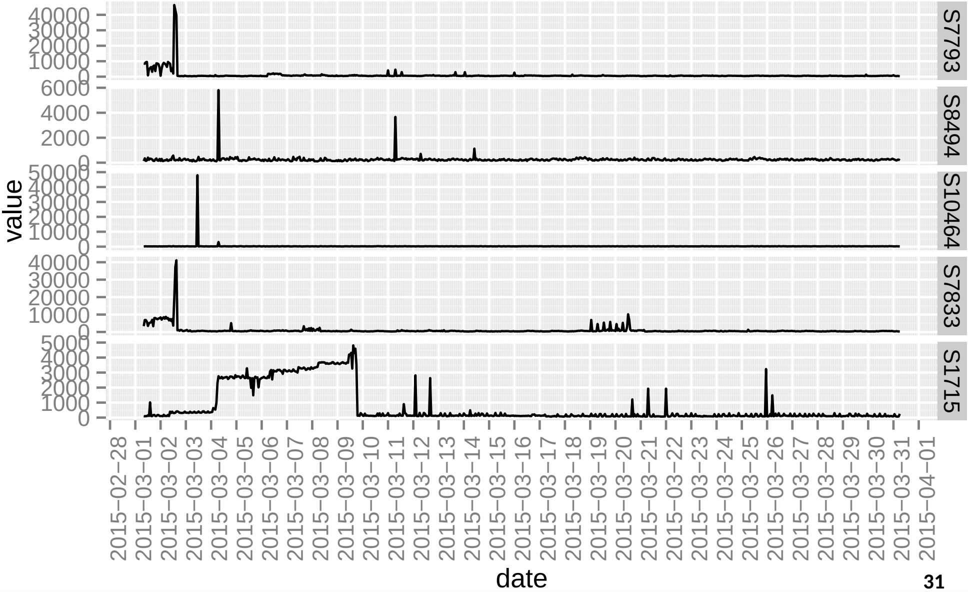
$$\hat{f}(\mathbf{x};\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i)$$

- $\mathbf{X}_i \in$ a bivariate random sample $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$
- $K_{\mathbf{H}}(\mathbf{x})$ is the standard normal kernel function
- $\mathbf{H}$ estimated by minimizing the sum of AMISE
- Rank points based on $\hat{f}$ values in 2d PCA space.

Bivariate density ranking

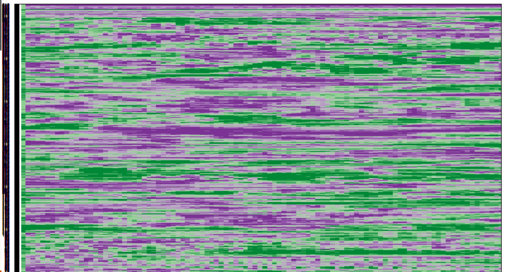


Bivariate density ranking



Security monitoring

Security monitoring



Time series anomaly detection

- Density-based outliers vs distance-based outliers.
- Fast feature calculation using windows needed for streaming data.
- **Oddstream**: Density-based anomalies, requiring a “typical” training period.
- **Stray**: Distance-based anomalies, requiring no “typical” training period.

Papers and packages

Hyndman, Wang & Laptev (2015). Large-scale unusual time series detection. *Proceedings of the IEEE International Conference on Data Mining*.


MONASH University

Fast computation of reconciled forecasts for hierarchical and grouped time series

Talagala, Hyndman, Smith-Miles, Kandanaarachchi & Muñoz (2018) Anomaly detection in streaming nonstationary temporal data. `robjhyndman.com/publications/oddstream/`

Hyndman, Wang, Kang, Talagala & Montero-Manso (2018). **tsfeatures**: Time Series Feature Extraction. `github.com/robjhyndman/tsfeatures/`

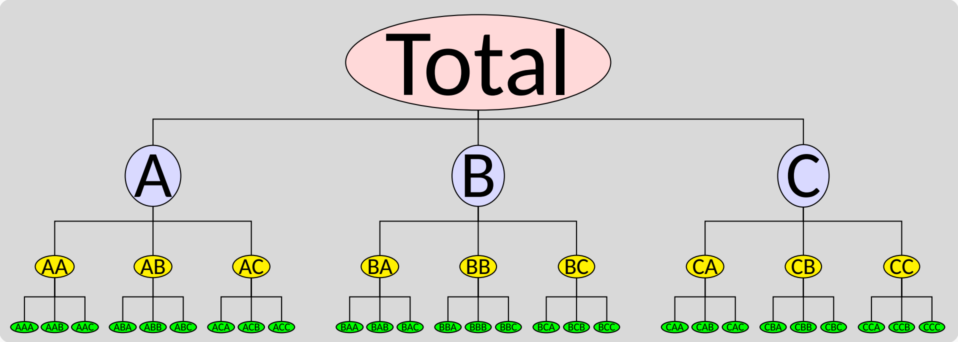
Talagala, Hyndman & Smith-Miles (2018) **oddstream**: Outlier Detection in Data Streams. `github.com/pridiltal/oddstream/`

Hyndman, Wang, Kang, Talagala & Montero-Manso (2018). **stray**: Robust Anomaly Detection in Data Streams with Concept Drift. `github.com/pridiltal/stray/`

Outline

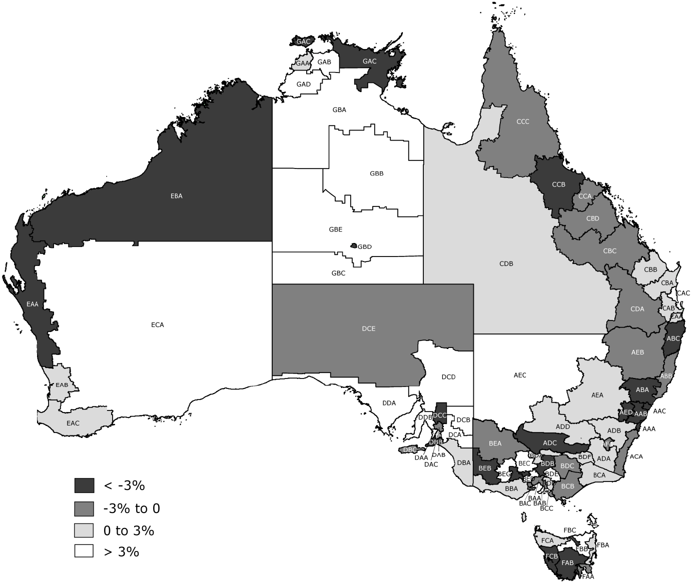
1. Visualizing many time series
2. Finding weird time series
3. **Reconciling many forecasts**
4. Forecasting temporal hierarchies

Forecast reconciliation



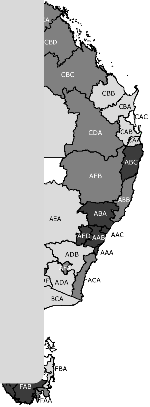
- Walmart sales by division, group, sub-group, etc.
- Australian tourism demand by state, region, zone.

Australian tourism demand

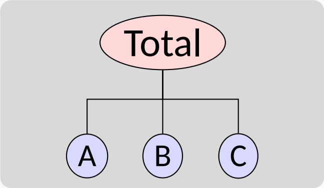


Australian tourism demand

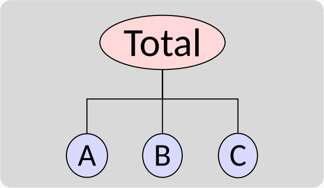
- Quarterly data on visitor night from 1998:Q1 – 2013:Q4
- From *National Visitor Survey*, based on annual interviews of 120,000 Australians aged 15+, collected by Tourism Research Australia.
- Split by 7 states, 27 zones and 76 regions (a geographical hierarchy)
- Also split by purpose of travel
    - Holiday
    - Visiting friends and relatives (VFR)
    - Business
    - Other
- 304 bottom-level series



Hierarchical time series



Hierarchical time series

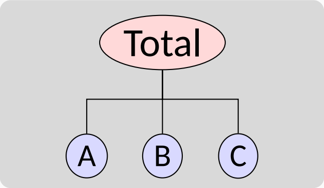


y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

Hierarchical time series



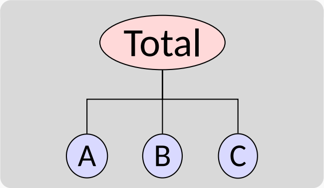
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

Hierarchical time series



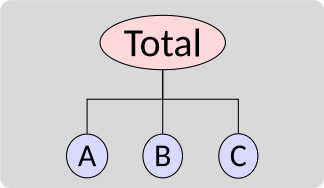
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

Hierarchical time series



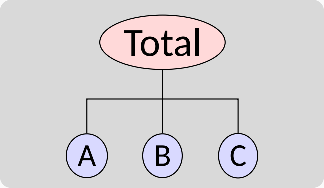
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

Hierarchical time series



y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Disaggregated time series

Every collection of time series with aggregation constraints can be written as

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

where

- $\boldsymbol{y}_t$ is a vector of all series at time t
- $\boldsymbol{b}_t$ is a vector of the most disaggregated series at time t
- $\boldsymbol{S}$ is a "summing matrix" containing the aggregation constraints.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (In general, they will not "add up".)

