MONASH University

MONASH BUSINESS SCHOOL

Tidy Time Series & Forecasting in R

4. Seasonality and trends

robjhyndman.com/workshop2020

Outline

1. Time series decompositions
2. Lab Session 7
3. Seasonal adjustment

Outline

1. Time series decompositions
2. Lab Session 7
3. Seasonal adjustment

Time series decomposition

Trend-Cycle aperiodic changes in level over time.

Seasonal (almost) periodic changes in level due to seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Additive decomposition

$$y_t = S_t + T_t + R_t$$

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

STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”
- Very versatile and robust.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Optionally 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.

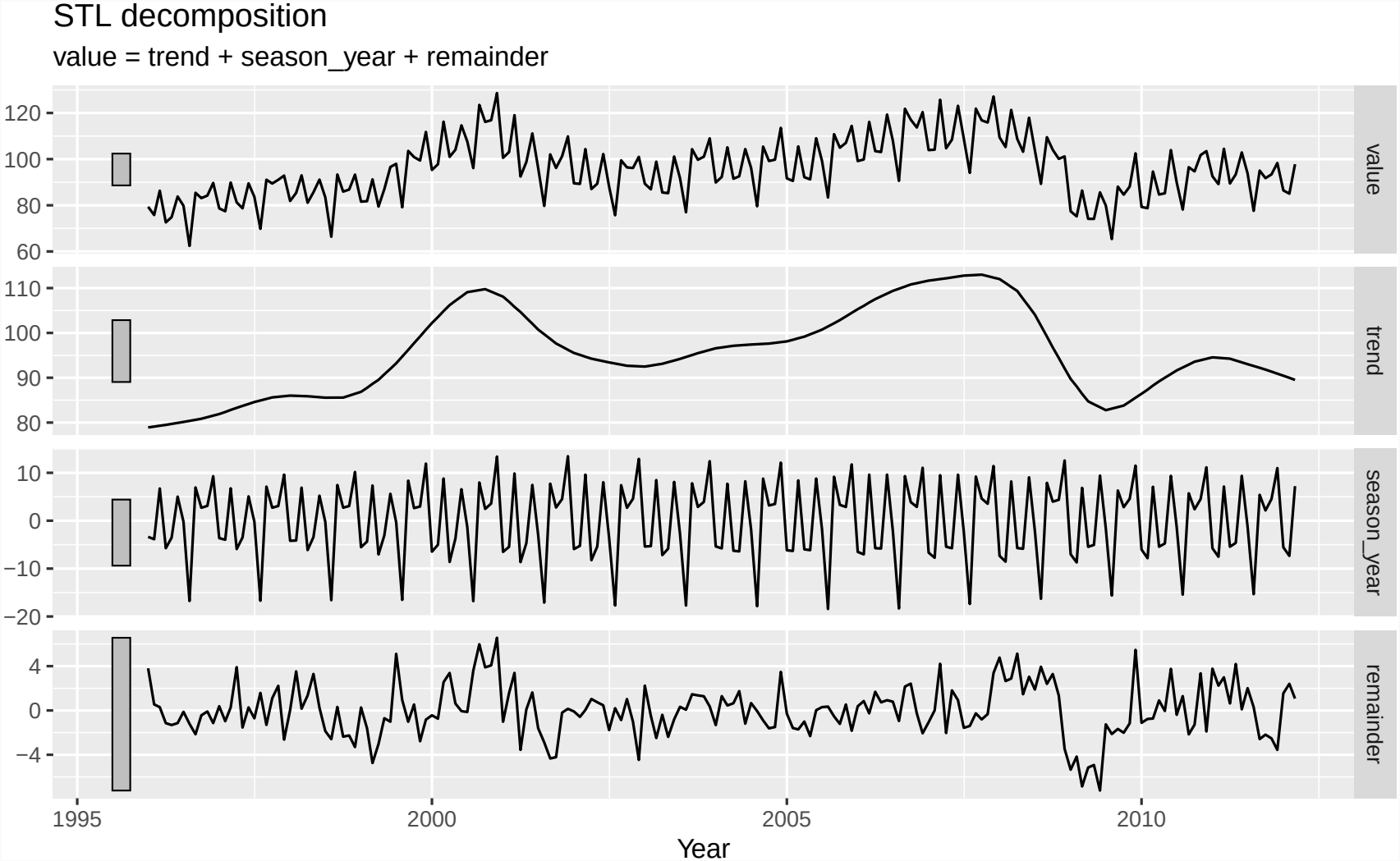
Decomposition dable

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

```
## # A dable:            195 x 6 [1M]
## # STL Decomposition: value = trend + season_year +
## #   remainder
##       index value trend season_year remainder season_adjust
##       <mth> <dbl> <dbl>       <dbl>     <dbl>          <dbl>
##  1 1996 Jan  79.4  78.9      -3.37      3.81           82.7
##  2 1996 Feb  75.8  79.1      -3.87      0.547          79.7
##  3 1996 Mar  86.3  79.3       6.73      0.301          79.6
##  4 1996 Apr  72.6  79.5      -5.74     -1.15           78.3
##  5 1996 May  74.9  79.7      -3.53     -1.31           78.4
##  6 1996 Jun  83.8  79.9       5.03     -1.14           78.8
##  7 1996 Jul  79.8  80.1      -0.222    -0.119          80.0
##  8 1996 Aug  62.4  80.4     -16.8      -1.21           79.2
##  9 1996 Sep  85.4  80.6       6.94     -2.15           78.5
## 10 1996 Oct  83.1  80.9       2.70     -0.442          80.4
```

