Analyzing Synthetic Sales Data with tstodfpkg

Abiyu Giday

2025-07-26

Contents

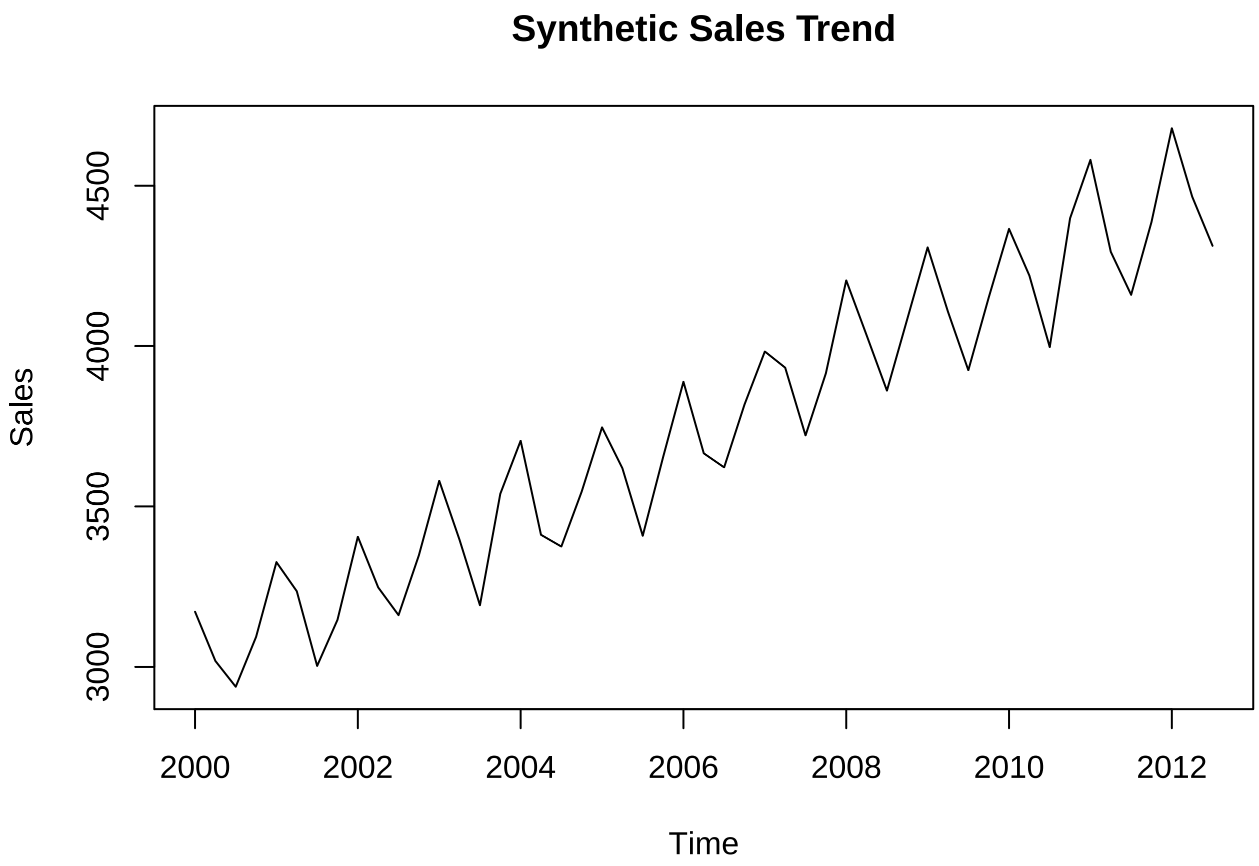
Analyzing Time Series Sales Data

This vignette demonstrates how to use `tstodfpkg` to analyze the `synth_gap` dataset, a synthetic quarterly sales time series. We will show `ts_to_df` working with both the original `ts` structure and the converted `data.frame`, comparing the results to highlight its versatility.

