MONASH University

MONASH BUSINESS SCHOOL

Time Series and Forecasting

Benchmark forecasting and time series decomposition

Outline

1 Some simple forecasting methods

2 Forecast residuals

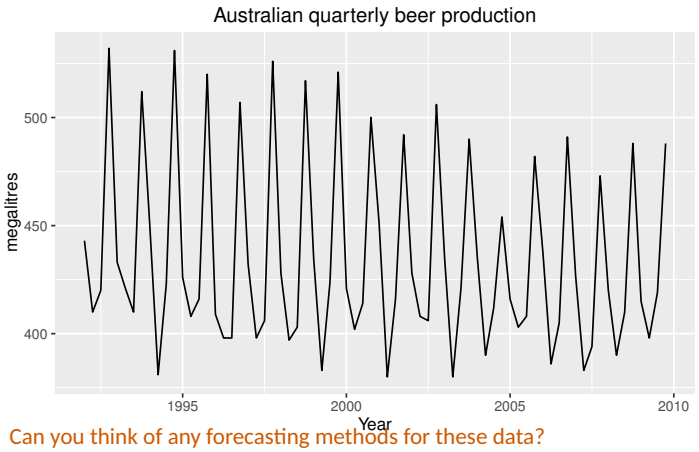
3 Evaluating forecast accuracy

4 Transformations

5 STL decomposition

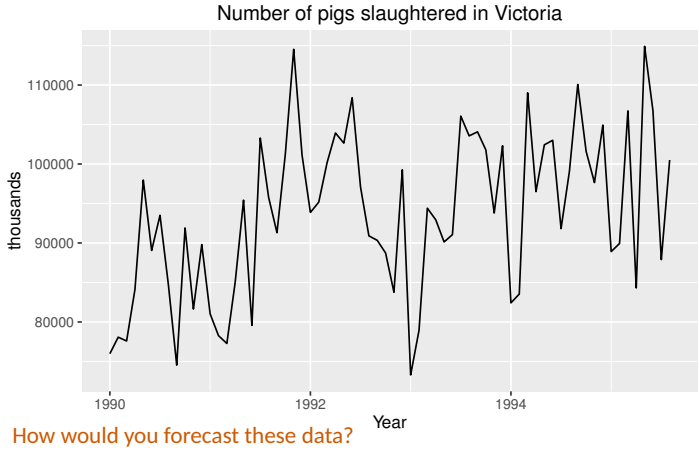
6 Forecasting and decomposition

Some simple forecasting methods



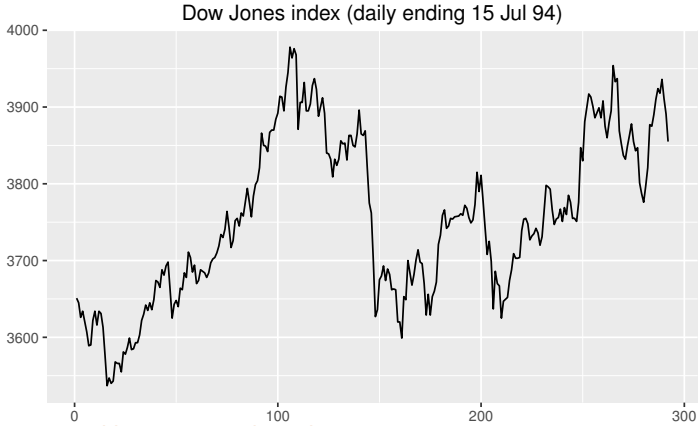
Can you think of any forecasting methods for these data?

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: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$
- Optimal for white noise.

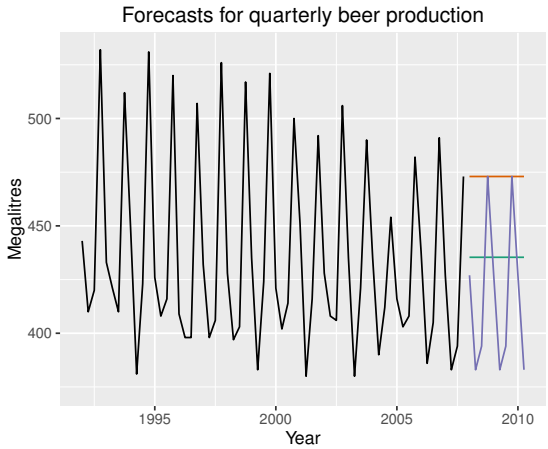
Naïve method

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Optimal for random walks
- 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



Which method is which?

