

Road map!

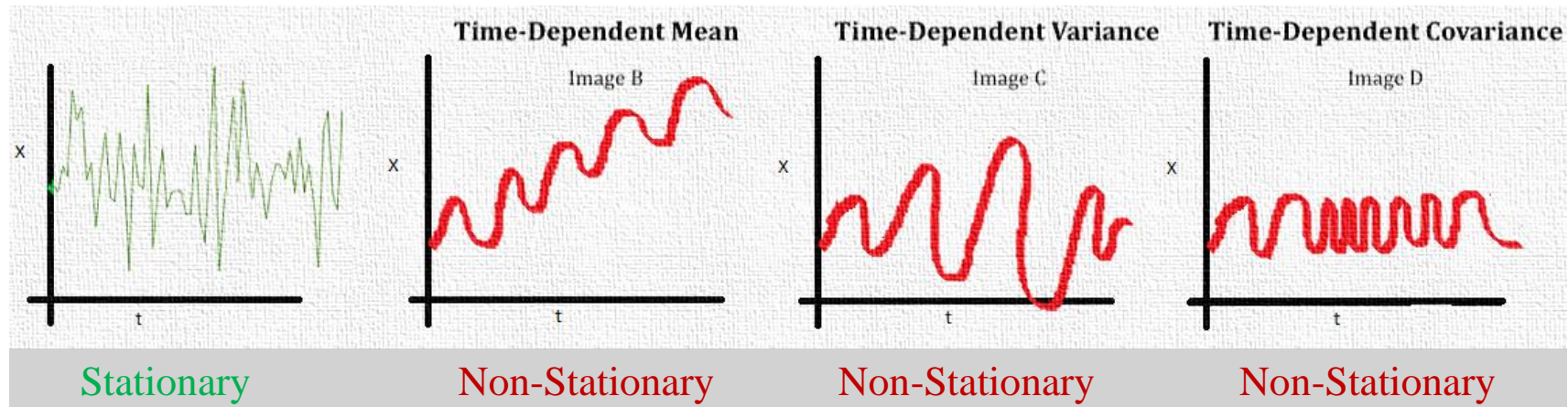
- Module 1- Introduction to Deep Forecasting
- Module 2- Setting up Deep Forecasting Environment
- Module 3- Exponential Smoothing
- **Module 4- ARIMA models**
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Prophet and Neural Prophet



Module 4 – Part I

ARIMA models' Prerequisites

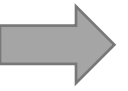
ACF, PACF, Stationarity, Differencing



→ ARIMA models prerequisites

- ARIMA stands for AutoRegressive Integrated Moving Average. It is a class of **statistical models** for analyzing and forecasting time series data.
- ETS and ARIMA models are two popular models for forecasting time series data. They offer **complementary approaches** to addressing the challenges of time series forecasting.
- **ARIMA** models describe **autocorrelations** in the data, whereas **ETS** models describe **trends** and **seasonality**.
- Let's review some prerequisites before moving forward with the models:

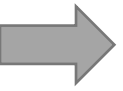




Autocorrelation

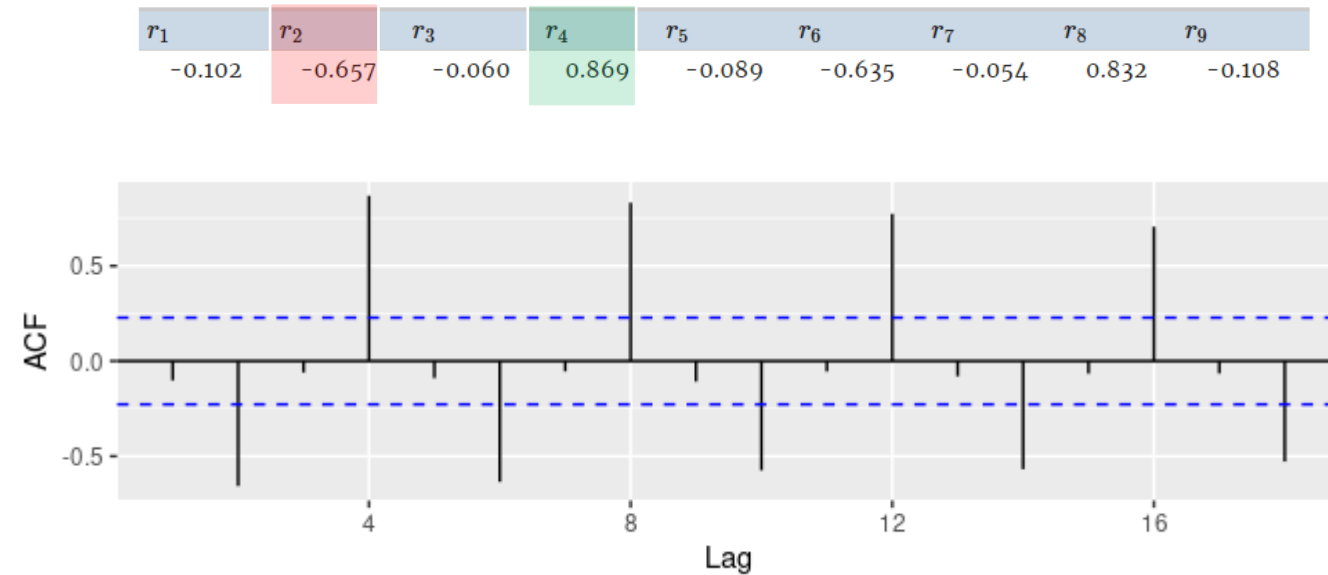
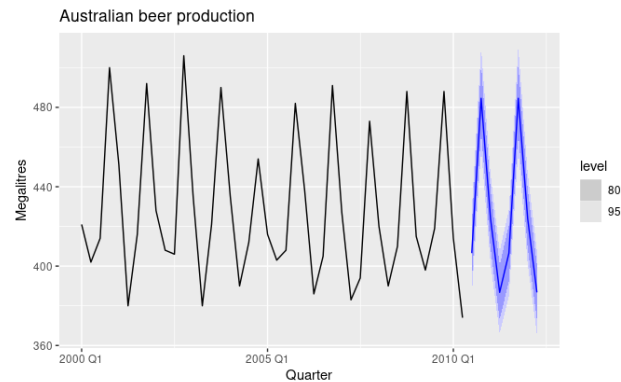
- **Autocorrelation**, also known as **serial correlation**, is a measure of the correlation between a time series and a lagged version of itself.
- It is used to assess the **degree to which** the past values of a time series **are predictive** of its future values.

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}$$



ACF: Autocorrelation Function

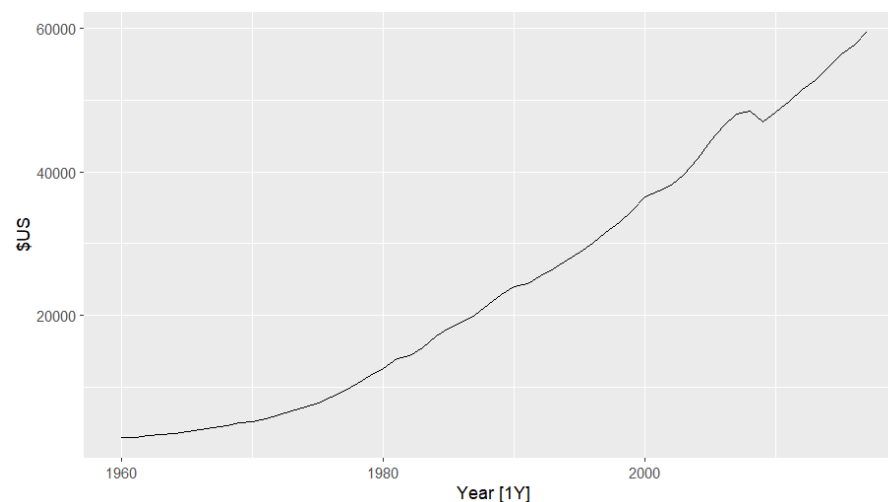
- The autocorrelation function (**ACF**) is a statistical tool that can be used to measure the autocorrelation of a time series.
- It calculates the correlation between the time series and lagged versions of itself at different lag periods.

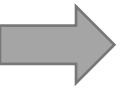


➔ Partial Autocorrelation

- **Partial autocorrelation**, also known as **partial serial correlation**, is a measure of the correlation between a time series and a lagged version of itself, **controlling for the effects of intermediate lag periods**.
- y_t and y_{t-2} might be correlated, simply because they are both connected to y_{t-1} , rather than because of any new information contained in y_{t-2} . **Partial autocorrelation** overcomes this problem.

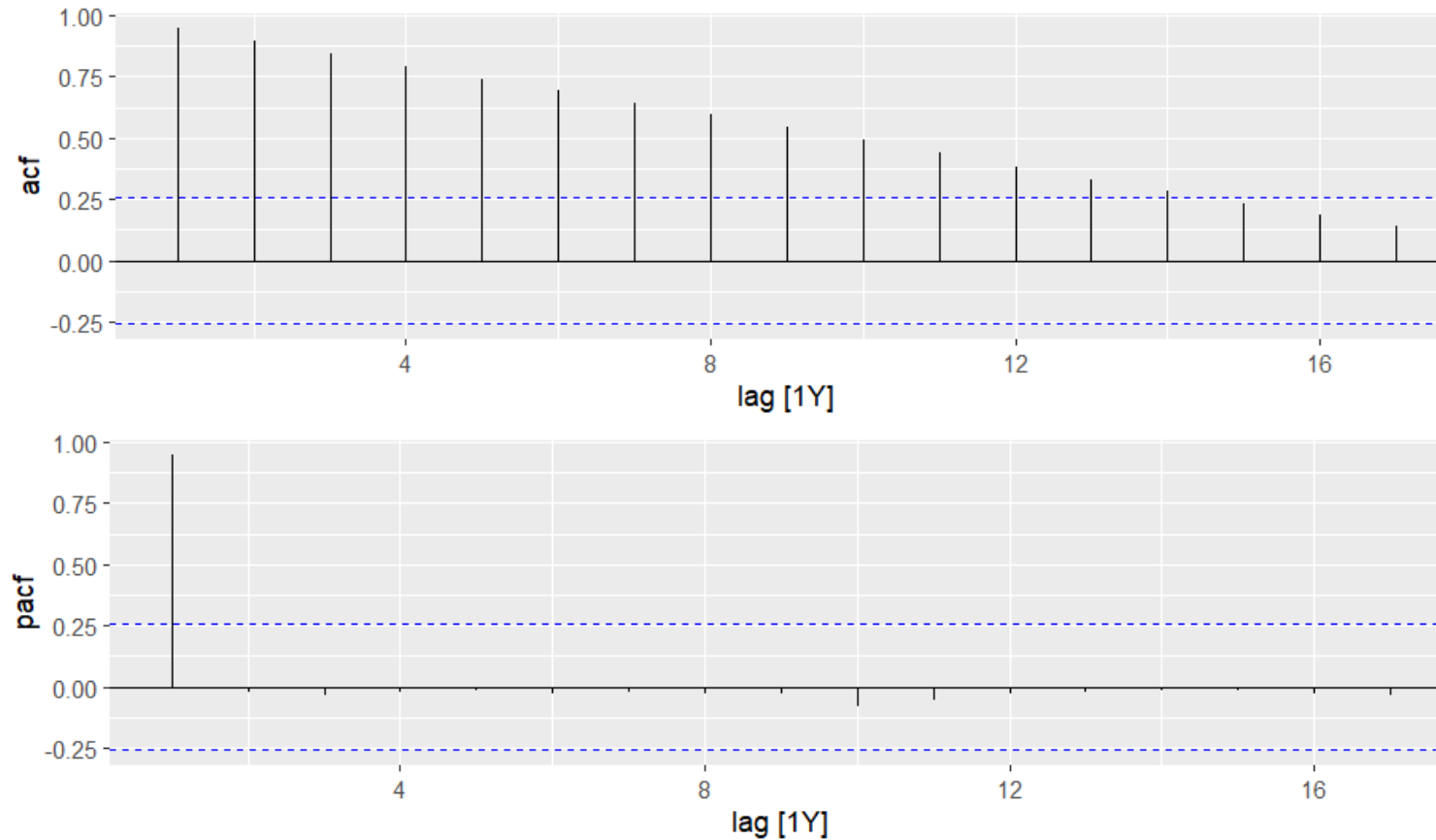
US Annual GDP per capita (1960-2017)

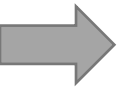




PACF: Partial Autocorrelation Function

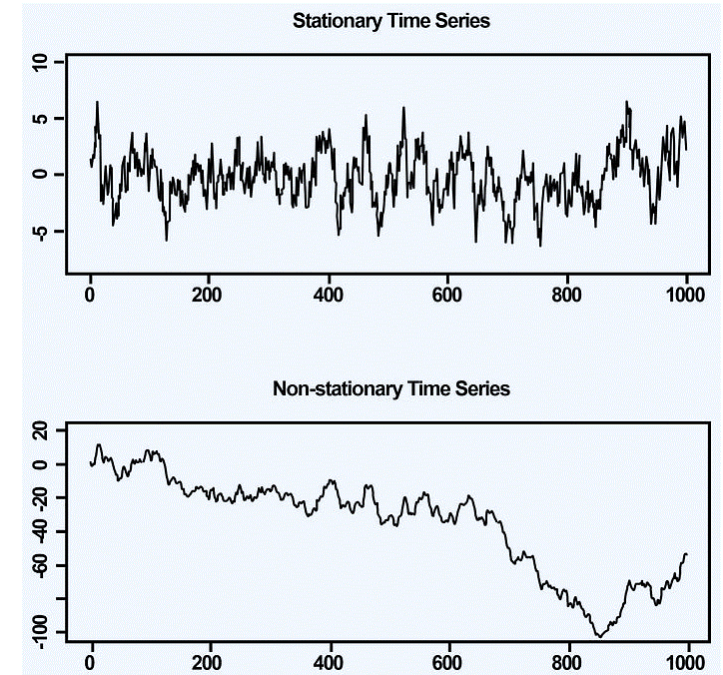
- PACF is a statistical tool that can be used to measure the partial autocorrelation of a time series.





