Forecasting for Data Scientists

Theory Session 1 – Forecasting Basics

Christoph Bergmeir

October, 2025

Universidad de Granada, Spain
Monash University, Melbourne, Australia
https://www.cbergmeir.com

Introduction

What this training course is about

- This training course will give you a broad overview over most (all?) relevant aspects of forecasting
- Traditional stats methods, traditional machine learning
- Deep learning, foundational models (GPT-like models)
- Won't go into much detail for any of them
- Give you an overview and general understanding of where the field is at, its history, main techniques and solutions, main problems

The main intention of this training course is that you understand the problems and their general solutions in forecasting, both for modelling and evaluation, so that you are enabled to use and judge results of complicated ML methods.

My background

- Have a Dipl.-Inf. degree from University of Ulm, Germany
- Have been in forecasting since the start of my PhD at University of Granada (Spain), in 2009
- Have worked at Monash University (Australia), Dept. Data Science and AI, from 2013-2022 (as Senior Lecturer in the end)
- In 2022/2023 I did a 6-months sabbatical at Meta Inc.
- I'm now back to University of Granada, Spain, as a María Zambrano (Senior) Fellow.
- I'm still an Adjunct Senior Research Fellow at Monash, DSAI

What I forecast

- Sales/demand forecasting in the supply chain, retail
- Forecasting in Energy
  - Demand
  - Prices
  - Renewable energy production
  - building efficiency
- Predictive maintenance (road, railway, ...mining)
- ...

What people think I forecast (but I don't)

- The weather
    - Lots of domain knowledge and specialised models exist
    - We leave it to the meteorologists
    - We often use weather forecasts as inputs to our models
- The stock market

Stock market forecasting (1)

Pros:

- Lots of freely available high-frequency data
- Clear motivation and use case (making money)

Stock market forecasting (2)

Cons:

- Shareprice not a function of its own past, but of its anticipated future
- Low signal to noise ratio
- Markets tend to be close to "efficient"
- Forecasting in efficient markets is not possible (beyond the naive forecast)
- R. Engle: got the Nobel prize in Economics in 2003 "for methods of analyzing economic time series with time-varying volatility (ARCH)."
- All about predicting risk, not returns (i.e., predicting the tails of the distributions)

Stock market forecasting (3)

$\rightarrow$ Predicting stock market returns from just a couple of lags won't work, no matter what method you are using.

$\rightarrow$ All this is true in a similar way for exchange rates!

What can we forecast?

- Fundamental assumption: Future depends on the past
- We need past data/experience to extrapolate
- But predictability varies widely
- Some more examples:
    - next Monday's time of sunset
    - the TV schedule in 2 weeks' time
    - next Saturday's lottery numbers
    - financial variables: exchange rates, stock prices, etc.

What can we forecast? (2)

Important factors for predictability are:

- How well do we understand the underlying phenomena?
- How much data do we have?
- Will the forecast affect the future?

Forecasting with mathematical models:

- Model adequately repetitive patterns
- Do not let you distract by noise and unique, unforeseeable events

$\rightarrow$ Separate the time series into a part with predictive value and a part of noise

What can we forecast? (3)

- Forecasts are always wrong
- Use prediction intervals to quantify uncertainty
- If you don't have to forecast, then don't forecast!
  - forecasting usually for decision making
  - can the decision be made without a forecast?
  - focus on resilience of the decision-making
  - Bad idea: "If I had a perfect forecast, I could make an optimal decision"
- Forecasting will give you the biggest wins in situations where you are already using a (bad) ad-hoc forecast in your decision making

Traditional (statistical univariate) forecasting techniques

“Forecasting: principles and practice” book

The standard resource for traditional time series forecasting that I somewhat follow in this presentation is:

Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

The Python version using StatsForecast is:

Hyndman, R.J., Athanasopoulos, G., Garza, A., Challu, C., Mergenthaler, M., & Olivares, K.G. (2024). Forecasting: Principles and Practice, the Pythonic Way. OTexts: Melbourne, Australia. Available at: OTexts.com/fpppy

A bit of forecasting terminology

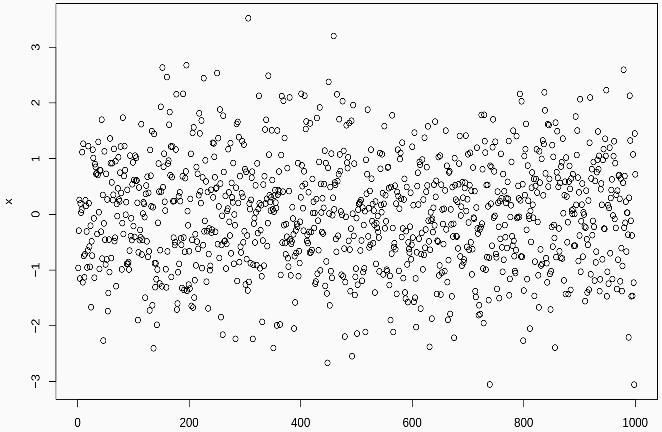
- model fitting, parameter estimation: model training
- in-sample: training set
- out-of-sample: test set
- forecast horizon: target variable
- forecast origin: from where you do the forecasting from, the last known observation
- rolling origin: the forecast origin changes (for every point in the test set)
- fixed origin: the forecast origin is fixed

A bit of forecasting terminology (2)

- forecast combination: ensembling
- lags, independent variables, regressors, covariates, predictors: features, inputs
- dummy variable, indicator variable: one-hot encoding
- seasonality, seasonal period: cyclic change in mean of the series. Its length is know a priori and will not change in the future
- trend: (smooth) change in the mean of the series

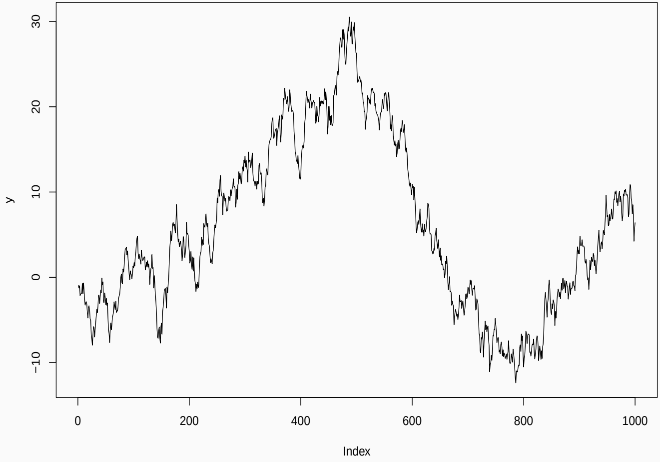
Some random numbers

```r
set.seed(3)
x <- rnorm(1000)
plot(x)
```



