Week 10 Solutions - AR1 Process Analysis

Assistant Solution

2024-11-18

1 Problem 1: Auto regressive process

Consider the AR(1) process defined as: Yt = phi * Y(t-1) + et, where et ~ N(0, sigma2), phi in (0,1), Y1 ~ N(0, sigma2)

1.1 Model Parameters

```r
params <- data.frame(
  Parameter = c("phi", "sigma2", "n", "Initial Distribution"),
  Value = c("0.6", "3", "60", "N(0, sigma2)"),
  Description = c(
    "Auto-regression coefficient",
    "Innovation variance",
    "Time series length",
    "Initial value distribution"
  )
)

kable(params,
      caption = "Model Parameters",
      align = c('l', 'c', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                position = "center",
                full_width = FALSE)
```

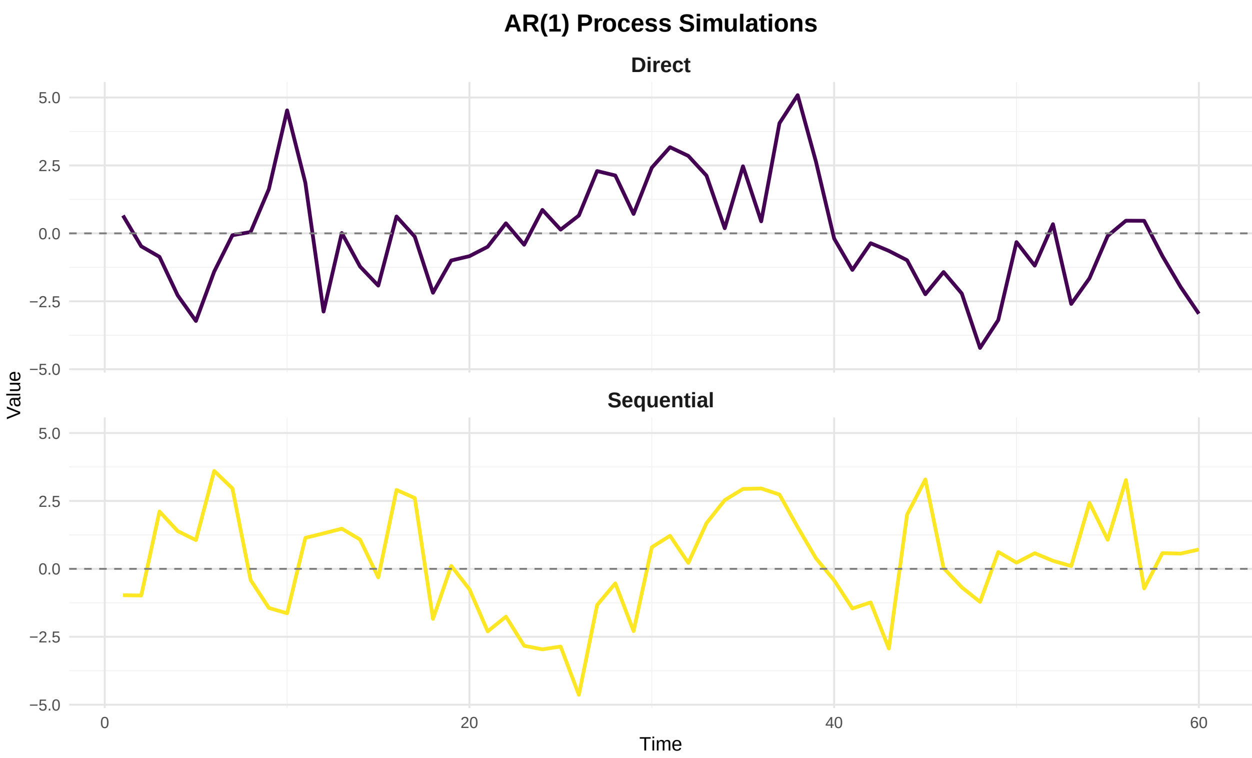
Table 1: Model Parameters

Parameter	Value	Description
phi	0.6	Auto-regression coefficient
sigma2	3	Innovation variance
n	60	Time series length
Initial Distribution	N(0, sigma2)	Initial value distribution

