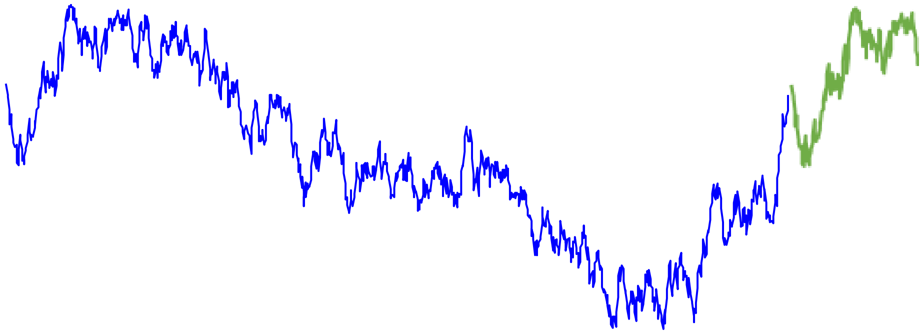
Forecasting

Riccardo Guidotti

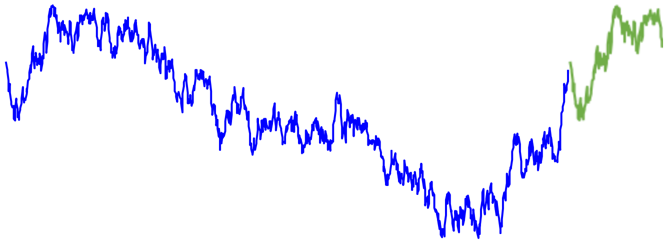
Syllabus

- Simple Forecasting
- Exponential Smoothing
- ARIMA
- Forecasting via Reduced Regression
- Deep-learning Models
- Probabilistic Forecasting



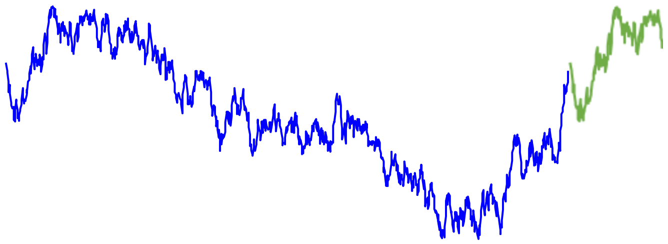
Time Series Forecasting

- Main difference between forecasting and regression: forecasting is about predicting a future state/value, rather than a current one.
- Applications:
  - Temperature, Humidity, CO2 Emissions
  - Epidemics
  - Pricing, Sales Volumes, Stocks
  - Forewarning of Natural Disasters (flooding, hurricane, snowstorm),
  - Electricity Consumption/Demands
- Techniques:
  - Statistical Methods,
  - Machine Learning Classifiers
  - Deep Neural Networks



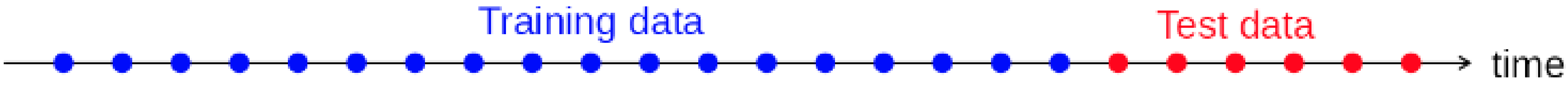
Forecasting vs Regression

- Forecasting is **time dependent:** the basic assumption of a linear regression model that the observations are independent does not hold.
- Along with an increasing or decreasing **trend**, most TS have some form of **seasonality** trends, i.e., variations specific to a particular time frame.



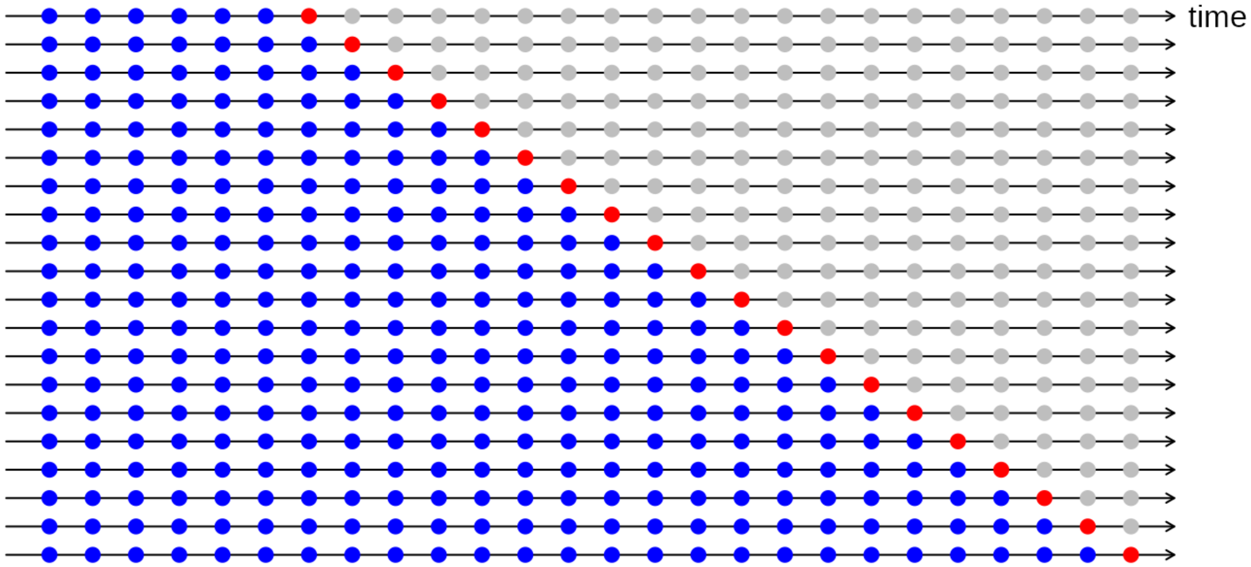
Evaluating Forecast Accuracy

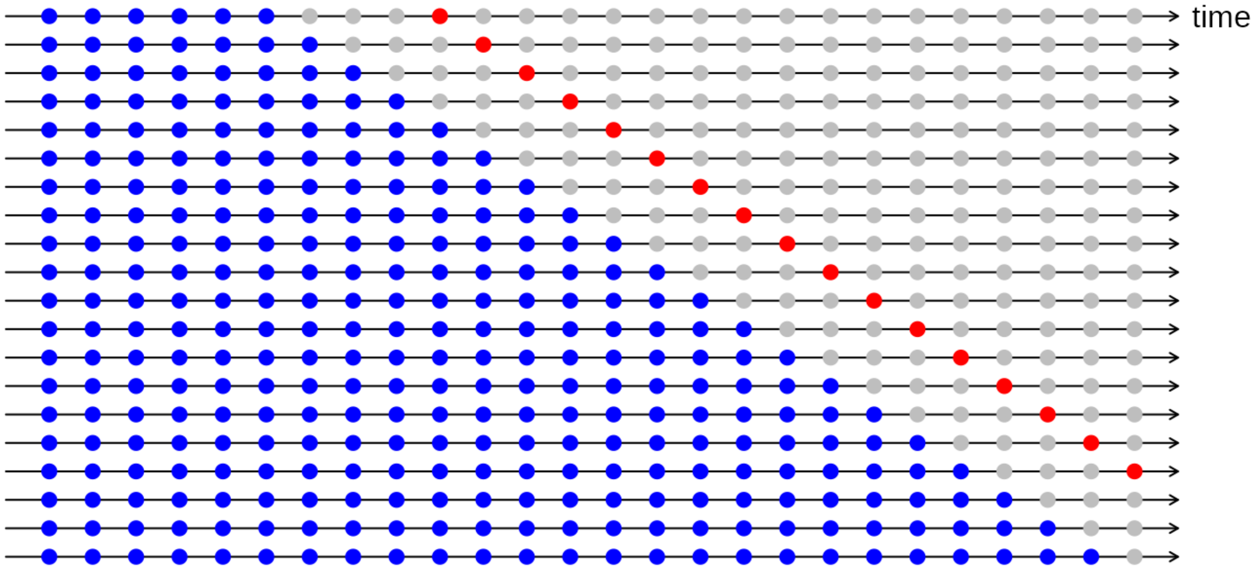
- Separate the data into two portions: training and test data
  - the training data is used to estimate any parameters of a forecasting method
  - the test data (typically about 20%) is used to evaluate its accuracy.
- the test data should provide a reliable indication of how well the model is likely to forecast on new data.
- The size of the test set should depend on how long the sample is and how far ahead you want to forecast. It should ideally be at least as large as the maximum forecast horizon required.
- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.



Time Series Cross Validation

- The test set consists into a single observation.
- The corresponding training set consists only of observations that occurred prior to the observation that forms the test set.
- The forecast accuracy is computed by averaging over the test sets.





Evaluating Forecast Accuracy

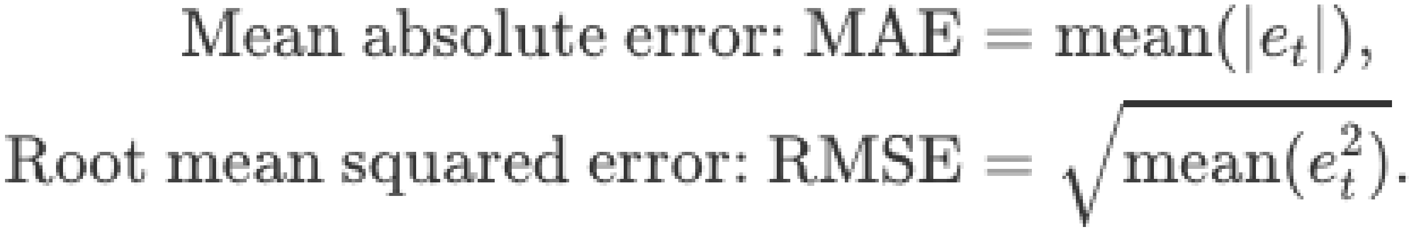
- A forecast "error" is the difference between an observed value and its forecast. An "error" is the unpredictable part.

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$$

- Forecast errors are different from residuals:
  - Residuals are calculated on the training set while forecast errors are calculated on the test set.
  - Residuals are based on one-step forecasts while forecast errors can involve multi-step forecasts.
- We can measure forecast accuracy by summarizing the forecast errors in different ways.

Scale-Dependent Errors

- Cannot be used to make comparisons between TS that involve different units.
- The two most commonly used scale-dependent measures are based on the absolute errors or squared errors:



$$\text{Mean absolute error: MAE} = \text{mean}(|e_t|),$$

$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_t^2)}.$$

Percentage Errors

- Percentage errors are unit-free, and so are frequently used to compare forecast performances between data sets.
- The percentage error is given by

$$p_t = 100e_t/y_t$$

- The most commonly used measure is:

$$\text{Mean absolute percentage error: MAPE} = \text{mean}(|p_t|)$$

- Total and Median Absolute Percentage Error (TAPE, MedianAPE) are also used.

Evaluation Measures from Regression

- **Coefficient of determination** R^2
  - is the proportion of the variance in the dependent variable that is predictable from the independent variable(s)

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

hat means predicted

$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}\epsilon_i^2$

- **Mean Squared/Absolute Error** MSE/MAE
  - a risk metric corresponding to the expected value of the squared (quadratic)/absolute error or loss

$$\text{MSE}(y,\hat{y}) = \frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2 \quad \text{MAE}(y,\hat{y}) = \frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1}\left|y_i - \hat{y}_i\right|$$

Simple Forecasting Methods

Simple Forecasting Methods

- **Average Method**: the forecasts of all future values are equal to the average (or “mean”) of the historical data.

$$\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T.$$

- **Naïve Method**: the forecasts of all future values are equal to the last value of the historical data.

$$\hat{y}_{T+h|T} = y_T.$$

- **Drift Method**: increase/decrease last value w.r.t. the amount of change over time (*drift*) as the average change in the historical data.

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right)$$

Exponential Smoothing

Simple Exponential Smoothing (SES or ETS)

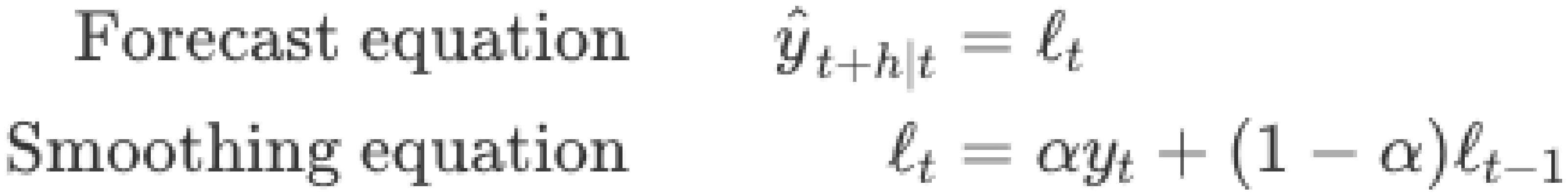
- Is suitable for data with no clear trend or seasonal pattern.
- SES is in between the average and naive method.
- SES attaches larger weights to more recent observations than to observations from the distant past, while smallest weights are associated with the oldest observations
- Forecasts are calculated using weighted averages, where the weights decrease exponentially as observations come from further in the past.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

- $0 \le \alpha \le 1$ is the smoothing parameter

SES – Formalization in Components

- For SES the only component used is the level.
- Component form representations of SES comprise a forecast equation and a smoothing equation for each of the components in the method.



$$\begin{aligned} \text{Forecast equation} \qquad & \hat{y}_{t+h|t} = \ell_t \\ \text{Smoothing equation} \qquad & \ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1} \end{aligned}$$

- where l_t is the level of the TS at time t

Holt’s Linear Trend Method

- Holt extended SES to allow the forecasting of data with a trend.

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \end{aligned}$$

- where l_t is the level of the TS at time t,
- b_t estimates the trend of TS,
- $0 \leq \alpha \leq 1$ is the smoothing parameter for the level and
- $0 \leq \beta^* \leq 1$ is the smoothing parameter for the trend.

Holt-Winters’ Seasonal Method

- Holt (1957) and Winters (1960) extended Holt’s method to capture seasonality.
- m denotes the frequency of the seasonality, i.e., the number of seasons in a reference period, while $0 \leq \gamma \leq 1 - \alpha$ is the smoothing parameter for the seasonality.
- The additive method is preferred when the seasonal variations are constant through the TS
- The multiplicative method is preferred when the seasonal variations are changing proportional to the level of the TS.

