MONASH University

MONASH BUSINESS SCHOOL

Forecasting: principles and practice

Rob J Hyndman

2.3 Stationarity and differencing

Outline

Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \dots, y_{t+s})$ does not depend on t.

Stationarity

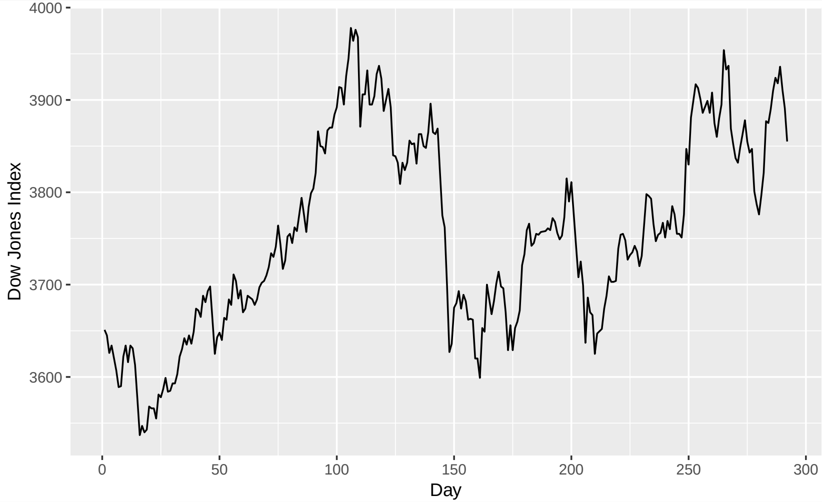
Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

A **stationary series** is:

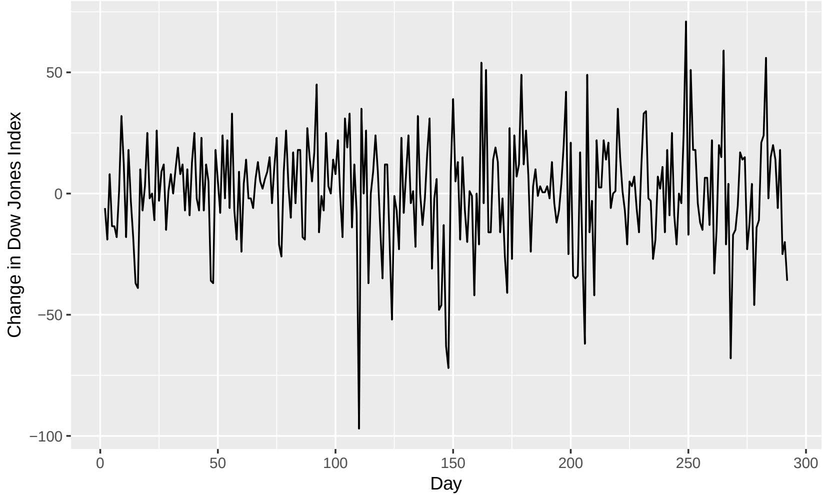
- roughly horizontal
- constant variance
- no patterns predictable in the long-term

Stationary?



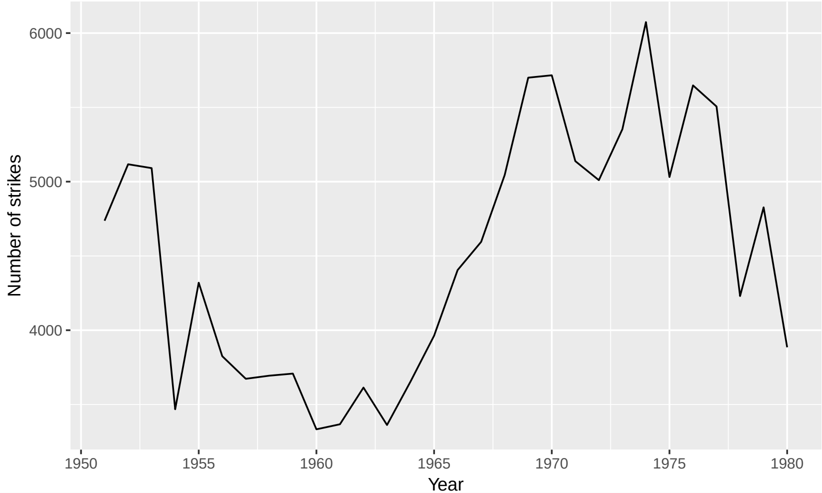
4000
3900
3800
3700
3600
Dow Jones Index
0
50
100
150
200
250
300
Day

Stationary?



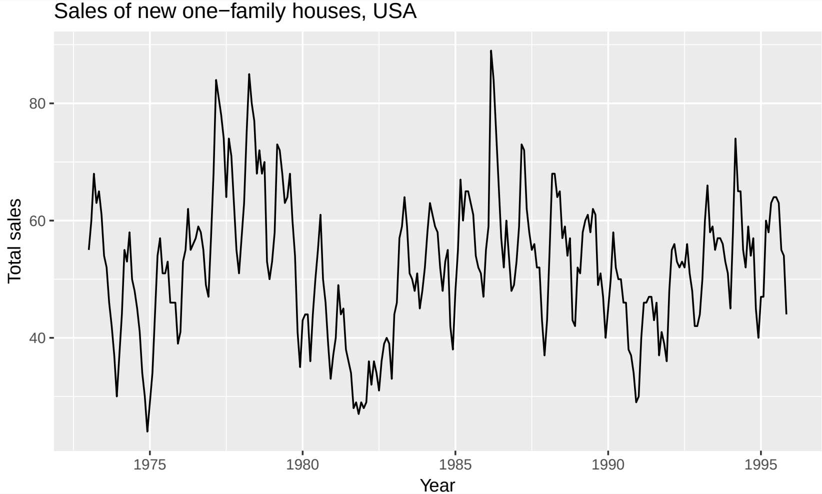
Change in Dow Jones Index
50
0
−50
−100
0
50
100
150
200
250
300
Day

Stationary?



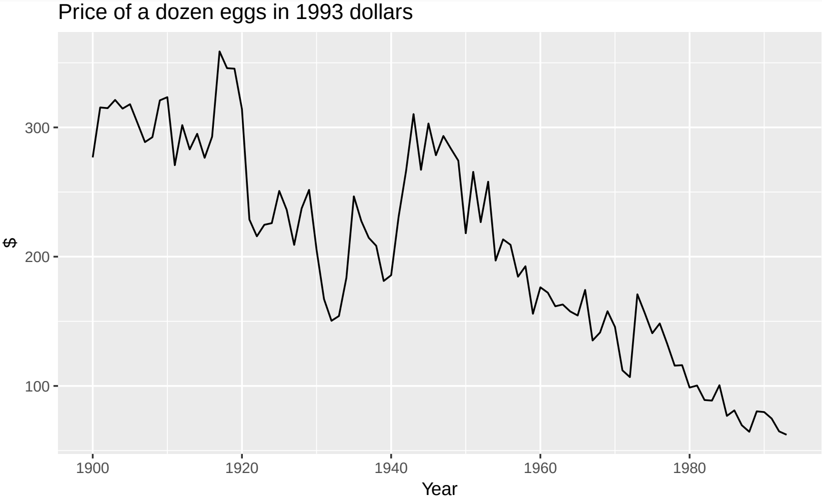
6000
5000
4000
Number of strikes
1950
1955
1960
1965
1970
1975
1980
Year

Stationary?

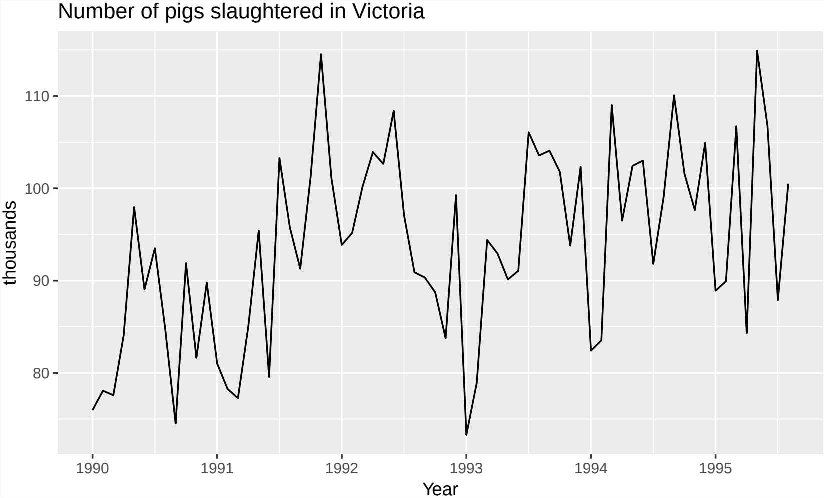


Sales of new one−family houses, USA
80
60
40
Total sales
1975
1980
1985
1990
1995
Year

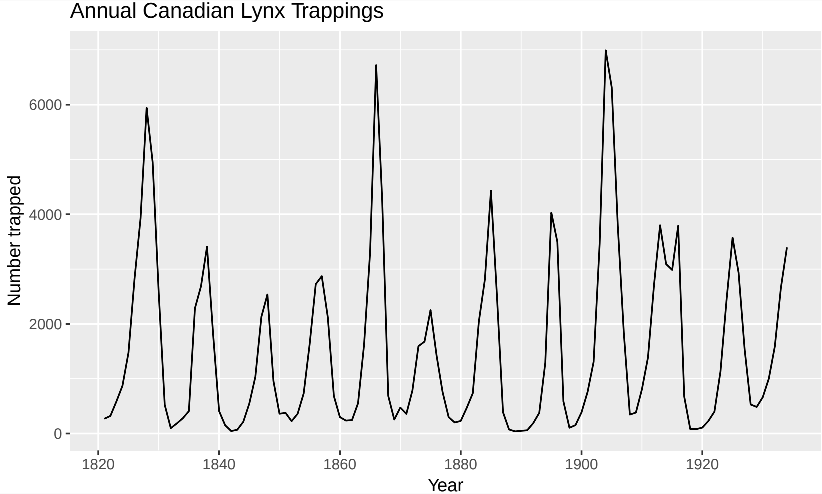
Stationary?


Price of a dozen eggs in 1993 dollars
$
300
200
100
1900
1920
1940
1960
1980
Year

Stationary?

