



# Forecasting: principles and practice

Rob J Hyndman

2.1 Transformations

# Outline

- 1 Variance stabilization
- 2 Box-Cox transformations
- 3 Back-transformation
- 4 Lab session 10

# 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	$w_t = \sqrt{y_t}$	↓
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.

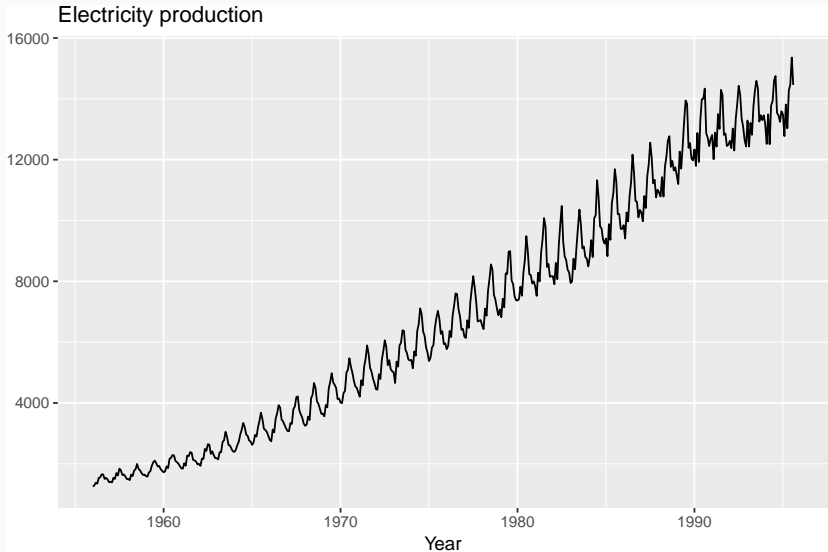
Denote original observations as  $y_1, \dots, y_n$  and transformed observations as  $w_1, \dots, w_n$ .

## Mathematical transformations for stabilizing variation

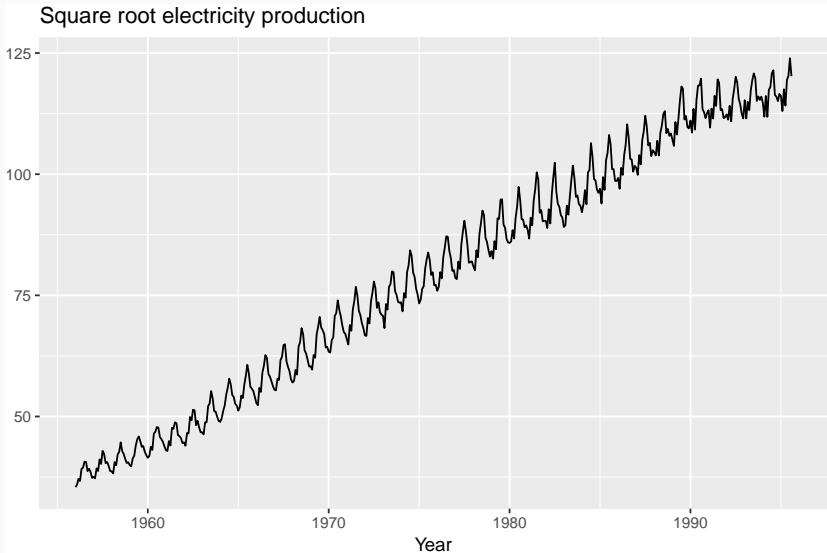
Square root	$w_t = \sqrt{y_t}$	↓
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

