Introduction to Time Series Analyis

Dr Rafael de Andrade Moral
Associate Professor of Statistics, Maynooth University

rafael.deandrademoral@mu.ie
https://rafamoral.github.io

Forecasting Approaches

- In time series analysis, methods typically rely on one (or both) of the following approaches:

1. Base your future predictions on previous values of the data
2. Base your future predictions on how wrong you were in your past predictions

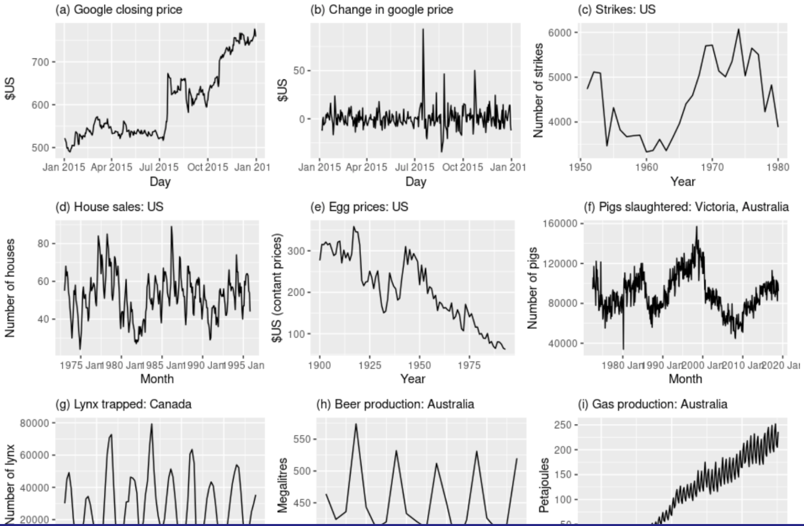
ARIMA Models

- ARIMA models provide another approach to time series forecasting.
- Exponential smoothing and ARIMA models are the two most widely used approaches to time series forecasting, and provide complementary approaches to the problem.
- While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the *autocorrelations* in the data.
- Before we introduce ARIMA models, we must first discuss the concept of stationarity and the technique of differencing time series.

Recap: Stationarity

- A stationary time series is one whose statistical properties do not depend on the time at which the series is observed.
- Thus, time series with trends, or with seasonality, are not stationary – the trend and seasonality will affect the value of the time series at different times.
- On the other hand, a white noise series is stationary – it does not matter when you observe it, it should look much the same at any point in time.
- In general, a stationary time series will have no predictable patterns in the long-term.

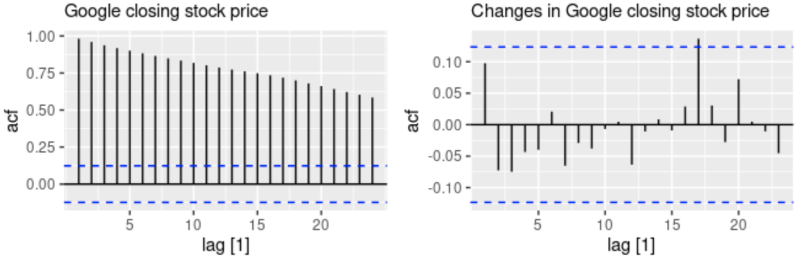
Recap: which time series is stationary?



Differencing

- Note that the Google stock price was non-stationary in panel (a), but the daily changes were stationary in panel (b).
- This shows one way to make a non-stationary time series stationary: compute the differences between consecutive observations.
- This is known as **differencing**.
- Transformations such as logarithms can help to stabilise the variance of a time series.
- Differencing can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

Differencing




```
google_2015 %>%
  mutate(diff_close = difference(Close)) %>%
  features(diff_close, ljung_box, lag = 10)
```

```
## # A tibble: 1 x 3
##   Symbol lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 GOOG      7.91     0.637
```

Recap: Random Walk model

- The differenced series is the *change* between consecutive observations in the original series, and can be written as

$$y'_t = y_t - y_{t-1}$$

- The differenced series will have $T - 1$ values since it is not possible to calculate a difference for the first observation.
- When the differenced series is white noise, the model for the original series can be written as

$$y_t - y_{t-1} = \varepsilon_t$$

  where ε_t denotes white noise.
- Rearranging yields the random walk model

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

- A closely related model allows the differences to have a non-zero mean:

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

  where c is the average of the changes between consecutive observations.

