MONASH University

Rob J Hyndman

Forecasting:

Principles and Practice

5. Time series decomposition and cross-validation

OTexts.org/fpp/6/

OTexts.org/fpp/2/5/

Outline

1 STL decomposition

2 Forecasting and decomposition

3 Cross-validation

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: $Y_t = S_t + T_t + E_t$.

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: $Y_t = S_t + T_t + E_t.$

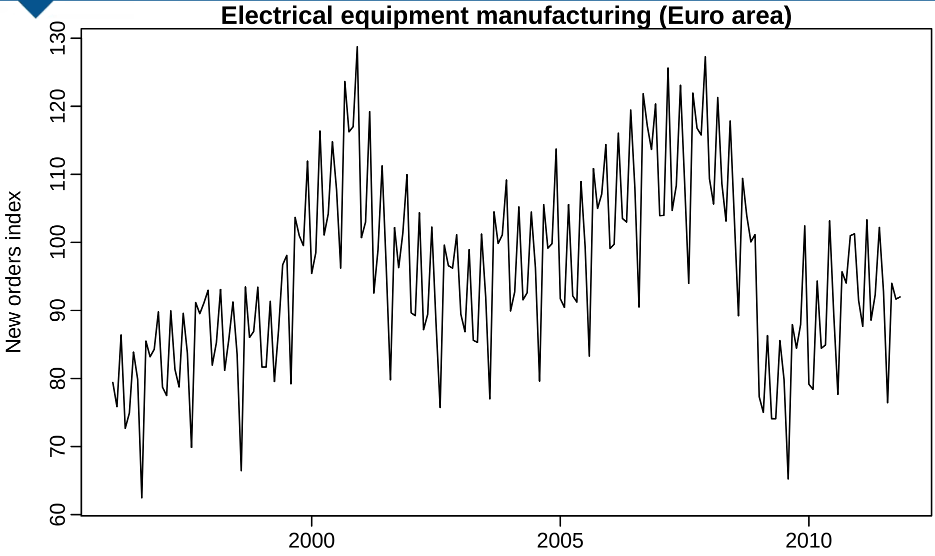
Time series decomposition

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

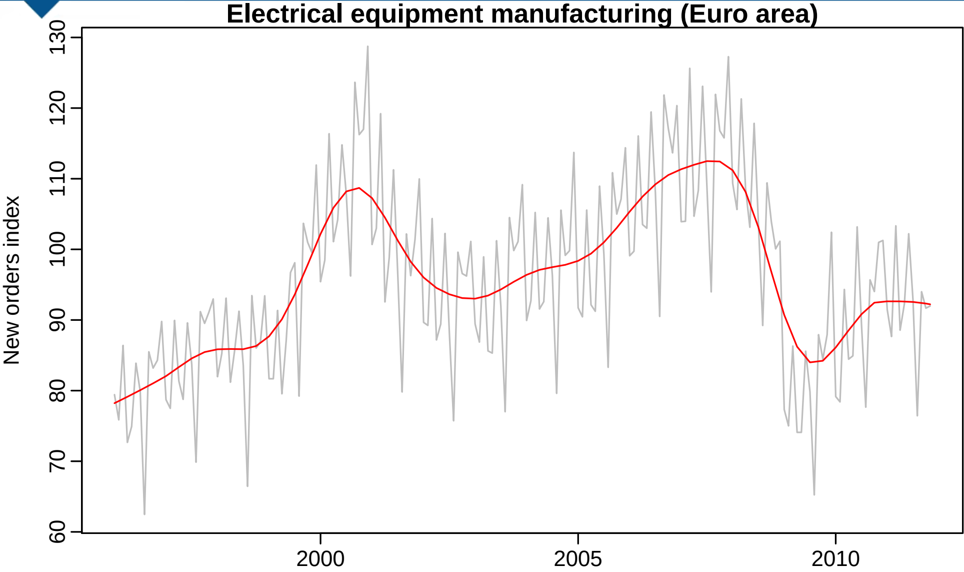
where
$Y_t =$ data at period t
$S_t =$ seasonal component at period t
$T_t =$ trend component at period t
$E_t =$ remainder (or irregular or error) component at period t

