MONASH University

MONASH BUSINESS SCHOOL

Time Series in R: Forecasting and Visualisation

Some automatic forecasting algorithms

29 May 2017

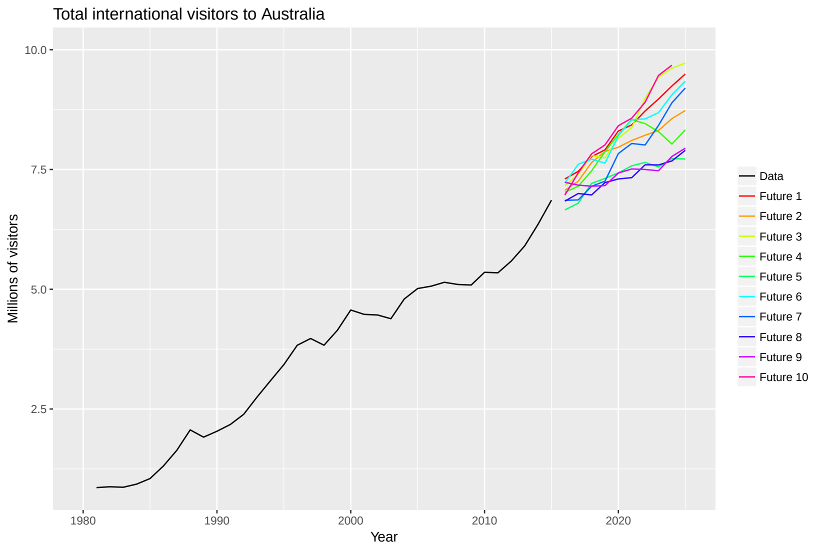
Outline

Forecasting

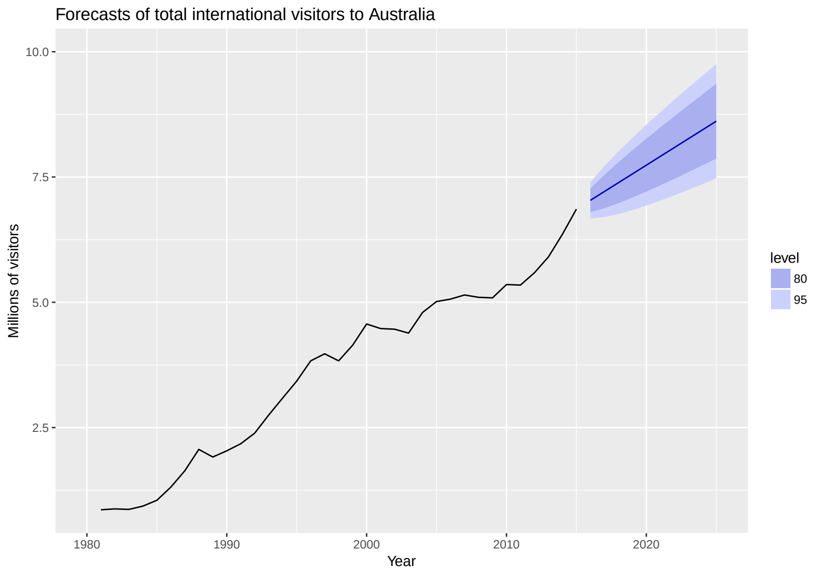
Forecasting is estimating how the sequence of observations will continue into the future.

- We usually think probabilistically about future sample paths
- What range of values covers the possible sample paths with 80% probability?

Sample futures



Forecast intervals



Automatic forecasting algorithms

1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.

Automatic forecasting algorithms

1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.

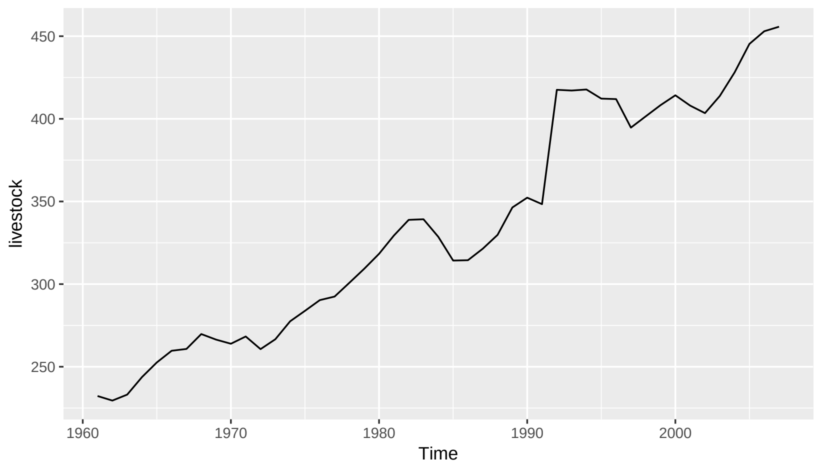
Specifications

Automatic forecasting algorithms must:

- determine an appropriate time series model;
- estimate the parameters;
- compute the forecasts with prediction intervals.

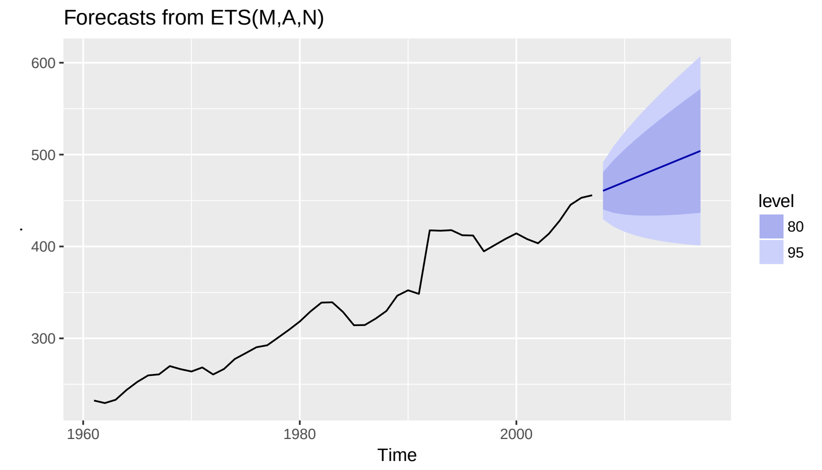
Example: Asian sheep

```
autoplot(livestock)
```