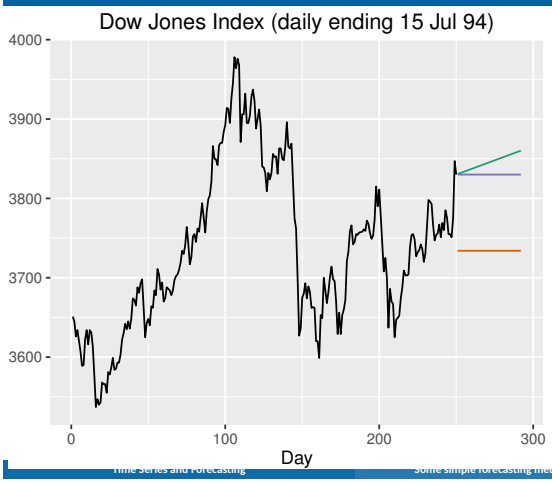
Drift method

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

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

- Optimal for random walks with drift
- Equivalent to extrapolating a line drawn between first and last observations.

Some simple forecasting methods



Which method is which?

Some simple forecasting methods

- Mean: `meanf(x, h=20)`
- Naive: `naive(x, h=20)` or `rwf(x, h=20)`
- Seasonal naive: `snaive(x, h=20)`
- Drift: `rwf(x, drift=TRUE, h=20)`

Outline

1 Some simple forecasting methods
2 Forecast residuals
3 Evaluating forecast accuracy
4 Transformations
5 STL decomposition
6 Forecasting and decomposition

Forecasting residuals

Residuals in forecasting: difference between observed value and its forecast based on all previous observations: $e_t = y_t - \hat{y}_{t|t-1}$.

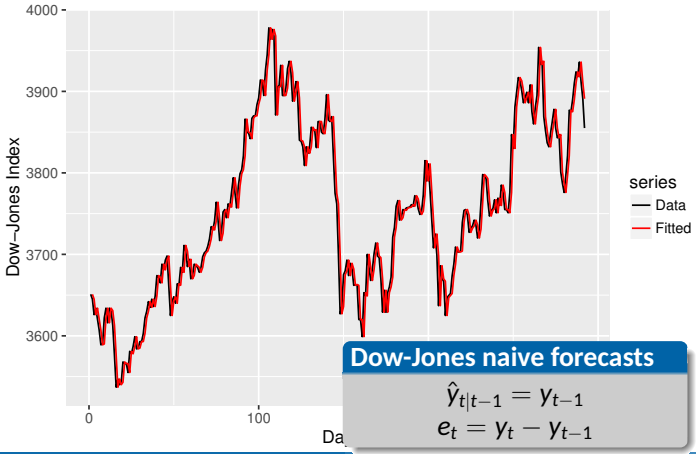
Assumptions

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.

