

Stochastic processes III

The ASTA team

Contents

1	Models with exogenous variables	1
1.1	Exogenous variables	1
1.2	Data example	1
1.3	Regression models with exogenous variables	3
1.4	Example	3
1.5	Simulation of the example	3
1.6	Estimation and model checking	4
1.7	Fitting AR(1) model to data example	6
1.8	Prediction	8
1.9	An example with delay	9
1.10	The cross-correlation function	10
1.11	Fitting models with lag	12
1.12	ARMAX models	12
2	Continuous time processes	13
2.1	Discrete vs. continuous time	13
2.2	Continuous time stochastic processes	13
2.3	The Wiener process	13
2.4	Stochastic differential equations	14
2.5	Stochastic differential equations	15

1 Models with exogenous variables

1.1 Exogenous variables

- The ARMA processes are flexible models for a time series Y_t , $t = 1, \dots, n$ evolving randomly over time, but they do not include the possibility that anything is influencing Y_t .
 - An exogeneous variable is another variable, say X_t , that influences the behaviour of Y_t
 - Wind power production Y_t is influenced by the wind speed X_t
 - The velocity of a DC motor Y_t is influenced by the input voltage X_t
 - Here X_t may be another stochastic process, which we do not model, but only consider as given, or it might be something we can control.
-

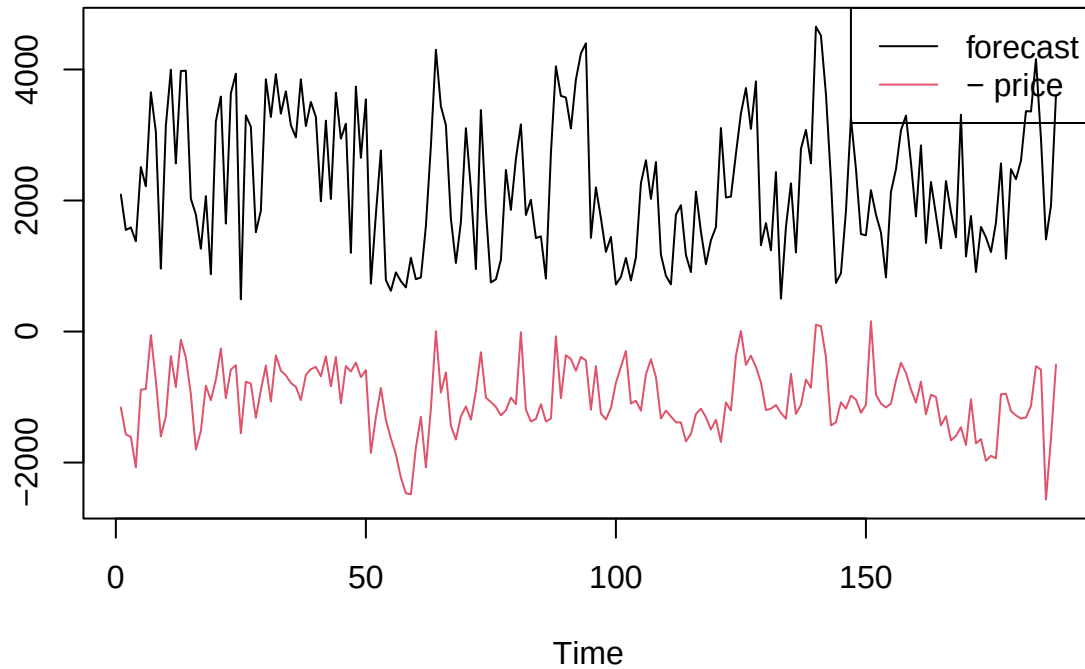
1.2 Data example

- The dataset below contains data from Jan 7 to Jul 13 2022 on two variables
 - forecast: Total day ahead forecasted wind and solar energy production
 - price: Day ahead elspot prices with weekly variation removed

```

elspot<-read.csv("https://asta.math.aau.dk/eng/static/datasets?file=elspot.csv", header = TRUE)
forecast<-elspot[,2]
price<-elspot[,3]
ts.plot(ts(forecast),ts(-price),col=1:2)
legend("topright",legend=c("forecast","- price"),col=1:2,lty=1)

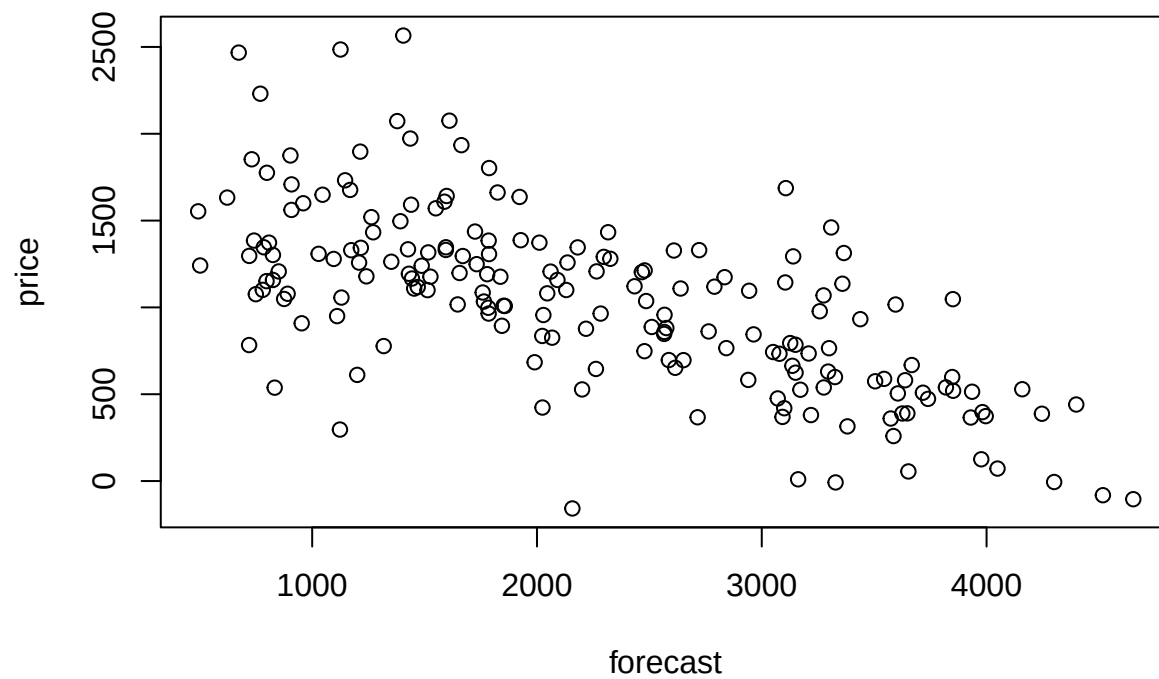
```



```

plot(forecast,price)

```



1.3 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 of Y_t on X_t , but where the noise term is an ARMA process:

$$Y_t = \gamma_0 + \gamma_1 X_t + \epsilon_t, \quad \alpha(B)\epsilon_t = \beta(B)W_t$$

