Lec 10: ARIMA Pt 2 (Chapter 8) and combination forecast

Forecasting: principles and practice

book by Rob Hyndman and George Athanasopoulos
Parts of the slides are taken from the lecture notes of Robert Nau, Duke University

```
library(fpp2)
library(tseries)
```

Seasonal ARIMA models

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$\text{ARIMA} \quad \underbrace{(p, d, q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Non-seasonal part}\\ \text{of the model}\end{array}\right)}} \quad \underbrace{(P, D, Q)_m}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Seasonal part}\\ \text{of the model}\end{array}\right)}}$$

where m= number of periods per season. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are very similar to the non-seasonal components of the model, but they involve backshifts of the seasonal period. For example, an ARIMA$(1,1,1)(1,1,1)_4$ model (without a constant) is for quarterly data ($m = 4$) and can be written as

$$\underbrace{(1-\phi_1 B)}_{\text{Non-seasonal AR(1)}}\ \underbrace{(1-\Phi_1 B^4)}_{\text{Seasonal AR(1)}}\ \underbrace{(1-B)}_{\text{Non-seasonal difference}}\ \underbrace{(1-B^4)}_{\text{Seasonal difference}}y_t = \underbrace{(1+\theta_1 B)}_{\text{Non-seasonal MA(1)}}\ \underbrace{(1+\Theta_1 B^4)}_{\text{Seasonal MA(1)}}e_t.$$

ACF/PACF

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, an ARIMA(0,0,0)(0,0,1)$_{12}$ model will show:

- a spike at lag 12 in the ACF but no other significant spikes.
- The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36, .

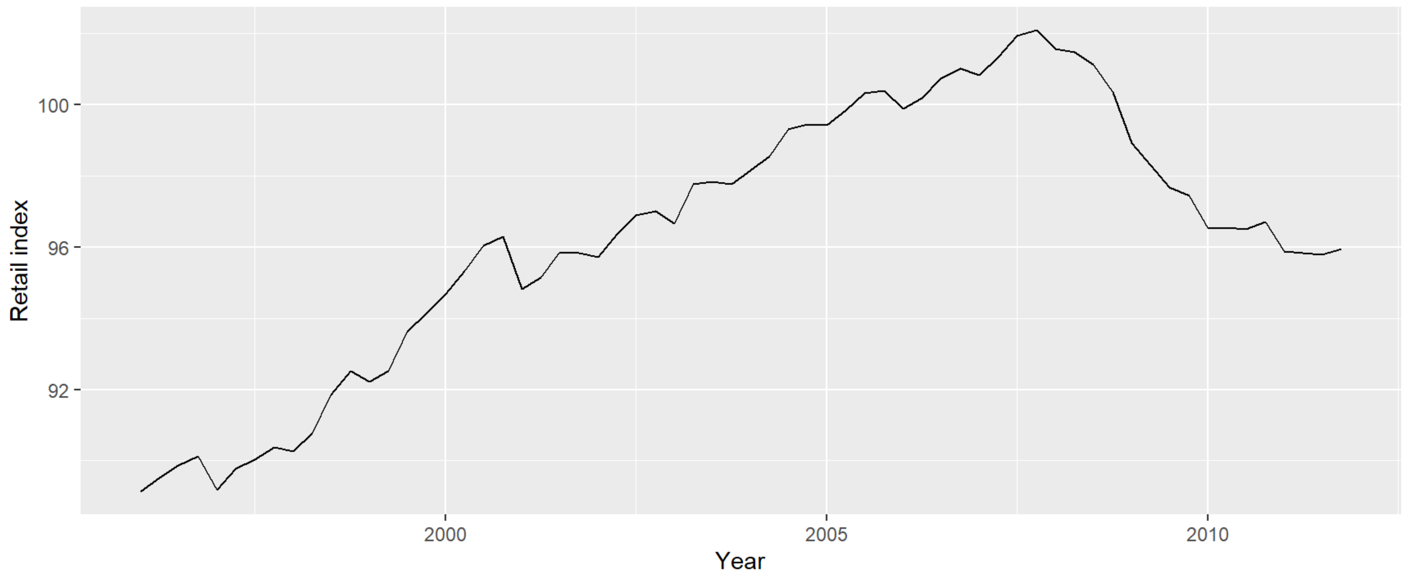
Similarly, an ARIMA(0,0,0)(1,0,0)$_{12}$ model will show:

- exponential decay in the seasonal lags of the ACF
- a single significant spike at lag 12 in the PACF.

The modelling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model. The process is best illustrated via examples.

Example European quarterly retail trade (1996 to 2011)

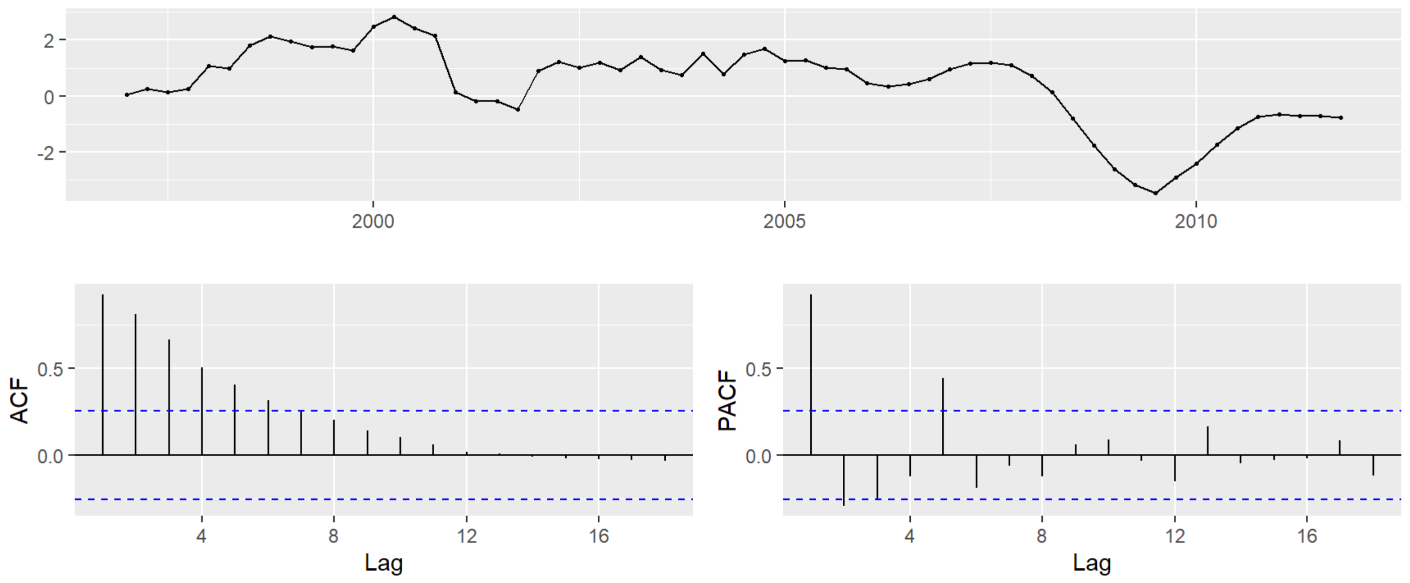
```r
autoplot(euretail) + ylab("Retail index") + xlab("Year")
```



Quarterly retail trade index in the Euro area (17 countries), 1996-2011, covering wholesale and retail trade, and the repair of motor vehicles and motorcycles. (Index: 2005 = 100).

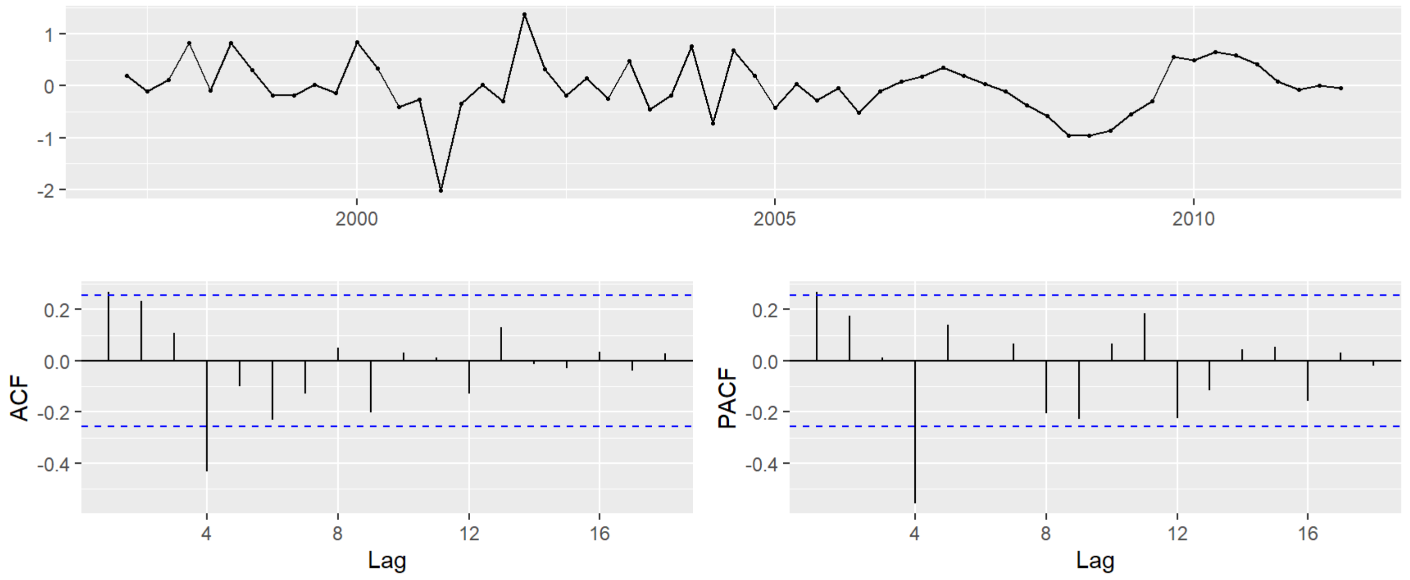
The data are clearly non-stationary, with some seasonality, so we will first take a seasonal difference

```
euretail %>% diff(lag=4) %>% ggtsdisplay
```



Seasonally differenced European retail trade index.

```r
euretail %>% diff(lag=4) %>% diff() %>% ggtsdisplay
```



Double differenced European retail trade index.

