

Stochastic processes II

The ASTA team

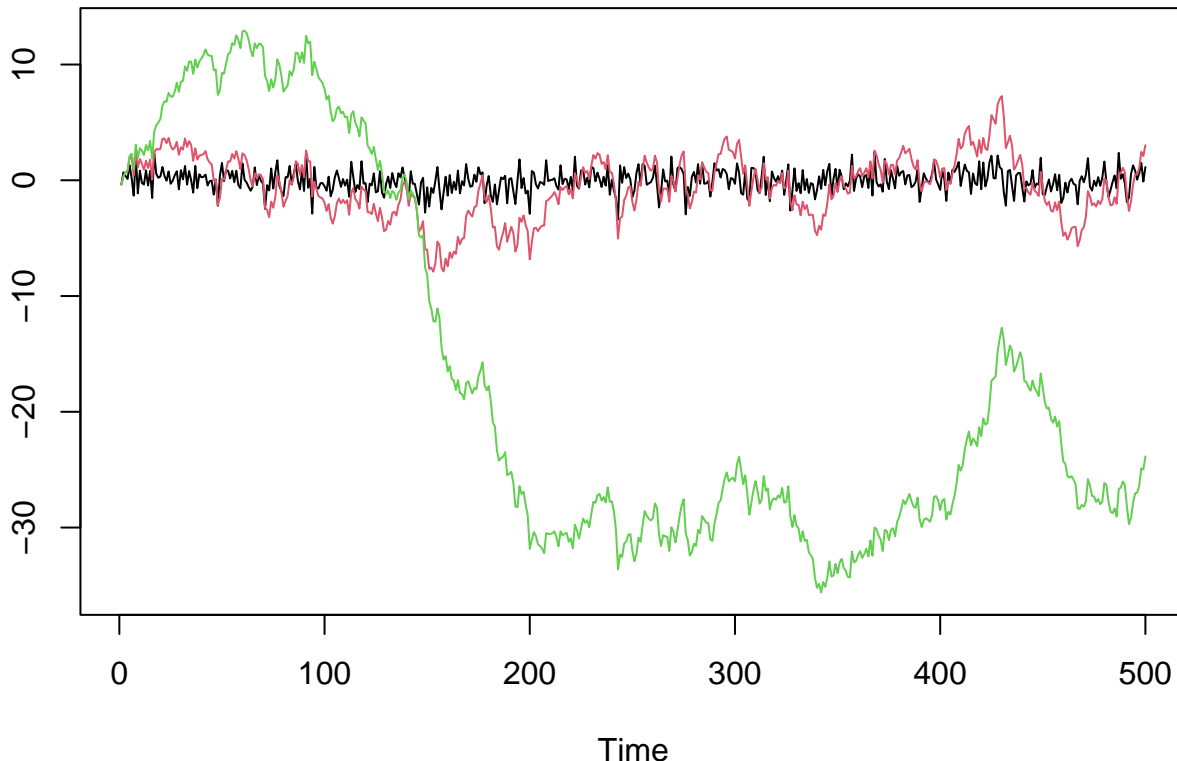
Contents

1	Basic stochastic process models	1
1.1	Stochastic processes	2
2	White noise	2
2.1	White noise	2
2.2	Correlogram for white noise	3
3	Random walk	4
3.1	Random walk	4
3.2	Properties of random walk	4
3.3	Simulation and correlogram of random walk	4
3.4	Differencing	5
3.5	Example: Exchange rate	6
4	First order auto-regressive models AR(1)	8
4.1	Auto-regressive model of order 1: AR(1)	8
4.2	Properties of AR(1) models	8
4.3	Simulation of AR(1) models	8
4.4	Fitted AR(1) models	10
4.5	Residuals for AR(1) models	10
4.6	AR(1) model fitted to exchange rate	11
4.7	Prediction/forecasting from AR(1) model	13
5	Auto-regressive models of higher order (AR(p)-models)	14
5.1	Auto-regressive models of higher order	14
5.2	Stationarity for AR(p) models	15
5.3	Estimation of AR(p) models	15
5.4	Example of AR(p) model for electricity data	15
6	Moving average models (MA(q)-models)	17
6.1	The moving average model	17
6.2	Simulation of MA(q) processes	18
6.3	Estimation of MA(q) models	19
7	Auto-regressive moving average models (ARMA)	20
7.1	Mixed models: Auto-regressive moving average models	20
7.2	Example with exchange rate data	20

1 Basic stochastic process models

1.1 Stochastic processes

- Last time we introduced stochastic processes



- 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 white noise - non-stationary, tendency to wander off.
 - **Autoregressive process AR(1):** weaker dependence on previous value than random walk, is stationary or not depending on choice of parameters.
- Today we will consider more stochastic process models.
- First we take a recap of the three models above, and add some details.

2 White noise

2.1 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
 - identically distributed
 - have mean zero and variance σ^2
- From the definition it follows that white noise is a second order stationary process since
 - The mean is constant ($= 0$)
 - The variance function $\sigma^2(t) = \sigma^2$ is constant
 - The autocorrelation $Cor(W_t, W_{t+k}) = 0$ for all $k \neq 0$, which does not depend on t .