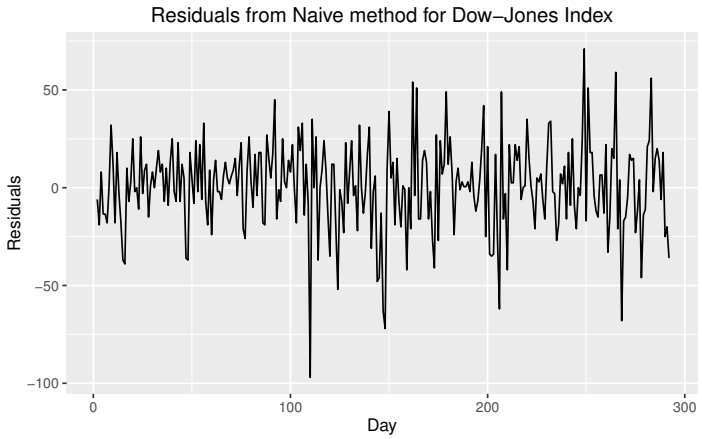
Forecasting Dow-Jones index



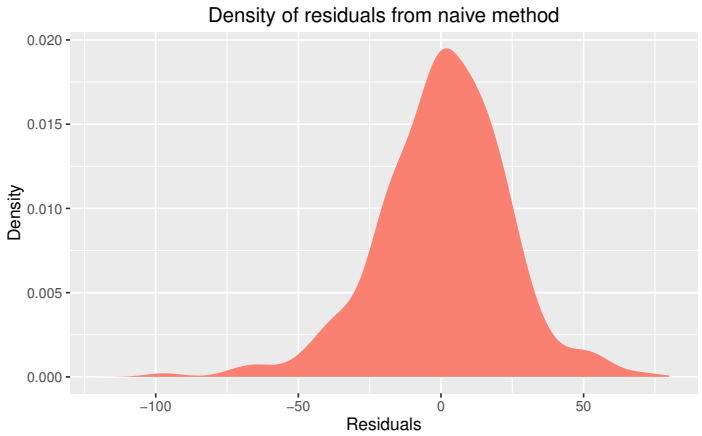
Dow-Jones naive forecasts

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

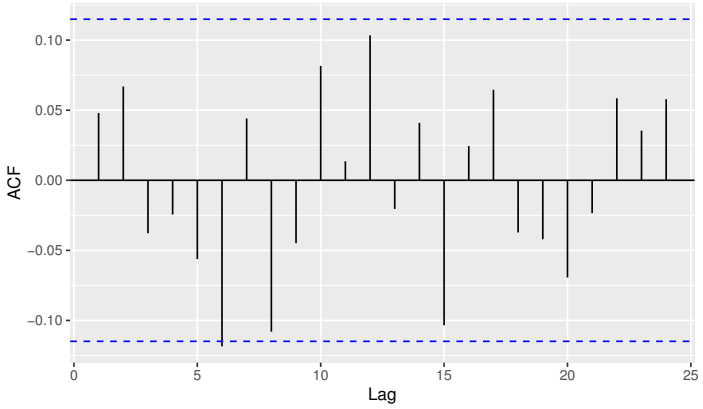
Forecasting Dow-Jones index



Forecasting Dow-Jones index



Forecasting Dow-Jones index



Forecasting Dow-Jones index

```
fc <- rwf(dj)

res <- residuals(fc)

autoplot(res)

ggplot(res, aes(res)) +
  geom_density(fill = "salmon", bw = "SJ",
  colour = NA)

ggAcf(res,main="")
```


Exercise

1. Calculate the residuals from a seasonal naive forecast applied to the quarterly Australian beer production data from 1992.
2. Test if the residuals are white noise.

Exercise

1. Calculate the residuals from a seasonal naive forecast applied to the quarterly Australian beer production data from 1992.
2. Test if the residuals are white noise.

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


Outline

1. Some simple forecasting methods
2. Forecast residuals
3. **Evaluating forecast accuracy**
4. Transformations
5. STL decomposition
6. Forecasting and decomposition

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 $t = 1, \ldots, T$. 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 $\{y_t\}$.

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 naive method.

Measures of forecast accuracy

Mean Absolute Scaled Error

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

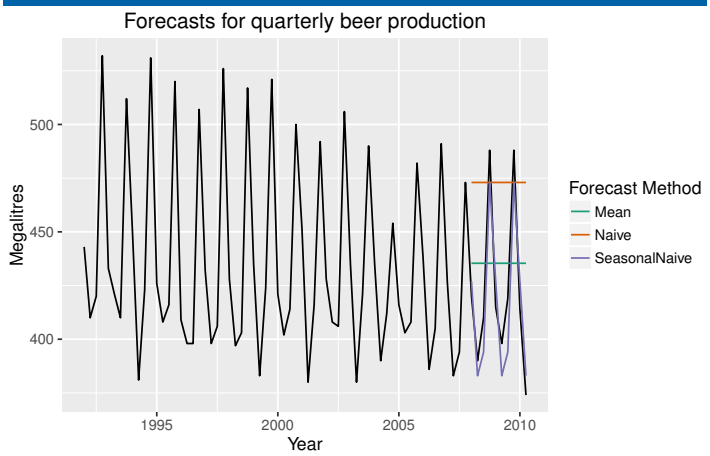
where Q is a stable measure of the scale of the time series $\{y_t\}$.

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 naive method.

