MONASH University

MONASH BUSINESS SCHOOL

# ETC3550 Applied forecasting for business and economics

## Ch9. ARIMA models

OTexts.org/fpp3/

# Outline


1. Stationarity and differencing
2. Non-seasonal ARIMA models
3. Estimation and order selection
4. ARIMA modelling in R
5. Forecasting
6. Seasonal ARIMA models
7. ARIMA vs ETS

# ARIMA models

| | |
|---|---|
| **AR**: | autoregressive (lagged observations as inputs) |
| **I**: | integrated (differencing to make series stationary) |
| **MA**: | moving average (lagged errors as inputs) |

# ARIMA models

**AR**: autoregressive (lagged observations as inputs)

**I**: integrated (differencing to make series stationary)

**MA**: moving average (lagged errors as inputs)

An ARIMA model is rarely interpretable in terms of visible data structures like trend and seasonality. But it can capture a huge range of time series patterns.

# Outline


1 Stationarity and differencing

2 Non-seasonal ARIMA models

3 Estimation and order selection

4 ARIMA modelling in R

5 Forecasting

6 Seasonal ARIMA models

7 ARIMA vs ETS

# 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

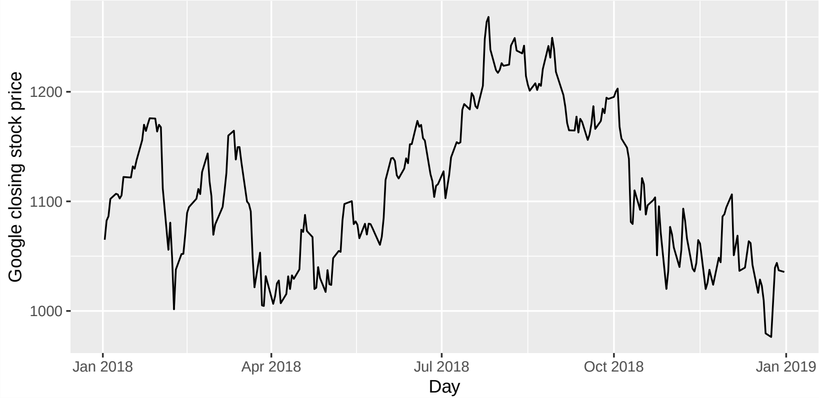
**Definition**

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

A **stationary series** is:

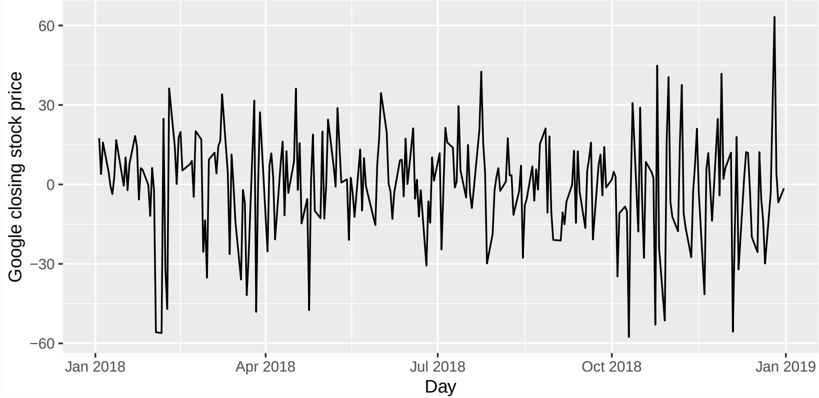
- roughly horizontal
- constant variance
- no patterns predictable in the long-term

# Stationary?



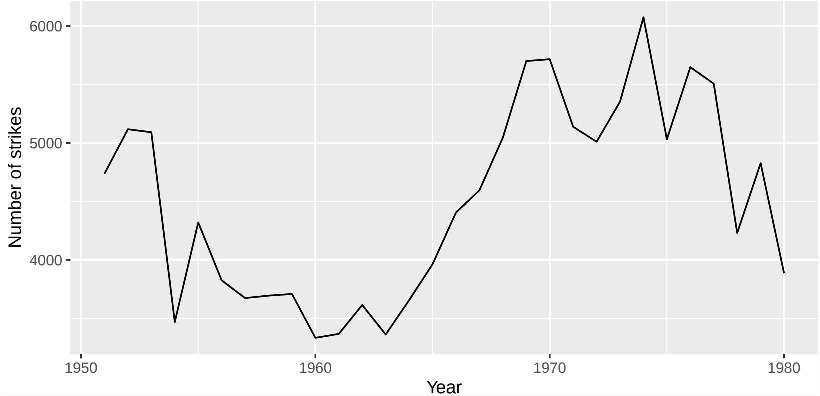
Google closing stock price
1200
1100
1000
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Jan 2019
Day

# Stationary?

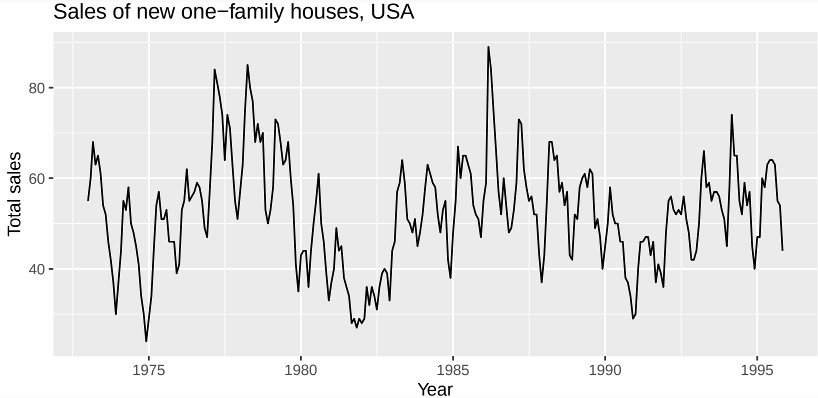


60
30
0
−30
−60
Google closing stock price
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Jan 2019
Day

# Stationary?

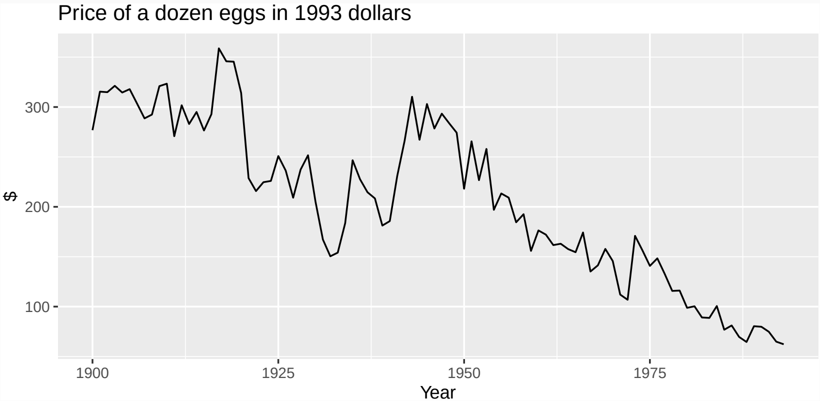


# Stationary?



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

# Stationary?

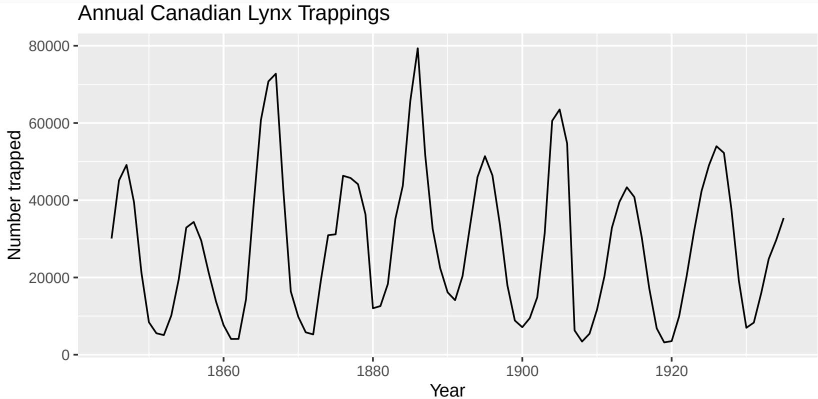


Price of a dozen eggs in 1993 dollars
300
200
100
$
1900
1925
1950
1975
Year

# Stationary?

Number of pigs slaughtered in Victoria
100
80
60
thousands
2010
2012
2014
2016
2018
Year

# Stationary?



Annual Canadian Lynx Trappings
80000
60000
40000
20000
0
Number trapped
1860
1880
1900
1920
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 $(y_t, \dots, y_{t+s})$ does not depend on $t$.

Transformations help to **stabilize the variance**.

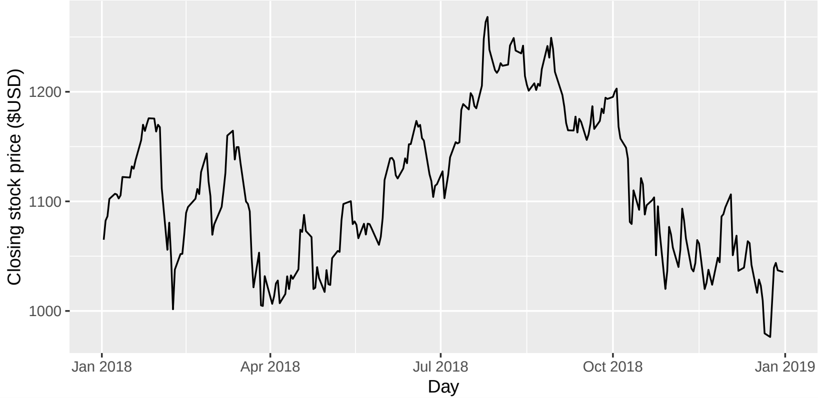
For ARIMA modelling, we also need to **stabilize the mean**.

# Non-stationarity in the mean

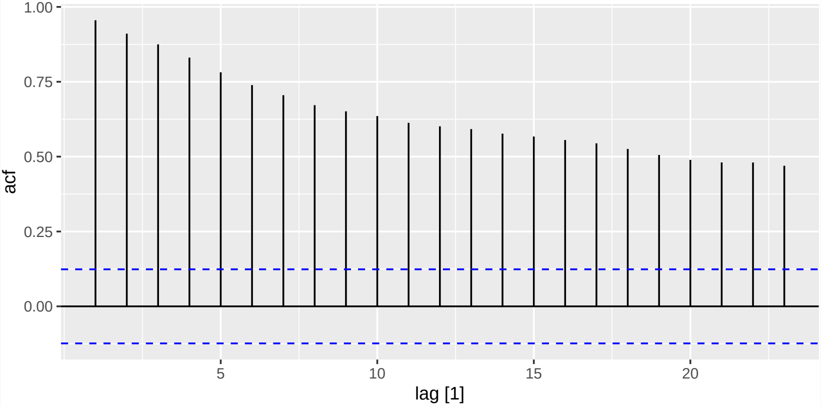
## Identifying non-stationary series

- 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.