- If we isolate $\epsilon_t = Y_t - \gamma_0 - \gamma_1 X_t$ 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:

$$\alpha(B)(Y_t - \gamma_0 - \gamma_1 X_t) = \beta(B)W_t$$

- 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 .
 - Above, X_t is a single stochastic process, but we can also include multiple stochastic processes by making a multiple regression model with an ARMA model for the errors.
-

1.4 Example

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

$$Y_t = \gamma_0 + \gamma_1 X_t + \epsilon_t, \quad \epsilon_t = \alpha_1 \epsilon_{t-1} + W_t$$

or, since $\epsilon_{t-1} = Y_{t-1} - \gamma_0 - \gamma_1 X_{t-1}$,

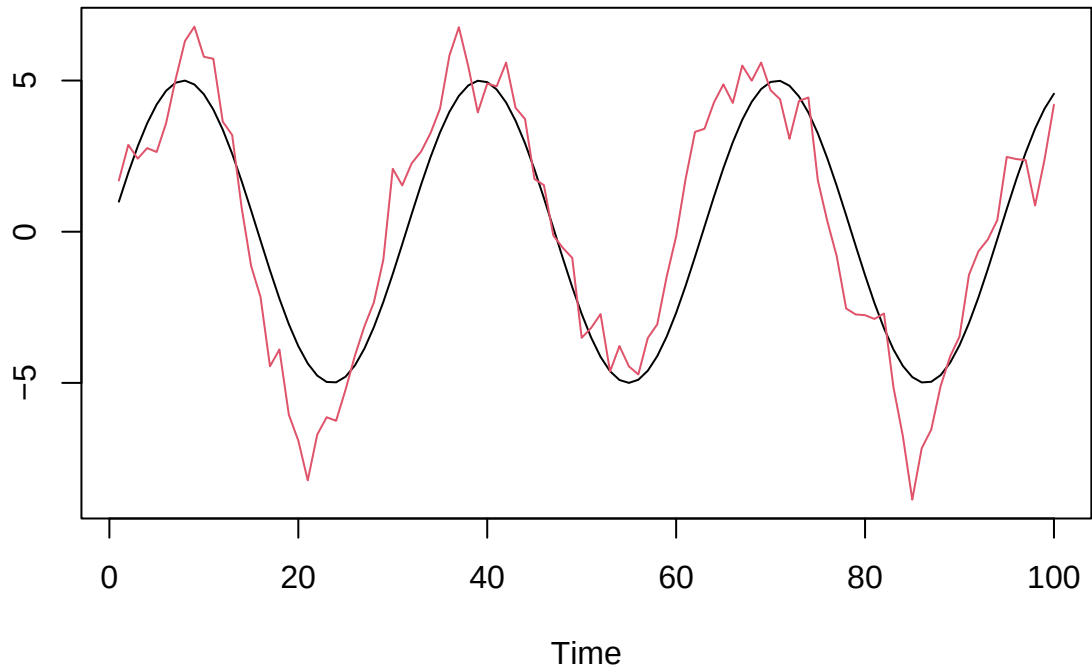
$$Y_t = \alpha_1 Y_{t-1} + (1 - \alpha_1)\gamma_0 + \gamma_1(X_t - \alpha_1 X_{t-1}) + W_t$$

- 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.
-

1.5 Simulation of the example

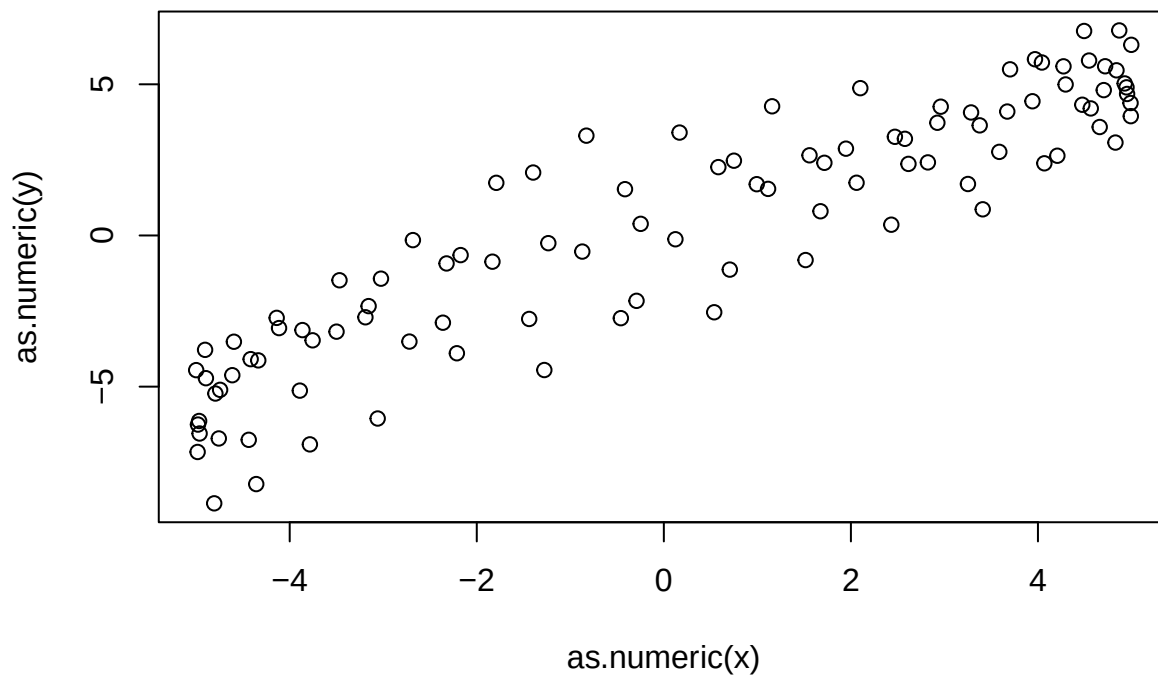
- We simulate some data resembling the example, where we let X_t follow a sine curve:

```
alpha = 0.9; gamma = 1; n = 100
x = as.ts(5*sin(1:n/5))
eps = arima.sim(model=list(ar=alpha),n=n)
y = gamma*x+eps
ts.plot(x,y,col=1:2)
```



- We should think of the red curve as some data we want to model, and the black curve as another variable which we believe may influence the data.
- We can also plot X_t against Y_t to get a view of the relation between the two variables.

```
plot(as.numeric(x),as.numeric(y))
```



1.6 Estimation and model checking

- We can estimate the parameters using the `arima` function in R.