1.2 (a) AR(1) Process Simulation

```r
# Set random seed for reproducibility
set.seed(123)

# Parameters
n <- 60
phi <- 0.6
sigma2 <- 3

# Method 1: Sequential simulation
ar1_sequential <- numeric(n)
ar1_sequential[1] <- rnorm(1, 0, sqrt(sigma2))
for(t in 2:n) {
  ar1_sequential[t] <- phi * ar1_sequential[t-1] + rnorm(1, 0, sqrt(sigma2))
}

# Method 2: Direct simulation using matrix multiplication
epsilon <- rnorm(n, 0, sqrt(sigma2))
phi_matrix <- matrix(0, n, n)
for(i in 1:n) {
  for(j in 1:i) {
    phi_matrix[i,j] <- phi^(i-j)
  }
}
ar1_direct <- phi_matrix %*% epsilon

# Create data frames for ggplot
df_sequential <- data.frame(
  Time = 1:n,
  Value = ar1_sequential,
  Method = "Sequential"
)

df_direct <- data.frame(
  Time = 1:n,
  Value = ar1_direct,
  Method = "Direct"
)

df_combined <- rbind(df_sequential, df_direct)

# Create enhanced plot
ggplot(df_combined, aes(x = Time, y = Value, color = Method)) +
  geom_line(size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50") +
  facet_wrap(~Method, ncol = 1) +
  scale_color_viridis(discrete = TRUE) +
  theme_minimal() +
  theme(
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_line(color = "gray95"),
    strip.text = element_text(size = 12, face = "bold"),
```

```r
    legend.position = "none",
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  labs(title = "AR(1) Process Simulations",
       x = "Time",
       y = "Value")
```



1.2.1 Parameter Effects Summary

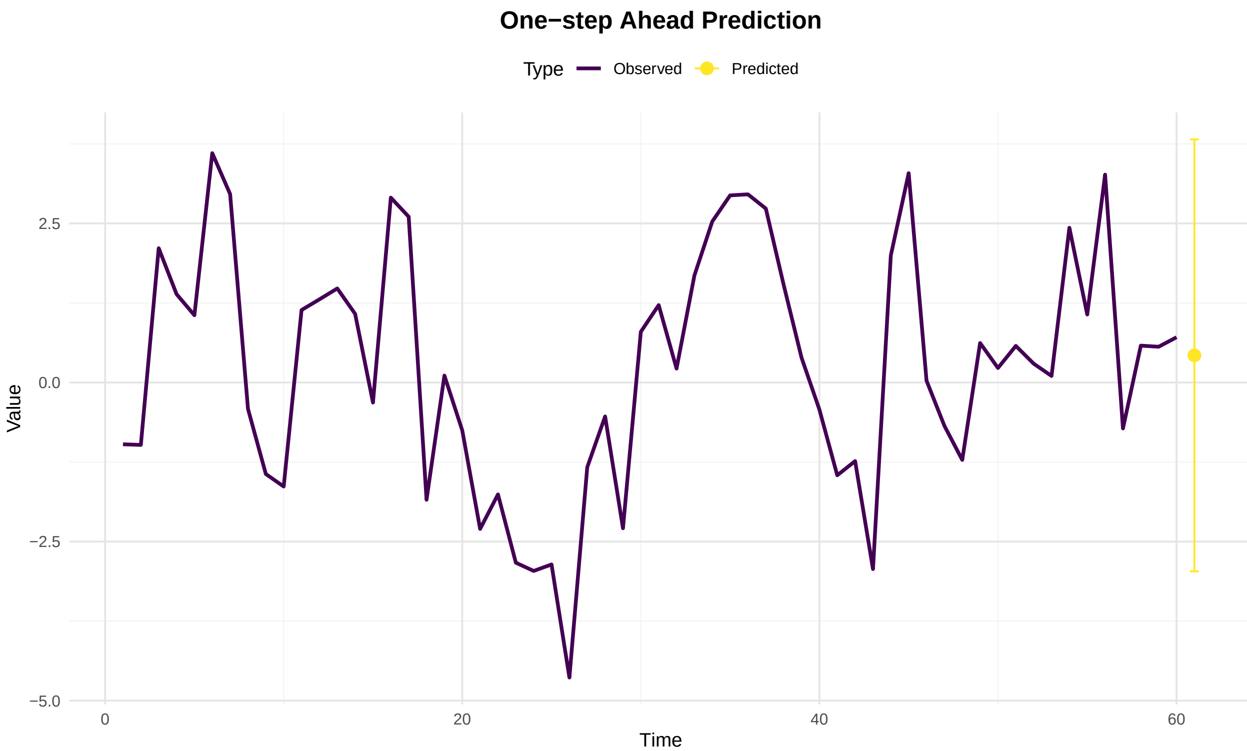
```r
effects <- data.frame(
  Parameter = c("Increasing sigma2", "Increasing phi", "Decreasing phi"),
  Effect = c(
    "Increases process variance and volatility",
    "Creates more persistent/smoother process (phi -> 1)",
    "Creates more random/noisy process (phi -> 0)"
  )
)

kable(effects,
      caption = "Effects of Parameter Changes",
      align = c('l', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                position = "center",
                full_width = FALSE)
```