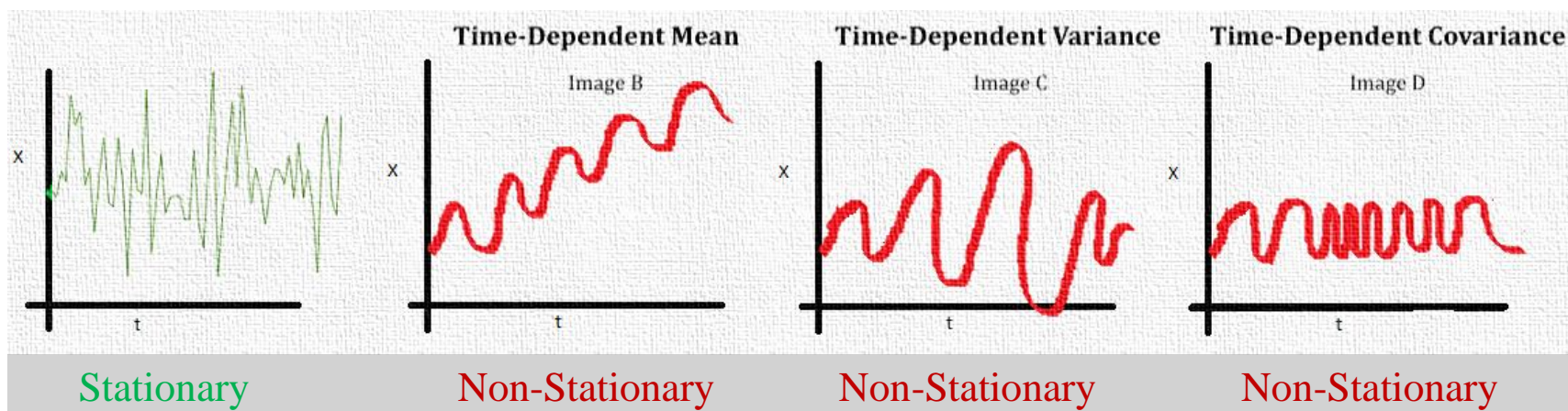
Stationarity

- Stationary vs Non-Stationary Data. What makes a data set **Stationary**?
- In a stationary timeseries, the statistical properties **do not depend on the time**
- **Predictability**: Stationary time series are easier to predict because you can assume that future statistical properties will not change.
- This doesn't mean we cannot predict non-stationary data!
- Data with **trend** and **seasonality** are **NOT** stationary!
- **Data granularity matters**: The level of detail in your data can impact its stationarity



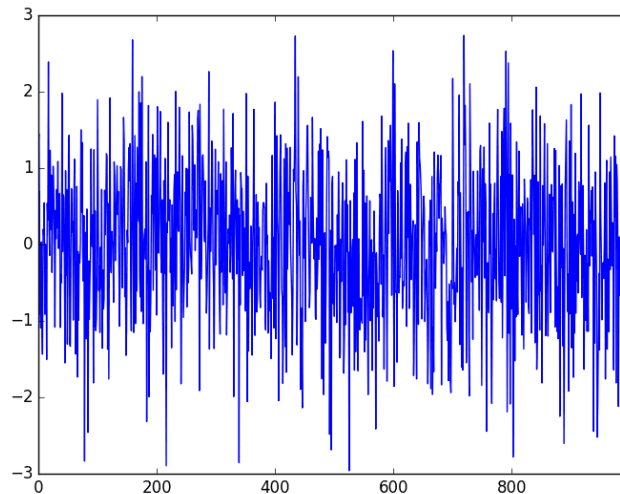
➔ Weak vs Strong Stationarity

- **Weak** Stationarity (**Covariance** Stationary): A time series is considered weakly stationary if the following conditions hold:
 1. Constant **Mean**: The mean of the process is constant over time.
 2. Constant **Variance**: The variance of the process is constant over time.
 3. **Covariance** Depends Only on Lag: The covariance between two points in the time series depends only on the time difference (lag) between the points, not their absolute position in time.

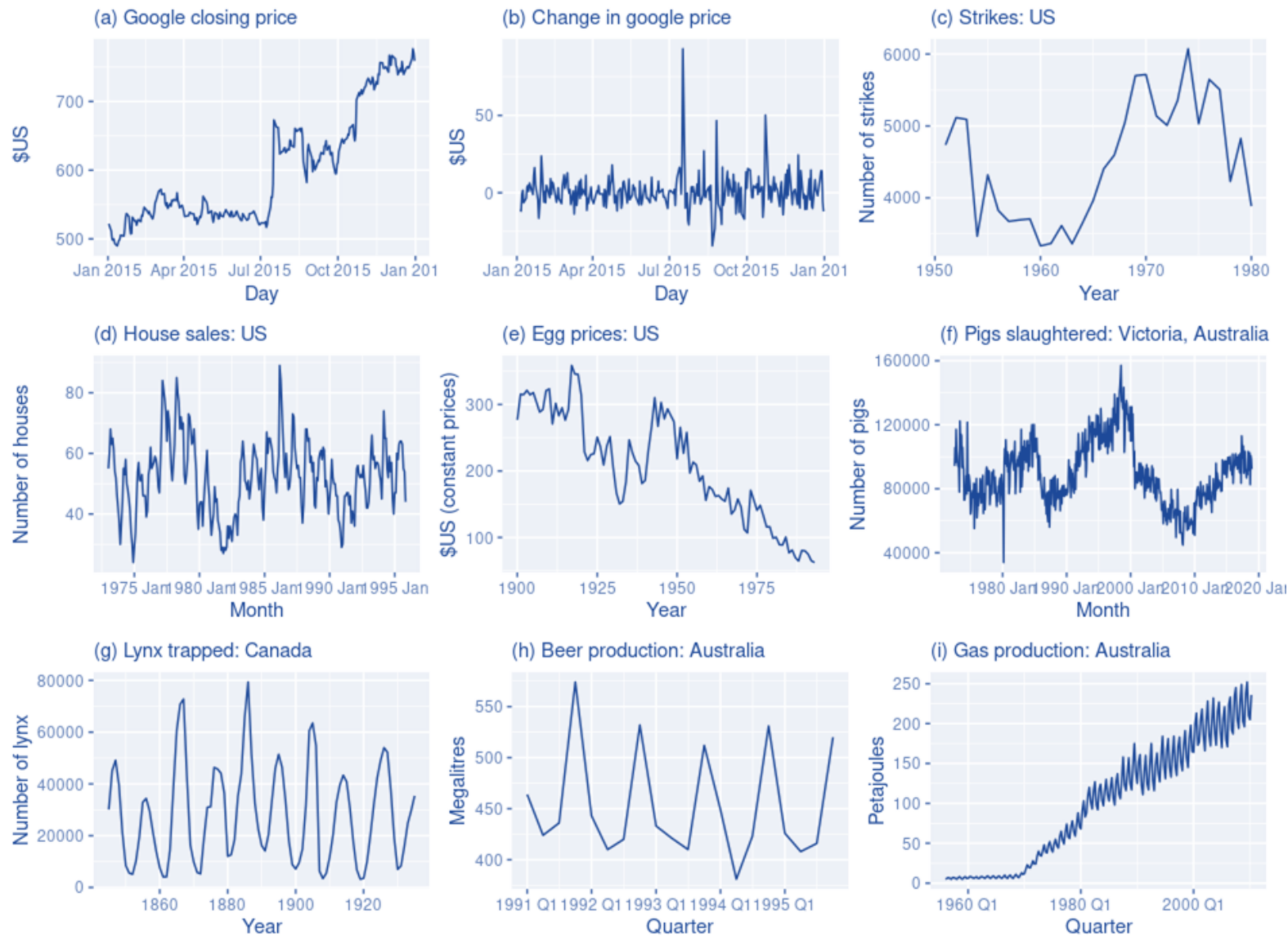


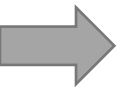
→ Weak vs Strong Stationarity

- **Strong stationarity:** A time series is considered strongly stationary if its **joint probability distribution** does not change when shifted in time
 - **All moments** of the series (mean, variance, skewness, kurtosis, etc) and joint distributions remain constant, irrespective of the time period



➔ Which ones are stationary?





Differencing

- Differencing: Computing the difference between consecutive observations.
- **Differencing** helps to **stabilize the mean** of a time series by removing changes in the level and therefore reducing the trend and seasonality.
- Recall: **Transformations** help to **stabilize the variance** of a time series.

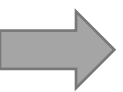
Original data 1st differenced 2nd differenced

Time1	10
Time2	12
Time3	8
Time4	14
Time5	7

Time1	NA
Time2	2
Time3	-4
Time4	6
Time5	-7

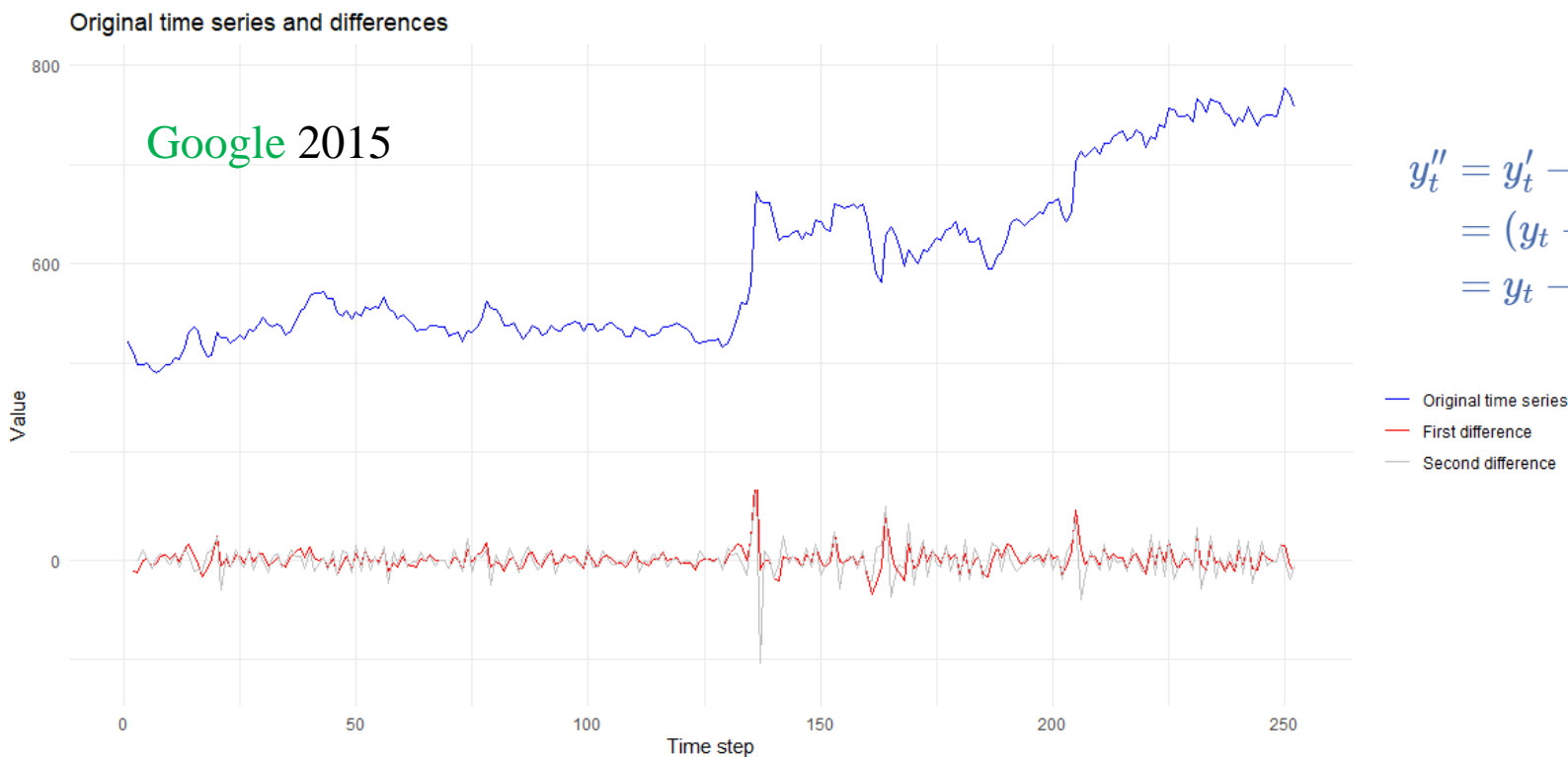
Time1	NA
Time2	NA
Time3	-6
Time4	10
Time5	-13





2nd Differencing

- Occasionally the differenced data will not appear to be stationary, and it may be necessary to difference the data a **second time** to obtain a stationary series.
- Second differencing is **change in change**.
- In practice, it is almost never necessary to go beyond second-order differences.



$$\begin{aligned}y_t'' &= y_t' - y_{t-1}' \\&= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\&= y_t - 2y_{t-1} + y_{t-2}.\end{aligned}$$

→ Seasonal Differencing

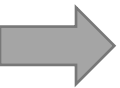
- A seasonal difference is the difference between an observation and the previous observation **from the same season**.

