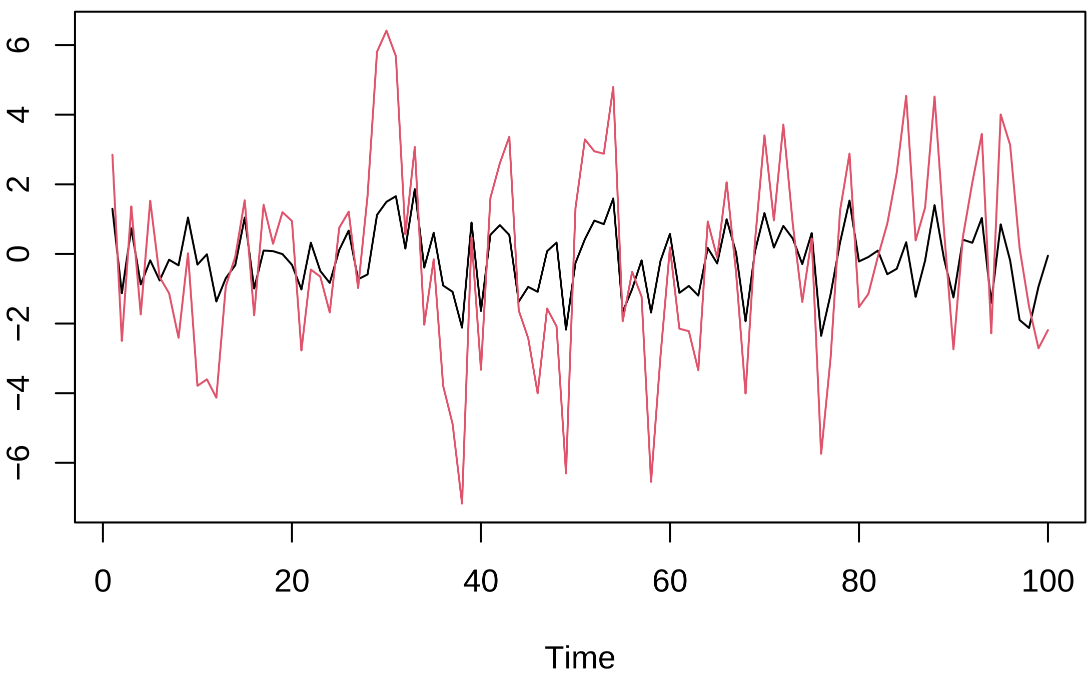
linear regression with ARMA noise

Linear regression with ARMA noise

Simulate data from a linear regression with ARMA noise in the following way: - Create an exogenous variable x_t - either simulate this randomly, or pick deterministic values for example using some function of t - Simulate the noise terms ϵ_t using an ARMA(1,1) model with parameters $\alpha_1 = 0.5$ and $\beta_1 = 0.5$. Is this process stationary? - Calculate y_t from x_t and ϵ_t using the model $y_t = 2x_t + \epsilon_t$.

Simulated data:

```
alpha = 0.5; beta = 0.5; gamma0=0;gamma1 = 2; n = 100
x = rnorm(n)
eps = arima.sim(model=list(ar=alpha,ma=beta),n=n)
y = gamma0 +gamma1*x+eps
ts.plot(x,y,col=1:2)
```



Next we fit a regression model to the simulated data

- Fit a regression model with ARMA(p,q) noise to the simulated data for various p and q. Compare the models using AIC. Which p and q give the best fit?

Fitting true model, with ARMA(1,1) noise:

```
mod11=arima(y,order=c(1,0,1),xreg=x); mod11
```

```
##
## Call:
## arima(x = y, order = c(1, 0, 1), xreg = x)
##
```

```
## Coefficients:
##          ar1     ma1  intercept       x
##       0.5526  0.5783     0.1963  2.1496
## s.e.  0.0929  0.0802     0.3366  0.0608
##
## sigma^2 estimated as 0.9384:  log likelihood = -139.38,  aic = 288.76
```