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

Euro electrical equipment

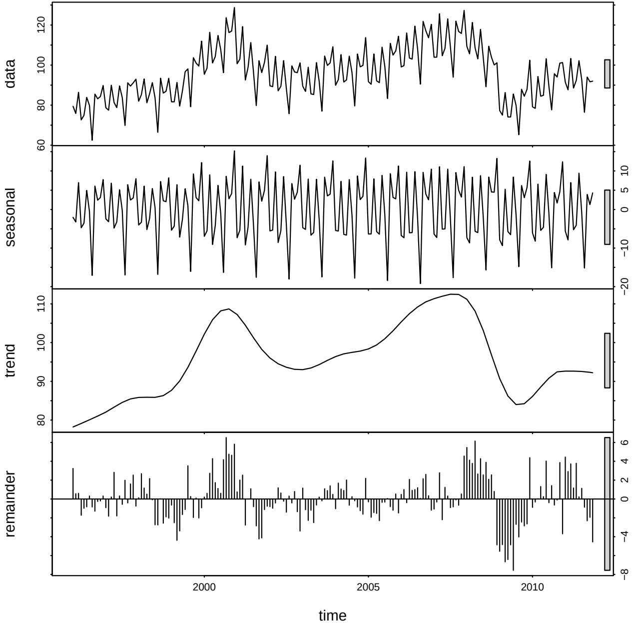


Euro electrical equipment

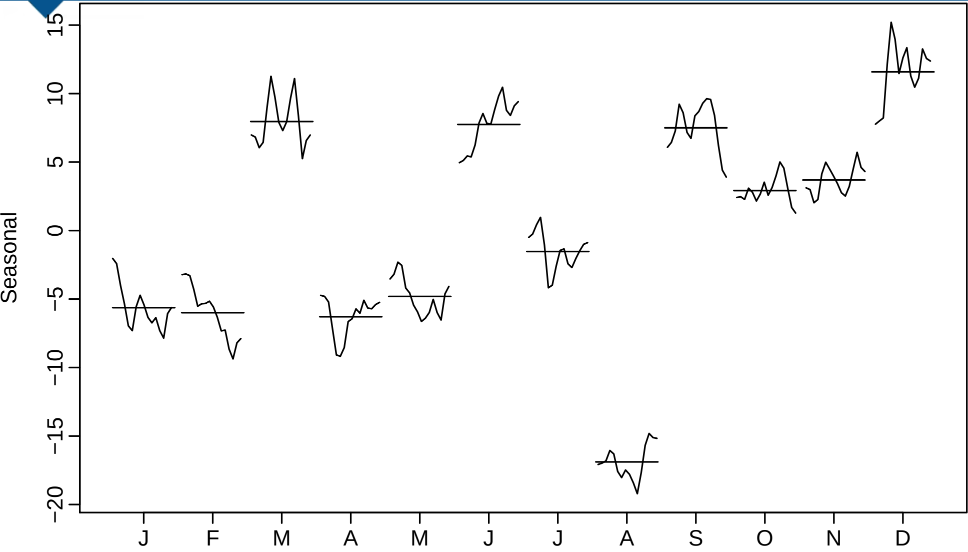


Euro electrical equipment

STL decomposition



Euro electrical equipment



Seasonal sub-series plot of the seasonal component

Seasonal adjustment

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

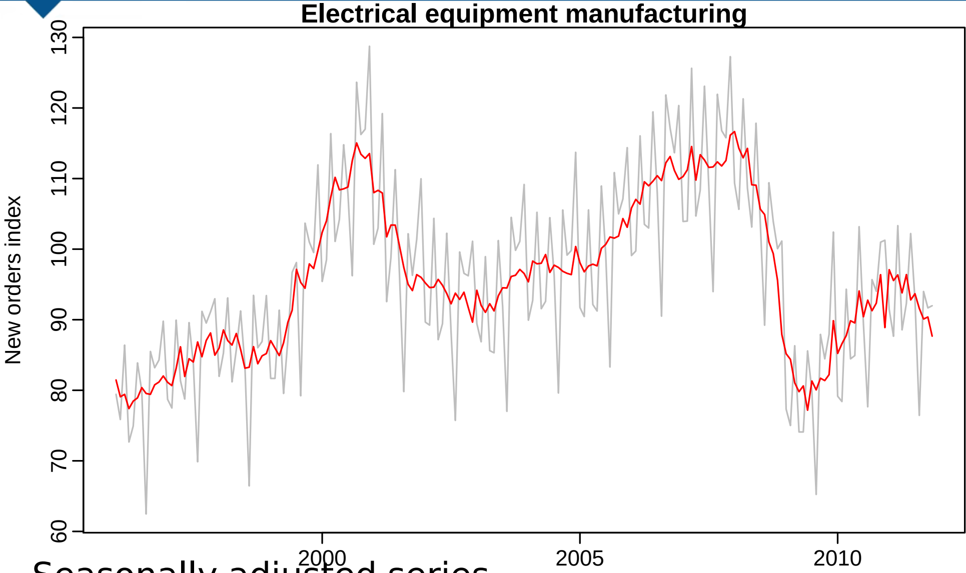
$$Y_t - S_t = T_t + E_t$$

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



Seasonally adjusted series

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: “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: "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: “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: "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.

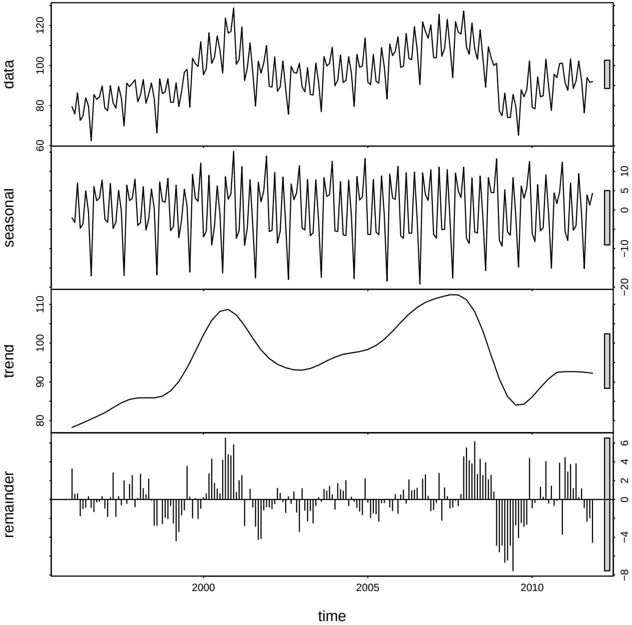
STL decomposition

- STL: “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

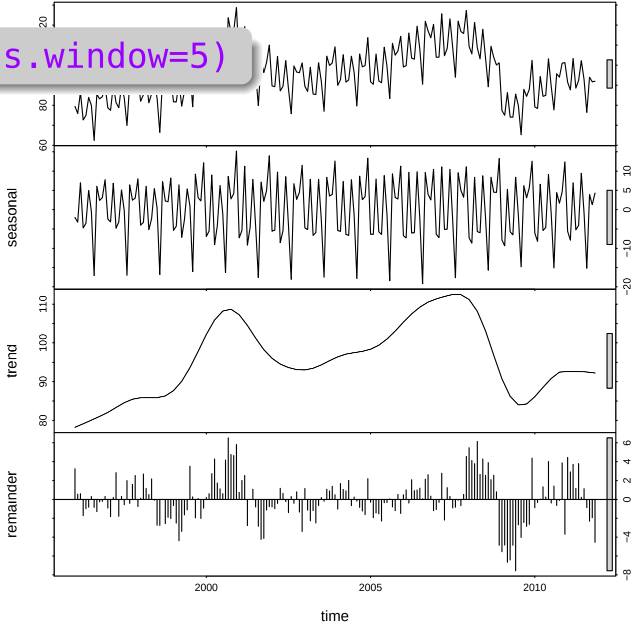
- STL: "Seasonal and Trend decomposition using Loess",
- Very versatile and robust.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers
- Only additive.
- Use Box-Cox transformations to get other decompositions.

