8.10 ARIMA vs ETS

It is a commonly held myth that ARIMA models are more general than exponential smoothing. While linear exponential smoothing models are all special cases of ARIMA models, the non-linear exponential smoothing models have no equivalent ARIMA counterparts. On the other hand, there are also many ARIMA models that have no exponential smoothing counterparts. In particular, all ETS models are non-stationary, while some ARIMA models are stationary.

The ETS models with seasonality or non-damped trend or both have two unit roots (i.e., they need two levels of differencing to make them stationary). All other ETS models have one unit root (they need one level of differencing to make them stationary).

Table 8.3 gives the equivalence relationships for the two classes of models. For the seasonal models, the ARIMA parameters have a large number of restrictions.

Table 8.3: Equivalence relationships between ETS and ARIMA models.

ETS model	ARIMA model	Parameters
ETS(A,N,N)	ARIMA(0,1,1)	$\theta_1 = \alpha - 1$
ETS(A,A,N)	ARIMA(0,2,2)	$\theta_1 = \alpha + \beta - 2$ $\theta_2 = 1 - \alpha$
ETS(A,A_d,N)	ARIMA(1,1,2)	$\phi_1 = \phi$ $\theta_1 = \alpha + \phi\beta - 1 - \phi$ $\theta_2 = (1 - \alpha)\phi$
ETS(A,N,A)	ARIMA(0,1,m)(0,1,0)$_m$	
ETS(A,A,A)	ARIMA(0,1,$m+1$)(0,1,0)$_m$	
ETS(A,A_d,A)	ARIMA(0,1,$m+1$)(0,1,0)$_m$	

The AICc is useful for selecting between models in the same class. For example, we can use it to select an ARIMA model between candidate ARIMA models[17] or an ETS model between candidate ETS models. However, it cannot be used to compare between ETS and ARIMA models because they are in different model classes, and the likelihood is computed in different ways. The examples below demonstrate selecting between these classes of models.

Example: Comparing `auto.arima()` and `ets()` on non-seasonal data

We can use time series cross-validation to compare an ARIMA model and an ETS model. The code below provides functions that return forecast objects from `auto.arima()` and `ets()` respectively.

```
fets <- function(x, h) {
  forecast(ets(x), h = h)
}
farima <- function(x, h) {
  forecast(auto.arima(x), h=h)
}
```

The returned objects can then be passed into `tsCV()`. Let's consider ARIMA models and ETS models for the `air` data as introduced in Section 7.2 where, `air <- window(ausair, start=1990)`.

```
# Compute CV errors for ETS as e1
e1 <- tsCV(air, fets, h=1)
# Compute CV errors for ARIMA as e2
e2 <- tsCV(air, farima, h=1)
# Find MSE of each model class
mean(e1^2, na.rm=TRUE)
#> [1] 7.864
mean(e2^2, na.rm=TRUE)
#> [1] 9.622
```

In this case the ets model has a lower tsCV statistic based on MSEs. Below we generate and plot forecasts for the next 5 years generated from an ETS model.

```
air %>% ets() %>% forecast() %>% autoplot()
```