- We fit a linear regression model with AR(1) noise to the simulated data (i.e. the true model used for simulation):

```
mod=arima(y,order=c(1,0,0),xreg=x); mod
```

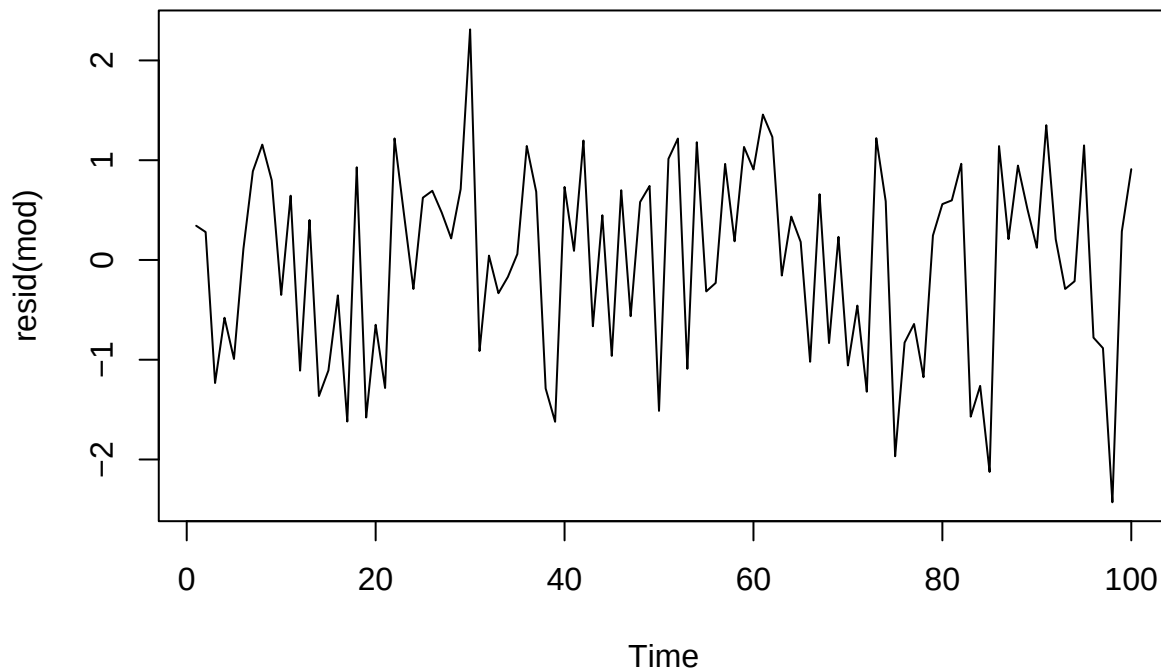
```
##
## Call:
## arima(x = y, order = c(1, 0, 0), xreg = x)
##
## Coefficients:
##          ar1  intercept          x
##       0.8069   0.0679   1.0551
## s.e.  0.0569    0.4795   0.1018
##
## sigma^2 estimated as 0.923:  log likelihood = -138.41,  aic = 284.82
```

- The fitted model becomes

$$Y_t = 0.0679 + 1.0551 \cdot X_t + \epsilon_t, \quad \epsilon_t = 0.8069 \cdot \epsilon_{t-1} + W_t$$

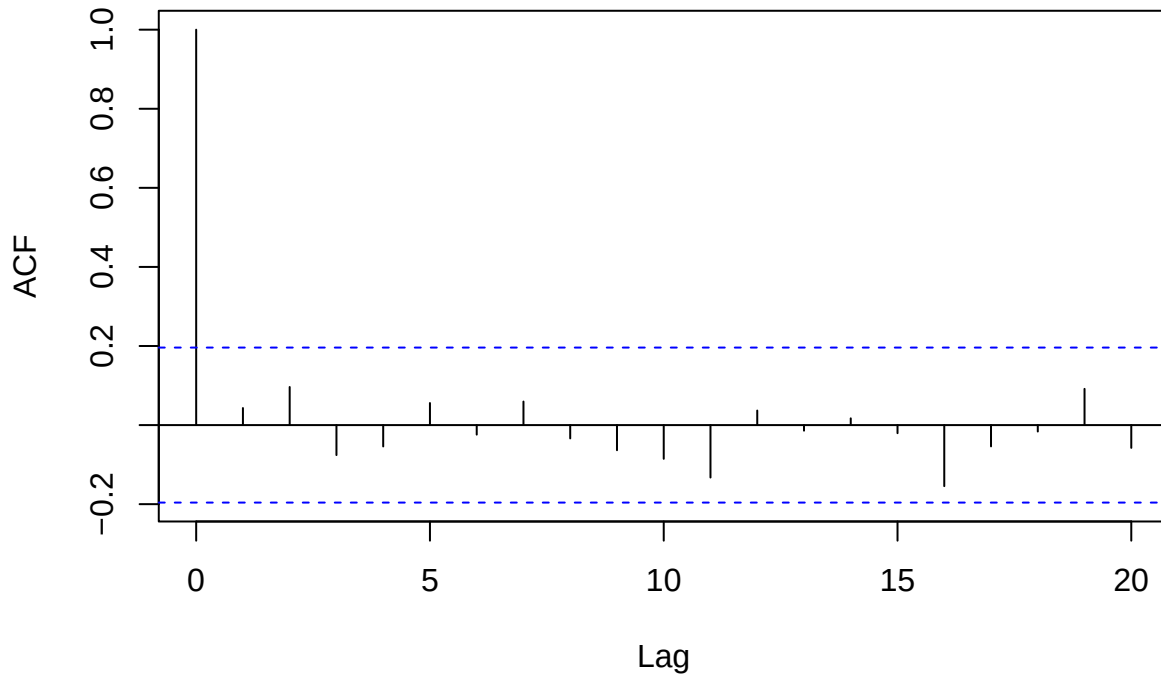
- The errors $\hat{\epsilon}_t = y_t - 0.0679 + 1.0551 \cdot x_t$ should behave like an AR(1)-model with $\hat{\alpha} = 0.8069$.
 - So the residuals $\epsilon_t - 0.8069 \cdot \epsilon_{t-1}$ should look like white noise.

```
plot(resid(mod))
```



```
acf(resid(mod))
```

Series resid(mod)