STL decomposition

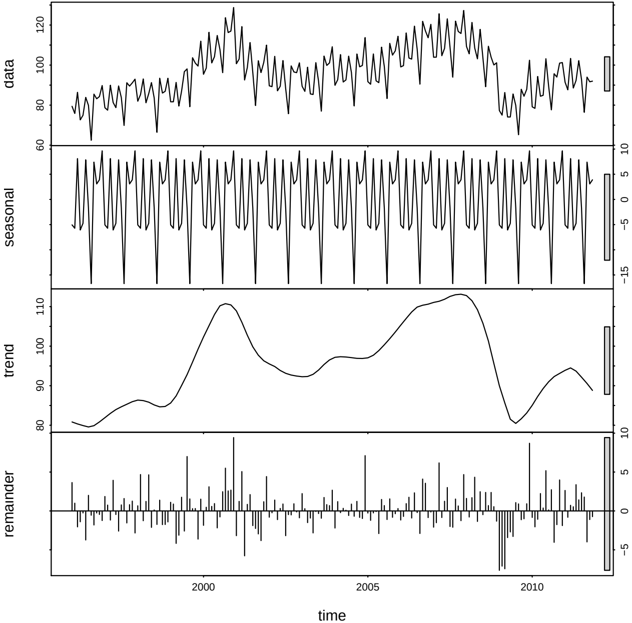


STL decomposition

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

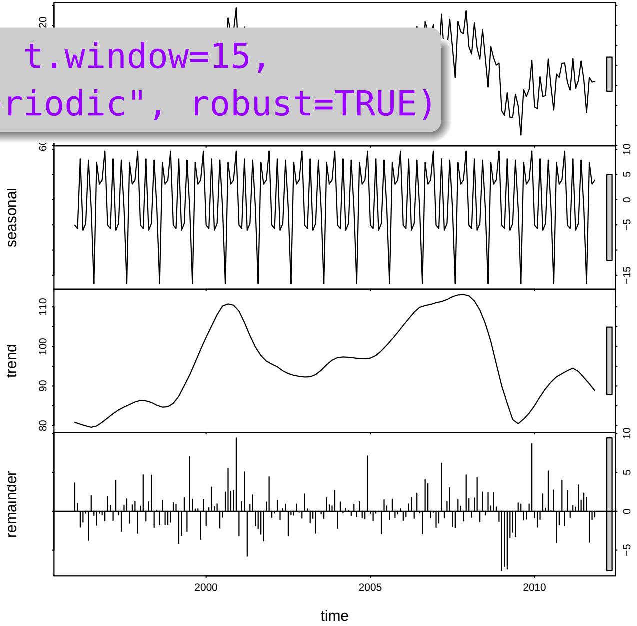


STL decomposition



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)
plot(fit)
```

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

STL decomposition in R

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

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

STL decomposition in R

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

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

Outline

1 STL decomposition

2 Forecasting and decomposition

3 Cross-validation

Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g., ETS model.
- 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.

Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g., ETS model.
- 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.

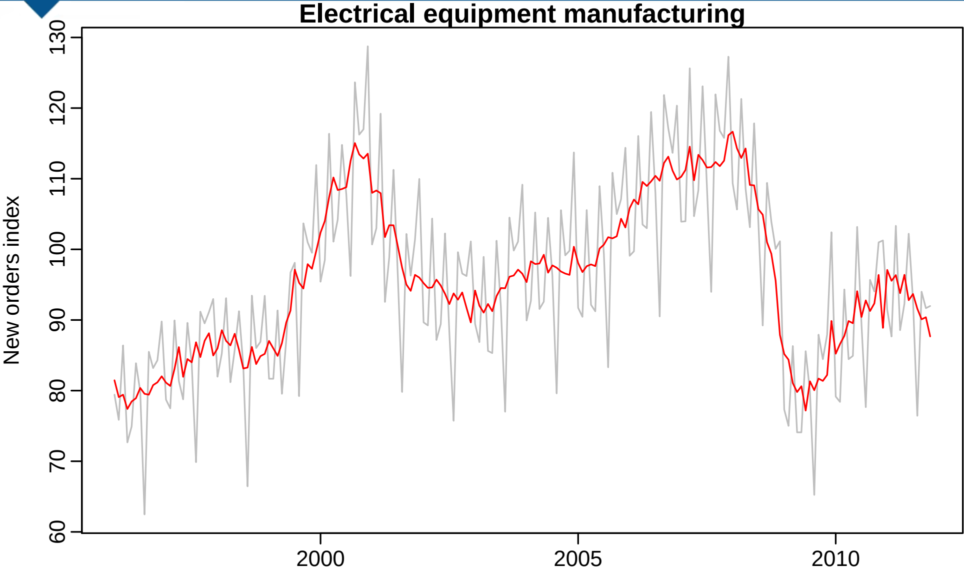
Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g., ETS model.
- 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.

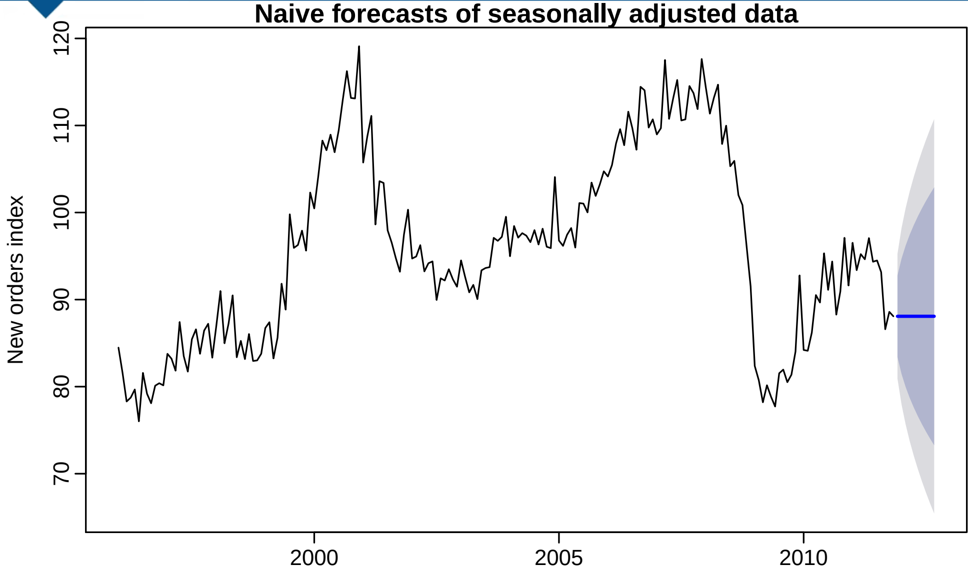
Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g., ETS model.
- 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.

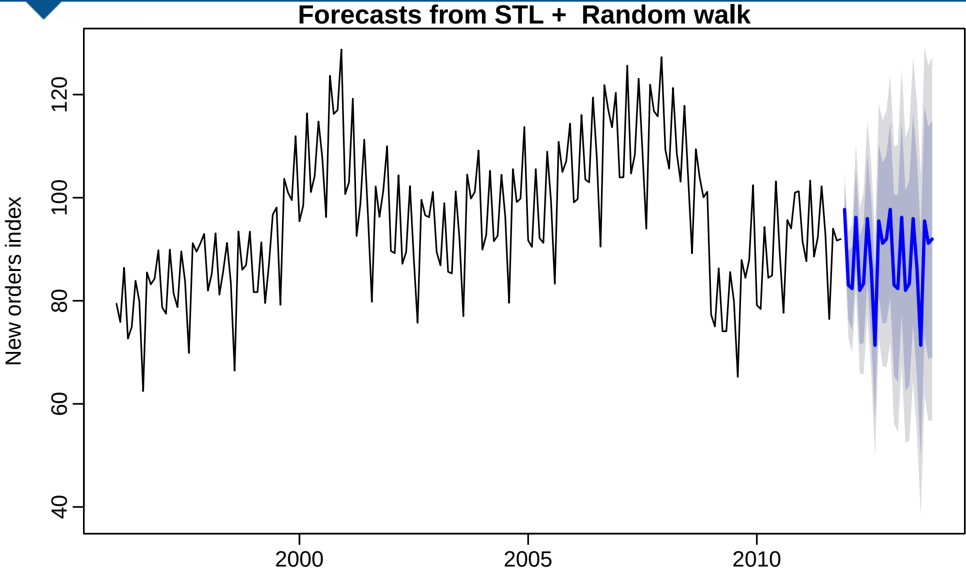
Seas adj elec equipment



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)
plot(naive(eeadj), xlab="New orders index")

fcast <- forecast(fit, method="naive")
plot(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.

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.

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.

Outline

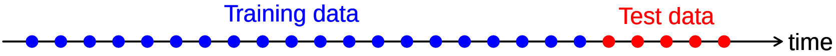
1 STL decomposition

2 Forecasting and decomposition

3 Cross-validation

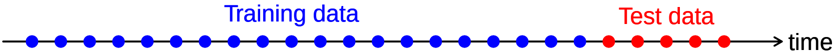
Cross-validation

Traditional evaluation

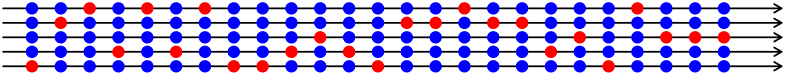


Cross-validation

Traditional evaluation

