ARMA

Sijia

2025-05-29

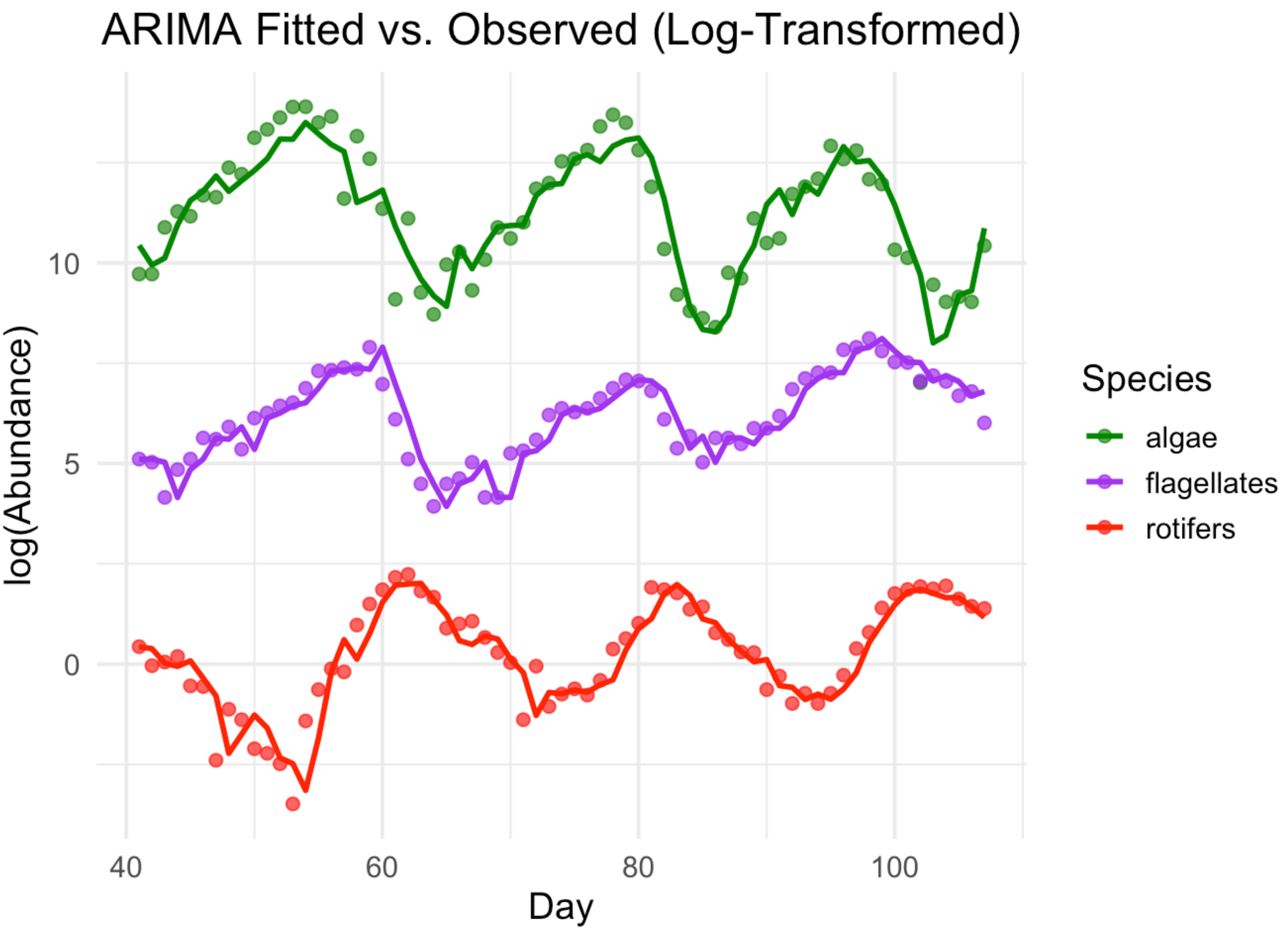
```
X123 <- read.csv("NoEvoData=SE=Feb14 2012.csv")

# Prepare log-transformed orange panel data
orange_data <- data.frame(
  day = X123$day,
  algae = log(X123$Algae.orange1),
  flagellates = log(X123$Flag.orange1),
  rotifers = log(X123$Rot.orange1)
)

# Subset data between day 41 and 107
orange_data <- orange_data[orange_data$day >= 41 & orange_data$day <= 107, ]

# Handle infinite values in rotifers
orange_data$rotifers[!is.finite(orange_data$rotifers)] <-
  min(orange_data$rotifers[is.finite(orange_data$rotifers)]) - 1

summary(orange_data)
```

```
##       day            algae         flagellates       rotifers
##  Min.   : 41.0   Min.   : 7.013   Min.   :3.932   Min.   :-3.4849
##  1st Qu.: 57.5   1st Qu.: 9.856   1st Qu.:5.366   1st Qu.:-0.6360
##  Median : 74.0   Median :11.283   Median :6.209   Median : 0.3011
##  Mean   : 74.0   Mean   :11.215   Mean   :6.180   Mean   : 0.2567
##  3rd Qu.: 90.5   3rd Qu.:12.589   3rd Qu.:7.056   3rd Qu.: 1.4142
##  Max.   :107.0   Max.   :13.892   Max.   :8.114   Max.   : 2.2357
```

```
species_names <- c("algae", "flagellates", "rotifers")
arima_models <- list()

for(species in species_names) {

  ts_data <- ts(orange_data[[species]], frequency = 1)

  # Auto ARIMA
  auto_model <- auto.arima(ts_data, seasonal = FALSE, stepwise = FALSE)
  arima_models[[species]] <- auto_model

}

n_total <- nrow(orange_data)
h_forecast <- 8  # Forecast horizon
n_train <- n_total - h_forecast
train_data <- orange_data[1:n_train, ]
test_data <- orange_data[(n_train + 1):n_total, ]
forecast_performance <- data.frame()

for(species in species_names) {
  # Refit model on training data
  train_ts <- ts(train_data[[species]], frequency = 1)

  # Get ARIMA order from full model
  full_model <- arima_models[[species]]
  arima_order <- arimaorder(full_model)

  # Refit on training data
  train_model <- Arima(train_ts, order = arima_order)

  # Generate forecasts
  forecasts <- forecast(train_model, h = h_forecast)

  # Calculate performance metrics
  actual <- test_data[[species]]
