

ETC3550/ETC5550

Applied forecasting

Ch3. Time series decomposition

OTexts.org/fpp3/



Outline

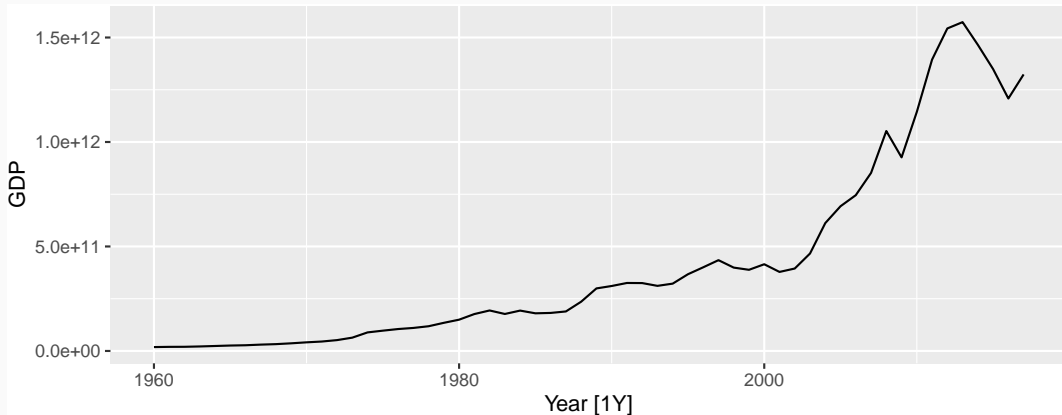
- 1 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong

Outline

- 1 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong

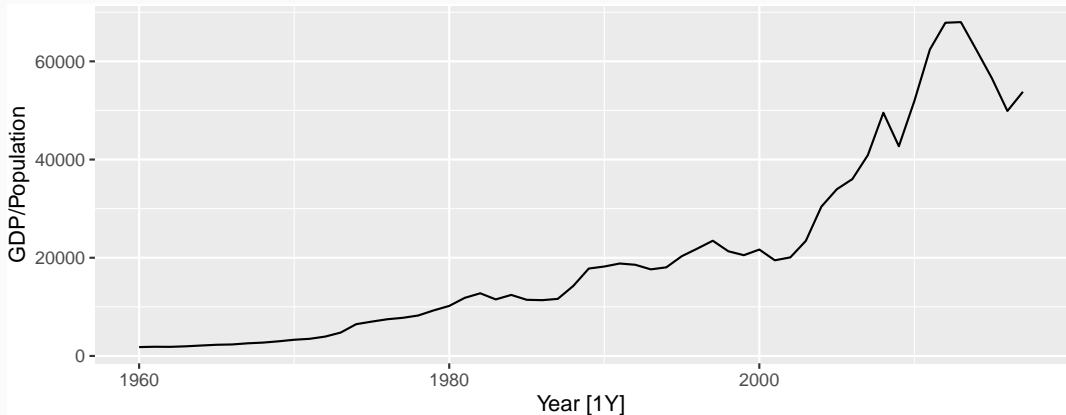
Per capita adjustments

```
global_economy %>%  
  filter(Country == "Australia") %>%  
  autoplot(GDP)
```



Per capita adjustments

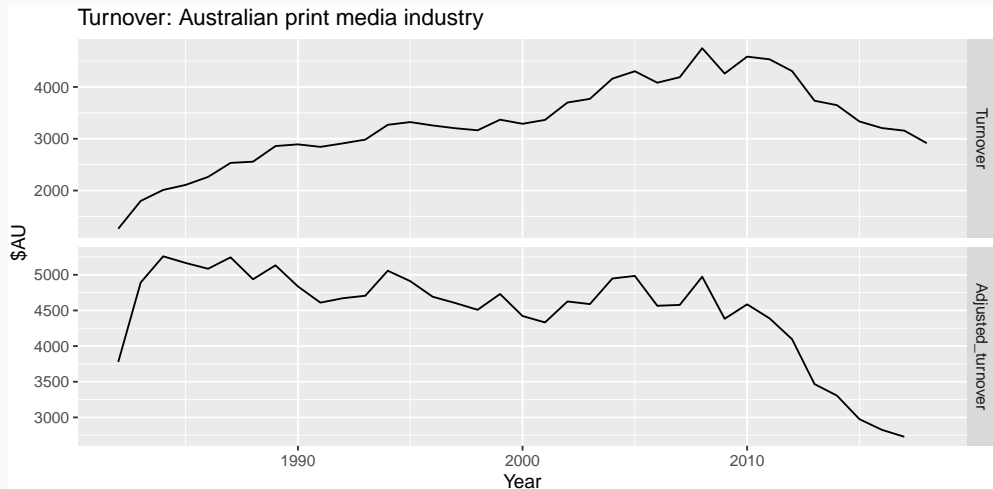
```
global_economy %>%  
  filter(Country == "Australia") %>%  
  autoplot(GDP / Population)
```



Inflation adjustments

```
print_retail <- aus_retail %>%  
  filter(Industry == "Newspaper and book retailing") %>%  
  group_by(Industry) %>%  
  index_by(Year = year(Month)) %>%  
  summarise(Turnover = sum(Turnover))  
aus_economy <- global_economy %>%  
  filter(Code == "AUS")  
print_retail %>%  
  left_join(aus_economy, by = "Year") %>%  
  mutate(Adjusted_turnover = Turnover / CPI * 100) %>%  
  pivot_longer(c(Turnover, Adjusted_turnover), values_to = "Turnover") %>%  
  mutate(name = factor(name, levels=c("Turnover", "Adjusted_turnover"))) %>%  
  ggplot(aes(x = Year, y = Turnover)) +  
  geom_line() +  
  facet_grid(name ~ ., scales = "free_y") +  
  labs(title = "Turnover: Australian print media industry", y = "$AU")
```

Inflation adjustments



Mathematical transformations

If the data show different variation at different levels of the series, then a transformation can be useful.

Mathematical transformations

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as y_1, \dots, y_T and transformed observations as w_1, \dots, w_T .