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

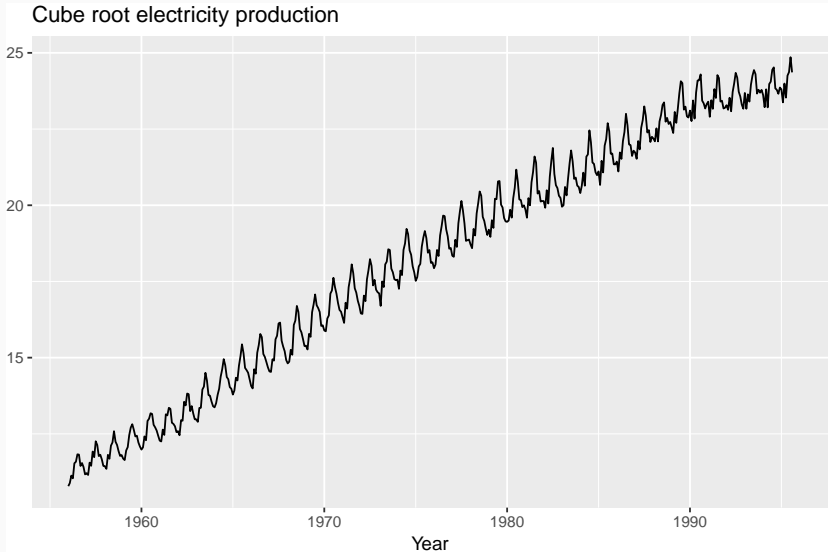


# Variance stabilization



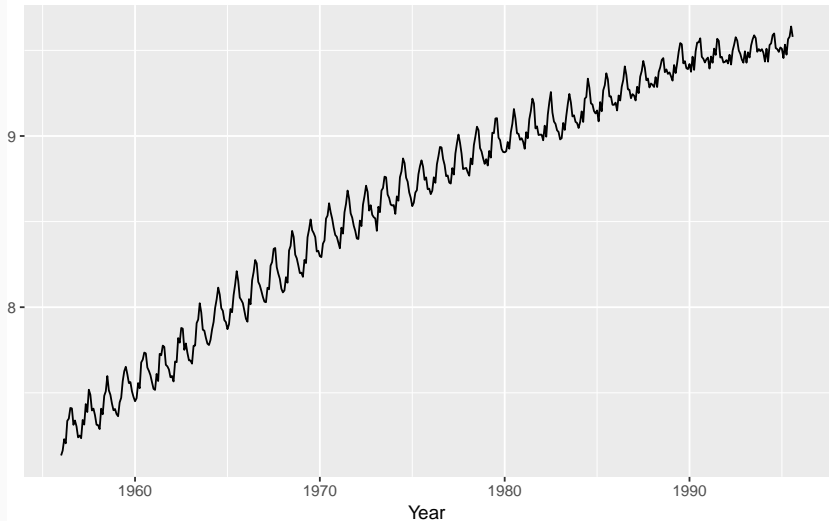


# Variance stabilization

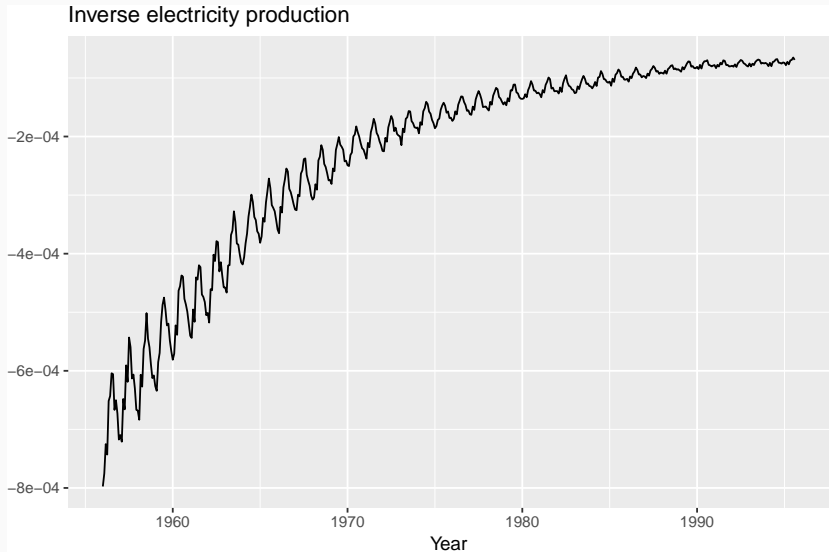


# Variance stabilization

Log electricity production



# Variance stabilization



# Outline

- 1 Variance stabilization
- 2 Box-Cox transformations
- 3 Back-transformation
- 4 Lab session 10

