INTRO

- Recall that for a stationary time series $\{X_t\}$, with mean μ, and auto-covariance function $\gamma(h)$, the <u>best linear predictor</u> in terms of mean square error, of X_{n+h} based on $X_1, \dots, X_n$ is given by

$$Pred(X_{n+h}|X_n, \dots, X_1) := \mu + (a_1, \dots, a_n)^T \begin{pmatrix} X_n - \mu \\ X_2 - \mu \\ \vdots \\ X_1 - \mu \end{pmatrix}$$

  - Where $a := (a_1, \dots, a_n)$ satisfies the equation

$$\Gamma a = \gamma_{(n,h)} := \big(\gamma(h), \dots, \gamma(h+n-1)\big)^T$$

  - Where Γ is the n*n matrix with ij$^{\text{th}}$-element, $\Gamma_{ij} = \gamma(|i-j|)$

- While this is important as a theoretical result it does not have many direct applications because most observed time series are not stationary.
- Two key observations, though, do mean we can extend the theorem to cases of practical importance
  - (1) Fortunately, a large number of observed time series do fall into the class of being linear transformations of stationary process
    - These include examples with non-constant means, strong seasonal effects, and non-constant variances, which are not themselves stationary.
  - (2) The BEST LINEAR PREDICTOR only uses linear functions, means and covariances and we know how all these behave under linear transformations.
    - This allows a powerful generalization of the following theorem
    - These generalizations do not come completely for "free", and some extra assumptions will be needed
    - Note that this theorem does not use the stationarity condition

> ***Theorem.*4.1.1** Let U be a random variable, such that $E(U^2) < \infty$, and $\boldsymbol{W}$ an n-dimensional random vector with a finite variance-covariance matrix,
>
> $$\Gamma := Cov(\boldsymbol{W}, \boldsymbol{W}).$$
>
> In terms of mean square error the best linear predictor, $Pred(U|\boldsymbol{W})$, of U based on $\boldsymbol{W} = (X_1, \dots, X_n)$ satisfies the following results:

***Theorem.*4.1.1**

(i) $Pred(U|\mathbf{W}) = E(U) + \mathbf{a}^T(\mathbf{W} - E(\mathbf{W}))$ where $\Gamma\mathbf{a} = Cov(U, \mathbf{W})$.

(ii) $E([U - Pred(U|\mathbf{W})]) = 0$ i.e. the estimator is unbiased.

(iii) $E([U - Pred(U|\mathbf{W})]\,\mathbf{W}) = \mathbf{0}$, the zero vector.

(iv) $Pred(\alpha_0 + \alpha_1 U_1 + \alpha_2 U_2|\mathbf{W}) = \alpha_0 + \alpha_1 Pred(U_1|\mathbf{W}) + \alpha_2 Pred(U_2|\mathbf{W})$, i.e. the predictor is linear.

(v) $Pred(W_i|\mathbf{W}) = W_i$.

NON-STATIONARY TIMES SERIES

- The following set of examples illustrates the range of useful linear transformations of stationary processes which can be used in modelling real data

- **Example 4.2.1** Consider the random walk with drift, defined by initial condition X_0 and for $t \geq 1$

$$X_t := X_0 + \sum_{i=1}^{t} Z_i$$

  - Where $Z_t \sim (i.i.d.)N(\mu, \sigma^2)$, or white noise, then $E(X_t) = \mu t$ and $Var(X_t) = t\sigma^2$.
  - If $\mu \neq 0$, the mean is dependent on t, so it is not stationary.
  - Further, the variance depends on t so its not variance stationary

