Project Write-up of 'Visualization of Autoregressive(AR) Models'

DataWaves - Libin N George, Ajeet Singh, Saharsh Bhave, Navyasree Madhu

Objective

Presenting and visualizing technical topic in statistics, focusing on Autoregressive (AR) models, to demonstrate how parameters affect patterns in time series data.

Introduction

This project aims to explore and visually demonstrate how autoregressive (AR) models operate, specifically how different parameters and AR orders impact the structure and behavior of a time series. AR models are essential tools in time series analysis, with broad applications across economics, finance, weather forecasting, and more. Understanding the visual and statistical characteristics of AR(1), AR(2), and AR(3) models, as well as how different parameter values influence the model's output, provides insight into data pattern recognition and prediction capabilities.

We will use a generated dataset to simulate various AR model scenarios. This approach allows us to precisely control parameters and observe how each setting impacts the time series and how to enable precise, tailored demonstrations of AR model behavior. Our visualizations will feature multiple plots for AR(1), AR(2), and AR(3) models, showcasing stationary and non-stationary processes, cyclical patterns, and convergence to the mean. We will also develop an interactive Shiny app that allows users to work on different AR model parameters and visualize the resulting time series. This approach helps to understand through direct engagement.

Real-World Applications of AR Models

Autoregressive(AR) models are foundational tools in time series analysis and have significant real-world applications across various domains. In finance, they are used to forecast stock

prices, assess risks, and optimize portfolios. In meteorology, AR models aid in weather prediction and climate analysis, while in healthcare, they help track disease outbreaks and forecast hospital resource needs. Industries like energy and telecommunications rely on AR models for demand forecasting and network traffic prediction. These applications highlight the power of AR models in analyzing historical patterns to make informed predictions, enabling better decision-making and resource optimization.

Generic AR Model Equation

The general formula for an auto regressive model of order p , or AR(p), is:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

where:

- X_t is the value of the time series at time t,
- $\phi_1, \phi_2, \ldots, \phi_p$ are the auto regressive coefficients,
- p is the order of the AR model, and
- ϵ_t is a white noise error term at time t, with mean zero and constant variance. ## Partial Autocorrelation Function (PACF)

The Partial Autocorrelation Function (PACF) is a statistical tool used in time series analysis to measure the direct relationship between a variable and its lagged values, after removing the influence of intermediate lags.

Justification of approach

To understand the behavior of AR (AutoRegressive) models under different configurations, we can simulate time series data using the arima.sim() function in R. This function is well-suited for generating synthetic data that follows ARIMA processes, allowing us to study how different AR and MA configurations affect the patterns in a time series. Specifically, we use it to generate data for AR models with varying parameters. In the data analysis plan, we did the following variations in AutoRegressive models and generate data for these models and created visualizations from the data.

1. AR(1) - Random Walk
2. AR(1) - Having moderate positive correlation, decaying toward the mean over time.
3. AR(1) - Data will oscillate, alternating signs and showing a negative correlation
4. AR(2) - Data with oscillatory behavior
5. AR(2) - Data without oscillatory behavior

6. AR(3) - Data with Complex behavior

We created a line graph and associated Autocorrelation graph for the above models to visualize the statistical information and how the data behaves. We have included a shiny app that enables the user to change parameters and plot the line graph.

AR(1) model

Formula:

$$X_t = \phi_1 X_{t-1} + \epsilon_t$$

Key Characteristics:

- The **AR(1)** model defines the current value (X_t) as a function of the **immediately preceding value** (X_{t-1}) and random noise (ϵ_t).
- The behavior of the series heavily depends on the coefficient ϕ_1:

Effect of ϕ_1:

1. $|\phi_1| < 1$**: Stationary Process**

    - The series will show **fluctuations around a mean** and gradually decay toward it.
    - Smaller $|\phi_1|$ results in faster decay, while larger $|\phi_1|$ implies slower decay.

2. $\phi_1 > 0$**: Positive Correlation**

    - The series exhibits a **moderate positive correlation** between successive values.
    - Values persist but eventually decay toward the mean over time.

3. $\phi_1 < 0$**: Oscillatory Behavior**

    - The series alternates in sign, showing **negative correlation** between successive values.
    - Oscillations diminish gradually and decay toward the mean.

4. $|\phi_1| = 1$**: Non-Stationary Process**

    - The series behaves like a **random walk**, where changes are not bound to a mean.
    - Values can drift indefinitely without reverting to a mean.