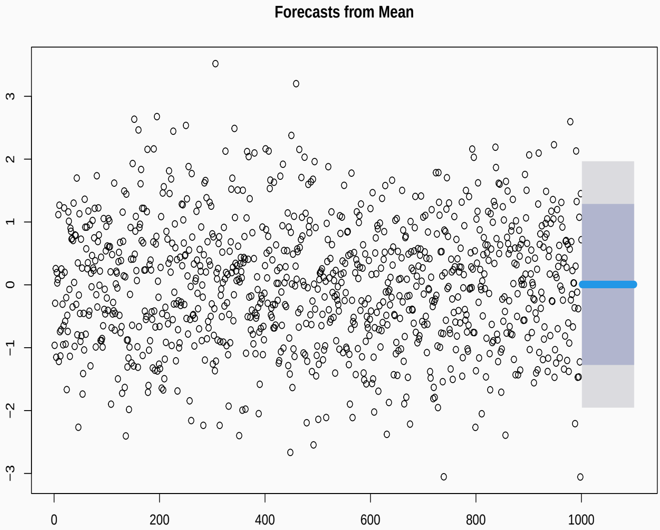
A random walk

```r
y <- cumsum(x)
plot(y, type="l")
```



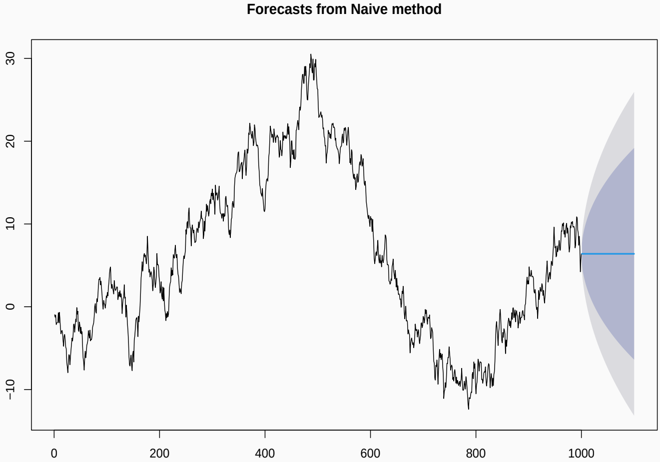
Mean forecast

```r
plot(meanf(x, h=100), type="l")
```



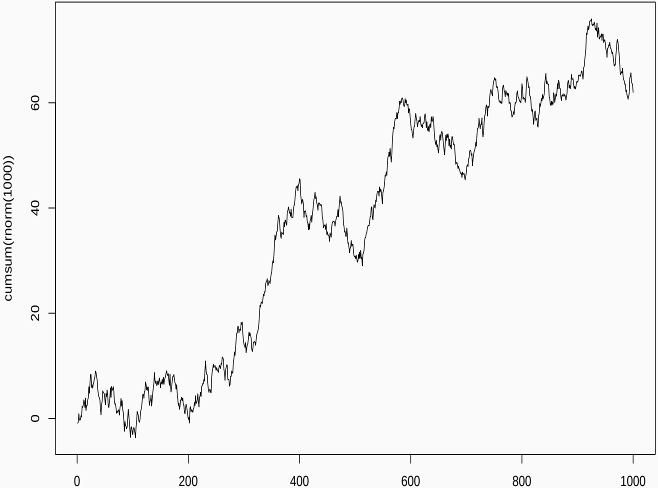
Naive (persistence/random walk) forecast

```
plot(naive(y, h=100))
```



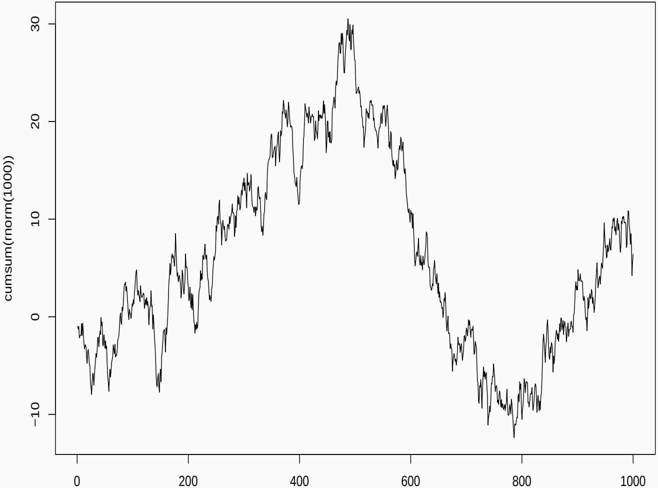
Random walk (3)

```r
set.seed(2)
plot(cumsum(rnorm(1000)), type="l")
```



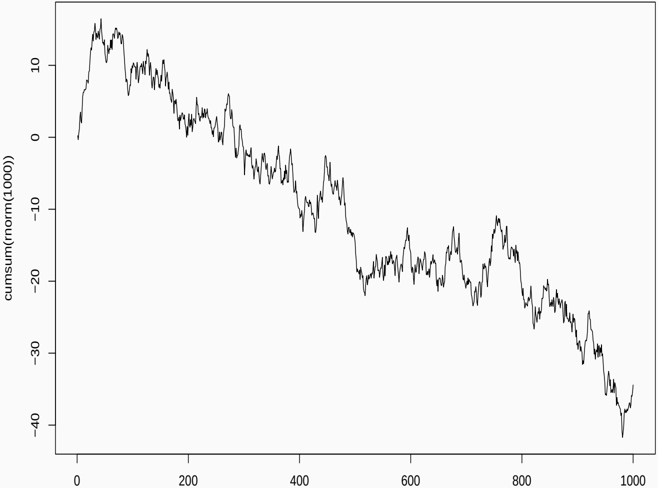
Random walk (4)

```
set.seed(3)
plot(cumsum(rnorm(1000)), type="l")
```



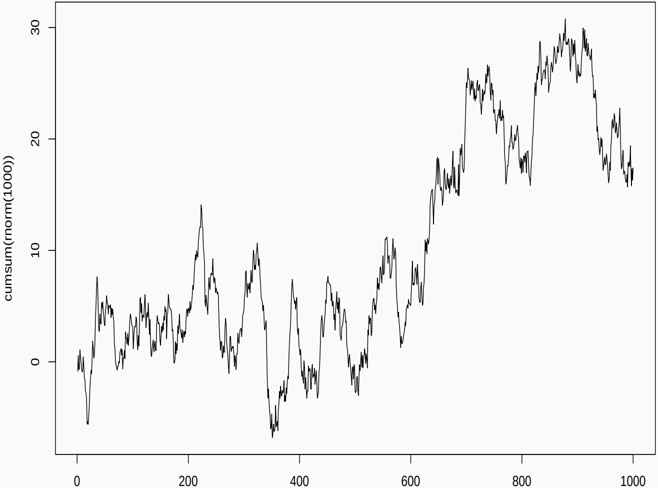
Random walk (5)

```
set.seed(4)
plot(cumsum(rnorm(1000)), type="l")
```