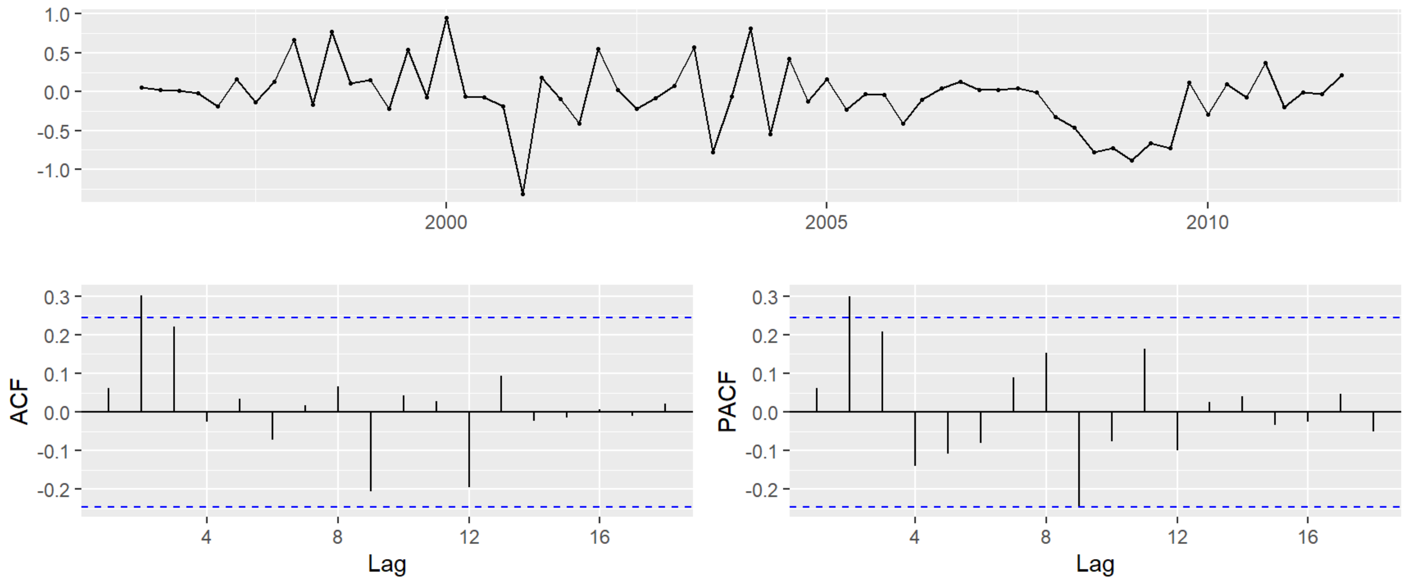
Now we need to find an appropriate ARIMA model based on the ACF and PACF shown in the figure above.

The significant spike at lag 1 in the ACF suggests a non-seasonal MA(1) component, and the significant spike at lag 4 in the ACF suggests a seasonal MA(1) component.

Consequently, we begin with an ARIMA$(0,1,1)(0,1,1)$ model, indicating a first and seasonal difference, and non-seasonal and seasonal MA(1) components.

The residuals for the fitted model are shown in the next few slides (By analogous logic applied to the PACF, we could also have started with an ARIMA$(1,1,0)(1,1,0)$ model.)

```r
fit <- Arima(euretail, order=c(0,1,1), seasonal=c(0,1,1))
ggtsdisplay(residuals(fit))
```

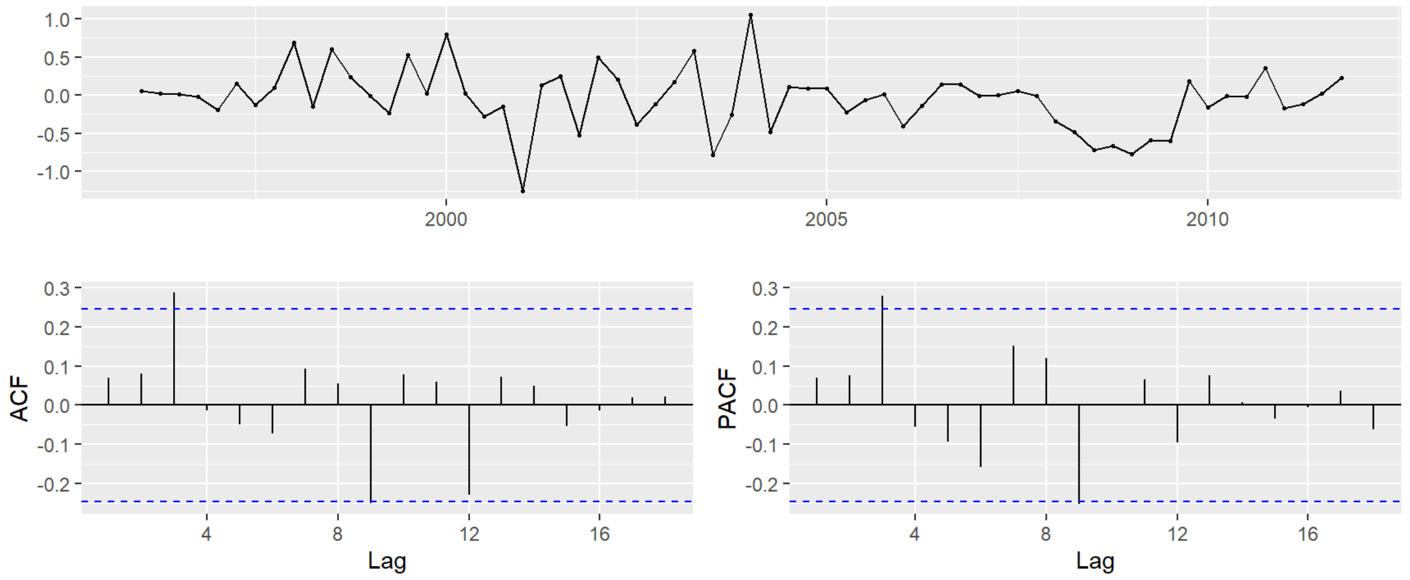


```
summary(fit)
```

```
## Series: euretail
## ARIMA(0,1,1)(0,1,1)[4]
##
## Coefficients:
##          ma1     sma1
##       0.2903  -0.6913
## s.e.  0.1118   0.1193
##
## sigma^2 estimated as 0.188:  log likelihood=-34.64
## AIC=75.28   AICc=75.72   BIC=81.51
##
## Training set error measures:
##                      ME     RMSE       MAE         MPE      MAPE      MASE
## Training set -0.0514931 0.409237 0.2843862 -0.04949815 0.2945738 0.2313337
##                    ACF1
## Training set 0.06136677
```

Both ACF and PACF show significant spikes at lag 2 and almost significant spikes at lag 3 indicating non-seasonal terms need to be included in the model. We could try ARIMA$(0,1,2)(0,1,1)_4$ and ARIMA$(0,1,3)(0,1,1)_4$.

```r
fit3 <- Arima(euretail, order=c(0,1,2), seasonal=c(0,1,1))
ggtsdisplay(residuals(fit3))
```



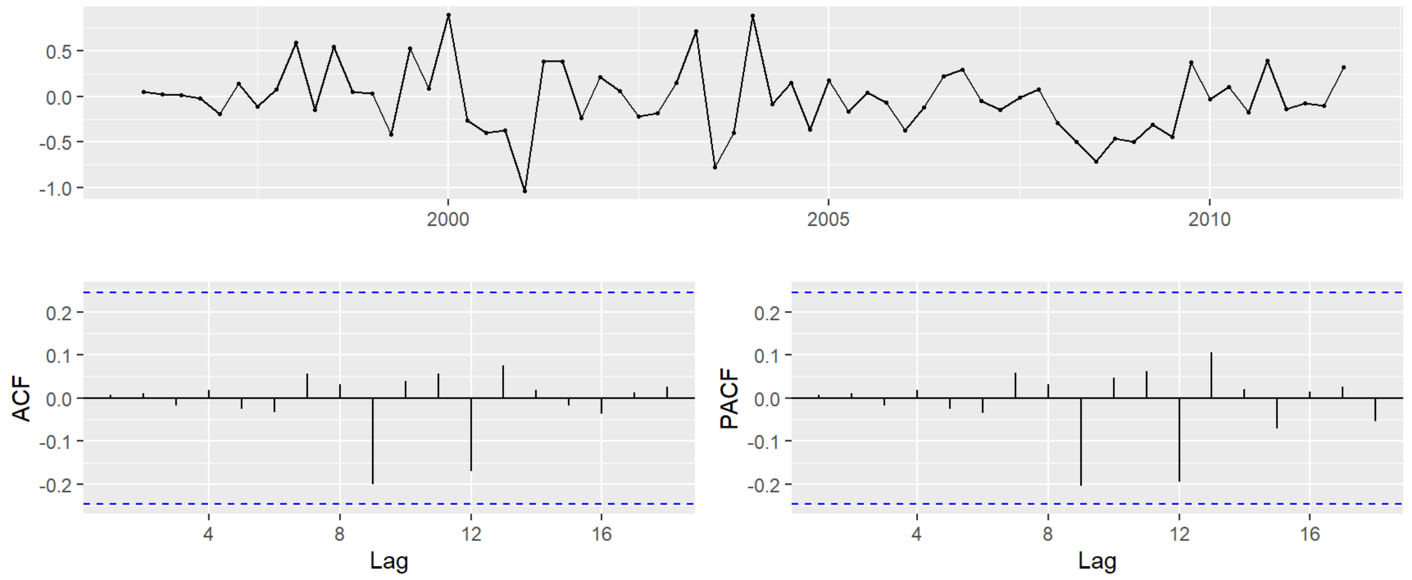
With ARIMA$(0,1,2)(0,1,1)_4$, both ACF and PACF show significant spike at lag 3 so we try ARIMA$(0,1,3)(0,1,1)_4$

```r
summary(fit3)
```