1.7 Fitting AR(1) model to data example

- Recall the elspot price dataset

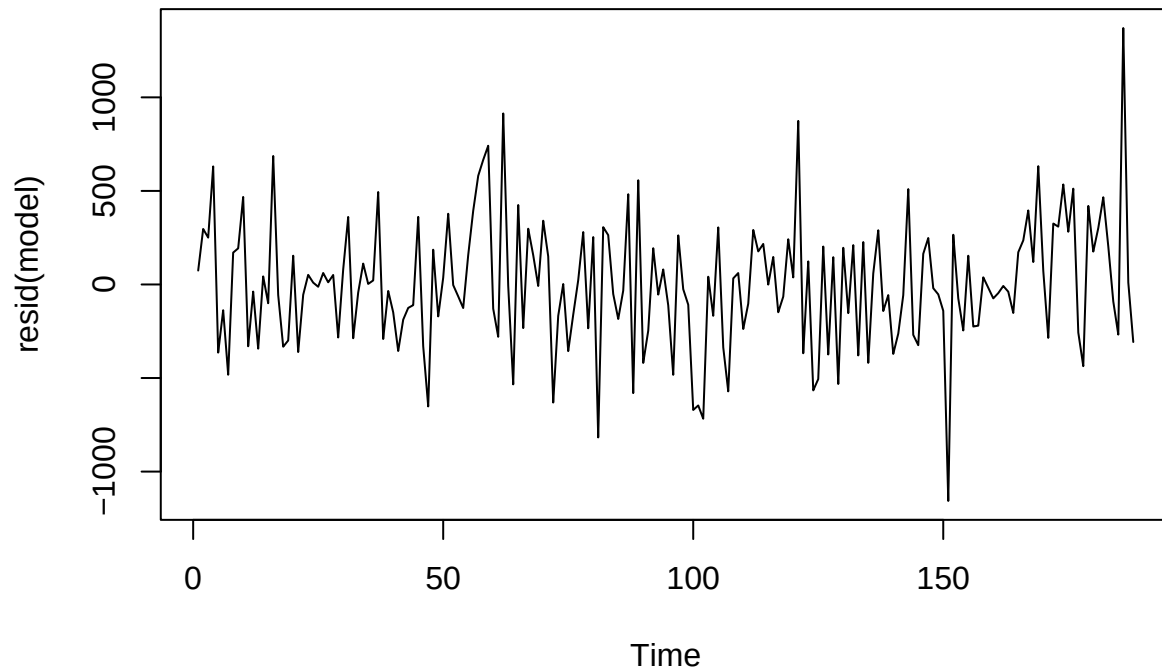
```
forecast<- ts(forecast)
price<-ts(price)
model=arima(price,order=c(1,0,0),xreg=forecast); model
```

```
##
## Call:
## arima(x = price, order = c(1, 0, 0), xreg = forecast)
##
## Coefficients:
##          ar1  intercept  forecast
##      0.3886  1715.8412   -0.3053
## s.e.  0.0680    73.2894    0.0271
##
## sigma^2 estimated as 117486:  log likelihood = -1364.2,  aic = 2736.41
```

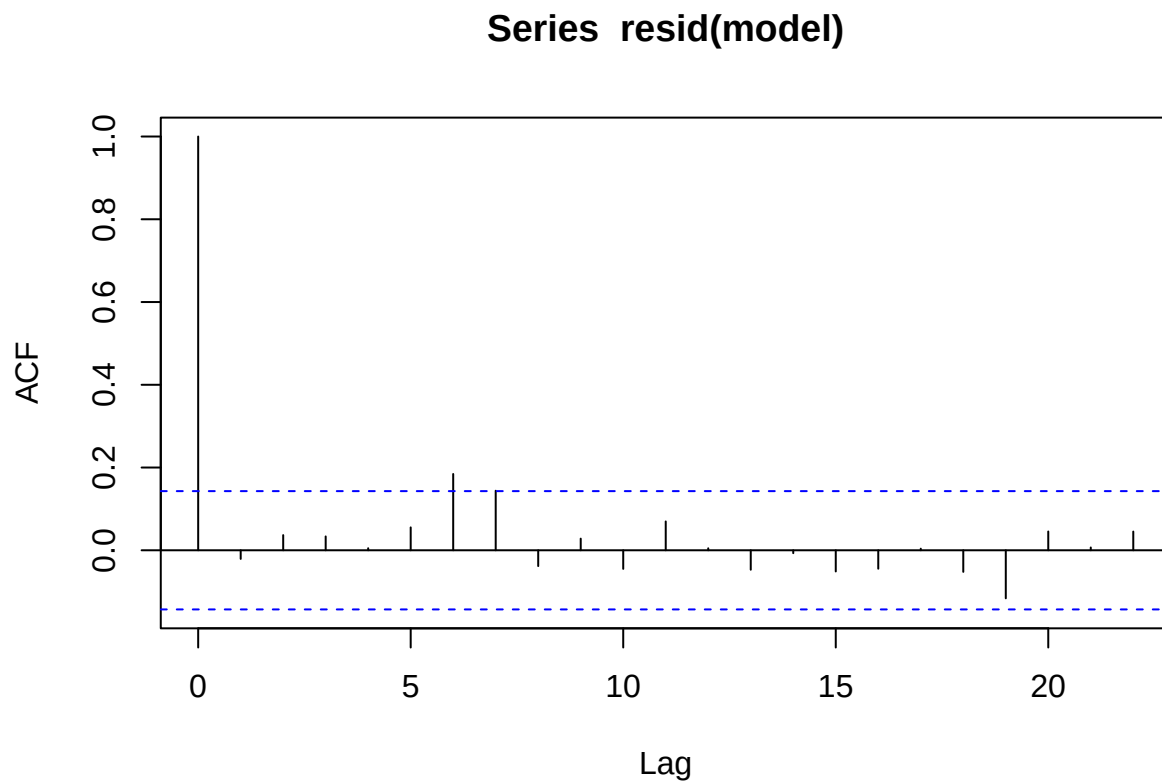
- So we get the model

$$\text{price}_t = 1715.8412 - 0.3053 \cdot \text{forecast}_t + \epsilon_t, \quad \epsilon_t = 0.3886 \cdot \epsilon_{t-1} + W_t.$$

```
plot(resid(model))
```



```
acf(resid(model))
```

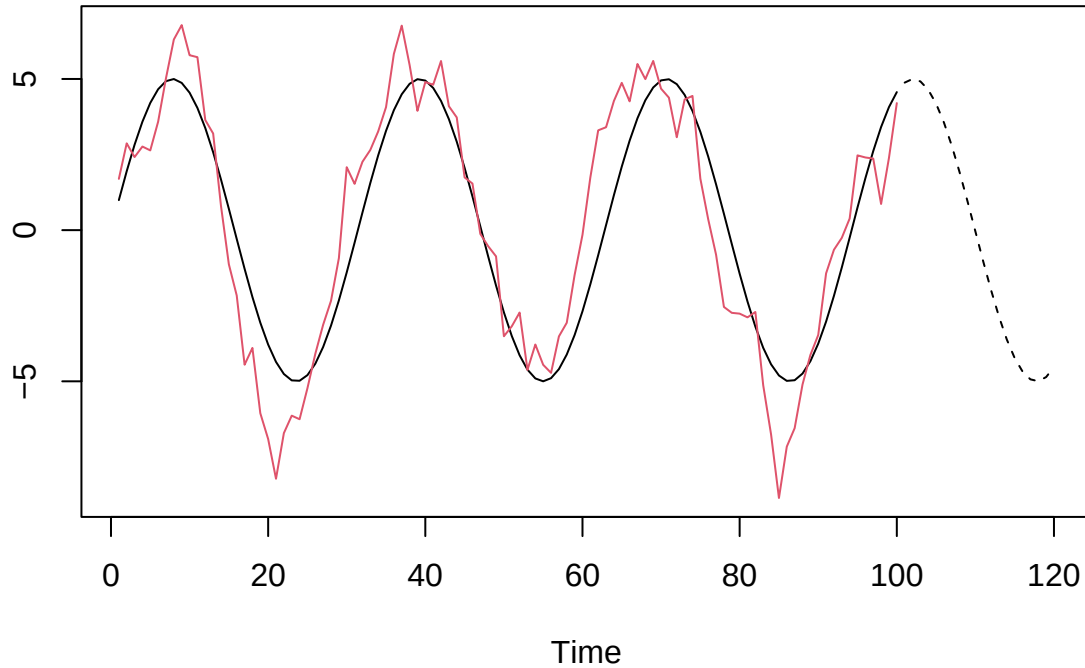


- Residuals indicate that there could be some weekly variation not accounted for.