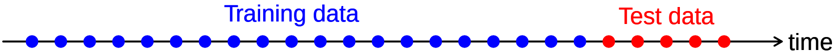


Standard cross-validation

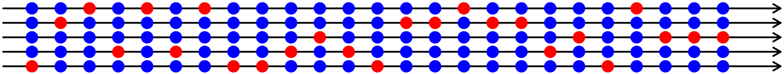


Cross-validation

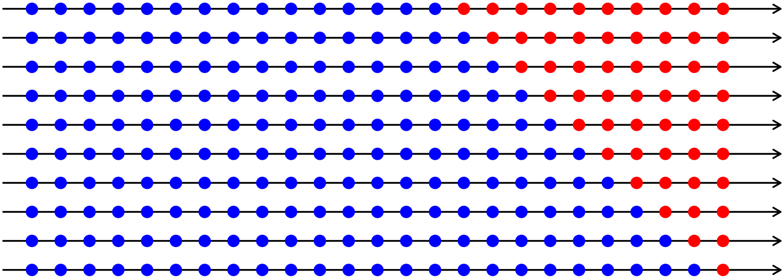
Traditional evaluation



Standard cross-validation

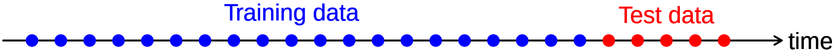


Time series cross-validation

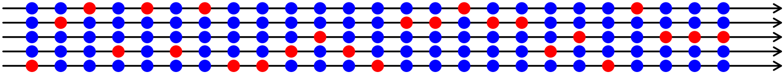


Cross-validation

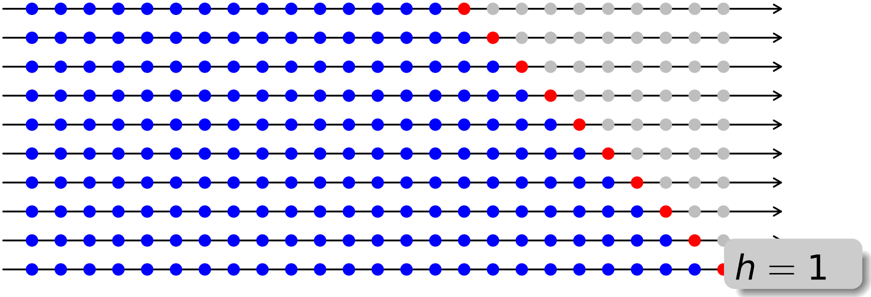
Traditional evaluation



Standard cross-validation

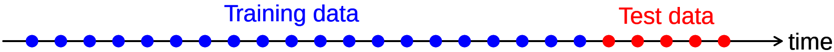


Time series cross-validation

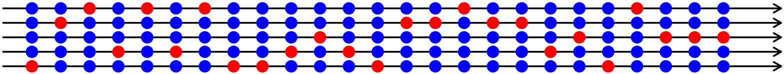


Cross-validation

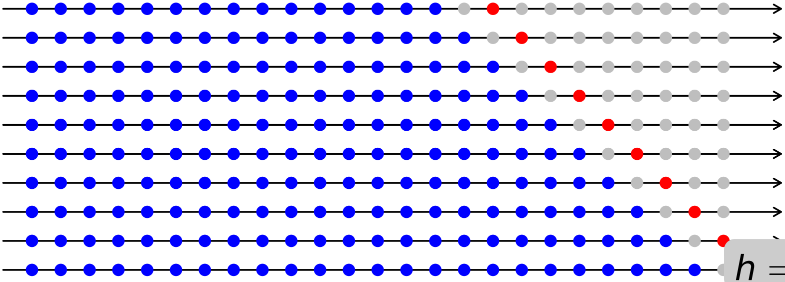
Traditional evaluation



Standard cross-validation



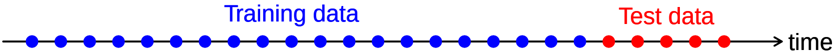
Time series cross-validation



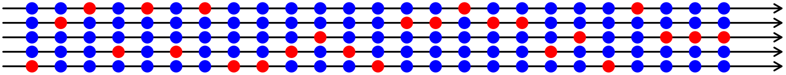
$h = 2$

Cross-validation

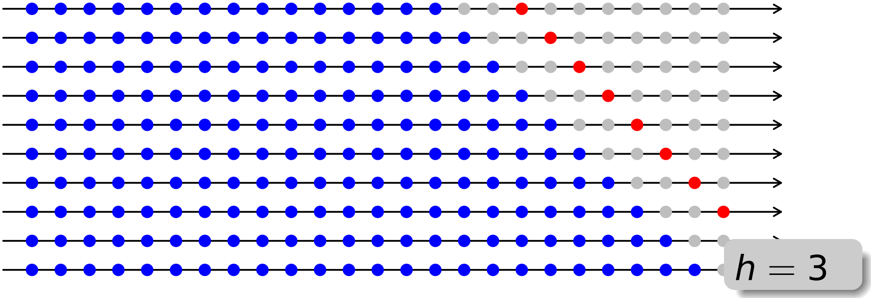
Traditional evaluation



Standard cross-validation

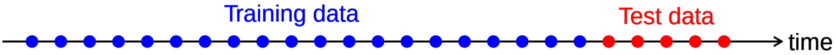


Time series cross-validation

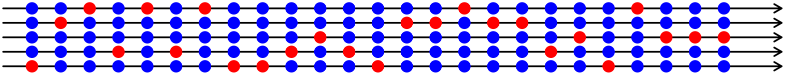


Cross-validation

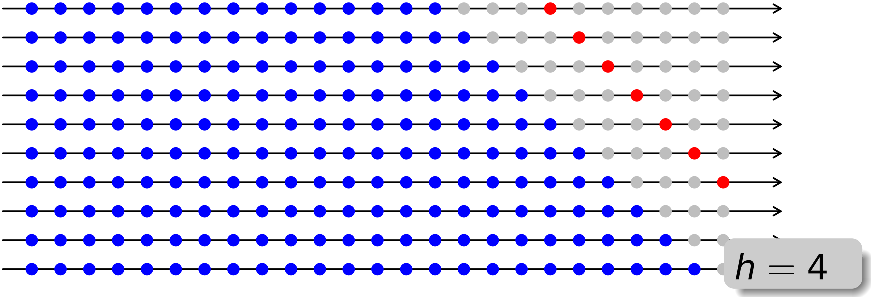
Traditional evaluation



Standard cross-validation

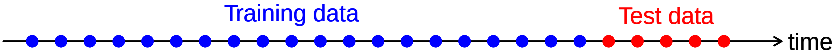


Time series cross-validation

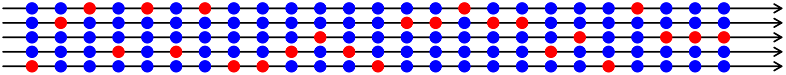


Cross-validation

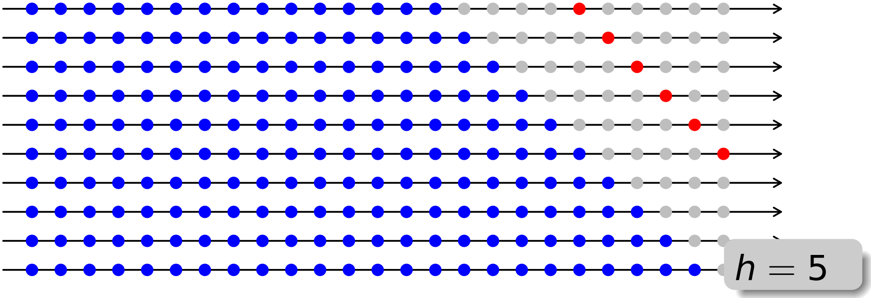
Traditional evaluation



Standard cross-validation

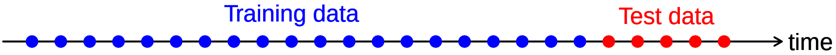


Time series cross-validation

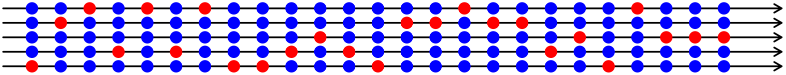


Cross-validation

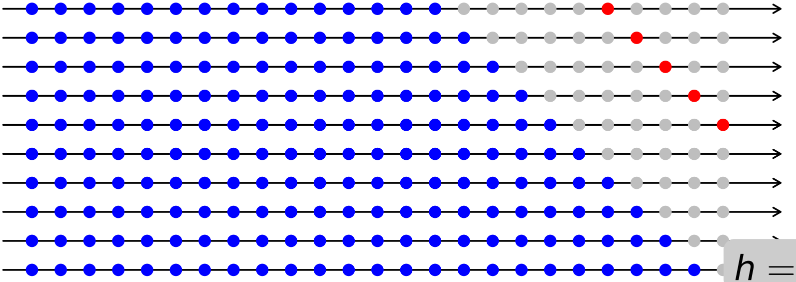
Traditional evaluation



Standard cross-validation



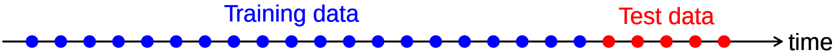
Time series cross-validation



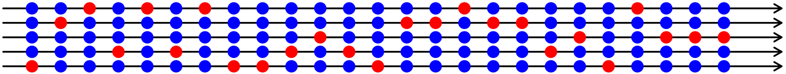
$h = 6$

Cross-validation

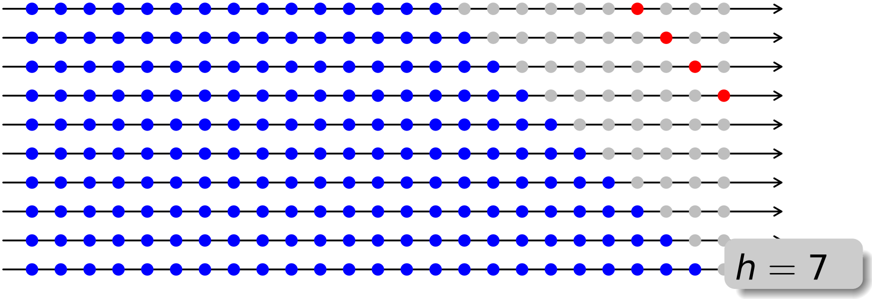
Traditional evaluation



Standard cross-validation

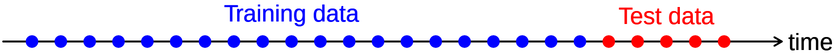


Time series cross-validation

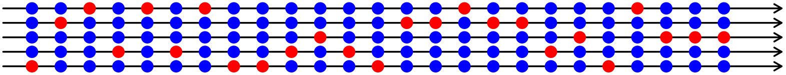


Cross-validation

