

The IMF Central Bank Liquidity Forecasting Framework

Romain Lafarguette, Ph.D.

Quant & IMF External Expert¹

<https://romainlafarguette.github.io/>

Singapore Training Institute, 09 November 2022



¹ This training material is the property of the International Monetary Fund (IMF) and is intended for use in IMF courses. Any reuse requires the permission of the IMF.

Outline of the Course

- ① The IMF Forecasting Framework
- ② Q & A
- ③ ETS Model
- ④ ARIMA Models
 - Modelling and Forecasting with ARIMA Models
- ⑤ Seasonal ARIMA (SARIMA)
 - ARIMA versus ETS

Table of Contents

1 The IMF Forecasting Framework

2 Q & A

3 ETS Model

4 ARIMA Models

- Modelling and Forecasting with ARIMA Models

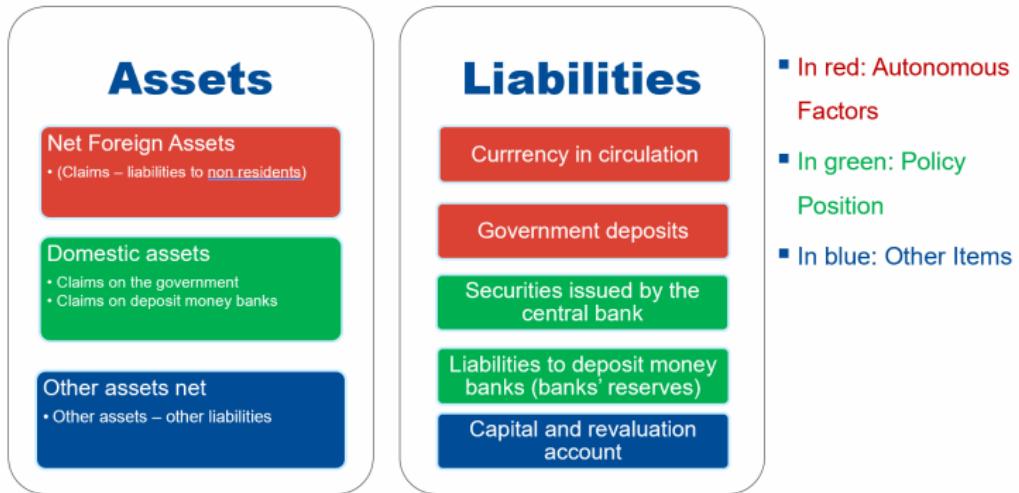
5 Seasonal ARIMA (SARIMA)

- ARIMA versus ETS

Overview: The IMF Liquidity Forecasting Framework

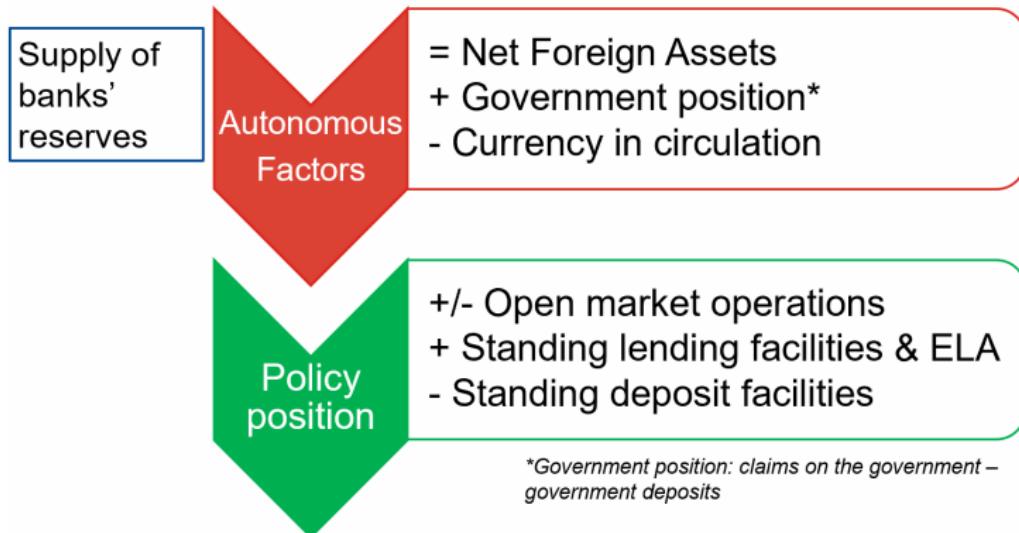
- Forecasting infrastructure based on high-quality **open-source codes**
 - Forecast the three main autonomous factors separately and the net liquidity position
 - Using the most recent methods in the statistical literature
- **Easy to use** for central banks: complex modeling issues are taken care of automatically
 - Open-source codes, free to use, up to the best programming standards
 - Provides automatic reporting and data visualization for daily operations
- **Team of experts to help** the authorities during missions
 - Deploy the model and help the authorities to customize it and use it

Refresher: Analytical Central Bank Balance Sheet



Credit: Author

Autonomous Factors and Policy Position



Credit: Author

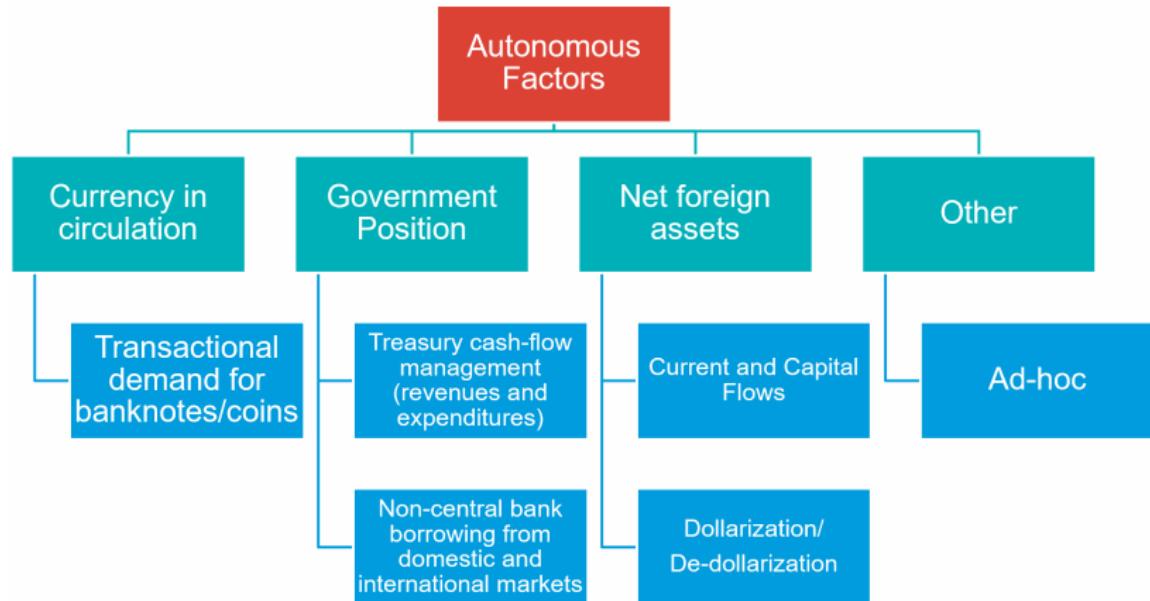
Liquidity Forecasting and Management

- Calibration of the regular, **short-term OMO** (Open Market Operations)
 - OMO to reach central bank optimal level/growth rate, based on the autonomous factors forecasts
- **Fine-tuning operations**
 - Fine-tuning operations to smooth unexpected, unforecasted liquidity developments
- Calibration of the **required reserves with averaging**
 - Banks use reserves averaging to smooth their liquidity during the reserves maintenance period
 - Calibrate required reserves to absorb the expected forecasting errors
- Model-based approach to **calibrate the different monetary policy instruments**
 - Short-term: steer the domestic rate and fine-tune
 - Stable: long term instruments, market and yield curve development
 - Buffer: required reserves

Why Accurate Liquidity Forecasting is Important?

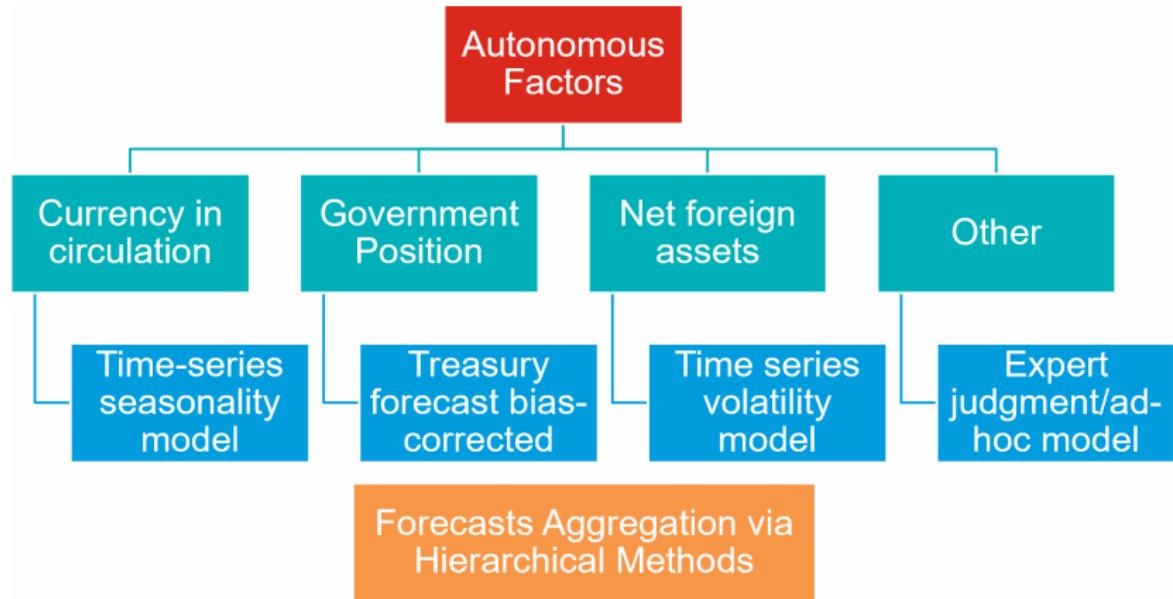
- Calibrate central bank operations accurately
 - ▶ Planify structural operations, calibrate the monetary policy instruments
- Mitigate money markets volatility and strengthen financial stability
- Improve monetary policy transmission
- Support market development

Autonomous Factors Drivers



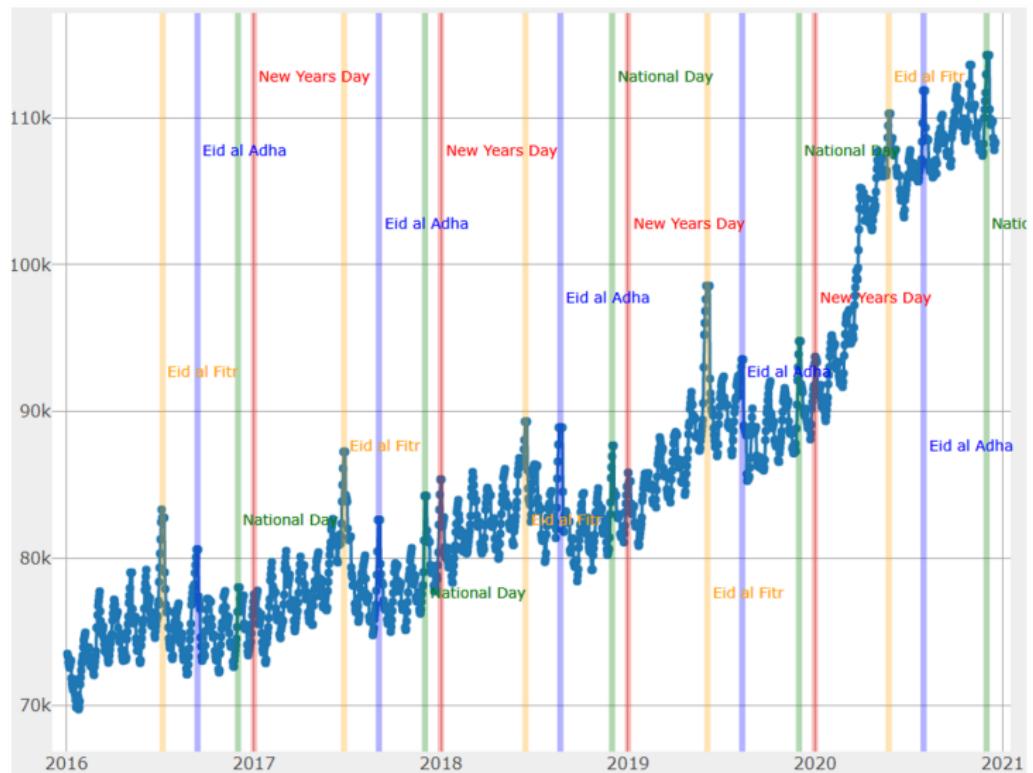
Credit: Author

Use Specific Models for Each Autonomous Factor



Credit: Author

Seasonality in Practice: Currency in Circulation



Credit: IMF TA mission

Models Family

Seasonality

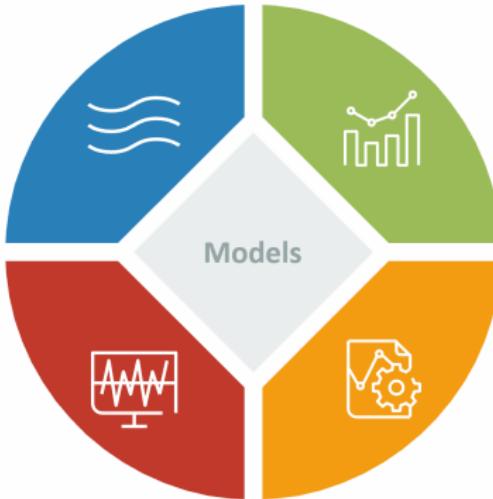


- Use regression based methods
- Capture regular and irregular seasonality
- With new parsimonious approaches