1.8 Prediction

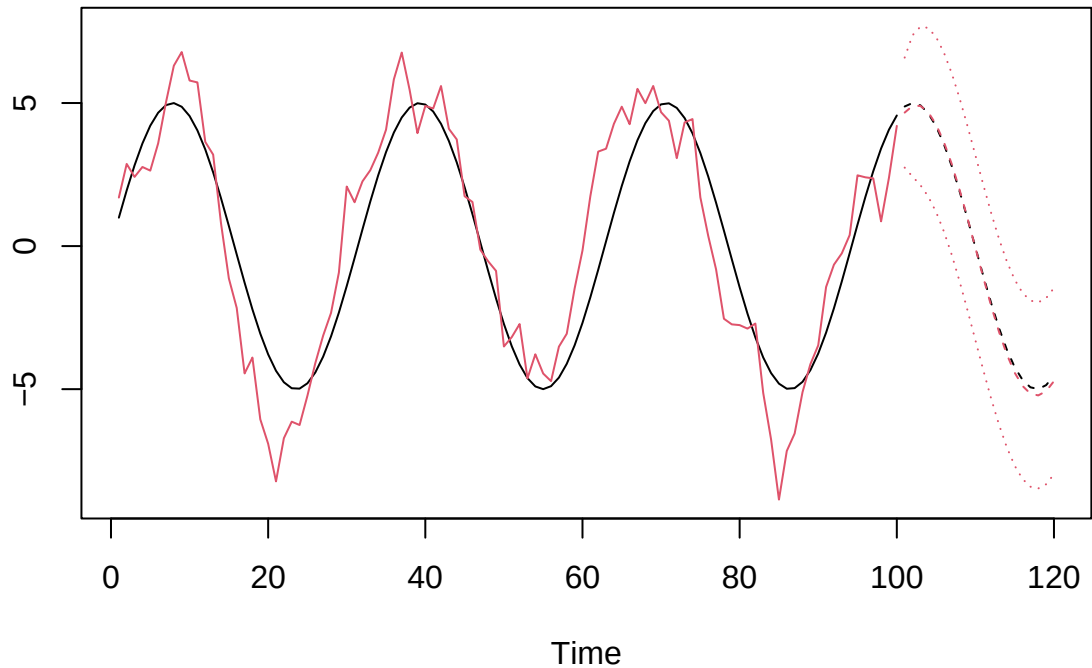
- Prediction can only be performed if we know the behavior of X_t for future time points, for example if we are able to control it.
- For the previous example we assume that the sine curve continues:

```
nnew = 20  
xnew = lag(as.ts(5*sin(((n+1):(n+nnew))/5)),-n)  
ts.plot(x,y,xnew,col=c(1,2,1),lty=c(1,1,2))
```



- We use the predict function.

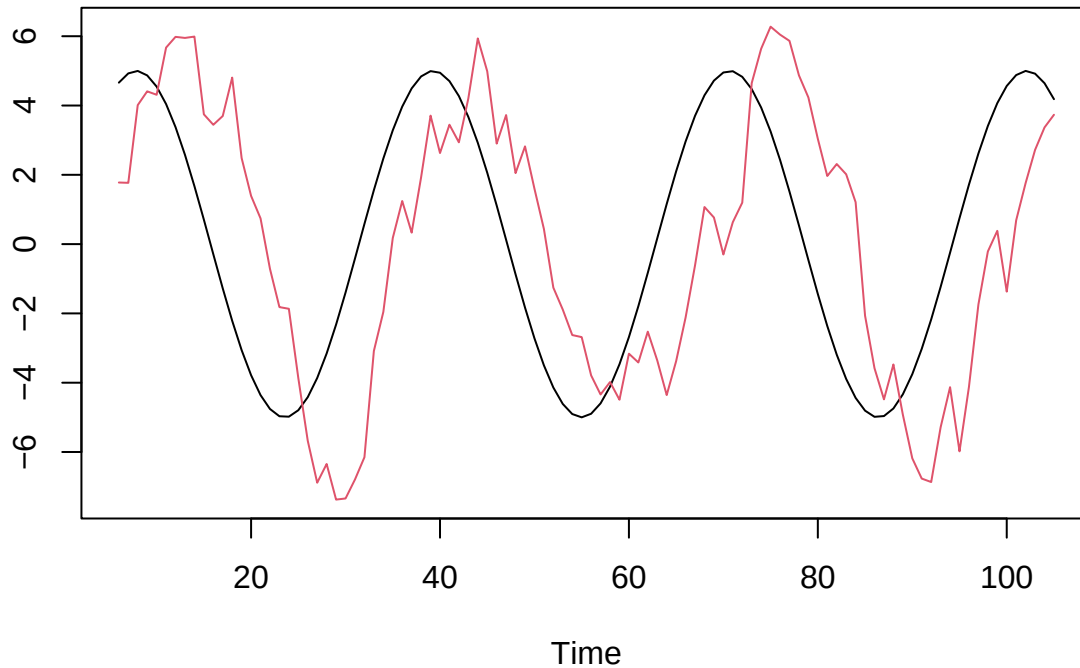
```
p = predict(mod,n.ahead=nnew,newxreg=xnew)  
ts.plot(x,y,xnew,p$pred,p$pred+2*p$se,p$pred-2*p$se,col=c(1,2,1,2,2,2),lty=c(1,1,2,2,3,3))
```

1.9 An example with delay

- If we model the influence of X_t on Y_t , it may take some time before Y_t responds to a change in X_t .
 - Say the delay is k time steps.
- We want to model the effect of X_{t-k} on Y_t .
 - We may not know the delay k , so we may need to estimate it first.
- We simulate a dataset with a built-in delay, and then we model this afterwards.

```
alpha = 0.5; gamma = 1; n = 100; delay = 5
x = as.ts(5*sin(1:(n+delay)/5))
eps = arima.sim(model=list(ar=alpha),n=n+delay)
y = gamma*lag(x,-delay)+eps
dat_lag = ts.intersect(x,y)
ts.plot(dat_lag[,1],dat_lag[,2],col=1:2)
```



