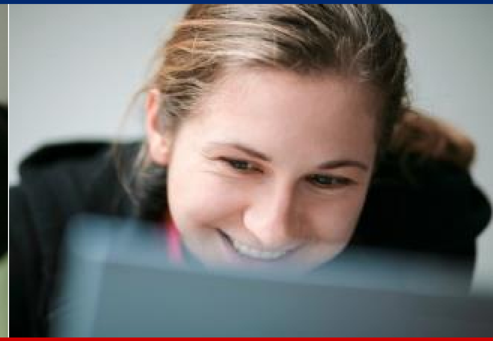






Applied Time Series Analysis
Decomposition and Smoothing

Prof. Dr. Stephan Trahasch

Outline

- Decomposition
- Smoothing
- Trend and seasonal methods
- Summary

Time Series Patterns

- **Trend**

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

- **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).

- **Cyclic**

  pattern exists when data exhibit rises and falls that are (duration usually of at least 2 years).

Time series decomposition

$$y_t = \mathrm{f}(S_t, T_t, R_t)$$

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

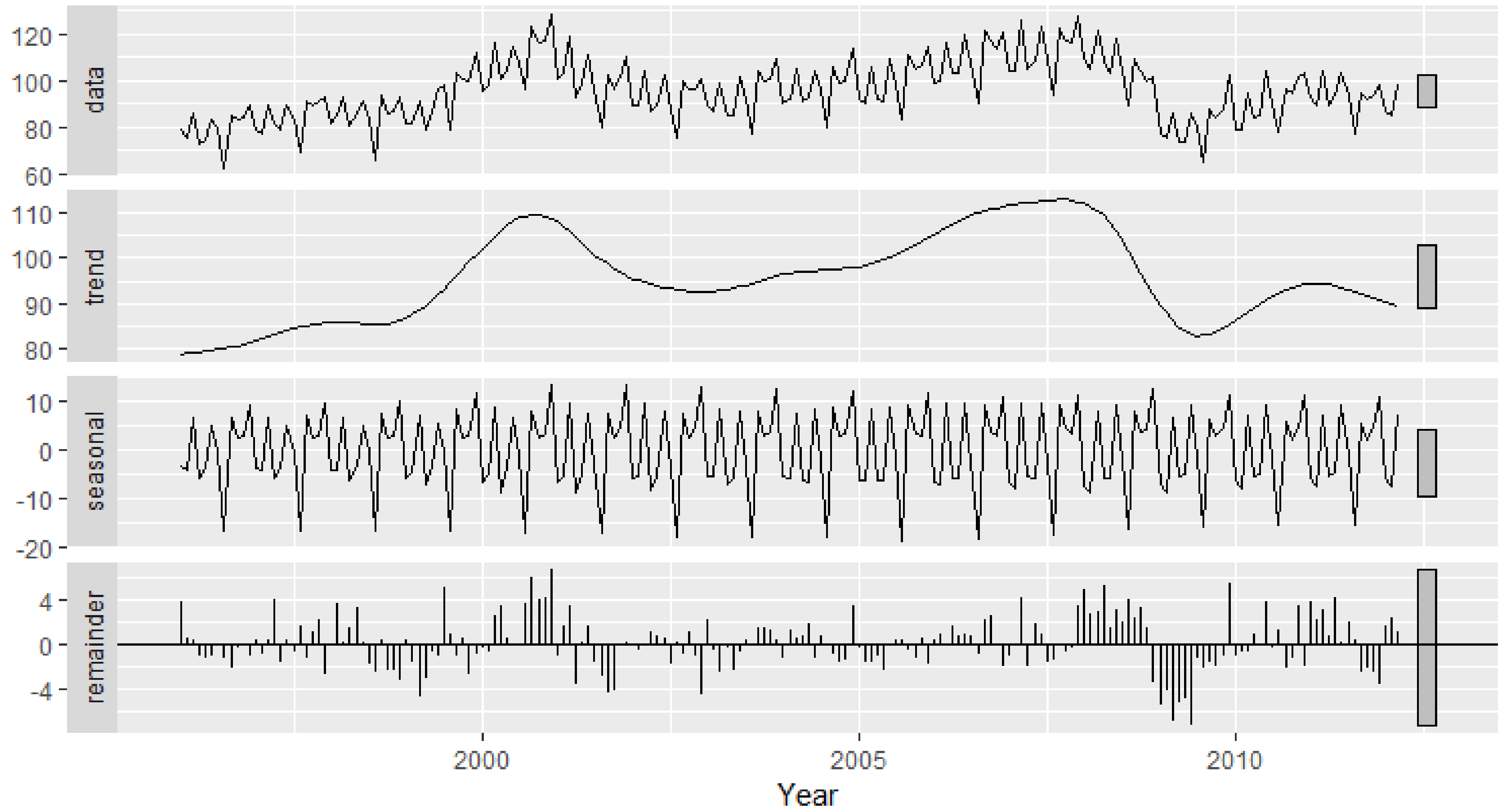
Time series decomposition

- 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:

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

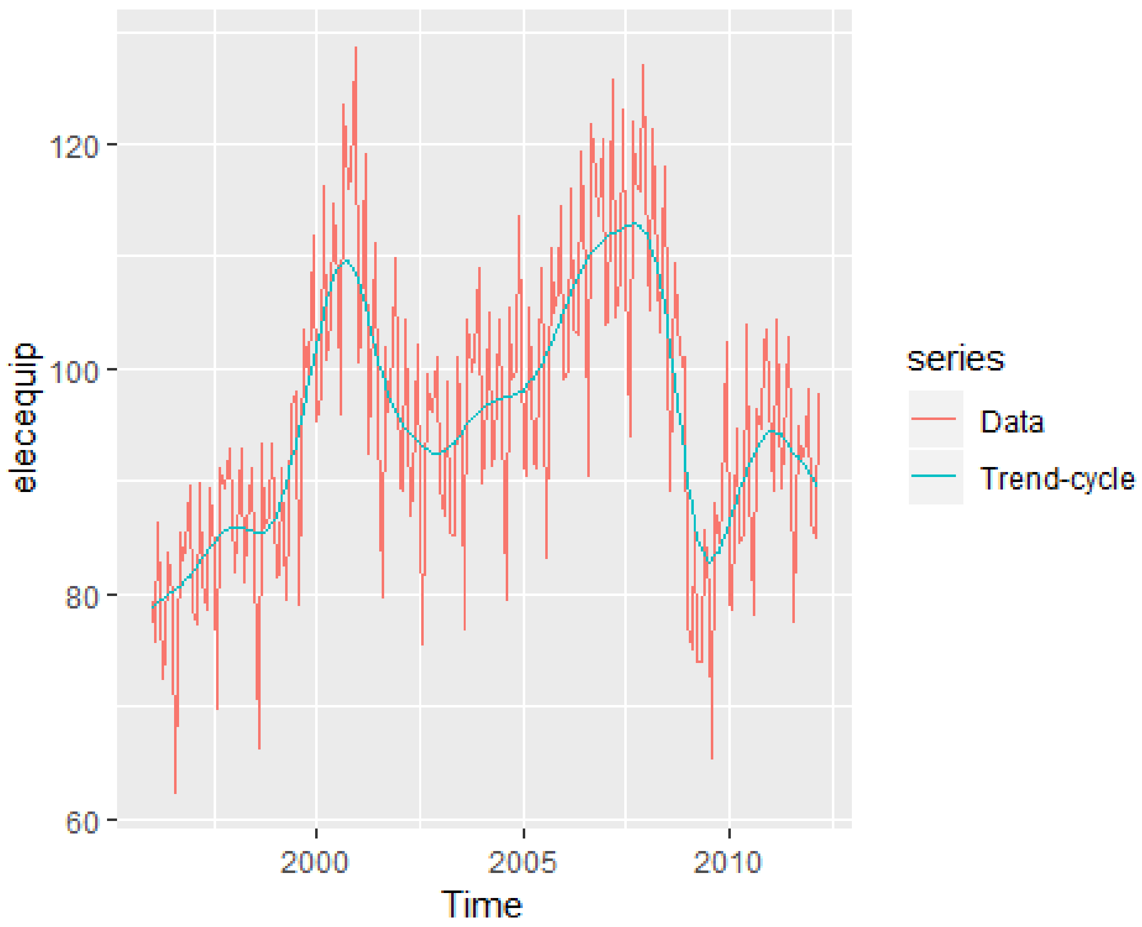
Euro electrical equipment

```
fit <- stl(elecequip, s.window=7)
autoplot(fit) + xlab("Year")
```



Euro electrical equipment

```
autoplot(elecequip, series="Data") +
  autolayer(trendcycle(fit),
series="Trend-cycle")
```



Helper functions

- seasonal() extracts the seasonal component
- trendcycle() extracts the trend-cycle component
- remainder() extracts the remainder component.
- seasadj() returns the seasonally adjusted series.

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

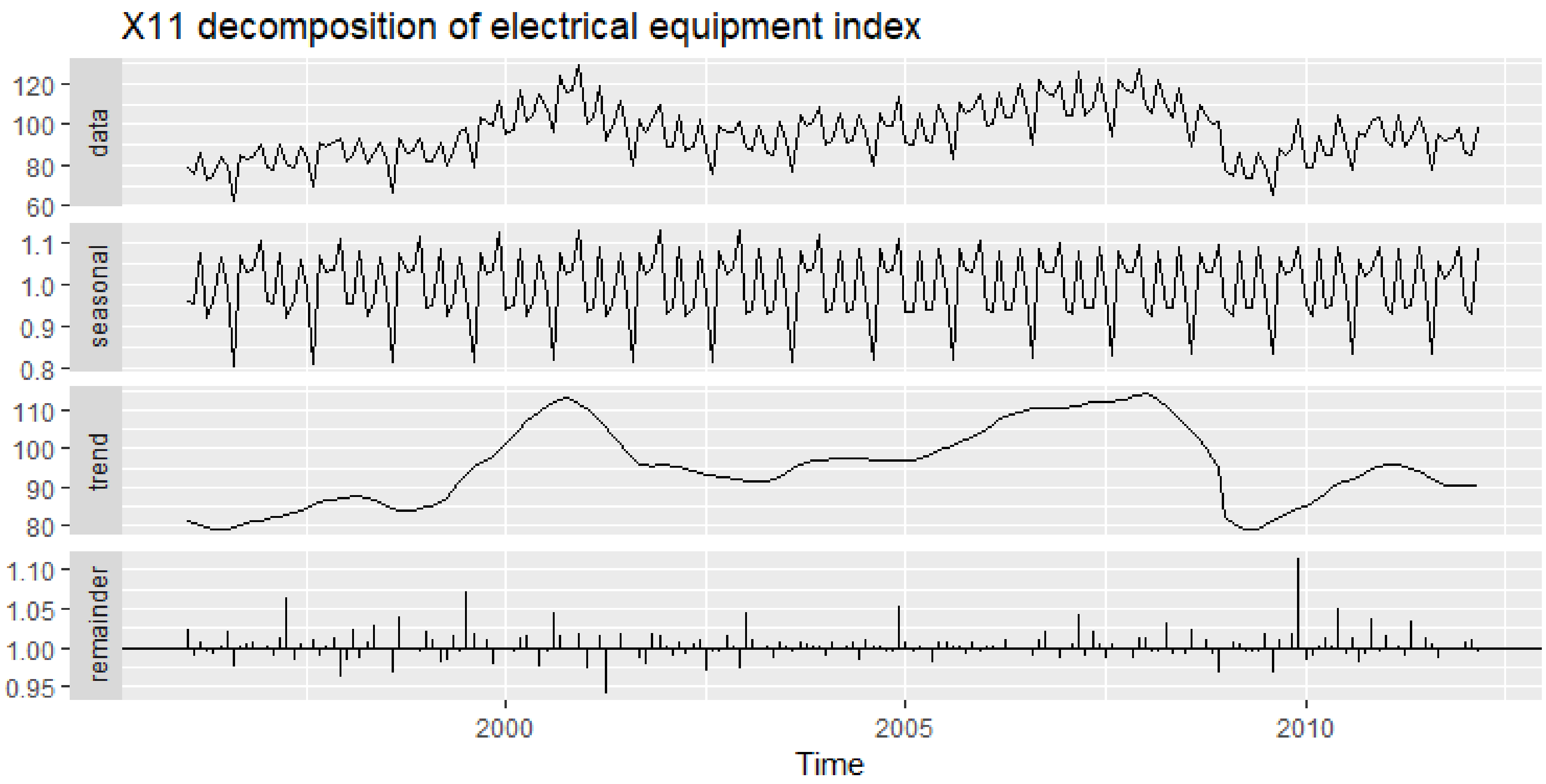
- We use estimates of S based on past values to seasonally adjust a current value.
- Seasonally adjusted series reflect **remainders** as well as **trend**. Therefore they are not “smooth”" and “downturns”" or “upturns” can be misleading.
- It is better to use the trend-cycle component to look for turning points.