Fitting model with AR(1) noise:

```
mod10=arima(y,order=c(1,0,0),xreg=x); mod10
```

```
##
## Call:
## arima(x = y, order = c(1, 0, 0), xreg = x)
##
## Coefficients:
##          ar1  intercept       x
##       0.7370     0.1655  2.1650
## s.e.  0.0672     0.4101  0.0958
##
## sigma^2 estimated as 1.223:  log likelihood = -152.37,  aic = 312.74
```

Fitting model with MA(1) noise:

```
mod01=arima(y,order=c(0,0,1),xreg=x); mod01
```

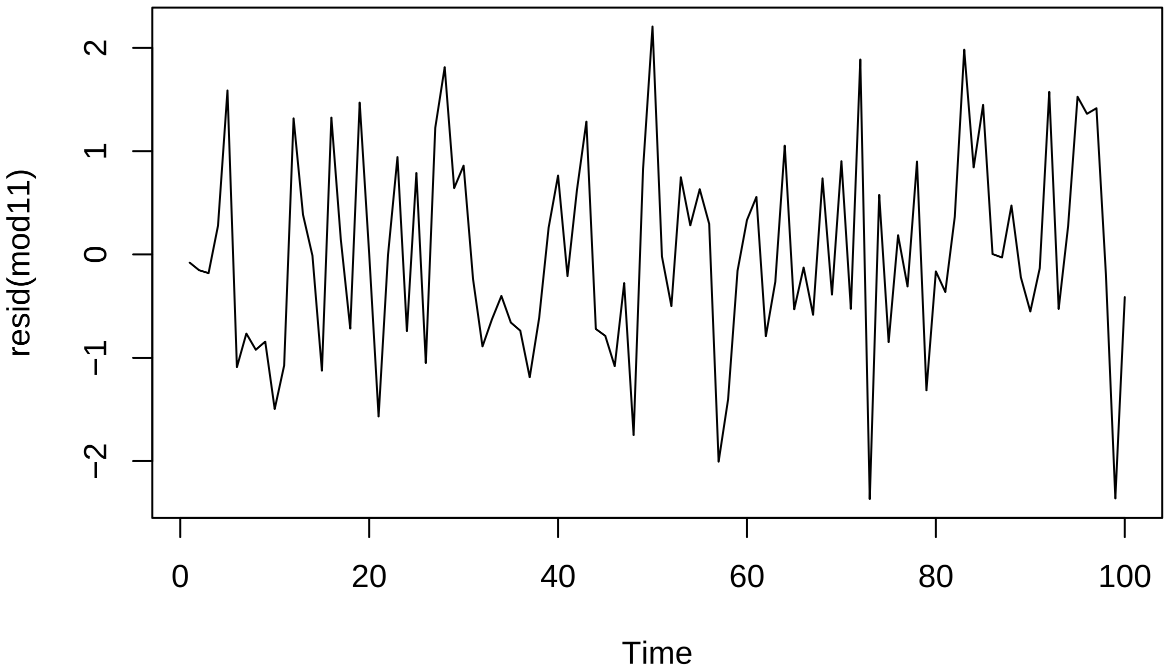
```
##
## Call:
## arima(x = y, order = c(0, 0, 1), xreg = x)
##
## Coefficients:
##          ma1  intercept       x
##       0.7829      0.223  2.1512
## s.e.  0.0472      0.196  0.0688
##
## sigma^2 estimated as 1.216:  log likelihood = -152.14,  aic = 312.28
```

- For the ARMA(1,1) model: estimate the parameters and compare them to the true values used for simulation. Check that the model is reasonable by checking that the residuals look like white noise.