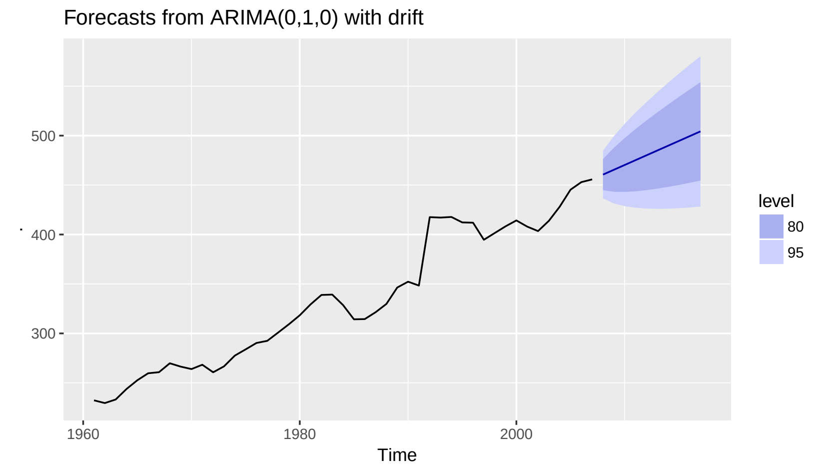
Example: Asian sheep

```
livestock %>% ets %>% forecast %>% autoplot
```



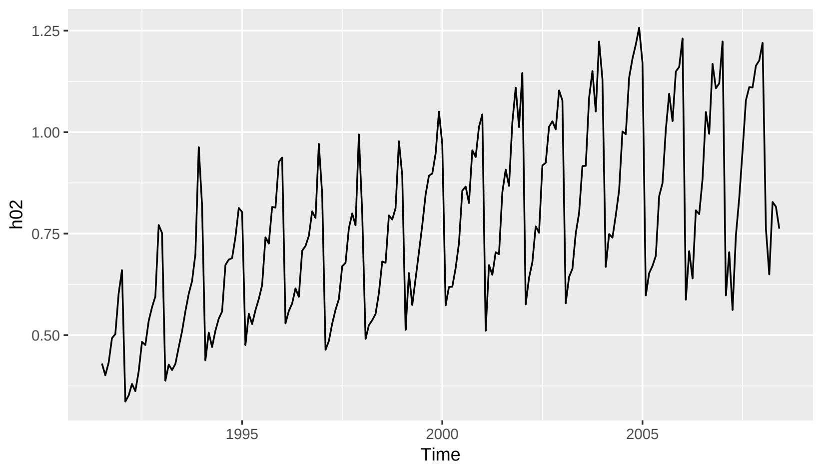
Example: Asian sheep

```
livestock %>% auto.arima %>% forecast %>% autoplot
```



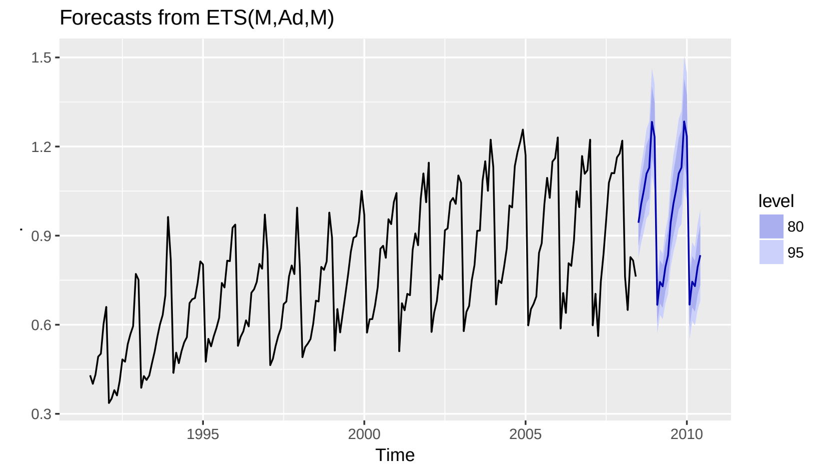
Example: Cortecosteroid sales

```r
autoplot(h02)
```



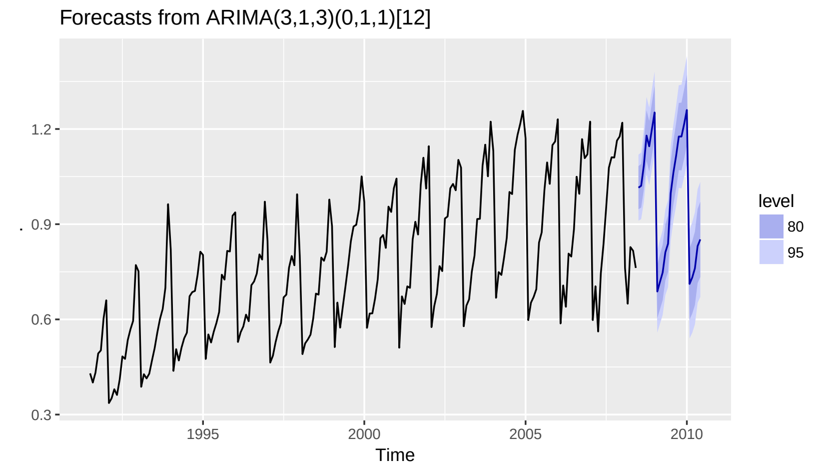
Example: Cortecosteroid sales

```
h02 %>% ets %>% forecast %>% autoplot
```



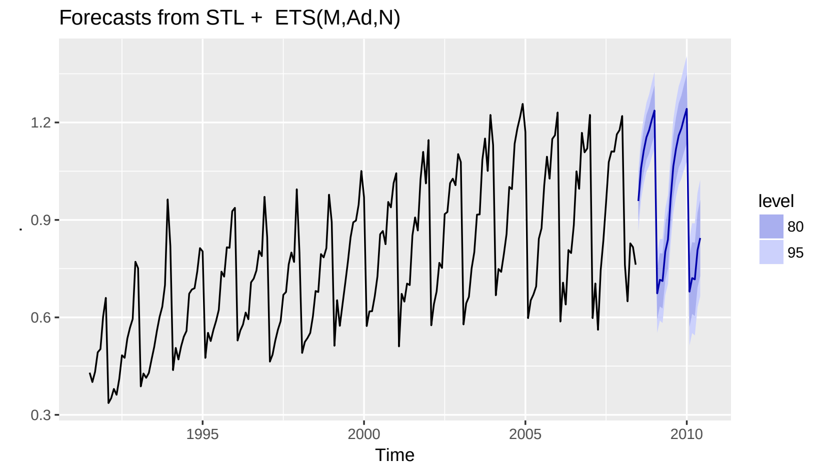
Example: Cortecosteroid sales

```
h02 %>% auto.arima %>% forecast %>% autoplot
```



Example: Cortecosteroid sales

```
h02 %>% stlf %>% autoplot
```



Outline

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
A,A: Additive Holt-Winters' method
A,M: Multiplicative Holt-Winters' method
A_d,M: Damped multiplicative Holt-Winters' method

There are also multiplicative trend methods (not recommended).

Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

- There are 9 separate exponential smoothing methods.
- Each can have an additive or multiplicative error, giving 18 separate models.
- Models with additive and multiplicative errors give the same point forecasts but different prediction intervals.

Exponential smoothing methods

Trend Component		Seasonal Component		
		N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**easonal

Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**easonal

ETS(Error,Trend,Seasonal):

- Error = {A,M}
- Trend = {N,A,A_d}
- Seasonal = {N,A,M}.

Automatic ETS forecasting

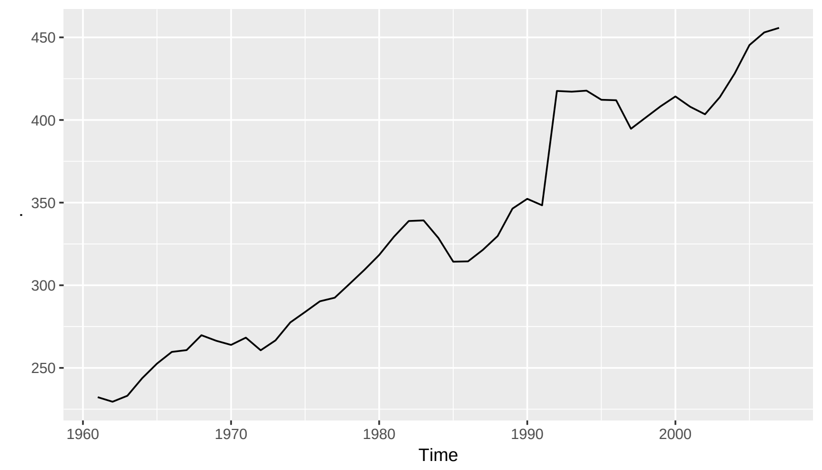
From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AICc:
- Produce forecasts using best method.
- Obtain forecast intervals using underlying state space model.

Method performed very well in M3 competition.

Example: Asian sheep

```
livestock %>% autoplot
```



Example: Asian sheep

```r
livestock %>% ets
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = .)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##
##   Initial states:
##     l = 225.1784
##     b = 4.8307
##
##   sigma:  0.0344
##
##   AIC  AICc   BIC
## 418.7 420.2 427.9
```

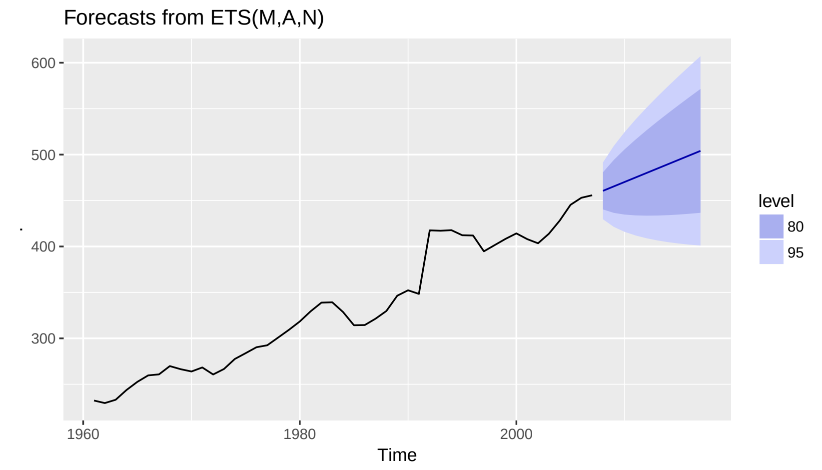
Example: Asian sheep

```
livestock %>% ets %>% forecast
```

```
##      Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2008          460.6 440.3 480.9 429.6 491.6
## 2009          465.4 436.6 494.2 421.3 509.5
## 2010          470.2 434.7 505.8 415.9 524.6
## 2011          475.1 433.8 516.3 412.0 538.1
## 2012          479.9 433.5 526.3 409.0 550.8
## 2013          484.7 433.7 535.8 406.6 562.8
## 2014          489.6 434.1 545.0 404.7 574.4
## 2015          494.4 434.8 554.0 403.2 585.6
## 2016          499.2 435.6 562.8 402.0 596.5
## 2017          504.1 436.7 571.4 401.0 607.1
```

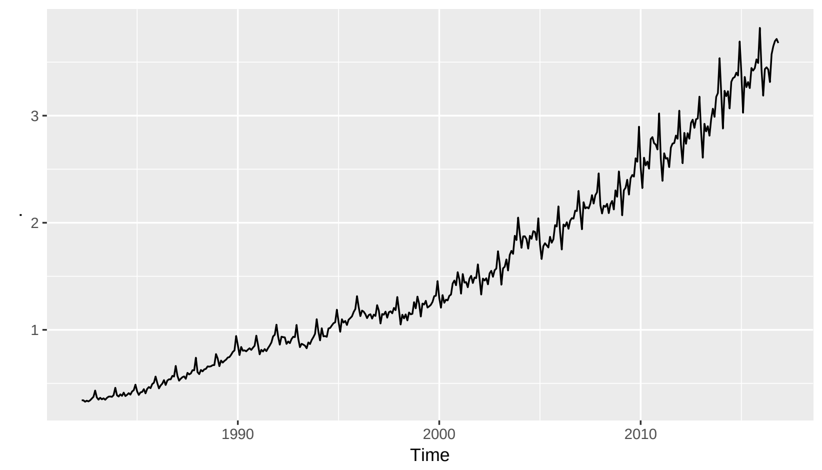
Example: Asian sheep

```
livestock %>% ets %>% forecast %>% autoplot
```



Example: Australian eating-out expenditure

```
auscafe %>% autoplot
```



Example: Australian eating-out expenditure

```r
auscafe %>% ets
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = .)
##
##   Smoothing parameters:
##     alpha = 0.5793
##     beta  = 0.0061
##     gamma = 0.2098
##
##   Initial states:
##     l = 0.3458
##     b = 0.0038
##     s=0.9875 0.9452 1.021 1.181 1.026 1.008
##            0.9728 0.9793 0.9796 0.9379 0.9931 0.9686
##
##   sigma:  0.0245
##
##     AIC    AICc     BIC
## -339.1 -337.6 -270.6
```

