

STAT481/581: Introduction to Time Series Analysis

Ch3. Time series decomposition

OTexts.org/fpp3/

Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

Time series patterns

Recall

Trend pattern exists when there is a long-term increase or decrease in the data.

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

Seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where $y_t =$ data at period t
 $T_t =$ trend-cycle component at period t
 $S_t =$ seasonal component at period t
 $R_t =$ remainder component at period t

Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where $y_t =$ data at period t

$T_t =$ trend-cycle component at period t

$S_t =$ seasonal component at period t

$R_t =$ remainder component at period t

Additive decomposition: $y_t = S_t + T_t + R_t$.

Multiplicative decomposition: $y_t = S_t \times T_t \times R_t$.

Time series decomposition

- Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.
- If seasonal are proportional to level of series, then multiplicative model appropriate.
- Multiplicative decomposition more prevalent with economic series
- Alternative: use a Box-Cox transformation, and then use additive decomposition.
- Logs turn multiplicative relationship into an additive relationship:

$$y_t = S_t \times T_t \times E_t \quad \Rightarrow \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

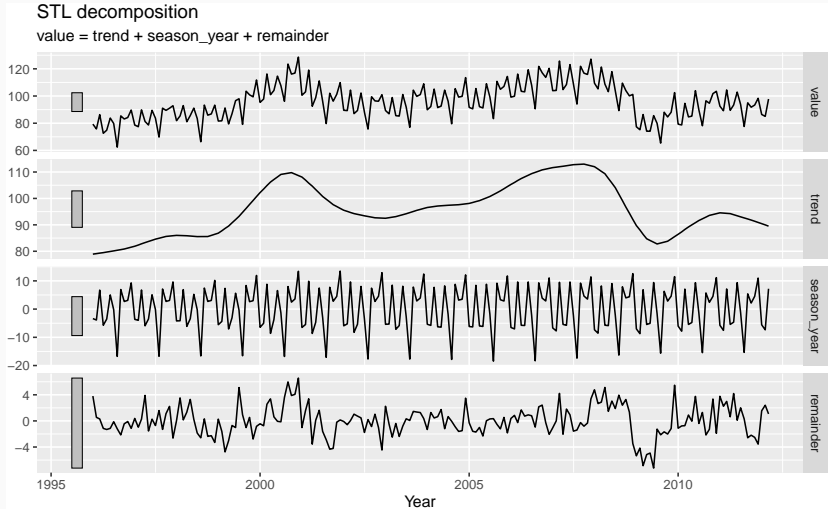
Decomposition

```
dcmp <- elecequip %>% model(STL(value ~ season(window = 7)))  
dcmp
```

```
## # A tibble: 1 x 1  
##   STL(value ~ season(window = 7))  
##   <model>  
## 1 <STL>
```

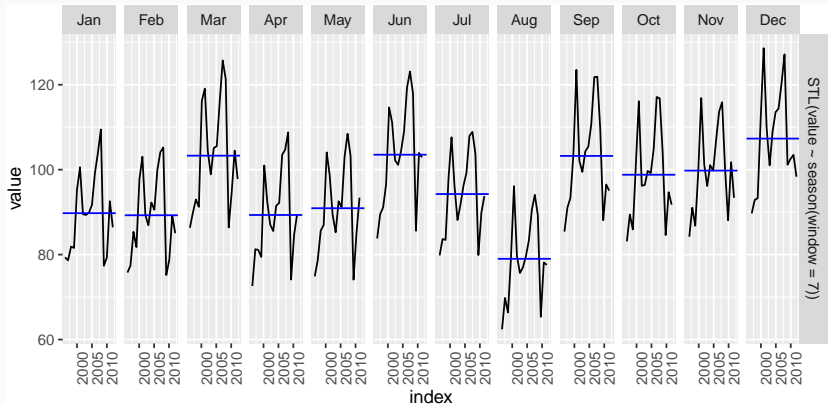

Euro electrical equipment

```
dcmp %>% components()%>% autoplot() + xlab("Year")
```