Second-order differencing

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

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

- In this case y''_t will have $T - 2$ values.
- We would model the "change in the changes" of the original data.
- In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

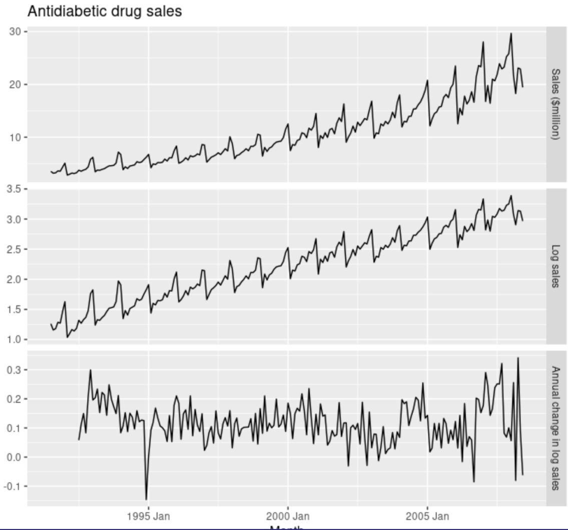
- A seasonal difference is the difference between an observation and the previous observation from the same season:

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

- Also called "lag-m differences".
- If seasonally differenced data appear to be white noise, then an appropriate model for the original data is

$$y_t = y_{t-m} + \varepsilon_t$$

Seasonal differencing



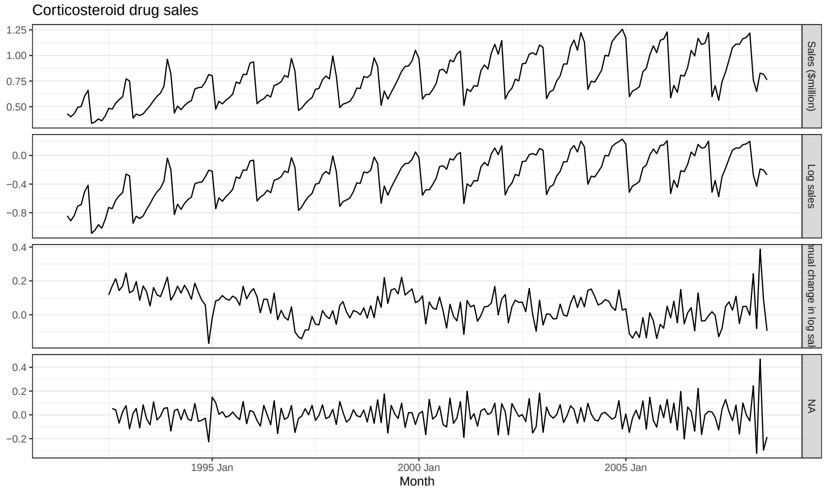
Multiple differencing

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

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

- When both seasonal and first differences are applied, it makes no difference which is done first – the result will be the same.
- However, if the data have a strong seasonal pattern, it is recommended that seasonal differencing be done first, because the resulting series will sometimes be stationary and there will be no need for a further first difference. If first differencing is done first, there will still be seasonality present.

Multiple differencing



Differencing

- Beware that applying more differences than required will induce false dynamics or autocorrelations that do not really exist in the time series.
- Therefore, do as few differences as necessary to obtain a stationary series.
- It is important that if differencing is used, the differences are interpretable.
- First differences are the change between one observation and the next.
- Seasonal differences are the change between one year to the next.
- Other lags are unlikely to make much interpretable sense and should be avoided.