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.

X-11 decomposition

```
library(seasonal)
fit <- seas(elecequip, x11="")
autoplot(fit)
```



(Dis)advantages of X-11

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

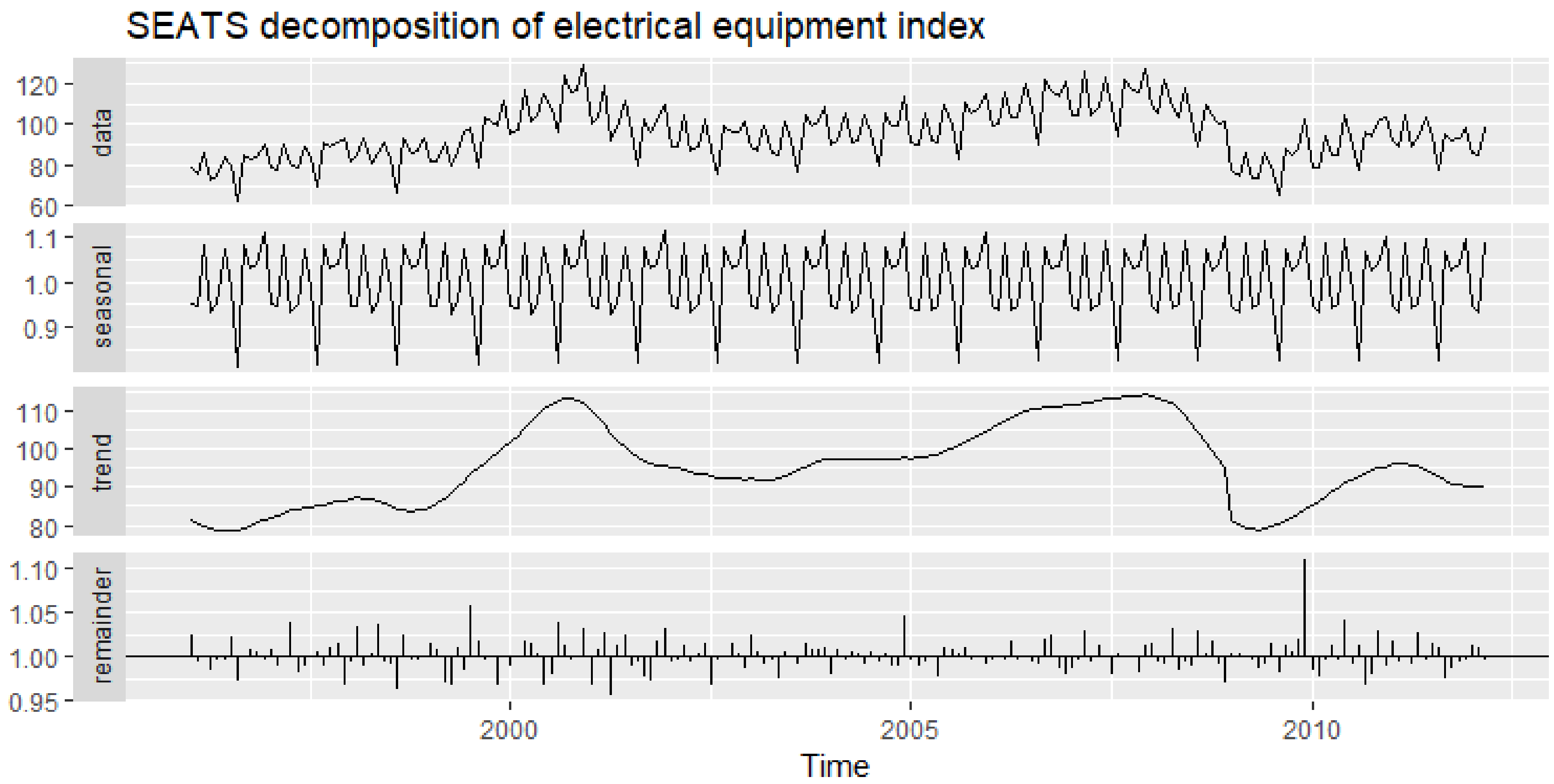
- No prediction/confidence intervals
- Ad hoc method with no underlying model
- Only developed for quarterly and monthly data

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.

SEATS decomposition

```
library(seasonal)
fit <- seas(elecequip)
autoplot(fit)
```