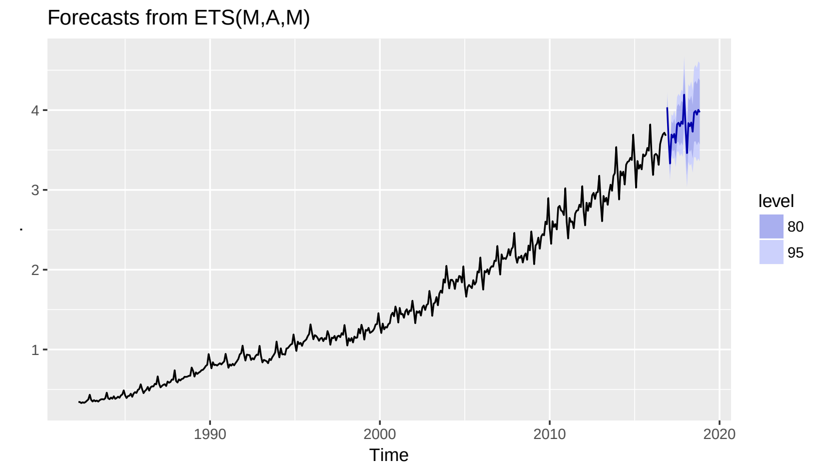
Example: Australian eating-out expenditure

```
auscafe %>% ets %>% forecast
```

```
##          Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## Dec 2016          4.035 3.908 4.162 3.841 4.229
## Jan 2017          3.637 3.505 3.769 3.435 3.839
## Feb 2017          3.331 3.195 3.467 3.123 3.539
## Mar 2017          3.693 3.528 3.859 3.440 3.946
## Apr 2017          3.656 3.478 3.833 3.384 3.928
## May 2017          3.701 3.508 3.894 3.406 3.996
## Jun 2017          3.591 3.392 3.790 3.286 3.896
## Jul 2017          3.823 3.599 4.047 3.480 4.166
## Aug 2017          3.841 3.604 4.079 3.479 4.204
## Sep 2017          3.798 3.552 4.044 3.422 4.174
## Oct 2017          3.856 3.596 4.117 3.458 4.255
## Nov 2017          3.827 3.558 4.096 3.415 4.239
## Dec 2017          4.194 3.878 4.509 3.711 4.676
## Jan 2018          3.780 3.486 4.074 3.330 4.229
## Feb 2018          3.461 3.183 3.739 3.036 3.886
## Mar 2018          3.837 3.520 4.155 3.351 4.323
```

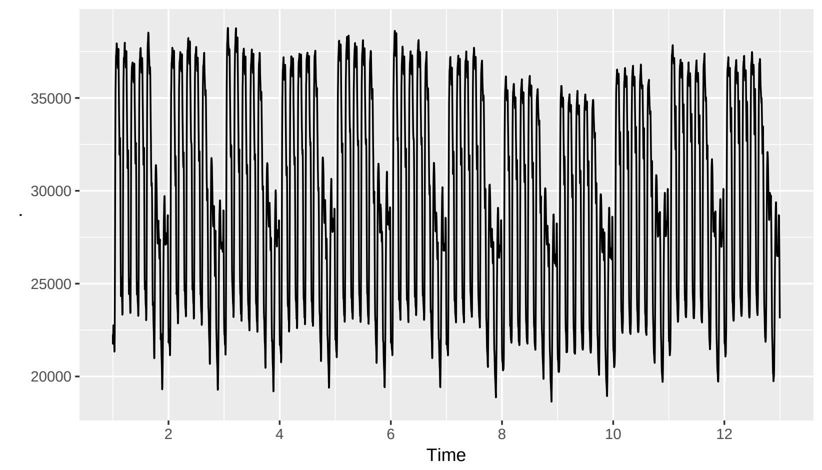
Example: Australian eating-out expenditure

```
auscafe %>% ets %>% forecast %>% autoplot
```



Example: Half-hourly electricity demand

```
taylor %>% autoplot
```



Example: Half-hourly electricity demand

```r
taylor %>% ets
```

```
## Warning in ets(.): I can't handle data with
## frequency greater than 24. Seasonality will
## be ignored. Try stlf() if you need seasonal
## forecasts.

## ETS(A,Ad,N)
##
## Call:
##  ets(y = .)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.9999
##     phi   = 0.8696
##
##   Initial states:
##     l = 22509.4085
```

Outline

Lab Session 5

Outline

Variance stabilization

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

Variance stabilization

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

Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

Variance stabilization

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

Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

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

Variance stabilization

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

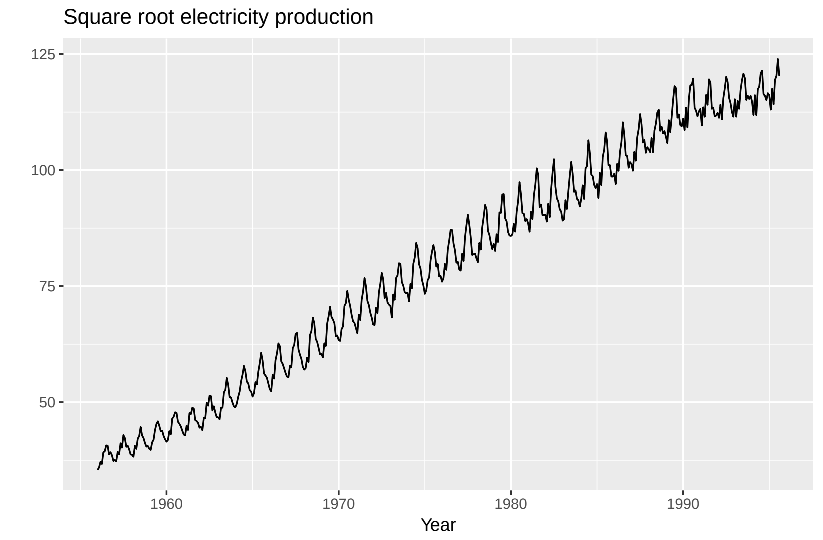
Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

Mathematical transformations for stabilizing variation

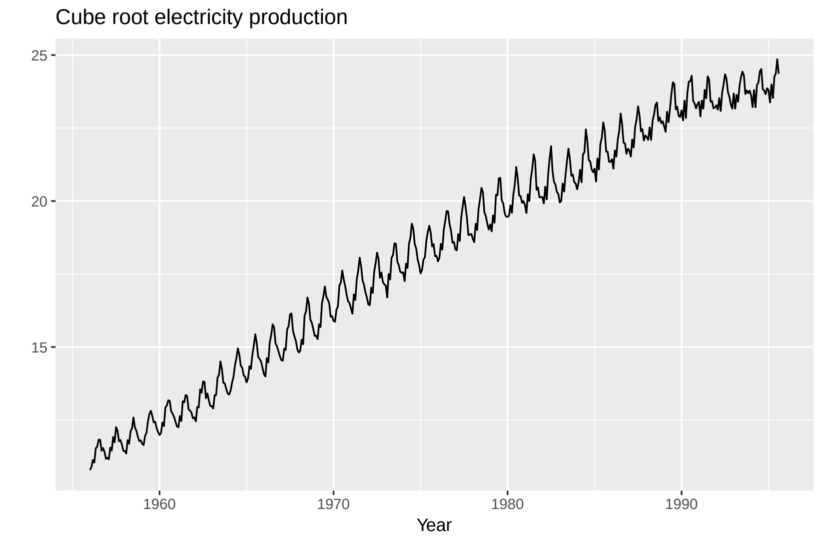
Square root	$w_t = \sqrt{y_t}$	$\downarrow$
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