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ maps base forecasts $\hat{\boldsymbol{y}}_n(h)$ to bottom level.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ maps base forecasts $\hat{\boldsymbol{y}}_n(h)$ to bottom level.
- $\boldsymbol{S}$ adds them up

General properties: bias and variance

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

General properties: bias and variance

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Bias

Reconciled forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

General properties: bias and variance

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Bias

Reconciled forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

Variance

Let error variance of h-step base forecasts $\hat{\boldsymbol{y}}_n(h)$ be

$$\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

Then error variance of the reconciled forecasts is

$$\text{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}'$$

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Theorem: MinT Reconciliation

If $\boldsymbol{P}$ satisfies $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Theorem: MinT Reconciliation

If $\boldsymbol{P}$ satisfies $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$.

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

Reconciled forecasts

Base forecasts

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Assume that $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ to simplify computations.

Optimal forecast reconciliation

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

Reconciled forecasts Base forecasts

Assume that $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ to simplify computations.

WLS solution

- Approximate $\boldsymbol{W}_1$ by its diagonal.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Assume that $\boldsymbol{W}_h = k_h \boldsymbol{W}_1$ to simplify computations.

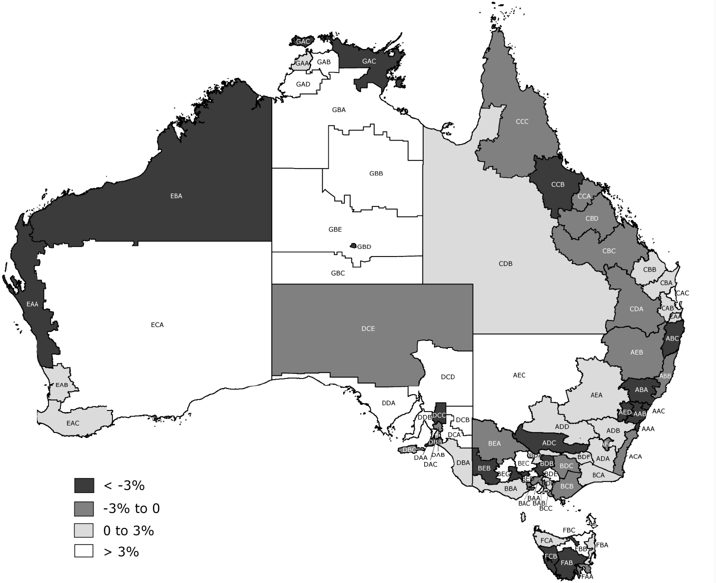
WLS solution

- Approximate $\boldsymbol{W}_1$ by its diagonal.

GLS solution

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal.

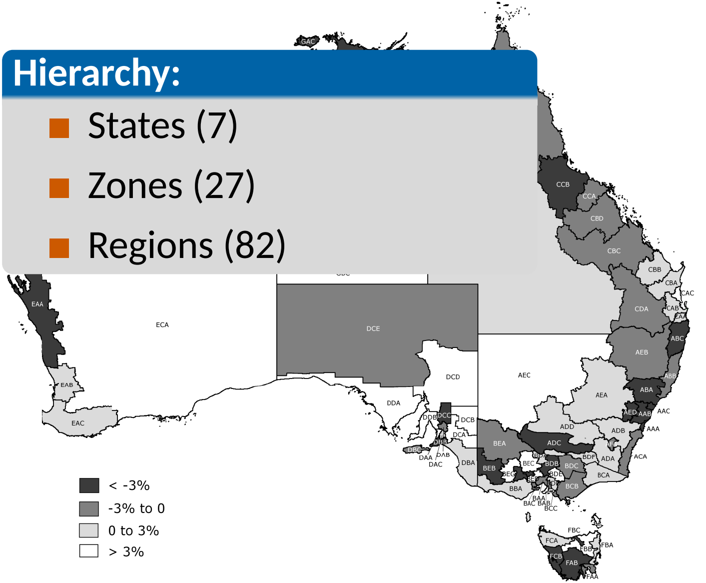
Australian tourism



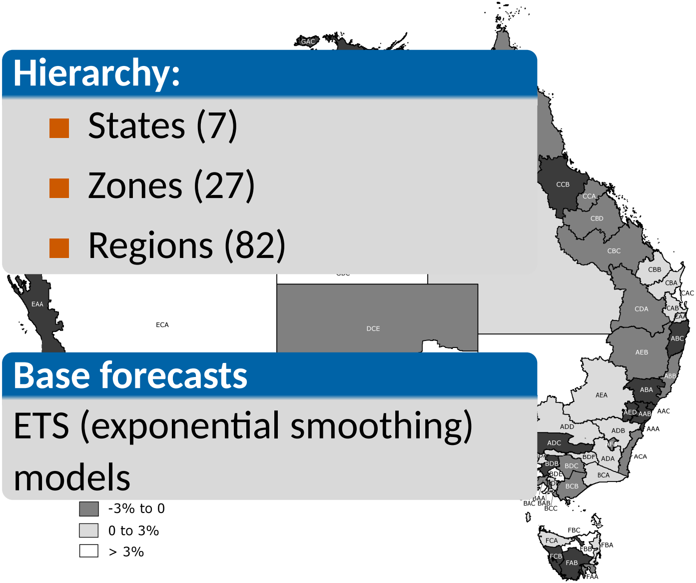
Australian tourism

Hierarchy:

- States (7)
- Zones (27)
- Regions (82)



Australian tourism



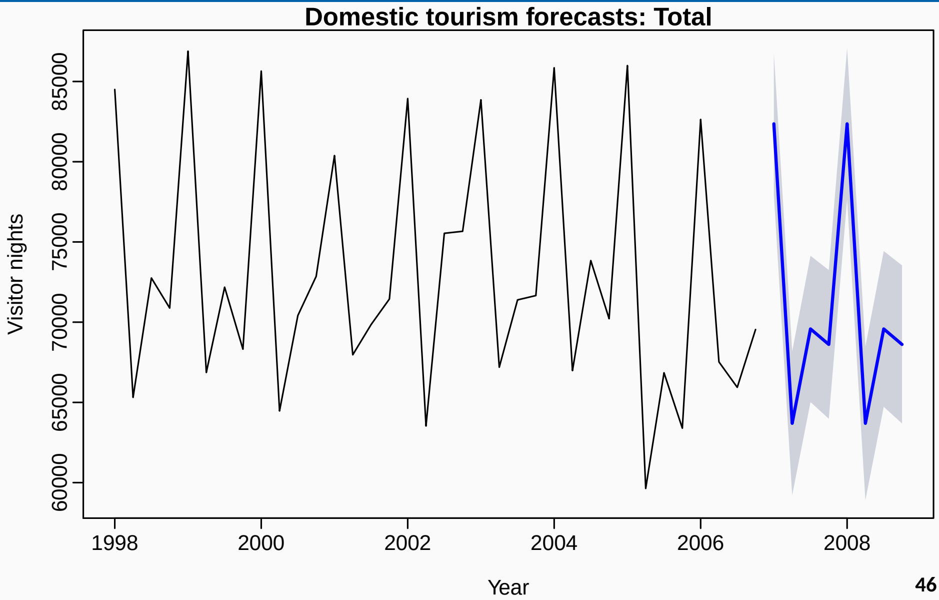
Hierarchy:

- States (7)
- Zones (27)
- Regions (82)

