MONASH University

MONASH BUSINESS SCHOOL

ETC3550: Applied forecasting for business and economics

Ch3. The forecasters’ toolbox

OTexts.org/fpp2/

Outline

1 **Forecasting**

2 Some simple forecasting methods

3 Box-Cox transformations

4 Forecasting residuals

5 Evaluating forecast accuracy

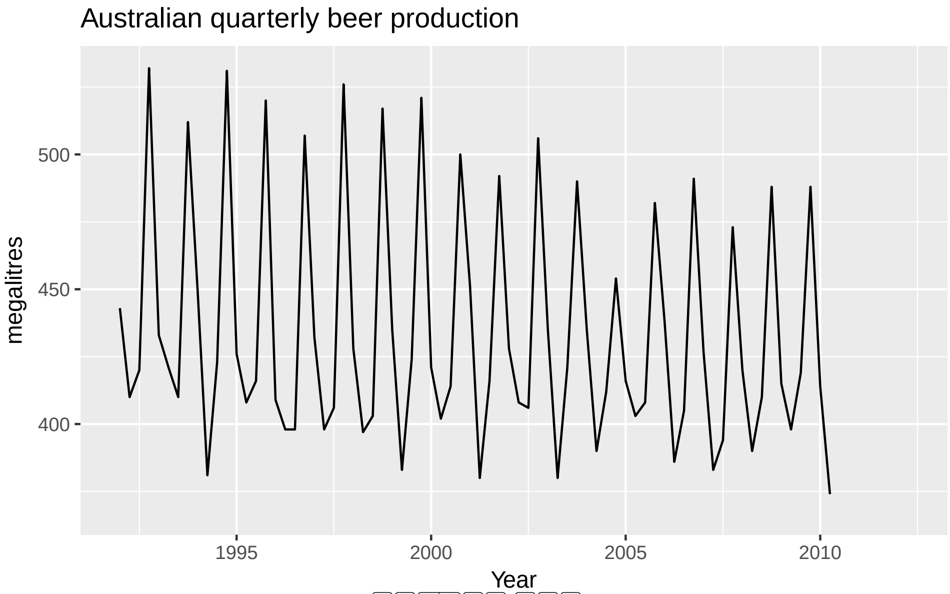
6 The forecast package in R

Forecasting

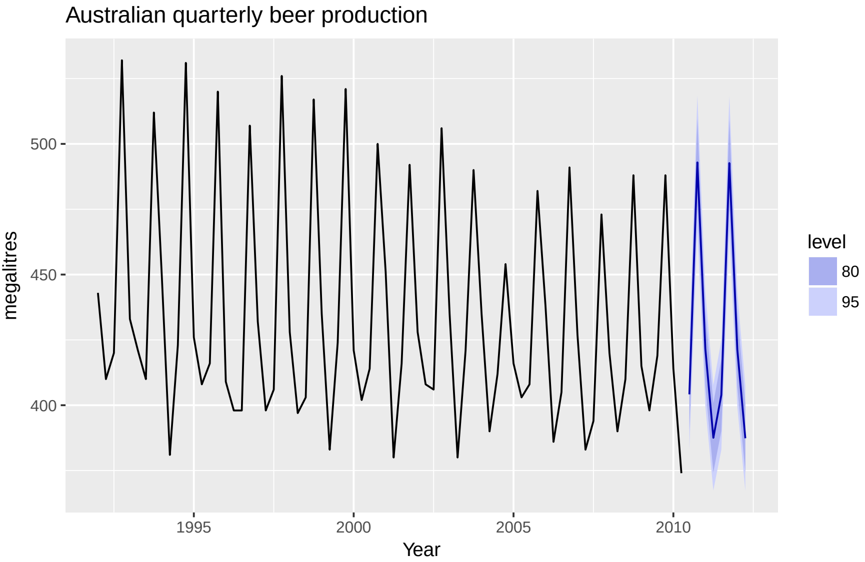
Forecasting is estimating how the sequence of observations will continue into the future.

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

Australian beer production



Australian beer production



Outline

1 Forecasting

2 **Some simple forecasting methods**

3 Box-Cox transformations

4 Forecasting residuals

5 Evaluating forecast accuracy

6 The forecast package in R

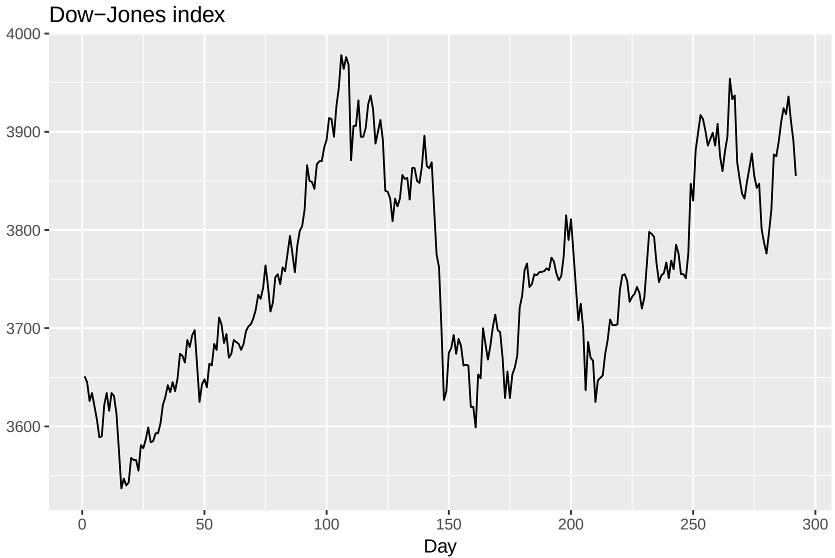
Some simple forecasting methods

How would you forecast these data?

Some simple forecasting methods

How would you forecast these data?

Some simple forecasting methods



How would you forecast these data?

Some simple forecasting methods

Average method

- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$

Naïve method

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

Seasonal naïve method

- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-km}$ where m = seasonal period and $k = \lfloor (h-1)/m \rfloor + 1$.

Some simple forecasting methods

Average method

- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$

Naïve method

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

Seasonal naïve method

- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-km}$ where m = seasonal period and $k = \lfloor (h-1)/m \rfloor + 1$.

Some simple forecasting methods

Average method

- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$

Naïve method

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

Seasonal naïve method

- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-km}$ where m = seasonal period and $k = \lfloor (h-1)/m \rfloor + 1$.