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

- **m** is the number of seasons. This is also called lag-**m** difference.
- If seasonal differenced is white noise, then

$$y_t = y_{t-m} + \varepsilon_t$$

- Recall:
 - **Seasonal Naïve forecast**: each forecast set to be equal to the last observed value **from the same season**



Put it together!

Original data



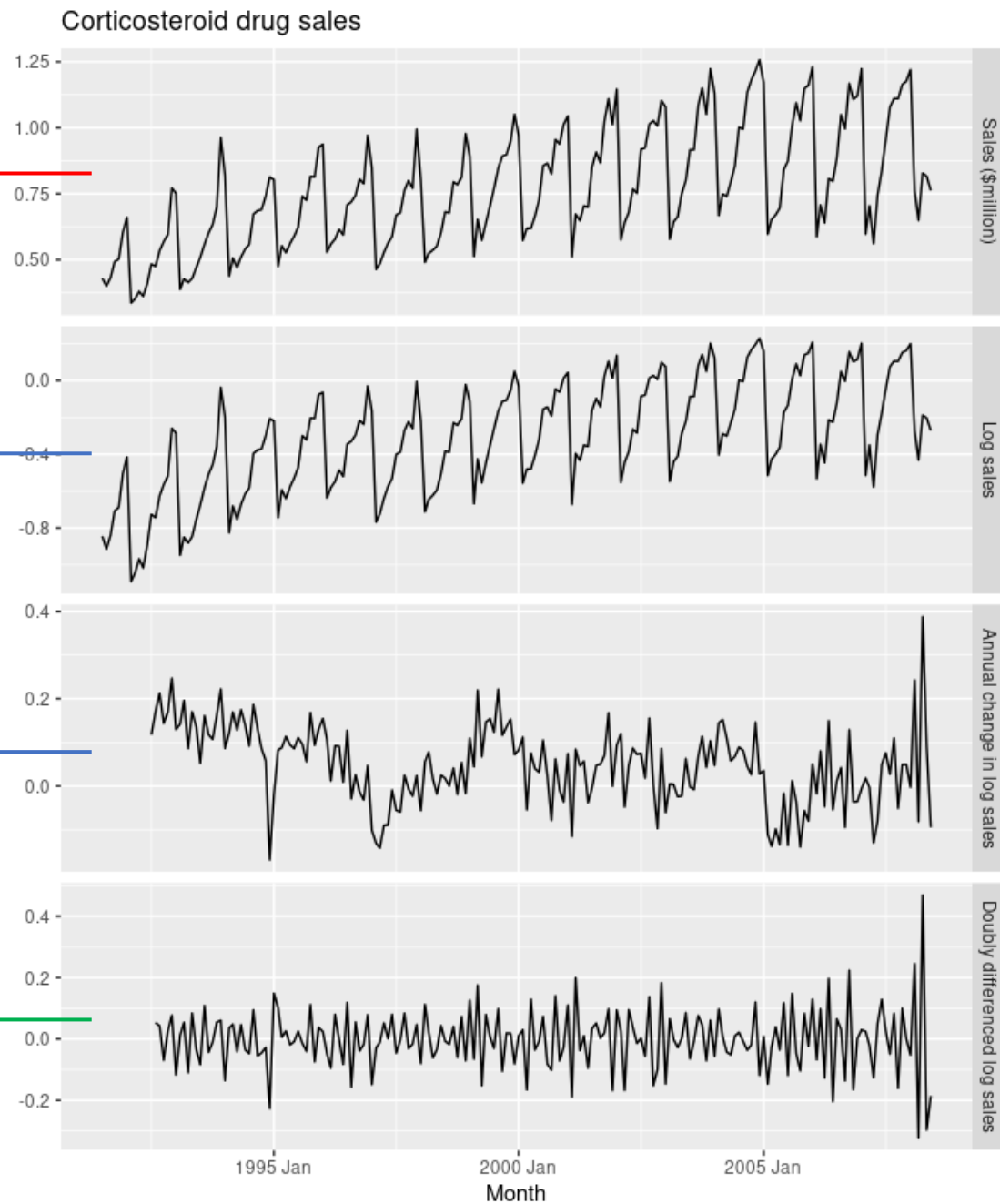
Log transformed



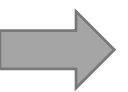
Seasonal difference



1st differenced seasonal difference



Interpretable?



Random Walk

- Random Walk: When the 1st differenced series is white noise

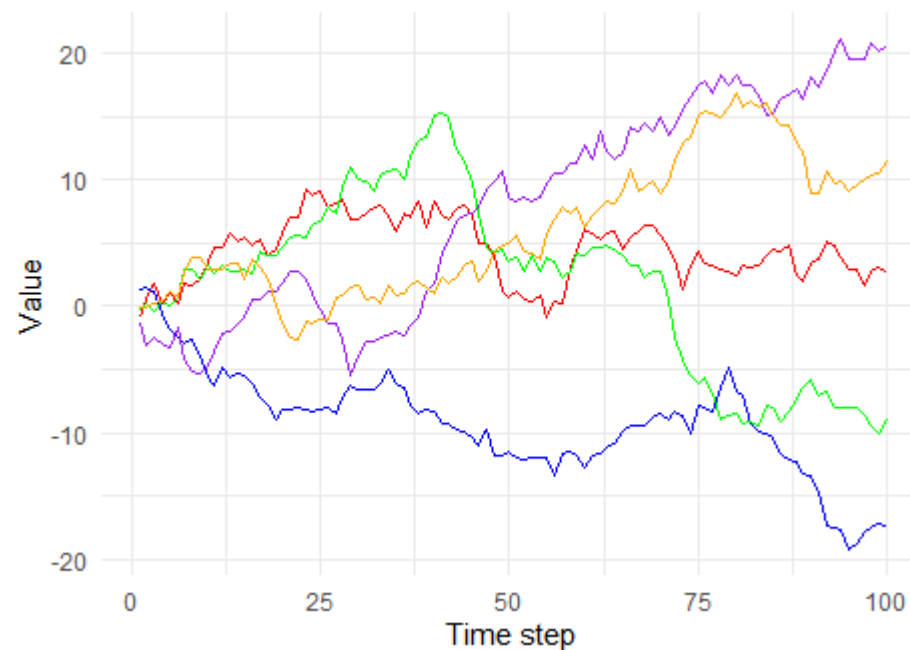
$$y_t - y_{t-1} = \varepsilon_t$$



$$y_t = y_{t-1} + \varepsilon_t$$

- Random walk models are widely used for non-stationary data, particularly **financial** and economic data.
- Random walks typically have **long periods of up or down trend** + **sudden change in direction**.
- Random walk with **no drift** = **Naïve** forecasting model

Five random walks without drift

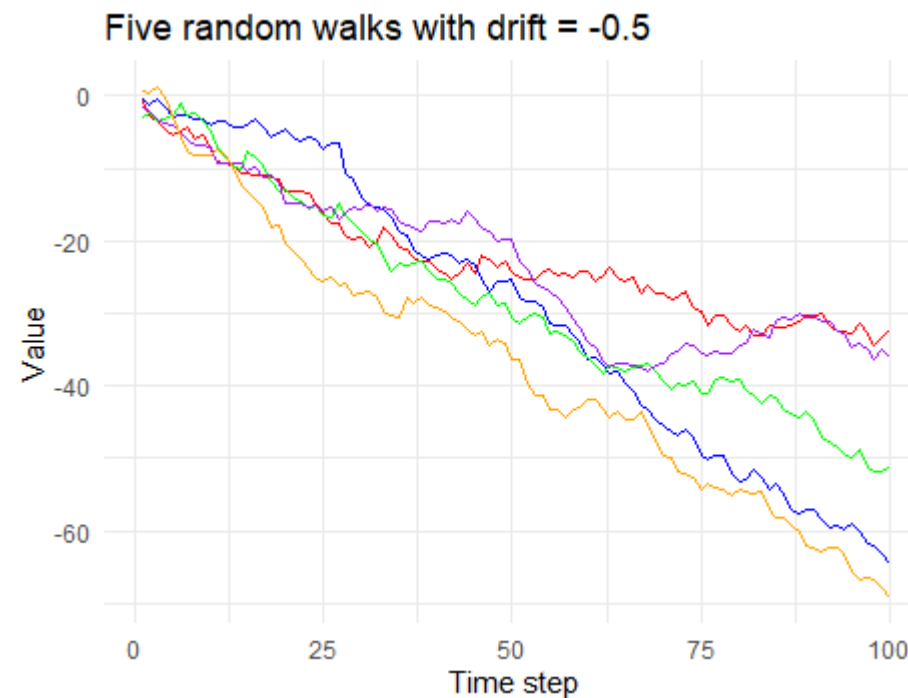
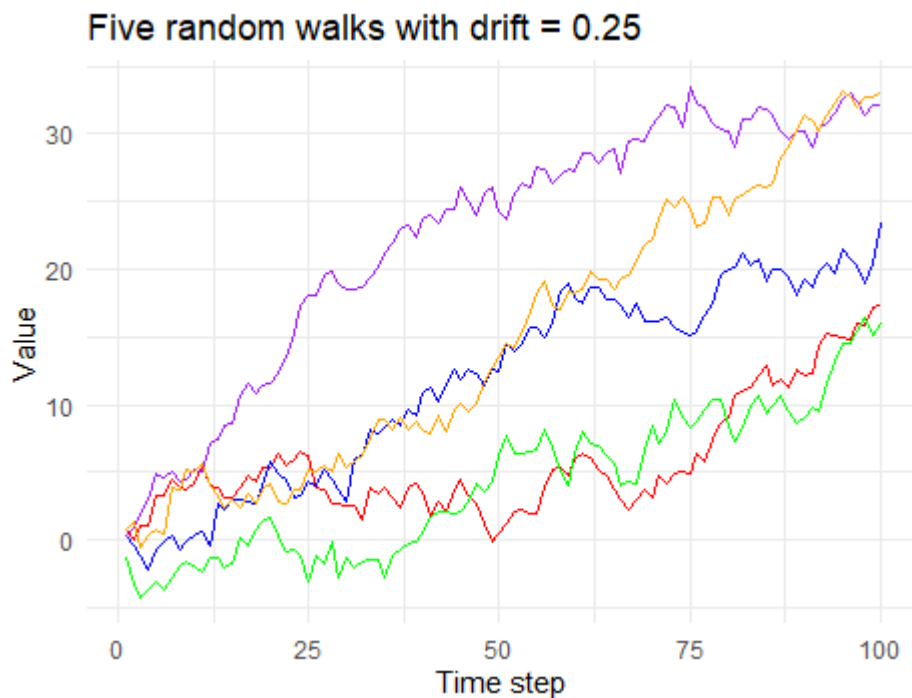


➔ Random Walk with Drift

- Random walk with drift c (the 1st difference does not have zero average):

$$y_t - y_{t-1} = c + \varepsilon_t \quad \text{or} \quad y_t = c + y_{t-1} + \varepsilon_t$$

- C is the average change between consecutive observations.



→ Testing for Stationarity

- Unit root test is a statistical test used to determine whether a time series **has a unit root**, which is a characteristic of a **non-stationary time series**
- There are several different unit root tests including:
 1. Augmented Dickey-Fuller (**ADF**) test.
 2. Kwiatkowski-Phillips-Schmidt-Shin (**KPSS**) test.

Hypothesis Test	Null	Alternative	P-value to get stationarity
ADF	Non-Stationary	Stationary	Small
KPSS	Stationary	Non-Stationary	Large

	ADF	KPSS
ADF statistic	-12.533939	0.012944
p-value	0.0	0.1
should we difference?	?	
conclusion		

→ Testing for Stationarity

- Unit root test is a statistical test used to determine whether a time series **has a unit root**, which is a characteristic of a **non-stationary time series**
- There are several different unit root tests including:
 1. Augmented Dickey-Fuller (**ADF**) test.
 2. Kwiatkowski-Phillips-Schmidt-Shin (**KPSS**) test.