(Dis)advantages of SEATS

Advantages

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

Disadvantages

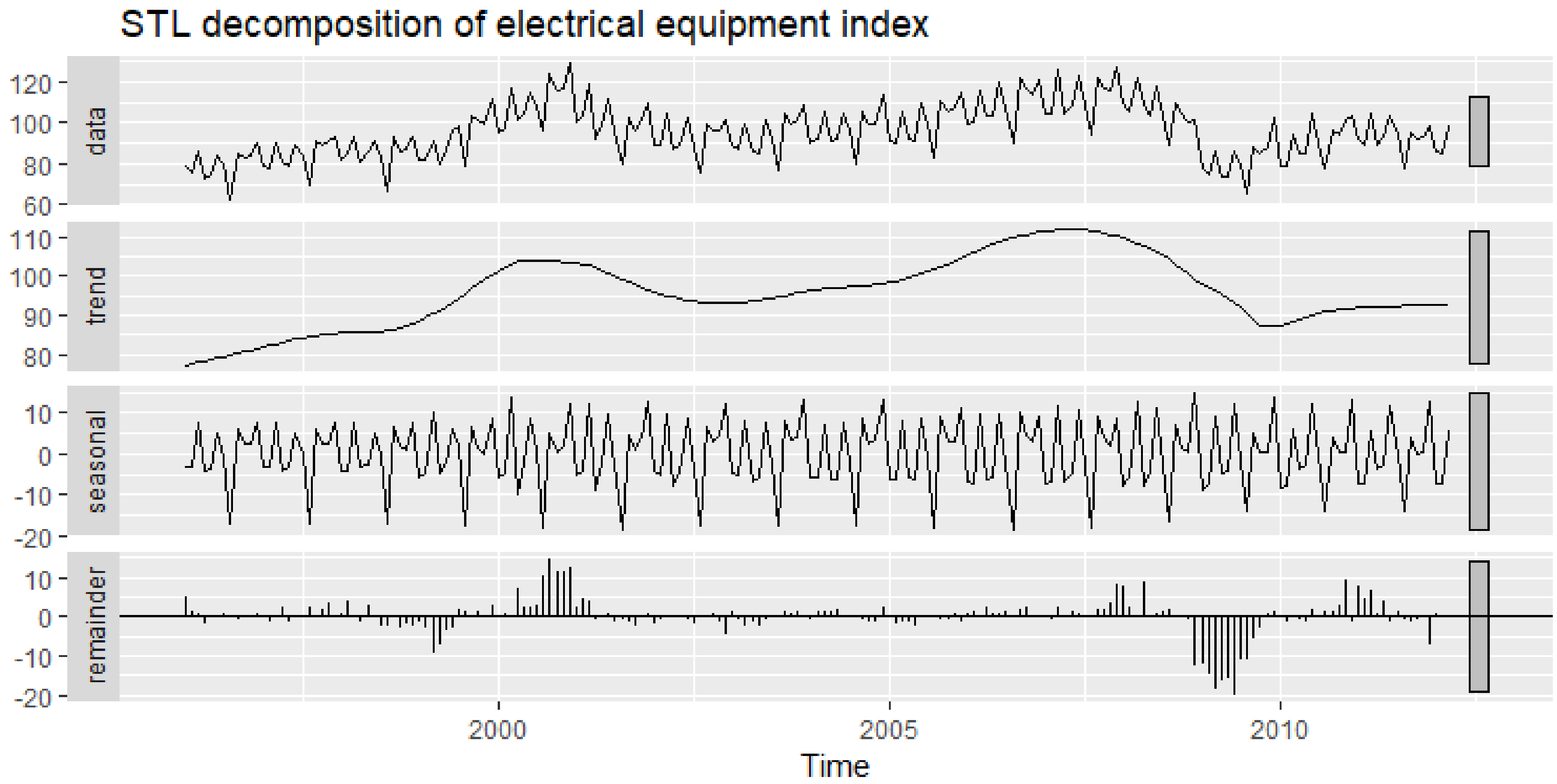
- Only developed for quarterly and monthly data

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

```
fit <- stl(elecequip, s.window=5, robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment index")
```



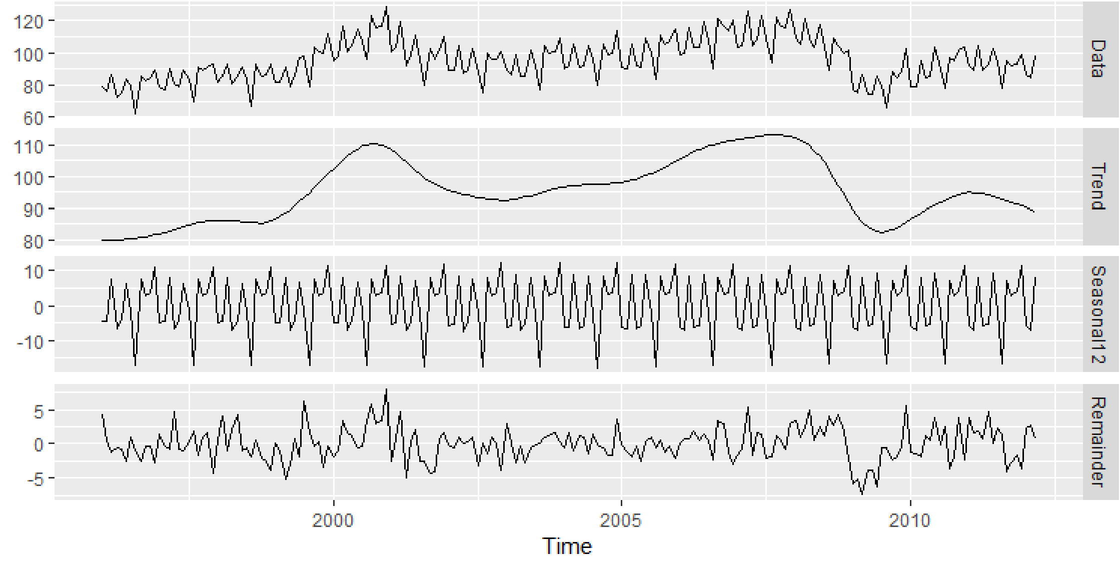
STL decomposition

```
stl(elecequip, s.window=5)
```

```
stl(elecequip, t.window=15,
  s.window="periodic", robust=TRUE)
```

- `t.window` controls wiggliness of trend component.
- `s.window` controls variation on seasonal component.

```
elecequip %>% mstl() %>% autoplot()
```