Unit Root Tests

- One way to determine more objectively whether differencing is required is to use a **unit root test**.
- These are statistical hypothesis tests of stationarity that are designed for determining whether differencing is required.
- A number of unit root tests are available, which are based on different assumptions and may lead to conflicting answers, e.g. augmented Dickey-Fuller test, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.
- For the KPSS test the null hypothesis is that the series is stationary.

Backshift Notation

- The backward shift operator B is a useful notational device when working with time series lags:

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

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

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

- For monthly data if we with to consider "the same month last year" the notation is

$$B^{12} y_t = y_{t-12}$$

Backshift Notation

- The backshift operator is convenient for describing the process of differencing.
- A first difference can be written as

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

- So a first difference can be represented by $(1 - B)$.
- Similarly, second-order differences yield

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

- In general, a $d-$order difference can be written as

$$(1 - B)^d y_t$$

- Combining differences is conveniently done using backshift notation. E.g. a seasonal difference followed by a first difference:

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

Unit Roots

- The backshift notation helps us discover whether there is differencing in a time series.
- When factorising a time series using the backshift operator, if we obtain an expression of the form $(1-B)^d$ multiplying y_t, then we conclude that the series is $d-$differenced.
- This means that the polynomial $(1-B)^d$ has d **unit roots**, and therefore has been differenced d times.
- This will be very useful to determine the order of ARIMA models, as will be discussed later.

Autoregression

- A linear regression model can be written as

$$y_i = c + \beta_1 x_{1,i} + \ldots + \beta_p x_{p,i} + \varepsilon_i$$

where $x_1, \ldots, x_p$ are explantory variables, β_j are parameters, and $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$.

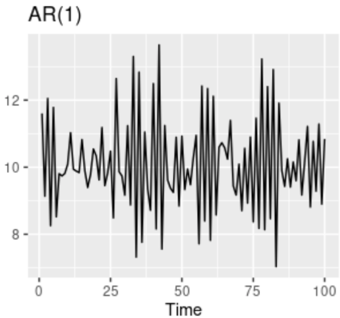
- In an autoregression model, we forecast the variable of interest using a linear combination of *past values of the variable*.
- The term *auto*regression indicates that it is a regression of the variable against itself.
- An autoregressive model of order p can be written as

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

where ε_t is white noise.

- We refer to this as an **AR(**p**)** model.
- Autoregressive models are remarkably flexible at handling a wide range of different time series patterns.

Autoregression



- Left panel: $y_t = 18 - 0.8y_{t-1} + \varepsilon_t$
- Right panel: $y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \varepsilon_t$
- $\varepsilon \sim \mathrm{N}(0, 1)$

Autoregression

For an AR(1) model

- when $\phi_1 = 0$ and $c = 0$, y_t is equivalent to white noise
- when $\phi_1 = 1$ and $c = 0$, y_t is equivalent to a random walk
- when $\phi_1 = 1$ and $c \neq 0$, y_t is equivalent to a random walk with drift
- when $-1 < \phi_1 < 1$, y_t tends to oscillate around the mean $c/(1 - \phi_1)$

We normally restrict autoregressive models to stationary data, in which case some constraints on the values of the parameters are required.

- For an AR(1) model: $-1 < \phi_1 < 1$
- For an AR(2) model: $-1 < \phi_2 < 1, \phi_1 + \phi_2 < 1, \phi_2 - \phi_1 < 1$
- For an AR(p) model with $p \geq 3$ the restrictions are more complicated