- We first note that $\{X_t\}$ is a linear function of the stationary process $\{Z_t\}$ and X_0, and for $t \geq 1$

$$Z_t = \nabla X_t = X_t - X_{t-1} = (1 - B)X_t$$

  - So there is a linear transformation, differencing, which recovers the stationary process. Here we have

$$\nabla X_t \sim \mu + ARMA(0,0)$$

  - A stationary process with a mean that needs to be estimated

- **Example 4.2.2** Let $Z_t \sim WN(0, \sigma^2)$ and define $X_t = a_0 + a_1 t + Z_t$, using a linear transformation, we have that $E(X_t) = a_0 + a_1 t$, so $\{X_t\}$ is not mean stationary. Applying the differencing operator gives

$$\nabla X_t = a_1 + (Z_t - Z_{t-1})$$

- Now $(Z_t - Z_{t-1})$ is an MA(1) process with $\theta_1 = -1$, and is stationary, hence we have

$$\nabla X_t \sim a_1 + ARMA(0,1)$$

- Where a mean parameter needs to be estimated

ARIMA MODELLING

- The above examples show that non-stationary processes can be closely linked to stationary ones, with differencing being the key tool. We can define a broad class of such examples

- **Definition (d^{th} – order differencing)** the operator ∇^d is called the d^{th} – order differencing operator and is defined by

$$\nabla^d X_t = (1-B)^d X_t$$

  - For example

$$\nabla^2 X_t = (1-B)^2 (X_t - X_{t-1}) = X_t - 2X_{t-1} + X_{t-2}$$

- **Definition (ARIMA model)** if d is a non-negative integer, then $\{X_t\}$ is an $ARIMA(p,d,q)$ process if

$$Y_t \coloneqq \nabla^d X_t$$

  - Is a causal $ARMA(p,q)$ process

FORECASTING ARIMA MODELS

- If a non-stationary time series is close to being stationary – in the sense that there exists simple linear transformation which reduce the process to a stationary one and with only a small number of parameters – then we might hope that optimal h-step ahead would be possible

- **Example (4.2.1)** suppose we wish to compute the best linear prediction of X_{n+1} based on $X_0, \dots, X_n$, i.e., $Pred(X_{n+1} | X_0, \dots, X_n)$
  - Note that

$$X_{n+1} \coloneqq X_0 + \sum_{i=1}^{n+1} Z_i$$

  - Then we have

$$Pred(X_{n+1}|X_0,\dots,X_n) = Pred\left(X_0 + \sum_{i=1}^{n+1} Z_i\,|X_0,\dots,X_n\right)$$
$$= Pred(X_n + Z_{n+1}|X_0,\dots,X_n)$$
$$= X_n + Pred(Z_{n+1}|X_0,\dots,X_n)$$

- Since $X_0,\dots,X_n$ are an invertible linear function of $X_0,\ Z_1,\dots,Z_n$ they will have the same best linear predictors, so we have
$$Pred(X_{n+1}|X_0,\dots,X_n) = X_n + Pred(Z_{n+1}|X_0,Z_1\dots,Z_n)$$

- To compute this with the same tools we have seen in Chapter 3, we have to make an additional assumption: Y_t is uncorrelated with X_0. This means that we have
$$Pred(X_{n+1}|X_0,\dots,X_n) = X_n + Pred(Z_{n+1}|,Z_1\dots,Z_n)$$
  - Where the second term is the best linear predictor for an $ARMA(0,0)$ process with an unknown mean