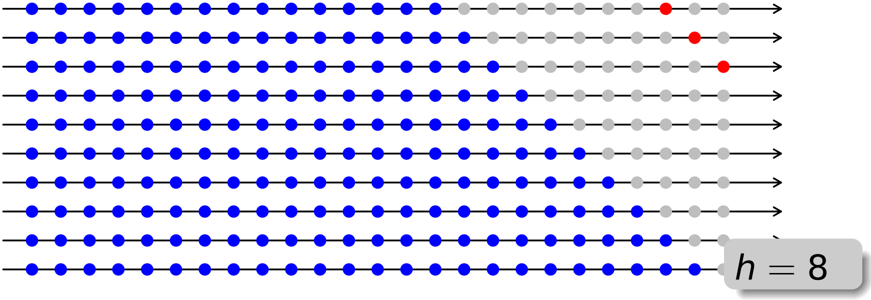
Traditional evaluation



Standard cross-validation

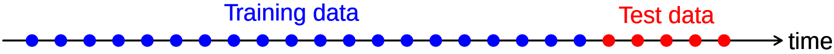


Time series cross-validation

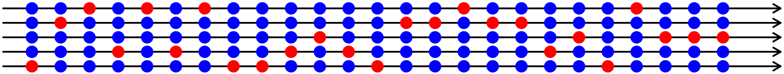


Cross-validation

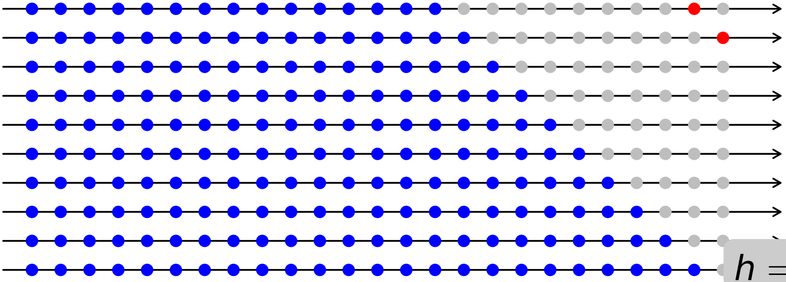
Traditional evaluation



Standard cross-validation



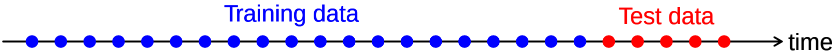
Time series cross-validation



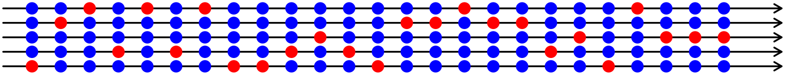
$h = 9$

Cross-validation

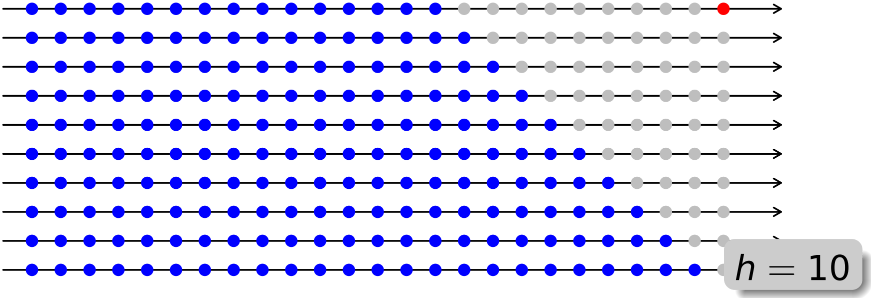
Traditional evaluation



Standard cross-validation

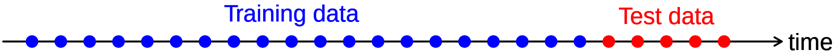


Time series cross-validation

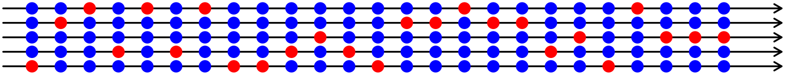


Cross-validation

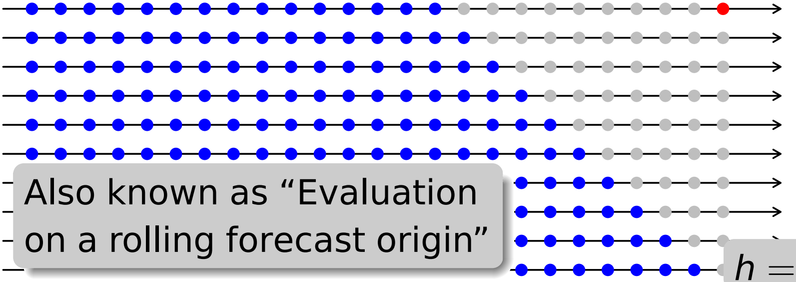
Traditional evaluation



Standard cross-validation



Time series cross-validation



Also known as "Evaluation on a rolling forecast origin"

$h = 10$

Some connections

Cross-sectional data

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via leave-one-out cross-validation. (Stone, 1977).

Time series cross-validation

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via one-step cross-validation. (Akaike, 1969,1973).

Some connections

Cross-sectional data

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via leave-one-out cross-validation. (Stone, 1977).

Time series cross-validation

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via one-step cross-validation. (Akaike, 1969,1973).

Some connections

Cross-sectional data

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via leave-one-out cross-validation. (Stone, 1977).

Time series cross-validation

- Minimizing the AIC is asymptotically equivalent to minimizing MSE via one-step cross-validation. (Akaike, 1969,1973).

Time series cross-validation

Assume k is the minimum number of observations for a training set.

- Select observation $k+i$ for test set, and use observations at times $1, 2, \dots, k+i-1$ to estimate model. Compute error on forecast for time $k+i$.