Hypothesis Test	Null	Alternative	P-value to get stationarity
ADF	Non-Stationary	Stationary	Small
KPSS	Stationary	Non-Stationary	Large

	ADF	KPSS
ADF statistic	-12.533939	0.012944
p-value	0.0	0.1
should we difference?	False	False
conclusion	stationary	stationary

Dickey-Fuller Test Variants

There are three main variants of the DF test, depending on the inclusion of a constant (α) and a deterministic trend (βt):

1. **Model without a drift and without a trend:**

$$\Delta y_t = \gamma y_{t-1} + \varepsilon_t$$

2. **Model with a drift but without a trend:**

$$\Delta y_t = \alpha + \gamma y_{t-1} + \varepsilon_t$$

3. **Model with both a drift and a trend:**

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \varepsilon_t$$

In each case, the focus is on testing the coefficient γ . The null hypothesis is that $\gamma = 0$, indicating a unit root (non-stationarity). Rejecting the null hypothesis suggests that the series does not have a unit root and is stationary.

The Dickey-Fuller test's limitation, which the Augmented Dickey-Fuller test addresses, is its inability to handle higher-order serial correlation. The ADF extends the DF test by including lagged differences of the dependent variable to account for this serial correlation, thus providing a more general approach to testing for unit roots.

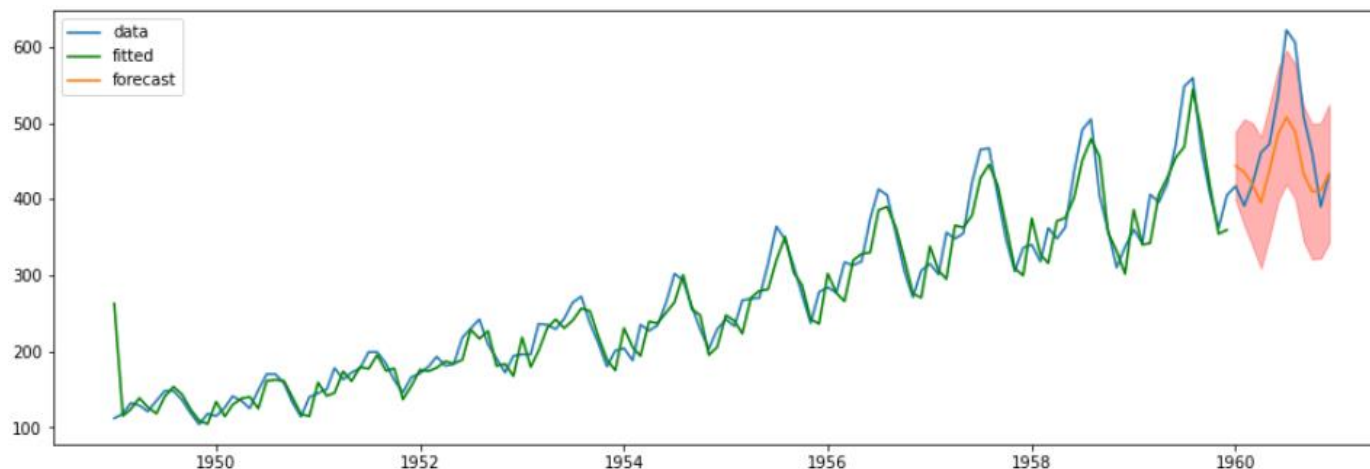
$$\Delta y_t = \alpha + \beta t + \gamma(y_{t-1} - y_t) + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t$$

→ Components of ARIMA model

1. Autoregressive (AR) term - captures the autocorrelation in the data
2. Integrated (I) term - removes the non-stationarity in the data
3. Moving Average (MA) term - captures the error term or noise in the data

How it works?

- The AR term models the current **value** of the time series as a **linear combination** of its past **values**.
- The I term models the **differences** between the current **value** and the past **value**.
- The MA term models the current **error** term as a **linear combination** of the past **error** terms.

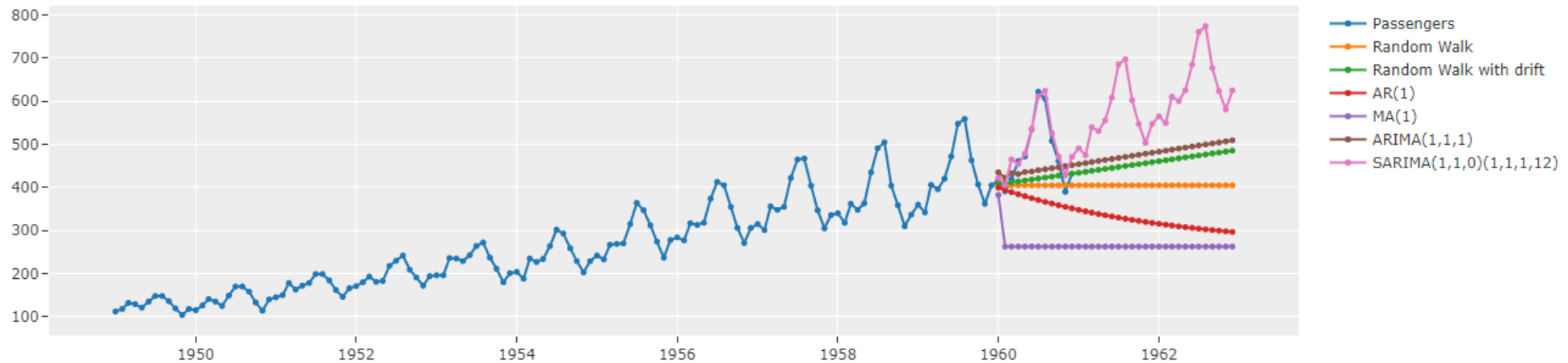


Module 4 – Part II

ARIMA models



Actual vs. Forecast (Out-of-Sample)



→ Components of ARIMA model

ARIMA

1. Autoregressive (**AR**) term - captures the autocorrelation in the data
2. Integrated (**I**) term - removes the non-stationarity in the data
3. Moving Average (**MA**) term - captures the error term or noise in the data

→ Autoregressive models

- An autoregressive (AR) model is a statistical model (multiple linear regression model) that uses **lagged** variable as **predictors**
- **Autoregression** = regression of the variable against **itself**
- AR(**p**) model, autoregressive model of order **p**.

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

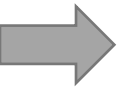
- In AR(1) model:
 - when $\phi_1 = 0$ and $c = 0$, y_t is equivalent to ?
 - when $\phi_1 = 1$ and $c = 0$, y_t is equivalent to ?
 - when $\phi_1 = 1$ and $c \neq 0$, y_t is equivalent to ?
 - when $\phi_1 < 0$, y_t tends to oscillate around the mean.

→ Autoregressive models

- An autoregressive (AR) model is a statistical model (multiple linear regression model) that uses **lagged** variable as **predictors**
- **Autoregression** = regression of the variable against **itself**
- AR(**p**) model, autoregressive model of order **p**.

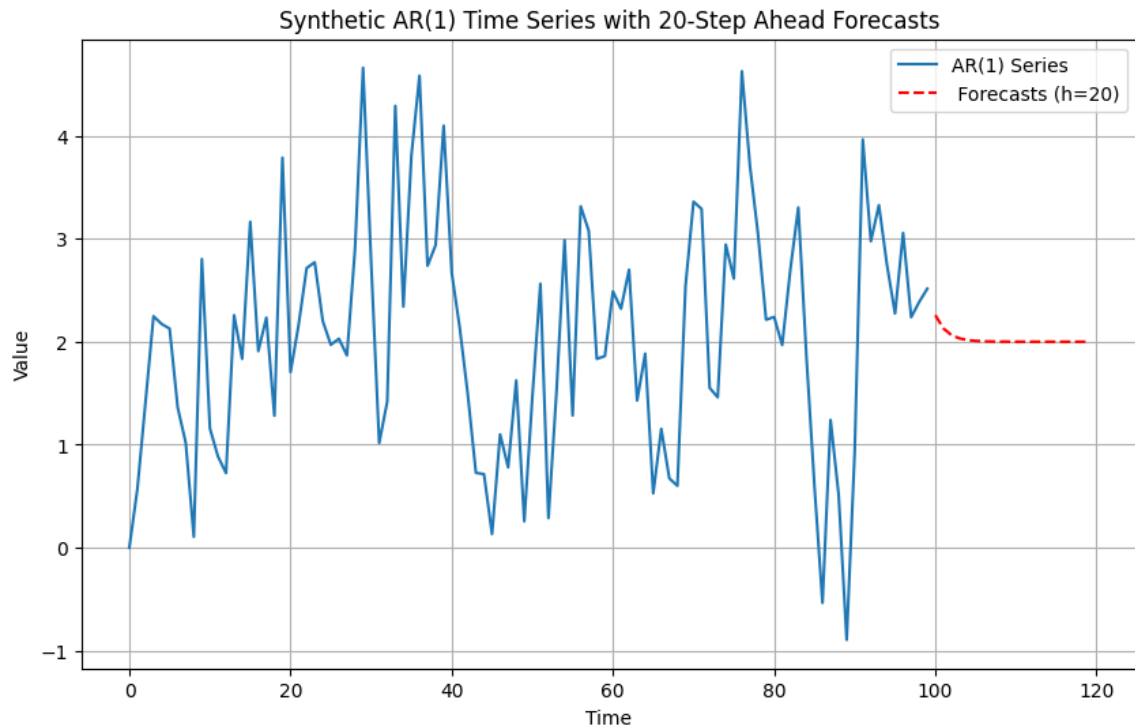
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

- In AR(1) model:
 - when $\phi_1 = 0$ and $c = 0$, y_t is equivalent to 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 around the mean.



Autoregressive Models (Forecasting)

$$y_t = c + \phi y_{t-1} + \varepsilon_t$$



$$c = 1, \quad \phi = 0.5, \quad \varepsilon_t \sim N(0, \sigma = 1)$$

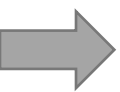
Forecasting Equation:

$$\hat{y}_{t+1} = c + \phi y_t$$

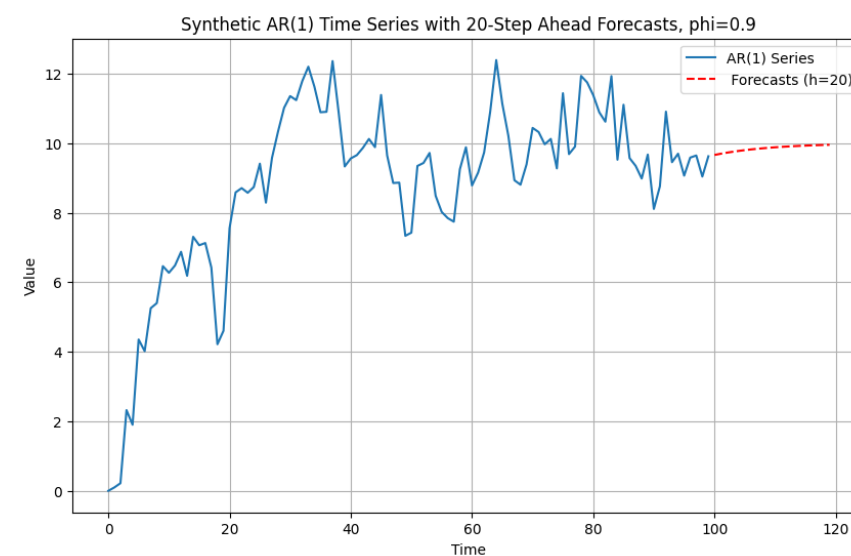
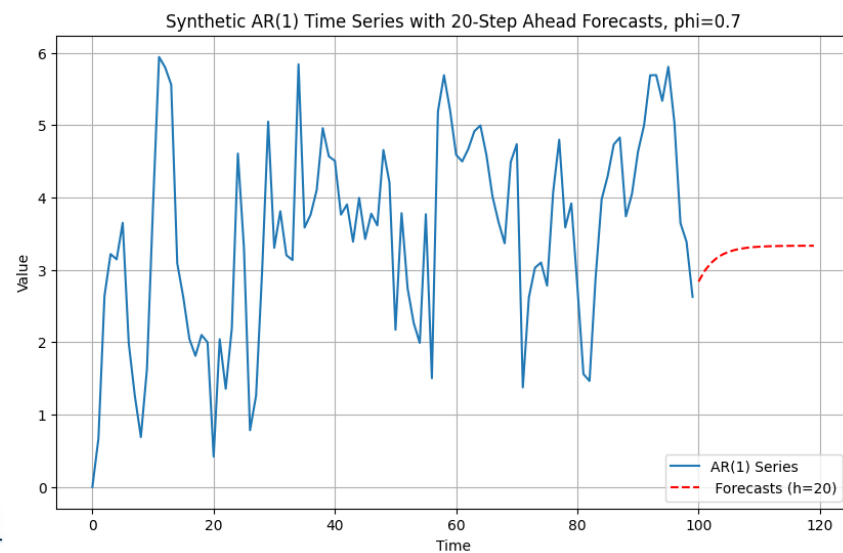
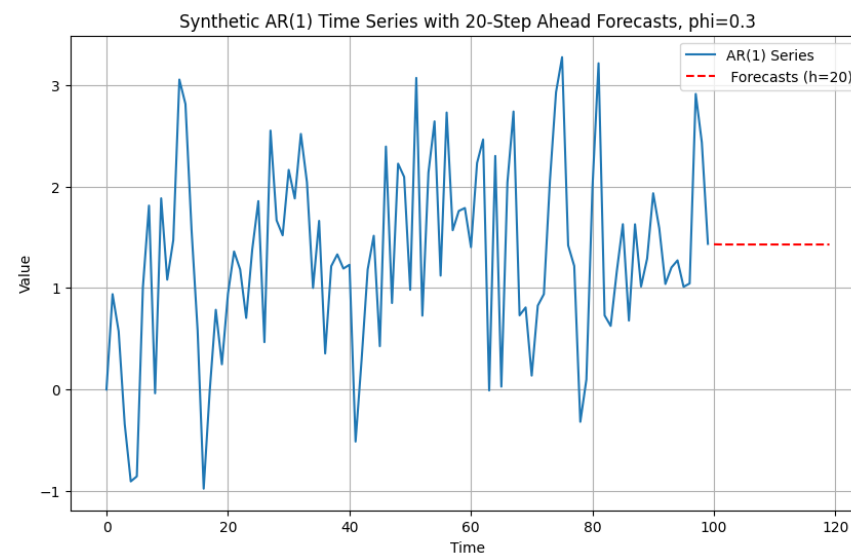
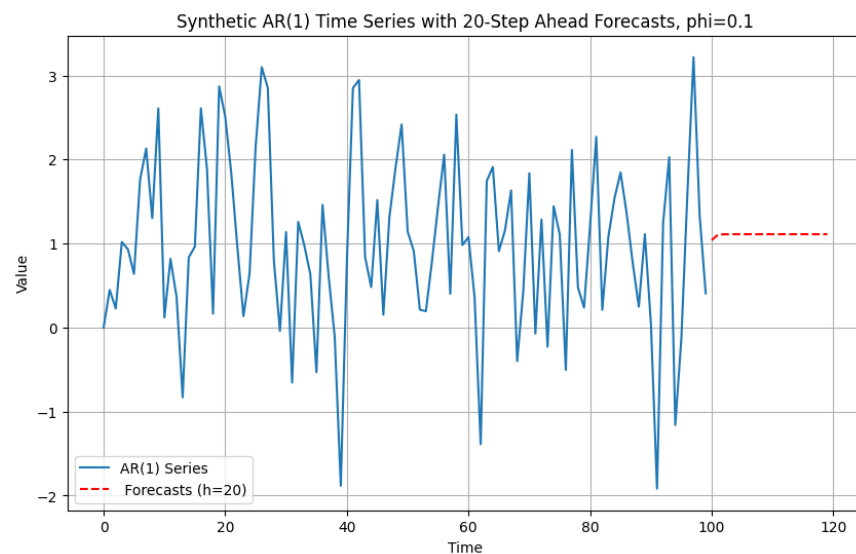