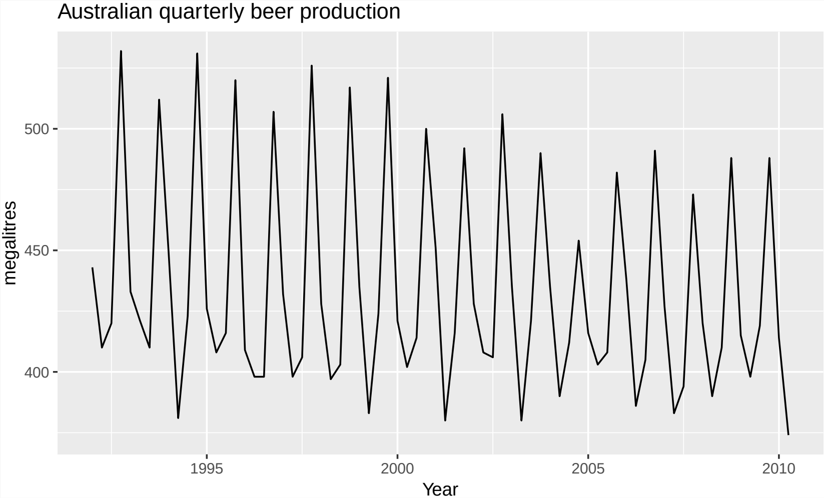

Number of pigs slaughtered in Victoria
110
100
90
80
thousands
1990
1991
1992
1993
1994
1995
Year

Stationary?



Annual Canadian Lynx Trappings
Number trapped
0
2000
4000
6000
1820
1840
1860
1880
1900
1920
Year

Stationary?



Australian quarterly beer production
megalitres
500
450
400
1995
2000
2005
2010
Year

Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

Transformations help to **stabilize the variance**.

For ARIMA modelling, we also need to **stabilize the mean**.

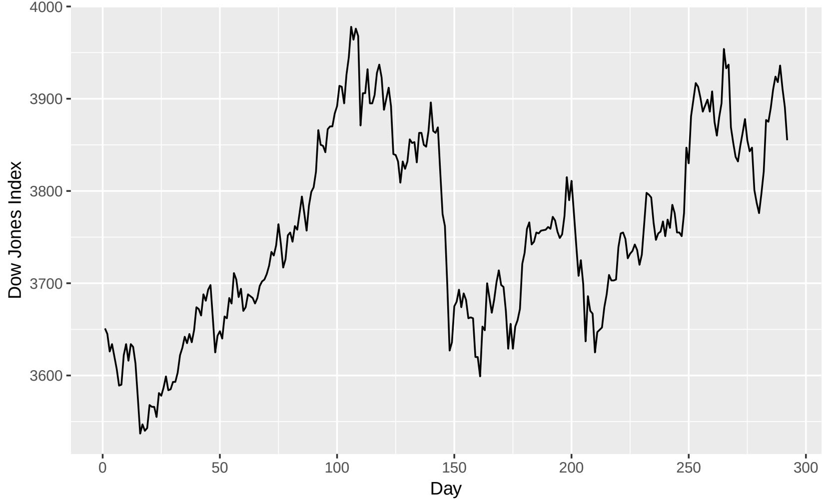
Non-stationarity in the mean

Identifying non-stationary series

- time plot.
- The ACF of stationary data drops to zero relatively quickly
- The ACF of non-stationary data decreases slowly.
- For non-stationary data, the value of r_1 is often large and positive.

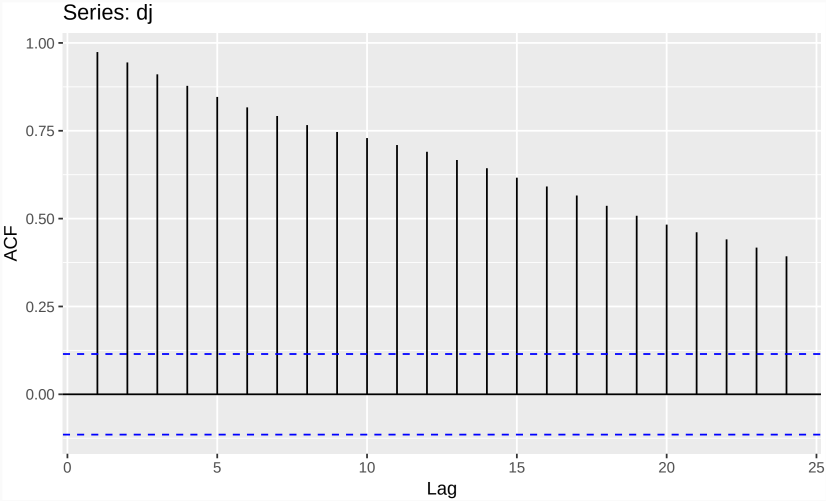
Example: Dow-Jones index

```
autoplot(dj) + ylab("Dow Jones Index") + xlab("Day")
```



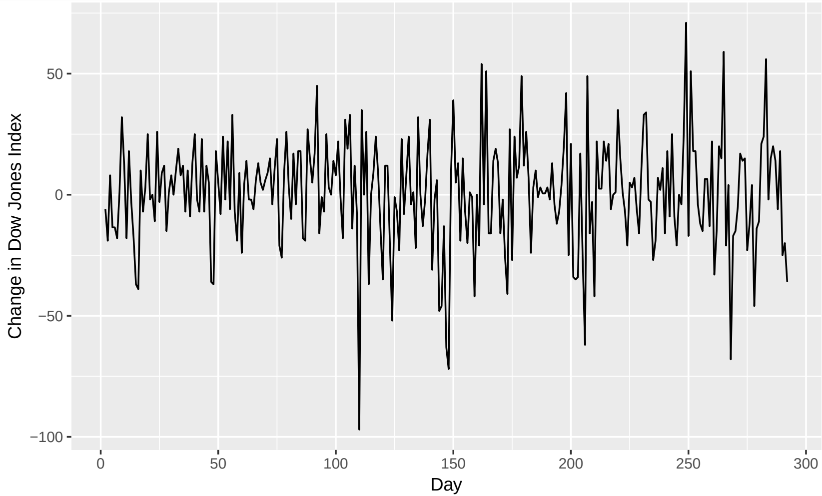
Example: Dow-Jones index

```r
ggAcf(dj)
```



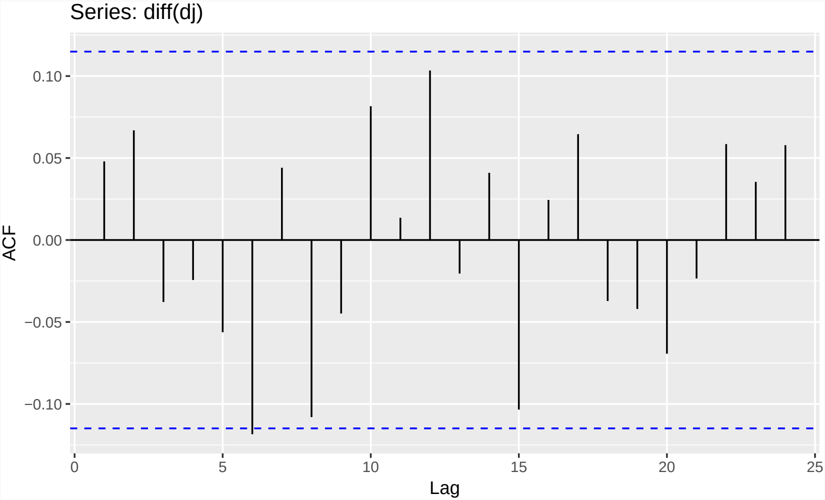
Example: Dow-Jones index

```r
autoplot(diff(dj)) +
  ylab("Change in Dow Jones Index") + xlab("Day")
```



Example: Dow-Jones index

```r
ggAcf(diff(dj))
```



Outline

Differencing

- Differencing helps to **stabilize the mean**.
- The differenced series is the *change* between each observation in the original series: $y'_t = y_t - y_{t-1}$.
- The differenced series will have only $T - 1$ values since it is not possible to calculate a difference y'_1 for the first observation.

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

- y''_t will have $T - 2$ values.
- In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

$$y'_t = y_t - y_{t-m}$$

where m = number of seasons.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

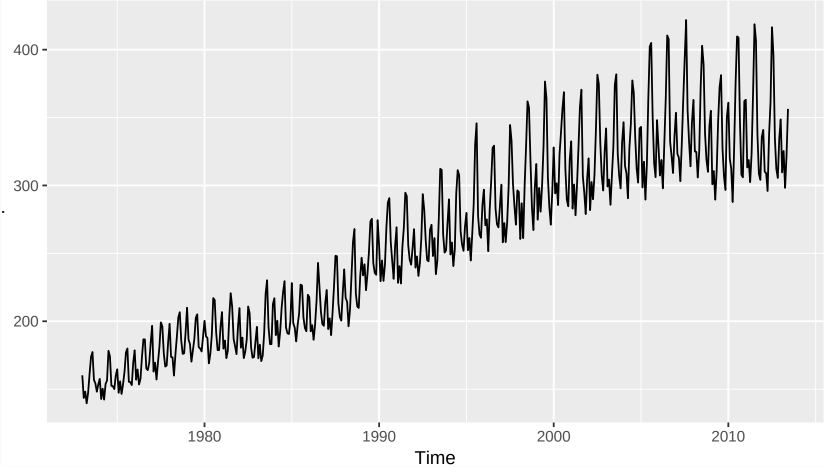
$$y'_t = y_t - y_{t-m}$$

where m = number of seasons.

- For monthly data $m = 12$.
- For quarterly data $m = 4$.