- Repeat for $i = 0, 1, \dots, T-k$ where T is total number of observations.
- Compute accuracy measure over all errors.

Also called **rolling forecasting origin** because the origin ($k+i-1$) at which forecast is based rolls forward in time.

Time series cross-validation

Assume k is the minimum number of observations for a training set.

- Select observation $k+i$ for test set, and use observations at times $1, 2, \dots, k+i-1$ to estimate model. Compute error on forecast for time $k+i$.
- Repeat for $i = 0, 1, \dots, T-k$ where T is total number of observations.
- Compute accuracy measure over all errors.

Also called **rolling forecasting origin** because the origin ($k+i-1$) at which forecast is based rolls forward in time.

Time series cross-validation

Assume k is the minimum number of observations for a training set.

- Select observation $k + i$ for test set, and use observations at times $1, 2, \ldots, k + i - 1$ to estimate model. Compute error on forecast for time $k + i$.
- Repeat for $i = 0, 1, \ldots, T - k$ where T is total number of observations.
- Compute accuracy measure over all errors.

Also called **rolling forecasting origin** because the origin ($k + i - 1$) at which forecast is based rolls forward in time.

Time series cross-validation

Assume k is the minimum number of observations for a training set.

- Select observation $k+i$ for test set, and use observations at times $1, 2, \dots, k+i-1$ to estimate model. Compute error on forecast for time $k+i$.
- Repeat for $i = 0, 1, \dots, T-k$ where T is total number of observations.
- Compute accuracy measure over all errors.

Also called **rolling forecasting origin** because the origin ($k+i-1$) at which forecast is based rolls forward in time.

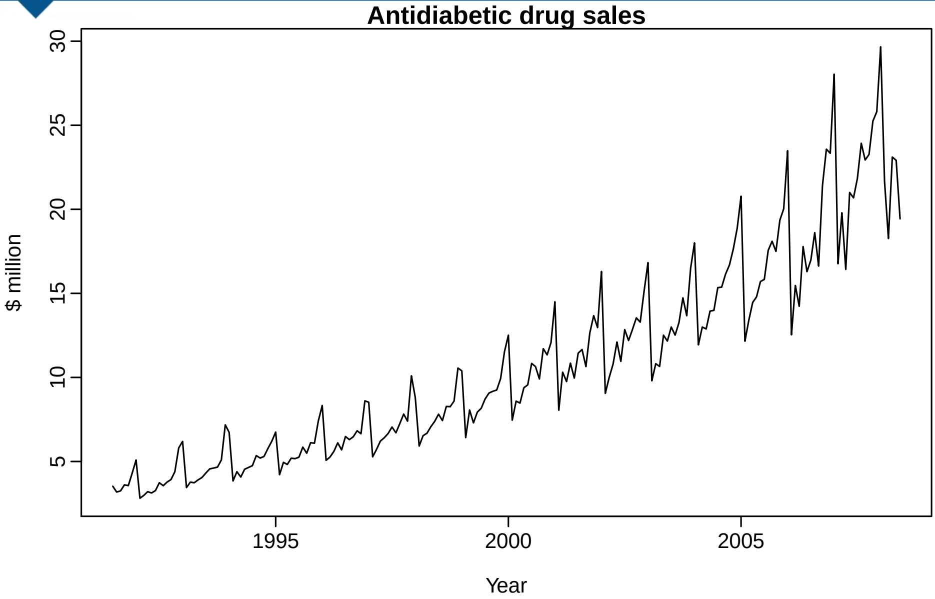
Time series cross-validation

Assume k is the minimum number of observations for a training set.