Volatility



- Deal with complex data
- Model volatility as a separate process



Smoothing

- Memory effect
- Adjust quickly to structural changes



Combination

- Combine multiple approaches
- State space, regression & smoothing
- Can handle more complex patterns

Credit: Author

Statistical Tradeoffs

Data-driven decision



Credit: Author

Models Selection

Leverage on Models Diversity to Reduce Modeling Risks



SINGLE MODELS

- Test each of them
- Ranked by performance



DYNAMIC SELECTION

- Take the most recent best model
- Adjust quickly to structural changes

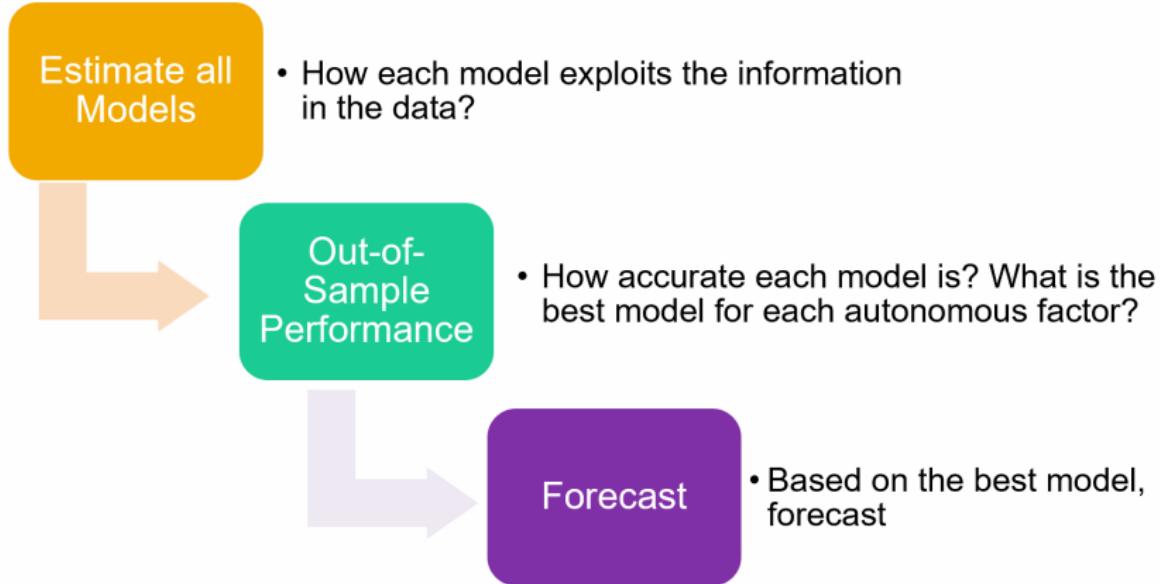


MODEL AVERAGING

- Average the top 3 models
- Reduce modeling risk

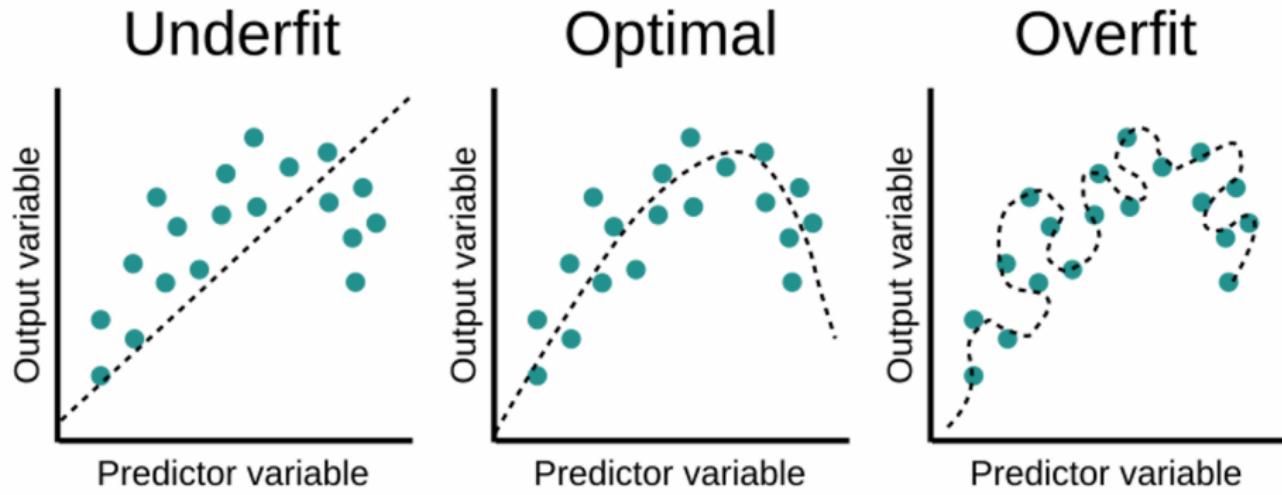
Credit: Author

Empirical Strategy



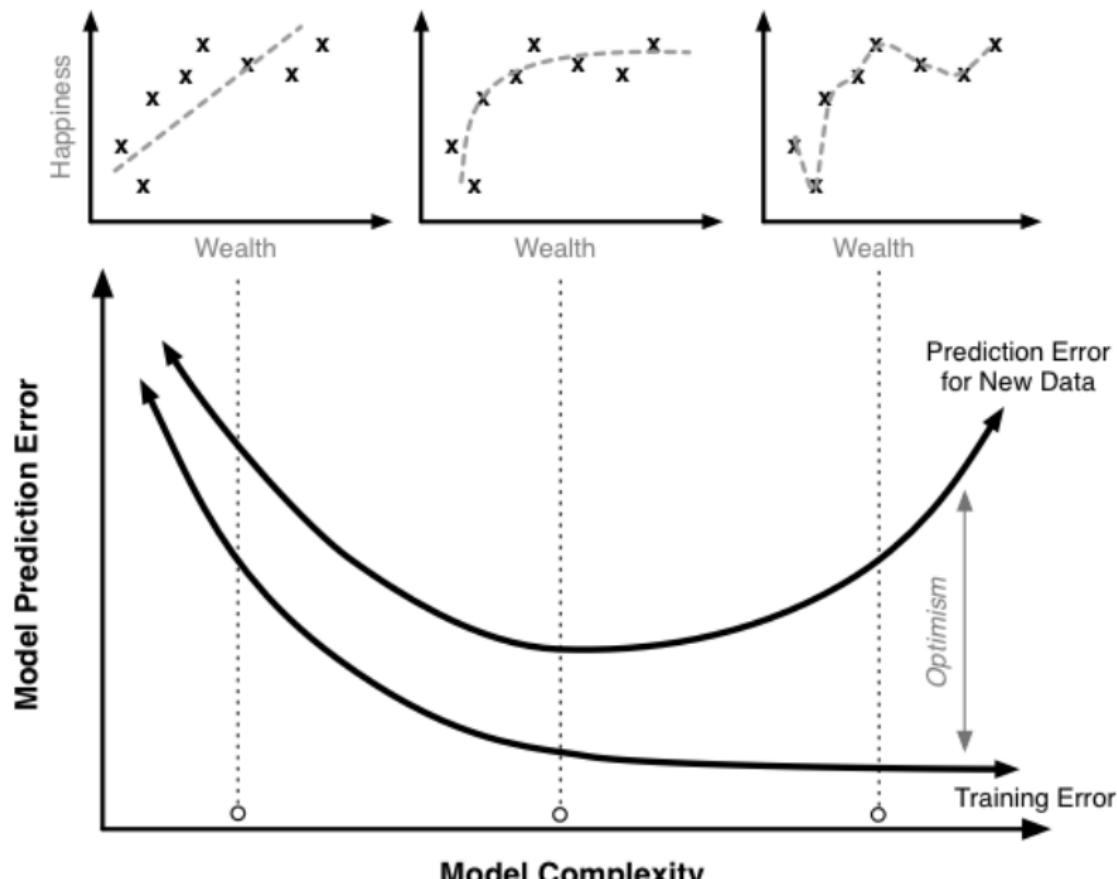
Credit: *Author*

Underfit, Optimal, Overfit: Intuition

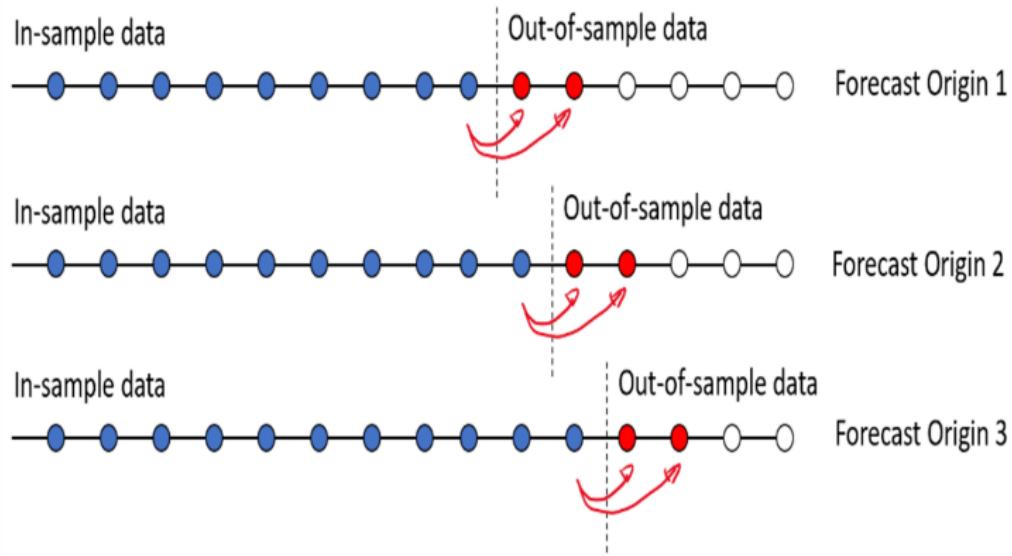


Credit: *towardsdatascience*

Underfit, Optimal, Overfit and Model Complexity

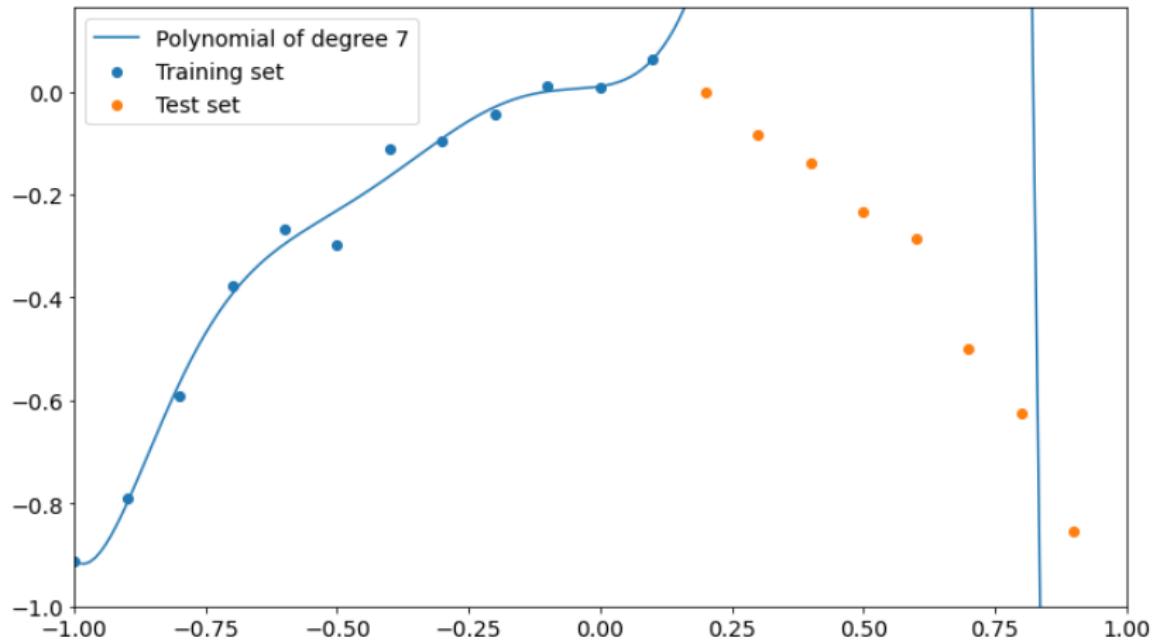


Out of Sample Concept



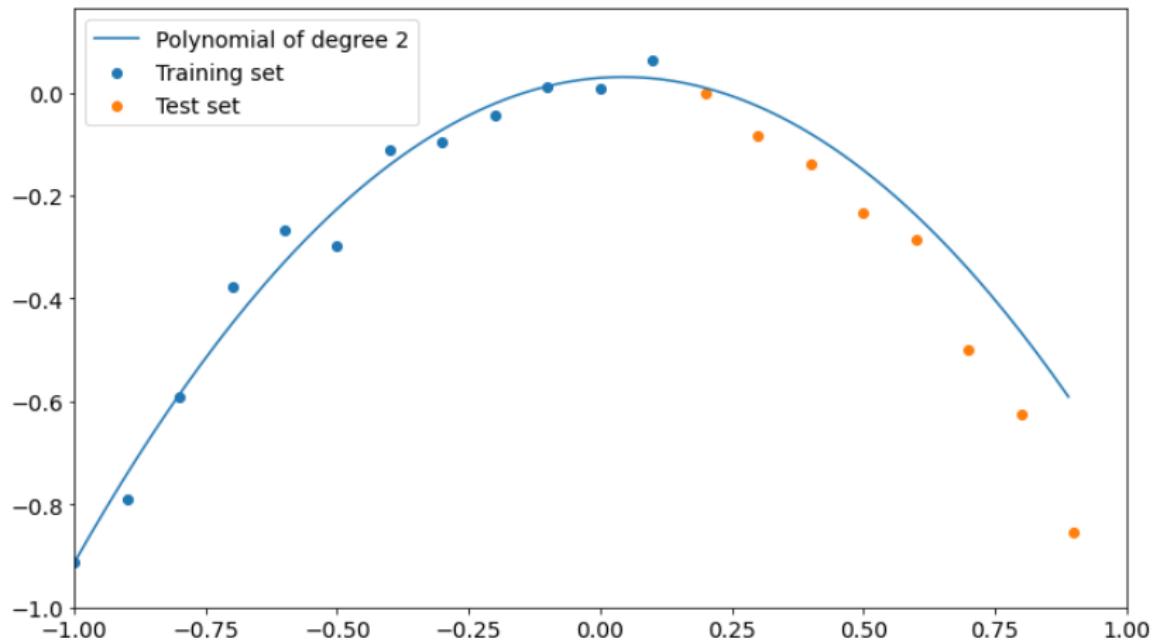
Credit: Author

Out of Sample Example: Overfit



Credit: towardsdatascience.com/an-example-of-overfitting-and-how-to-avoid-it

Out of Sample Example: Correct Fit



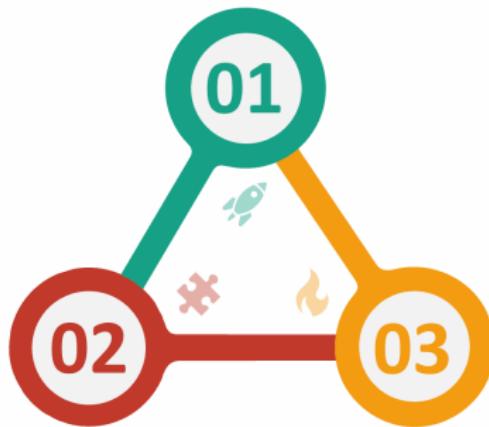
Credit: towardsdatascience.com/an-example-of-overfitting-and-how-to-avoid-it

Operational Procedure

Operational Procedure

Forecast

- Every day, forecast the autonomous factors & total liquidity projections
- Execute in a few seconds
- Warnings for outdated validation



Data Feed

- Every day update the data
- Use an Excel template

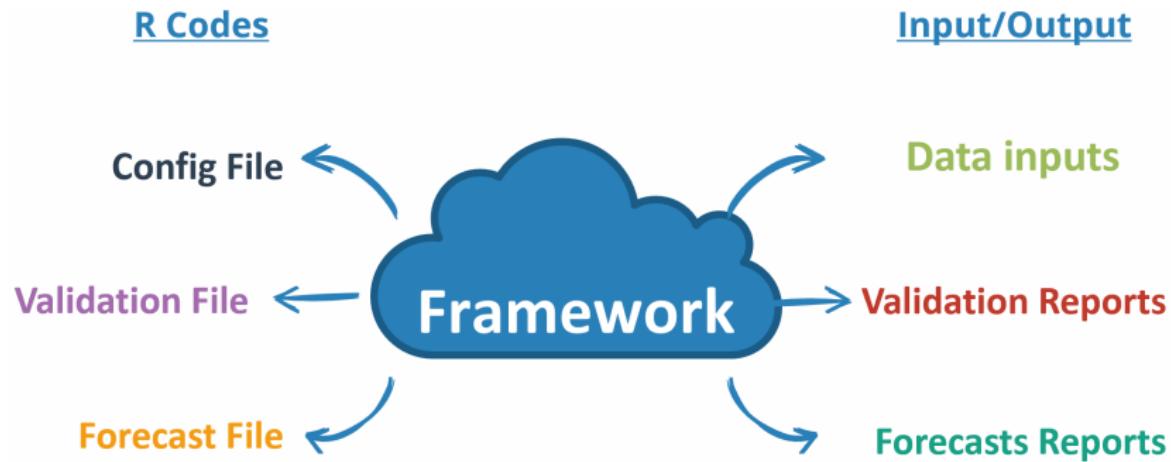


Validation

- Once every quarter, or after structural break
- Investigate models performance
- Can take a few hours
- Update config file

Credit: Author

Software Infrastructure



Credit: Author

Beyond Models: Institutional Arrangements to Improve Liquidity Forecasts

- The Treasury should anticipate the government position based on the government expected cash flows and debt issuance schedule
- Ease liquidity management and forecasting via Treasury Single Account
- Memorandum of Understanding to share the information between the Treasury and the central bank
- Gather information from important participants (liquidity contact group) including:
 - Large companies with FX transactions (e.g. commodities importers/exporters)
 - Sovereign wealth fund, large money market funds, etc.

Forecasts Publication and Transparency

- Once the forecasting model is well functioning and accurate, the central bank should publish its liquidity forecasts
 - Help market participants to manage their liquidity
- The central bank should also publish its forecasting methodology
 - Share technical capacity with the market
 - Increase transparency
 - Reinforce central bank credibility

Summary and Contributions

- ① Forecast the three main autonomous factors with advanced models
- ② Combine the autonomous factors forecasts into a single liquidity measure
- ③ Incorporate risk into liquidity forecasts
- ④ Use a robust modeling framework to deal with structural breaks and complex data samples
- ⑤ Automatic data reporting and data visualization
- ⑥ User-friendly interface based on an open-source and free statistical software (R)
- ⑦ Community of experts, working on public codes (Github)

Main References

- Cabrero, Alberto, Gonzalo Camba-Mendez, Astrid Hirsh, and Fernando Nieto. 2002. "Modeling the Daily Banknotes in Circulation in the Context of Liquidity Management of the European Central Bank," European Central Bank Working Paper No. 142, May.