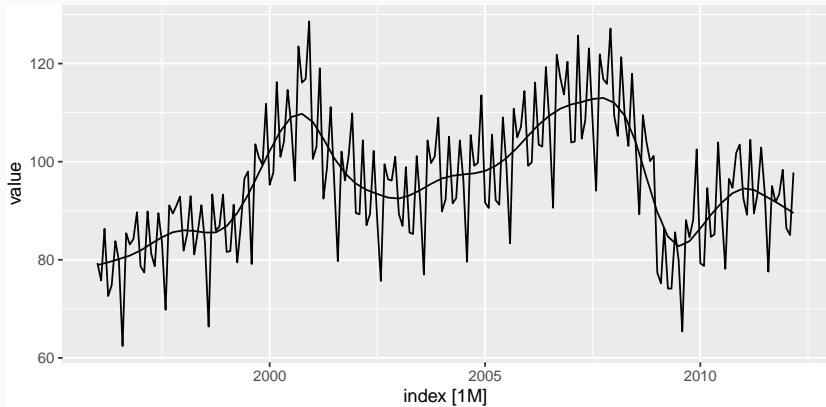
Euro electrical equipment

```
dcmp %>% components(season_year)%>%  
gg_subseries(value)
```



Euro electrical equipment

```
autoplot(elecequip) +  
  autolayer(components(dcmp), trend)
```



Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

Seasonal adjustment

- Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.
- Additive decomposition: seasonally adjusted data given by

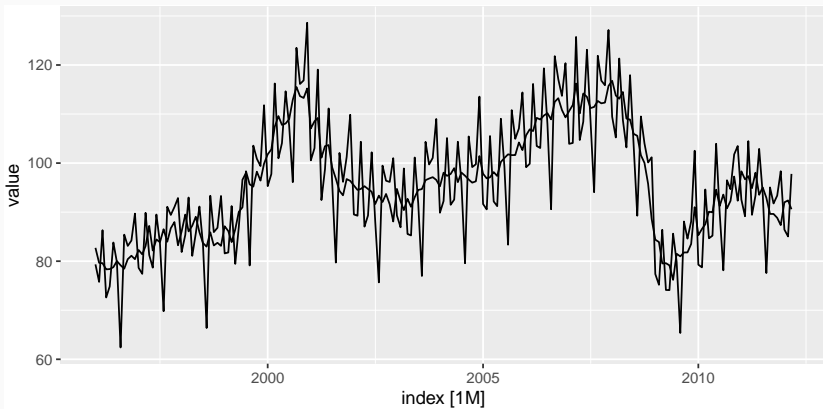
$$y_t - S_t = T_t + R_t$$

- Multiplicative decomposition: seasonally adjusted data given by

$$y_t/S_t = T_t \times R_t$$

Euro electrical equipment

```
dcmp <- elecequip %>% model(STL(value ~ season(window=7)))  
elecequip %>% autoplot(value) +  
  autolayer(components(dcmp), trend + remainder)
```



Seasonal adjustment

- If the variation due to seasonality is not of primary interest, the seasonally adjusted series can be useful. For example, monthly unemployment data are usually seasonally adjusted in order to highlight variation due to the underlying state of the economy rather than the seasonal variation.
- We use estimates of S based on past values to seasonally adjust a current value.
- Seasonally adjusted series reflect **remainders** as well as **trend**. Therefore they are not “smooth”. “Downturns” or “upturns” can be misleading.

History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.

History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.

National Statistics Offices

- ABS uses X-12-ARIMA
- US Census Bureau uses X-13ARIMA-SEATS
- Statistics Canada uses X-12-ARIMA
- ONS (UK) uses X-12-ARIMA
- EuroStat use X-13ARIMA-SEATS

Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition**
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

Moving averages

- The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle.
- A moving average of order m can be written as

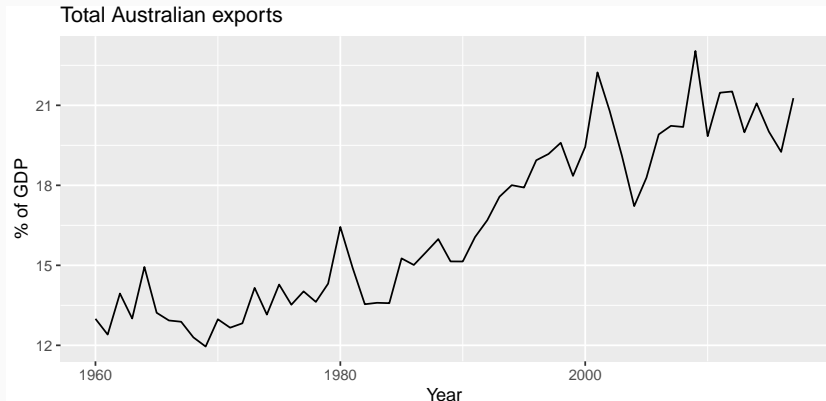
$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j}$$

where $m = 2k + 1$. We call this an m -MA.