- Hence, we have a well-defined mean,

$$\mu = 0,$$

autocovariance function

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

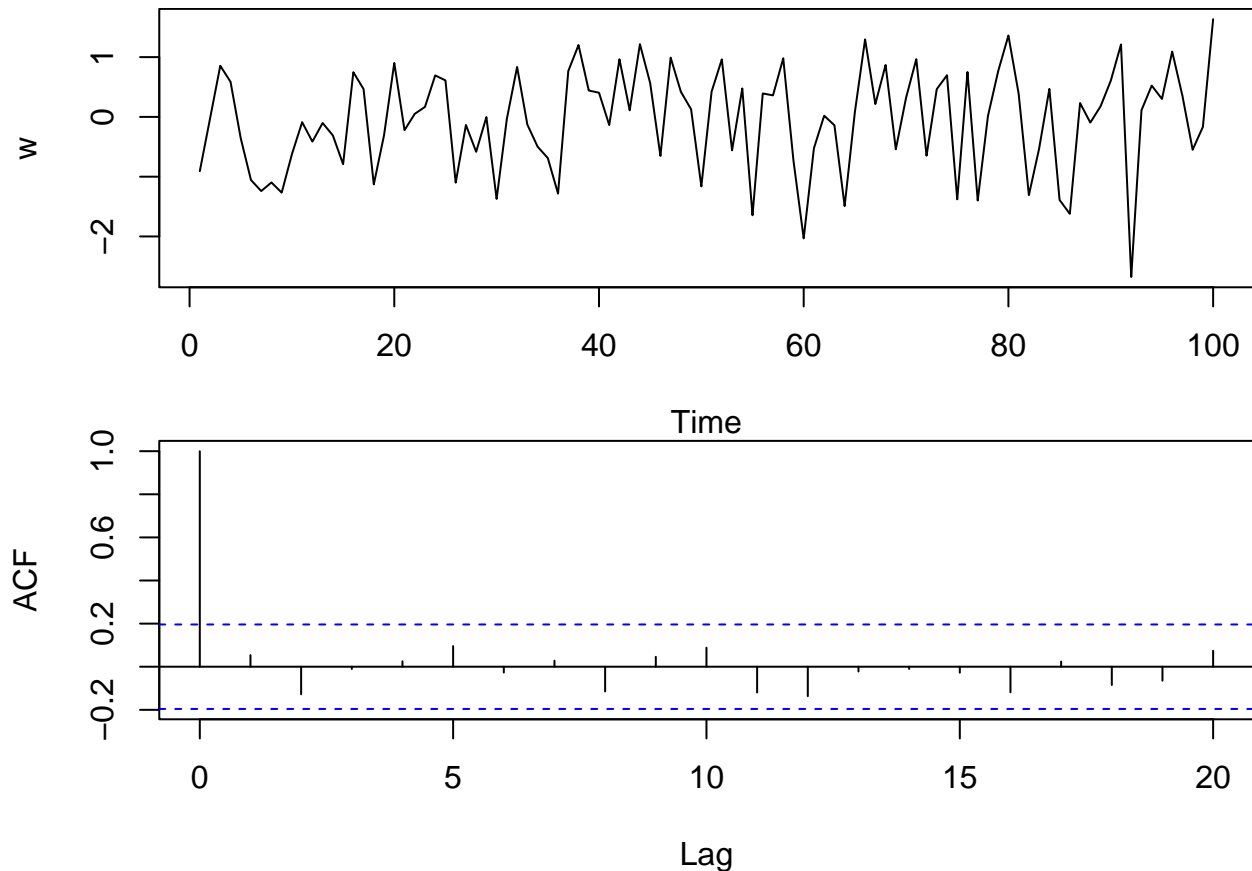
and autocorrelation function

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

2.2 Correlogram for 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)
```



- 95% of the estimated autocorrelations at lag $k > 0$ lie between the blue lines.
- It is a good idea to repeat this simulation and plot a few times to appreciate the variability of the results.

3 Random walk

3.1 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, assuming we start at $X_0 = 0$,

$$X_t = W_1 + W_2 + \dots + W_t$$

3.2 Properties of random walk

- A random walk X_t has a constant mean function

$$\mu(t) = 0.$$

- The variance function is given by

$$\sigma^2(t) = \text{Var}(X_t) = \text{Var}(W_1 + \dots + W_t) = \text{Var}(W_1) + \dots + \text{Var}(W_t) = t \cdot \sigma^2,$$

which clearly depends on the time t , so the process is not stationary.

- Similarly, one can compute the autocovariance and autocorrelation function (see Cowpertwait and Metcalfe for details)

$$\text{Cov}(X_t, X_{t+k}) = t\sigma^2.$$

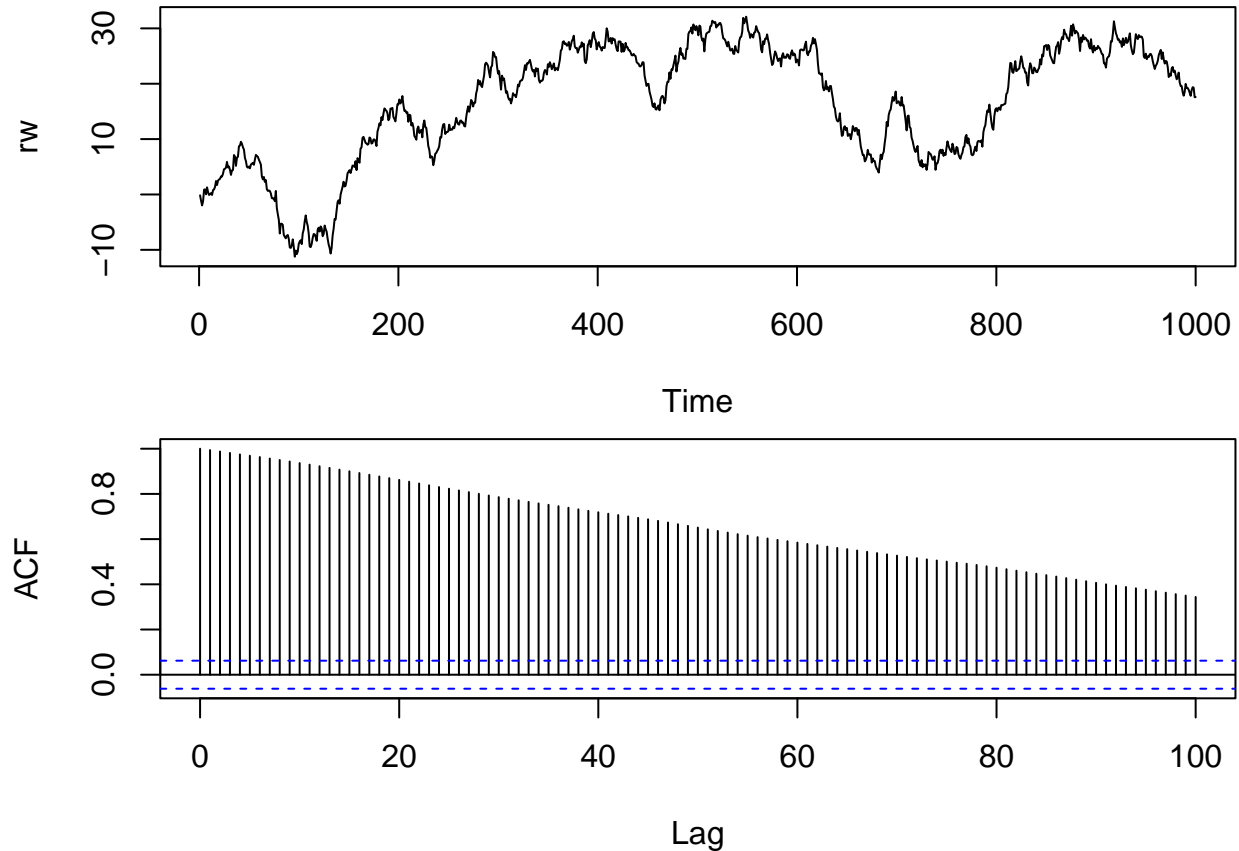
$$\text{Cor}(X_t, X_{t+k}) = \frac{1}{\sqrt{1 + k/t}}$$

- The autocorrelation is non-stationary.
 - When t is large compared to k we have very high correlation (close to 1)
 - We expect the correlogram of a reasonably long random walk to show very slow decay.
-

3.3 Simulation and correlogram of random walk

- We already know how to simulate Gaussian white noise W_t (with `rnorm`) and the random walk is just generated by repeatedly adding noise terms:

```
w <- rnorm(1000)
rw<-w
for(t in 2:1000) rw[t]<- rw[t-1] + w[t]
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(rw)
acf(rw, lag.max = 100)
```



- 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 unpredictably as a **stochastic trend**.

3.4 Differencing

- If a non-stationary time series shows a stochastic trend 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 $X_0 = 0$ we get

$$\nabla X_1 = X_1.$$

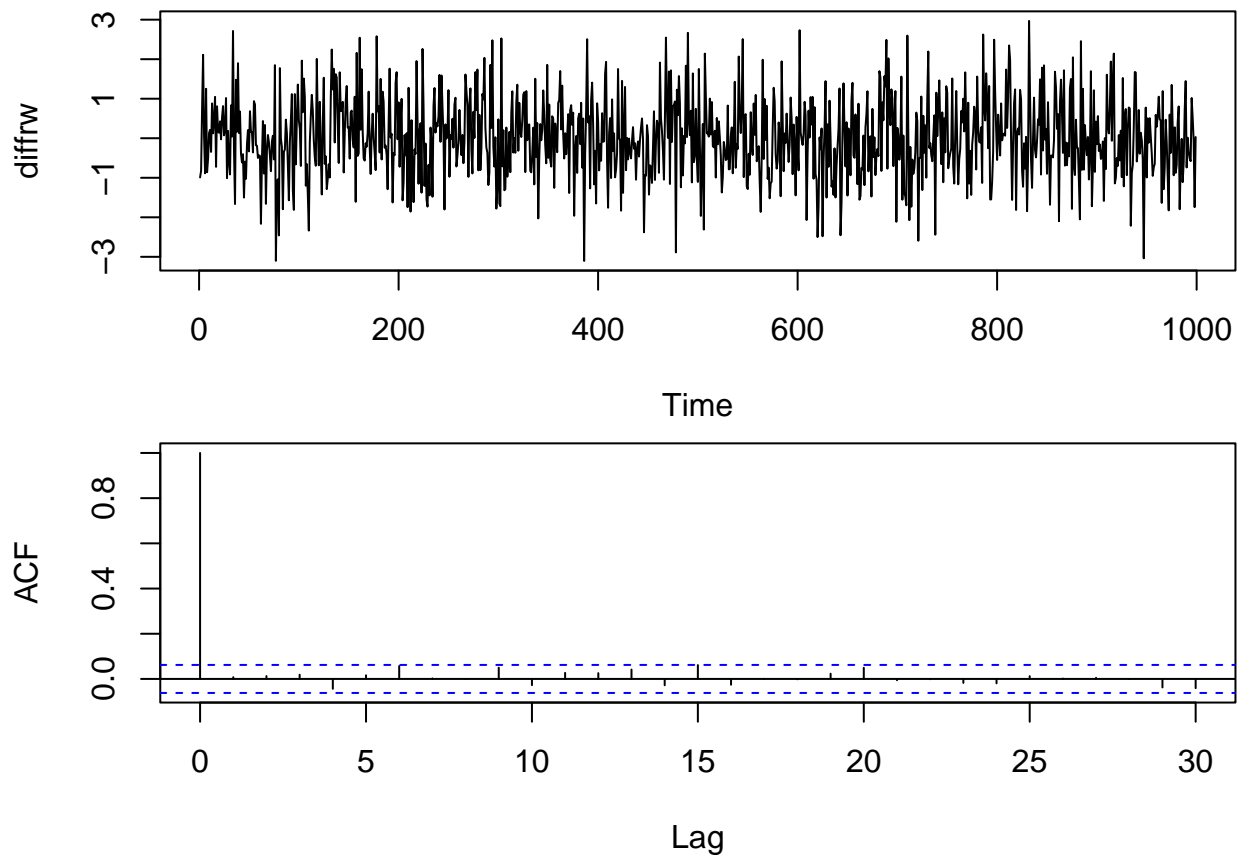
- Specifically when we difference a random walk $X_t = X_{t-1} + W_t$ we recover the white noise series

$$\nabla X_t = W_t$$

