

# Time Series Analysis & Forecasting Using R

[bit.ly/fable2023](https://bit.ly/fable2023)

## 4. Seasonality and trends



# Outline

1 Time series decompositions

2 Lab Session 8

3 Multiple seasonality

4 The ABS stuff-up

# Outline

1 Time series decompositions

2 Lab Session 8

3 Multiple seasonality

4 The ABS stuff-up

# 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
- No trading day or calendar adjustments.
- Only additive.
- Take logs to get multiplicative decomposition.
- Use Box-Cox transformations to get other decompositions.

# US Retail Employment

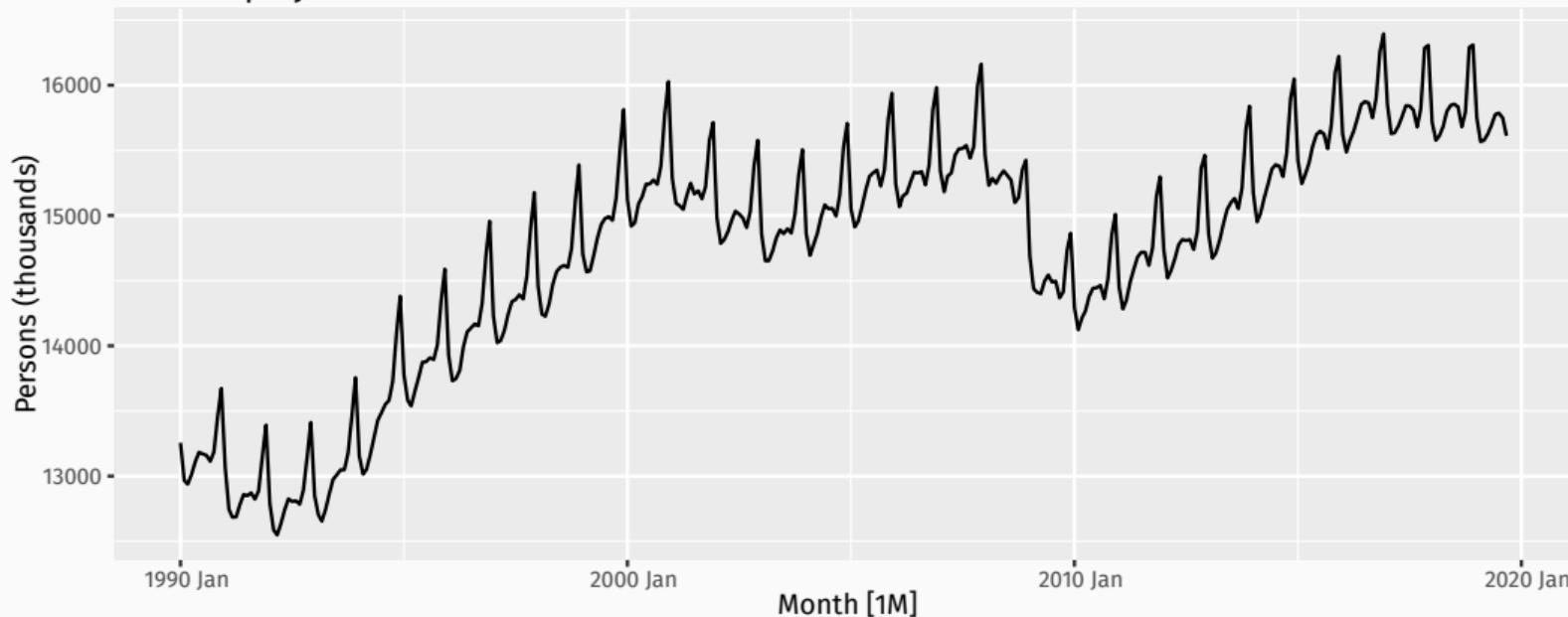
```
us_retail_employment <- us_employment |>
  filter(year(Month) >= 1990, Title == "Retail Trade") |>
  select(-Series_ID)
us_retail_employment
```

```
# 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 13113.
10 1990 Oct Retail Trade 13185
```

# US Retail Employment

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

Total employment in US retail



# US Retail Employment

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

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

# US Retail Employment

## components (dcmp)

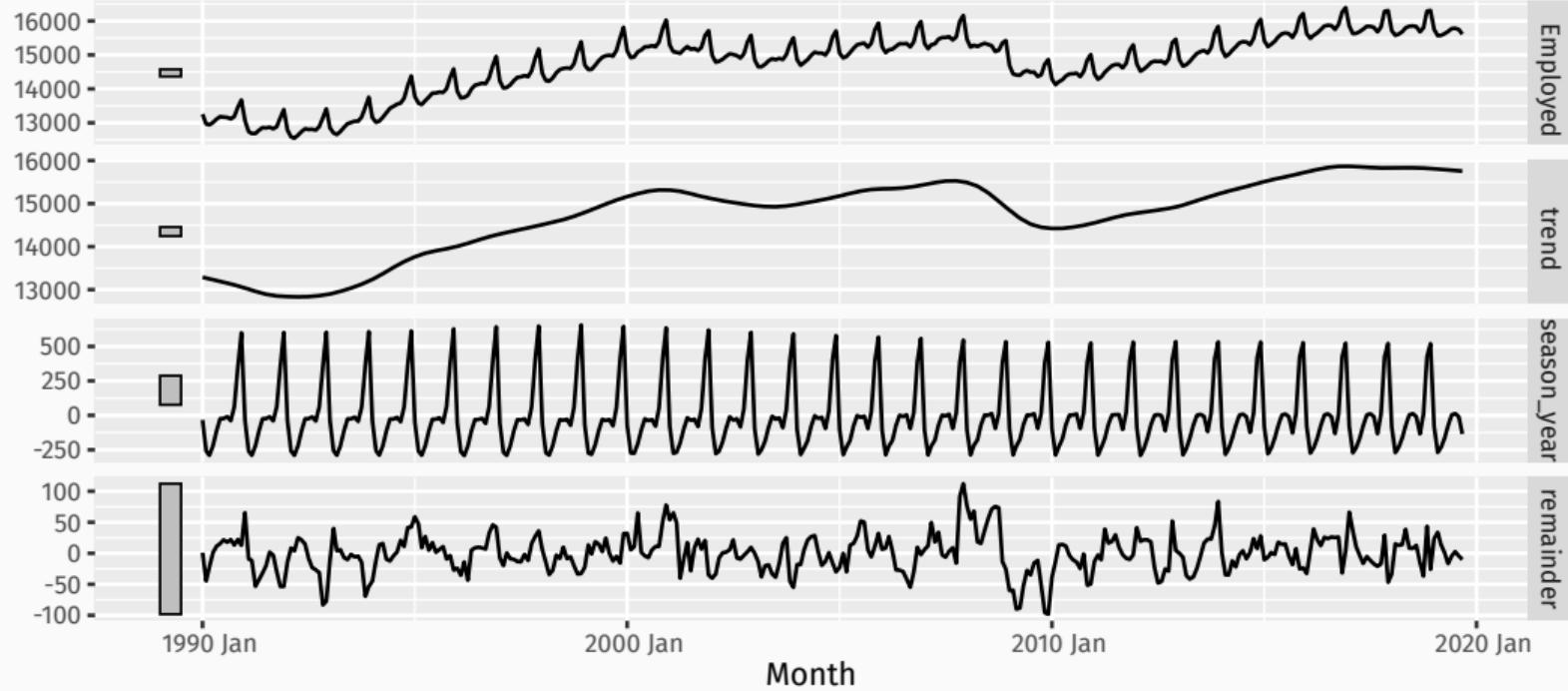
```
# A dable: 357 x 7 [1M]
# Key:     .model [1]
# :       Employed = trend + season_year + remainder
#             .model   Month Employed   trend season_year remainder season_adjust
#             <chr>    <mth>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 stl      1990 Jan   13256. 13288. -33.0    0.836 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.
8 stl      1990 Aug   13160. 13151. -9.52   17.8 13169.
9 stl      1990 Sep   13113. 13131. -39.5   22.0 13153.
10 stl     1990 Oct   13185. 13110.  61.6   13.2 13124.
# ... with 347 more rows
```

# US Retail Employment

components(dcmp) |> autoplot()