  predicted <- as.numeric(forecasts$mean)

  mae <- mean(abs(actual - predicted))
  rmse <- sqrt(mean((actual - predicted)^2))
  mape <- mean(abs((actual - predicted) / actual)) * 100

  # Naive forecast (random walk)
  naive_forecast <- rep(tail(train_data[[species]], 1), h_forecast)
  mae_naive <- mean(abs(actual - naive_forecast))
  rmse_naive <- sqrt(mean((actual - naive_forecast)^2))

  forecast_performance <- rbind(forecast_performance, data.frame(
    species = species,
    mae_arima = mae,
    rmse_arima = rmse,
    mape_arima = mape,
    mae_naive = mae_naive,
    rmse_naive = rmse_naive,
    improvement = (rmse_naive - rmse) / rmse_naive * 100
  ))
}
```

```r
# Combine all fitted values into one long dataframe
fitted_all <- data.frame()

for(species in species_names) {
  model <- arima_models[[species]]
  fitted_vals <- fitted(model)

  df <- data.frame(
    day = orange_data$day,
    observed = orange_data[[species]],
    fitted = as.numeric(fitted_vals),
    species = species
  )

  fitted_all <- bind_rows(fitted_all, df)
}

# Pivot longer for ggplot
fitted_long <- fitted_all %>%
  pivot_longer(cols = c(observed, fitted), names_to = "type", values_to = "log_abunda
nce")

# Unified plot: points for observed, dashed lines for fitted
ggplot(fitted_long, aes(x = day, y = log_abundance, color = species)) +
  geom_point(data = filter(fitted_long, type == "observed"), size = 2, alpha = 0.7) +
  geom_line(data = filter(fitted_long, type == "fitted"), linewidth = 1) +
  scale_color_manual(values = c(
    algae = "green4",
    flagellates = "purple",
    rotifers = "red"
  )) +
  scale_linetype_manual(values = c(fitted = "dashed")) +
  labs(
    title = "ARIMA Fitted vs. Observed (Log-Transformed)",
    x = "Day",
    y = "log(Abundance)",
    color = "Species",
    linetype = "Type"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and th
e
## data's linetype values.
```


ARIMA Fitted vs. Observed (Log-Transformed)

```
total_loglik <- sum(sapply(arima_models, function(model) logLik(model)))
total_loglik
```

```
## [1] -176.4332
```