Holt-Winters’ Seasonal Method

- Additive

$$\begin{aligned}\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)}\\ \ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})\\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}\\ s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},\end{aligned}$$

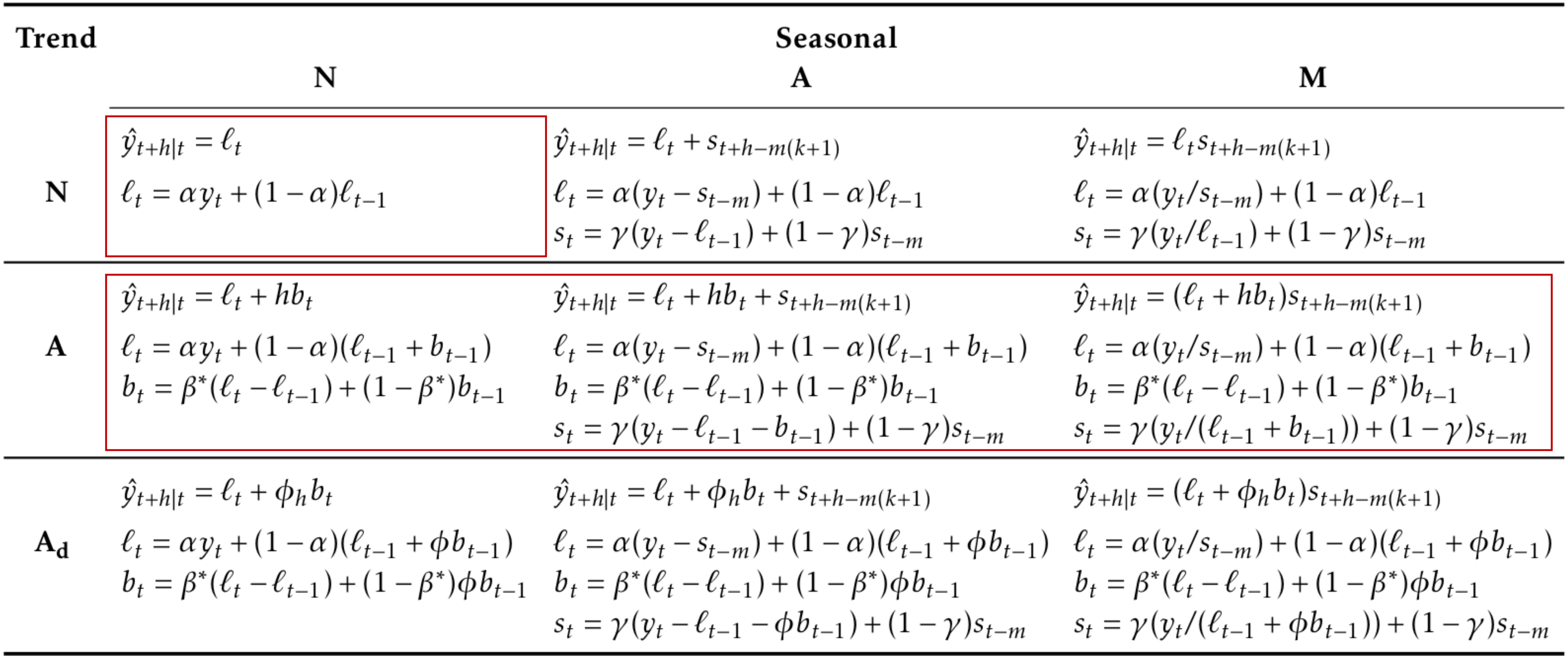
- Multiplicative

$$\begin{aligned}\hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)}\\ \ell_t &= \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})\\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}\\ s_t &= \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}\end{aligned}$$

k is the integer part of (h−1)/m, which ensures that the estimates of the seasonal indices come from the final period of the sample.

More on Exponential Smoothing

- ES methods are not restricted to those we have presented.



Trend		Seasonal	
	N	A	M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$

ARIMA Models

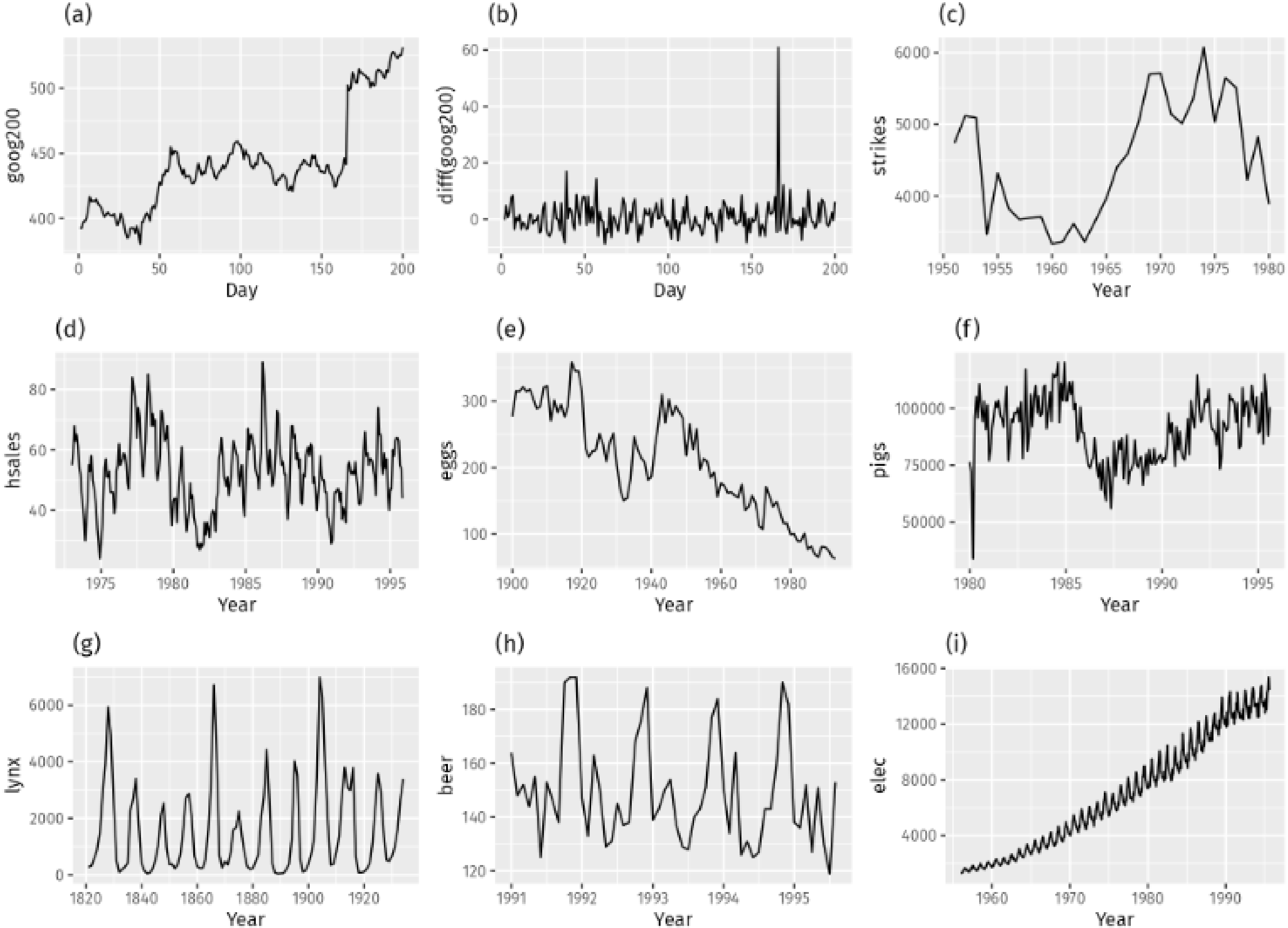
ARIMA Models

- While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data.
- Before we introduce ARIMA models, we recall the concepts of stationarity and the technique of differencing TS.

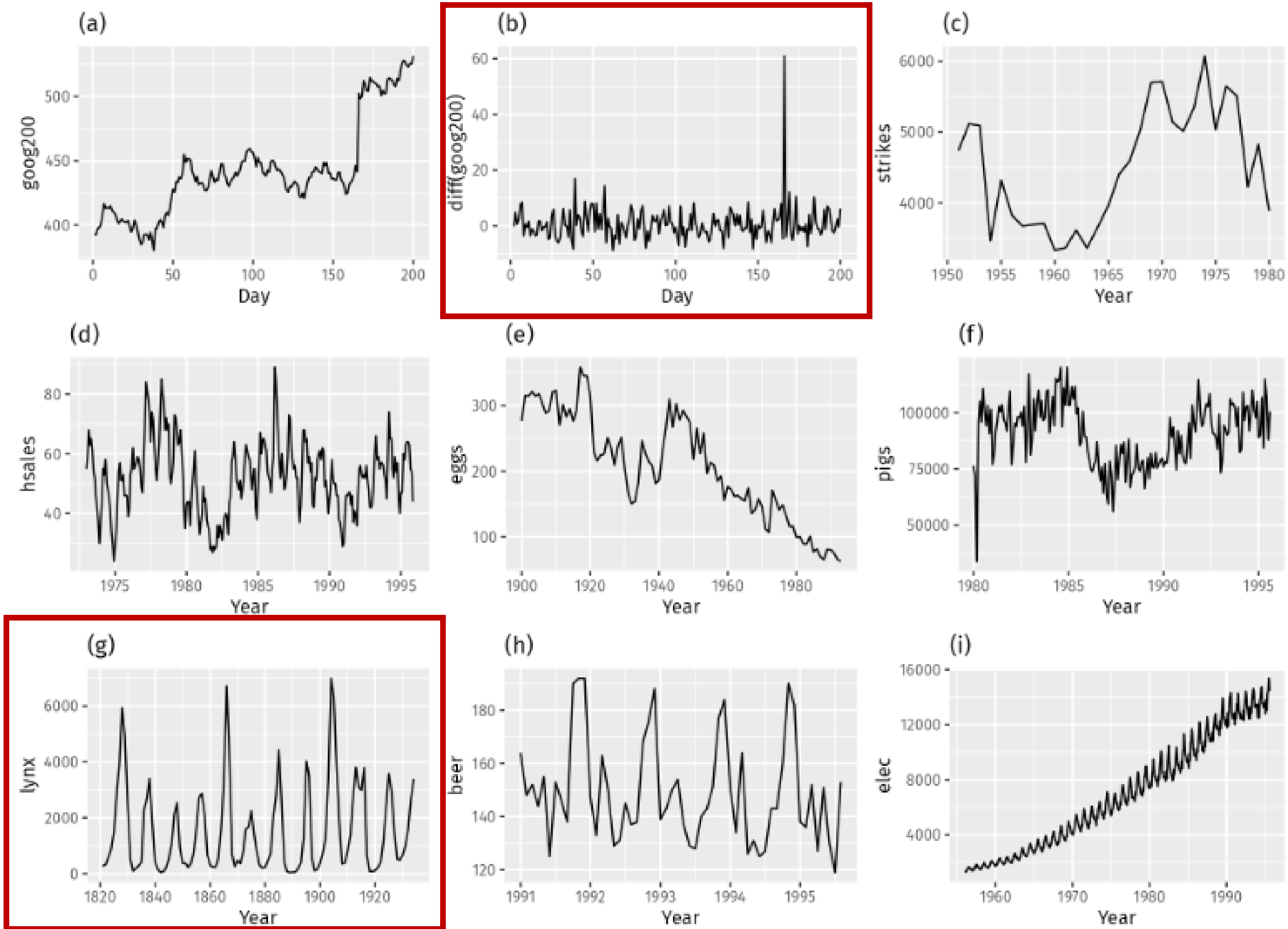
Stationarity

- A stationary TS is one whose properties do not depend on the time at which the series is observed.
- TS with trends, with heteroskedasticity, or seasonality, are not stationary as these aspects affect the value of the TS at different times.
- A TS with cyclic behavior but with no trend or seasonality is stationary.
- A white noise series is stationary: it does not matter when you observe it, it looks much the same at any point in time.
- In general, a stationary time series will have no predictable patterns in the long-term.

Which are Stationary?



Which are Stationary?

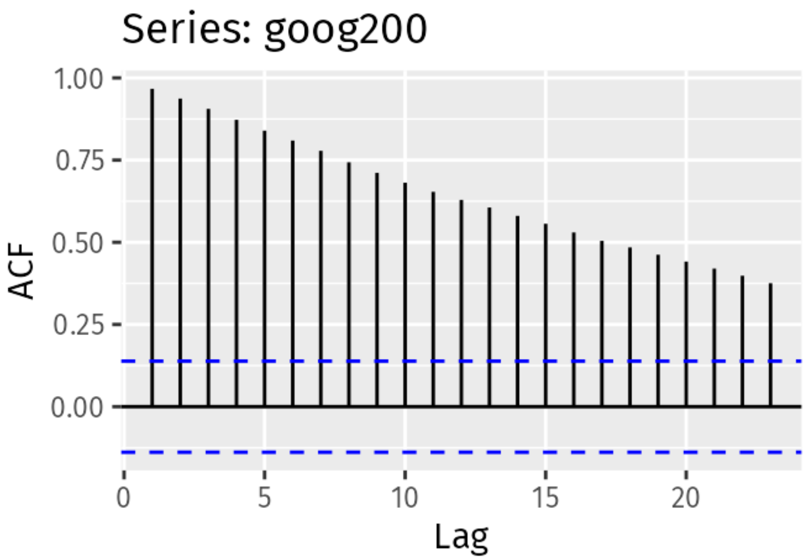


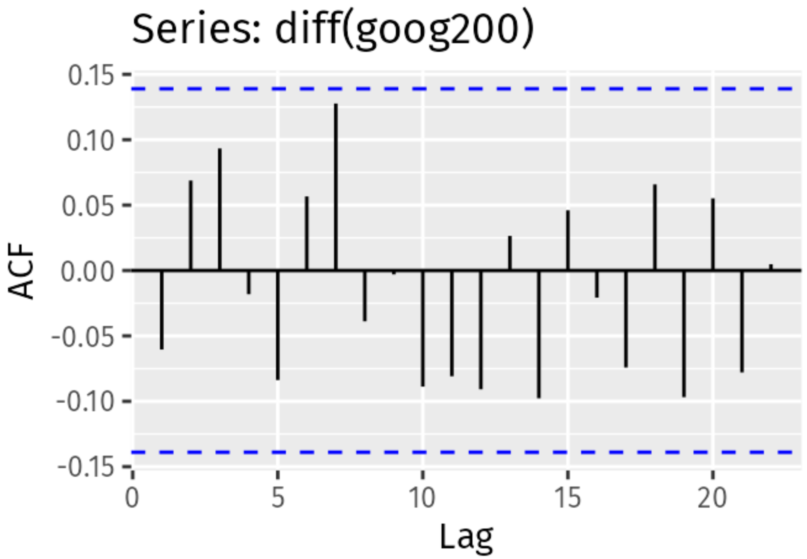
Differencing

- Differencing: compute the differences between consecutive observations.
- Differencing can make a non-stationary TS stationary.
- Differencing can help stabilize the mean of a TS by removing changes in the level, and thus eliminating (or reducing) trend and seasonality.
- In addition, transformations such as logarithms can help to stabilize the variance of a time series.

ACF Plot and Stationariety

- As well as looking at the plot of the TS
- The ACF plot is also useful for identifying non-stationary TS:
- For a stationary TS the ACF will drop to zero relatively quickly
- For non-stationary TS the ACF decreases slowly.
- Also, for non-stationary TS, the value of is often large and positive.


