# Chapter 4

Below is a piece of code to load a file sample\_file\_poly.csv. It produces a decomposition of a time series using two methods: STL and decomposition() and compares the two. It plots the residual (or errors) of the two methods. The data file (sample\_file\_poly.csv) is downloadable from BlackBoard. Run the code on the data file. Look at the output.

# Load required packages

library(readr)

library(tsibble)

library(lubridate)

library(feasts)

library(fabletools)

library(ggplot2)

# 1. Load and prepare data

data <- read.csv("C:/Users/trout/Downloads/sample\_file\_poly.csv", stringsAsFactors = FALSE)

data$Date <- mdy(data$Date)

# 2. Convert to tsibble

ts\_data <- tsibble::as\_tsibble(data, index = Date)

# 3. Plot Value vs Date

ggplot(ts\_data, aes(x = Date, y = Value)) +

geom\_line() +

labs(title = "Time Series: Value vs Date", x = "Date", y = "Value")

# 4. STL decomposition

stl\_fit <- model(ts\_data, STL(Value ~ season(window = "periodic")))

stl\_comp <- components(stl\_fit)

# 5. Classical decomposition (corrected)

classical\_fit <- model(ts\_data, classical\_decomposition(Value ~ season()))

classical\_comp <- components(classical\_fit)

# 6. Plot both decompositions

autoplot(stl\_comp) + ggtitle("STL Decomposition")

autoplot(classical\_comp) + ggtitle("Classical Decomposition")

# 7. Extract residuals

stl\_resid <- stl\_comp[, c("Date", "remainder")]

names(stl\_resid)[2] <- "STL\_Residual"

# Classical uses 'random' instead of 'remainder'

classical\_resid <- classical\_comp[, c("Date", "random")]

names(classical\_resid)[2] <- "Classical\_Residual"

# 8. Merge residuals by Date

merged\_resid <- merge(stl\_resid, classical\_resid, by = "Date", all = TRUE)

# === OPTION 2: Show missing rows, if any

na\_rows <- merged\_resid[!complete.cases(merged\_resid), ]

if (nrow(na\_rows) > 0) {

cat("⚠️ Missing values in residuals at:\n")

print(na\_rows)

}

# 9. Do NOT drop NAs (retain for plotting)

# merged\_resid <- na.omit(merged\_resid) # <- Optional, but not used here

# 10. Convert to long format for plotting residuals

merged\_resid\_long <- data.frame(

Date = rep(merged\_resid$Date, 2),

Residual = c(merged\_resid$STL\_Residual, merged\_resid$Classical\_Residual),

Method = rep(c("STL", "Classical"), each = nrow(merged\_resid))

)

# 11. Plot residual comparison

ggplot(merged\_resid\_long, aes(x = Date, y = Residual, color = Method)) +

geom\_line(na.rm = TRUE) + # === OPTION 3: Suppress NA warning

labs(title = "Residuals: STL vs Classical", x = "Date", y = "Residual")

# 12. Residual summary stats

stl\_sd <- sd(merged\_resid$STL\_Residual, na.rm = TRUE)

classical\_sd <- sd(merged\_resid$Classical\_Residual, na.rm = TRUE)

stl\_mean <- mean(merged\_resid$STL\_Residual, na.rm = TRUE)

classical\_mean <- mean(merged\_resid$Classical\_Residual, na.rm = TRUE)

cat("\nResidual Summary:\n")

cat(sprintf("STL - Mean: %.4f SD: %.4f\n", stl\_mean, stl\_sd))

cat(sprintf("Classical - Mean: %.4f SD: %.4f\n", classical\_mean, classical\_sd))

## What can you say about the output? Did the code do a good job at decomposition? Are there any glaring issues? What are they?