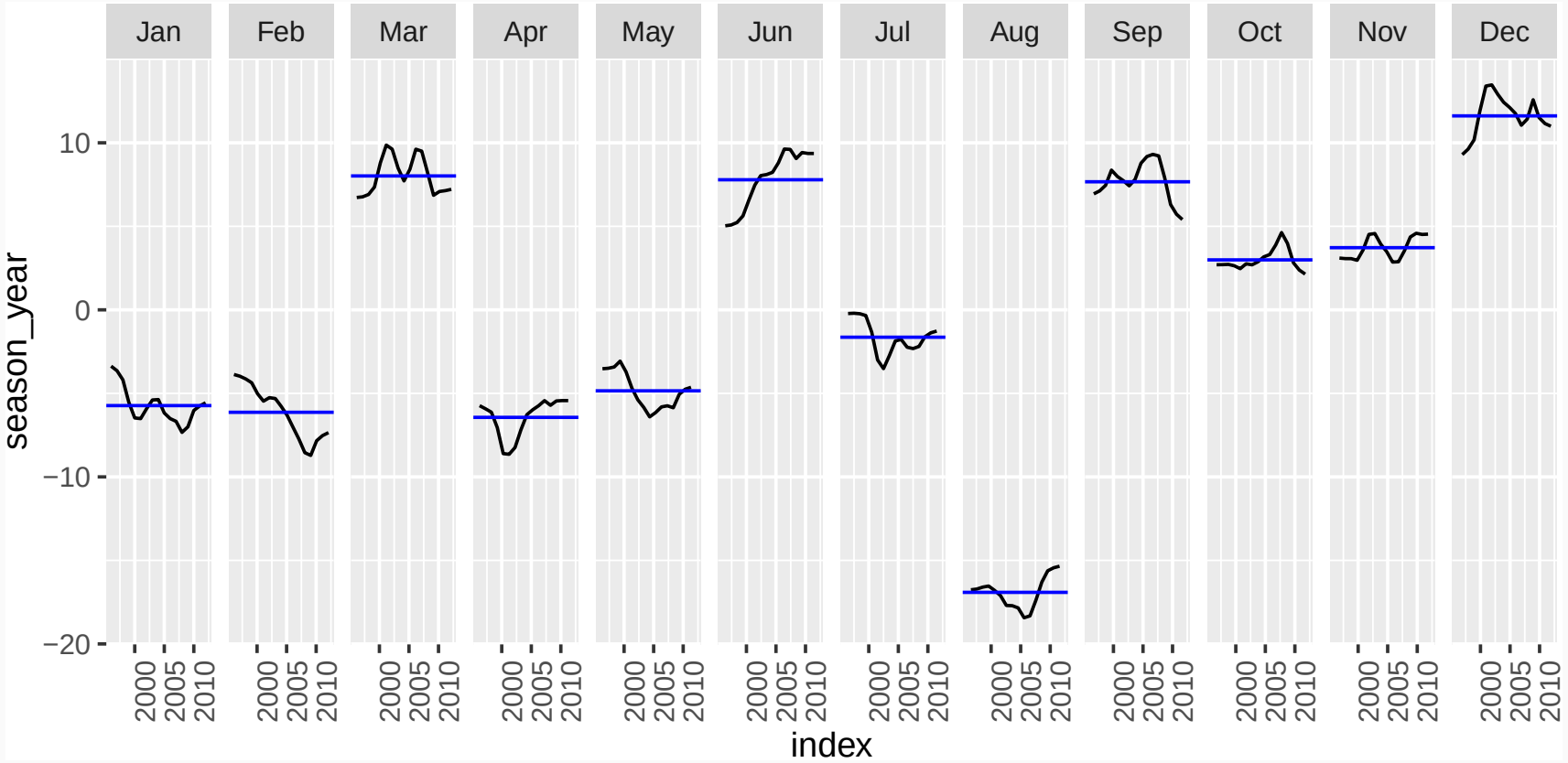
Euro electrical equipment

```
autoplot(dcmp) + xlab("Year")
```