Mathematical transformations

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as y_1, \dots, y_T and transformed observations as w_1, \dots, w_T .

Mathematical transformations for stabilizing variation

Square root	$w_t = \sqrt{y_t}$	\downarrow
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

Mathematical transformations

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as y_1, \dots, y_T and transformed observations as w_1, \dots, w_T .

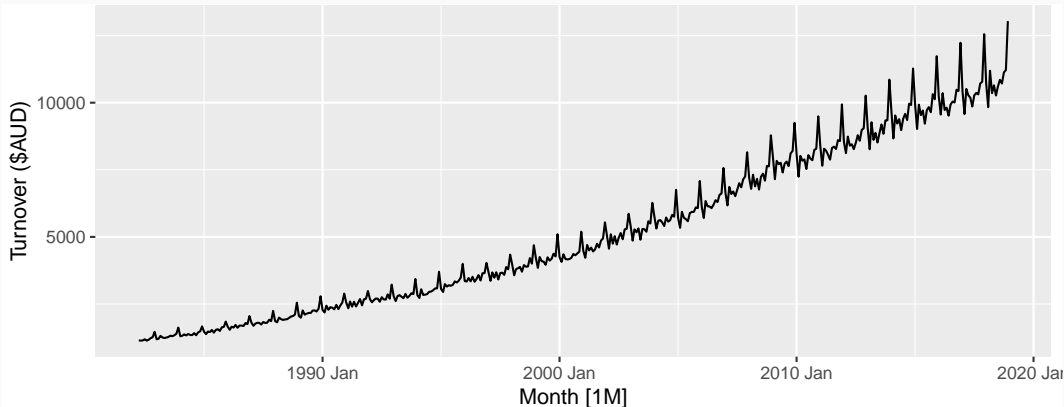
Mathematical transformations for stabilizing variation

Square root	$w_t = \sqrt{y_t}$	\downarrow
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are **relative (percent) changes on the original scale**.

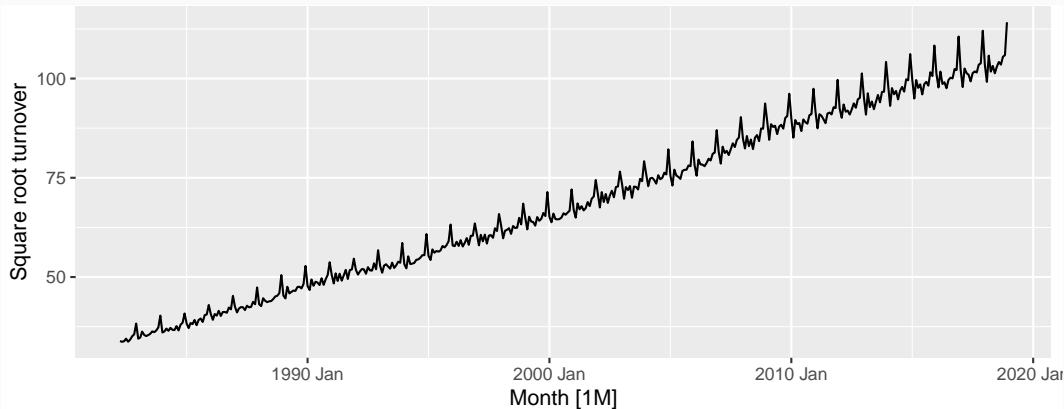
Mathematical transformations

```
food <- aus_retail %>%  
  filter(Industry == "Food retailing") %>%  
  summarise(Turnover = sum(Turnover))
```



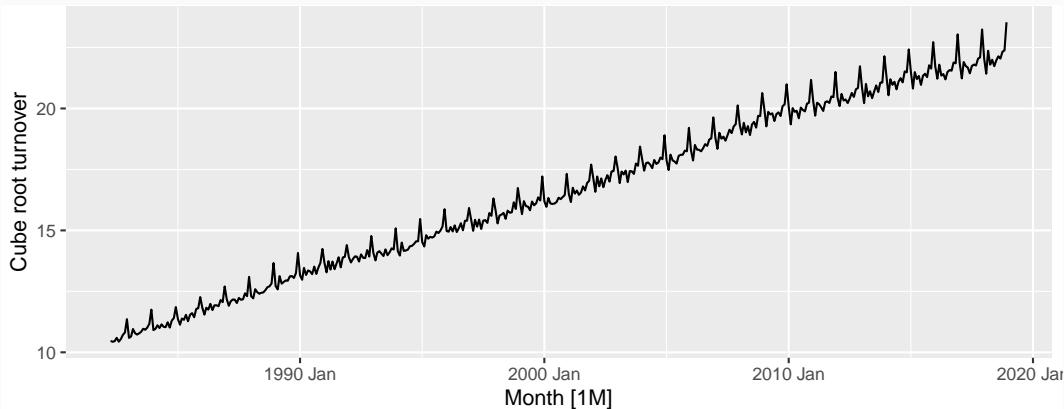
Mathematical transformations

```
food %>% autoplot(sqrt(Turnover)) +  
  labs(y = "Square root turnover")
```



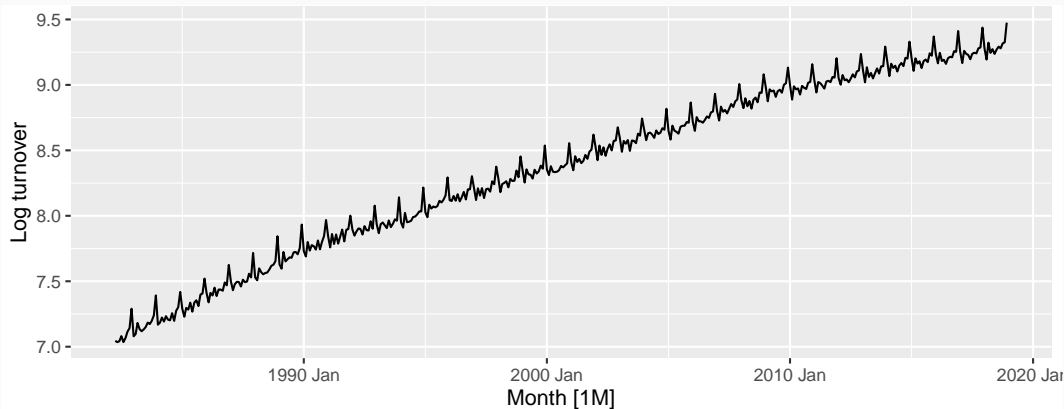
Mathematical transformations

```
food %>% autoplot(Turnover^(1/3)) +  
  labs(y = "Cube root turnover")
```