## STL decomposition

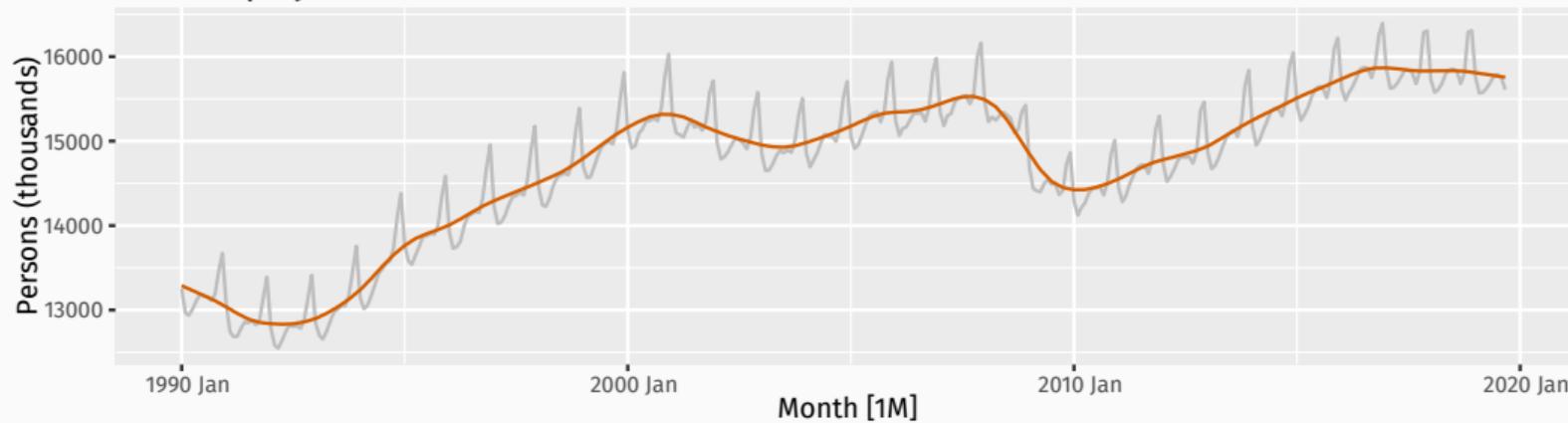
Employed = trend + season\_year + remainder



# US Retail Employment

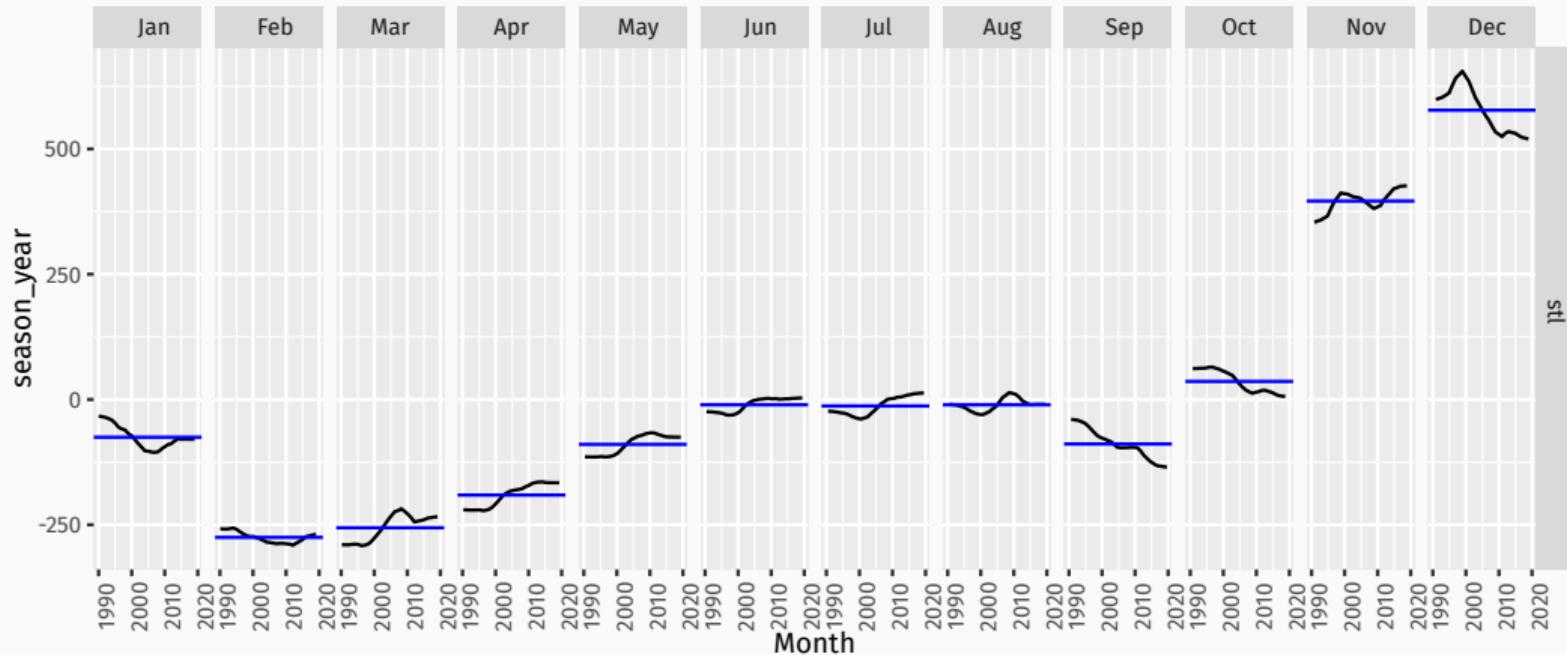
```
us_retail_employment |>  
  autoplot(Employed, color = "gray") +  
  autolayer(components(dcmp), trend, color = "#D55E00") +  
  labs(y = "Persons (thousands)", title = "Total employment in US retail")
```

Total employment in US retail



# 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

$$y_t - S_t = T_t + R_t$$

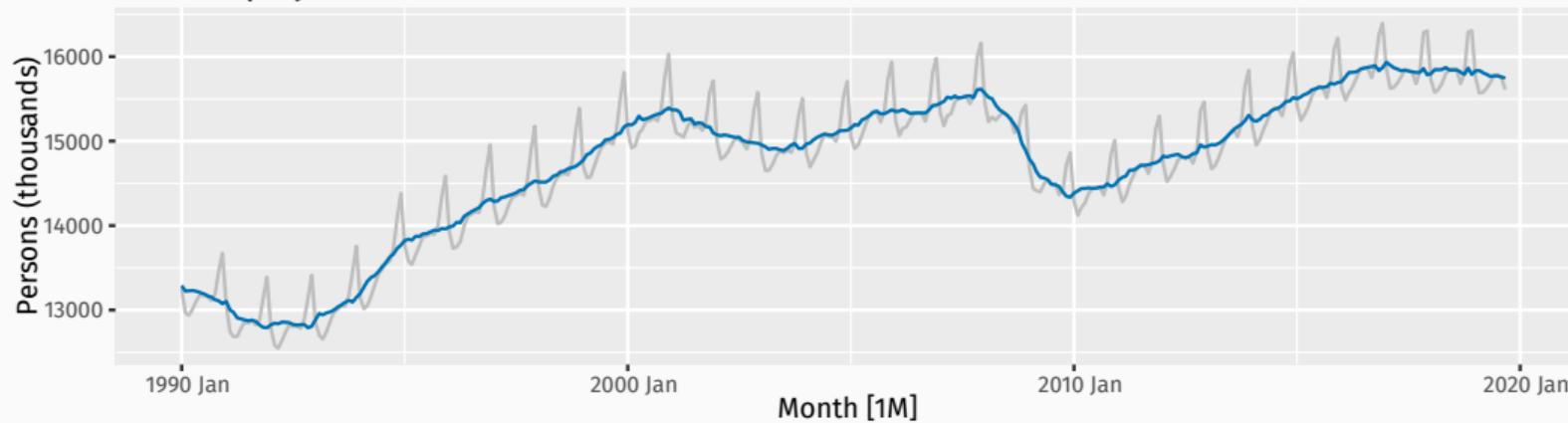
- Multiplicative decomposition: seasonally adjusted data given by