1.10 The cross-correlation function

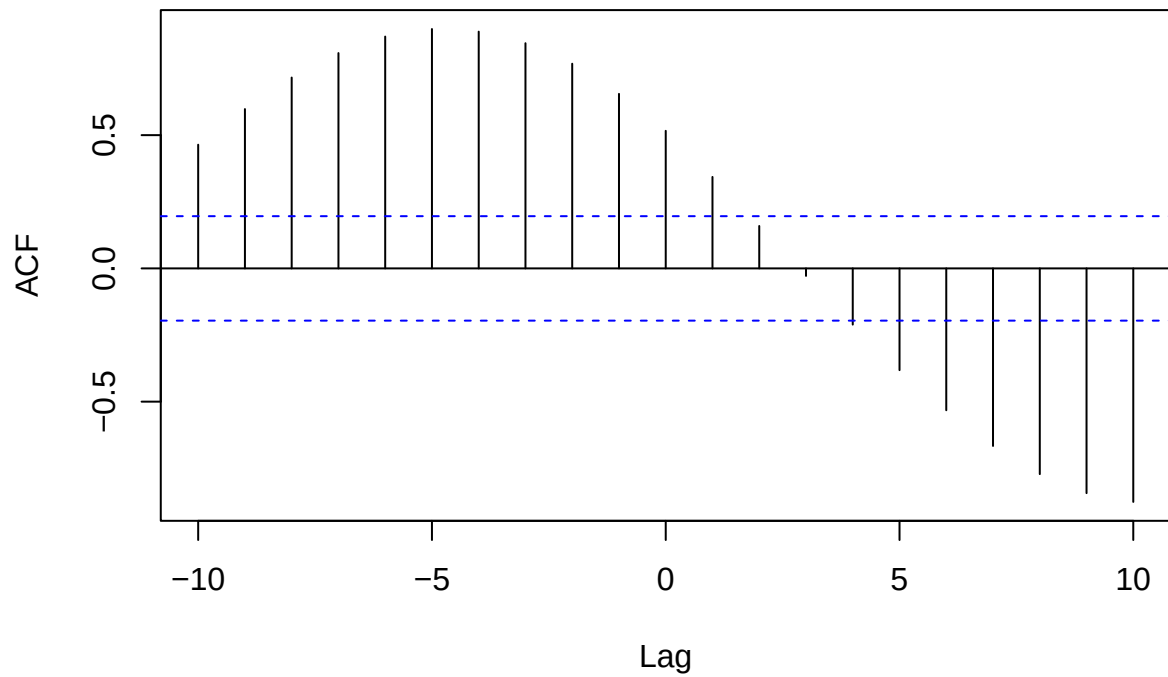
- The **cross correlation function** is used for checking the relation between two time series at different time points:

$$\rho_{xy}(t+k, t) = \text{Cor}(X_{t+k}, Y_t).$$

- Values that are close to 1 or -1 indicate that the two time series are closely related if X_t is delayed by k time steps.
- Cross-correlation function for the simulated data

```
cc = ccf(dat_lag[,1], dat_lag[,2], lag.max=10)
```

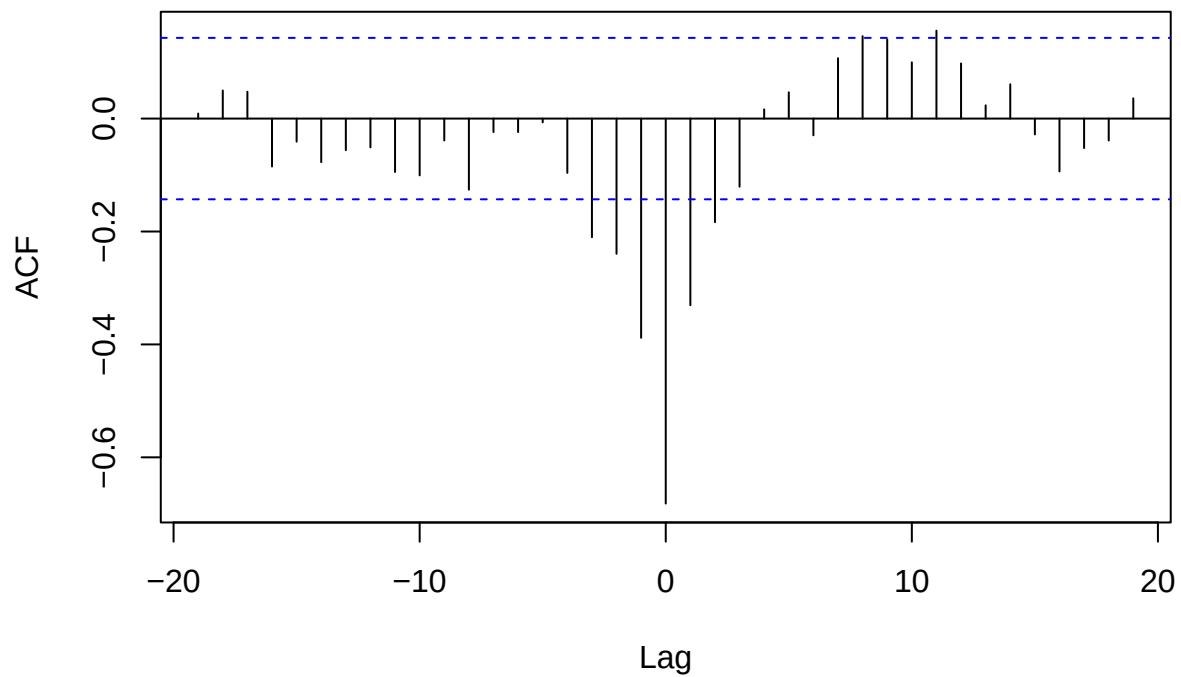
dat_lag[, 1] & dat_lag[, 2]



- Cross-correlation function for the elspot data:

```
ccf(forecast,price)
```

forecast & price



1.11 Fitting models with lag

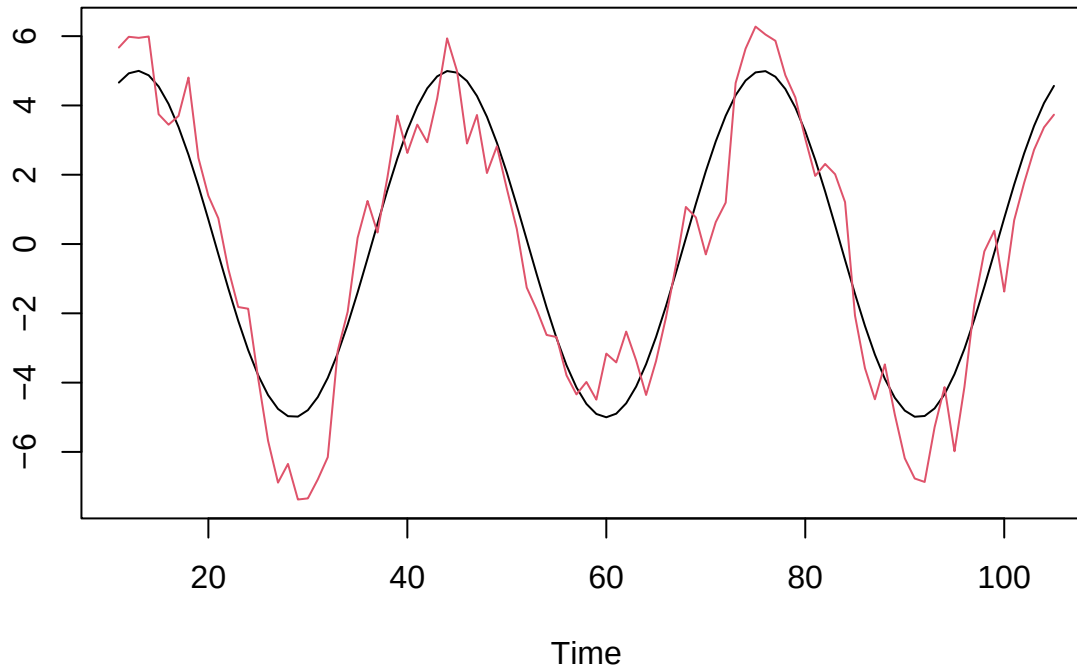
- We estimate the lag to be the one where the cross-correlation function is maximal:

```
estlag = cc$lag[which(cc$acf==max(abs(cc$acf)))]
estlag
```

```
## [1] -5
```