Series: goog200


Series: diff(goog200)

AR - AutoRegressive Models

- In multiple *regression* model, we predict the variable of interest using a linear combination of predictors.
- In an autoregression model, we forecast the variable of interest using a linear combination of past values of the variable.
- The term *autoregression* indicates that it is a regression of the variable against itself.
- An autoregressive model of order *p* can be written as

white noise

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

- This is as an **AR(p) model** of order *p* (p = lag in the past)

AR - AutoRegressive Models

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

For an AR(1) model:

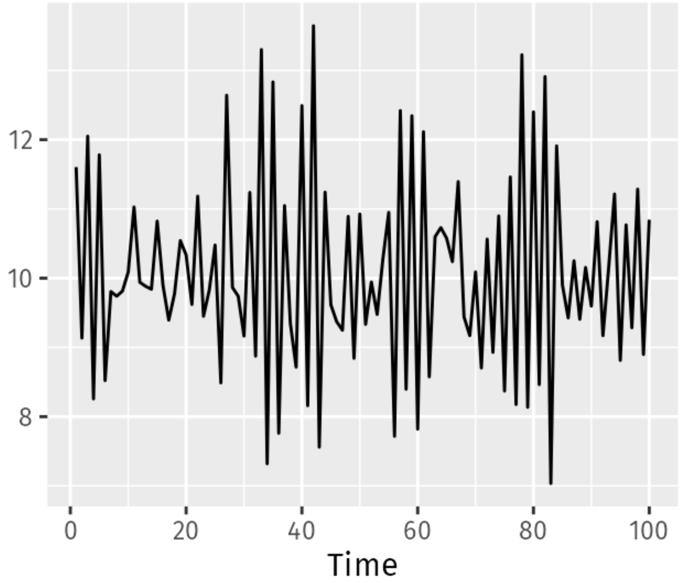
- when $\phi_1 = 0$ is equivalent to white noise
- when $\phi_1 = 1$ and $c = 0$ is equivalent to a random walk
- when $\phi_1 = 1$ and $c \neq 0$ is equivalent to a random walk with drift
- when $\phi_1 < 0$ tends to oscillate around the mean

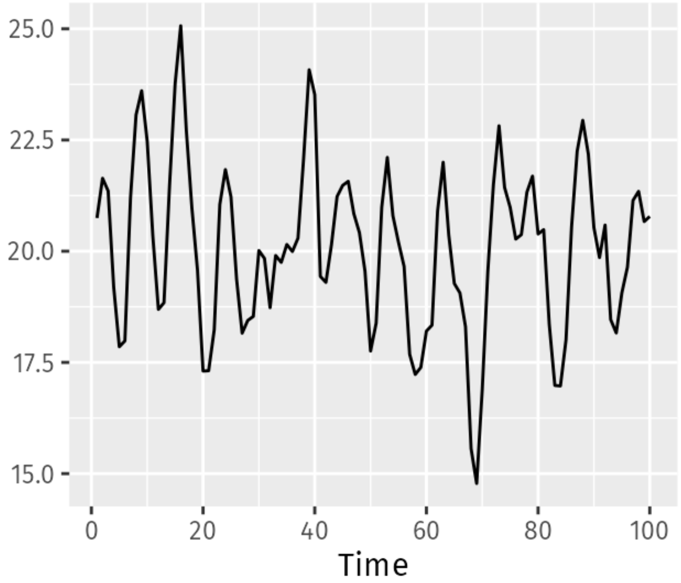
We normally restrict AR models to stationary data, in which case some constraints on the values of the parameters are required.

- For AR(1): $-1 \leq \phi_1 \leq 1$
- For AR(2): $-1 \leq \phi_2 \leq 1,\ \phi_1 + \phi_2 < 1,\ \phi_2 - \phi_1 < 1$
- When *p>2* the restriction are much more complicated.

AR - AutoRegressive Models Example

- Two examples of data from AR models with different parameters.
- Left: AR(1) with y_t = 18 − 0.8 y_{t-1} + ε_t
- Right: AR(2) with y_t = 8 + 1.3 y_{t-1} − 0.7 y_{t-2} + ε_t
- In both cases, ε_t is normally distributed white noise.


AR(1)


AR(2)

MA - Moving Average Models

- Rather than using past values of the forecast variable in a regression, a MA model uses past forecast errors in a regression-like model.

white noise

second order differencing

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}$$

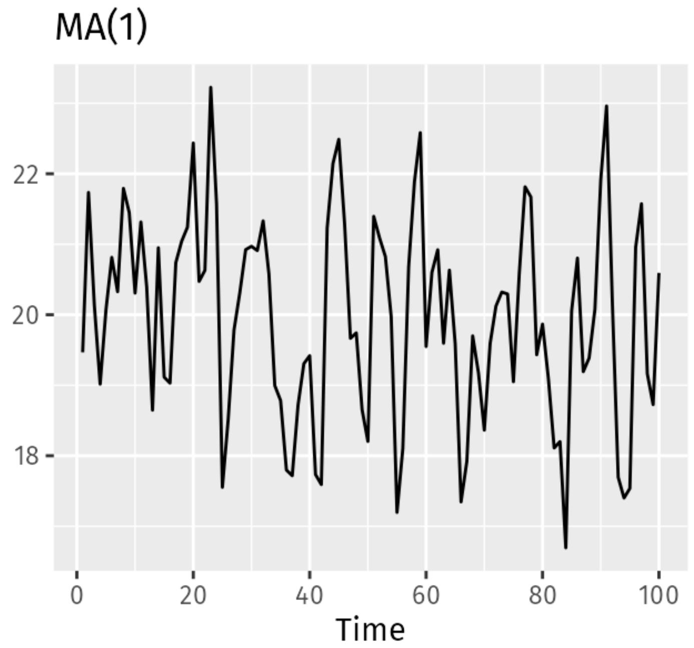
mean of
the series

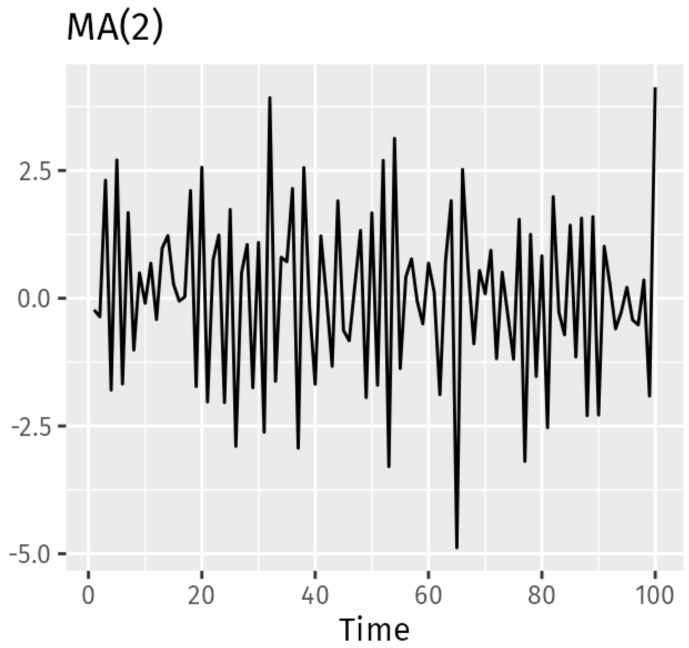
first order differencing

- This is as a **MA(q) model** of order *q* (q = lag in the past).
- It can be interpreted as: each observation is regressed on previous noise
- A weighted sum of previous and current noise is called Moving Average (MA)
- MA models should not be confused with the moving average smoothing.
- It is possible to write any stationary AR(p) as MA(∞)

MA - Moving Average Models Example

- Two examples of data from MA models with different parameters.
- Left: MA(1) with $y_t = 20 - \varepsilon_t + 0.8\ \varepsilon_{t-1}$
- Right: MA(2) with $y_t = 0 + \varepsilon_t + 1\ \varepsilon_{t-1} + 0.8\ \varepsilon_{t-2}$
- In both cases, ε_t is normally distributed white noise.





Moving Average Models

- It is possible to write any stationary AR(p) as MA(∞)
- The reverse result holds if we impose some constraints on the MA parameters.
- Then the MA model is called **invertible.**
- The invertibility constraints for other models are similar to the stationarity constraints.
- For MA(1): $-1 \leq \theta_1 \leq 1$
- For MA(2): $-1 \leq \theta_2 \leq 1, \theta_1 + \theta_2 > -1, \theta_1 - \theta_2 < 1$
- When p>2 the restriction are much more complicated.

ARIMA Models (Non-Seasonal)

- If we combine differencing with an AR model and a MA model, we obtain a non-seasonal ARIMA model.
- ARIMA is an acronym for AutoRegressive Integrated Moving Average ("integration" is the reverse of differencing).

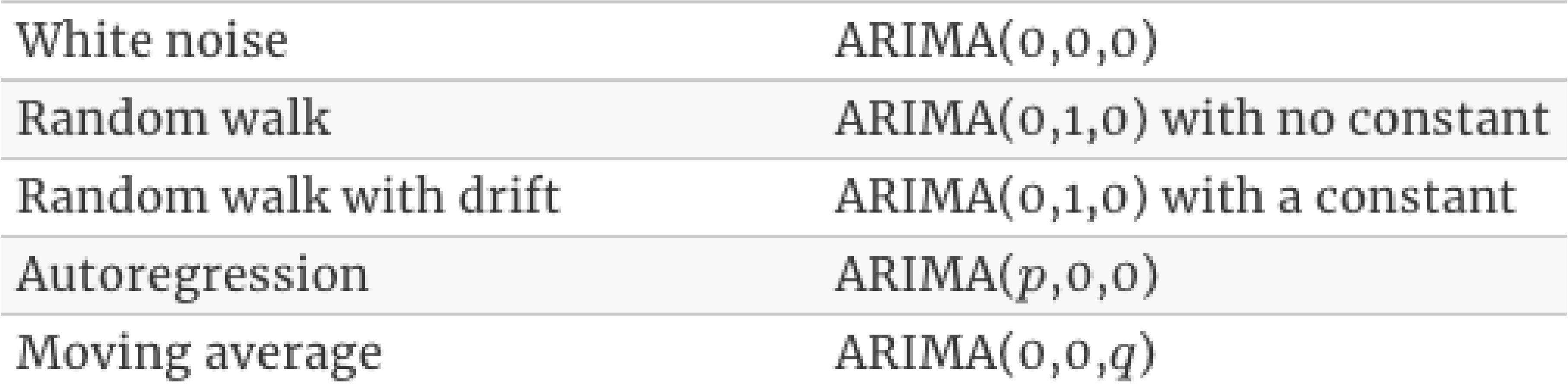
$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

AR(p) MA(q)

- where y'_t is the differenced series.
- We call this model **ARIMA(p,d,q) model**, where *p* is the order of the autoregressive part, *d* is the degree of first differencing involved, *q* is the order of the moving average part

ARIMA Models (Non-Seasonal)

- The same stationarity and invertibility conditions that are used for AR and MA models also apply to an ARIMA model.
- Special cases of ARIMA models

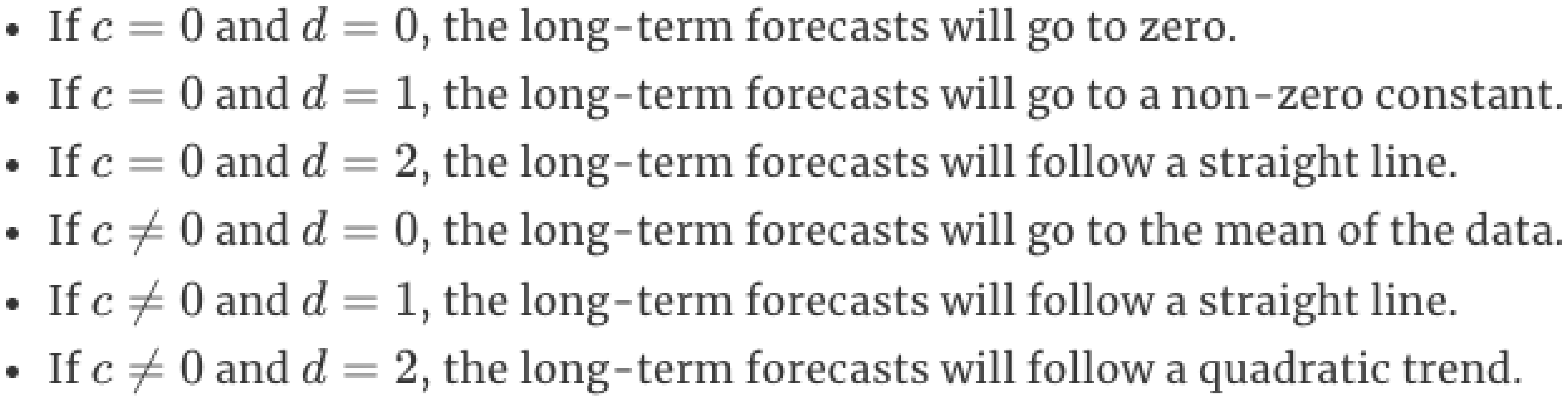


White noise	ARIMA(0,0,0)
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)

- ARIMA(p,0,q) is also called ARMA(p,q)

Understanding ARIMA Models

- The constant *c* has an important effect on the long-term forecasts.



  - If $c = 0$ and $d = 0$, the long-term forecasts will go to zero.
  - If $c = 0$ and $d = 1$, the long-term forecasts will go to a non-zero constant.
  - If $c = 0$ and $d = 2$, the long-term forecasts will follow a straight line.
  - If $c \neq 0$ and $d = 0$, the long-term forecasts will go to the mean of the data.
  - If $c \neq 0$ and $d = 1$, the long-term forecasts will follow a straight line.
  - If $c \neq 0$ and $d = 2$, the long-term forecasts will follow a quadratic trend.

Backshift Notation

- The backward shift operator B is a useful notational when working with TS lags:
- $By_t = y_{t-1}$
- In other words, B has the effect of shifting the data back to one period.
- Two applications of B to y_t shifts the data back two periods
- $B(By_t) = B^2 y_t = y_{t-2}$
- The backshift operator is convenient to describe differencing
- First order differencing $y' = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t$
- d-th order differencing $(1 - B)^2 y_t$
- Seasonal difference followed by first difference: $(1 - B)(1 - B)^m y_t$

ARIMA in Backshift Notation

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

$$\underset{\substack{\uparrow \\ \text{AR}(p)}}{(1 - \phi_1 B - \cdots - \phi_p B^p)} \quad \underset{\substack{\uparrow \\ d \text{ differences}}}{(1 - B)^d y_t} \quad = \quad \underset{\substack{\uparrow \\ \text{MA}(q)}}{c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t}$$

- Example: ARIMA(3,1,1)

$$(1 - \hat{\phi}_1 B - \hat{\phi}_2 B^2 - \hat{\phi}_3 B^3)(1 - B) y_t = (1 + \hat{\theta}_1 B)\varepsilon_t.$$

ARIMA Lags Estimations with ACF and PACF

- It is sometimes possible to use the ACF plot, and the closely related PACF plot, to determine appropriate values for *p* and *q*.
- **ACF** plot shows the autocorrelations which measure the relationship between y_t and y_{t-k} for different values of *k*.
- **PACF** measure the relationship between y_t and y_{t-k} after removing the effects of lags $1,2,3,\ldots,k-1$.
- If the TS are from an ARIMA(p,d,0) or ARIMA(0,d,q), then the ACF and PACF plots can be helpful in determining the value of p or q.
- If *p* and *q* are both positive, then the plots do not help in finding suitable values of *p* and *q*.

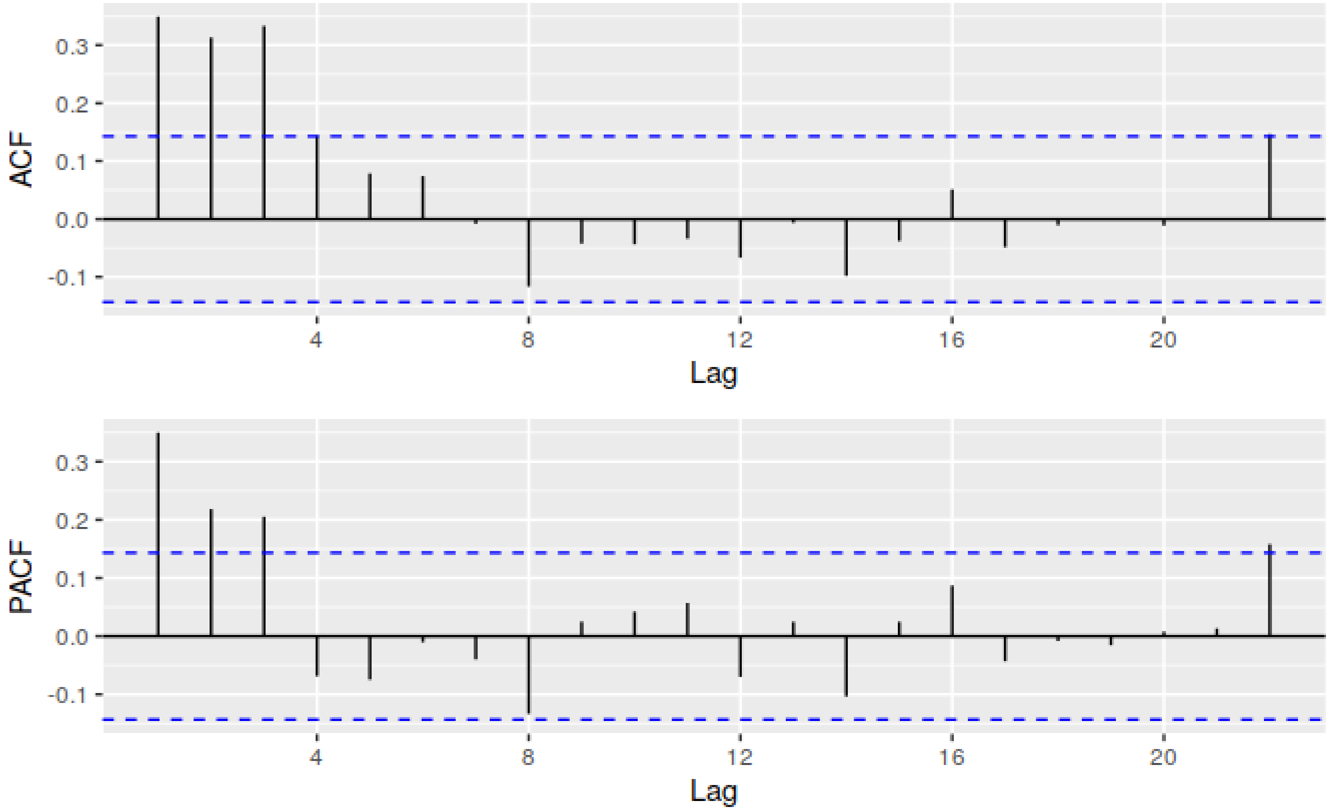
ARIMA Lags Estimations with ACF and PACF

- The TS may follow an ARIMA(p,d,0) model if the ACF and PACF plots of the differenced TS show the following patterns:
  - the ACF is exponentially decaying or sinusoidal;
  - there is a significant spike at lag *p* in the PACF, but none beyond lag *p*.