- Gray, Simon T. 2008. "Liquidity Forecasting," Bank of England, Centre for Central Banking Studies Handbook—No. 27, June.
- Andrea Shaechter, 2000 "Liquidity Forecasting", International Monetary Fund
- Veyrune, Romain, Shaoyu Guo. 2019. "Autonomous Factor Forecast Quality: The Case of the Eurosystem", IMF Working paper No. 19/296, December.

- Anderson, D., & Burnham, K. (2004). Model selection and multi-model inference. 2nd ed. NY: Springer-Verlag.
- Athanasiopoulos, G., Hyndman, R. J., Kourentzes, N., & Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1), 60-74.
- De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American statistical association*, 106(496), 1513-1527.
- Wickramasuriya, S. L., Athanasiopoulos, G., & Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526), 804-819.

Table of Contents

1 The IMF Forecasting Framework

2 Q & A

3 ETS Model

4 ARIMA Models

- Modelling and Forecasting with ARIMA Models

5 Seasonal ARIMA (SARIMA)

- ARIMA versus ETS

Question 1

What are the benefits of accurate liquidity forecasting for central banks operations? Multiple answers possible:

- ① Reduce inflation
- ② Better calibrate the operations
- ③ Reduce the interbank interest rate volatility
- ④ Stabilize the exchange rate
- ⑤ Support the interbank market development

Question 2

Which strategy is valid to reduce the risk of being reliant of a single model (modeling risks)? Multiple answers possible:

- ① Model averaging
- ② Use large data sample
- ③ Remove outliers
- ④ Dynamic model selection
- ⑤ Complement the results with expert judgment

Question 3

Why is it more appropriate to evaluate the performance of models out-of-sample rather than in-sample? Multiple answers possible:

- ① Reduce the impact of outliers
- ② Avoid structural breaks
- ③ Avoid overfitting
- ④ Test forecasting in real conditions

Question 4

Under which circumstances should you revalidate the forecasting models? Multiple answers possible:

- ① A few weeks after a structural break
- ② A long time (e.g., a quarter) after the last validation
- ③ After you observe a loss in forecasting accuracy

Table of Contents

1 The IMF Forecasting Framework

2 Q & A

3 ETS Model

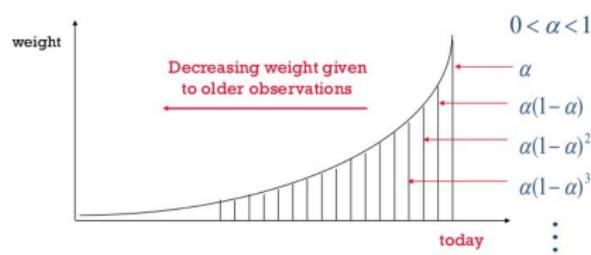
4 ARIMA Models

- Modelling and Forecasting with ARIMA Models

5 Seasonal ARIMA (SARIMA)

- ARIMA versus ETS

Exponential Smoothing Patterns



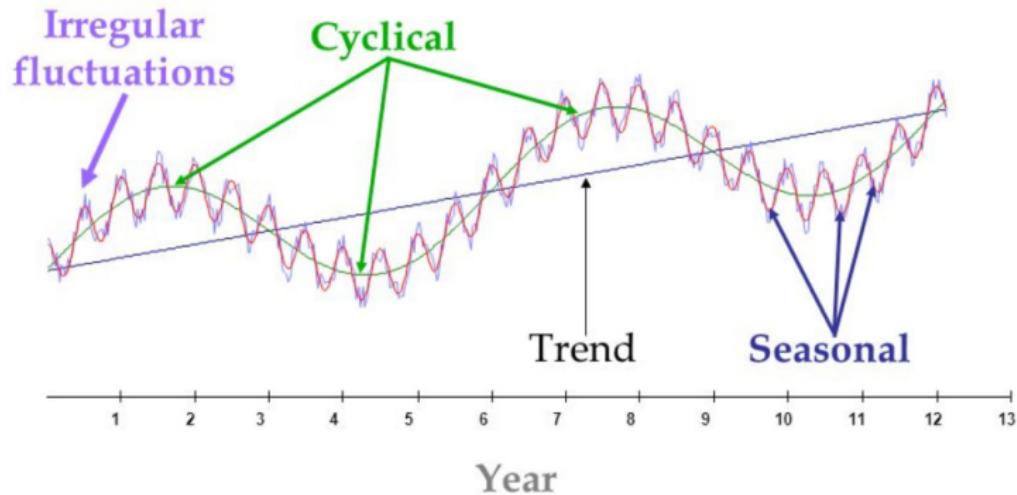
	Nonseasonal	Additive Seasonality	Multiplicative Seasonality
Constant Level			
Linear Trend			
Exponential Trend			
Damped Trend			

Credit: Medium

Perspective

- ETS (Error, Trend, Season) model was developed in the 1950s as algorithms to produce point forecasts
- ETS combines a "level" (l_{t-1}), a "trend" ((b_{t-1})) and a "seasonal" (s_{t-m}) components to describe a time series
- The combination $f(l_{t-1}, b_{t-1}, s_{t-m})$ can be additive, multiplicative, etc.
- The rate of change of the components are controlled by "smoothing" parameters: α for the level, β for the trend, γ for the seasonal
- The researcher has to:
 - ➊ To choose the best values for the smoothing parameters
 - ➋ The initial state of the parameters
- Equivalent ETS state-space models have been developed in the 1990s and the 2000s

Decomposing A Time-Series



Credit: medium.com

Combining Level, Trend and Seasonal Components

- **Additively:** $y_t = l_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t$
- **Multiplicatively:** $y_t = l_{t-1} \times b_{t-1} \times s_{t-m} \times (1 + \epsilon_t)$
- **Mixed:** $y_t = (l_{t-1} + b_{t-1}) \times s_{t-m} + \epsilon_t$

Notations:

- **Error** can be additive ("A") or multiplicative ("M")
- **Trend** can be None ("N"), additive ("A"), multiplicative ("M") or damped ("Ad" or "Md")
- Seasonality can be None ("N"), additive ("A") or multiplicative ("M")

Point forecasts and forecast distribution

- Models generates point forecasts estimates, based on a given conditional value $y_{t+1|y_0} = f_y(y_0)$
- However, need to generate the forecast distribution to assess the quality of models and select the best. **Model selection**
- A stochastic data generating process (DGP) can generate an entire forecast distribution
- Core idea: the residual (ϵ) is the only stochastic (=random) element in $y_t = l_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t$
 - ▶ Therefore, the distribution of the residuals will determine the distribution of the estimator

Level-Only Model ETS(A, N, N) (Simple Smoothing)

Component Form

- Forecast equation: $\hat{y}_{t+h|t} = l_t$
 - Smoothing equation: $l_t = \alpha y_t + (1 - \alpha)l_{t-1}$
-
- l_t is the level ("smoothed value") of the series at time t
 - $\hat{y}_{t+1|t} = \alpha y_t + (1 - \alpha)\hat{y}_{t|t-1}$

Iterate to get exponentially weighted moving average form:

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1 - \alpha)^j y_{T-j} + (1 - \alpha)^T l_0$$

Error Correction Form

Component Form

- Forecast equation: $\hat{y}_{t+h|t} = l_t$
- Smoothing equation: $l_t = \alpha y_t + (1 - \alpha)l_{t-1}$

Forecast error: $e_t = y_t - \hat{y}_{t|t-1} = y_t - l_{t-1}$

Error Correction Form

- $y_t = l_{t-1} + e_t$
- $l_t = l_{t-1} + \alpha * \underbrace{(y_t - l_{t-1})}_{e_t}$
- Intuition: α updates the next-period estimate based on the forecasting error
- Specify probability distribution for e_t , often assumed that $e_t = \epsilon_t \sim \mathcal{N}(0, \sigma^2)$

- Need to choose the best values for α and l_0
- Similarly to regression, choose optimal parameters by minimizing SSE:

$$\text{SSE} = \sum_{t=1}^T (y_t - \hat{y}_{t|t-1})^2$$

- Unlike regression, there is no closed form solution: need to use numerical optimization

State-Space Representation for Additive Models

State-Space Representation

- **Measurement equation:** $y_t = l_{t-1} + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$
 - ▶ The measurement equation is the relationship between observations (y_t) and state (=structure) l_t
- **State equation** $l_t = l_{t-1} + \alpha * \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$
 - ▶ The state equation is evolution of the state variable (l_t) through time
- Both equations have the same error process ϵ_t

State-Space Representation for Multiplicative Models

- Instead of differential errors, specify relative errors: $\eta_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}}$
- Some easy algebra, substituting $\hat{y}_{t|t-1} = l_{t-1}$ gives

State-Space Representation

- **Measurement equation:** $y_t = l_{t-1}(1 + \epsilon_t)$
- **State equation:** $l_t = l_{t-1}(1 + \alpha * \epsilon_t)$

Implication: Models with addititative and multiplicative errors with the same parameters generate **the same point forecasts but with different prediction intervals**

Holt's Linear Trend

Component Form

- Level: $l_t = \alpha_t + (1 - \alpha)(l_{t-1} + b_{t-1})$
 - Trend: $b_t = \beta * (l_t - l_{t-1}) + (1 - \beta)b_{t-1}$
 - Forecast: $\hat{y}_{t+h|t} = l_t + h b_t$
-
- Two smoothing parameters: α and β ($0 \leq \alpha, \beta \leq 1$)
 - l_t level: weighted average between y_t and one-step ahead forecast for time t : $l_{t-1} + b_{t-1} = \hat{y}_{t|t-1}$
 - b_t slope: weighted average of $(l_t - l_{t-1})$ and b_{t-1} , current and previous estimate of slope
 - Choose α, β, l_0, b_0 to minimize the SSE (sum squared errors)

Damped Trend Method

Component Form

- $l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1})$
- $b_t = \beta * (l_t - l_{t-1}) + (1 - \beta)\phi b_{t-1}$
- $\hat{y}_{t+h|t} = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$

- ϕ is the damping parameter, $0 \leq \phi \leq 1$
- If $\phi = 1$, the method boils down to Holt's linear trend
- As $h \rightarrow \infty$, then $\hat{y}_{T+h|T} \rightarrow l_T + \frac{\phi b_T}{1-\phi}$
- Application: short-run forecasts are trended, long-run forecasts constant

Holt-Winters Seasonal Model

Holt and Winters extended Holt's method to capture seasonality

Component Form

- Level: $l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$
- Trend: $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$
- Season: $\gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$
- Forecast: $\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)}$
- m is the period of seasonality (e.g. $m = 4$ for quarterly data)

Seasonal Component

- The seasonal component is usually expressed as:

$$s_t = \gamma * (y_t - l_t) + (1 - \gamma) * s_{t-m} \quad 0 \leq \gamma \leq 1$$

- By substitution, we can derive the dynamic of the seasonal term as:

$$s_t = \gamma * (1 - \alpha)(y_t - l_{t-1} - b_{t-1}) + [1 - \gamma(1 - \alpha)]s_{t-m}$$

Forecasting with ETS models

Traditional point forecasts: iterate the equations for $t = T + 1, T + 2, \dots, T + h$. By construction, $\epsilon_t = 0 \forall t > T$

- Equals to $E[y_{t+h}|x_t]$ in the case of additive seasonality only
- Point forecasts for additive ETS are the same as for multiplicative ETS if the parameters are the same

Prediction Intervals

- They can only be generated using the models
- The prediction intervals will differ between models with additive and multiplicative errors
- Some simple ETS models offer exact formula
- For more complex ETS models, the only solution for generating the confidence intervals is by bootstrapping
 - Simulate future sample paths, conditional on the last estimates of the states
 - Obtain the prediction intervals from the percentiles of these simulated future paths

Main Idea: Control the Rate of Change

- α controls the flexibility of the **level**
 - If $\alpha = 0$, the level never updates (stays at the mean)
 - If $\alpha = 1$, the level updates completely (naive, start from yesterday)
- β controls the flexibility of the **trend**
- If $\beta = 0$, the trend is linear
- If $\beta = 1$, the trend changes suddenly at each observation
- γ controls the flexibility of the **seasonality**
 - If $\gamma = 0$ the seasonality is fixed (seasonal mean)
 - If $\gamma = 1$ the seasonality updates completely (seasonal naive)

Stability and Forecastability

- Stability and forecastability have deep mathematical implications that I will only explain intuitively
 - **Stability:** the weights of the observations decay over time, guaranteeing that the newer ones will have higher weights than old one.
 - ★ *This is the core principle behind exponential weights: the model captures information update as time goes through.*
 - **Forecastability:** The forecastability does not guarantee that the weights decay, but it guarantees that the initial value of the state vector will have a constant impact on forecasts, i.e. will not increase in weight with the increase of the forecast horizon.
 - ★ *Forecastability is a variation around the concept of ergodicity: it implies that, as the model embeds new observations, the impact of "old information" from the old observations does not "pollute" the new information brought by the new observations. In other words, the model is "learning" relevant information*
- More information: <https://otexts.com/fpp3/>

Table of Contents

1 The IMF Forecasting Framework

2 Q & A

3 ETS Model

4 ARIMA Models

- Modelling and Forecasting with ARIMA Models

5 Seasonal ARIMA (SARIMA)

- ARIMA versus ETS

ARIMA Model

- AR: autoregressive (lagged observations as inputs)
- I: integrated (differencing to make series stationary)
- MA: moving average (lagged errors as inputs)

Intuition

Contrary to an ETS, an ARIMA model is rarely interpretable in terms of visible data structures like trend and seasonality. But it can capture a huge range of time series patterns

Refresher: Intuitive Definition of Stationarity

Intuitive Characterization

If y_t is a stationary time series, then for any period s in the future, the distribution $\{y_t, \dots, y_{t+s}\}$ doesn't depend on t

A **stationary series** is:

- Roughly horizontal
- Constant variance
- No predictable patterns in the long term

Stabilization:

- Transformations help to **stabilize the variance**
- For ARIMA modelling, we also need to **stabilize the mean**

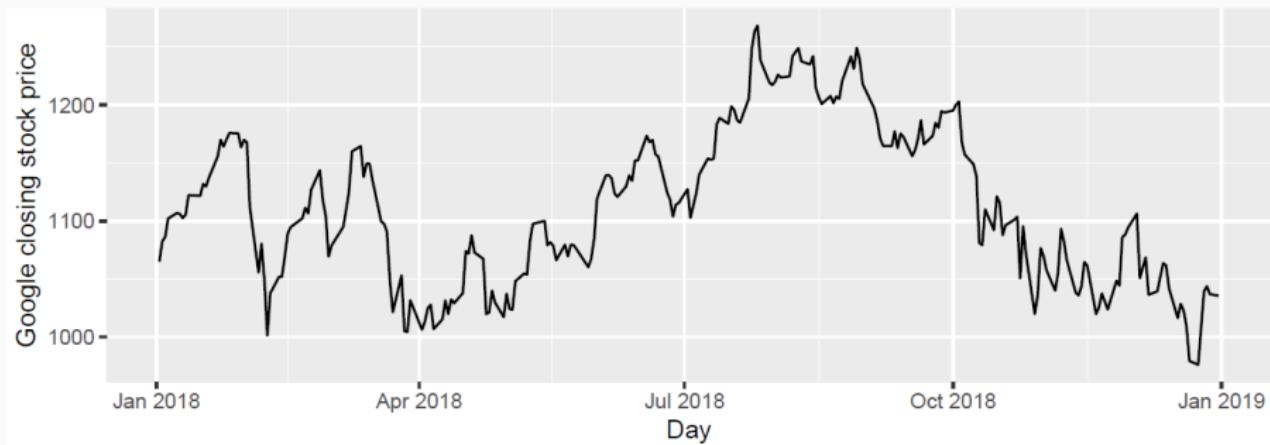
Identifying Non-Stationary Series

Tips:

- Time plot
- The ACF of stationary data drops to zero relatively quickly
- The ACF of non-stationary data decreases slowly
- For non-stationary data, the value of the first coefficient is often large and positive

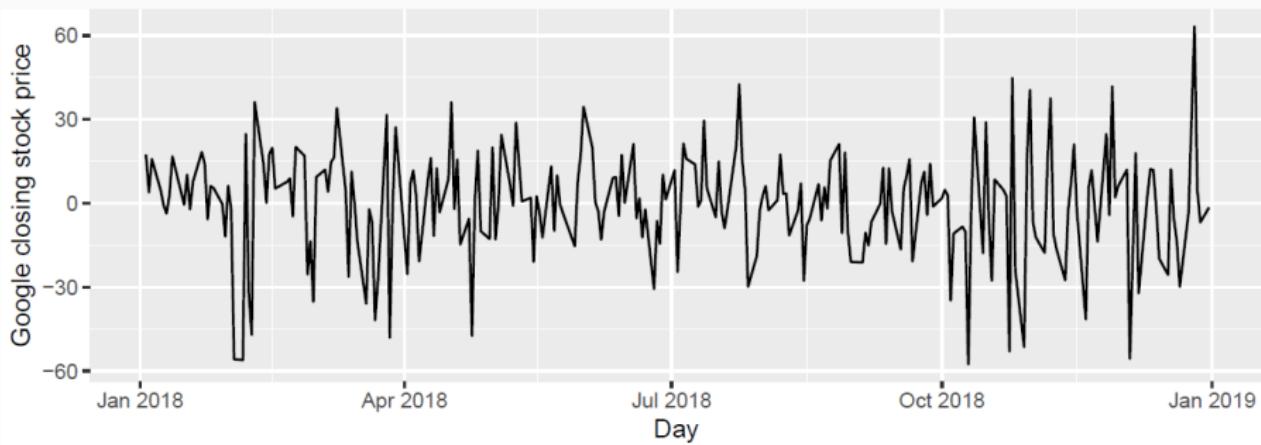
Stationary?

```
gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) == 2018) %>%
  autoplot(Close) +
  labs(y = "Google closing stock price", x = "Day")
```

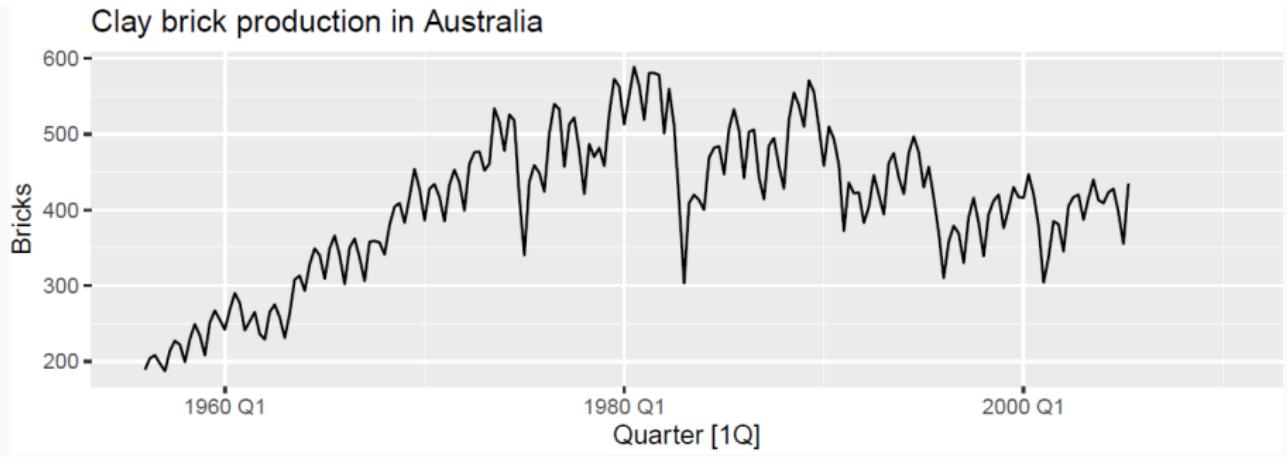


Stationary?

```
gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) == 2018) %>%
  autoplot(difference(Close)) +
  labs(y = "Google closing stock price", x = "Day")
```



Stationary?