```

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

```



- If a time series needs to be differenced to become stationary, we say that the series is **integrated** (of order 1).

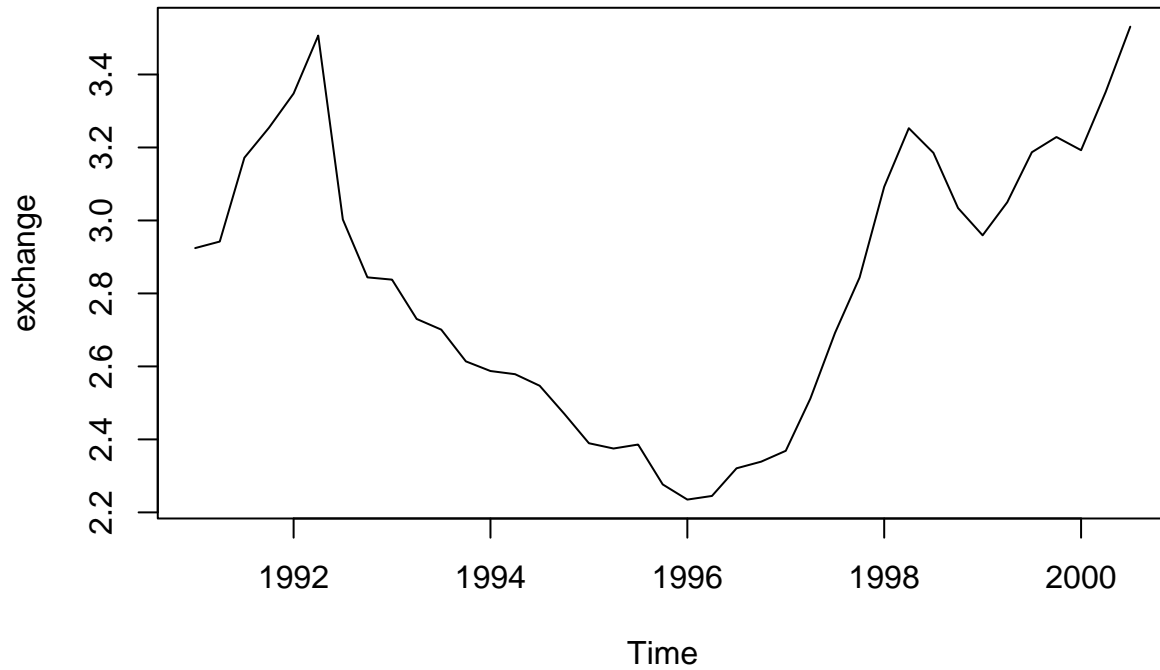
3.5 Example: Exchange rate

- We consider a data set for 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.