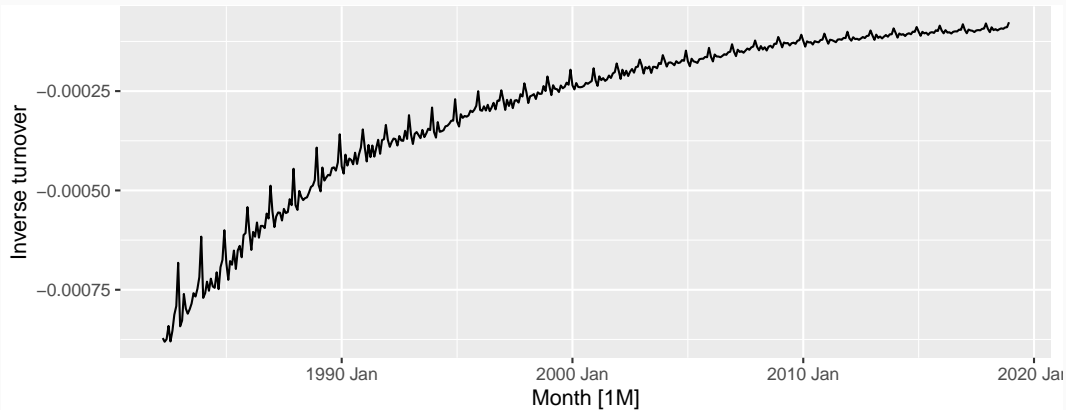
Mathematical transformations

```
food %>% autoplot(log(Turnover)) +  
  labs(y = "Log turnover")
```



Mathematical transformations

```
food %>% autoplot(-1/Turnover) +  
  labs(y = "Inverse turnover")
```



Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (\text{sign}(y_t)|y_t|^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (\text{sign}(y_t)|y_t|^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

- Actually the Bickel-Doksum transformation (allowing for $y_t < 0$)
- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Box-Cox transformations

Box-Cox transformations

```
food %>%  
  features(Turnover, features = guerrero)
```

```
## # A tibble: 1 x 1  
##   lambda_guerrero  
##               <dbl>  
## 1              0.0895
```

Box-Cox transformations

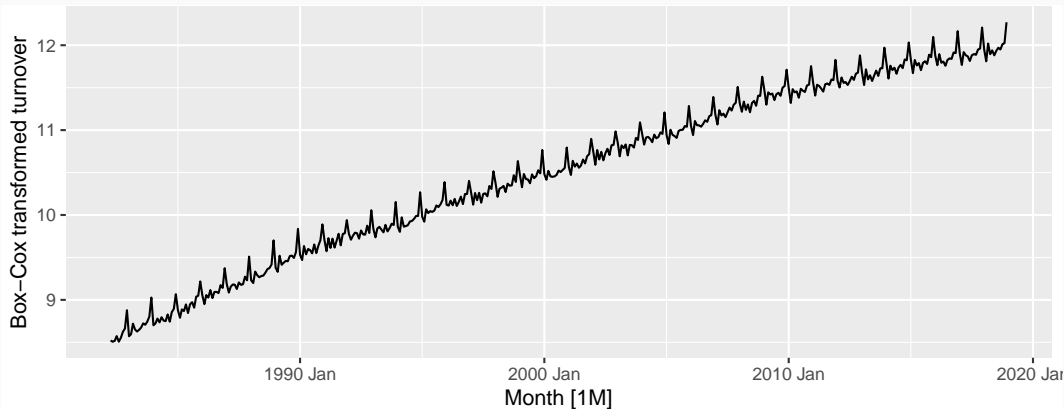
```
food %>%  
  features(Turnover, features = guerrero)
```

```
## # A tibble: 1 x 1  
##   lambda_guerrero  
##               <dbl>  
## 1               0.0895
```

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of λ can give extremely large prediction intervals.

Box-Cox transformations

```
food %>% autoplot(box_cox(Turnover, 0.0524)) +  
  labs(y = "Box-Cox transformed turnover")
```



Transformations

- Often no transformation needed.
- Simple transformations are easier to explain and work well enough.
- Transformations can have very large effect on PI.
- If some data are zero or negative, then use $\lambda > 0$.
- `log1p()` can also be useful for data with zeros.
- Choosing logs is a simple way to force forecasts to be positive
- Transformations must be reversed to obtain forecasts on the original scale. (Handled automatically by `fab1e`.)

Outline

- 1 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong

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:

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

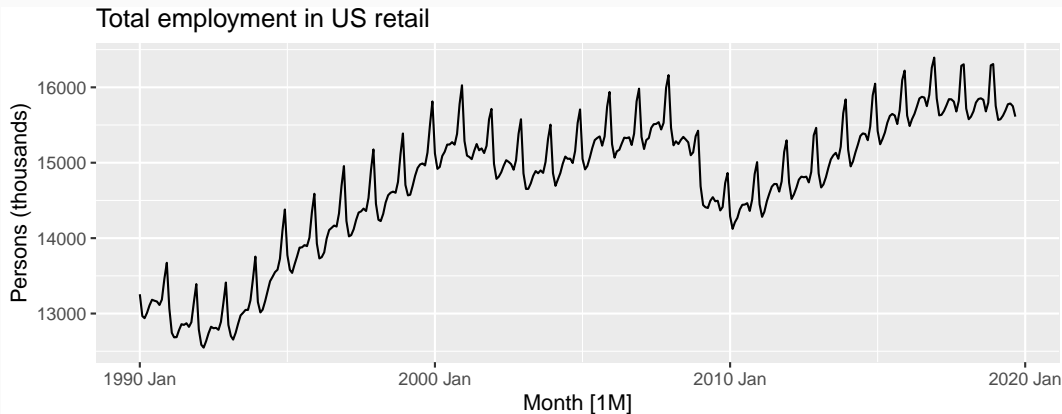
US Retail Employment

```
us_retail_employment <- us_employment %>%  
  filter(year(Month) >= 1990, Title == "Retail Trade") %>%  
  select(-Series_ID)  
us_retail_employment
```

```
## # A tsibble: 357 x 3 [1M]  
##      Month Title      Employed  
##      <mth> <chr>      <dbl>  
## 1 1990 Jan Retail Trade 13256.  
## 2 1990 Feb Retail Trade 12966.  
## 3 1990 Mar Retail Trade 12938.  
## 4 1990 Apr Retail Trade 13012.  
## 5 1990 May Retail Trade 13108.  
## 6 1990 Jun Retail Trade 13183.  
## 7 1990 Jul Retail Trade 13170.  
## 8 1990 Aug Retail Trade 13160.  
## 9 1990 Sep Retail Trade 13112.
```

US Retail Employment

```
us_retail_employment %>%  
  autoplot(Employed) +  
  labs(y="Persons (thousands)", title="Total employment in US retail")
```



US Retail Employment

```
us_retail_employment %>%  
  model(stl = STL(Employed))
```

```
## # A mable: 1 x 1  
##      stl  
##    <model>  
## 1    <STL>
```

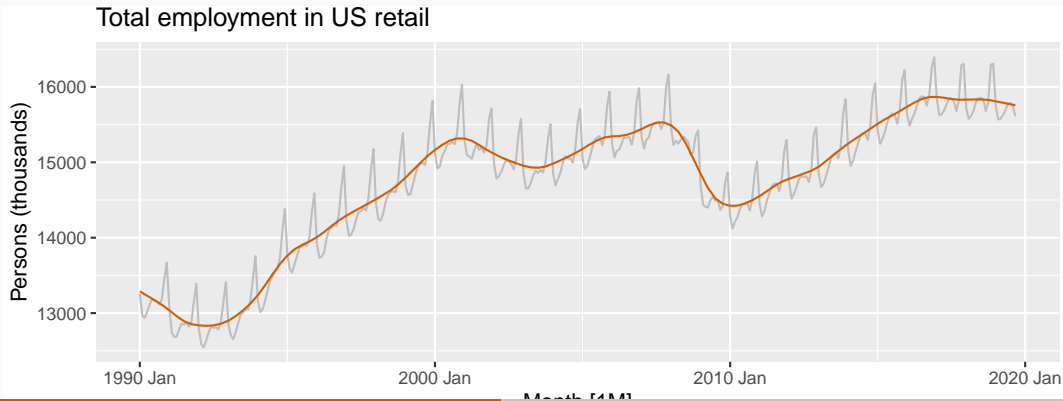
US Retail Employment

```
dcmp <- us_retail_employment %>%  
  model(stl = STL(Employed))  
components(dcmp)
```

```
## # A dable: 357 x 7 [1M]  
## # Key:      .model [1]  
## # :      Employed = trend + season_year + remainder  
##   .model      Month Employed  trend season_year remainder season_adjust  
##   <chr>      <mth>    <dbl>  <dbl>      <dbl>      <dbl>      <dbl>  
## 1 stl      1990 Jan    13256. 13288.    -33.0      0.836     13289.  
## 2 stl      1990 Feb    12966. 13269.   -258.     -44.6     13224.  
## 3 stl      1990 Mar    12938. 13250.   -290.     -22.1     13228.  
## 4 stl      1990 Apr    13012. 13231.   -220.      1.05     13232.  
## 5 stl      1990 May    13108. 13211.   -114.     11.3     13223.  
## 6 stl      1990 Jun    13183. 13192.   -24.3     15.5     13207.  
## 7 stl      1990 Jul    13170. 13172.   -23.2     21.6     13193.26  
## 8 stl      1990 Aug    13160. 13151.    -9.5     17.0     13160.
```

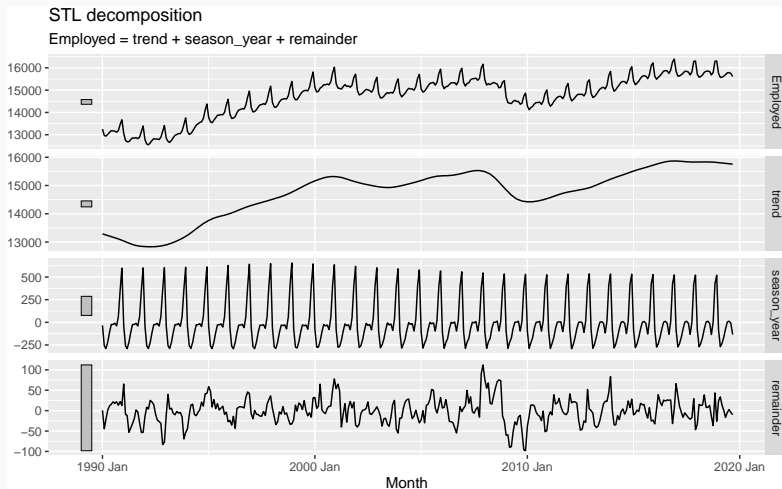

US Retail Employment

```
us_retail_employment %>%  
  autoplot(Employed, color='gray') +  
  autolayer(components(dcmp), trend, color='#D55E00') +  
  labs(y="Persons (thousands)", title="Total employment in US retail")
```