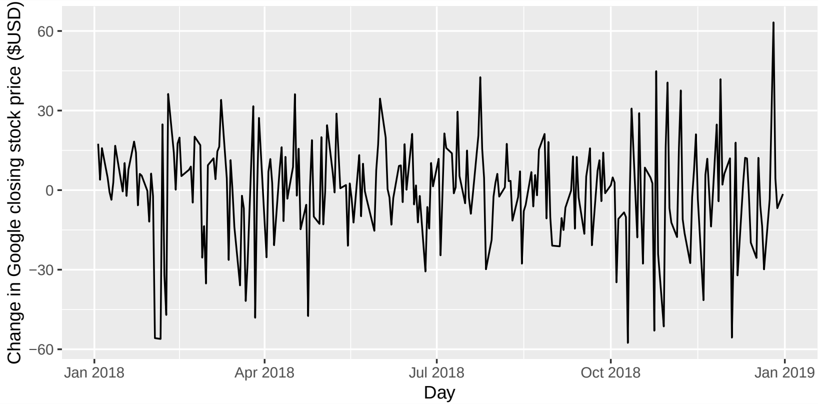
# Example: Google stock price



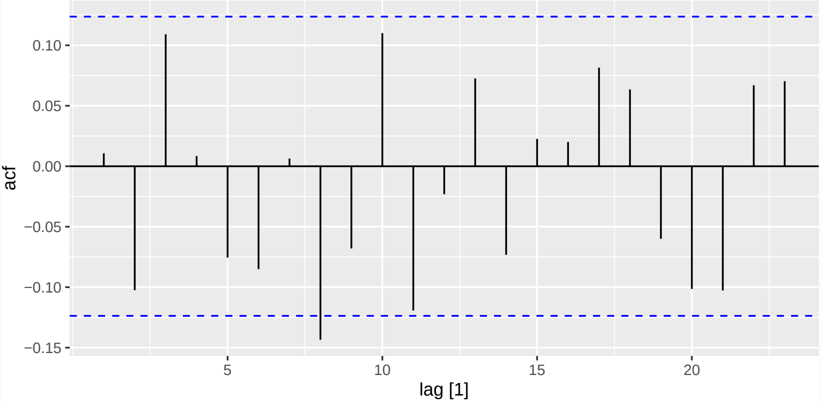
# Example: Google stock price



# Example: Google stock price



# Example: Google stock price



# 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.

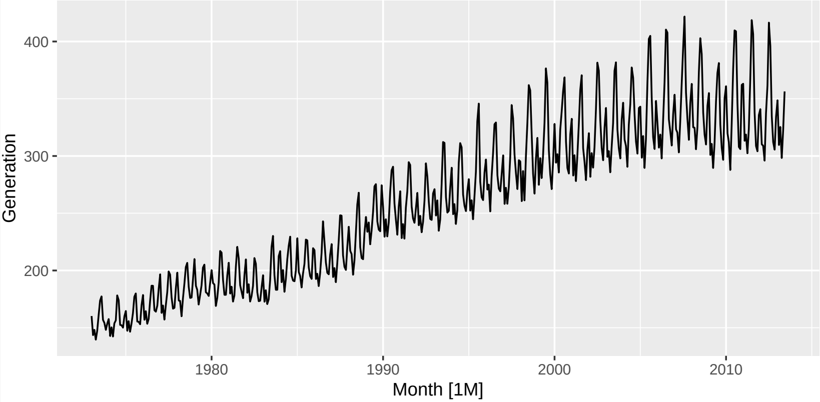
$$y'_t = y_t - y_{t-m}$$

where $m$ = number of seasons.

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

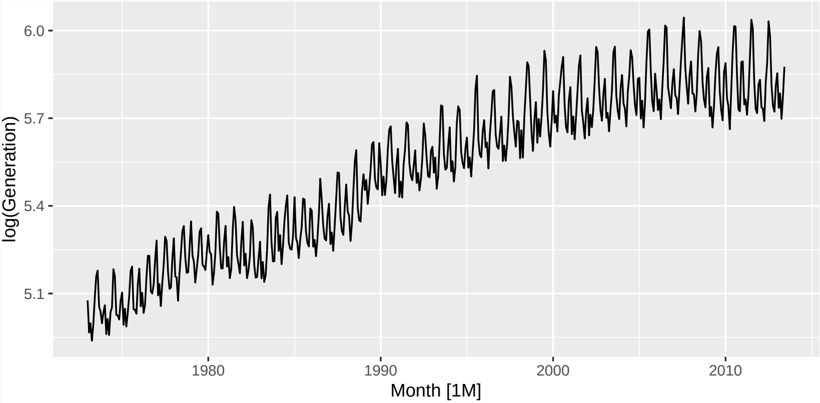
# Electricity production

```
usmelec %>% autoplot(
  Generation
)
```



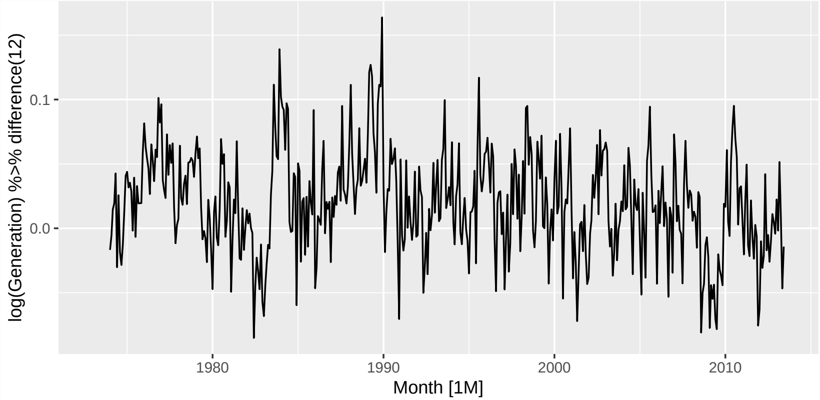
# Electricity production

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



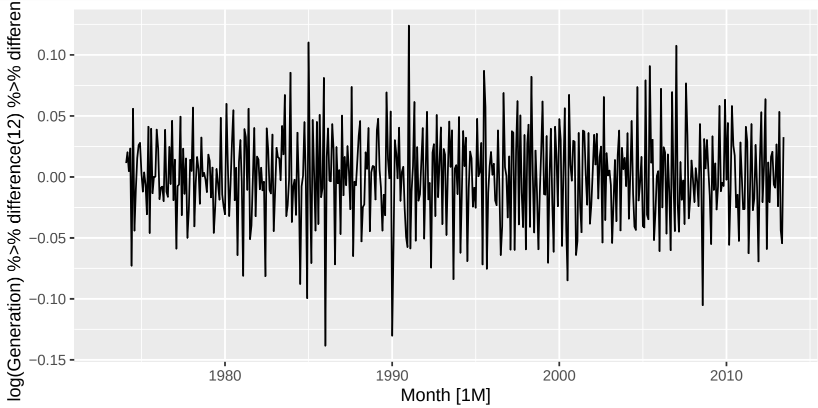
# Electricity production

```
usmelec %>% autoplot(
  log(Generation) %>% difference(12)
)
```



# Electricity production

```
usmelec %>% autoplot(
  log(Generation) %>% difference(12) %>% difference()
)
```



# 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.

# 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
google_2018 %>%
  features(Close, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   Symbol kpss_stat kpss_pvalue
##   <chr>      <dbl>       <dbl>
## 1 GOOG       0.573      0.0252
```

# KPSS test

```r
google_2018 %>%
  features(Close, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   Symbol kpss_stat kpss_pvalue
##   <chr>      <dbl>       <dbl>
## 1 GOOG       0.573      0.0252
```

```r
google_2018 %>%
  features(Close, unitroot_ndiffs)
```

```
## # A tibble: 1 x 2
##   Symbol ndiffs
##   <chr>   <int>
## 1 GOOG        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.

```
usmelec %>% mutate(log_gen = log(Generation)) %>%
  features(log_gen, list(unitroot_nsdiffs, feat_stl))
```

```
## # A tibble: 1 x 10
##   nsdiffs trend_strength seasonal_streng~ seasonal_peak_y~
##     <int>          <dbl>            <dbl>            <dbl>
## 1       1          0.994            0.941                7
## # ... with 6 more variables: seasonal_trough_year <dbl>,
## #   spikiness <dbl>, linearity <dbl>, curvature <dbl>,
## #   stl_e_acf1 <dbl>, stl_e_acf10 <dbl>
```

# Automatically selecting differences

```
usmelec %>% mutate(log_gen = log(Generation)) %>%
  features(log_gen, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

```
usmelec %>% mutate(d_log_gen = difference(log(Generation), 12)) %>%
  features(d_log_gen, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

# Your turn

For the `tourism` dataset, compute the total number of trips and find an appropriate differencing (after transformation if necessary) to obtain stationary data.

# 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**:

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

# 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 $d$th-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

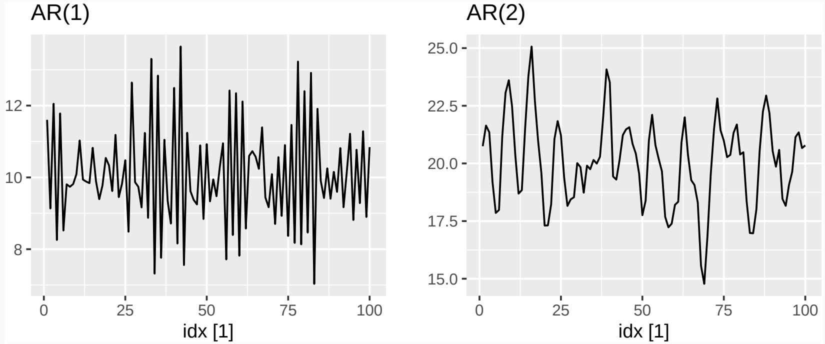

1. Stationarity and differencing
2. Non-seasonal ARIMA models
3. Estimation and order selection
4. ARIMA modelling in R
5. Forecasting
6. Seasonal ARIMA models
7. ARIMA vs ETS

# Autoregressive models

## Autoregressive (AR) models:

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

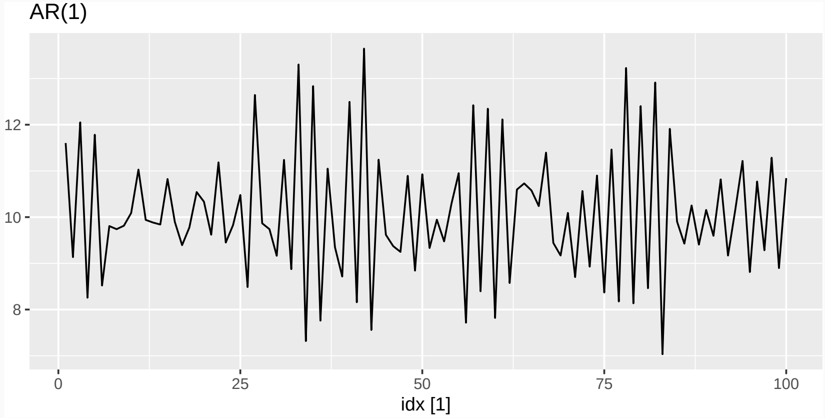
where $\varepsilon_t$ is white noise. This is a multiple regression with **lagged values** of $y_t$ as predictors.



