ASTA

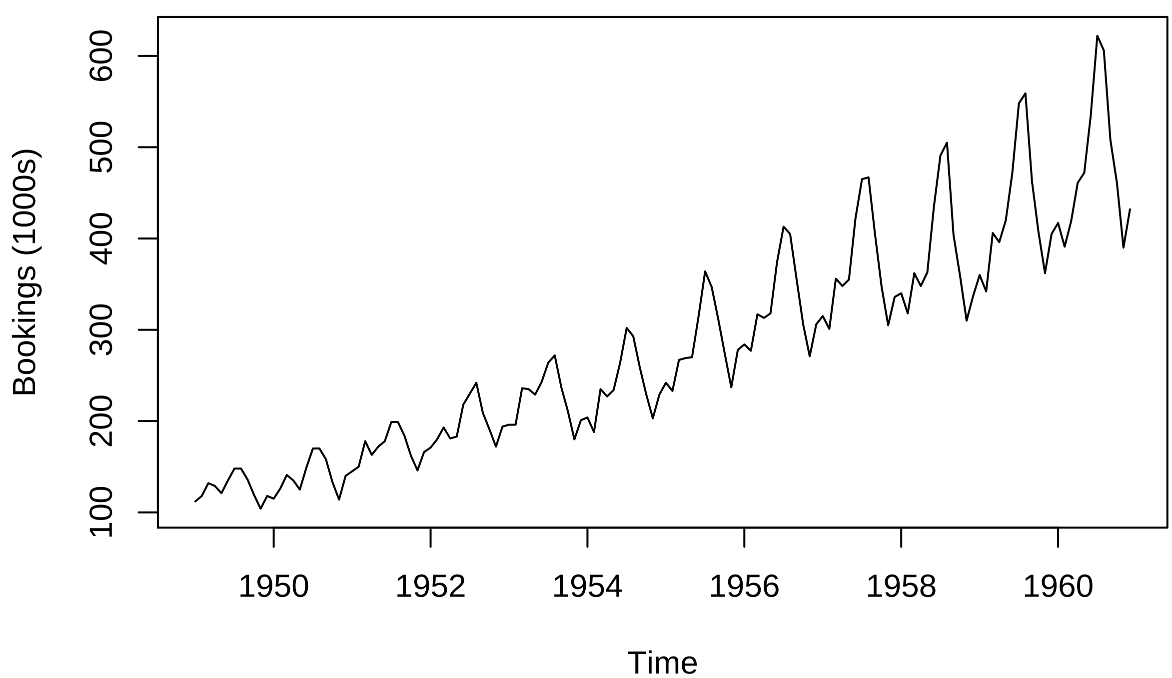
The ASTA team

Contents

1 Data examples

- Before we define a stochastic process, we start by considering some examples of data.
- Here we are considering data which is a collection of the same variable measured at different points in time, i.e. x_t indexed by time t in some discrete set, often $t = 1, \dots, n$. We will always assume the data is observed at equidistant points in time (i.e. same time difference between consecutive observations).
- An example is the monthly number of international airline passengers 1949-1960:

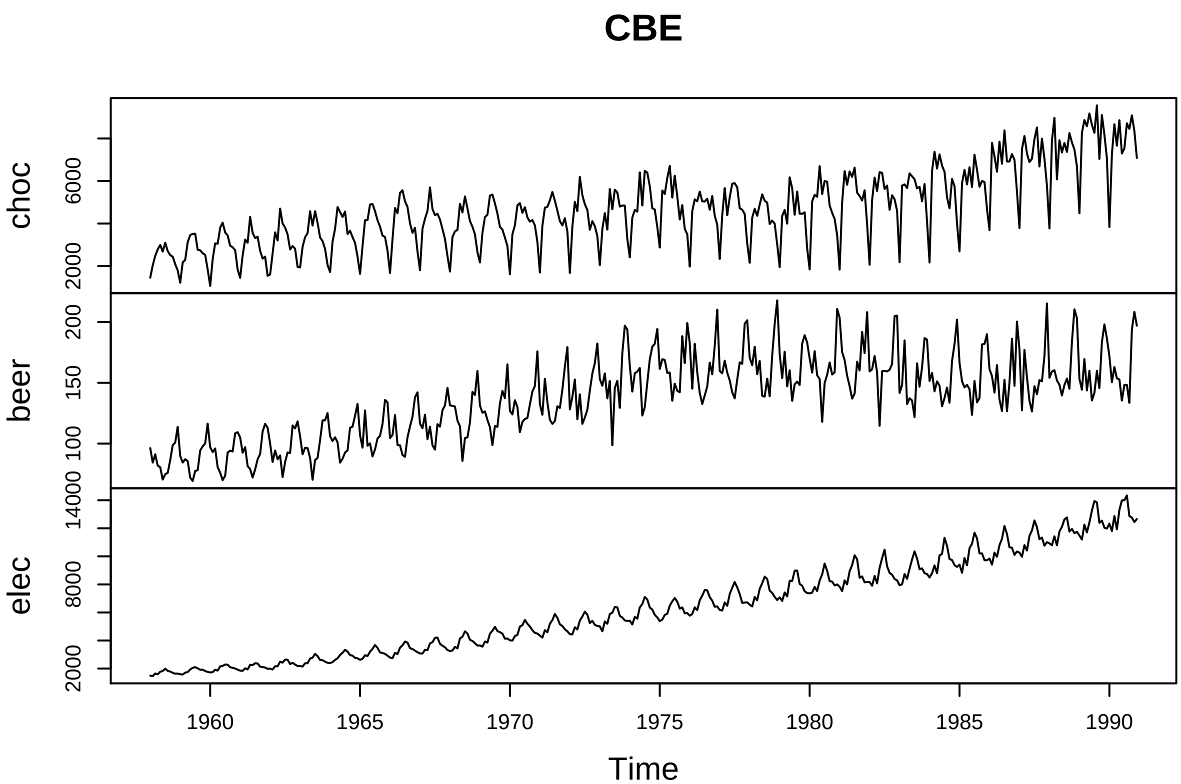
```
AP <- AirPassengers
plot(AP, ylab = "Bookings (1000s)")
```



- Another example is monthly time series from Jan. 1958 to Dec. 1990 of supply of three goods in Australia:

  - Electricity (Giga Watt hours)
  - Beer (Mega liters)
  - Chocolate (tonnes)

```
CBEdata <- read.table("https://asta.math.aau.dk/eng/static/datasets?file=cbe.dat", header = TRUE)
CBE <- ts(CBEdata, start = 1958, freq = 12)
plot(CBE)
```



- There are many examples of data evolving over time and measured at discrete times, and we would like to make realistic models to analyse and make predictions for such datasets.

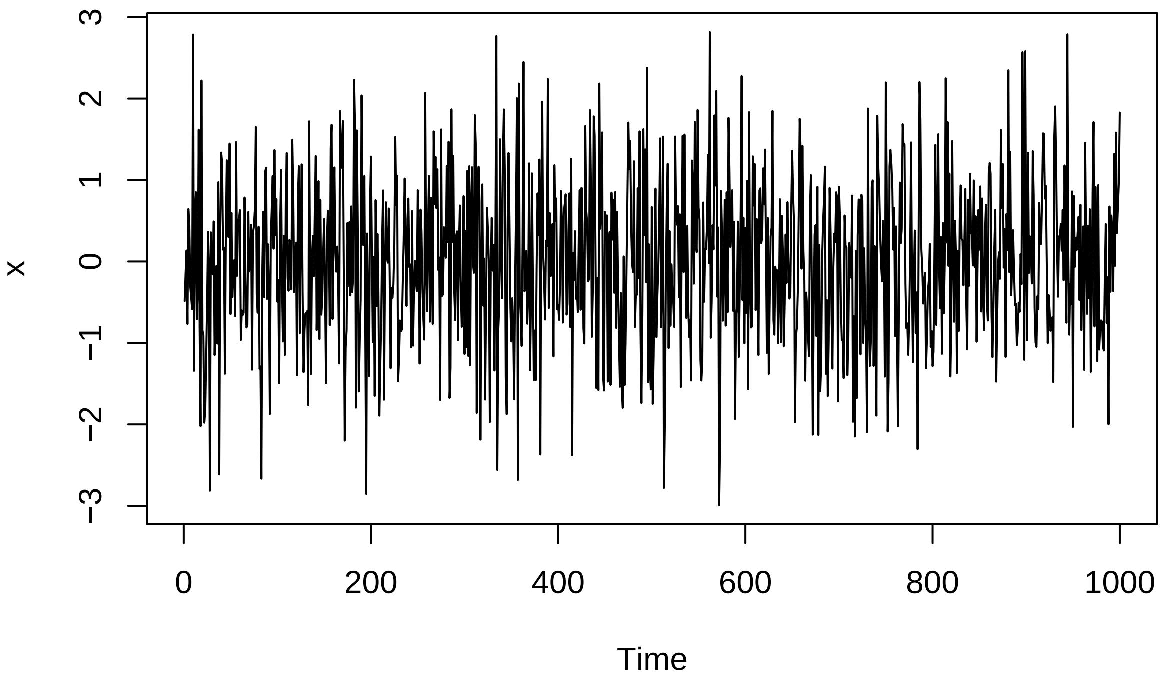
2 Stochastic processes

- A stochastic process is a number of stochastic variables X_t indexed by time t, for example $t = 1, \dots, n$ or $t \in \mathbb{Z}$.
- Typically there will be dependence between X_t at different time points.
- We will often define a stochastic process by defining the distribution of X_t conditionally on X_s for $s < t$.

3 Example 1: White noise

- White noise is the simplest example of a stochastic process: Here X_t are independent and identically distributed random variables with mean 0 and variance σ^2. It is called Gaussian white noise, if X_t is Gaussian.
- Due to independence everywhere, white noise is typically not a good model for real data, but it is a building block for more complicated stochastic processes.

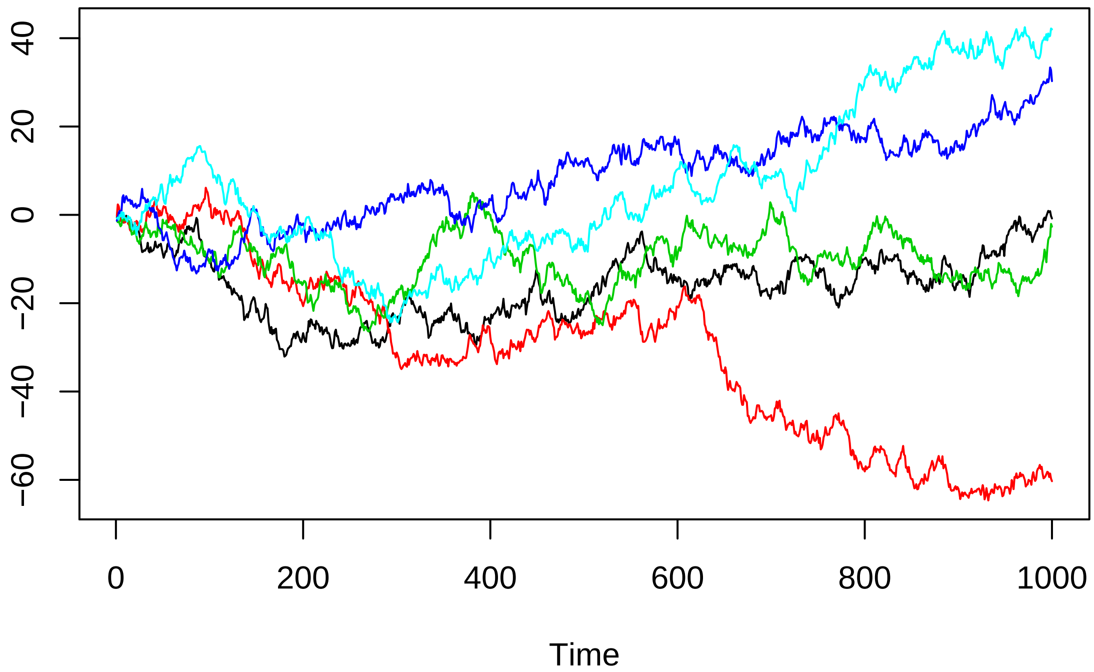
```
x = rnorm(1000,0,1)
ts.plot(x)
```



4 Example 2: Random walk

- A random walk is defined by $x_t = x_{t-1} + w_t$, where w_t is white noise.
- The random walk may come back to zero after some time, but often it has a tendency to wander of in some random direction.
- Examples:

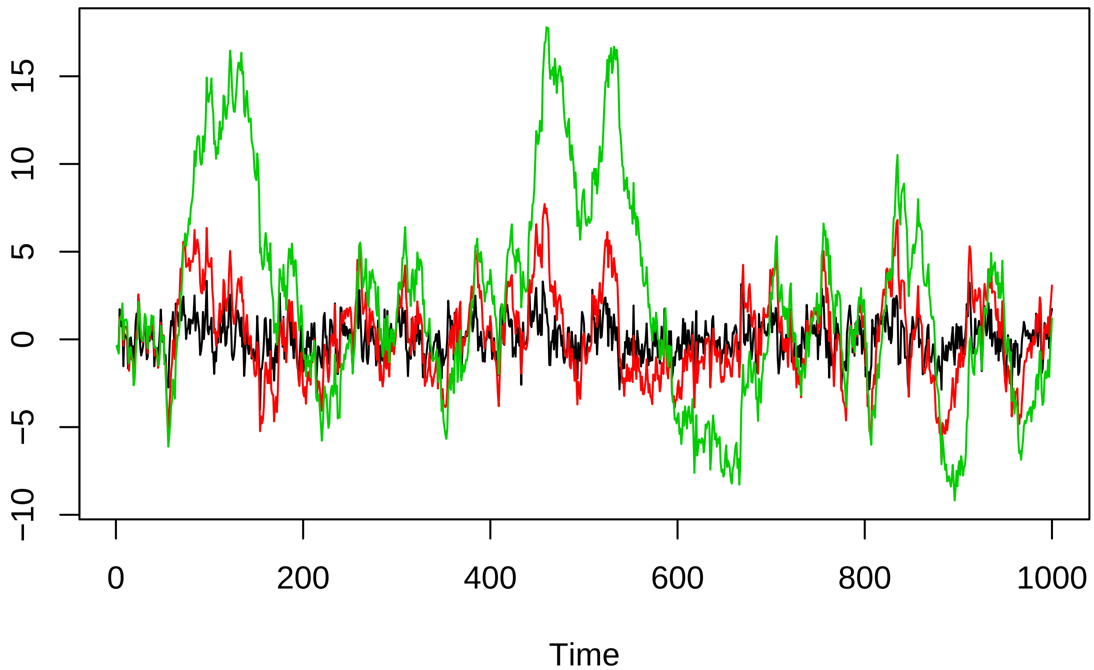
```
x = matrix(0,1000,5)
for (i in 1:5) x[,i] = cumsum(rnorm(1000,0,1))
ts.plot(x,col=1:5)
```



5 Example 3: First order autoregressive process

- A first order autoregressive process, AR(1), is defined by $x_t = \alpha x_{t-1} + w_t$, where w_t is white noise and $\alpha \in \mathbb{R}$.
- For $\alpha = 0$ we get white noise, and for $\alpha = 1$ we get a random walk.
- Next time we will consider autoregressive processes in much more detail and higher order, where they become quite flexible models for data.
- Examples:

```
w = ts(rnorm(1000))
x1 = filter(w,0.5,method="recursive")
x2 = filter(w,0.9,method="recursive")
x3 = filter(w,0.99,method="recursive")
ts.plot(x1,x2,x3,col=1:3)
```



6 Mean function

- The mean function of a stochastic process is given by

$$\mu_t = \mathbb{E}(x_t)$$

- All three examples have a constant mean of $\mu_t = 0$. For example, the random walk:

$$\mathbb{E}(x_t) = \mathbb{E}(x_{t-1} + w_t) = \mathbb{E}(x_{t-1}) + \mathbb{E}(w_t) = \mathbb{E}(x_{t-1})$$

$$\Rightarrow \quad \mathbb{E}(x_t) = \mathbb{E}(x_{t-1}) = \cdots = \mathbb{E}(x_0) = \mathbb{E}(0) = 0$$

- The mean function shows the mean behavior of the process, but individual simulations may move far away from this. For example, the random walk has a tendency to move far away from the mean. White noise on the other hand will stay close to the mean.