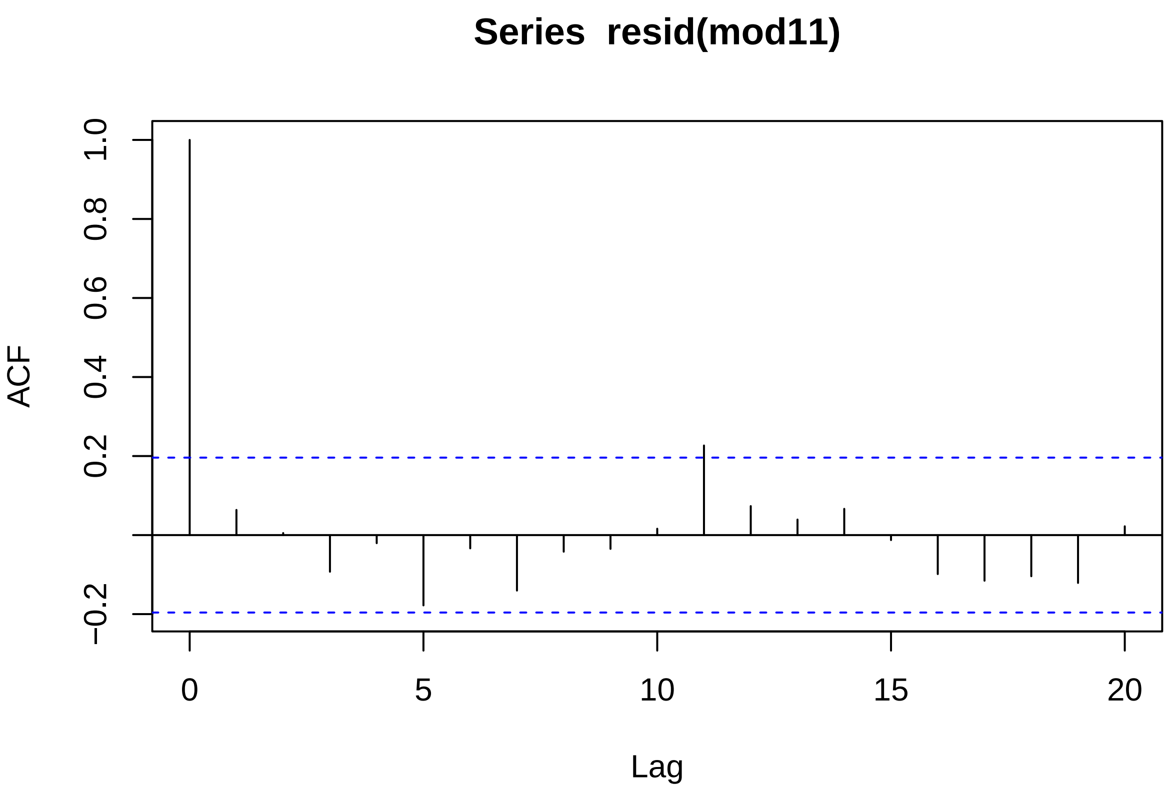
```
mod11
```

```
##
## Call:
## arima(x = y, order = c(1, 0, 1), xreg = x)
##
## Coefficients:
##          ar1     ma1  intercept       x
##       0.5526  0.5783     0.1963  2.1496
## s.e.  0.0929  0.0802     0.3366  0.0608
##
## sigma^2 estimated as 0.9384:  log likelihood = -139.38,  aic = 288.76
```

```r
plot(resid(mod11))
```