Differencing

- Differencing helps to **stabilize the mean**
- The differenced series is the *change* (or first difference) between each observation in the original series: $y'_t = y_t - y_{t-1}$
- The differenced series will have only $T - 1$ values since it is not possible to calculate a difference y'_1 for the first observation

Random Walk Model

If the differenced series is white noise with zero mean:

Specification

$$y_t - y_{t-1} = \epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- Very widely used for non-stationary data
- This is the model behind the **naive method**
- Random walks typically have:
 - Long periods of apparent trends up or down
 - Sudden/unpredictable changes in direction
- In a random walk, the forecasts are equal to the last observation
 - Future movements up or down are equally likely

Random Walk with Drift Model

If the differenced series has a drift c :

$$y_t - y_{t-1} = c + \epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- c is the **average change** between consecutive observations
- If $c > 0$, y_t will tend to drift upwards and vice-versa
- This is the model behind the **drift method**

Second-Order Differencing

Occasionally, the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$y_t'' = y_t' - y_{t-1}' = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

- y_t'' will have $T - 2$ values
- In practice, it is almost never necessary to go beyond second-order differences

Seasonal Differencing

Definition: Seasonal Difference

A seasonal difference is the difference between an observation and the corresponding observation from the previous year

$$y'_t = y_t - y_{t-m}$$

where m = number of seasons

- For monthly data, $m = 12$
- For quarterly data, $m = 4$

Differencing in Practice

When both seasonal and first differences are applied:

- It makes no difference which one is done first - the result will be the same
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference
- It is important that, if differencing is used, the differences are **interpretable**: for instance, taking lag 3 differences for yearly data is difficult to interpret

Unit Root Tests

Statistical tests can be used to determine the required order of differencing

- ① **Augmented Dickey Fuller test:** null hypothesis is that the data is non-stationary and non-seasonal
- ② KPSS (Kwiatkowski-Phillips-Schmidt Shin) test: the null hypothesis is that the data is stationary is non-seasonal
- ③ Other tests are available for seasonal data

Backshift Notation

Notation

The backshift notational device, B is used as follows:

$$By_t = y_{t-1}$$

- B operating on y_t has the effect of **shifting the data back one period**
- Two applications of B to y_t shifts the data back **two periods**

$$B(By_t) = B^2y_t = y_{t-2}$$

B depends on the period/frequency considered. Shifting monthly data by a year supposes using B^{12}

Relationship with Differencing

Importantly, the backshift operator is convenient for describing differencing

Backshift Operator and Differencing

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t$$

- Likewise, second-order differences are obtained with:
 $y''_t = (1 - B)^2 y_t$
- Pay attention !! Second-order difference is not second difference
 - Second order difference: $(1 - B)^2 y_t = y''_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$
 - Second difference: $1 - B^2 y_t = y_t - y_{t-2}$

Combined Effects

Assume that you want to combine a first difference with a seasonal difference:

- First difference: $(1 - B)$
- Seasonal difference: $(1 - B^m)$

Then, the backshift operator can be directly combined, as a polynomial, to represent the transformed time series:

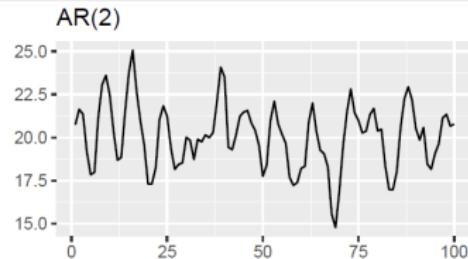
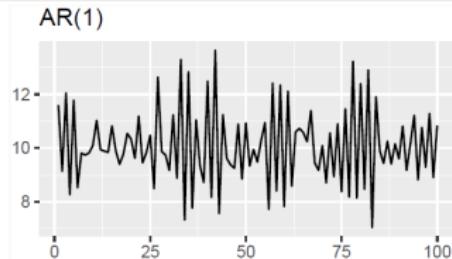
$$(1 - B)(1 - B^m)y_t = (1 - B - B^m + B^{m+1})y_t = y_t - y_{t-1} - y_{t-m} + y_{t-m-1}$$

Autoregressive (AR) Models

Definition

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

- where ϵ_t is a white noise
- This is a multiple regression with **lagged variables**



AR(1) Model

Specification

$$y_t = c + \phi_1 y_{t-1} + \epsilon_t$$

- When $\phi_1 = 0$, y_t is equivalent to a **white noise**
- When $\phi_1 = 1$ and $c = 0$, y_t is equivalent to a **random walk**
- When $\phi_1 = 1$ and $c \neq 0$, y_t is equivalent to a **random walk with drift**
- When $\phi_1 < 0$ y_t tends to oscillate between positive and negative values

Stationarity Conditions

To restrict AR models to stationary data, some constraints on the coefficients are needed

General Condition for Stationarity

Complex roots of the polynomial $\mathcal{P}(z) = 1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$ lie outside the unit circle of the complex plane

- Intuition: For an AR(1) model, the backshift polynomial is
 $y_t = \phi_1 y_{t-1} + \epsilon_t \leftrightarrow y_t(1 - \phi_1 B) = \epsilon_t$
- $(1 - \phi_1 B) = 0 \leftrightarrow B = \underbrace{\frac{1}{\phi_1}}$
Not explosive
- To get the AR expression not explosive, we need $|\frac{1}{\phi_1}| < 1$ and therefore $|\phi_1| > 1$

The Case for Low Lags Orders

For low lags orders, the stationarity conditions are simply:

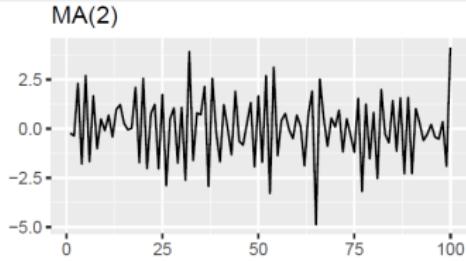
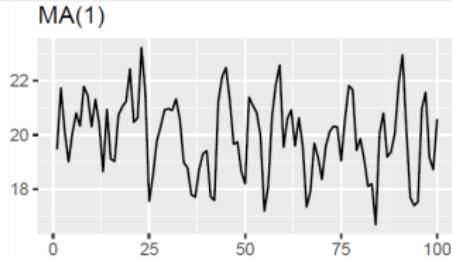
- For $p = 1$: $-1 < \phi_1 < 1$
- For $p = 2$: $-1 < \phi_2 < 1$, $\phi_1 + \phi_2 < 1$ and $\phi_2 - \phi_1 < 1$
- More complex conditions hold for $p \geq 3$
- Estimation software (R, Python, Eviews, etc.) takes care of this

Moving Average Model

Definition: Moving Average Model

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

- ϵ_t is a white noise
- This is a multiple regression with **past errors** as predictors
- Do NOT confuse this with *moving average smoothing!*



Wold Decomposition: From AR(p) to MA(∞) Model

Wold Decomposition

It is possible to write any **stationary** AR(p) model as an MA(∞)

- Intuitive: just go backward!

- $y_t = \phi_1 \underbrace{y_{t-1}} + \epsilon_t$

- $y_t = \phi_1(\phi_1 y_{t-1} + \epsilon_{t-1}) + \epsilon_t = \phi_1^2 y_{t-2} + \phi_1 \epsilon_{t-1} + \epsilon_t$

- ...

- Providing that $1 < |\phi_1| < 1$:

$$y_t = \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_1^2 \epsilon_{t-2} + \phi_1^3 \epsilon_{t-3} + \dots$$

Invertibility: From MA(q) to AR(∞)

- Under certain conditions, an MA(q) process can be written as an AR(∞) process
- In this case, the MA model is said to be **invertible**
- Invertible models have some mathematical properties that make them easier to use in practice
- Invertability of an MA model is equivalent to the **forecastability** of an ETS model
 - This is intuitive: AR processes are embedding new information on the most recent lags

Invertibility

General Condition for MA(q) Invertibility

Complex roots of $1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q$ lie outside the unit circle of the complex plane

- For $q = 1$: $-1 < \theta_1 < 1$
- For $q=2$:
 - $-1 < \theta_2 < 1$
 - $\theta_1 + \theta_2 > -1$ and $\theta_1 - \theta_2 < 1$
- More complicated solutions hold for $q \geq 3$
- Estimation software takes care of this

ARMA(p, q) Model

Specification

$$y_t = c + \underbrace{\phi_1 y_{t-1} + \cdots + \phi_p y_{t-p}}_{\text{AR}} + \underbrace{\theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}}_{\text{MA}} + \epsilon_t$$

- The predictors include both **lagged values of y_t** and **lagged errors**
- Important specification: the future value of the series depends both on the past values it took (dynamic), as well as recent random noise / error term
- This simple model is "learning" both from the dynamic of the past values and from its inherent randomness
- Conditions on AR coefficients ensure **stationarity**
- Conditions on MA coefficients ensure **invertibility**

Autoregressive Integrated Moving Average (ARIMA)

ARIMA: ARIMA stands for: Autoregressive **Integrated** Moving Average model

- Basically, it is a non-stationary model that can be made stationary by differencing
- $(1 - B)^d y_t$ follows an ARMA model: d is the degree of differencing
- Once differenced d times, it is stationary and behaves as an ARMA model

Generalization

- $ARIMA(p, d, q)$ where p is the autoregressive order, d the degree of differencing and q the order of the moving average part
- All linear models we discussed are special cases of the ARIMA model:
 - White noise model: $ARIMA(0, 0, 0)$
 - Random walk: $ARIMA(0, 1, 0)$ with no constant
 - Random walk with drift: $ARIMA(0, 1, 0)$ with constant
 - $AR(p) = ARIMA(p, 0, 0)$, $MA(q) = ARIMA(0, 0, q)$

Backshift Notation for ARIMA

ARIMA(1, 1, 1) model:

$$\underbrace{(1 - \phi_1 B)}_{AR(1)} \quad \underbrace{(1 - B)y_t}_{\text{First difference}} = c + \underbrace{(1 + \theta_1 B)\epsilon_t}_{MA(1)}$$