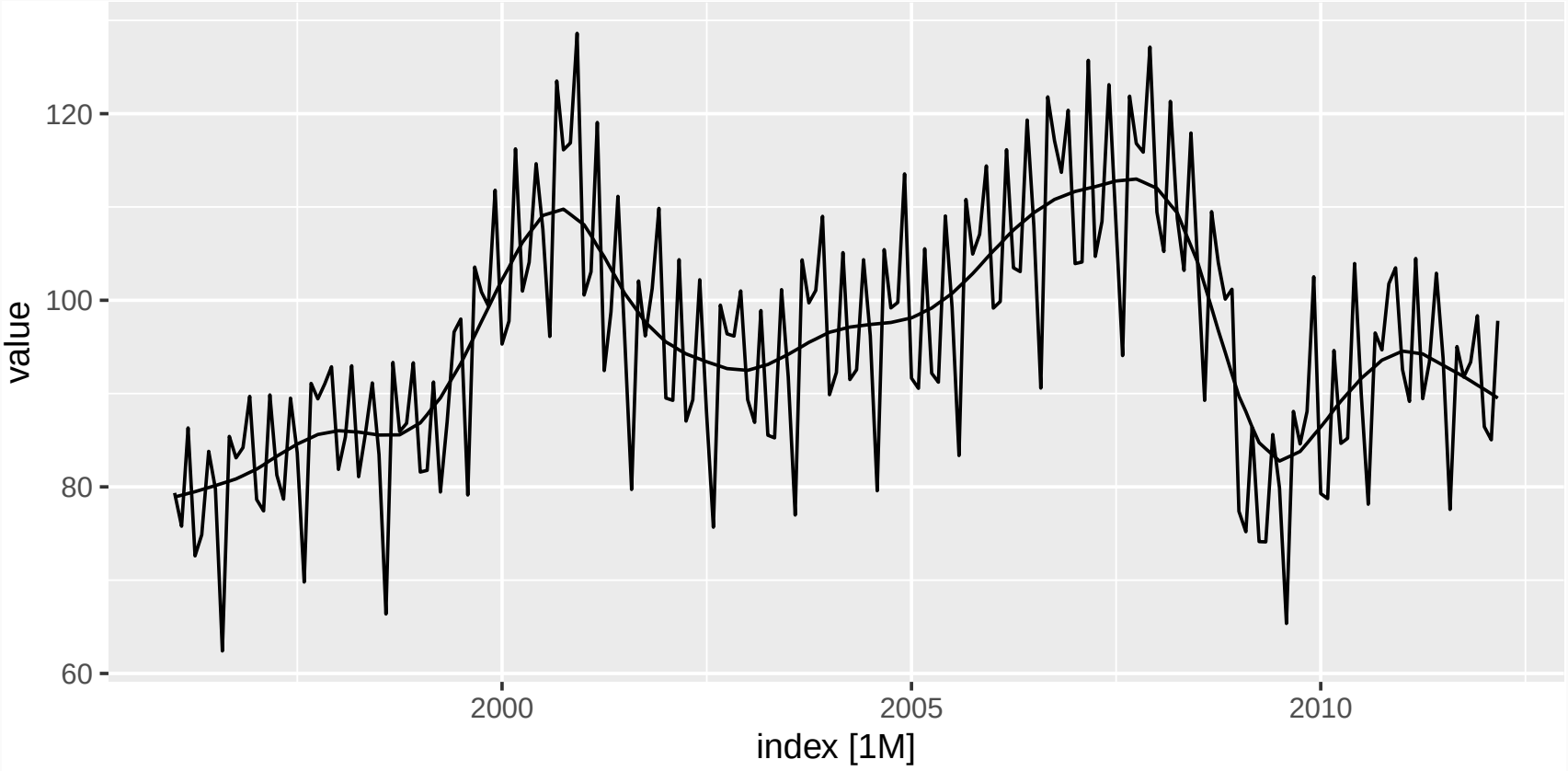
Euro electrical equipment

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



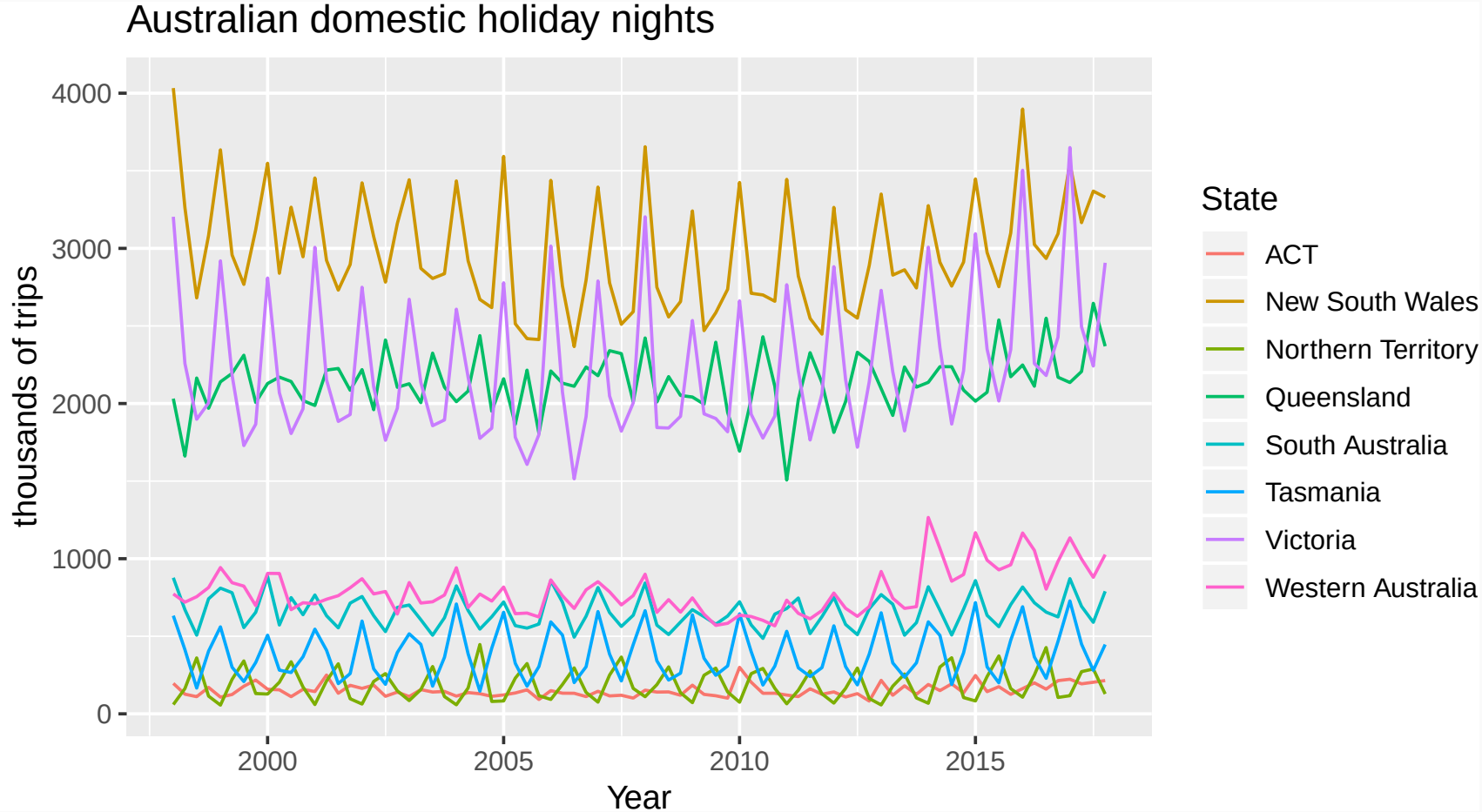
Euro electrical equipment

```
autoplot(elecequip, series="Data") +
  autolayer(dcmp, trend, series="Trend-cycle")
```