# 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**:

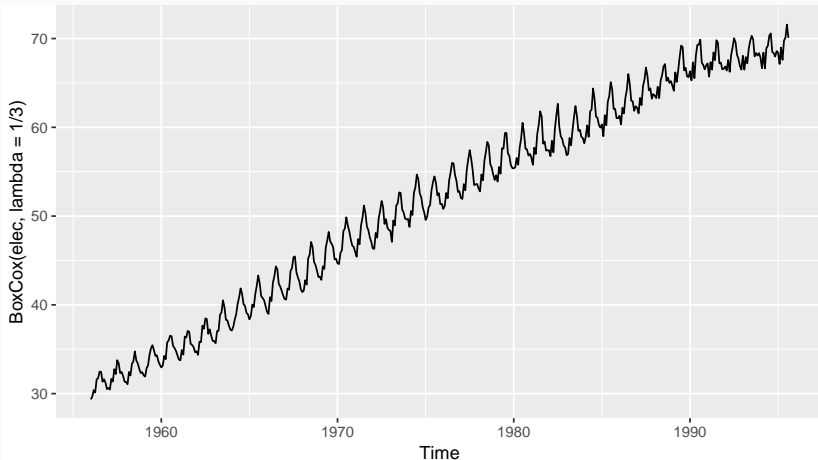
$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

- $\lambda = 1$ : (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))
```





# Box-Cox transformations

- $y_t^\lambda$  for  $\lambda$  close to zero behaves like logs.
- If some  $y_t = 0$ , then must have  $\lambda > 0$
- if some  $y_t < 0$ , no power transformation is possible unless all  $y_t$  adjusted by **adding a constant to all values**.
- Simple values of  $\lambda$  are easier to explain.
- Results are relatively insensitive to  $\lambda$ .
- Often no transformation ( $\lambda = 1$ ) needed.
- Transformation can have very large effect on PI.
- Choosing  $\lambda = 0$  is a simple way to force forecasts to be positive

# 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  $\lambda$  can give extremely large prediction intervals.

# Outline

- 1 Variance stabilization
- 2 Box-Cox transformations
- 3 Back-transformation
- 4 Lab session 10

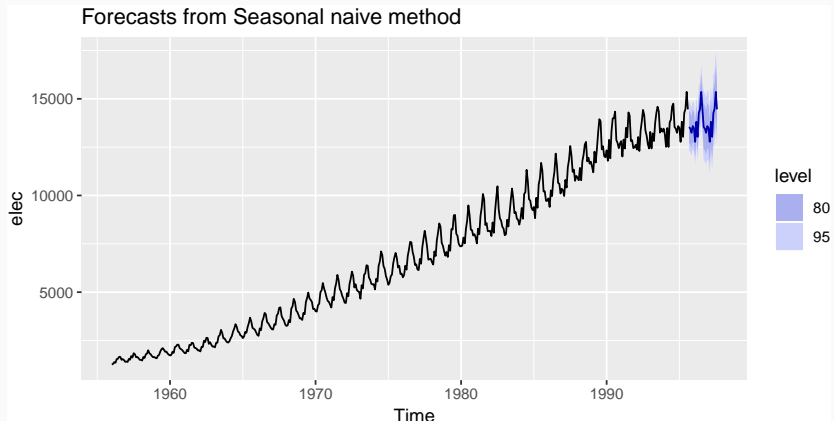
# Back-transformation

We must reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

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

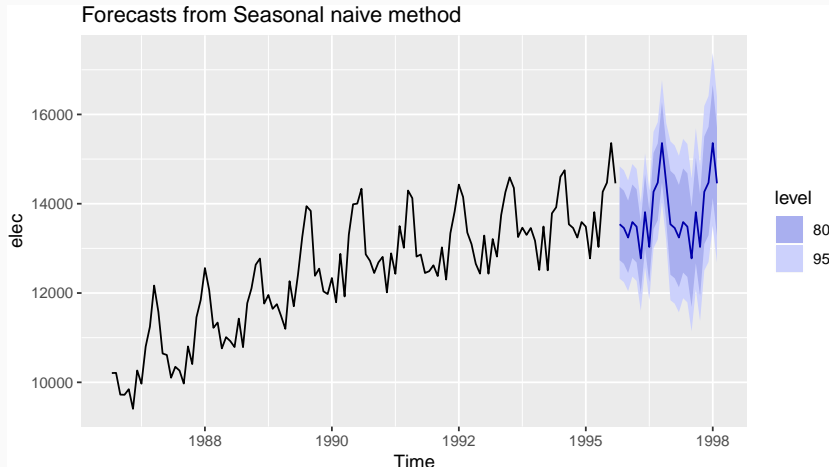
# Back-transformation