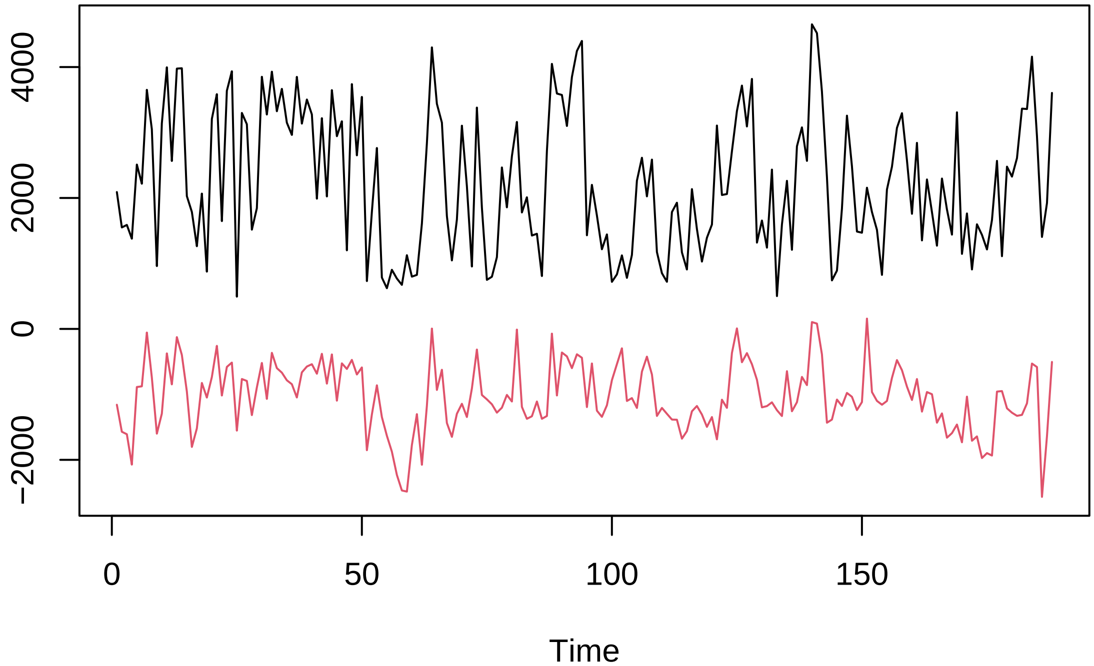

```r
acf(resid(mod11))
```



Data example

Recall the elspot dataset from the lectures.

```r
elspot<-read.csv("https://asta.math.aau.dk/eng/static/datasets?file=elspot.csv", header = TRUE)
forecast<-ts(elspot[,2])
price<-ts(elspot[,3])
ts.plot(forecast,-price,col=1:2) # Price is negative to make the series positively correlated
```



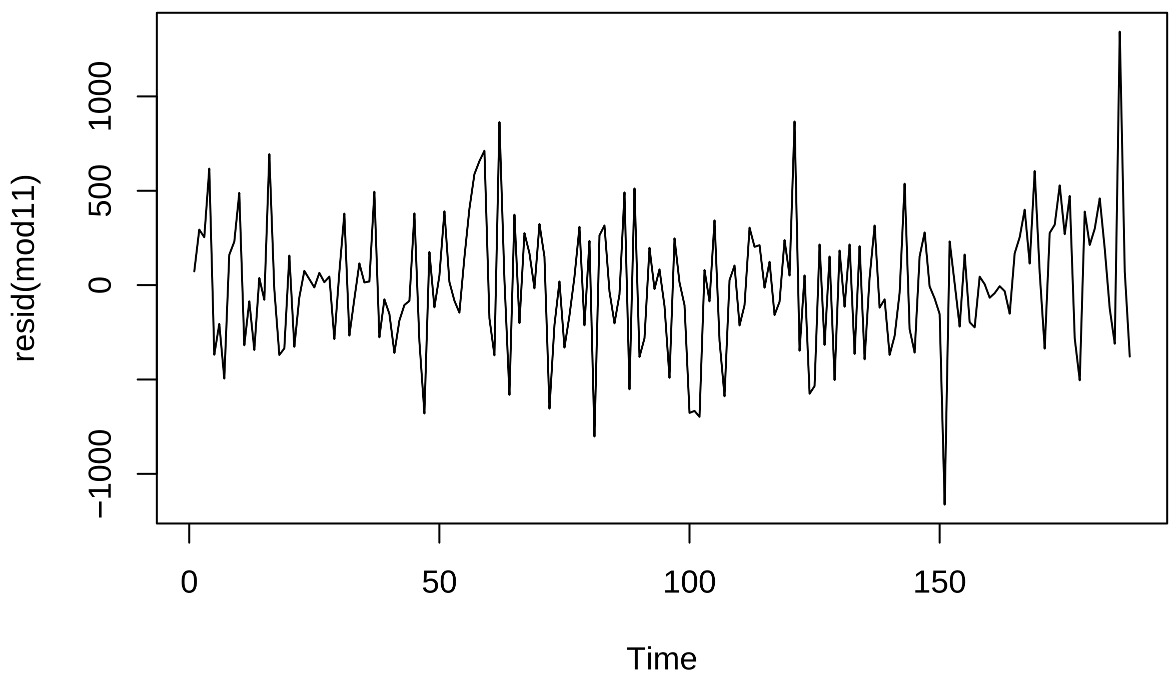
In the lectures we fitted a regression model with AR(1) noise to the data.