Some simple forecasting methods

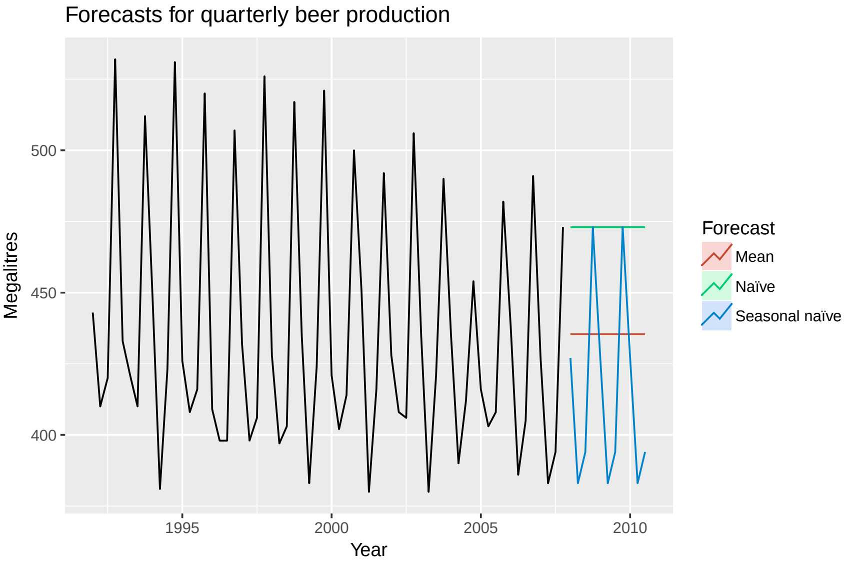
Drift method

- Forecasts equal to last value plus average change.
- Forecasts:

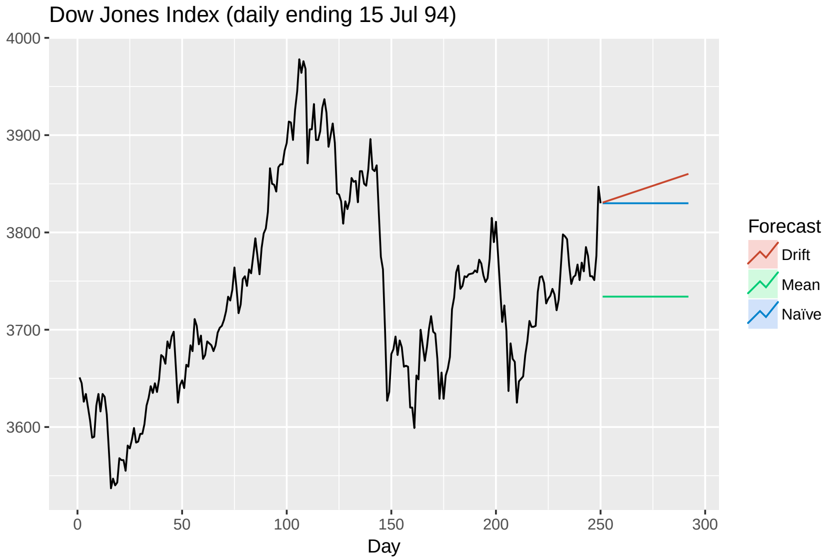
$$\begin{aligned}\hat{y}_{T+h|T} &= y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) \\ &= y_T + \frac{h}{T-1}(y_T - y_1).\end{aligned}$$

- Equivalent to extrapolating a line drawn between first and last observations.

Some simple forecasting methods



Some simple forecasting methods



Some simple forecasting methods

- Mean: `meanf(y, h=20)`
- Naïve: `naive(y, h=20)`
- Seasonal naïve: `snaive(y, h=20)`
- Drift: `rwf(y, drift=TRUE, h=20)`

Outline

1 Forecasting

2 Some simple forecasting methods

3 **Box-Cox transformations**

4 Forecasting residuals

5 Evaluating forecast accuracy

6 The forecast package in R

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

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

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

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

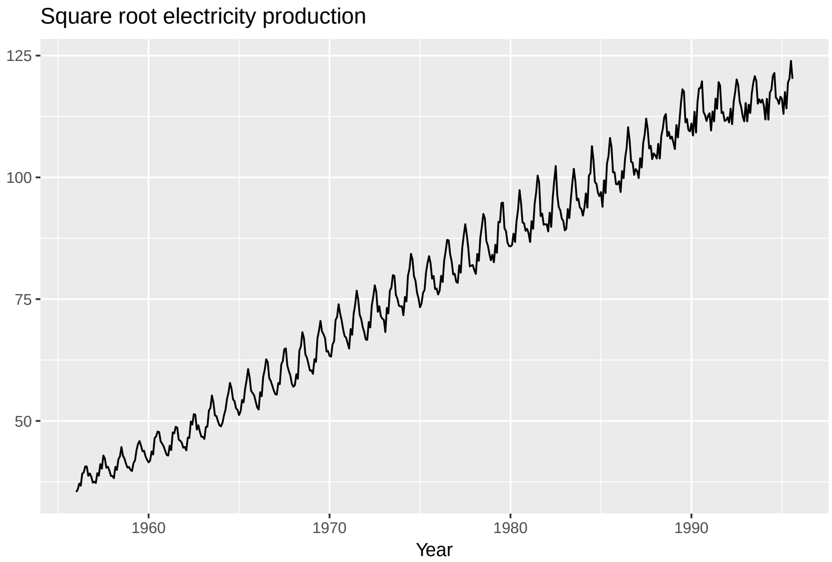
Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

Mathematical transformations for stabilizing variation

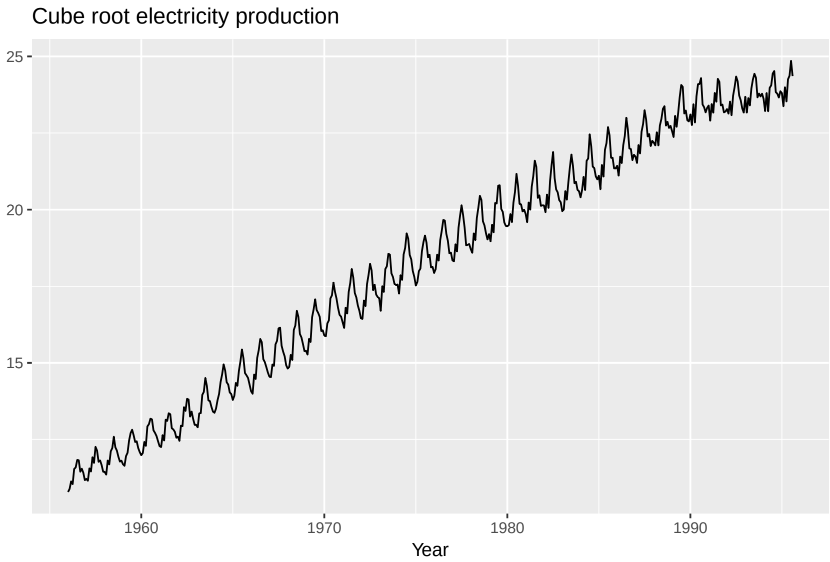
Square root	$w_t = \sqrt{y_t}$	$\downarrow$
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