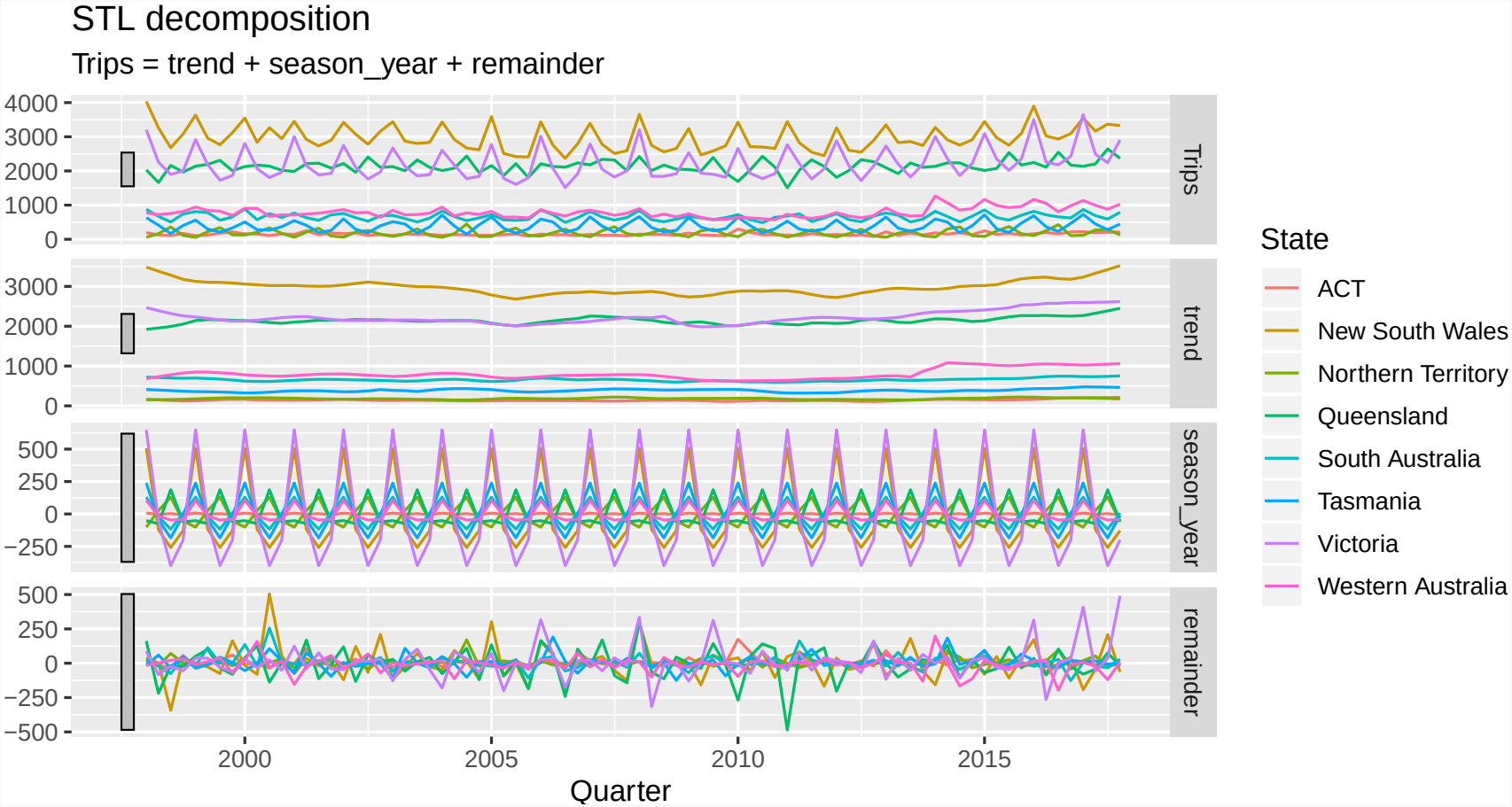
Australian holidays

```
holidays %>% autoplot(Trips) +
  ylab("thousands of trips") + xlab("Year") +
  ggtitle("Australian domestic holiday nights")
```