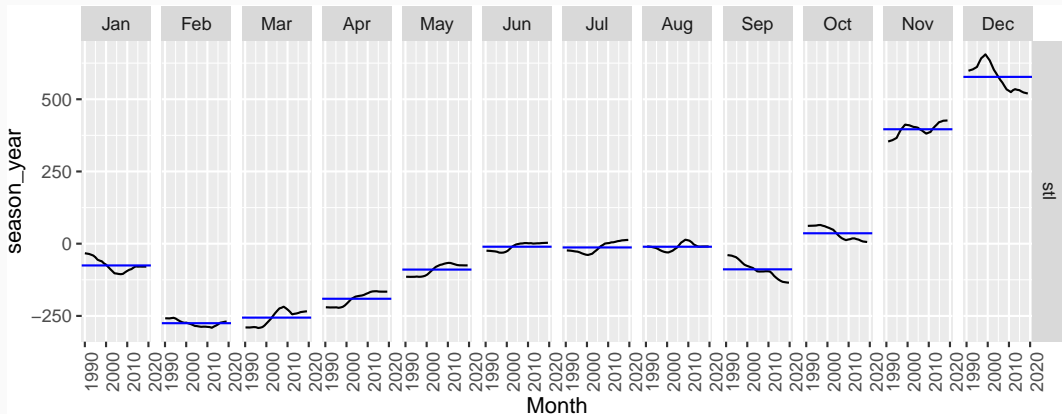
US Retail Employment

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



US Retail Employment

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



Seasonal adjustment

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

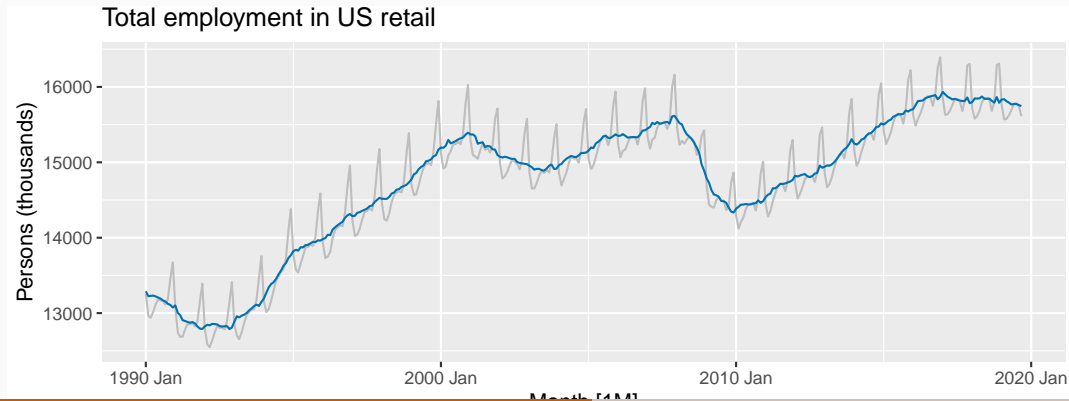
$$y_t - S_t = T_t + R_t$$

- Multiplicative decomposition: seasonally adjusted data given by

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

US Retail Employment

```
us_retail_employment %>%  
  autoplot(Employed, color='gray') +  
  autolayer(components(dcmp), season_adjust, color='#0072B2') +  
  labs(y="Persons (thousands)", title="Total employment in US retail")
```



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.

Outline

- 1 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong

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

X-11 decomposition

Advantages

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

X-11 decomposition

Advantages

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

Disadvantages

- 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.

Advantages

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

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 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong

STL decomposition

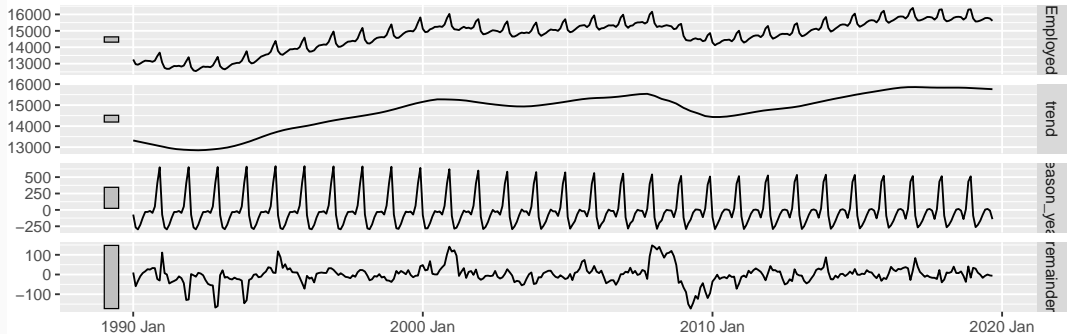
- STL: “Seasonal and Trend decomposition using Loess”
- Very versatile and robust.
- Unlike X-12-ARIMA, STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers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

STL decomposition

```
us_retail_employment %>%  
  model(STL(Employed ~ season(window=9), robust=TRUE)) %>%  
  components() %>% autoplot() +  
    labs(title = "STL decomposition: US retail employment")
```

STL decomposition: US retail employment

Employed = trend + season_year + remainder



STL decomposition

STL decomposition

```
us_retail_employment %>%  
  model(STL(Employed ~ season(window=5))) %>%  
  components()  
  
us_retail_employment %>%  
  model(STL(Employed ~ trend(window=15) +  
            season(window="periodic"),  
        robust = TRUE)  
  ) %>% components()
```

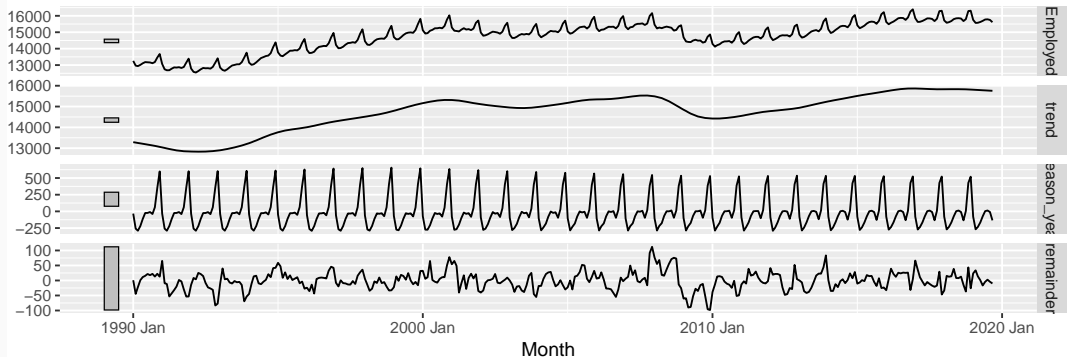
- `trend(window = ?)` controls wiggleness of trend component.
- `season(window = ?)` controls variation on seasonal component.
- `season(window = 'periodic')` is equivalent to an infinite window.

