Applied Time Series Analysis

Zem Wang

1/1/24

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

Part I

The Basics

1 Time Series Data

Raw data: The raw values without any transformation. We are not so interested in the raw data, as it is hard to read information from it. Take the GDP plot as an example (Figure 1.1, upper-left subplot). There is an overall upward trend. But we are more interested in: how much does the economy grow this year? Is it better or worse than last year? The answers are not obvious from the raw data. Besides, there are obvious seasonal fluctuations. Usually the first quarter has the lowest value in a whole year, due to the Spring Festival, which significantly reduces the working days in the first quarter. The seasonal fluctuations prohibit us from sensibly comparing two consecutive values.

Growth rate: The headline GDP growth is usually derived by comparing the current quarter with the same quarter from last year. $g = \frac{x_t - x_{t-4}}{x_{t-4}} \times 100$. This makes sense. As mentioned above, due to seasonal patterns, comparing two consecutive quarters directly does not make sense. The year-on-year growth rate directly tells us how fast the economy grows. However, by dividing the past values, it loses the absolute level information. For instance, it is hard to tell after the pandemic, whether or not the economy recovers from its pre-pandemic output level. Besides, it is sensitive to the values of last year. For example, due to the pandemic, the GDP for 2020 is exceptionally low, which makes growth rate for 2021 exceptionally high. This is undesirable, because it does not mean the economy in 2021 is actually good. We would like a growth rate that shirks off past burdens.

That's why we sometimes prefer (annualized) quarterly growth rate. $g = \frac{x_t - x_{t-1}}{x_{t-1}} \times 400$. Due to seasonally patterns, two consecutive quarters are not comparable directly. A first quarter value is usually much lower than the fourth quarter of last year due to holidays, which does not necessarily mean the economy condition is getting worse. Since this pattern is the same every year, it is possible to remove the seasonal fluctuations. This is called *seasonally adjustment*. We won't cover seasonally adjustment in detail, but the next section will give some intuitions on how this can possibly be done. After seasonally adjusting the time series, we can calculate the growth rate based on two consecutive values (annualized by multiplying 4). The bottom-right panel of Figure 1.1 is the seasonally-adjusted quarterly growth. Note that it is no longer biased upward in 2021 as the YoY growth.

Seasonally-adjusted series: This is usually the data format we prefer in time series analysis. FRED reports both seasonally-adjusted and non-seasonally-adjusted series. Seasonal adjustment algorithm is a science in itself. Popular algorithms include X-13-ARIMA developed by the United States Census Bureau, TRAMO/SEATS developed by the Bank of Spain, and so on.