# AR(1) model

$$y_t = 18 - 0.8y_{t-1} + \varepsilon_t$$

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



# AR(1) model

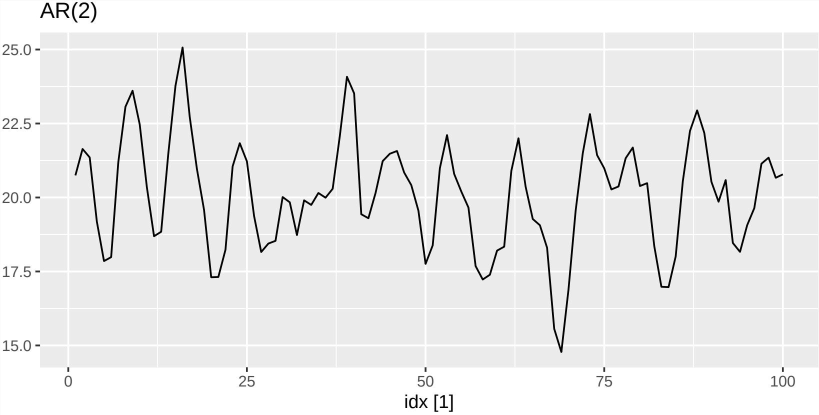
$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

- 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 $1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$ 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.

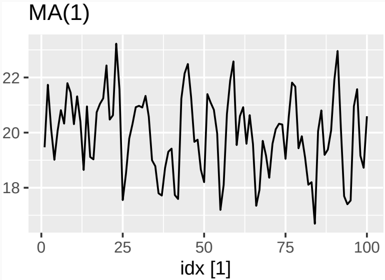
- For $p = 1$: $-1 < \phi_1 < 1$.
- For $p = 2$:
  $-1 < \phi_2 < 1 \qquad \phi_2 + \phi_1 < 1 \qquad \phi_2 - \phi_1 < 1$.
- More complicated conditions hold for $p \geq 3$.
- Estimation software takes care of this.

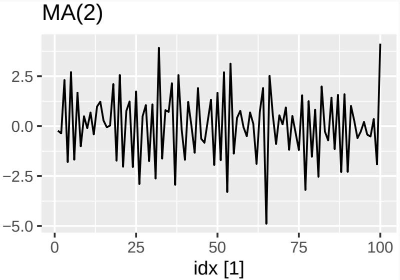
# Moving Average (MA) models

## 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 $\varepsilon_t$ is white noise. This is a multiple regression with ***past errors*** as predictors. *Don't confuse this with moving average smoothing!*

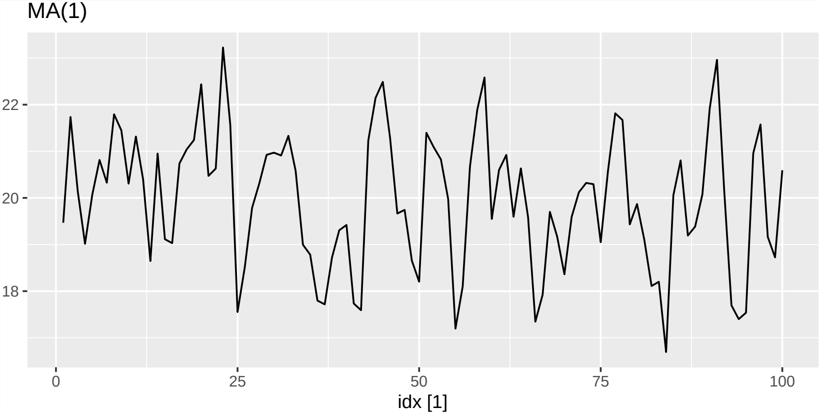


# MA(1) model

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

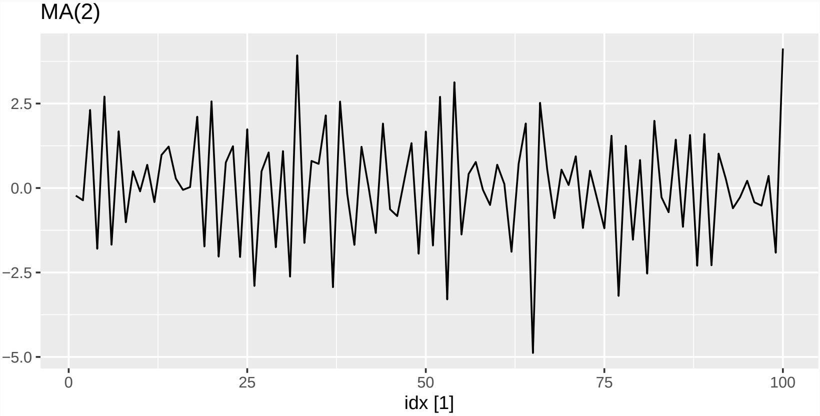
$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



# MA(2) model

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

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



# MA($\infty$) models

It is possible to write any stationary AR($p$) process as an MA($\infty$) process.

**Example: AR(1)**

$$\begin{aligned}
y_t &= \phi_1 y_{t-1} + \varepsilon_t \\
&= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\
&= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\
&= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\
&\dots
\end{aligned}$$

# MA($\infty$) models

It is possible to write any stationary AR($p$) process as an MA($\infty$) process.

**Example: AR(1)**

$$\begin{aligned} y_t &= \phi_1 y_{t-1} + \varepsilon_t \\ &= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\ &= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\ &\dots \end{aligned}$$

Provided $-1 < \phi_1 < 1$:

$$y_t = \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} + \phi_1^3 \varepsilon_{t-3} + \cdots$$

# Invertibility

- Any MA($q$) process can be written as an AR($\infty$) process if we impose some constraints on the MA parameters.
- Then the MA model is called "invertible".
- Invertible models have some mathematical properties that make them easier to use in practice.
- Invertibility of an ARIMA model is equivalent to forecastability of an ETS model.

# Invertibility

**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

**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 $q \geq 3$.
- Estimation software takes care of this.

# ARIMA models

## Autoregressive Moving Average models:

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

# ARIMA models

## Autoregressive Moving Average models:

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

- 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:**

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

- 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:

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

- 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**

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

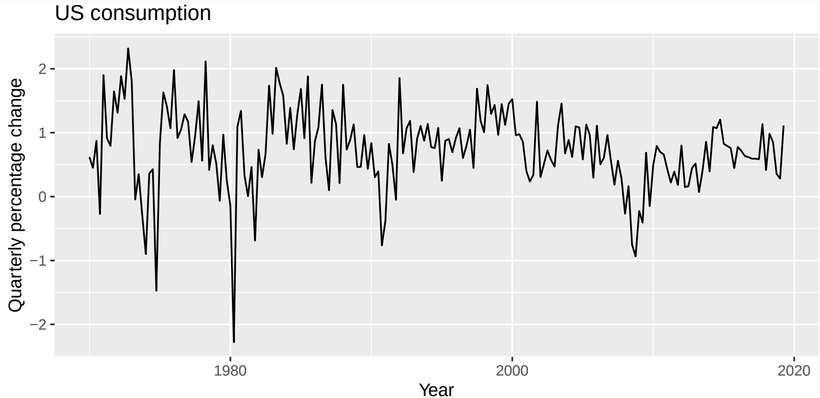
**Mean form**

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

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

# US consumption expenditure

```
us_change %>% autoplot(Consumption) +
  xlab("Year") +
  ylab("Quarterly percentage change") +
  ggtitle("US consumption")
```



# US personal consumption

```r
fit <- us_change %>% model(arima = ARIMA(Consumption ~ PDQ(0,0,0)))
report(fit)
```

```
## Series: Consumption
## Model: ARIMA(1,0,3) w/ mean
##
## Coefficients:
##          ar1      ma1     ma2     ma3  constant
##       0.5731  -0.3617  0.0925  0.1934    0.3160
## s.e.  0.1503   0.1607  0.0787  0.0824    0.0371
##
## sigma^2 estimated as 0.3334:  log likelihood=-169.9
## AIC=351.8   AICc=352.2   BIC=371.5
```

# US personal consumption

```r
fit <- us_change %>% model(arima = ARIMA(Consumption ~ PDQ(0,0,0)))
report(fit)
```

```
## Series: Consumption
## Model: ARIMA(1,0,3) w/ mean
##
## Coefficients:
##          ar1      ma1     ma2     ma3  constant
##       0.5731  -0.3617  0.0925  0.1934    0.3160
## s.e.  0.1503   0.1607  0.0787  0.0824    0.0371
##
## sigma^2 estimated as 0.3334:  log likelihood=-169.9
## AIC=351.8   AICc=352.2   BIC=371.5
```

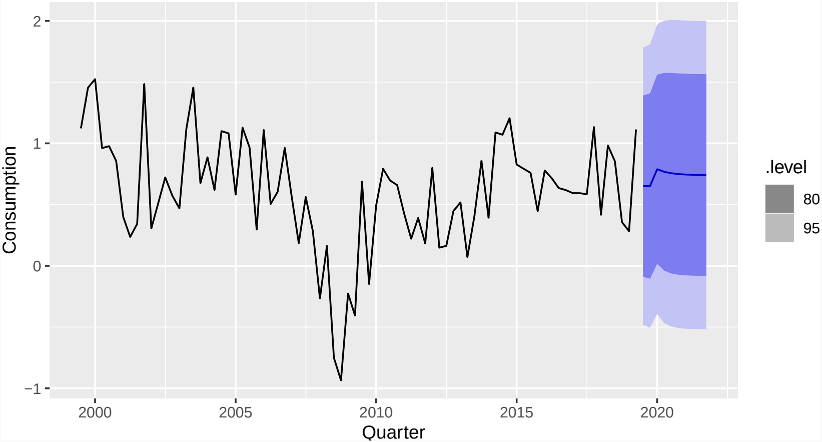
## ARIMA(1,0,3) model:

$$y_t = 0.316 + 0.573 y_{t-1} - 0.362\varepsilon_{t-1} + 0.0925\varepsilon_{t-2} + 0.193\varepsilon_{t-3} + \varepsilon_t,$$

where $\varepsilon_t$ is white noise with a standard deviation of $0.577 = \sqrt{0.333}$.

# US personal consumption

```
fit %>% forecast(h=10) %>%
  autoplot(tail(us_change, 80))
```



# Understanding ARIMA models

- If $c = 0$ and $d = 0$, the long-term forecasts will go to zero.
- If $c = 0$ and $d = 1$, the long-term forecasts will go to a non-zero constant.
- If $c = 0$ and $d = 2$, the long-term forecasts will follow a straight line.
- If $c \neq 0$ and $d = 0$, the long-term forecasts will go to the mean of the data.
- If $c \neq 0$ and $d = 1$, the long-term forecasts will follow a straight line.
- If $c \neq 0$ and $d = 2$, the long-term forecasts will follow a quadratic trend.

# Understanding ARIMA models

## 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.

## 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

1 Stationarity and differencing

2 Non-seasonal ARIMA models

3 Estimation and order selection

4 ARIMA modelling in R

5 Forecasting

6 Seasonal ARIMA models

7 ARIMA vs ETS

# 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()` function allows CLS or MLE estimation.
- Non-linear optimization must be used in either case.
- Different software will give different estimates.

# Partial autocorrelations

**Partial autocorrelations** measure relationship between $y_t$ and $y_{t-k}$, when the effects of other time lags — $1, 2, 3, \dots, k-1$ — are removed.