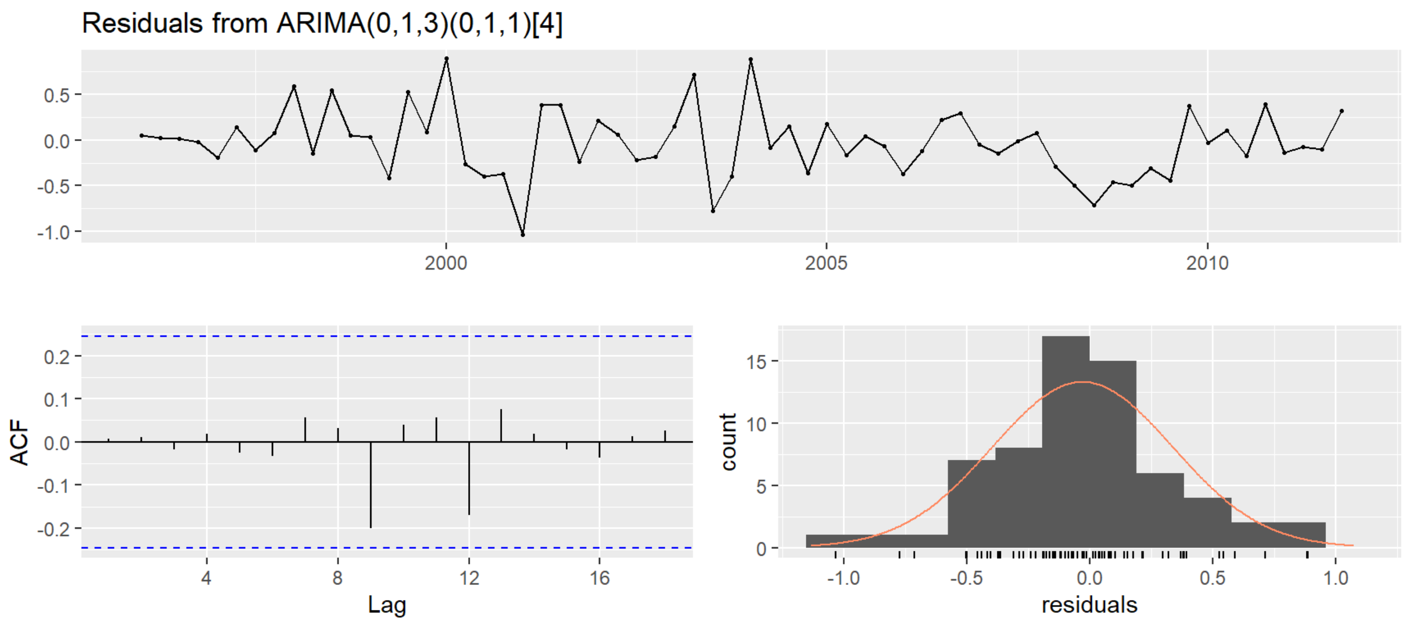
```
## Series: euretail
## ARIMA(0,1,2)(0,1,1)[4]
##
## Coefficients:
##          ma1     ma2     sma1
##       0.2303  0.2502  -0.6991
## s.e.  0.1484  0.1188   0.1284
##
## sigma^2 estimated as 0.1789:  log likelihood=-32.76
## AIC=73.53   AICc=74.27   BIC=81.84
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE      MAPE
## Training set -0.0448743 0.3956301 0.2816785 -0.04296557 0.2916949
##                   MASE       ACF1
## Training set 0.2291311 0.06982161
```

In the previous slide, ARIMA$(0,1,2)(0,1,1)_4$ has both ACF and PACF show significant spike at lag 3 so we want to try ARIMA$(0,1,3)(0,1,1)_4$

```
fit4 <- Arima(euretail, order=c(0,1,3), seasonal=c(0,1,1))
ggtsdisplay(residuals(fit4))
```



```
checkresiduals(fit4)
```



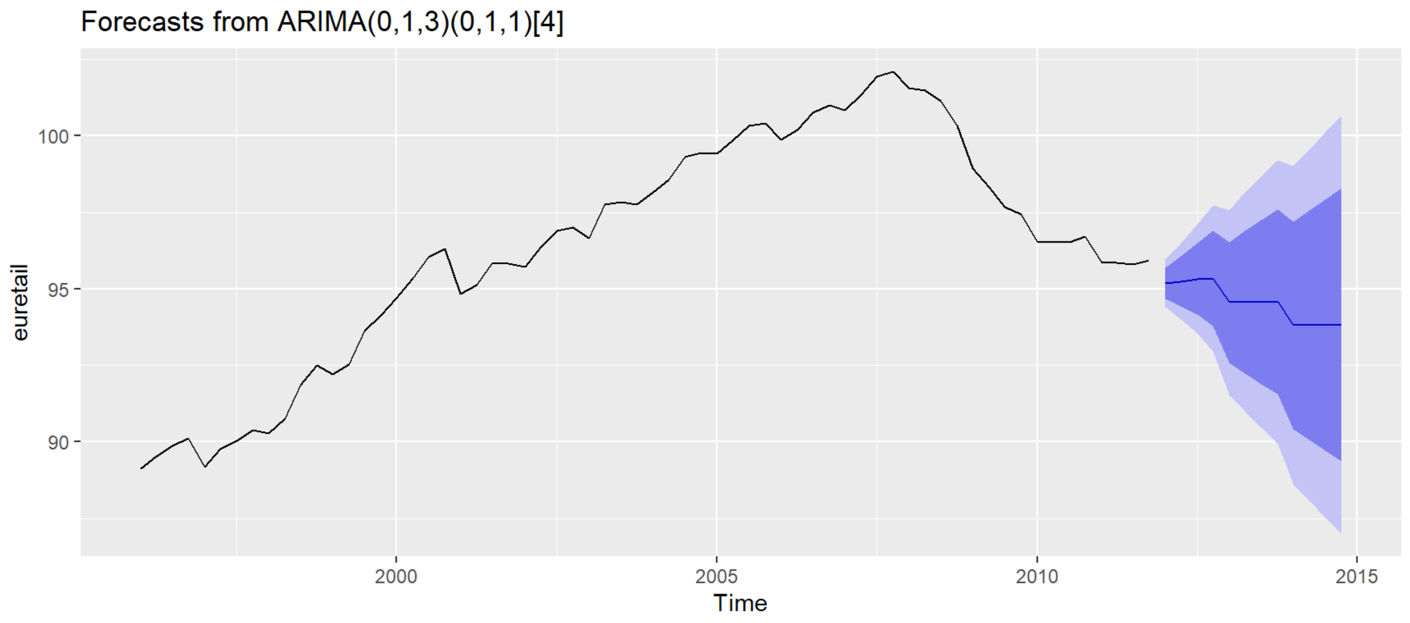
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,3)(0,1,1)[4]
## Q* = 0.51128, df = 4, p-value = 0.9724
##
## Model df: 4.   Total lags used: 8
```

ARIMA$(0,1,3)(0,1,1)_4$ gives the lowest AICc, and the residuals looks like white noise.

We now have a seasonal ARIMA model that passes the required checks and is ready for forecasting. Forecasts from the model for the next three years is shown in the next slide.

The graph in the next slide shows large and rapidly increasing prediction intervals. It shows that the retail trade index could start increasing or decreasing at any time – while the point forecasts trend downwards, the prediction intervals allow for the data to trend upwards during the forecast period.