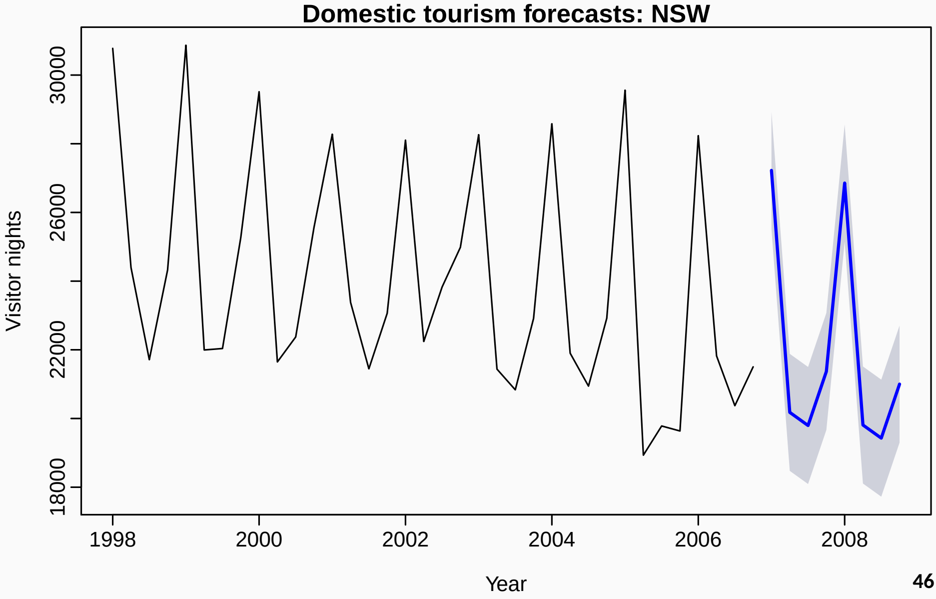
Base forecasts

ETS (exponential smoothing) models

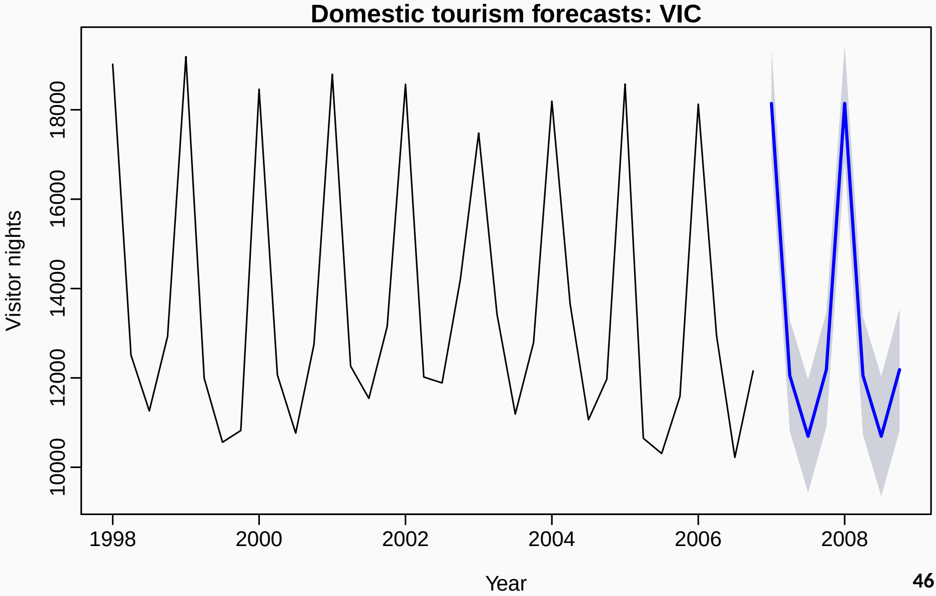
Base forecasts



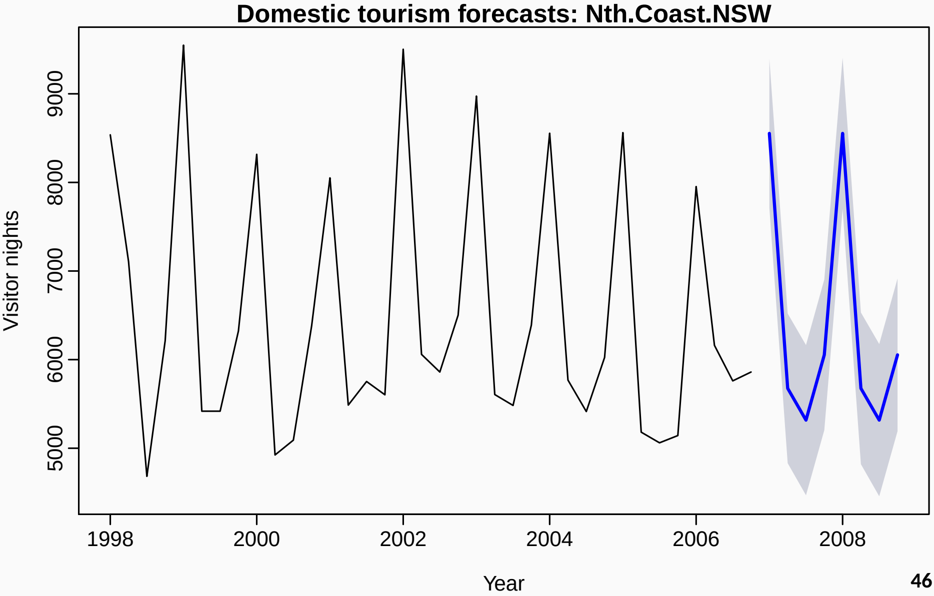
Base forecasts



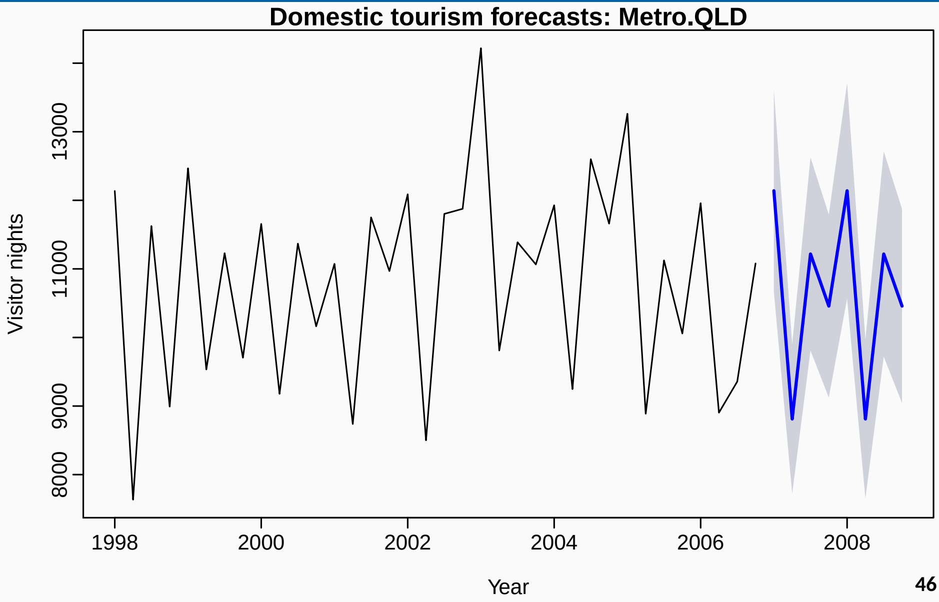
Base forecasts



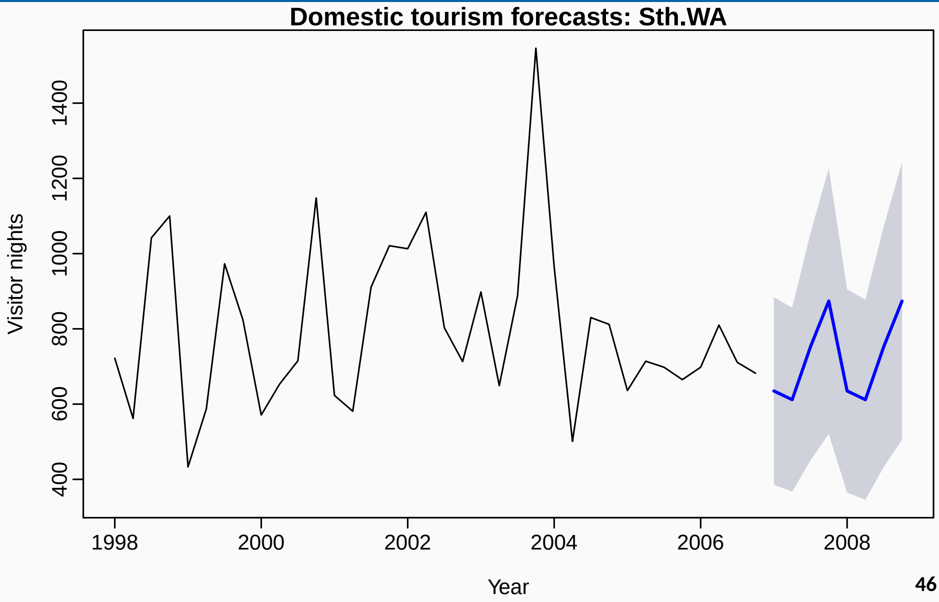
Base forecasts



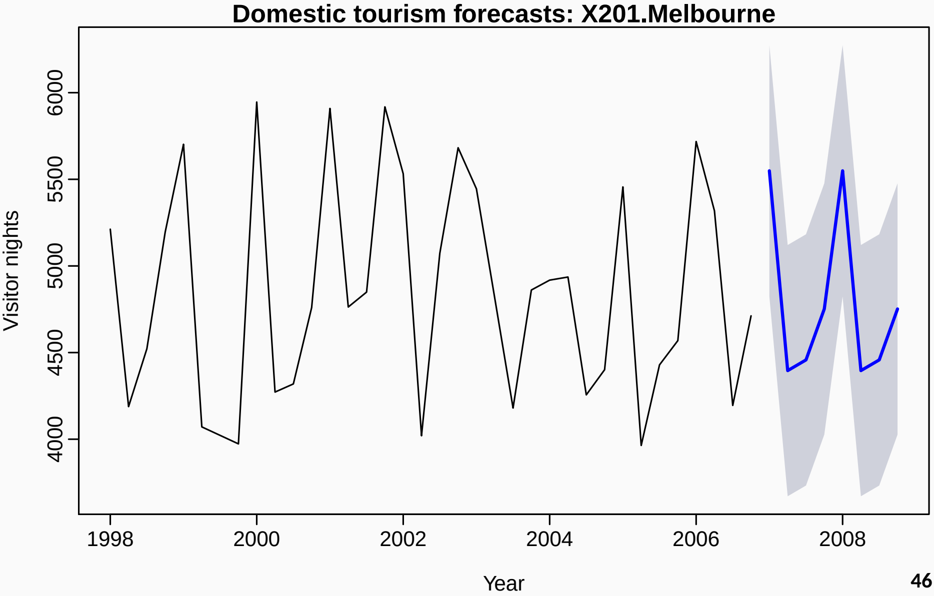
Base forecasts



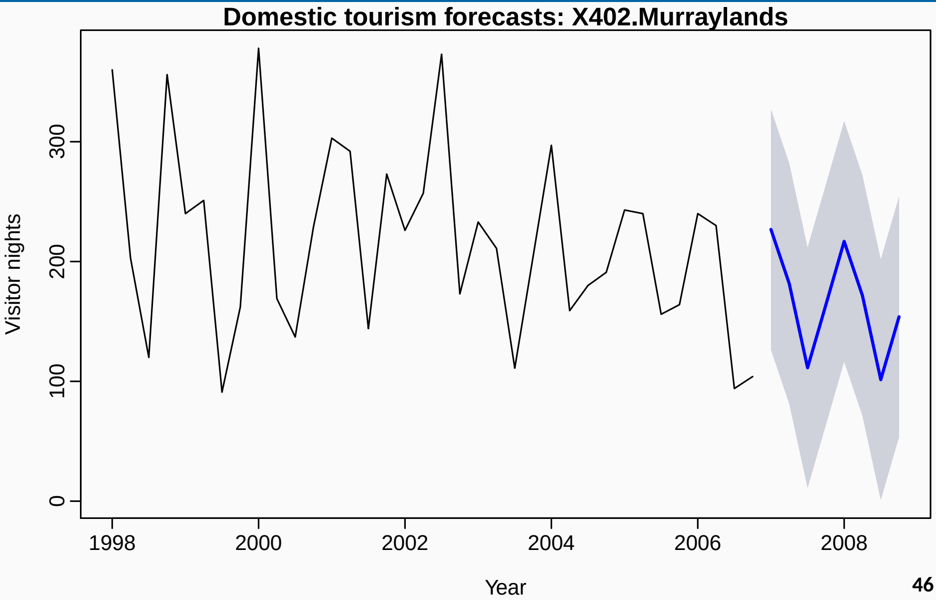
Base forecasts



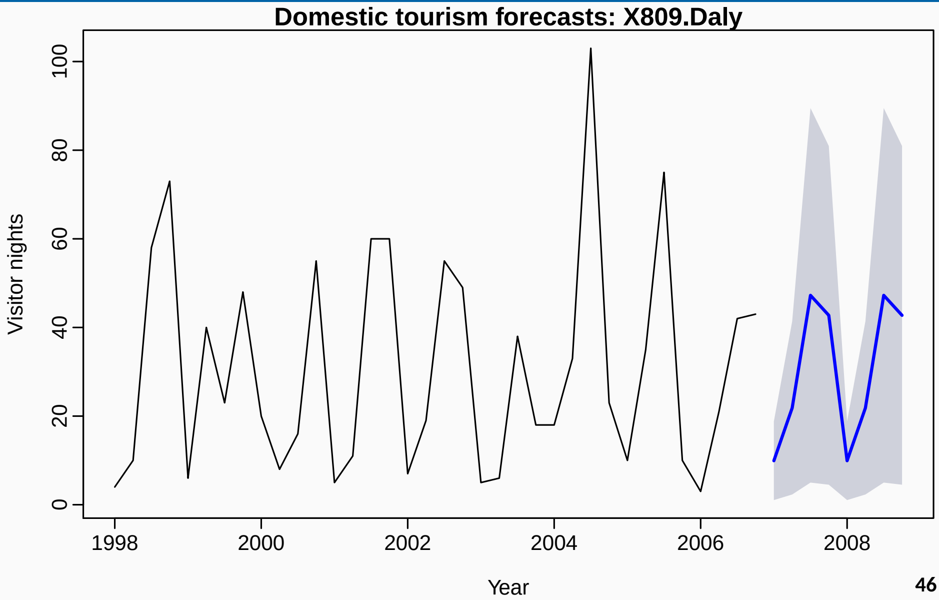
Base forecasts



Base forecasts

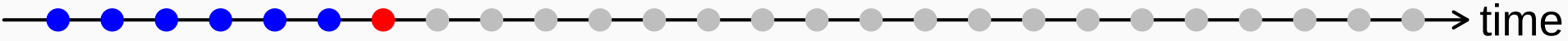


Base forecasts



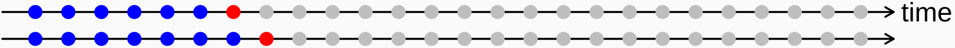
Forecast evaluation

Training sets **Test sets $h = 1$**



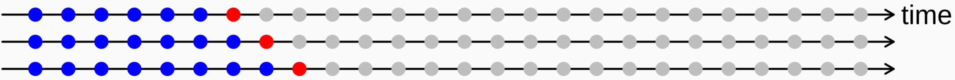
Forecast evaluation

Training sets **Test sets $h = 1$**



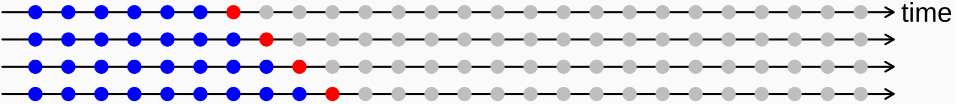
