ETF3231/5231 Week 3 - Time series patterns, transformations, and decomposition

Sao Yang Hew

Friday - 10 March 2023 (12:30pm - 2:00pm)

2023

Table of Contents

Revision for last week

- Last week, we looked into the `autoplot` function to visualise time series.
- We also looked at the following to identify **seasonal patterns** available in time series:
  - `gg_season()`
  - `gg_subseries()`

Effect of lags on time series

- This week, we are going to assess other components in the time series and how we can visualise them. These include:
  - `gg_lag()`
  - `ACF() %>% autoplot()`
  - Both these answer the same question: How much does the past influence the future?
- Central idea: for **trend, seasonality, and cyclicity**, the past has some indication of the future. And this is reflected using the above plot functions.
- **Trends and seasons** are reflected by very correlated `gg_lag()` values across the board, and positive `ACF()` values across different lags.
- **Cycles** are reflected by fluctuating `ACF()` values - from negative, to positive, then negative again. It is unclear whether cycles show very different `gg_lag()` values.

Seasons and cycles

- Question: what is the difference between **seasonal** and **cyclical** variation?
  - Seasons usually occur **within the year**. And has regular patterns - e.g., December or Q4 reports the highest retail sales in Australia, and the same can be said about every December or Q4.
  - Cycles can occur **beyond a year**. It does not have regular patterns:
    - Recession in 1981 lasted for 15 months
    - Recession in 1990 lasted for 8 months
    - Recession in 2008 lasted for 18 months
- `gg_lag()` can usually pick out seasonality and trend well, but they are less straightforward when assessing cyclical patterns.
  - If an item has quarterly seasonality - lags 1 and 4 are usually correlated
  - If an item has monthly seasonality - lags 1 and 12 are usually correlated
  - If an item exhibits a huge trend - the lags are all very correlated

Why do we transform time series?

There are a few reasons why we need to transform our time series before forecasting, but they can be categorised into 2 main reasons:

- Mathematical transformations (e.g., log, and Box-Cox) help with making our time series easier to forecast, especially if **time series have higher variation at high positive, or low negative values**.
- It makes sense in a business or contextual setting, usually for comparisons:
  - Population adjustments, e.g., GDP per capita time series are more comparable between countries
  - Inflation adjustments, e.g., real wages are more comparable between different periods (e.g., 1980 vs 2010).
  - Calendar adjustments, e.g., for monthly sales ensures that months with different days (e.g., November vs December) are comparable

Mathematical transformations

- log transformation
- Box-Cox transformation (note that Box-Cox transformation with $\lambda = 0$ is equivalent to log transformation)
- Important tip: log and Box-Cox transformations are **only** appropriate when the variance of the time series is proportional to the level of the time series

Why do we decompose time series?

- Time series can usually be decomposed into 3 main components, using mathematical formulae:
  - Trend-cycle components, we keep these together because:
    - It is hard to separate these mathematically
    - When forecasting trend-cycle components, we do not usually care about the cycles as they are very hard to forecast, and would require a lot of specialist knowledge
  - Seasonal components
  - Remainder/residual components
- We decompose time series so we can do a few things:
  - To allow us to split and forecast time series separately, which will then be combined at the end
  - To observe if the remainder component is entirely random - if they are not, it would mean that we are not taking into account some information we can use to forecast our time series better. We would have to then assess if our decomposition is done appropriately, or if we need to mathematically transform our time series again.

Types of decompositions

- Classical
  - Available in additive and multiplicative form
  - Lose first and last few observations due to moving average calculations
  - Assumes seasonality remains constant over time
  - Not robust to outliers
- X11
  - Available for all observations
  - Robust to outliers
- SEATS
  - Multiplicative only
- STL
  - Any type of seasonality can be used
  - Seasonal component can change smoothly over time
  - Smoothness of the trend can be controlled by the user
  - Robust to outliers
  - Additive only