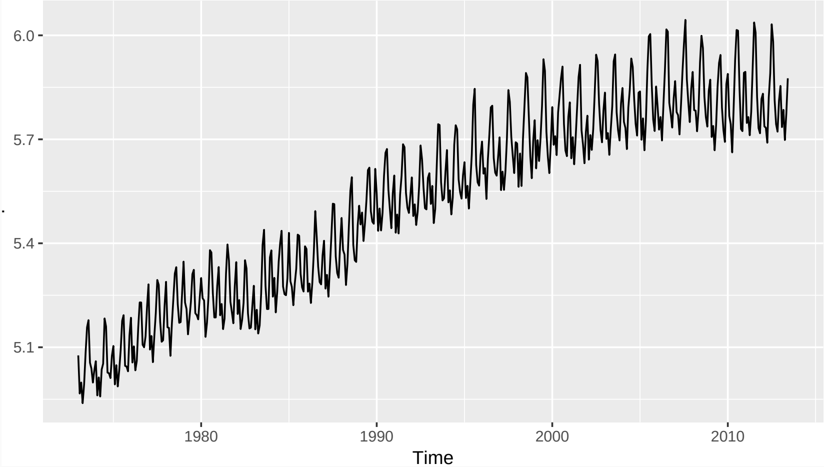
Electricity production

```r
usmelec %>% autoplot()
```



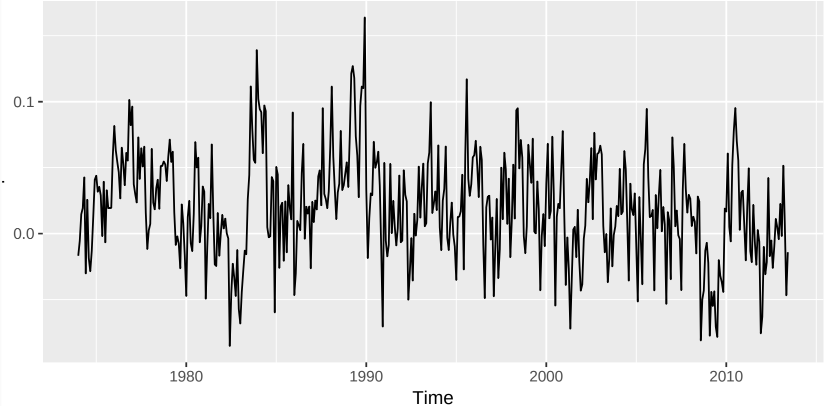
Electricity production

```r
usmelec %>% log() %>% autoplot()
```



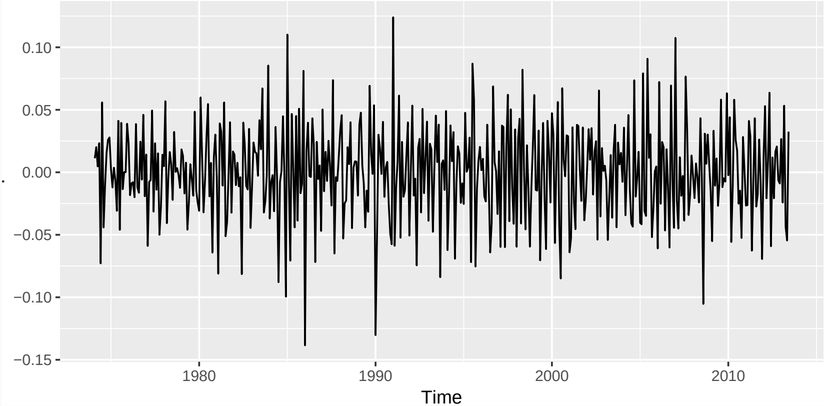
Electricity production

```
usmelec %>% log() %>% diff(lag=12) %>%
  autoplot()
```



Electricity production

```r
usmelec %>% log() %>% diff(lag=12) %>%
  diff(lag=1) %>% autoplot()
```



Electricity production

- Seasonally differenced series is closer to being stationary.
- Remaining non-stationarity can be removed with further first difference.

If $y'_t = y_t - y_{t-12}$ denotes seasonally differenced series, then twice-differenced series is

$$\begin{aligned} y^*_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \\ &= y_t - y_{t-1} - y_{t-12} + y_{t-13}\,. \end{aligned}$$

Seasonal differencing

When both seasonal and first differences are applied…

Seasonal differencing

When both seasonal and first differences are applied…

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

Seasonal differencing

When both seasonal and first differences are applied…

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

It is important that if differencing is used, the differences are interpretable.

Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

But taking lag 3 differences for yearly data, for example, results in a model which cannot be sensibly interpreted.

Outline

Unit root tests

Statistical tests to determine the required order of differencing.

1. Augmented Dickey Fuller test: null hypothesis is that the data are non-stationary and non-seasonal.
2. Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: null hypothesis is that the data are stationary and non-seasonal.
3. Other tests available for seasonal data.

KPSS test

```r
library(urca)
summary(ur.kpss(goog))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 10.72
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

KPSS test

```r
library(urca)
summary(ur.kpss(goog))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 10.72
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
ndiffs(goog)
```

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

Automatically selecting differences

STL decomposition: $y_t = T_t + S_t + R_t$

Seasonal strength $F_s = \max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t+R_t)}\right)$

If $F_s > 0.64$, do one seasonal difference.

Automatically selecting differences

STL decomposition: $y_t = T_t + S_t + R_t$

Seasonal strength $F_s = \max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t+R_t)}\right)$

If $F_s > 0.64$, do one seasonal difference.

```
usmelec %>% log() %>% nsdiffs()
```

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