- The estimate of the trend-cycle at time t is obtained by averaging values of the time series within k periods of t . Observations that are nearby in time are also likely to be close in value.

Australia economy plot

```
global_economy %>% filter(Country == "Australia") %>%  
  autoplot(Exports) + xlab("Year") + ylab("% of GDP") +  
  ggtitle("Total Australian exports")
```

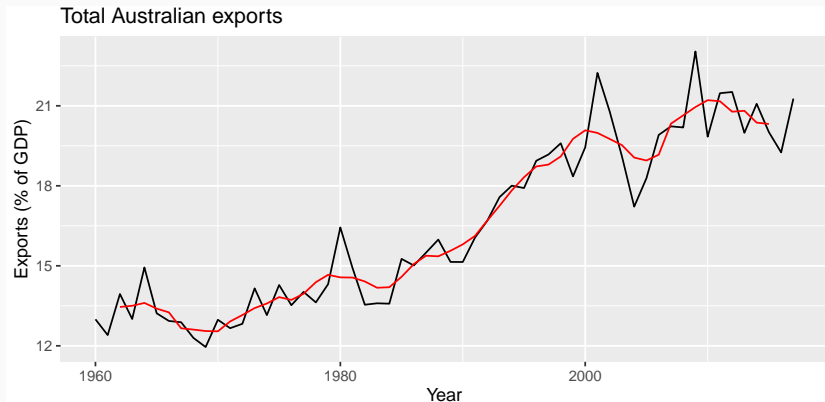


Australia economy

```
aus_exports <- global_economy %>%  
  filter(Country == "Australia") %>%  
  mutate(5-MA = slide_dbl(Exports, mean,  
    .size = 5, .align = "center"))
```

Australia economy trend

Warning: Removed 4 rows containing missing values (geom_p



Moving averages of moving averages

- m can be even but the moving average would no longer be symmetric
- It is possible to apply a moving average to a moving average. One reason for doing this is to make an even-order moving average symmetric.

```
beer <- aus_production %>%  
  filter(year(Quarter) >= 1992) %>%  
  select(Quarter, Beer)  
beer_ma <- beer %>%  
  mutate(  
    4-MA = slide_dbl(Beer, mean, .size = 4,  
      .align = "center-left"),  
    2x4-MA = slide_dbl(4-MA, mean, .size = 2,  
      .align = "center-right")  
  )
```

Moving averages of moving averages

- When a 2-MA follows a moving average of an even order (such as 4), it is called a “centred moving average of order 4”–written as 2×4 -MA.

- $$\hat{T}_t = \frac{1}{2} \left[\frac{1}{4}(y_{t-2} + y_{t-1} + y_t + y_{t+1}) + \frac{1}{4}(y_{t-1} + y_t + y_{t+1} + y_{t+2}) \right]$$

- It is now a weighted average of observations that is symmetric.

Estimating the trend-cycle with seasonal data

- The most common use of centred moving averages is for estimating the trend-cycle from seasonal data.
- For quarterly data, we can use 2×4 -MA

$$\hat{T}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}$$

Consequently, the seasonal variation will be averaged out and the resulting values of

\hat{T}_t will have little or no seasonal variation remaining.

Estimating the trend-cycle with seasonal data

- For monthly data, we can use 2×12 -MA to estimate the trend-cycle.
- For weekly data, we can use 7-MA.

Classical decomposition

- There are two forms of classical decomposition: an additive decomposition and a multiplicative decomposition.
- Let m be the seasonal period.