Forecast evaluation

Training sets **Test sets $h = 1$**



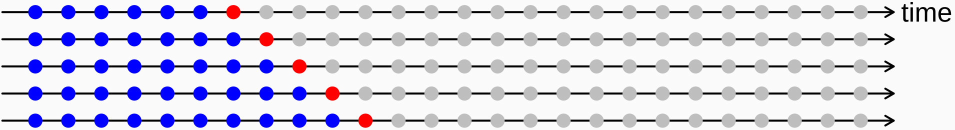
Forecast evaluation

Training sets **Test sets $h = 1$**



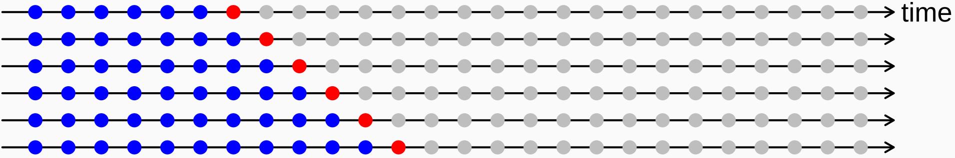
Forecast evaluation

Training sets **Test sets $h = 1$**



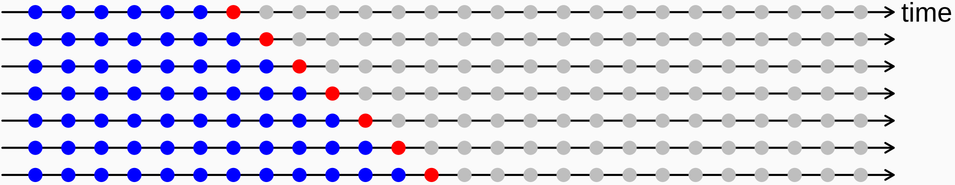
Forecast evaluation

Training sets **Test sets $h = 1$**



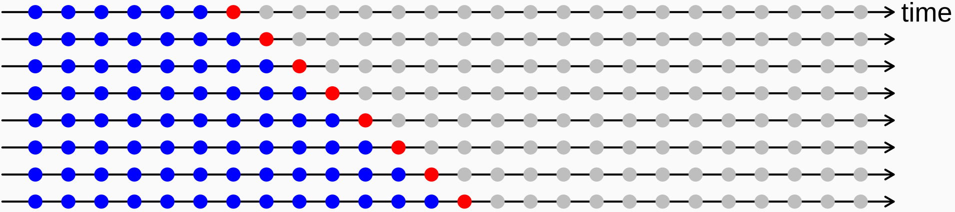
Forecast evaluation

Training sets **Test sets $h = 1$**



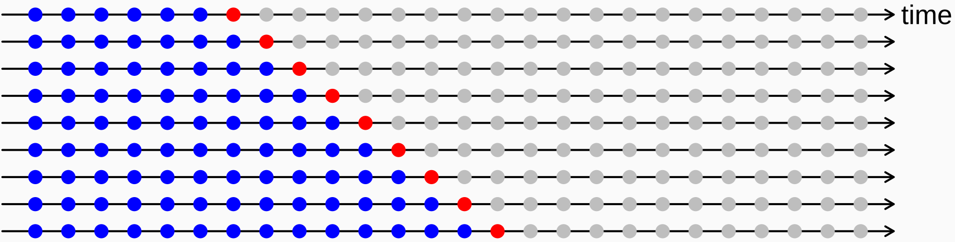
Forecast evaluation

Training sets **Test sets $h = 1$**



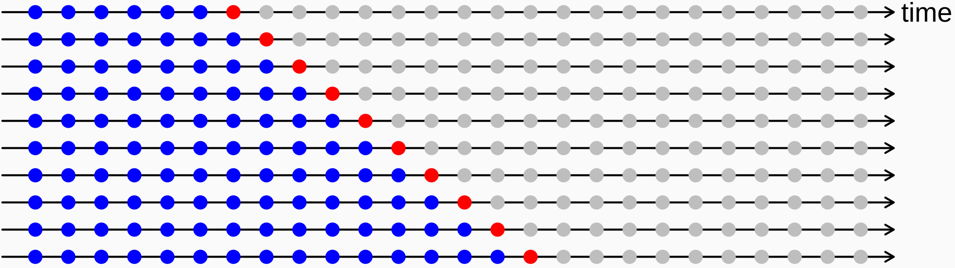
Forecast evaluation

Training sets **Test sets $h = 1$**



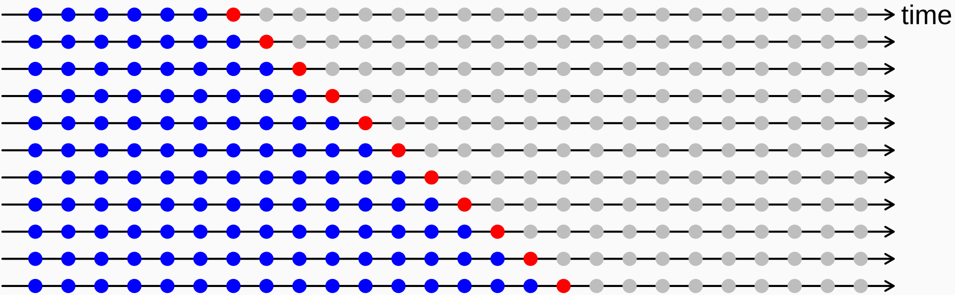
Forecast evaluation

Training sets **Test sets $h = 1$**



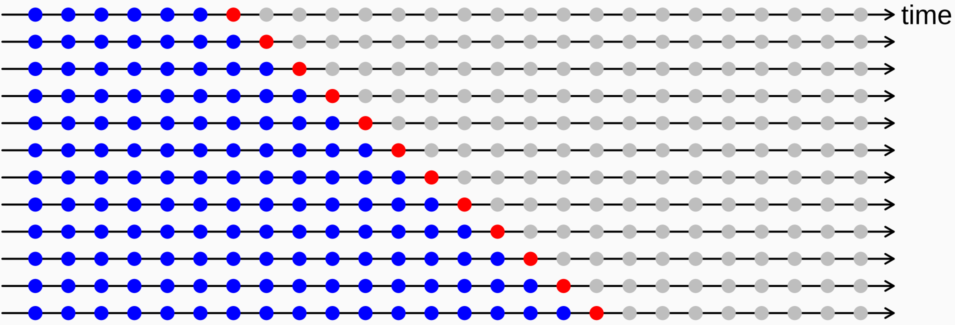
Forecast evaluation