Table 2: Effects of Parameter Changes

Parameter	Effect
Increasing sigma2	Increases process variance and volatility
Increasing phi	Creates more persistent/smoother process (phi -> 1)
Decreasing phi	Creates more random/noisy process (phi -> 0)

1.3 (b) One-step ahead prediction

```r
# Predict Y61
y61_pred <- phi * ar1_sequential[n]
ci_width <- qnorm(0.975) * sqrt(sigma2)
y61_ci <- c(y61_pred - ci_width, y61_pred + ci_width)

# Create prediction plot
df_pred <- data.frame(
  Time = c(1:n, n+1),
  Value = c(ar1_sequential, y61_pred),
  Type = c(rep("Observed", n), "Predicted")
)

ggplot(df_pred, aes(x = Time, y = Value, color = Type)) +
  geom_line(data = subset(df_pred, Type == "Observed"), size = 1) +
  geom_point(data = subset(df_pred, Type == "Predicted"), size = 3) +
  geom_errorbar(data = subset(df_pred, Type == "Predicted"),
                aes(ymin = y61_ci[1], ymax = y61_ci[2]),
                width = 0.5) +
  scale_color_manual(values = c("Observed" = "#440154", "Predicted" = "#FDE725")) +
  theme_minimal() +
  theme(
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_line(color = "gray95"),
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  labs(title = "One-step Ahead Prediction",
       x = "Time",
       y = "Value")
```




```r
# Results table
pred_results <- data.frame(
  Metric = c("Point Prediction", "Lower 95% CI", "Upper 95% CI"),
  Value = round(c(y61_pred, y61_ci[1], y61_ci[2]), 3)
)

kable(pred_results,
      caption = "One-step Ahead Prediction (Y61)",
      align = c('l', 'r'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                position = "center",
                full_width = FALSE)
```