```

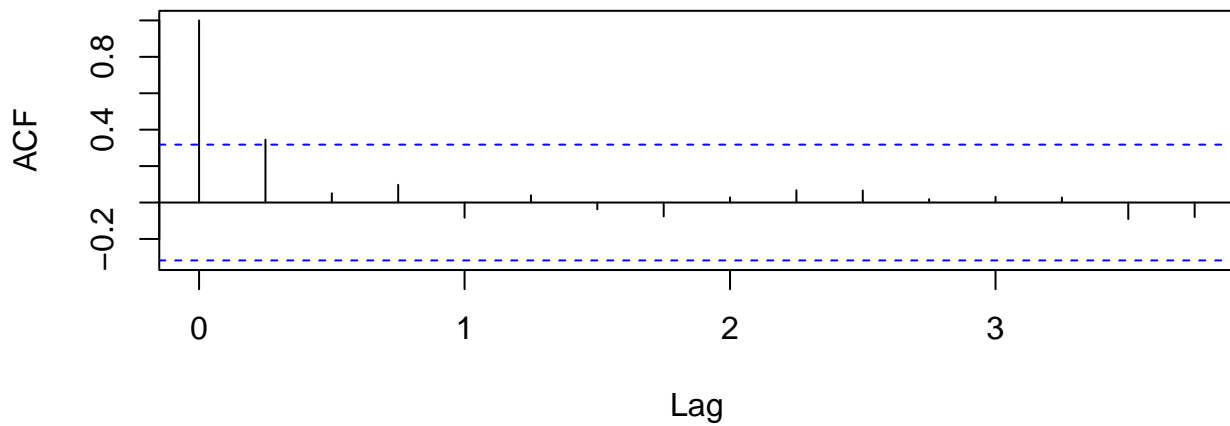
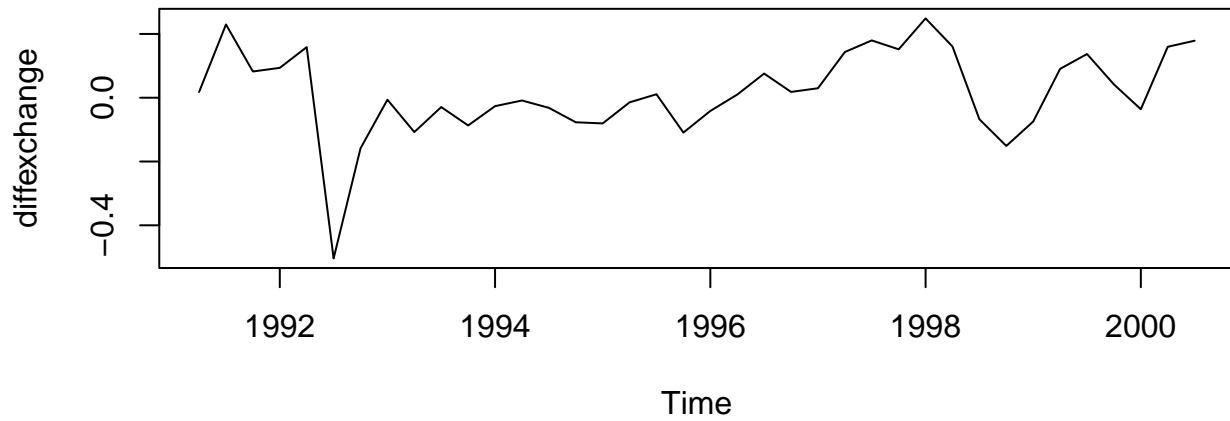
www <- "https://asta.math.aau.dk/eng/static/datasets?file=pounds_nz.dat"
exchange_data <- read.table(www, header = TRUE)[[1]]
exchange <- ts(exchange_data, start = 1991, freq = 4)
plot(exchange)

```



- We difference the series and see if the differences 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.

4 First order auto-regressive models AR(1)

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

- If the correlogram shows a significant auto-correlation at lag 1, it means that X_t and X_{t-1} are correlated.
- The simplest way to model this, is the 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.

4.2 Properties of AR(1) models

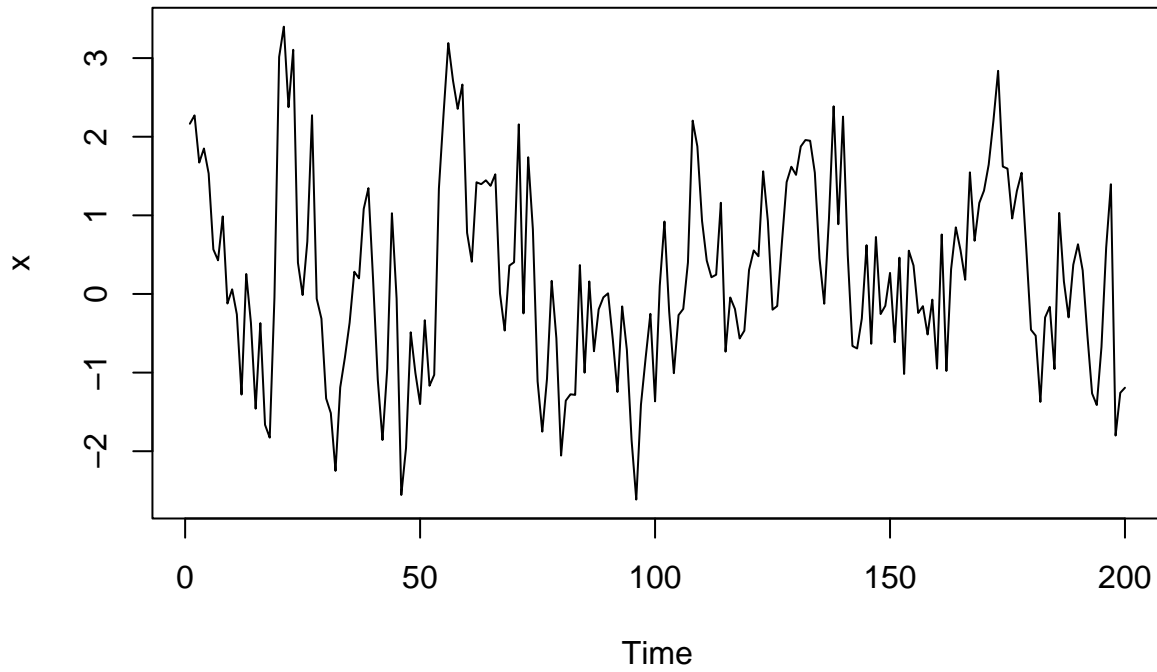
- From last time: The model can only be stationary if $-1 < \alpha_1 < 1$ such that the dependence on the past decreases with time.
- For a stationary AR(1) model with $-1 < \alpha_1 < 1$ we found

- $\mu(t) = 0$
- $Var(X_t) = \sigma_t^2 = \sigma^2 / (1 - \alpha_1^2)$
- $\gamma(k) = \alpha_1^k \sigma^2 / (1 - \alpha_1^2)$
- $\rho(k) = \alpha_1^k$

4.3 Simulation of AR(1) models

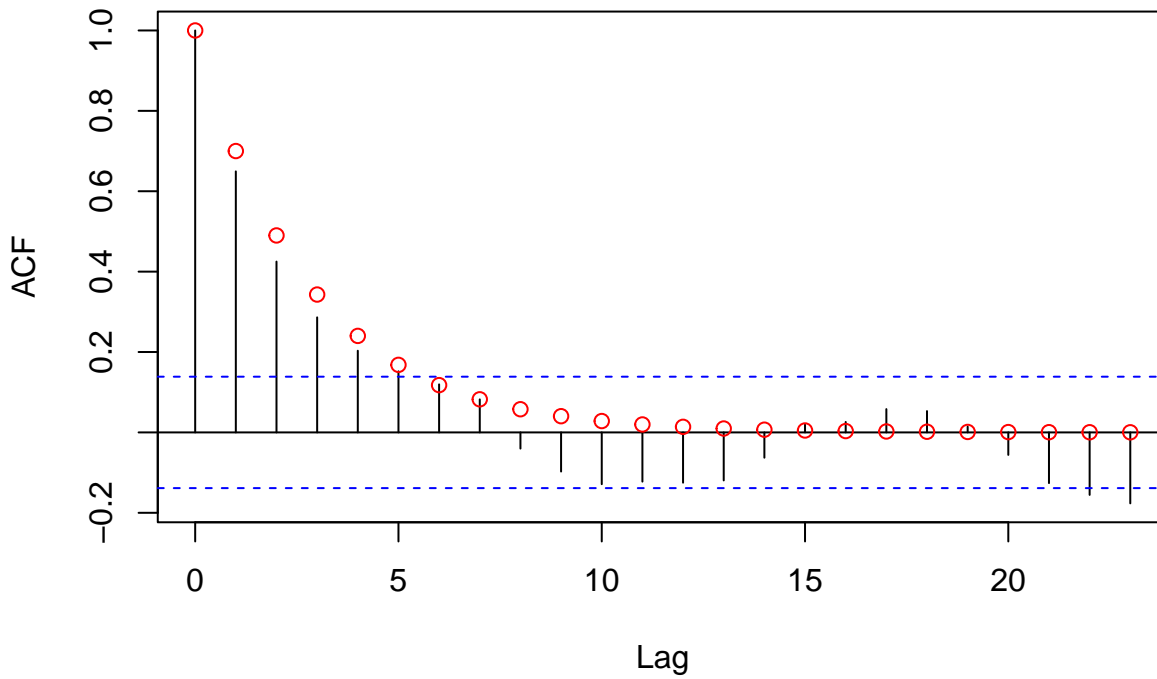
- R has a built-in function `arima.sim` to simulate AR(1) models (and other more complicated models called ARMA and ARIMA).
- It needs the model (i.e. the autoregressive coefficient α_1) and the desired number of time steps `n`.
- To simulate 200 time steps of AR(1) with $\alpha_1 = 0.7$ we do:

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

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

Series x



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

4.4 Fitted AR(1) models

- There are several ways to estimate the parameters in an AR(1) process. We use the so-called maximum likelihood estimation method (MLE).
- We skip the details of this, and simply use the function `ar`:

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

- The resulting object contains the value of the estimated parameter $\hat{\alpha}_1$ and a bunch of other information.
- In this example, the input data are artificial so we know that $\hat{\alpha}_1$ should ideally be close to 0.7:

```
fit$ar
```

```
## [1] 0.6587599
```

- An estimate of the variance (squared standard error) of $\hat{\alpha}_1$ is given in `fit$asy.var.coef`

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

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

```
## [1,] 0.002849969
```

- The estimated std. error is the square root of this:

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