```
usmelec %>% log() %>% diff(lag=12) %>% ndiffs()
```

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

Outline

Lab Session 15

Outline

Backshift notation

A very useful notational device is the backward shift operator, B, which is used as follows:

$$By_t = y_{t-1} \, .$$

Backshift notation

A very useful notational device is the backward shift operator, B, which is used as follows:

$$By_t = y_{t-1} \; .$$

In other words, B, operating on y_t, has the effect of **shifting the data back one period**.

Backshift notation

A very useful notational device is the backward shift operator, B, which is used as follows:

$$By_t = y_{t-1} \ .$$

In other words, B, operating on y_t, has the effect of **shifting the data back one period**. Two applications of B to y_t **shifts the data back two periods**:

$$B(By_t) = B^2y_t = y_{t-2} \ .$$

Backshift notation

A very useful notational device is the backward shift operator, B, which is used as follows:

$$By_t = y_{t-1} \; .$$

In other words, B, operating on y_t, has the effect of **shifting the data back one period**. Two applications of B to y_t **shifts the data back two periods**:

$$B(By_t) = B^2 y_t = y_{t-2} \; .$$

For monthly data, if we wish to shift attention to "the same month last year," then B^{12} is used, and the notation is $B^{12}y_t = y_{t-12}$.

Backshift notation

The backward shift operator is convenient for describing the process of *differencing*.

Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t\ .$$

Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t \ .$$

Note that a first difference is represented by $(1 - B)$.

Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t\ .$$

Note that a first difference is represented by $(1 - B)$.

Similarly, if second-order differences (i.e., first differences of first differences) have to be computed, then:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1 - B)^2 y_t\ .$$

Backshift notation

- Second-order difference is denoted $(1 - B)^2$.
- *Second-order difference* is not the same as a *second difference*, which would be denoted $1 - B^2$;
- In general, a dth-order difference can be written as
$$(1 - B)^d y_t.$$
- A seasonal difference followed by a first difference can be written as
$$(1 - B)(1 - B^m)y_t\ .$$

Backshift notation

The “backshift” notation is convenient because the terms can be multiplied together to see the combined effect.

$$\begin{aligned}(1-B)(1-B^m)y_t &= (1-B-B^m+B^{m+1})y_t \\ &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}.\end{aligned}$$

Backshift notation

The "backshift" notation is convenient because the terms can be multiplied together to see the combined effect.

$$\begin{aligned}(1-B)(1-B^m)y_t &= (1-B-B^m+B^{m+1})y_t \\ &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}.\end{aligned}$$

For monthly data, $m = 12$ and we obtain the same result as earlier.

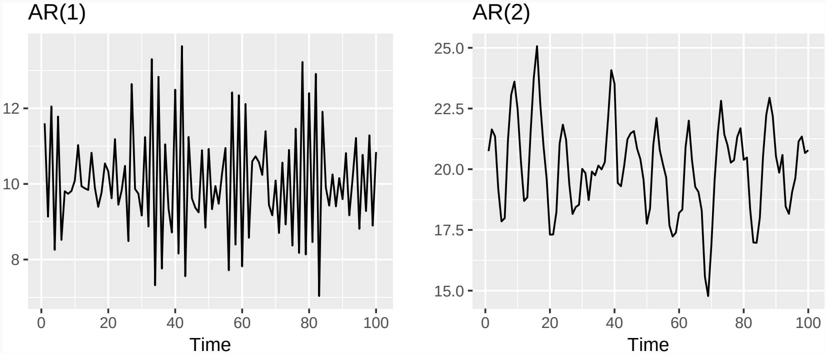
Outline

Autoregressive models

Autoregressive (AR) models:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

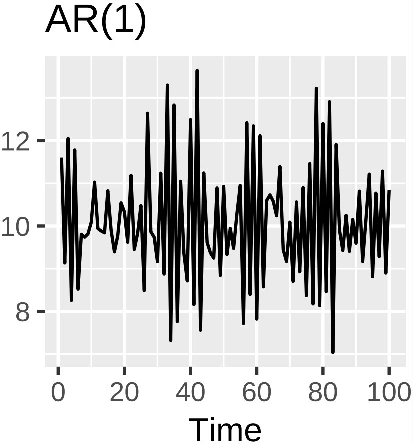
where ε_t is white noise. This is a multiple regression with **lagged values** of y_t as predictors.



AR(1) model

$$y_t = 2 - 0.8y_{t-1} + \varepsilon_t$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



AR(1) model

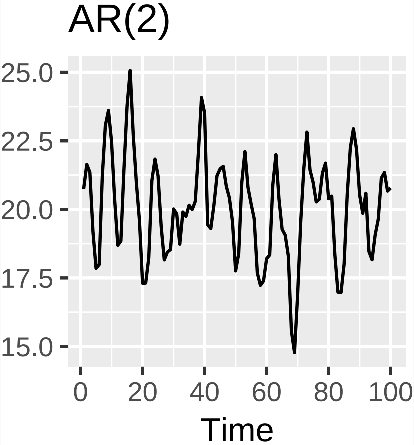
$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

- When $\phi_1 = 0$, y_t is **equivalent to WN**
- When $\phi_1 = 1$ and $c = 0$, y_t is **equivalent to a RW**
- When $\phi_1 = 1$ and $c \neq 0$, y_t is **equivalent to a RW with drift**
- When $\phi_1 < 0$, y_t tends to **oscillate between positive and negative values**.

AR(2) model

$$y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \varepsilon_t$$

$$\varepsilon_t \sim N(0,1), \qquad T = 100.$$



Stationarity conditions

We normally restrict autoregressive models to stationary data, and then some constraints on the values of the parameters are required.

General condition for stationarity

Complex roots of $1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$ lie outside the unit circle on the complex plane.

Stationarity conditions

We normally restrict autoregressive models to stationary data, and then some constraints on the values of the parameters are required.

General condition for stationarity

Complex roots of $1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$ lie outside the unit circle on the complex plane.

- For $p = 1$: $-1 < \phi_1 < 1$.
- For $p = 2$:

  $-1 < \phi_2 < 1 \qquad \phi_2 + \phi_1 < 1 \qquad \phi_2 - \phi_1 < 1.$
- More complicated conditions hold for $p \geq 3$.

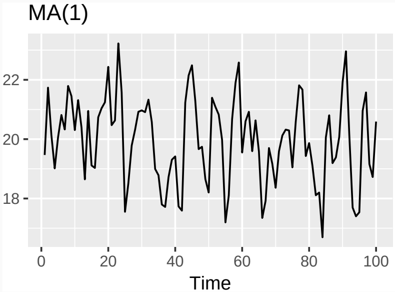
Outline

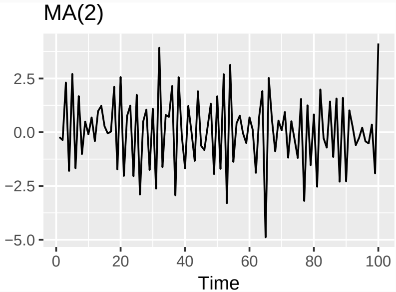
Moving Average (MA) models

Moving Average (MA) models:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

where ε_t is white noise. This is a multiple regression with ***past errors*** as predictors. *Don't confuse this with moving average smoothing!*

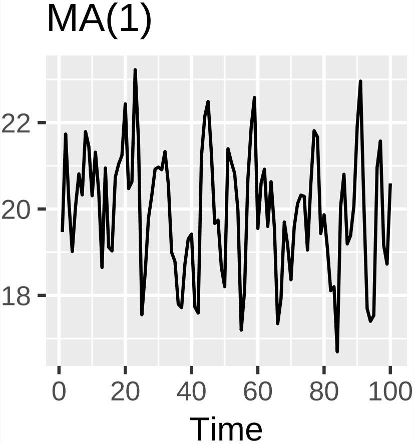




MA(1) model

$$y_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$$

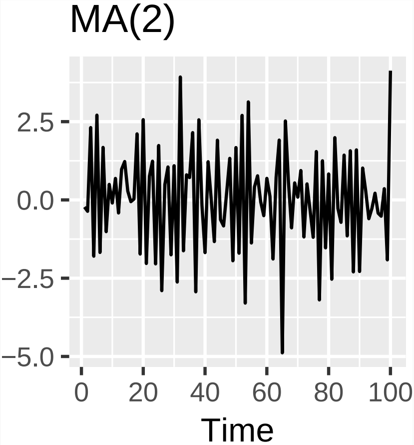
$$\varepsilon_t \sim N(0,1), \quad T = 100.$$



MA(2) model

$$y_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



Invertibility

- Invertible models have property that distant past has negligible effect on forecasts. Requires consraints on MA parameters.

General condition for invertibility

Complex roots of $1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q$ lie outside the unit circle on the complex plane.

Invertibility

- Invertible models have property that distant past has negligible effect on forecasts. Requires consraints on MA parameters.

General condition for invertibility

Complex roots of $1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q$ lie outside the unit circle on the complex plane.

- For $q = 1$: $-1 < \theta_1 < 1$.
- For $q = 2$:

  $$-1 < \theta_2 < 1 \qquad \theta_2 + \theta_1 > -1 \qquad \theta_1 - \theta_2 < 1.$$

- More complicated conditions hold for $q \geq 3$.

Outline

ARIMA models

Autoregressive Moving Average models:

$$\begin{aligned} y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} \\ + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t. \end{aligned}$$

ARIMA models

Autoregressive Moving Average models:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t.$$

- Predictors include both **lagged values of y_t and lagged errors.**
- Conditions on coefficients ensure stationarity.
- Conditions on coefficients ensure invertibility.

ARIMA models

Autoregressive Moving Average models:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t.$$

- Predictors include both **lagged values of y_t and lagged errors.**
- Conditions on coefficients ensure stationarity.
- Conditions on coefficients ensure invertibility.

Autoregressive Integrated Moving Average models

- Combine ARMA model with **differencing**.
- $(1 - B)^d y_t$ follows an ARMA model.

ARIMA models

Autoregressive Integrated Moving Average models

ARIMA(p, d, q) model

AR: p = order of the autoregressive part

I: d = degree of first differencing involved

MA: q = order of the moving average part.

- White noise model: ARIMA(0,0,0)
- Random walk: ARIMA(0,1,0) with no constant
- Random walk with drift: ARIMA(0,1,0) with const.
- AR(p): ARIMA(p,0,0)
- MA(q): ARIMA(0,0,q)

Backshift notation for ARIMA

- ARMA model:

$$y_t = c + \phi_1 B y_t + \cdots + \phi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

$$\text{or} \quad (1 - \phi_1 B - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

- ARIMA(1,1,1) model:

$$\begin{array}{ccccc} (1 - \phi_1 B) & (1 - B) y_t & = & c + (1 + \theta_1 B)\varepsilon_t \\ \uparrow & \uparrow & & \uparrow \\ \text{AR(1)} & \text{First} & & \text{MA(1)} \\ & \text{difference} & & \end{array}$$

Backshift notation for ARIMA

- ARMA model:

$$y_t = c + \phi_1 B y_t + \cdots + \phi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

$$\text{or} \quad (1 - \phi_1 B - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

- ARIMA(1,1,1) model:

$$\underset{\substack{\uparrow \\ \text{AR(1)}}}{(1 - \phi_1 B)} \quad \underset{\substack{\uparrow \\ \text{First} \\ \text{difference}}}{(1 - B) y_t} \quad = \quad c + \underset{\substack{\uparrow \\ \text{MA(1)}}}{(1 + \theta_1 B)\varepsilon_t}$$

Written out:

$$y_t = c + y_{t-1} + \phi_1 y_{t-1} - \phi_1 y_{t-2} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

R model

Intercept form

$$(1 - \phi_1 B - \cdots - \phi_p B^p) y_t' = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

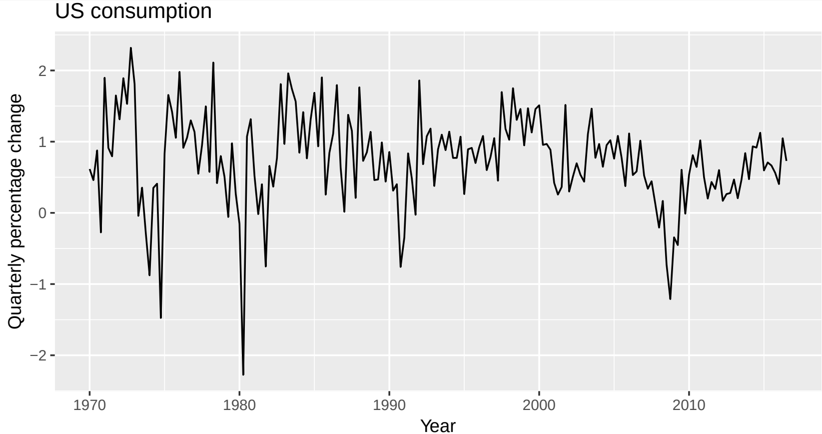
Mean form

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(y_t' - \mu) = (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

- $y_t' = (1 - B)^d y_t$
- μ is the mean of y_t'.
- $c = \mu(1 - \phi_1 - \cdots - \phi_p)$.
- R uses mean form.

US personal consumption

```r
autoplot(uschange[,"Consumption"]) +
  xlab("Year") + ylab("Quarterly percentage change") +
  ggtitle("US consumption")
```



US personal consumption

```r
(fit <- auto.arima(uschange[,"Consumption"]))
```

```
## Series: uschange[, "Consumption"]
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ma1    ma2   mean
##        1.391  -0.581  -1.180  0.558  0.746
## s.e.   0.255   0.208   0.238  0.140  0.084
##
## sigma^2 estimated as 0.351:  log likelihood=-165.1
## AIC=342.3   AICc=342.8   BIC=361.7
```

US personal consumption

```r
(fit <- auto.arima(uschange[,"Consumption"]))
```

```
## Series: uschange[, "Consumption"]
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ma1    ma2   mean
##        1.391  -0.581  -1.180  0.558  0.746
## s.e.   0.255   0.208   0.238  0.140  0.084
##
## sigma^2 estimated as 0.351:  log likelihood=-165.1
## AIC=342.3   AICc=342.8   BIC=361.7
```

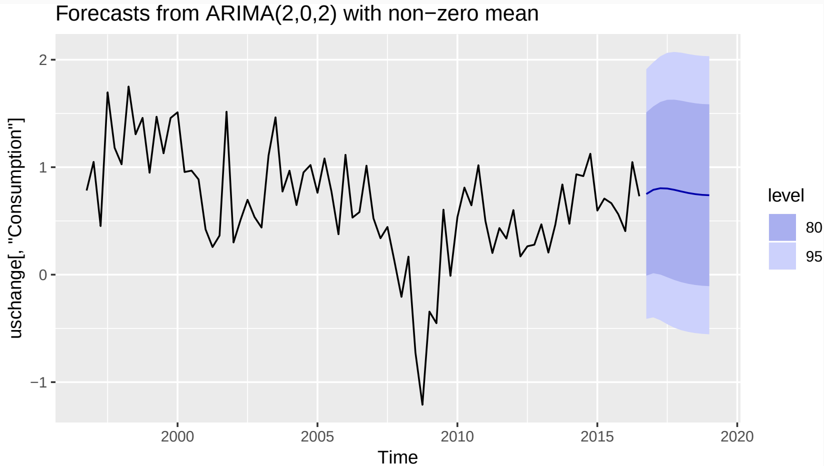
ARIMA(2,0,2) model:

$$y_t = c + 1.391y_{t-1} - 0.581y_{t-2} - 1.180\varepsilon_{t-1} + 0.558\varepsilon_{t-2} + \varepsilon_t,$$

where $c = 0.746 \times (1 - 1.391 + 0.581) = 0.142$ and $\varepsilon_t \sim N(0, 0.351)$.

US personal consumption