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

# 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")
```

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.

# STL decomposition

# STL decomposition

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

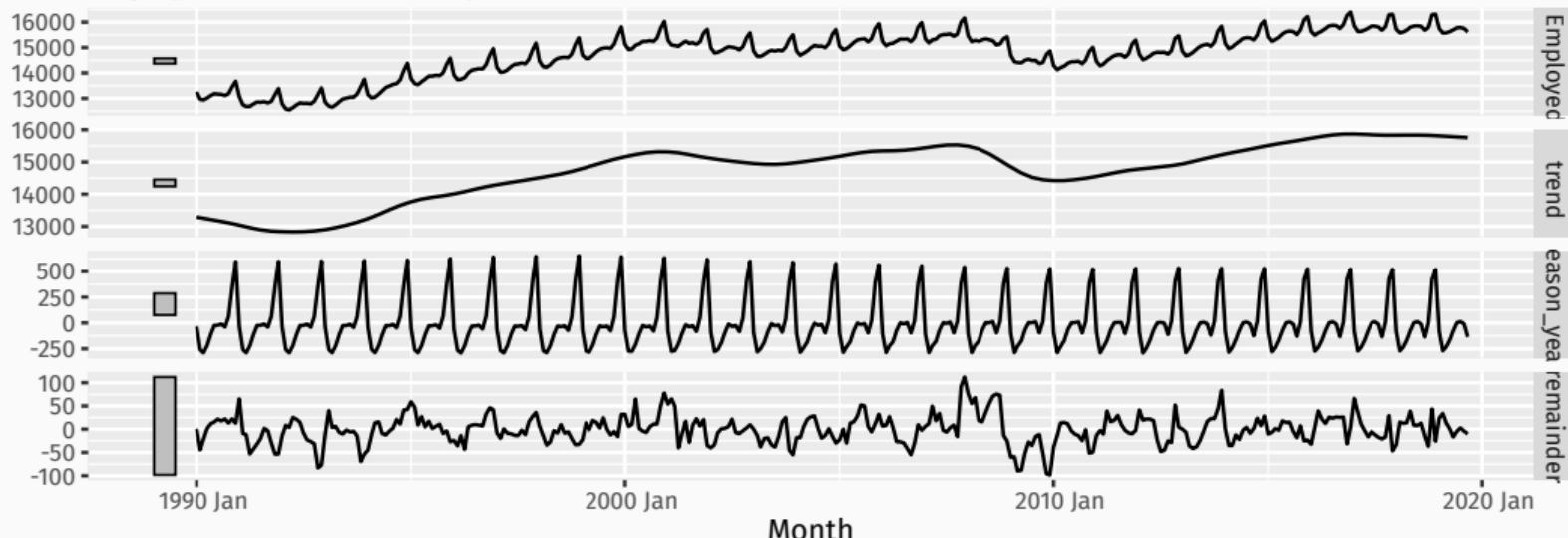
- `trend(window = ?)` controls wiggliness of trend component.
- `season(window = ?)` controls variation on seasonal component.
- `season(window = 'periodic')` is equivalent to an infinite window.

# STL decomposition

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

STL decomposition

Employed = trend + season\_year + remainder



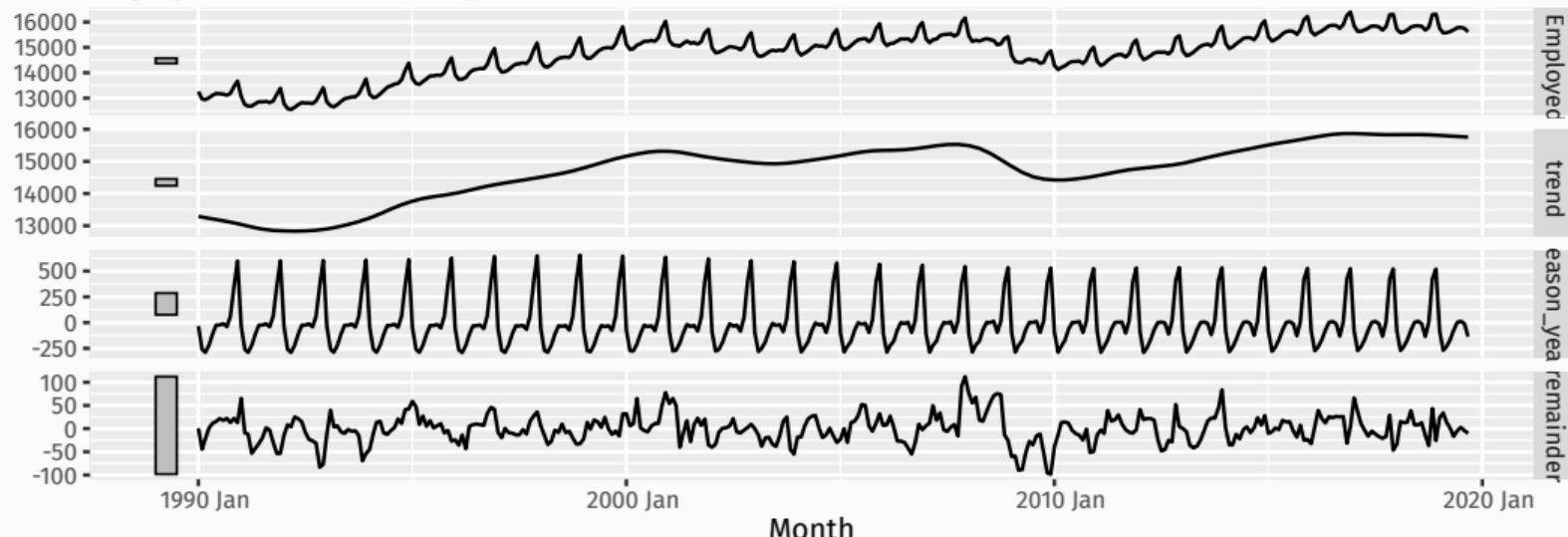
# STL decomposition

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

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

STL decomposition

Employed = trend + season\_year + remainder



# STL decomposition

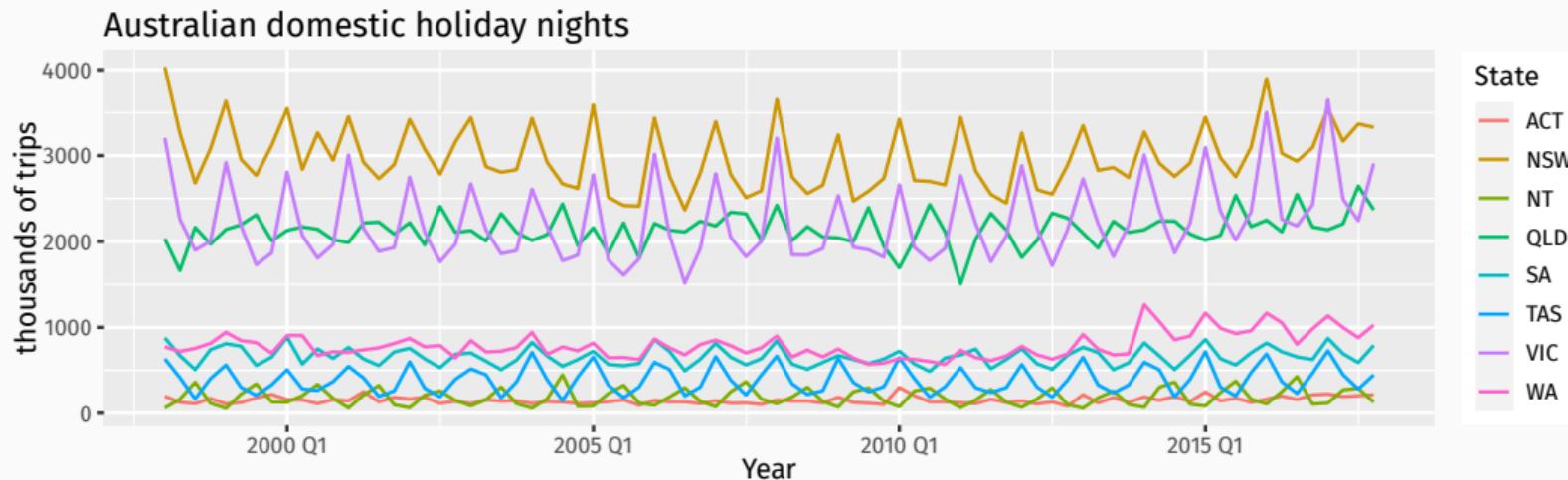
- Algorithm that updates trend and seasonal components iteratively.
- Starts with  $\hat{T}_t = 0$
- Uses a mixture of loess and moving averages to successively refine the trend and seasonal estimates.
- trend window controls loess bandwidth on deasonalised values.
- season window controls loess bandwidth on detrended subseries.
- Robustness weights based on remainder.
- Default season: window = 13
- Default trend:

window =