Random walk (6)

```
set.seed(5)
plot(cumsum(rnorm(1000)), type="l")
```



Naive and mean forecast

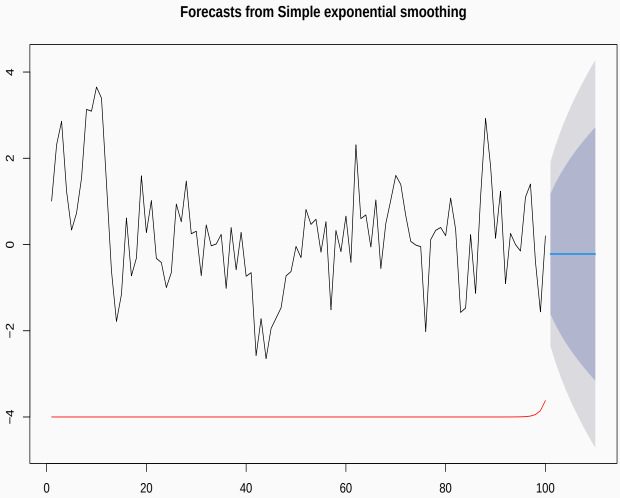
- Naive forecast: We use the last known observation as our forecast
  - Effectively weighting the last observation with 1, all the others with 0
- Mean forecast: We calculate the mean over all observations
  - All observations are weighted equally
- Naive and mean forecast are two extreme cases.

A central assumption in forecasting is often that the more recent past is more important than the more distant past.

Idea: What about weighting the observations with an exponential decay?

$\rightarrow$ Exponential smoothing

Simple exponential smoothing (SES)



red line, shifted from zero for better visualisation: exponentially decaying weights, with decay rate $(1 - \alpha) = 0.38$.

Point forecasts are probabilistic!

- When we forecast a future value y_{t+h}, we are really forecasting its **distribution**:

$$p(y_{t+h} \mid \text{information at time } t).$$

- A **point forecast** $\hat{y}_{t+h}$ is a **single-number summary** of this distribution.
- It is **not necessarily** the "most likely future value," but the value that is **optimal under a chosen loss function**.
- So a point forecast is a **statistic** of the forecast distribution, and therefore has **statistical properties** (bias, variance, etc.).

Point forecasts as optimal decisions

Let $L(a, y)$ be the **loss** from predicting a when the outcome is y.

- A point forecast solves:

$$\hat{y}_{t+h} = \arg\min_a \mathbb{E}[L(a, Y_{t+h})] .$$

Different loss functions $\rightarrow$ different optimal summaries:

- Squared error: $L(a, y) = (a - y)^2 \rightarrow$ **mean**
- Absolute error: $L(a, y) = |a - y| \rightarrow$ **median**
- 0–1 loss: $L(a, y) = \mathbf{1}\{a \neq y\} \rightarrow$ **mode**

Only in the 0–1 loss case is the point forecast the **most likely value**.

Squared error loss (L2 loss)

$$L(a, y) = (a - y)^2$$

$$\hat{y}_{t+h} = \mathbb{E}[Y_{t+h}]$$

- Most common in practice (e.g. least squares, many time series models).
- Forecast = **expected value** of the predictive distribution.

Absolute error loss (L1 loss)

$$L(a, y) = |a - y|$$

$$\hat{y}_{t+h} = \text{median of } Y_{t+h}$$

- More robust to outliers.
- Forecast = **median** of the predictive distribution.

0–1 loss (L0 loss)

$$L(a, y) = \begin{cases} 0 & \text{if } a = y, \\ 1 & \text{otherwise} \end{cases}$$

- Forecast = **mode** (most probable value).

The “most likely future value”: the mode

- The phrase **“most likely future value”** corresponds to the **mode** of the forecast distribution:

$$\text{mode}(Y_{t+h}) = \arg\max_{y} \; p(y \mid \text{information at time } t).$$

- The **mode** is the forecast distribution’s *peak*:
  - the value with the highest probability mass (discrete case), or
  - the highest density (continuous case).

But typical point forecasts are *not* modes

- Most classical forecasting methods (ARIMA, ETS, regression) produce **mean forecasts**, not mode forecasts.
- These coincide only under *symmetric, unimodal* predictive distributions (e.g. Gaussian), where:

$$\text{mean} = \text{median} = \text{mode}.$$

$\rightarrow$ Always think “what loss function is implicit here?” when interpreting or evaluating point forecasts.

Example: when mode, median, and mean differ

Forecast distribution for tomorrow's rainfall:

- 80% chance: 0 mm
- 15% chance: 5 mm
- 5% chance: 40 mm

Then:

- **Mode:** 0 mm — *most likely outcome*
- **Median:** 0 mm — 50% quantile
- **Mean:** $\mathbb{E}[Y] = 0 \cdot 0.8 + 5 \cdot 0.15 + 40 \cdot 0.05 = 3.25$ mm

$\rightarrow$ A model using squared-error loss would forecast **3.25 mm**, even though **0 mm is much more likely**.

$\rightarrow$ 3.25 mm is **not** a "likely" value in the everyday sense.

$\rightarrow$ It is the **mean** of the forecast distribution, chosen to minimize expected squared error.

Strict stationarity

Definition:

A time series $\{y_t\}$ is **strictly stationary** if for any $k \geq 1$, any time points $t_1, \dots, t_k$, and any shift h:

$$(y_{t_1}, y_{t_2}, \dots, y_{t_k}) \stackrel{d}{=} (y_{t_1+h}, y_{t_2+h}, \dots, y_{t_k+h})$$

Key idea:

- The *entire joint distribution* is unchanged by time shifts.
- All moments (mean, variance, skewness, etc.) are time-invariant.