7 Autocovariance/autocorrelation functions

- The autocovariance function is given by

$$\gamma(t, t+h) = \text{Cov}(x_t, x_{t+h}) = \mathbb{E}((x_t - \mu_t)(x_{t+h} - \mu_{t+h}))$$

- Note that $\gamma(t,t) = \sigma_t^2$ is the variance at time t.
- The autocorrelation function (ACF) is a normalised version of the autocovariance function

$$\rho(t, t+h) = \frac{\text{Cov}(x_t, x_{t+h})}{\sigma_t \sigma_{t+h}}$$

- It holds that $\rho(t,t) = 1$, and $\rho(t, t+h)$ is in the interval [-1,1] for any h.
- The autocorrelation function shows how much x_t and x_{t+h} are related:
- If x_t and x_{t+h} are independent, then $\rho(t, t+h) = 0$
- If $\rho(t, t+h)$ is close to one, then x_t and x_{t+h} tends to be either high or low at the same time.
- If $\rho(t, t+h)$ is close to minus one, then when x_t is high x_{t+h} tends to be low and vice versa.

8 Stationarity

- We call a stochastic process second order stationary if

- the mean is constant, $\mu_t = \mu$, and
- the autocovariance function only depends on the time difference, $\gamma(t, t+h) = \gamma(h)$.
- In this case the variance $\sigma_t^2 = \gamma(t,t) = \gamma(0)$ is also constant.
- If a process is second order stationary, then also $\rho(t, t+h) = \rho(h)$, i.e. it is a function of only h and is easier to work with and plot.
- Intuitively stationarity means that the process behaves in the same way nomatter which times we look at.
- There are other kinds of stationarity, but in this course by stationarity we always mean second order stationarity.

9 Stationarity and autocorrelation - example

- The autoregressive process is stationary if $\alpha \in (-1, 1)$. We can calculate the autocovariance and autocorrelation.
- First observe

$$x_{t+h} = \alpha x_{t+h-1} + w_{t+h} = \cdots = \alpha^h x_t + \sum_{i=0}^{h-1} \alpha^i w_{t+h-i}$$

- Then we calculate the autocovariance:

$$\gamma(h) = \mathbb{E}((x_t - \mu_t)(x_{t+h} - \mu_{t+h})) = \mathbb{E}(x_t x_{t+h}) = \mathbb{E}(x_t(\alpha^h x_t + \sum_{i=0}^{h-1} \alpha^i w_{t+h-i})) = \mathbb{E}(x_t \alpha^h x_t) + \mathbb{E}(x_t \sum_{i=0}^{h-1} \alpha^i w_{t+h-i}) = \alpha^h \mathbb{E}(x_t^2$$

- We need to calculate the variance of x_t, where we use $\text{Var}(x_t) = \text{Var}(x_{t-1})$ by stationarity:

$$x_t = \alpha x_{t-1} + w_t \Rightarrow \text{Var}(x_t) = \text{Var}(\alpha x_{t-1}) + \text{Var}(w_t)$$

$$\Rightarrow \text{Var}(x_t) = \alpha^2 \text{Var}(x_t) + \sigma^2 \Rightarrow \text{Var}(x_t) = \frac{\sigma^2}{1 - \alpha^2}$$

- Finally the autocorrelation:

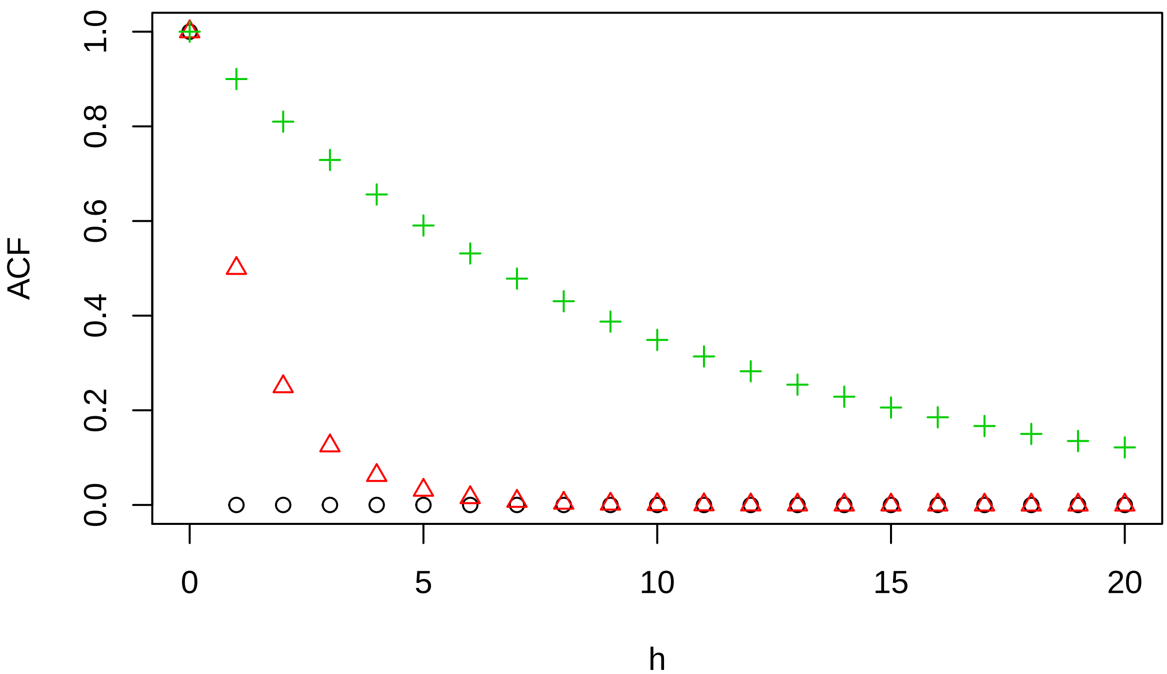
$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \frac{\alpha^h \sigma^2/(1-\alpha^2)}{\sigma^2/(1-\alpha^2)} = \alpha^h$$

- Note that this is only for $h \geq 0$. For arbitrary h we have $\rho(h) = \alpha^{|h|}$ since $\rho(-h) = \rho(h)$.
- White noise is a special case of an autoregressive process with $\alpha = 0$.

$$\rho(h) = 0^{|h|} = \begin{cases} 1, & h = 0 \\ 0, & h \neq 0 \end{cases}$$

- Random walk is not stationary.
- The ACF for a stationary AR(1):

```
h = 0:20
acf1 = 0^h  # AR(1) with alpha = 0 (or white noise)
acf2 = 0.5^h  # AR(1) with alpha = 0.5
acf3 = 0.9^h  # Ar(1) with alpha = 0.9
plot(matrix(rep(h,3),3),cbind(acf1,acf2,acf3),col=rep(1:3,each=length(h)),
     pch=rep(1:3,each = length(h)),xlab="h",ylab="ACF")
```



10 Estimation

- The mean and autocovariance/autocorrelation functions are theoretical constructions defined for stochastic processes, but what about data? Here we have to estimate them.
- We will focus on the stationary case, i.e. the case where the data looks stationary.
- The (constant) mean can be estimated the usual way:

$$\hat{\mu} = \bar{x} = \sum_{t=1}^{n} x_t$$

- The autocovariance function can be estimated as follows (remember it only depends on h, not on t in the case of stationarity):

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_t - \bar{x})(x_{t+h} - \bar{x})$$

- The (constant) variance is estimated as $\hat{\sigma}^2 = \hat{\gamma}(0)$.
- An estimate of the autocorrelation function is obtained as

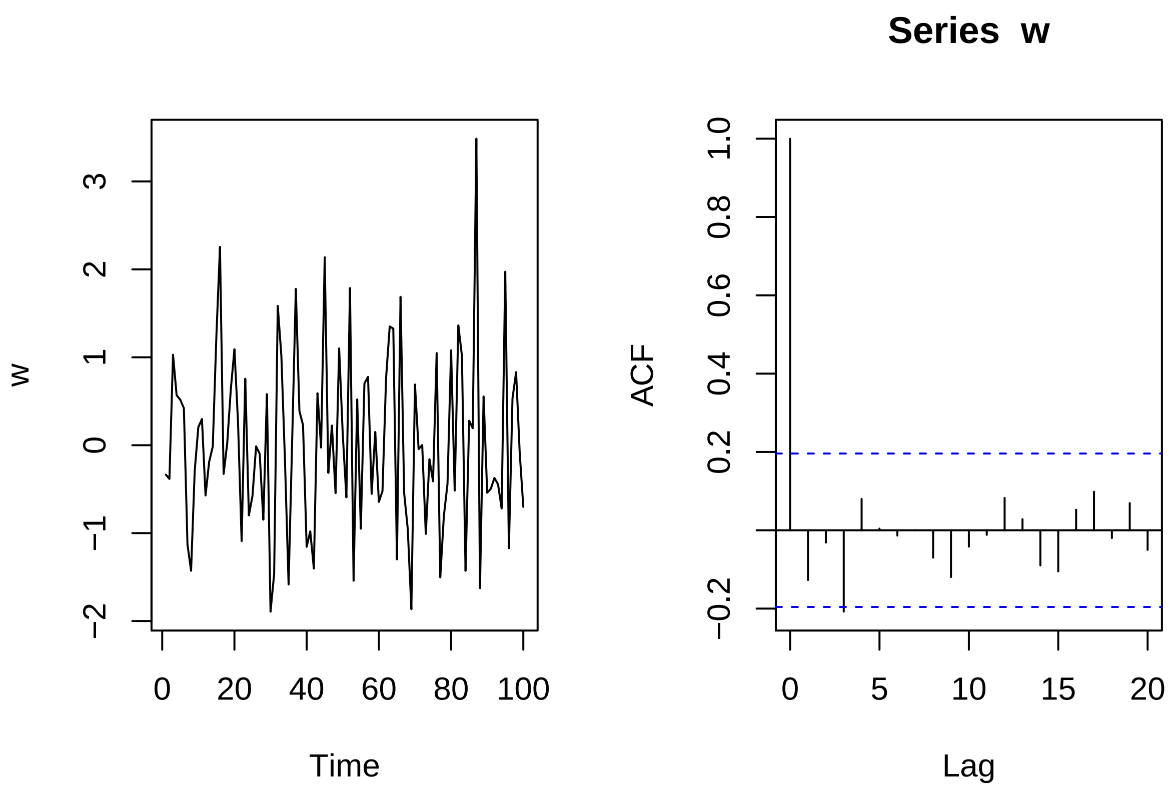
$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

  This is known as the correlogram, and has many practical uses.

11 Examples of correlograms

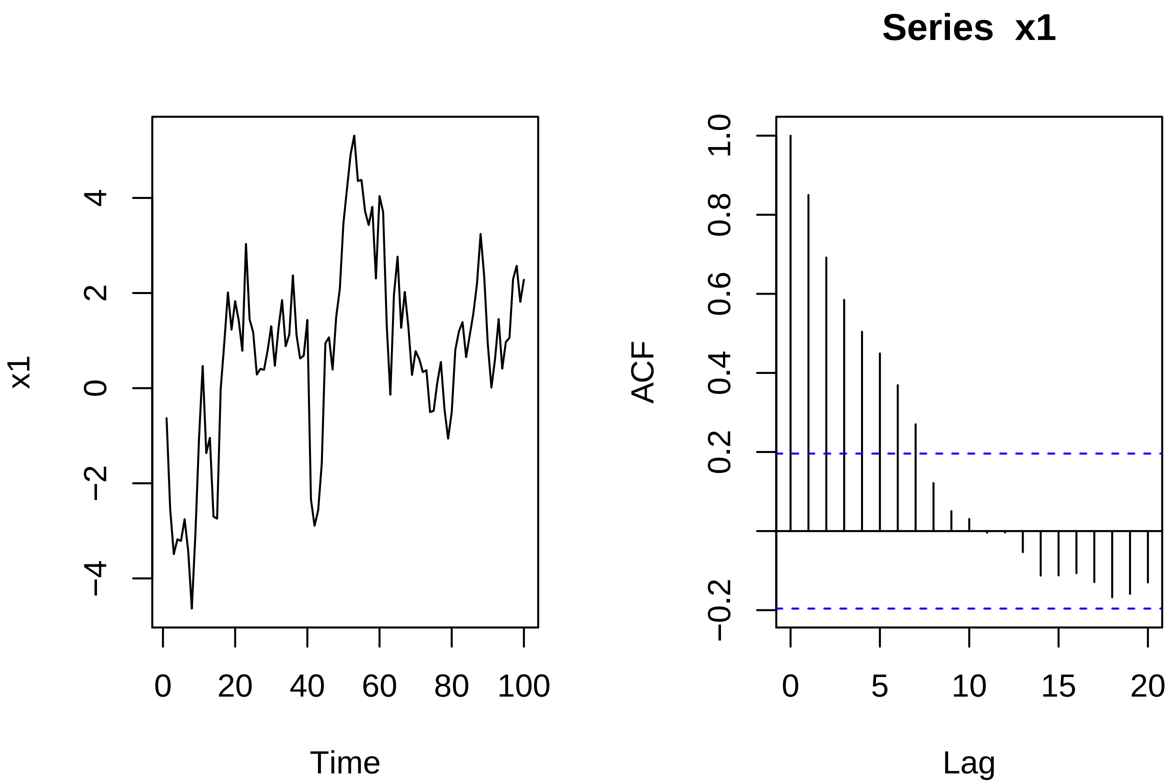
- To get an idea of how a correlogram looks, we make simulated data, and use the estimation formulas.
- White noise:

```r
w = ts(rnorm(100))
par(mfrow=c(1,2)); plot(w); acf(w)
```



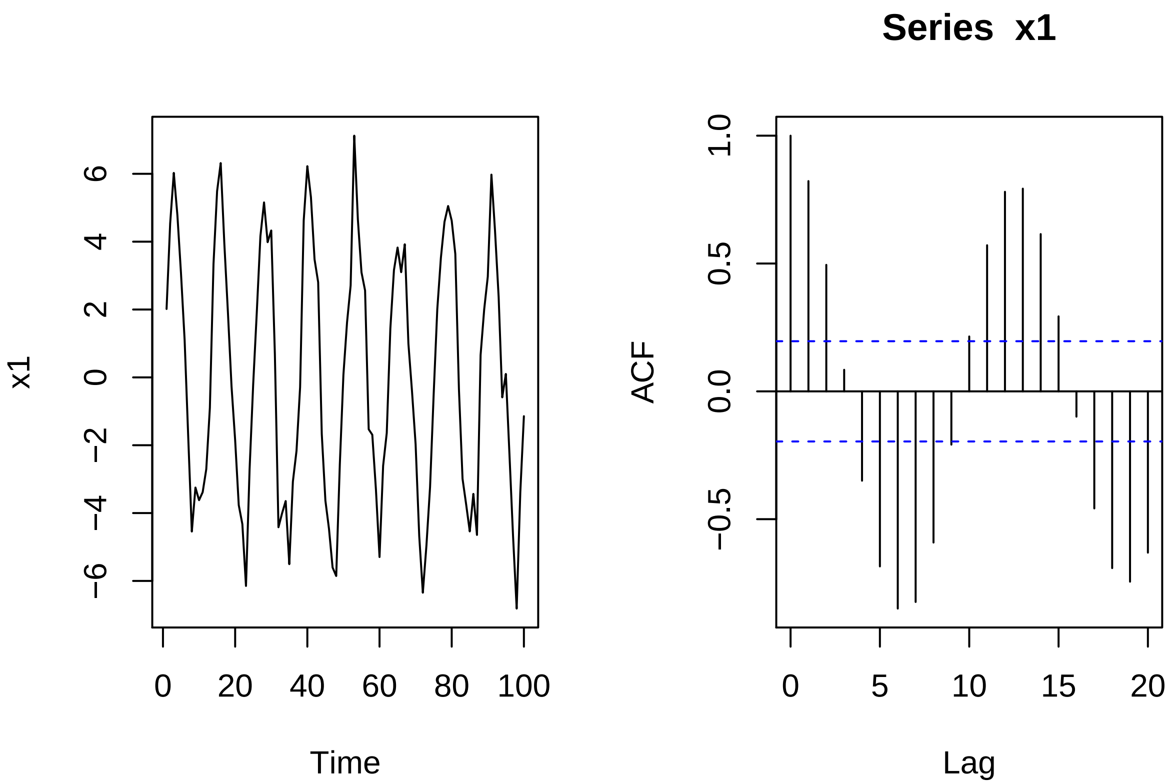
- AR(1) process with $\alpha = 0.9$:

```r
w = ts(rnorm(100))
x1 = filter(w,0.9,method="recursive")
par(mfrow=c(1,2)); plot(x1); acf(x1)
```