- **Example (ARIMA(1,2,1) forecast)** Suppose that we have the model
$$X_t \sim ARIMA(1,2,1)$$
  - And we want for forecast X_{n+2} based on $X_{-1},\dots,X_n$. By definition we have that $Z_t \coloneqq \nabla^2 X_t = X_t - 2X_{t-1} + X_{t-2}$ is a causal $ARMA(1,1)$ process. So
$$X_{n+2} = Z_{n+2} + 2X_{n+1} - X_n$$
  - And
$$P(X_{t+2}|X_{-1},\dots,X_n) = P(Z_{n+2}|X_{-1},\dots,X_n) + P(X_{n+1}|X_{-1},\dots,X_n) - X_n$$
- We treat of these terms in turn
  - First we have, since $Z_1,\dots,Z_n$ is a linear function of $X_1,\dots,X_n$, that
$$P(Z_{n+2}|X_{-1},\dots,X_n) = P(Z_{n+2}|X_{-1},X_0,Z_1,\dots,Z_n)$$
    - And if we make the new assumption that the SZ variables are uncorrelated with X_{-1},X_0, then we have
$$P(Z_{n+2}|X_{-1},\dots,X_n) = P(Z_{n+2}|Z_1,\dots,Z_n)$$
    - Which is the h = 2 forecast of a causal $ARMA(1,1)$
  - The term $P(X_{n+1}|X_{-1},\dots,X_n)$, using the same argument as above, can be written as
$$P(Z_{n+1}|Z_1,\dots,Z_n) + 2X_n - X_{n-1}$$
    - And hence, again is computable using the theory of stationary processes in Chapter 3

- Thus by recursively estimating previous terms we can build us an estimate of X_{n+2} as long as we include extra conditions on the lack of correlation between initial conditions and the driving stationary process

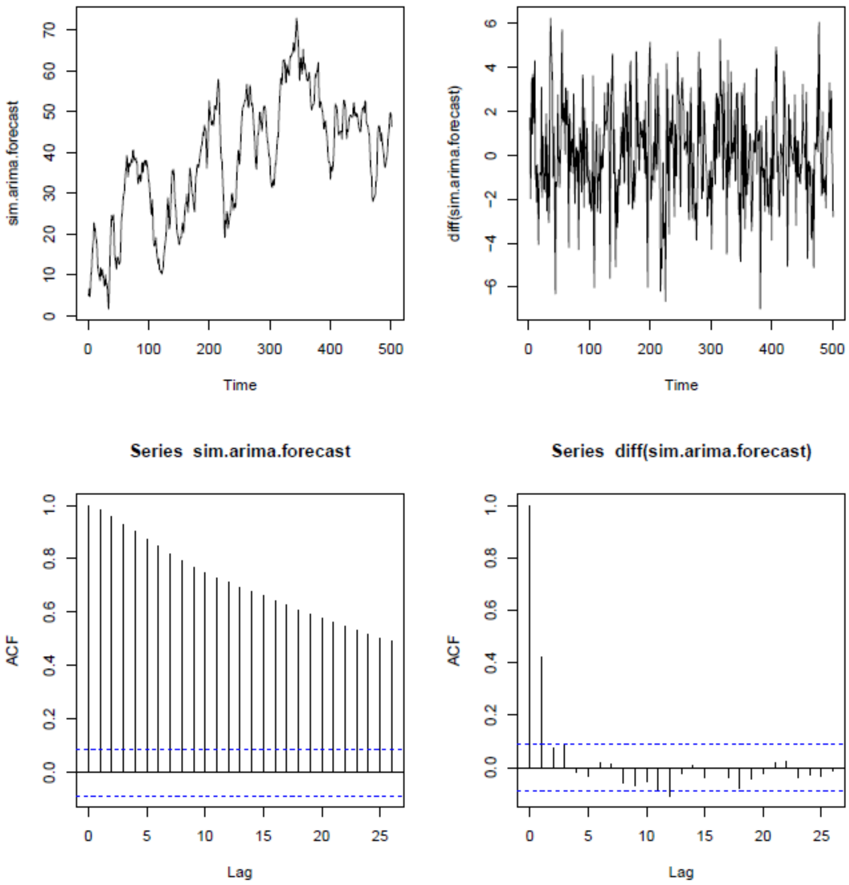
- The ideas behind the above example directly extends to produce h-step ahead forecasts for any $ARIMA(p,d,q)$ models, where we work recursively building the h-step forecast from (h-1) step forecast … Each time we do need to add assumptions on the lack of correlation between initial conditions and the driving causal $ARMA(p,q)$ process.