Measures of forecast accuracy



Measures of forecast accuracy

Mean method

RMSE	MAE	MAPE	MASE
38.45	34.83	8.28	2.44

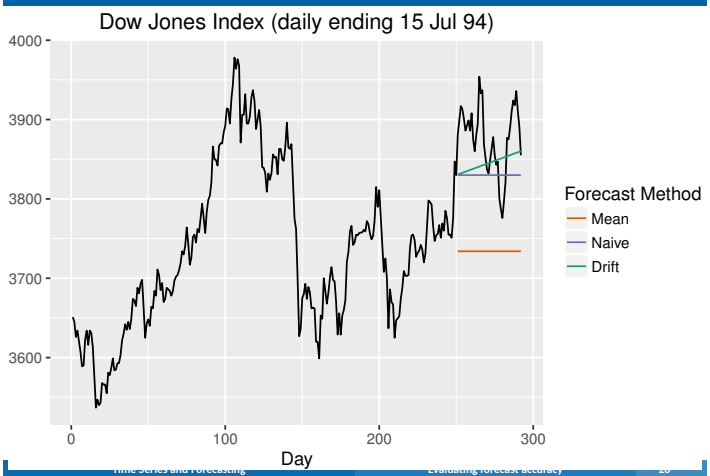
Naïve method

RMSE	MAE	MAPE	MASE
62.69	57.40	14.18	4.01

Seasonal naïve method

RMSE	MAE	MAPE	MASE
14.31	13.40	3.17	0.94

Measures of forecast accuracy



Measures of forecast accuracy

Mean method

RMSE	MAE	MAPE	MASE
148.24	142.42	3.66	8.70

Naïve method

RMSE	MAE	MAPE	MASE
62.03	54.44	1.40	3.32

Drift model

RMSE	MAE	MAPE	MASE
53.70	45.73	1.18	2.79

Training and test sets

Available data

Training set (e.g., 80%)	Test set (e.g., 20%)

- 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.

Training and test sets

```
beer3 <- window(ausbeer,start=1992,end=2005.99)
beer4 <- window(ausbeer,start=2006)

fit1 <- meanf(beer3,h=20)
fit2 <- rwf(beer3,h=20)

accuracy(fit1,beer4)
accuracy(fit2,beer4)
```

In-sample accuracy (one-step forecasts)

```
accuracy(fit1)
accuracy(fit2)
```


Beware of over-fitting

- A model which fits the data well does not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters. (Compare R^2)
- Over-fitting a model to data is as bad as failing to identify the systematic pattern in the data.
- Problems can be overcome by measuring true *out-of-sample* forecast accuracy. That is, total data divided into "training" set and "test" set. Training set used to estimate parameters. Forecasts are made for test set.
- Accuracy measures computed for errors in test set only.

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.

Outline

1. Some simple forecasting methods
2. Forecast residuals
3. Evaluating forecast accuracy
4. **Transformations**
5. STL decomposition
6. Forecasting and decomposition

Transformations to stabilize the variance

If the data show different variation at different levels of the series, then a transformation can be useful.
Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

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.**

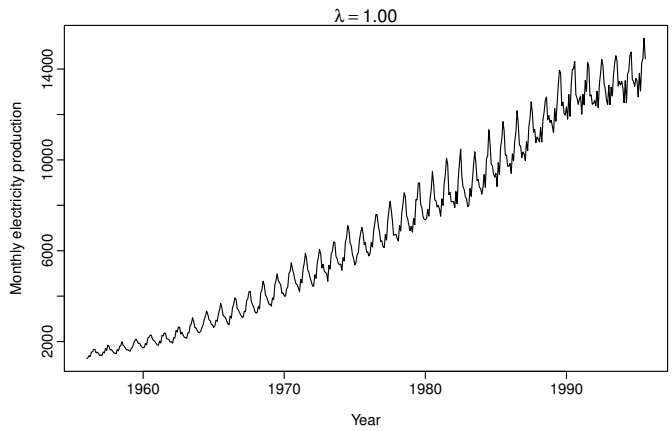
Box-Cox transformations

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

- 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

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

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

```
autoplot(BoxCox(elec,lambda=1/3))
fit <- snaive(elec, lambda=1/3)
autoplot(fit)
autoplot(fit, include=120)
```