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

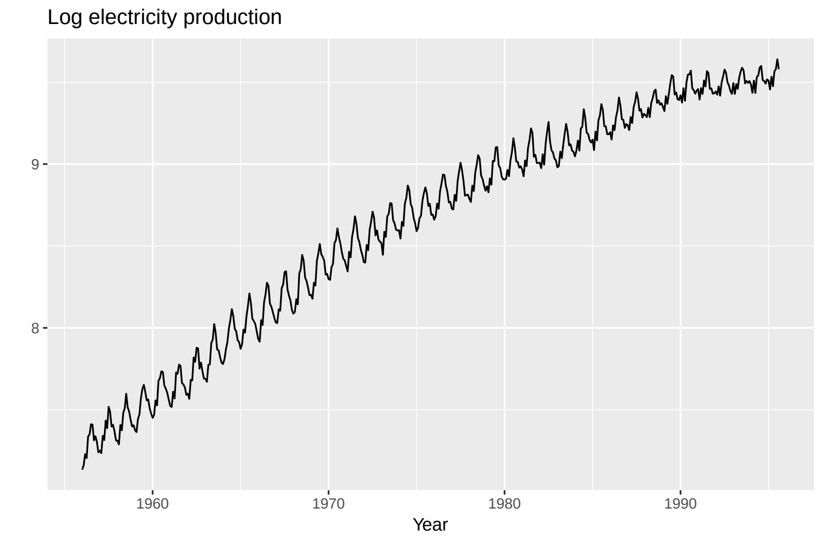
Variance stabilization



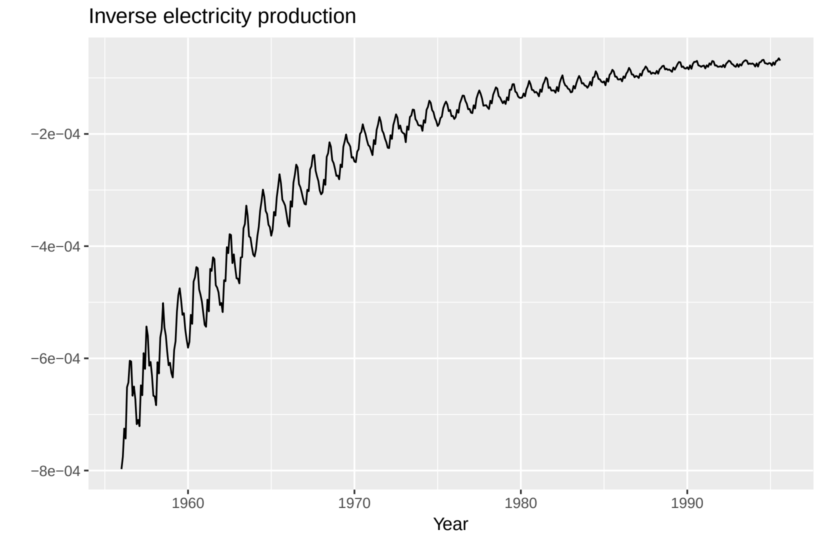
Variance stabilization



Variance stabilization



Variance stabilization



Box-Cox transformations

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

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

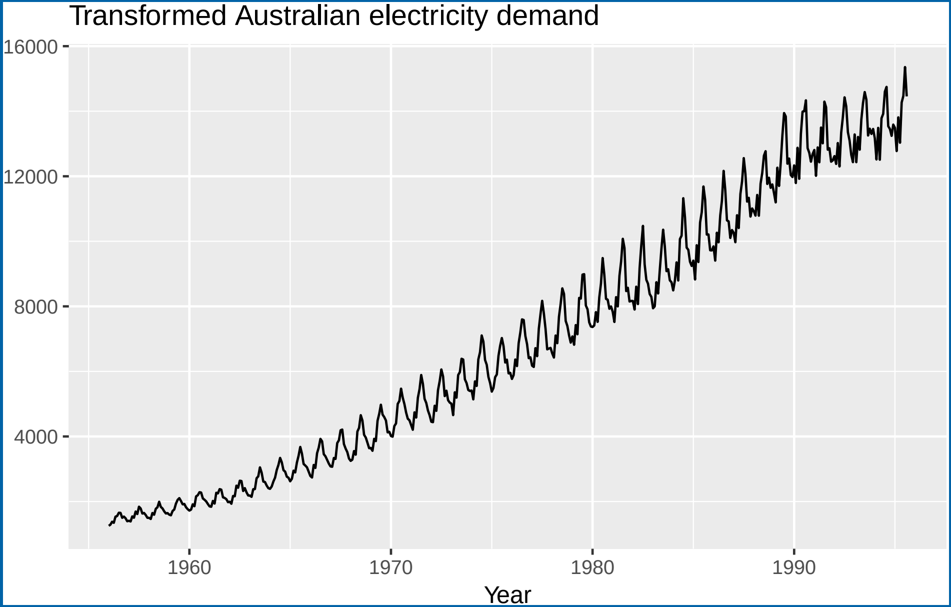
Box-Cox transformations

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

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

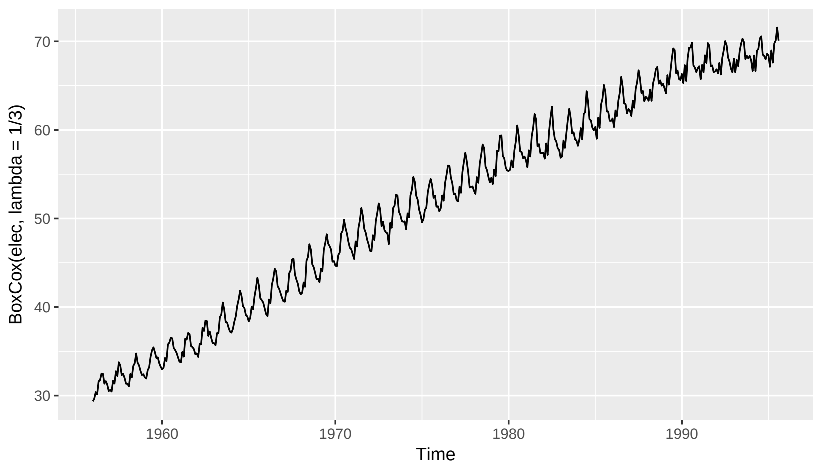
- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Box-Cox transformations



Box-Cox transformations

```
autoplot(BoxCox(elec,lambda=1/3))
```



Automated Box-Cox transformations

```
(BoxCox.lambda(elec))
```

```
## [1] 0.2654
```

Automated Box-Cox transformations

```
(BoxCox.lambda(elec))
```

```
## [1] 0.2654
```

- 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.