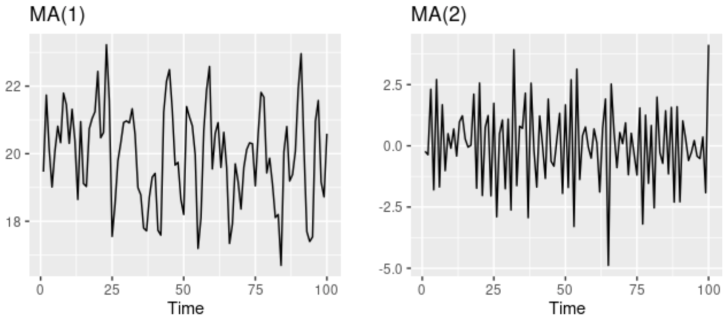
Moving average models

- Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model

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

  where ε_t is white noise.
- We refer to this as an **MA(q)** model, a moving average of order q.
- Of course, we do not observe the values of ε_t, so it's not really a regression in the usual sense.
- Notice that each value of y_t can be thought of as a weighted moving average of the past few forecast errors (although the coefficients will not normally sum to one).
- However, moving average models should not be confused with the *moving average smoothing* we discussed previously.
- A moving average model is used for forecasting future values, while moving average smoothing is used for estimating the trend-cycle of past values.

Moving average models



- Left panel: $y_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$
- Right panel: $y_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$
- $\varepsilon \sim \mathrm{N}(0, 1)$

AR as an MA process and vice-versa

- It is possible to write any AR(p) model as an MA(∞) model:
- If we impose some constraints on the MA parameters, we are also able to write an MA(q) process as an AR(∞) process.
- All MA(q) series are stationary!
- Under some parameter constraints, AR(p) series are also stationary

The ARMA Model

- If we combine autoregression and a moving average model, we obtain a non-seasonal ARMA model.
- ARMA is an acronym for "AutoRegressive Moving Average"
- We can write it as

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

  or in backshift notation

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) \varepsilon_t$$

- This is called an **ARMA(p,q)** model, where p is the order of the AR part and q is the order of the MA part.

The ARIMA Model

- Recall that we may write a $d-$differenced time series as

$$(1-B)^d y_t = c + \varepsilon_t,$$

  i.e. the polynomial $(1-B)^d$ has d unit roots.
- Combining differencing with an ARMA model yields a non-seasonal ARIMA model, which stands for "AutoRegressive Integrated Moving Average" and can be written in backshift notation as

$$(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)(1-B)^d y_t = c+(1+\theta_1 B+\theta_2 B^2+\cdots+\theta_q B^q)\varepsilon_t$$

- This is called **ARIMA(p,d,q)**, where d is the degree of first differencing involved.

Special cases of ARIMA models

ARIMA(p,d,q)	Special case
ARIMA(0,0,0)	White noise
ARIMA(0,1,0)	Random walk
ARIMA(0,1,0) with constant	Random walk with drift
ARIMA(p,0,0)	AR(p)
ARIMA(0,0,q)	MA(q)
ARIMA(0,d,0)	I(d)

- MA(q) models are always stationary
- AR(p) models are stationary under certain conditions
- I(d) models are **not** stationary

Selecting the order of an ARIMA model

- Selecting appropriate values for p, d and q can be difficult. The `ARIMA()` function from the `fable` package does this automatically. We will see how to choose them manually later.
- Example: Egyptian exports as a percentage of GDP from 1960 to 2017.

ARIMA Models

- The `ARIMA()` function is useful, but anything automated can be dangerous
- It is worth understanding something of the behaviour of the models even when you rely on an automatic procedure to choose the model for you.

1. If $c = 0$ and $d = 0$, the long-term forecasts will go to zero.
2. If $c = 0$ and $d = 1$, the long-term forecasts will go to a non-zero constant.
3. If $c = 0$ and $d = 2$, the long-term forecasts will follow a straight line.
4. If $c \neq 0$ and $d = 0$, the long-term forecasts will go to the mean of the data.
5. If $c \neq 0$ and $d = 1$, the long-term forecasts will follow a straight line.
6. If $c \neq 0$ and $d = 2$, the long-term forecasts will follow a quadratic trend.
7. As d grows the prediction intervals become wider.
8. For $d = 0$ the long-term forecast standard deviation converges to the standard deviation of the historical data.
9. For cyclic forecasts it is necessary to have $p \geq 2$ along with other conditions.