STL decomposition

```
us_retail_employment %>%  
  model(STL(Employed)) %>%  
  components() %>% autoplot()
```

STL decomposition

Employed = trend + season_year + remainder



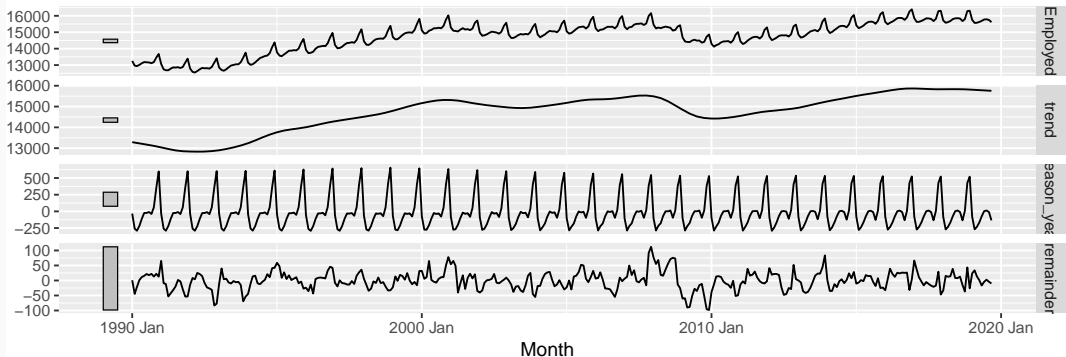
STL decomposition

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

```
us_retail_employment %>%  
  model(STL(Employed)) %>%  
  components() %>% autoplot()
```

STL decomposition

Employed = trend + season_year + remainder




STL decomposition

- Algorithm that updates trend and seasonal components iteratively.
- Starts with $\hat{T}_t = 0$
- Uses a mixture of loess and moving averages to successively refine the trend and seasonal estimates.
- The trend window controls loess bandwidth applied to deasonalised values.
- The season window controls loess bandwidth applied to detrended subseries.
- Robustness weights based on remainder.
- Default season window = 13
- Default trend window = `nextodd(`
`ceiling((1.5*period)/(1-(1.5/s.window)))`


Outline

- 1 Transformations and adjustments
- 2 Time series components
- 3 History of time series decomposition
- 4 STL decomposition
- 5 When things go wrong






The ABS stuff-up

NEWS 

LOCATION:
📍 Clayton, Vic [Change ▾](#)

 Just In [Australia](#) [World](#) [Business](#) [Sport](#) [Analysis & Opinion](#) [Fact Check](#) [Programs](#)

BREAKING NEWS Police arrest man in connection with stabbing death of 17-year-old Masa Vukotic in M

 Print  Email  Facebook  Twitter  More

Treasurer Joe Hockey calls for answers over Australian Bureau of Statistics jobs data

By [Michael Vincent](#) and [Simon Frazer](#)

Updated 9 Oct 2014, 12:17pm

Federal Treasurer Joe Hockey says he wants answers to the problems the Australian Bureau of Statistics (ABS) has had with unemployment figures.

Mr Hockey, who is in the US to discuss Australia's G20 agenda, said last month's unemployment figures were "extraordinary".

The rate was 6.1 per cent after jumping to a 12-year high of 6.4 per cent the previous month.


The ABS has now taken the rare step of abandoning seasonal adjustment for its latest employment data.




PHOTO: Joe Hockey says he is unhappy with the volatility of ABS unemployment figures. (AAP: Alan Porritt)

RELATED STORY: [ABS abandons seasonal adjustment for latest jobs data](#)

The ABS stuff-up

NEWS 

LOCATION:
📍 Clayton, Vic [Change](#)



[Home](#) [Just In](#) [Australia](#) [World](#) [Business](#) [Sport](#) [Analysis & Opinion](#) [Fact Check](#) [Programs](#) [More](#)

BREAKING NEWS

Police arrest man in connection with stabbing death of 17-year-old Masa Vukotic in Mel

[Print](#) [Email](#) [Facebook](#) [Twitter](#) [More](#)

ABS abandons seasonal adjustment for latest jobs data

By business reporter [Michael Janda](#)

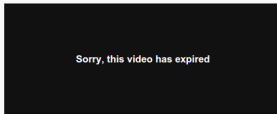
Updated 8 Oct 2014, 4:19pm

The Australian Bureau of Statistics is taking the rare step of abandoning seasonal adjustment for its latest employment data.

The ABS uses seasonal adjustment, based on historical experience, to account for the normal variation between hiring and firing patterns between different months.

However, after a winter where the seasonally adjusted unemployment rate swung wildly from 6.1 to 6.4 and back to 6.1 per cent, [the bureau released a statement](#) saying it will not adjust the original figure for September for seasonal factors.

It will also reset the seasonal adjustment for July and August to one, meaning that these months will also reflect the original figures.