```
fit <- snaive(elec, lambda=1/3)  
autoplot(fit)
```



# Back-transformation

```
autoplot(fit, include=120)
```



# Back transformation

- Back-transformed point forecasts are medians.
- Back-transformed PI have the correct coverage.



# Back transformation

- Back-transformed point forecasts are medians.
- Back-transformed PI have the correct coverage.

## Back-transformed means

Let  $X$  be have mean  $\mu$  and variance  $\sigma^2$ .

Let  $f(x)$  be back-transformation function, and  $Y = f(X)$ .

$$E[Y] = E[f(X)] = f(\mu) + \frac{1}{2}\sigma^2[f''(\mu)]^2.$$

# Back transformation

**Box-Cox back-transformation:**

$$y_t = \begin{cases} \exp(w_t) & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f(x) = \begin{cases} e^x & \lambda = 0; \\ (\lambda x + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f''(x) = \begin{cases} e^x & \lambda = 0; \\ (1 - \lambda)(\lambda x + 1)^{1/\lambda - 2} & \lambda \neq 0. \end{cases}$$

# Back transformation

**Box-Cox back-transformation:**

$$y_t = \begin{cases} \exp(w_t) & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f(x) = \begin{cases} e^x & \lambda = 0; \\ (\lambda x + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

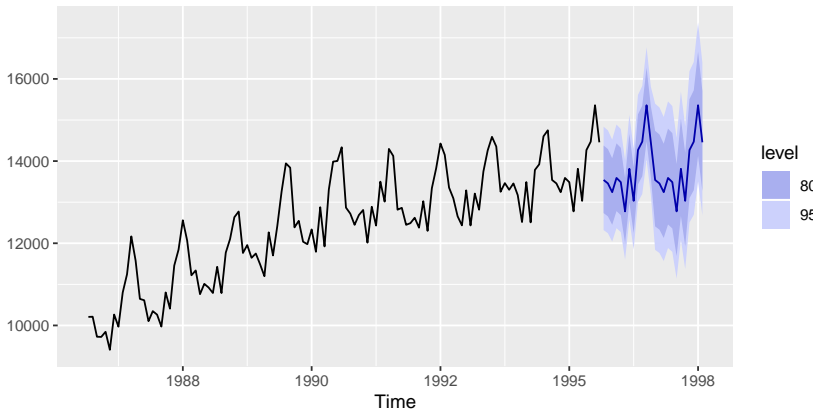
$$f''(x) = \begin{cases} e^x & \lambda = 0; \\ (1 - \lambda)(\lambda x + 1)^{1/\lambda - 2} & \lambda \neq 0. \end{cases}$$

$$E[Y] = \begin{cases} e^{\mu} \left[ 1 + \frac{\sigma^2}{2} \right] & \lambda = 0; \\ (\lambda \mu + 1)^{1/\lambda} \left[ 1 + \frac{\sigma^2(1-\lambda)}{2(\lambda \mu + 1)^2} \right] & \lambda \neq 0. \end{cases}$$

# Back-transformation

