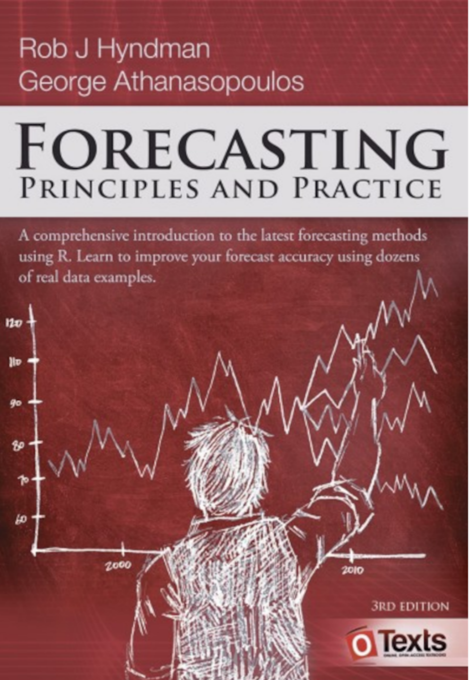


9. ARIMA models

9.7 ARIMA modelling in fable

OTexts.org/fpp3/

Traditional modelling procedure for ARIMA models

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.

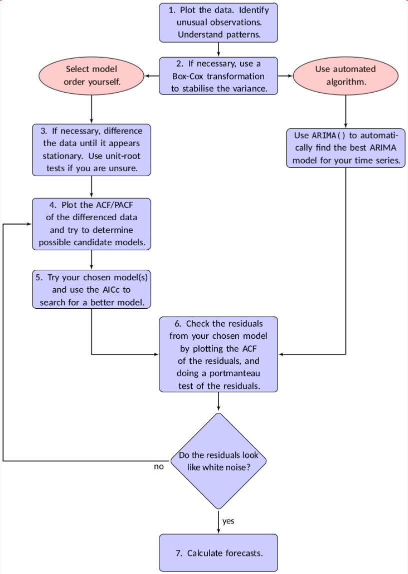
Automatic modelling procedure with `ARIMA()`

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

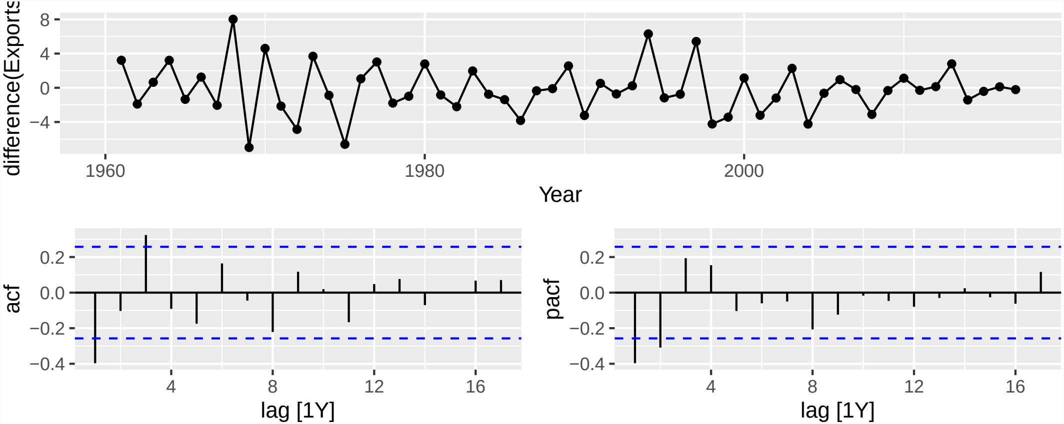


Central African Republic exports

```r
global_economy |>
  filter(Code == "CAF") |>
  autoplot(Exports) +
  labs(title = "Central African Republic exports", y = "% of GDP")
```

Central African Republic exports

```r
global_economy |>
  filter(Code == "CAF") |>
  gg_tsdisplay(difference(Exports), plot_type = "partial")
```



Central African Republic exports

```r
caf_fit <- global_economy |>
  filter(Code == "CAF") |>
  model(
    arima210 = ARIMA(Exports ~ pdq(2, 1, 0)),
    arima013 = ARIMA(Exports ~ pdq(0, 1, 3)),
    stepwise = ARIMA(Exports),
    search = ARIMA(Exports, stepwise = FALSE)
  )
```