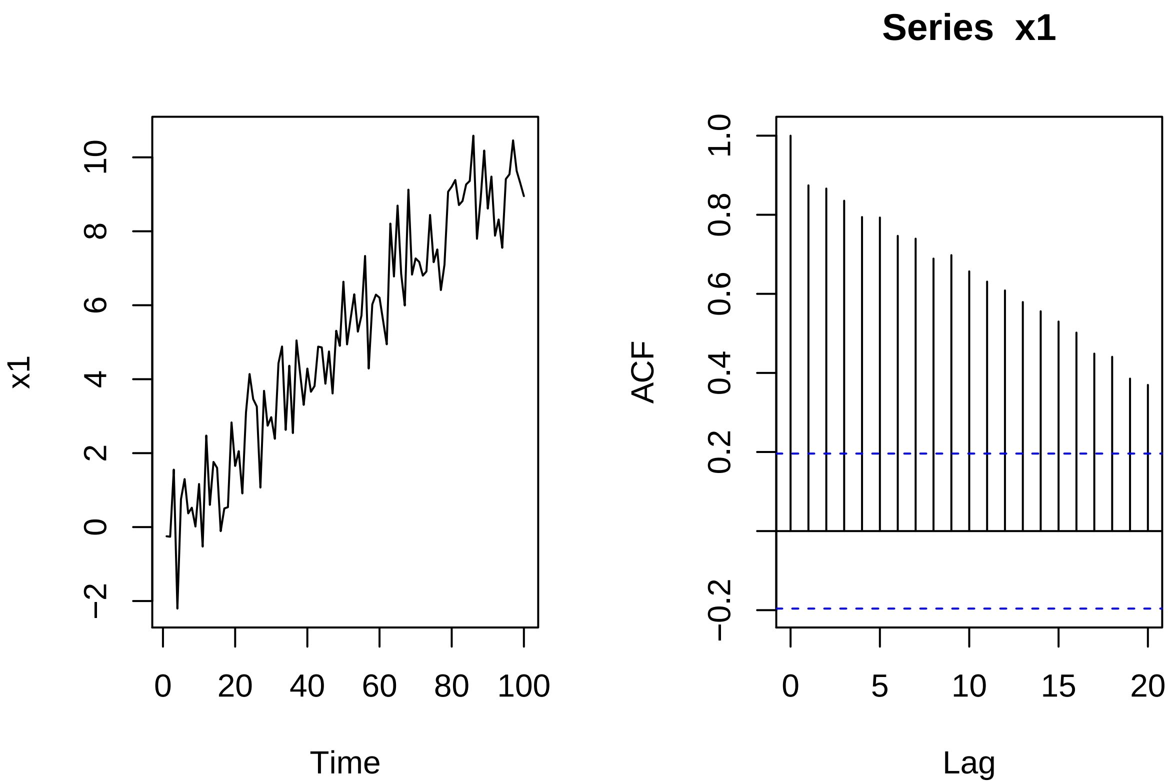
- Sine curve with added white noise:

```
w = ts(rnorm(100))
x1 = 5*sin(0.5*(1:100)) + w
par(mfrow=c(1,2)); plot(x1); acf(x1)
```



- Straight line with added white noise:

```
w = ts(rnorm(100))
x1 = 0.1*(1:100) + w
par(mfrow=c(1,2)); plot(x1); acf(x1)
```

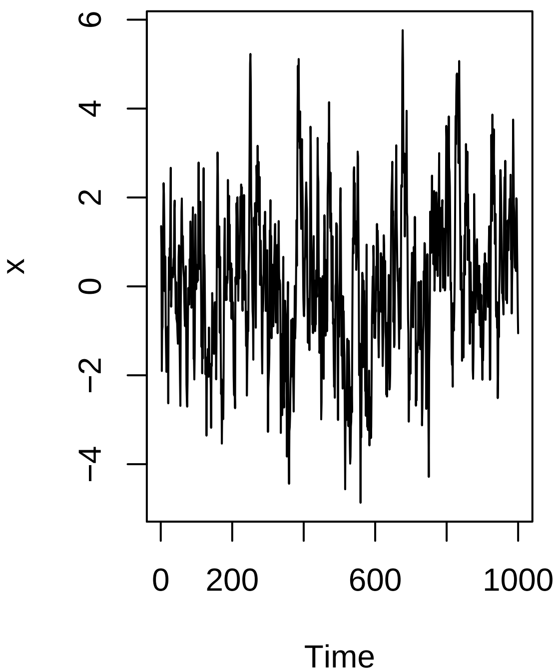


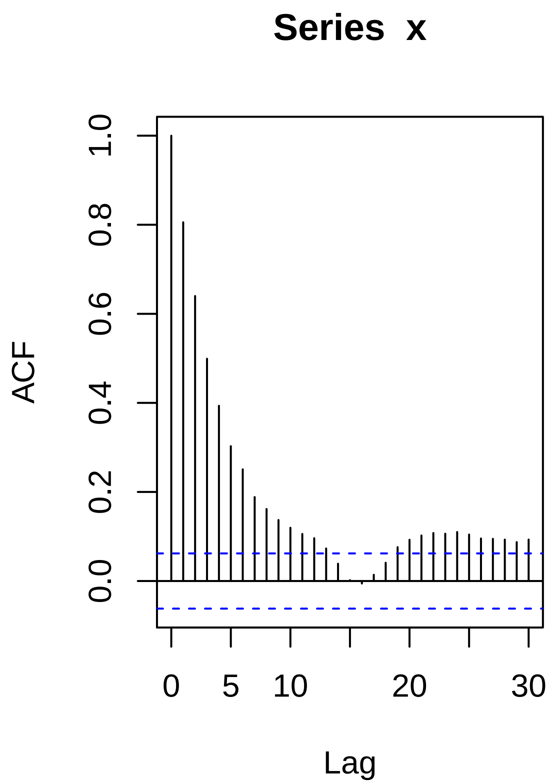
- Note that even though $\rho(h)$ is only well-defined for stationary models, we can plug any data (stationary or otherwise) into the estimation formula. The estimate tells a lot about the data.

12 Stationary and non-stationary data

- We will primarily look at stationary processes the next time, but these will not be good models for data, if the data does not look stationary. First we need to check whether the assumption of stationarity is okay.
- One check is visual inspection of a plot of x_t vs t to see whether there is any indication of non-stationarity.
- Another visual check is a plot of the correlogram. If this tends very slowly to zero, this indicates non-stationarity.
- Simulated data - stationary:

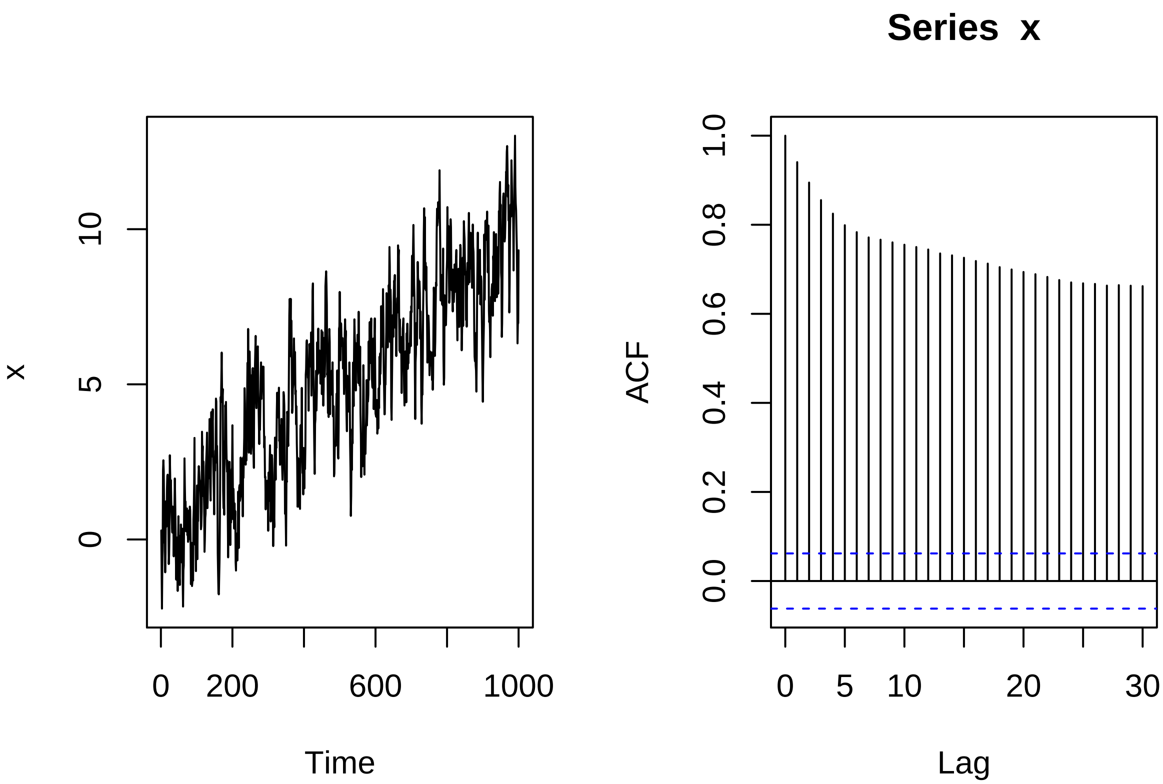
```
w = ts(rnorm(1000))
x = filter(w,0.8,method="recursive")
par(mfrow=c(1,2)); plot(x); acf(x)
```





- Non-stationary simulated example (there is an increasing tendency):

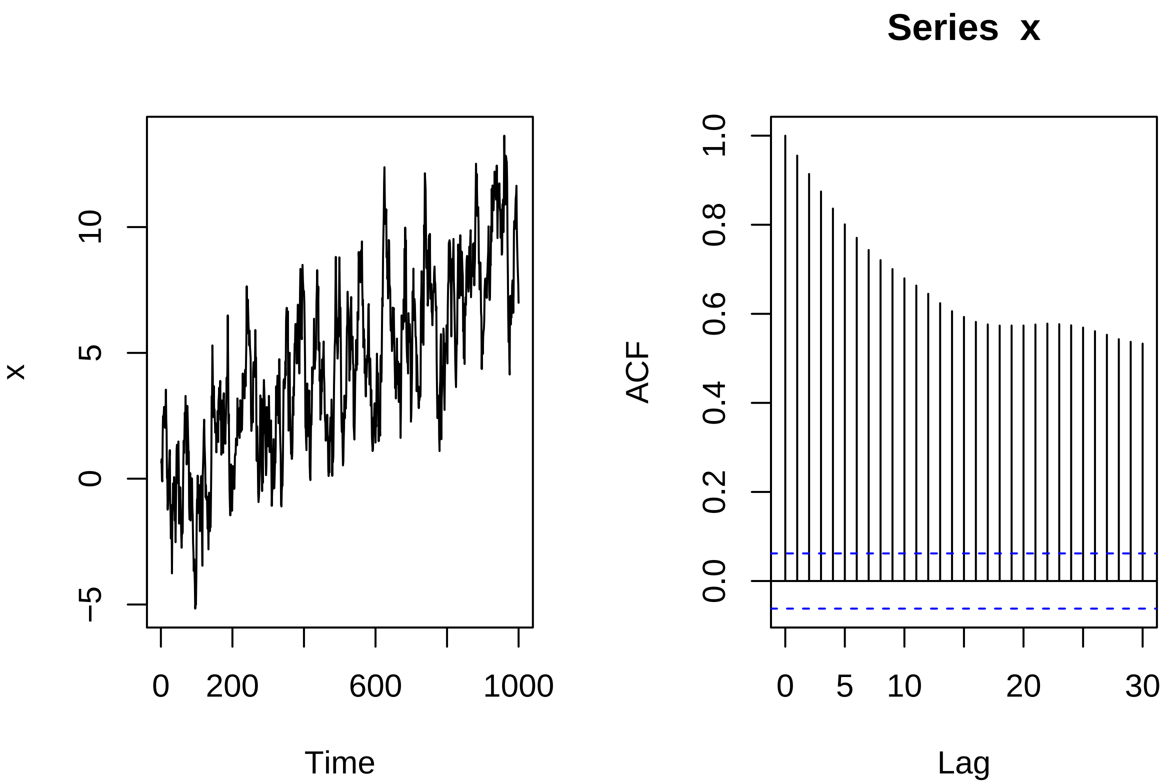
```
w = ts(rnorm(1000))
x = filter(w,0.8,method="recursive") + (1:1000)/100
par(mfrow=c(1,2)); plot(x); acf(x)
```



13 Detrending data

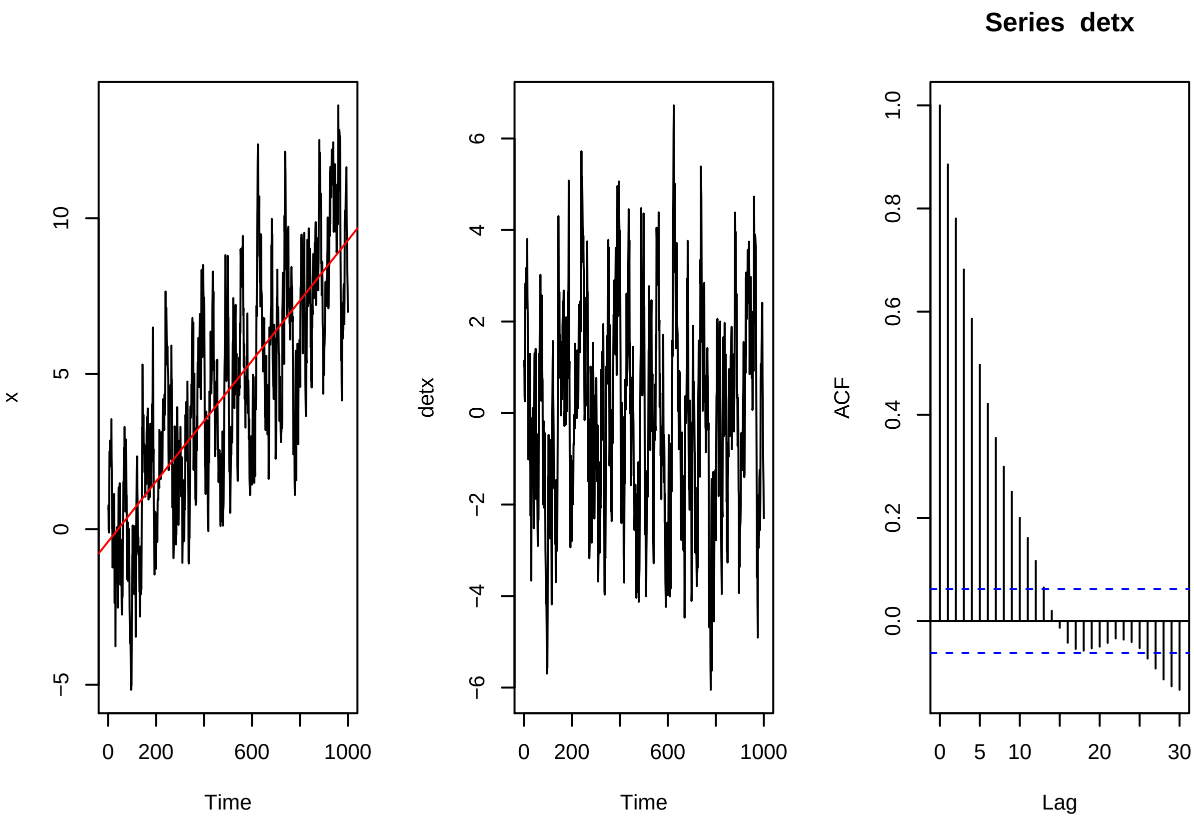
- If the data does not seem stationary, we can try find a stationary data hidden in it by transforming the data. There are many methods for this, and we will only have a brief look at one possibility based on the linear regression models you have seen earlier.
- Consider the following simulated data:

```
w = ts(rnorm(1000))
x = filter(w,0.9,method="recursive") + (1:1000)/100
par(mfrow=c(1,2)); plot(x); acf(x)
```



- Ignoring that this is simulated data, it would seem that the data is located around a straight line, so one possibility is to fit a straight line, and the consider the residuals as our new data. We then do all the model fitting to this detrended data, and work with this.

```
lin = lm(x~I(1:1000))
detx = resid(lm(x~I(1:1000)))
par(mfrow=c(1,3)); ts.plot(x); abline(lin,col=2); ts.plot(detx); acf(detx)
```



#

Basic stochastic models

Today we will consider various stochastic process models for modeling data evolving over time.