Weak (covariance) stationarity

A series $\{y_t\}$ is **weakly stationary** if:

1. **Constant mean:**

$$\mathbb{E}[y_t] = \mu$$

2. **Constant variance:**

$$\mathrm{Var}(y_t) = \sigma^2$$

3. **Autocovariance depends only on the lag:**

$$\gamma(k) = \mathrm{Cov}(y_t, y_{t+k}) \quad \text{is independent of } t$$

A random walk is non-stationary

- Consider the random walk:

$$y_t = y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{i.i.d. } (0, \sigma^2)$$

- Rewrite:

$$y_t = y_0 + \sum_{i=1}^{t} \varepsilon_i$$

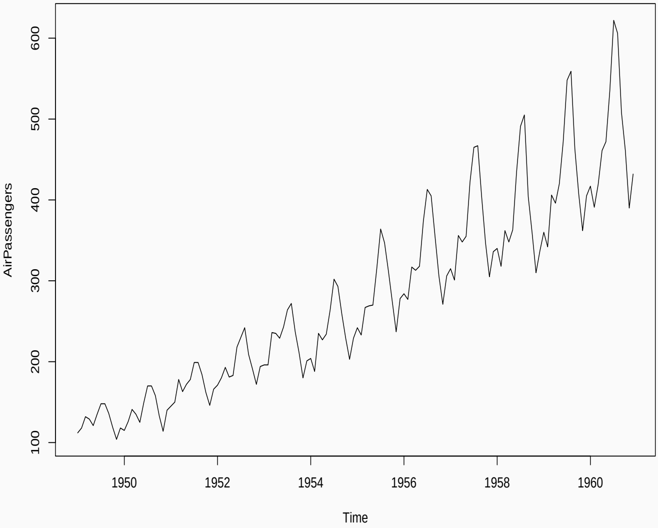
- Variance grows with time: $\mathrm{Var}(y_t) = t\sigma^2$

$\rightarrow$ not constant, violates weak stationarity

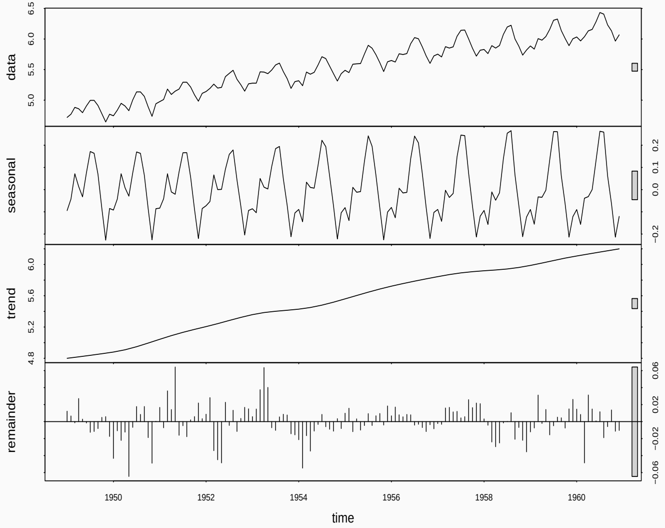
- Shocks accumulate permanently: No mean reversion $\rightarrow$ distribution spreads out without bound.

$\rightarrow$ a random walk is **neither strictly nor weakly stationary**.

Trend and seasonality in a series



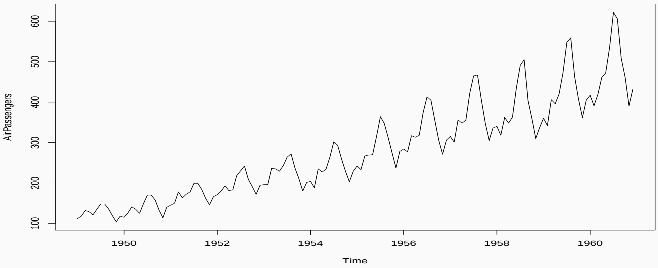
Time series decompositions: STL

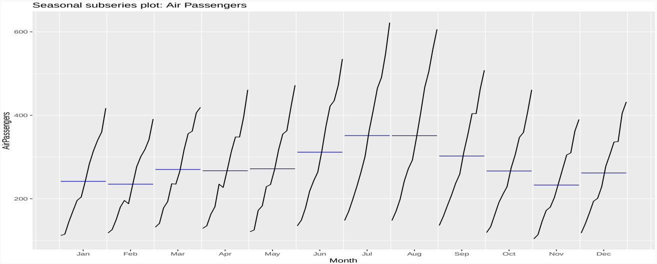


Time series decompositions: STL (2)

- iterative smoothing procedure
- extract the trend
- build series for each component, smooth those
- put series together, smooth
- remove seasonality

Time series decompositions: STL (3)





Time series decompositions

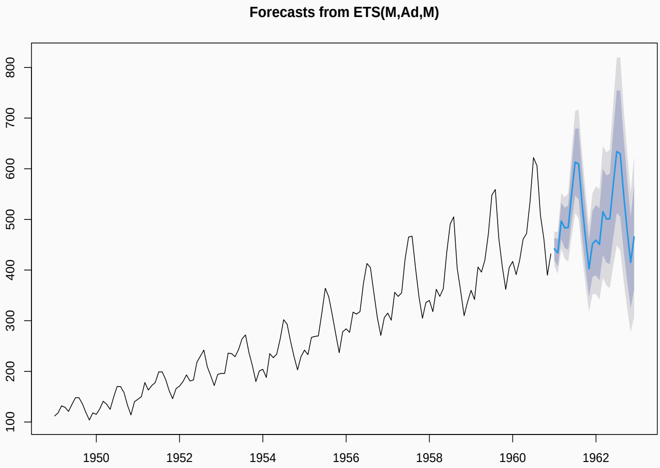
- when you hear in the news about "seasonally adjusted" unemployment data
- there are methods that do decompositions, using information from both the past and the future to calculate the decompositions
- STL, MSTL, STR, X11ARIMA, TRAMO-SEATS
- many forecasting methods decompose the series as well, usually into trend, seasonalities, reminder
- goal: extract components that are easier to predict (i.e., smooth)
- forecasting methods can only use information from one side (the past) for decomposition
- beware of data leakage: never smooth (or extract components from) a whole time series before partitioning into training and test

Exponential smoothing

- Exponential smoothing applied to components: level, seasonality, and trend
- Holt (1957), and his student Winters (1960); (see Goodwin, 2010, for an overview)
- Was used for a long time as a relatively ad-hoc method without a theoretical underpinning
- Hyndman et al. (2002) gave it a solid statistical foundation in state-space models
- ETS stands for both ExponenTial Smoothing and Error, Trend, and Seasonality