```r
fit %>% forecast(h=10) %>% autoplot(include=80)
```



Understanding ARIMA models

Long-term forecasts

zero	$c = 0, d = 0$	
non-zero constant	$c = 0, d = 1$	$c \neq 0, d = 0$
linear	$c = 0, d = 2$	$c \neq 0, d = 1$
quadratic	$c = 0, d = 3$	$c \neq 0, d = 2$

Forecast variance and *d*

- The higher the value of d, the more rapidly the prediction intervals increase in size.
- For $d = 0$, the long-term forecast standard deviation will go to the standard deviation of the historical data.

Understanding ARIMA models

Cyclic behaviour

- For cyclic forecasts, $p \geq 2$ and some restrictions on coefficients are required.
- If $p = 2$, we need $\phi_1^2 + 4\phi_2 < 0$. Then average cycle of length

$$(2\pi)/\left[\text{arc cos}(-\phi_1(1-\phi_2)/(4\phi_2))\right].$$

Outline

Maximum likelihood estimation

Having identified the model order, we need to estimate the parameters $c, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$.

Maximum likelihood estimation

Having identified the model order, we need to estimate the parameters $c, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$.

- MLE is very similar to least squares estimation obtained by minimizing
$$\sum_{t-1}^{T} e_t^2.$$
- The `Arima()` command allows CLS or MLE estimation.
- Non-linear optimization must be used in either case.
- Different software will give different estimates.

Information criteria

Akaike's Information Criterion (AIC):

$$\text{AIC} = -2\log(L) + 2(p + q + k + 1),$$

where L is the likelihood of the data,

$k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

Information criteria

Akaike's Information Criterion (AIC):

$$\text{AIC} = -2\log(L) + 2(p+q+k+1),$$

where L is the likelihood of the data,

$k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

Information criteria

Akaike's Information Criterion (AIC):

$$\text{AIC} = -2\log(L) + 2(p + q + k + 1),$$

where L is the likelihood of the data,

$k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2}.$$

Bayesian Information Criterion:

$$\text{BIC} = \text{AIC} + \log(T)(p + q + k - 1).$$

Information criteria

Akaike's Information Criterion (AIC):

$$\text{AIC} = -2\log(L) + 2(p+q+k+1),$$

where L is the likelihood of the data,

$k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

Bayesian Information Criterion:

$$\text{BIC} = \text{AIC} + \log(T)(p+q+k-1).$$

Good models are obtained by minimizing either the AIC, AICc or BIC. My preference is to use the AICc.

Outline

How does auto.arima() work?

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 and D via KPSS test and seasonal strength measure.
- Select p, q by minimising AICc.
- Use stepwise search to traverse model space.

How does auto.arima() work?

Step 1: Select values of d and D.

Step 2: 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?

Step 1: Select values of d and D.

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

Step 3: Consider variations of current model:

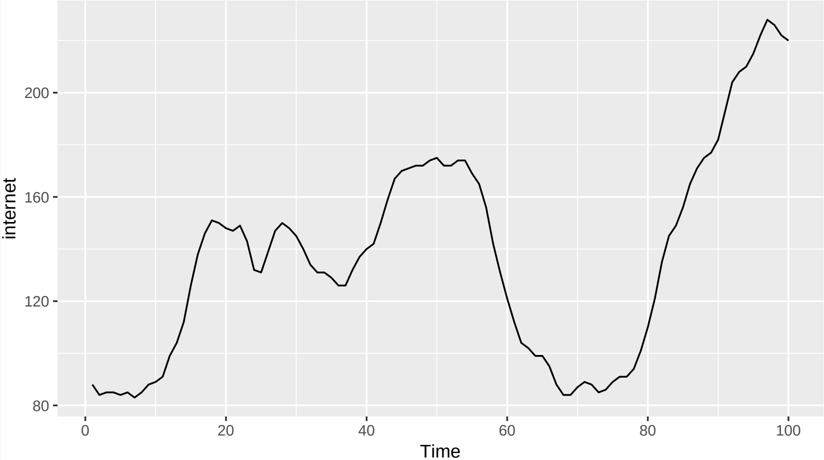
- vary one of $p, q,$ from current model by ± 1;
- p, q both vary from current model by ± 1;
- Include/exclude c from current model.

Model with lowest AICc becomes current model.

Repeat Step 3 until no lower AICc can be found.

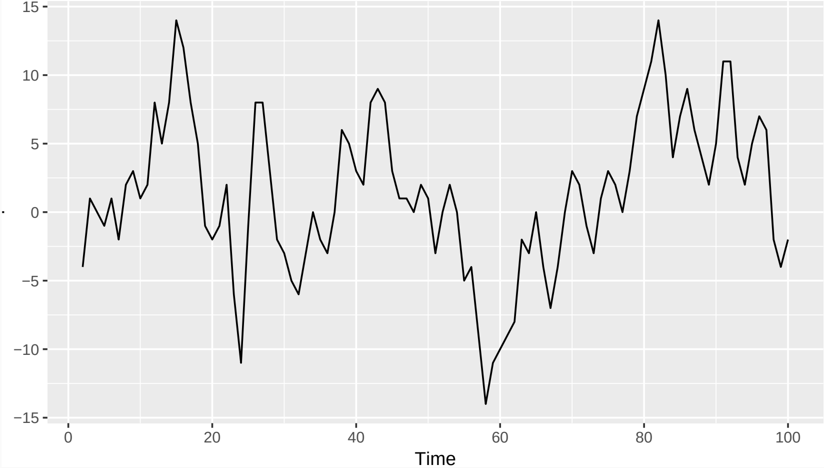
Choosing an ARIMA model

```r
autoplot(internet)
```



Choosing an ARIMA model

```r
internet %>% diff() %>% autoplot()
```



Choosing an ARIMA model

```r
(fit <- auto.arima(internet))
```

```
## Series: internet
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1    ma1
##        0.650  0.526
## s.e.   0.084  0.090
##
## sigma^2 estimated as 10:  log likelihood=-254.2
## AIC=514.3   AICc=514.5   BIC=522.1
```

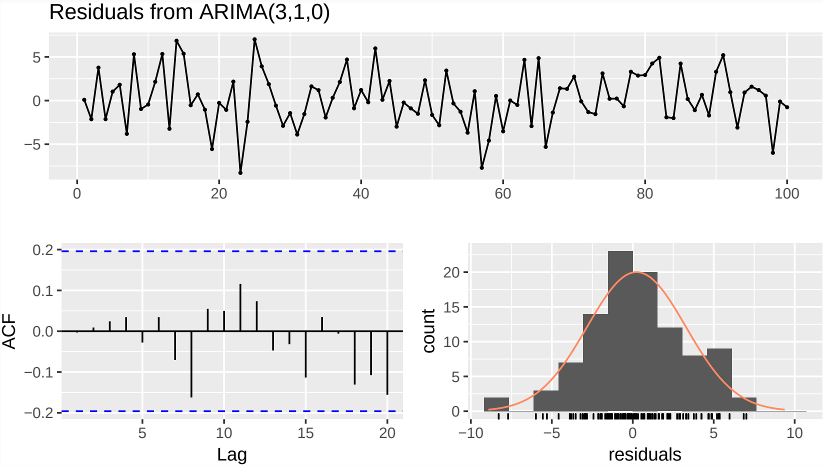
Choosing an ARIMA model

```
(fit <- auto.arima(internet, stepwise=FALSE,
  approximation=FALSE))
```

```
## Series: internet
## ARIMA(3,1,0)
##
## Coefficients:
##          ar1     ar2    ar3
##        1.151  -0.661  0.341
## s.e.   0.095   0.135  0.094
##
## sigma^2 estimated as 9.66:  log likelihood=-252
## AIC=512   AICc=512.4   BIC=522.4
```

Choosing an ARIMA model

```r
checkresiduals(fit, plot=TRUE)
```



#

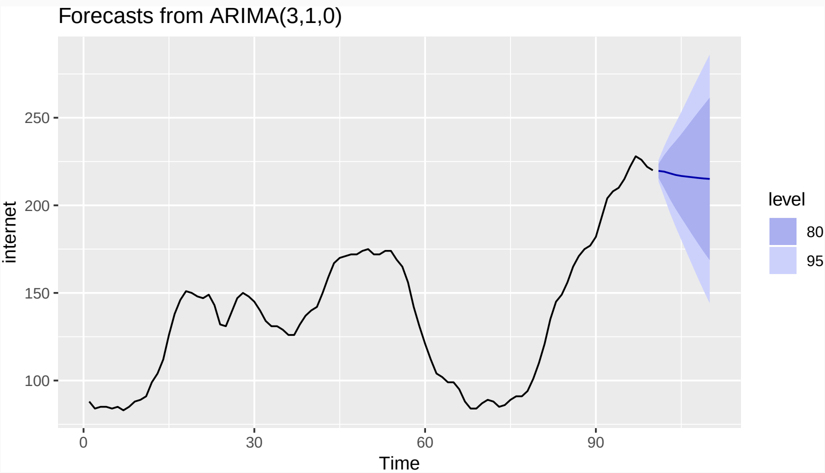
Choosing an ARIMA model