Last time we saw three basic models:

- White noise: independent random variables - not very interesting by itself, but an essential building block in more complicated/realistic models.
- Random walk: cumulatively adding random variables - moves around randomly, and resulting in non-stationary behaviour.
- Autoregressive process: AR(1) is a simple model for something evolving over time, and is stationary or not depending on choice of parameters. The main model class considered today is ARMA (autogressive moving average), which is a generalisation of AR(1).

First we take a recap of the three models above, and add some details.

---

14 White noise

A time series w_t, $t = 1, \dots, n$ is *white noise* if the variables $w_1, w_2, \dots, w_n$ are *independent* and *identically* distributed with a mean of zero.

From the definition it follows that white noise is a second order stationary process since the variance function $\sigma^2(t) = \sigma^2$ is the same constant for all t and the autocovariance is $Cov(w_t, w_{t+k}) = 0$ for all $k \neq 0$ which does not depend on t. We summarize this as:

$$\mu = 0$$

$$\gamma(k) = Cov(w_t, w_{t+k}) = \begin{cases} \sigma^2 \text{ for } k = 0 \\ 0 \text{ for } k \neq 0 \end{cases}$$

$$\rho(k) = \begin{cases} 1 \text{ for } k = 0 \\ 0 \text{ for } k \neq 0 \end{cases}$$

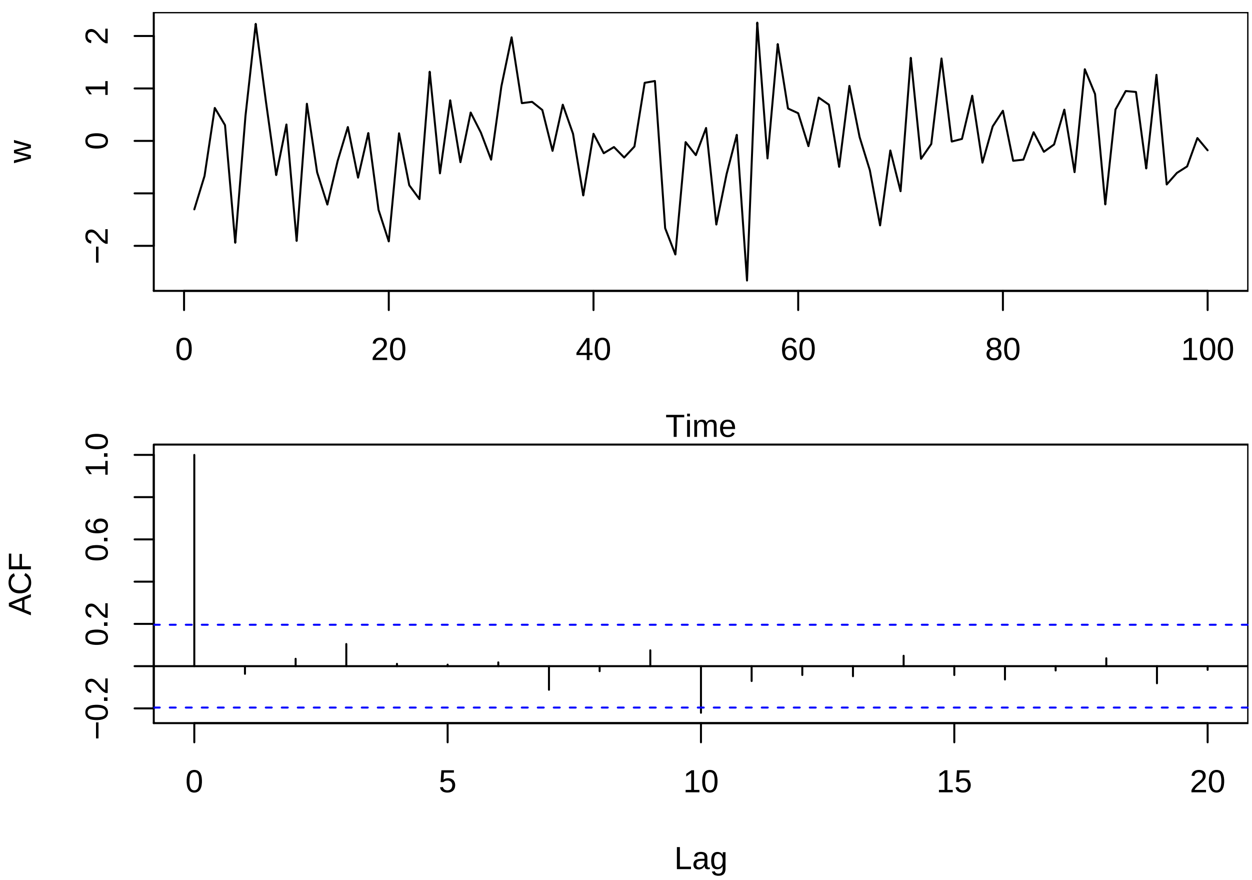
Often we will also assume the distribution of each w_t is Gaussian (i.e. $w_t \sim \texttt{norm}(0, \sigma)$) and then we call it *Gaussian white noise.*

---

15 Simulation of white noise

To understand how white noise behaves we can simulate it with R and plot both the series and the autocorrelation:

```
w <- rnorm(100, mean = 0, sd = 1)
par(mfrow = c(2,1), mar = c(4,4,0,0))
ts.plot(w)
acf(w)
```



It is a good idea to repeat this simulation and plot a few times to appreciate the variability of the results.

---

16 Random walk

A time series x_t is called a random walk if

$$x_t = x_{t-1} + w_t$$

where w_t is a white noise series. Using $x_{t-1} = x_{t-2} + w_{t-1}$ we get

$$x_t = x_{t-2} + w_{t-1} + w_t$$

Substituting for x_{t-2} we get

$$x_t = x_{t-3} + w_{t-2} + w_{t-1} + w_t$$

Continuing this way we would get an infinite sum of white noise

$$x_t = w_t + w_{t-1} + w_{t-2} + w_{t-3} + \dots$$

However, we will assume we have a fixed starting point $x_0 = 0$ such that

$$x_t = w_1 + w_2 + \cdots + w_t$$

---

17 Properties of random walk

A random walk x_t has a constant mean function

$$\mu(t) = 0$$

since the random walk at time t is a sum of t white noise terms that all have mean zero.

However, the variance function

$$\sigma^2(t) = t \cdot \sigma^2$$

clearly depends on the time t, so the process is not stationary. The variance function is derived from the general fact that for **independent random variables**, y_1 and y_2, the variance of the sum is

$$Var(y_1 + y_2) = Var(y_1) + Var(y_2).$$

Thus,

$$Var(x_t) = Var(w_1 + w_2 + \cdots + w_t) = \sigma^2 + \sigma^2 + \cdots + \sigma^2 = t\sigma^2$$

The non-stationary autocovariance function is

$$Cov(x_t, x_{t+k}) = t\sigma^2, \quad k = 0, 1, \dots$$

which only depends on how many white noise terms x_t and x_{t+k} have in common (t) and not how far they are separated (k).

By combining the two results we obtain the non-stationary autocorrelation function

$$Cor(x_t, x_{t+k}) = \frac{Cov(x_t, x_{t+k})}{\sqrt{Var(x_t)Var(x_{t+k})}} = \frac{t\sigma^2}{\sqrt{t\sigma^2(t+k)\sigma^2}} = \frac{1}{\sqrt{1+k/t}}$$

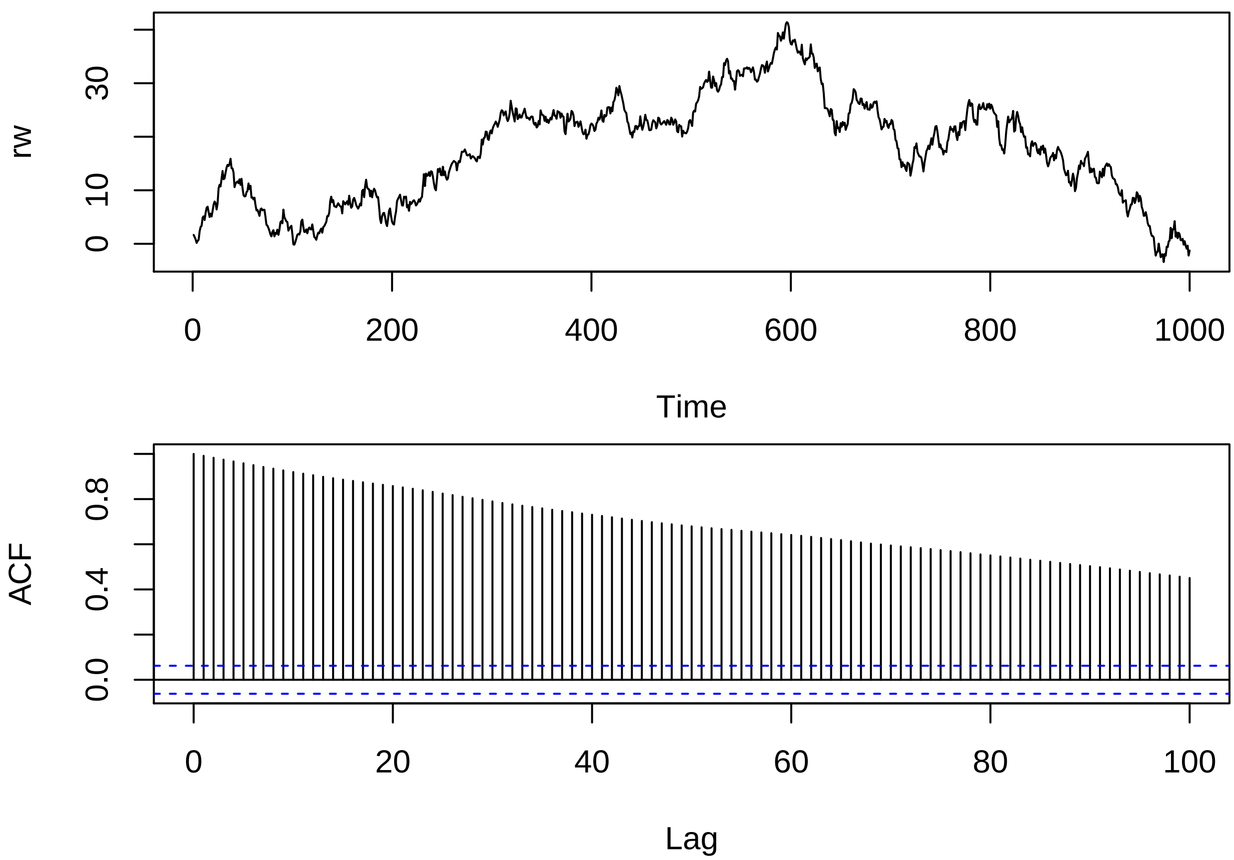
When t is large compared to k we have very high correlation (close to one) and even though the process is not stationary we expect the correlogram of a reasonably long random walk to show very slow decay.

---

18 Simulation of random walk

We already know how to simulate Gaussian white noise (with `rnorm`) and the random walk is just a cumulative sum of white noise:

```
w <- rnorm(1000)
rw <- cumsum(w)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(rw)
acf(rw, lag.max = 100)
```



---

19 Differencing

The slowly decaying acf for random walk is a classical sign of non-stationarity, indicating there may be some kind of trend. In this case there is no real trend, since the theoretical mean is constant zero, but we refer to the apparent trend which seems to change directions unpredictiably as a stochastic trend.

If a time series shows these signs of non-stationarity we can try to study the time series of differences and see if that is stationary and easier to understand:

$$\nabla x_t = x_t - x_{t-1}.$$

Since we assume/define $x_0 = 0$ we get

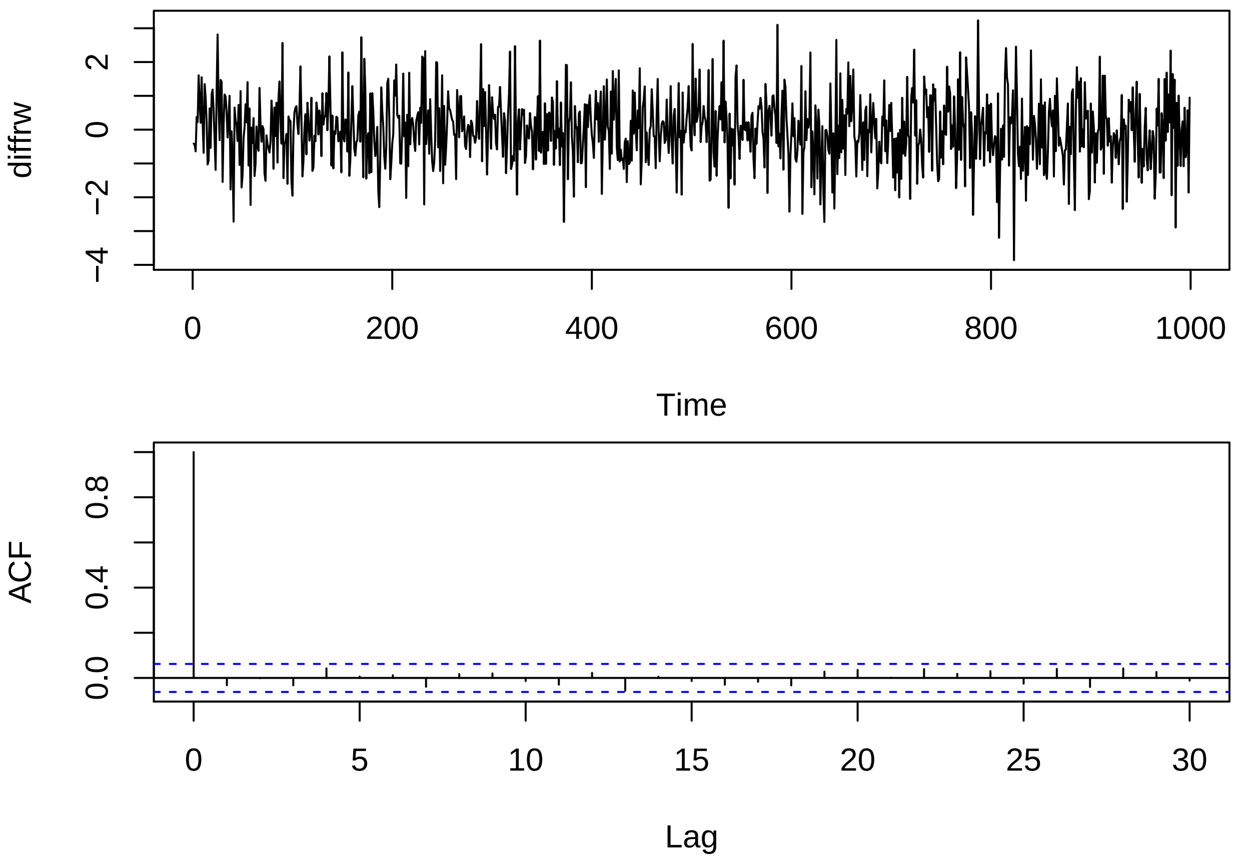
$$\nabla x_1 = x_1$$

$$\nabla x_2 = x_2 - x_1$$
$$\nabla x_3 = x_3 - x_2$$

etc.