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

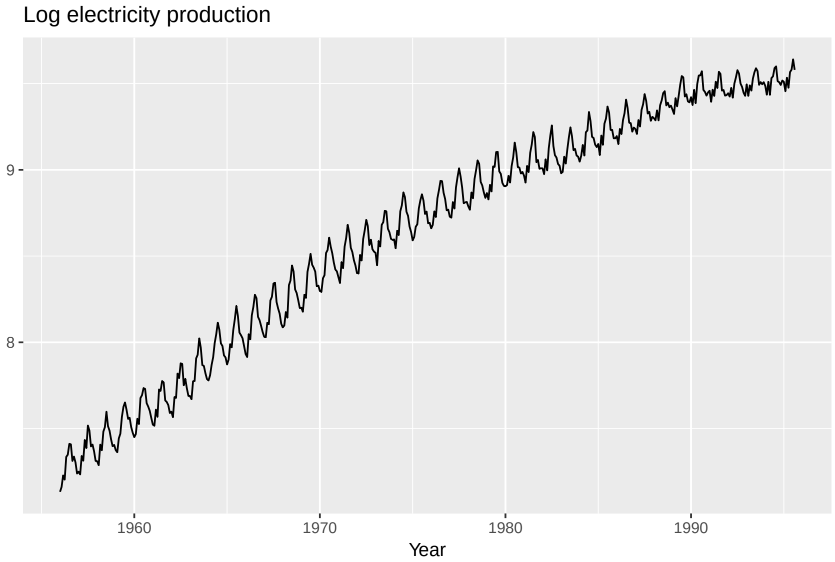
Variance stabilization



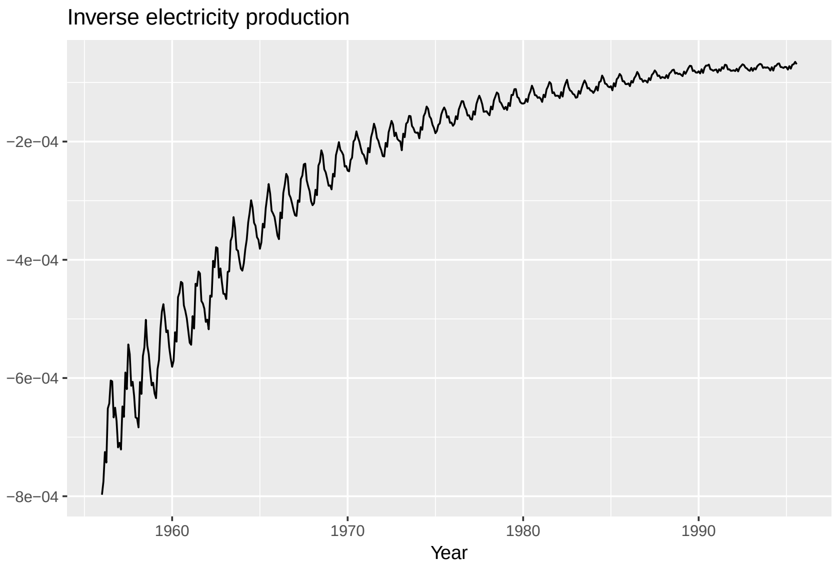
Variance stabilization



Variance stabilization



Variance stabilization



Box-Cox transformations

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

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

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

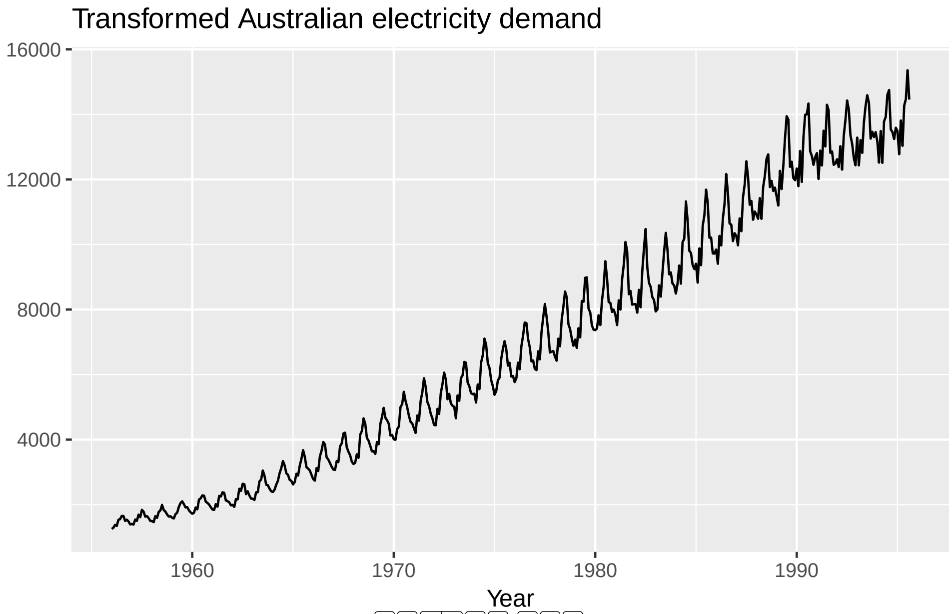
Box-Cox transformations

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

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

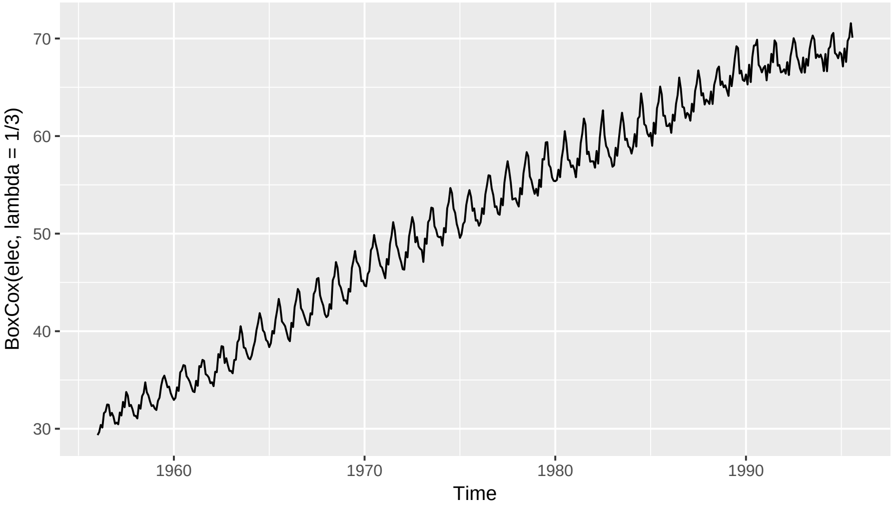
- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Box-Cox transformations



Box-Cox transformations

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



Box-Cox transformations

- y_t^λ for λ 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**.
- Choose a simple value of λ. It makes explanation easier.
- Results are relatively insensitive to value of λ
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive

Automated Box-Cox transformations

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

```
## [1] 0.2654076
```

- 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.

Automated Box-Cox transformations

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

```
## [1] 0.2654076
```

- 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.

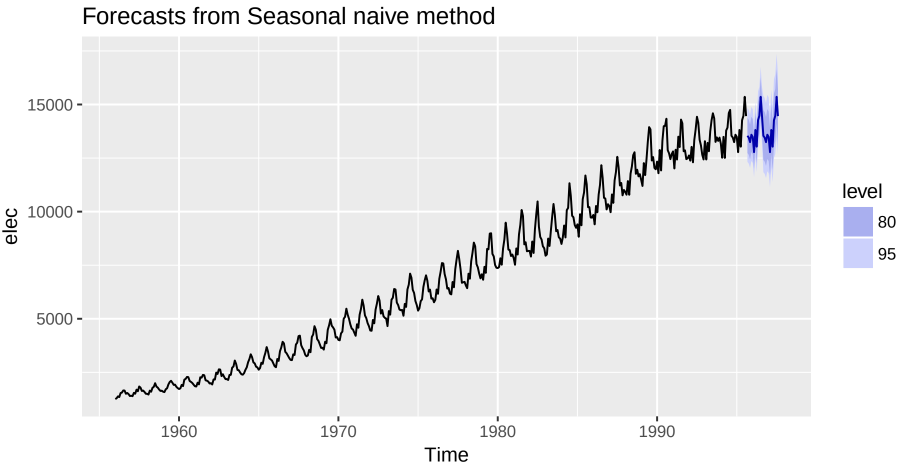
Back-transformation

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

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

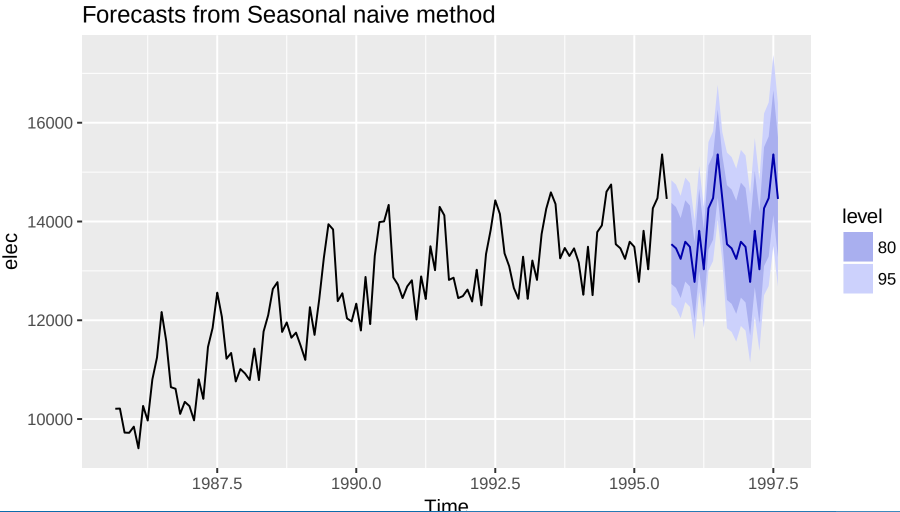
Back-transformation

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



Back-transformation

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



Back transformation

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

Back-transformed means

Let X be have mean μ and variance σ^2.

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