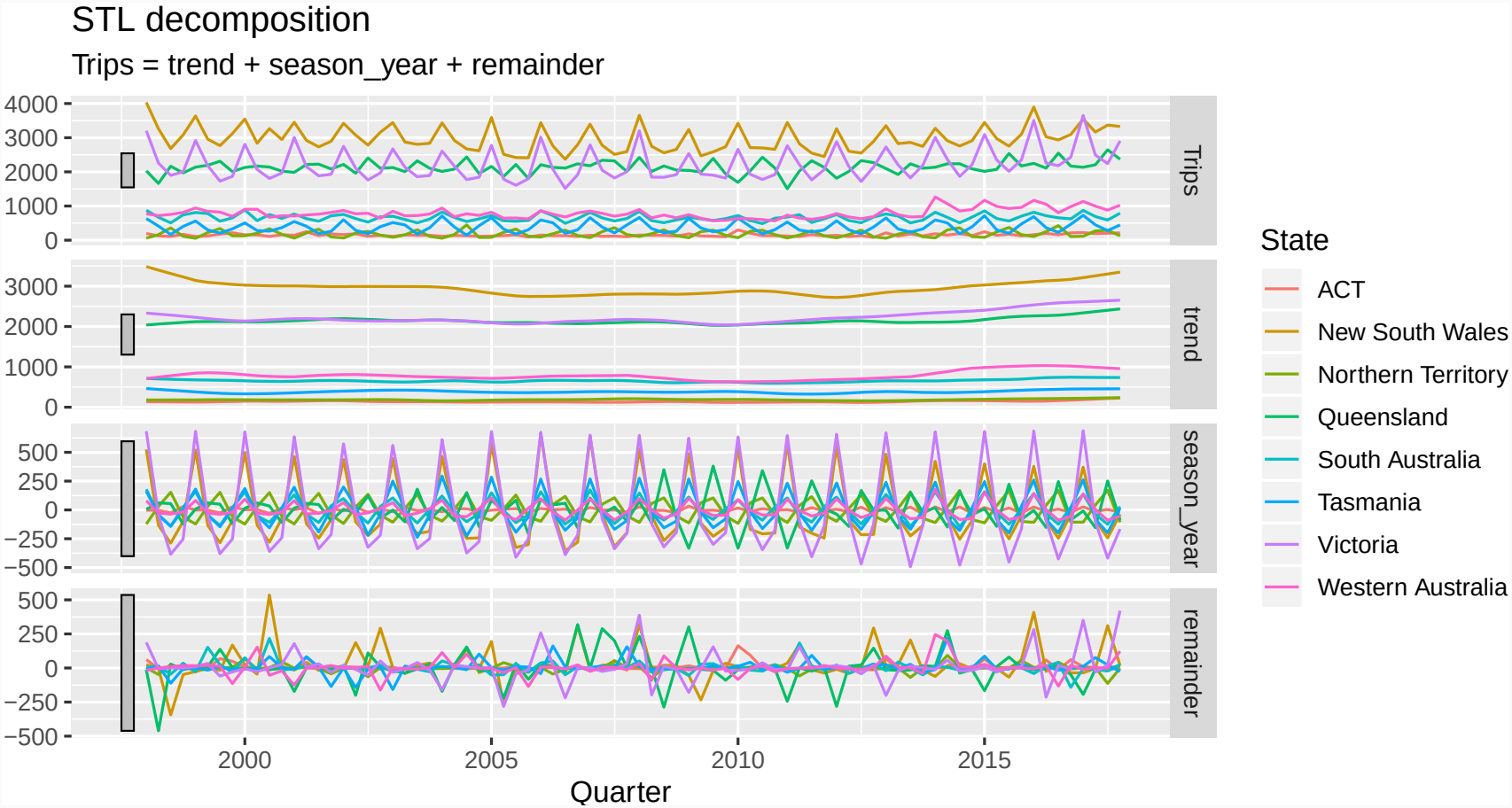
Holidays decomposition

```
holidays %>%
  STL(Trips ~ season(window="periodic"), robust=TRUE) %>%
  autoplot()
```



Holidays decomposition

```
holidays %>%
  STL(Trips ~ season(window = 5), robust = TRUE) %>%
  autoplot()
```



STL decomposition

```
holidays %>%
  STL(Trips ~ trend(window=15) + season(window=13),
      robust = TRUE)
```

- `trend(window = ?)` controls wiggliness of trend component.
- `season(window = ?)` controls variation on seasonal component.
- `STL()` chooses `season(window=13)` by default
- A large seasonal window is equivalent to setting `window="periodic"`.
- Odd numbers should be used for symmetry.