```
nextodd(ceiling((1.5*period)/(1-(1.5/s.window))))
```

# Australian holidays

```
holidays |> autoplot(Trips) +  
  labs(y = "thousands of trips", x = "Year",  
        title = "Australian domestic holiday nights")
```



# Australian holidays

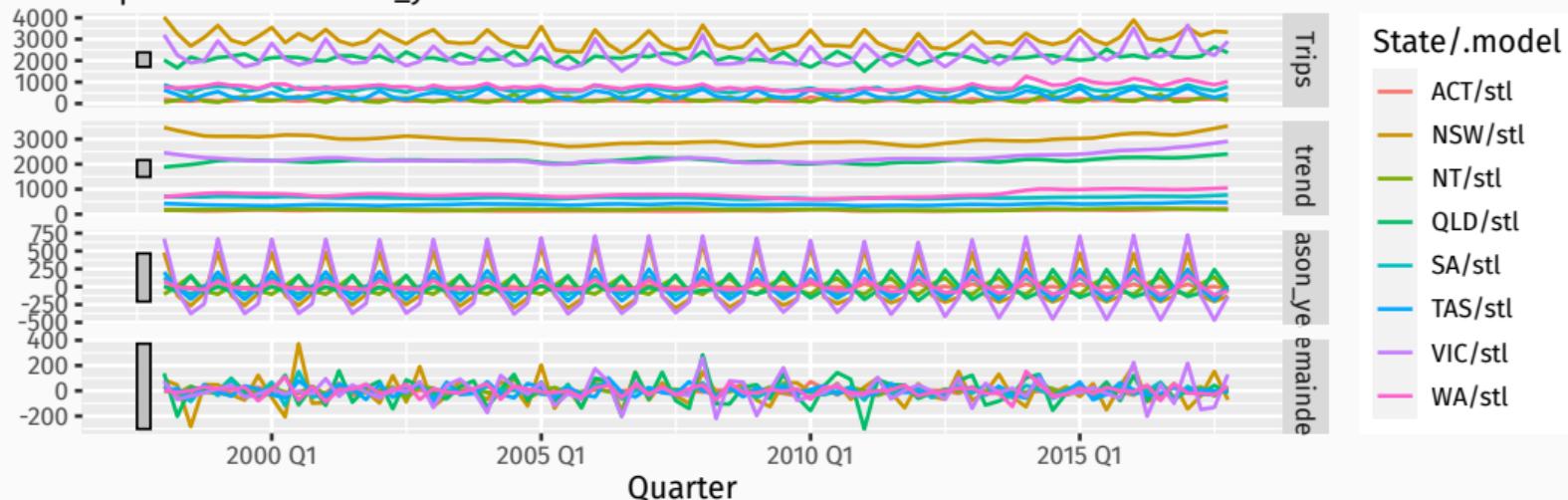
```
library(fpp2)
library(tsibble)
library(dplyr)
library(ggplot2)
library(tidyverse)

# Load the data
airline_trips <- read_csv("airline_trips.csv") %>%  
  # Create a tsibble  
  as_tsibble()  
  
# Set the time variable  
airline_trips <- airline_trips %>%  
  mutate(date = as.Date(date))
```

```
## #> # A tsibble: 1,000 x 2  
## #> #> #> date     trips  
## #> #> #>   <date>    <dbl>  
## #> #> #> 1 1999-01-01  2000  
## #> #> #> 2 1999-04-01  2000  
## #> #> #> 3 1999-07-01  2000  
## #> #> #> 4 1999-10-01  2000  
## #> #> #> 5 2000-01-01  2000  
## #> #> #> 6 2000-04-01  2000  
## #> #> #> 7 2000-07-01  2000  
## #> #> #> 8 2000-10-01  2000  
## #> #> #> 9 2001-01-01  2000  
## #> #> #> 10 2001-04-01  2000  
## #> #> #> 11 2001-07-01  2000  
## #> #> #> 12 2001-10-01  2000  
## #> #> #> 13 2002-01-01  2000  
## #> #> #> 14 2002-04-01  2000  
## #> #> #> 15 2002-07-01  2000  
## #> #> #> 16 2002-10-01  2000  
## #> #> #> 17 2003-01-01  2000  
## #> #> #> 18 2003-04-01  2000  
## #> #> #> 19 2003-07-01  2000  
## #> #> #> 20 2003-10-01  2000  
## #> #> #> 21 2004-01-01  2000  
## #> #> #> 22 2004-04-01  2000  
## #> #> #> 23 2004-07-01  2000  
## #> #> #> 24 2004-10-01  2000  
## #> #> #> 25 2005-01-01  2000  
## #> #> #> 26 2005-04-01  2000  
## #> #> #> 27 2005-07-01  2000  
## #> #> #> 28 2005-10-01  2000  
## #> #> #> 29 2006-01-01  2000  
## #> #> #> 30 2006-04-01  2000  
## #> #> #> 31 2006-07-01  2000  
## #> #> #> 32 2006-10-01  2000  
## #> #> #> 33 2007-01-01  2000  
## #> #> #> 34 2007-04-01  2000  
## #> #> #> 35 2007-07-01  2000  
## #> #> #> 36 2007-10-01  2000  
## #> #> #> 37 2008-01-01  2000  
## #> #> #> 38 2008-04-01  2000  
## #> #> #> 39 2008-07-01  2000  
## #> #> #> 40 2008-10-01  2000  
## #> #> #> 41 2009-01-01  2000  
## #> #> #> 42 2009-04-01  2000  
## #> #> #> 43 2009-07-01  2000  
## #> #> #> 44 2009-10-01  2000  
## #> #> #> 45 2010-01-01  2000  
## #> #> #> 46 2010-04-01  2000  
## #> #> #> 47 2010-07-01  2000  
## #> #> #> 48 2010-10-01  2000  
## #> #> #> 49 2011-01-01  2000  
## #> #> #> 50 2011-04-01  2000  
## #> #> #> 51 2011-07-01  2000  
## #> #> #> 52 2011-10-01  2000  
## #> #> #> 53 2012-01-01  2000  
## #> #> #> 54 2012-04-01  2000  
## #> #> #> 55 2012-07-01  2000  
## #> #> #> 56 2012-10-01  2000  
## #> #> #> 57 2013-01-01  2000  
## #> #> #> 58 2013-04-01  2000  
## #> #> #> 59 2013-07-01  2000  
## #> #> #> 60 2013-10-01  2000  
## #> #> #> 61 2014-01-01  2000  
## #> #> #> 62 2014-04-01  2000  
## #> #> #> 63 2014-07-01  2000  
## #> #> #> 64 2014-10-01  2000  
## #> #> #> 65 2015-01-01  2000  
## #> #> #> 66 2015-04-01  2000  
## #> #> #> 67 2015-07-01  2000  
## #> #> #> 68 2015-10-01  2000  
## #> #> #> 69 2016-01-01  2000  
## #> #> #> 70 2016-04-01  2000  
## #> #> #> 71 2016-07-01  2000  
## #> #> #> 72 2016-10-01  2000  
## #> #> #> 73 2017-01-01  2000  
## #> #> #> 74 2017-04-01  2000  
## #> #> #> 75 2017-07-01  2000  
## #> #> #> 76 2017-10-01  2000  
## #> #> #> 77 2018-01-01  2000  
## #> #> #> 78 2018-04-01  2000  
## #> #> #> 79 2018-07-01  2000  
## #> #> #> 80 2018-10-01  2000  
## #> #> #> 81 2019-01-01  2000  
## #> #> #> 82 2019-04-01  2000  
## #> #> #> 83 2019-07-01  2000  
## #> #> #> 84 2019-10-01  2000  
## #> #> #> 85 2020-01-01  2000  
## #> #> #> 86 2020-04-01  2000  
## #> #> #> 87 2020-07-01  2000  
## #> #> #> 88 2020-10-01  2000  
## #> #> #> 89 2021-01-01  2000  
## #> #> #> 90 2021-04-01  2000  
## #> #> #> 91 2021-07-01  2000  
## #> #> #> 92 2021-10-01  2000  
## #> #> #> 93 2022-01-01  2000  
## #> #> #> 94 2022-04-01  2000  
## #> #> #> 95 2022-07-01  2000  
## #> #> #> 96 2022-10-01  2000  
## #> #> #> 97 2023-01-01  2000  
## #> #> #> 98 2023-04-01  2000  
## #> #> #> 99 2023-07-01  2000  
## #> #> #> 100 2023-10-01  2000
```

## STL decomposition

Trips = trend + season\_year + remainder



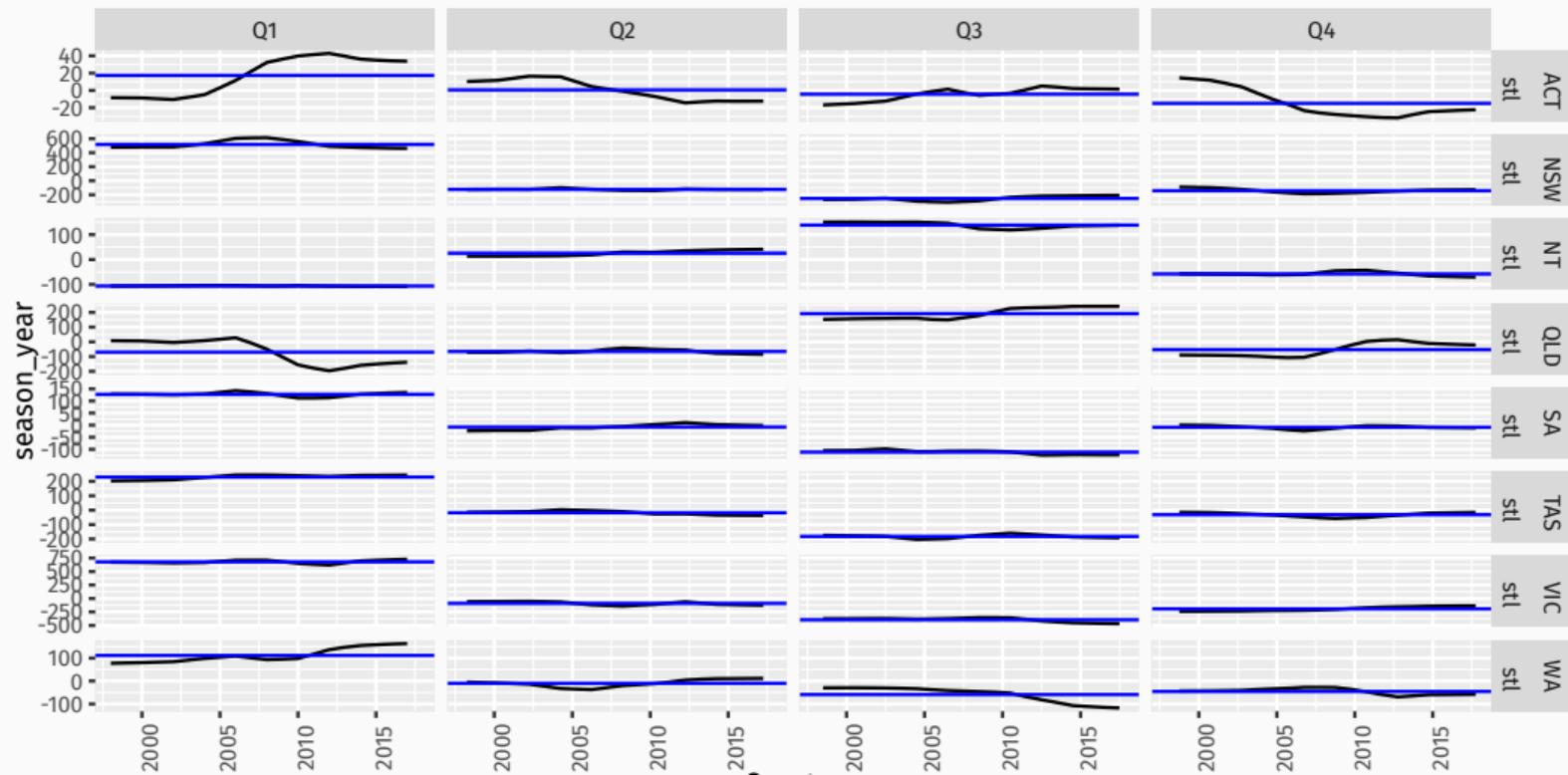
# Holidays decomposition