- The data may follow an ARIMA(0,d,q) model if the ACF and PACF plots of the differenced TS show the following patterns:
  - the PACF is exponentially decaying or sinusoidal;
  - there is a significant spike at lag *q* in the ACF, but none beyond lag *q*.

ACF and PACF plots - Example

- There are three spikes in the ACF, followed by an almost significant spike at lag 4. In the PACF, there are three significant spikes, and then no significant spikes.
- The pattern in the first three spikes is what we would expect from an ARIMA(3,0,0), as the PACF tends to decrease.
- In this case, the ACF and PACF lead us to think an ARIMA(3,0,0) model might be appropriate.



ARIMA – Parameters Estimation

- Once the model order has been identified (i.e., the values of *p,d,q*), we need to estimate the parameters $c, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_p$.
- *Maximum Likelihood Estimation* (MLE) can be used to find the values for these parameters.
- For ARIMA models, MLE is similar to the *least squares* estimates that would be obtained by minimizing

$$\sum_{t=1}^{T} \varepsilon_t^2.$$

- Once the parameters are estimated they are placed in the equation and used to make the prediction of $y_{t+1}, y_{t+2}, \dots, y_{t+n}$

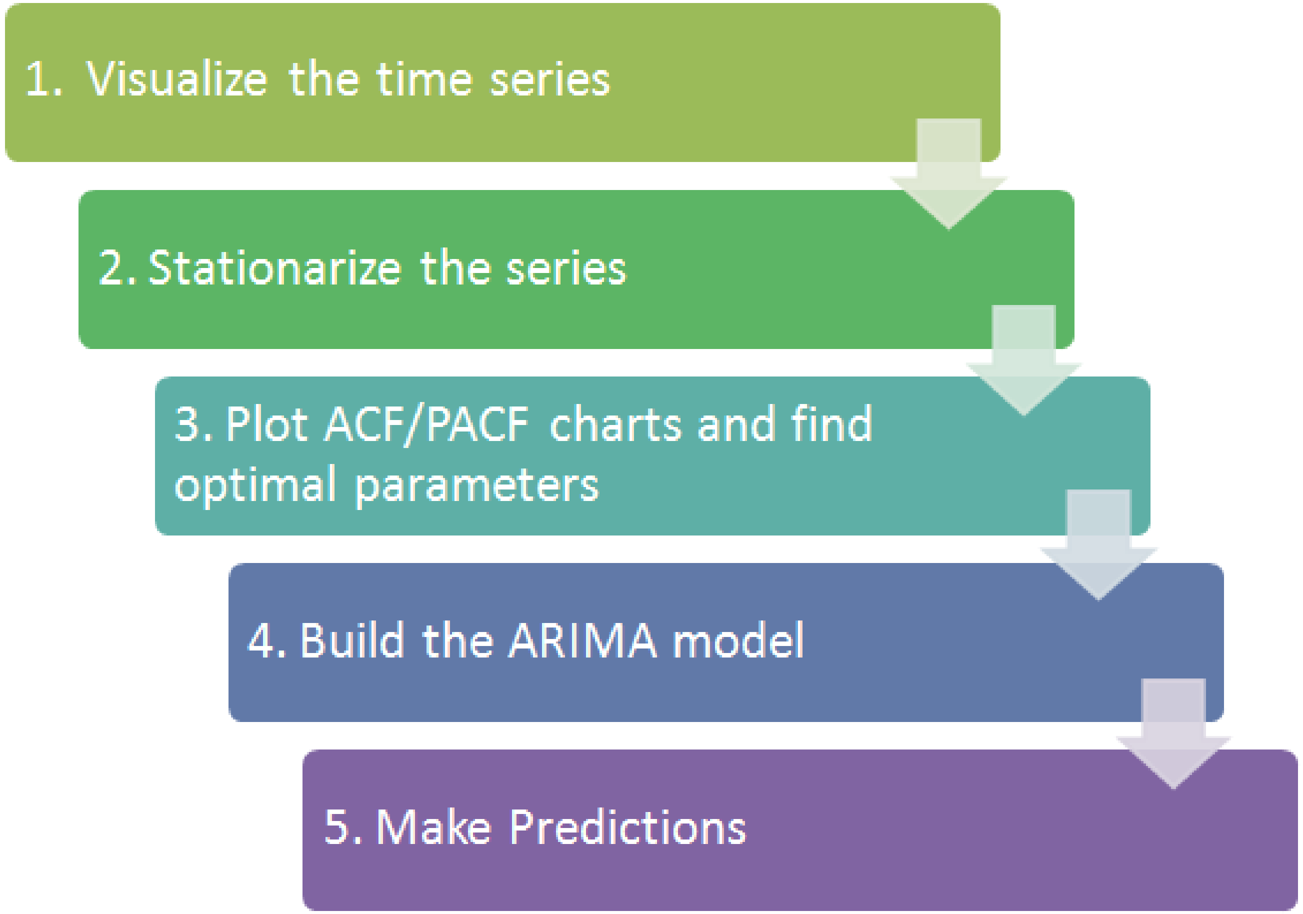
Determining the Order of an ARIMA model

- Akaike's Information Criterion (AIC) $\mathrm{AIC} = -2\log(L) + 2(p + q + k + 1)$

- Bayesian Information Criterion (BIC) $\mathrm{BIC} = \mathrm{AIC} + [\log(T) - 2](p + q + k + 1)$

- *L* is the likelihood of the data
- *k=1* if *c=0*, *k=0* otherwise
- Good models are obtained by minimizing the AIC, or BIC
- AIC, or BIC are not good guides to selecting the appropriate *d*.
- This is because the differencing changes the data on which the likelihood is computed, making the AIC values between models with different orders of differencing not comparable.

Modelling Procedure



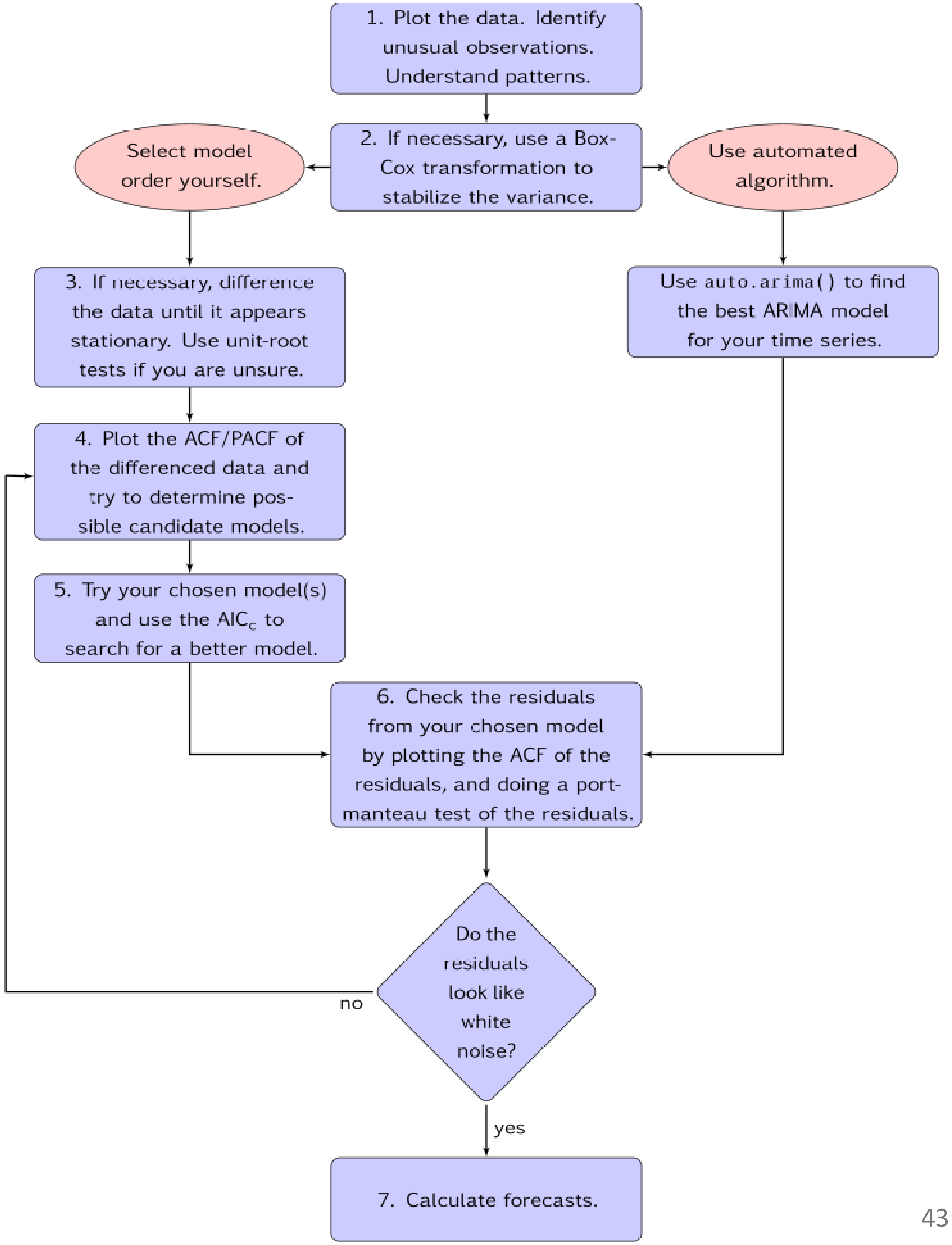
1. Visualize the time series

2. Stationarize the series

3. Plot ACF/PACF charts and find optimal parameters

4. Build the ARIMA model

5. Make Predictions



1. Plot the data. Identify unusual observations. Understand patterns.

2. If necessary, use a Box-Cox transformation to stabilize the variance.

Select model order yourself.

Use automated algorithm.

3. If necessary, difference the data until it appears stationary. Use unit-root tests if you are unsure.

Use `auto.arima()` to find the best ARIMA model for your time series.

4. Plot the ACF/PACF of the differenced data and try to determine possible candidate models.

5. Try your chosen model(s) and use the AIC_c to search for a better model.

6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals.

Do the residuals look like white noise?

7. Calculate forecasts.

SARIMA - Seasonal ARIMA Models

- ARIMA models can also model a wide range of seasonal data.
- A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far.

$$\text{ARIMA} \qquad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

- where *m* is the number of observations per period.
- The seasonal part consists of terms that are similar to the non-seasonal components but involve backshifts of the seasonal period.

SARIMA - Example

- ARIMA(1,1,1)(1,1,1)$_4$

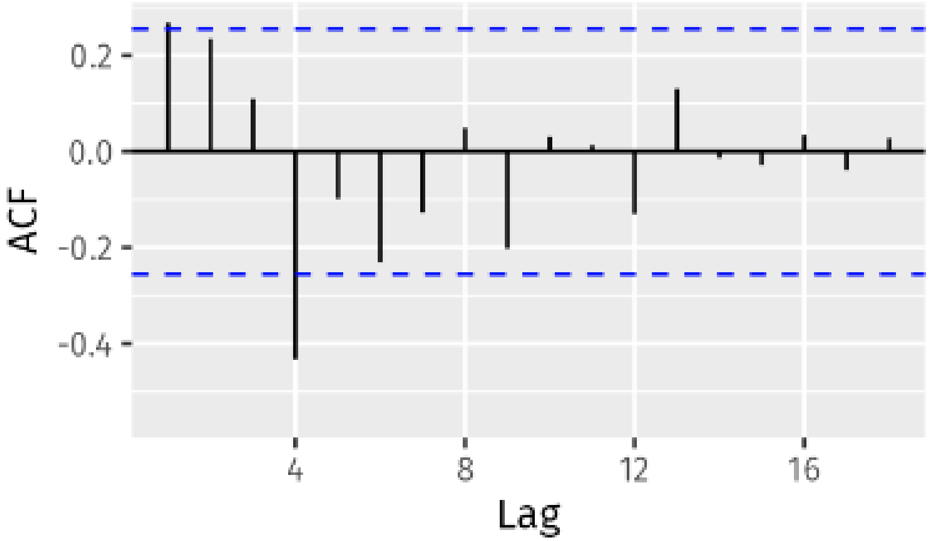
$$(1-\phi_1 B)\ (1-\Phi_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)\ (1+\Theta_1 B^4)\varepsilon_t$$

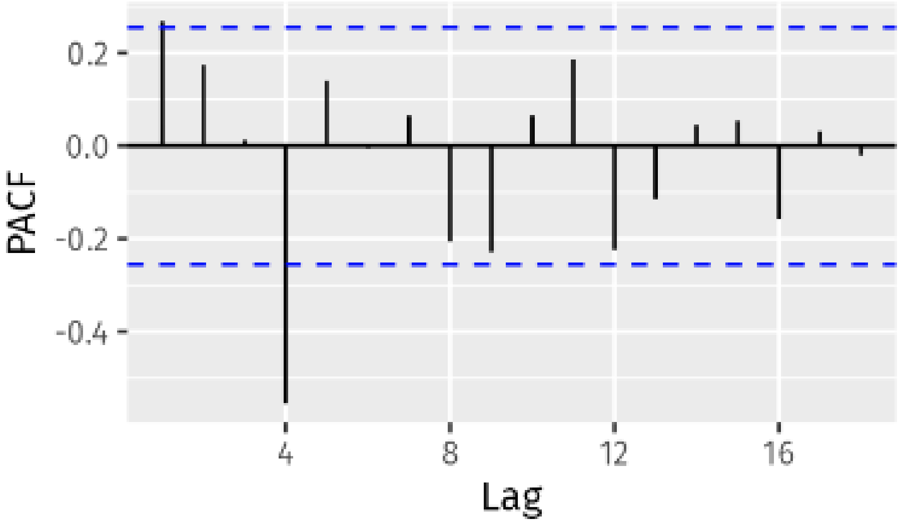
SARIMA Lags Estimations with ACF and PACF

- The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF.
- For example, an ARIMA(0,0,0)(0,0,1)$_{12}$ will show:
  - a spike at lag 12 in the ACF but no other significant spikes;
  - exponential decay in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, …)
- Similarly, an ARIMA(0,0,0)(1,0,0)$_{12}$ will show:
  - exponential decay in the seasonal lags of the ACF;
  - a single significant spike at lag 12 in the PACF.
- The procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components.
- In considering the appropriate seasonal orders for a seasonal ARIMA model, restrict attention to the seasonal lags.