```
elec %>% snaive(lambda=1/3, biasadj=FALSE) %>%  
  autoplot(include=120)
```

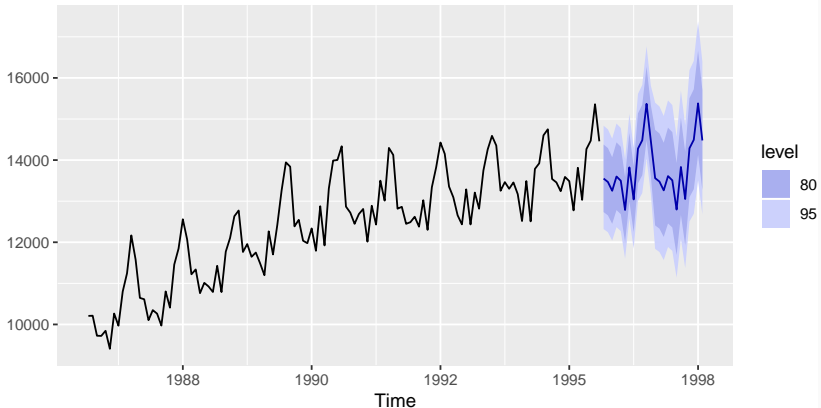
Forecasts from Seasonal naive method



# Back-transformation

```
elec %>% snaive(lambda=1/3, biasadj=TRUE) %>%  
  autoplot(include=120)
```

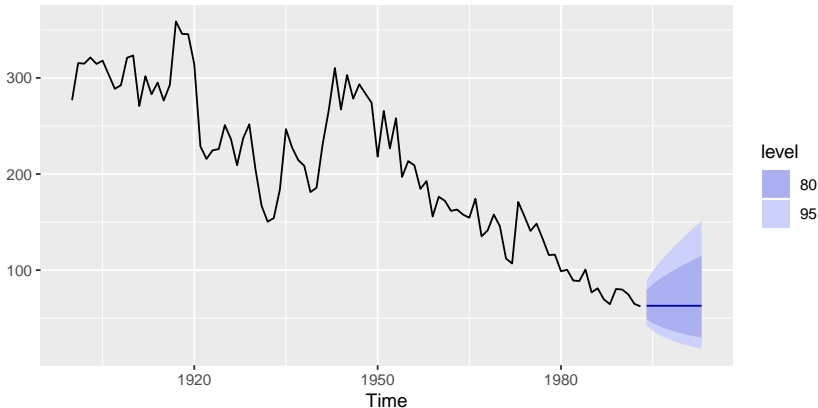
Forecasts from Seasonal naive method



# Back-transformation

```
eggs %>% ses(lambda=1/3, biasadj=FALSE) %>%  
  autoplot
```

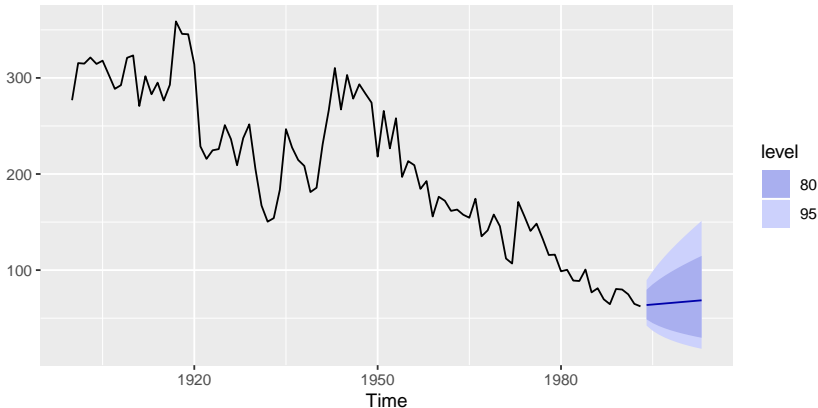
Forecasts from Simple exponential smoothing



# Back-transformation

```
eggs %>% ses(lambda=1/3, biasadj=TRUE) %>%  
  autoplot
```

Forecasts from Simple exponential smoothing



# Outline

- 1 Variance stabilization
- 2 Box-Cox transformations
- 3 Back-transformation
- 4 Lab session 10



# Lab Session 10