```
dcmp <- holidays |>
  model(stl = STL(Trips)) |>
  components()
dcmp
```

```
# A dable: 640 x 8 [1Q]
# Key:      State, .model [8]
# :          Trips = trend + season_year + remainder
#             State .model Quarter Trips trend season_year remainder season_adjust
#             <chr> <chr>    <qtr> <dbl> <dbl>      <dbl>      <dbl>      <dbl>
# 1 ACT   stl     1998 Q1  196.  172.     -8.48     32.6     205.
# 2 ACT   stl     1998 Q2  127.  157.      10.3     -40.6     116.
# 3 ACT   stl     1998 Q3  111.  142.     -16.8     -14.5     128.
# 4 ACT   stl     1998 Q4  170.  130.      14.6      25.6     156.
# 5 ACT   stl     1999 Q1  108.  135.     -8.63     -18.3     116.
# 6 ACT   stl     1999 Q2  125.  148.      11.0     -34.6     114.
# 7 ACT   stl     1999 Q3  178.  166.     -16.0      28.3     194.
# 8 ACT   stl     1999 Q4  218.  177.      13.2      27.5     204.
# 9 ACT   stl     2000 Q1  158.  169.     -8.75     -1.96     167.
```

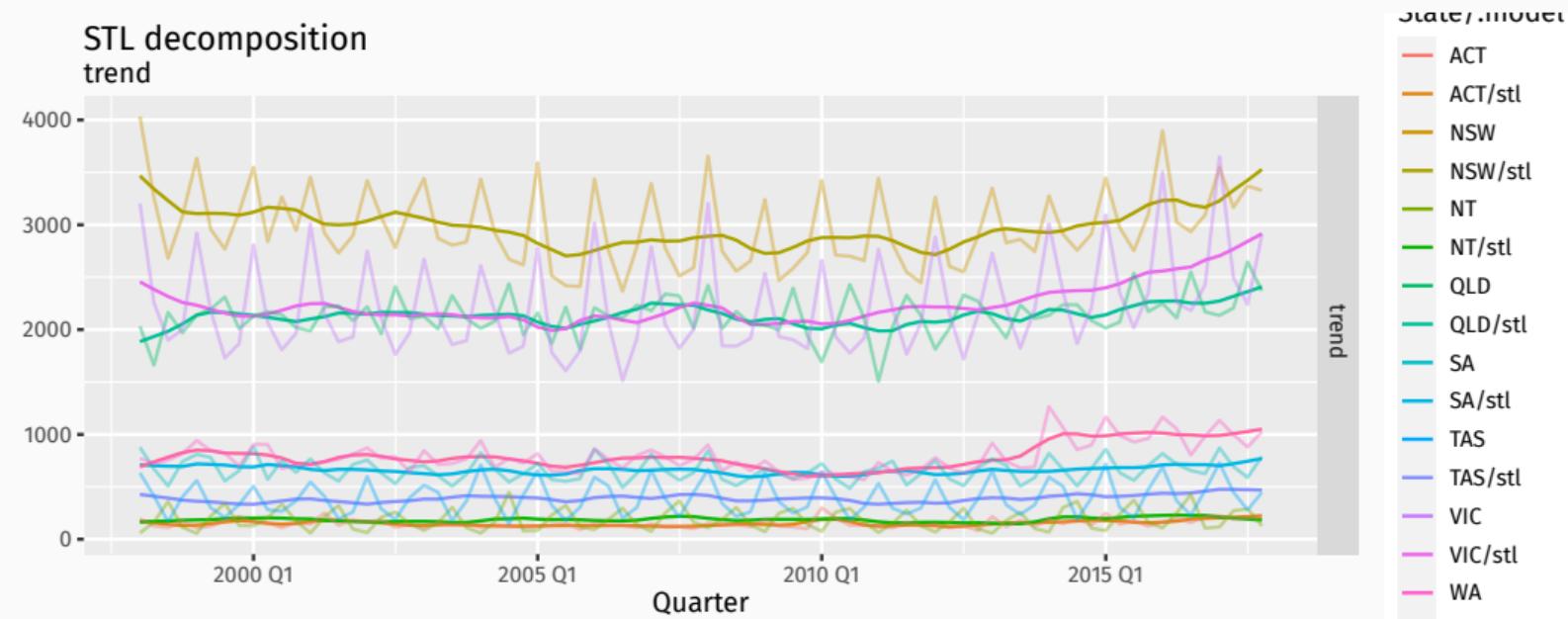
# Holidays decomposition

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 8

3 Multiple seasonality

4 The ABS stuff-up

# Lab Session 8

- 1 Produce the following decomposition