ACF and PACF plots

- It is usually not possible to tell, simply from a time plot, what values of p and q are appropriate for the data.
- However, it is sometimes possible to use the **ACF plot**, and the closely related **PACF plot**, to determine appropriate values for p and q.
- Recall that an ACF plot shows the autocorrelations which measure the relationship between y_t and y_{t-k} for different values of k.
- If y_t and y_{t-1} are correlated, then y_{t-1} and y_{t-2} must also be correlated. Then y_t and y_{t-2} might be correlated simply because they are both connected to y_{t-1} rather than because new information in y_{t-2} could be useful to forecast y_t.
- To overcome this problem we use **partial autocorrelations**.
- These measure the relationship between y_t and y_{t-k} after removing the effects of lags $1, 2, \dots, k-1$.
- Each partial autocorrelation can be estimated as the last coefficient in an AR model.
- Specifically, the $k-$th partial autocorrelation coefficient α_k is equal to ϕ_k in an AR(k) model. (In practice we use more efficient algorithms rather than fitting k AR models.)

ACF and PACF

- ACF: Corr(y_t, y_{t-k})
- Useful to determine the order of the MA terms in an ARIMA(0,d,q) model.
- There will be a significant spike at lag q in the ACF, but none beyond lag q.
- PACF: Corr$(y_t, y_{t-k} | y_{t-1}, \ldots, y_{t-k+1})$
- Useful to determine the order of the AR terms in an ARIMA(p,d,0) model.
- There will be a significant spike at lag p in the PACF, but none beyond lag p.

ARIMA Models: Estimation

- After we identify the model order (i.e. the values of p, d and q), we need to estimate the parameters $c, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$.
- There are different estimation methods we can use to fit a statistical model.
- `fable` uses **maximum likelihood estimation** to estimate ARIMA models.
- This technique finds the values of the parameters which maximise the probability of obtaining the data that we have observed.
- Note that ARIMA models are much more complicated to estimate than regression models, and different software will give slightly different answers as they use different methods of estimation, and different optimisation algorithms.
- In practice, `fable` will report the *log-likelihood* of the data; that is, the logarithm of the probability of the observed data coming from the estimated model.
- For given values of p, d and q, `ARIMA()` will try to maximise the log likelihood when finding parameter estimates.

Information Criteria

- We can use information criteria based on the log-likelihood to select the values of p and q for an ARIMA model (but not d).
- Akaike's Information Criterion (AIC) is written as

$$\text{AIC} = -2\log L + 2k$$

  where k is the number of parameters in the model ($k = p + q + 1$ if $c = 0$ and $k = p + q + 2$ if c is estimated)
- The corrected AIC

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

  is used to correct for small sample sizes
- The Bayesian Information Criterion (BIC) is written as

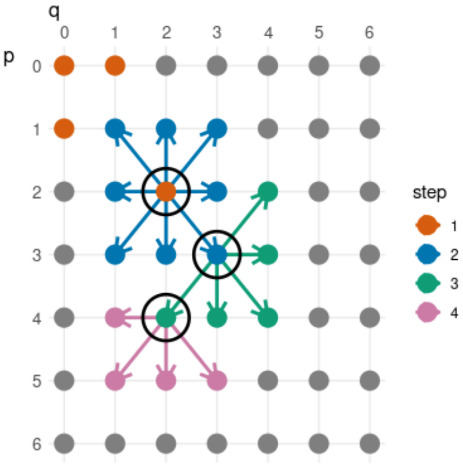
$$\text{BIC} = -2\log L + k\log n$$

- Good models are obtained by minimising the AIC, AICc, or BIC.

Automatic selection

Hyndman-Khandakar algorithm for automatic ARIMA modelling
1. The number of differences $0 \leq d \leq 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
a. Four initial models are fitted: ∘ $\text{ARIMA}(0, d, 0)$, ∘ $\text{ARIMA}(2, d, 2)$, ∘ $\text{ARIMA}(1, d, 0)$, ∘ $\text{ARIMA}(0, d, 1)$. A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted: ∘ $\text{ARIMA}(0, d, 0)$ without a constant.
b. The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model."
c. Variations on the current model are considered: ∘ vary p and/or q from the current model by ± 1; ∘ include/exclude c from the current model. The best model considered so far (either the current model or one of these variations) becomes the new current model.
d. Repeat Step 2(c) until no lower AICc can be found.