```
se
```

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

```
## [1,] 0.0533851
```

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

```
ci
```

```
## [1] 0.5519897 0.7655301
```

4.5 Residuals for AR(1) models

- 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:

$$X_t - \bar{X} = \alpha_1 \cdot (X_{t-1} - \bar{X}) + W_t$$

- The fitted model can be used to make predictions. To predict X_t from 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

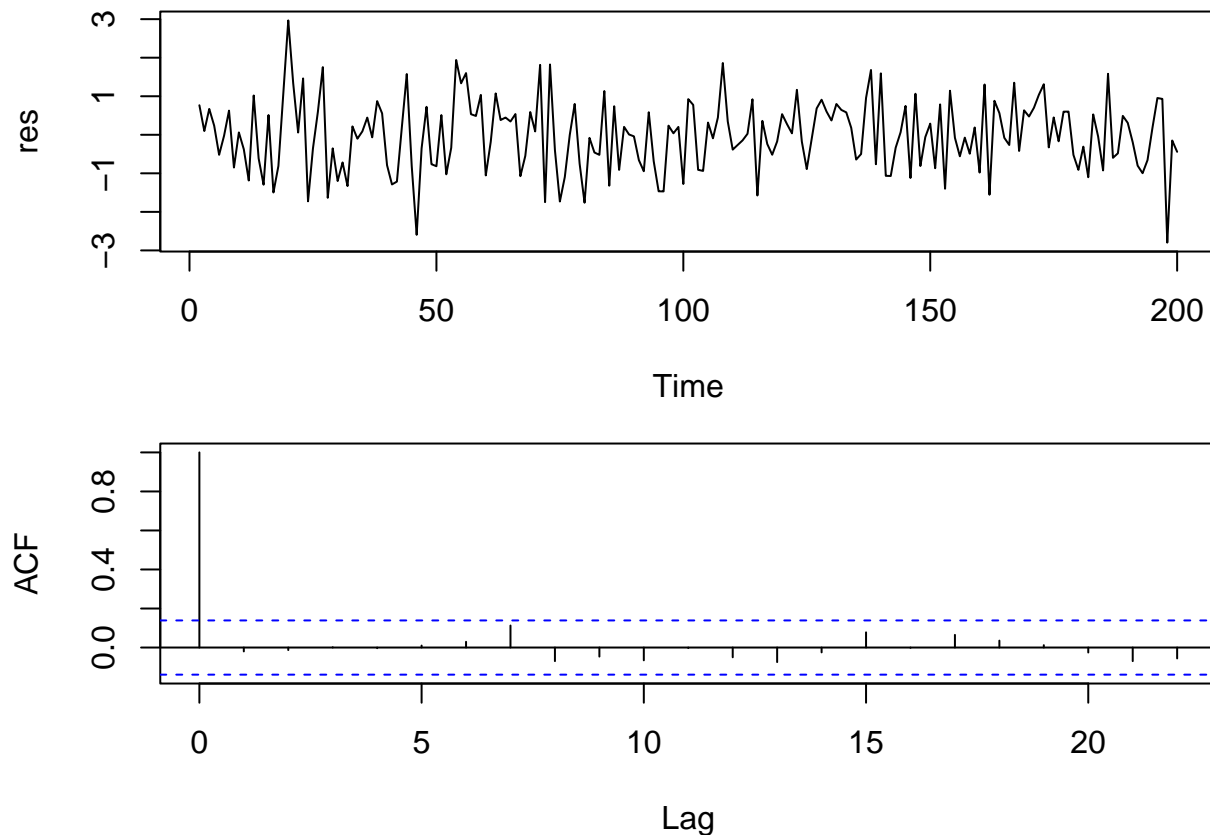
$$\hat{x}_t = \bar{x} + \hat{\alpha}_1 \cdot (x_{t-1} - \bar{x}), \quad t \geq 2.$$

- We can estimate the white noise terms as usual by the model residuals:

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

- If the model is correct, the residuals should look like a sample of white noise. We check this by plotting the acf:

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



- It naturally looks good for this artificial dataset.

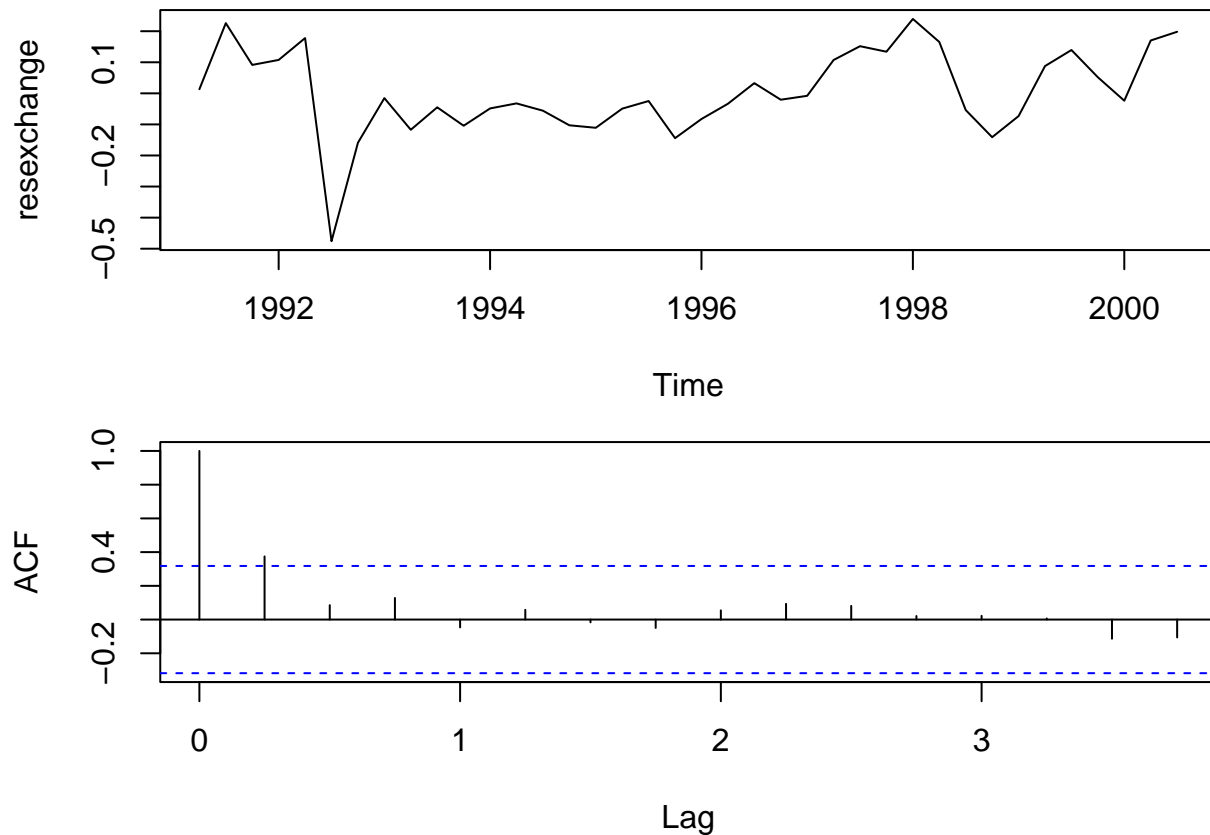
4.6 AR(1) model fitted to exchange rate

- A random walk is an example of an AR(1) model with $\alpha_1 = 1$. It 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:

```
fitexchange <- ar(exchange, order.max = 1, method="mle")
fitexchange$ar
```

```
## [1] 0.9437125
```

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

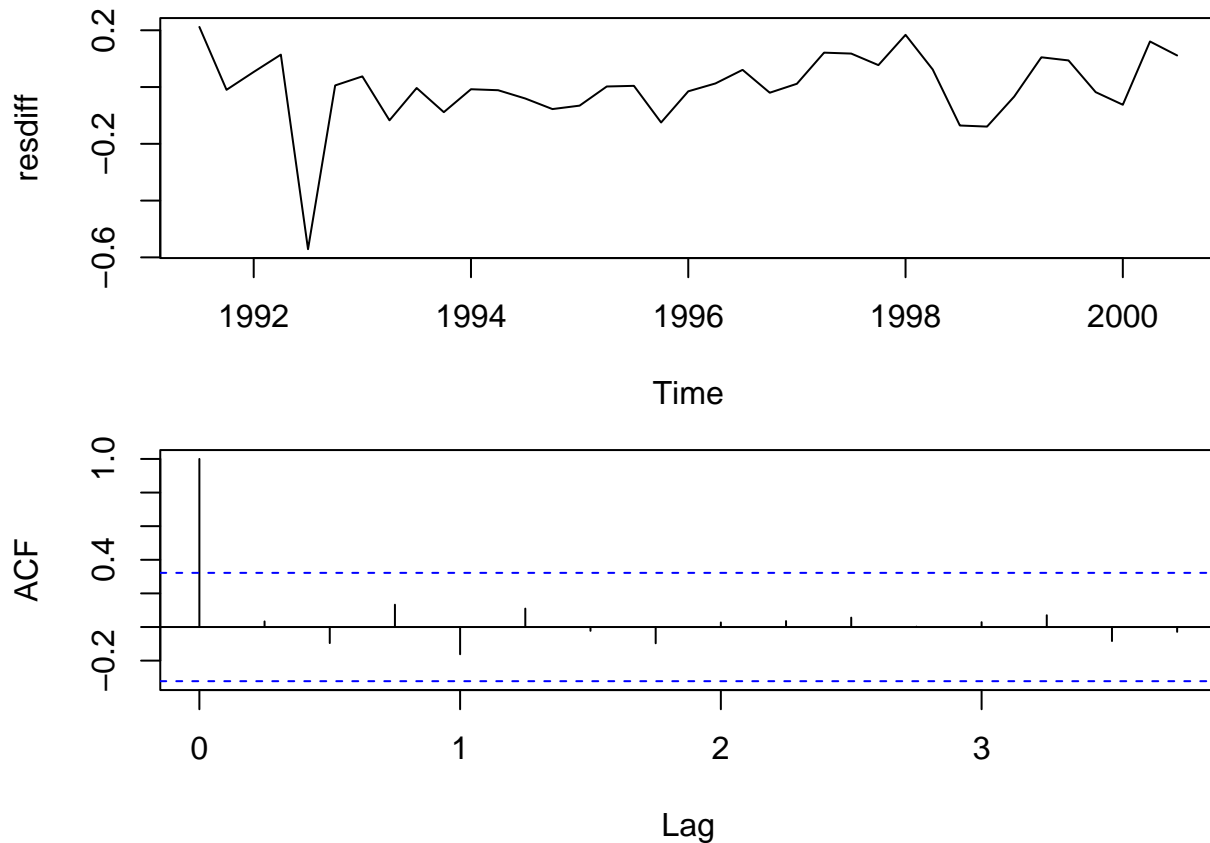


- This does not appear to really provide a better fit than the random walk model proposed earlier.
- An alternative would be to propose an AR(1) model for the differenced time series $\nabla X_t = X_t - X_{t-1}$:

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

```
## [1] 0.3495624
```

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



4.7 Prediction/forecasting from AR(1) model

- We can use a fitted AR(1) model to predict the next value of an observed time series x_1, \dots, x_n .

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

- This can be iterated to predict x_{n+2}

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

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:

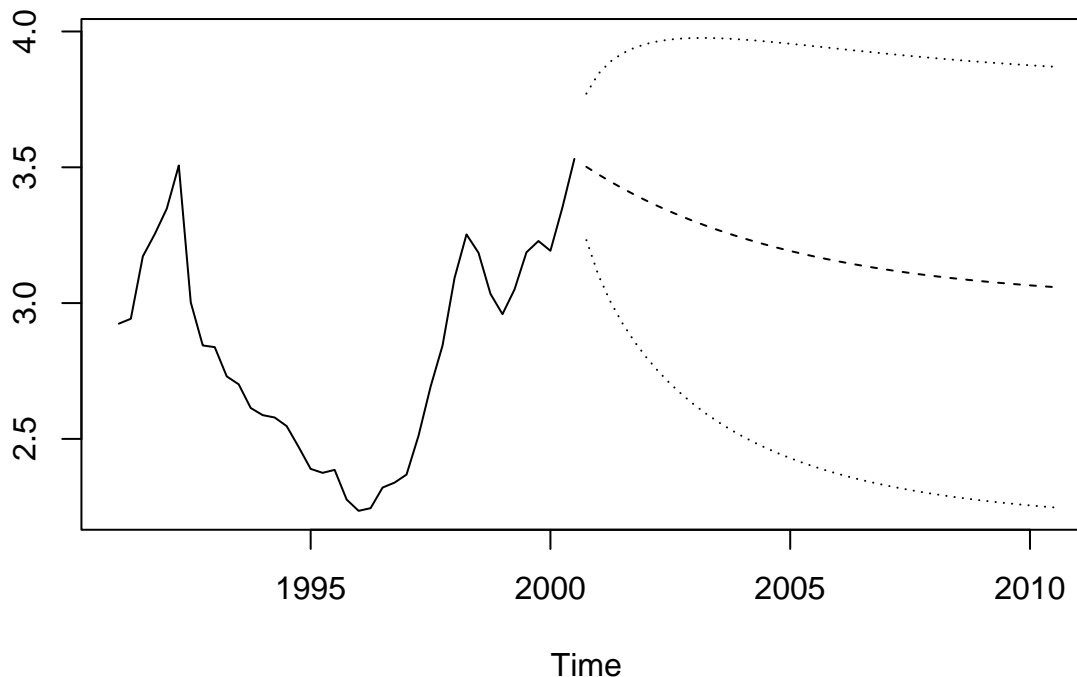
```
pred1 <- predict(fitexchange, n.ahead = 4)
pred1
```

```
## $pred
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2000                    3.501528
## 2001 3.473716 3.447469 3.422699
##
## $se
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2000                    0.1348262
```

```
## 2001 0.1853845 0.2208744 0.2482461
```

- Note how the prediction returns both the predicted value and a standard error for this value.
- We can use this to say that we are 95% confident that the exchange rate in third quarter of 2001 would be within $3.42 \pm 2 \cdot 0.25$.
- 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))
```



5 Auto-regressive models of higher order (AR(p)-models)

5.1 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

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} + W_t$$

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

- To simplify the notation, 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 $B^2X_t = BBX_t = BX_{t-1} = X_{t-2}$. We can then make a “polynomial” of backshift operators

$$\alpha(B) = 1 - \alpha_1B - \alpha_2B^2 - \dots - \alpha_pB^p$$

and write the AR(p) process as

$$W_t = \alpha(B)X_t.$$

5.2 Stationarity for AR(p) models

- Not all AR(p) models are stationary. To check that a given AR(p) model is stationary we consider the **characteristic equation**

$$1 - \alpha_1z - \alpha_2z^2 - \dots - \alpha_pz^p = 0.$$

- The characteristic equation can be written with the polynomial α from previous slide, but with z inserted instead of B , i.e.

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

- The process is stationary if and only if all roots of the characteristic equation have an absolute value that is greater than 1.
 - Solving a p -order polynomial is hard for high values of p , so we will let R do this for us.
-

5.3 Estimation of AR(p) models

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

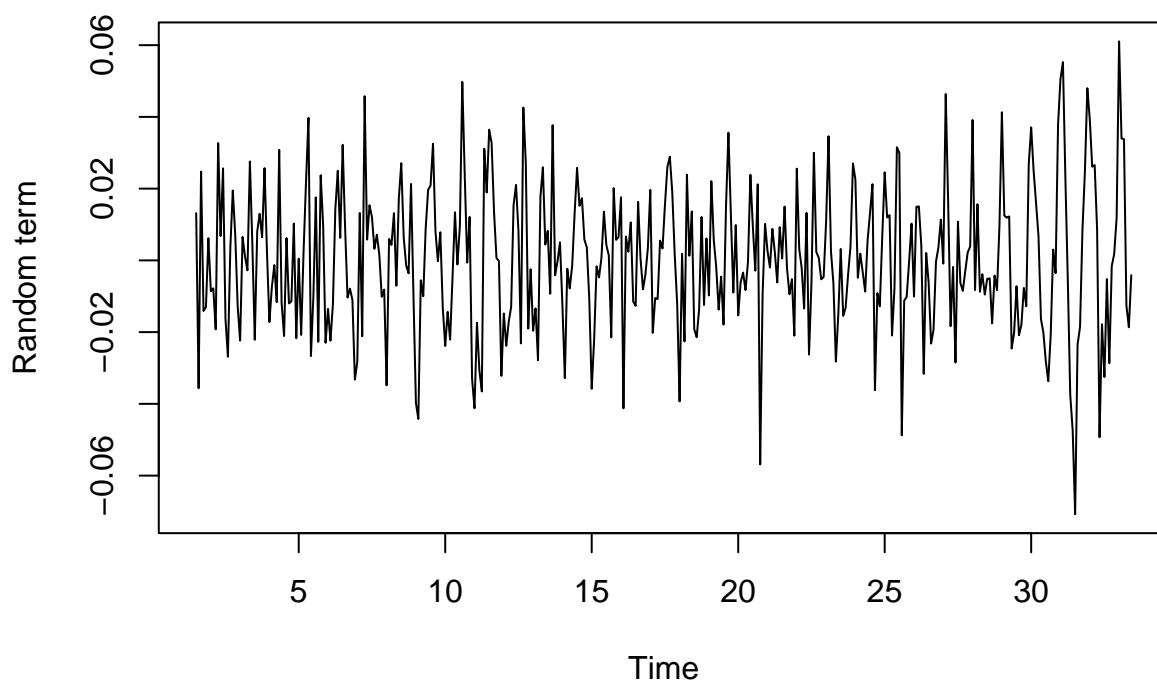
1. The maximal non-zero lag p in the model.
2. The autoregressive coefficients/parameters $\alpha_1, \dots, \alpha_p$.