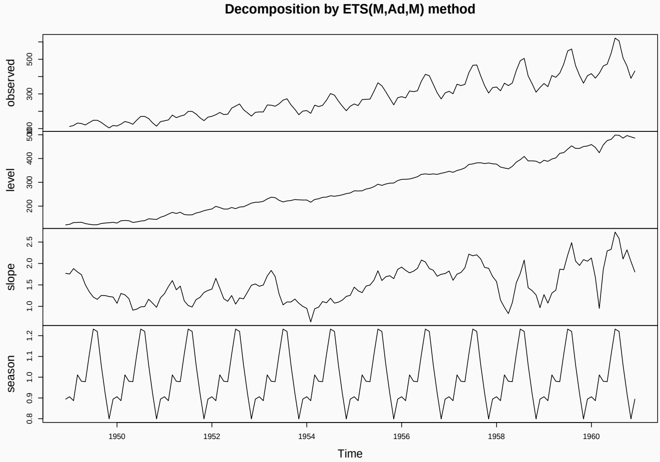
ETS

```
plot(forecast(ets(AirPassengers)))
```

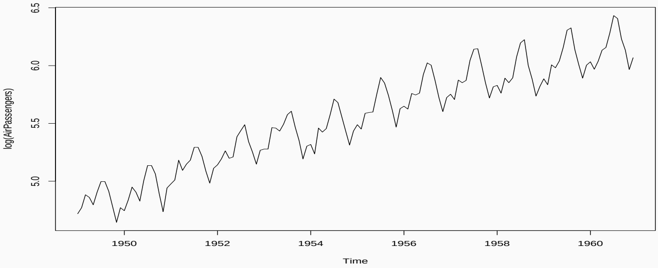


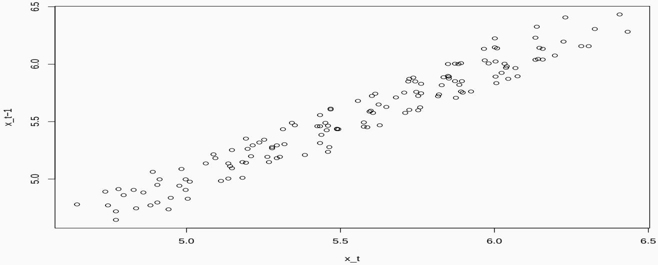
ETS (2)

```
plot(ets(AirPassengers))
```

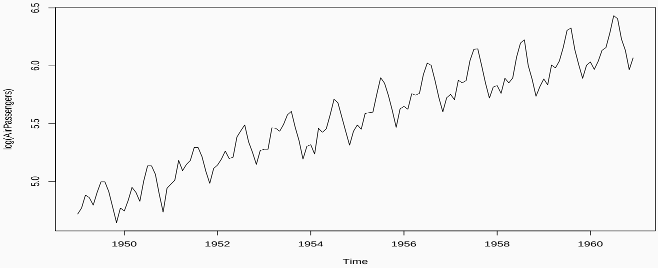


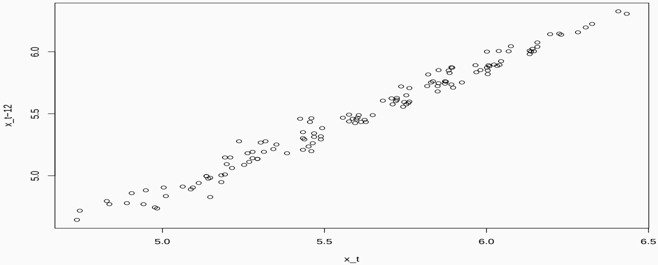
Linear modelling with lags





Linear modelling with lags (2)

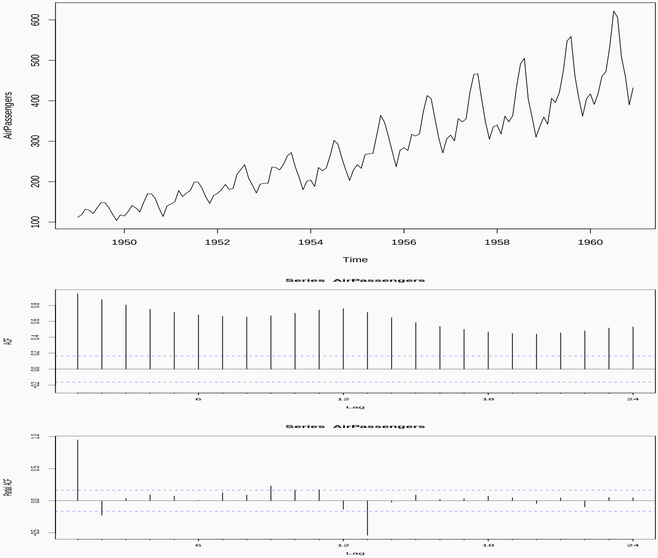




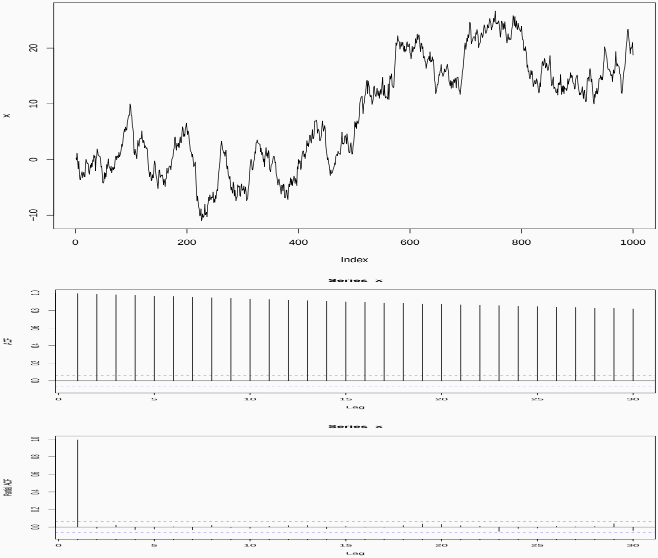
Autocorrelation

- correlation between lags (e.g., x_t and x_t-1)
- measures the extent of a linear relationship
- usually calculated between x_t and all lags up to a maximum number of lags
- trend: large autocorrelation in the first lags that slowly decreases
- seasonality: large autocorrelation in seasonal lags
- both trend and seasonality: combination of the two effects

(Partial) Autocorrelation function



Acf and Pacf for random walks



Box and Jenkins (1970): Linear modelling...