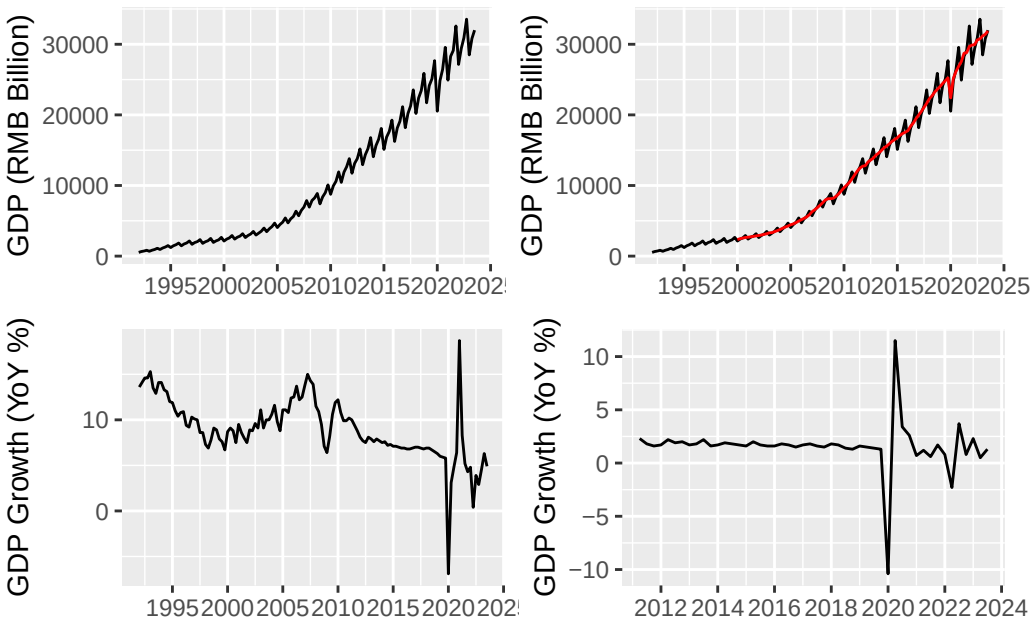

Figure 1.1: Quarterly GDP Time Series

Log levels and log growth rates: We like to work with log levels. A lot of economic time series exhibit exponential growth, such as GDP. Taking logs convert them to linear. Another amazing thing about logs is the difference of two log values can be interpreted as percentage growth. We know from Taylor expansion that for small values of Δx : $\ln(\Delta x + 1) \approx \Delta x$. Therefore,

$$\ln x_t - \ln x_{t-1} = \ln\left(\frac{x_t}{x_{t-1}}\right) = \ln\left(\frac{x_t - x_{t-1}}{x_{t-1}} + 1\right) \approx \frac{x_t - x_{t-1}}{x_{t-1}}.$$

So it is very handy to just difference the log levels to get the growth rates. Log growth is even preferred to percentage growth, because it has the property of summability: summing up a series of log growth rates gives the log level given the initial level. It is not as handy if you want to recover the level values from a series of percentage growth.

$$\ln x_t = x_0 + \sum_{j=1}^{t} (\ln x_j - \ln x_{j-1}).$$

2 Time Series Decomposition

2.1 Time Series Components

It is helpful to think about a time series as composed of different components: a trend component, a seasonal component, and a remainder.

$$x_t = T_t + S_t + R_t.$$

The formula assumes the "additive" composition. This assumption is appropriate if the magnitude of the fluctuations does not vary with the absolute levels of the time series. If the magnitude of fluctuations is proportional to the absolute levels, a "multiplicative" decomposition is more appropriate:

$$x_t = T_t \times S_t \times R_t.$$

Note that a multiplicative decomposition of a time series is equivalent to an additive decomposition on its log levels:

$$\ln x_t = \ln T_t + \ln S_t + \ln R_t.$$

Decomposing a time series allows us to extract information that is not obvious from the original time series. It also allows us to manipulate the time series. For example, if the seasonal component can be estimated, we can remove it to obtain seasonally-adjusted series, $x_t^{SA} = x_t - S_t$, or $x_t^{SA} = x_t / S_t$. The question is how to estimate the components given a time series.

2.2 Moving Averages

Moving averages turn out to be handy in estiming trend-cycles by averaging out noisy fluctuations. A moving average of order m (assuming m is an odd number) is defined as

$$\text{MA}(x_t, m) = \frac{1}{m}\sum_{j=-k}^{k} x_{t+j},$$

where $m = 2k + 1$. For example, a moving average of order 3 is

$$\text{MA}(x_t, 3) = \frac{1}{3}(x_{t-1} + x_t + x_{t+1}).$$

Note that x_t is centered right in the middle and the average is symmetric. This also means, if we apply this formula to real data, the first and last observation will have to be discarded. If the order m is an even number, the formula will no longer be symmetric. To overcome this, we can estimate a moving average over another moving average. For example, we can estimate a moving average of order 4, followed by a moving average of order 2. This is denoted as 2×4-MA. Mathematically,

$$\begin{aligned}
\text{MA}(x_t, 2 \times 4) &= \frac{1}{2}[\text{MA}(x_{t-1}, 4) + \text{MA}(x_t, 4)] \\
&= \frac{1}{2}\left[\frac{1}{4}(x_{t-2} + x_{t-1} + x_t + x_{t+1}) + \frac{1}{4}(x_{t-1} + x_t + x_{t+1} + x_{t+2})\right] \\
&= \frac{1}{8}x_{t-2} + \frac{1}{4}x_{t-1} + \frac{1}{4}x_t + \frac{1}{4}x_{t+1} + \frac{1}{8}x_{t+2}.
\end{aligned}$$

Note that how the 2×4-MA averages out the seasonality for time series with seasonal period 4, e.g. quarterly series. The formula puts equal weight on every quarter — the first and last terms refer the same quarter and their weights combined to $\frac{1}{4}$.