$$\hat{y}_{t+h|t} = c(1 + \phi + \phi^2 + \dots + \phi^{h-1}) + \phi^h y_t$$

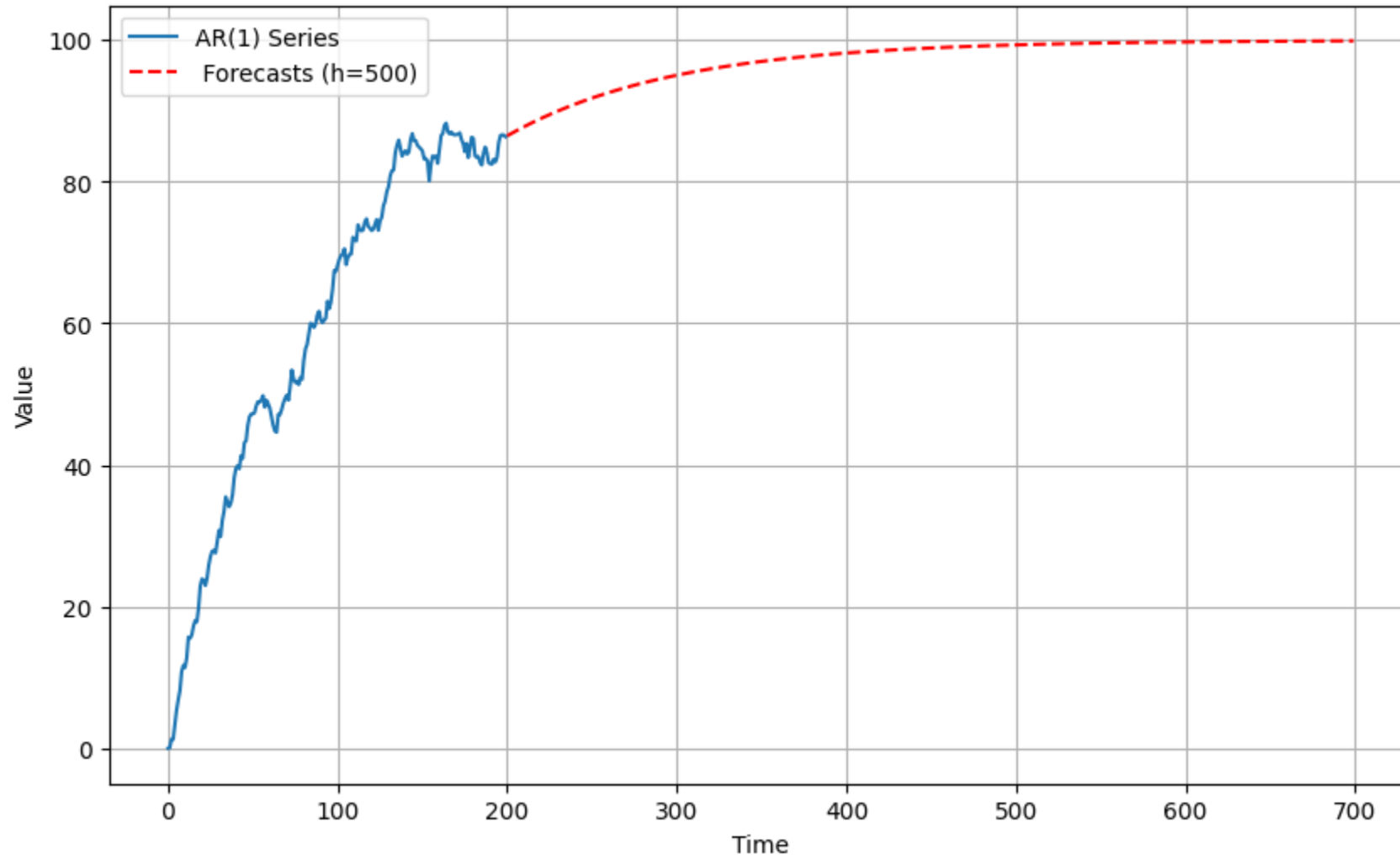
$$\hat{y}_{t+h|t} = c \frac{1 - \phi^h}{1 - \phi} + \phi^h y_t$$



Autoregressive Models (Examples)



Synthetic AR(1) Time Series with 500-Step Ahead Forecasts, $\phi=0.99$



→ Moving Average Models

- A moving average model uses past forecast **errors** in a regression-like model
- MA(**q**) model, a moving average model of order **q**.

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

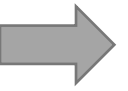
- y_t can be thought of as a weighted moving average of the past few forecast errors
- We require $|\theta| < 1$, the most recent observations carry a greater weight than those from the distant past.

→ Moving Average Models

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

- Do **NOT** confuse this model with simple moving average method or exponentially weighted moving average method.
- Moving average **models** is used for forecasting future values!
- Moving average **smoothing** (SMA, EWMA, ...) is used for estimating the trend-cycle of past values.

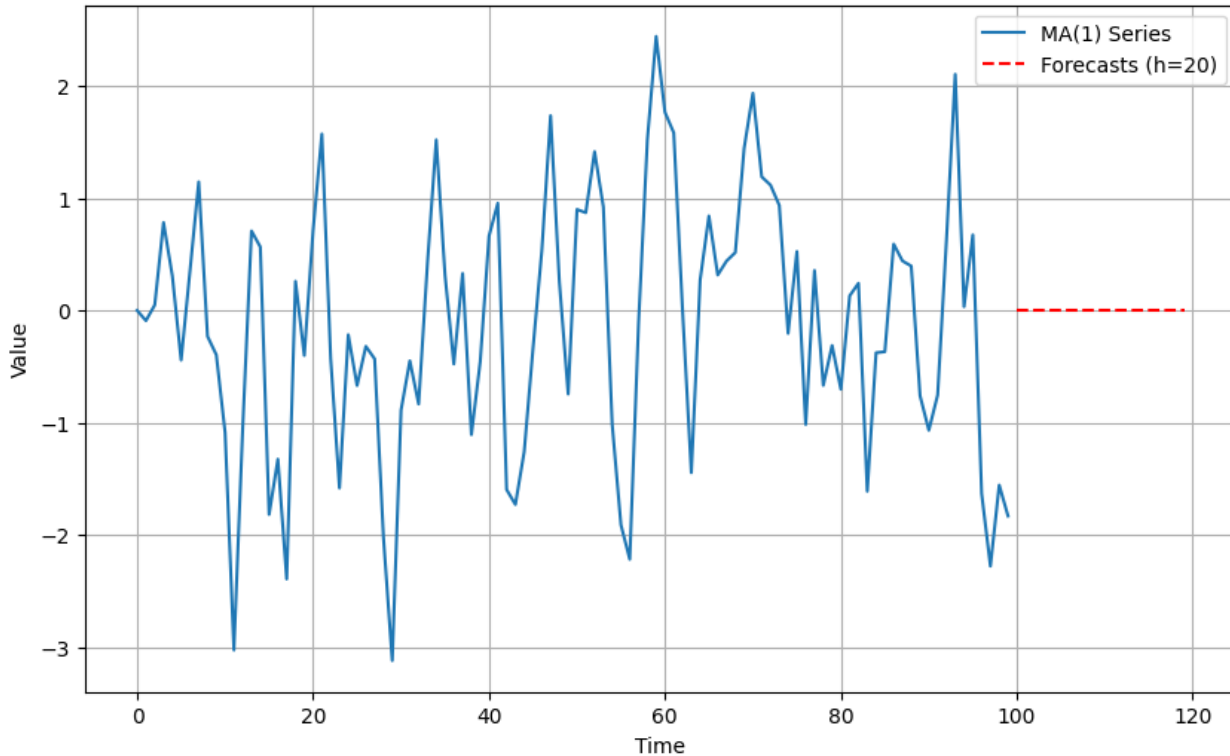




Moving Average Models (Forecasting)

$$y_t = \mu + \theta \varepsilon_{t-1} + \varepsilon_t$$

Synthetic MA(1) Time Series with 20-Step Ahead Forecasts, theta=0.5



$$\mu = 0, \quad \theta = 0.5, \quad \varepsilon_t \sim N(0, \sigma = 1)$$

Forecasting Equation:

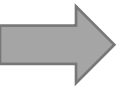
$$\hat{y}_{t+1|t} = \mu + \theta \varepsilon_t$$