Taylor series expansion about μ:

$$f(X) = f(\mu) + (X - \mu)f'(\mu) + \frac{1}{2}(X - \mu)^2 f''(\mu).$$

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

Back transformation

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

Back-transformed means

Let X be have mean μ and variance σ^2.

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

Taylor series expansion about μ:

$$f(X) = f(\mu) + (X - \mu)f'(\mu) + \frac{1}{2}(X - \mu)^2 f''(\mu).$$

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

Back transformation

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

Back-transformed means

Let X be have mean μ and variance σ^2.

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

Taylor series expansion about μ:

$$f(X) = f(\mu) + (X - \mu)f'(\mu) + \frac{1}{2}(X - \mu)^2 f''(\mu).$$

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

Back transformation

Box-Cox back-transformation:

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

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

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

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

Back transformation

Box-Cox back-transformation:

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

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

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

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

Your turn

Find a Box-Cox transformation that works for the `gas` data.

Outline

1 Forecasting

2 Some simple forecasting methods

3 Box-Cox transformations

4 **Forecasting residuals**

5 Evaluating forecast accuracy

6 The forecast package in R

Fitted values

- $\hat{y}_{t|t-1}$ is the forecast of y_t based on observations $y_1, \dots, y_t$.
- We call these "fitted values".
- Sometimes drop the subscript: $\hat{y}_t \equiv \hat{y}_{t|t-1}$.
- Often not true forecasts since parameters are estimated on all data.