5. $|\phi_1| > 1$**: Divergent Process**

    - The series is **unstable** and grows without bounds, either positively or negatively.
    - It does not converge to a mean and becomes non-stationary.

```r
# GENERIC CODE FOR CREATING AR MODELS
library(ggplot2)
set.seed(123)
# Function to plot AR(3) process
plot_ar_process <- function(ar_coeffs, n, title_expression,subtitle) {
  # Generate AR process
  ar_process <- as.numeric(arima.sim(model = list(ar = ar_coeffs), n = 2000))

  # Create data frame for plotting
  plot_data <- data.frame(Time = 1:2000, AR_Oscillatory = ar_process)

  # Generate the plot
  plot <- ggplot(subset(plot_data, Time < n), aes(x = Time, y = AR_Oscillatory)) +
    geom_line(color = "#B0CFEA", linewidth=0.8) +
    geom_point(color = "#5D4375", size = 2, shape = 20) +
    labs(
      title = title_expression,
      subtitle = subtitle,
      x = "Time",
      y = "Value"
    ) +
    theme_minimal()
  print(plot)
  # Return the AR process for further use
  return(ar_process)
}

# Function to plot PACF
plot_pacf_process <- function(ar_process, title, subtitle) {
  # Compute PACF
  pacf_data <- pacf(ar_process, plot = FALSE)

  # Convert PACF results to data frame
  pacf_df <- data.frame(
    lag = as.numeric(pacf_data$lag),
    pacf = as.numeric(pacf_data$acf)
  )

  # Generate PACF plot using ggplot
  ggplot(pacf_df, aes(x = lag, y = pacf)) +
    geom_bar(stat = "identity", fill = "blue", color = "black") +
    geom_hline(yintercept = 0, color = "black", linetype = "dashed") +
```

```
    theme_minimal() +
    labs(
      title = title,
      subtitle = subtitle,
      x = "Lag",
      y = "PACF"
    ) +
    theme(
      plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
      axis.title = element_text(size = 14),
      axis.text = element_text(size = 12)
    )
}
```
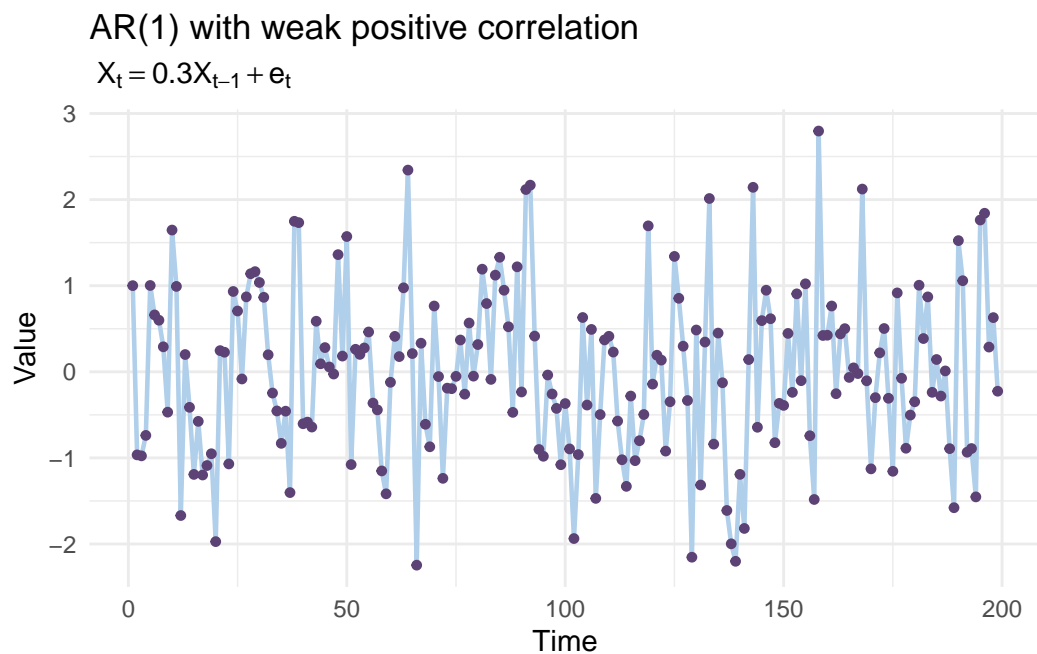
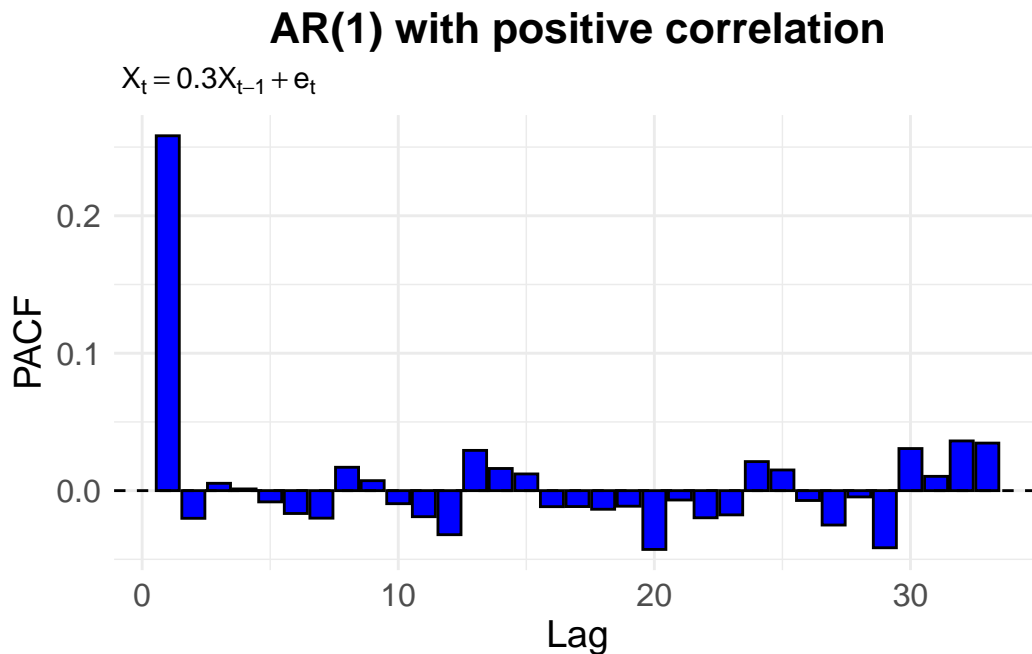## 1. AR(1) with positive coefficients

```
ar_process <- plot_ar_process(
  ar_coeffs = c(0.3), n =200,
  title_expression ="AR(1) with weak positive correlation",
  subtitle=~X[t] == 0.3 * X[t-1] + e[t])
```