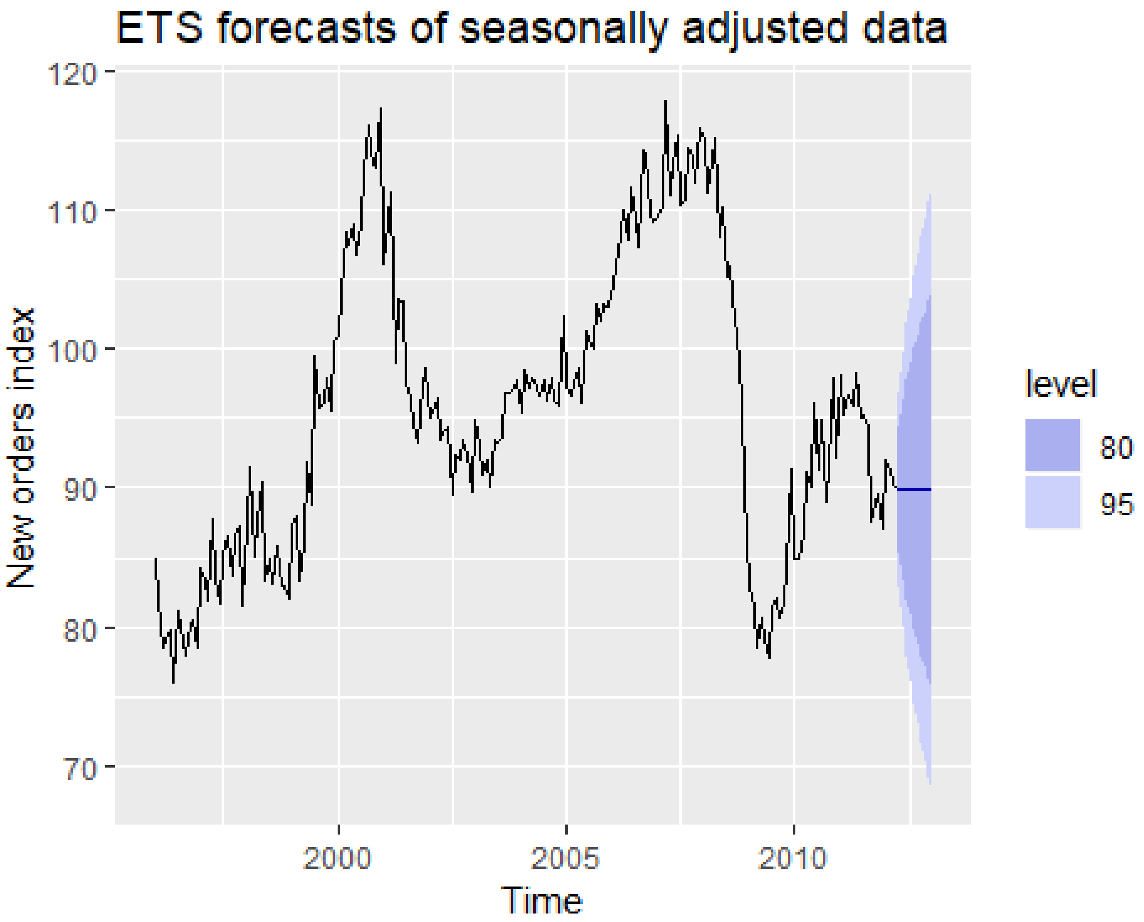


Forecasting and decomposition

- 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.

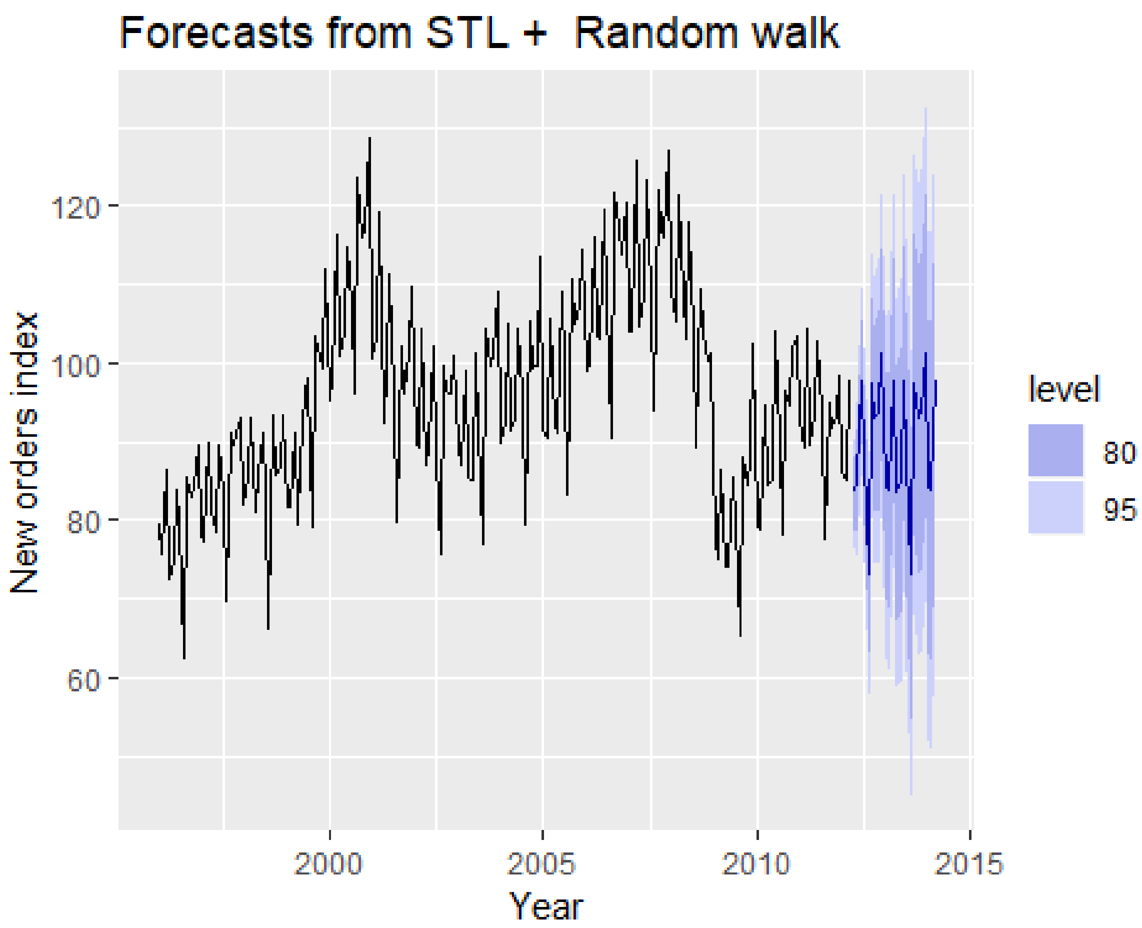
Electrical equipment

```
fit <- stl(elecequip, t.window=13, s.window="periodic")
fit %>% seasadj() %>% naive() %>%
  autoplot() + ylab("New orders index") +
  ggtitle("ETS forecasts of seasonally adjusted data")
```