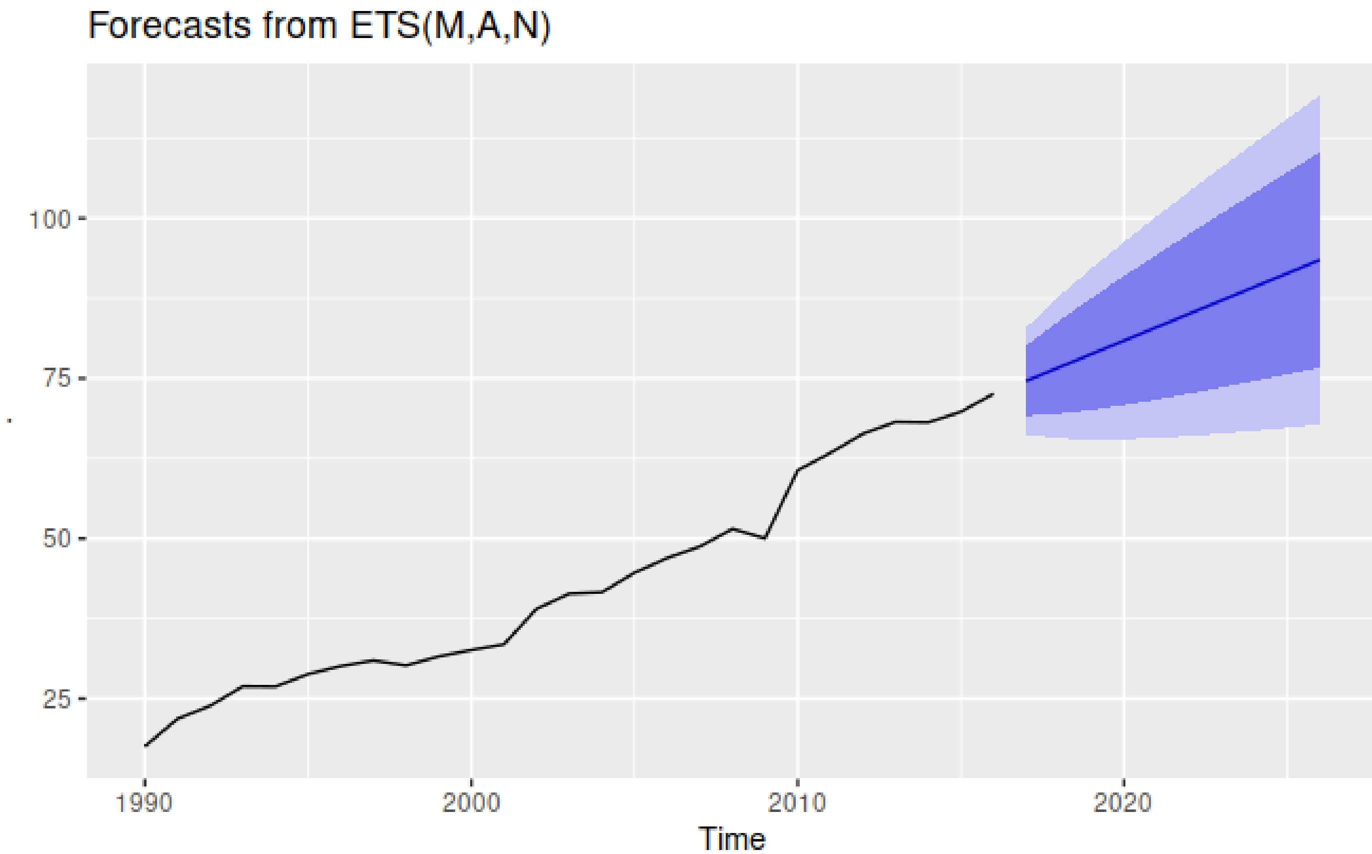

Figure 8.27: Forecasts from an ETS model fitted to monthly totals of air transport passengers in Australia.

Example: Comparing `auto.arima()` and `ets()` on seasonal data

In this case we want to compare seasonal ARIMA and ETS models applied to the quarterly cement production data `qcement`. Because the series is relatively long, we can afford to use a training and a test set rather than time series cross-validation. The advantage is that this is much faster. We create a training set from the beginning of 1988 to the end of 2007 and select an ARIMA and an ETS model using the `auto.arima()` and `ets()` functions.

```
# Consider the qcement data beginning in 1988
cement <- window(qcement, start=1988)
# Use 20 years of the data as the training set
train <- window(cement, end=c(2007,4))
```

The output below shows the ARIMA model selected and estimated by `auto.arima()`. The ARIMA model does well in capturing all the dynamics in the data as the residuals seem to be white noise.

```
(fit.arima <- auto.arima(train))
```

```
#> Series: train
#> ARIMA(1,0,1)(2,1,1)[4] with drift
#>
#> Coefficients:
#>          ar1     ma1    sar1     sar2     sma1   drift
#>        0.889  -0.237   0.081   -0.235   -0.898   0.010
#> s.e.   0.084   0.133   0.157    0.139    0.178   0.003
#>
#> sigma^2 estimated as 0.0115:  log likelihood=61.47
#> AIC=-109   AICc=-107.3   BIC=-92.63
checkresiduals(fit.arima)
```

