

# Forecasting with ARIMA models

FISH 550 – Applied Time Series Analysis   Download Rmd

Eli Holmes

6 Apr 2023

# Forecasting with an ARIMA model

The basic idea of forecasting with an ARIMA model to estimate the parameters and forecast forward.

For example, let's say we want to forecast with this ARIMA(2,1,0) model:

$$y_t = \mu + \beta_1 y_{t-1} + \beta_2 y_{t-2} + e_t$$

where  $y_t = x_t - x_{t-1}$ , the first difference.

`Arima()` would write this model:

$$(y_t - m) = \beta_1(y_{t-1} - m) + \beta_2(y_{t-2} - m) + e_t$$

The relationship between  $\mu$  and  $m$  is  $\mu = m(1 - \beta_1 - \beta_2)$ .

Let's estimate the  $\beta$ 's for this model from the anchovy data.

```
fit <- forecast::Arima(anchovyts, order=c(2,1,0), include.drift=TRUE)
coef(fit)
```

```
##           ar1           ar2           drift
## -0.53850433 -0.44732522  0.05367062
```

```
mu <- coef(fit)[3]*(1-coef(fit)[1]-coef(fit)[2])
mu
```

```
##           drift
## 0.1065807
```

So we will forecast with this model:

$$y_t = 0.1065807 - 0.53850433y_{t-1} - 0.44732522y_{t-2} + e_t$$

To get our forecast for 1990, we do this

$$(x_{90} - x_{89}) = 0.106 - 0.538(x_{89} - x_{88}) - 0.447(x_{88} - x_{87})$$

Thus

$$x_{90} = x_{89} + 0.106 - 0.538(x_{89} - x_{88}) - 0.447(x_{88} - x_{87})$$

Here is R code to do that:

```
anchovyts[26]+mu+coef(fit)[1]*(anchovyts[26]-anchovyts[25])  
  coef(fit)[2]*(anchovyts[25]-anchovyts[24])
```

```
##      drift  
## 9.962083
```

## Forecasting with forecast()

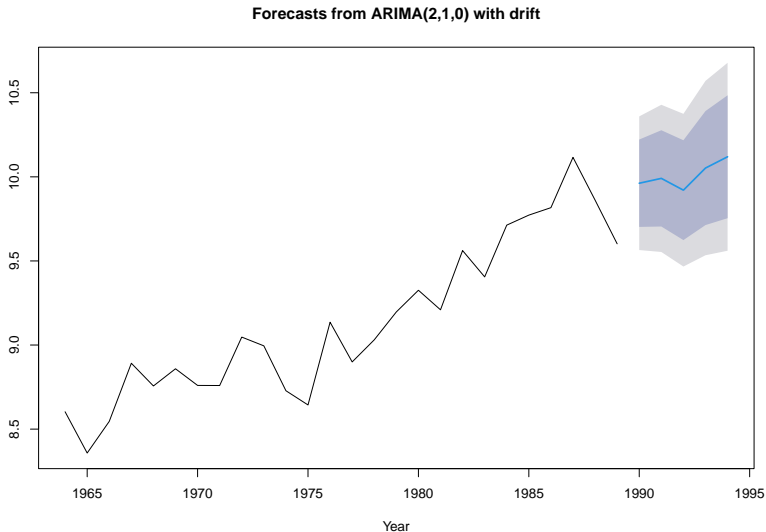
`forecast(fit, h=h)` automates the forecast calculations for us and computes the upper and lower prediction intervals. Prediction intervals include uncertainty in parameter estimates plus the process error uncertainty.

```
fr <- forecast::forecast(fit, h=5)
fr
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 1990	9.962083	9.702309	10.22186	9.564793	10.35937
## 1991	9.990922	9.704819	10.27703	9.553365	10.42848
## 1992	9.920798	9.623984	10.21761	9.466861	10.37473
## 1993	10.052240	9.713327	10.39115	9.533917	10.57056
## 1994	10.119407	9.754101	10.48471	9.560719	10.67809

# Plotting our forecasts

```
plot(fr, xlab="Year")
```



## Missing values

Missing values are allowed for `Arima()` and `arima()`. We can produce forecasts with the same code.