In general, we can use m-MA to estimate the trend if the seasonal period is an odd number, and use $2 \times m$-MA if the seasonal period is an even number.

```
library(zoo)
library(ggplot2)
library(forecast)

data = read.csv.zoo("data/gdp.csv")
gdp2x4MA = ma(ma(data$GDP, 4), 2)

autoplot(data$GDP, series = "Data") +
  autolayer(gdp2x4MA, series = "2x4 MA") +
  labs(x = "Year", y = "GDP (RMB Billion)") +
  scale_x_continuous(breaks = pretty)
```

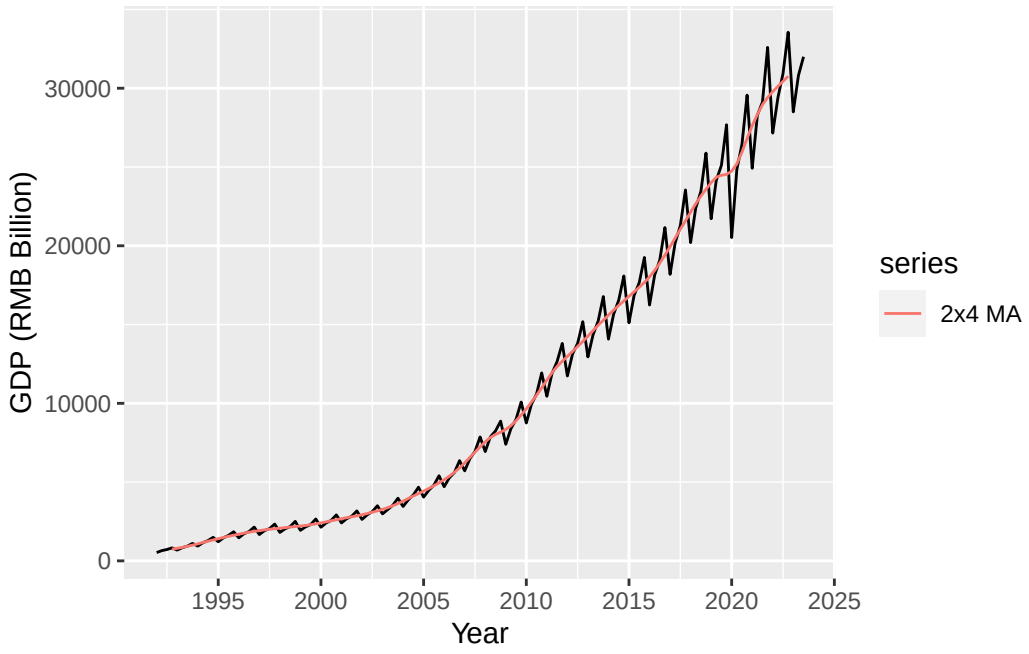

Figure 2.1: Quarterly GDP with 2x4-MA estimate of the trend-cycle

2.3 Classical Decomposition

Moving averages give us everything we need to perform classical decomposition. Classical decomposition, invented 1920s, is the simplest method decompose a time series into trend, seasonality and remainder. It is outdated nowadays and has been replaced by more advanced algorithms. Nonetheless, it serves as a good example for introductory purpose on how time series decomposition could possibly be achieved.

The algorithm for additive decomposition is as follows.

1. Estimate the trend component T_t by applying moving averages. If the seasonal period is an odd number, apply the m-th order MA. If the seasonal period is even, apply the $2 \times m$ MA.
2. Calculate the detrended series $x_t - T_t$.
3. Calculate the seasonal component S_t by averaging all the detrended values of the season. For example, for quarterly series, the value of S_t for Q1 would be the average of all values in Q1. This assumes the seasonal component is constant over time. S_t is then adjusted to ensure all values summed up to zero.
4. Subtracting the seasonal component to get the remainder $R_t = x_t - T_t - S_t$.

```r
as.ts(log(data$GDP)) %>%
  decompose() %>%
  autoplot()
```