ACF and PACF plots - Example

- From ACF: the significant spike at lag 1 suggests a non-seasonal MA(1), and the significant spike at lag 4 suggests a seasonal MA(1).
- From PACF: the significant spike at lag 1 suggests a non-seasonal AR(1), and the significant spike at lag 4 suggests a seasonal AR(1).
- Thus, we can consider
  - ARIMA(0,1,1)(0,1,1)$_4$ or
  - ARIMA(1,1,0)(1,1,0)$_4$





In Practice use AutoARIMA

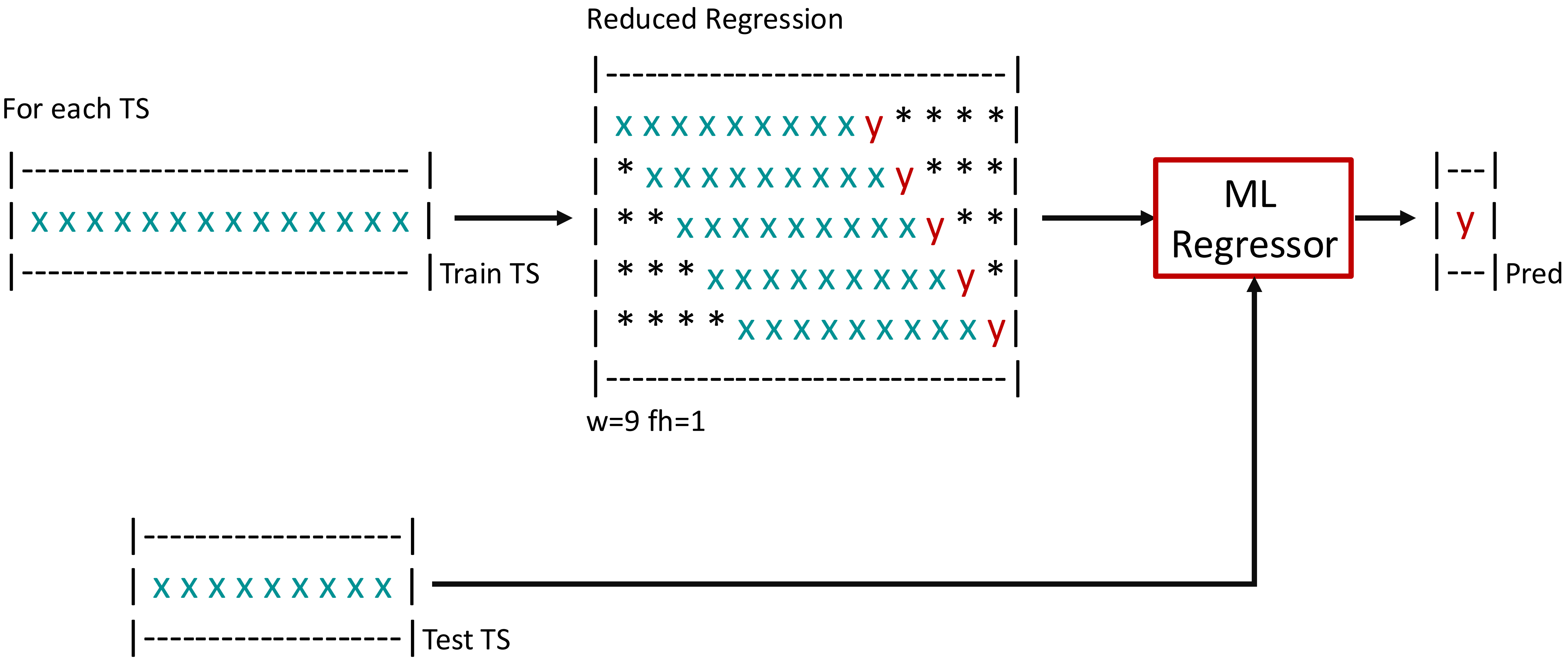
- AutoARIMA seeks to identify the most optimal parameters for an ARIMA model, settling on a single fitted ARIMA model.
- Auto-ARIMA works by conducting differencing tests (i.e., Kwiatkowski–Phillips–Schmidt–Shin, Augmented Dickey-Fuller or Phillips–Perron) to determine the order of differencing, d, and then fitting models within ranges of defined start_p, max_p, start_q, max_q ranges.
- If the seasonal optional is enabled, AutoARIMA also seeks to identify the optimal P and Q hyper-parameters after conducting the Canova-Hansen to determine the optimal order of seasonal differencing, D.

Prophet – Facebook TSF Method

- A “curve fitting” algorithm to build TSF models
- Additive Model: build a model by finding a best smooth line which can be represented as the sum of the following components
- **Trend**: Logistic Growth/Linear Trend with Changepoints
- **Multiple Seasonality** (Yearly, Monthly, Weekly, Daily, etc.): DFT
- **Holiday Effect** (Xmas, New Year, Easter, Halloween, etc.): custom list of dates with normal distribution to account for days surrounding the holiday
- **External Predictors** (normally distributed noise)
- Model Fitting with Stan’s L-BFGS optimization

Forecasting via Reduced Regression

Reduced Regression


Reduced Regression
For each TS
x x x x x x x x x x x x x x
Train TS
x x x x x x x x x y * * * *
* x x x x x x x x x y * * *
* * x x x x x x x x x y * *
* * * x x x x x x x x x y *
* * * * x x x x x x x x x y
w=9 fh=1
ML Regressor
y
Pred
x x x x x x x x x
Test TS

Reduced Regression

```
|------------------------------ |
| x x x x x x x x x x x x x x x |
|------------------------------ |TS len = 15
```

- Objective: train a TS forecaster based on a reduction of TS into tabular data.
- A sliding-window is used to transform the TS into tabular data which is then used to fit any ML regressor.
- By default, the first data point after the window is used as target variable, i.e., forecasting horizon = 1.
- The target variable can be set also after a certain forecasting horizon > 1.

```
|------------------------------|
| x x x x x x x x x y * * * *|
| * x x x x x x x x x y * * *|
| * * x x x x x x x x x y * *|
| * * * x x x x x x x x x y *|
| * * * * x x x x x x x x x y|
|------------------------------|
w=9 fh=1
```

```
|------------------------------|
| x x x x x x x x x * y * * *|
| * x x x x x x x x x * y * *|
| * * x x x x x x x x x * y *|
| * * * x x x x x x x x x * y|
|------------------------------|
w=9 fh=2
```

```
|------------------------------|
| x x x x x x x x x * * * y *|
| * x x x x x x x x x * * * y|
|------------------------------|
w=9 fh=4
```

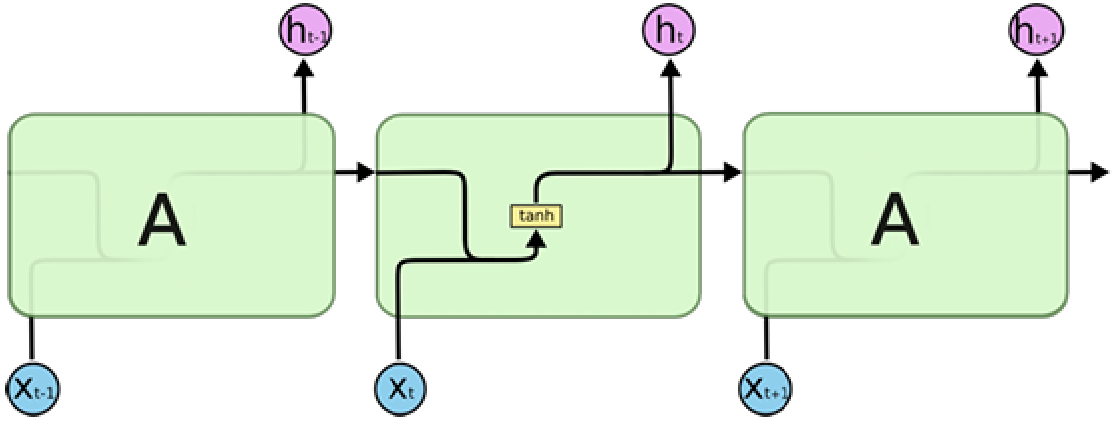
ML + Reduced Regression

- After Reduced Regression any ML model can be used!
- ROCKET
- Ensemble Methods
- Shapelets
- HIVE-COTE
- DrCIF
- Decision Tres
- etc.

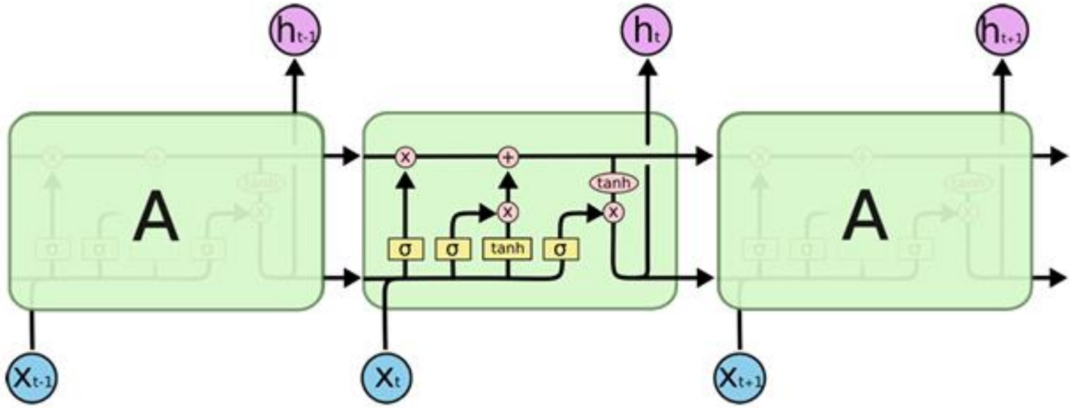
Deep-learning Models

LSTM - Long Short Term Memory

- LSTM are a special RNN, capable of learning long-term dependencies.
- All RNN have the form of a chain of repeating modules of neural network with a single layer typically with tanh activation.
- LSTM have the same structure but different repeating module.



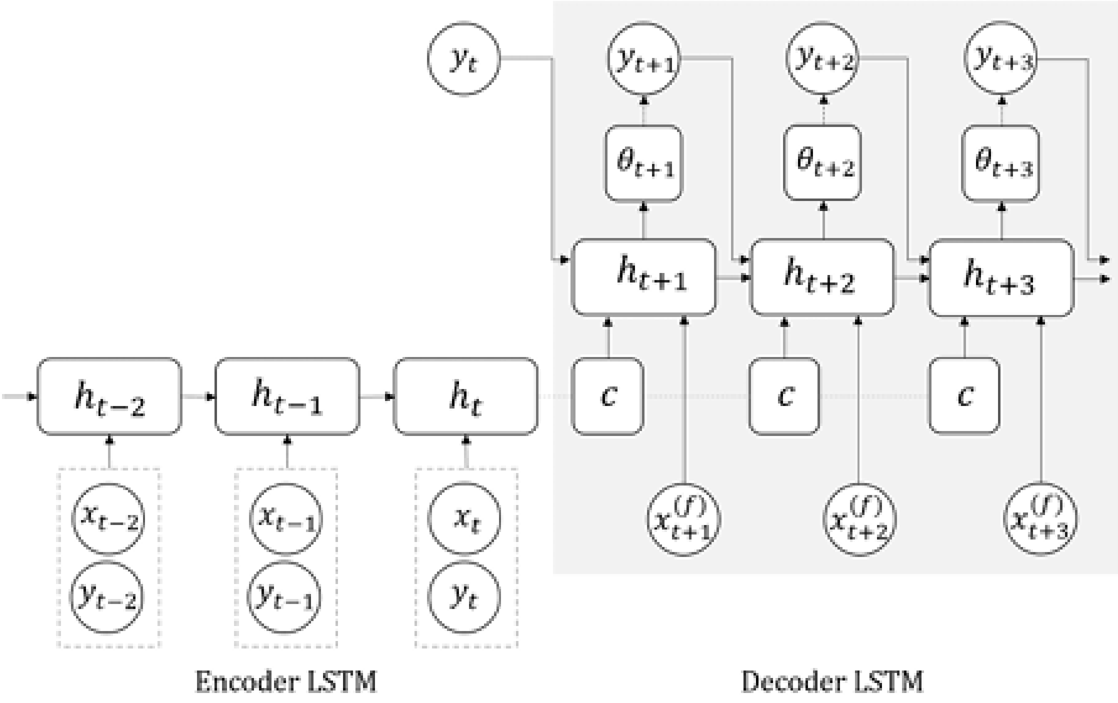
RNN



LSTM

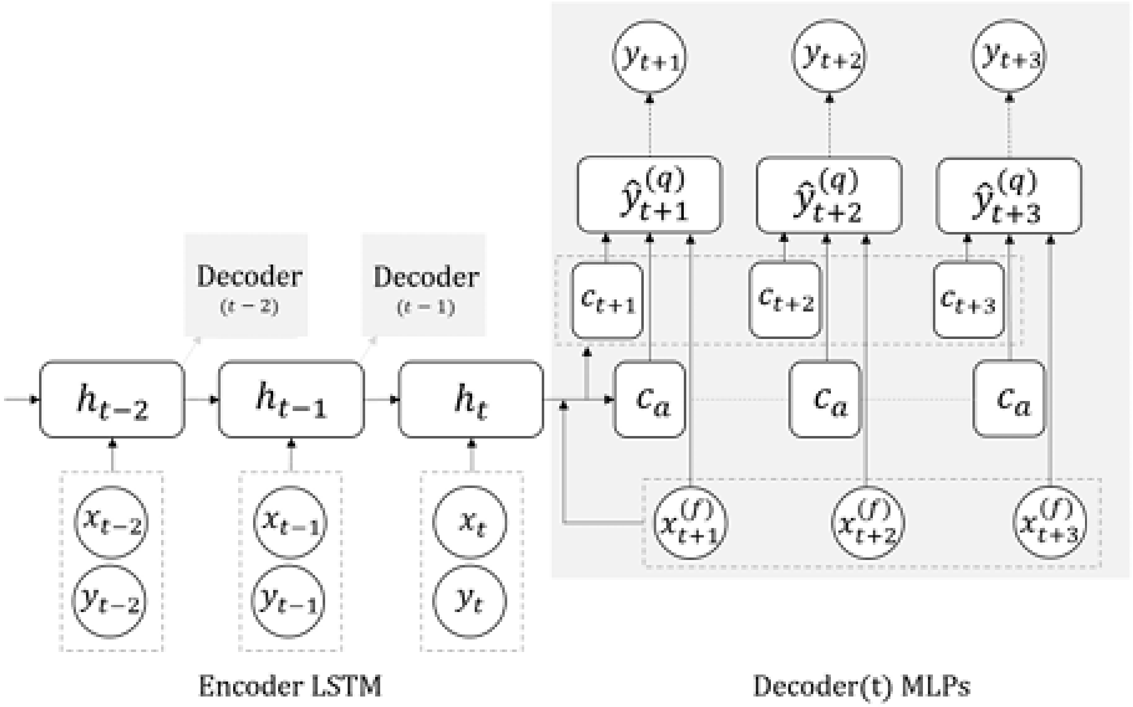
LSTM for Time Series Forecasting

- Sequence to Sequence Architecture: LSTM encoder + LSTM decoder.
- The network has *tanh* or *relu* non-linearities and is trained using ADAM SGD.
- It accepts historic and future data.
- Predictions are obtained by transforming the hidden states into a horizon-agnostic context which captures common information that are decoded and adapted into the predicted value through LSTM.