Training sets **Test sets $h = 1$**



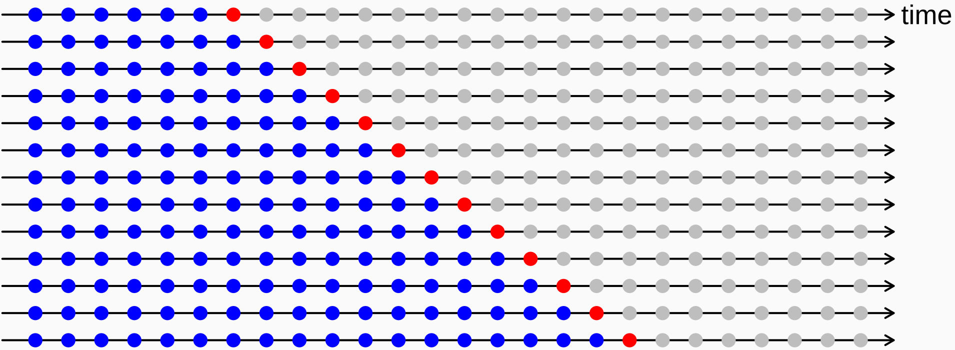
Forecast evaluation

Training sets **Test sets $h = 1$**



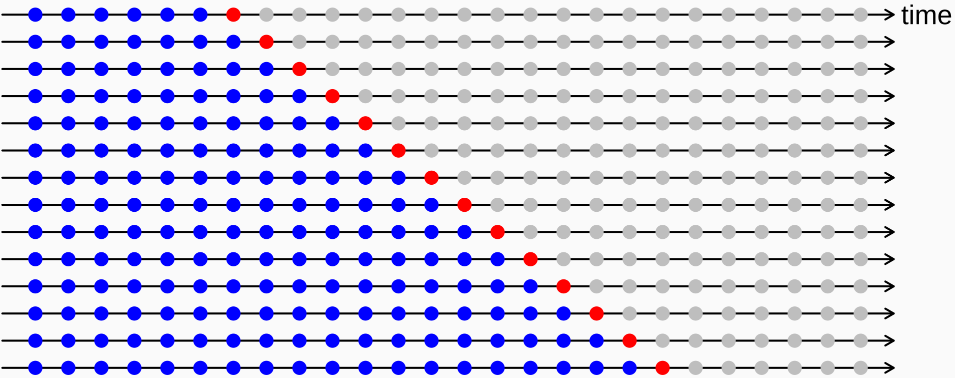
Forecast evaluation

Training sets **Test sets $h = 1$**



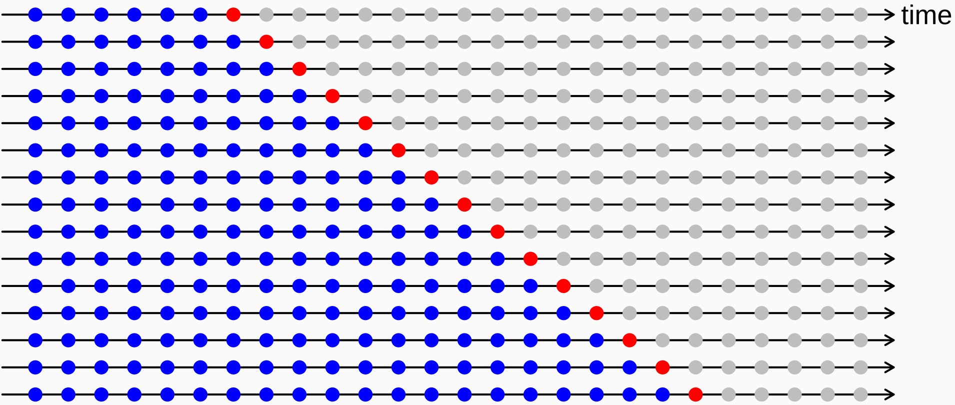
Forecast evaluation

Training sets **Test sets $h = 1$**



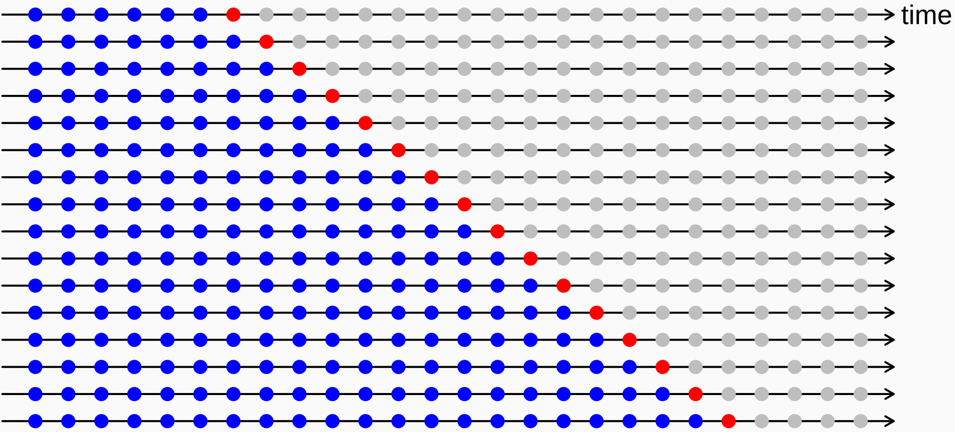
Forecast evaluation

Training sets **Test sets $h = 1$**



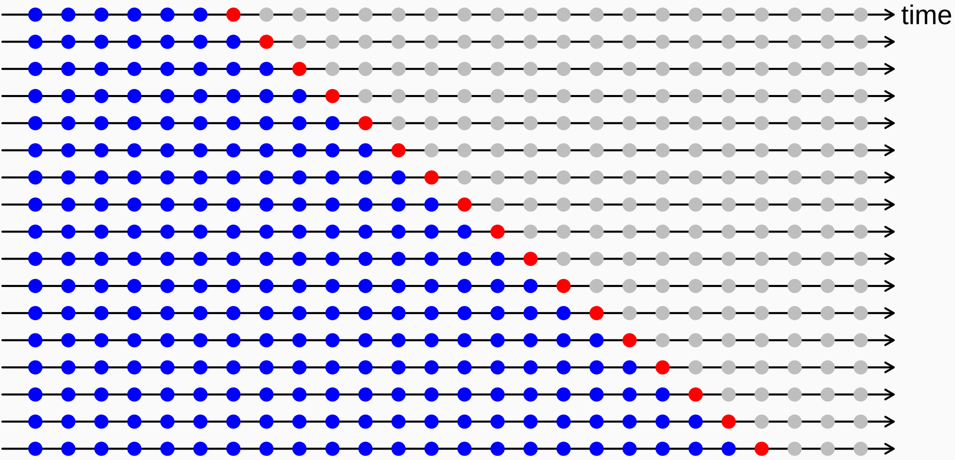
Forecast evaluation

Training sets **Test sets $h = 1$**



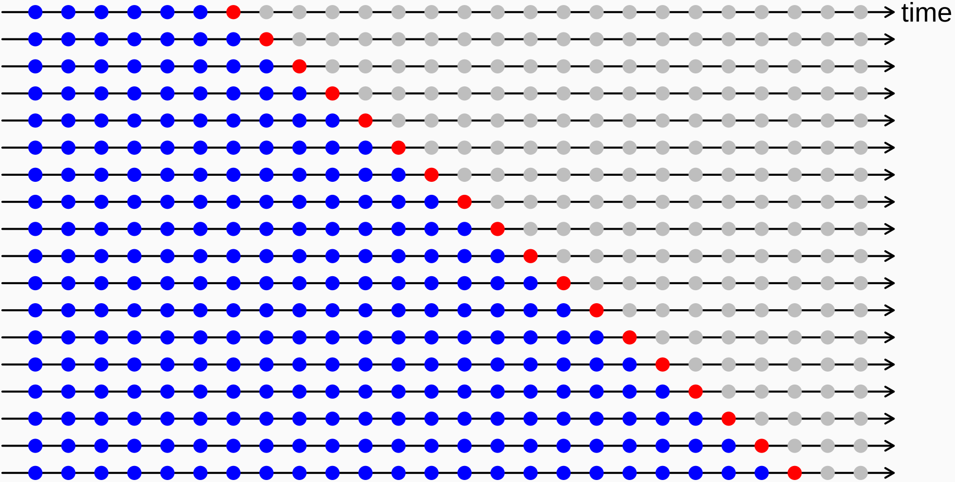
Forecast evaluation

Training sets **Test sets $h = 1$**

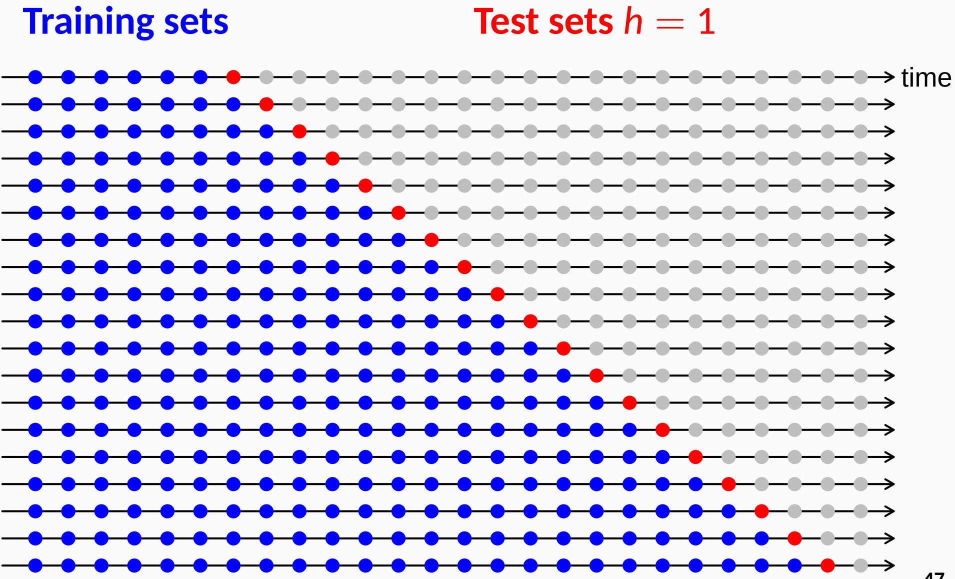


Forecast evaluation

Training sets **Test sets $h = 1$**

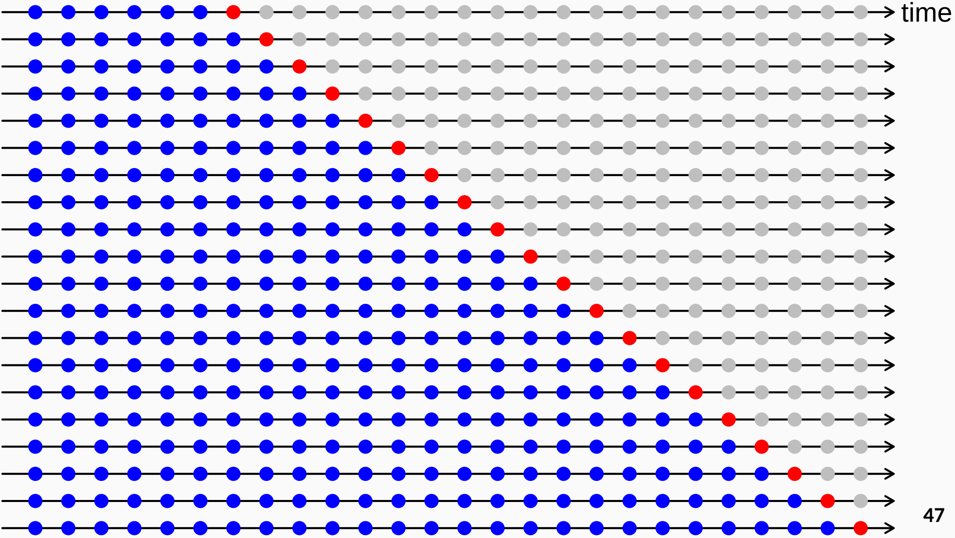


Forecast evaluation

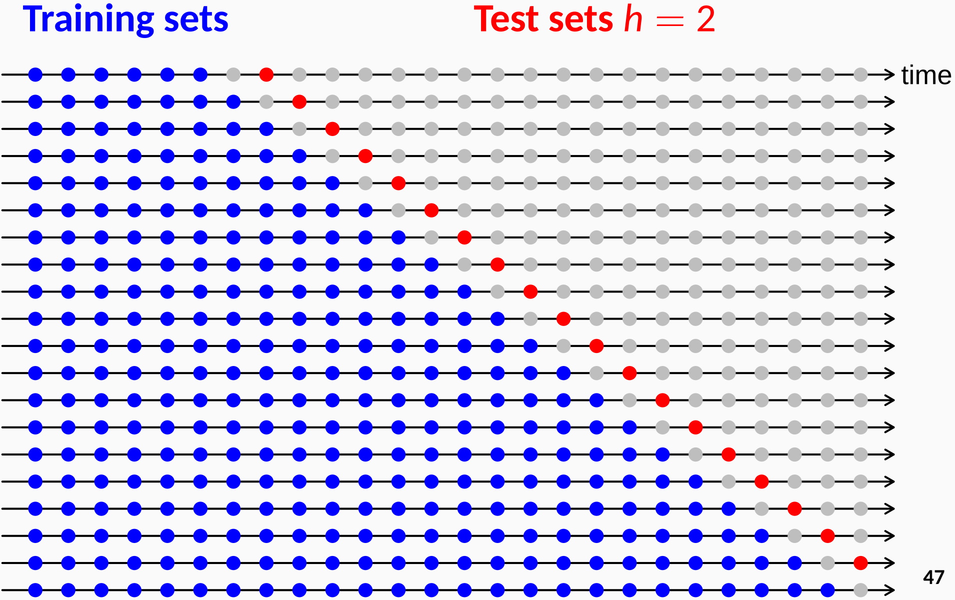