### Outputs

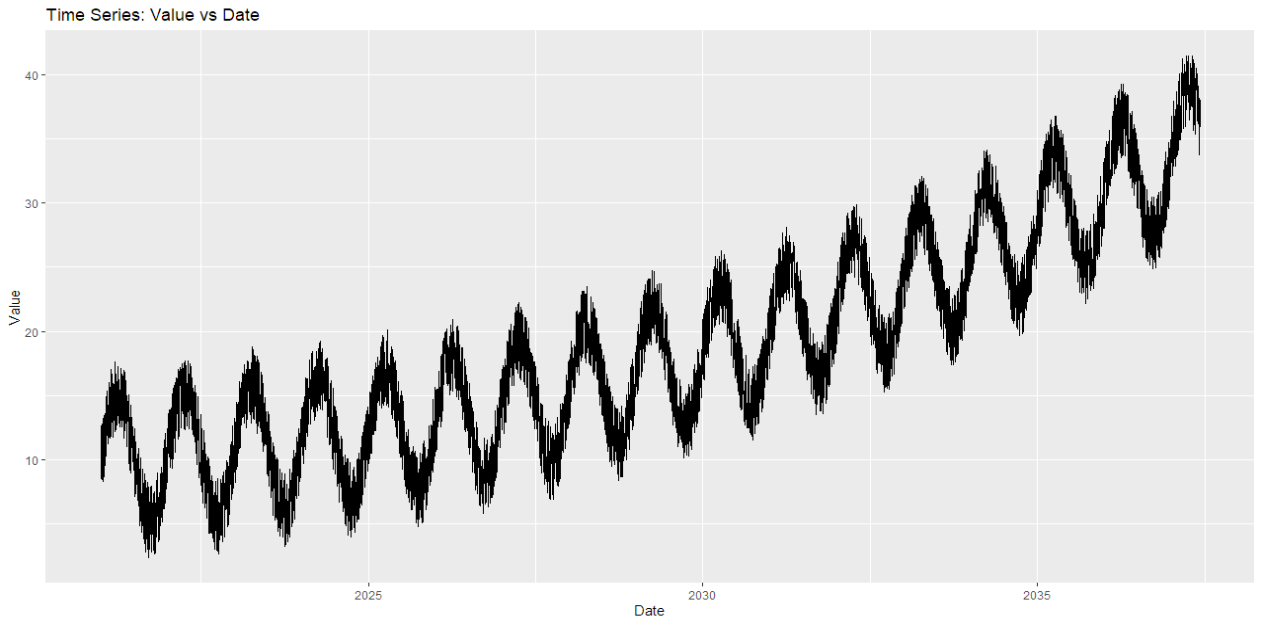


Figure : Time Series: Value vs Date

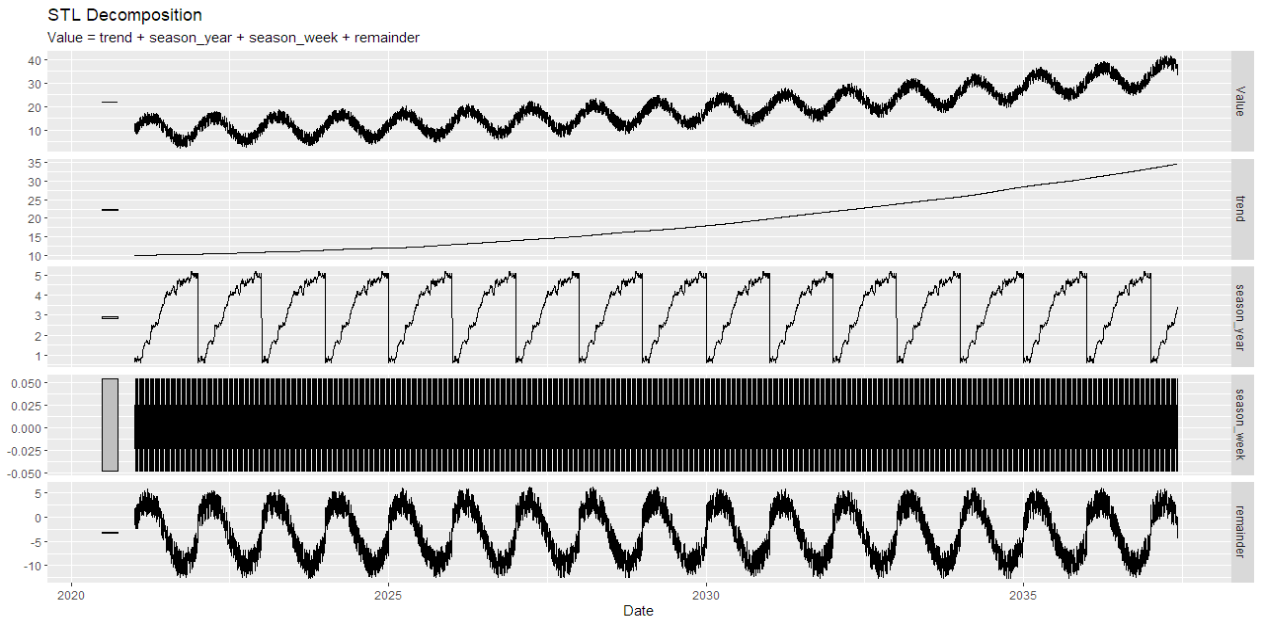


Figure : STL Decomposition

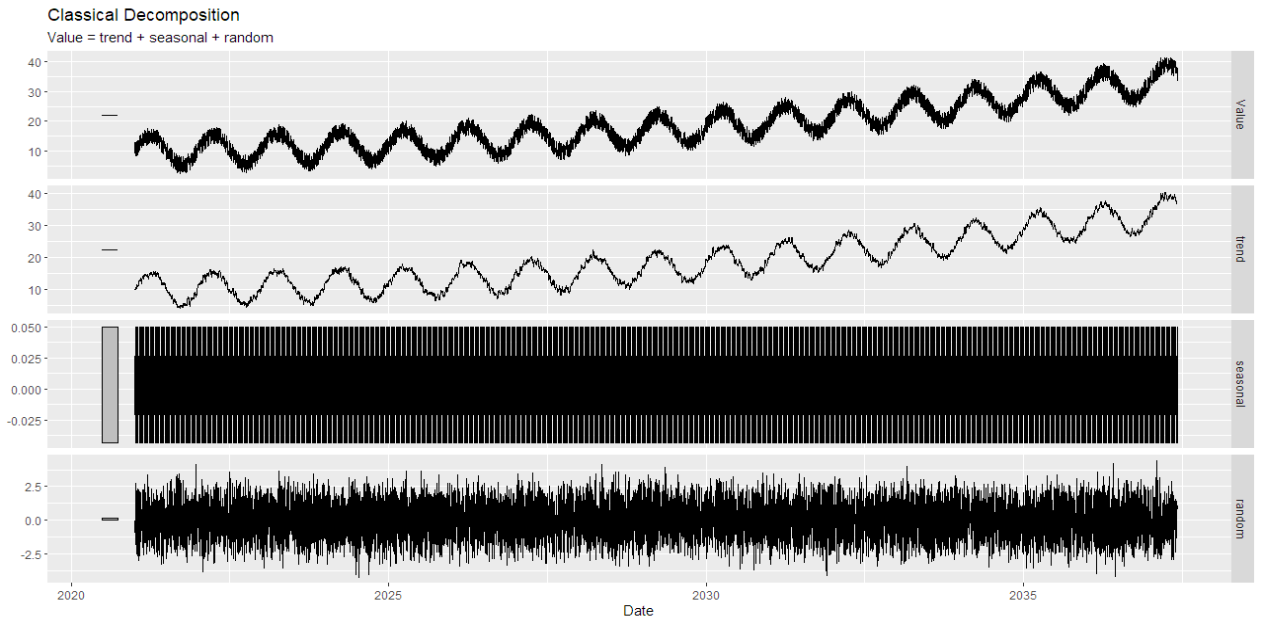


Figure : Classical Decomposition

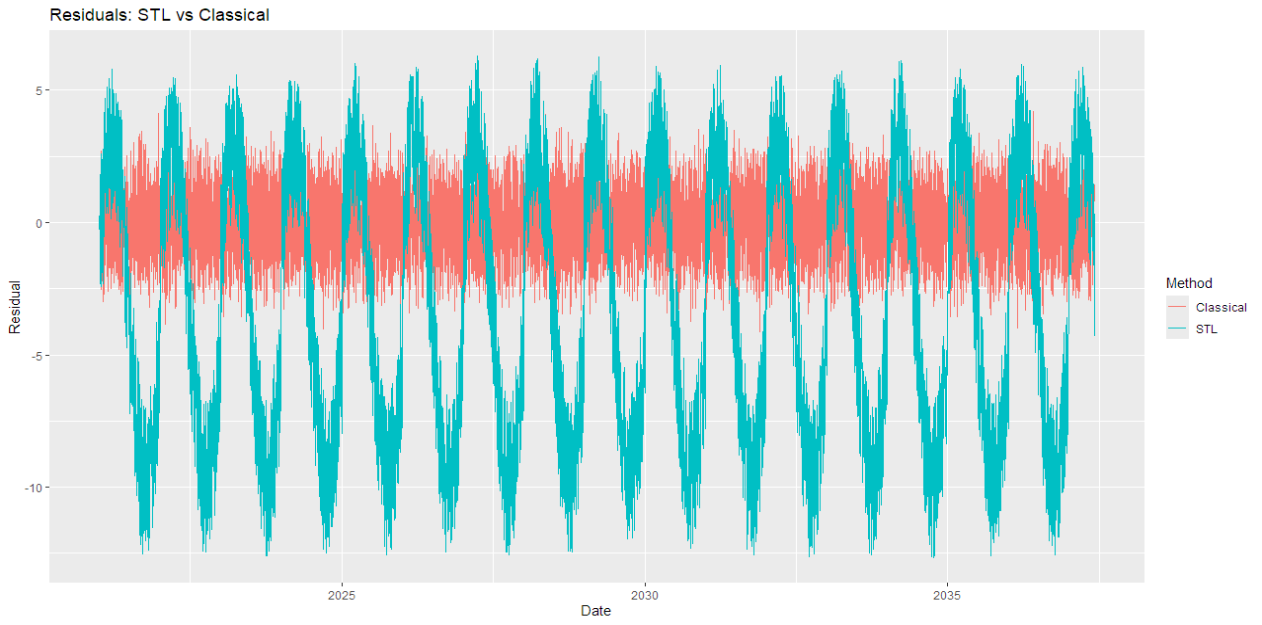


Figure : Residuals: STL vs Classical

### Comments on the outputs

The time series shows an upward trend, meaning the values are increasing over time. There is also a clear yearly seasonal pattern, where similar changes repeat each year.

In the STL decomposition, we get four parts: trend, season\_year, season\_week, and remainder. The trend shows a steady and smooth increase. The season\_year is easy to see and clearly shows the yearly repeating pattern. The season\_week, however, is hard to understand, it mostly looks like a dark or noisy area, so it’s not clear if there’s any real weekly pattern. The remainder still shows some yearly seasonality, which might mean STL didn’t fully remove all seasonal effects, or that the seasonality is more complex than the model can handle.

In the classical decomposition, the trend also increases but has more ups and downs, showing more short-term changes. The seasonal part looks similar to the season\_week in STL, it’s not very clear and mostly looks like noise. The remainder here looks more like random noise, without any clear pattern, which might suggest that the classical model captured most of the trend and seasonal parts.

## Hopefully you found that even though the code completed successfully, the output was not satisfactory. Two simple corrections should be made. Make corrections on the code.

Fix the STL decomposition model:

stl\_fit <- model(

ts\_data,

STL(Value ~ **season(period = "1 year")** + **season(period = "1 week")**)

)

Fix the classical decomposition model:

classical\_fit <- model(

ts\_data,

classical\_decomposition(Value ~ season(**"year"**))

)

## Run the program again and discuss the output. Discuss the trend and seasonality. Use the residual information to discuss the quality of the methods. Which was best? Why?