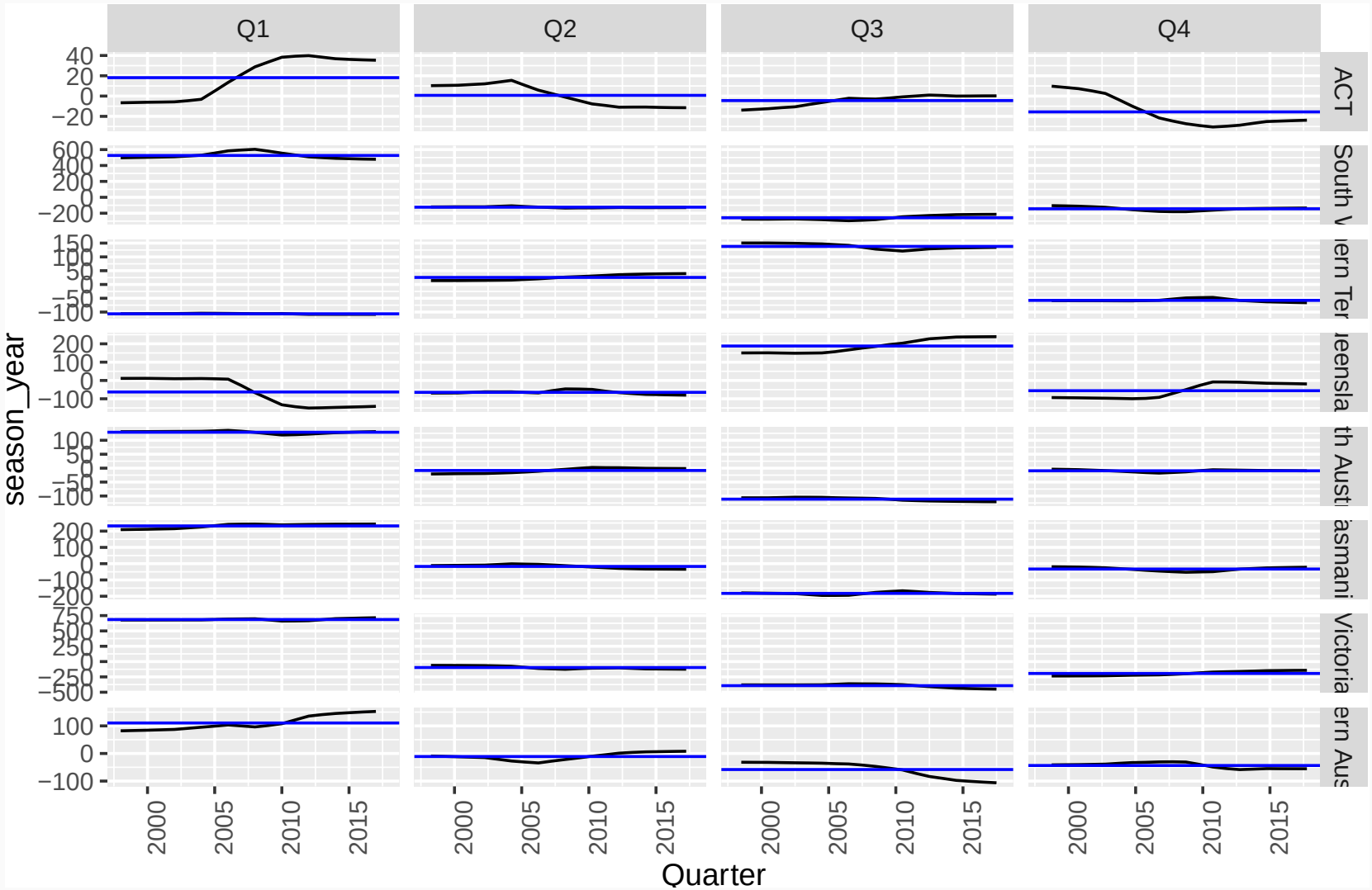
Holidays decomposition

```
dcmp <- holidays %>% STL(Trips)
dcmp
```

```
## # A dable:           640 x 7 [1Q]
## # Key:               State [8]
## # STL Decomposition: Trips = trend + season_year +
## #   remainder
##    State    Quarter Trips trend season_year remainder
##    <chr>      <qtr> <dbl> <dbl>       <dbl>     <dbl>
##  1 ACT      1998 Q1  196.  171.       -6.60     32.3
##  2 ACT      1998 Q2  127.  156.       10.3     -39.7
##  3 ACT      1998 Q3  111.  142.      -13.9     -17.2
##  4 ACT      1998 Q4  170.  130.        9.76     30.3
##  5 ACT      1999 Q1  108.  135.       -6.35    -20.7
##  6 ACT      1999 Q2  125.  148.       10.5     -33.9
##  7 ACT      1999 Q3  178.  166.      -13.2       25.5
##  8 ACT      1999 Q4  218.  177.        8.56     32.0
##  9 ACT      2000 Q1  158.  169.       -6.09     -4.74
## 10 ACT      2000 Q2  155.  151.       10.7      -7.00
```