AR(1) with weak positive correlation
$X_t = 0.3X_{t-1} + e_t$

The Behavior of the Model Low Coefficient (ϕ=0.3) The value of $\phi = 0.3$ is relatively low, which means the influence of X_{t-1} is moderate. The series will exhibit limited persistence; therefore, the deviations from the mean will fade quickly as the lagged influence diminishes over time. Stationarity The process is stationary because $|\phi| < 1$. This implies that the time series will fluctuate around a constant mean (0 for this example). The variance of the series remains finite and does not grow over time. White Noise Contribution Random white noise (ϵ_t) plays a more significant role in shaping the series than lagged values. This results in a more stochastic (less deterministic) series. Smoothness and Short-Term Correlation The small ϕ value results in limited smoothness; is only weakly correlated with X_{t-1}. Short-term trends are minimal, and the series appears more random. Mean-Reverting Property The series will revert to the mean quickly after a deviation because the influence of the lagged term diminishes rapidly.
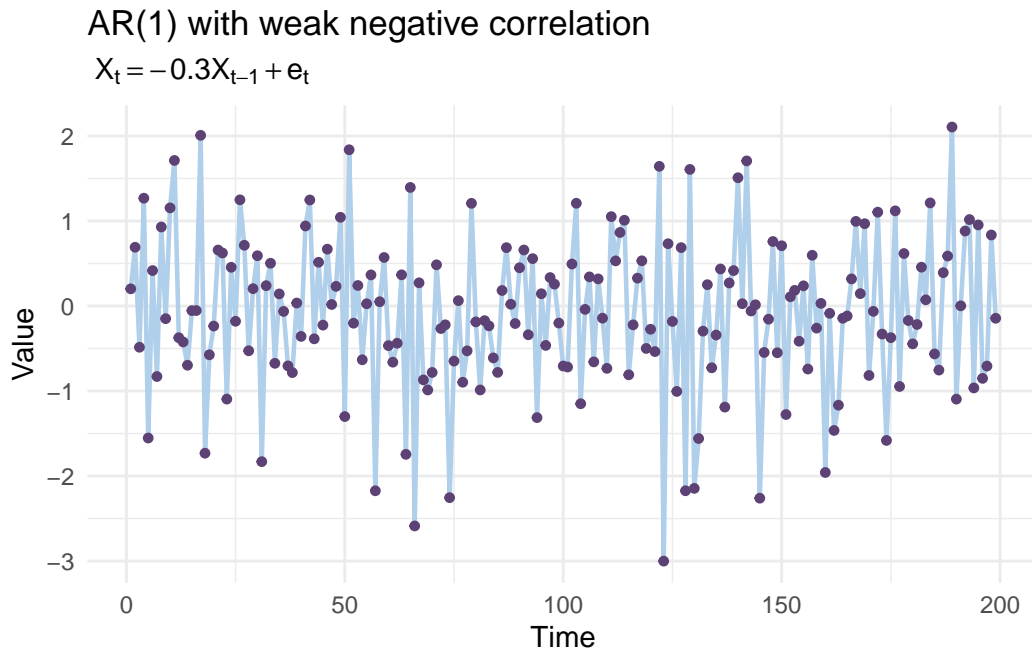
```
plot_pacf_process(ar_process,
                  title ="AR(1) with positive correlation",
                  subtitle=~X[t] == 0.3 * X[t-1] + e[t])
```



We can clearly see dominance of 1 lag term here.

## 2. AR(1) with negative coefficients

```
ar_process <- plot_ar_process(
  ar_coeffs = c(-0.3), n =200,
  title_expression ="AR(1) with weak negative correlation",
  subtitle=~X[t] == -0.3 * X[t-1] + e[t])
```

AR(1) with weak negative correlation
$X_t = -0.3X_{t-1} + e_t$

The Behavior of the Model Low Negative Coefficient ($\phi=-0.3$) A small negative coefficient indicates: A weak negative correlation between X_t and X_{t-1}. X_t tends to move in the opposite direction to X_{t-1} but with low persistence. Oscillatory Behavior The negative coefficient introduces an alternating pattern: If X_{t-1} is positive, X_t is more likely to be negative, and vice versa. However, the oscillations are not pronounced due to the small magnitude of ϕ. Stationarity The process is stationary because $\phi < 1$. This ensures the series fluctuates around a constant mean (0 for this example). The variance remains stable over time, preventing explosive growth or decay. Randomness and White Noise Random white noise (ϵ_t) dominates the series: The weak ϕ value means that the current value is heavily influenced by random noise rather than the lagged term. Limited Persistence The impact of deviations from the mean dissipates quickly because ϕ is small. The series lacks strong trends or long-term dependencies.

```
plot_pacf_process(ar_process,
                  title ="AR(1) with positive correlation",
                  subtitle=~X[t] == -0.3 * X[t-1] + e[t])
```

AR(1) with positive correlation

$$X_t = -0.3X_{t-1} + e_t$$
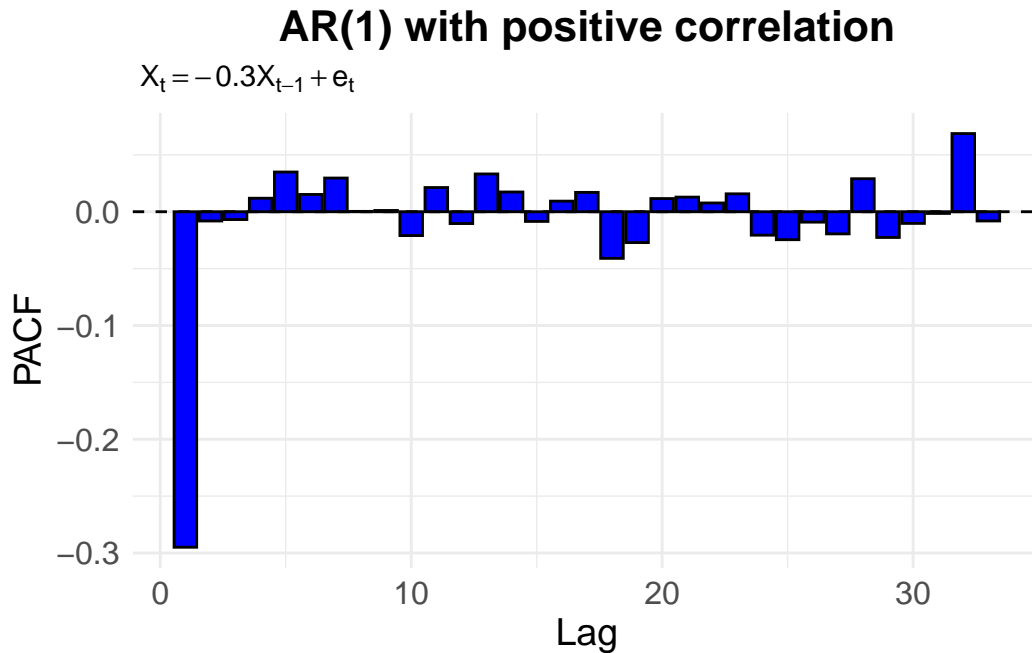
We can see significant negative correlation for 1st lag term.