Specifically when we difference a random walk $x_t = x_{t-1} + w_t$ we recover the white noise series $\nabla x_t = w_t$:

```
diffrw <- diff(rw)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(diffrw)
acf(diffrw, lag.max = 30)
```



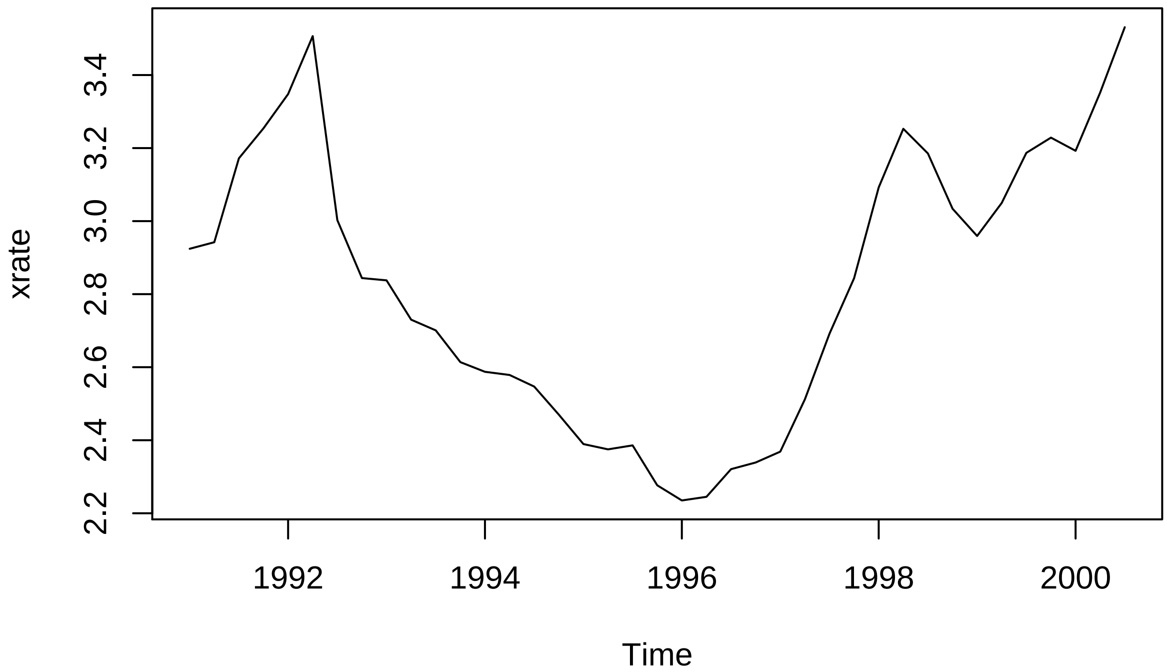
In general if a time series needs to be differenced to become stationary we say that the series is integrated (of order 1).

---

20 Example: Exchange rate

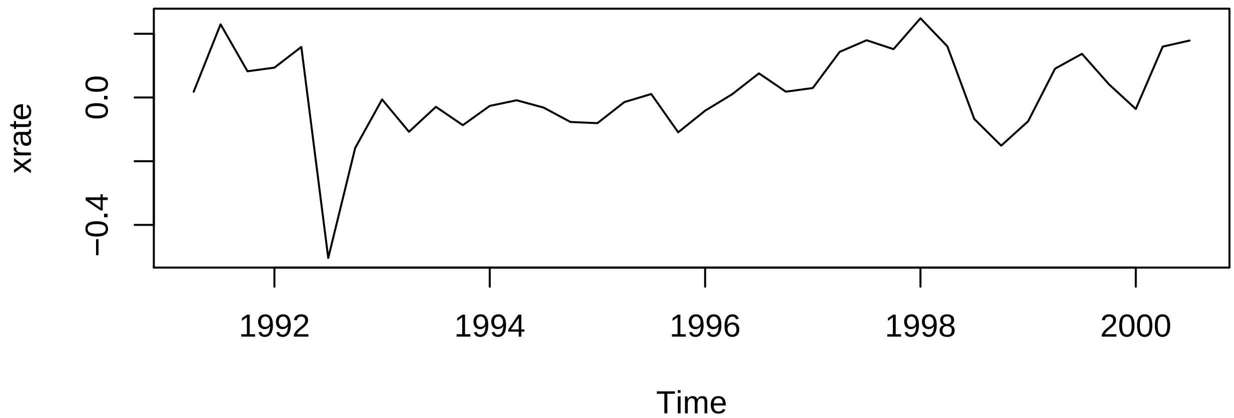
If we look at the exchange rate from GBP to NZD, we observe what looks like an unpredictable stochastic trend and we would like to see if it could reasonably be described as a random walk.

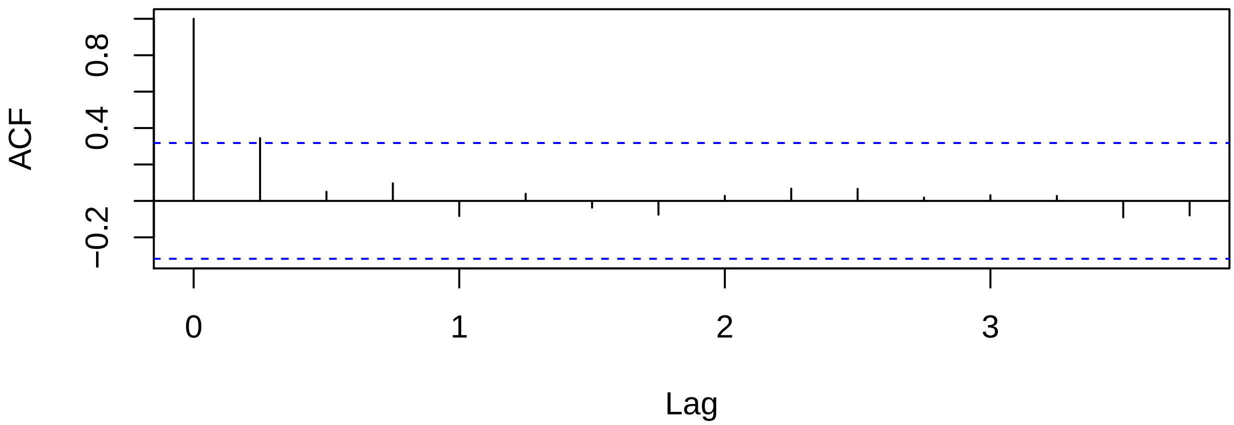
```
www <- "https://asta.math.aau.dk/eng/static/datasets?file=pounds_nz.dat"
exchange_data <- read.table(www, header = TRUE)
exchange <- ts(exchange_data, start = 1991, freq = 4)
plot(exchange)
```



To this end we difference the series and see if the difference looks like white noise:

```
diffexchange <- diff(exchange)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
plot(diffexchange)
acf(diffexchange)
```





The first order difference looks reasonably stationary, so the original exchange rate series could be considered integrated of order 1. However, there is an indication of significant autocorrelation at lag 1, so a random walk might not be a completely satisfactory model for this dataset.

21 Auto-regressive (AR) models

21.1 Auto-regressive model of order 1: AR(1)

A significant auto-correlation at lag 1 means that x_t and x_{t-1} are correlated so the previous value x_{t-1} can be used to predict the current value x_t. This is the idea behind an auto-regressive model of order one AR(1):

$$x_t = \alpha_1 x_{t-1} + w_t$$

where w_t is white noise and the auto-regressive coefficient α_1 is a parameter to be estimated from data.

The model is only stationary if $-1 < \alpha_1 < 1$ such that the dependence of the past decreases with time.

21.1.1 Properties of AR(1) models

For a stationary AR(1) model with $-1 < \alpha_1 < 1$ we have previously shown that

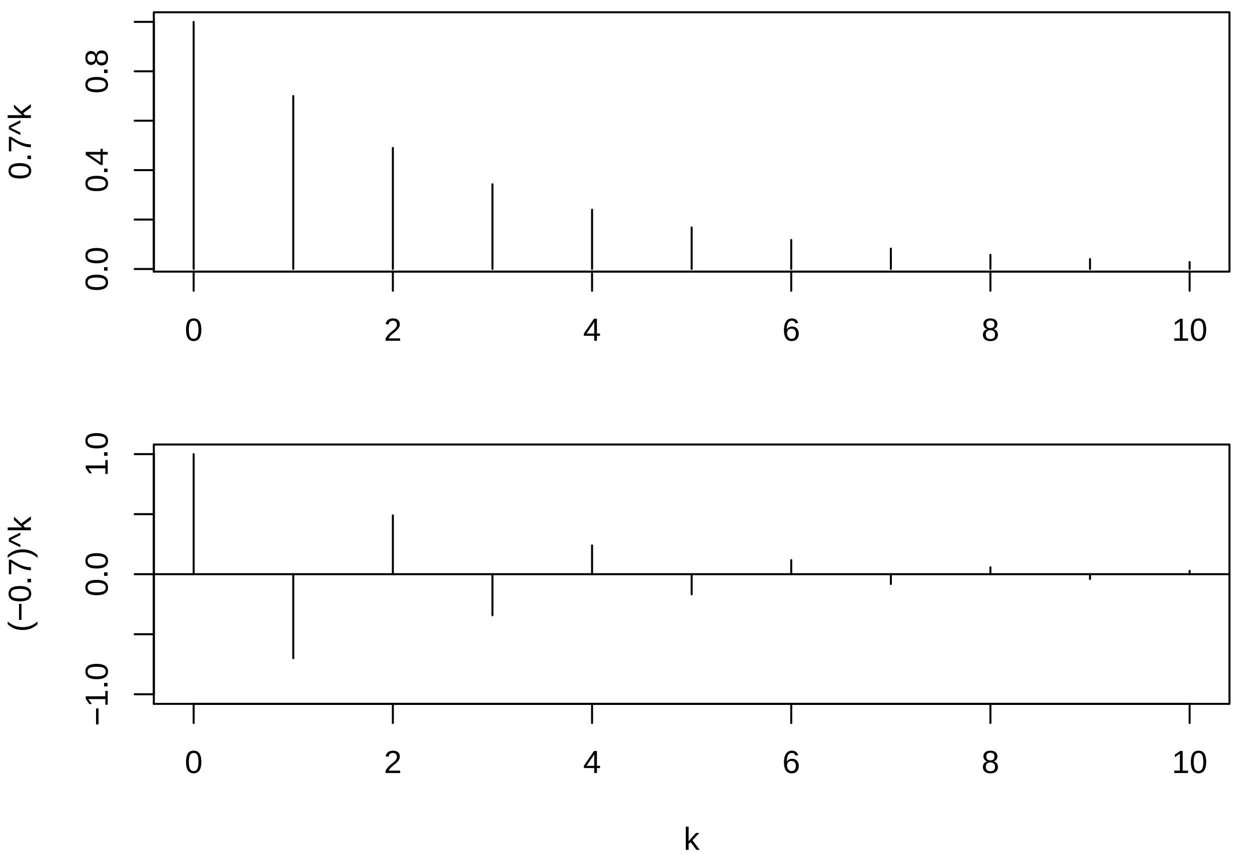
- $\mu(t) = 0$
- $Var(x_t) = \sigma^2(t) = \sigma^2/(1-\alpha_1^2)$

- $\gamma(k) = \alpha_1^k \sigma^2/(1-\alpha_1^2)$
- $\rho(k) = \alpha_1^k$

Below are the theoretical autocorrelation functions for the following AR(1) models:

- Model 1: $x_t = 0.7x_{t-1} + w_t$
- Model 2: $x_t = -0.7x_{t-1} + w_t$

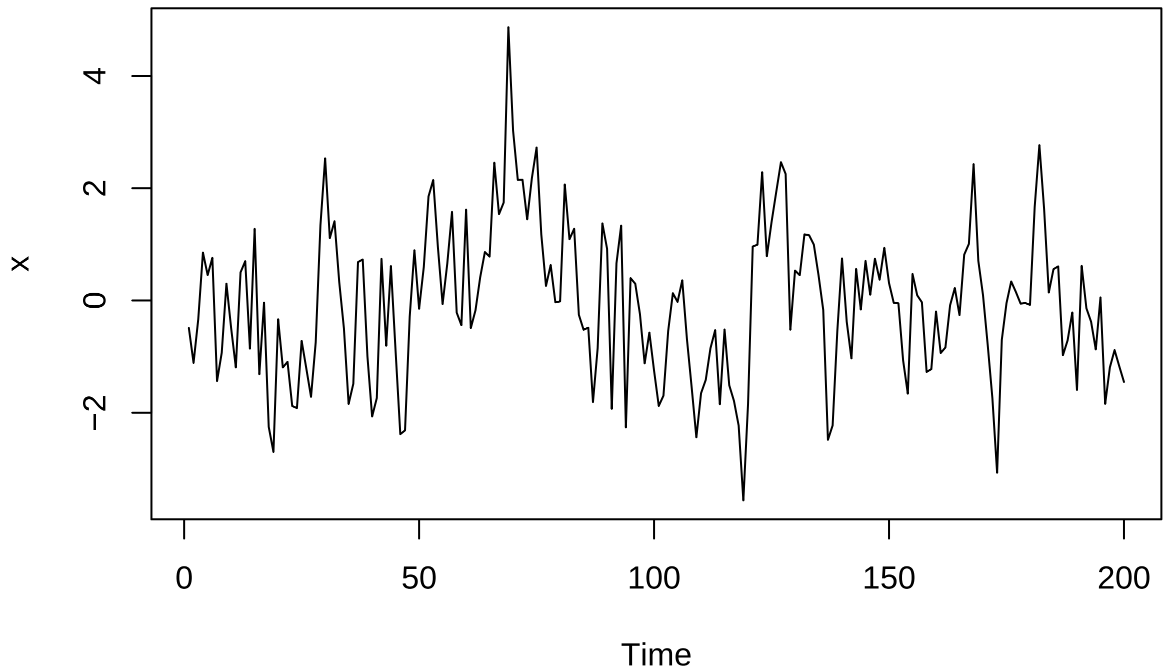
```
k <- 0:10
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
plot(k, 0.7^k, type = "h", xlab = "")
plot(k, (-0.7)^k, type = "h", ylim = c(-1,1))
abline(h=0)
```