Classical additive decomposition

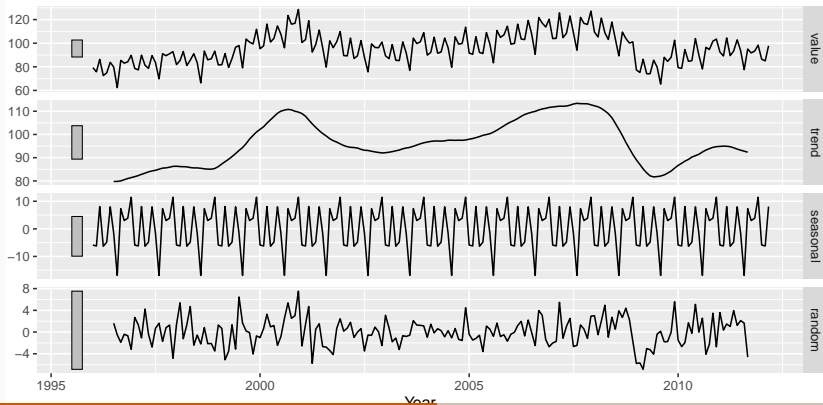
- step 1: If m is an even number, compute the trend-cycle component \hat{T}_t using $2 \times m$ -MA. If m is an odd number, use m -MA.
- Step 2: Calculate the detrended series: $T_t - \hat{T}_t$.
- Step 3: To estimate the seasonal component for each season, simply average the detrended values for that season.
- Step 4: The remainder component is calculated by subtracting the estimated seasonal and trend-cycle components.

Classical additive decomposition

```
elecequip %>% model(classical_decomposition(value,  
type = "additive")) %>% components() %>% autoplot() + xlab("  
ggtitle("Classical additive decomposition of electrical equ
```

Classical additive decomposition of electrical equipment index

value = trend + seasonal + random



Classical additive decomposition

- The trend-cycle estimate tends to over-smooth rapid rises and falls in the data.
- Classical decomposition methods assume that the seasonal component repeats from year to year. For many series, this is a reasonable assumption, but for some longer series it is not.
- The classical method is not robust to unusual values.

Outline

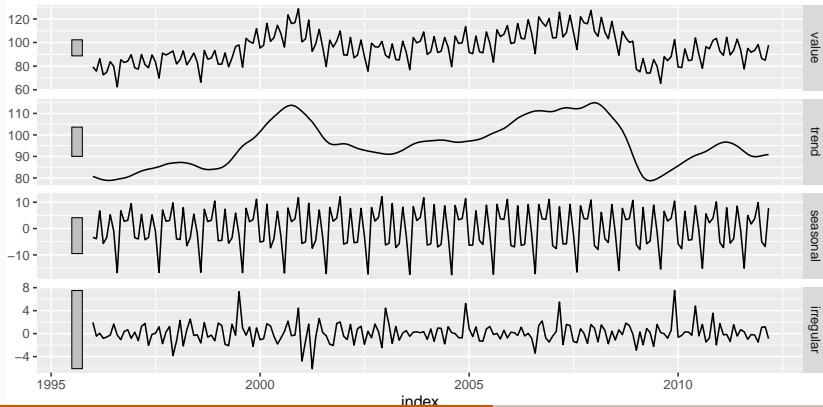
- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

X-11 decomposition

```
elecequip %>% model(feasts::X11(value,type="additive"))%>%  
  components()%>%autoplot() +  
  ggtitle("X11 decomposition of electrical equipment index")
```

X11 decomposition of electrical equipment index

value = trend + seasonal + irregular



Advantages of X-11

- Details are in Dagum, E. B., & Bianconcini, S. (2016). Seasonal adjustment methods and real time trend-cycle estimation. Springer
- Relatively robust to outliers and level shifts.
- Has some sophisticated methods for handling trading day variation, holiday effects and the effects of known predictors.
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

Disadvantages of X-11

- No prediction/confidence intervals
- Ad hoc method with no underlying model
- Only developed for quarterly and monthly data

Extensions: X-12-ARIMA and X-13-ARIMA

- The X-11, X-12-ARIMA and X-13-ARIMA methods are based on Census II decomposition.
- These allow adjustments for trading days and other explanatory variables.
- Known outliers can be omitted.
- Level shifts and ramp effects can be modelled.
- Missing values estimated and replaced.
- Holiday factors (e.g., Easter, Labour Day) can be estimated.