- Plotting the data with this lags can be useful to check the choice:

```
dat_shifted = ts.intersect(lag(as.ts(dat_lag[,1]),estlag),dat_lag[,2] )
ts.plot(dat_shifted[,1],dat_shifted[,2],col=1:2)
```



- We can now fit a model with this lag:

```
mod=arima(dat_shifted[,2],order=c(1,0,0),xreg=dat_shifted[,1]); mod
```

```
##
## Call:
## arima(x = dat_shifted[, 2], order = c(1, 0, 0), xreg = dat_shifted[, 1])
##
## Coefficients:
##          ar1  intercept  dat_shifted[, 1]
##       0.5938   -0.2047         1.0526
## s.e.  0.0820     0.2347         0.0615
##
## sigma^2 estimated as 0.8884:  log likelihood = -129.39,  aic = 266.79
```

1.12 ARMAX models

- An alternative way of including exogenous variables into an ARMA model is an ARMAX model.
- The $ARMAX(p, q, b)$ model is an $ARMA(p, q)$ model including b terms of an exogenous variable, i.e. it

is defined by

$$Y_t = \sum_{i=1}^p \alpha_i Y_{t-i} + \sum_{i=1}^b \gamma_i X_{t-i} + W_t + \sum_{i=1}^q \beta_i W_{t-i}$$

- Using the backshift operator, this can be written as

$$\alpha(B)Y_t = \gamma(B)X_t + \beta(B)W_t$$

with $\alpha(B) = 1 - \sum_{i=1}^p \alpha_i B^i$, $\beta(B) = 1 + \sum_{i=1}^q \beta_i B^i$, and $\gamma(B) = \sum_{i=1}^b \gamma_i B^i$.

- Compare with the regression with ARMA noise:

$$\alpha(B)(Y_t - \gamma_0 - \gamma_1 X_t) = \beta(B)W_t \quad \Rightarrow \quad \alpha(B)Y_t = \alpha(B)(\gamma_0 + \gamma_1 X_t) + \beta(B)W_t$$

- The difference is only how the model includes the exogenous variable.
- It is mostly a matter of taste which kind of model you should choose.
- Only the regression with ARMA noise is included into R as standard.

2 Continuous time processes

2.1 Discrete vs. continuous time

There are two fundamentally different model classes for time series data.

- Discrete time stochastic processes
 - Variables given at equally spaced time points
- Continuous time stochastic processes
 - Variables that evolve over a continuous time scale

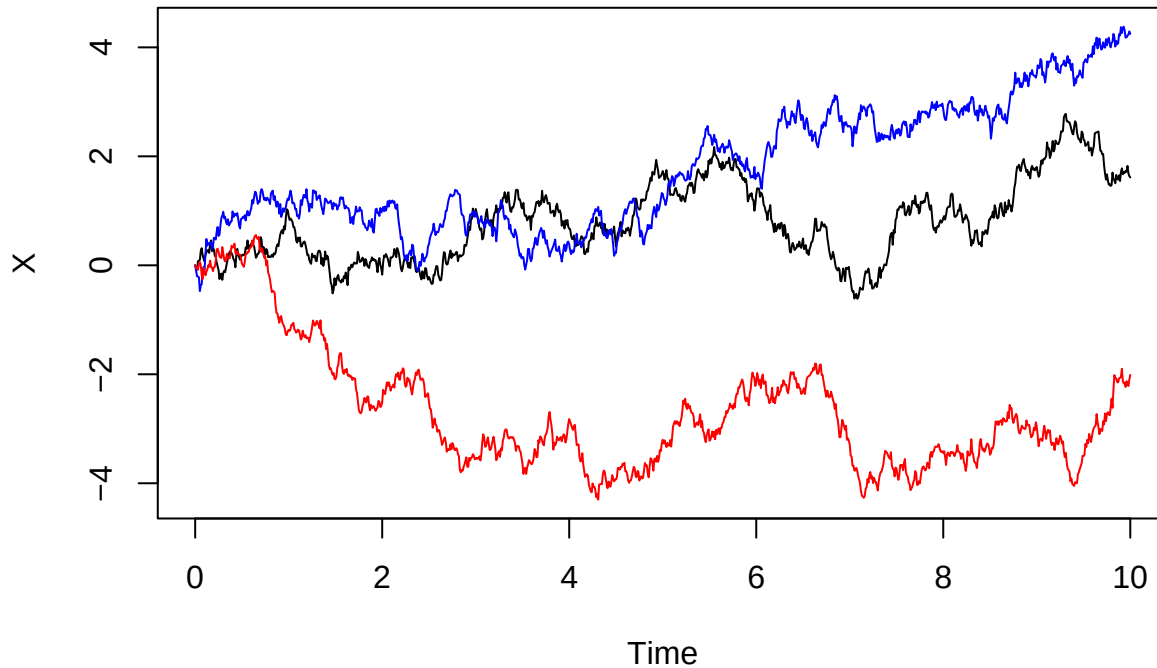
So far we have only looked at the discrete time case. We will finish today's lecture by looking a bit at the continuous time case, just to give you an idea of this topic.

2.2 Continuous time stochastic processes

- In this setup we see the underlying X_t as a continuous function of t for t in some interval $[0, T]$.
 - In principle we imagine that there are infinitely many data points, simply because there are infinitely many time points between 0 and T .
 - In practice we will always only have finitely many data points.
 - But we can imagine that the real data actually contains all the data points. We are just not able to measure them (and to store them in a computer).
 - With a model for all datapoints, we are - through simulation - able to describe the behaviour of data. Also between the observations.
-

2.3 The Wiener process

- A key example of a process in continuous time will be the so-called **Wiener process** or **Brownian motion**.
- Here are three simulated realizations (black, blue and red) of this process: here



- A Wiener process has the following properties:
 - It starts in 0: $B_0 = 0$.
 - It has independent increments: For $0 < s < t$ it holds that $B_t - B_s$ is independent of everything that has happened up to time s , that is B_u for all $u \leq s$.
 - It has normally distributed increments: For $0 < s < t$ it holds that the increment $B_t - B_s$ is normally distributed with mean zero and variance $t - s$:

$$B_t - B_s \sim \text{norm}(0, t - s).$$

- The intuition of this process is that it somehow changes direction all the time: How the process changes after time s will be independent of what has happened before time s . So whether the process should increase or decrease after s will not be affected by how much it was increasing or decreasing before. This gives the very bumpy behaviour over time.