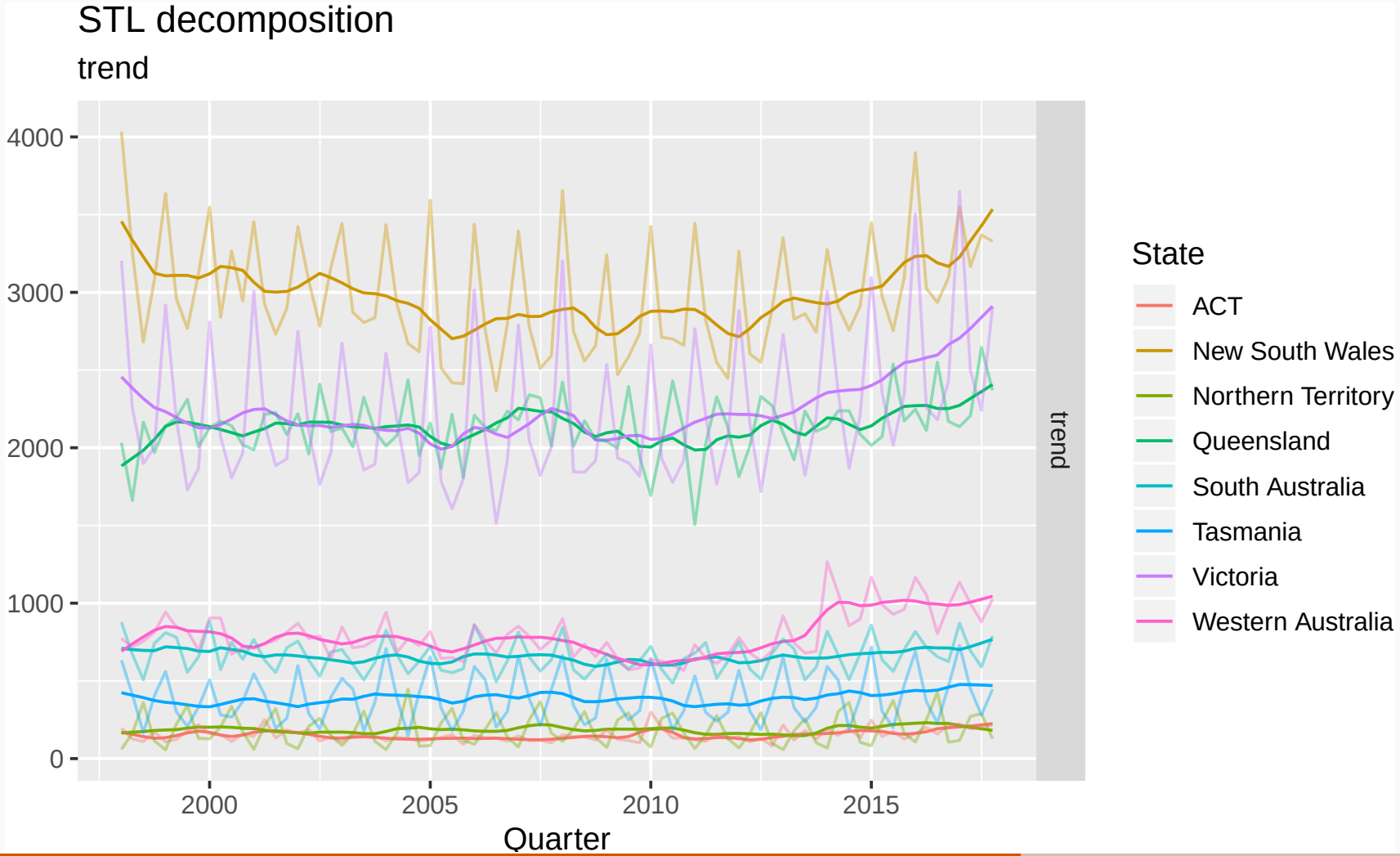
Holidays decomposition

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



Holidays decomposition

```
autoplot(dcmp, trend, scale_bars=FALSE) +
  autolayer(holidays, alpha=0.4)
```



Outline

1 Time series decompositions

2 Lab Session 7

3 Seasonal adjustment

Lab Session 7

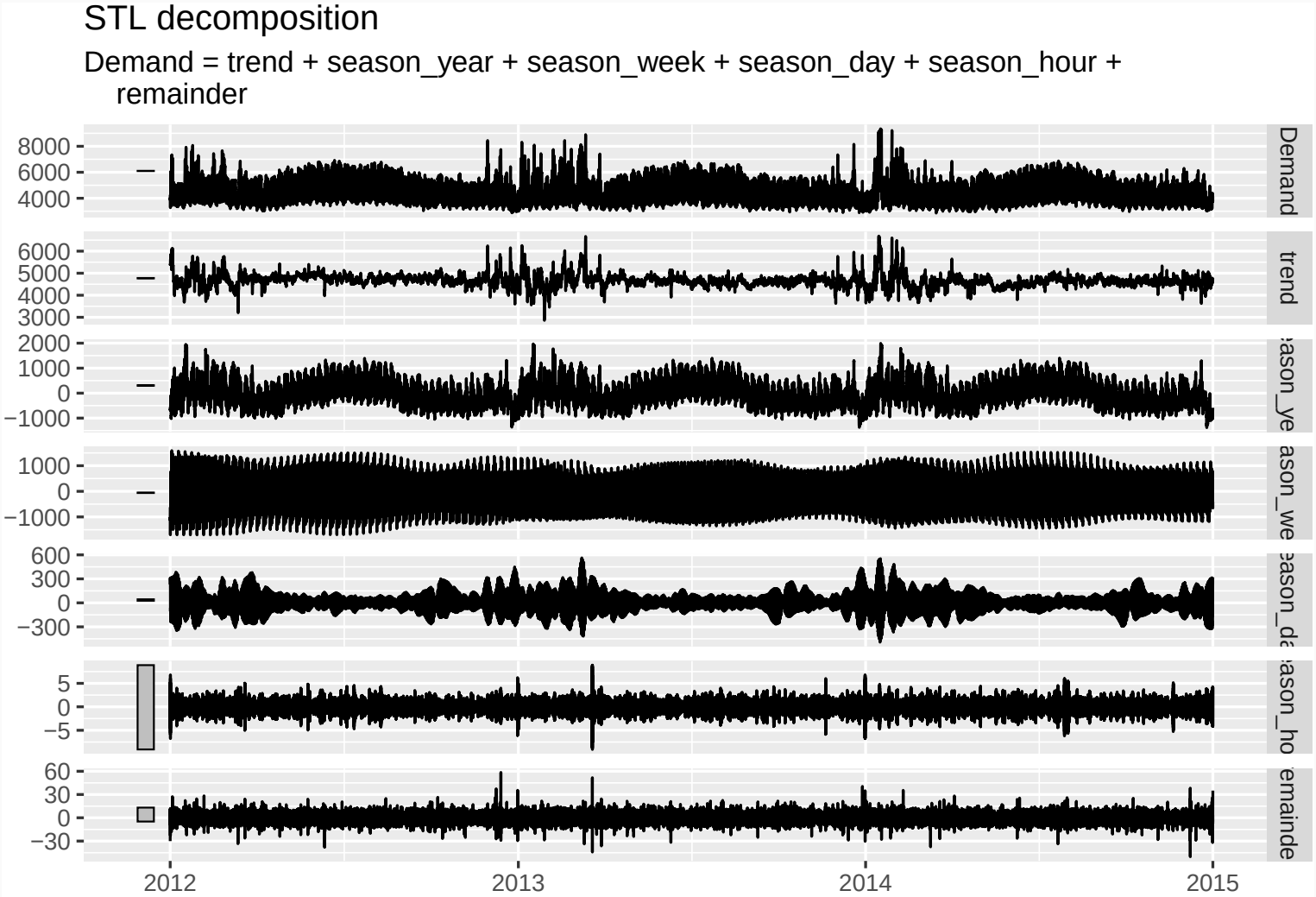
Repeat the decomposition using

```
holidays %>%
  STL(Trips ~ season(window=7) + trend(window=11)) %>%
  autoplot()
```