- Try to fit various ARMA(p,q) models to the data with various p and q. Compare using AIC and residual plots.

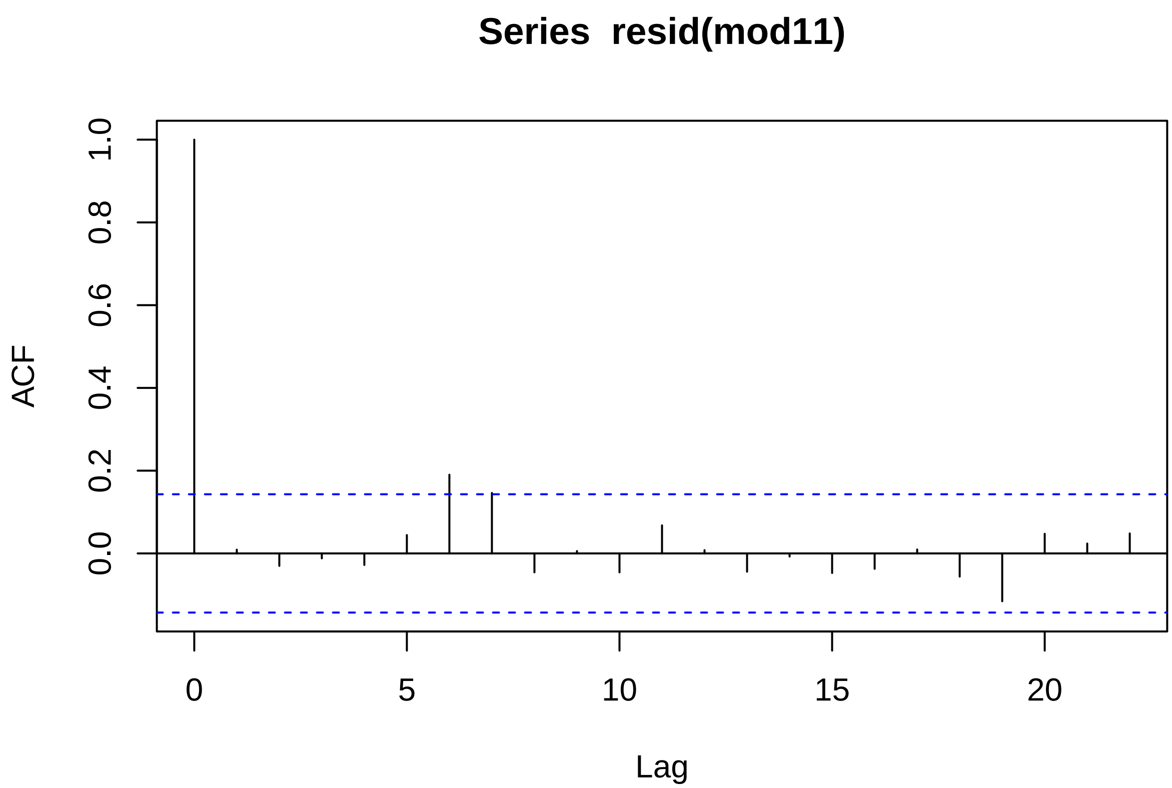
```r
mod11=arima(price,order=c(1,0,1),xreg=forecast); mod11
```

```
##
## Call:
## arima(x = price, order = c(1, 0, 1), xreg = forecast)
##
## Coefficients:
##          ar1      ma1  intercept  forecast
##       0.5634  -0.2112  1721.8864   -0.3076
## s.e.  0.1989   0.2464    76.2764    0.0274
##
## sigma^2 estimated as 116940:  log likelihood = -1363.77,  aic = 2737.55
```

```r
plot(resid(mod11))
```




```
acf(resid(mod11))
```

