Time-Series Forecasting

ECON20222 - Lecture 10

Ralf Becker and Martyn Andrews

Aim for today

- Identify the presence of seasonal features in a time-series
- Use AR models to produce single step ahead forecasts
- Use AR models to produce multiple step ahead forecasts
- Evaluate and compare different forecasts

Forecast Sources

Forecasts are either

- totally made up
- obtained from betting markets (usually for binary events), e.g. https://www.predictit.org/markets
- obtained from surveys, e.g. https://www.ecb.europa.eu/stats/ecb_surveys/survey_of_professional_forecasters/html/index.en.html
- obtained from formal forecasting models

Forecasting basics

Let's say we have a time series y_t, for $t = 1, ..., T$ where T is the last available observation.

The aim is to use the observations available to obtain 1 step - or more generally h step ahead forecasts.

$$E(y_{T+1}|y_T, y_{T-1}, y_{T-2}, ...) = E(y_{T+1}|I_T) = \hat{y}_{T+1|T}$$

$$E(y_{T+h}|y_T, y_{T-1}, y_{T-2}, ...) = E(y_{T+h}|I_T) = \hat{y}_{T+h|T}$$

We call I_t the information set.

1. We use the data in the information set to estimate a model representing the process
2. We then use this estimated model to obtain a forecast

Forecasting basics

- We may want to use information from other time-series, x_t, z_t etc.
- This opens up more complex models and the additional information may add quality to the forecast.
- But if you forecast multiple steps ahead then we need forecasts for these (x and z) to obtain forecasts for y.

Forecasting basics - Uncertainty

When forecasting we know from the outset that our forecast is not going to hit the actual outcome and hence we should expect the forecast error

$$\epsilon_{T+1|T} = y_{T+1} - \hat{y}_{T+1|T}$$

to be unequal to 0. Note that y_{T+1} is the actual observation which we don't have at time T.

$\rightarrow$ interval forecasts (see below)

Forecasting basics - Uncertainty

We should expect forecasts to be imperfect for the following reasons:

- Even the best model will not capture all the random variation
- Which variables are relevant for forecasting y?
- What is the right model?
- When estimating a model we will have uncertainty about the parameters.

All of these are actually quite harmless when carefully modelled, **but** significant forecast errors will arise if there are changes in the process which effect the process such that:

- the overall (unconditional mean) of the process changes
- the trend of a series changes

Our working example - female unemployment rate

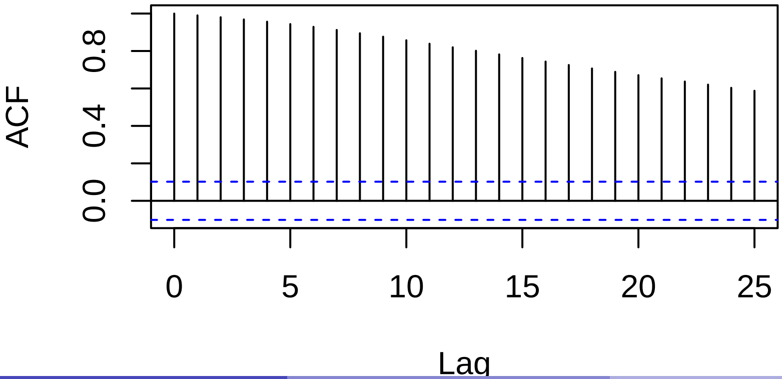
```r
# Download: Female unemployment rate (YCPL in database LMS)
ur_female <- pdfetch_ONS("YCPL","LMS")
names(ur_female) <- "Unemp Rate (female)"
```

Our working example - female unemployment rate

Our working example - female unemployment rate

Here we focus on a forecasting model which only uses the unemployment rate itself. In particular we look at an autoregressive (AR) forecasting model.

ACF contains all the information which can be used for forecasting



Data properties

Recall that we want to work with series which are stationary. Clearly the unemployment rate, ur_t is not.