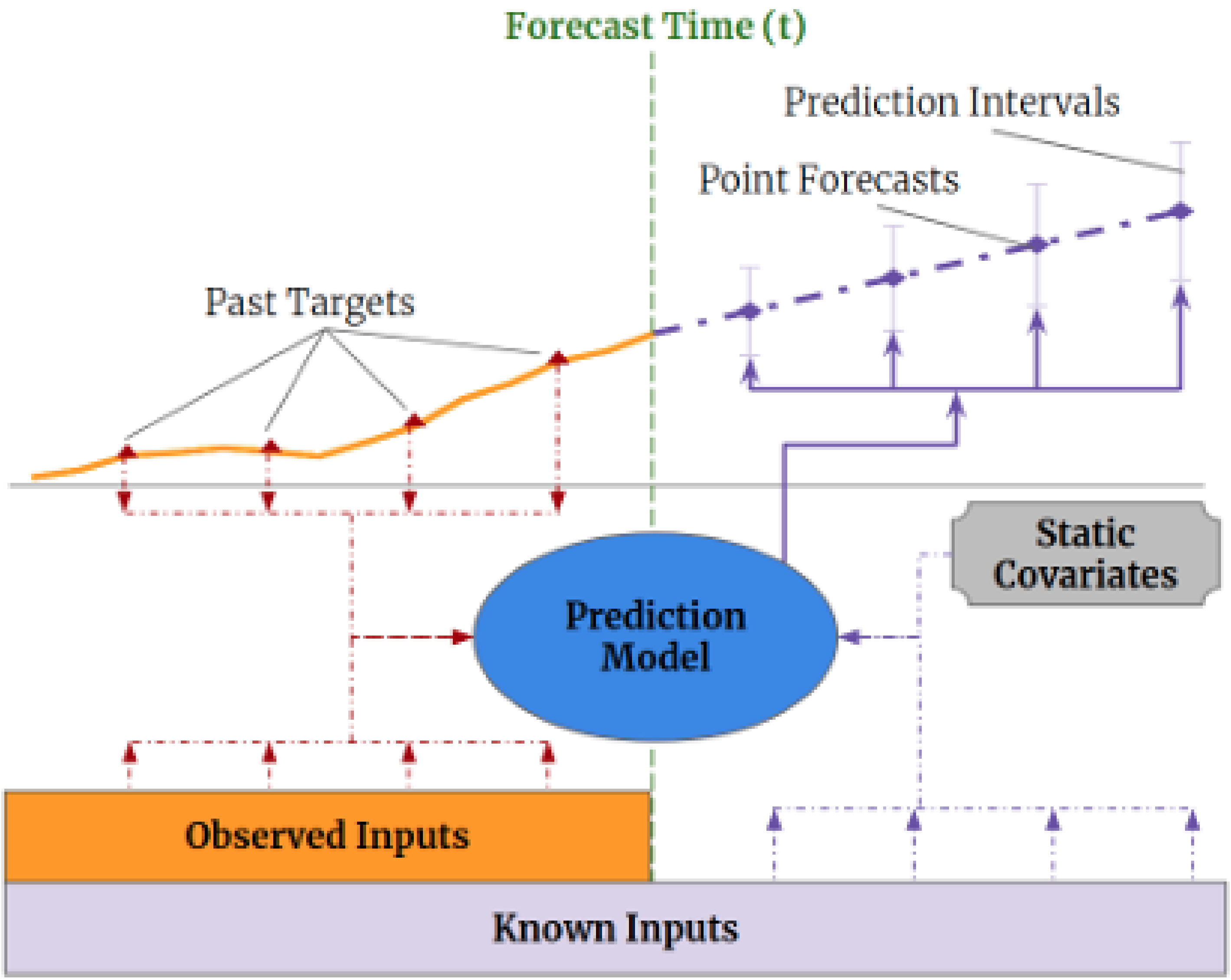


LSTM for Time Series Forecasting

- Sequence to Sequence Architecture: LSTM encoder + MLP decoder.
- The network has *tanh* or *relu* non-linearities and is trained using ADAM SGD.
- It accepts static, historic and future exogenous data.
- Predictions are obtained by transforming the hidden states into horizon-specific contexts c_{t+k} for each future points, and a horizon-agnostic context c_a which captures common information that are decoded and adapted into the predicted value through MLPs.



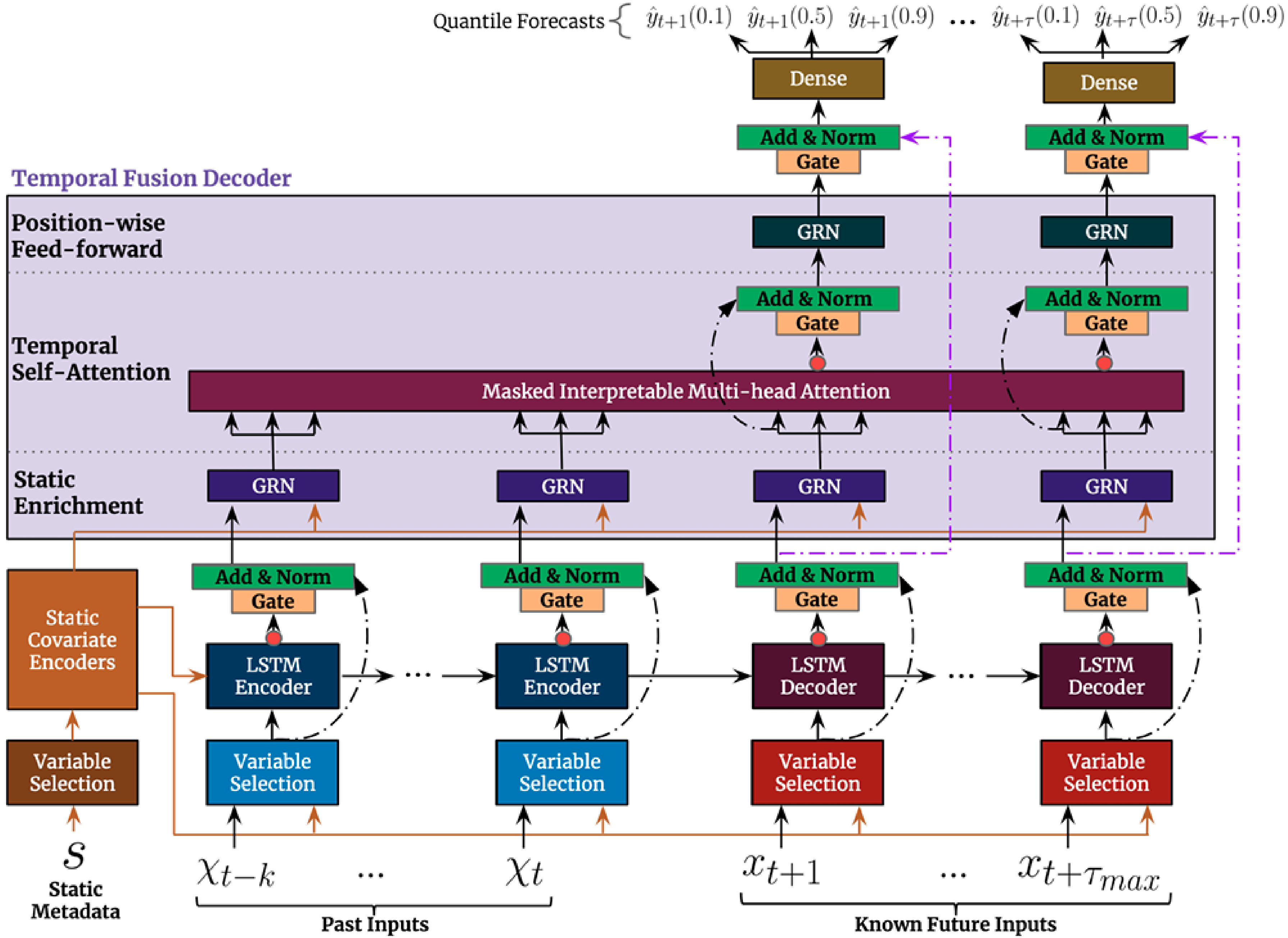
Forecasting with Static Covariates

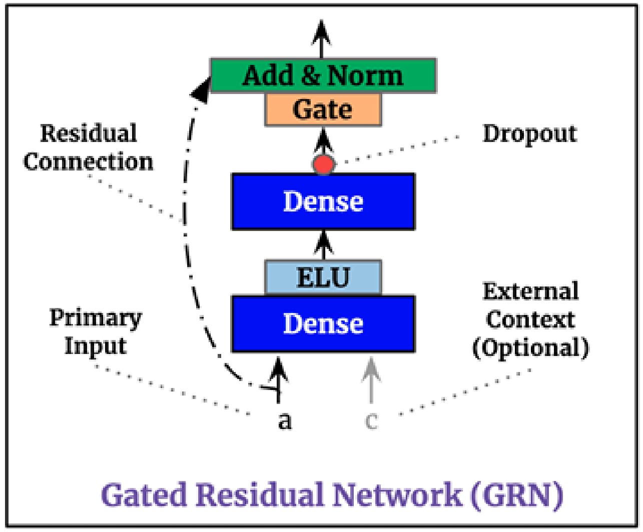


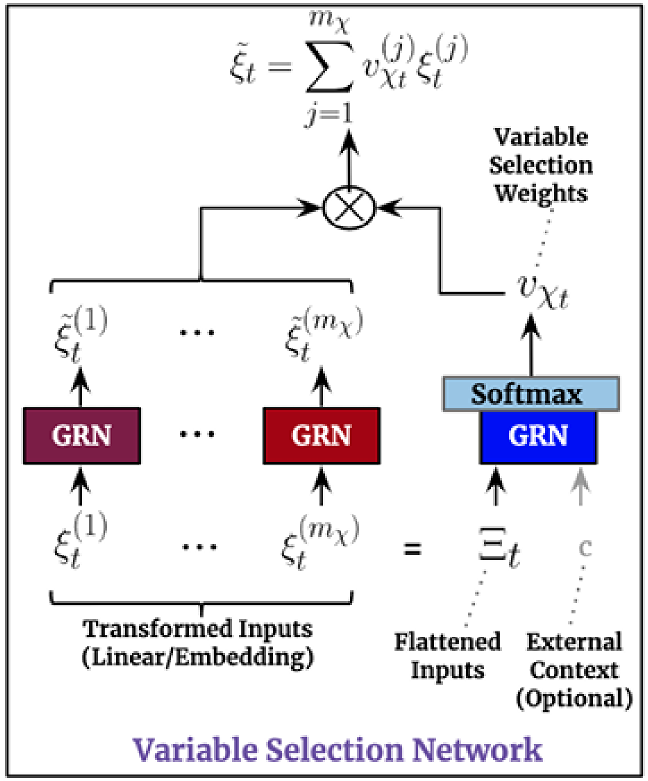
TFT - Temporal Fusion Transformers

- TFT is an attention-based architecture which combines high-performance multi-horizon forecasting with interpretable attention.
- TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies.
- TFT utilizes specialized components to select relevant features and a series of gating layers to suppress unnecessary components.

TFT Architecture


Quantile Forecasts
$\hat{y}_{t+1}(0.1)$ $\hat{y}_{t+1}(0.5)$ $\hat{y}_{t+1}(0.9)$ … $\hat{y}_{t+\tau}(0.1)$ $\hat{y}_{t+\tau}(0.5)$ $\hat{y}_{t+\tau}(0.9)$
Dense
Add & Norm
Gate
Temporal Fusion Decoder
Position-wise Feed-forward
GRN
Temporal Self-Attention
Masked Interpretable Multi-head Attention
Static Enrichment
Static Covariate Encoders
LSTM Encoder
LSTM Decoder
Variable Selection
s
Static Metadata
χ_{t-k} … χ_t
x_{t+1} … $x_{t+\tau_{max}}$
Past Inputs
Known Future Inputs



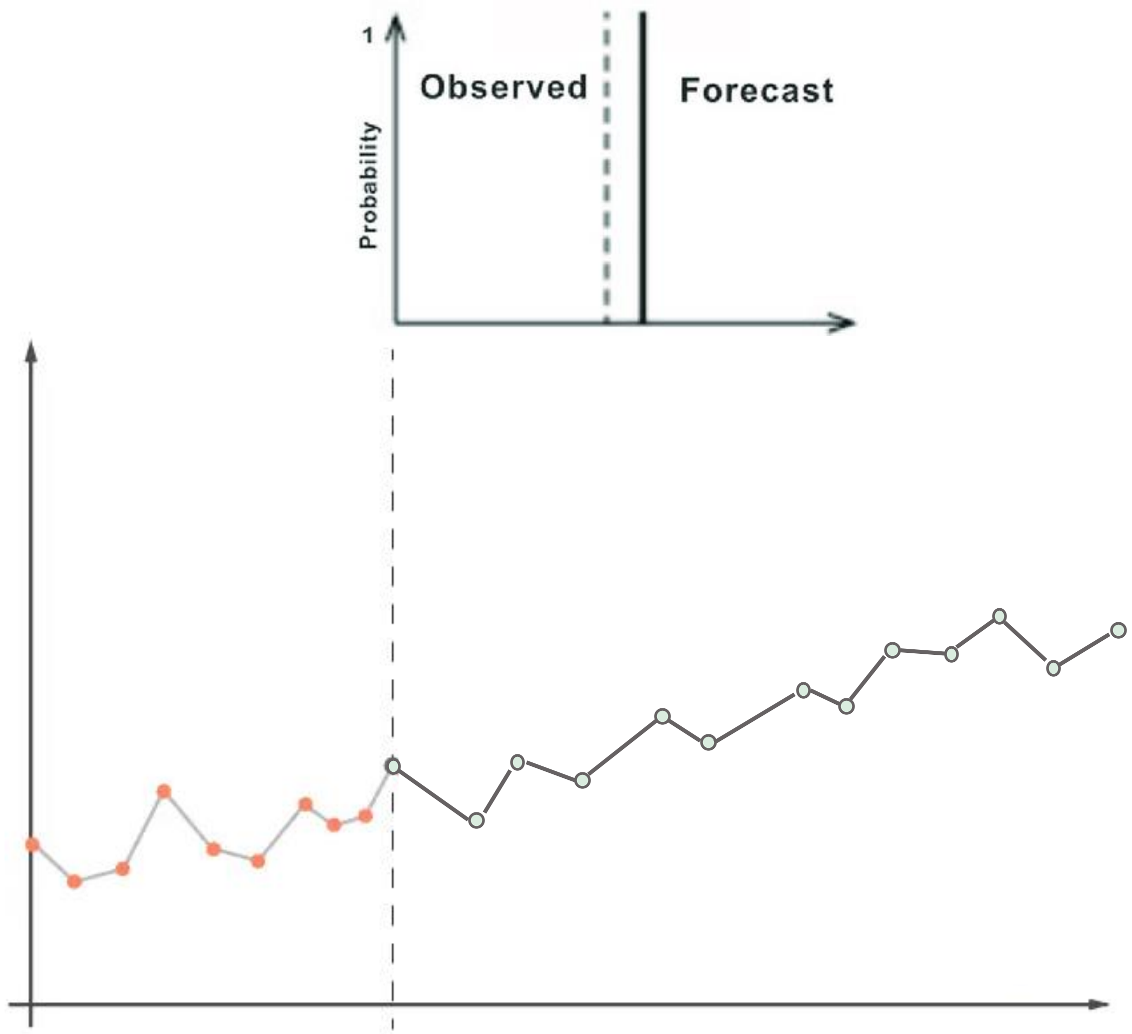

$\tilde{\xi}_t = \sum_{j=1}^{m_\chi} v_{\chi_t}^{(j)} \tilde{\xi}_t^{(j)}$
Variable Selection Weights
v_{χ_t}
$\tilde{\xi}_t^{(1)}$ … $\tilde{\xi}_t^{(m_\chi)}$
Softmax
GRN … GRN
GRN
$\xi_t^{(1)}$ … $\xi_t^{(m_\chi)}$ = Ξ_t
c
Transformed Inputs (Linear/Embedding)
Flattened Inputs
External Context (Optional)
Variable Selection Network

