Time series decomposition

Print-friendly PDF version

Before you start

This is an in-class lab: work through the material below at your own pace (in pairs or small groups is fine) — I'll walk around to help, and we'll stop along the way to discuss as a group.

This builds on the moving averages, classical decomposition, X11/SEATS and STL material from the decomposition chapter. Today you apply it: decompose real series into trend-cycle, seasonal and remainder components, compare methods, and interpret what you get.

```
library(fpp3)
library(tidyverse)
```

Cheat sheet: decomposition methods

Method	Syntax
Classical	`model(classical_decomposition(variable, type = "additive"))`
X11	`model(X_13ARIMA_SEATS(variable ~ x11()))`
SEATS	`model(X_13ARIMA_SEATS(variable ~ seats()))`
STL	`model(STL(variable ~ trend(window = ...) + season(window = ...), robust = ...))`

After fitting, use `components()` to extract the trend-cycle/seasonal/remainder columns, then `autoplot()` to plot them.

Cheat sheet: features

```
data %>%
  features(variable, feat_stl)
```

Gives one row per series with columns like `trend_strength` and `seasonal_strength_year` - see Section 4.3 of the textbook for the full list.

Task 1: Offshore wind power

Use the `windpower` data from `ban430data` (the same one from the Time series graphics lab) again — this time for decomposition.

```
# install.packages("remotes")
remotes::install_github("holleland/ban430data")
library(ban430data)
data(windpower)
```

a) Create a tsibble containing the **daily** wind power data from **Sørlig Nordsjø 2** only (aggregate `windpower` from hourly to daily with the mean, same as in the graphics lab).

Code hint

```
wp_day <- windpower %>%
  filter(Place == "Sørlig Nordsjø 2") %>%
  index_by(date = as_date(datetime)) %>%
  summarise(powerprod = mean(powerprod, na.rm = TRUE))
```

b) Decompose the series into trend-cycle T_t, season S_t and remainder R_t, using a suitable decomposition method. Which method(s) from the cheat sheet above are *not* usable here, and why? Why did you choose the method you did?

Code hint

```
wp_day %>%
  model(STL(powerprod)) %>%
  components() %>%
  autoplot()
```

c) We can also use **STL features** to assess and compare *how* seasonal or trending a series is rather than just eyeballing a decomposition (see Section 4.3 of the textbook). Aggregate `windpower` to **monthly** data for **both** locations (like `wp_month` in the graphics lab), then run:

```
windpower %>%
  group_by_key() %>%
  index_by(yearmonth = yearmonth(datetime)) %>%
  summarise(powerprod = mean(powerprod, na.rm = TRUE)) %>%
  features(powerprod, feat_stl)
```

Which location has the stronger `seasonal_strength_year`? What about `trend_strength`?

Task 2: Norwegian wholesale and retail sales index

The data is a monthly index for *Retail trade, except of motor vehicles and motorcycles*, from Statistics Norway (table 07129) (Jan 2000 onwards). It's the `wholesale_raw_lab` object from `ban430data`.

```
library(ban430data)
data(decomposition)

wholesale <- wholesale_raw_lab
head(wholesale, 3)
```

```
# A tibble: 3 x 2
  month   `wholesale and retail sales index`
  <chr>                                <dbl>
1 2000M01                               39.9
2 2000M02                               38.9
3 2000M03                               42.4
```

a) Convert the `month` column to a `yearmonth` type and turn `wholesale` into a tsibble.

Code hint

```
wholesale <- wholesale_raw_lab %>%
  rename(index = `wholesale and retail sales index`) %>%
  mutate(month = yearmonth(month)) %>%
  as_tsibble(index = month)
```

b) Make a time plot.

Code hint

```
wholesale %>% autoplot(index)
```

c) Decompose the series using the classical, X11, SEATS and STL methods. Can you detect any prominent differences between the methods?

Code hint

```
wholesale %>%
  model(
    classical = classical_decomposition(index, type = "additive"),
    x11       = X_13ARIMA_SEATS(index ~ x11()),
    seats     = X_13ARIMA_SEATS(index ~ seats()),
    stl       = STL(index)
  ) %>%
```

```
  components() %>%
  autoplot()
```

d) Try adjusting the trend and season windows of the STL (default values are 21 and 11 respectively). What happens?

Code hint

```
wholesale %>%
  model(STL(index ~ trend(window = ...) + season(window = ...))) %>%
  components() %>%
  autoplot()
```

e) Using X11 or SEATS - the "statistics agency methods" from the cheat sheet - plot your seasonally-adjusted time series.

Code hint

```
wholesale %>%
  model(seats = X_13ARIMA_SEATS(index ~ seats())) %>%
  components() %>%
  autoplot(season_adjust)
```

f) Statistics Norway publishes its own official seasonally-adjusted version of this exact series - it's the `wholesale_sa` object in `ban430data` (column `index_sa`). Plot it alongside your series from (e). Do they agree?

Code hint

```
own_sa <- wholesale %>%
  model(seats = X_13ARIMA_SEATS(index ~ seats())) %>%
  components() %>%
  as_tibble() %>%
  select(month, own = season_adjust)

official_sa <- wholesale_sa %>%
  rename(month = yearmonth, official = index_sa)

own_sa %>%
  left_join(official_sa, by = "month") %>%
  pivot_longer(c(own, official), names_to = "source", values_to = "index") %>%
  as_tsibble(index = month, key = source) %>%
  autoplot(index)
```

g) Using your method of choice, plot the detrended series.

Code hint

```
wholesale %>%
  model(STL(index)) %>%
  components() %>%
  mutate(detrended = index - trend) %>%
  autoplot(detrended)
```

h) *Optional*: Implement your own additive classical decomposition by hand (moving averages + seasonal averaging), and compare to `classical_decomposition()`. (See Chapter 3 exercises, exercise 6 if you want to check your approach against a worked solution.)

Wrap-up discussion

For both series today: did the different decomposition methods roughly agree, or did they tell noticeably different stories? When might you prefer STL over classical decomposition, or vice versa? What made X11/SEATS unsuitable for the daily wind power data?