```r
fit4 %>% forecast(h=12) %>% autoplot
```



Using auto.arima

```
auto.arima(euretail, stepwise=FALSE, approximation=FALSE)
```

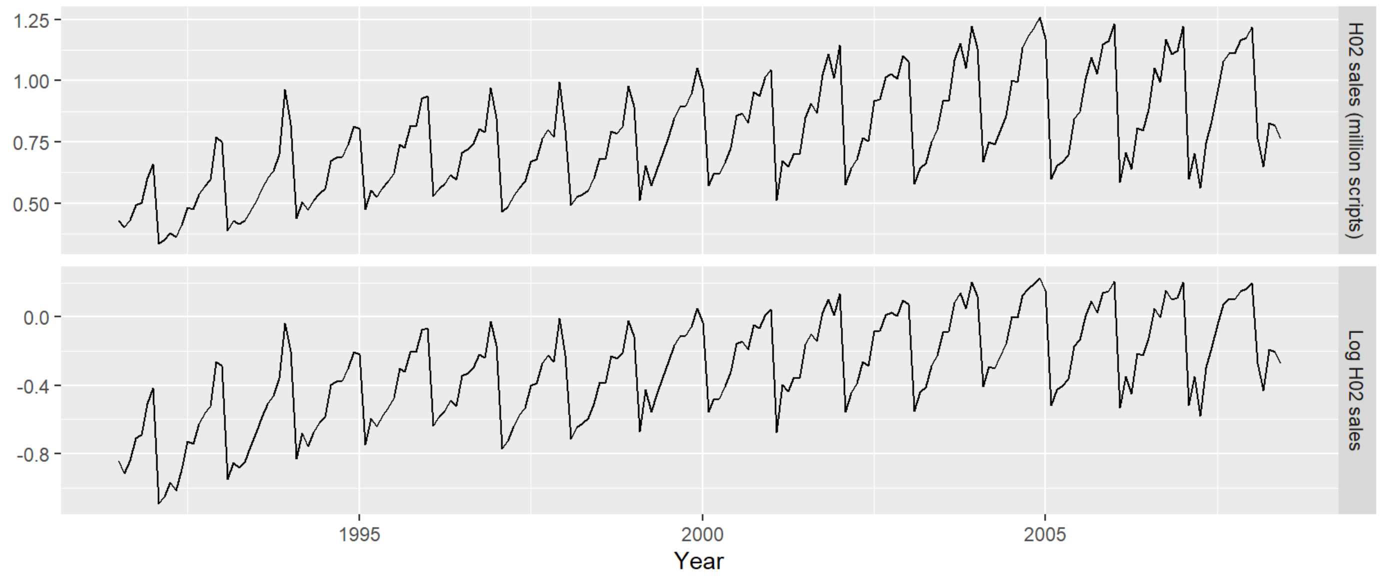
```
## Series: euretail
## ARIMA(0,1,3)(0,1,1)[4]
##
## Coefficients:
##          ma1     ma2     ma3     sma1
##       0.2630  0.3694  0.4200  -0.6636
## s.e.  0.1237  0.1255  0.1294   0.1545
##
## sigma^2 estimated as 0.156:  log likelihood=-28.63
## AIC=67.26   AICc=68.39   BIC=77.65
```

auto.arima with stepwisestepwise=FALSE, approximation=FALSE give the same results

Example: Cortecosteroid drug sales in Australia

The second example is more difficult. We will try to forecast monthly cortecosteroid drug sales in Australia. These are known as H02 drugs under the Anatomical Therapeutic Chemical classification scheme.

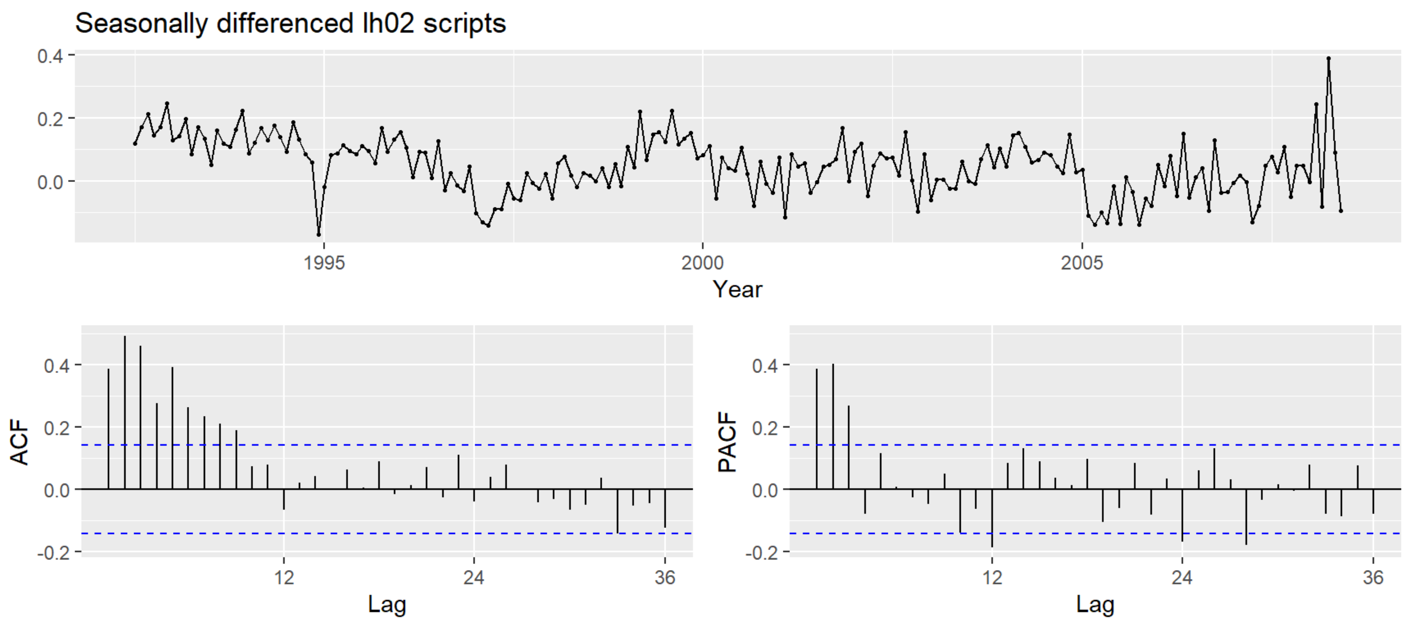
```
lh02 <- log(h02)
cbind("H02 sales (million scripts)" = h02,
       "Log H02 sales"=lh02) %>%
   autoplot(facets=TRUE) + xlab("Year") + ylab("")
```



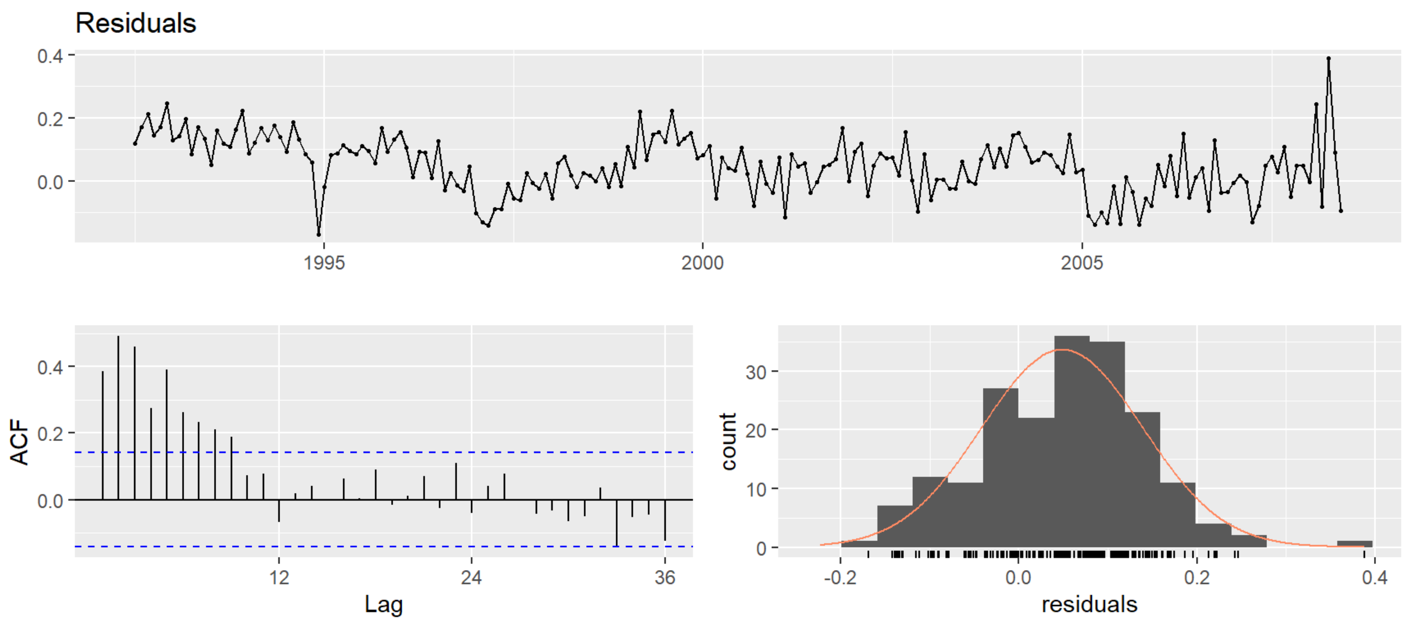
Cortecosteroid drug sales in Australia (in millions of scripts per month) from July 1991 to June 2008. Logged data which stabilizes the variance are shown in bottom panel.

- There is a small increase in the variance with the level, so we take logarithms to stabilize the variance.
- The data are strongly seasonal and obviously non-stationary, so seasonal differencing will be used.
- The seasonally differenced data are shown in in the next slide. It is not clear at this point whether we should do another difference or not. We decide not to, but the choice is not obvious.
- The last few observations appear to be different (more variable) from the earlier data. This may be due to the fact that data are sometimes revised when earlier sales are reported late.

```
ggtsdisplay(diff(lh02,12), main="Seasonally differenced lh02 scripts", xlab="Year")
```



```r
checkresiduals(diff(lh02,12))
```



In the plots of the seasonally differenced data, we observe the following:

- There are spikes in the PACF at lags 12 and 24, but nothing at seasonal lags in the ACF. This may be suggestive of a seasonal AR(2) term.
- In the nonseasonal lags, there are three significant spikes in the PACF, suggesting a possible AR(3) term.
- This initial analysis suggests that a possible model for these data is an ARIMA$(3,0,0)(2,1,0)_{12}$. This model is fitted, along with some variations of it, and compute the AICc values shown in the next slides.