# Partial autocorrelations

**Partial autocorrelations** measure relationship between $y_t$ and $y_{t-k}$, when the effects of other time lags — $1, 2, 3, \dots, k-1$ — are removed.

$$\begin{aligned}\alpha_k &= k\text{th partial autocorrelation coefficient}\\ &= \text{equal to the estimate of } \phi_k \text{ in regression:}\\ &\qquad y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_k y_{t-k}.\end{aligned}$$

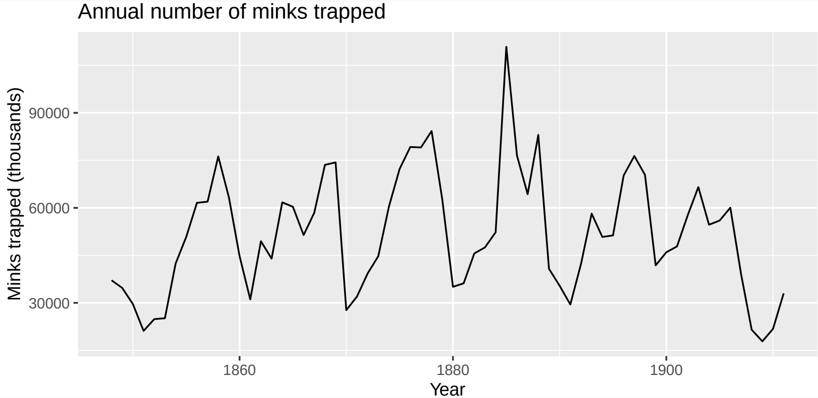
# Partial autocorrelations

**Partial autocorrelations** measure relationship between $y_t$ and $y_{t-k}$, when the effects of other time lags — $1, 2, 3, \dots, k-1$ — are removed.

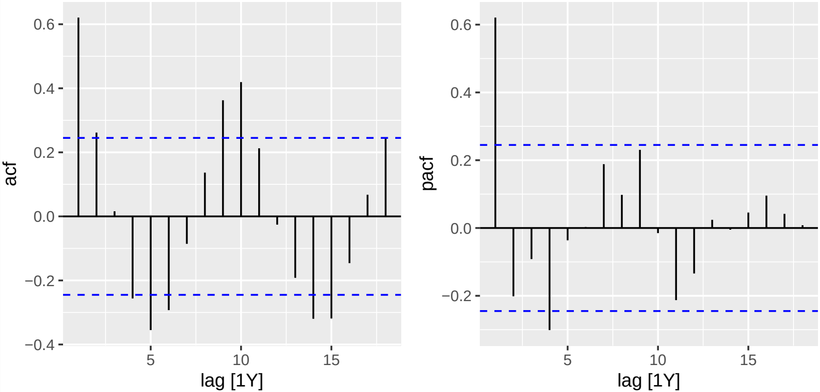
$$\begin{aligned} \alpha_k &= k\text{th partial autocorrelation coefficient} \\ &= \text{equal to the estimate of } \phi_k \text{ in regression:} \\ & \quad y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_k y_{t-k}. \end{aligned}$$

- Varying number of terms on RHS gives $\alpha_k$ for different values of $k$.
- $\alpha_1 = \rho_1$
- same critical values of $\pm 1.96/\sqrt{T}$ as for ACF.
- Last significant $\alpha_k$ indicates the order of an AR model.

# Example: Mink trapping

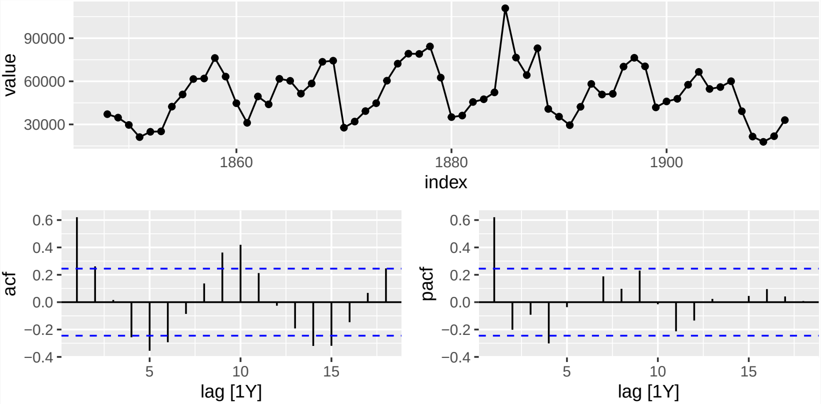


# Example: Mink trapping



# Example: Mink trapping

```r
mink %>% gg_tsdisplay(value, plot_type='partial')
```



# ACF and PACF interpretation

**AR(1)**

$$\rho_k = \phi_1^k \qquad \text{for } k = 1, 2, \ldots;$$

$$\alpha_1 = \phi_1 \qquad \alpha_k = 0 \qquad \text{for } k = 2, 3, \ldots.$$

So we have an AR(1) model when

- autocorrelations exponentially decay
- there is a single significant partial autocorrelation.

# ACF and PACF interpretation

**AR($p$)**

- ACF dies out in an exponential or damped sine-wave manner
- PACF has all zero spikes beyond the $p$th spike

So we have an AR($p$) model when

- the ACF is exponentially decaying or sinusoidal
- there is a significant spike at lag $p$ in PACF, but none beyond $p$

# ACF and PACF interpretation

**MA(1)**

$$\rho_1 = \theta_1 \qquad \rho_k = 0 \qquad \text{for } k = 2, 3, \dots;$$
$$\alpha_k = -(-\theta_1)^k$$

So we have an MA(1) model when

- the PACF is exponentially decaying and
- there is a single significant spike in ACF

# ACF and PACF interpretation

**MA($q$)**

- PACF dies out in an exponential or damped sine-wave manner
- ACF has all zero spikes beyond the $q$th spike

So we have an MA($q$) model when

- the PACF is exponentially decaying or sinusoidal
- there is a significant spike at lag $q$ in ACF, but none beyond $q$

# 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:**

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p + q + k + 1).$$

# 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:**

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p+q+k+1).$$

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

# Outline


1 Stationarity and differencing

2 Non-seasonal ARIMA models

3 Estimation and order selection

4 ARIMA modelling in R

5 Forecasting

6 Seasonal ARIMA models

7 ARIMA vs ETS

# How does 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 ARIMA() work?

$$\text{AICc} = -2\log(L) + 2(p+q+k+1)\left[1 + \frac{(p+q+k+2)}{T-p-q-k-2}\right].$$

where $L$ is the maximised likelihood fitted to the *differenced* data, $k = 1$ if $c \neq 0$ and $k = 0$ otherwise.

# How does ARIMA() work?

$$\text{AICc} = -2\log(L) + 2(p+q+k+1)\left[1 + \frac{(p+q+k+2)}{T-p-q-k-2}\right].$$

where $L$ is the maximised likelihood fitted to the *differenced* data, $k = 1$ if $c \neq 0$ and $k = 0$ otherwise.

**Step1:** 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 ARIMA() work?

$$\text{AICc} = -2\log(L) + 2(p+q+k+1)\left[1 + \frac{(p+q+k+2)}{T-p-q-k-2}\right].$$

where $L$ is the maximised likelihood fitted to the *differenced* data, $k = 1$ if $c \neq 0$ and $k = 0$ otherwise.

**Step1:** Select current model (with smallest AICc) from:
ARIMA$(2, d, 2)$
ARIMA$(0, d, 0)$
ARIMA$(1, d, 0)$
ARIMA$(0, d, 1)$

**Step 2:** Consider variations of current model:

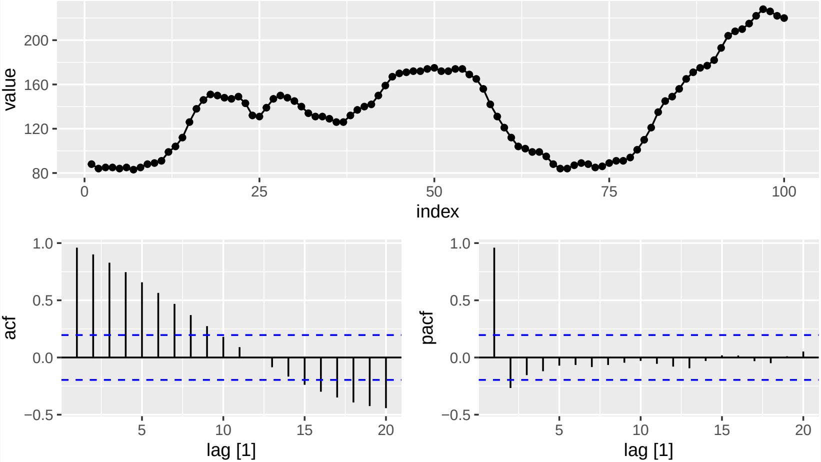
- vary one of $p, q$, from current model by $\pm 1$;
- $p, q$ both vary from current model by $\pm 1$;
- Include/exclude $c$ from current model.

Model with lowest AICc becomes current model.

**Repeat Step 2 until no lower AICc can be found.**

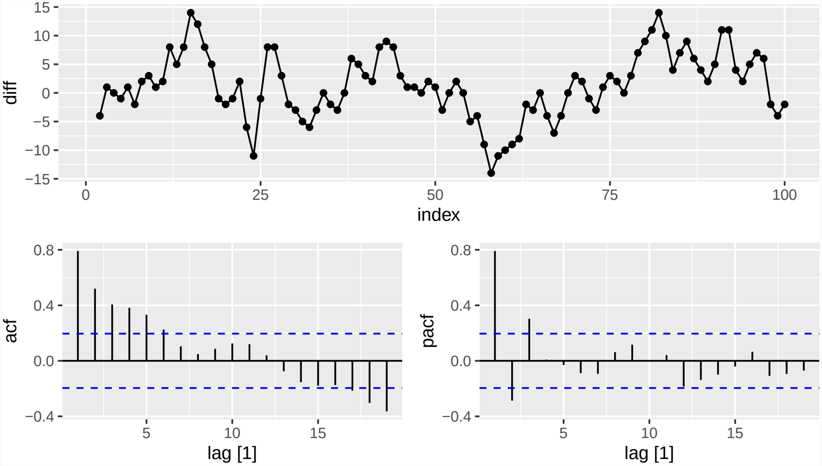
# Choosing your own model

```
web_usage <- as_tsibble(WWWusage)
web_usage %>% gg_tsdisplay(value, plot_type = 'partial')
```



# Choosing your own model

```
web_usage %>% mutate(diff = difference(value)) %>%
  gg_tsdisplay(diff, plot_type = 'partial')
```



# Choosing your own model

```r
fit <- web_usage %>%
  model(arima = ARIMA(value ~ pdq(3, 1, 0)))
report(fit)
```

```
## Series: value
## Model: ARIMA(3,1,0)
##
## Coefficients:
##          ar1      ar2     ar3
##        1.151  -0.6612  0.3407
## s.e.   0.095   0.1353  0.0941
##
## sigma^2 estimated as 9.656:  log likelihood=-252
## AIC=512   AICc=512.4   BIC=522.4
```

# Choosing your own model

```
web_usage %>%
  model(ARIMA(value ~ pdq(d=1))) %>%
report()
```

```
## Series: value
## Model: ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.6504  0.5256
## s.e.  0.0842  0.0896
##
## sigma^2 estimated as 9.995:  log likelihood=-254.2
## AIC=514.3   AICc=514.5   BIC=522.1
```