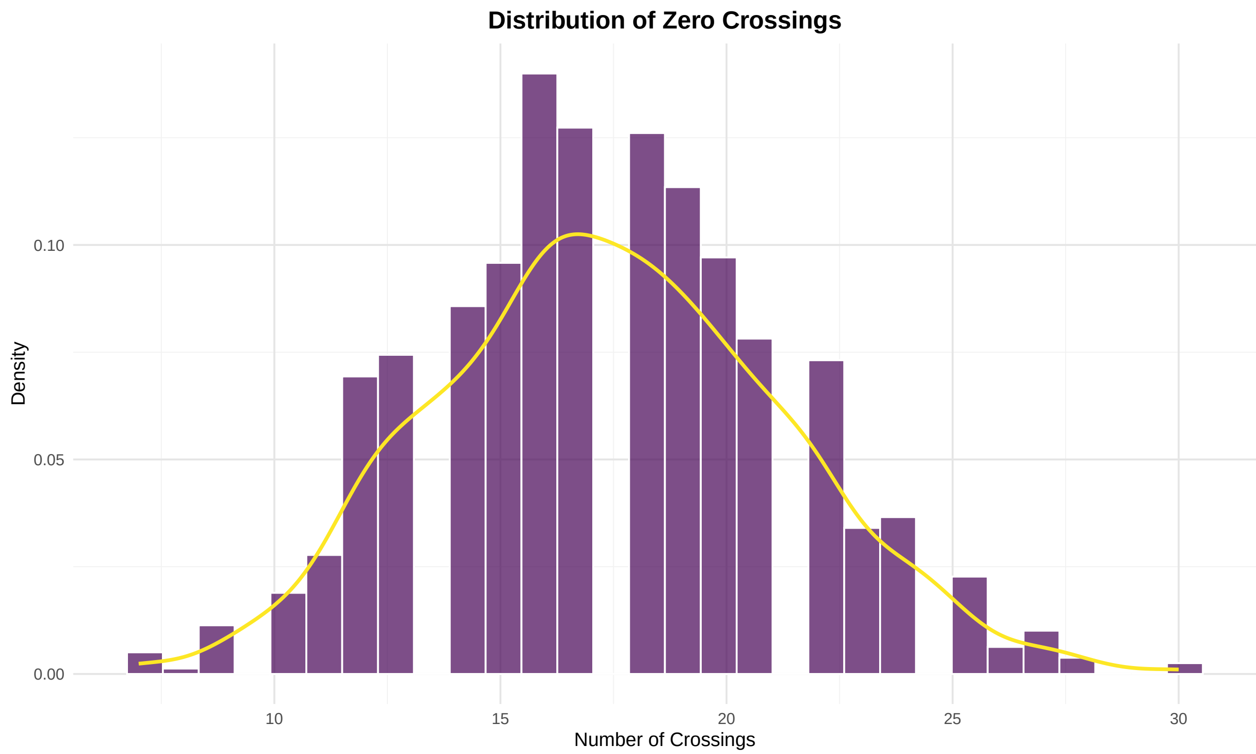
Table 3: One-step Ahead Prediction (Y61)

Metric	Value
Point Prediction	0.427
Lower 95% CI	-2.968
Upper 95% CI	3.822

1.4 (d) Zero crossings simulation

```r
# Function to count zero crossings
count_crossings <- function(series) {
  sum(series[-1] * series[-length(series)] < 0)
```

```r
}

# Simulate multiple series and count crossings
n_sims <- 1000
crossings <- numeric(n_sims)

for(i in 1:n_sims) {
  y <- numeric(n)
  y[1] <- rnorm(1, 0, sqrt(sigma2))
  for(t in 2:n) {
    y[t] <- phi * y[t-1] + rnorm(1, 0, sqrt(sigma2))
  }
  crossings[i] <- count_crossings(y)
}

# Create enhanced histogram
df_crossings <- data.frame(crossings = crossings)

ggplot(df_crossings, aes(x = crossings)) +
  geom_histogram(aes(y = ..density..),
                 fill = "#440154",
                 color = "white",
                 bins = 30,
                 alpha = 0.7) +
  geom_density(color = "#FDE725", size = 1) +
  theme_minimal() +
  theme(
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_line(color = "gray95"),
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  labs(title = "Distribution of Zero Crossings",
       x = "Number of Crossings",
       y = "Density")
```