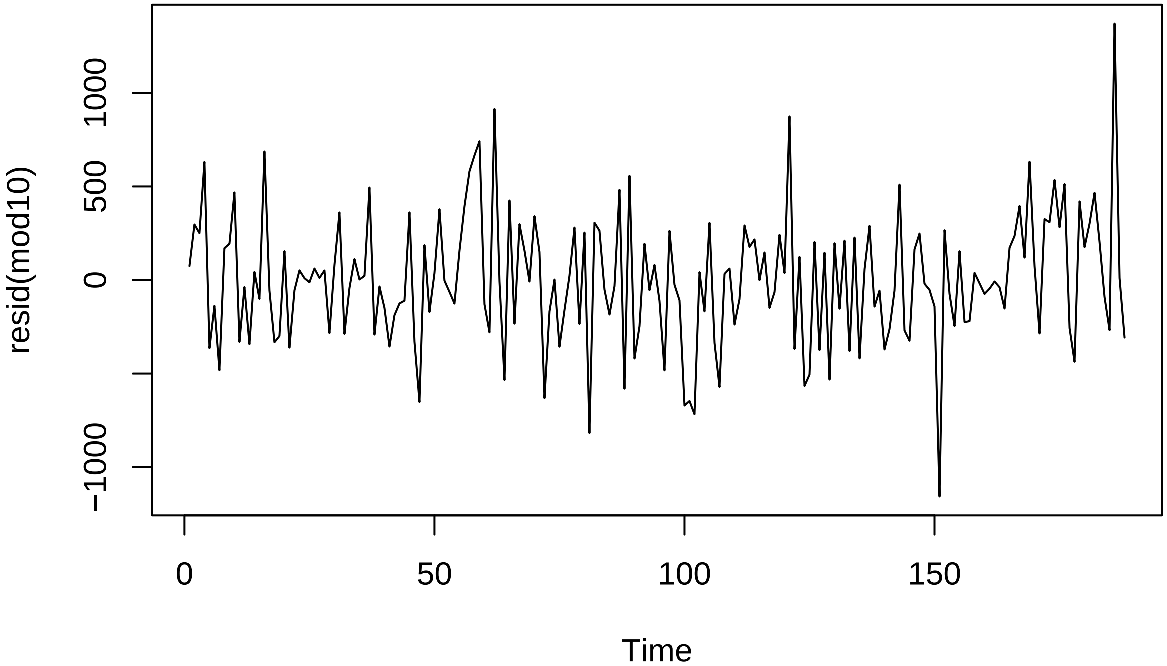



```
mod10=arima(price,order=c(1,0,0),xreg=forecast); mod10
```

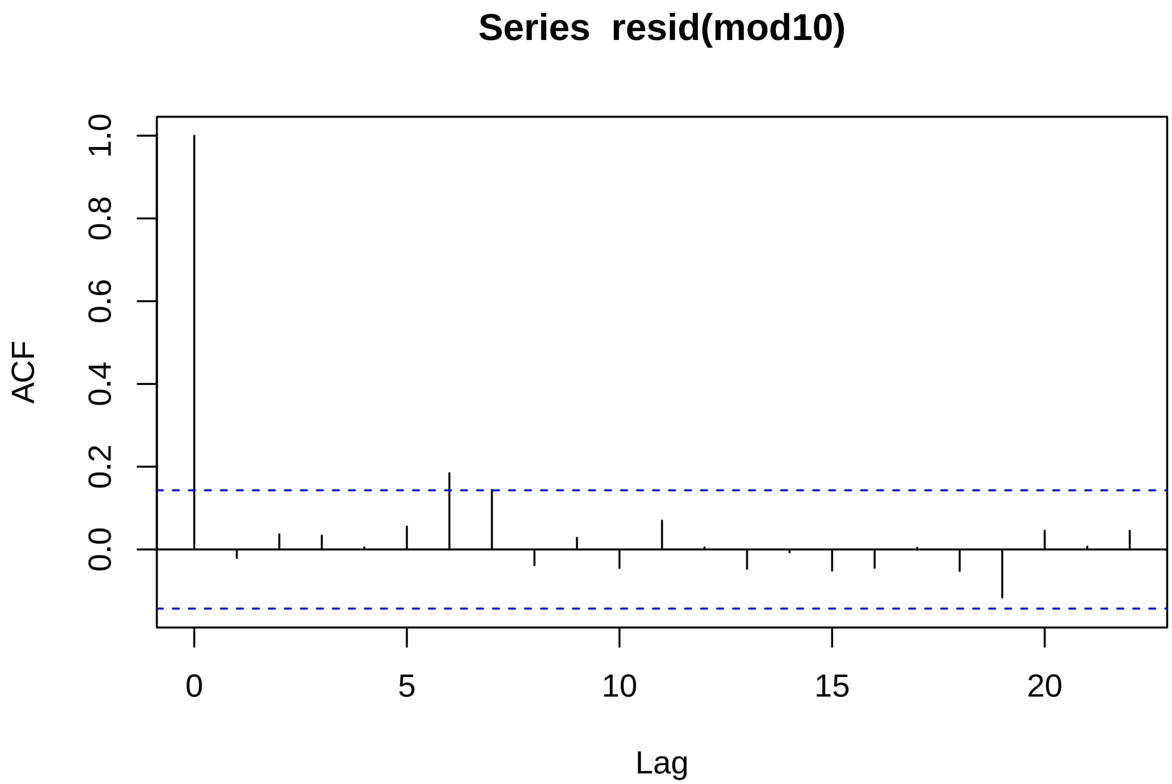
```
##
## Call:
## arima(x = price, order = c(1, 0, 0), xreg = forecast)
##
```

```
## 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
```

```r
plot(resid(mod10))
```