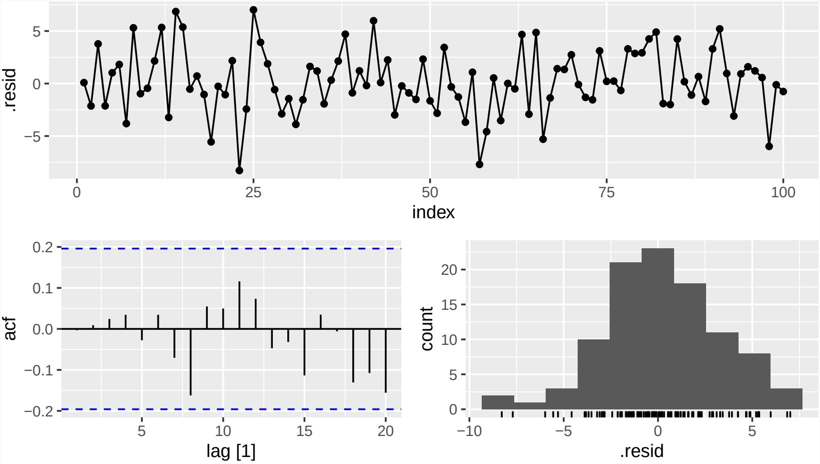
# Choosing your own model

```r
web_usage %>%
  model(ARIMA(value ~ pdq(d=1),
    stepwise = FALSE, approximation = FALSE)) %>%
  report()
```

```
## Series: value
## Model: ARIMA(3,1,0)
##
## Coefficients:
##          ar1      ar2     ar3
##        1.151  -0.6612  0.3407
## s.e.   0.095   0.1353  0.0941
##
## sigma^2 estimated as 9.656:  log likelihood=-252
## AIC=512   AICc=512.4   BIC=522.4
```

# Choosing your own model

```
gg_tsresiduals(fit)
```



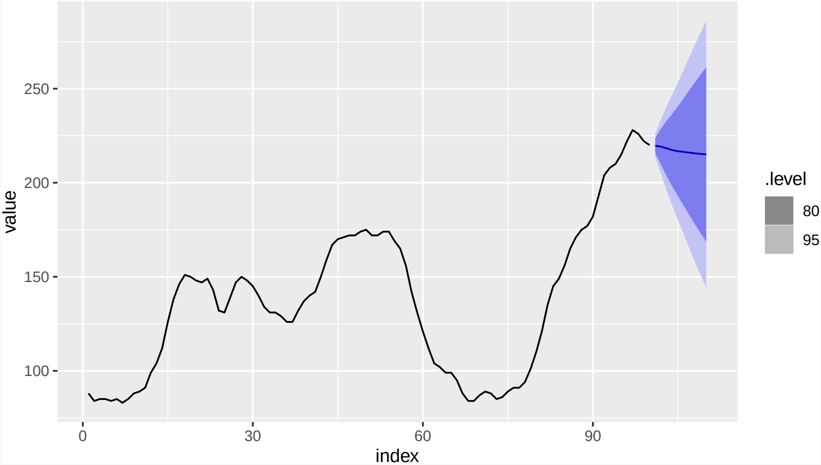
# Choosing your own model

```r
augment(fit) %>%
  features(.resid, ljung_box, lag = 10, dof = 3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 arima     4.49     0.722
```

# Choosing your own model

```
fit %>% forecast(h = 10) %>%
  autoplot(web_usage)
```



# Modelling procedure with `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. If the data are non-stationary: take first differences of the data until the data are stationary.
4. Examine the ACF/PACF: Is an AR($p$) or MA($q$) model appropriate?
5. Try your chosen model(s), and use the AICc to search for a better model.
6. 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.
7. Once the residuals look like white noise, calculate forecasts.

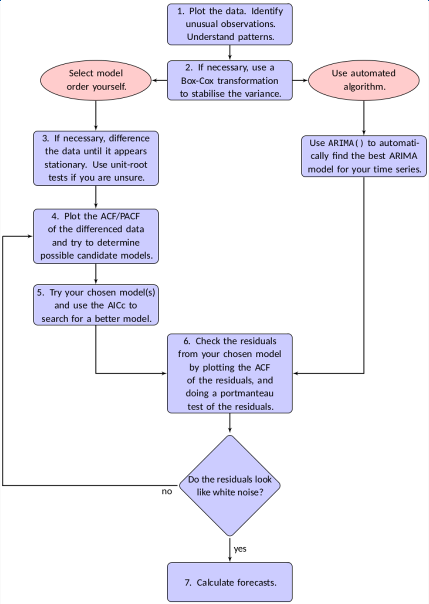
# Automatic modelling procedure with `ARIMA()`

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 `ARIMA` to automatically select a model.

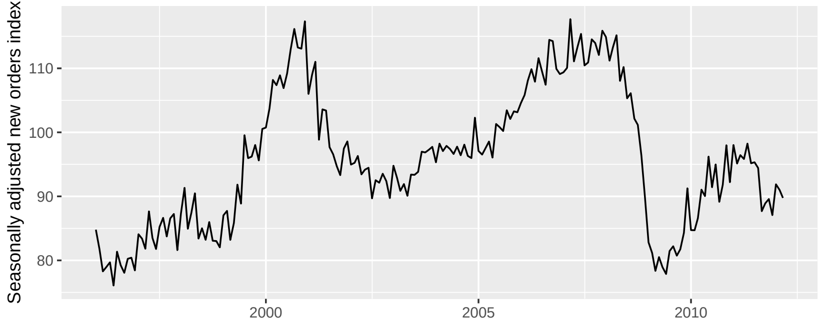
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.

# Modelling procedure



# Seasonally adjusted electrical equipment

```
elecequip <- as_tsibble(fpp2::elecequip)
dcmp <- elecequip %>%
  model(STL(value ~ season(window = "periodic"))) %>%
  components() %>% select(-.model)
dcmp %>% as_tsibble %>%
  autoplot(season_adjust) + xlab("Year") +
  ylab("Seasonally adjusted new orders index")
```

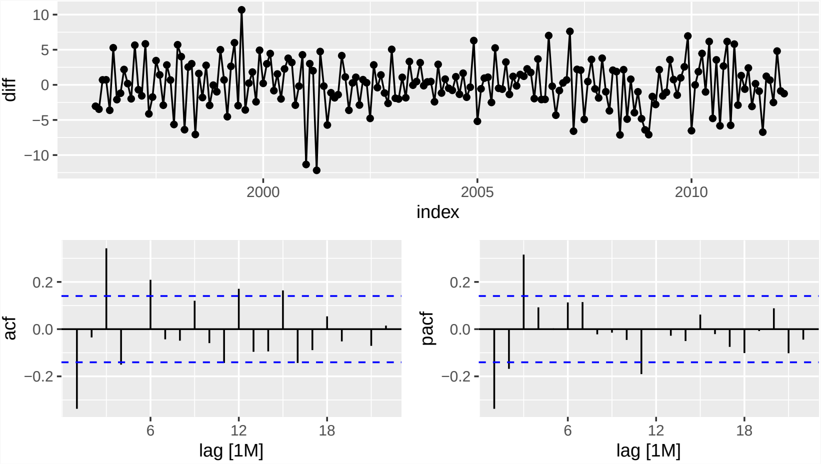


# Seasonally adjusted electrical equipment

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

```
dcmp %>% mutate(diff = difference(season_adjust)) %>%
  gg_tsdisplay(diff, plot_type = 'partial')
```



# Seasonally adjusted electrical equipment

4. PACF is suggestive of AR(3). So initial candidate model is ARIMA(3,1,0). No other obvious candidates.
5. Fit ARIMA(3,1,0) model along with variations: ARIMA(4,1,0), ARIMA(2,1,0), ARIMA(3,1,1), etc. ARIMA(3,1,1) has smallest AICc value.

# Seasonally adjusted electrical equipment

```r
fit <- dcmp %>%
  model(arima = ARIMA(season_adjust))
report(fit)
```

```
## Series: season_adjust
## Model: ARIMA(3,1,0)
##
## Coefficients:
##           ar1      ar2     ar3
##       -0.3418  -0.0426  0.3185
## s.e.   0.0681   0.0725  0.0682
##
## sigma^2 estimated as 9.639:  log likelihood=-493.8
## AIC=995.6   AICc=995.8   BIC=1009
```

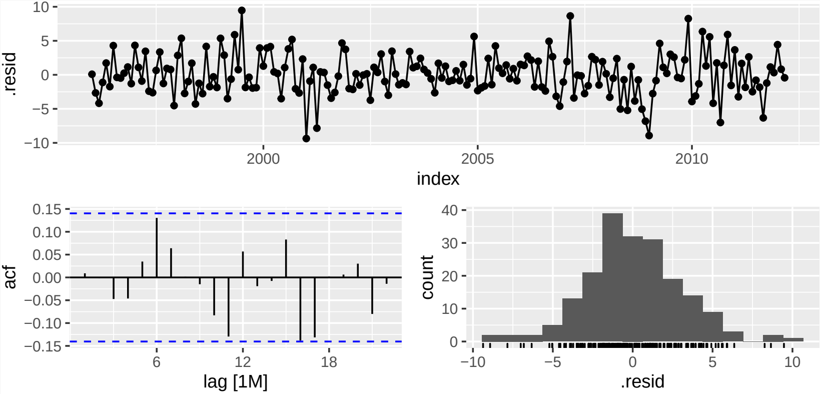
# Seasonally adjusted electrical equipment

```r
fit <- dcmp %>%
  model(arima = ARIMA(season_adjust, approximation=FALSE))
report(fit)
```

```
## Series: season_adjust
## Model: ARIMA(3,1,1)
##
## Coefficients:
##          ar1     ar2     ar3      ma1
##       0.0044  0.0916  0.3698  -0.3921
## s.e.  0.2201  0.0984  0.0669   0.2426
##
## sigma^2 estimated as 9.577:  log likelihood=-492.7
## AIC=995.4   AICc=995.7   BIC=1012
```