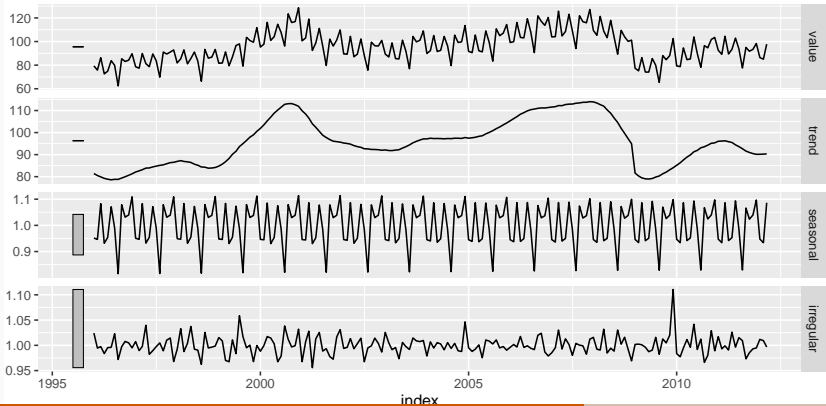
Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

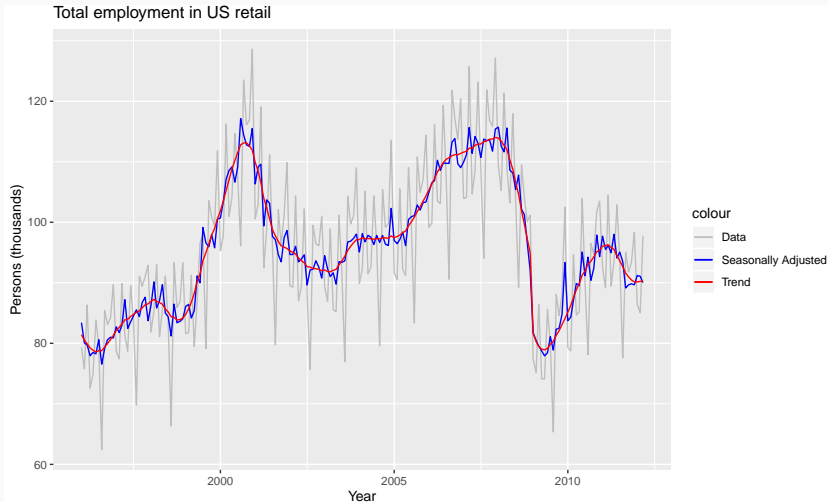
X-13ARIMA-SEATS decomposition

```
elecequip %>% model(feasts::SEATS(value))%>% components()%>%  
  autoplot() +  
  ggtitle("X-13ARIMA-SEATS decomposition of electrical equipment i
```

X-13ARIMA-SEATS decomposition of electrical equipment index
value = trend * seasonal * irregular



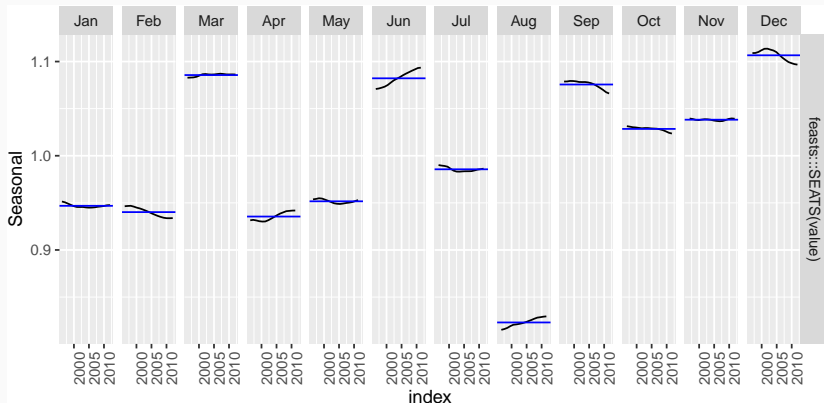
X-13ARIMA-SEATS decomposition



X-13ARIMA-SEATS decomposition

```
fit %>%
```

```
gg_subseries(seasonal) + ylab("Seasonal")
```



(Dis)advantages of X-13ARIMA-SEATS

Advantages

- Model-based
- Smooth trend estimate
- Allows estimates at end points
- Allows changing seasonality
- Developed for economic data

(Dis)advantages of X-13ARIMA-SEATS

Advantages

- Model-based
- Smooth trend estimate
- Allows estimates at end points
- Allows changing seasonality
- Developed for economic data