3. Random Walk

```
set.seed(123)
ar_process <- arima.sim(model = list(order = c(0, 1, 0)), n = 200)

#Create a data frame
plot_data <- data.frame(Time = 1:201, Value = ar_process)

# Plotting the random walk (ARIMA(0,1,0))
ggplot(plot_data, aes(x = Time, y = Value)) +
  geom_line(color = "#B0CFEA", linewidth=0.8) +
  geom_point(color = "#5D4375", size = 2, shape = 20) +
  labs(title = "Simulated Random Walk",
       subtitle = ~  X[t] == X[t-1] + e[t],
       x = "Time",
       y = "Value") +
  theme_minimal()
```
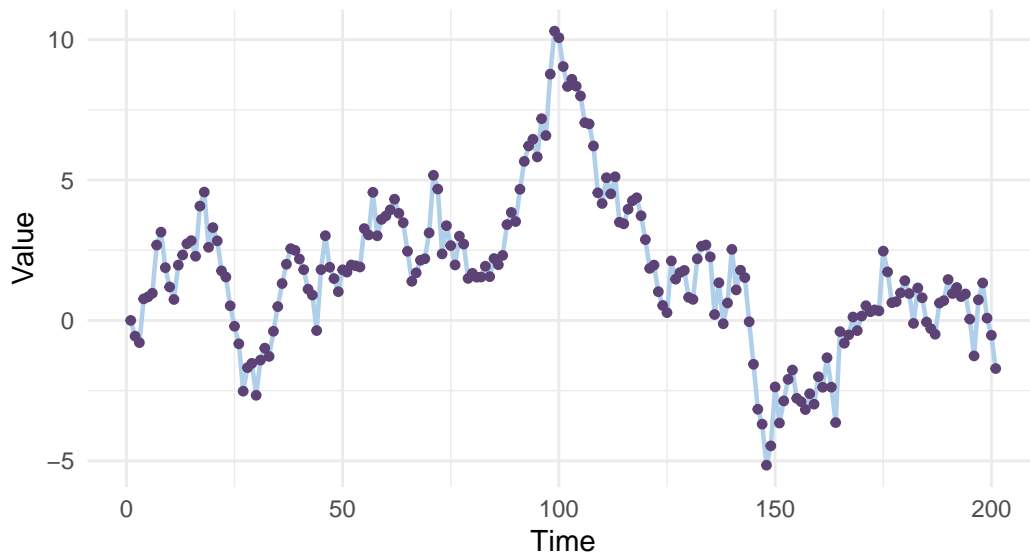
Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

8

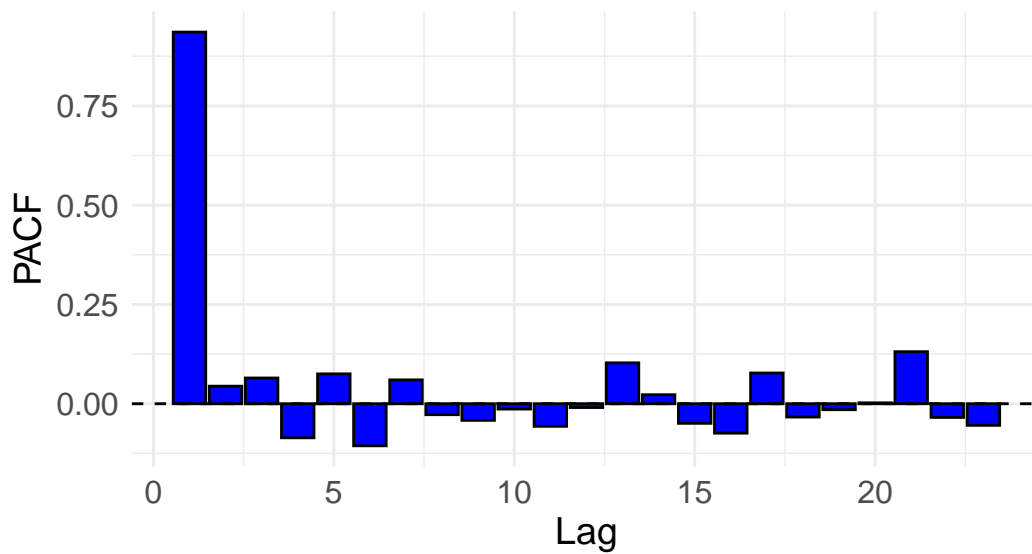Simulated Random Walk

$X_t = X_{t-1} + e_t$


```
plot_pacf_process(ar_process,
                  title ="Simulated Random Walk",
                  subtitle=~X[t] ==  X[t-1] + e[t])
```

Simulated Random Walk

$X_t = X_{t-1} + e_t$

We can see string correlatin for first lag term in above PACF graph.