Automatic selection



Modelling procedure

1. Plot the data and identify unusual observations.
2. Transform the data, if necessary, to stabilise the variance.
3. If the data are non-stationary, take first differences until the data is stationary.
4. Examine the ACF/PACF: Is an ARIMA(p,d,0) or ARIMA(0,d,q) model appropriate?
5. Try your chosen model and use AIC/AICc/BIC to compare with other models.
6. Check the residuals by plotting an ACF and doing a portmanteau test. If they don't look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.

NB: The Hyndman-Khandakar algorithm only takes care of steps 3–5.

Seasonal ARIMA

- ARIMA models are also capable of modelling a wide range of seasonal data.
- A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far.
- It is written as

$$\text{ARIMA}(p, d, q)(P, D, Q)_m$$

  where m is the seasonal period and we use upper-case notation for the seasonal parts of the model.
- The seasonal terms are similar to the non-seasonal terms, but involve backshifts of the seasonal period:

$$\begin{aligned}(1 - \phi_1 B - \cdots - \phi_p B^p)(1 - \Phi_1 B^m - \cdots - \Phi_P B^{mP})(1 - B)^d(1 - B^m)^D y_t = \\ = (1 + \theta_1 B + \cdots + \theta_q B^q)(1 + \Theta_1 B^m + \cdots + \Theta_Q B^{mQ})\varepsilon_t\end{aligned}$$

- Example: an ARIMA$(1, 1, 1)(1, 1, 1)_4$ model can be written as

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

Seasonal ARIMA

- The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF, respectively.
  - e.g.[1]: an ARIMA$(0,0,0)(0,0,1)_{12}$ will have a significant spike at lag 12 in the ACF and exponential decay in the seasonal lags of the PACF.
  - e.g.[2]: Example 2: an ARIMA$(0,0,0)(1,0,0)_{12}$ will have a significant spike at lag 12 in the PACF and exponential decay in the seasonal lags of the ACF.
- Example[1]: Monthly US leisure and hospitality employment
- Example[2]: Corticosteroid drug sales in Australia

Dynamic Regression

- The time series models in the previous two chapters allow for the inclusion of information from past observations of a series, but not for the inclusion of other information that may also be relevant.
- For example, the effects of holidays, competitor activity, changes in the law, the wider economy, or other external variables, may explain some of the historical variation and may lead to more accurate forecasts.
- Recall the multiple linear regression model

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t$$

  where y_t is a linear function of the k predictor variables $(x_{1,t}, \ldots, x_{k,t})$ and ε_t is white noise.
- In dynamic regression models, we allow the errors from a regression to contain autocorrelation:

$$\begin{aligned} y_t &= \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t \\ (1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d \eta_t &= (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t \\ \varepsilon_t &\sim \mathsf{N}(0, \sigma^2) \end{aligned}$$

- The error η_t follows an ARIMA(p, d, q) model and only the ARIMA model errors

Estimation

- When estimating a multiple regression model, we typically look for the values of $\beta_0, \ldots, \beta_k$ that minimise the residual sum of squares $\sum_{i=1}^{T} \varepsilon_i^2$.
- If we ignore autocorrelation in the errors, it is equivalent to minimising $\sum_{i=1}^{T} \eta_i^2$ instead, and several problems arise:

1. The estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer the best estimates, since some information has been ignored in the calculation;
2. Any statistical tests associated with the model (e.g. t-tests for the model coefficients) will be incorrect;
3. The AIC/AICc/BIC values are no longer a good guide as to which model is best for forecasting;
4. The $p-$values associated with the coefficients will typically be too small, and some predictor variables will appear important when they are not (this is known as *spurious regression*).

Estimation

- Minimising $\sum_{i=1}^{T} \varepsilon_i^2$ avoids all these problems.
- Alternatively, maximum likelihood estimation can be used and will give similar estimates of the coefficients.
- An important consideration when estimating a regression with ARMA errors is that all of the variables in the model **must first be stationary**.
- We have to check that y_t and all of the predictors $(x_{1,t}, \ldots, x_{k,t})$ appear to be stationary.
- If we estimate the model when any of these are non-stationary we will not obtain consistent estimates of the coefficients.