Outline

How does auto.arima() work?

Non-seasonal version:

A non-seasonal ARIMA process

$$\phi(B)(1-B)^d y_t = c + \theta(B)\varepsilon_t$$

Need to select appropriate orders: p, q, d.

Hyndman and Khandakar (JSS, 2008) algorithm:

- Select no. differences d via unit root tests.
- Select p, q by minimising AICc.
- Use stepwise search to traverse model space.

How does auto.arima() work?

Non-seasonal version:

Step1: Select current model (with smallest AICc) from:
ARIMA(2, d, 2)
ARIMA(0, d, 0)
ARIMA(1, d, 0)
ARIMA(0, d, 1)

How does auto.arima() work?

Non-seasonal version:

Step1: Select current model (with smallest AICc) from:
ARIMA$(2, d, 2)$
ARIMA$(0, d, 0)$
ARIMA$(1, d, 0)$
ARIMA$(0, d, 1)$

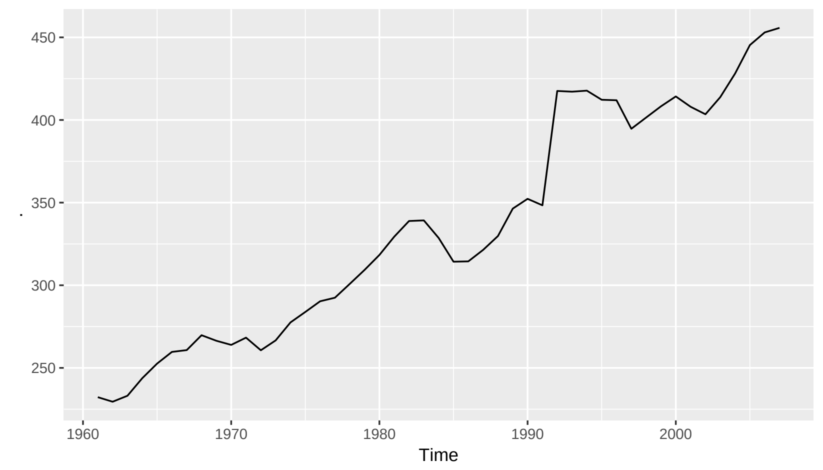
Step 2: Consider variations of current model:

- vary one of $p, q,$ from current model by ± 1;
- p, q both vary from current model by ± 1;
- Include/exclude c from current model.

Model with lowest AICc becomes current model.
Repeat Step 2 until no lower AICc can be found.

Example: Asian sheep

```
livestock %>% autoplot
```



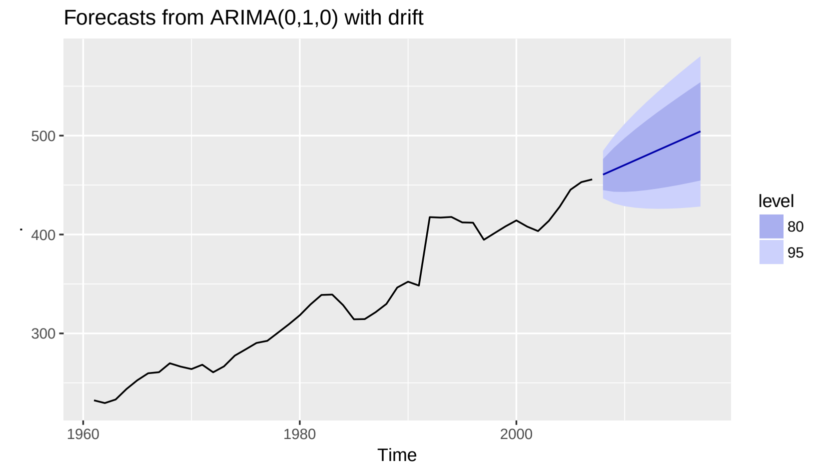
Example: Asian sheep

```
livestock %>% auto.arima
```

```
## Series: .
## ARIMA(0,1,0) with drift
##
## Coefficients:
##       drift
##       4.858
## s.e.  1.789
##
## sigma^2 estimated as 150:  log likelihood=-180.1
## AIC=364.2   AICc=364.4   BIC=367.8
```

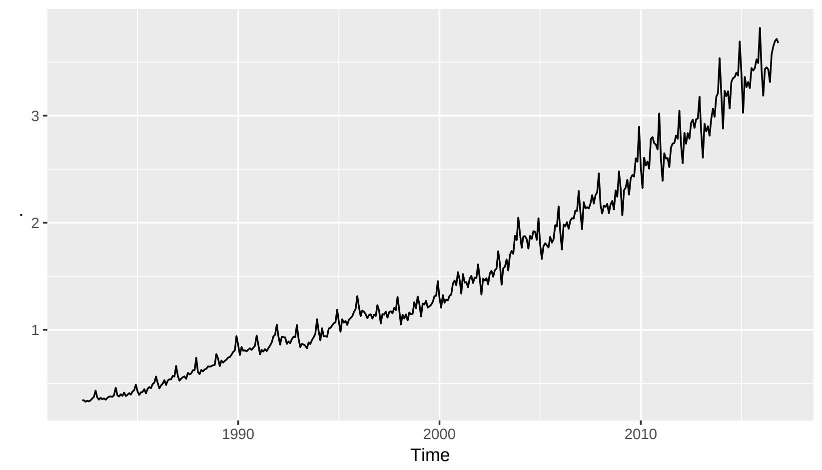
Example: Asian sheep

```
livestock %>% auto.arima %>% forecast %>% autoplot
```



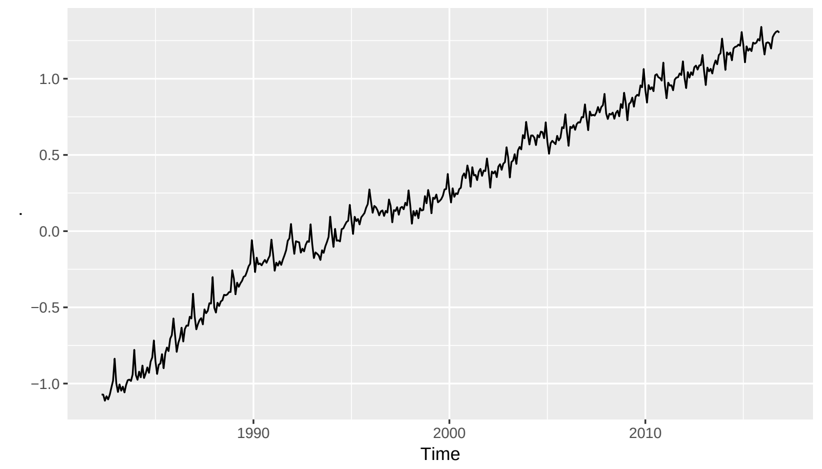
Example: Australian eating-out expenditure

```
auscafe %>% autoplot
```