- To select p we can use **AIC (Akaike’s Information Criterion)**.
 - More complex models can always fit a dataset better
 - AIC is essentially a balance between model simplicity and good fit.
 - 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 p is chosen, the estimates $\hat{\alpha}_1, \dots, \hat{\alpha}_p$ can be computed by the ar function.
 - 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.
-

5.4 Example of AR(p) model for electricity data

- We try to fit an AR(p) model to the detrended data for the log of the electricity production in Australia:

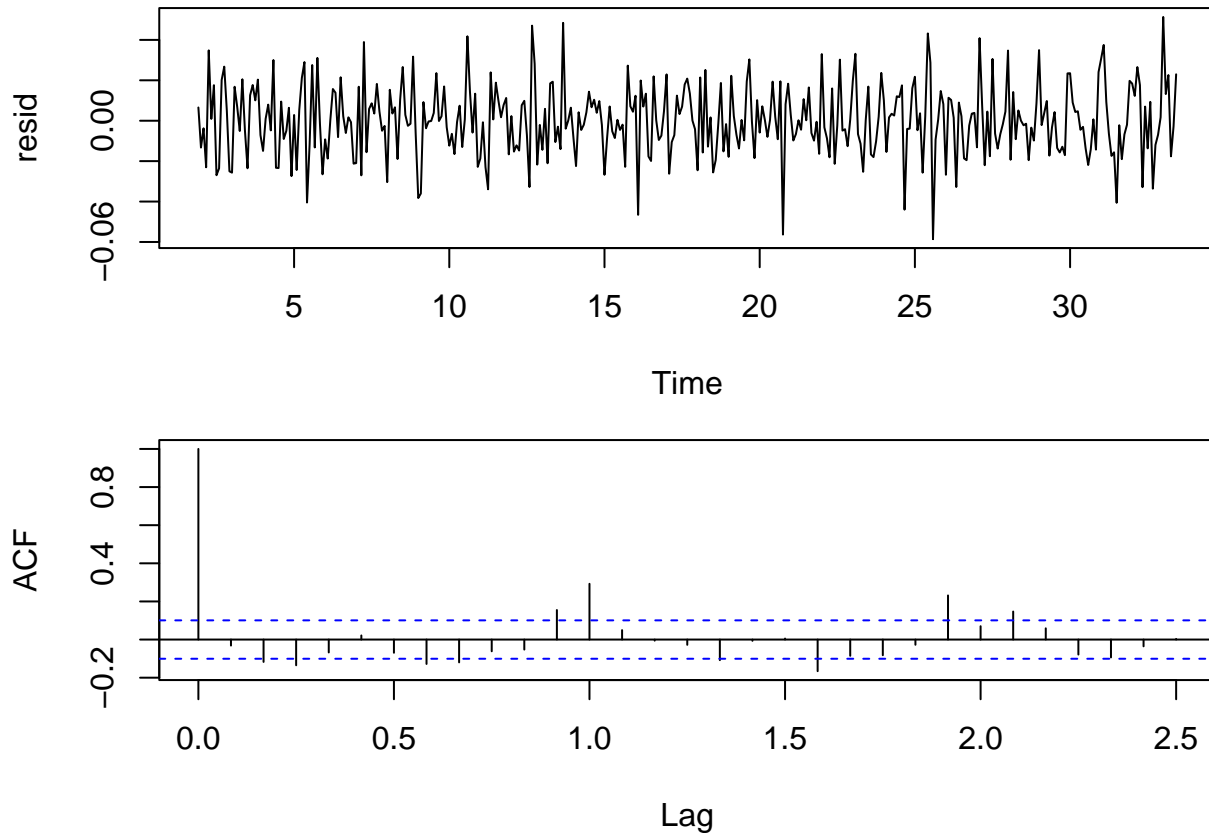
Electricity production



```
fit <- ar(random, order.max = 5, method="mle")
fit

##
## Call:
## ar(x = random, order.max = 5, method = "mle")
##
## Coefficients:
##      1      2      3      4      5
## 0.2047 0.0714 -0.0280 -0.1975 -0.2440
##
## Order selected 5  sigma^2 estimated as 0.0003103

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

- There are too many significant autocorrelations. Suggests the model is not a good fit.
- We are not assured that the estimated model will be stationary. To check stationarity we solve $\alpha(z) = 0$:

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