ARIMA Analysis with Egyptian Exports

Egyptian Exports

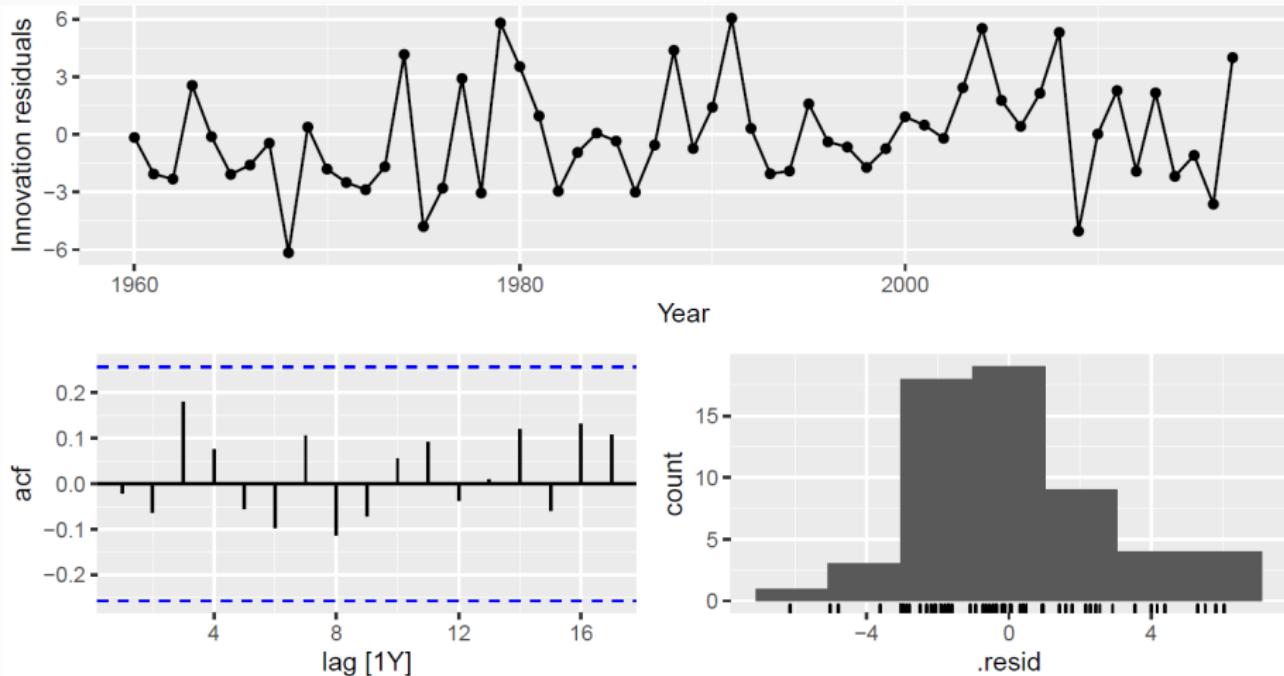


Egyptian Exports Fit

```
fit <- global_economy %>% filter(Code == "EGY") %>%
  model(ARIMA(Exports))
report(fit)

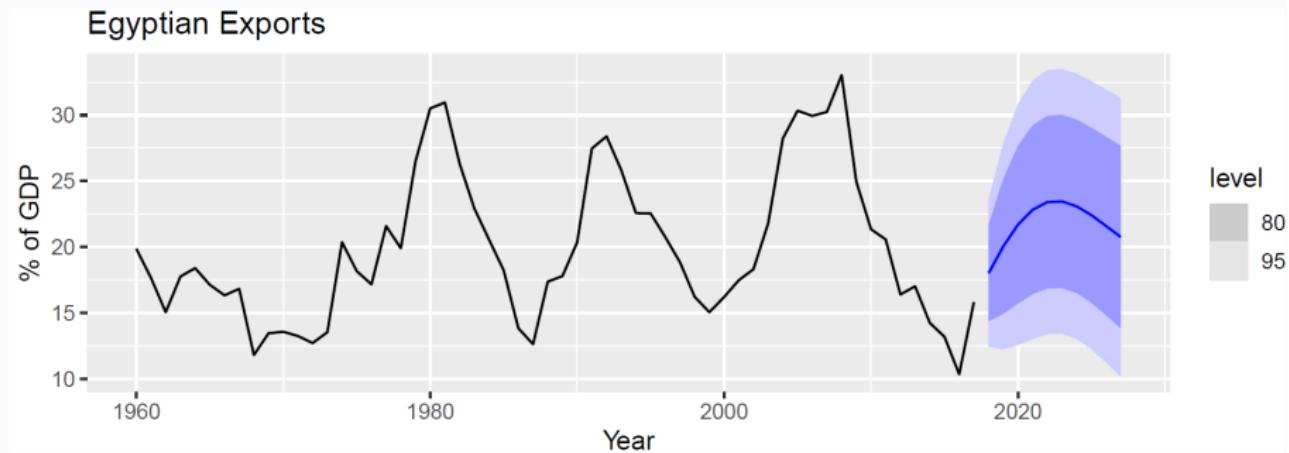
## Series: Exports
## Model: ARIMA(2,0,1) w/ mean
##
## Coefficients:
##             ar1      ar2      ma1  constant
##             1.676   -0.8034  -0.690    2.562
## s.e.  0.111    0.0928   0.149    0.116
##
## sigma^2 estimated as 8.046:  log likelihood=-142
## AIC=293    AICc=294    BIC=303
```

Egyptian Exports Fit Residuals



Egyptian Exports Forecasts

```
fit %>% forecast(h=10) %>%
  autoplot(global_economy) +
  labs(y = "% of GDP", title = "Egyptian Exports")
```



General Approach

- **Plot the data.** Identify any unusual observations
- If necessary, **transform the data** (using a Box-Cox transformation) to stabilize the variance
- If the data are non-stationary, **first-difference** it until stationarity
- **Examine the ACF/PACF:** is an $AR(p)$ or $MA(q)$ reasonable assumptions
- Try your chosen models: use AIC to compare with other models
- Plot the residuals, look at the residuals ACF. Residuals should look like a **white noise**.
- Once the residuals look like a white noise, **compute the forecasts**

Parameters Selection: Algorithm

ARIMA Specification

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\epsilon_t$$

- Need to select the appropriate order for p, q, d

Hyndman and Khandasar (2008) selection algorithm:

- Select the differencing order (=number of differences) d and D using KPSS test
- Select p, q by minimizing AICs
- Use stepwise search to traverse model space (= test different combinations of p, q for robustness)

Selection Algorithm

Aikaike Information Criteria (AIC)

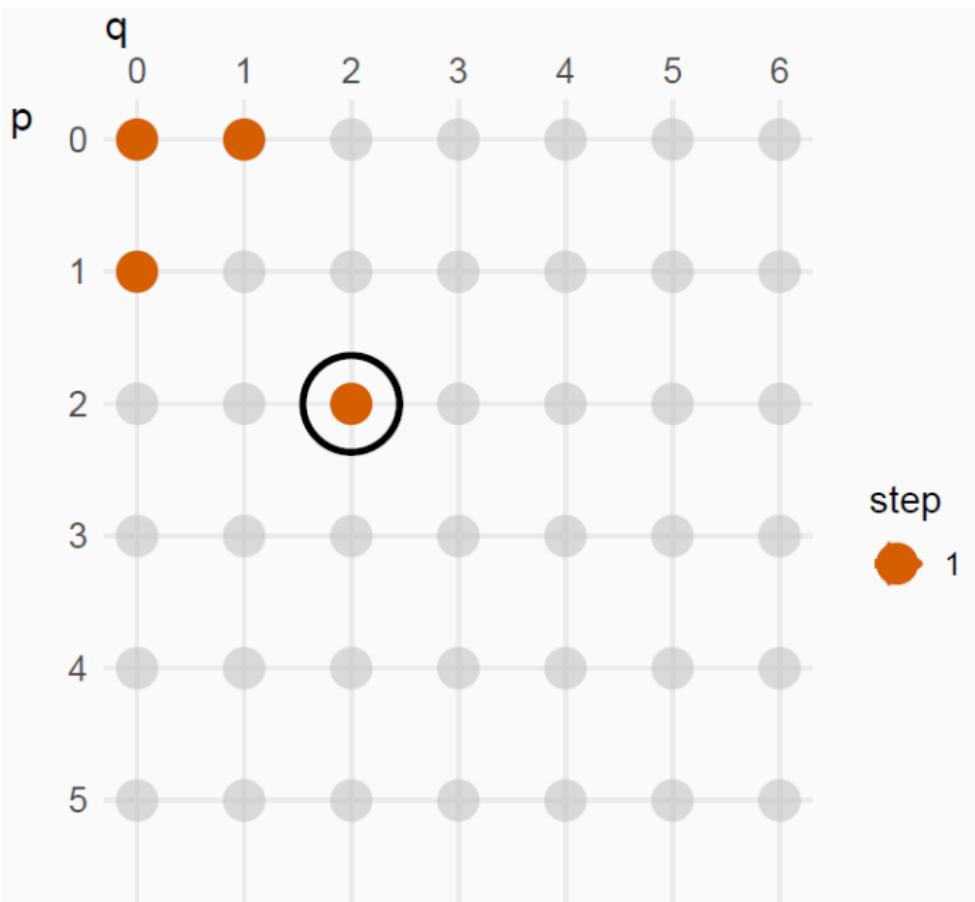
$$AIC = -2\log(L) + 2(p + q + k + 1) \left[1 + \frac{(p + q + k + 2)}{T - p - q - k - 2} \right]$$

- where L is the maximized likelihood fitted to the *differenced data*
- Degree of freedom: if the model embedds a constant c , $k = 1$, else $k = 0$

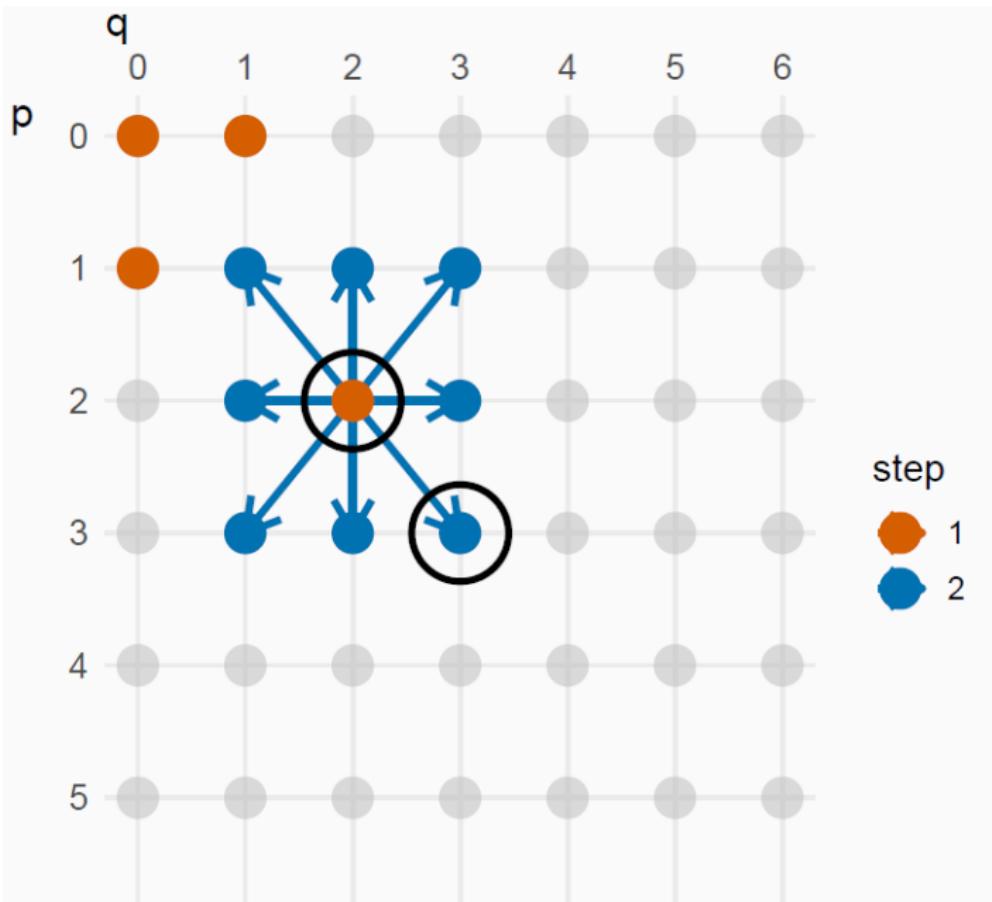
Steps:

- ① Select the current model, with the smallest AIC from the most parsimonious
 - ARIMA(0,d, 0), ARIMA(1, d, 0), ARIMA(0, d, 1), ARIMA(1, d, 1) and ARIMA(2, d, 2)
- ② Consider **variations** of the current model:
 - Vary one of p, q from current model by ± 1
 - p, q both vary from current model by ± 1
 - Include/exclude the constant c from the current model