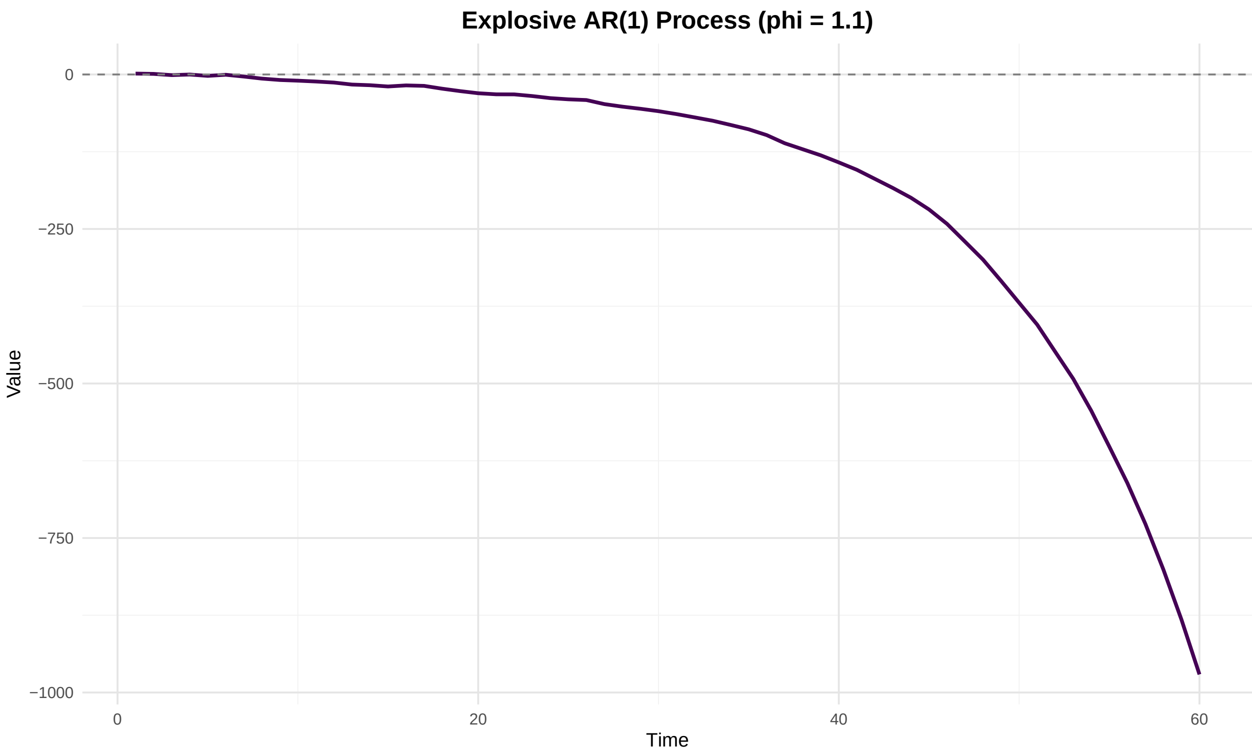

```r
# Summary statistics
crossing_stats <- data.frame(
  Statistic = c("Mean", "Median", "SD", "Min", "Max"),
  Value = round(c(mean(crossings), median(crossings),
                  sd(crossings), min(crossings), max(crossings)), 2)
)

kable(crossing_stats,
      caption = "Zero Crossings Statistics",
      align = c('l', 'r'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                position = "center",
                full_width = FALSE)
```

Table 4: Zero Crossings Statistics

Statistic	Value
Mean	17.41
Median	17.00
SD	3.85
Min	7.00
Max	30.00

1.5 (e) Explosive AR process

```r
# Parameters for explosive AR
phi_explosive <- 1.1
n <- 60

# Simulate explosive process
ar_explosive <- numeric(n)
ar_explosive[1] <- rnorm(1, 0, sqrt(sigma2))

for(t in 2:n) {
  ar_explosive[t] <- phi_explosive * ar_explosive[t-1] + rnorm(1, 0, sqrt(sigma2))
}

# Create enhanced plot
df_explosive <- data.frame(
  Time = 1:n,
  Value = ar_explosive
)

ggplot(df_explosive, aes(x = Time, y = Value)) +
  geom_line(color = "#440154", size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50") +
  theme_minimal() +
  theme(
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_line(color = "gray95"),
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  labs(title = "Explosive AR(1) Process (phi = 1.1)",
       x = "Time",
       y = "Value")
```




```r
# Process characteristics
explosive_chars <- data.frame(
  Characteristic = c("Process Type", "Behavior", "Stationarity", "Long-term Distribution"),
  Description = c(
    "Explosive AR(1)",
    "Exponential growth",
    "Non-stationary",
    "No stable distribution"
  )
)

kable(explosive_chars,
      caption = "Characteristics of Explosive AR(1) Process",
      align = c('l', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                position = "center",
                full_width = FALSE)
```

Table 5: Characteristics of Explosive AR(1) Process

Characteristic	Description
Process Type	Explosive AR(1)
Behavior	Exponential growth
Stationarity	Non-stationary
Long-term Distribution	No stable distribution