We will therefore work with the first difference

The `forecast` package

- To implement forecasting in R we will rely on the `forecast` package (remember to install and then load)
- This is written by Rob Hyndman (Monash University).
- The `forecast` package requires the data to be in `ts` format not the `xts` format in which they are delivered from the `pdfetch` package

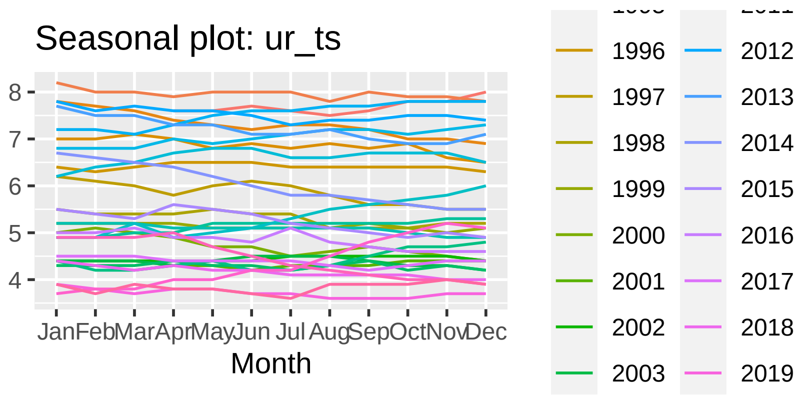
```
ur_ts <- ts(as.numeric(ur_female),
            start = c(1992,4), frequency = 12)
```

- Use `?ts` to see details of the `ts` function

Looking out for seasonalties

Use a special function which recognises the frequency of data and plots them “year-by-year”

```r
ggseasonplot(ur_ts)
```



No obvious seasonalities appear. But see later.

Forecasting from AR Model

In our dataset $t = 1, ..., T$ from April 1992 ($t = 1$) to Jan 2022 ($t = T$). We want to obtain forecasts for Feb 2022 ($T + 1$), Mar 2022 ($T + 2$), etc.

The model we want to forecast from

$$\Delta ur_t = \alpha + \beta_1 \ \Delta ur_{t-1} + \beta_2 \ \Delta ur_{t-2} + u_t \tag{1}$$

1. We obtain parameter estimates by estimating this model on all available data (I_T) $\Rightarrow \hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2$
   We also obtain the sample variance of the residuals (s_u^2)
2. Use this model for forecasting

Forecasting from AR Model - Point forecasts

Recall that the information set we used to estimate the model is the set of information available to us: $I_T = ur_T, ur_{T-1}, ur_{T-2}, ...$ or in differences $I_T = \Delta ur_T, \Delta ur_{T-1}, \Delta ur_{T-2}, ...$

Restate the AR model for the period for which you want to forecast

$$\Delta ur_{T+1} = \alpha + \beta_1 \ \Delta ur_T + \beta_2 \ \Delta ur_{T-1} + u_{T+1}$$

Now form expectations conditional on I_T

$$\begin{aligned} E[\Delta ur_{T+1}|I_T] &= E[\alpha + \beta_1 \ \Delta ur_T + \beta_2 \ \Delta ur_{T-1} + u_{T+1}|I_T] \\ &= \alpha + \beta_1 \ E[\Delta ur_T|I_T] + \beta_2 \ E[\Delta ur_{T-1}|I_T] \end{aligned} \tag{2}$$

where we recognise that

$$E[u_{T+1}|I_T] = 0, E[\alpha|I_T] = \alpha, E[\beta_1|I_T] = \beta_1, E[\beta_2|I_T] = \beta_2$$

$E[\Delta ur_{T+1}|I_T]$ is a point forecast, the **one best** forecast value.

Forecasting from AR Model - Point forecasts

The forecasting equation (2) requires two more steps before we can use it:

- Replace the unknown fixed coefficients with sample estimates, eg replace α with $\hat{\alpha}$
- Notice that: $E[\Delta ur_T|I_T] = \Delta ur_T$ and $E[\Delta ur_{T-1}|I_T] = \Delta ur_{T-1}$

$$E[\Delta ur_{T+1}|I_T] = \hat{\alpha} + \hat{\beta}_1\ \Delta ur_T + \hat{\beta}_2\ \Delta ur_{T-1} \tag{3}$$

After estimating the AR(2) model with information up to time T we have all the information on the right-hand side.