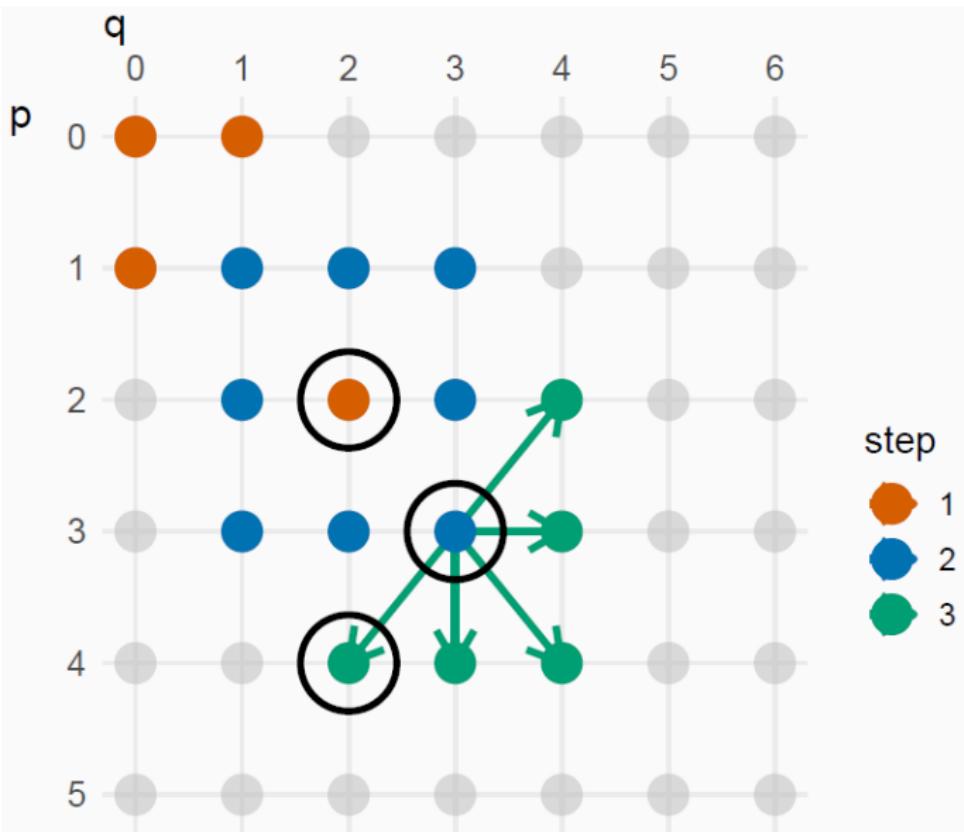
ARIMA Selection



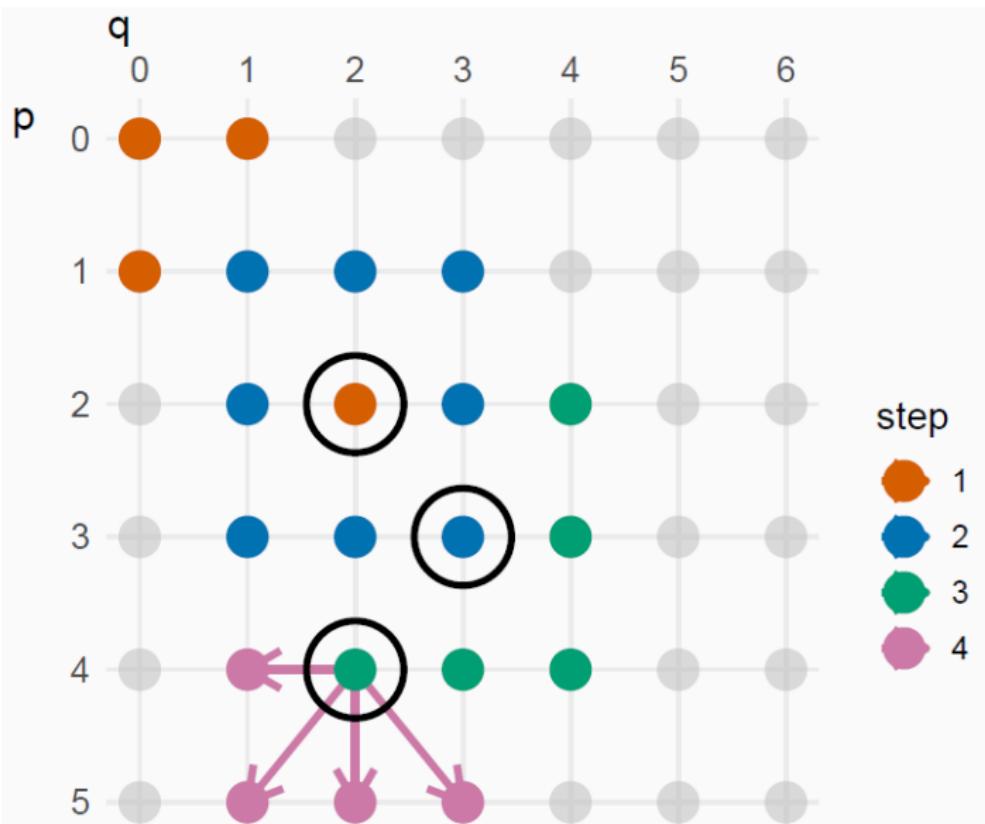
ARIMA Selection



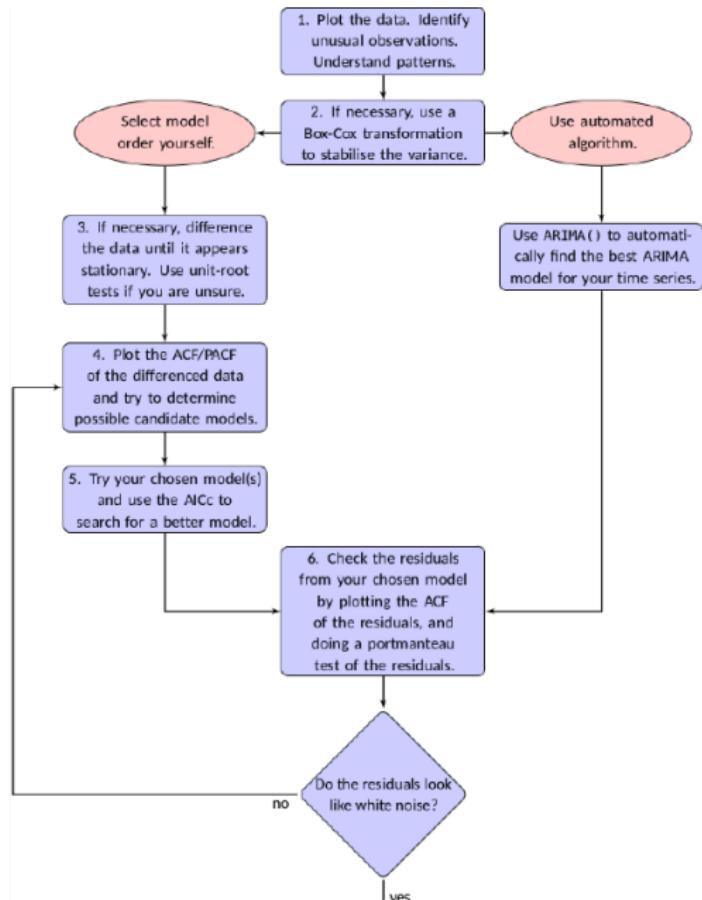
ARIMA Selection



ARIMA Selection



Modelling Process



Point Forecasts

General approach:

- Rearrange the ARIMA equation so that y_t is on the left-hand side
- Convert from polynomial backshift to companion form
- On the right hand-side, replace the future observations by their forecasts, the future errors by zeros, and the past errors by the corresponding residuals
- Start with $h = 1$; iterate with $h = 2, 3, \dots$

Example with an ARIMA(3, 1, 1): Rearrangements

Specification

$$\underbrace{(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)}_{AR(3)} \underbrace{(1 - B)}_{I(1)} y_t = \underbrace{(1 + \theta_1 B)}_{MA(1)} \epsilon_t$$

- Distribute the backshift:

$$[1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4] y_t = (1 + \theta_1 B) \epsilon_t$$

- Express as a function of y_t :

$$y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} + \phi_3 y_{t-4} = \epsilon_t + \theta_1 \epsilon_{t-1}$$

- Rearrange:

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

Example with an ARIMA(3, 1, 1): Iterative Replacement

Component Form

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

- ① Start at a given T , known. Plug T in the formula to obtain the expression of $T + 1$

$$y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \epsilon_{T+1} + \theta_1 \epsilon_T$$

- ② Problem: ϵ_{T+1} is unknown. Put it at 0 and replace ϵ_T by the fitted residuals $\hat{\epsilon}_T$

$$y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \theta_1 \hat{\epsilon}_T$$

- ③ Likewise, iterate for next periods, $T + 2, T + 3, etc.$

Prediction Intervals

95% Prediction Interval

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$$

- $v_{T+1|T} = \hat{\sigma}^2$ for all ARIMA models, regardless of parameters and orders
- Multi-step prediction interval for $ARIMA(0, 0, q)$:
 - $y_t = \epsilon_t + \sum_{i=1}^q \theta_i \epsilon_{t-i}$
 - $V_{T|T+h} = \hat{\sigma}^2 \left[\sum_{i=1}^{h-1} \theta_i^2 \right] \quad \forall h = 2, 3, \dots$

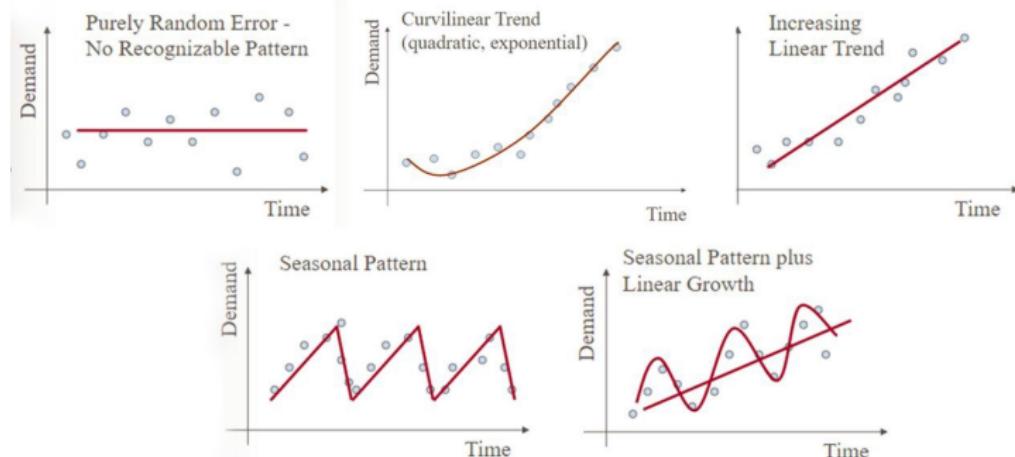
Properties of Prediction Intervals

- ❶ Prediction intervals increase in size with the forecasting horizon
- ❷ Prediction intervals can be difficult to calculate by hand
- ❸ Calculations assume that the residuals are **uncorrelated** and **normally distributed**
- ❹ Prediction intervals tend to be too narrow compared to the true uncertainty:
 - The uncertainty in the parameter estimates has not been accounted for
 - The ARIMA model assumes that the recent historical patterns won't change during the forecast period
 - The ARIMA model assumes uncorrelated future errors

Table of Contents

- 1 The IMF Forecasting Framework
- 2 Q & A
- 3 ETS Model
- 4 ARIMA Models
 - Modelling and Forecasting with ARIMA Models
- 5 Seasonal ARIMA (SARIMA)
 - ARIMA versus ETS

Non-Stationarity and Seasonal Patterns



Credit: medium.com

Seasonal ARIMA (SARIMA)

Specification

ARIMA

$$\underbrace{(p, d, q)}$$

Non-Seasonal part of the model

$$\underbrace{(P, D, Q)_m}$$

Seasonal part of the model

where m = number of observations per year

Example: $ARIMA(1, 1, 1)(1, 1, 1)_4$

Example: $ARIMA(1, 1, 1)(1, 1, 1)_4$, without constant

$$\underbrace{(1 - \phi_1 B)}_{\text{Non-Seas. AR1}} \underbrace{(1 - \Phi_1 B^4)}_{\text{Seas. AR1}} \underbrace{(1 - B)}_{\text{Seas. Diff.}} = \underbrace{(1 + \theta_1 B)}_{\text{Non-seas. MA}(1)} \underbrace{(1 + \Theta_1 B^4)}_{\text{Seasonal MA}(1)} \epsilon_t$$

All factors can be multiplied out to obtain the model's component form

US Census Canonical Models

The US Census Bureau uses the following models the most often:

- $ARIMA(0, 1, 1)(0, 1, 1)_m$ with log transformation
- $ARIMA(0, 1, 2)(0, 1, 1)_m$ with log transformation
- $ARIMA(2, 1, 0)(0, 1, 1)_m$ with log transformation
- $ARIMA(0, 2, 2)(0, 1, 1)_m$ with log transformation
- $ARIMA(2, 1, 2)(0, 1, 1)_m$ with **no** transformation

Intuition

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF

$ARIMA(0, 0, 0)(0, 0, 1)_{12}$ will show

- A spike at lag 12 in the ACF but no other significant spikes
- The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36, etc.