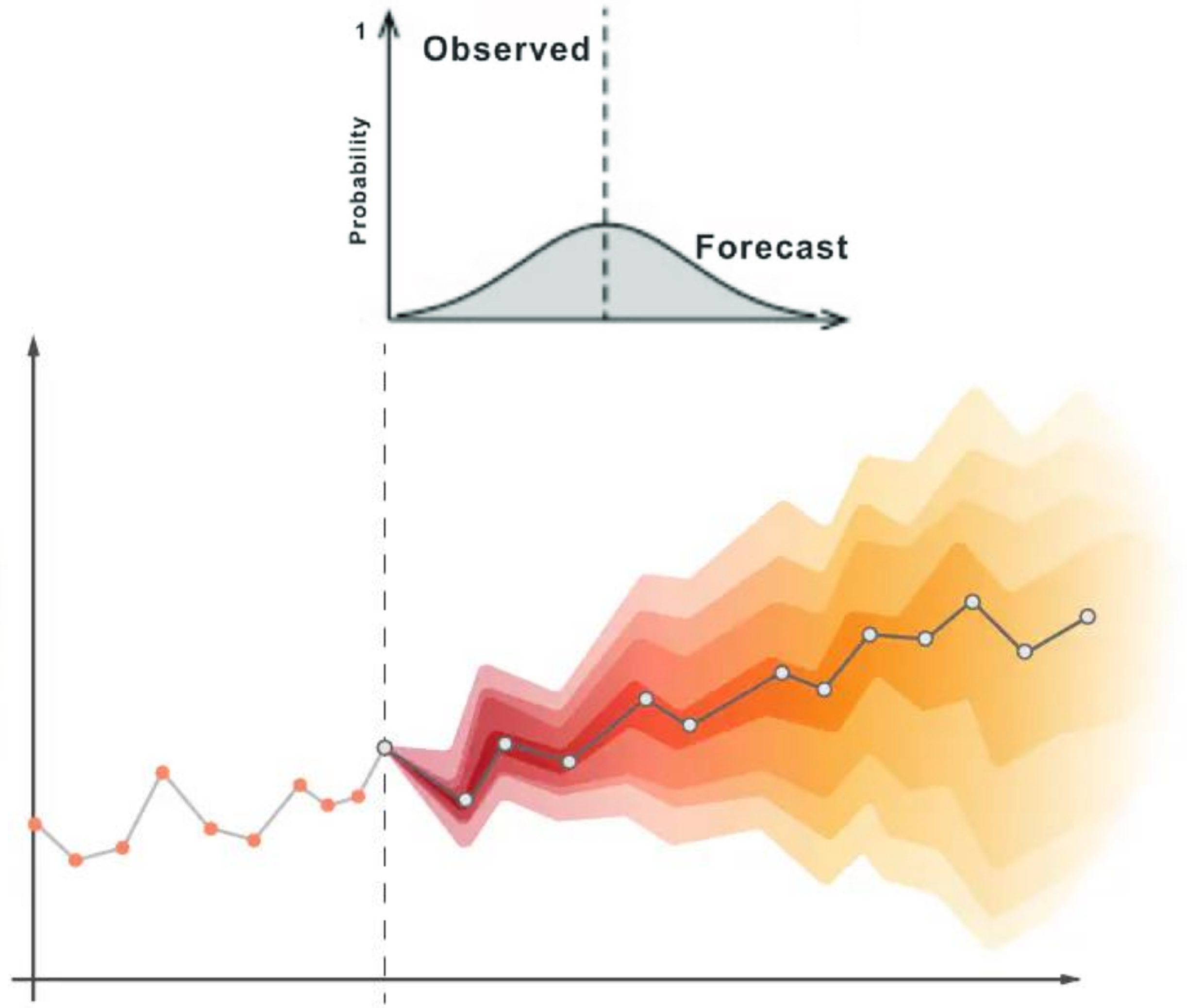
Probabilistic Forecasting

Probabilistic Forecasting

- Deterministic point forecasts produces a single estimate for each time step in the forecast horizon, i.e., one value for each of the future time periods $t = N + 1,..., N + k$ assuming a forecast horizon of k steps ahead starting at forecast origin N.
- Deterministic point estimates are limited in their description of the future demand, especially when the underlying data has a large degree of uncertainty.
- A more detailed picture of the possible future values can be produced by **estimating the distribution of the future values** for each period in the forecast horizon.
- Forecasts which estimate the spread of the distribution are called **probabilistic forecasts**.

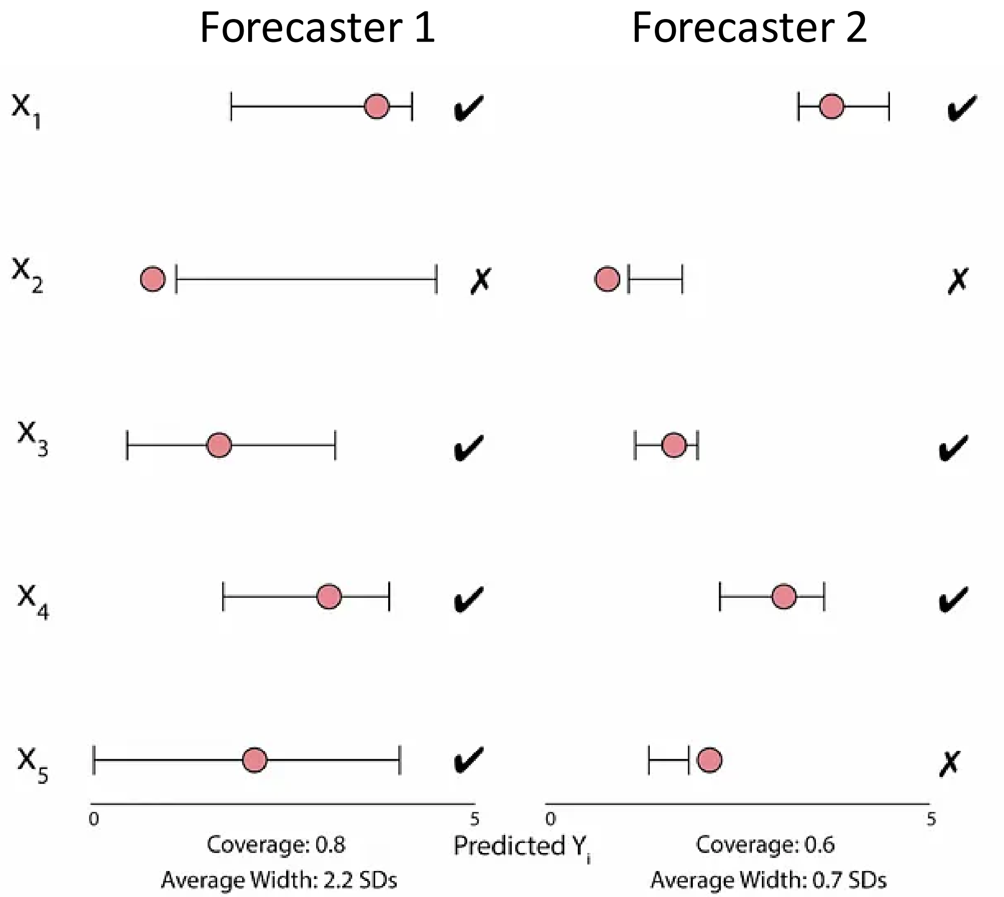
Deterministic vs Probabilistic Forecasting





Evaluating Probabilistic Forecasting

- Validity/Coverage: fraction of real values contained in predicted intervals.



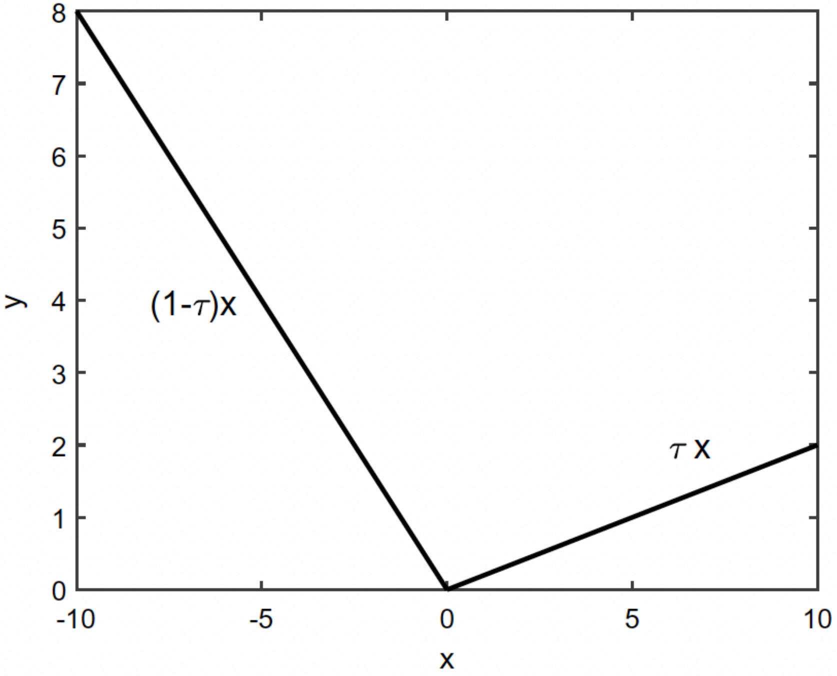
Evaluating Probabilistic Forecasting

- Pinball Loss Score (PLS): average over each loss over each quantile.
- The pinball function is an asymmetric function which takes the difference between the observation and the quantile and then weights the difference differently depending on whether the value is positive or negative.

$$\mathrm{PLS} = \frac{1}{N}\sum_{k=1}^{N} L_{\tau_k}(L, z_{\tau_k})$$

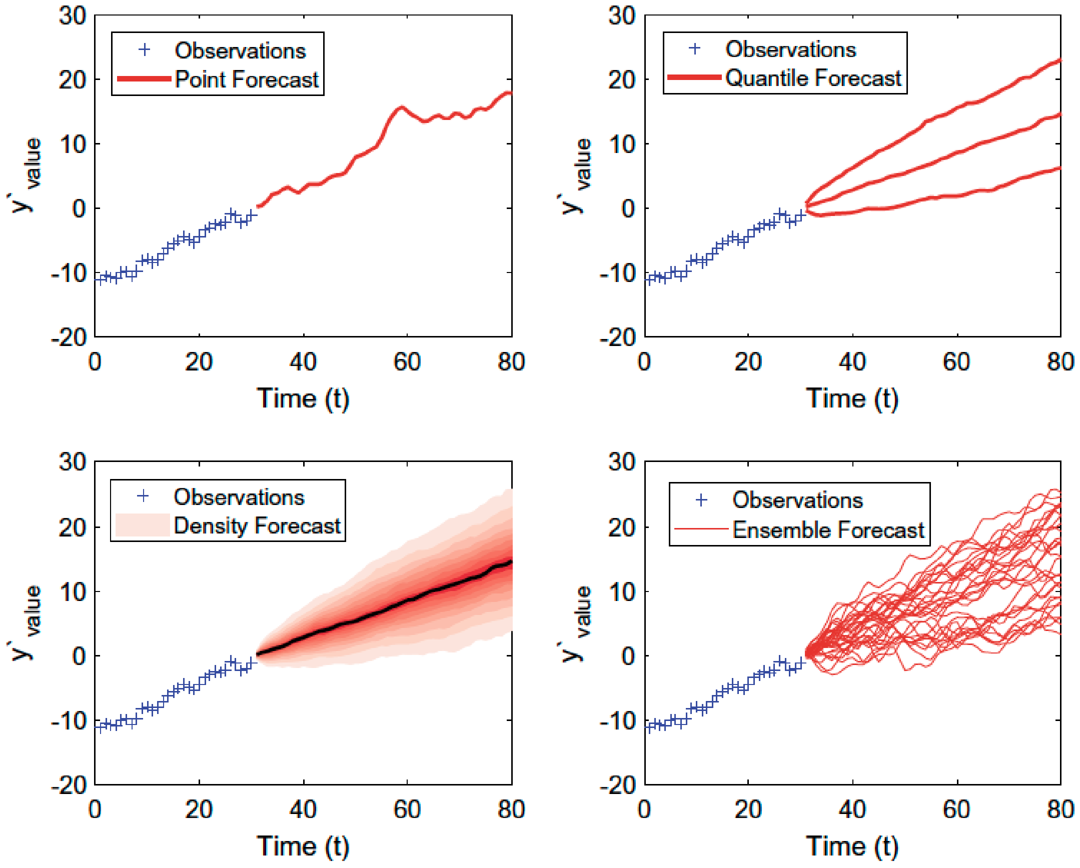
$$L_\tau(L, z_\tau) = \begin{cases} \tau(L - z_\tau) & L \geq z_\tau \\ (1-\tau)(z_\tau - L) & L < z_\tau \end{cases}$$

τ quantile in [0, 1]; z_τ value of the quantile



Forms of Probabilistic Forecast

- **Quantile Forecast**: several quantiles of the future values are estimated. If two quantiles are used (a high and low) then the area between the two values is called the **forecast interval**. The 10% and 90% quantiles are common choices.
- **Density Forecast**: the full continuous distribution is estimated for each time step.
- **Ensemble/Scenario Forecast**: quantile and density forecasts only estimate a distribution at each time step in the forecast horizon. The time steps are often interdependent with the values at earlier time periods influencing the values at later time periods. Ensemble forecasts estimate realisations from the full joint multivariate distribution for the set of random variables

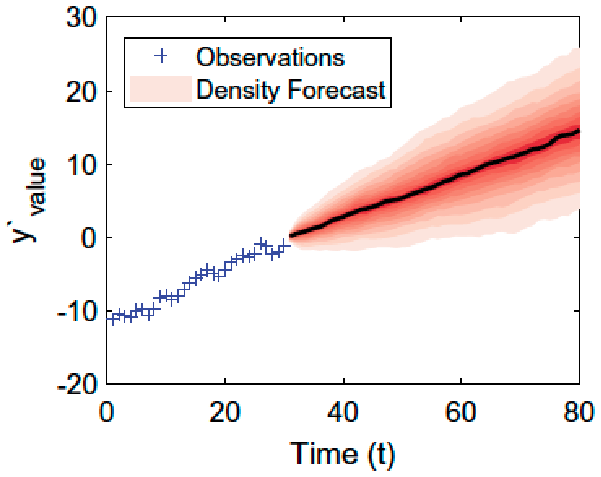


Estimating Future Distributions

- To estimate the future distributions all the different models make an assumption about how the past distribution will relate to the future demand.
- Types of Models:
  - Parametric Models
  - Kernel Density Estimation Models
  - Quantile Regression Models
- For each model the aim is to train the parameters or hyperparameters of a distribution model directly (e.g. the Gaussian model) or use a model which will estimate the distribution (e.g. quantiles).