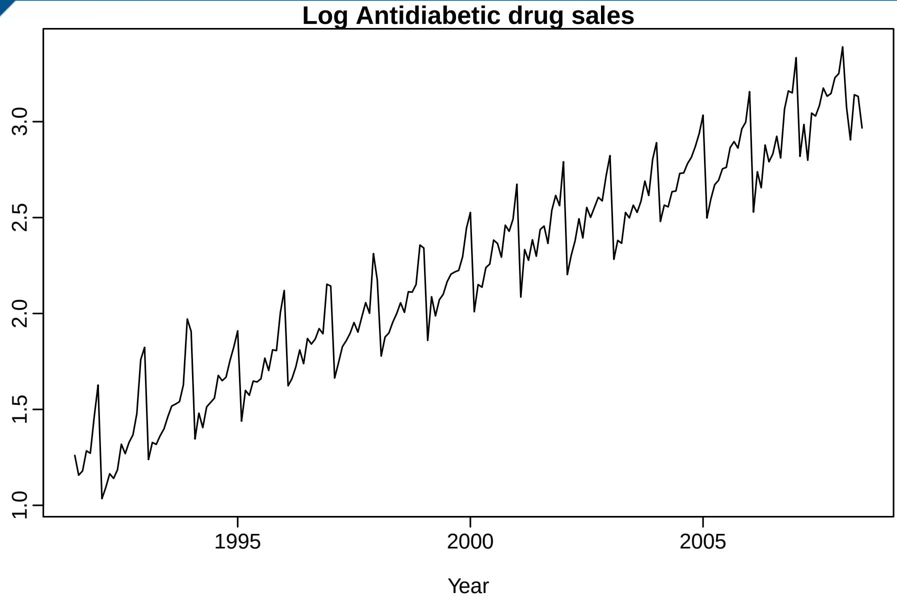
- Select observation $k+i$ for test set, and use observations at times $1, 2, \dots, k+i-1$ to estimate model. Compute error on forecast for time $k+i$.
- Repeat for $i = 0, 1, \dots, T-k$ where T is total number of observations.
- Compute accuracy measure over all errors.

Also called **rolling forecasting origin** because the origin $(k+i-1)$ at which forecast is based rolls forward in time.

Example: Pharmaceutical sales



Example: Pharmaceutical sales



Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.

Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.
- Forecast 12 steps ahead based on data to time $k + i - 1$ for $i = 1, 2, \ldots, 156$.
- Compare MAE values for each forecast horizon.

Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.
- Forecast 12 steps ahead based on data to time $k + i - 1$ for $i = 1, 2, \dots, 156$.
- Compare MAE values for each forecast horizon.

Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.
- Forecast 12 steps ahead based on data to time $k + i - 1$ for $i = 1, 2, \dots, 156$.
- Compare MAE values for each forecast horizon.

Example: Pharmaceutical sales

Which of these models is best?

1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.
- Forecast 12 steps ahead based on data to time $k + i - 1$ for $i = 1, 2, \dots, 156$.
- Compare MAE values for each forecast horizon.

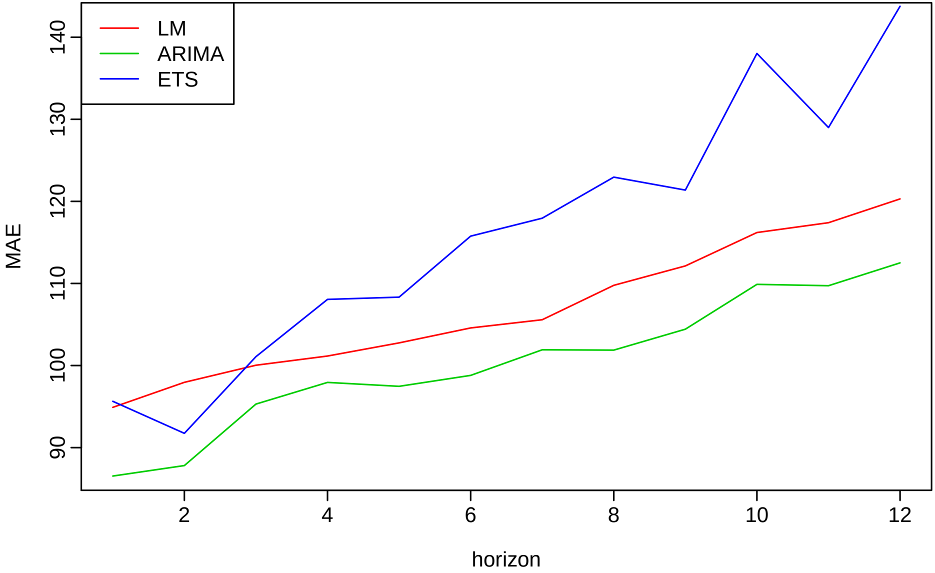
Example: Pharmaceutical sales

Which of these models is best?

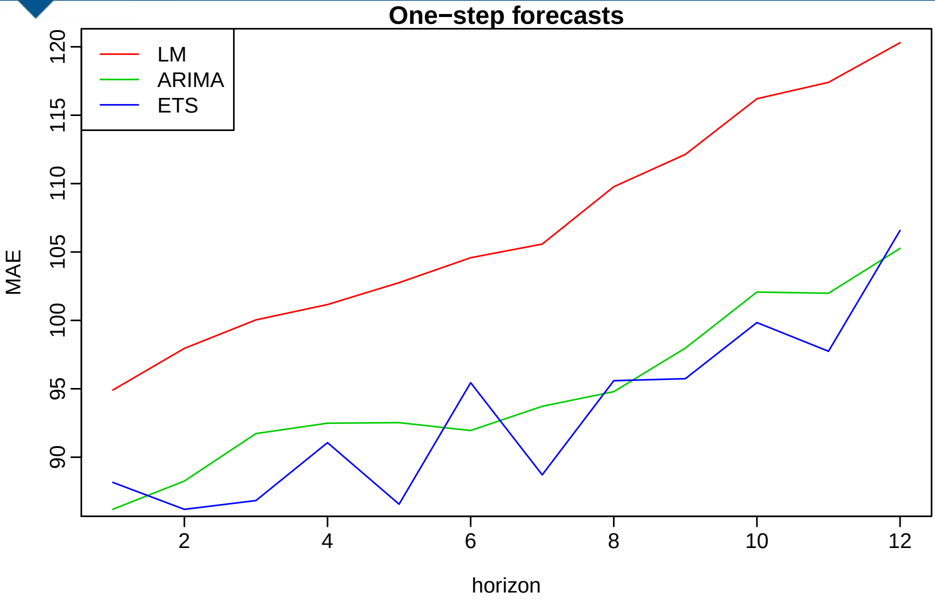
1. Linear model with trend and seasonal dummies applied to log data.
2. ARIMA model applied to log data
3. ETS model applied to original data

- Set $k = 48$ as minimum training set.
- Forecast 12 steps ahead based on data to time $k + i - 1$ for $i = 1, 2, \ldots, 156$.
- Compare MAE values for each forecast horizon.

Example: Pharmaceutical sales



Example: Pharmaceutical sales



Example: Pharmaceutical sales