Visualizing Trends with Time Series

Load the dataset and plot the time series to identify trends:

```r
data(synth_gap)
plot(synth_gap, main = "Synthetic Sales Trend", xlab = "Time", ylab = "Sales", type = "l")
```



Analyzing with Data Frame Using dplyr and ggplot2

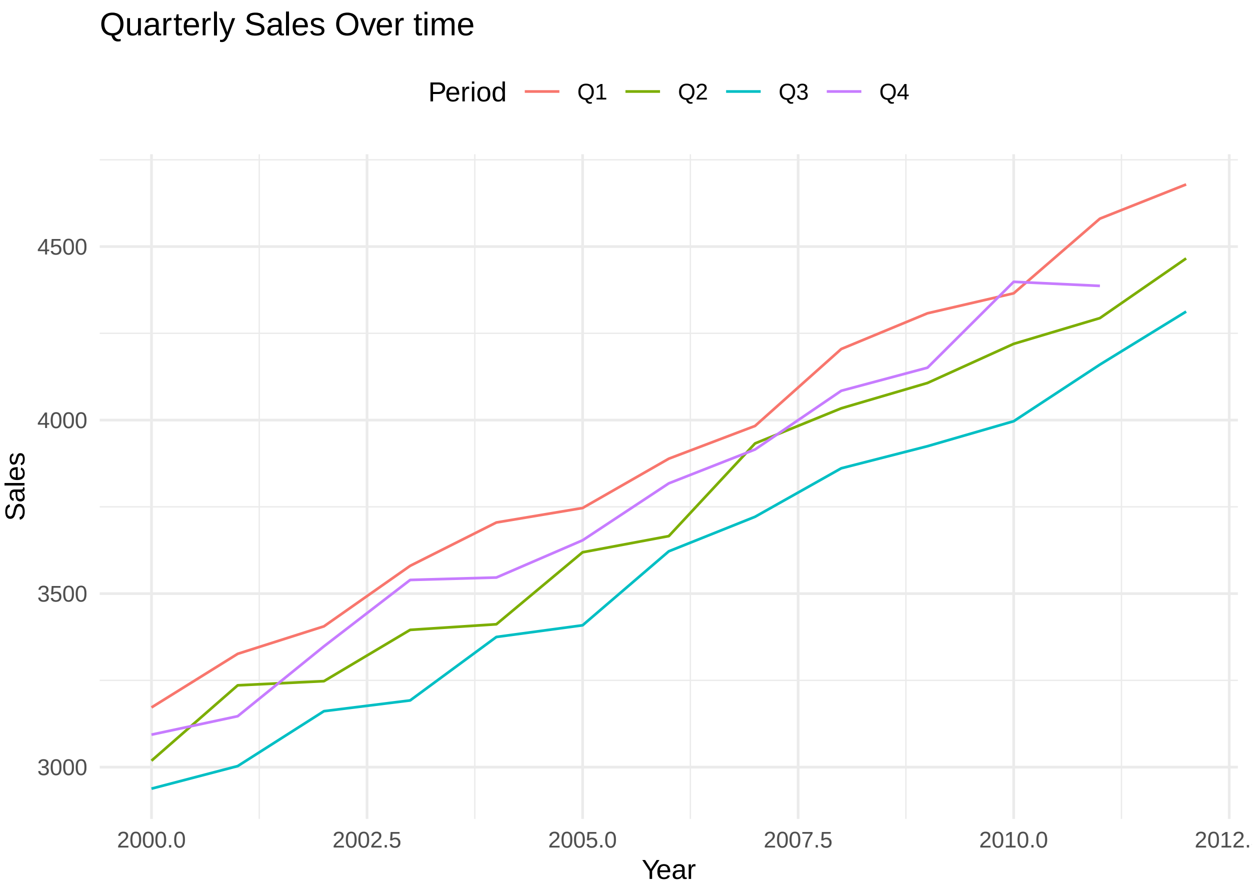
Convert the time series to a data frame and use `dplyr` for aggregation:

```
df <- ts_to_df(synth_gap, name = "Sales")
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
sales_summary <- df %>% group_by(Period) %>% summarise(Avg_Sales = mean(Sales))
print(sales_summary)
#> # A tibble: 4 x 2
#>   Period Avg_Sales
#>   <fct>      <dbl>
#> 1 Q1         3919.
#> 2 Q2         3742.
#> 3 Q3         3590.
#> 4 Q4         3757.
```

Visualize the quarterly sales with `ggplot2`:

```r
library(ggplot2)

ggplot(df, aes(x = Year, y = Sales, color = Period)) +
  geom_line() +
  #facet_wrap(~ Year, ncol = 4) +
  theme_minimal() +
  theme(legend.position = "top") +
  labs(title = "Quarterly Sales Over time")
#> Don't know how to automatically pick scale for object of type <ts>. Defaulting
#> to continuous.
```



This approach leverages the `data.frame` structure for flexible manipulation and enhanced visualization, making it ideal for detailed sales pattern analysis.