# Seasonally adjusted electrical equipment

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

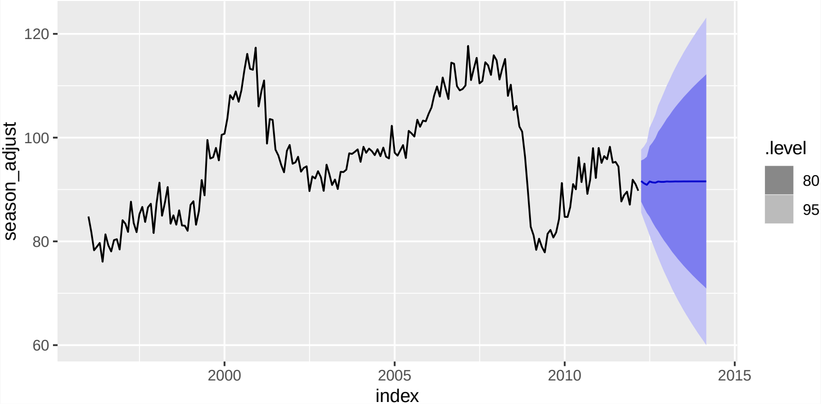


# Seasonally adjusted electrical equipment

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 arima     24.0     0.241
```

# Seasonally adjusted electrical equipment

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



# Outline


1. Stationarity and differencing
2. Non-seasonal ARIMA models
3. Estimation and order selection
4. ARIMA modelling in R
5. Forecasting
6. Seasonal ARIMA models
7. ARIMA vs ETS

# Point forecasts

1. Rearrange ARIMA equation so $y_t$ is on LHS.
2. Rewrite equation by replacing $t$ by $T + h$.
3. On RHS, replace future observations by their forecasts, future errors by zero, and past errors by corresponding residuals.

Start with $h = 1$. Repeat for $h = 2, 3, \ldots$.

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B) y_t = (1 + \theta_1 B)\varepsilon_t,$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1-\phi_1B-\phi_2B^2-\phi_3B^3)(1-B)y_t = (1+\theta_1B)\varepsilon_t,$$

$$\begin{aligned}\left[1-(1+\phi_1)B+(\phi_1-\phi_2)B^2+(\phi_2-\phi_3)B^3+\phi_3B^4\right]y_t \\ = (1+\theta_1B)\varepsilon_t,\end{aligned}$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B)y_t = (1 + \theta_1 B)\varepsilon_t,$$

$$\begin{aligned}\left[1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4\right] y_t \\ = (1 + \theta_1 B)\varepsilon_t,\end{aligned}$$

$$\begin{aligned}y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} \\ + \phi_3 y_{t-4} = \varepsilon_t + \theta_1 \varepsilon_{t-1}.\end{aligned}$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B) y_t = (1 + \theta_1 B)\varepsilon_t,$$

$$\left[1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4\right] y_t \\ = (1 + \theta_1 B)\varepsilon_t,$$

$$y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} \\ + \phi_3 y_{t-4} = \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

# Point forecasts (h=1)

$$\begin{aligned}y_t = (1+\phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1\varepsilon_{t-1}.\end{aligned}$$

# Point forecasts (h=1)

$$
\begin{aligned}
y_t = (1+\phi_1)y_{t-1} - (\phi_1-\phi_2)y_{t-2} - (\phi_2-\phi_3)y_{t-3} \\
- \phi_3 y_{t-4} + \varepsilon_t + \theta_1\varepsilon_{t-1}.
\end{aligned}
$$

**ARIMA(3,1,1) forecasts: Step 2**

$$
\begin{aligned}
y_{T+1} = (1+\phi_1)y_T - (\phi_1-\phi_2)y_{T-1} - (\phi_2-\phi_3)y_{T-2} \\
- \phi_3 y_{T-3} + \varepsilon_{T+1} + \theta_1\varepsilon_T.
\end{aligned}
$$

# Point forecasts (h=1)

$$\begin{aligned} y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}. \end{aligned}$$

## ARIMA(3,1,1) forecasts: Step 2

$$\begin{aligned} y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} \\ - \phi_3 y_{T-3} + \varepsilon_{T+1} + \theta_1 \varepsilon_T. \end{aligned}$$

## ARIMA(3,1,1) forecasts: Step 3

$$\begin{aligned} \hat{y}_{T+1|T} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} \\ - \phi_3 y_{T-3} + \theta_1 e_T. \end{aligned}$$

# Point forecasts (h=2)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

# Point forecasts (h=2)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1\varepsilon_{t-1}.$$

**ARIMA(3,1,1) forecasts: Step 2**

$$y_{T+2} = (1 + \phi_1)y_{T+1} - (\phi_1 - \phi_2)y_{T} - (\phi_2 - \phi_3)y_{T-1} \\ - \phi_3 y_{T-2} + \varepsilon_{T+2} + \theta_1\varepsilon_{T+1}.$$

# Point forecasts (h=2)

$$\begin{aligned}y_t = (1+\phi_1)y_{t-1} - (\phi_1-\phi_2)y_{t-2} - (\phi_2-\phi_3)y_{t-3}\\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1\varepsilon_{t-1}.\end{aligned}$$

## ARIMA(3,1,1) forecasts: Step 2

$$\begin{aligned}y_{T+2} = (1+\phi_1)y_{T+1} - (\phi_1-\phi_2)y_{T} - (\phi_2-\phi_3)y_{T-1}\\ - \phi_3 y_{T-2} + \varepsilon_{T+2} + \theta_1\varepsilon_{T+1}.\end{aligned}$$

## ARIMA(3,1,1) forecasts: Step 3

$$\begin{aligned}\hat{y}_{T+2|T} = (1+\phi_1)\hat{y}_{T+1|T} - (\phi_1-\phi_2)y_{T} - (\phi_2-\phi_3)y_{T-1}\\ - \phi_3 y_{T-2}.\end{aligned}$$

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$$

where $v_{T+h|T}$ is estimated forecast variance.

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$$

where $v_{T+h|T}$ is estimated forecast variance.

- $v_{T+1|T} = \hat{\sigma}^2$ for all ARIMA models regardless of parameters and orders.
- Multi-step prediction intervals for ARIMA(0,0,$q$):

$$y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[1 + \sum_{i=1}^{h-1} \theta_i^2\right], \qquad \text{for } h = 2, 3, \dots.$$

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$$

where $v_{T+h|T}$ is estimated forecast variance.

- Multi-step prediction intervals for ARIMA(0,0,$q$):

$$y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[1 + \sum_{i=1}^{h-1} \theta_i^2\right], \qquad \text{for } h = 2, 3, \ldots.$$

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$$

where $v_{T+h|T}$ is estimated forecast variance.

- Multi-step prediction intervals for ARIMA(0,0,$q$):

$$y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[1 + \sum_{i=1}^{h-1} \theta_i^2\right], \qquad \text{for } h = 2, 3, \ldots.$$

- AR(1): Rewrite as MA($\infty$) and use above result.
- Other models beyond scope of this subject.

# Prediction intervals

- Prediction intervals **increase in size with forecast horizon**.
- Prediction intervals can be difficult to calculate by hand
- 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

# Your turn

For the GDP data (from `global_economy`):

- fit a suitable ARIMA model to the logged data for all countries
- check the residual diagnostics for Australia;
- produce forecasts of your fitted model for Australia.

# Outline

1 Stationarity and differencing

2 Non-seasonal ARIMA models

3 Estimation and order selection

4 ARIMA modelling in R

5 Forecasting

6 Seasonal ARIMA models

7 ARIMA vs ETS

# 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)

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

(Non-seasonal AR(1))

(Seasonal AR(1))

(Non-seasonal difference)

(Seasonal difference)

(Non-seasonal MA(1))

(Seasonal MA(1))

# 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:

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

# 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 |

# Seasonal ARIMA models

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF.

**ARIMA(0,0,0)(0,0,1)$_{12}$ will show:**

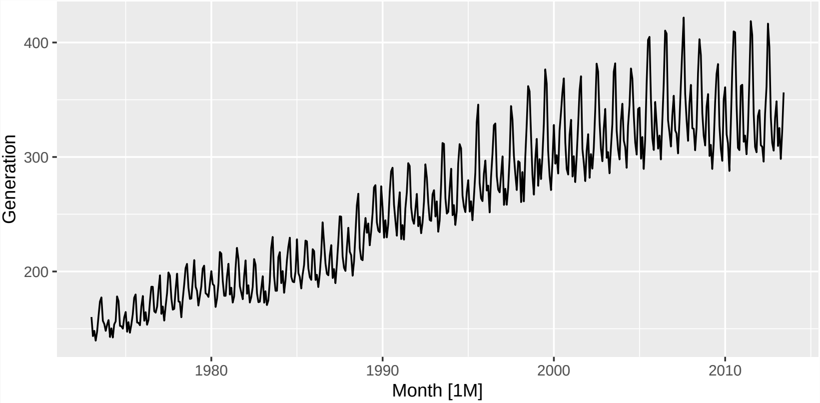
- a spike at lag 12 in the ACF but no other significant spikes.
- The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36, . . . .

**ARIMA(0,0,0)(1,0,0)$_{12}$ will show:**

- exponential decay in the seasonal lags of the ACF
- a single significant spike at lag 12 in the PACF.

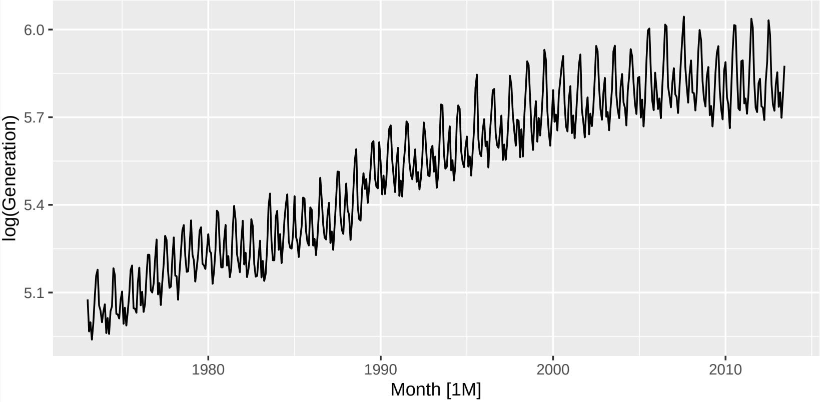
# Example: US electricity production

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



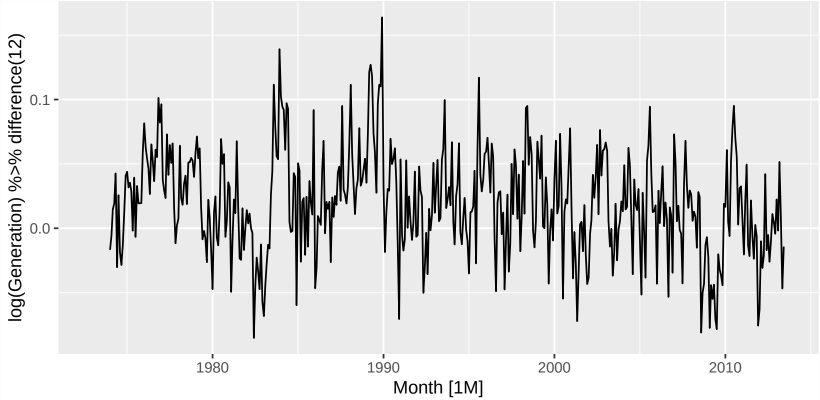
# Example: US electricity production

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



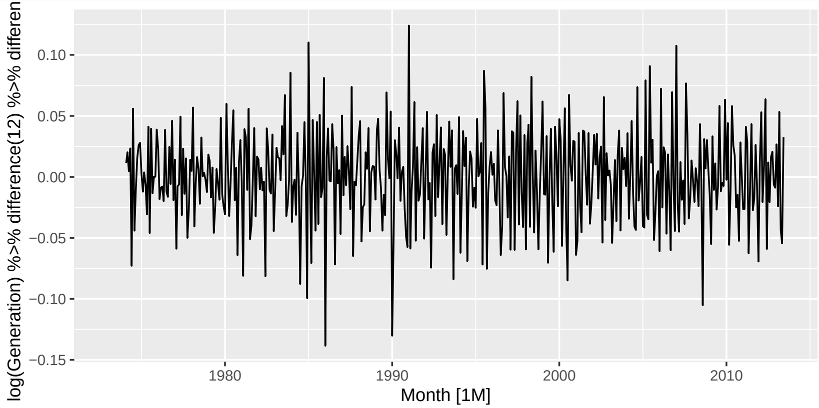
# Example: US electricity production