```
canadian_gas |>  
  model(STL(Volume ~ season(window=7) + trend(window=11))) |>  
  components() |>  
  autoplot()
```

- 2 What happens as you change the values of the two window arguments?
- 3 How does the seasonal shape change over time? [Hint: Try plotting the seasonal component using gg\_season.]
- 4 Can you produce a plausible seasonally adjusted series? [Hint: season\_adjust is one of the variables returned by STL.]

# Outline

1 Time series decompositions

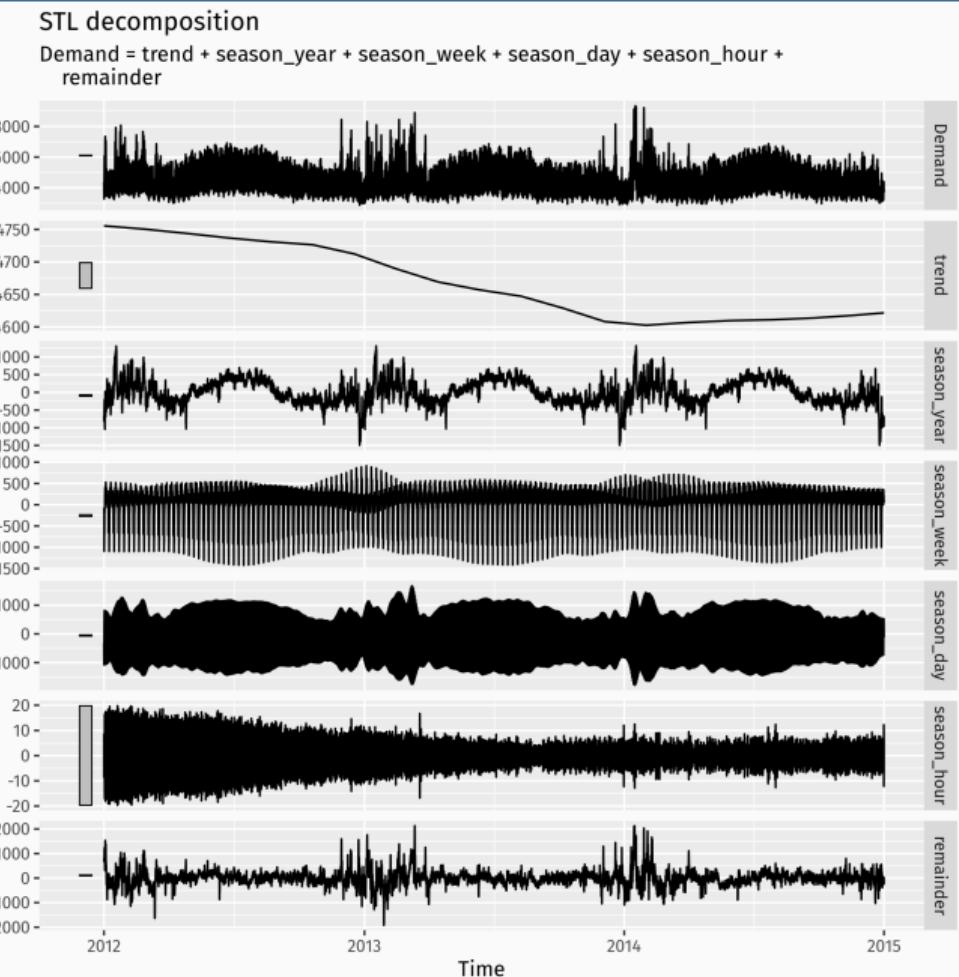
2 Lab Session 8

3 Multiple seasonality

4 The ABS stuff-up

# Multiple seasonality

```
vic_elec |>  
model(STL(Demand)) |>  
components() |>  
autoplot()
```



# Outline

1 Time series decompositions

2 Lab Session 8

3 Multiple seasonality

4 The ABS stuff-up

# The ABS stuff-up

NEWS 

LOCATION:  
Clayton, Vic [Change](#)

[Home](#) 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.

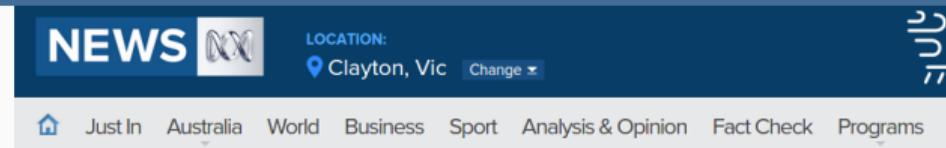
The ABS has now taken the rare step of abandoning seasonal adjustment for its latest employment data.



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



The header of the ABC News website. It features the 'NEWS' logo with a stylized 'N' icon. Below it is a location dropdown set to 'Clayton, Vic' with a 'Change' button. On the right is a decorative graphic of three vertical bars. A navigation bar below includes links for Home, Just In, Australia, World, Business, Sport, Analysis & Opinion, Fact Check, Programs, and a partially visible 'M'.

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

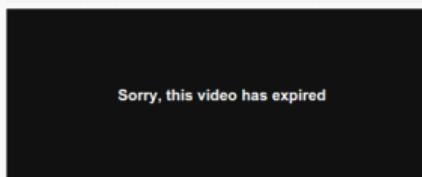
Updated 8 Oct 2014, 4:19pm

**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.



Sorry, this video has expired

**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

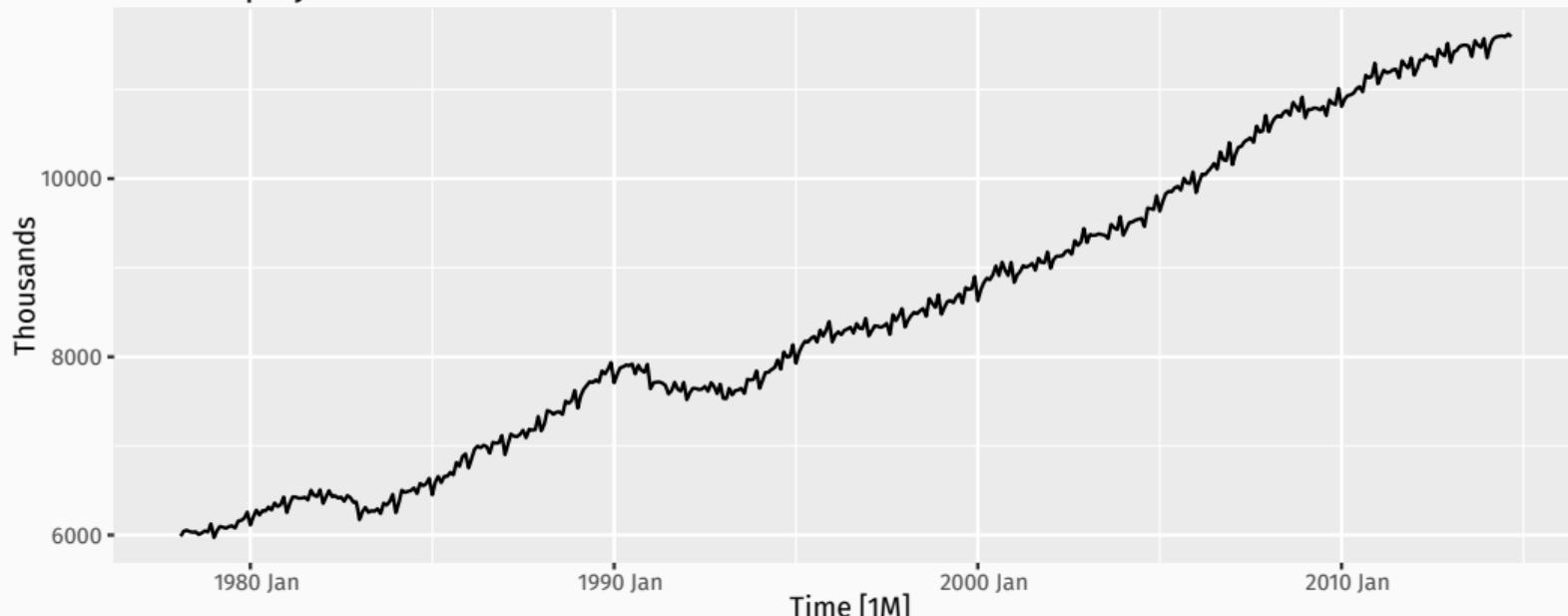
employed