Automated Box-Cox transformations

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

- 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

1 Some simple forecasting methods

2 Forecast residuals

3 Evaluating forecast accuracy

4 Transformations

5 STL decomposition

6 Forecasting and decomposition

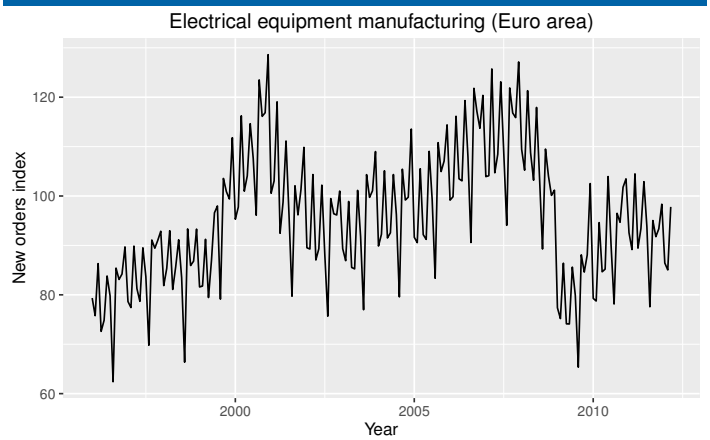
Time series decomposition

$$Y_t = f(S_t, T_t, E_t)$$

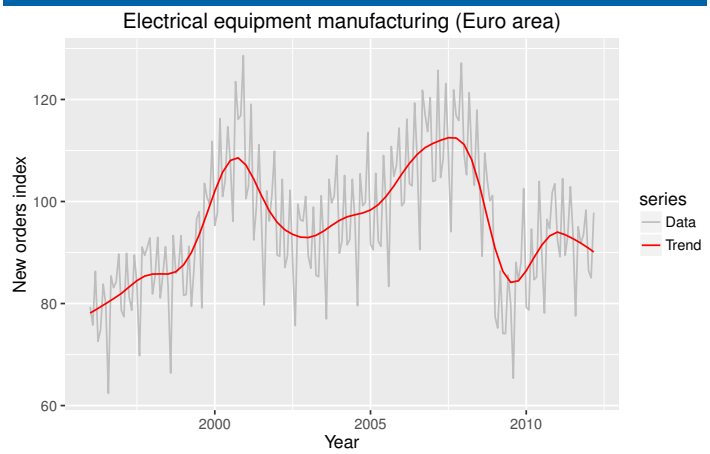
where $Y_t =$ data at period t
$S_t =$ seasonal component at period t
$T_t =$ trend component at period t
$E_t =$ remainder (or irregular or error) component at period t

Additive decomposition: $Y_t = S_t + T_t + E_t$.