```
fit <- Arima(h02, order=c(3,0,0), seasonal=c(2,1,0), lambda=0); fit
```

```
## Series: h02
## ARIMA(3,0,0)(2,1,0)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2     ar3     sar1    sar2
##       0.0983  0.4060  0.4311  -0.4361  -0.290
## s.e.  0.0695  0.0611  0.0726   0.0795   0.078
##
## sigma^2 estimated as 0.004622:  log likelihood=243.79
## AIC=-475.58   AICc=-475.12   BIC=-456.03
```

Alternatively, we could also use auto.arima to check the initial analysis and fit models among variations of the intial analysis and auto.arima results as a further check. In running auto.arima(), we need to set the difference ($d = 0$), seasonal difference ($D = 1$) and the Box-Cox transformation (lambda=0) similar to our initial model so the AICc may be comparable. Recall that differencing transforms the data which make AICc difficult to compare.

In running auto.arima(), we use the options *stepwise=FALSE, approximation=FALSE* which returns a model ARIMA $(3, 0, 1)(0, 1, 2)_{12}$ with drift.

```
fit <- auto.arima(h02, lambda=0, d=0, D=1,
                  stepwise=FALSE, approximation=FALSE); fit
```

```
## Series: h02
## ARIMA(3,0,0)(0,1,2)[12] with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2     ar3     sma1     sma2   drift
##       0.1367  0.3788  0.3540  -0.5242  -0.1598  0.0039
## s.e.  0.0743  0.0654  0.0761   0.0837   0.0855  0.0010
##
## sigma^2 estimated as 0.00427:  log likelihood=250.78
## AIC=-487.55   AICc=-486.94   BIC=-464.75
```

In this case, auto.arima found a model ARIMA$(3,0,0)(0,1,2)_{12}$ with drift

To make the initial model comparable, we rerun Arima with drift and without drift

```
fit2 <- Arima(h02, order=c(3,0,0), seasonal=c(2,1,0), lambda=0, include.drift = TRUE); fit2
```

```
## Series: h02 
## ARIMA(3,0,0)(2,1,0)[12] with drift 
## Box Cox transformation: lambda= 0 
## 
## Coefficients:
##          ar1     ar2     ar3      sar1      sar2   drift
##       0.0740  0.3751  0.4054  -0.4271  -0.2850  0.0042
## s.e.  0.0702  0.0631  0.0737   0.0799   0.0781  0.0016
## 
## sigma^2 estimated as 0.004566:  log likelihood=245.78
## AIC=-477.56   AICc=-476.96   BIC=-454.76
```

```
fit3 <- Arima(h02, order=c(3,0,0), seasonal=c(2,1,0), lambda=0, include.drift = FALSE); fit3
```

```
## Series: h02
## ARIMA(3,0,0)(2,1,0)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2     ar3     sar1    sar2
##       0.0983  0.4060  0.4311  -0.4361  -0.290
## s.e.  0.0695  0.0611  0.0726   0.0795   0.078
##
## sigma^2 estimated as 0.004622:  log likelihood=243.79
## AIC=-475.58   AICc=-475.12   BIC=-456.03
```

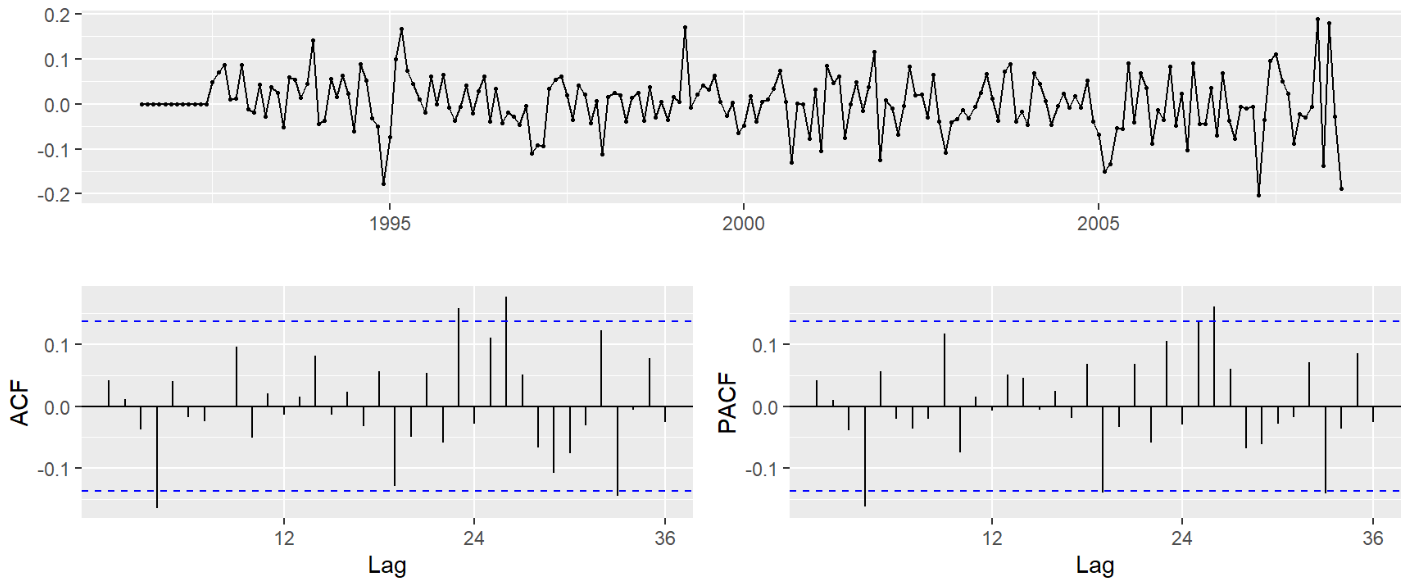
Comparing AICc we have:

- ARIMA$(3,0,0)(2,1,0)_{12}$ AICc = -475.12
- ARIMA$(3,0,0)(2,1,0)_{12}$ with drift,AICc = -476.96
- ARIMA$(3,0,0)(0,1,2)_{12}$ AICc = -475.58
- ARIMA$(3,0,0)(0,1,2)_{12}$ with drift AICc = -486.94

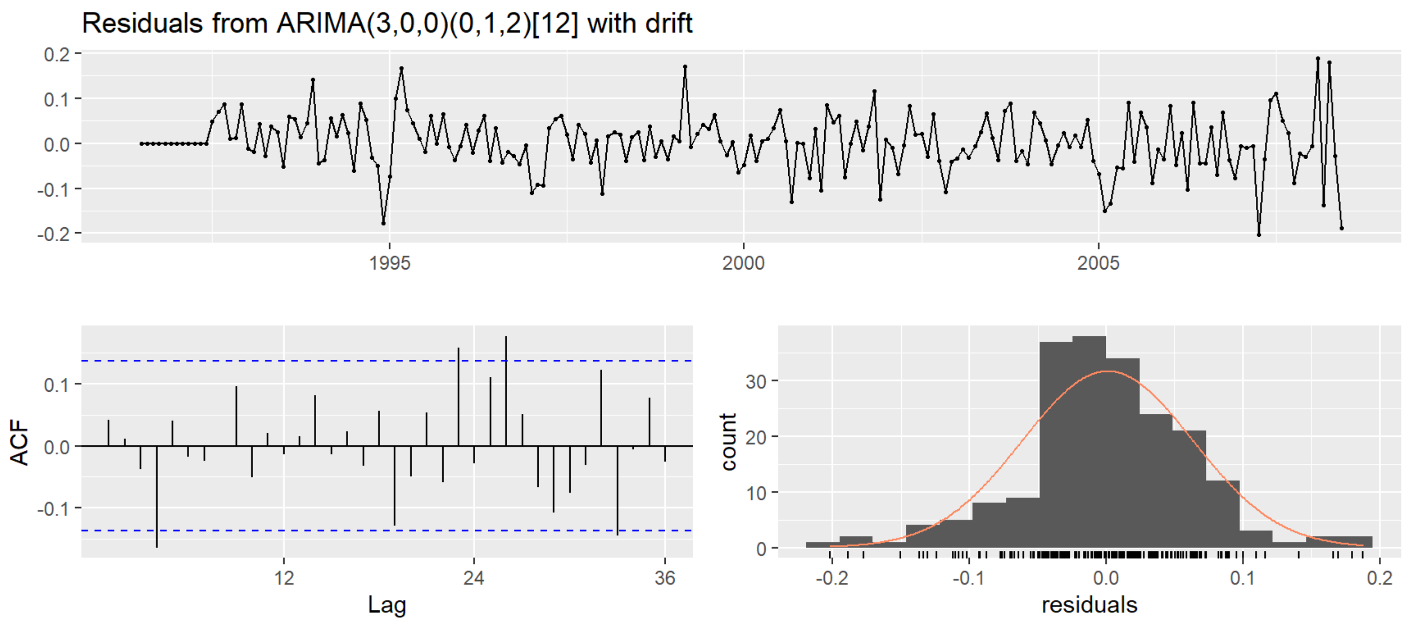
Hence, we pick ARIMA$(3,0,0)(0,1,2)_{12}$ with drift based on lowest AICc = -486.94.

The residual diagnostics on this best model are in the next slides