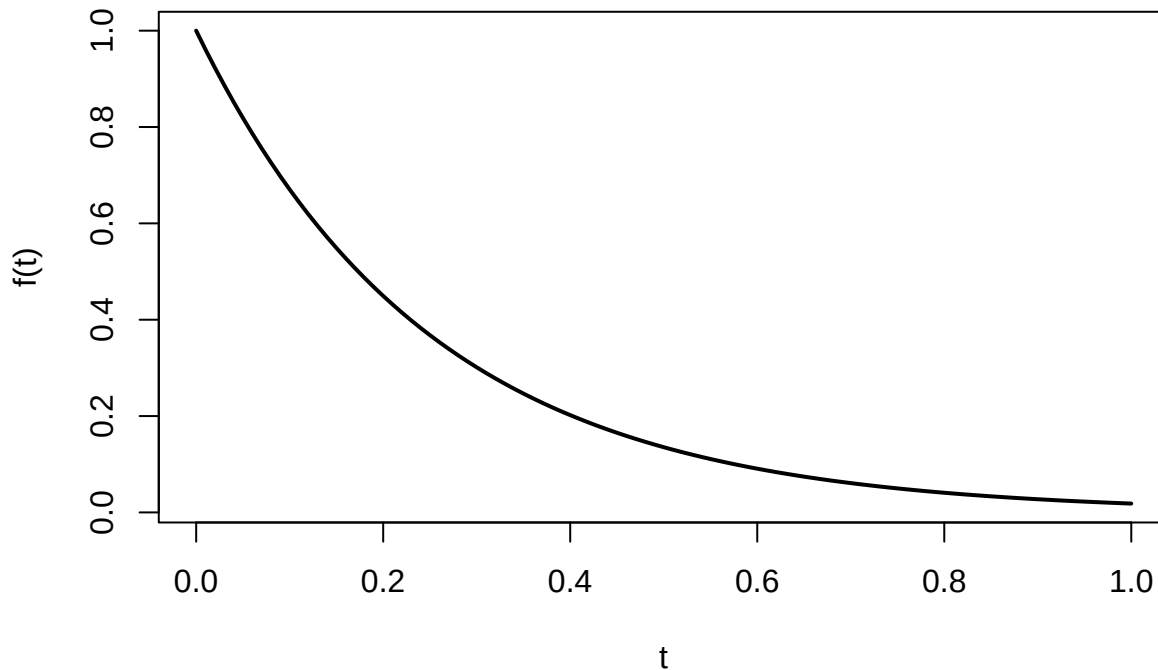
2.4 Stochastic differential equations

- A common way to define a continuous time stochastic process model is through a stochastic differential equation (SDE) which we will turn to shortly, but before doing so we will recall some basic things about ordinary differential equations.
- **Example:** Suppose f is an unknown differentiable function satisfying the differential equation

$$\frac{df(t)}{dt} = -4f(t)$$

with initial condition $f(0) = 1$. This equation has the solution

$$f(t) = \exp(-4t)$$



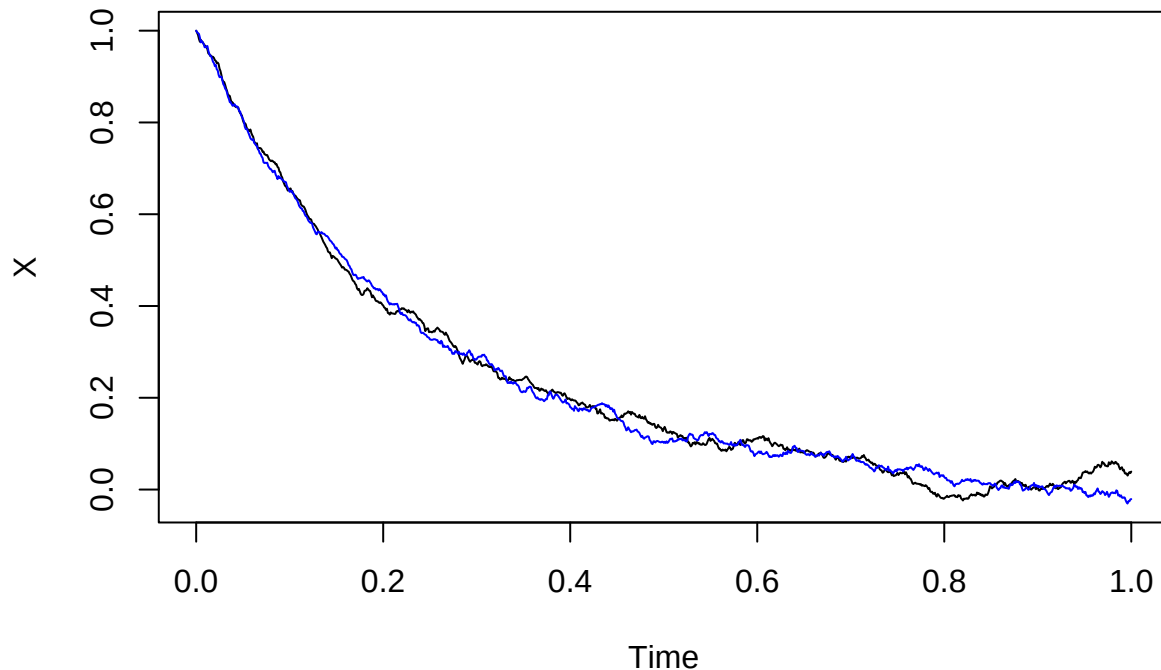
- With a slightly unusual notation we can rewrite this as

$$df(t) = -4 \cdot f(t)dt$$

- This equation has the following (hopefully intuitive) interpretation:
 - When time increases by a small amount dt (from t to $t + dt$) the value of f changes (approximately) by $-4f(t) \cdot dt$.
- So when t is increased, then f is decreased, and the decrease is proportional to the value of $f(t)$. That is why f decreases slower and slower, when t is increased.
- We say that the function has a **drift** towards zero, and this drift is determined by the value of the function.

2.5 Stochastic differential equations

- It will probably never be true that data behaves exactly like the exponentially decreasing curve on the previous slide.
- Instead we will consider a model, where some random noise from a Wiener process has been added to the growth rate. Two different (black/blue) simulated realizations can be seen below



- The type of process that is simulated above is described formally by the equation

$$dX_t = -4X_t dt + 0.1dB_t$$

- This is called a **Stochastic Differential Equation** (SDE), and the processes simulated above are called solutions of the stochastic differential equation.
- The SDE $dX_t = -4X_t dt + 0.1dB_t$ has two terms:
 - $-4X_t dt$ is the **drift term**.
 - $0.1dB_t$ is the **diffusion term**.
- The intuition behind this notation is very similar to the intuition in the equation $df(t) = -4 \cdot f(t) dt$ for an ordinary differential equation. When the time is increased by the small amount dt , then the process X_t is increased by $-4X_t dt$ AND by how much the process $0.1B_t$ has increased on the time interval $[t, t + dt]$.
- So this process has a **drift** towards zero, but it is also pushed in a random direction (either up or down) by the Wiener process (more precisely, the process $0.1B_t$)