USING R FOR ARIMA MODELLING

- There are two types of problems
  - How to work with the models for which p, d, q are known, and
  - How to find appropriate values for p, d, q

FITTING ARIMA MODELS

- **Example (ARIMA model in R)** Suppose we have a time series in the time series object *sim.arima.forecast* and that this has been correctly identified as being an $ARIMA(0,1,1)$ model – that is of the form Example 4.2.2
  - we can plot the data in the following figures



  - The top left panel shows the time series and the right panel shows the plot of the first difference, by using the *diff()* function
  - We see that the original data does not "look" stationary – it is not centered around a time

independent mean with a fixed variance. If we use the *acf()* function – which really we shouldn't as its only defined for stationary series – we get a plot where the correlations only very slowly decay to zero. On the other hand the auto-correlation of the first difference function looks like that of a stationary series

- If we know the structure of the model, we can fit it using the ARIMA() function:

```
> arima(sim.arima.forecast, order=c(0,1,1))
Coefficients:
         ma1
      0.5407
s.e.  0.0414
sigma^2 estimated as 4.095:  log likelihood = -1062.07,  aic = 2128.14
```

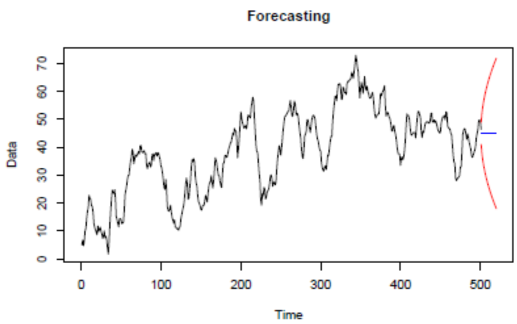
getting the parameter estimates, standard errors, log-likelihood and AIC values with an assumption of Gaussian errors. The output from this function can go into the `predict( )` function in the following way.

```
> predict(arima(sim.arima.forecast, order=c(0,1,1)), n.ahead=2)
$pred
Time Series:
Start = 502
End = 503
Frequency = 1
[1] 44.97003 44.97003

$se
Time Series:
Start = 502
End = 503

 Frequency = 1
 [1] 2.023546 3.71677
```

- The following figure shows the prediction of h = 20 steps ahead for this model.



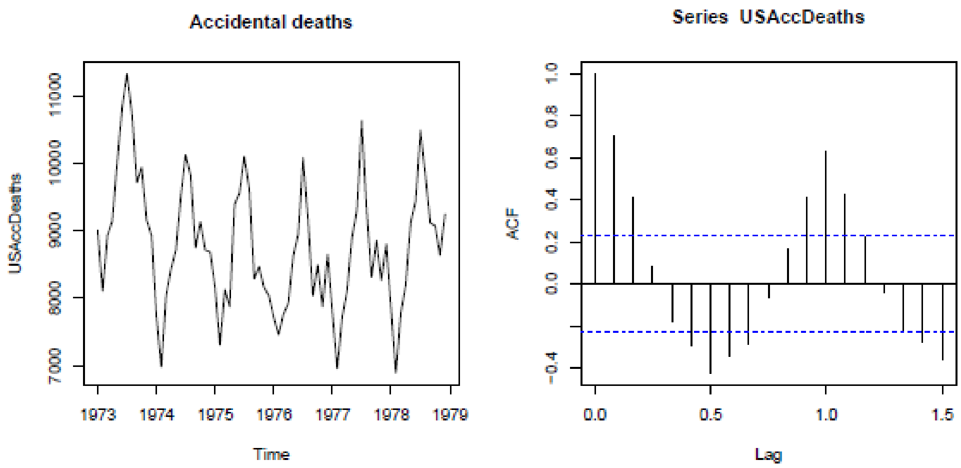
➢ We see, unlike the stationary time series example, that the uncertainty in the forecast grows very large for large h. This is a result of the fact that we are trying to forecast a non-stationary process