Autoregressive moving average model (ARMA):

$$\hat{x}_{t+1} = c + \phi_1 x_t + \cdots + \phi_p x_{t-p+1} + \theta_1 \epsilon_t + \cdots + \theta_q \epsilon_{t-q+1}$$

- Model is linear in the lags and linear in the errors
- When fitting the model, where do the errors come from?
- Need to estimate some initial conditions and step through the whole series (in ETS as well)
- No closed-form solution
- Need a non-linear fitting procedure. By default in R: BFGS
- fitting can be slow in long time series

ARIMA (cont'd)

- Any (stationary) AR(1) model has an equivalent MA(∞) model, and any (invertible) MA(1) model has an equivalent AR(∞) model (Hyndman and Athanasopoulos, 2018)
- Thus, we can approximate the MA part of the ARMA model with a higher order AR part
- In practice, this "higher order" often is not very high to get satisfactory results (in Econometrics, 5 is often a "high order" already)

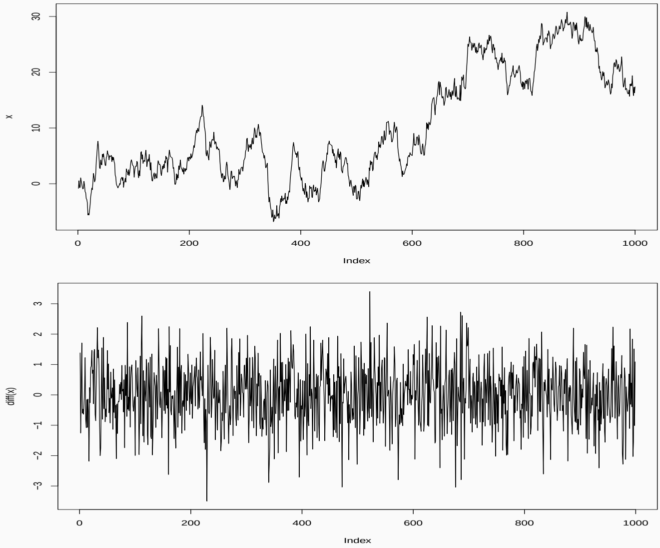
–> Can be seen as a re-parametrisation, to get fewer parameters and a smaller input window, at the cost of more complex model fitting

ARIMA

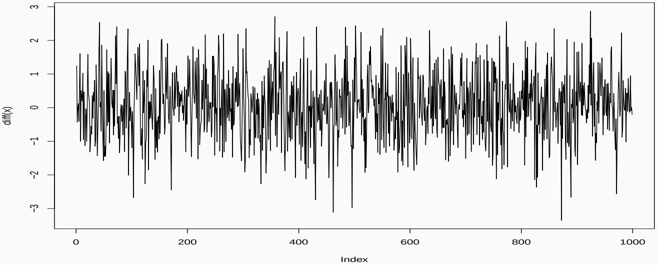
Integrated:

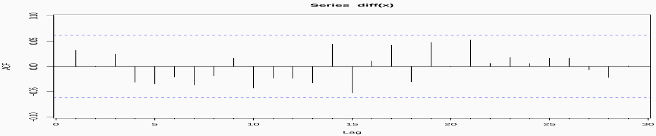
- Do we want to model the values directly or the change to the last value?
- addresses non-stationarity
- Preprocessing step: Do differencing of the series before we do the ARMA
- Pro: Hopefully makes the series stationary
- Contra: We loose some information about the scale in the preprocessing

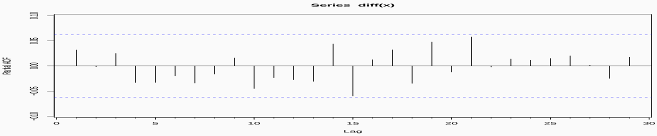
Integration (cont'd)



Acf and Pacf for random walk with differencing

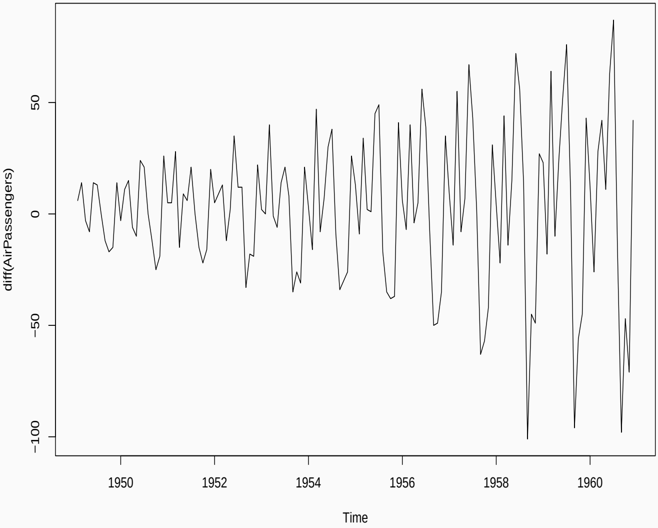






Transformations

```r
plot(diff(AirPassengers))
```



Logarithm or Box-Cox transform

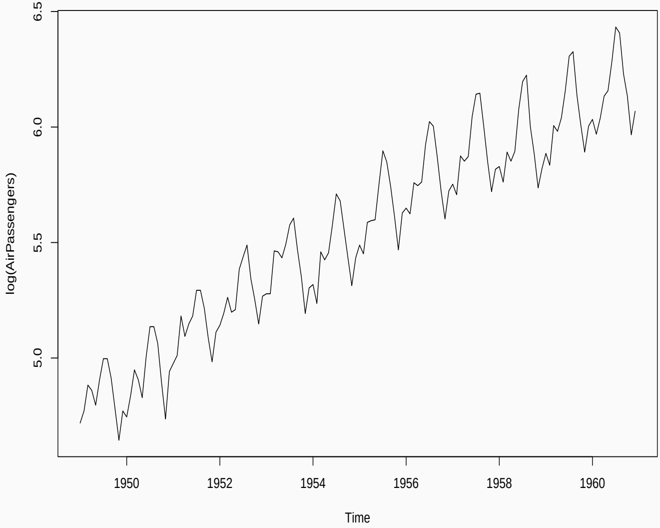
- makes exponential trends linear
- makes multiplicative seasonalities additive (e.g., STL is an additive decomposition)
- also stabilises the variance
- log is a very strong transform
- Box-Cox Transformation

$$w_t = \begin{cases} log(y_t) & \text{if } \lambda = 0, \\ (y_t^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \end{cases}$$

- Choice of optimal value for λ is difficult

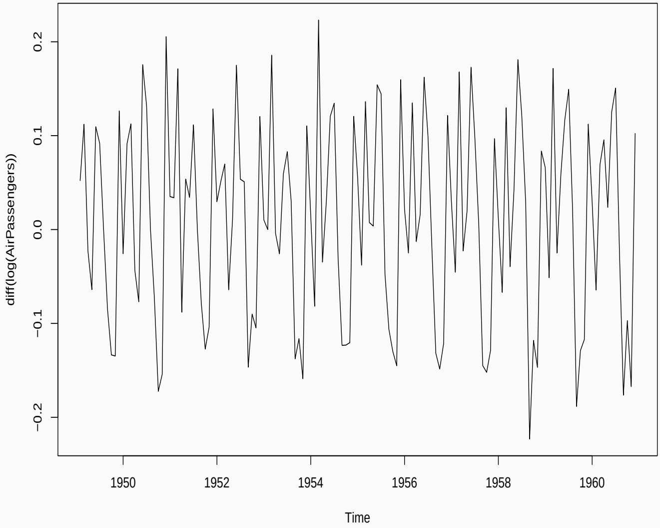
Transformations (2)

```
plot(log(AirPassengers))
```