```
# A tsibble: 440 x 4 [1M]
# ... with variables:
#   - Time: Date
#   - Month: Month
#   - Year: Year
#   - Employed: Double
Time Month Year Employed
<mth> <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.
# i 430 more rows
```

# The ABS stuff-up

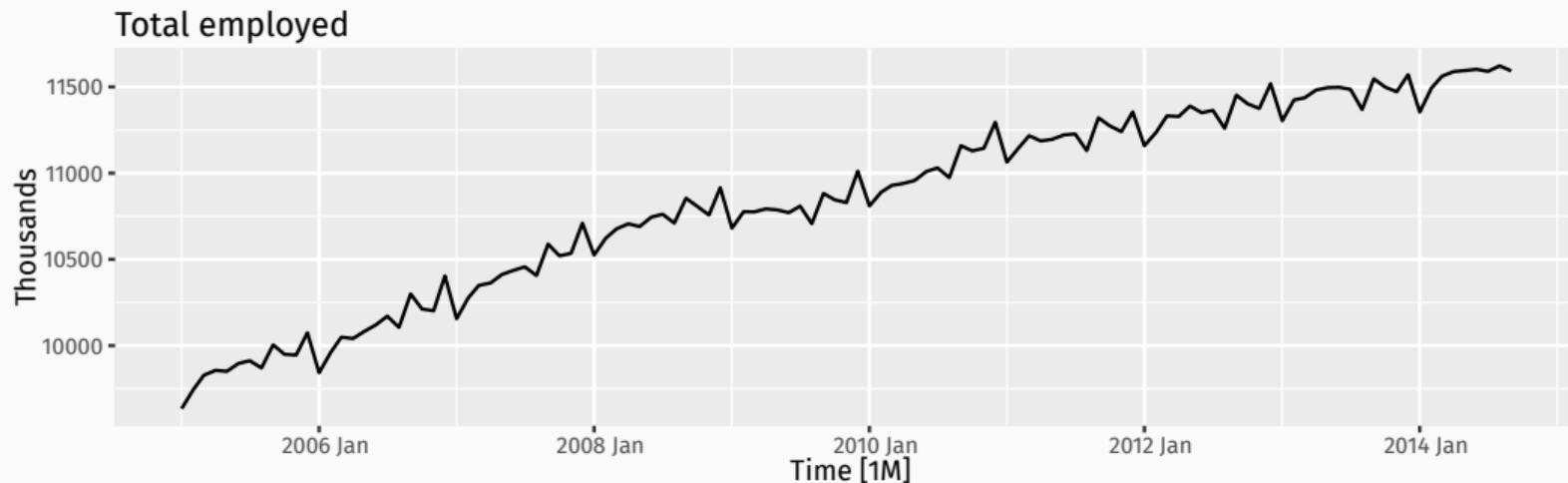
```
employed |>  
  autoplot(Employed) +  
  labs(title = "Total employed", y = "Thousands")
```

Total employed



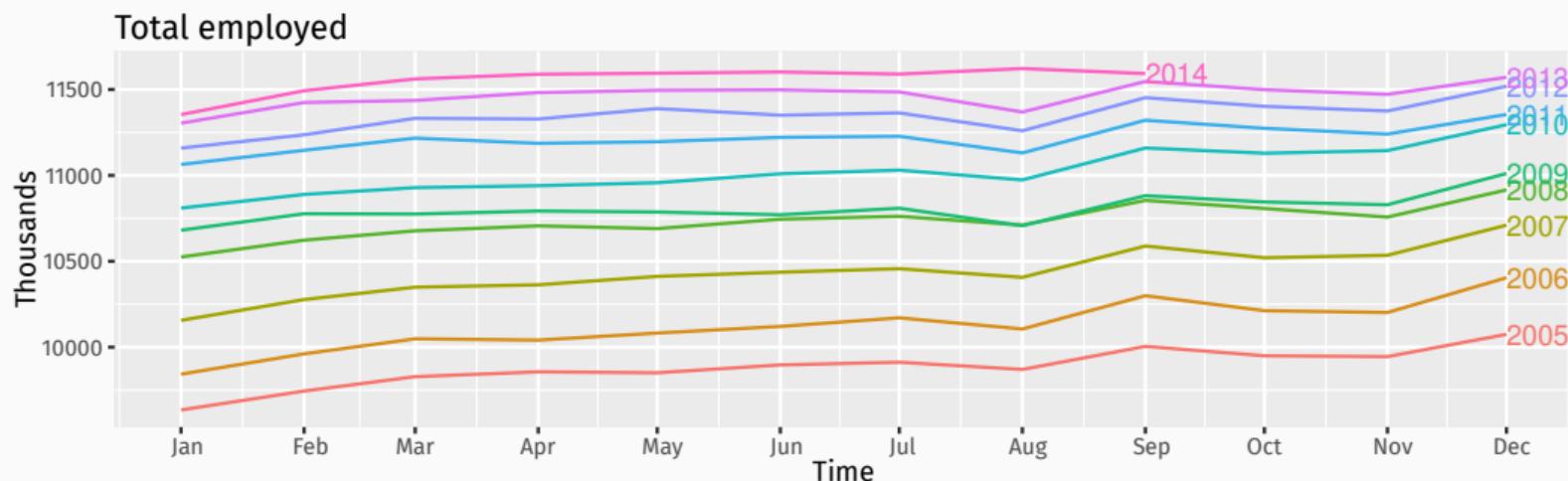
# The ABS stuff-up

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



# The ABS stuff-up

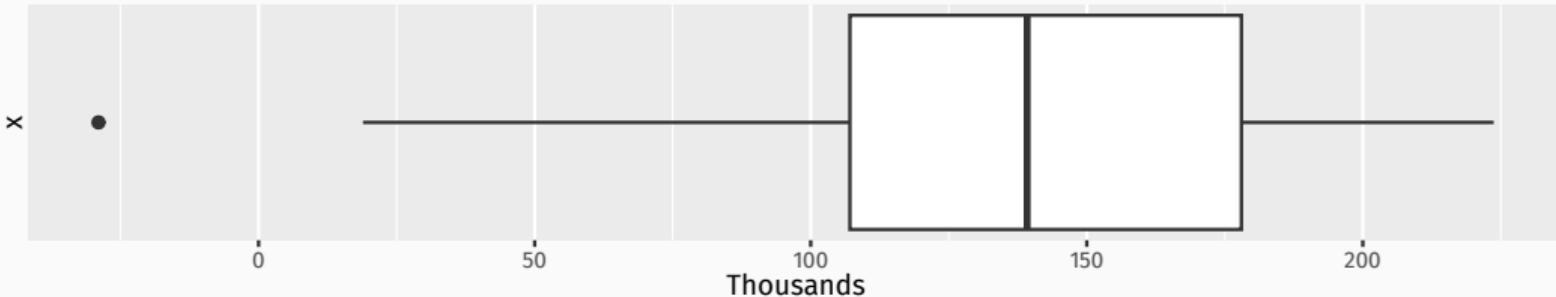
```
employed |>  
filter(Year >= 2005) |>  
gg_season(Employed, labels = "right") +  