What happens as you change `season(window = ???)` and `trend(window = ???)`?

Multiple seasonality

```
vic_elec %>% STL(Demand) %>% autoplot()
```



Outline

1 Time series decompositions

2 Lab Session 7

3 Seasonal adjustment

Seasonal adjustment

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

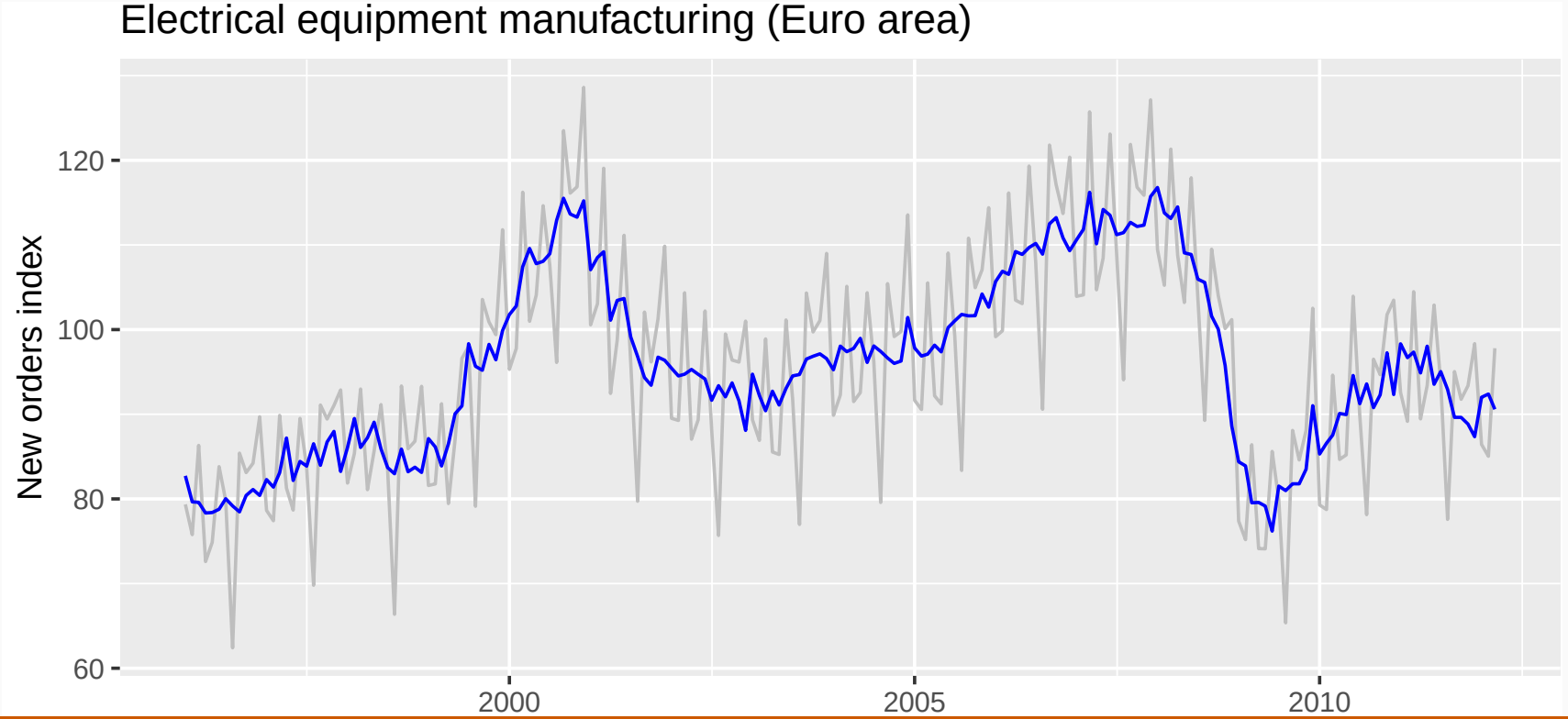
$$y_t - S_t = T_t + R_t$$

- Multiplicative decomposition: seasonally adjusted data given by

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

Euro electrical equipment

```
dcmp <- elecequip %>% STL(value ~ season(window=7))
elecequip %>% autoplot(value, col='gray') +
    autolayer(dcmp, season_adjust, col='blue') +
    xlab("Year") + ylab("New orders index") +
    ggtitle("Electrical equipment manufacturing (Euro area)"
```



Seasonal adjustment

- 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”" and “downturns”" or “upturns” can be misleading.
- It is better to use the trend-cycle component to look for turning points.