```
usmelec %>% autoplot(
  log(Generation) %>% difference(12)
)
```



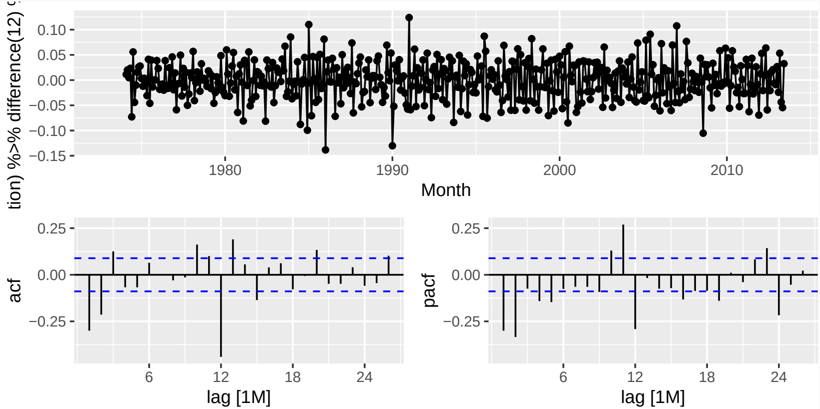
# Example: US electricity production

```
usmelec %>% autoplot(
  log(Generation) %>% difference(12) %>% difference()
)
```



# Example: US electricity production

```
usmelec %>% gg_tsdisplay(
  log(Generation) %>% difference(12) %>% difference(),
  plot_type = "partial")
```



# Example: US electricity production

- $d = 1$ and $D = 1$ seems necessary
- $P = 0$ and $Q = 1$ suggested by seasonal lags
- $p = 0$ and $q = 3$ suggested by non-seasonal lags.

# Example: US electricity production

```r
usmelec %>%
  model(arima = ARIMA(log(Generation) ~ pdq(0,1,3) + PDQ(0,1,1))) %>%
  report()
```

```
## Series: Generation
## Model: ARIMA(0,1,3)(0,1,1)[12]
## Transformation: log(.x)
##
## Coefficients:
##           ma1      ma2      ma3     sma1
##       -0.4266  -0.2496  -0.0439  -0.8358
## s.e.   0.0462   0.0516   0.0430   0.0262
##
## sigma^2 estimated as 0.0006904:  log likelihood=1045
## AIC=-2080   AICc=-2080   BIC=-2059
```

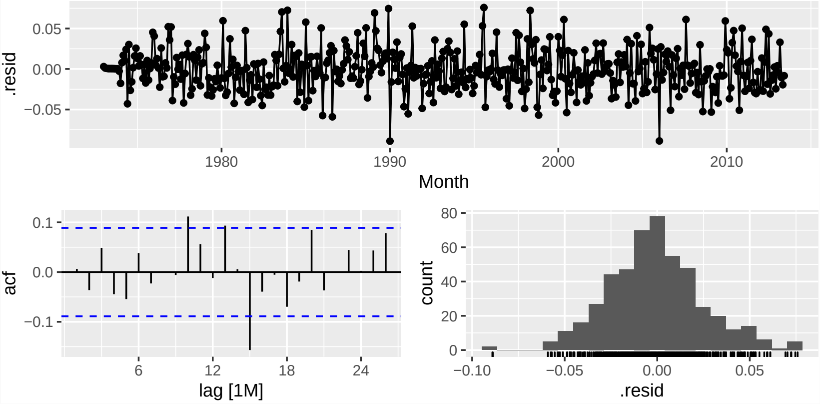
# Example: US electricity production

```
usmelec %>%
  model(arima = ARIMA(log(Generation))) %>%
  report()
```

```
## Series: Generation
## Model: ARIMA(1,1,1)(2,1,1)[12]
## Transformation: log(.x)
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1
##       0.4116  -0.8483  0.0100  -0.1017  -0.8204
## s.e.  0.0617   0.0348  0.0561   0.0529   0.0357
##
## sigma^2 estimated as 0.0006841:  log likelihood=1047
## AIC=-2082   AICc=-2082   BIC=-2057
```

# Example: US electricity production

```r
fit <- usmelec %>%
  model(arima = ARIMA(log(Generation)))
gg_tsresiduals(fit)
```



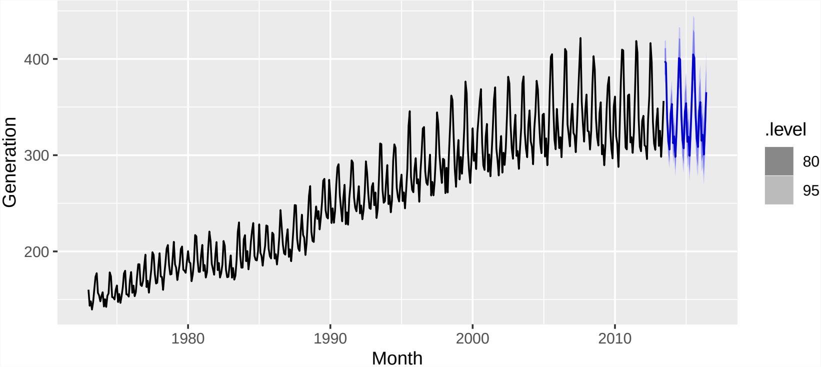
# Example: US electricity production

```r
augment(fit) %>%
  features(.resid, ljung_box, lag = 24, dof = 5)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 arima     38.7   0.00484
```

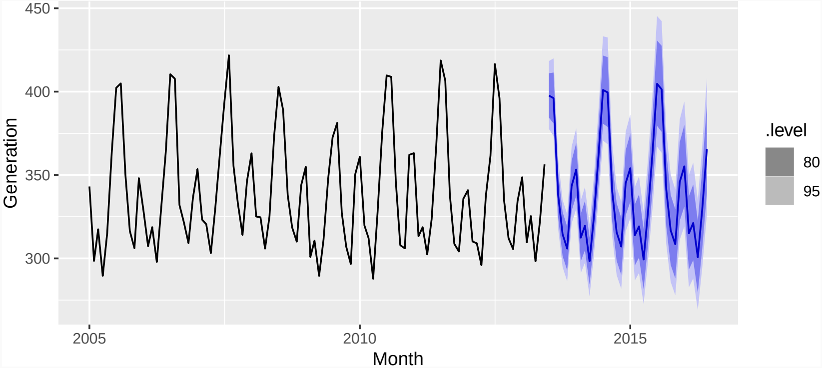
# Example: US electricity production

```r
usmelec %>%
  model(arima = ARIMA(log(Generation))) %>%
  forecast(h = "3 years") %>%
  autoplot(usmelec)
```

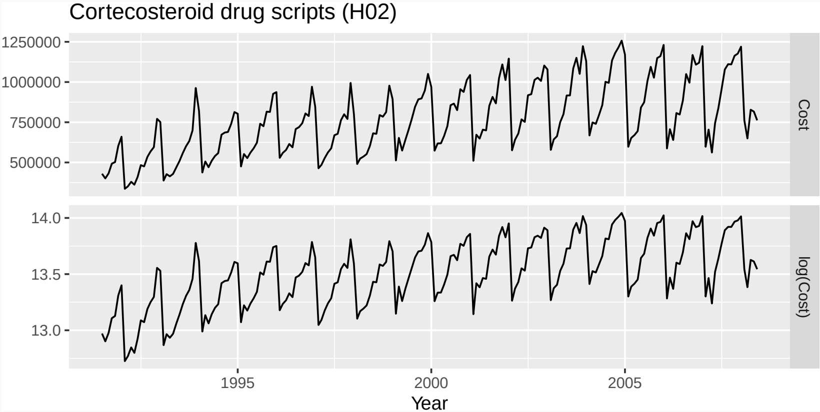


# Example: US electricity production

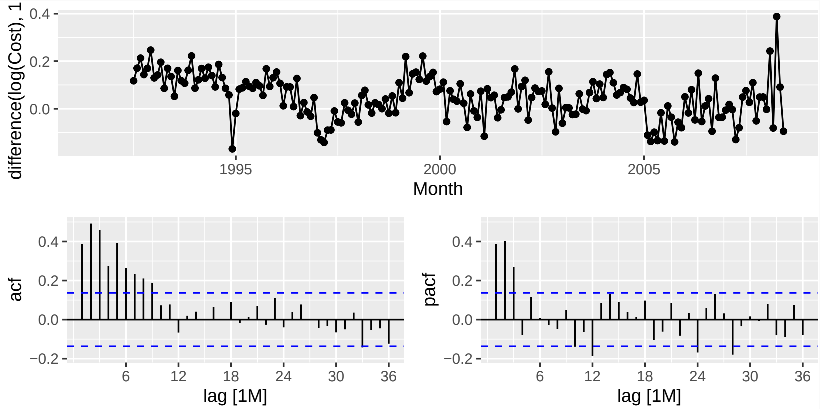
```r
usmelec %>%
  model(arima = ARIMA(log(Generation))) %>%
  forecast(h = "3 years") %>%
  autoplot(filter(usmelec, year(Month) >= 2005))
```



# Cortecosteroid drug sales



# Cortecosteroid drug sales



# Cortecosteroid drug sales

- Choose $D = 1$ and $d = 0$.
- Spikes in PACF at lags 12 and 24 suggest seasonal AR(2) term.
- Spikes in PACF suggests possible non-seasonal AR(3) term.
- Initial candidate model: ARIMA$(3,0,0)(2,1,0)_{12}$.

# Cortecosteroid drug sales

| .model | AICc |
|---|---|
| ARIMA(3,0,1)(0,1,2)[12] | -485.5 |
| ARIMA(3,0,1)(1,1,1)[12] | -484.3 |
| ARIMA(3,0,1)(0,1,1)[12] | -483.7 |
| ARIMA(3,0,1)(2,1,0)[12] | -476.3 |
| ARIMA(3,0,0)(2,1,0)[12] | -475.1 |
| ARIMA(3,0,2)(2,1,0)[12] | -474.9 |
| ARIMA(3,0,1)(1,1,0)[12] | -463.4 |

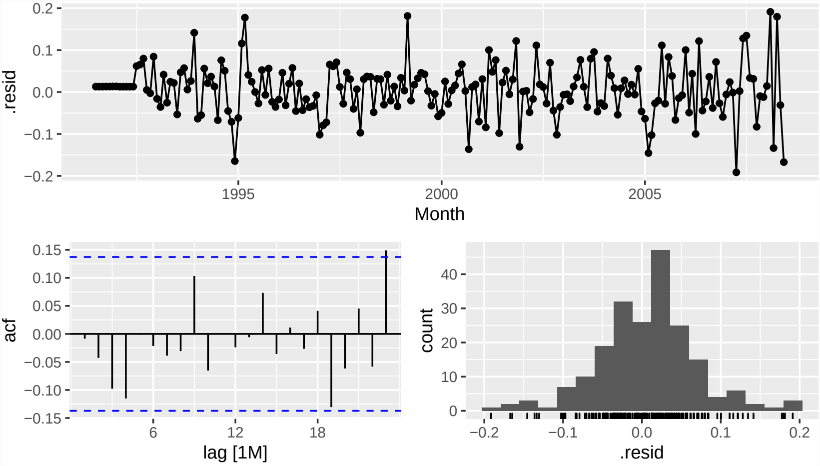
# Cortecosteroid drug sales

```r
fit <- h02 %>%
  model(best = ARIMA(log(Cost) ~ 0 + pdq(3,0,1) + PDQ(0,1,2)))
report(fit)
```