```r
checkresiduals(fit, plot=FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0)
## Q* = 4.5, df = 7, p-value = 0.7
##
## Model df: 3.   Total lags used: 10
```

Choosing an ARIMA model

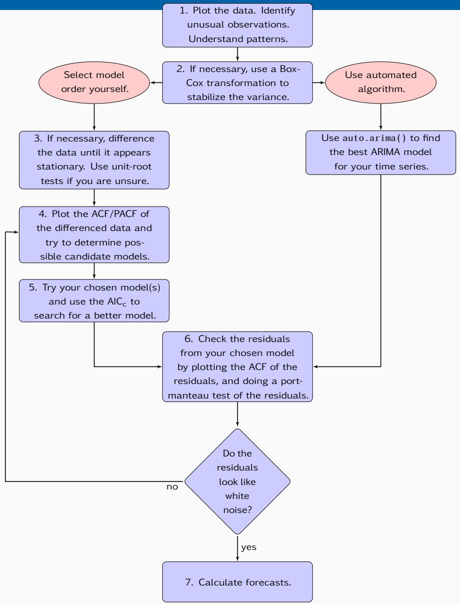
```
fit %>% forecast() %>% autoplot()
```



Modelling procedure with `auto.arima`

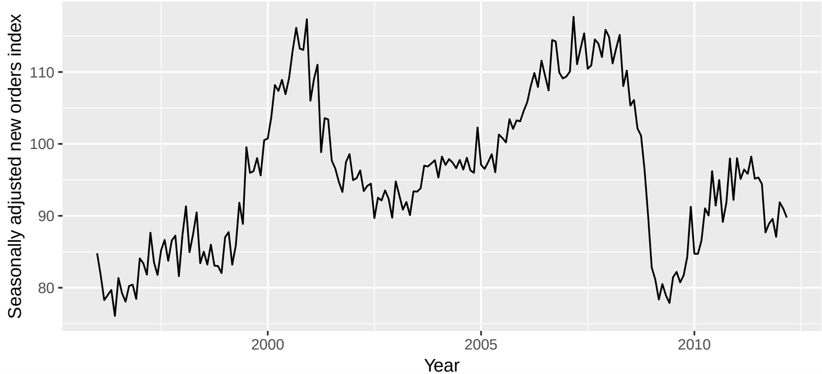
1. Plot the data. Identify any unusual observations.
2. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
3. Use `auto.arima` to select a model.
4. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
5. Once the residuals look like white noise, calculate forecasts.

Modelling procedure



Seasonally adjusted electrical equipment

```
eeadj <- seasadj(stl(elecequip, s.window="periodic"
autoplot(eeadj) + xlab("Year") +
  ylab("Seasonally adjusted new orders index")
```

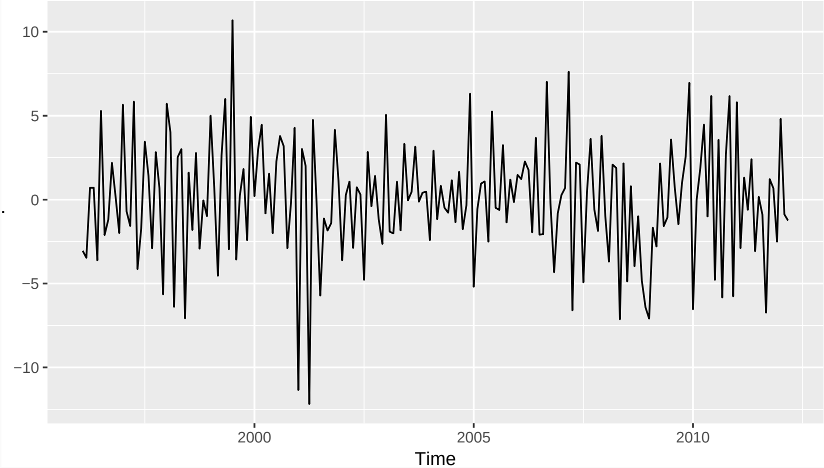


Seasonally adjusted electrical equipment

1. Time plot shows sudden changes, particularly big drop in 2008/2009 due to global economic environment. Otherwise nothing unusual and no need for data adjustments.
2. No evidence of changing variance, so no Box-Cox transformation.
3. Data are clearly non-stationary, so we take first differences.

Seasonally adjusted electrical equipment

```r
eeadj %>% diff() %>% autoplot()
```



Seasonally adjusted electrical equipment

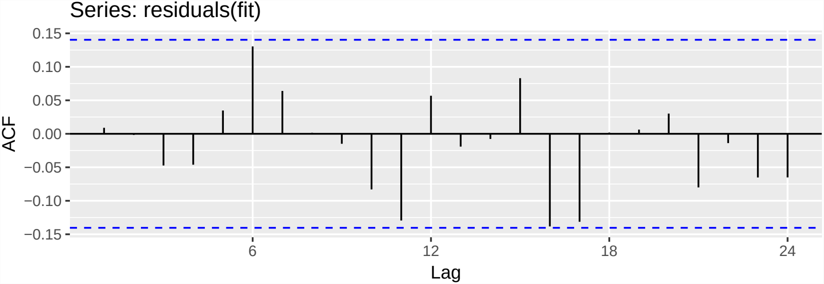
```
fit <- auto.arima(eeadj, stepwise=FALSE, approximation=FALSE)
summary(fit)
```

```
## Series: eeadj
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1    ar2    ar3     ma1
##        0.004  0.092  0.370  -0.392
## s.e.   0.220  0.098  0.067   0.243
##
## sigma^2 estimated as 9.58:  log likelihood=-492.7
## AIC=995.4   AICc=995.7   BIC=1012
##
## Training set error measures:
##                     ME  RMSE   MAE      MPE  MAPE   MASE
## Training set 0.03288 3.055 2.357 -0.00647 2.482 0.2884
##                    ACF1
```

Seasonally adjusted electrical equipment

6. ACF plot of residuals from ARIMA(3,1,1) model look like white noise.

```
ggAcf(residuals(fit))
```



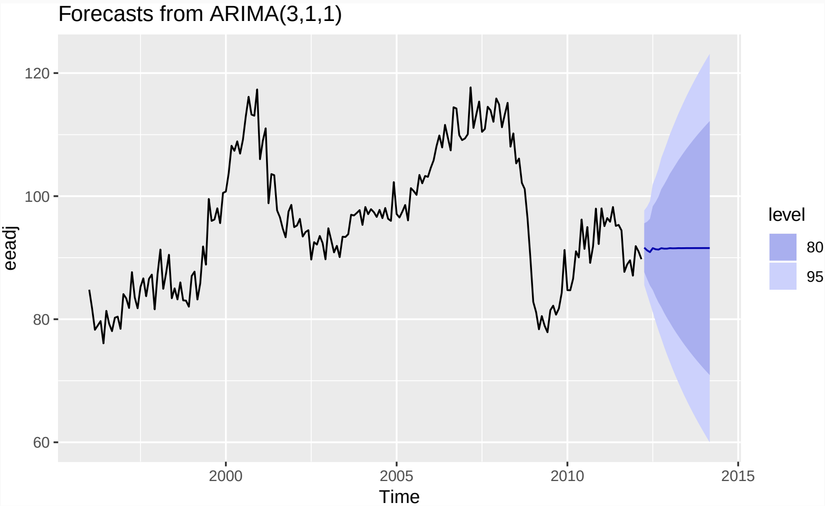
Seasonally adjusted electrical equipment

```r
checkresiduals(fit, plot=FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,1)
## Q* = 24, df = 20, p-value = 0.2
##
## Model df: 4.   Total lags used: 24
```

Seasonally adjusted electrical equipment

```
fit %>% forecast() %>% autoplot()
```



Outline

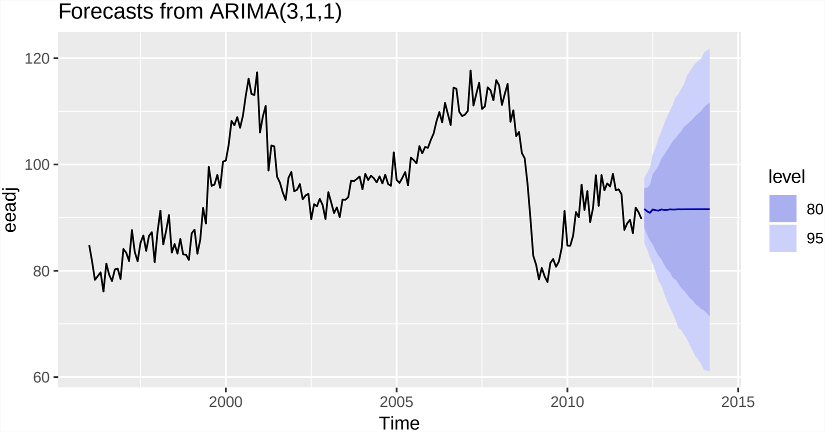
Lab Session 16

Prediction intervals

- Prediction intervals **increase in size with forecast horizon**.
- Calculations assume residuals are **uncorrelated** and **normally distributed**.
- Prediction intervals tend to be too narrow.
  - the uncertainty in the parameter estimates has not been accounted for.
  - the ARIMA model assumes historical patterns will not change during the forecast period.
  - the ARIMA model assumes uncorrelated future errors

Bootstrapped prediction intervals