$E[\Delta ur_{T+1}|I_T]$ is our best estimate for the unemployment rate next period $(T+1)$ made at time T. Note that we "lost" the error term as we assumed $E[u_{T+1}|I_T] = 0$. But we still recognise that in any particular period we are likely to get a non-zero u_{T+1} ($\rightarrow$ Interval forecasts)

Forecasting from AR Model - Point forecasts

Multi-step ahead forecasts

Say we want to forecast a value two months ahead ($T+2$). Restate (2) for $T+2$:

$$E[\Delta ur_{T+2}|I_T] = \alpha + \beta_1\ E[\Delta ur_{T+1}|I_T] + \beta_2\ E[\Delta ur_T|I_T] \tag{4}$$

Replace the parameters with sample estimates and check which values are already in I_T:

$$E[\Delta ur_{T+2}|I_T] = \hat{\alpha} + \hat{\beta}_1\ E[\Delta ur_{T+1}|I_T] + \hat{\beta}_2\ \Delta ur_T \tag{5}$$

We are left with $E[\Delta ur_{T+1}|I_T]$, but that is of course the value of our one-step ahead forecast, and so we can substitute this value from (3).

Recursively we can build up forecasts one period at a time.

Forecasting from AR Model - R implementation

At the core of estimating these models in R is the `Arima` function.
`Arima` stands for **A**uto**R**egressive **I**ntegrated **M**oving **A**verage.
`order = c(2,0,0)` estimates an AR(2) model.

```
dur_ts <- diff(ur_ts) # create differenced series
fit_dur <- Arima(dur_ts,order = c(2,0,0))
summary(fit_dur)
```

```
## Series: dur_ts
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2     mean
##       0.0189  0.2042  -0.0096
## s.e.  0.0509  0.0510   0.0077
##
## sigma^2 = 0.01336:  log likelihood = 274.14
## AIC=-540.28   AICc=-540.17   BIC=-524.63
##
## Training set error measures:
##                         ME      RMSE        MAE MPE MAPE      MASE        ACF1
## Training set -7.366606e-05 0.1150975 0.08752183 NaN  Inf 0.7166352 0.001525754
```

Forecasting from AR Model - R implementation

Above we introduced the AR(2) model as in equation (1).

$$\Delta ur_t = \alpha + \beta_1 \ \Delta ur_{t-1} + \beta_2 \ \Delta ur_{t-2} + u_t$$

The `Arima` function estimates a slightly different (but equivalent) version:

$$(\Delta ur_t - m) = \beta_1 \ (\Delta ur_{t-1} - m) + \beta_2 \ (\Delta ur_{t-2} - m) + u_t$$

Note that instead of a constant we have the new term m. For stationary series this is the value towards which a very long-range forecast will converge (and it is basically the same as the sample mean - see `mean` in output).

Forecasting from AR Model - R implementation

So far we did the equivalent to Step 1

Step 1
We obtain parameter estimates by estimating this model on all available data $(I_T) \Rightarrow \hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2$. We also obtain the sample variance of the residuals (s_u^2). Saved in `fit_dur`.

Step 2 is to use this model to forecast. This is done using the `forecast` function (here we forecast `h=12` months ahead).

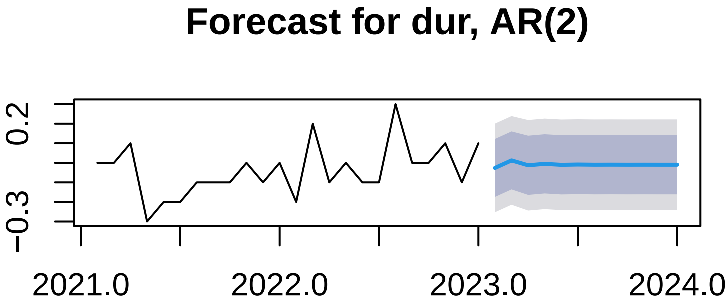
```
for_dur <- forecast(fit_dur,h=12)
for_dur$mean    # the actual forecast values are in mean
```

```
##               Jan          Feb          Mar          Apr          May
## 2023              -0.026020164  0.012426488 -0.012573746 -0.005197684
## 2024 -0.009641738
##               Jun          Jul          Aug          Sep          Oct
## 2023 -0.010162052 -0.008750119 -0.009736906 -0.009467323 -0.009663680
## 2024
##               Nov          Dec
## 2023 -0.009612359 -0.009651475
## 2024
```

The object `for_dur` now contains the forecasts (and a whole lot more)

Forecasting from AR Model - R implementation

```
plot(for_dur, include = 24, main="Forecast for dur, AR(2)")  # includes last 24 obse
```



There isn't a whole lot happening in the forecast. It quickly converges to `mean = -0.0098`.