Central African Republic exports

```r
caf_fit |> pivot_longer(!Country,
  names_to = "Model name",
  values_to = "Orders"
)
```

```
## # A mable: 4 x 3
## # Key:     Country, Model name [4]
##   Country                  `Model name`         Orders
##   <fct>                    <chr>               <model>
## 1 Central African Republic arima210     <ARIMA(2,1,0)>
## 2 Central African Republic arima013     <ARIMA(0,1,3)>
## 3 Central African Republic stepwise     <ARIMA(2,1,2)>
## 4 Central African Republic search       <ARIMA(3,1,0)>
```

Central African Republic exports

```r
glance(caf_fit) |>
  arrange(AICc) |>
  select(.model:BIC)
```

```
## # A tibble: 4 x 6
##   .model   sigma2 log_lik   AIC  AICc   BIC
##   <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 search     6.52   -133.  274.  275.  282.
## 2 arima210   6.71   -134.  275.  275.  281.
## 3 arima013   6.54   -133.  274.  275.  282.
## 4 stepwise   6.42   -132.  274.  275.  284.
```

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 d, p, q, and whether to include the intercept c.

Hyndman and Khandakar (JSS, 2008) algorithm:

- Select no. differences d via KPSS test.
- Select p, q and c 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$(0, d, 1)$

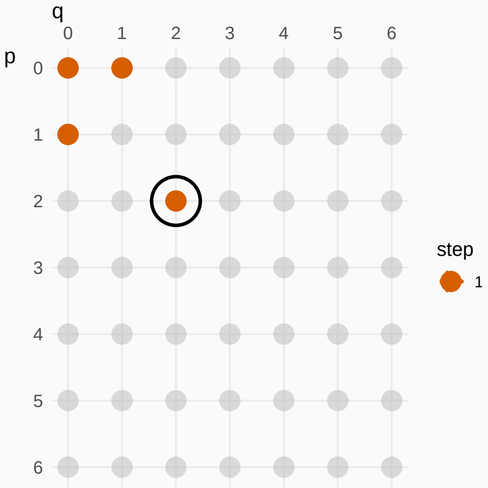
Step 2: Consider variations of current model:

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

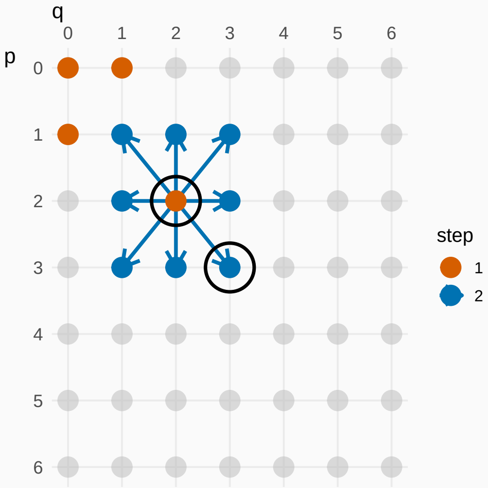
Model with lowest AICc becomes current model.

Repeat Step 2 until no lower AICc can be found.

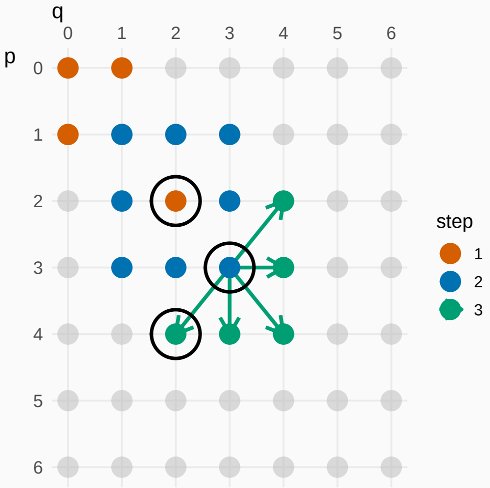
How does ARIMA() work?