```
anchovy.miss <- anchovyts
anchovy.miss[10:11] <- NA
anchovy.miss[20:21] <- NA
fit <- forecast::auto.arima(anchovy.miss)
fr <- forecast::forecast(fit, h=5)
fr
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 1990	9.922074	9.654009	10.19014	9.512103	10.33205
## 1991	9.976827	9.698782	10.25487	9.551595	10.40206
## 1992	10.031579	9.743902	10.31926	9.591615	10.47154
## 1993	10.086332	9.789334	10.38333	9.632113	10.54055
## 1994	10.141084	9.835049	10.44712	9.673044	10.60912



```
plot(fr)
```

Forecasts from ARIMA(0,1,1) with drift



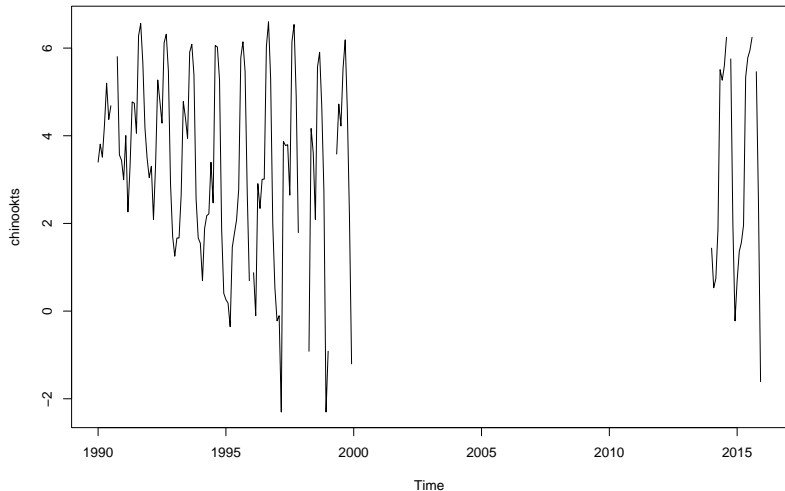
# Forecasting with a Seasonal model

Load the chinook salmon data

```
load("chinook.RData")  
chinookts <- ts(chinook$log.metric.tons, start=c(1990,1),  
               frequency=12)
```

## Plot seasonal data

```
plot(chinookts)
```



## Seasonal ARIMA model

Seasonally differenced data, e.g. chinook data January 1990 - chinook data January 1989.

$$z_t = x_t - x_{t+s} - m$$

Basic structure of a seasonal AR model

$$z_t = \text{AR}(p) + \text{AR}(\text{season}) + \text{AR}(p+\text{season})$$

$$\text{e.g. } z_t = \text{AR}(1) + \text{AR}(12) + \text{AR}(1+12)$$

Example AR(1) non-seasonal part + AR(1) seasonal part

$$z_t = \phi_1 z_{t-1} + \Phi_1 z_{t-12} - \phi_1 \Phi_1 z_{t-13}$$

# Notation

ARIMA  $(p,d,q)(ps,ds,qs)S$

ARIMA  $(1,0,0)(1,1,0)[12]$

## auto.arima() for seasonal ts

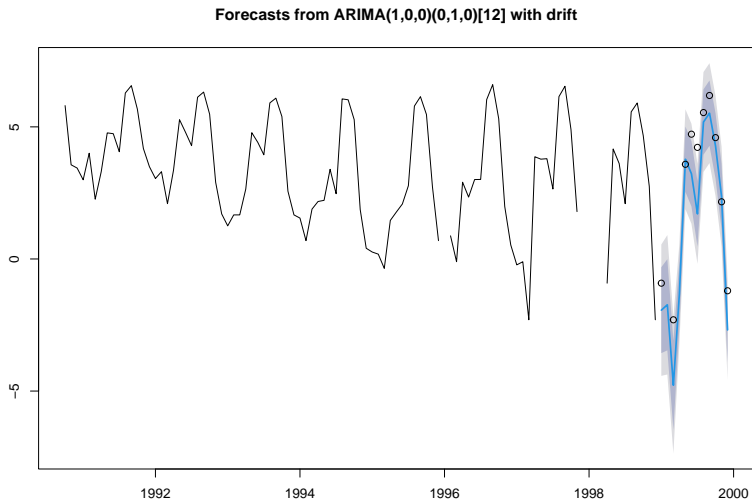
auto.arima() will recognize that our data has season and fit a seasonal ARIMA model to our data by default. We will define the training data up to 1998 and use 1999 as the test data.

```
traindat <- window(chinookts, c(1990,10), c(1998,12))
testdat <- window(chinookts, c(1999,1), c(1999,12))
fit <- forecast::auto.arima(traindat)
fit
```

```
## Series: traindat
## ARIMA(1,0,0)(0,1,0)[12] with drift
##
## Coefficients:
##          ar1      drift
##      0.3676  -0.0320
## s.e.  0.1335   0.0127
##
## sigma^2 = 0.8053:  log likelihood = -107.37
## AIC=220.73   AICc=221.02   BIC=228.13
```

## Forecast using seasonal model

```
fr <- forecast::forecast(fit, h=12)  
plot(fr)  
points(testdat)
```



# Missing values

Missing values are ok when fitting a seasonal ARIMA model