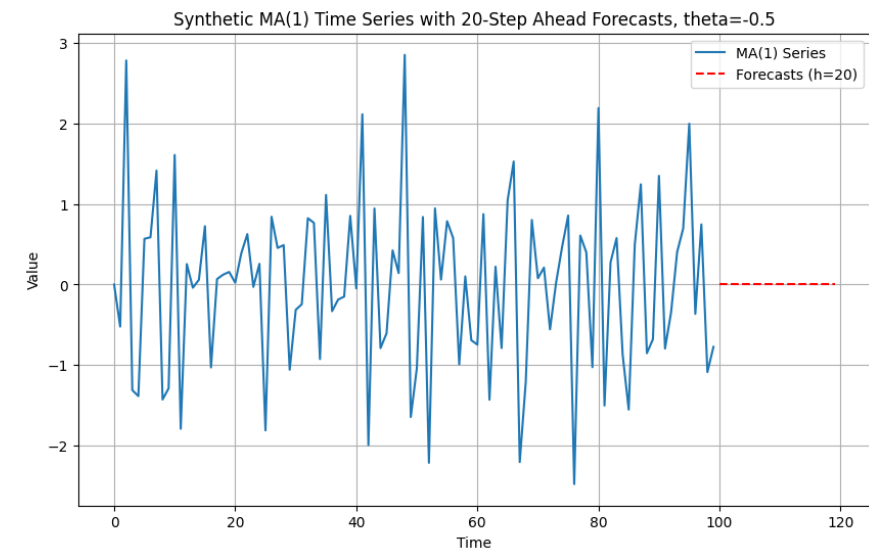
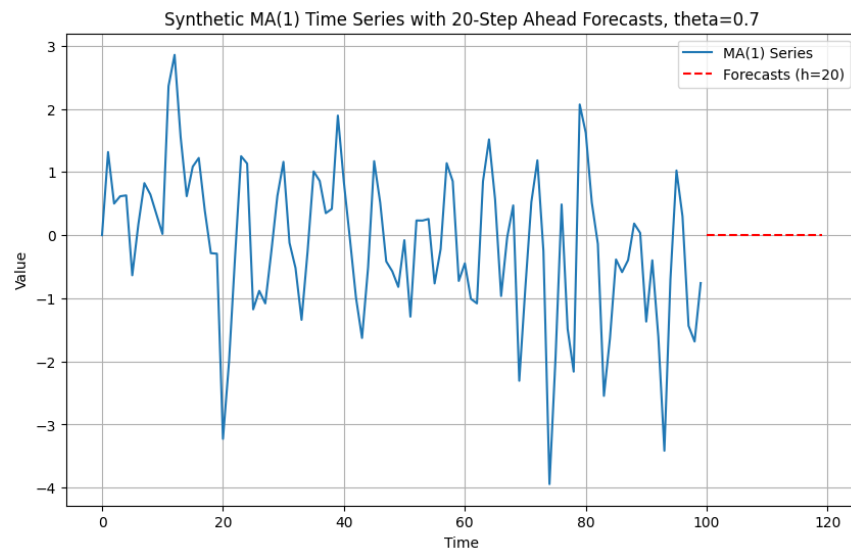
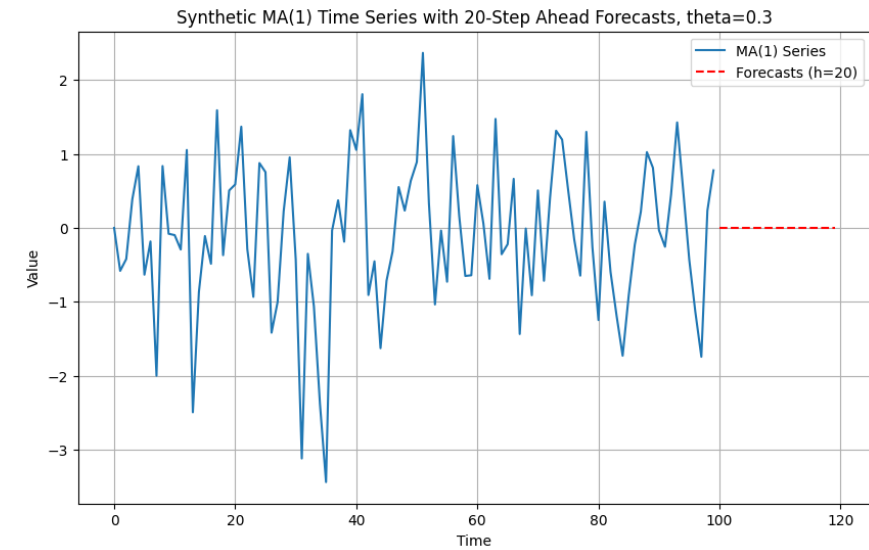
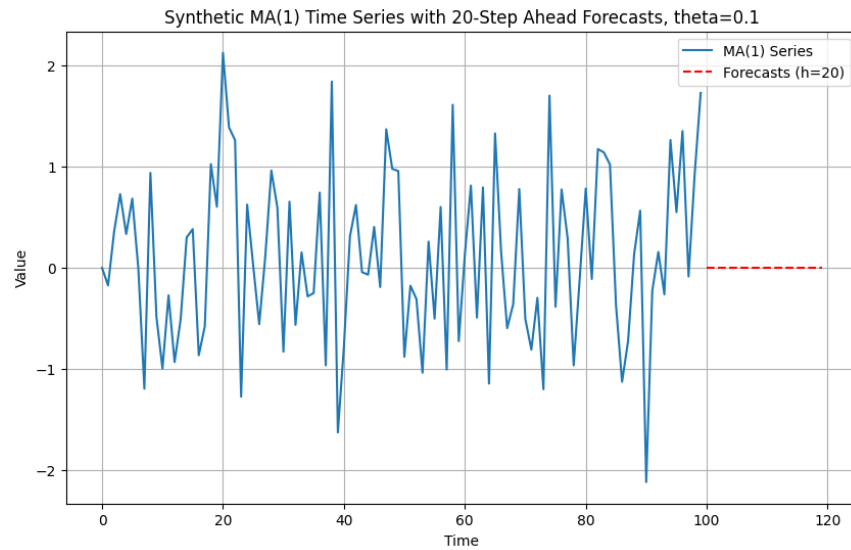
Since future error terms are unknown

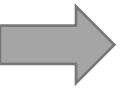


$$\hat{y}_{t+h|t} = \mu$$



Moving Average Models (Examples)





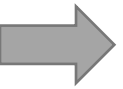
ARIMA (AutoRegressive Integrated Moving Average)

- ARIMA model combines three models, autoregressive (**AR**) model, an integrated (**I**) model, and a moving average (**MA**) model.
- ARIMA(**p**, **d**, **q**) model.

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

- y'_t is the differenced time series.
- d degree of first difference involved.
- Note: p, d, and q are estimated using **MLE**.

?	ARIMA(0,0,0) with no constant
?	ARIMA(0,1,0) with no constant
?	ARIMA(0,1,0) with a constant
?	ARIMA(p,0,0)
?	ARIMA(0,0,q)



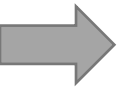
ARIMA (AutoRegressive Integrated Moving Average)

- ARIMA model combines three models, autoregressive (**AR**) model, an integrated (**I**) model, and a moving average (**MA**) model.
- ARIMA(**p**, **d**, **q**) model.

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

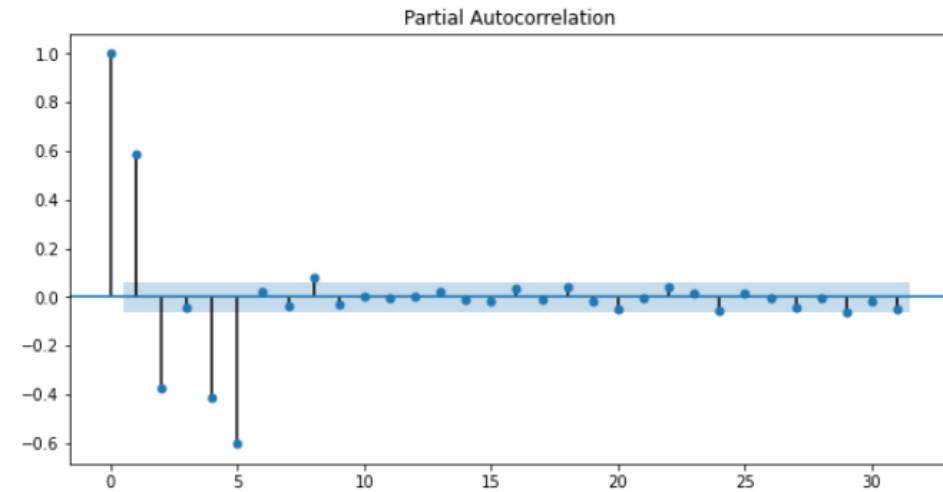
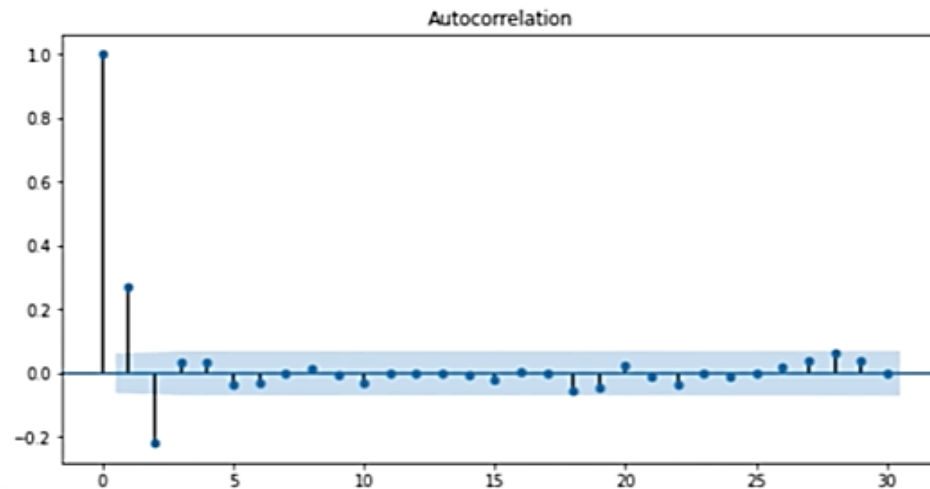
- y'_t is the differenced time series.
- d degree of first difference involved.
- Note: p , d , and q are estimated using **MLE**.

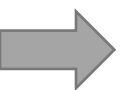
White noise	ARIMA(0,0,0) with no constant
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p ,0,0)
Moving average	ARIMA(0,0, q)



Selecting (p, q) orders using ACF and PAC

- Some rough guidelines:
- Identification of an **AR** model is often best done with the **PACF**
 - **p** set to be the maximum significant non-zero lag in PACF typically followed by a **sharp decline**.
- Identification of an **MA** model is often best done with the **ACF**
 - **q** set to be the maximum significant non-zero lag in ACF typically followed by a **sharp decline**.





Model selection

- For model selection we can either use **information criteria** or any **cross validated** performance metrics like R^2 , MSE, RMSE, MAPE, sMAPE.

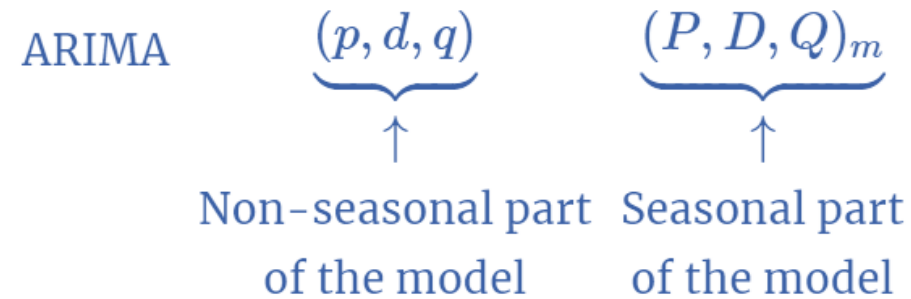
Information Criteria	Formula
Akaike's Information Criterion (AIC)	$AIC = -2 \log(L) + 2k$
AIC corrected for small sample bias (AICc)	$AIC_c = AIC + \frac{2k(k+1)}{T-k-1}$
Bayesian Information Criterion (BIC)	$BIC = AIC + k[\log(T) - 2]$

- **L** is the likelihood of the model and **K** is the total number of parameters (including the variance of residuals)
- The model with the **minimum information criteria** is often the best model for forecasting

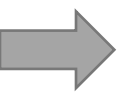
→ ARIMA model estimation (optional)

➔ SARIMA (Seasonal ARIMA) models

- SARIMA is an extension of an ARIMA model that includes **additional seasonal terms**.
- It is used to model time series data that exhibits seasonal patterns

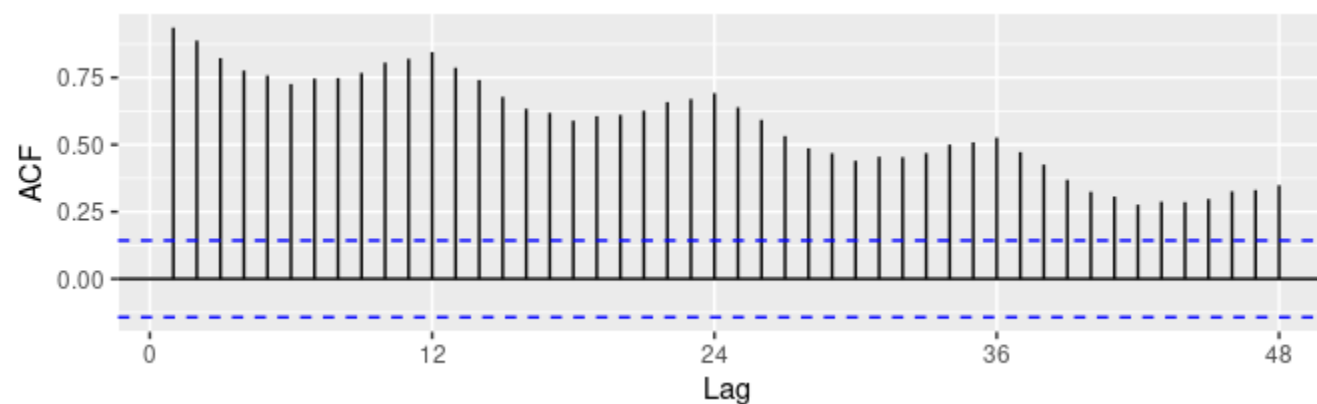
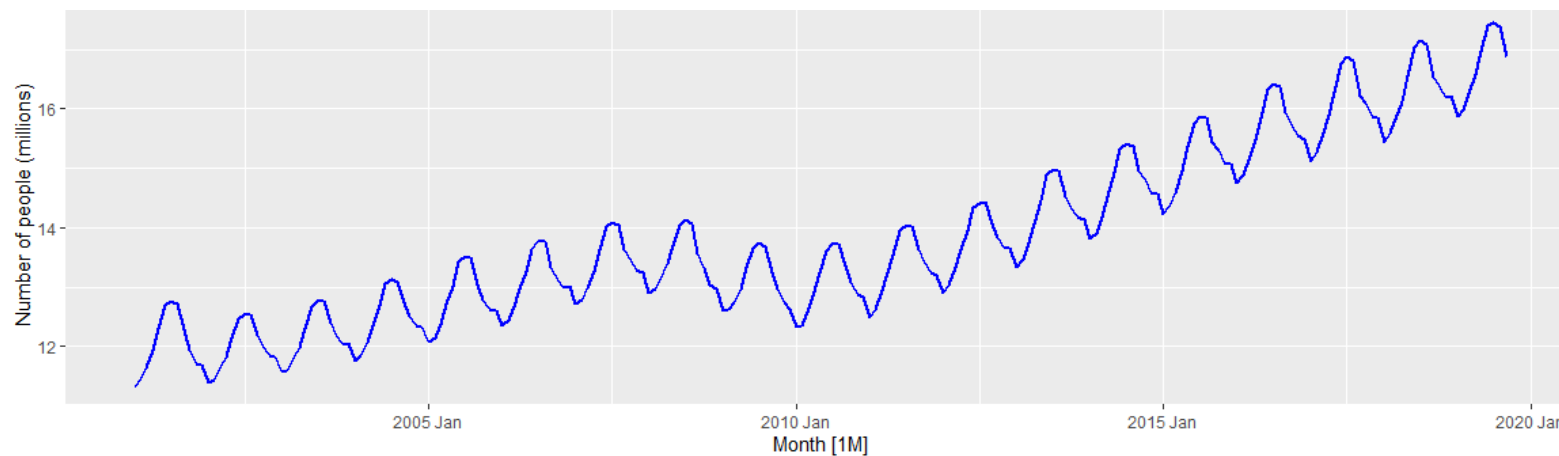