4. White noise

When $\phi_1 = 0$ in an AR(1) process, the series is driven entirely by white noise, with no dependence on past values. This results in a stationary process with constant mean and variance, and no autocorrelation or predictable patterns.

```
white_noise <- rnorm(200)
df <- data.frame(Time = 1:200, Value = white_noise)

white_noise <- rnorm(1000)

# Plot using ggplot
ggplot(df, aes(x = Time, y = Value)) +
  geom_line(color = "#B0CFEA", linewidth=0.8) +
  geom_point(color = "#5D4375", size = 2, shape = 20) +
  labs(title = "White Noise", x = "Time", y = "Value") +
  theme_minimal()
```
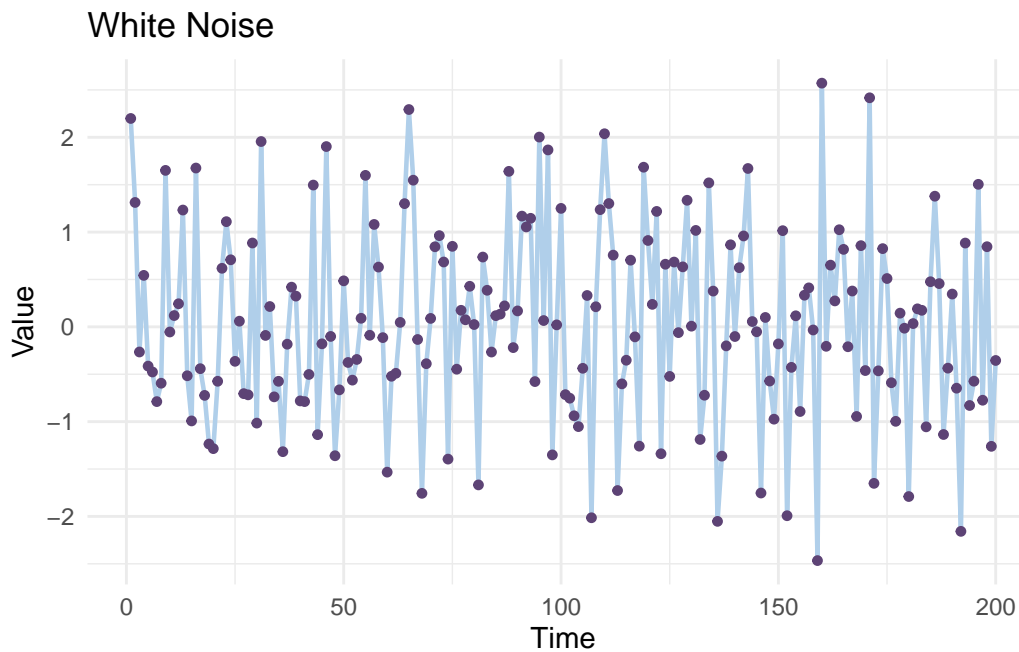
AR(2) Model

- Formula: $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \epsilon_t$
- More complex dynamics emerge as the series now depends on the previous two points.
- Effect of ϕ_1 and ϕ_2:
- Oscillations: If $\phi_2 < 0$, the series may display oscillatory or cyclical behavior.
- Damping: If both coefficients are positive but less than 1, the series will dampen oscillations toward the mean.
- Explosive behavior: Larger values of ϕ_1 and ϕ_2 lead to explosive growth in the series.

AR(2) positive coeifficient

```
ar_process <- plot_ar_process(
  ar_coeffs = c(0.4, 0.1), n =200,
  title_expression ="AR(2) without oscillatory behavior",
  subtitle=~Y[t] == 0.4 * Y[t-1] + 0.1 * Y[t-2] + e[t])
```

### AR(2) without oscillatory behavior
$Y_t = 0.4Y_{t-1} + 0.1Y_{t-2} + e_t$
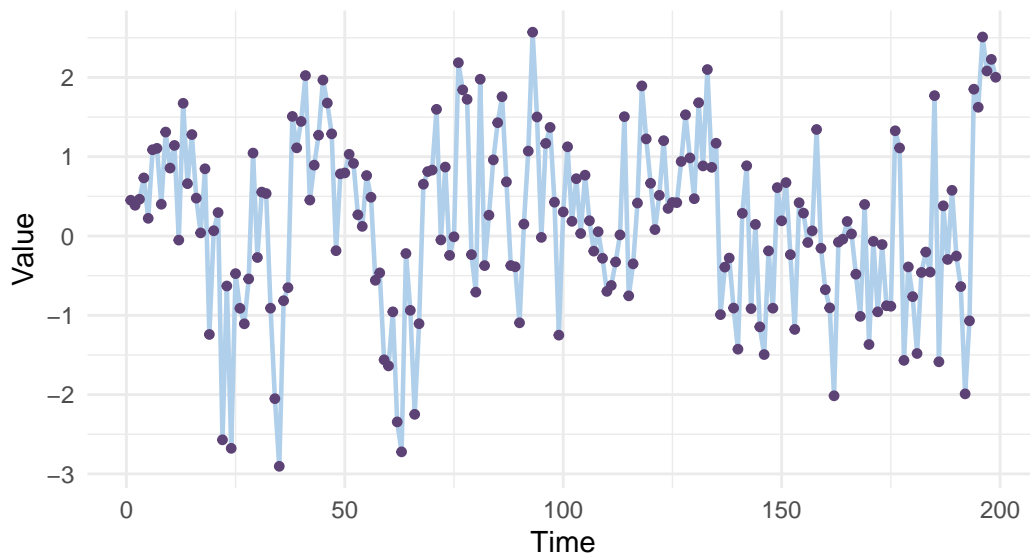
Equation:

$$Y_t = 0.4 \cdot Y_{t-1} + 0.1 \cdot Y_{t-2} + e_t$$

Key Observations:

- **Low coefficients (0.4, 0.1):**
  - These relatively small coefficients result in **weak persistence**, meaning the current value of Y_t depends only moderately on the previous two values.
  - Changes in Y_t are largely driven by random noise (e_t) rather than strong autoregressive patterns.