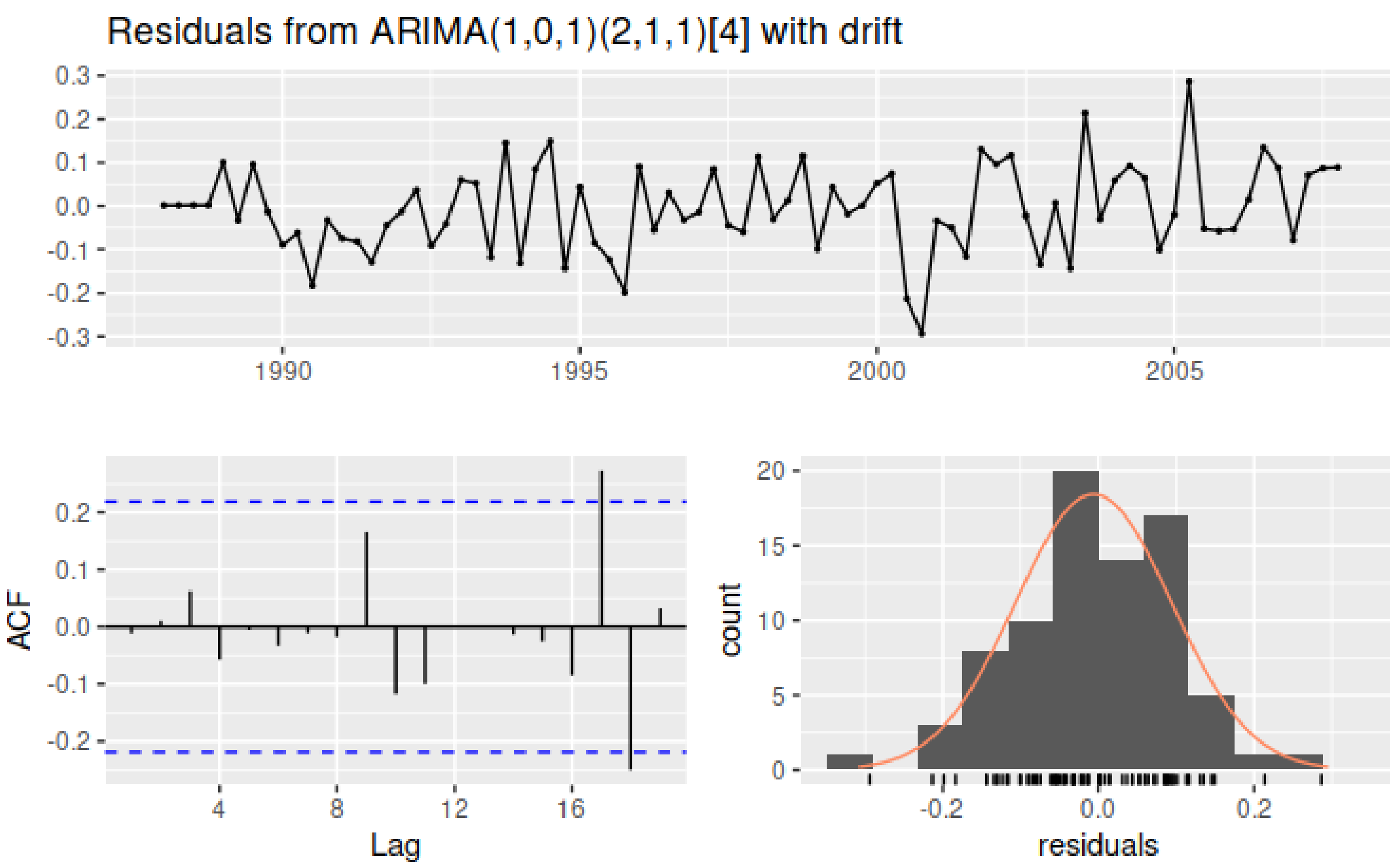

Figure 8.28: Residual diagnostic plots for the ARIMA model fitted to the quarterly cement production training data.

```
#>
#>  Ljung-Box test
#>
#> data:  Residuals from ARIMA(1,0,1)(2,1,1)[4] with drift
#> Q* = 3.3, df = 3, p-value = 0.3
#>
#> Model df: 6.   Total lags used: 9
```

The output below also shows the ETS model selected and estimated by `ets()`. This model also does well in capturing all the dynamics in the data, as the residuals similarly appear to be white noise.

```
(fit.ets <- ets(train))
```

```
#> ETS(M,N,M)
#>
#> Call:
#>  ets(y = train)
#>
#>   Smoothing parameters:
#>     alpha = 0.7341
#>     gamma = 1e-04
#>
#>   Initial states:
#>     l = 1.6439
#>     s = 1.031 1.044 1.01 0.9148
#>
#>   sigma:  0.0581
#>
#>     AIC    AICc     BIC
#> -2.1967 -0.6411 14.4775
checkresiduals(fit.ets)
```