Forecasting from AR Model - Differences and Levels

We build the model in differences Δur_t as the levels (ur_t) were not stationary. But in the end we may be interested in the level of the unemployment rate.

Getting a forecast for the level is fairly straightforward

Time	t	ur_t	Δur_t	$E[\Delta ur_t \vert I_T]$	$E[ur_t \vert I_T]$
Nov 2022	$T-2$	3.8	0.1		
Dec 2022	$T-1$	3.8	0.0		
Jan 2023	T	3.9	0.1		
Feb 2023	$T+1$			-0.005	3.895
Mar 2023	$T+2$			0.013	3.908
...					

Forecasting from AR Model - Differences and Levels

The forecast package makes this procedure somewhat simpler.

We feed into `Arima` the series we are really interested in (`ur_ts`) and if we need to difference once then we use `order = c(2,1,0)`, where

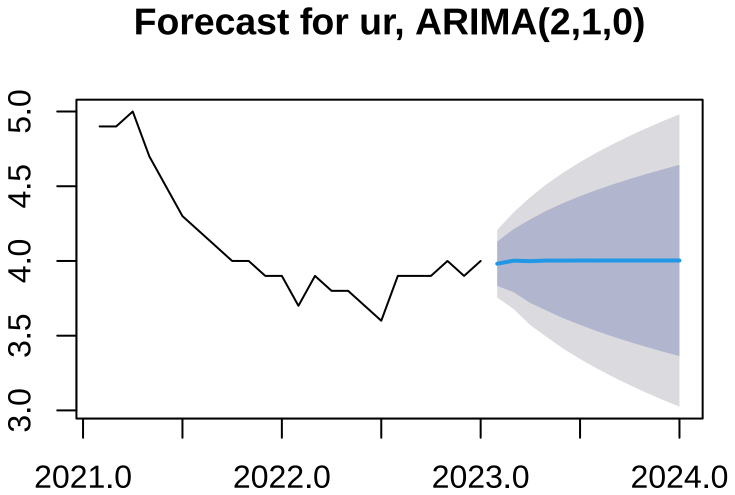
- the first entry tells R that we want an AR(2) model,
- the second entry tells R that we want to difference the series **once** and
- the third entry tells R that we need an MA(0) component (not covered here)

```
fit_ur <- Arima(ur_ts,order = c(2,1,0))
for_ur <- forecast(fit_ur, h =12)
```

The forecasts will then be calculated for `ur_ts` directly.

Forecasting from AR Model - Differences and Levels

```
plot(for_ur, include = 24, main="Forecast for ur, ARIMA(2,1,0)
```



Forecasting from AR models - Interval forecasts

We know that forecasts will not be perfect. (Just like the in-sample fit is not perfect).

The way this is expressed is by creating interval forecasts:

$$P(lb \leq E[\Delta ur_{T+1}|I_T]) \leq ub) = 1 - sig \tag{6}$$

The smaller sig the wider the interval, and the more certain $(1 - sig)$ we are that the actual value will end up in that interval.

The 80% and 95% forecast intervals are shown in the previous graph.

Automatic order selection

The `forecast` package has a function (`auto.arima`) which instructs R to find the model with the best order (according to an information criterion).

```
fit_ur_a <- auto.arima(ur_ts)
summary(fit_ur_a)
```

```
## Series: ur_ts
## ARIMA(1,1,4)(2,0,0)[12]
##
## Coefficients:
##          ar1      ma1     ma2      ma3     ma4    sar1     sar2
##       0.7571  -0.7579  0.1841  -0.1854  0.2254  0.0033  -0.1312
## s.e.  0.1013   0.1054  0.0631   0.0683  0.0584  0.0552   0.0551
##
## sigma^2 = 0.01263:  log likelihood = 286.11
## AIC=-556.22   AICc=-555.82   BIC=-524.94
##
## Training set error measures:
##                        ME      RMSE        MAE        MPE     MAPE      MASE
## Training set -0.005423602 0.1111646 0.08470913 -0.1009932 1.578093 0.1793369
##                       ACF1
## Training set -0.000420403
```