VIDEO: [Westpac chief economist Bill Evans discusses the ABS jobs data changes](#) (ABC News)

RELATED STORY: [Doubt the record breaking jobs figures? So does the ABS](#)

RELATED STORY: [Jobs increase record sees unemployment slashed](#)

RELATED STORY: [Unemployment surges to 12-year high at 6.4 pc](#)

MAP: [Australia](#)



The ABS stuff-up

ABS jobs and unemployment figures – key questions answered by an expert

A professor of statistics at Monash University explains exactly what is seasonal adjustment, why it matters and what went wrong in the July and August figures



📷 School leavers come on to the jobs market at the same time, causing a seasonal fluctuation. Photograph: Brian Snyder/Reuters

The Australian Bureau of Statistics has [retracted its seasonally adjusted employment data for July and August](#), which recorded huge swings in the jobless rate. The ABS is also planning to review the methods it uses for seasonal adjustment to ensure its figures are as accurate as possible. Rob Hyndman, a professor of statistics at Monash University and member of the bureau's methodology advisory board, answers our questions:

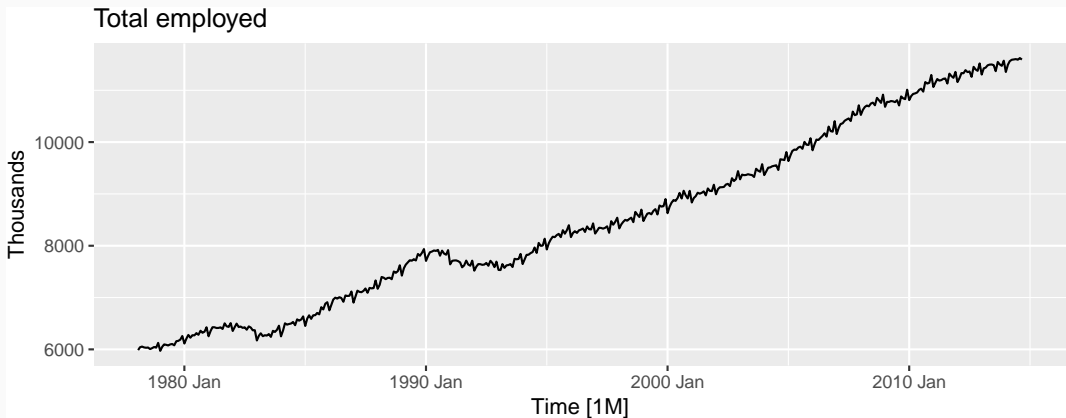
The ABS stuff-up

employed

```
## # A tsibble: 440 x 4 [1M]
##       Time Month Year Employed
##       <mt> <ord> <dbl>   <dbl>
## 1 1978 Feb Feb    1978    5986.
## 2 1978 Mar Mar    1978    6041.
## 3 1978 Apr Apr    1978    6054.
## 4 1978 May May    1978    6038.
## 5 1978 Jun Jun    1978    6031.
## 6 1978 Jul Jul    1978    6036.
## 7 1978 Aug Aug    1978    6005.
## 8 1978 Sep Sep    1978    6024.
## 9 1978 Oct Oct    1978    6046.
## 10 1978 Nov Nov    1978    6034.
## # ... with 430 more rows
```

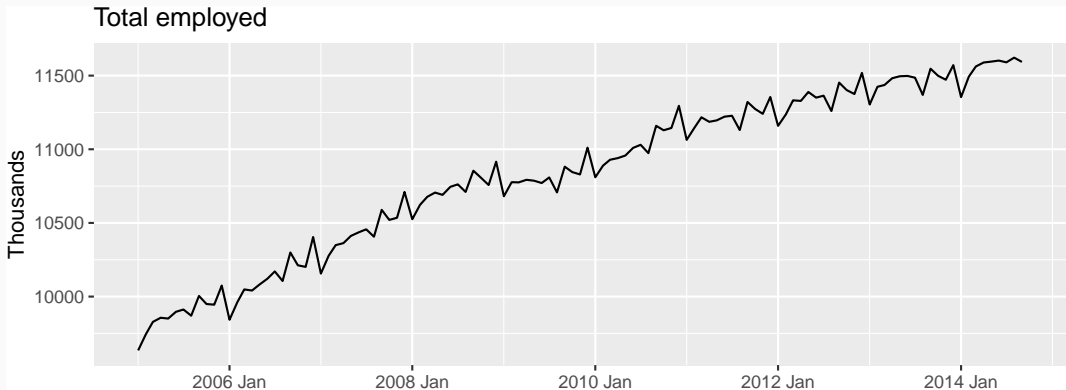
The ABS stuff-up

```
employed %>%  
  autoplot(Employed) +  
  labs(title = "Total employed", y = "Thousands")
```



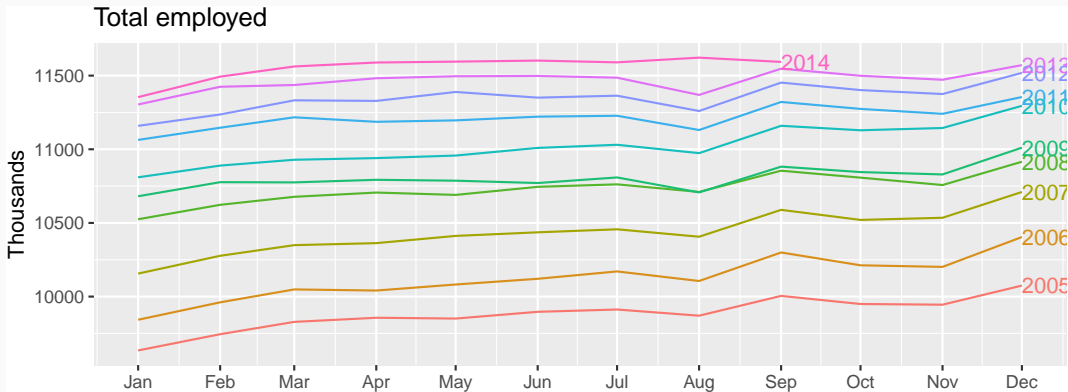
The ABS stuff-up

```
employed %>%  
  filter(Year >= 2005) %>%  
  autoplot(Employed) +  
  labs(title = "Total employed", y = "Thousands")
```