Forecast evaluation

Training sets **Test sets $h = 1$**

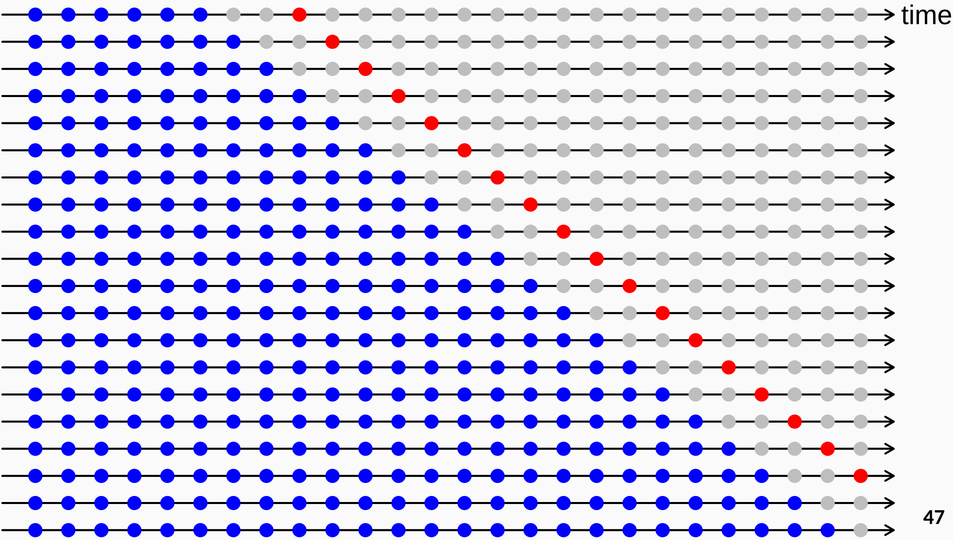


Forecast evaluation



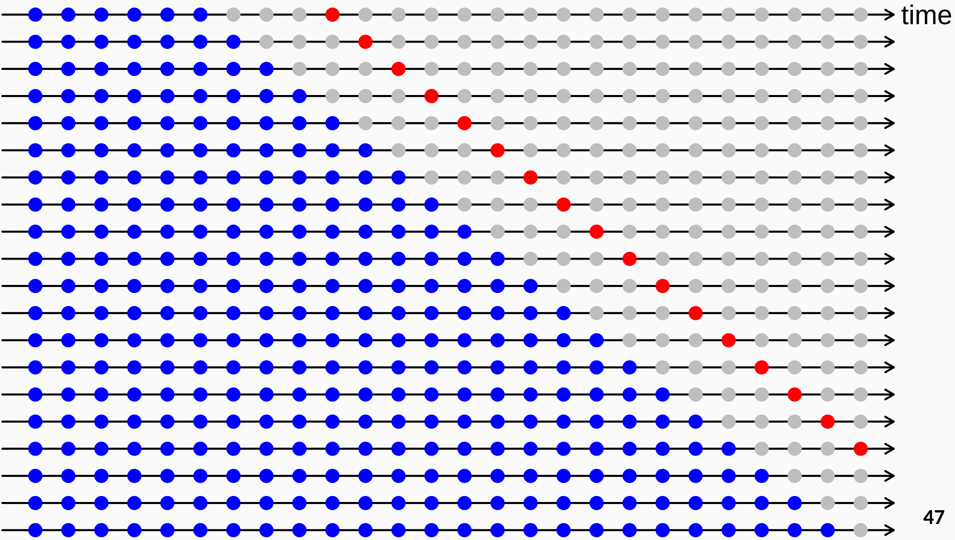
Forecast evaluation

Training sets **Test sets $h = 3$**

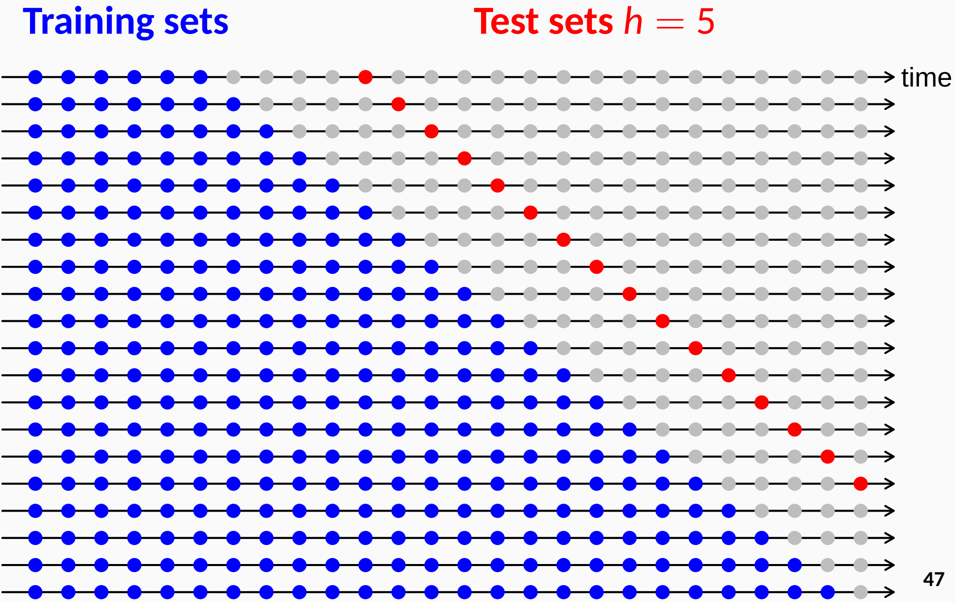


Forecast evaluation

Training sets **Test sets $h = 4$**

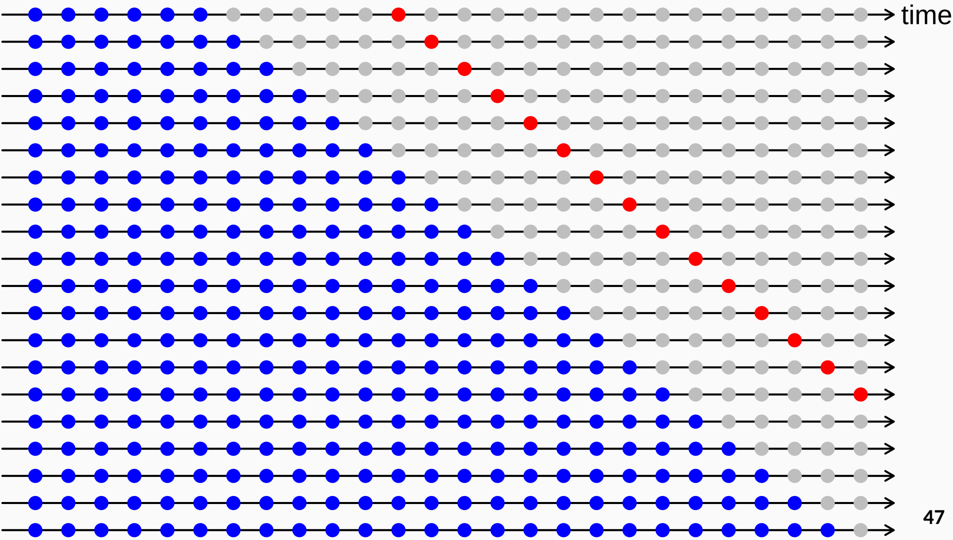


Forecast evaluation



Forecast evaluation

Training sets **Test sets $h = 6$**



Hierarchy: states, zones, regions

RMSE	$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=6$	Ave
			Forecast horizon				
			Australia				
Base	1762.04	1770.29	1766.02	1818.82	1705.35	1721.17	**1757.28**
Bottom	1736.92	1742.69	1722.79	1752.74	1666.73	1687.43	**1718.22**
WLS	1705.21	1715.87	1703.75	1729.56	1627.79	1661.24	**1690.57**