```
k <- 48
n <- length(a10)
mae1 <- mae2 <- mae3 <- matrix(NA,n-k-12,12)
for(i in 1:(n-k-12))
{
  xshort <- window(a10,end=1995+(5+i)/12)
  xnext <- window(a10,start=1995+(6+i)/12,end=1996+(5+i)/12)
  fit1 <- tslm(xshort ~ trend + season, lambda=0)
  fcast1 <- forecast(fit1,h=12)
  fit2 <- auto.arima(xshort,D=1, lambda=0)
  fcast2 <- forecast(fit2,h=12)
  fit3 <- ets(xshort)
  fcast3 <- forecast(fit3,h=12)
  mae1[i,] <- abs(fcast1[['mean']]-xnext)
  mae2[i,] <- abs(fcast2[['mean']]-xnext)
  mae3[i,] <- abs(fcast3[['mean']]-xnext)
}
plot(1:12,colMeans(mae1),type="l",col=2,xlab="horizon",ylab="MAE",
     ylim=c(0.58,1.0))
lines(1:12,colMeans(mae2),type="l",col=3)
lines(1:12,colMeans(mae3),type="l",col=4)
legend("topleft",legend=c("LM","ARIMA","ETS"),col=2:4,lty=1)
```

Variations on time series cross validation

- Keep training window of fixed length.

```
xshort <- window(a10,start=i+1/12,end=1995+(5+i)/12)
```

- Compute one-step forecasts in out-of-sample period.

```
for(i in 1:(n-k))
{
  xshort <- window(a10,end=1995+(5+i)/12)
  xlong <- window(a10,start=1995+(6+i)/12)
  fit2 <- auto.arima(xshort,D=1, lambda=0)
  fit2a <- Arima(xlong,model=fit2)
  fit3 <- ets(xshort)
  fit3a <- ets(xlong,model=fit3)
  mae2a[i,] <- abs(residuals(fit3a))
  mae3a[i,] <- abs(residuals(fit2a))
}
```