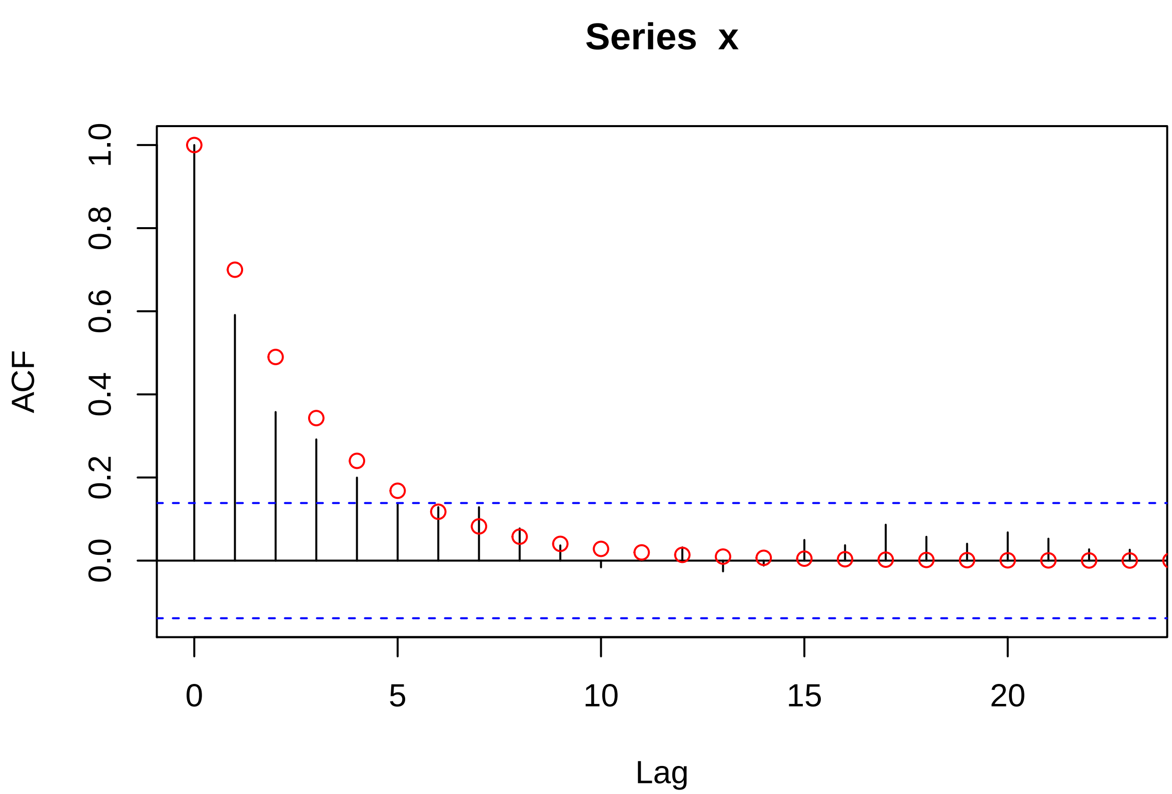
21.1.2 Simulation of AR(1) models

R has a built-in function `arima.sim` to simulate AR(1) and other more complicated models called ARMA and ARIMA. It needs the model (i.e. the autoregressive coefficient α_1) and the desired number of time steps `n`. To simulate 200 time steps of AR(1) with $\alpha_1 = 0.7$ we do:

```
x <- arima.sim(model = list(ar = 0.7), n = 200)
plot(x)
```




```r
acf(x)
k = 0:30
points(k, 0.7^k, col = "red")
```



Here we have compared the empirical correlogram with the theoretical values of the model.

21.1.3 Fitted AR(1) models

To estimate the parameters in an AR(1) process, we use the so-called Yule-Walker equations. This essentially means that we pick the parameter α_1 such that the theoretical autocorrelation function fits the correlogram of the data as close as possible. We skip the details of this, and simple use the function `ar`:

```
fit <- ar(x, order.max = 1)
```

The resulting object contains the value of the estimated parameter and a bunch of other information. In this case the input data are artificial so we know we should ideally get a value close to 0.7:

```
fit$ar
```

```
## [1] 0.5911432
```

An estimate of the variance of the estimate $\hat{\alpha}_1$ is given in `fit$asy.var.coef` (the estimated std. error is the square root of this):

```
fit$asy.var.coef
```

```
##             [,1]
## [1,] 0.003285605
```

```
se <- sqrt(fit$asy.var.coef)
se
```

```
##           [,1]
## [1,] 0.0573202
```

```
ci <- c(fit$ar - 2*se, fit$ar + 2*se)
ci
```

```
## [1] 0.4765028 0.7057836
```

The AR(1) model defined earlier has mean 0. However, we cannot expect data to fulfill this. This is fixed by subtracting the average $\bar{x}$ of the data before doing anything else, so the model that is fitted is actually:

$$x_t - \bar{x} = \alpha_1 \cdot (x_{t-1} - \bar{x}) + w_t$$

To predict the value of x_t given x_{t-1} we use that w_t is white noise so we expect it to be zero on average:

$$\hat{x}_t - \bar{x} = \hat{\alpha}_1 \cdot (x_{t-1} - \bar{x})$$

So the predictions are given by

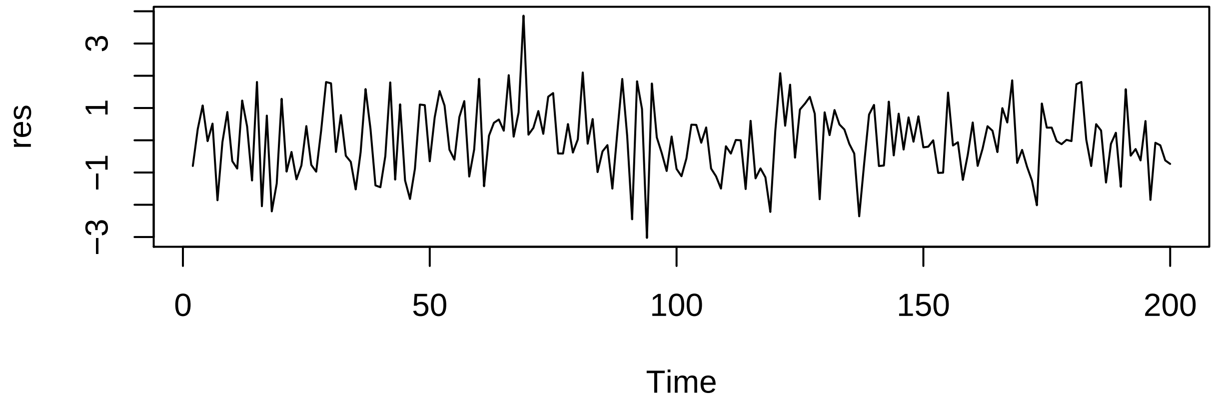
$$\hat{x}_t = \bar{x} + \hat{\alpha}_1 \cdot (x_{t-1} - \bar{x}), \quad t \geq 2.$$

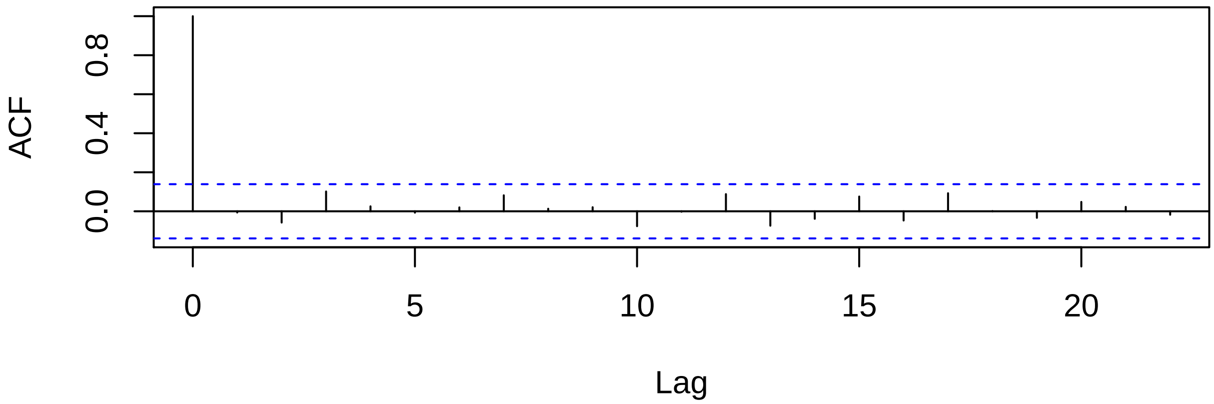
Given the predictions we can estimate the model errors as usual by the model residuals:

$$\hat{w}_t = x_t - \hat{x}_t, \quad t \geq 2.$$

If we believe the model describes the dataset well the residuals should look like a sample of white noise:

```r
res <- na.omit(fit$resid)
par(mfrow = c(2,1), mar = c(4,4,1,1))
plot(res)
acf(res)
```





This naturally looks good for this artificial dataset.

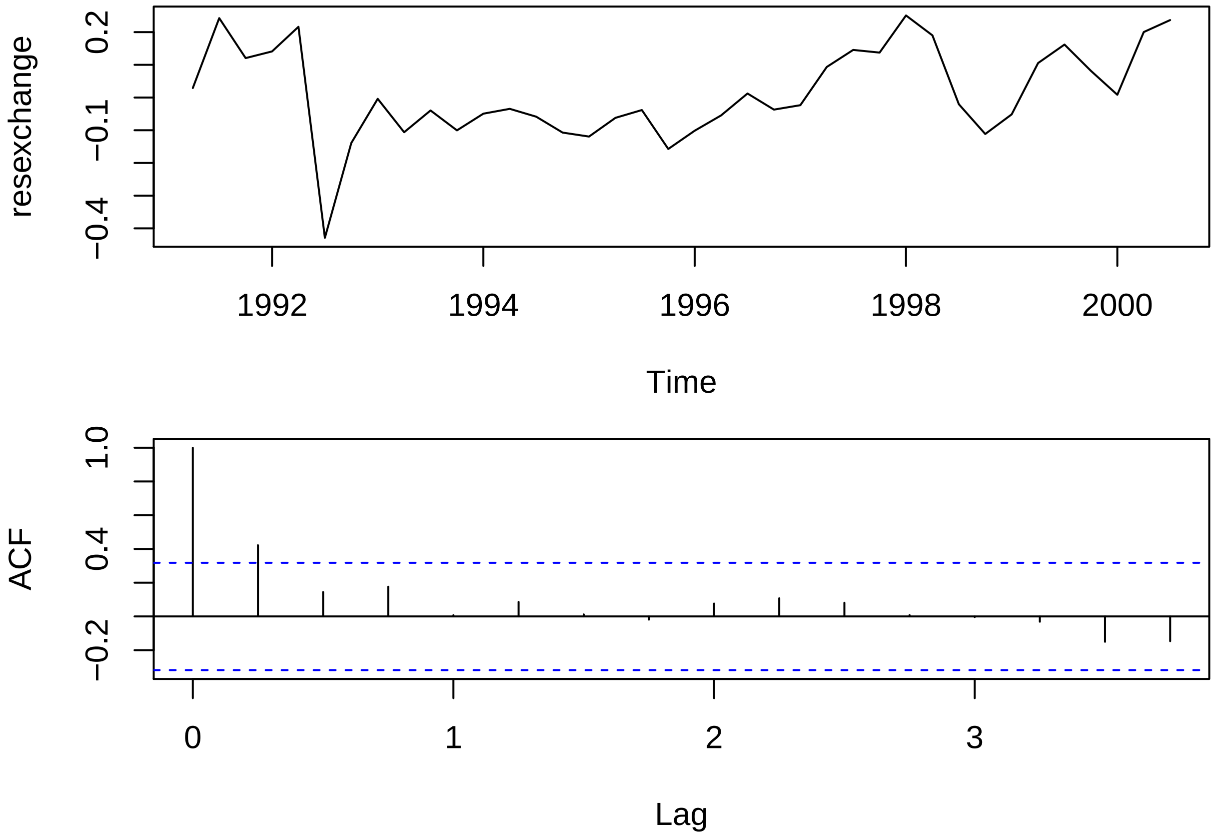
21.1.4 AR(1) model fitted to exchange rate

A random walk is an example of a AR(1) model with $\alpha_1 = 1$, and it is non-stationary. This didn't provide an ideal fit for the exchange rate dataset, so we might suggest a stationary AR(1) model with α_1 as a parameter to be estimated from data:

```r
fitexchange <- ar(exchange, order.max = 1)
fitexchange$ar
```

```
## [1] 0.890261
```

```r
resexchange <- na.omit(fitexchange$resid)
par(mfrow = c(2,1), mar = c(4,4,1,1))
plot(resexchange)
acf(resexchange)
```



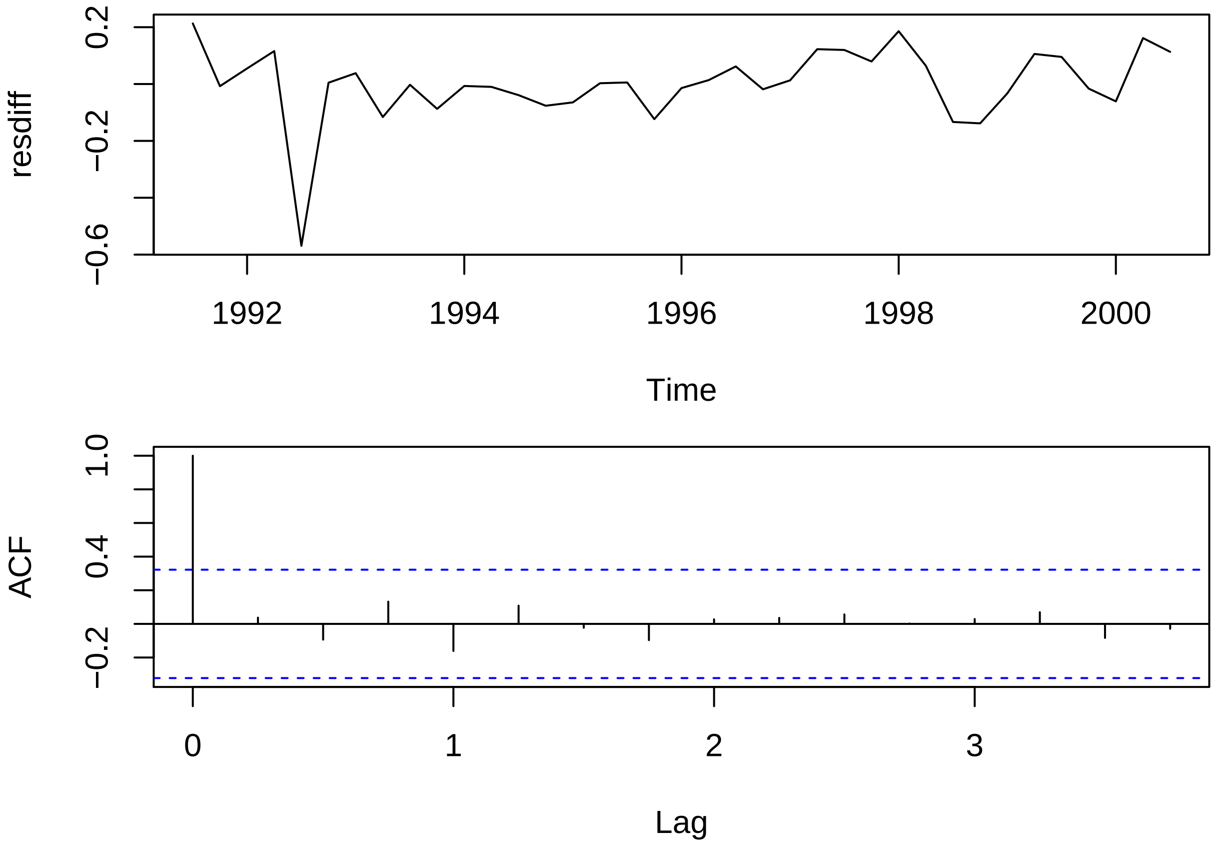
This does not appear to really provide a better fit than the random walk model proposed earlier.