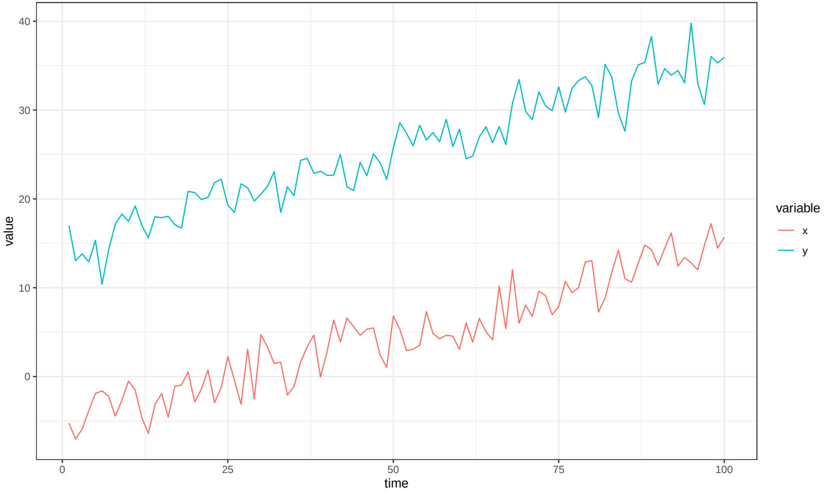
Spurious Regression: Example

- Is there a relationship between the two time series below?

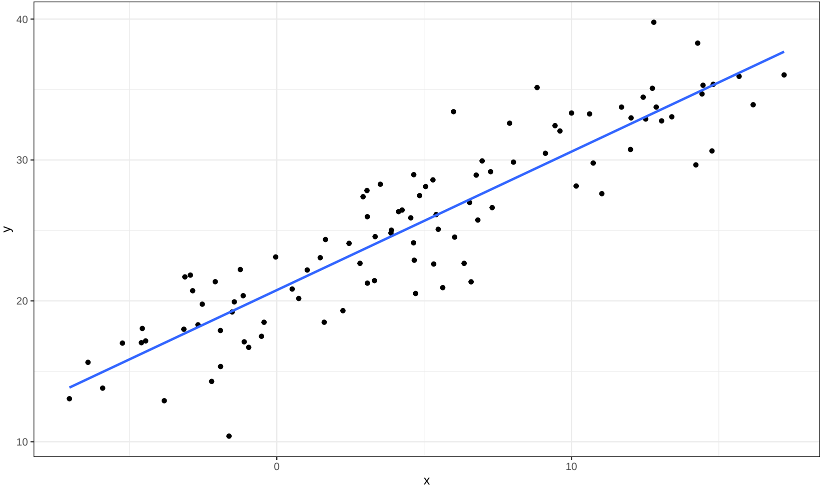
$$
\begin{aligned}
y_t &= 15 + 0.2t + \varepsilon_t \\
x_t &= -5 + 0.2t + \epsilon_t
\end{aligned}
$$

- y is not explained by x, they are independent.
- Let's have a look at a few plots.