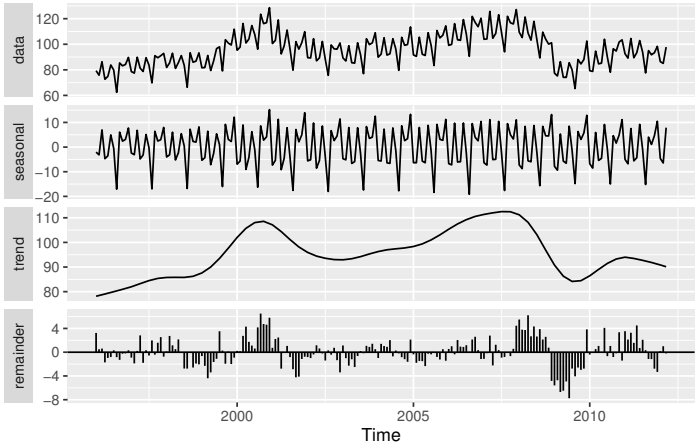
Euro electrical equipment



Euro electrical equipment

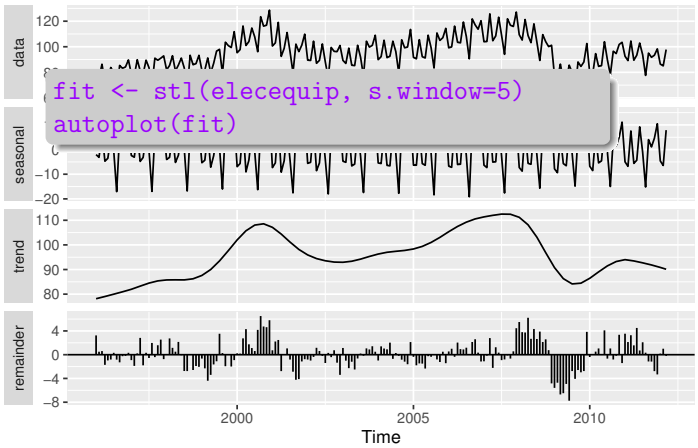


STL decomposition

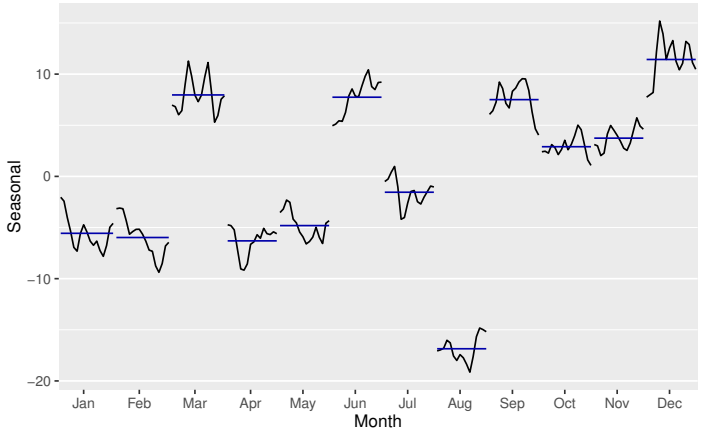


STL decomposition

```
fit <- stl(elecequip, s.window=5)
autoplot(fit)
```



Euro electrical equipment

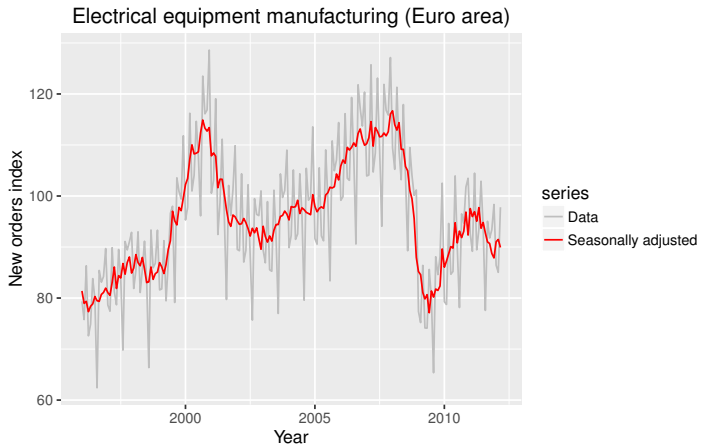


Seasonal adjustment

- Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.
- Additive decomposition: seasonally adjusted data given by

$$Y_t - S_t = T_t + E_t.$$

Euro electrical equipment

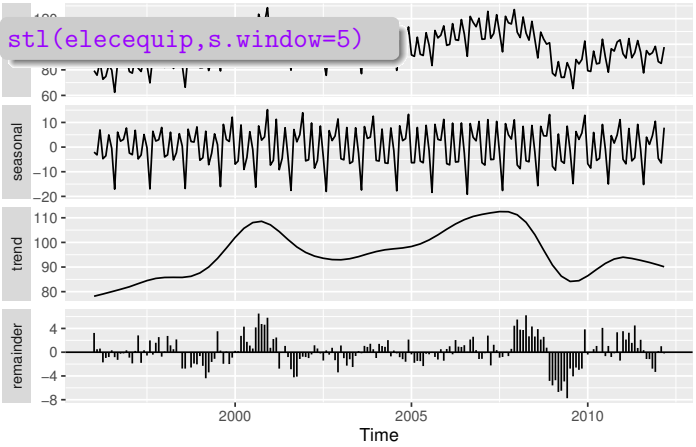


STL decomposition

- 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.
- Robust to outliers
- Only additive.
- Use Box-Cox transformations to get other decompositions.