For example:

- $\hat{y}_t = \bar{y}$ for average method.
- $\hat{y}_t = y_{t-1} + (y_T - y_1)/(T-1)$ for drift method.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Assumptions

1. $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
2. $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Useful properties (for prediction intervals)

3. $\{e_t\}$ have constant variance.
4. $\{e_t\}$ are normally distributed.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Assumptions

1. $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
2. $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Useful properties (for prediction intervals)

3. $\{e_t\}$ have constant variance.
4. $\{e_t\}$ are normally distributed.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

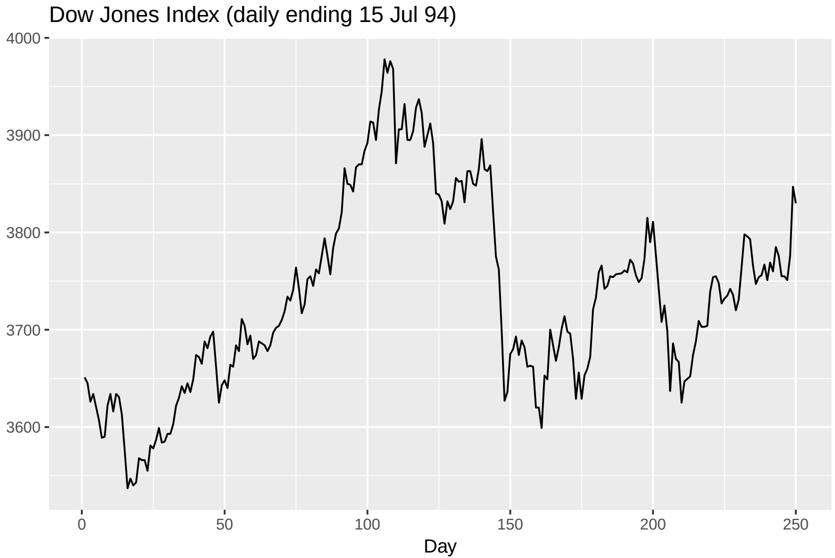
Assumptions

1. $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
2. $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Useful properties (for prediction intervals)

3. $\{e_t\}$ have constant variance.
4. $\{e_t\}$ are normally distributed.

Example: Dow-Jones index



Example: Dow-Jones index

Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

$$e_t = y_t - y_{t-1}$$

Note: e_t are one-step-forecast residuals

Example: Dow-Jones index

Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

$$e_t = y_t - y_{t-1}$$

Note: e_t are one-step-forecast residuals

Example: Dow-Jones index

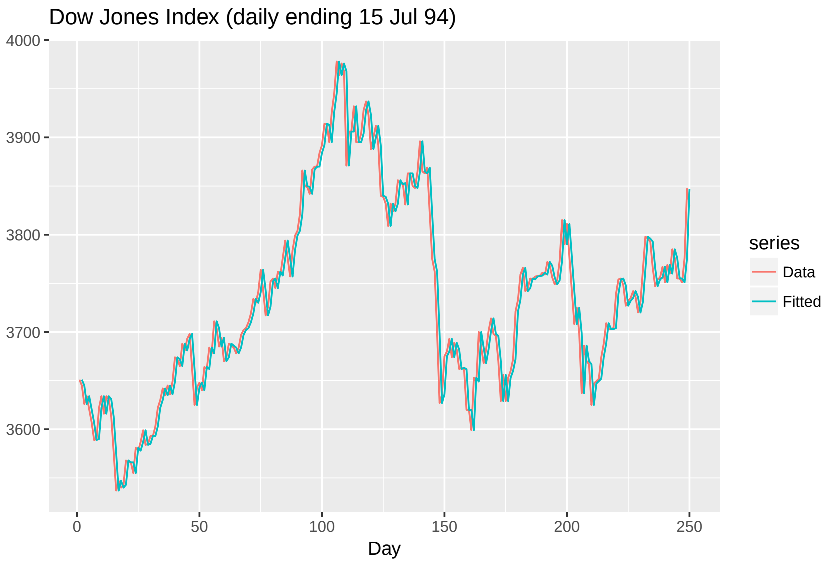
Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

$$e_t = y_t - y_{t-1}$$

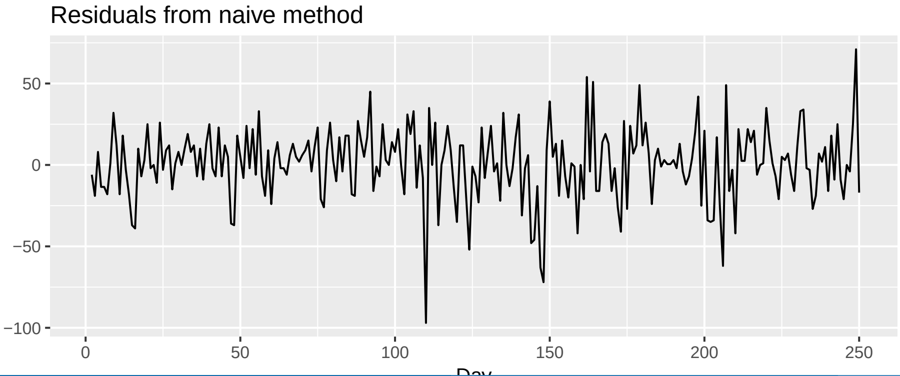
Note: e_t are one-step-forecast residuals

Example: Dow-Jones index



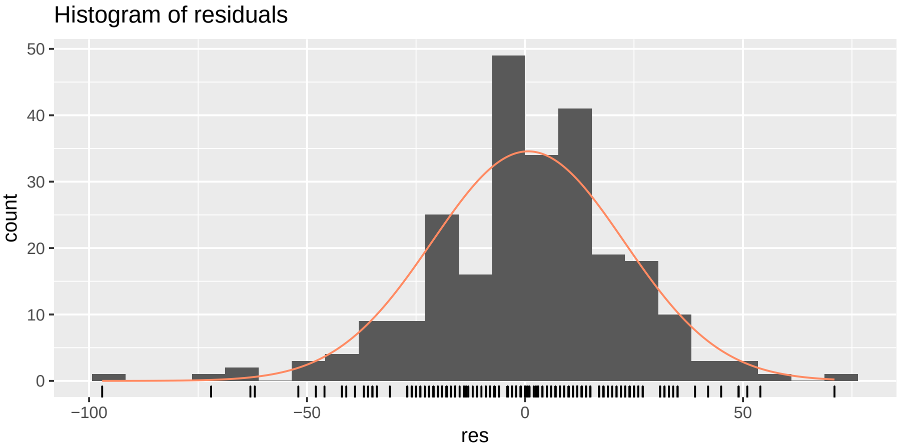
Example: Dow-Jones index

```r
dj2 <- window(dj, end=250)
res <- residuals(naive(dj2))
autoplot(res) + xlab("Day") + ylab("") +
  ggtitle("Residuals from naive method")
```