Spurious Regression: Example



Spurious Regression: Example



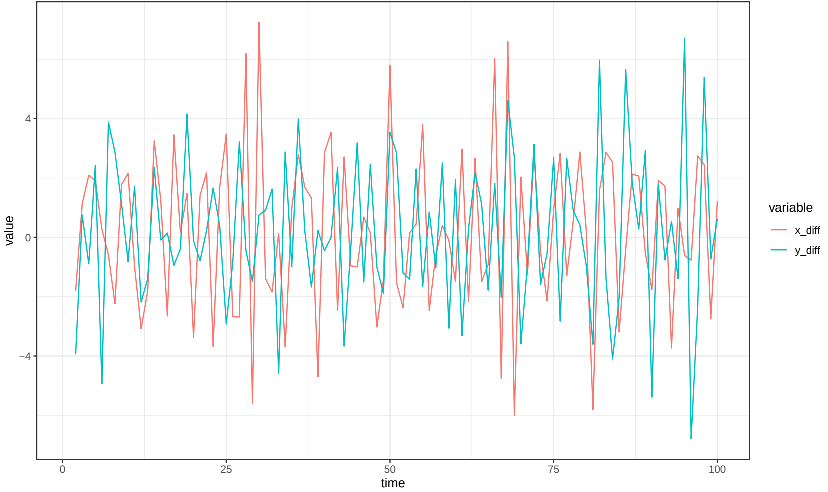
Spurious Regression: Example

- Although they are independent, the correlation between x and y is 0.91
- In a linear regression setting,

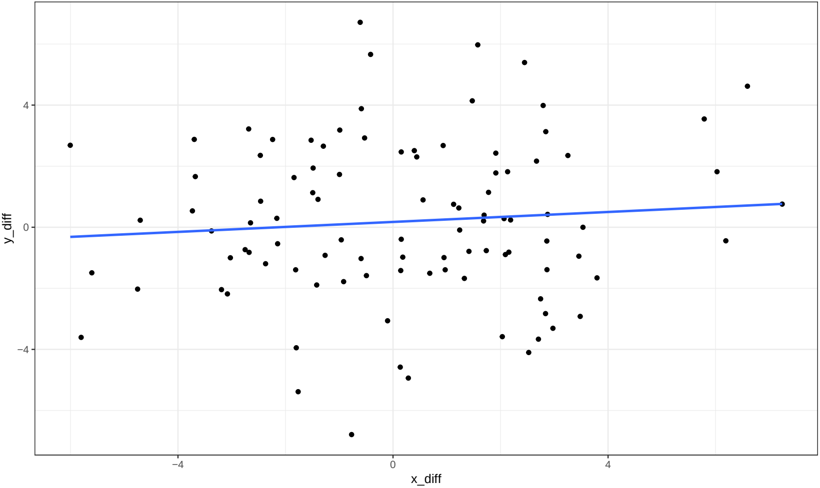
$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$$

  our estimate for β_1 would be significant, even though the variables are unrelated.
- This is because they have the same time trend.
- Now see what happens if we apply differencing to obtain stationary series.

Spurious Regression: Example



Spurious Regression: Example



Regression with ARIMA errors

- The function `ARIMA()` will fit a regression model with ARIMA errors if exogenous regressors are included in the formula.
- E.g. the command

```
ARIMA(y ~ x + pdq(1,1,0))
```

fits the model

$$
\begin{aligned}
y'_t &= \beta_1 x'_t + \eta'_t \\
\eta'_t &= \phi_1 \eta'_{t-1} + \varepsilon_t
\end{aligned}
$$

which is equivalent to fitting

$$y_t = \beta_0 + \beta_1 x_t + \eta_t$$

where η_t is an ARIMA$(1, 1, 0)$ error.

- If we specify `ARIMA(y ~ x)` only, the function will select the best values for p, d and q for the ARIMA error.

Example: US Personal Consumption and Income

- Data on quarterly changes in personal consumption expenditure and personal disposable income from 1970 to 2019 Q2.
- We would like to forecast changes in expenditure based on changes in income.
- A change in income does not necessarily translate to an instant change in consumption (e.g., after the loss of a job, it may take a few months for expenses to be reduced to allow for the new circumstances).
- However, we will ignore this complexity in this example and try to measure the instantaneous effect of the average change of income on the average change of consumption expenditure.

Forecasting

- To forecast using a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model, and combine the results.
- As with ordinary regression models, in order to obtain forecasts we first need to forecast the predictors.
- When the predictors are known into the future (e.g., calendar-related variables such as time, day-of-week, etc.), this is straightforward.
- But when the predictors are themselves unknown, we must either model them separately, or use assumed future values for each predictor.
- Example[1]: US personal consumption and income
- Example[2]: Daily electricity demand