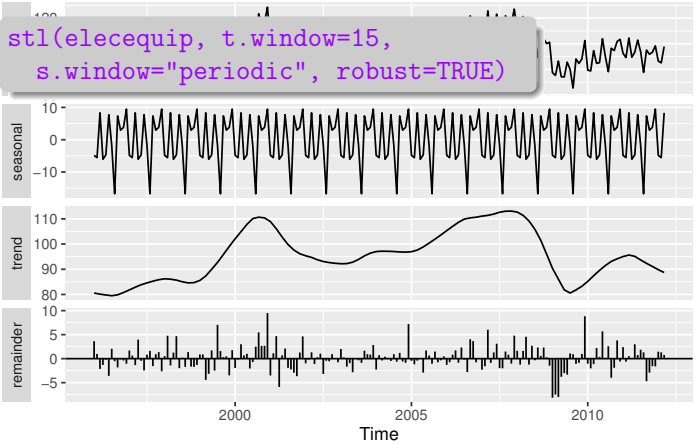
STL decomposition

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



STL decomposition

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



STL decomposition in R

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

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

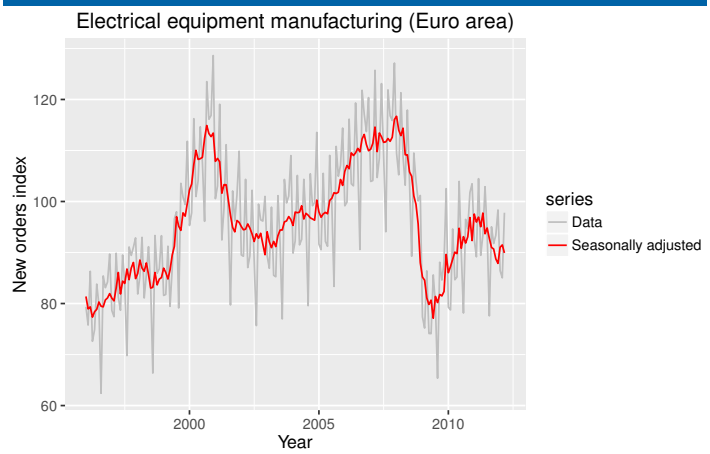
Outline

1 Some simple forecasting methods
2 Forecast residuals
3 Evaluating forecast accuracy
4 Transformations
5 STL decomposition
6 **Forecasting and decomposition**

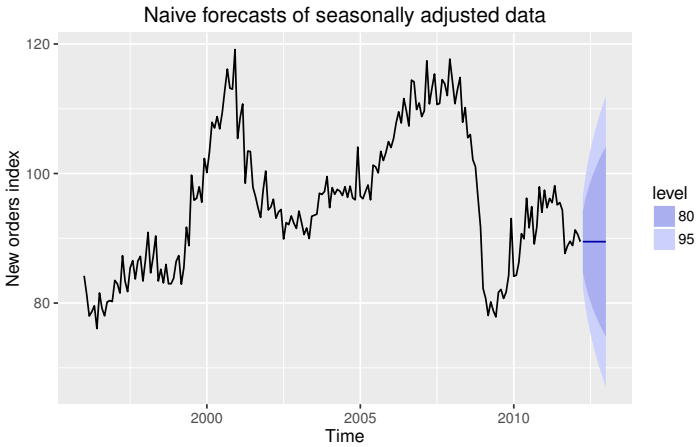
Forecasting and decomposition

- 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.

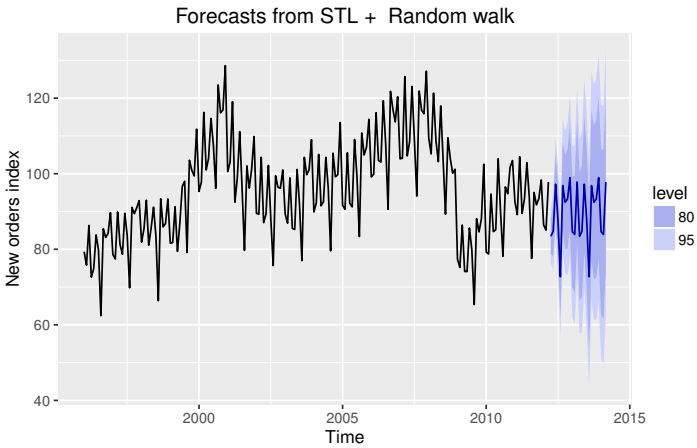
Seas adj elec equipment



Seas adj elec equipment



Seas adj elec equipment



How to do this in R

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

eeadj <- seasadj(fit)
autoplot(naive(eeadj), xlab="New orders index")

fcast <- forecast(fit, method="naive")
autoplot(fcast, ylab="New orders index")
```


Decomposition and prediction intervals

- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.