```r
fit %>% forecast(bootstrap=TRUE) %>% autoplot()
```



- No assumption of normally distributed residuals. # Seasonal ARIMA models

Seasonal ARIMA models

ARIMA	$\underbrace{(p, d, q)}$	$\underbrace{(P, D, Q)_m}$
	$\uparrow$	$\uparrow$
	Non-seasonal part of the model	Seasonal part of of the model

where m = number of observations per year.

Seasonal ARIMA models

E.g., ARIMA$(1, 1, 1)(1, 1, 1)_4$ model (without constant)

Seasonal ARIMA models

E.g., ARIMA$(1, 1, 1)(1, 1, 1)_4$ model (without constant)

$$(1-\phi_1 B)(1-\Phi_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)(1+\Theta_1 B^4)\varepsilon_t.$$

Seasonal ARIMA models

E.g., ARIMA$(1, 1, 1)(1, 1, 1)_4$ model (without constant)

$$\underbrace{(1-\phi_1 B)}_{\text{Non-seasonal AR(1)}}\underbrace{(1-\Phi_1 B^4)}_{\text{Seasonal AR(1)}}\underbrace{(1-B)}_{\text{Non-seasonal difference}}\underbrace{(1-B^4)}_{\text{Seasonal difference}}y_t = \underbrace{(1+\theta_1 B)}_{\text{Non-seasonal MA(1)}}\underbrace{(1+\Theta_1 B^4)}_{\text{Seasonal MA(1)}}\varepsilon_t.$$

Seasonal ARIMA models

E.g., ARIMA(1, 1, 1)(1, 1, 1)$_4$ model (without constant)

$$(1-\phi_1 B)(1-\Phi_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)(1+\Theta_1 B^4)\varepsilon_t.$$

All the factors can be multiplied out and the general model written as follows:

$$\begin{aligned} y_t &= (1+\phi_1)y_{t-1} - \phi_1 y_{t-2} + (1+\Phi_1)y_{t-4} \\ &\quad - (1+\phi_1+\Phi_1+\phi_1\Phi_1)y_{t-5} + (\phi_1+\phi_1\Phi_1)y_{t-6} \\ &\quad - \Phi_1 y_{t-8} + (\Phi_1+\phi_1\Phi_1)y_{t-9} - \phi_1\Phi_1 y_{t-10} \\ &\quad + \varepsilon_t + \theta_1\varepsilon_{t-1} + \Theta_1\varepsilon_{t-4} + \theta_1\Theta_1\varepsilon_{t-5}. \end{aligned}$$

Common ARIMA models

The US Census Bureau uses the following models most often:

ARIMA(0,1,1)(0,1,1)$_m$	with log transformation
ARIMA(0,1,2)(0,1,1)$_m$	with log transformation
ARIMA(2,1,0)(0,1,1)$_m$	with log transformation
ARIMA(0,2,2)(0,1,1)$_m$	with log transformation
ARIMA(2,1,2)(0,1,1)$_m$	with no transformation

Understanding ARIMA models

Long-term forecasts

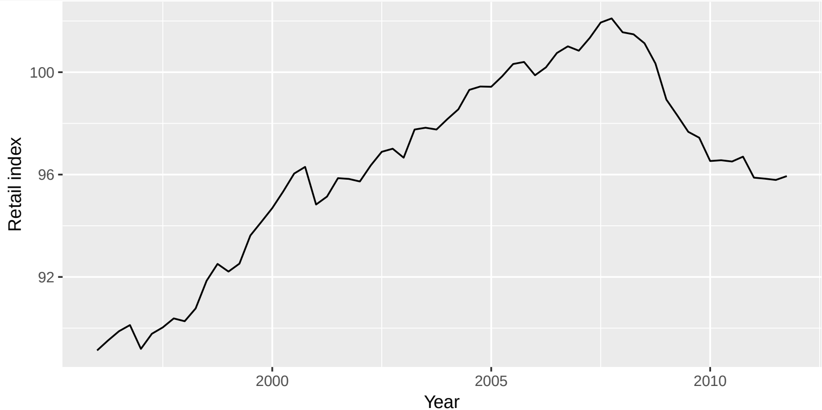
zero	$c = 0, d + D = 0$	
non-zero constant	$c = 0, d + D = 1$	$c \neq 0, d + D = 0$
linear	$c = 0, d + D = 2$	$c \neq 0, d + D = 1$
quadratic	$c = 0, d + D = 3$	$c \neq 0, d + D = 2$

Forecast variance and $d + D$

- The higher the value of $d + D$, the more rapidly the prediction intervals increase in size.
- For $d + D = 0$, the long-term forecast standard deviation will go to the standard deviation of the historical data.

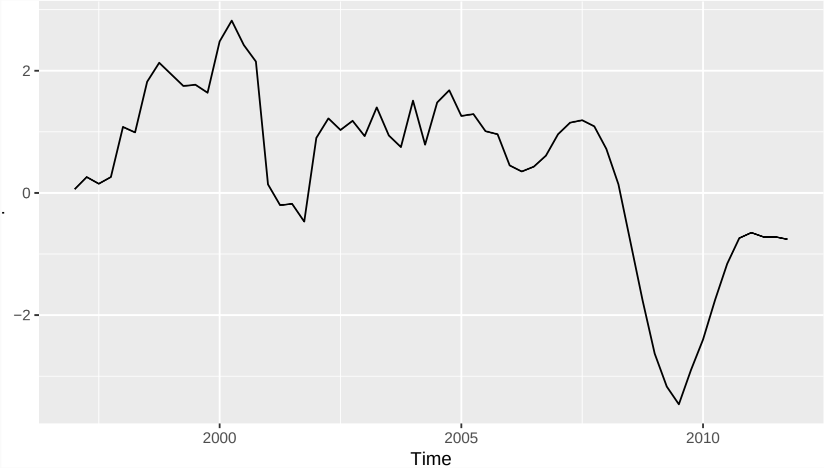
European quarterly retail trade

```r
autoplot(euretail) +
  xlab("Year") + ylab("Retail index")
```



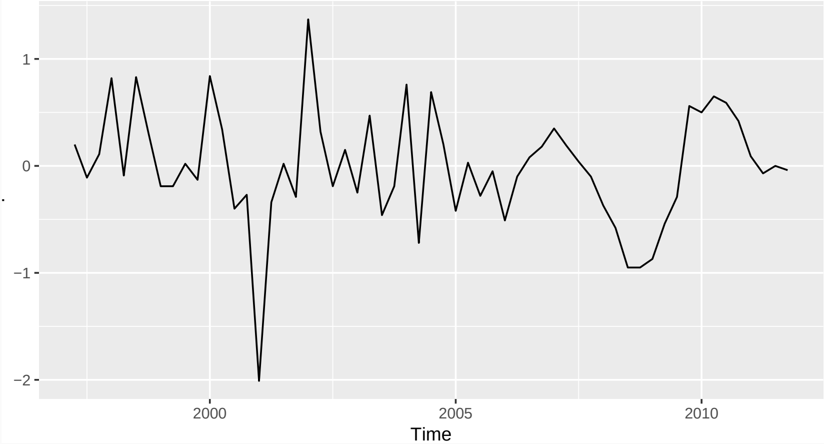
European quarterly retail trade

```
euretail %>% diff(lag=4) %>% autoplot()
```



European quarterly retail trade

```r
euretail %>% diff(lag=4) %>% diff() %>%
  autoplot()
```



European quarterly retail trade

```r
(fit <- auto.arima(euretail))
```

```
## Series: euretail
## ARIMA(1,1,2)(0,1,1)[4]
##
## Coefficients:
##          ar1      ma1    ma2     sma1
##        0.736  -0.466  0.216  -0.843
## s.e.   0.224   0.199  0.210   0.188
##
## sigma^2 estimated as 0.159:  log likelihood=-29.62
## AIC=69.24   AICc=70.38   BIC=79.63
```

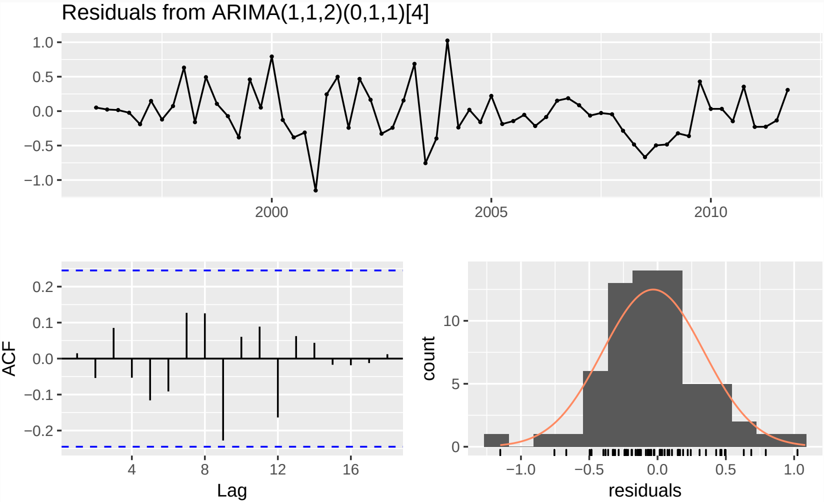
European quarterly retail trade

```
(fit <- auto.arima(euretail, stepwise=TRUE,
  approximation=FALSE))
```

```
## Series: euretail
## ARIMA(1,1,2)(0,1,1)[4]
##
## Coefficients:
##          ar1      ma1     ma2     sma1
##        0.736  -0.466  0.216  -0.843
## s.e.   0.224   0.199  0.210   0.188
##
## sigma^2 estimated as 0.159:  log likelihood=-29.62
## AIC=69.24   AICc=70.38   BIC=79.63
```

European quarterly retail trade

```r
checkresiduals(fit, test=FALSE)
```



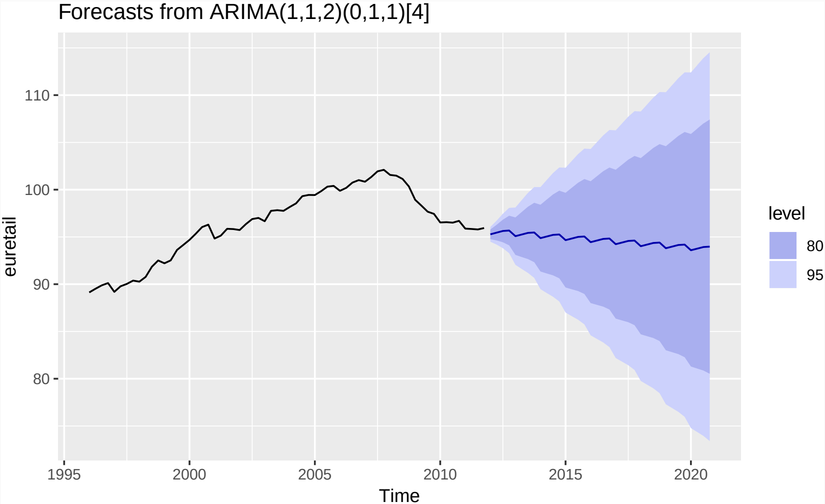
European quarterly retail trade

```r
checkresiduals(fit, plot=FALSE)
```

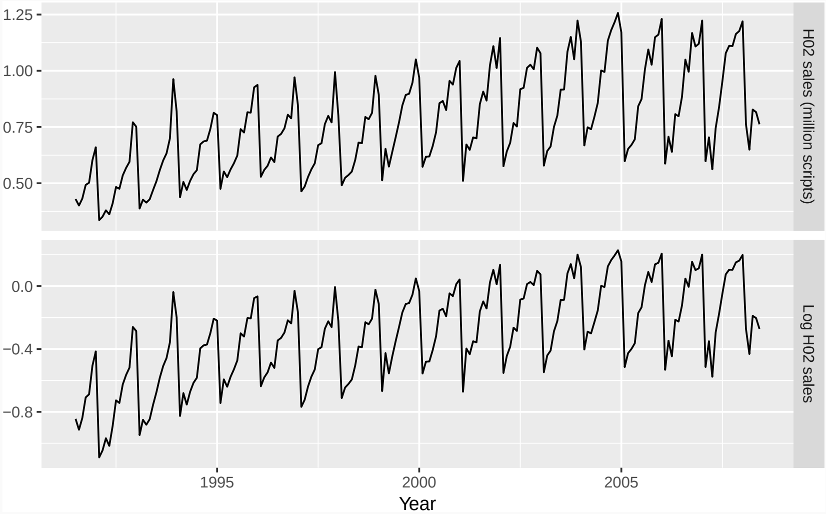
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(0,1,1)[4]
## Q* = 4.9, df = 4, p-value = 0.3
##
## Model df: 4.   Total lags used: 8
```