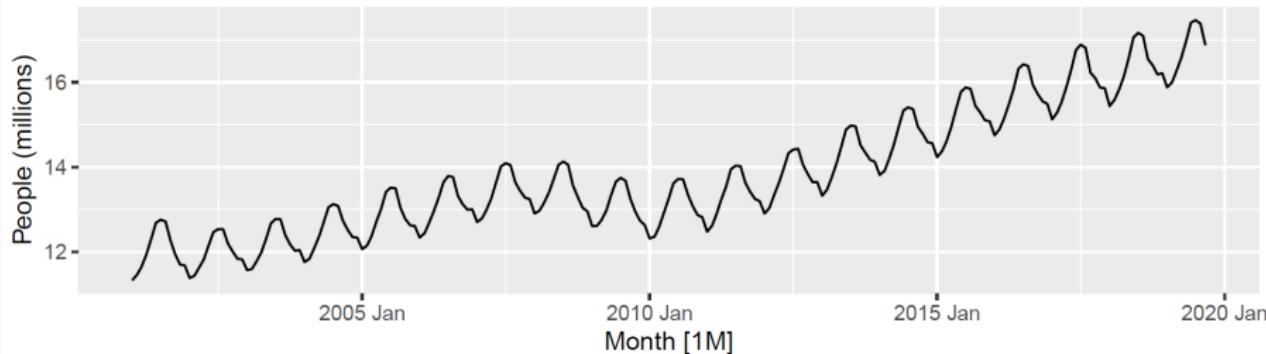
$ARIMA(0, 0, 0)(1, 0, 0)_{12}$ will show

- Exponential lags in the seasonal lags of the ACF
- A single significant spike at lag 12 in the PACF

Application: US Leisure Employment and Hospitality

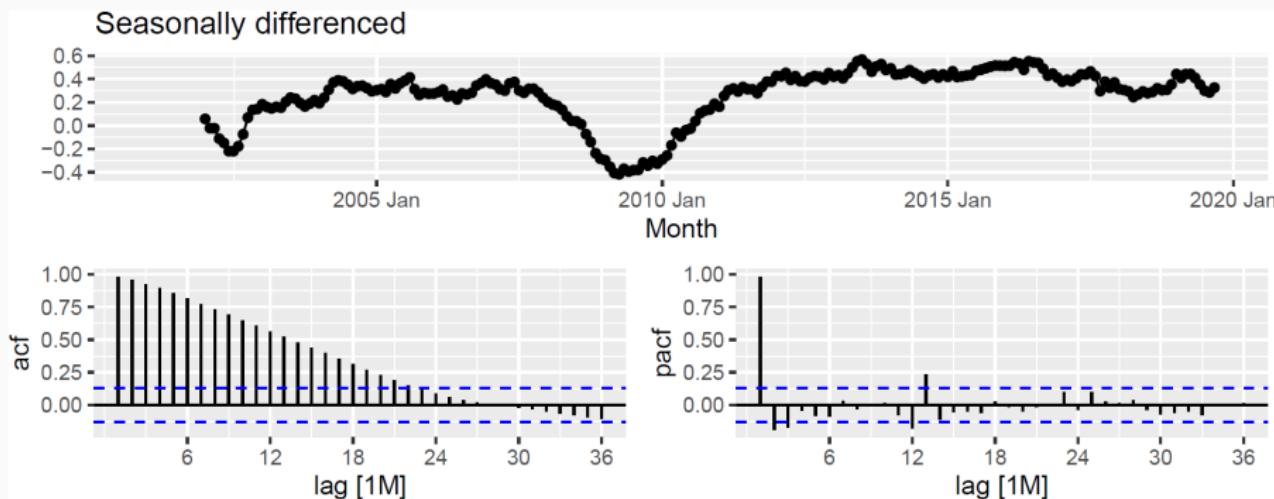
```
leisure <- us_employment %>%
  filter>Title == "Leisure and Hospitality", year(Month) > 2000) %>%
  mutate(Employed = Employed/1000) %>%
  select(Month, Employed)
autoplot(leisure, Employed) +
  labs(title = "US employment: leisure & hospitality", y="People (millions)")
```

US employment: leisure & hospitality



Application: US Leisure Employment and Hospitality

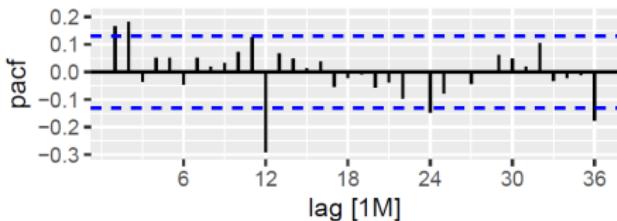
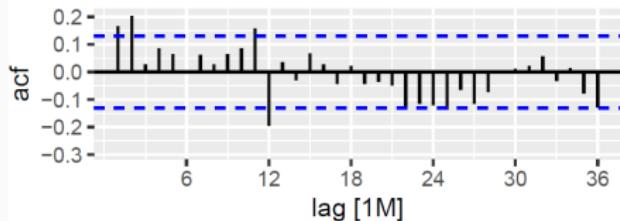
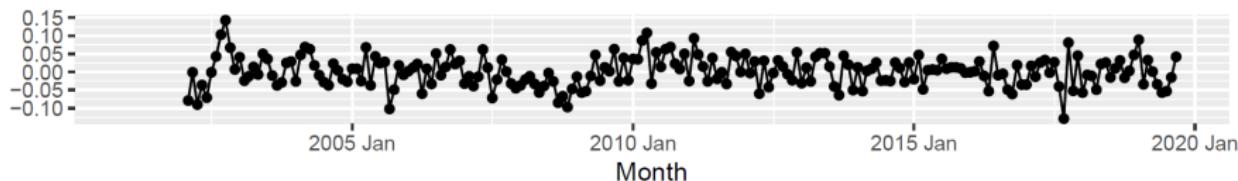
```
leisure %>%
  gg_tsdisplay(difference(Employed, 12), plot_type='partial', lag=36) +
  labs(title="Seasonally differenced", y="")
```



Application: US Leisure Employment and Hospitality

```
leisure %>%
  gg_tsdisplay(difference(Employed, 12) %>% difference(),
    plot_type='partial', lag=36) +
  labs(title = "Double differenced", y="")
```

Double differenced

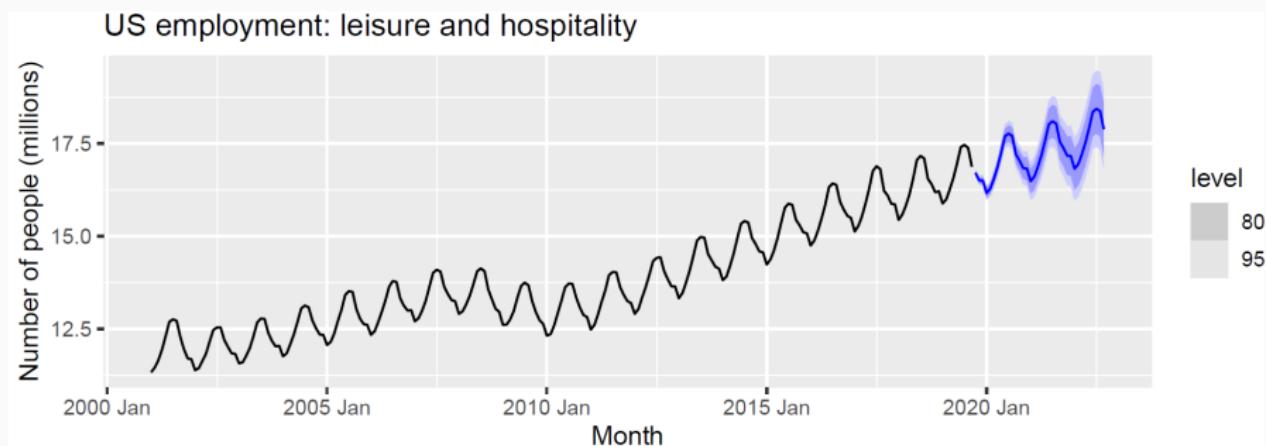


Application: US Leisure Employment and Hospitality

```
fit <- leisure %>%
  model(
    arima012011 = ARIMA(Employed ~ pdq(0,1,2) + PDQ(0,1,1)),
    arima210011 = ARIMA(Employed ~ pdq(2,1,0) + PDQ(0,1,1)),
    auto = ARIMA(Employed, stepwise = FALSE, approx = FALSE)
  )
fit %>% pivot_longer(everything(), names_to = "Model name",
                      values_to = "Orders")  
  
## # A mable: 3 x 2
## # Key:   Model name [3]
##   `Model name`          Orders
##   <chr>                <model>
## 1 arima012011  <ARIMA(0,1,2)(0,1,1)[12]>
## 2 arima210011  <ARIMA(2,1,0)(0,1,1)[12]>
## 3 auto        <ARIMA(2,1,0)(1,1,1)[12]>
```

Application: US Leisure Employment and Hospitality

```
forecast(fit, h=36) %>%
  filter(.model=='auto') %>%
  autoplot(leisure) +
  labs(title = "US employment: leisure and hospitality", y="Number of people (mi
```

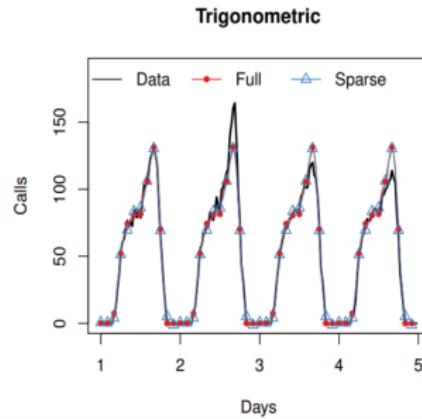
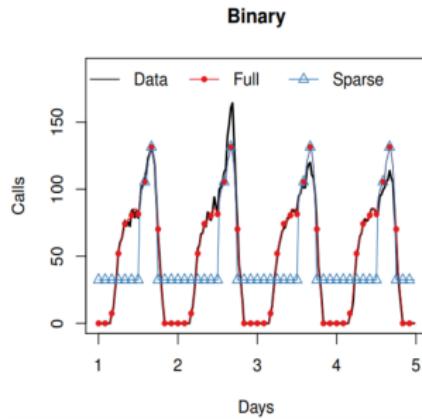


Treatment of Seasonality

- Daily data can exhibit multiple seasonal periodicities. This is a complication for all high-frequency forecasting problems: day in the month, day in the week, etc.
- This comes with additional complexity:
 - Months has different number of days
 - Leap years with different number of days
 - Weeks do not align with daily cycles (the year is not divisible in an exact number of weeks)
- Seasonality can be irregular: Ramadan and some other religious festivities for instance
- We use two approaches to deal with complex seasonality
 - ① Trigonometric representation of seasonality
 - ② Simplification of seasonal terms

Binary and Trigonometric Seasonality

- Consider a single deterministic seasonal cycle. We can model this either by:
 - Indicator variables** (e.g. friday's dummy: equals to 1 on Friday and 0 otherwise)
 - Trigonometric variables** (cf. Fourier Analysis): Trigonometric approach



Binary and Trigonometric Seasonality: Pros and Cons

- Elimination of terms:
 - Eliminating terms from the binary (indicator variable) representation, removes the effect of seasonal periods
 - Eliminating terms in the trigonometric representation results in smoother approximations of the seasonal shape
 - Elimination of terms can be done using information criteria
- Besides, trigonometric representation is more parsimonious, as it involves less parameters to estimate: reduce parametric noise
- Model-wise: ARIMA with dummy variables for seasonality versus SARIMA (“Seasonal ARIMA”) that suggests stochastic seasonality

ARIMA with Trigonometric Seasonality

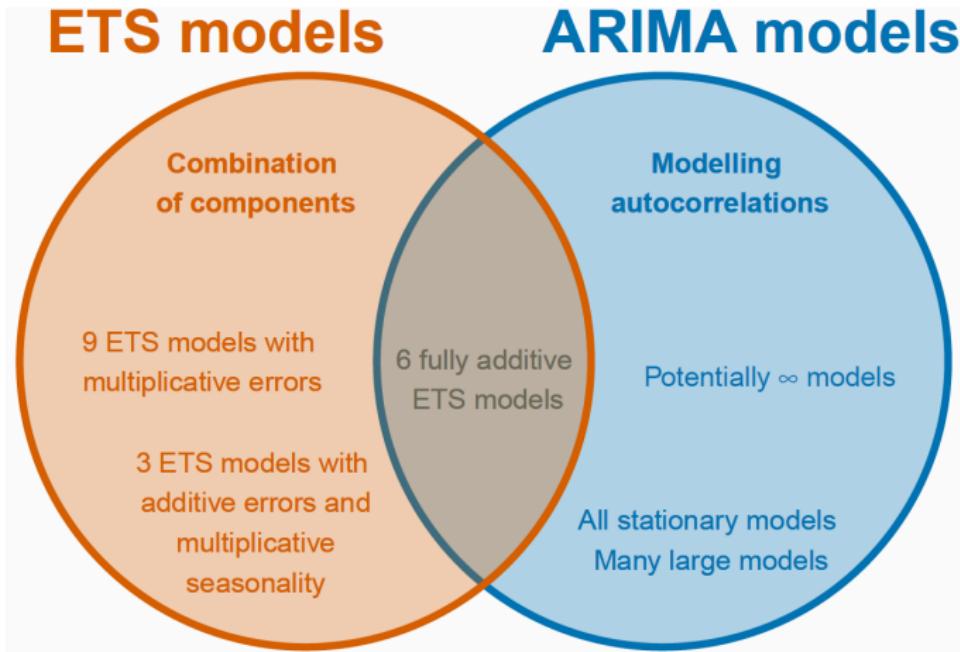
- $ARIMA(p, d, q)(P_1, D_1, Q_1)_{S_1}$
- $S_1 = 5$ days in the week
- The model contains trigonometric variables of the form $\sin \frac{2t \times i_j \pi}{S_j}$ and $\cos \frac{2t \times i_j \pi}{S_j}$
 - ▶ i_j

ARIMA versus ETS

While the general perception sees ARIMA has more general than ETS, this is not exactly true:

- Linear ETS are all special cases of ARIMA models
- Non-linear exponential smoothing model (e.g. multiplicative) have no ARIMA equivalent
- Many ARIMA have no linear exponential smoothing model equivalent
- ETS models are non-stationary. The number of roots depends on the inclusion of seasonal terms or not

ARIMA versus ETS Models



Question 1

What are the problems associated with using many binary variables to model seasonality? Multiple answers possible:

- ① Complicated to estimate
- ② Takes excessive computation time
- ③ Add parametric noise
- ④ Risk of overfit