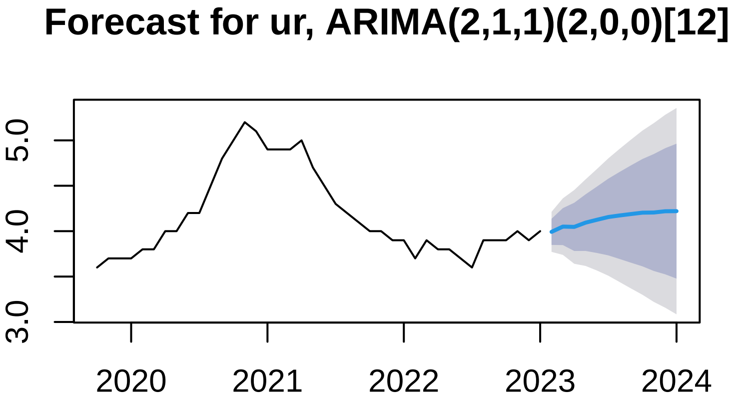
Automatic order selection

The chosen model is `ARIMA(1,1,4)(2,0,0)[12]`. Without any details, this output indicates that

- the procedure has found an AR(1) component
- the procedure wants to difference once to achieve stationarity
- the procedure has found an MA(4) component (no further details here)
- the procedure has found that there is some seasonality (at the 12 month / 1 year period)
- to cater for this seasonality it also included ur_{t-12} and some other lags to capture this seasonal dependence

Automatic order selection

```
for_ur_a <- forecast(fit_ur_a, h =12)
plot(for_ur_a, include = 40,
     main="Forecast for ur, ARIMA(2,1,1)(2,0,0)[12]")
```



Forecast Evaluation

So far we forecast for 2023 and if we wanted to evaluate how well we did we would have to wait for a year to obtain realised values.

Let's turn back time and pretend we are at the end of 2017 and want to forecast for 2018. As we have these values we can then evaluate how well the forecast model did.

We create a shortened series which finishes in Dec 2017.

```
ur_ts_17 <- window(ur_ts,end = c(2017,12))
```

Forecast Evaluation

Repeat the earlier exercise.

- Estimate AR(2) model and forecast 12 months ahead (Jan to Dec 2018)
- Let `auto.arima` select the best model and forecast Jan to Dec 2018

```
fit_ur_17 <- Arima(ur_ts_17,order = c(2,1,0))
for_ur_17 <- forecast(fit_ur_17, h =12)

fit_ur_a_17 <- auto.arima(ur_ts_17)
for_ur_a_17 <- forecast(fit_ur_a_17, h =12)
```

Forecast Evaluation

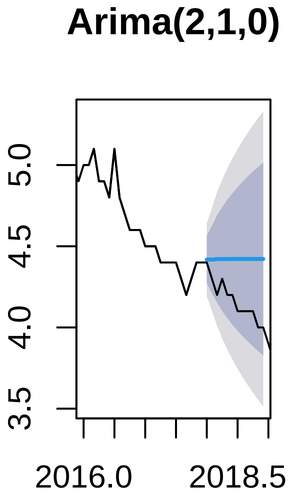
Then we plot the two forecast series against the actual realisations

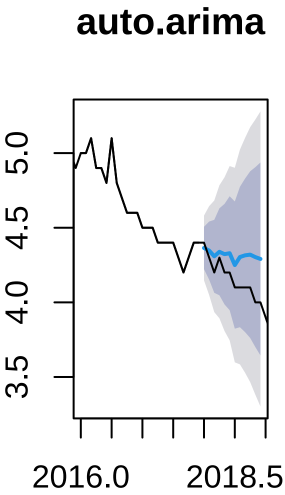
```
par(mfrow=c(1,2))    # This plots the next two plots
                     # next to each other, in a (1,2) grid

plot(for_ur_17, include = 24, main = "Arima(2,1,0)")
lines(ur_ts)         # this adds ur_ts to the previous plot

plot(for_ur_a_17, include = 24, main = "auto.arima")
lines(ur_ts)
```

Forecast Evaluation





Forecast Evaluation - Numerical

We can see that the seasonal model from `auto.arima` fits the data better. It, at least anticipates that the unemployment rate continued to drop.