Disadvantages

- Only developed for quarterly and monthly data

Outline

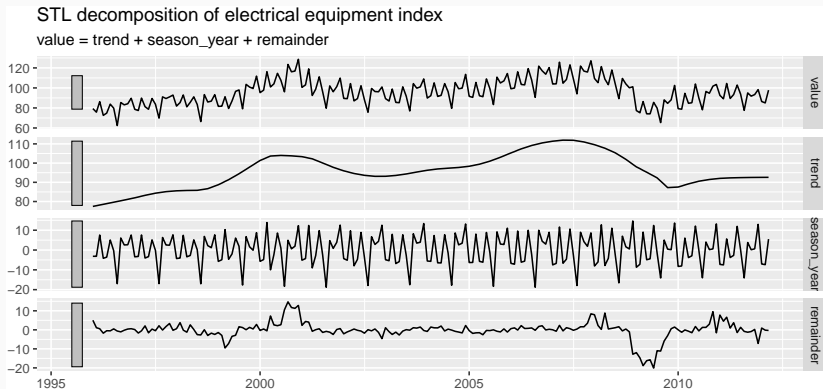
- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition**
- 7 Forecasting and decomposition

STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”
- Very versatile and robust.
- Unlike X-12-ARIMA, STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers
- Not trading day or calendar adjustments.
- Only additive.
- Take logs to get multiplicative decomposition.
- Use Box-Cox transformations to get other decompositions.

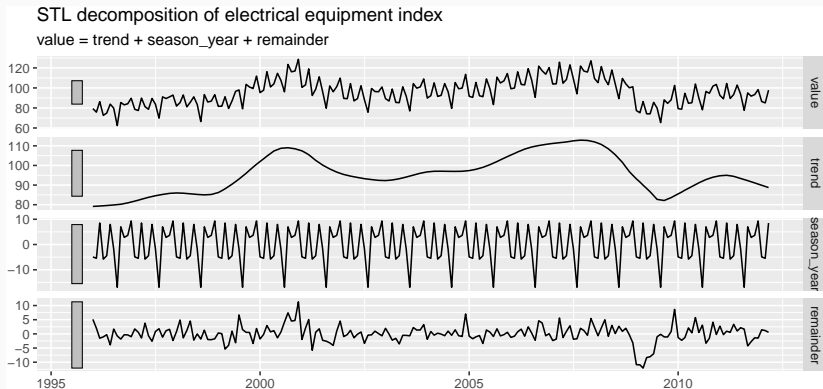
STL decomposition

```
dcmp <- elecequip %>%  
  model(STL(value ~ season(window = 5), robust = TRUE))  
autoplot(components(dcmp)) +  
  ggtitle("STL decomposition of electrical equipment index")
```



STL decomposition

```
fit <- elecequip %>%  
  model(STL(value ~ season(window="periodic"), robust=TRUE))  
autoplot(components(fit)) +  
  ggtitle("STL decomposition of electrical equipment index")
```



STL decomposition

```
elecequip %>%
```

```
  model(STL(value ~ season(window = 5)))
```

```
elecequip %>%
```

```
  model(STL(value ~ trend(window=15) + season(window="periodic"),  
         robust = TRUE))
```

STL decomposition

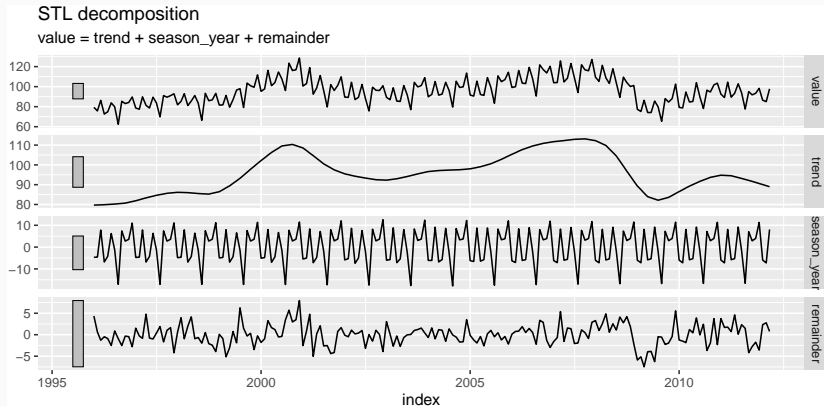
- `trend(window = ?)` controls wiggleness of trend component.
- `season(window = ?)` controls variation on seasonal component.
- Smaller values of both parameters allow for more rapid changes.
- Both trend and seasonal windows should be odd numbers;
- Trend/season window is the number of consecutive observations to be used when estimating the trend-cycle/season component.