```
## [1] 1.148425 1.415987 1.415987 1.148425 1.549525
```

- All absolute values of the roots are greater than 1, indicating stationarity.

6 Moving average models (MA(q)-models)

6.1 The moving average model

- A **moving average process** of order q , $MA(q)$, is defined by

$$X_t = W_t + \beta_1 W_{t-1} + \beta_2 W_{t-2} + \dots + \beta_q W_{t-q}$$

where W_t is a white noise process with mean 0 and variance σ_w^2 and $\beta_1, \beta_2, \dots, \beta_q$ are parameters to be estimated.

- The moving average process can also be written using the backshift operator:

$$X_t = \beta(B)W_t$$

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

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

- Since a moving average process is a finite sum of stationary white noise terms it is itself stationary with mean and variance:

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

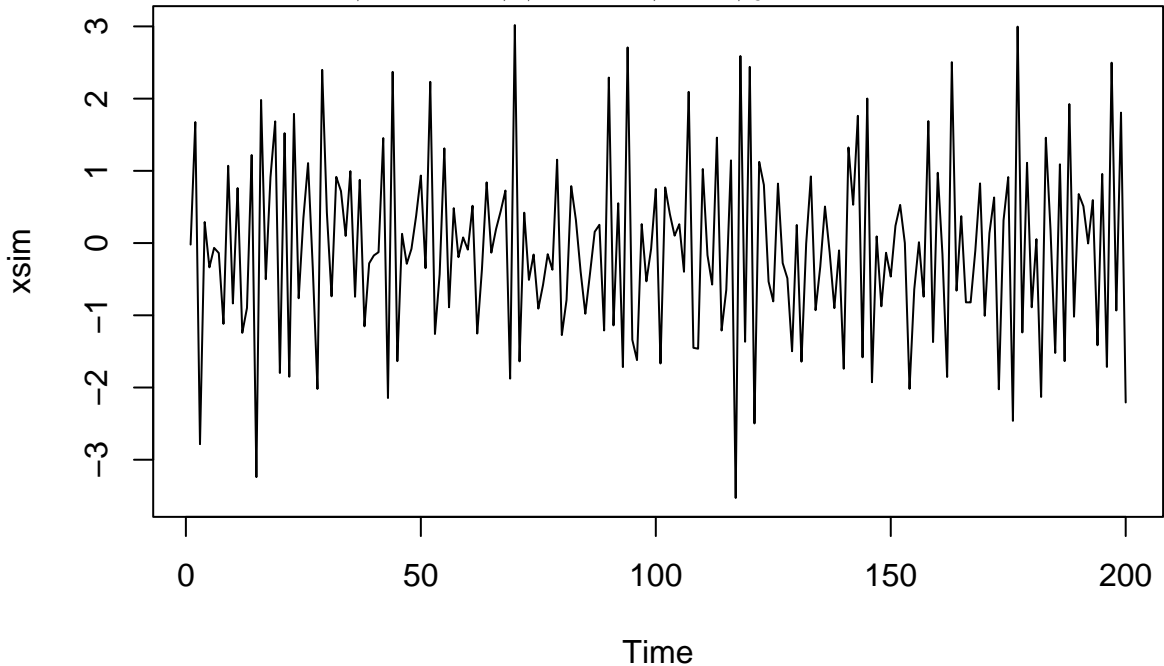
- The autocorrelation function is

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

where $\beta_0 = 1$.

6.2 Simulation of MA(q) processes

- An MA(q) process can be simulated by first generating a white noise process W_t and then transforming it using the MA coefficients.
- To simulate a model with $\beta_1 = -0.7$, $\beta_2 = 0.5$, and $\beta_3 = -0.2$ we can use `arma.sim`:



- The theoretical autocorrelations are in this case:

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

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

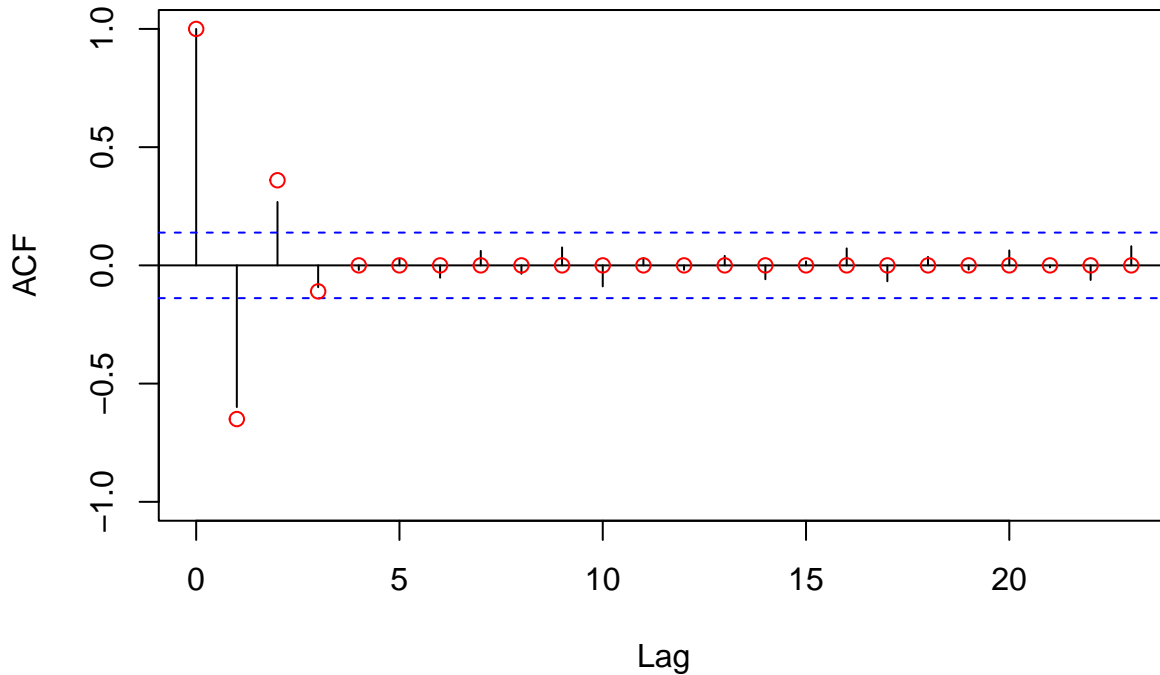
$$\rho(3) = \frac{1 \cdot (-0.2)}{1 + (-0.7)^2 + 0.5^2 + (-0.2)^2} = -0.11$$

and $\rho(k) = 0$ for $k > 3$.

- We plot the acf of the simulated series together with the theoretical one.

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

Series xsim



6.3 Estimation of MA(q) models

- To estimate the parameters of an MA(3) model we use `arima`:

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

```
##
## Call:
## arima(x = xsim, order = c(0, 0, 3))
##
## Coefficients:
##      ma1      ma2      ma3  intercept
## -0.6941  0.3363 -0.1554  -0.0653
## s.e.   0.0706  0.0836  0.0662   0.0341
##
## sigma^2 estimated as 0.9702:  log likelihood = -281.03,  aic = 572.05
```

- 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 choosing the model with the minimal AIC.

```
AIC(xfit)
```

```
## [1] 572.0522
```

7 Auto-regressive moving average models (ARMA)

7.1 Mixed models: Auto-regressive moving average models

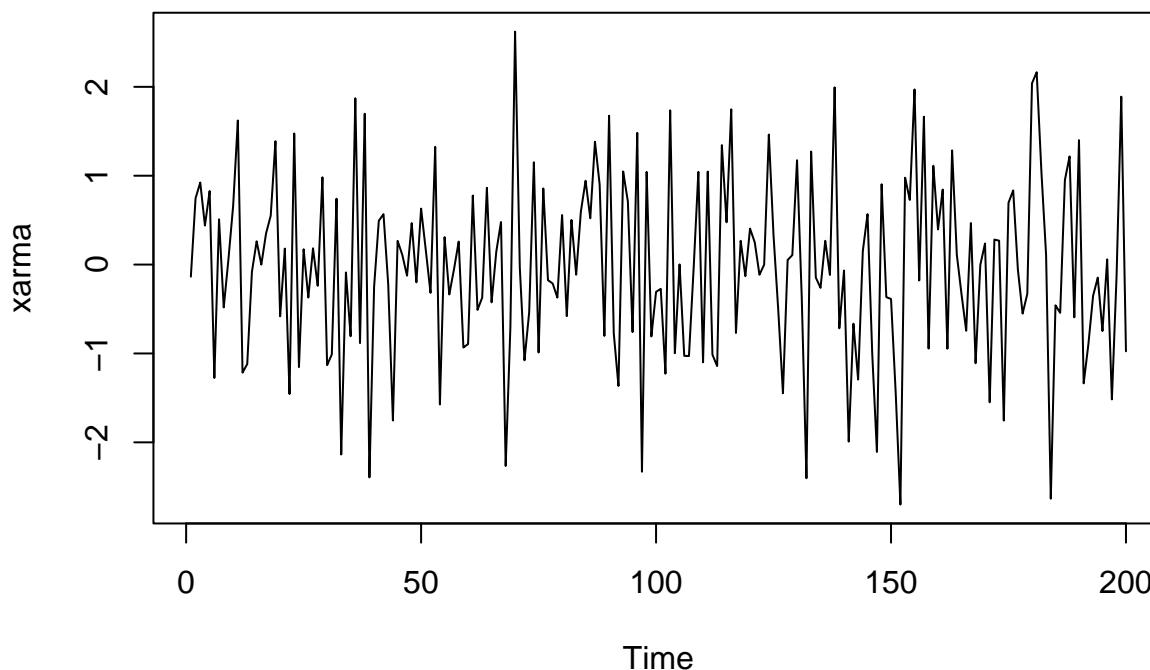
- A time series X_t follows an auto-regressive moving average (ARMA) process of order (p, q) , denoted $ARMA(p, q)$, if

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} + W_t + \beta_1 W_{t-1} + \beta_2 W_{t-2} + \dots + \beta_q W_{t-q}$$

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

- We can simulate an ARMA model with `arima.sim`. E.g. an ARMA(1,1) model with $\alpha_1 = -0.6$ and $\beta_1 = 0.5$:

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



- Estimation is done with `arima` as before.
-

7.2 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 models 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
```

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

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

```
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
```

```
par(mfrow = c(2,1), mar = c(4,4,1,1))
```

```
resid_arma <- na.omit(exchange_arma$residuals)
```

```
plot(resid_arma)
```

```
acf(resid_arma)
```