European quarterly retail trade

```r
forecast(fit, h=36) %>% autoplot()
```

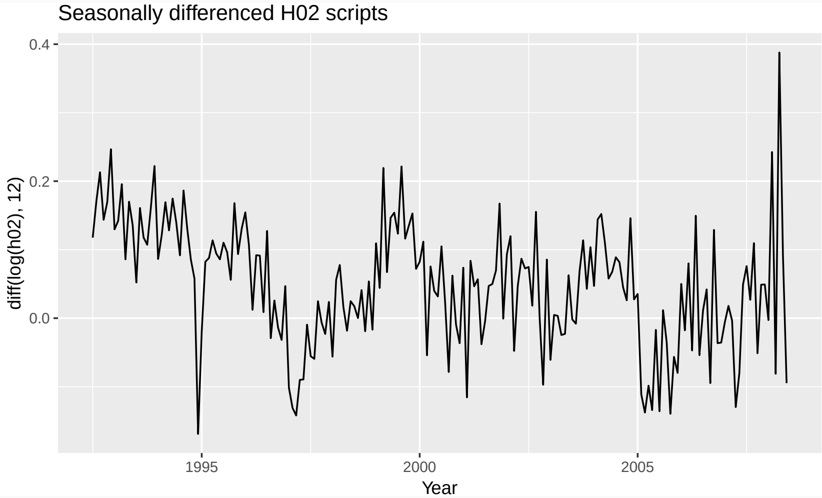


Cortecosteroid drug sales



Cortecosteroid drug sales

```
autoplot(diff(log(h02),12), xlab="Year",
  main="Seasonally differenced H02 scripts")
```



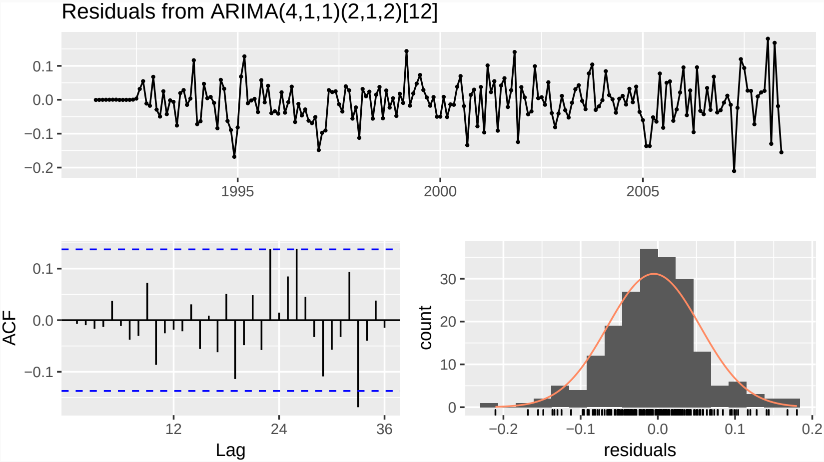
Cortecosteroid drug sales

```r
(fit <- auto.arima(h02, lambda=0, max.order=9,
  stepwise=FALSE, approximation=FALSE))
```

```
## Series: h02
## ARIMA(4,1,1)(2,1,2)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1    ar2    ar3     ar4     ma1   sar1
##       -0.042  0.210  0.202  -0.227  -0.742  0.621
## s.e.   0.217  0.181  0.114   0.081   0.207  0.242
##         sar2    sma1   sma2
##       -0.383  -1.202  0.496
## s.e.   0.118   0.249  0.214
##
```

Cortecosteroid drug sales

```r
checkresiduals(fit)
```



##

Cortecosteroid drug sales

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(4,1,1)(2,1,2)[12]
## Q* = 16, df = 15, p-value = 0.4
##
## Model df: 9.   Total lags used: 24
```

Cortecosteroid drug sales

Training data: July 1991 to June 2006

Test data: July 2006–June 2008

```r
getrmse <- function(x,h,...)
{
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fit <- Arima(train,...)
  fc <- forecast(fit,h=h)
  return(accuracy(fc,test)[2,"RMSE"])
}
getrmse(h02,h=24,order=c(3,0,0),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,0),lambda=0)
```

Cortecosteroid drug sales

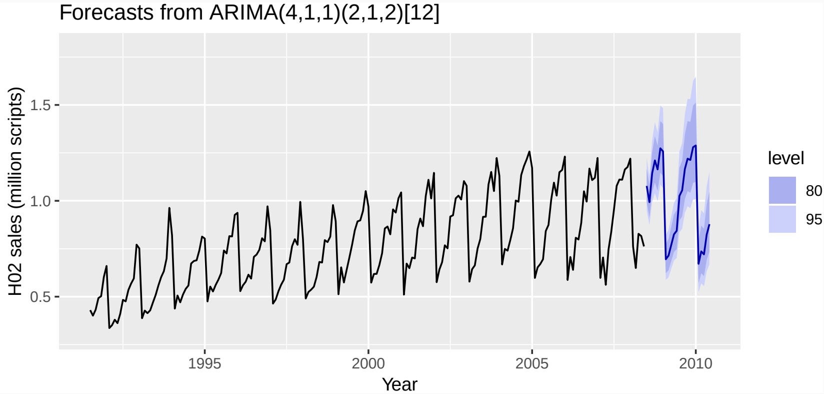
Model	RMSE
ARIMA(4,1,2)(2,1,2)[12]	0.0614
ARIMA(4,1,1)(2,1,2)[12]	0.0615
ARIMA(3,0,1)(0,1,2)[12]	0.0622
ARIMA(3,0,1)(1,1,1)[12]	0.0630
ARIMA(2,1,4)(0,1,1)[12]	0.0632
ARIMA(2,1,3)(0,1,1)[12]	0.0634
ARIMA(4,1,2)(1,1,2)[12]	0.0634
ARIMA(3,1,2)(2,1,2)[12]	0.0636
ARIMA(3,0,3)(0,1,1)[12]	0.0639
ARIMA(2,1,5)(0,1,1)[12]	0.0640
ARIMA(3,0,1)(0,1,1)[12]	0.0644
ARIMA(3,0,2)(0,1,1)[12]	0.0644
ARIMA(3,0,2)(2,1,0)[12]	0.0645

Cortecosteroid drug sales

- Models with lowest AICc values tend to give slightly better results than the other models.
- AICc comparisons must have the same orders of differencing. But RMSE test set comparisons can involve any models.
- Use the best model available, even if it does not pass all tests.

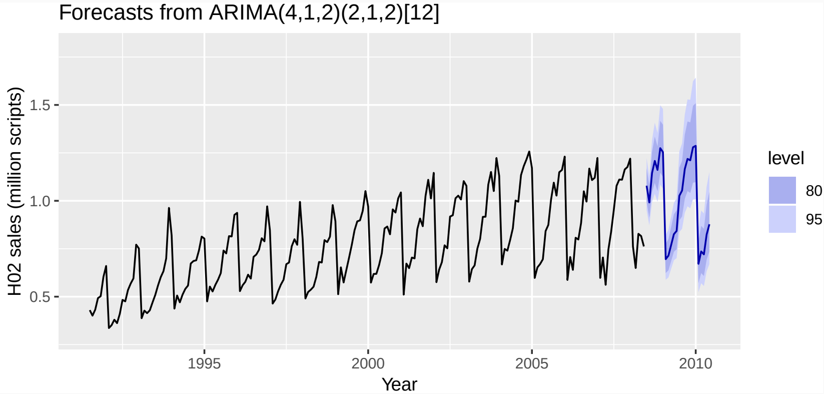
Cortecosteroid drug sales

```
fit <- Arima(h02, order=c(4,1,1), seasonal=c(2,1,2),
  lambda=0)
autoplot(forecast(fit)) + xlab("Year") +
  ylab("H02 sales (million scripts)") + ylim(0.3,1.8)
```



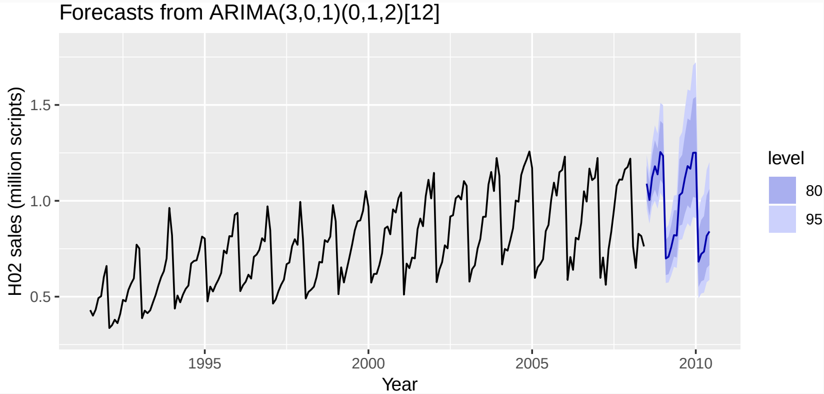
Cortecosteroid drug sales

```
fit <- Arima(h02, order=c(4,1,2), seasonal=c(2,1,2),
  lambda=0)
autoplot(forecast(fit)) + xlab("Year") +
  ylab("H02 sales (million scripts)") + ylim(0.3,1.8)
```



Cortecosteroid drug sales

```r
fit <- Arima(h02, order=c(3,0,1), seasonal=c(0,1,2),
  lambda=0)
autoplot(forecast(fit)) + xlab("Year") +
  ylab("H02 sales (million scripts)") + ylim(0.3,1.8)
```



Outline

Lab Session 17