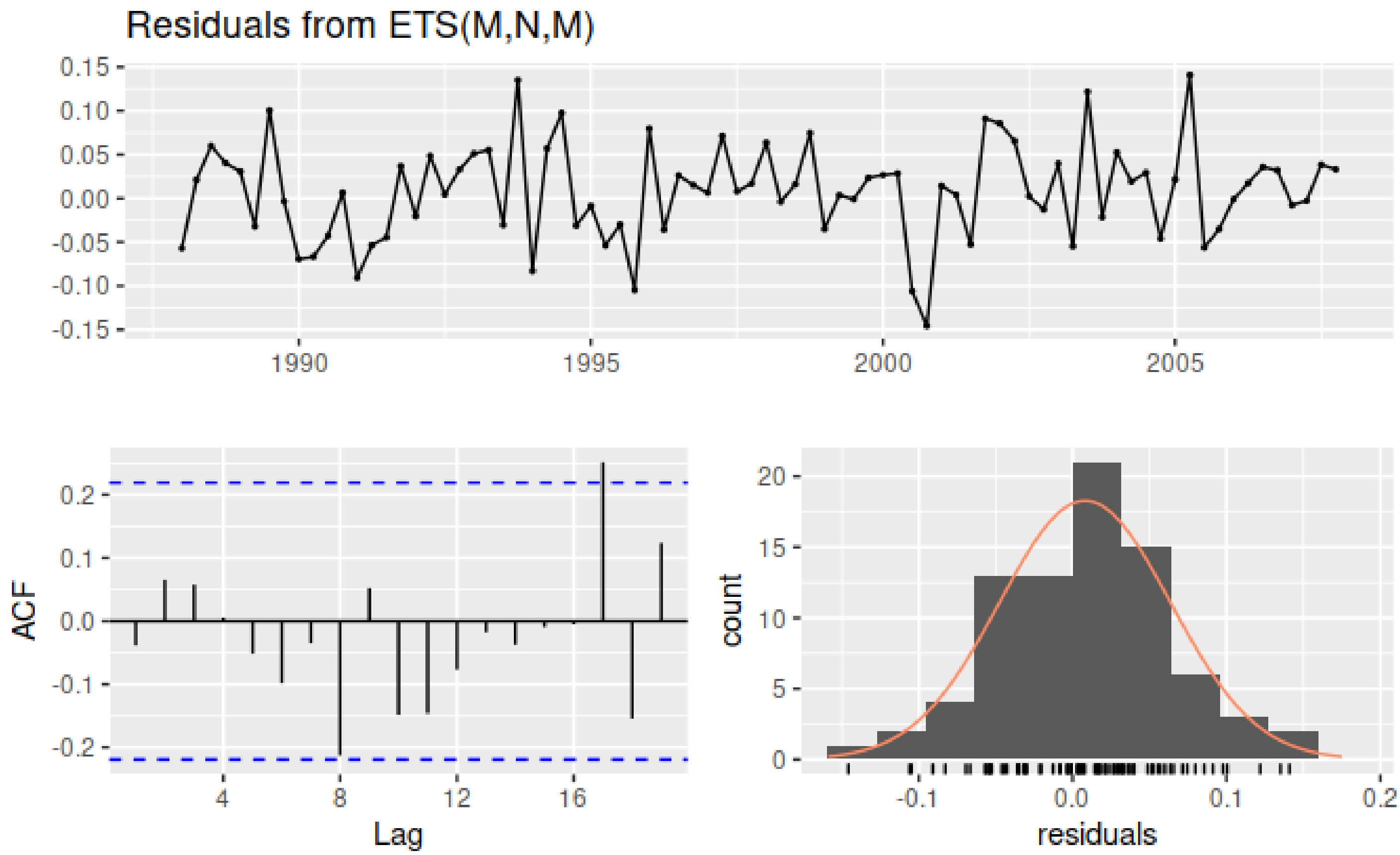

Figure 8.29: Residual diagnostic plots for the ETS model fitted to the quarterly cement production training data.

```
#>
#>  Ljung-Box test
#>
#> data:  Residuals from ETS(M,N,M)
#> Q* = 6.3, df = 3, p-value = 0.1
#>
#> Model df: 6.   Total lags used: 9
```

The output below evaluates the forecasting performance of the two competing models over the test set. In this case the ETS model seems to be the slightly more accurate model based on the test set RMSE, MAPE and MASE.

```
# Generate forecasts and compare accuracy over the test set
a1 <- fit.arima %>% forecast(h = 4*(2013-2007)+1) %>%
  accuracy(qcement)
a1[,c("RMSE","MAE","MAPE","MASE")]
#>                RMSE     MAE  MAPE   MASE
#> Training set 0.1001 0.07989 4.372 0.5458
#> Test set     0.1996 0.16882 7.719 1.1534
a2 <- fit.ets %>% forecast(h = 4*(2013-2007)+1) %>%
  accuracy(qcement)
a2[,c("RMSE","MAE","MAPE","MASE")]
#>                RMSE     MAE  MAPE   MASE
#> Training set 0.1022 0.07958 4.372 0.5437
#> Test set     0.1839 0.15395 6.986 1.0518
```

Notice that the ARIMA model fits the training data slightly better than the ETS model, but that the ETS model provides more accurate forecasts on the test set. A good fit to training data is never an indication that the model will forecast well.