```
ggtsdisplay(residuals(fit))
```



```
checkresiduals(fit, lag=36)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0)(0,1,2)[12] with drift
## Q* = 50.892, df = 30, p-value = 0.01
##
## Model df: 6.   Total lags used: 36
```

There are a few significant spikes in the ACF, and the model fails the Ljung-Box test.

The model can still be used for forecasting, but the prediction intervals may not be accurate due to the correlated residuals.

Sometimes it is just not possible to find a model that passes all of the tests

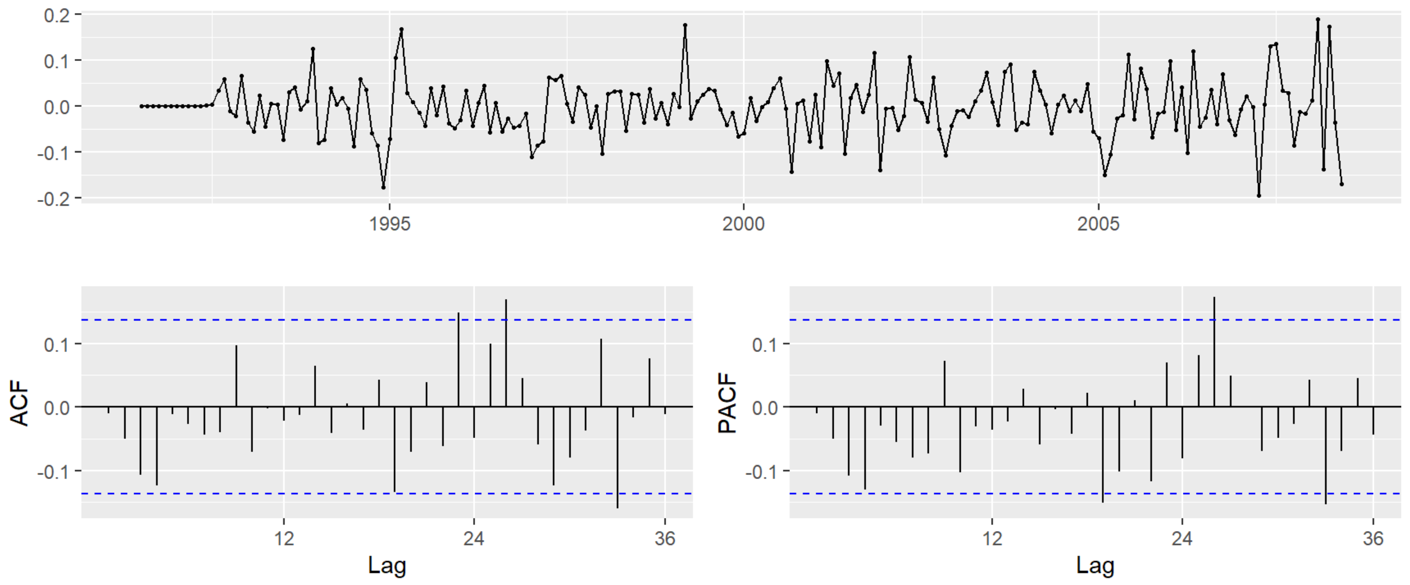
Note that above, we constrained auto.arima to D=1. Suppose we just run auto.arima as given below:

```
fitauto <- auto.arima(h02, lambda=0, stepwise=FALSE, approximation=FALSE); fitauto
```

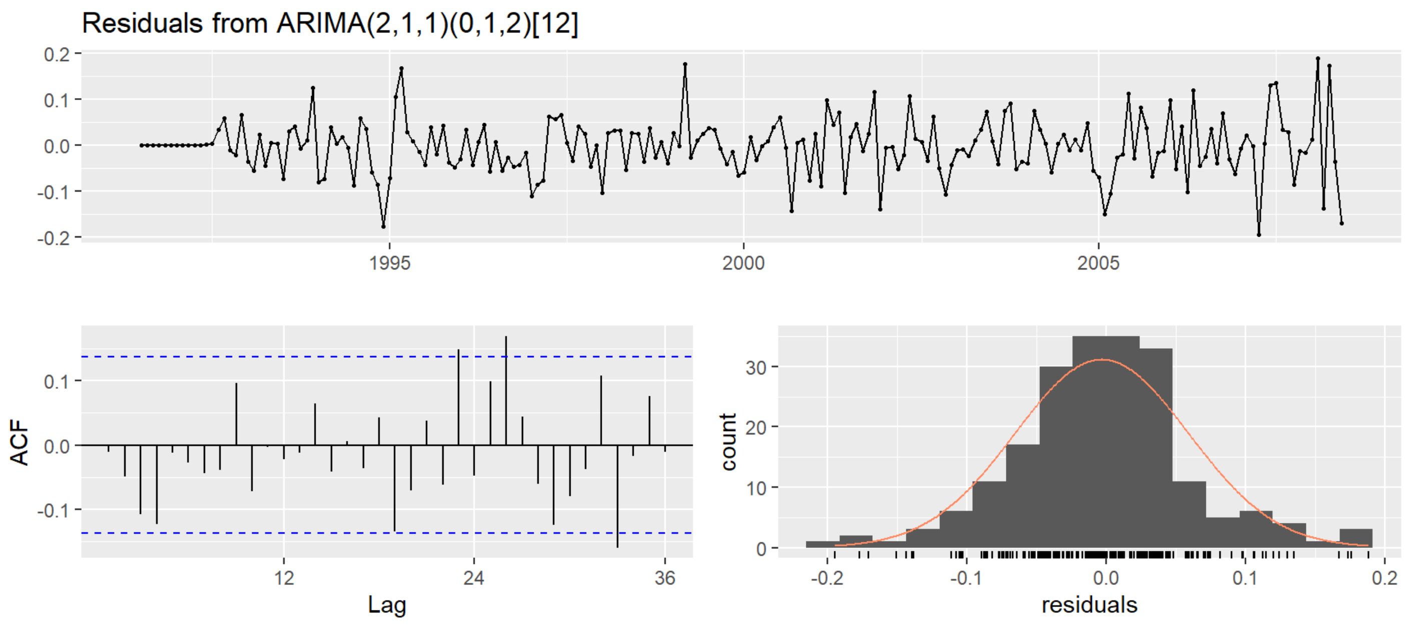
```
## Series: h02
## ARIMA(2,1,1)(0,1,2)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##           ar1      ar2     ma1     sma1     sma2
##       -1.1358  -0.5753  0.3683  -0.5318  -0.1817
## s.e.   0.1608   0.0965  0.1884   0.0838   0.0881
##
## sigma^2 estimated as 0.004278:  log likelihood=248.25
## AIC=-484.51   AICc=-484.05   BIC=-465
```

Gives a different model than ARIMA$(3,0,0)(0,1,2)_{12}$ with drift above. Here we cannot compare the two models using AICc because ARIMA$(3,0,0)(0,1,2)_{12}$ has d=0 and D=1 and ARIMA$(2,1,1)(0,1,2)_{12}$ has d=1 and D=1. However, we can compare the two models, among other models, using mean square errors below.

```
ggtsdisplay(residuals(fitauto))
```



```
checkresiduals(fitauto)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(0,1,2)[12]
## Q* = 24.514, df = 19, p-value = 0.1772
##
## Model df: 5.   Total lags used: 24
```

Here that ARIMA$(2,1,1)(0,1,2)_{12}$ cannot reject the null of no autocorrelation at the 10% level of significance

Test set evaluation

We will compare some of the models fitted so far using a test set consisting of the last two years of data.

Thus, we fit the models using data from July 1991 to June 2006, and forecast the script sales for July 2006 - June 2008.

The criterion is the RMSE

```r
getrmse <- function(x,h,...)
{
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fit <- Arima(train,...)
  fc <- forecast(fit,h=h)
  return(accuracy(fc,test)[2,"RMSE"])
}

c(getrmse(h02,h=24,order=c(3,0,0),seasonal=c(2,1,0),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(2,1,0),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(2,1,0),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,0),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(0,1,1),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(0,1,2),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,1),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(4,0,3),seasonal=c(0,1,1),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,3),seasonal=c(0,1,1),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(4,0,2),seasonal=c(0,1,1),lambda=0, include.drift=TRUE),
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(0,1,1),lambda=0, include.drift=TRUE))
```

```
##  [1] 0.08950036 0.09075583 0.09712397 0.07781312 0.08133615 0.09921112
##  [7] 0.09238534 0.08117906 0.07477604 0.09842800 0.08586048
```

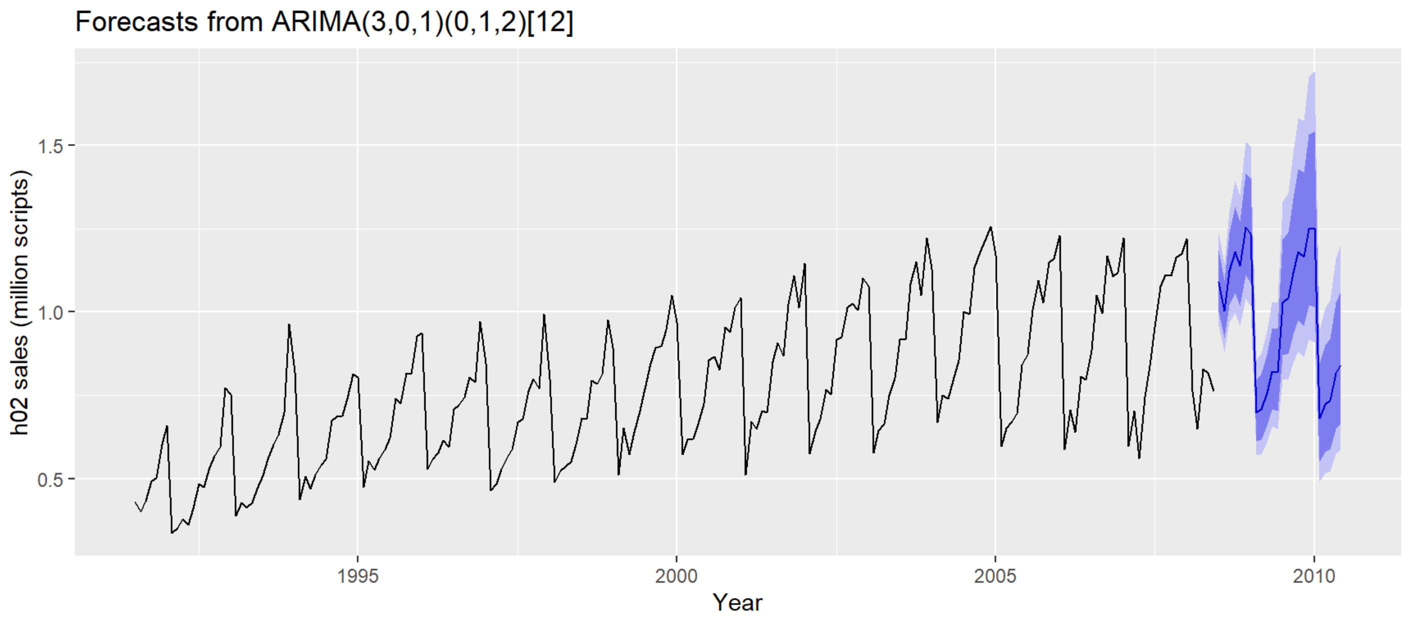
```r
getrmse <- function(x,h,...)
{
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fit <- Arima(train,...)
  fc <- forecast(fit,h=h)
  return(accuracy(fc,test)[2,"RMSE"])
}