Electrical equipment

```
fit %>% forecast(method='naive') %>%
  autoplot() + ylab("New orders index") + xlab("Year")
```

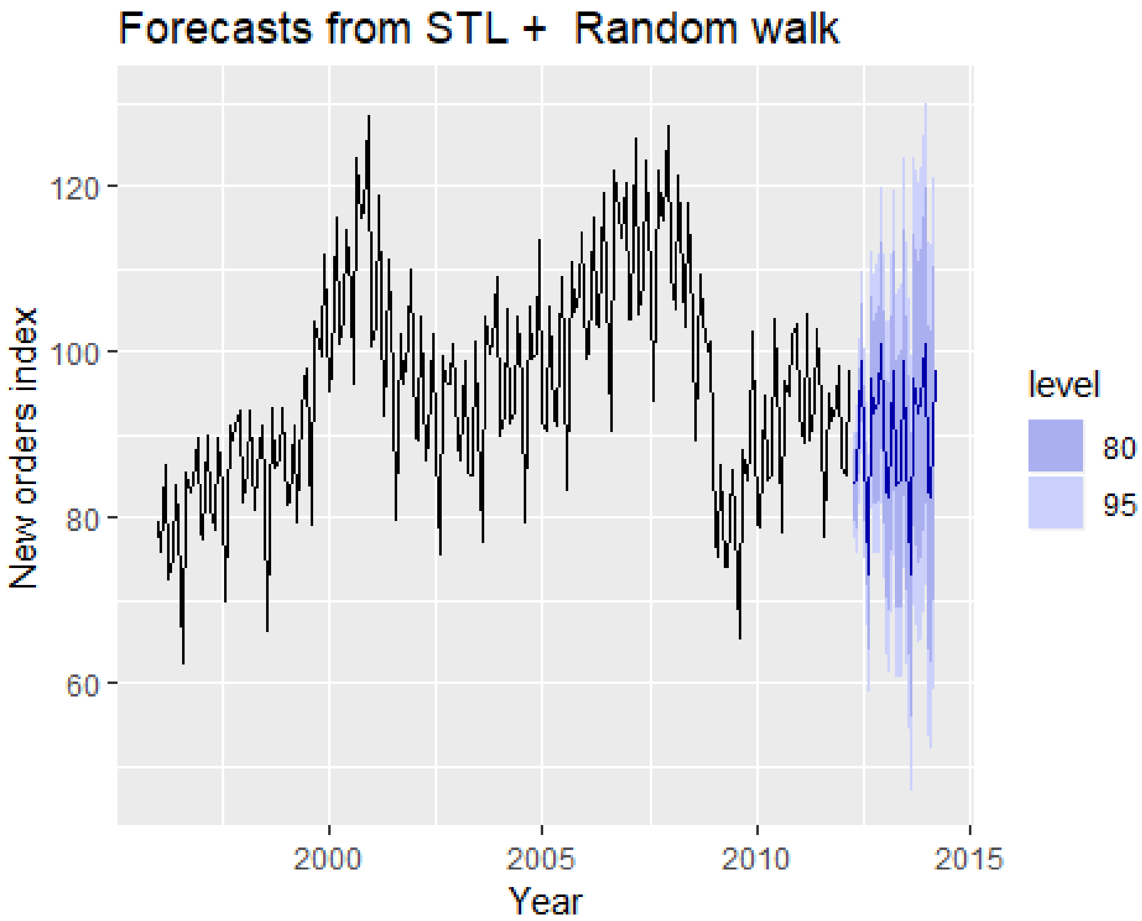


Forecasting and decomposition

A short-cut approach is to use the **stlf()** function. The following code will decompose the time series using STL, forecast the seasonally adjusted series, and return the reseasonalised forecasts.

If method is not specified, it will use the ETS approach applied to the seasonally adjusted series. This usually produces quite good forecasts for seasonal time series.

```
elecequip %>% stlf(method='naive') %>%
  autoplot() + ylab("New orders index") + xlab("Year")
```



Outline

- Decomposition
- Smoothing
- Trend and seasonal methods
- Summary

Simple methods

Time series $y_1, y_2, \dots, y_T$.

Random walk forecasts

$$\hat{y}_{T+h|T} = y_T$$

Average forecasts

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_T$$

- Want something in between that weights most recent data more highly.
- Simple exponential smoothing uses a weighted moving average with weights that decrease exponentially.

Simple Exponential Smoothing – SES

Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $0 \leq \alpha \leq 1$.

Weights assigned to observations for:

Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2	0.4	0.6	0.8
y_{T-1}	0.16	0.24	0.24	0.16
y_{T-2}	0.128	0.144	0.096	0.032
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{T-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

Simple Exponential Smoothing – SES

Component from

Forecast equation $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

- ℓ_t is the level or the smoothed value of the series at time t
- $\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1}$
  Iterate to get exponentially weighted moving average form

- Weighted average form

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T \ell_0$$

Optimisation

- Need to choose value for α and ℓ_0
- Similarly to regression — we choose α and ℓ_0 by minimising SSE:

$$SSE = \sum_{t=1}^{T} (y_t - \hat{y}_{t|t-1})^2$$

Unlike regression there is no closed form solution — use numerical optimization.