An alternative would be to propose a AR(1) model for the differenced time series $\nabla x_t = x_t - x_{t-1}$:

```
dexchange <- diff(exchange)
fitdiff <- ar(dexchange, order.max = 1)
fitdiff$ar
```

```
## [1] 0.3451507
```

```
resdiff <- na.omit(fitdiff$resid)
par(mfrow = c(2,1), mar = c(4,4,1,1))
plot(resdiff)
acf(resdiff)
```



21.1.5 Prediction from AR(1) model

We can use a fitted AR(1) model to predict future values of a time series. If the last observed time point is t then we predict x_{t+1} using the equation given previously:

$$\hat{x}_{t+1} = \bar{x} + \hat{\alpha}_1 \cdot (x_t - \bar{x}).$$

If we want to predict x_{t+2} we use

$$\hat{x}_{t+2} = \bar{x} + \hat{\alpha}_1 \cdot (\hat{x}_{t+1} - \bar{x}).$$

And we can continue this way. Prediction is performed by `predict` in R. E.g. for the AR(1) model fitted to the exchange rate data the last observation is in third quarter of 2000. If we want to predict 1 year ahead to third quarter of 2001 (probably a bad idea due to the stochastic trend):

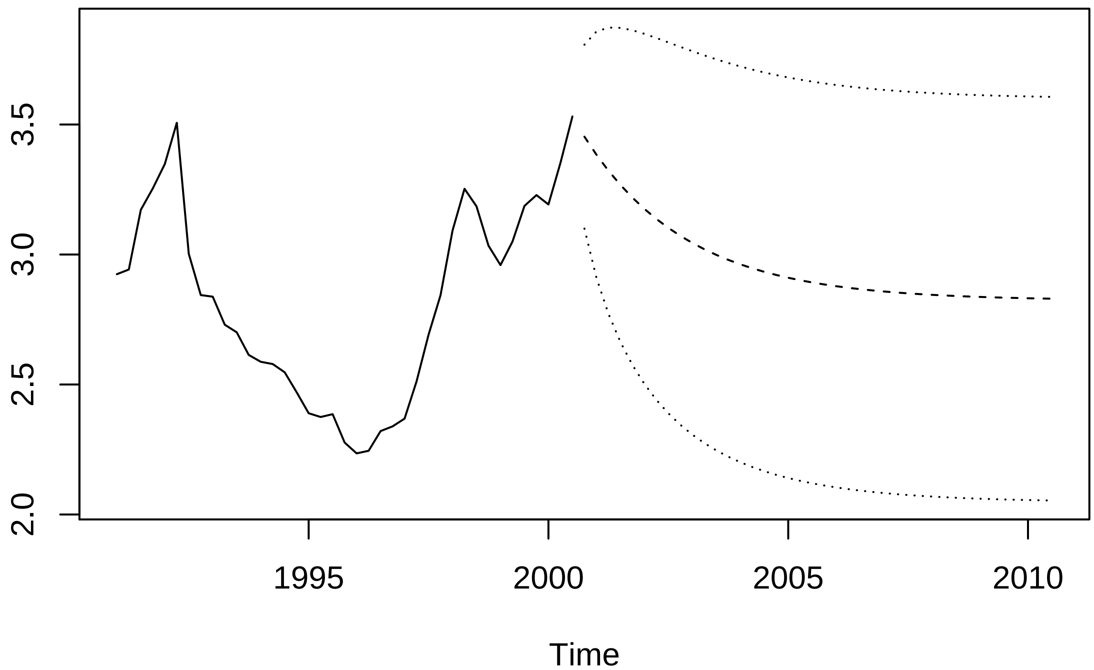
```
pred1 <- predict(fitexchange, n.ahead = 4)
pred1
```

```
## $pred
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2000                            3.453332
## 2001 3.384188 3.322631 3.267830
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2000                               0.1767767
## 2001 0.2366805 0.2750411 0.3020027
```

Note how the prediction returns both the predicted value and a standard error for this value. So we predict that the exchange rate in third quarter of 2001 would be within 3.27 ± 0.6 with approximately 95% probability.

We can plot a prediction and approximate 95% pointwise prediction intervals with `ts.plot` (where we use a 10 year prediction – which is a very bad idea – to see how it behaves in the long run):

```
pred10 <- predict(fitexchange, n.ahead = 40)
lower10 <- pred10$pred-2*pred10$se
upper10 <- pred10$pred+2*pred10$se
ts.plot(exchange, pred10$pred, lower10, upper10, lty = c(1,2,3,3))
```



21.2 Auto-regressive models of higher order

The first order auto-regressive model can be generalised to higher order by adding more lagged terms to explain the current value x_t. An AR(p) process is

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + w_t$$

where w_t is white noise and $\alpha_1, \alpha_2, \ldots, \alpha_p$ are parameters to be estimated from data.

The notation can be a little cumbersome when we have many terms, so we introduce the backshift operator B, that takes the time series one step back, i.e.

$$Bx_t = x_{t-1}$$

This can be used repeatedly, for example $B^2 x_t = BBx_t = Bx_{t-1} = x_{t-2}$. We can then make a polynomial of backshift operators

$$\alpha(B) = 1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p$$

and write the AR(p) process as

$$\alpha(B)x_t = w_t.$$

The parameters cannot be chosen arbitrarily if we want the model to be stationary. To check that a given AR(p) model is stationary we must find all the roots of the characteristic equation

$$1 - \alpha_1 z - \alpha_2 z^2 - \cdots - \alpha_p z^p = 0$$

and check that the absolute value of each root is greater than 1. This can be written with the polynomial from above, but with z inserted instead of B, i.e.

$$\alpha(z) = 0.$$

Solving a p-order polynomial is hard for high values of p, so we will let R do this for us.

21.2.1 Estimation of AR(p) models

For an AR(p) model there are typically two things we need to estimate:

1. The maximal non-zero lag p in the model.
2. The autoregressive coefficients/parameters $\alpha_1, \ldots, \alpha_p$.

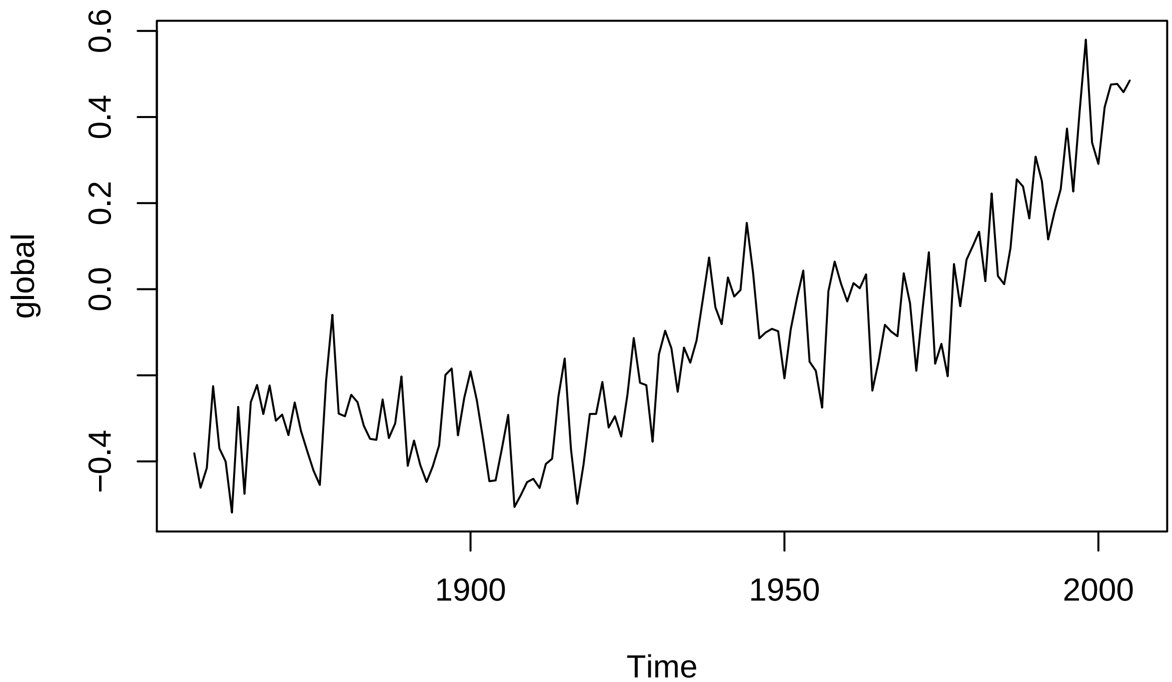
For the first point we can use AIC (Akaike's Information Criterion). This is essentially a balance between simplicity and good fit of a model. The AIC results in a single real number, where smaller is better. The ar function in R uses AIC to automatically select the value for p by calculating the AIC for all models with p between 1 and some chosen maximal value, and picking the one with the smallest AIC.

Once the order is chosen and the estimates $\hat{\alpha}_1, \ldots, \hat{\alpha}_p$ are found the corresponding standard errors can be found as the square root of the diagonal of the matrix stored as `asy.var.coef` in the fitted model object.

21.2.2 Example of AR(p) model

We use an example of monthly global temperatures expressed as anomalies from the monthly average in 1961-1990. We reduce the dataset to the yearly mean temperature, and fit an AR(p) model to this. A good fit would indicate that the higher temperatures over the last decade could be explained by a purely stochastic process which just has dependence on the temperature anomalies from previous year and eventually might as well start decreasing again. (However, this does not mean that there is no climate crisis! There is lots of scientific evidence of this based on much more complicated models and more detailed data.)

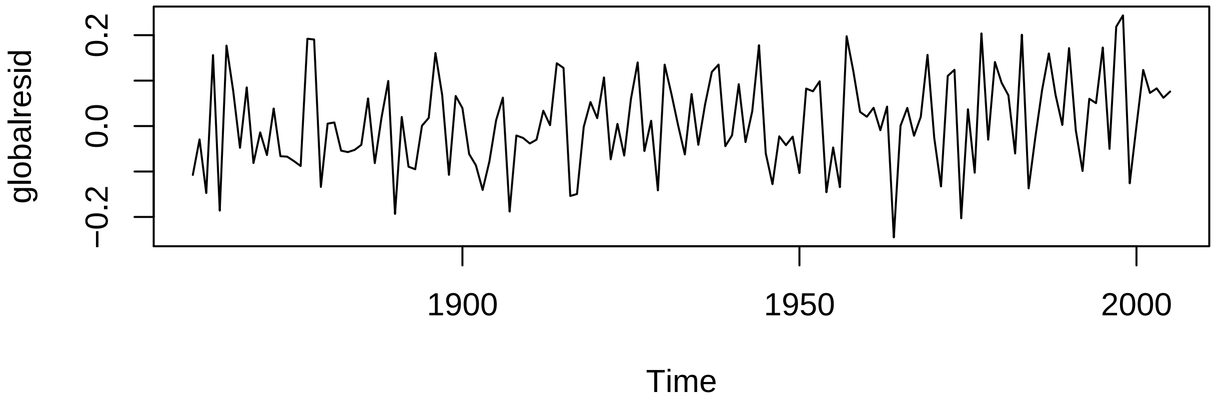
```
global_data <- scan("https://asta.math.aau.dk/eng/static/datasets?file=global.dat")
global_monthly <- ts(global_data, st = c(1856,1), end = c(2005,12), freq = 12)
global <- aggregate(global_monthly, FUN = mean)
plot(global)
```

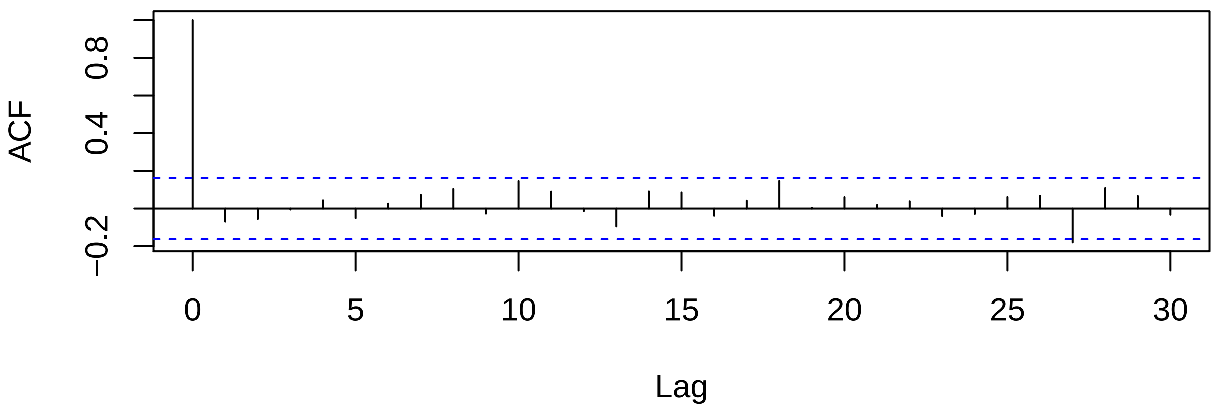



```
globalfit <- ar(global, order.max = 10)
globalfit
```

```
##
## Call:
## ar(x = global, order.max = 10)
##
## Coefficients:
##      1       2       3       4
## 0.6825  0.0032  0.0672  0.1730
##
## Order selected 4  sigma^2 estimated as  0.01371
```

```
globalresid <- na.omit(globalfit$resid)
par(mfrow = c(2,1), mar = c(4,4,1,1))
plot(globalresid)
acf(globalresid, lag.max = 30)
```





We are not assured that the estimated model will be stationary, but we can solve $\alpha(z) = 0$ and check if the solutions all have absolute values larger than 1, in which case the estimated model is stationary:

```
abs(polyroot(c(1,-globalfit$ar)))
```

```
## [1] 1.045256 2.029234 1.650963 1.650963
```

22 Moving average models

Another class of models are moving average (MA) models. An moving average process of order q, MA(q), is defined by

$$x_t = w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \cdots + \beta_q w_{t-q}$$

where w_t is a white noise process with mean zero and variance σ_w^2 and $\beta_1, \beta_2, \ldots, \beta_q$ are parameters to be estimated.

The moving average process also has a short notation using the backshift operator, given by

$$x_t = \beta(B) w_t$$

where the polynomial $\beta(B)$ is given by

$$\beta(B) = 1 + \beta_1 B + \beta_2 B^2 + \cdots + \beta_q B^q.$$

Since a moving average process is a finite sum of stationary white noise terms it is itself stationary and therefore the mean and variance is time-invariant (same constant mean and variance for all t):

- Mean $\mu(t) = 0$
- Variance $\sigma^2(t) = \sigma_w^2(1 + \beta_1^2 + \beta_2^2 + \cdots + \beta_q^2)$

The autocorrelation function, for $k \geq 0$, is

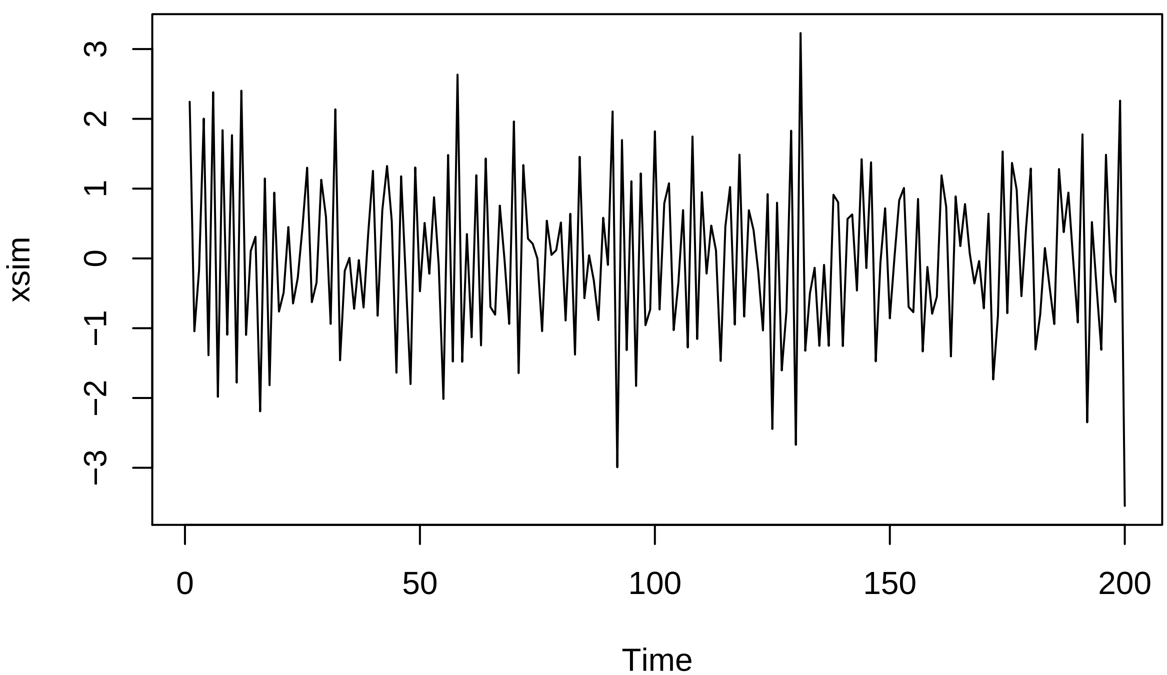
$$\rho(k) = \begin{cases} 1 & k = 0 \\ \sum_{i=0}^{q-k} \beta_i \beta_{i+k} / \sum_{i=0}^{q} \beta_i^2 & k = 1, 2, \ldots, q \\ 0 & k > q \end{cases}$$

where $\beta_0 = 1$.