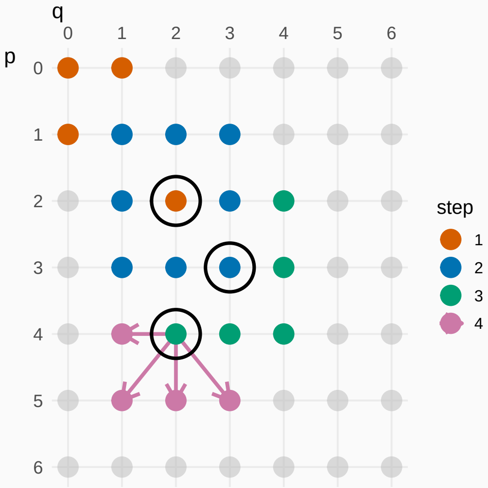
How does ARIMA() work?



How does ARIMA() work?



How does ARIMA() work?



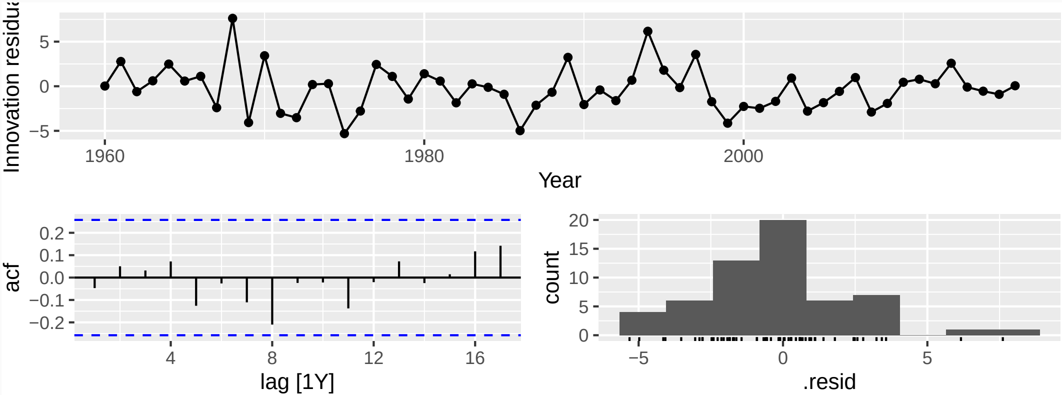
Central African Republic exports

```
caf_fit <- global_economy |>
  filter(Code == "CAF") |>
  model(
    arima210 = ARIMA(Exports ~ pdq(2, 1, 0)),
    arima013 = ARIMA(Exports ~ pdq(0, 1, 3)),
    stepwise = ARIMA(Exports),
    search = ARIMA(Exports, stepwise = FALSE)
  )
```

```
## # A tibble: 4 x 5
##   .model              Orders sigma2 log_lik  AICc
##   <chr>              <model>  <dbl>   <dbl> <dbl>
## 1 search   <ARIMA(3,1,0)>   6.52   -133.  275.
## 2 arima210 <ARIMA(2,1,0)>   6.71   -134.  275.
## 3 arima013 <ARIMA(0,1,3)>   6.54   -133.  275.
## 4 stepwise <ARIMA(2,1,2)>   6.42   -132.  275.
```

Central African Republic exports

```r
caf_fit |>
  select(search) |>
  gg_tsresiduals()
```



Portmanteau tests of residuals for ARIMA models

With ARIMA models, more accurate portmanteau tests obtained if degrees of freedom are adjusted to take account of number of parameters in the model.

- Use $\ell - K$ degrees of freedom, where $K = p + q$ = number of AR and MA parameters in the model.
- `dof` argument in `ljung_box()`.

Central African Republic exports

```r
augment(caf_fit) |>
  filter(.model == "search") |>
  features(.innov, ljung_box, lag = 10, dof = 3)
```

```
## # A tibble: 1 x 4
##   Country                  .model lb_stat lb_pvalue
##   <fct>                    <chr>    <dbl>     <dbl>
## 1 Central African Republic search    5.75     0.569
```

Central African Republic exports

```r
caf_fit |>
  forecast(h = 5) |>
  filter(.model == "search") |>
  autoplot(global_economy)
```