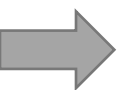
- p, d, q are defined as before.
- P is the order of the **seasonal** autoregressive component
- D is the degree of **seasonal** differencing
- Q is the order of the **seasonal** moving average component
- m is the **period of the seasonality**. $m = 4, 12$ is for quarterly and monthly seasonality, respectively.



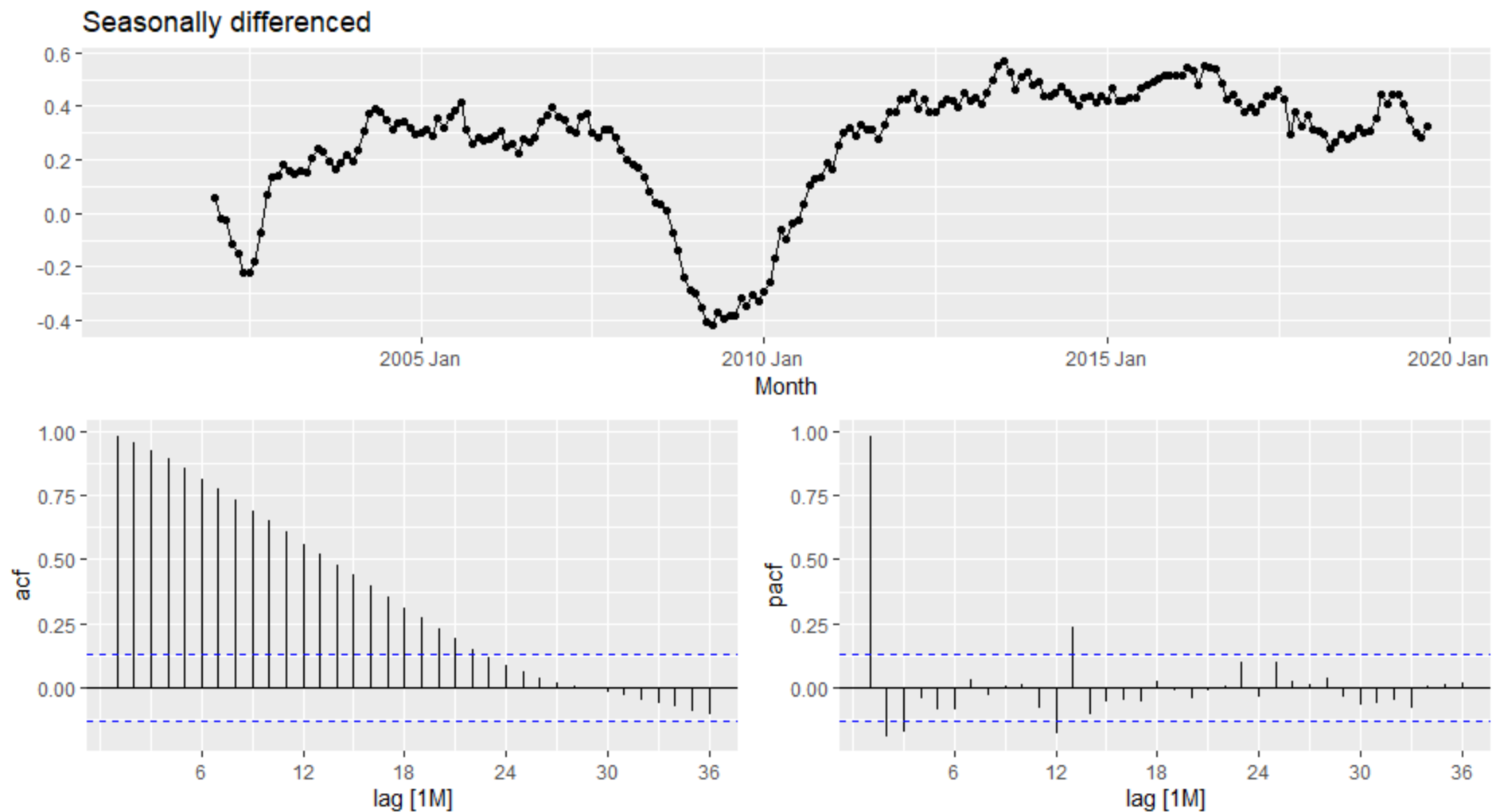
SARIMA example

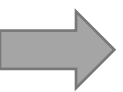
Monthly US leisure and hospitality employment, 2001-2019.