GLS	1704.64	1715.60	1705.31	1729.04	1626.36	1661.64	**1690.43**
			States				
Base	399.77	404.16	401.92	407.26	395.38	401.17	**401.61**
Bottom	404.29	406.95	404.96	409.02	399.80	401.55	**404.43**
WLS	398.84	402.12	400.71	405.03	394.76	398.23	**399.95**
GLS	398.84	402.16	400.86	405.03	394.59	398.22	**399.95**
			Regions				
Base	93.15	93.38	93.45	93.79	93.50	93.56	**93.47**
Bottom	93.15	93.38	93.45	93.79	93.50	93.56	**93.47**
WLS	93.02	93.32	93.38	93.72	93.39	93.53	**93.39**
GLS	92.98	93.27	93.34	93.66	93.34	93.46	**93.34**

Papers and packages

Hyndman, Ahmed, Athanasopoulos & Shang (2011) Optimal combination forecasts for hierarchical time series. *CSDA* **55**(9) 2579–2589

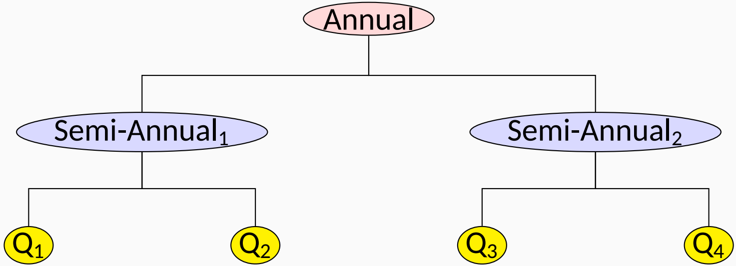
Wickramasuriya, Athanasopoulos & Hyndman (2018) Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *JASA*, to appear.