Example: Oil production

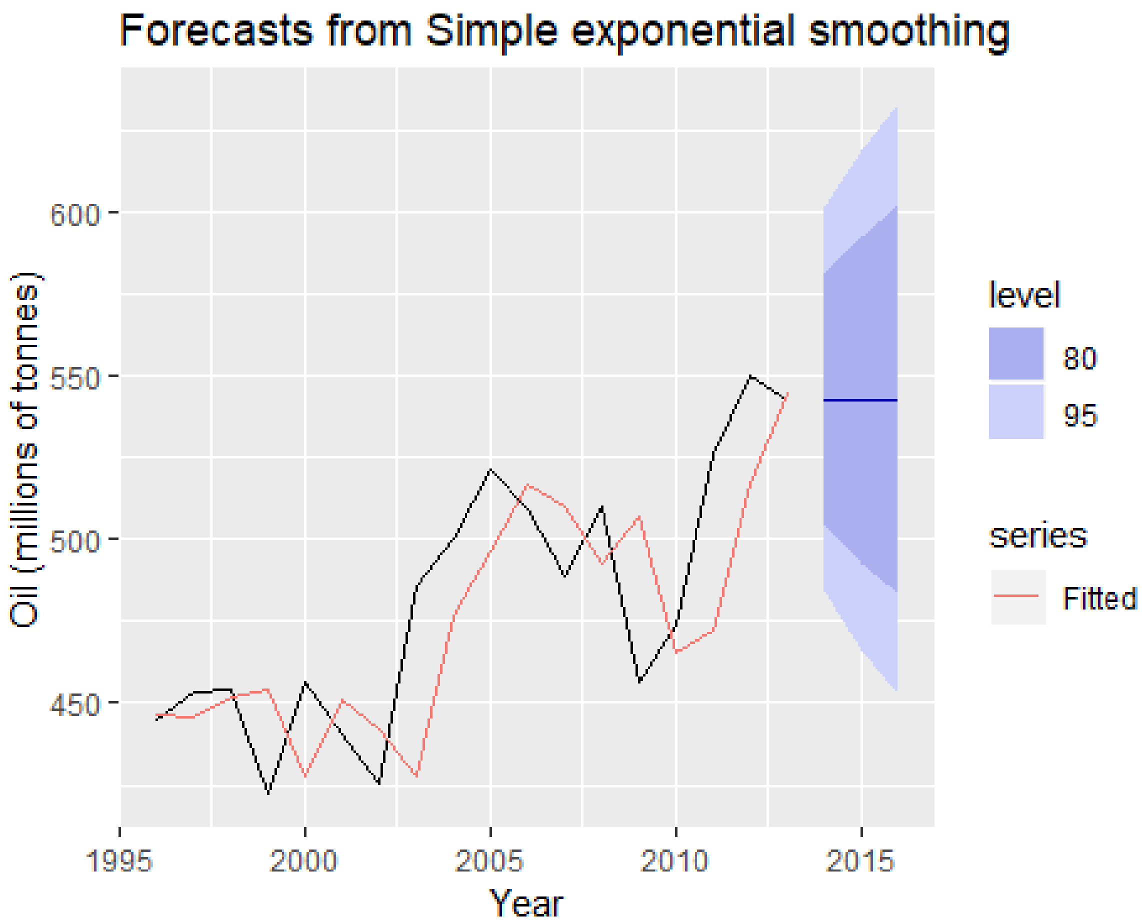
```r
oildata <- window(oil, start=1996)
# Estimate parameters
fc <- ses(oildata, h=5)
summary(fc[["model"]])

## Simple exponential smoothing
##
## Call:
##  ses(y = oildata, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.8339
##
##   Initial states:
##     l = 446.5868
##
##   sigma:  29.83
##
##   AIC  AICc   BIC
## 178.1 179.9 180.8
##
## Training set error measures:
##                 ME  RMSE   MAE   MPE  MAPE   MASE     ACF1
## Training set 6.402 28.12 22.26 1.098 4.611 0.9257 -0.03378
```

Example: Oil production

Year	Time	Observation	Level	Forecast
	t	y_t	ℓ_t	$\hat{y}_{t+1\|t}$
1995	0		446.59	
1996	1	445.36	445.57	446.59
1997	2	453.20	451.93	445.57
1998	3	454.41	454.00	451.93
1999	4	422.38	427.63	454.00
2000	5	456.04	451.32	427.63
2001	6	440.39	442.20	451.32
2002	7	425.19	428.02	442.20
2003	8	486.21	476.54	428.02
2004	9	500.43	496.46	476.54
2005	10	521.28	517.15	496.46
2006	11	508.95	510.31	517.15
2007	12	488.89	492.45	510.31
2008	13	509.87	506.98	492.45
2009	14	456.72	465.07	506.98
2010	15	473.82	472.36	465.07
2011	16	525.95	517.05	472.36
2012	17	549.83	544.39	517.05
2013	18	542.34	542.68	544.39
	h			$\hat{y}_{T+h\|T}$
2014	1			542.68