The ABS stuff-up

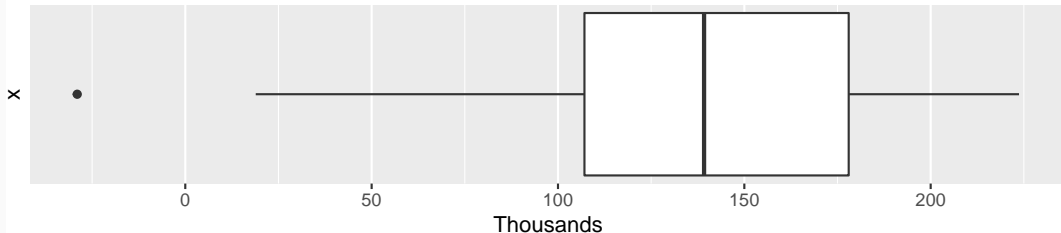
```
employed %>%  
  filter(Year >= 2005) %>%  
  gg_season(Employed, labels = "right") +  
  labs(title = "Total employed", y = "Thousands")
```



The ABS stuff-up

```
employed %>%  
  mutate(diff = difference(Employed)) %>%  
  filter(Month == "Sep") %>%  
  ggplot(aes(y = diff, x = 1)) +  
  geom_boxplot() + coord_flip() +  
  labs(title = "Sep - Aug: total employed", y = "Thousands") +  
  scale_x_continuous(breaks = NULL, labels = NULL)
```

Sep - Aug: total employed

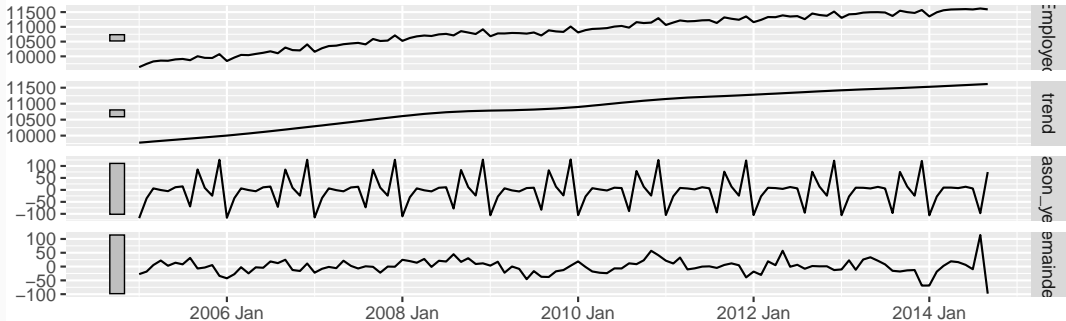


The ABS stuff-up

```
dcmp <- employed %>%  
  filter(Year >= 2005) %>%  
  model(stl = STL(Employed ~ season(window = 11), robust = TRUE))  
components(dcmp) %>% autoplot()
```

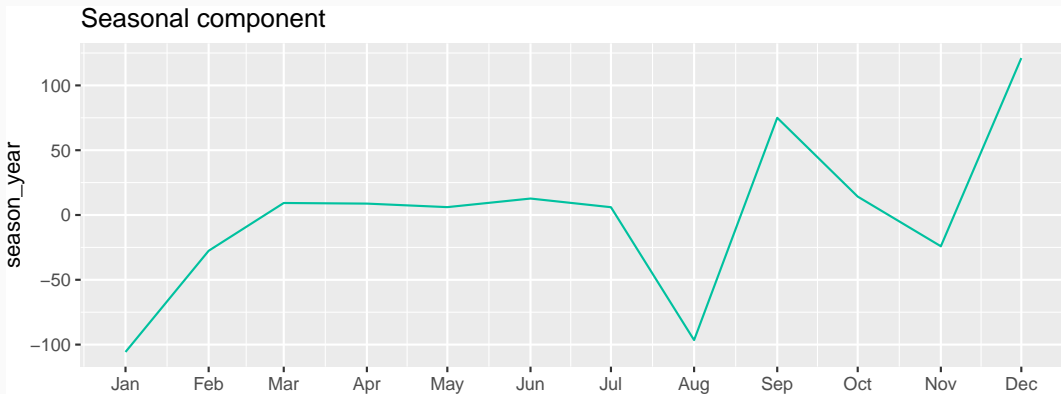
STL decomposition

Employed = trend + season_year + remainder



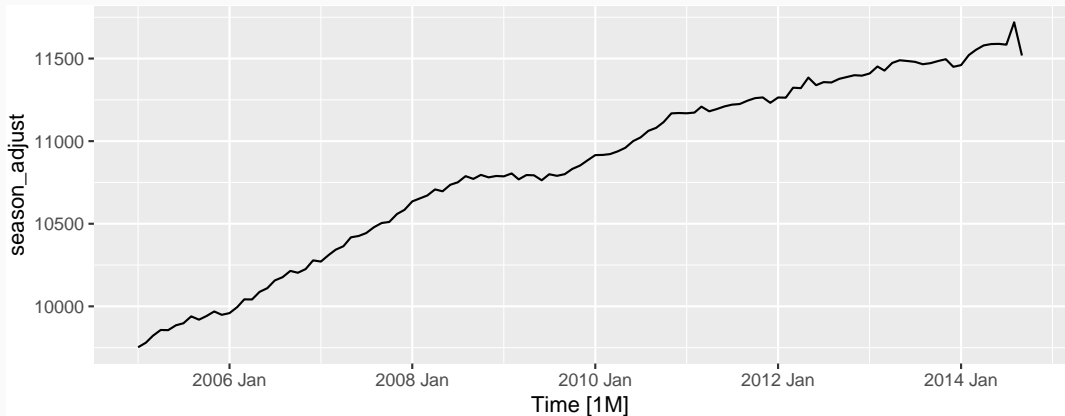
The ABS stuff-up

```
components(dcmp) %>%  
  filter(year(Time) == 2013) %>%  
  gg_season(season_year) +  
  labs(title = "Seasonal component") + guides(colour = "none")
```



The ABS stuff-up

```
components(dcmp) %>%  
  as_tsibble() %>%  
  autoplot(season_adjust)
```



The ABS stuff-up

- August 2014 employment numbers higher than expected.
- Supplementary survey usually conducted in August for employed people.
- Most likely, some employed people were claiming to be unemployed in August to avoid supplementary questions.
- Supplementary survey not run in 2014, so no motivation to lie about employment.
- In previous years, seasonal adjustment fixed the problem.
- The ABS has now adopted a new method to avoid the bias.