SARIMA Modelling

- There are other forms of non-stationarity that the Box-Jenkins approach can tackle. The most important is seasonality

- **Definition (Seasonal Component)** Let $Z_t \sim WN(0, \sigma^2)$ and, for ease of interpretation, let us assume t is recorded on a yearly time scale, but we have observation per month.
  - Following R, we would record January 2015 as *2015.00*, February 2015 as $2015.083 = 2015\frac{1}{12}$
  - Let $a_i\ i = 1, \dots, 12$ be constants associate with the month $(i-1)/12$ such that $\sum_{i=1}^{12} a_i = 0$ and define the function *mon(t)* as the function which returns the month associated with time t.
- The process $X_t := a_{mon(t)} + Z_t$ where $\{Z_t\} \sim ARIMA(p, d, q)$ is said to have a seasonal component. It will not be stationary if any of the a_i values are nonzero

- **Example 4.5.2**



- **Definition (Lag s difference)** The lag s difference operator B^s is defined by
$$B^s X_t = X_{t-s}$$
  - The corresponding difference operator Δ^s is then defined as
$$\Delta^s := (I - B^s)$$
- Differencing at appropriate lags can remove seasonality, hence are another example where a linear function of a non-stationary time series can be stationary. This gives rise, in a very similar way, to the definition of the ARIMA model, to a wider class of models which can model seasonal time series

- **Definition (Seasonal ARIMA (SARIMA) model)** If d and D are nonnegative integers, then $\{X_t\}$ is a seasonal $ARIMA(p, d, q)(P, D, Q)$ process with period s if the differenced series
$$Y_t = (1-B)^d (1-B^s)^D X_t$$
  - Is a causal ARMA process defined by
$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_t$$

➢ Where $Z_t \sim WN(0, \sigma^2)$ and

$$\begin{aligned}
\phi(z) &:= 1 - \phi_1 z - \cdots - \phi_p z^p \\
\Phi(z) &:= 1 - \Phi_1 z - \cdots - \Phi_P z^P \\
\theta(z) &:= 1 + \theta_1 z + \cdots + \theta_q z^q \\
\Theta(z) &:= 1 + \Theta_1 z + \cdots + \Theta_Q z^Q
\end{aligned}$$

➢ The corresponding process is causal if and only if all the roots of $\phi(z)$ and $\Phi(z)$ lie outside the unit circle

Using R for SARIMA Modelling

- There are two types of problem:
  - ➢ How to work with the model for which p, d, q, P, D, Q are known, and
  - ➢ How to find appropriate values for p, d, q, P, D, Q

- **Example (SARIMA model in R)** Let us look at a particular example for concreteness. The R data object USAccDeaths contains a monthly time series of accidental deaths in the USA in the period 1973-1978
  - ➢ The following figure shows
    - ✧ The data and we see a strong seasonal effect.
    - ✧ Differencing at lag 12,
    - ✧ And there seems to be a trend so another difference is done

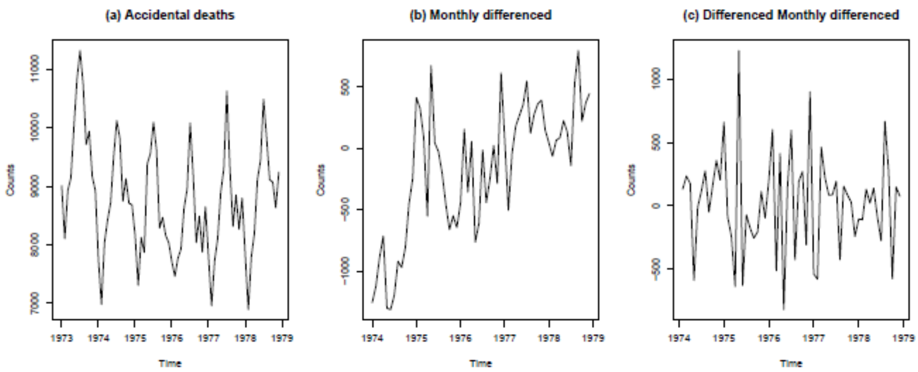


Figure 4.5: US accident data

➢ The R code is

```
> data(USAccDeaths)
> plot(USAccDeaths, main="(a) Accidental deaths", ylab="Counts")
> plot(diff(USAccDeaths, lag=12) )
> plot(diff(diff(USAccDeaths, lag=12)))
```