- **Behavior:**
  - The process is **stable** with no oscillatory behavior, as both coefficients are positive and their sum $(0.4 + 0.1 = 0.5)$ is less than 1.
  - The values show gradual variations around the mean but return to the mean quickly after small deviations.
  - This process exhibits **weak memory**, meaning the influence of past values fades quickly.

```
plot_pacf_process(ar_process, "APCF: AR(2)", expression(~
                          X[t] == 0.4 * X[t-1] + 0.1 * X[t-2] + e[t]))
```

APCF: AR(2)

$X_t = 0.4X_{t-1} + 0.1X_{t-2} + e_t$
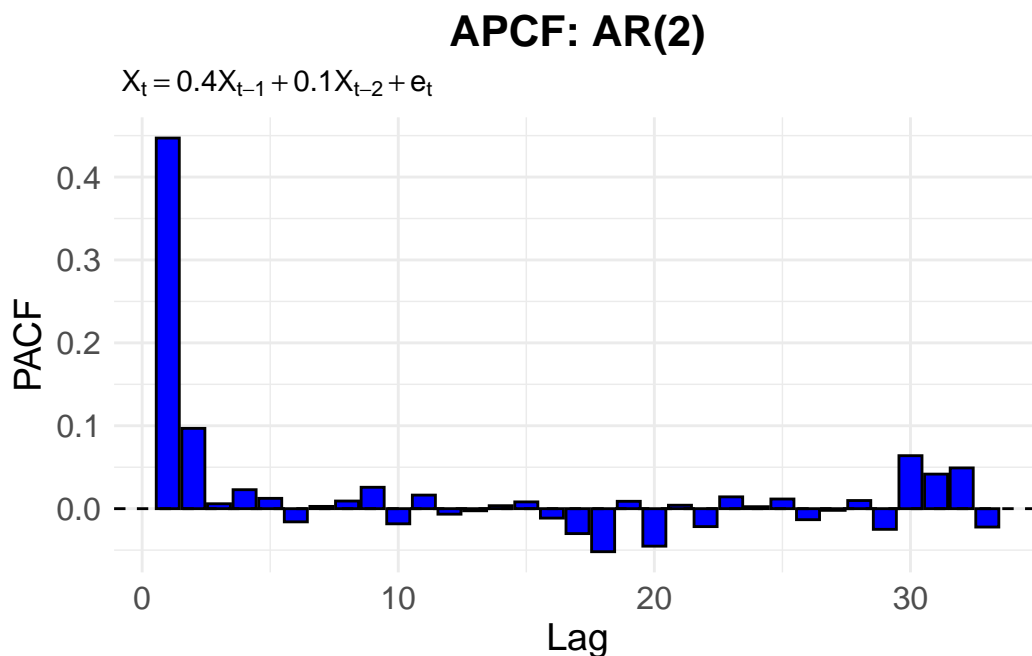
- The Partial Autocorrelation Function (PACF) shows significant spikes at lags 1 and 2, corresponding to the direct dependence of X_t on X_{t-1} and X_{t-2}, respectively. - Beyond lag 2, the PACF values drop close to zero, indicating that higher-order lags have minimal direct influence on X_t. We can see occasional spikes, but this wll go away of we use more lengthy series.

AR-2 with oscillation

,

```
ar_process <- plot_ar_process(
  ar_coeffs = c(0.4, -0.9), n =200,
  title_expression ="AR(2) with oscillatory behavior",
  subtitle=~Y[t] == 0.4 * Y[t-1] - 0.9 * Y[t-2] + e[t])
```

AR(2) with oscillatory behavior
$Y_t = 0.4Y_{t-1} - 0.9Y_{t-2} + e_t$
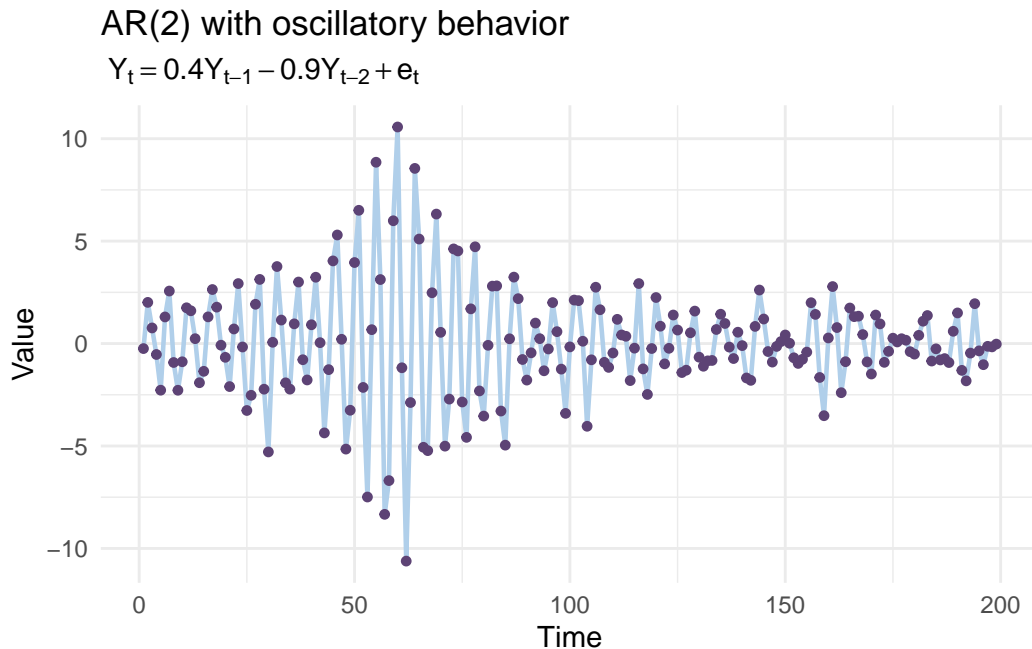
- **Negative Coefficients**: The coefficients indicate that X_t is positively influenced by X_{t-1} ($\phi_1 = 0.4$) and negatively influenced by X_{t-2} ($\phi_2 = -0.9$). This means X_t moves in the same direction as X_{t-1} but in the opposite direction of X_{t-2}. The influence of X_{t-2} is stronger due to the larger magnitude of its negative coefficient.