Example: Oil production

```
autoplot(fc) +
  autolayer(fitted(fc), series="Fitted") +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```



Outline

- Decomposition
- Smoothing
- Trend and seasonal methods
- Summary

Holt’s linear trend

Component from

Forecast $\hat{y}_{t+h|t} = \ell_t + hb_t$

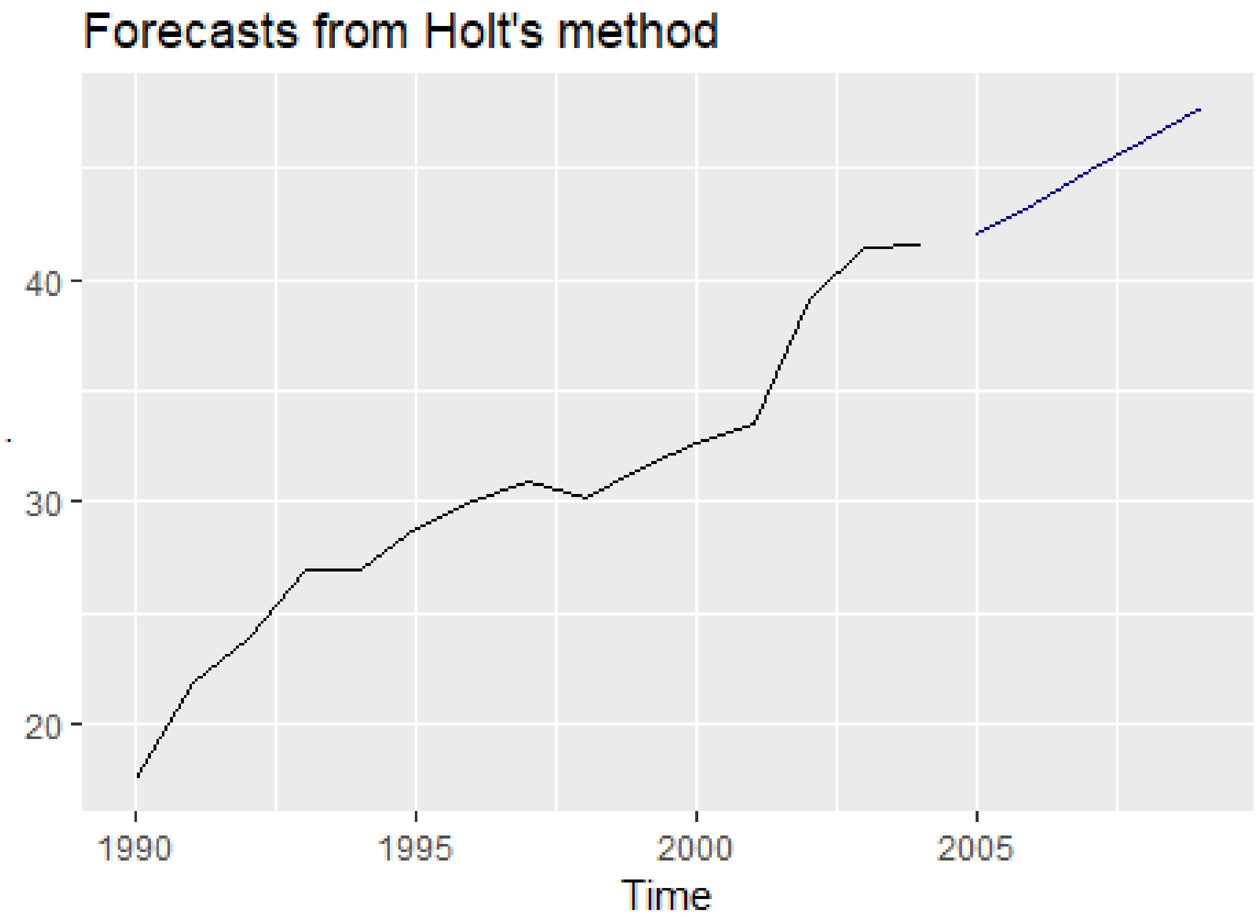
Level $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1}+b_{t-1})$

Trend $b_t = \beta^*(\ell_t+\ell_{t-1}) + (1-\beta^*)\, b_{t-1}$

- Two smoothing parameters α and β^* $(0 \leq \alpha, \beta^* \leq 1)$.
- ℓ_t level: weighted average between y_t and one-step ahead forecast for time t, $(\ell_{t-1}+b_{t-1} = \hat{y}_{t|t-1})$
- b_t slope: weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1} ,current and previous estimate of slope.
- Choose $\alpha, \beta^*, \ell_0, b_0$ to minimise SSE.

Holt’s method in R

```
window(ausair, start=1990, end=2004) %>%
  holt(h=5, PI=FALSE) %>%
  autoplot()
```



Seasonal methods: Holt-Winters additive method

Holt and Winters extended Holt's method to capture seasonality.

Component from

Forecast $\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$

Level $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1}+b_{t-1})$

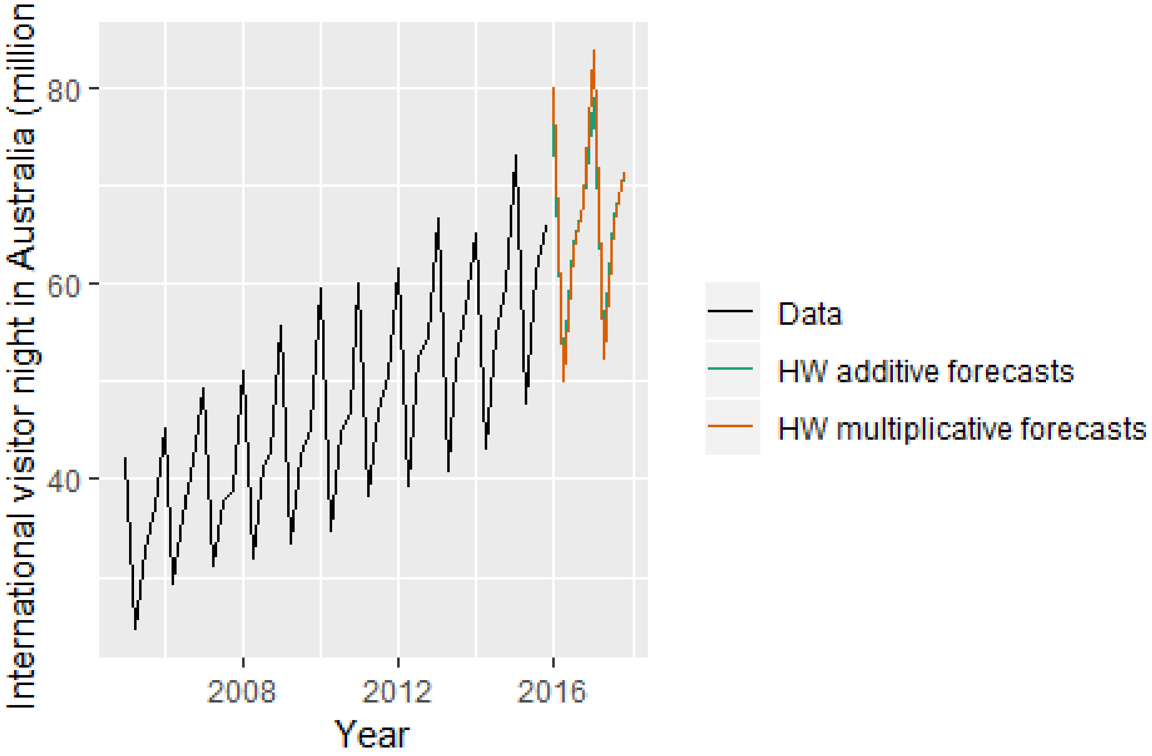
Trend $b_t = \beta^*(\ell_t+\ell_{t-1}) + (1-\beta^*)\, b_{t-1}$

Season $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)\, s_{t-m}$

- $k =$ integer part of $(h-1)/m$. Ensures estimates from the final year are used for forecasting.
- Parameters: $0 \le \alpha \le 1,\ 0 \le \beta^* \le 1$ $,0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality (e.g. $m = 4$ for quarterly data).

Example: Visitor Nights

```r
aust <- window(austourists,start=2005)
fit1 <- hw(aust,seasonal="additive")
fit2 <- hw(aust,seasonal="multiplicative")
```



Outline

- Decomposition
- Smoothing
- Trend and seasonal methods
- Summary