Below we generate and plot forecasts from an ETS model for the next 3 years.

```
# Generate forecasts from an ETS model
cement %>% ets() %>% forecast(h=12) %>% autoplot()
```

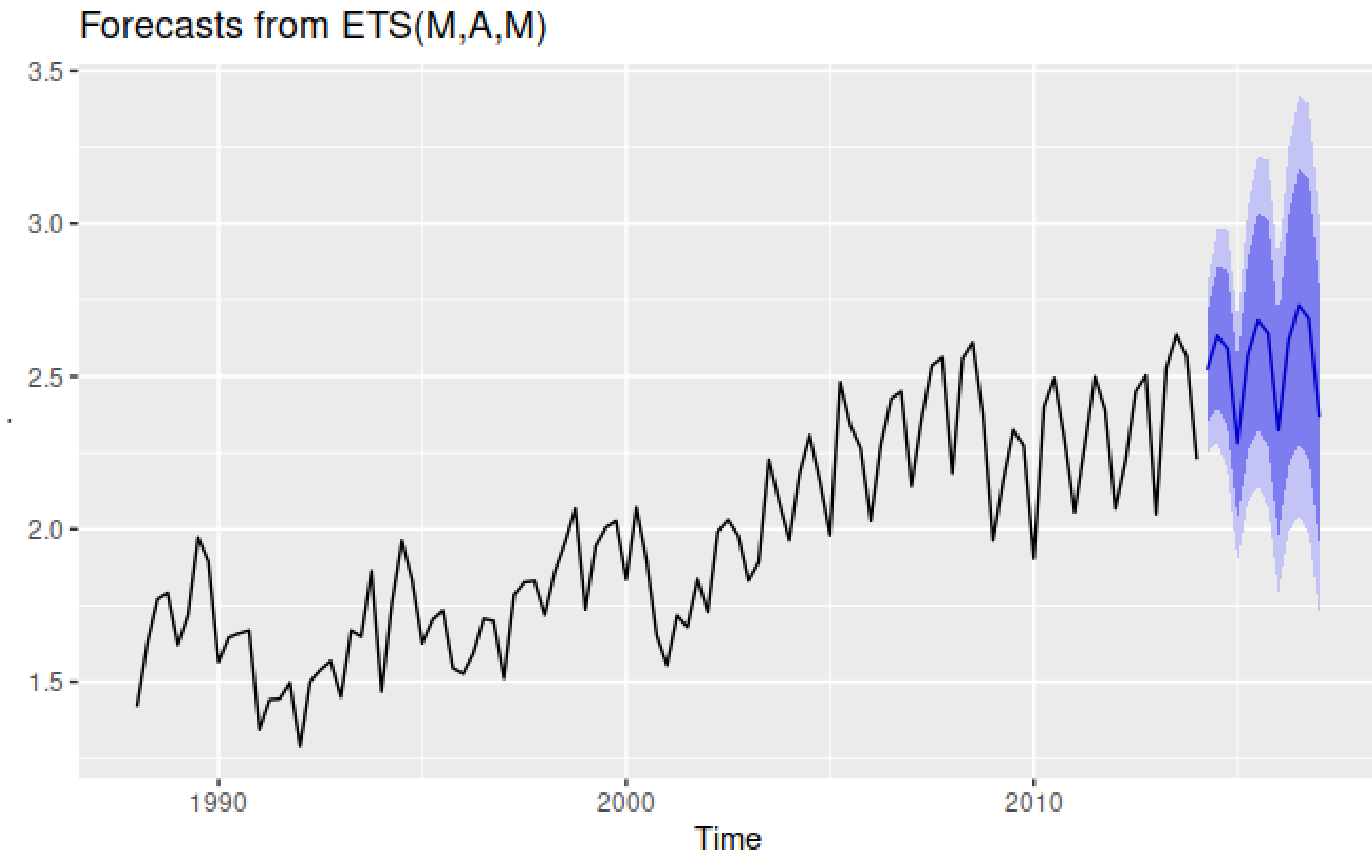

Figure 8.30: Forecasts from an ETS model fitted to all of the available quarterly cement production data.

---

17. As already noted, comparing information criteria is only valid for ARIMA models of the same orders of differencing.↩