STL decomposition

- Setting the seasonal window to be infinite is equivalent to forcing the seasonal component to be periodic `season(window='periodic')` (i.e., identical across years).
- By default, the `STL()` function provides a convenient automated STL decomposition using a seasonal window of `season(window=13)`, and the trend window chosen automatically from the seasonal period.

STL decomposition

```
elecequip %>% model(STL(value)) %>% components() %>% aut
```



- `STL()` chooses `season(window=13)` by default
- Can include transformations.

Outline

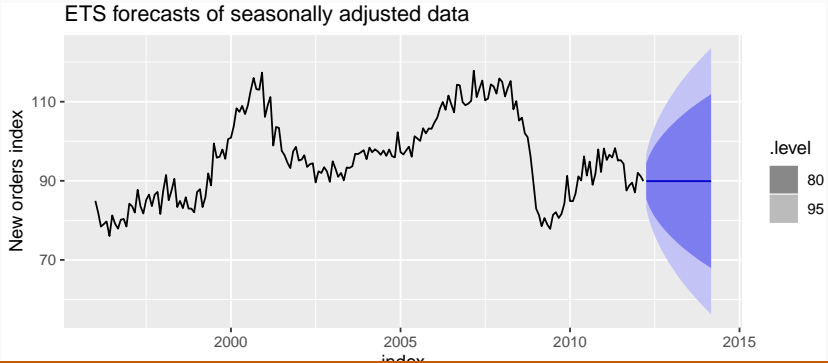
- 1 Time series components
- 2 Seasonal adjustment
- 3 Classical decomposition
- 4 X-11 decomposition
- 5 X-13ARIMA-SEATS decomposition
- 6 STL decomposition
- 7 Forecasting and decomposition

Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method.
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.

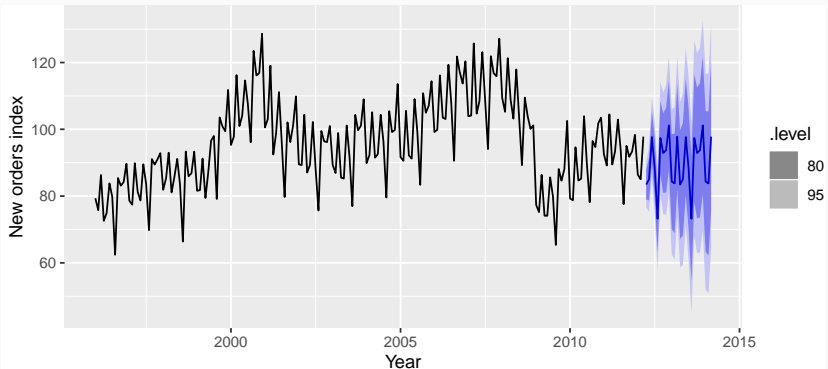
Electrical equipment

```
dcmp <- elecequip %>%  
  model(STL(value ~ trend(window=13) + season(window="periodic")))  
dcmp %>%  
  model(NAIVE(season_adjust)) %>% forecast() %>%  
  autoplot(dcmp) + ylab("New orders index") +  
  ggtitle("ETS forecasts of seasonally adjusted data")
```



Electrical equipment

```
elecequip%>% model(decomposition_model(STL(  
  value ~ trend(window = 13) + season(window = "periodic")),  
  NAIVE(season_adjust))%>% forecast() %>%  
  autoplot(elecequip) + ylab("New orders index") + xlab("Year")
```



Decomposition models

`decomposition_model()` creates a decomposition model

- You must provide a method for forecasting the `season_adjust` series.
- A seasonal naive method is used by default for the `seasonal` components.

Decomposition and prediction intervals

- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.

The `fable` package combines the uncertainty from both the seasonally adjusted and seasonal forecasts. When using `decomposition_model()` to produce forecasts, the seasonal uncertainty is not ignored!