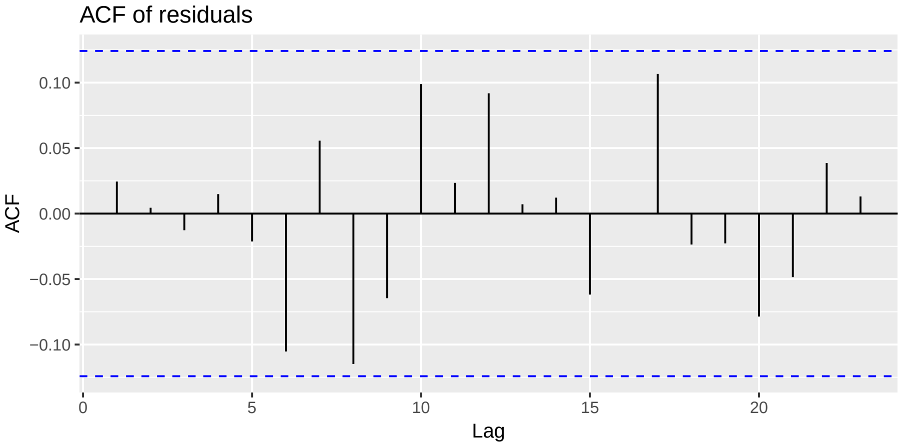
Example: Dow-Jones index

```r
gghistogram(res, add.normal=TRUE) +
  ggtitle("Histogram of residuals")
```



Example: Dow-Jones index

```r
ggAcf(res) + ggtitle("ACF of residuals")
```



ACF of residuals

- We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren't, then there is information left in the residuals that should be used in computing forecasts.
- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.
- We *expect* these to look like white noise.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Box-Pierce test

$$Q = T\sum_{k=1}^{h} r_k^2$$

where h is max lag being considered and T is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each r_k close to zero, Q will be **small**.
- If some r_k values large (positive or negative), Q will be **large**.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Box-Pierce test

$$Q = T\sum_{k=1}^{h} r_k^2$$

where h is max lag being considered and T is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each r_k close to zero, Q will be **small**.
- If some r_k values large (positive or negative), Q will be **large**.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Ljung-Box test

$$Q^* = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k^2$$

where h is max lag being considered and T is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- Better performance, especially in small samples.

Portmanteau tests

- If data are WN, Q^* has χ^2 distribution with $(h - K)$ degrees of freedom where K = no. parameters in model.
- When applied to raw data, set $K = 0$.
- For the Dow-Jones example,

```
# lag=h and fitdf=K
Box.test(res, lag=10, fitdf=0)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 10.655, df = 10, p-value =
## 0.385
```

Portmanteau tests

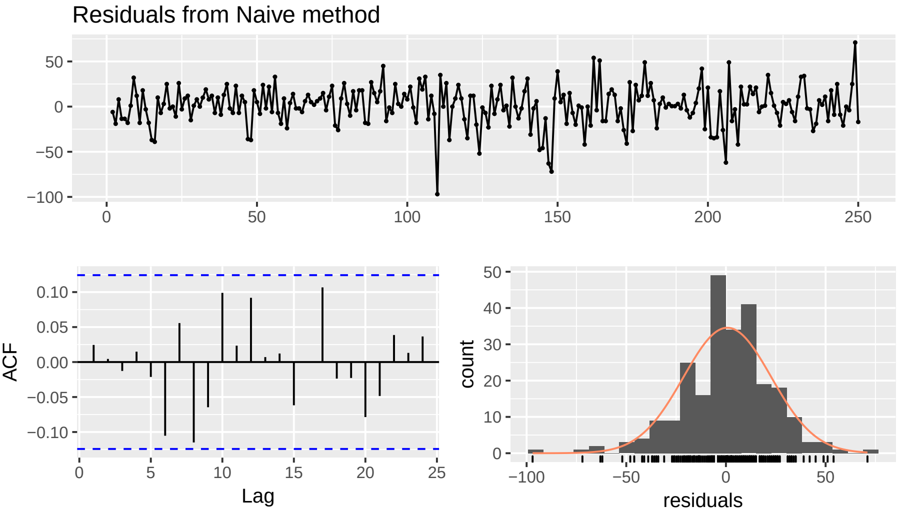
- If data are WN, Q^* has χ^2 distribution with $(h - K)$ degrees of freedom where K = no. parameters in model.
- When applied to raw data, set $K = 0$.
- For the Dow-Jones example,

```
# lag=h and fitdf=K
Box.test(res, lag=10, fitdf=0, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 11.088, df = 10, p-value =
## 0.3507
```

`checkresiduals` function

```
checkresiduals(naive(dj2))
```



`checkresiduals` function

```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 11.088, df = 10, p-value = 0.3507

## Model df: 0.   Total lags used: 10
```

Your turn

Compute seasonal naïve forecasts for quarterly Australian beer production from 1992.

```
beer <- window(ausbeer, start=1992)
fc <- snaive(beer)
autoplot(fc)
```

Test if the residuals are white noise.

```
checkresiduals(fc)
```

What do you conclude?

Outline

1 Forecasting

2 Some simple forecasting methods

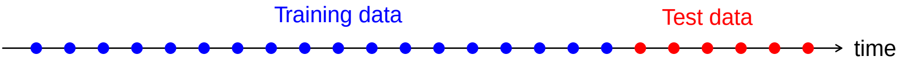
3 Box-Cox transformations

4 Forecasting residuals

5 **Evaluating forecast accuracy**

6 The forecast package in R

Training and test sets



- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data.
- The test set must not be used for *any* aspect of model development or calculation of forecasts.
- Forecast accuracy is based only on the test set.