Parametric Models

- Simple Univariate Distributions: assume a simple distribution such as Gaussian distribution and estimate the parameters on the past through sample mean and sample standard deviation and use them as probabilistic model.
- Mixture Models: combination of simple parametric distributions such as Gaussian Mixture Model where parameters can be estimated through Expectation Maximization algorithm (EM)



Quantile Regression Models

- Consider estimating the *q* quantiles (popular choices are deciles, i.e., 10-quantiles).

coefficients

TS

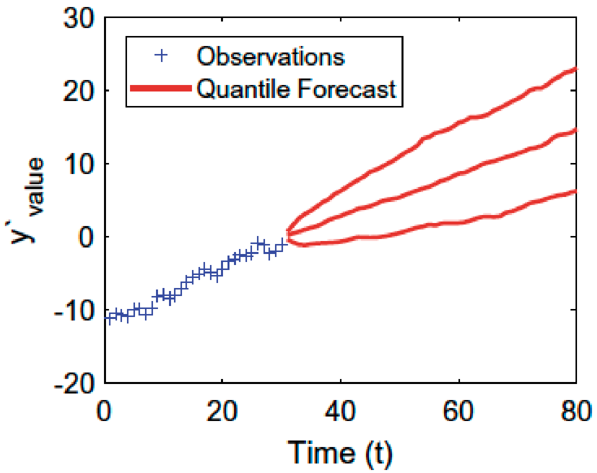
$$\hat{\beta} = \arg\min_{\beta\in\mathcal{B}} \left(\sum_{t=1}^{N} (L_t - f_t(\mathbf{Z}, \beta))^2 \right)$$

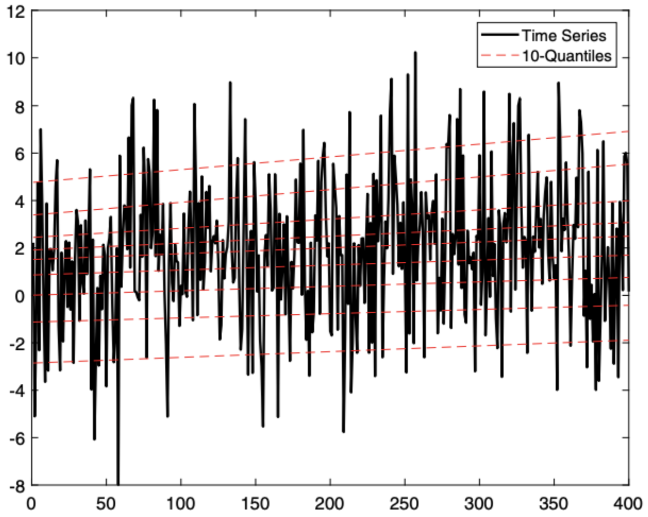
past data

repeated for all quantile

$$\hat{\beta}_\tau = \arg\min_{\beta\in\mathcal{B}} \left(\min \sum_{t=1}^{N} c_\tau(L_t, f_t(\mathbf{Z}, \beta)) \right)$$

Regression





Quantile Regression

$$c_\tau(x, y) = \begin{cases} \tau(x - y) & x \geq y \\ (1 - \tau)(y - x) & x < y \end{cases}$$

τ quantile

Kernel Density Estimation Models

- KDEs are summations of small smooth functions *K* called kernels which are defined around each observation of a variable to contribute to the PDF defined as

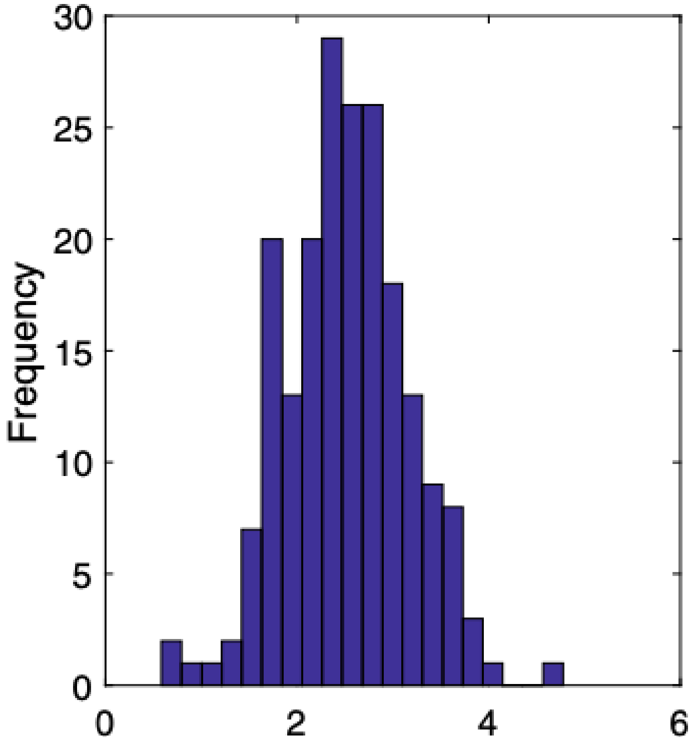
$$\hat{F}(X) = \frac{1}{Nh} \sum_{j=1}^{N} K\left(\frac{X - X_j}{h}\right)$$

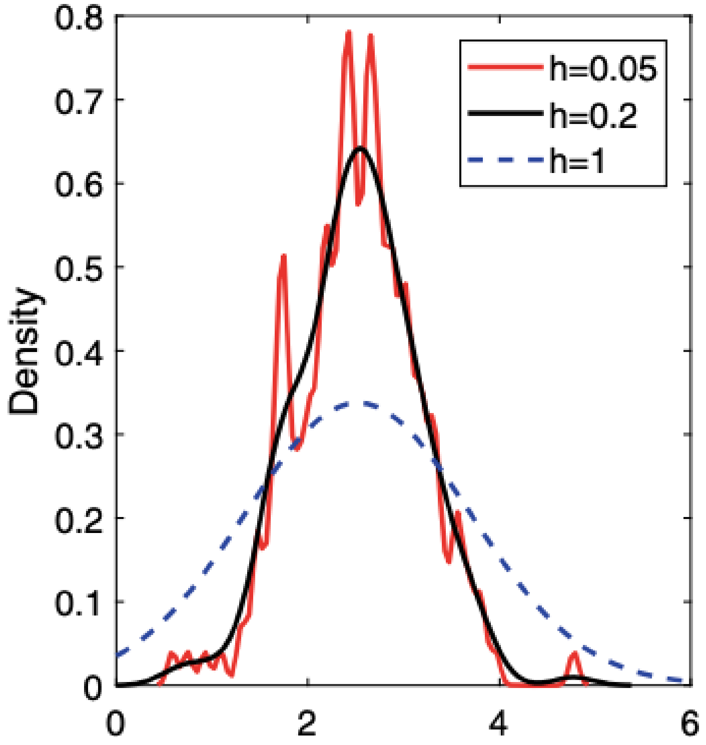
- where *h* is the bandwidth a smoothing parameter for the estimate, and *K()* is a kernel function. A widely used kernel function is the Gaussian kernel

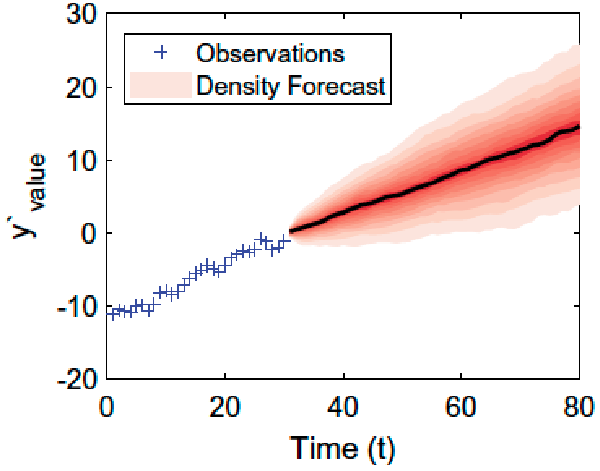
$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x\right)$$

Kernel Density Estimation Models

- The chosen kernel is often less important than the proper training of the bandwidth *h*.
- The choice of kernel function has no relationship to the true distribution, i.e., a Gaussian kernel does not mean the data is distributed as a Gaussian.







Ensemble/Scenario Forecast Models

- Ensemble Forecast: a set of point forecasts from the same forecast origin, estimating each time step with the same forecast horizon (of length *h* time steps).
- The point forecasts are samples of equal probability from a h-dimensional multivariate distribution representing the joint distribution over the forecast horizon.
- Each ensemble represents an equally likely forecast TS.

Residual Bootstrap Ensembles (Homoscedasticity)

- Let consider a 1-step ahead point forecast model such as ETS or ARIMA denoted as $f(L_t|Z, \beta)$ to estimate the future value
- $L_{t+1|t} = f(L_t|Z, \beta)$
- The next time step ahead can be forecasted by including the forecast from the previous time step as pesudo-historical input for the model
- $L_{t+2|t} = f(L_{t+1|t}| \mathbf{L_t}, Z, \beta) = f(f(L_t|Z, \beta)| \mathbf{L_t}, Z, \beta)$
- The process can be repeated for *k* steps ahead

Estimating Future Distributions - sktime

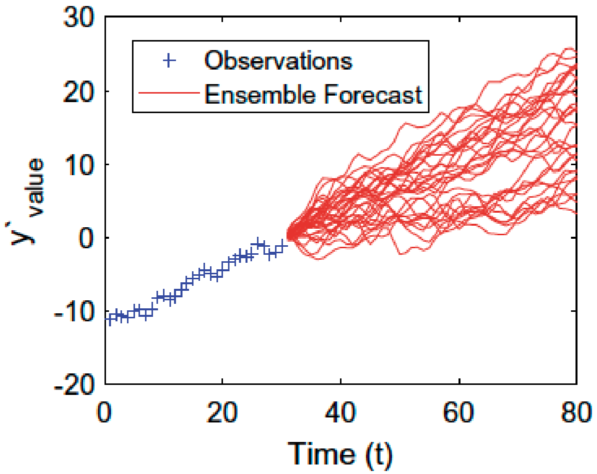
Let $y(t_1)$, …, $y(tn)$ be observations at fixed time stamps t_1, …, t_n.

Let y' be a (true) value, which will be observed at a future time stamp τ.

We have the following “types of forecasts” of y':

- **Point Forecast**: is a prediction of the conditional expectation $E[y'|y]$: out of many repetitions, this value is the arithmetic average of all observations (what we have seen with non probabilistic forecast).
- **Variance Forecast**: is a prediction of the conditional expectation $Var[y'|y]$: out of many repetitions, this value is the average squared distance of the observation to the perfect point forecast.
- **Distribution Forecast (Parametric Model)**: is a prediction of the distribution of $y'|y$, e.g., “it’s a normal distribution with mean 42 and variance 1”: exhaustive description of the generating mechanism of many repetitions.
- **Quantile Forecast**: at quantile point $\alpha \in (0,1)$ is a prediction of the α-quantile of $y'|y$: out of many repetitions, a fraction of exactly α will have equal or smaller than this value.
- **Interval Forecast**: with (symmetric) coverage $c \in (0,1)$ is a prediction pair of lower bound a and upper bound b such that $P(a \leq y' \leq b|y) = c$ and $P(y' \gneqq b|y) = P(y' \lneqq a|y) = (1-c)/2$: out of many repetitions, a fraction of exactly c will be contained in the interval $[a,b]$, and being above is equally likely as being below.

Residual Bootstrap Forecast



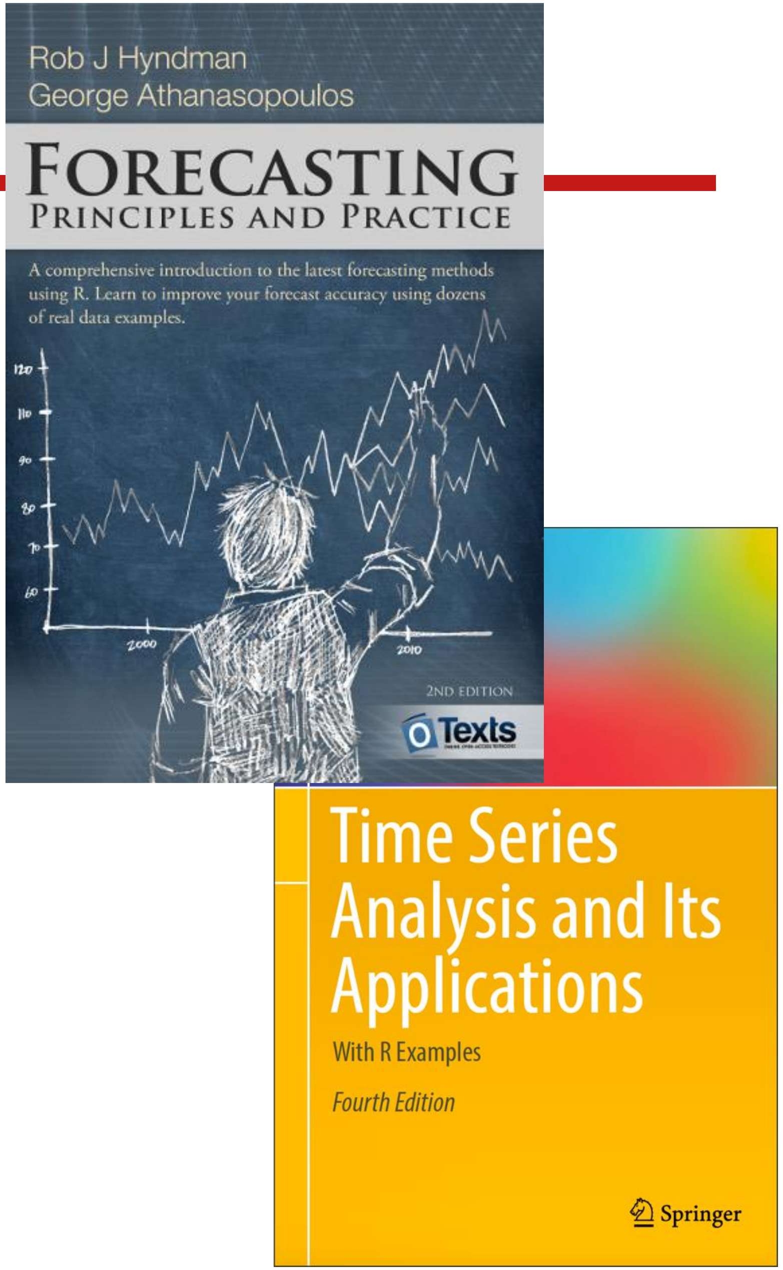
Consider the homoscedastic residual series $\{e_1, \ldots, e_k\}$ where $e_t = L_{t+1} - L'_{t+1}$

For each ensemble *b* repeat the following procedure:

- Randomly sample with replacement (bootstrap) a residual $e_1^{(b)}$ from the set of residuals
- Add $e_1^{(b)}$ to the current 1-step ahead forecast value to produce a new value $L_{t+1|t}{}^{(b)} = L_{t+1|t}{}^{(b)} + e_1^{(b)}$
- Include $L_{t+1|t}{}^{(b)}$ in the forecast model to generate an estimate for the next time step $L_{t+2|t}{}^{(b)} = f\,(L_{t+1|t}{}^{(b)} \mid L_t\,, Z, \beta)$
- Randomly sample with replacement a residual $e_2^{(b)}$ from the set of residuals
- Add $e_2^{(b)}$ to the current 2-step ahead forecast value to produce a new value $L_{t+2|t}{}^{(b)} = L_{t+2|t}{}^{(b)} + e_2^{(b)}$
- Continue this procedure *k* times
- The resulting $L_{t+1|t}{}^{(b)}, L_{t+2|t}{}^{(b)}, \ldots, L_{t+k|t}{}^{(b)}$ series is the b-the bootstrap enseble

References

- Forecasting: Principles and Practice. Rob J Hyndman and George Athanasaopoulus. (https://otexts.com/fpp2/)
- Time Series Analysis and Its Applications. Robert H. Shumway and David S. Stoffer. 4th edition.(http://www.stat.ucla.edu/~frederic/415/S23/tsa4.pdf)
- Time Series Analysis in R (https://s-ai-f.github.io/Time-Series/)
- Mining Time Series Data. Chotirat Ann Ratanamahatana et al. 2010. (https://www.researchgate.net/publication/227001229_Mining_Time_Series_Data)
- Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. Hansika Hewamalage et al. 2020.
- Forecasting at Scale. Sean J. Taylor et al. 2017.
- Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. Bryan Lim et al. 2020.
- Core Concepts and Methods in Load Forecasting. Stephen Haben et al. 2023. (https://library.oapen.org/bitstream/handle/20.500.12657/63002/1/978-3-031-27852-5.pdf)



Rob J Hyndman
George Athanasopoulos

FORECASTING
PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.

2ND EDITION

OTexts

Time Series Analysis and Its Applications

With R Examples

Fourth Edition

Springer