22.1 Simulation of MA(q) processes

To simulate a MA(q) process we just need the white noise process w_t and then transform it using the MA coefficients. If we e.g. want to simulate a model with $\beta_1 = -0.7$, $\beta_2 = 0.5$, and $\beta_3 = -0.2$ we can use `arima.sim`:

```
xsim <- arima.sim(list(ma = c(-.7, .5, -.2)), n = 200)
plot(xsim)
```



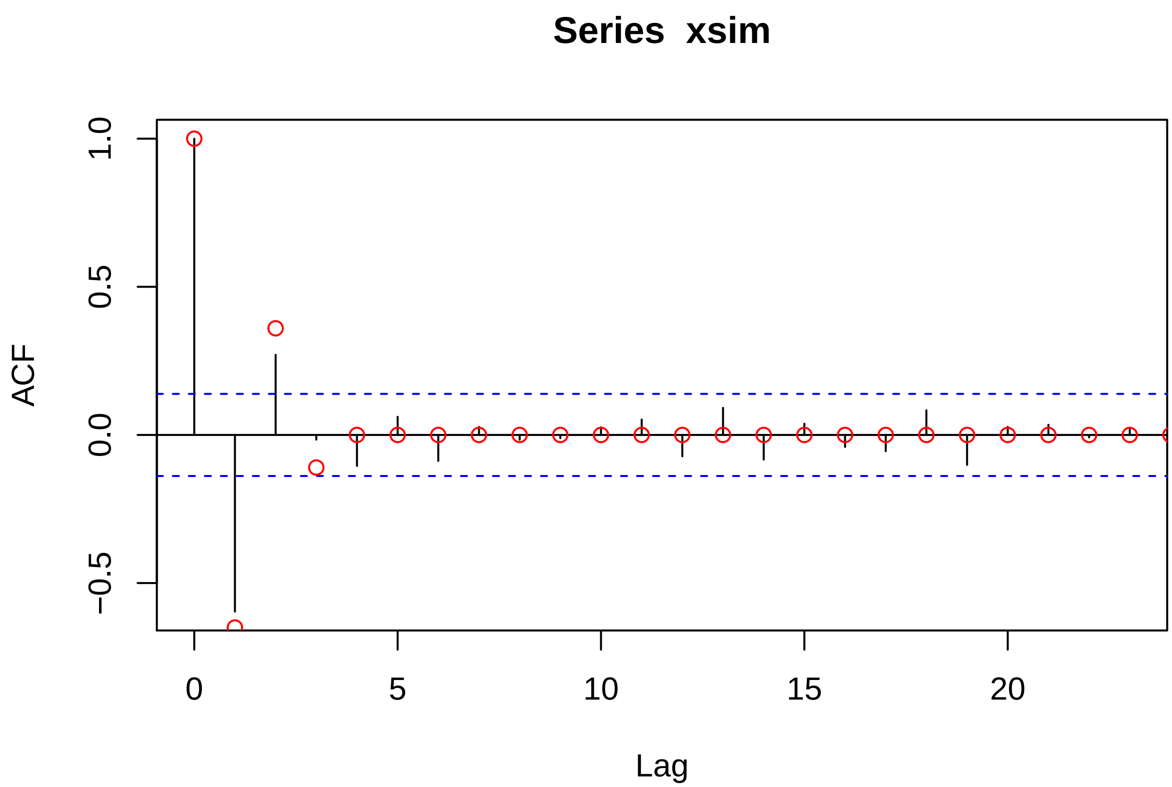
The theoretical autocorrelations are in this case:

$$\rho(1) = \frac{1 \cdot (-0.7) + (-0.7) \cdot 0.5 + 0.5 \cdot (-0.2)}{1 + (-0.7)^2 + 0.5^2 + (-0.2)^2} = -0.65$$

$$\rho(2) = \frac{1 \cdot 0.5 + (-0.7) \cdot (-0.2)}{1 + (-0.7)^2 + 0.5^2 + (-0.2)^2} = 0.36$$

$$\rho(3) = \frac{1 \cdot (-0.2)}{1 + (-0.7)^2 + 0.5^2 + (-0.2)^2} = -0.11$$

```r
acf(xsim)
points(0:25, c(1,-.65, .36, -.11, rep(0,22)), col = "red")
```



22.1.1 Estimation of MA(q) models

To estimate the parameters of a MA(q) model we use `arima`:

```r
xfit <- arima(xsim, order = c(0,0,3))
xfit
```

```
##
## Call:
## arima(x = xsim, order = c(0, 0, 3))
##
## Coefficients:
##           ma1     ma2      ma3  intercept
##       -0.7491  0.5072  -0.2626    -0.0037
## s.e.   0.0765  0.0846   0.0849     0.0320
##
## sigma^2 estimated as 0.8254:  log likelihood = -264.96,  aic = 539.93
```

The function `arima` does not include automatic selection of the order of the model so this has to be chosen beforehand or selected by comparing several proposed models and choosen the model with the minimal AIC.

23 Mixed models: Auto-regressive moving average models

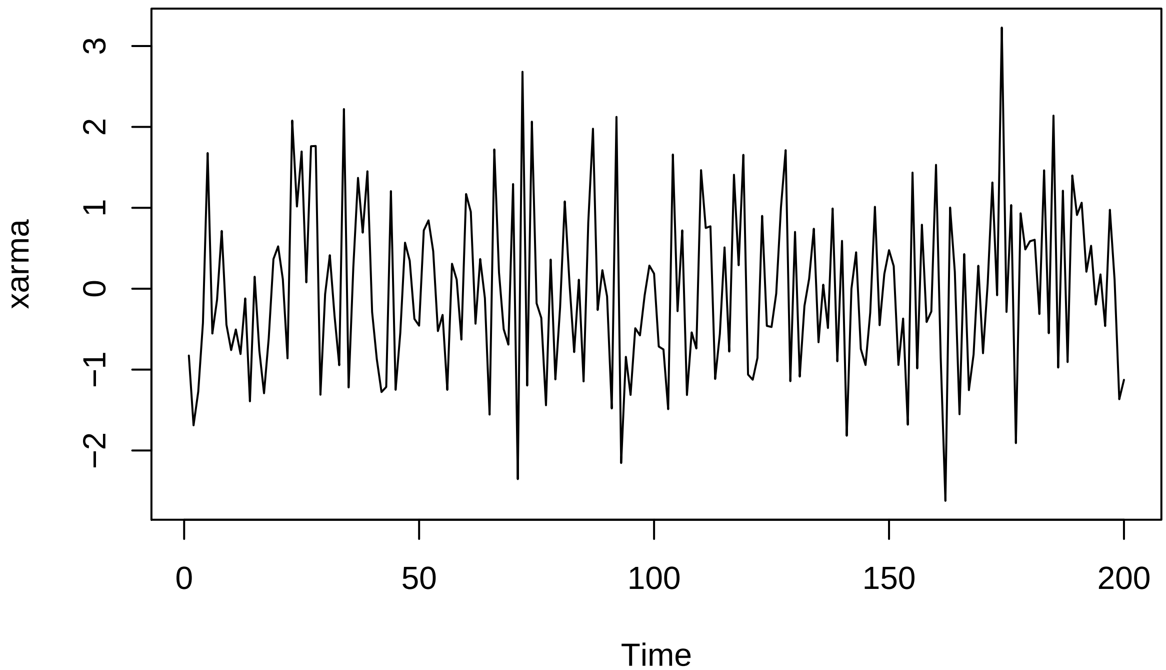
A time series x_t follows an auto-regressive moving average (ARMA) process of order (p, q), denoted $ARMA(p, q)$, if

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \cdots + \beta_q w_{t-q}$$

where w_t is a white noise process and $\alpha_1, \alpha_2, \ldots, \alpha_p, \beta_1, \beta_2, \ldots, \beta_q$ are parameters to be estimated.

We can simulate an ARMA model with `arima.sim`. E.g. an ARMA(1,1) model:

```
xarma <- arima.sim(model = list(ar = -0.6, ma = 0.5), n = 200)
plot(xarma)
```



Estimation is done with `arima` as before.

23.0.1 Example with exchange rate data

For the exchange rate data we may e.g. suggest either a AR(1), MA(1) or ARMA(1,1) model. We can compare fitted model using `AIC` (smaller is better):

```
exchange_ar <- arima(exchange, order = c(1,0,0))
AIC(exchange_ar)
```

```
## [1] -37.40417
```

```
exchange_ma <- arima(exchange, order = c(0,0,1))
AIC(exchange_ma)
```

```
## [1] -3.526895
```

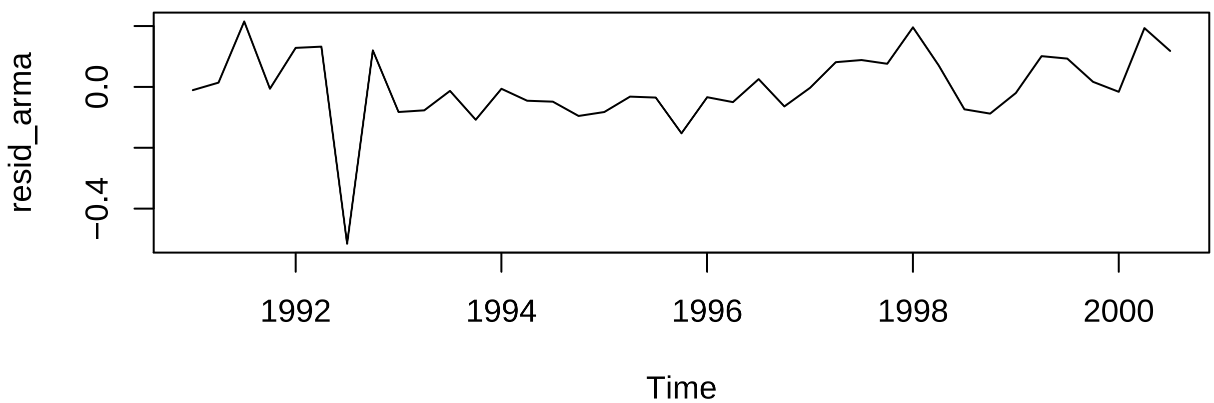
```r
exchange_arma <- arima(exchange, order = c(1,0,1))
AIC(exchange_arma)
```

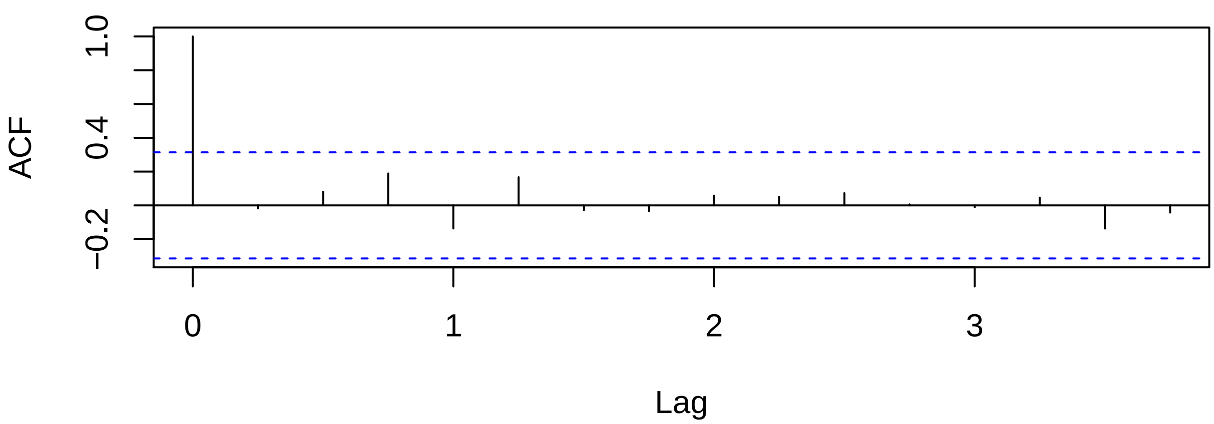
```
## [1] -42.27357
```

```r
exchange_arma
```

```
##
## Call:
## arima(x = exchange, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.8925  0.5319     2.9597
## s.e.  0.0759  0.2021     0.2435
##
## sigma^2 estimated as 0.01505:  log likelihood = 25.14,  aic = -42.27
```

```r
par(mfrow = c(2,1), mar = c(4,4,1,1))
resid_arma <- na.omit(exchange_arma$residuals)
plot(resid_arma)
acf(resid_arma)
```





24 Models with expogenous variables

24.1 Exogenous variables

- The ARMA processes are flexible models for x_t for $t = 1 \ldots, n$ evolving randomly over time, but it does not include the possibility that anything is influencing x_t
- An exogeneous variable is another variable, say y_t, that influences the behaviour of x_t
- Here y_t may be another stochastic process, which we do not model, but only consider as given, or it might be something we can control.

24.2 Regression models with exogenous variables

- We can combine regression models with ARMA models to obtain a stochastic process which is influenced by exogenous variables.
- Consider a linear regression, but where the noise term is an ARMA process:

$$y_t = \gamma x_t + \epsilon_t, \qquad \alpha(B)\epsilon_t = \beta(B)w_t$$

- If we isolate y_t and insert into the ARMA expression, we get something that looks more like an ARMA process but with y_t adjusted by the exogenous variable:

$$\alpha(B)(y_t - \gamma x_t) = \beta(B)w_t$$

- Modelling a dataset with this model will allow us to simultaneously include influence by earlier times of the process itself, but also from external sources.
- The purpose of fitting such a model is both to obtain a good model for the evolution of the data and to obtain an understanding of the relation between y_t and x_t.
- Note that here γ is a single number, and x_t is a single stochastic process, but we can also include multiple stochastic processes, and let γ be a vector instead.

24.3 Example

- As an example consider a simple linear regression combined with an AR(1) process for noise terms:

$$y_t = \gamma x_t + \epsilon_t, \qquad \epsilon_t = \alpha_1 \epsilon_{t-1} + w_t$$

  or

$$y_t = \alpha_1 y_{t-1} + \gamma(x_t - \alpha_1 x_{t-1}) + w_t$$

- Notice that the model behaves like an AR(1) process, but instead of having a constant mean of 0, its mean is constantly adjusted by the exogenous variable.