- So it looks like a SARIMA model might be appropriate. Let us suppose that we know that $(p, d, q) = (1,1,1)$ and $(P, D, Q) = (0,1,1)$ we than fit the appropriate SARIMA model using

```
> mod2 <- arima(USAccDeaths, order = c(1, 1,1),
     seasonal = list(order = c(0, 1, 1), period = 12)
```

- Where *mod2* is an arima object containing the estimated parameters and other details of the fit

```
> mod2
Call:
arima(x = USAccDeaths, order = c(1, 1, 1),
   seasonal = list(order = c(0, 1, 1),  period = 12))
Coefficients:
         ar1      ma1     sma1
      0.0979  -0.5109  -0.5437
s.e.  0.3111   0.2736   0.1784
sigma^2 estimated as 99453:  log likelihood = -425.39,  aic = 858.78
```

We need to see if the fit is 'good' so we look the residuals using

```
> acf(mod2$residuals, main="ACF residuals")
> acf(mod2$residuals, type="partial", main="PACF residuals")
```

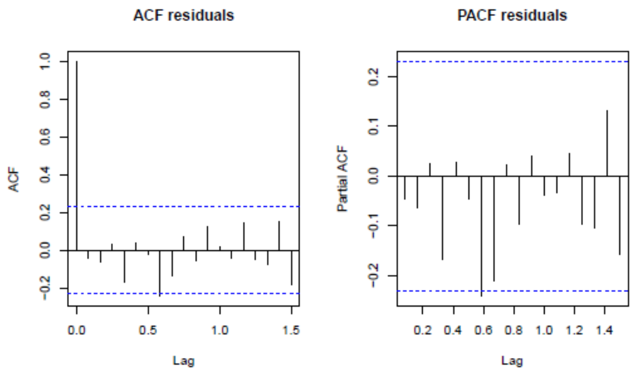


Figure 4.6: US accident data

- We can then make predictions for two years ahead using

```
> predict(mod2, n.ahead=24)
```

and the results can be found in Fig. 4.7 with the red line the point forecast and the blue the prediction intervals.

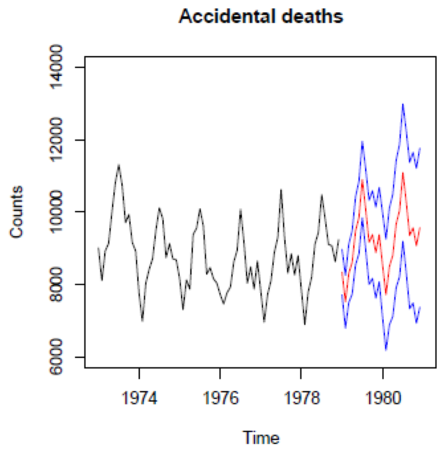


Figure 4.7: US accident data

Estimating the SARIMA Structure

- We can inspect acf and pacf plot to look for possible p, d, q and P, D, Q values. This usually requires a good deal of experience to do reliably and a good deal of data

- **Example (SARIMA model example**) We have already seen by looking at differencing plots that d = 1, D = 1 are plausible – although not the only possible values. We will stick to those and look at acf and pacf plots of these differences in the following figure to try and guess P, 1 we look for the same types of patterns discussed above – i.e. exponential decay and finding a bound after which all estimates are zero – but only for integer values.