Example: Australian eating-out expenditure

```r
auscafe %>% BoxCox(lambda=0) %>% autoplot
```



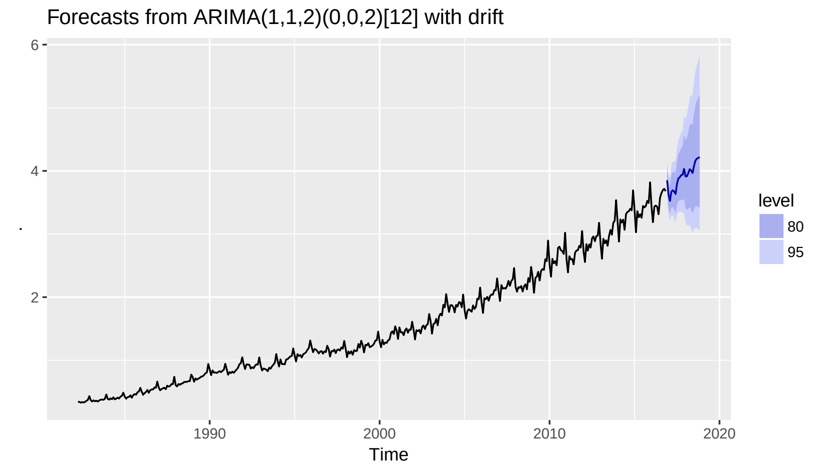
Example: Australian eating-out expenditure

```r
auscafe %>% auto.arima(lambda=0)
```

```
## Series: .
## ARIMA(1,1,2)(0,0,2)[12] with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ma1      ma2    sma1    sma2  drift
##       -0.787   0.516   -0.389   0.746   0.474  0.006
## s.e.   0.051   0.061    0.046   0.054   0.039  0.002
##
## sigma^2 estimated as 0.0013:  log likelihood=788.8
## AIC=-1564   AICc=-1563   BIC=-1535
```

Example: Australian eating-out expenditure

```r
auscafe %>% auto.arima(lambda=0) %>%
  forecast %>% autoplot
```



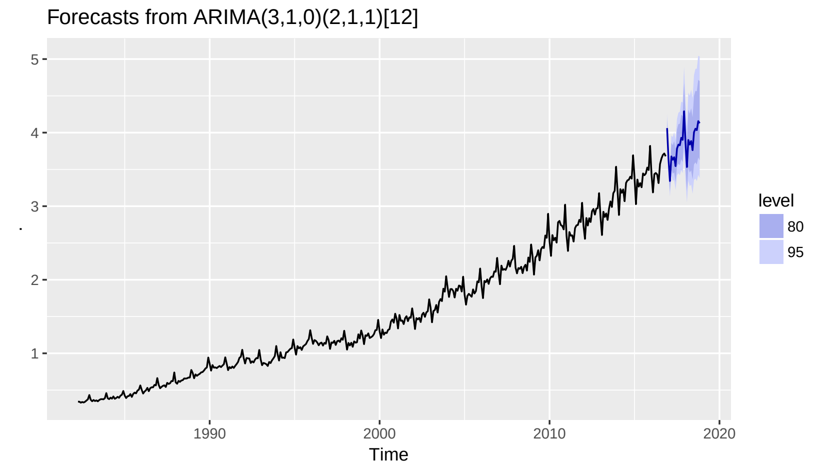
Example: Australian eating-out expenditure

```r
auscafe %>% auto.arima(lambda=0, D=1)
```

```
## Series: .
## ARIMA(3,1,0)(2,1,1)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2    ar3   sar1    sar2
##       -0.340  -0.108  0.095  0.125  -0.059
## s.e.   0.052   0.053  0.050  0.066   0.059
##         sma1
##       -0.828
## s.e.   0.044
##
## sigma^2 estimated as 0.000572:  log likelihood=929.5
## AIC=-1845   AICc=-1845   BIC=-1817
```

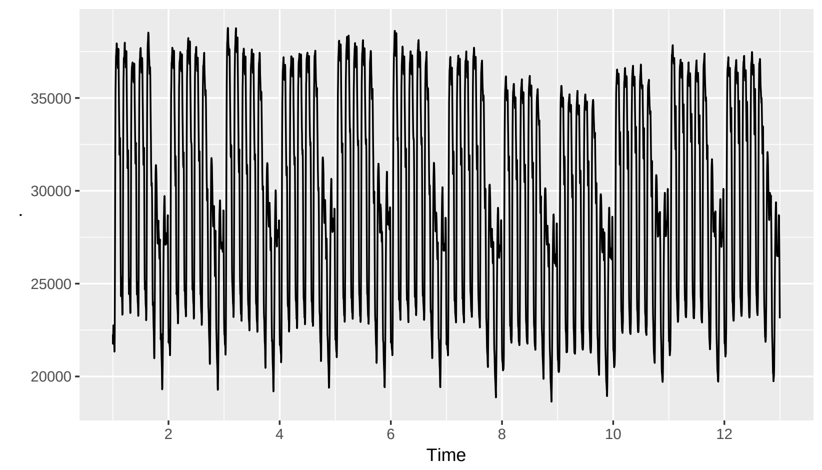
Example: Australian eating-out expenditure

```
auscafe %>% auto.arima(lambda=0, D=1) %>%
  forecast %>% autoplot
```



Example: Half-hourly electricity demand

```
taylor %>% autoplot
```



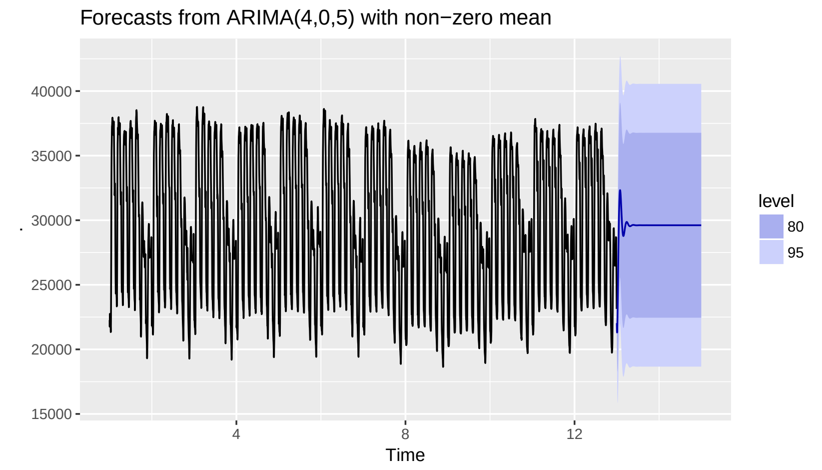
Example: Half-hourly electricity demand

```r
taylor %>% auto.arima
```

```
## Series: .
## ARIMA(4,0,5) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3    ar4    ma1     ma2
##        1.726  -0.378  -0.553  0.190  0.391  -0.075
## s.e.   0.080   0.180   0.151  0.053  0.080   0.052
##           ma3     ma4     ma5     mean
##        -0.332  -0.255  -0.177  29611.3
## s.e.    0.019   0.032   0.021    230.6
##
## sigma^2 estimated as 159426:  log likelihood=-29870
## AIC=59762   AICc=59762   BIC=59832
```