labs(title = "Total employed", y = "Thousands")
```



# The ABS stuff-up

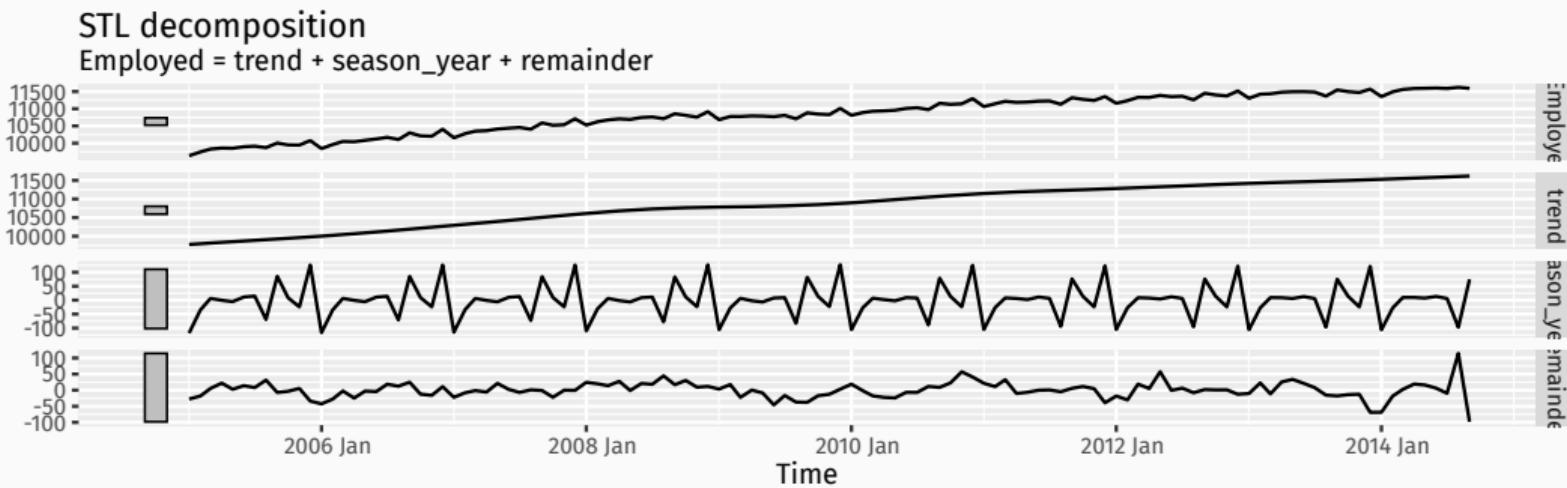
```
employed |>  
  mutate(diff = difference(Employed)) |>  
  filter(Month == "Sep") |>  
  ggplot(aes(y = diff, x = 1)) +  
  geom_boxplot() +  
  coord_flip() +  
  labs(title = "Sep - Aug: total employed", y = "Thousands") +  
  scale_x_continuous(breaks = NULL, labels = NULL)
```

Sep - Aug: total employed



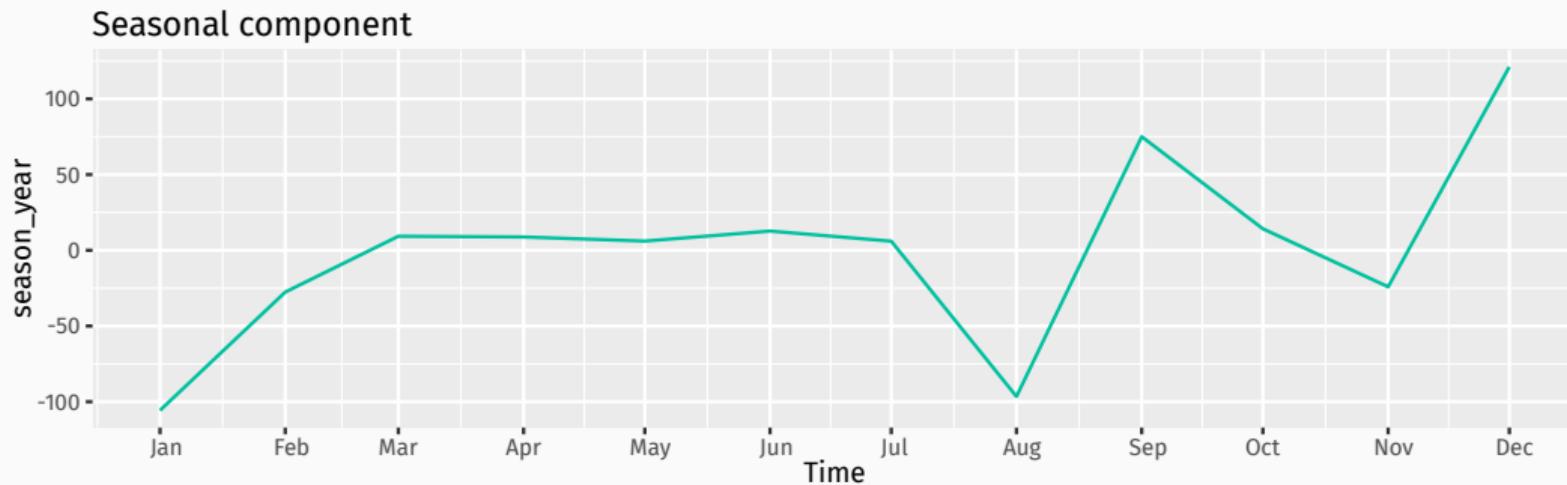
# The ABS stuff-up

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



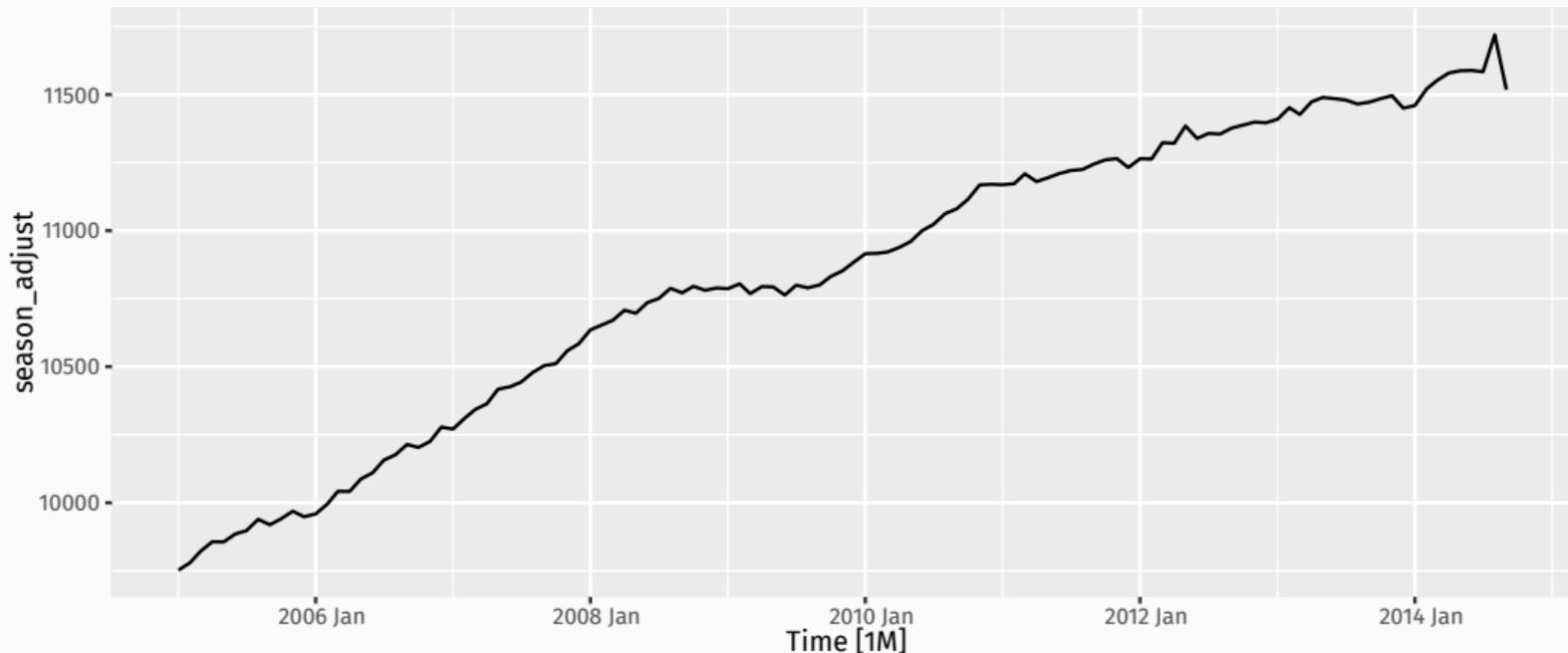
# The ABS stuff-up

```
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.