Forecast errors

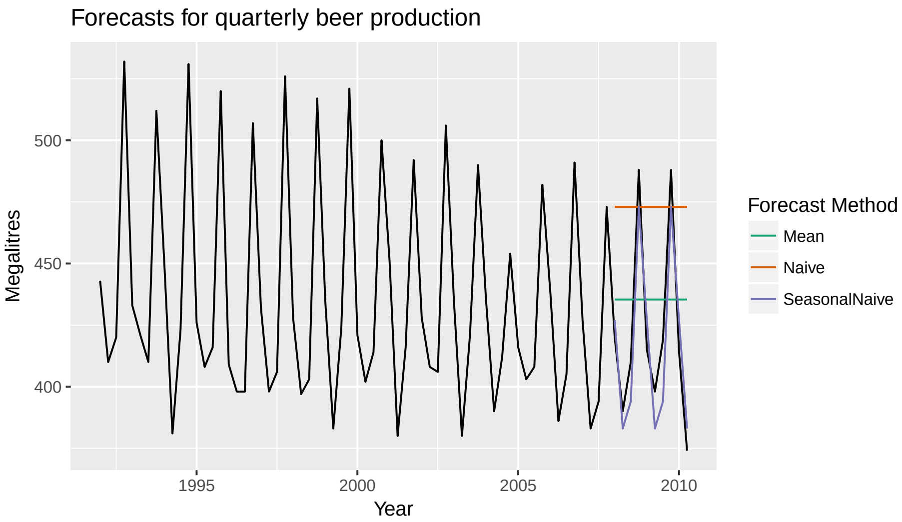
Forecast “error”: the difference between an observed value and its forecast.

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by $\{y_1, \dots, y_T\}$

- Unlike residuals, forecast errors on the test set involve multi-step forecasts.
- These are *true* forecast errors as the test data is not used in computing $\hat{y}_{T+h|T}$.

Measures of forecast accuracy



Measures of forecast accuracy

Let y_t denote the tth observation and $\hat{y}_{t|t-1}$ denote its forecast based on all previous data, where $t = 1, \dots, T$. Then the following measures are useful.

$$\text{MAE} = T^{-1}\sum_{t=1}^{T}|y_t - \hat{y}_{t|t-1}|$$

$$\text{MSE} = T^{-1}\sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2 \qquad \text{RMSE} = \sqrt{T^{-1}\sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2}$$

$$\text{MAPE} = 100T^{-1}\sum_{t=1}^{T}|y_t - \hat{y}_{t|t-1}|/|y_t|$$

- MAE, MSE, RMSE are all scale dependent.
- MAPE is scale independent but is only sensible if $y_t \gg 0$ for all t, and y has a natural zero.

Measures of forecast accuracy

Let y_t denote the tth observation and $\hat{y}_{t|t-1}$ denote its forecast based on all previous data, where $t = 1, \dots, T$. Then the following measures are useful.

$$\text{MAE} = T^{-1}\sum_{t=1}^{T}|y_t - \hat{y}_{t|t-1}|$$

$$\text{MSE} = T^{-1}\sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2 \qquad \text{RMSE} = \sqrt{T^{-1}\sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2}$$

$$\text{MAPE} = 100T^{-1}\sum_{t=1}^{T}|y_t - \hat{y}_{t|t-1}|/|y_t|$$

- MAE, MSE, RMSE are all scale dependent.
- MAPE is scale independent but is only sensible if $y_t \gg 0$ for all t, and y has a natural zero.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = T^{-1}\sum_{t=1}^{T}|y_t - \hat{y}_{t|t-1}|/Q$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

Proposed by Hyndman and Koehler (IJF, 2006).

For non-seasonal time series,

$$Q = (T-1)^{-1}\sum_{t=2}^{T}|y_t - y_{t-1}|$$

works well. Then MASE is equivalent to MAE relative to a naïve method.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = T^{-1} \sum_{t=1}^{T} |y_t - \hat{y}_{t|t-1}|/Q$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

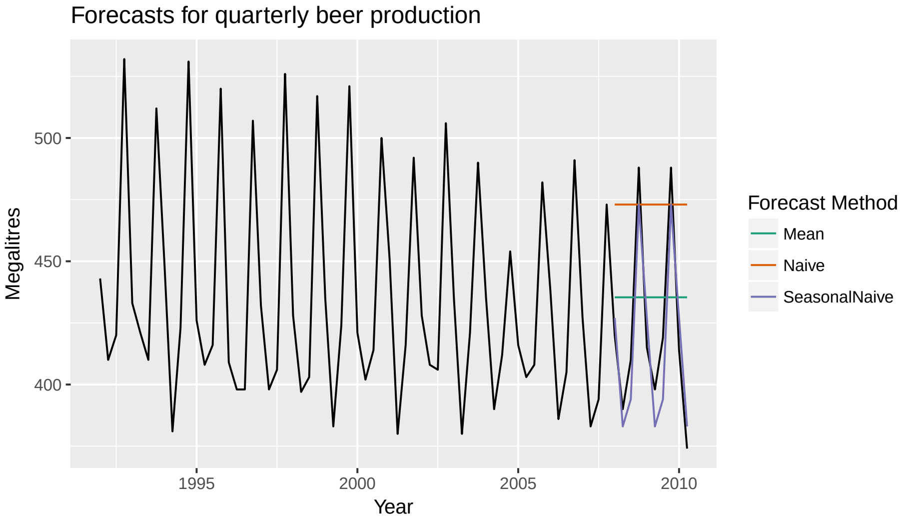
Proposed by Hyndman and Koehler (IJF, 2006).

For seasonal time series,

$$Q = (T - m)^{-1} \sum_{t=m+1}^{T} |y_t - y_{t-m}|$$

works well. Then MASE is equivalent to MAE relative to a seasonal naïve method.

Measures of forecast accuracy



Measures of forecast accuracy

```r
beer2 <- window(ausbeer, start=1992, end=c(2007,4))
beer3 <- window(ausbeer, start=2008)
beerfit1 <- meanf(beer2, h=10)
beerfit2 <- rwf(beer2, h=10)
beerfit3 <- snaive(beer2, h=10)
accuracy(beerfit1, beer3)
accuracy(beerfit2, beer3)
accuracy(beerfit3, beer3)
```

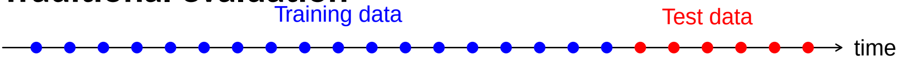
	RMSE	MAE	MAPE	MASE
Mean method	38.45	34.83	8.28	2.44
Naïve method	62.69	57.40	14.18	4.01
Seasonal naïve method	14.31	13.40	3.17	0.94

Poll: true or false?

1. Good forecast methods should have normally distributed residuals.
2. A model with small residuals will give good forecasts.
3. The best measure of forecast accuracy is MAPE.
4. If your model doesn't forecast well, you should make it more complicated.
5. Always choose the model with the best forecast accuracy as measured on the test set.