Hyndman, Bergmeir, Caceres, O'Hara-Wild, Razbash & Wang (2018). **forecast**: Forecasting functions for time series and linear models.
`pkg.robjhyndman.com/forecast/`

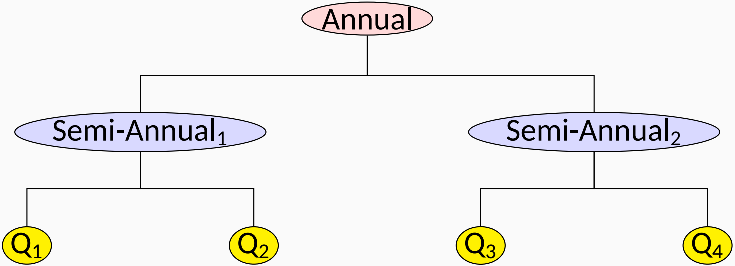
Hyndman, Lee, Wang & Wickramasuriya (2018). **hts**: Hierarchical and Grouped Time Series.
`pkg.earo.me/hts/`

Outline

Temporal hierarchies



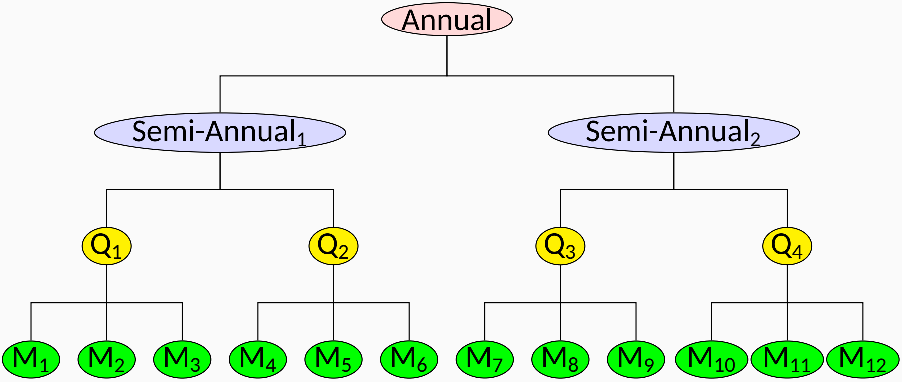
Temporal hierarchies



Basic idea:

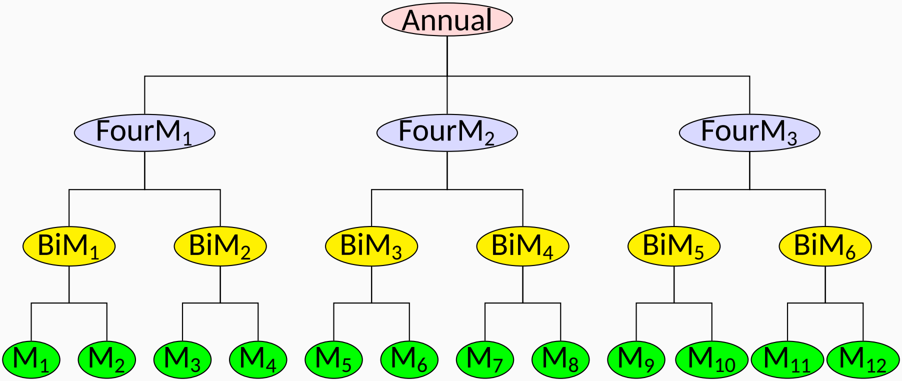
- Forecast series at each available frequency.
- Optimally reconcile forecasts within the same year.

Monthly series



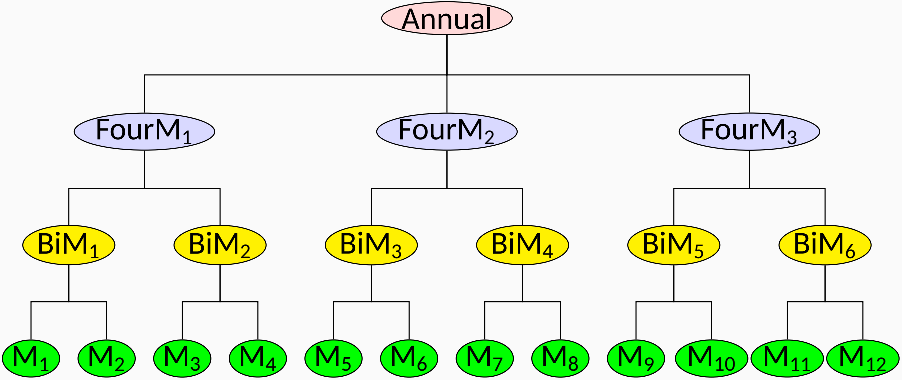
- $k = 2, 4, 12$ nodes

Monthly series



- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes

Monthly series



- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- Why not $k = 2, 3, 4, 6, 12$ nodes?

Monthly data

$$
\underbrace{\begin{pmatrix} A \\ SemiA_1 \\ SemiA_2 \\ FourM_1 \\ FourM_2 \\ FourM_3 \\ Q_1 \\ \vdots \\ Q_4 \\ BiM_1 \\ \vdots \\ BiM_6 \\ M_1 \\ \vdots \\ M_{12} \end{pmatrix}}_{(28\times 1)}
=
\underbrace{\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & \vdots & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & \vdots & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 & & & & & & & & & & & \\
 & & & & & & \boldsymbol{I}_{12} & & & & & \\
 & & & & & & & & & & &
\end{pmatrix}}_{\boldsymbol{S}}
\underbrace{\begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \\ M_{10} \\ M_{11} \\ M_{12} \end{pmatrix}}_{\boldsymbol{b}_t}
$$

In general

For a time series $y_1, \dots, y_T$, observed at frequency m, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.

In general

For a time series $y_1, \dots, y_T$, observed at frequency m, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.
- A single unique hierarchy is only possible when there are no coprime pairs in $F(m)$.

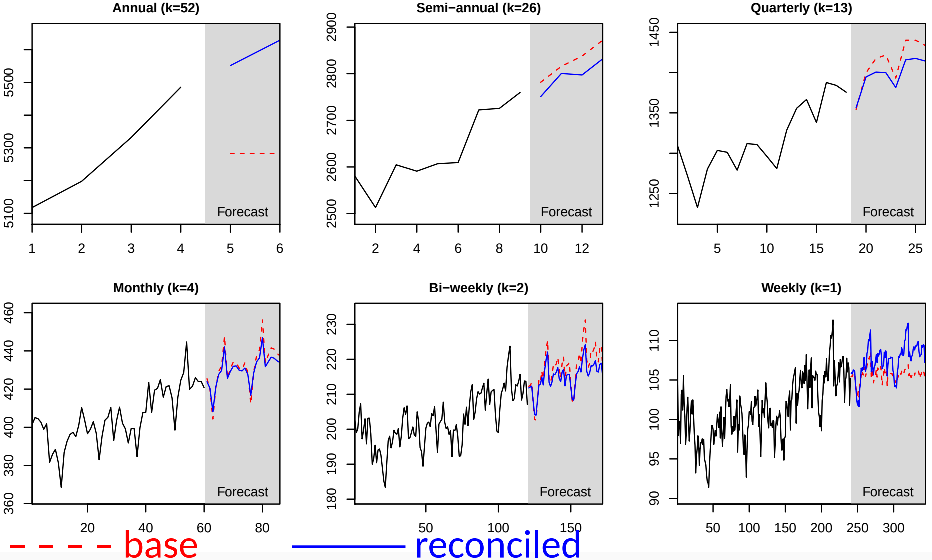
In general

For a time series $y_1, \dots, y_T$, observed at frequency m, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.
- A single unique hierarchy is only possible when there are no coprime pairs in $F(m)$.
- $M_k = m/k$ is seasonal period of aggregated series.

UK Accidents and Emergency Demand



Papers and packages

Athanasopoulos, Hyndman, Kourentzes & Petropoulos (2017) Forecasting with temporal hierarchies. *EJOR*, **262**(1) 60–74.

Hyndman & Kourentzes (2017). **thief**: Temporal HIErarchical Forecasting
`pkg.robjhyndman.me/thief/`

Papers, packages and slides available at **robjhyndman.com**