New outputs:

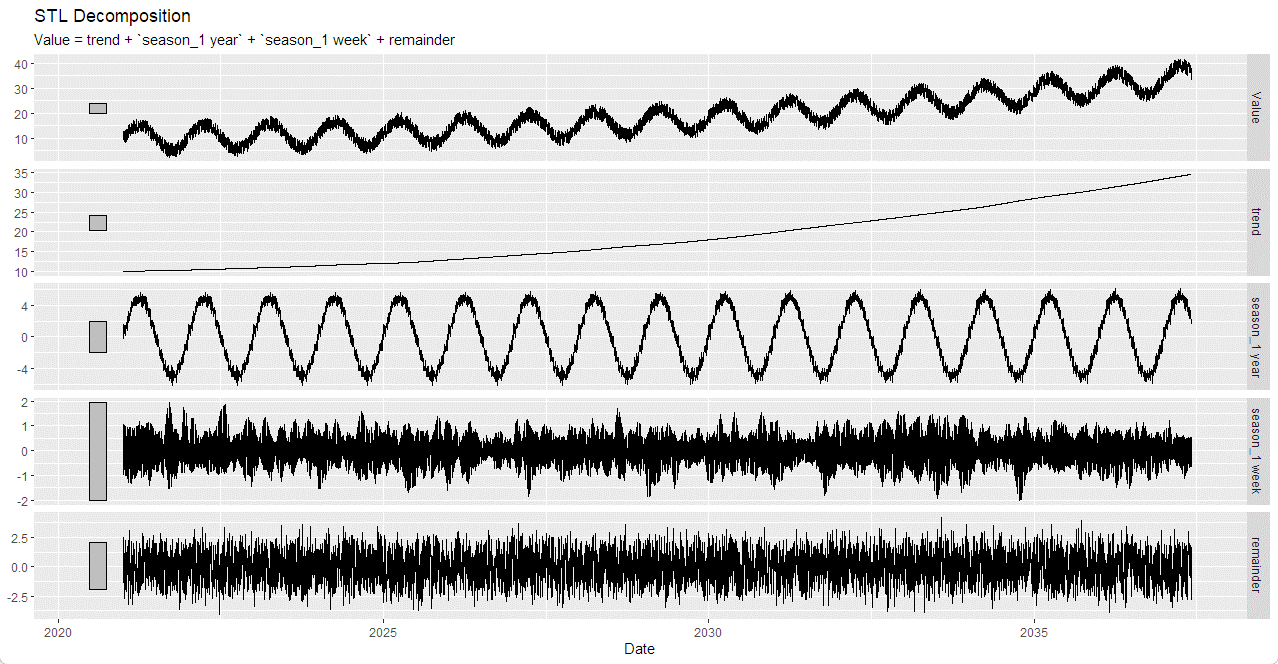


Figure 5: STL Decomposition

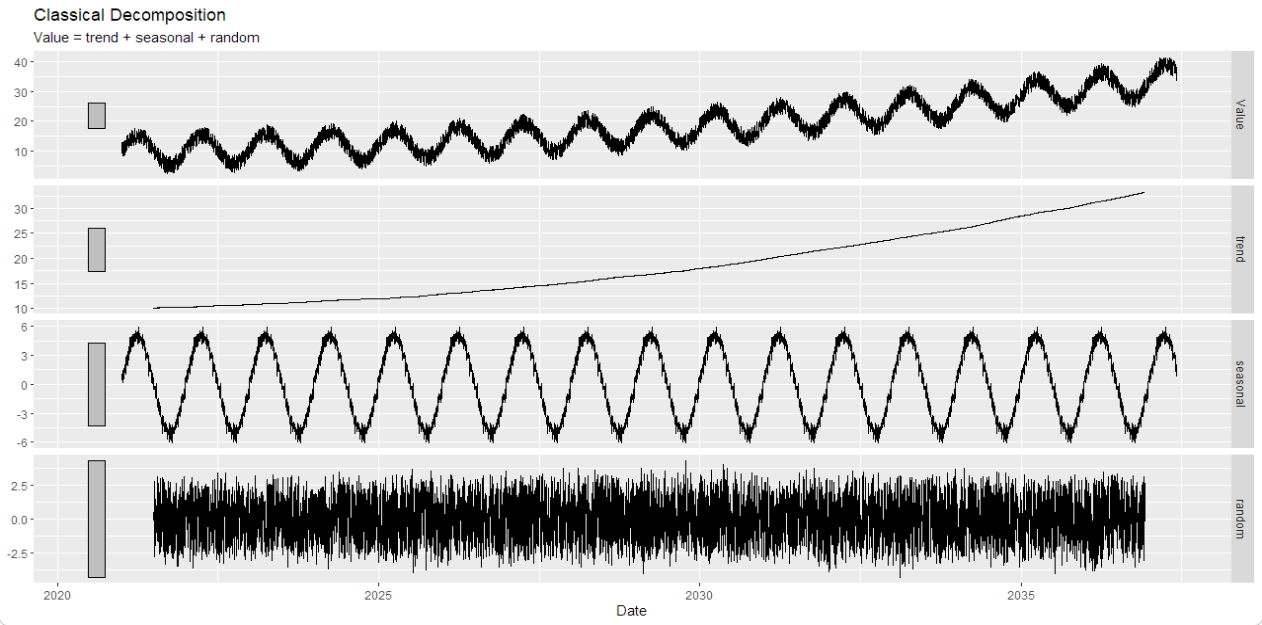


Figure 6: Classical Decomposition

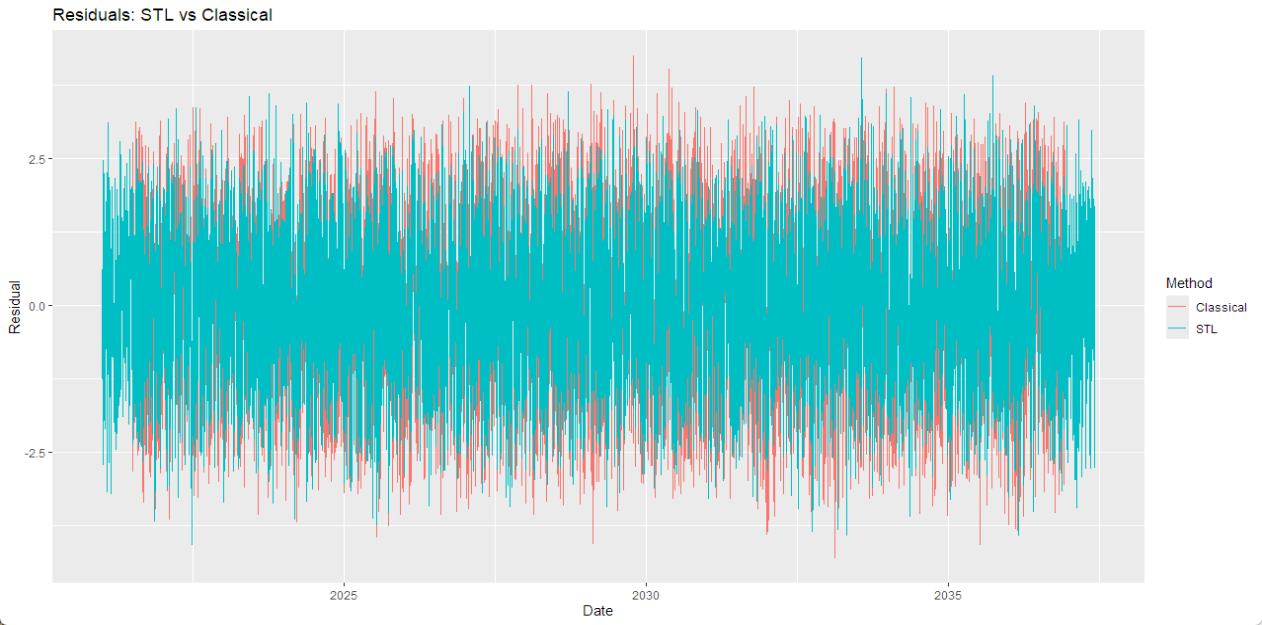


Figure 4: Residuals: STL vs Classical

The trend component now appears consistent between STL and Classical decomposition, indicating a likely multiplicative trend over time. Both methods reveal a clear yearly seasonal pattern. Additionally, the weekly seasonal component is now properly rendered and no longer contains blacked-out or missing areas, improving readability.

The residual plots from both STL and Classical decomposition display random noise without any discernible structure or autocorrelation. This suggests that the trend and seasonal components have been successfully extracted, leaving behind only the irregular (unexplained) component. The lack of patterns in the residuals indicates that neither method is systematically missing key features in the data.