c(getrmse(h02,h=24,order=c(3,0,0),seasonal=c(2,1,0),lambda=0),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(2,1,0),lambda=0),
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(2,1,0),lambda=0),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,0),lambda=0),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(0,1,2),lambda=0),
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,1),lambda=0),
getrmse(h02,h=24,order=c(4,0,3),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(3,0,3),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(4,0,2),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(2,1,3),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(2,1,4),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(2,1,5),seasonal=c(0,1,1),lambda=0),
getrmse(h02,h=24,order=c(2,1,1),seasonal=c(0,1,2),lambda=0))
```

```
##  [1] 0.06610388 0.06463807 0.06450797 0.06793329 0.06436838 0.06217316
##  [7] 0.06299956 0.06484697 0.06397438 0.06475151 0.06438796 0.06338313
## [13] 0.06315186 0.06398250 0.06342027
```

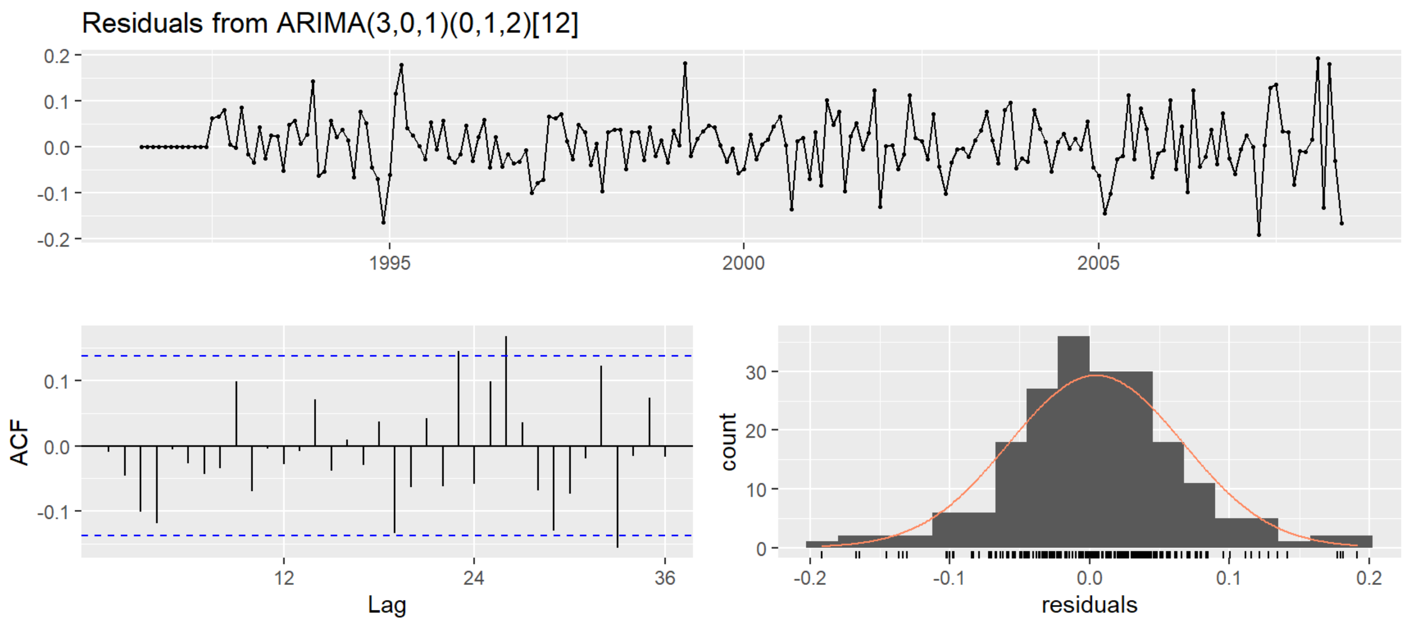
We note that even though AICc gave lowest values for the model with drift, RMSE was the least at 0.06217316 for ARIMA$(3,0,1)(0,1,2)_{12}$. This model is used for forecasting.

As mentioned above, our analysis had as our "best" a model which may still have autocorrelation and therefore prediction intervals that are inaccurate.

```
fit <- Arima(h02, order=c(3,0,1), seasonal=c(0,1,2), lambda=0)
autoplot(forecast(fit), ylab="h02 sales (million scripts)", xlab="Year")
```



```
checkresiduals(fit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,1)(0,1,2)[12]
## Q* = 23.663, df = 18, p-value = 0.1664
##
## Model df: 6.   Total lags used: 24
```

ARIMA vs ETS

It is a common 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. There are also many ARIMA models that have no exponential smoothing counterparts. In particular, every ETS model is non-stationary, while ARIMA models must be stationary.

The ETS models with seasonality and non-damped trend 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) – please refer to the table in the next slide.

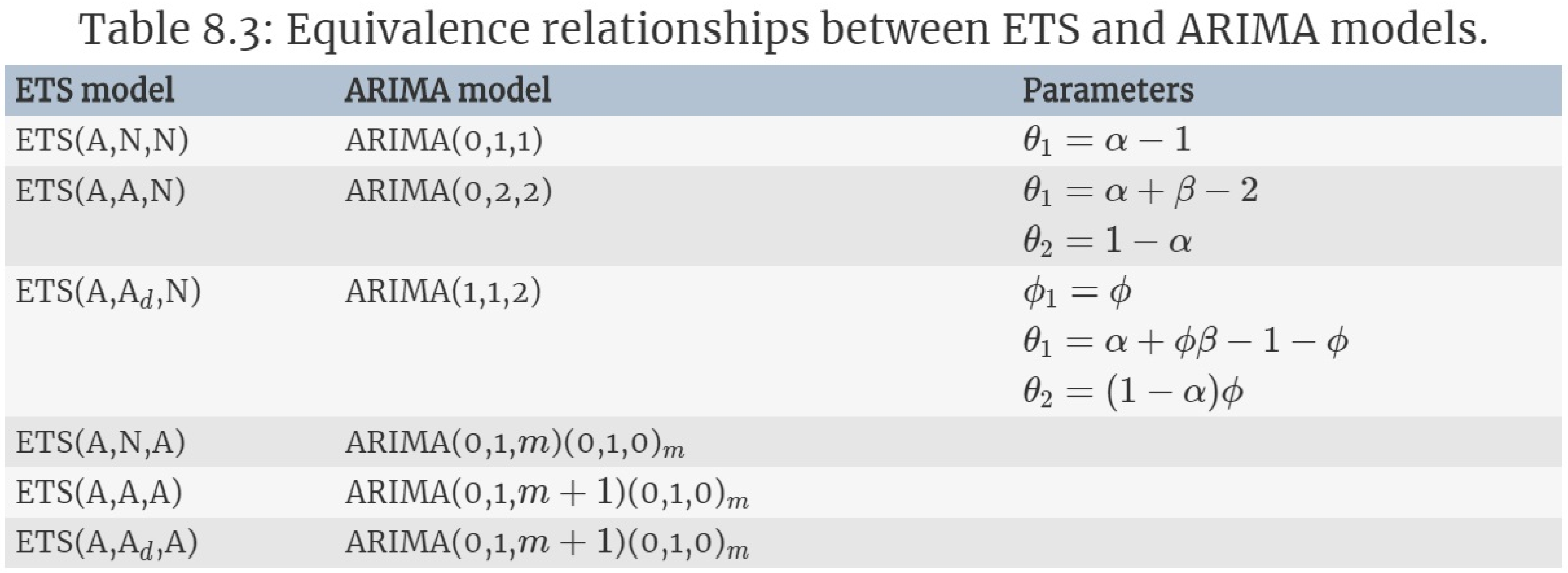


Table 8.3: Equivalence relationships between ETS and ARIMA models.

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)$_m$	

Equivalence relationships of some of the ETS and ARIMA models

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 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. An example would be as follows:

```
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

Using ausair data

```
air <- window(ausair, start=1990)
fets <- function(air, h) {
 forecast(ets(air), h = h)
}
farima <- function(air, h) {
forecast(auto.arima(air), h=h)
}
```

```
# 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] 7.864374
```