```
## Series: Cost
## Model: ARIMA(3,0,1)(0,1,2)[12]
## Transformation: log(.x)
##
## Coefficients:
##           ar1     ar2     ar3     ma1     sma1     sma2
##       -0.1602  0.5481  0.5678  0.3826  -0.5222  -0.1769
## s.e.   0.1636  0.0878  0.0942  0.1895   0.0861   0.0872
##
## sigma^2 estimated as 0.004289:  log likelihood=250.1
## AIC=-486.1   AICc=-485.5   BIC=-463.3
```

# Cortecosteroid drug sales

```r
gg_tsresiduals(fit)
```



# Cortecosteroid drug sales

```
augment(fit) %>%
  features(.resid, ljung_box, lag = 36, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 best      50.5    0.0109
```

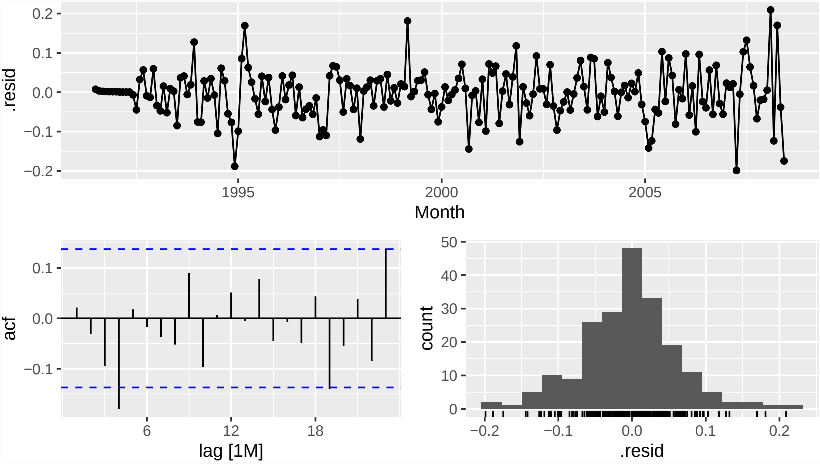
# Cortecosteroid drug sales

```r
fit <- h02 %>% model(auto = ARIMA(log(Cost)))
report(fit)
```

```
## Series: Cost
## Model: ARIMA(2,1,0)(0,1,1)[12]
## Transformation: log(.x)
##
## Coefficients:
##           ar1      ar2     sma1
##       -0.8491  -0.4207  -0.6401
## s.e.   0.0712   0.0714   0.0694
##
## sigma^2 estimated as 0.004399:  log likelihood=245.4
## AIC=-482.8   AICc=-482.6   BIC=-469.8
```

# Cortecosteroid drug sales

```r
gg_tsresiduals(fit)
```



# Cortecosteroid drug sales

```r
augment(fit) %>%
  features(.resid, ljung_box, lag = 36, dof = 3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      57.5   0.00513
```

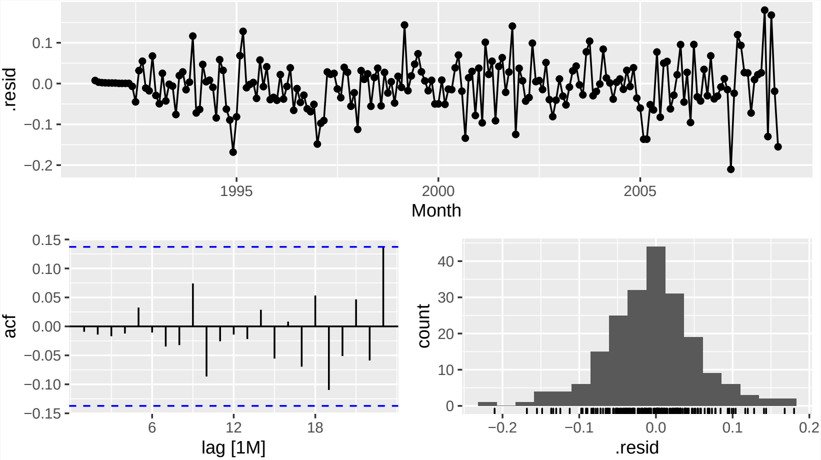
# Cortecosteroid drug sales

```r
fit <- h02 %>%
  model(best = ARIMA(log(Cost), stepwise = FALSE,
                     approximation = FALSE,
                     order_constraint = p + q + P + Q <= 9))
report(fit)
```

```
## Series: Cost
## Model: ARIMA(4,1,1)(2,1,2)[12]
## Transformation: log(.x)
##
## Coefficients:
##           ar1     ar2     ar3      ar4      ma1    sar1
##       -0.0426  0.2097  0.2016  -0.2273  -0.7423  0.6213
## s.e.   0.2167  0.1814  0.1144   0.0810   0.2075  0.2421
##          sar2     sma1    sma2
##       -0.3832  -1.2018  0.4958
## s.e.   0.1185   0.2492  0.2136
##
## sigma^2 estimated as 0.004061:  log likelihood=254.3
## AIC=-488.6   AICc=-487.4   BIC=-456.1
```

# Cortecosteroid drug sales

```
gg_tsresiduals(fit)
```



# Cortecosteroid drug sales

```
augment(fit) %>%
  features(.resid, ljung_box, lag = 36, dof = 9)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 best      35.1     0.136
```

# Cortecosteroid drug sales

Training data: July 1991 to June 2006

Test data: July 2006–June 2008

```
fit <- h02 %>%
  filter_index(~ "2006 Jun") %>%
  model(
    ARIMA(log(Cost) ~ pdq(3, 0, 0) + PDQ(2, 1, 0)),
    ARIMA(log(Cost) ~ pdq(3, 0, 1) + PDQ(2, 1, 0)),
    ARIMA(log(Cost) ~ pdq(3, 0, 2) + PDQ(2, 1, 0)),
    ARIMA(log(Cost) ~ pdq(3, 0, 1) + PDQ(1, 1, 0))
    # ... #
  )

fit %>%
  forecast(h = "2 years") %>%
  accuracy(h02)
```

# Cortecosteroid drug sales

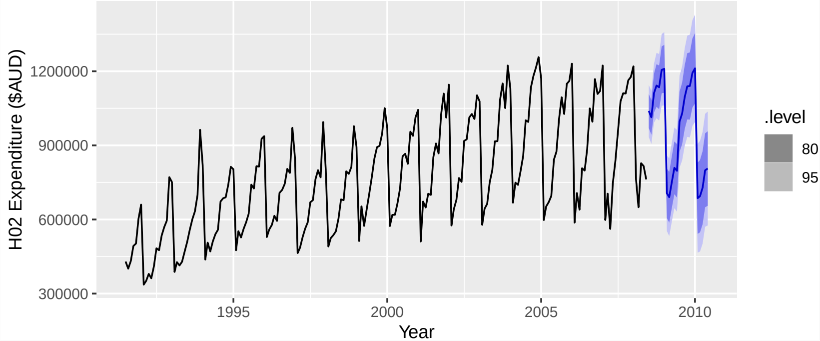
| .model | RMSE |
|---|---|
| ARIMA(3,0,1)(1,1,1)[12] | 61878 |
| ARIMA(3,0,1)(0,1,2)[12] | 62142 |
| ARIMA(3,0,1)(0,1,1)[12] | 62947 |
| ARIMA(2,1,0)(0,1,1)[12] | 62984 |
| ARIMA(4,1,1)(2,1,2)[12] | 63114 |
| ARIMA(3,0,2)(2,1,0)[12] | 65146 |
| ARIMA(3,0,1)(2,1,0)[12] | 65270 |
| ARIMA(3,0,1)(1,1,0)[12] | 66644 |
| ARIMA(3,0,0)(2,1,0)[12] | 66816 |

# 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 <- h02 %>%
  model(ARIMA(Cost ~ 0 + pdq(3,0,1) + PDQ(0,1,2)))
fit %>% forecast %>% autoplot(h02) +
  ylab("H02 Expenditure ($AUD)") + xlab("Year")
```



# Outline


1 Stationarity and differencing

2 Non-seasonal ARIMA models

3 Estimation and order selection

4 ARIMA modelling in R

5 Forecasting

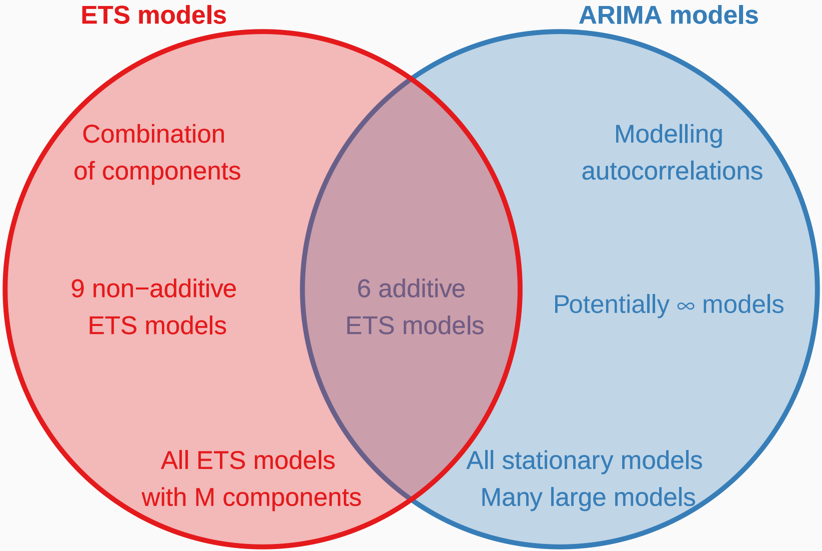
6 Seasonal ARIMA models

7 ARIMA vs ETS

# ARIMA vs ETS

- Myth that ARIMA models are more general than exponential smoothing.
- Linear exponential smoothing models all special cases of ARIMA models.
- Non-linear exponential smoothing models have no equivalent ARIMA counterparts.
- Many ARIMA models have no exponential smoothing counterparts.
- ETS models all non-stationary. Models with seasonality or non-damped trend (or both) have two unit roots; all other models have one unit root.

# ARIMA vs ETS


ETS models
ARIMA models
Combination of components
Modelling autocorrelations
9 non−additive ETS models
6 additive ETS models
Potentially ∞ models
All ETS models with M components
All stationary models
Many large models

# Equivalences

| ETS model | ARIMA model | Parameters |
|---|---|---|
| ETS(A,N,N) | ARIMA(0,1,1) | $\theta_1 = \alpha - 1$ |
| ETS(A,A,N) | ARIMA(0,2,2) | $\theta_1 = \alpha + \beta - 2$ |
| | | $\theta_2 = 1 - \alpha$ |
| ETS(A,$\text{A}_\text{d}$,N) | ARIMA(1,1,2) | $\phi_1 = \phi$ |
| | | $\theta_1 = \alpha + \phi\beta - 1 - \phi$ |
| | | $\theta_2 = (1 - \alpha)\phi$ |
| ETS(A,N,A) | ARIMA(0,0,$m$)(0,1,0)$_m$ | |
| ETS(A,A,A) | ARIMA(0,1,$m+1$)(0,1,0)$_m$ | |
| ETS(A,$\text{A}_\text{d}$,A) | ARIMA(1,0,$m+1$)(0,1,0)$_m$ | |