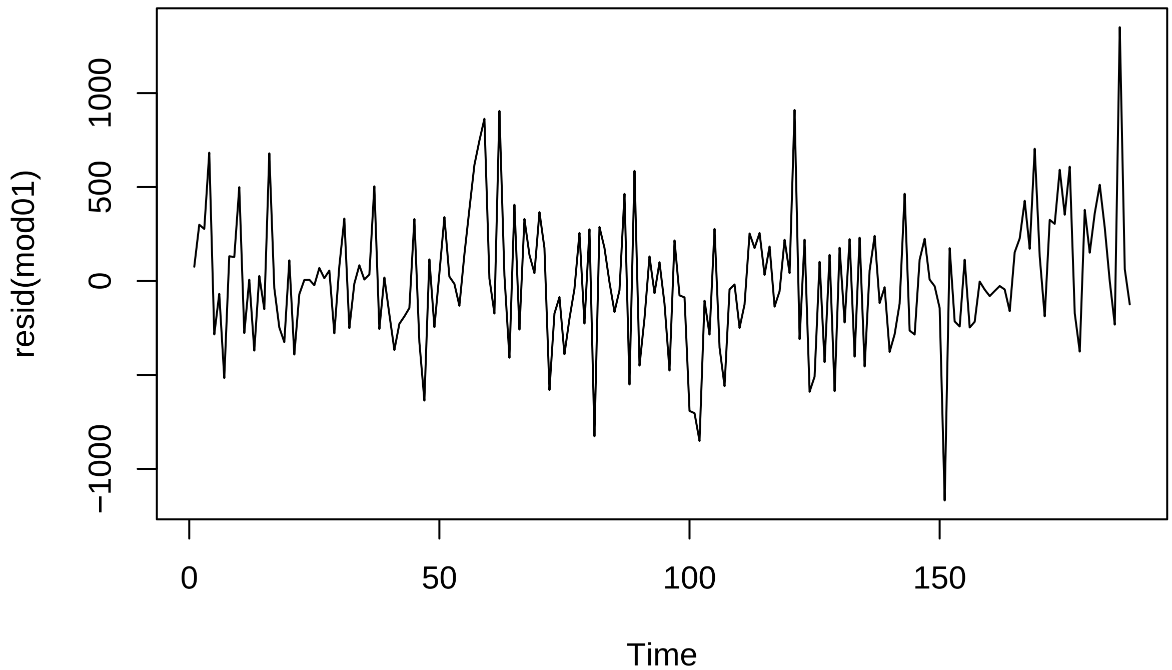

```r
acf(resid(mod10))
```



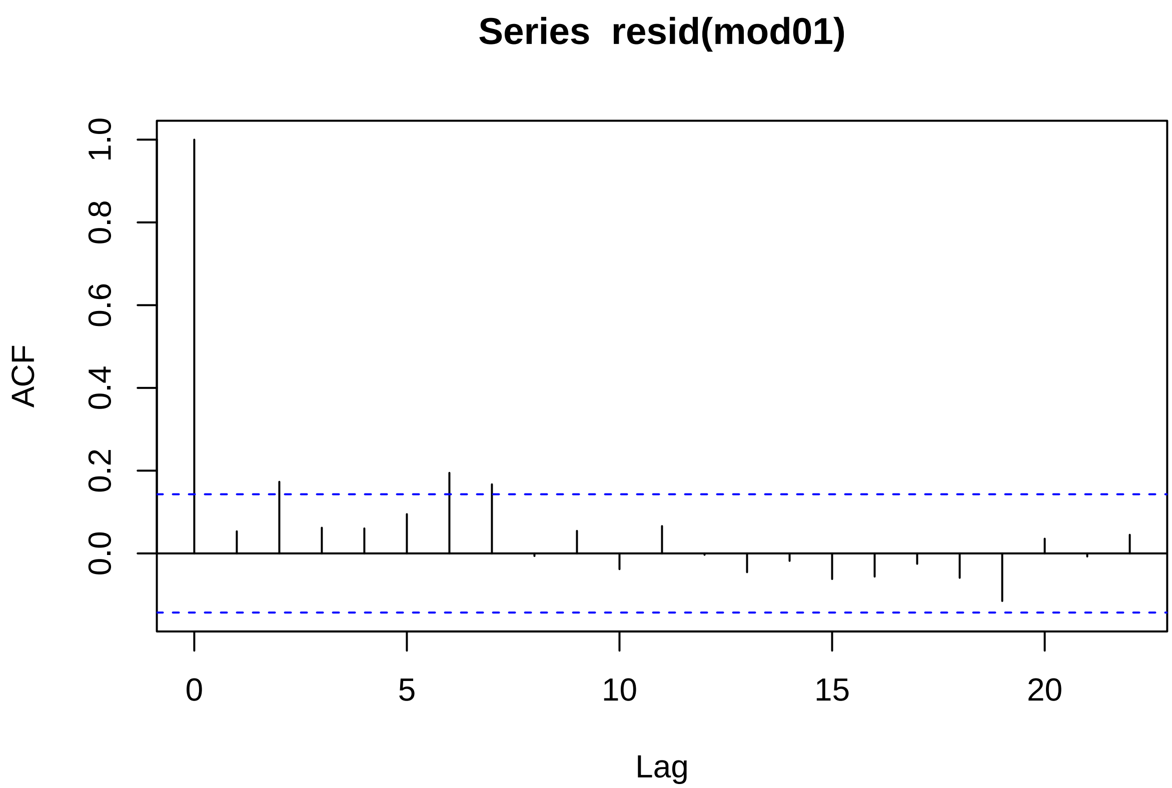

```
mod01=arima(price,order=c(0,0,1),xreg=forecast); mod01
```

```
##
## Call:
## arima(x = price, order = c(0, 0, 1), xreg = forecast)
##
## Coefficients:
##          ma1  intercept  forecast
##       0.3277  1726.3804   -0.3101
## s.e.  0.0623    69.0776    0.0269
##
## sigma^2 estimated as 121126:  log likelihood = -1367.05,  aic = 2742.1
```

```
plot(resid(mod01))
```




```
acf(resid(mod01))
```



Correlograms look similar, AR(1) model looks best using AIC

- The forecasted wind and solar power production for day n+1 is 3000. Predict the elspot price on day n+1 and make a 95% confidence interval for this price.

```
pred<-predict(mod10,n.ahead=1,newxreg=3000)$pred
pred
```

```
## Time Series:
## Start = 189
## End = 189
## Frequency = 1
## [1] 756.8968
```

```
se<-predict(mod10,n.ahead=1,newxreg=3000)$se
upper<- pred + 2*se
lower<-pred-2*se
upper
```

```
## Time Series:
## Start = 189
## End = 189
## Frequency = 1
##    pred
## 1442.42
```

```
lower
```

```
## Time Series:
## Start = 189
## End = 189
## Frequency = 1
##    pred
## 71.3733
```

The prediction is 756.9, the confidence interval is [71.3, 1442.42] (it seems that the prediction is quite uncertain).