Transformations (3)

```
plot(diff(log(AirPassengers)))
```



How to use Pacf and Acf for model selection

- Pacf is used to select model order of the AR model
- Acf is used to select model order of MA model

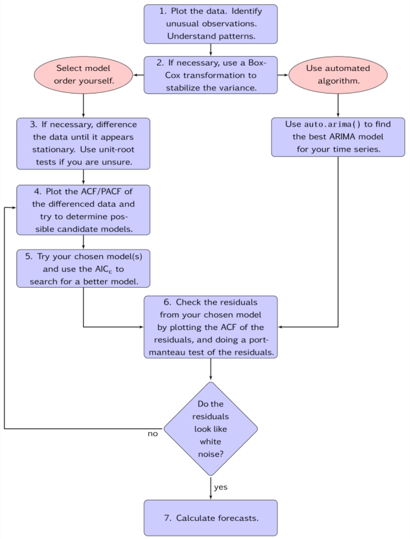
Other ways for model selection: AIC, BIC, AICc

- Akaike's Information Criterion (AIC):

$$AIC = 2k - 2ln(\hat{L})$$

- Basically log of in-sample MSE penalised by the amount of parameters in the model
- Method common in statistics for model selection / hyper-parameter selection
- No need for an additional validation set or cross-validation
- AICc: Bias-corrected version for small amounts of data
- Theoretical results that AICc is asymptotically equivalent than one-step-ahead out-of-sample
- Cannot be used across different models and model classes where the parameters are not comparable
- Therefore, not usually used in Machine Learning

ARIMA workflow



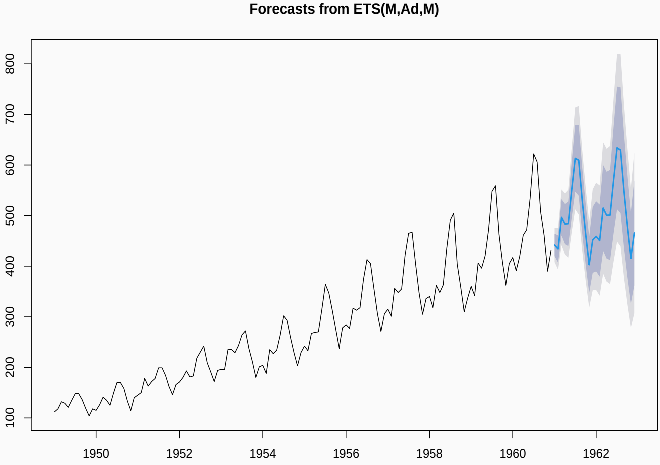
Source: https://otexts.com/fpp2/arima-r.html

ets() and auto.arima()

- standard implementations, from the “forecast” package in R (Hyndman and Khandakar, 2008)
- Wrappers for automatic model selection of ETS and ARIMA
- `ets()`: 15 models with different Error, Trend, and Seasonality
- `auto.arima()`: by default up to order 5 for p and q, and up to 2 differences
- choose the best one with AICc
- widely used standard benchmarks
- often very competitive

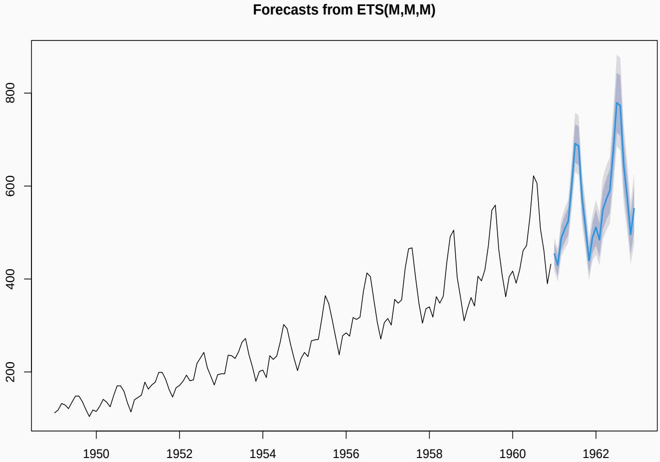
ETS

```
plot(forecast(ets(AirPassengers)))
```



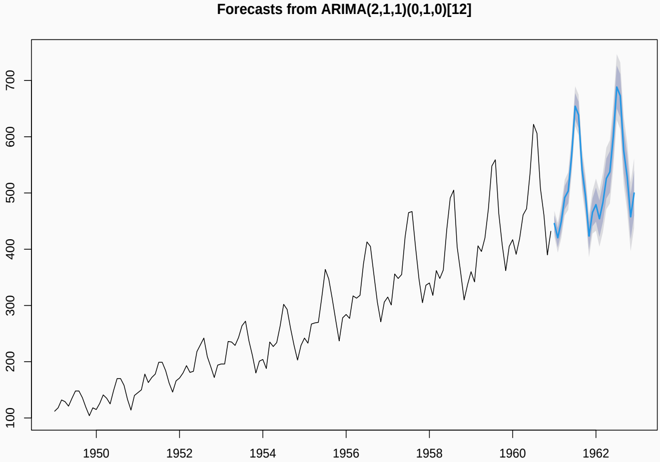
ETS (2)

```
plot(forecast(ets(AirPassengers,
                  model="MMM", damped=FALSE)))
```



auto.arima

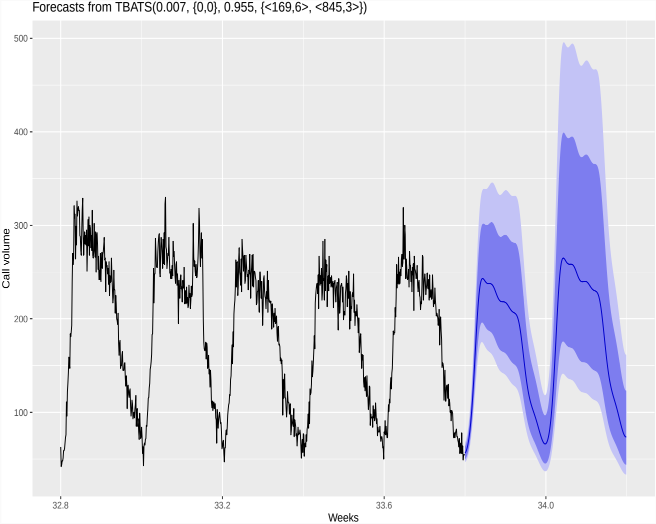
```r
plot(forecast(auto.arima(AirPassengers)))
```