- **Oscillatory Behavior**: The combination of positive and negative coefficients leads to oscillations in the series. When X_{t-1} is negative and X_{t-2} is also positive, X_t will be negative, etc. The oscillations are relatively weak but still noticeable, especially due to the higher magnitude of ϕ_2.

- **Stationarity**: The process remains stationary as long as the roots of the characteristic equation (determined by the AR coefficients) lie outside the unit circle. For this AR(2) model with $\phi_1 = 0.4$ and $\phi_2 = -0.9$, the process will be stationary because the magnitude of both coefficients is less than 1.

- **White Noise Contribution**: The noise term e_t introduces random variations in the series. Despite the autoregressive structure, the white noise remains an important factor in influencing the stochastic behavior of the series.

```
plot_pacf_process(ar_process,
                  title ="AR(2) with oscillatory behavior",
                  subtitle=~Y[t] == 0.4 * Y[t-1] - 0.9 * Y[t-2] + e[t])
```
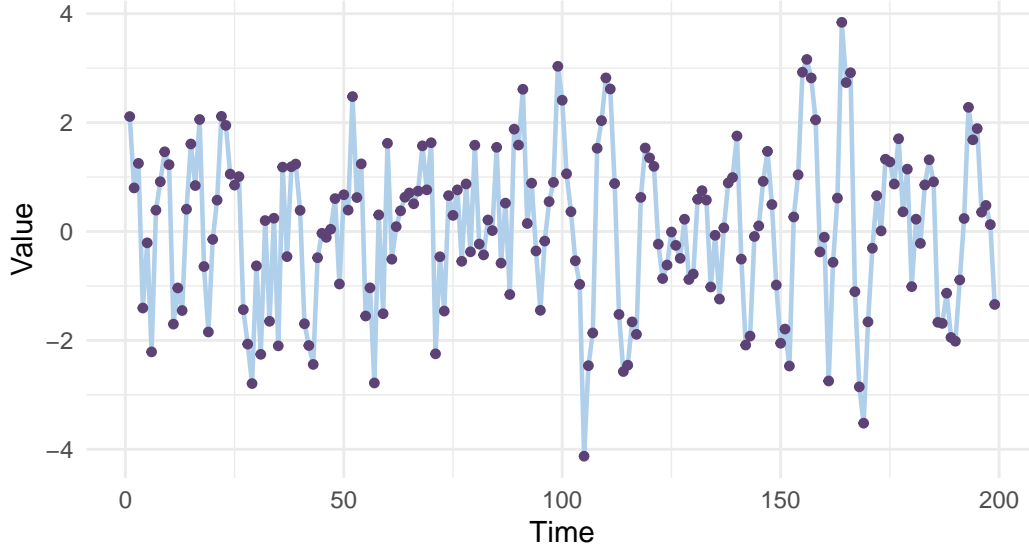
AR(2) with oscillatory behavior

$$Y_t = 0.4Y_{t-1} - 0.9Y_{t-2} + e_t$$
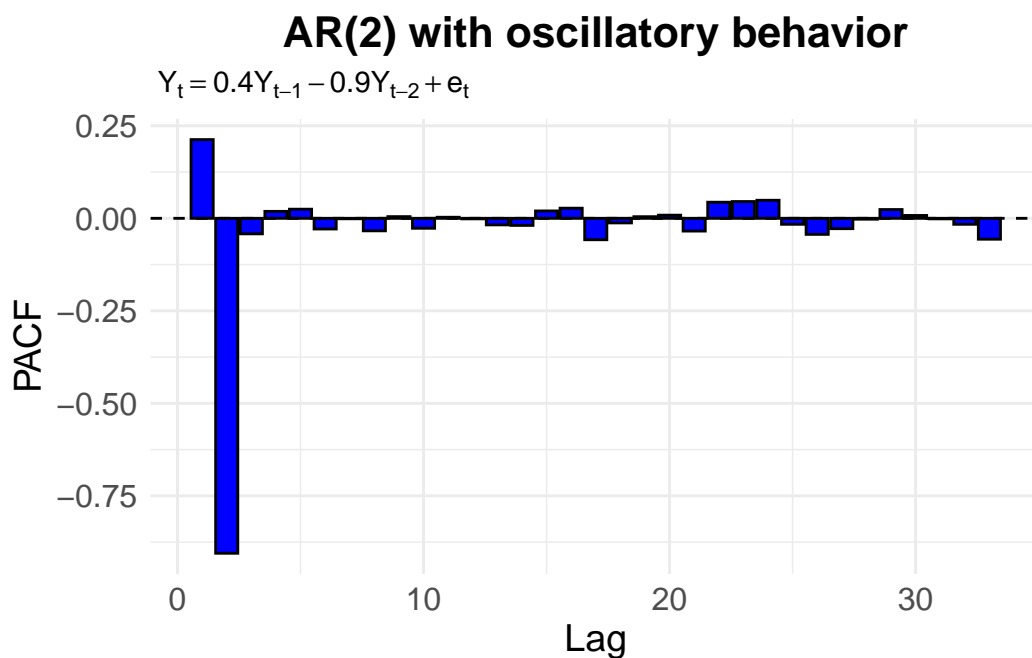
We can clearly see the two lag(X_{t-1} and X_{t-2}) dependencies in above plot. # AR(3) model with complex cycle

```
ar_process <- plot_ar_process(
  ar_coeffs = c(0.5, 0.3, -0.6), n =200,
  title_expression ="AR(3) without Complex cycles",
  subtitle=~X[t] == 0.4 * X[t-1] + 0.3 * X[t-2] - 0.6 * X[t-3]+ e[t])
```