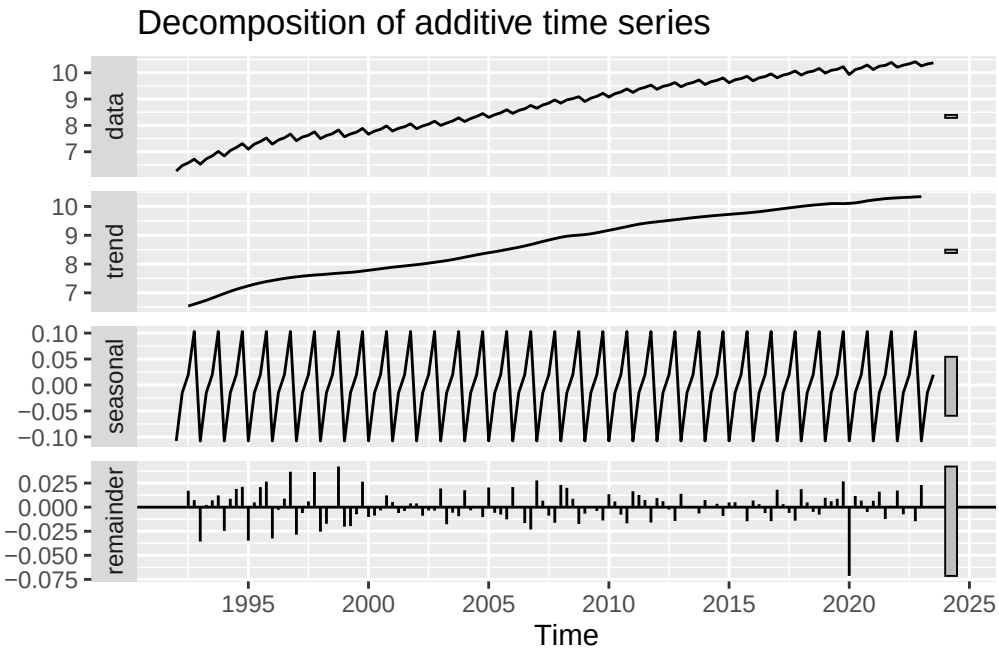

Figure 2.2: Classical multiplicative decomposition of quarterly GDP

The example performs additive decomposition to the logged quarterly GDP series. Note how the constant seasonal component is removed, leaving the smooth and nice-looking up-growing trend. The remainder component tells us the irregular ups and downs of the economy around the trend-cycle. Isn't it amazing that a simple decomposition of the time series tells us a lot about the economy?

2.4 Seasonal Adjustment

By decomposing a time series into trend, seasonality and remainder, it readily gives us a method for seasonal adjustment. Simply substracting the seasonal component from the original data, or equivalently, summing up the trend and the remainder components, would give us the seasonally-adjusted series.

The following example compares the seasonally-adjusted series using the classical decomposition with the state-of-the-art X-13ARIMA-SEATS algorithm. Despite the former is far more rudimentary than the latter, they look quite close if we simply eye-balling the plot. By taking

first-order differences, we can see the series based on classical decomposition is more volatile, suggesting the classical decomposition is less robust to unusual values.

```
logdata = as.ts(log(data)) %>% window(start=2000)

plotLevel = decompose(logdata[,'GDP']) %>%
  seasadj() %>%
  autoplot(series = "Classical") +
  autolayer(logdata[,'GDPSA'], series = "X-13") +
  scale_color_manual(values = 1:2)

plotDiff = decompose(logdata[,'GDP']) %>%
  seasadj() %>% diff() %>%
  autoplot(series = "Classical") +
  autolayer(logdata[,'GDPSA'] %>% diff(), series = "X-13") +
  scale_color_manual(values = 1:2)

library(patchwork)
plotLevel + plotDiff +
  plot_layout(guides = "collect")
```

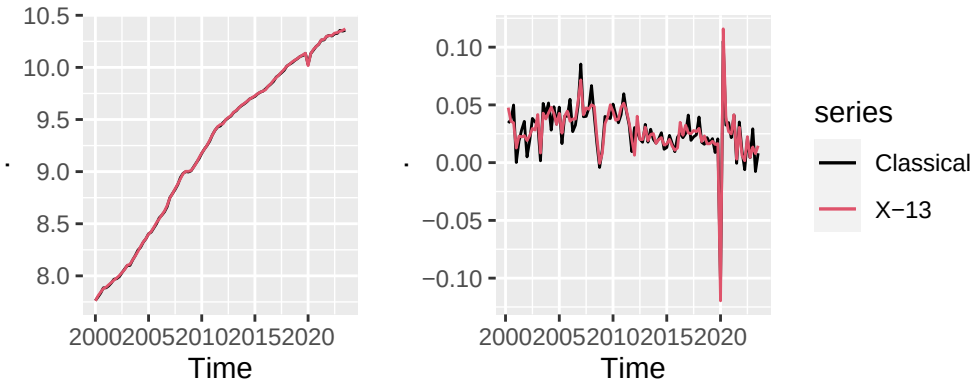

Figure 2.3: Comparing classical decomposition and X-13

Part II

ARIMA Model

3 Summary

In summary, this book has no content whatsoever.

References