TBATS

- Trigonometric seasonality, Box-Cox transformation, ARIMA errors, Trend, Seasonal components
- combination of Fourier terms with exponential smoothing model
- also performs a decomposition of the time series, same as ETS
- fully automatic method to forecast series with multiple seasonalities, non-integer seasonalities
- often slow to fit, especially for long series (and series with multiple seasonalities are often long)

TBATS (2)


Forecasts from TBATS(0.007, {0,0}, 0.955, {<169,6>, <845,3>})
500
400
300
200
100
Call volume
32.8
33.2
33.6
34.0
Weeks

Prophet

- Method developed initially by Facebook
- Native implementation in Python, at a time where there were not many forecasting methods in Python
- Easy to use and flexible
- Also based on decomposing the time series
- Trend can have change points
- It has no autoregressive components, it is a regression of the series at time t onto components at time t:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t$$

- g is the trend ("growth"), s the seasonality, h the holiday effects
- Is intended for interactive forecasting, and flexible model adjusting
- Not too special in terms of accuracy, especially when used fully automatically which is not its intended use case

A brief history of forecasting competitions

The M competitions

I'm following here mostly Hyndman (2020)

Makridakis and Hibon (1979). Journal of the Royal Statistical Society: Series A.

- They benchmarked a bunch of time series methods against each other on 111 time series.
- Simple methods (such as seasonal naive or SES) worked often better than ARIMA...which was considered a complex method

The M competitions (cont'd)

- Discussion published alongside the paper.
  "I find it hard to believe that Box-Jenkins, if properly applied, can actually be worse than so many of the simple methods." —Chris Chatfield

- Strong belief at that time that there is a true underlying data model that can be discovered
- And that this underlying model will necessarily give the best forecasts
- So, the discussants questioned the ability of Makridakis and Hibon to use methods like ARIMA properly.
- And they didn't agree with the finding that "forecast combination" works well (we would call this nowadays "ensembling").

M1 competition (1982)

- Results published in Makridakis et al. (1982)
- To answer to the criticism, Makridakis organised a competition where contestants could submit forecasts
- 1001 time series
- results largely confirmed the earlier results
- best method was a seasonal naive combined with SES
- forecast combination worked well
- after this, forecasters focused on OOS accuracy rather than model properties
- Forecasting emerged as an independent field from Time Series Analysis

M3 competition (1998)

- Results published in Makridakis and Hibon (2000)
- Had as a central conclusion (again) that complex methods do not necessarily outperform simple methods
- Again, ETS and ARIMA were the “complex” models, and simple models were models like a random walk with drift
- The only neural network that participated was quite bad
- Won by the “theta” method, which was later shown to be an average of a linear regression and simple exponential smoothing with drift
- **The** benchmark dataset in forecasting for almost 20 years
- A lot of focus (too much?) on this dataset

Controversy of ML vs Statistical methods

Controversy in the forecasting field, whether Machine Learning or Statistical methods work better for forecasting

- Forecasters "knew" that simple methods work best
  - ... because they had won the M3, the NN3, the NN5
- Machine Learners "knew" that Neural Networks work best
  - ... because, hey, it's a NN, it's a universal approximator
- Thousands of papers in Neural Network journals and others
- "A Novel method X for stock market forecasting..."
- Cherry-picked datasets, no proper benchmarking

Controversy of ML vs Statistical methods (cont'd)

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

- Makridakis et al. (2018b) benchmarked ML methods against statistical benchmarks
- not surprisingly, the ML methods lost
- The paper is published in PLOS ONE because it got rejected at 2-3 Neural Network outlets beforehand.

Controversy of ML vs Statistical methods (cont'd)

- Reasons for rejection were that: there are many ML methods that have "proven to overcome the results provided," though the editors and reviewers didn't name any in particular
- Makridakis was upset over this and did what he seemingly does best when he is upset...organise a competition: the M4
- so that Machine Learners could submit forecasts and show that their methods work well

Controversy of ML vs Statistical methods (cont'd)

- The history explains why the forecasting field got so hung up on this controversy
- Still somewhat surprising as Forecasting emerged from Statistics just in the same way Machine Learning emerged from Statistics

→ Focus on OOS accuracy, not model properties

→ Some things were discovered in parallel in both fields:

- forecast combination (Bates and Granger, 1969) vs ensembling (in ML in the 1990s)
- simple methods vs regularisation

→ Forecasting and Machine Learning should learn from each other!

M4 competition (2018)

- 100k series, across many different domains
- hourly, daily, weekly, monthly, quarterly, yearly series
- prediction intervals

Results

- published in Makridakis et al. (2018a)
- most forecasters expected that a combination approach of statistical methods would win
- however, such methods got 2nd and 3rd
- winner was RNN-ES, a hybrid between RNN and ETS, by Smyl (2020)
- a big surprise to many people that a ML model could win
- ... but not so much for me

Computational Intelligence in Forecasting (CIF) 2016 competition (Štěpnička and Burda, 2016)

- … I got beaten in a competition by Slawek Smyl 2 years earlier
- 72 monthly series (lengths 22-108)
- I participated with plain ETS and BaggedETS
- the BaggedETS won several sub-categories and was also winning on median sMAPE
- On the competition metric mean sMAPE, BaggedETS got 9th, plain ETS got 3rd
- 1st and 2nd place were LSTMs, from Slawek Smyl.
- Globally trained across series

Thank You

`https://www.cbergmeir.com`

christoph.bergmeir@monash.edu

References i

J. M. Bates and C. W. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969.

G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

P. Goodwin. The Holt-Winters approach to exponential smoothing: 50 years old and going strong. *Foresight: The International Journal of Applied Forecasting*, 19:30–33, 2010.

References ii

R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

R. Hyndman, A. Koehler, R. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

R. J. Hyndman. A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14, 2020.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 2nd edition*. OTexts, otexts.com/fpp2, 2018.

References iii

S. Makridakis and M. Hibon. Accuracy of forecasting: An empirical investigation. *Journal of the Royal Statistical Society: Series A (General)*, 142(2):97–125, 1979.

S. Makridakis and M. Hibon. The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153, 1982.

References iv

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018a.

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

S. Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

References v

M. Štěpnička and M. Burda. Computational intelligence in forecasting (CIF) 2016 time series forecasting competition. In *IEEE WCCI 2016, JCNN-13 Advances in Computational Intelligence for Applied Time Series Forecasting (ACIATSF)*, 2016.