14

AR(3) without Complex cycles

$X_t = 0.4X_{t-1} + 0.3X_{t-2} - 0.6X_{t-3} + e_t$

The Behavior of the Model Positive and Negative Coefficients The AR(3) model has a combination of both positive and negative coefficients. $\phi_1 = 0.5$: The current value X_t has a positive dependence on the previous value X_{t-1}, meaning that if X_{t-1} is large, X_t will likely also be large. $\phi_2 = 0.3$: The current value X_t has a moderate positive dependence on X_{t-2} but is weaker than X_{t-1}. $\phi_3 = -0.6$: The current value X_t has a negative dependence on X_{t-3}, meaning that if X_{t-3} is large, X_t will likely be smaller. Oscillatory Behavior The process exhibits oscillations when it has positive and negative coefficients. Positive coefficients contribute to values moving in the same direction, while negative coefficients introduce reversals. This can lead to a complex, oscillatory pattern, where the influence of earlier terms (both positive and negative) generates alternating cycles. The size of the coefficients determines the amplitude of these oscillations. Large positive coefficients (like $\phi_1 = 0.5$) cause the series to follow the previous term closely, while the negative coefficient $\phi_3 = -0.6$ creates some counteracting force. Stationarity As long as the roots of the characteristic equation (derived from the AR coefficients) are outside the unit circle, the process is stationary. Given that the coefficients here are moderate in size, this process is stationary, meaning that the mean and variance will remain constant over time. Impact of White Noise The white noise term t adds randomness to the model, making the series unpredictable and variable. While the autoregressive terms provide structure, the random noise plays a major role in shaping the series, especially when the influence of past values is not very strong. Persistence of Shocks Because of the combination of positive and negative coefficients, shocks (deviations from the mean) may persist longer than in models with only one or two coefficients, as multiple lags influence the series. However, the persistence of shocks is still limited by the relatively small size of the negative coefficient ($\phi_3 = -0.6$).

```
plot_pacf_process(ar_process, "APCF: AR(3)",
            subtitle=~X[t] == 0.4 * X[t-1] + 0.3 * X[t-2] - 0.7 * X[t-3]+ e[t])
```

APCF: AR(3)

$$X_t = 0.4X_{t-1} + 0.3X_{t-2} - 0.7X_{t-3} + e_t$$
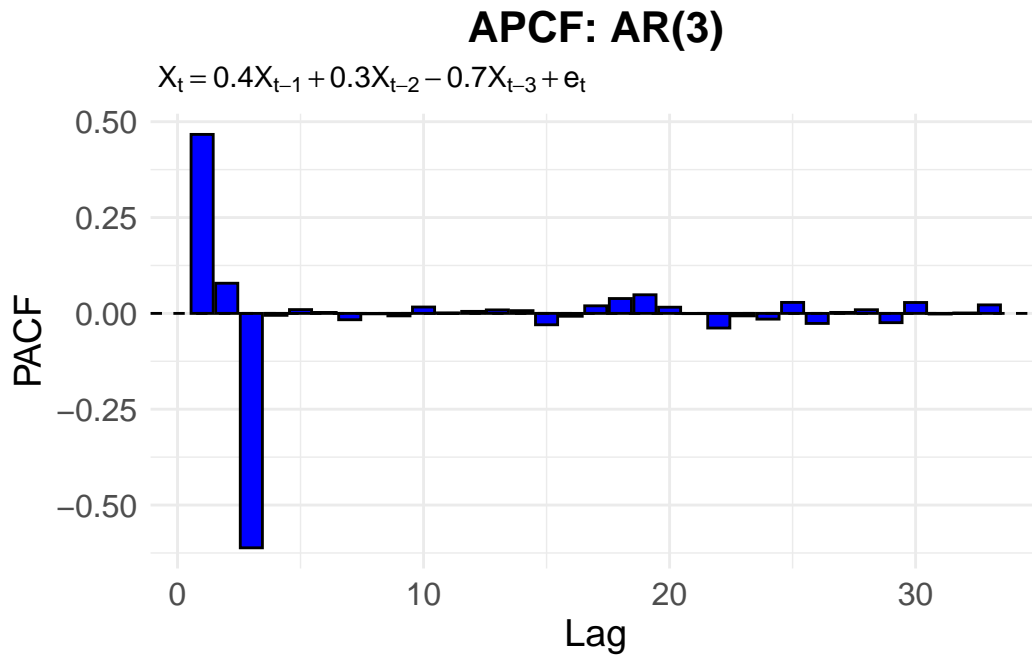
Interactive Shiny App:

An interactive Shiny app was developed to allow users to manipulate AR parameters and observe their real-time effects. User can select one of the 3 AR models and its variations discussed in the document and change its parameters, They can also create higher order AR models, but this will require the coefficients of MOdel to result in stationary series.

Discussion and Conclusions

This project is a demonstration of the characteristics and behavior of autoregressive (AR) models through simulations and visualizations. The insights are relayed to AR(1), AR(2), and AR(3) models under varying parameters. For AR(1) models, positive coefficients, such as $\phi = 0.5$, result in moderate positive correlations with values decaying toward the mean, while negative coefficients, such as $\phi = -0.5$, introduce oscillatory behavior with alternating signs. Both cases maintain stationarity as long as $|\phi| < 1$. AR(2) models exhibit more complex dynamics due to including a second lag term. Oscillatory patterns emerge when one coefficient is negative, as seen with $\phi_1 = 1.4$ and $\phi_2 = -0.9$, while positive coefficients

16

produce dampened oscillations. In AR(3) models, the interplay of three lag terms creates intricate cyclical behavior, with positive and negative coefficients driving alternating cycles and persistence influenced by the magnitude of these coefficients. The model white noise plays a crucial role in introducing randomness and variability, while the AR terms define structure. We have introduced an interactive Shiny app, which allows users to manipulate AR parameters and observe their effects in real-time. In Essence, any cyclical patterns that arise due to past data can be represented by higher order AR models.

Future Works:

In future work, it incorporatesis possible to real-world datasets and expands the analysis to mixed ARIMA models or higher-order processes. This study provides a comprehensive and engaging exploration of AR models, highlighting their versatility in time series analysis and real-world applications. We can also take a real world example to show how a sesonal data fits into an arima model.