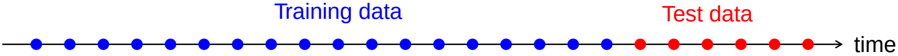
Time series cross-validation

Traditional evaluation

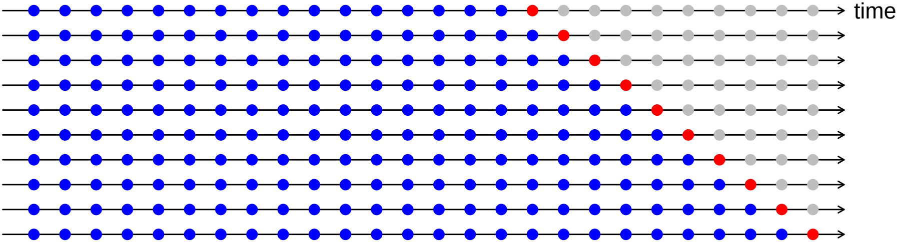


Time series cross-validation

Traditional evaluation



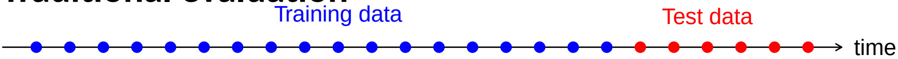
Time series cross-validation



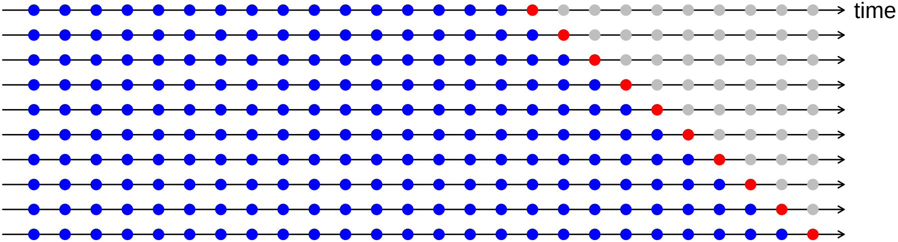
- Forecast accuracy averaged over test sets.
- Also known as “evaluation on a rolling forecasting origin”

Time series cross-validation

Traditional evaluation



Time series cross-validation



- Forecast accuracy averaged over test sets.
- Also known as “evaluation on a rolling forecasting origin”

tsCV function:

```
e <- tsCV(dj, rwf, drift=TRUE, h=1)
sqrt(mean(e^2, na.rm=TRUE))
```

```
## [1] 22.68249
```

```
sqrt(mean(residuals(rwf(dj, drift=TRUE))^2, na.rm=TRUE))
```

```
## [1] 22.49681
```

A good way to choose the best forecasting model is to find the model with the smallest RMSE computed using time series cross-validation.

Pipe function

Ugly code:

```
e <- tsCV(dj, rwf, drift=TRUE, h=1)
sqrt(mean(e^2, na.rm=TRUE))
sqrt(mean(residuals(rwf(dj, drift=TRUE))^2, na.rm=TRUE))
```

Better with a pipe:

```
dj %>% tsCV(forecastfunction=rwf, drift=TRUE, h=1) -> e
e^2 %>% mean(na.rm=TRUE) %>% sqrt
dj %>% rwf(drift=TRUE) %>% residuals -> res
res^2 %>% mean(na.rm=TRUE) %>% sqrt
```

Prediction intervals

- A forecast $\hat{y}_t$ is (usually) the mean of the conditional distribution $y_t \mid y_1, \dots, y_{t-1}$.
- A prediction interval gives a region within which we expect y_t to lie with a specified probability.
- Assuming $\{e_t\}$ are iid N(0,σ^2), then a simple 95% PI *for the first forecast* is

$$\hat{y}_t \pm 1.96\hat{\sigma}$$

where $\hat{\sigma}$ is the st dev of e_t.

Prediction intervals

Naive forecast with prediction interval:

```
djsd <- sqrt(mean(res^2, na.rm=TRUE))
djf <- tail(dj,1)
upper <- c(djf) + 1.96 * djsd
lower <- c(djf) - 1.96 * djsd
c(lower,upper)
```

```
## [1] 3811.88 3898.12
```

```
naive(dj, level=95)
```

```
##     Point Forecast    Lo 95    Hi 95
## 293           3855 3810.886 3899.114
## 294           3855 3792.613 3917.387
## 295           3855 3778.592 3931.408
## 296           3855 3766.771 3943.229
```

Prediction intervals

- Point forecasts are often useless without prediction intervals.
- Prediction intervals require a stochastic model (with random errors, etc).
- Multi-step forecasts for time series require a more sophisticated approach (with PI getting wider as the forecast horizon increases).

Prediction intervals

- Computed automatically using: `naive()`, `snaive()`, `rwf()`, `meanf()`, etc.
- Use `level` argument to control coverage.
- Check residual assumptions before believing them.
- Usually too narrow due to unaccounted uncertainty.

Outline

1 Forecasting

2 Some simple forecasting methods

3 Box-Cox transformations

4 Forecasting residuals

5 Evaluating forecast accuracy

6 The forecast package in R

The forecast package in R

Functions that output a forecast object:

- `meanf`, `naive`, `snaive`, `rwf`

forecast class contains

- original series
- point forecasts
- prediction interval(s)
- forecasting method used
- residuals and fitted values

Functions designed to work with forecast objects:

- `autoplot`, `summary`, `print`.

The forecast package in R

`forecast()` function

- Takes a time series or a time series model as first argument.
- If first argument is of class `ts`, it returns forecasts from automatic ETS algorithm.
- Returns object of class `forecast`.

```
forecast(ausbeer, level=90)
```

```
##         Point Forecast    Lo 90    Hi 90
## 2010 Q3       404.5681 380.9711 428.1652
## 2010 Q4       480.3658 451.5609 509.1706
## 2011 Q1       417.0146 391.1700 442.8592
## 2011 Q2       383.0830 358.4230 407.7430
## 2011 Q3       402.9795 374.2342 431.7249
```

The forecast package in R

`forecast()` function

- Takes a time series or a time series model as first argument.
- If first argument is of class `ts`, it returns forecasts from automatic ETS algorithm.
- Returns object of class `forecast`.

```
forecast(ausbeer, level=90)
```

```
##         Point Forecast    Lo 90    Hi 90
## 2010 Q3       404.5681 380.9711 428.1652
## 2010 Q4       480.3658 451.5609 509.1706
## 2011 Q1       417.0146 391.1700 442.8592
## 2011 Q2       383.0830 358.4230 407.7430
## 2011 Q3       402.9795 374.2342 431.7249
```