Example: Half-hourly electricity demand

```
taylor %>% auto.arima %>% forecast %>% autoplot
```



Outline

Lab Session 6

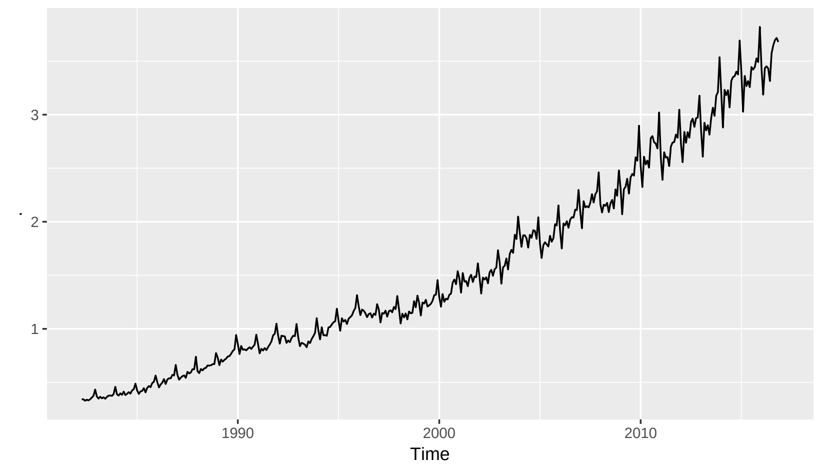
Outline

STLF

- Decomposes time series into a trend, seasonal and remainder component using STL decomposition
- Forecasts the seasonally adjusted series using ETS
- Forecasts the seasonal component using a “seasonal naive” approach (replicating last available year).
- Combines them to get forecasts for original series.
- May need a Box-Cox transformation

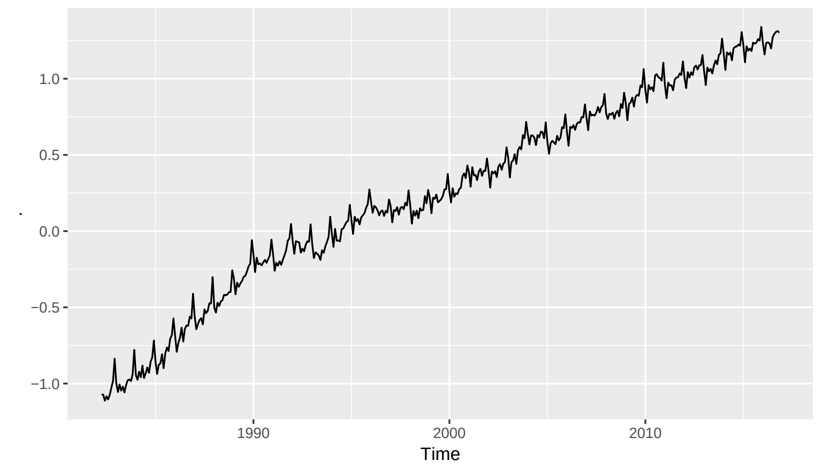
Example: Australian eating-out expenditure

```
auscafe %>% autoplot
```



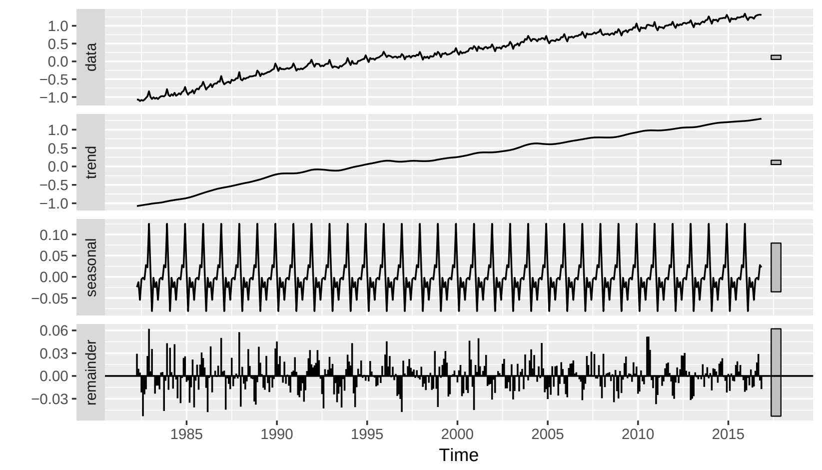
Example: Australian eating-out expenditure

```r
auscafe %>% BoxCox(lambda=0) %>% autoplot
```



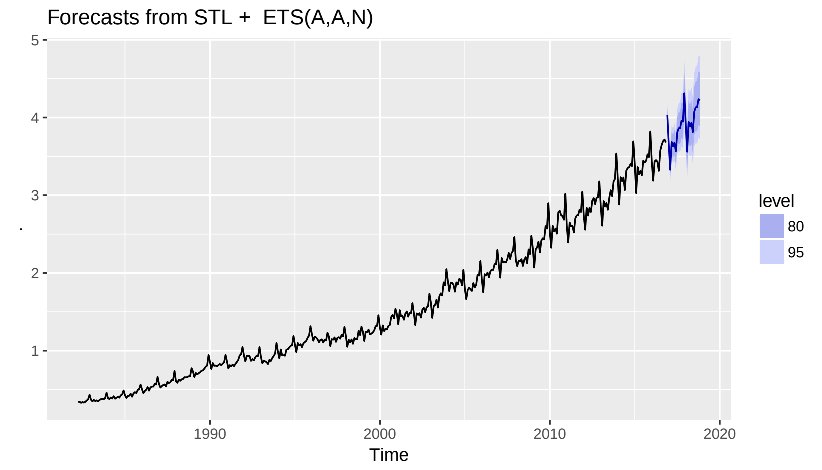
Example: Australian eating-out expenditure

```r
auscafe %>% BoxCox(lambda=0) %>%
  stl(s.window='periodic') %>% autoplot
```



Example: Australian eating-out expenditure

```
auscafe %>% stlf(lambda=0) %>% autoplot
```



Example: Half-hourly electricity demand

```
taylor %>% stlf %>% autoplot
```