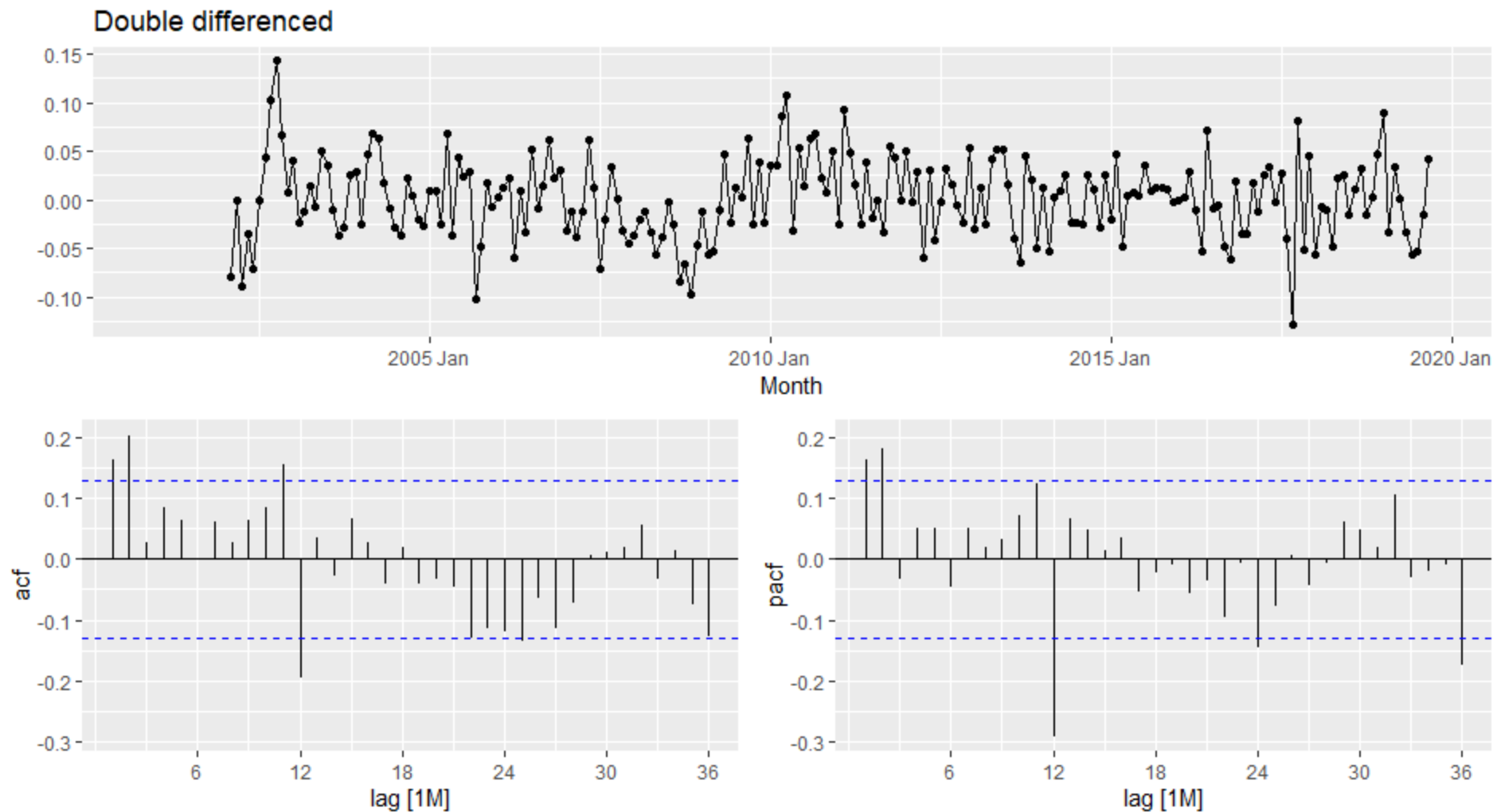


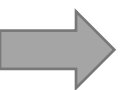
SARIMA example





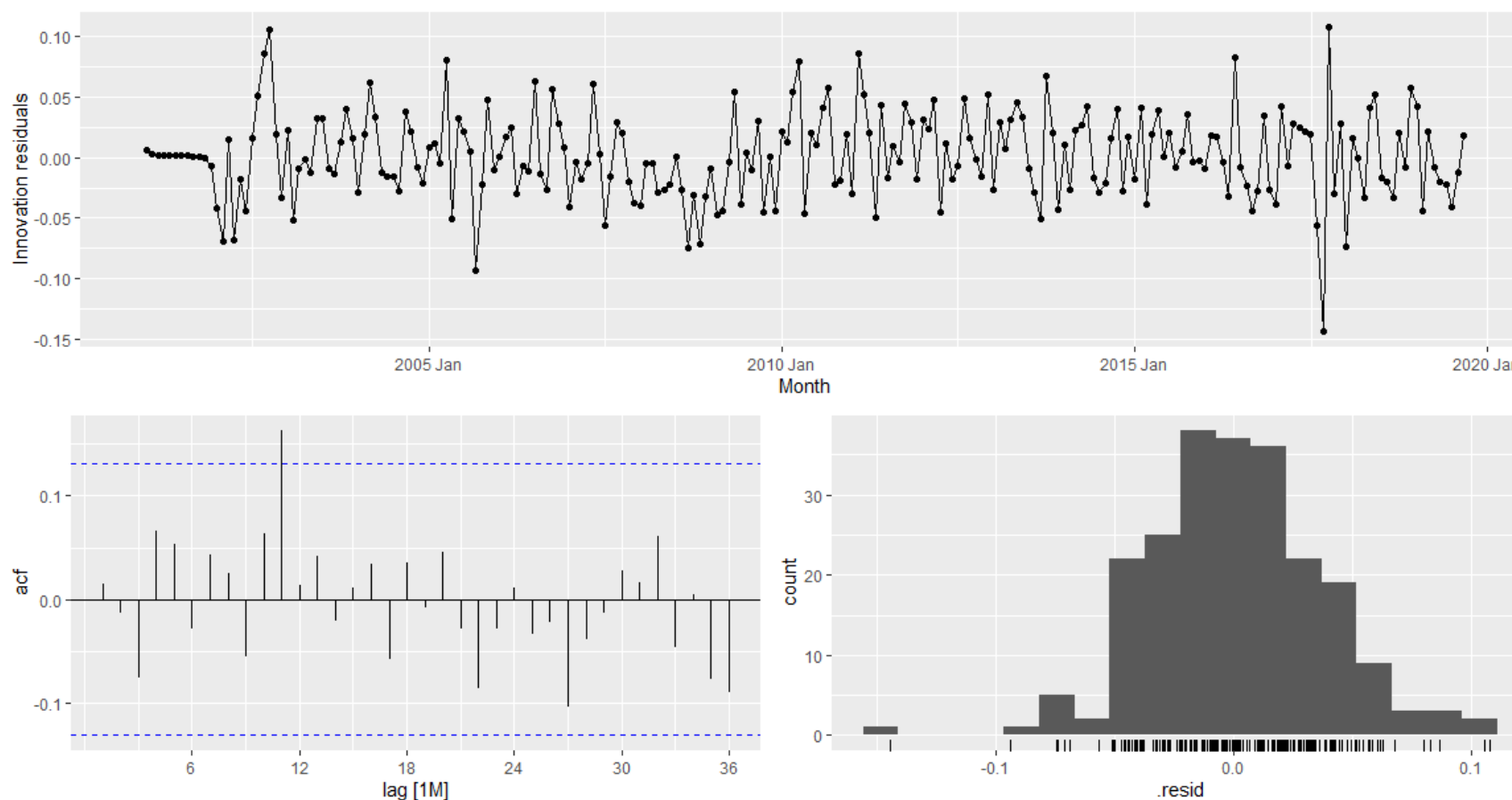
SARIMA example

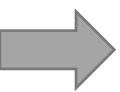




SARIMA example

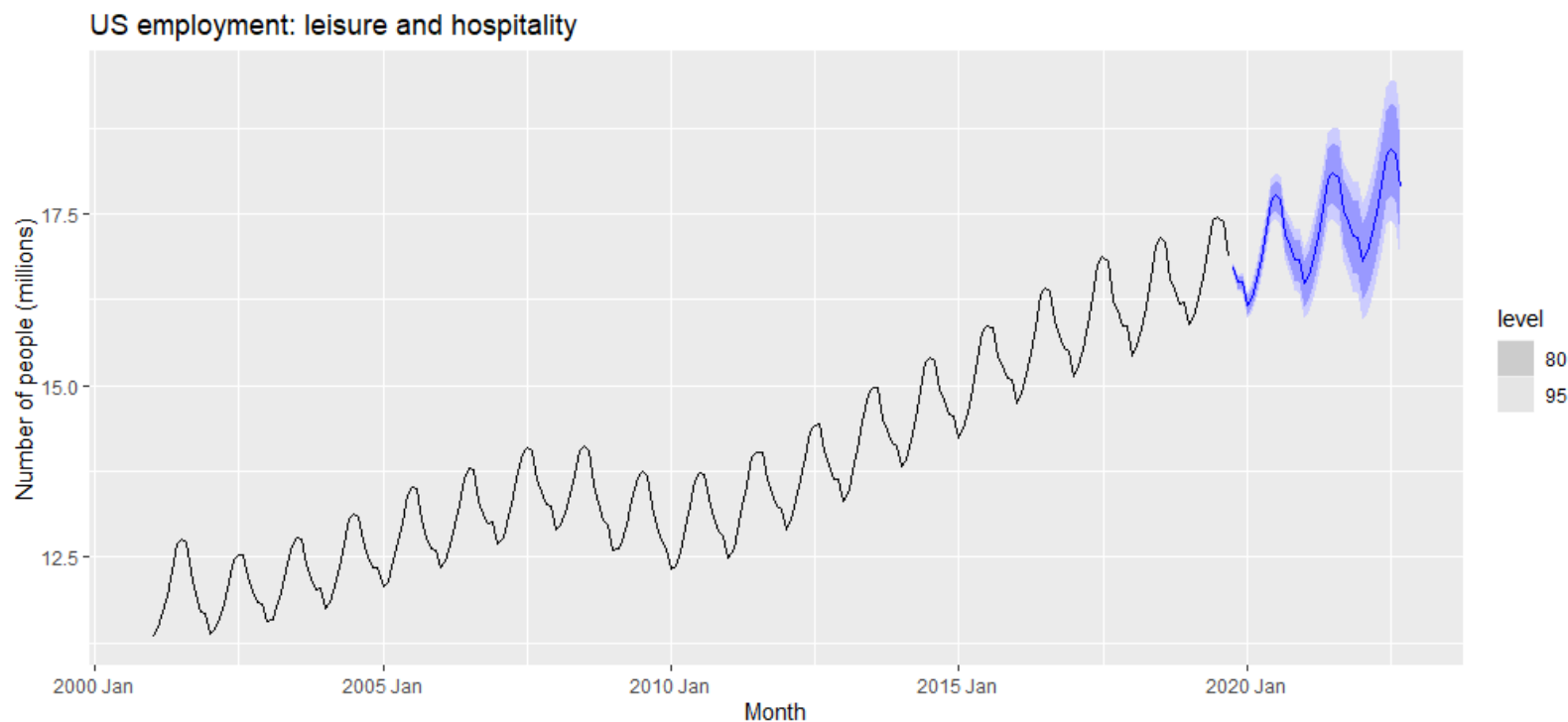
- Using Auto-SARIMA, the winning model is **SARIMA(2,1,0)(1,1,1)₁₂**
- Plotting residuals to confirm they are like Gaussian white noise.





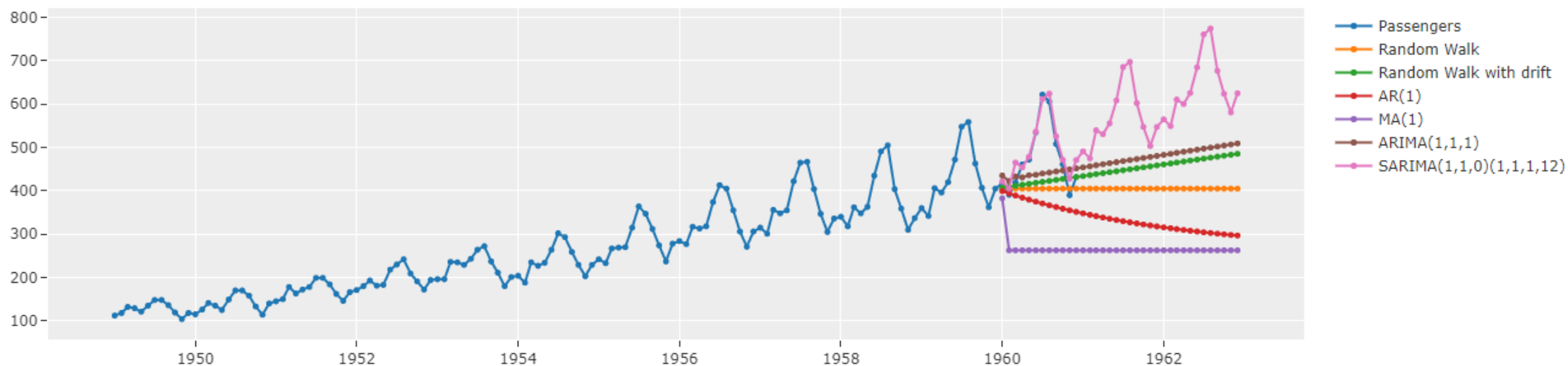
SARIMA example, Forecasting

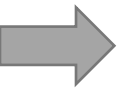
- We now have a seasonal ARIMA model that passes the required checks and is ready for **forecasting**.
- The forecasts have captured the **seasonal** pattern very well, and the increasing **trend** extends the recent pattern. The trend in the forecasts is induced by the double differencing.



➔ Comparing all the models

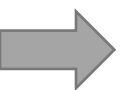
Actual vs. Forecast (Out-of-Sample)



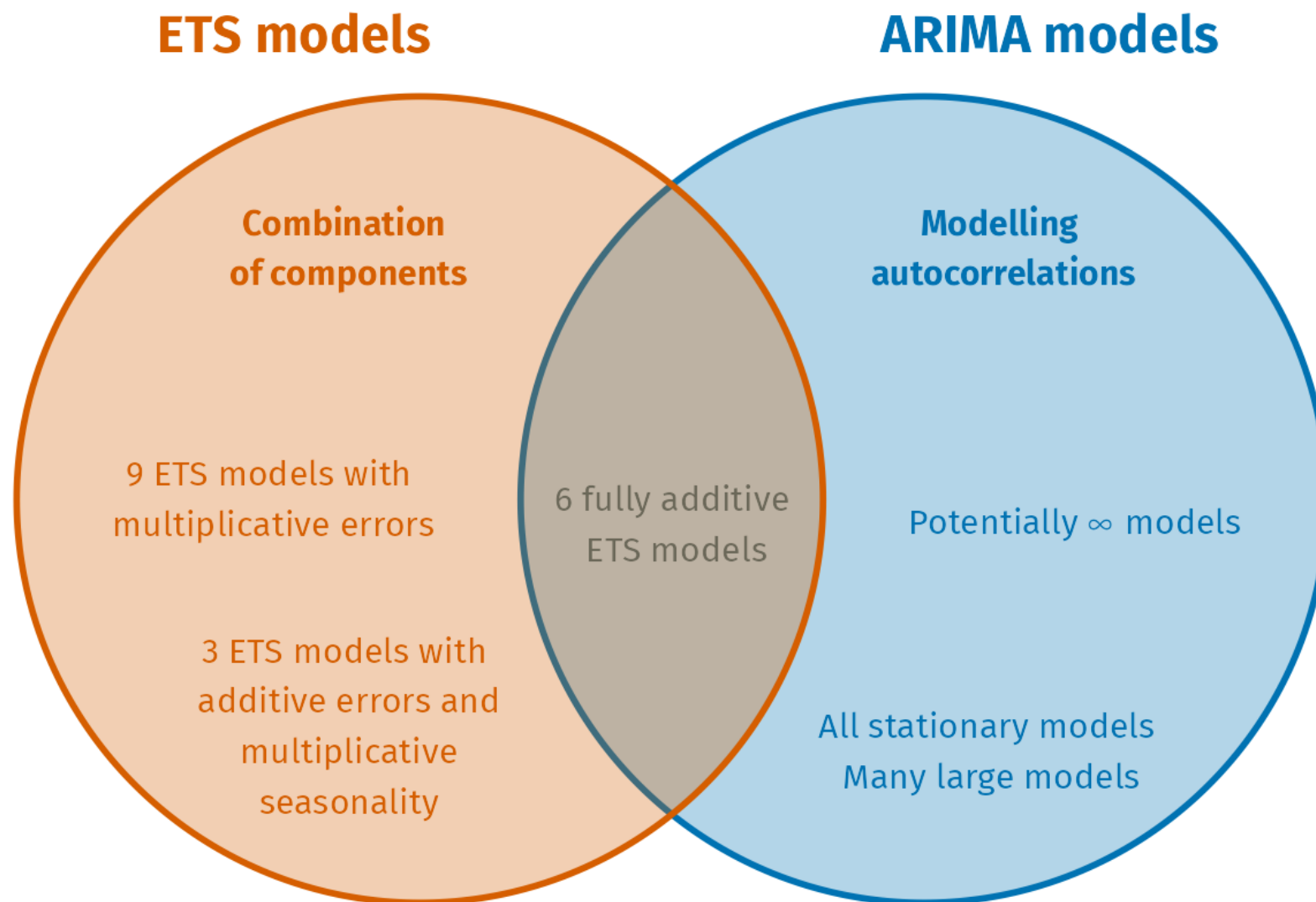


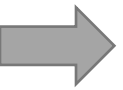
ARIMA vs ETS

- ARIMA and ETS models can be used **together** to enhance forecasting accuracy
- **Modeling Approach:**
 - **ARIMA**: Focuses on describing the **autocorrelations** in the data. It models the time series as a **linear** function of its past values (AR part), the past error terms (MA part), and differences of the series (I part) to ensure stationarity.
 - **ETS**: Models the time series by **explicitly decomposing it into error, trend, and seasonal** components, which can be combined additively or multiplicatively. ETS directly models the series' level, trend, and seasonality.



ARIMA vs ETS





ARIMA vs ETS

- ARIMA and ETS models can be used **together** to enhance forecasting accuracy
- **Data Characteristics:**
 - **ARIMA** is well-suited for time series that can be made stationary through differencing and that have significant autocorrelation patterns but may not have a clear trend or seasonal component.
 - **ETS** excels with time series that have a pronounced trend and/or seasonality.
- In summary, ARIMA and ETS models are **complementary** because they offer different approaches to modeling and forecasting time series data, each with its own set of advantages.
- The choice between them, or the decision to use them together, depends on the specific characteristics of the time series data and the forecasting goals.

➔ Handling Non-Linearities

- **Data Transformation:** Logarithms, Box-Cox, or similar transformations can sometimes linearize **mild non-linear relationships**, making the data suitable for ARIMA or ETS
- Approximating with **ARIMA**: Combining **multiple AR and MA terms** with differencing allows ARIMA models to **partially** capture some forms of non-linear trends or patterns.
- Approximating with **ETS**: selecting between **additive** and **multiplicative** components allows for **some** flexibility in handling non-linear trends and seasonal patterns
- **Limitations of ARIMA and ETS:** For data with strong, complex non-linear relationships, standard ARIMA and ETS models often prove insufficient
- **Transition to Machine Learning:** ML models like tree-based methods (e.g., Random Forests, Gradient Boosting) or neural networks (given enough data) are inherently designed to capture complex non-linear patterns in time series.

➔ Road map!

- ✓ Module 1- Introduction to Deep Forecasting
- ✓ Module 2- Setting up Deep Forecasting Environment
- ✓ Module 3- Exponential Smoothing
- ✓ Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Prophet and Neural Prophet





Module 4 – Part III

RIMA models in Python (Pycaret)



PYCARRET

Actual vs. Forecast (Out-of-Sample)