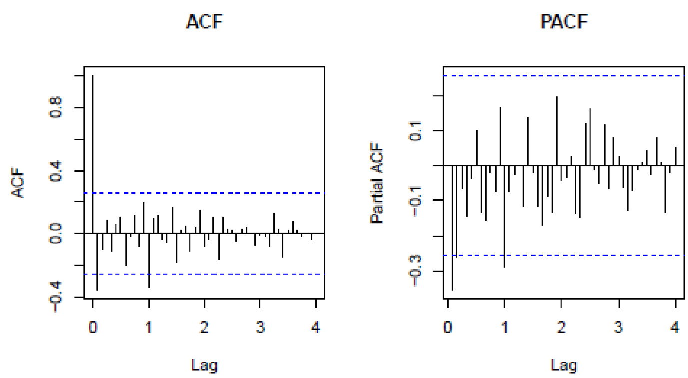


Figure 4.8: US accident data

- Here we see P = 0, Q = 1 is possible, although not the only possible choice.
- The think about the p, q values, we look for patterns at the month 1/12, 2/12, …, lags. Here we see that p = 1, q = 1 is possible, as are other choices

- An alternative method to interpreting this plots is look at all models with small p, q, P, Q values and look at the AIC value given by the *arima*() function output

The Box Jenkins Approach

- In this chapter we have looked at the Box-Jenkins methodology for forecasting. It is based on the idea that many stationary processes can be parsimoniously represented by ARMA(p, q) models – that is only a relatively small number of parameters are needed.
  - As we have seen in Chapter 2, if we can fit observed data with only a small number of parameters then often we can get reasonable forecasting models
- However, most applications involve some form of non-stationarity and the methodology deals with this by appropriate differencing combined with associated independence assumptions. We can summarize the steps as follows

> (a) Model Identification. This uses the R functions `plot`, `diff`, `acf` etc.
> (b) Estimation. Once the structure of the model is identified we used the `arima` function to estimate the parameters
> (c) Diagnostic checking. Using the residuals from the model fit we can check to see if the model is 'satisfactory'.
> (d) Consider alternative models if necessary.

Tests for Stationarity

- One of the important stages of the approach is to decide if a suitably difference time series is stationary. This can be rather subjective if purely graphical methods are used.
- We can use more formal tests which focus on different kinds ways that stationarity can fail

Definition.(Phillips-Perron test) The Phillips-Perron test is a test that the time series has a 'unit root' with a stationary alternative.
In R we use the function `PP.test`. For the data in Example 4.6.1 selecting $d = D = 1$ reduced the model to stationarity. If we try and test this formally we get

```
> PP.test(diff(diff(USAccDeaths, lag=12)))

Phillips-Perron Unit Root Test

data:  diff(diff(USAccDeaths, lag = 12))
Dickey-Fuller = -11.8917,
Truncation lag parameter = 3, p-value = 0.01
```

So the method rejects the hypothesis of a unit root.

Definition 4.7.2. (Runs test) If there was a trend in the sequence the series would have a pattern in if it was above or below its mean. The runs test looks at the the frequency of runs of a binary data set. The null hypothesis is that the series is random.

```
> library(tseries)
> x <- diff(diff(USAccDeaths, lag=12))
> y <- factor(sign(x - mean(x)))
> runs.test(y)

Runs Test

data:  y
Standard Normal = 0.2439, p-value = 0.8073
alternative hypothesis: two.sided
```

Here we convert the continuous time series to a binary one and then use the `runs.test( )` from the `tseries` library.

CRITICISMS OF APPROACH

- (a) Differencing to attain stationarity may introduce spurious auto-correlations. For example with monthly data it can occur at lag 11.
- (b) If can be difficult, just from data, to distinguish between different SARIMA models and these may give different forecasts
- (c) A reasonably large number of data points is needed to implement the approach with Box and Jenkins stating that 50-100 observations are needed
- (d) ARIMA models may be harder for practitioners to understand than linear trends plus seasonal effect models