Numerical measures of forecast accuracy compare the forecast to the observation. A range of measures exist, e.g. the Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{12}\sum_{h=1}^{12}(E[ur_{T+h}|I_T] - ur_{T+h})^2} \quad (7)$$

Smaller RMSE is better.

Forecast Evaluation - Numerical

```r
accuracy(for_ur_17,ur_ts)
```

```
##                        ME      RMSE        MAE        MPE     MAPE      MASE
## Training set -0.008496628 0.1141159 0.08595689 -0.1609257 1.504272 0.1884074
## Test set     -0.254293230 0.2807261 0.25429323 -6.1879077 6.187908 0.5573807
##                      ACF1 Theil's U
## Training set -0.004431789        NA
## Test set      0.573813816  4.090548
```

```r
accuracy(for_ur_a_17,ur_ts)
```

```
##                        ME      RMSE        MAE         MPE     MAPE      MASE
## Training set -0.004330286 0.1096765 0.08473188 -0.06255499 1.487238 0.1857223
## Test set     -0.149464958 0.1788153 0.15539167 -3.65504176 3.789740 0.3406002
##                     ACF1 Theil's U
## Training set 0.003385942        NA
## Test set     0.622004661  2.613773
```

Forecasting failure - Covid-19

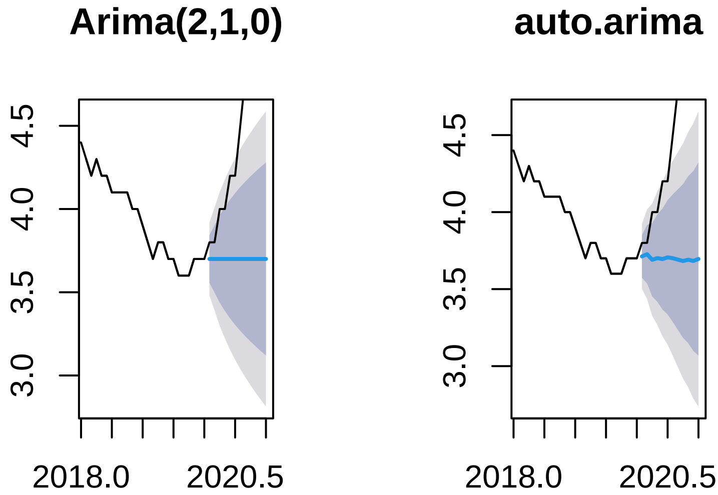
Let's consider data up until Jan 2020 and use the information available at that time to forecast the female unemployment rate for the 12 months after.

```
ur_ts_20 <- window(ur_ts,end = c(2020,1))

fit_ur_20 <- Arima(ur_ts_20,order = c(2,1,0))
for_ur_20 <- forecast(fit_ur_20, h =12)

fit_ur_a_20 <- auto.arima(ur_ts_20)
for_ur_a_20 <- forecast(fit_ur_a_20, h =12)
```

Forecasting failure - Covid-19



Forecasting failure - Covid-19

On a previous slide we argued that minor forecast “failures” will be due to random variation, failure to include all relevant variables, using the wrong model and parameter uncertainty.

We mentioned earlier that the key model features which will result in bad forecasts are:

- the overall (unconditional mean) of the process changes
- the trend of a series changes

An unforecastable event like Covid will affect both these and hence the forecast “failure”.

Forecasting - Scenarios

Importantly, all forecasts are **contingent** on certain assumptions. Basically we need to assume that the model used is and **will stay** to be the correct model.

Therefore you often get **scenarios** rather than **forecasts**. For instance the Office for Budget Responsibility's Covid-19 Scenario.

The key scenario assumption in April 2020 was

> *This was a scenario rather than a forecast, based on the illustrative assumption that people's movements (and thus economic activity) would be heavily restricted for three months and would get back to normal over the subsequent three months.*

Forecasting - Scenarios

We mentioned earlier that the key model features which will result in bad forecasts are:

- the overall (unconditional mean) of the process changes
- the trend of a series changes

If the above assumption is inappropriate, we will see changes in these model features and significant deviations from any scenario predictions.

Summary

- AR models can be used to produce forecasts (one and multiple step ahead)
- Order selection is very important
- But acknowledge forecast uncertainty (using intervals)
- Implementation in R is very straightforward
- Tend to work well for short-term forecasts as, in the short-run, it is likely that overall mean and trend remain unchanged