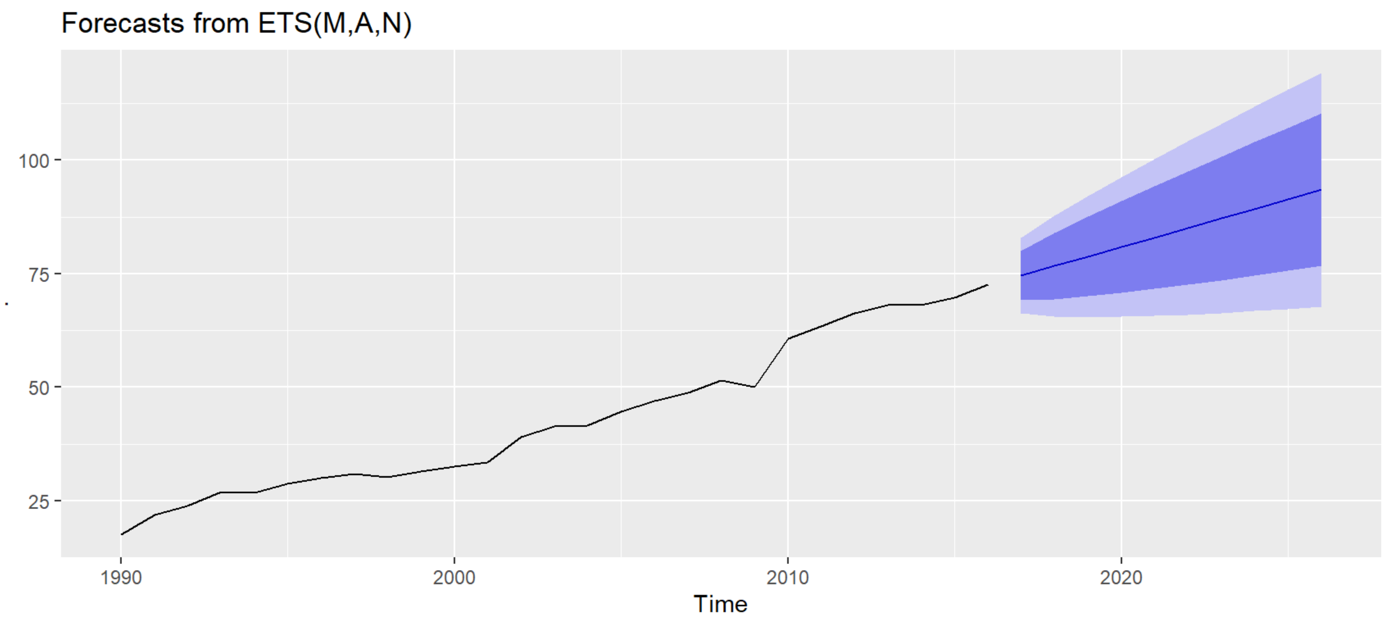
```
#> [1] 7.86
mean(e2^2, na.rm=TRUE)
```

```
## [1] 9.622164
```

```
#> [1] 9.62
```

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()
```



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 very long, we can afford to use a training and a test set only using accuracy () rather than tsCV() - the latter will take a lot of computing time.

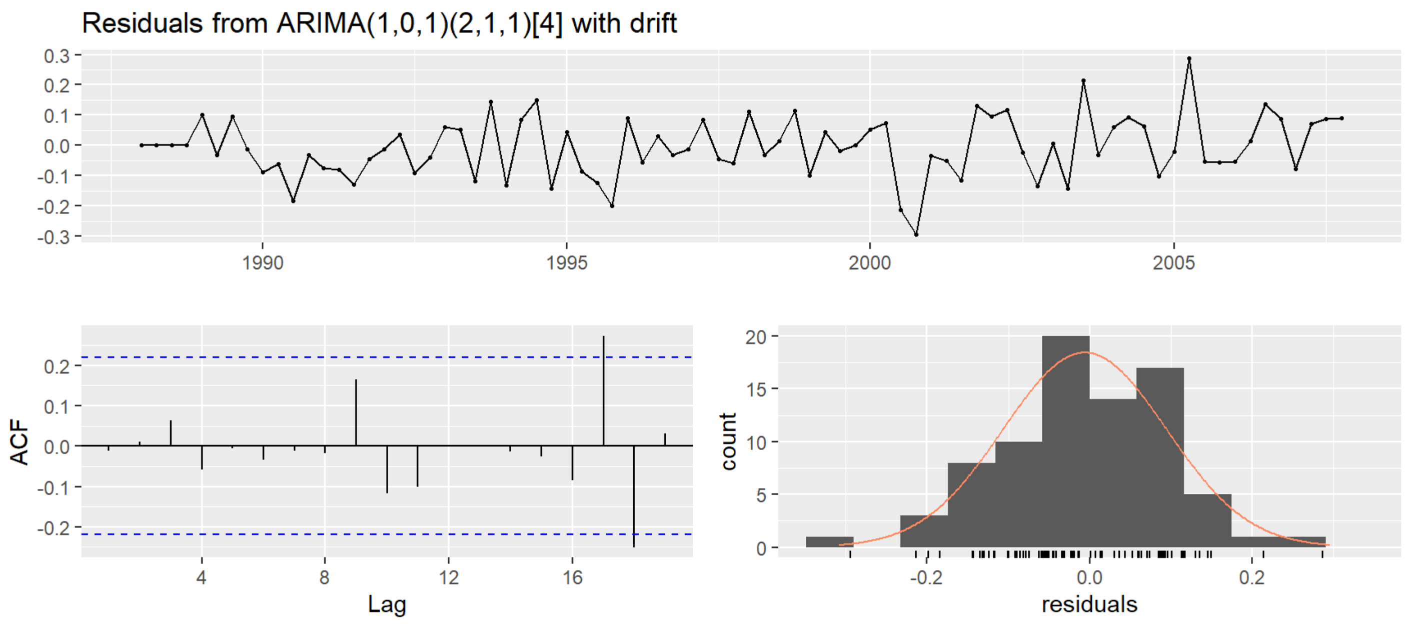
We create a training set from the beginning of 1988 to the end of 2007 and select an ARIMA and an ETS model using the *auto.arima* and *ets* functions.

```r
# 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))
```

```r
# Fit an ARIMA model to the training data
fit.arima <- auto.arima(train);
fit.arima
```

```
## Series: train
## ARIMA(1,0,1)(2,1,1)[4] with drift
##
## Coefficients:
##          ar1      ma1   sar1     sar2     sma1   drift
##       0.8886  -0.2366  0.081  -0.2345  -0.8979  0.0105
## s.e.  0.0842   0.1334  0.157   0.1392   0.1780  0.0029
##
## sigma^2 estimated as 0.01146:  log likelihood=61.47
## AIC=-108.95   AICc=-107.3   BIC=-92.63
```

```
checkresiduals(fit.arima)
```

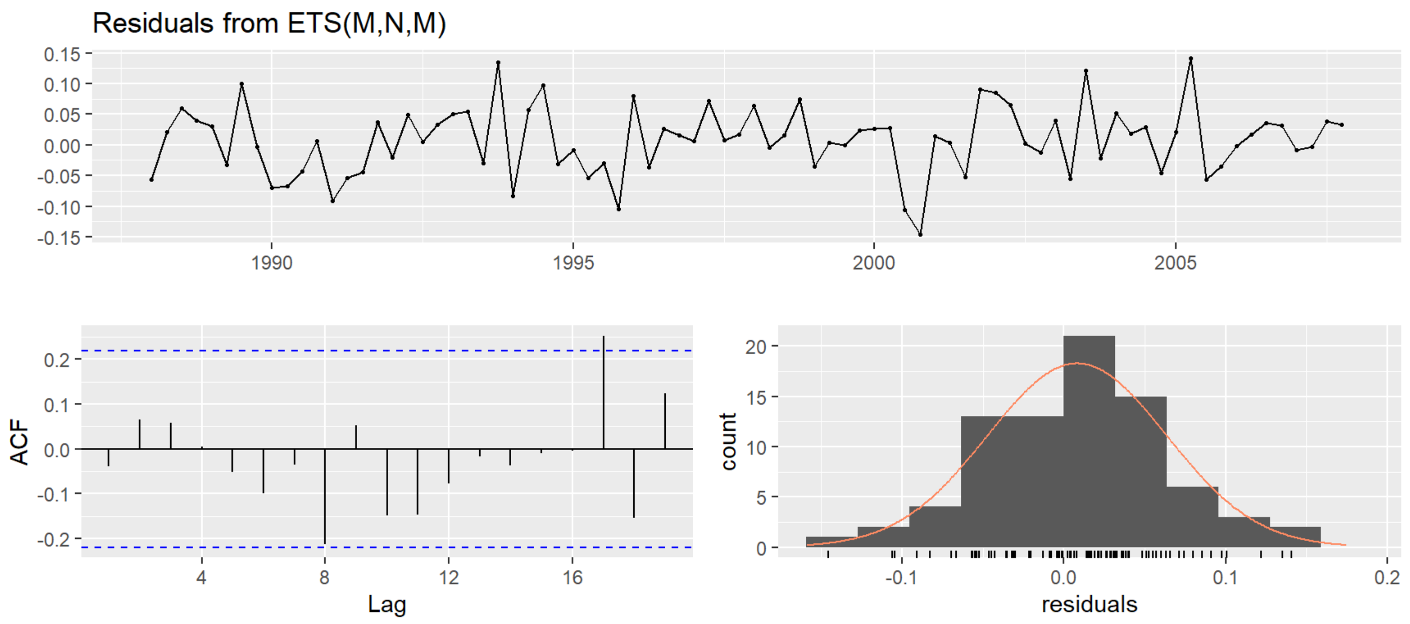


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(2,1,1)[4] with drift
## Q* = 3.3058, df = 3, p-value = 0.3468
##
## Model df: 6.   Total lags used: 9
```

The output below also shows the ETS model selected and estimated by ets . This models also does well in capturing all the dynamics in the data as the residuals also seem to be white noise.

```
# Fit an ETS model to the training data
fit.ets <- ets(train)
```

```
checkresiduals(fit.ets)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,N,M)
## Q* = 6.3457, df = 3, p-value = 0.09595
##
## 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 slighlty better model based on the test set.

```
# Generate forecasts and compare accuracy over the test set
fit.arima %>% forecast(h = 4*(2013-2007)+1) %>% accuracy(qcement)
```

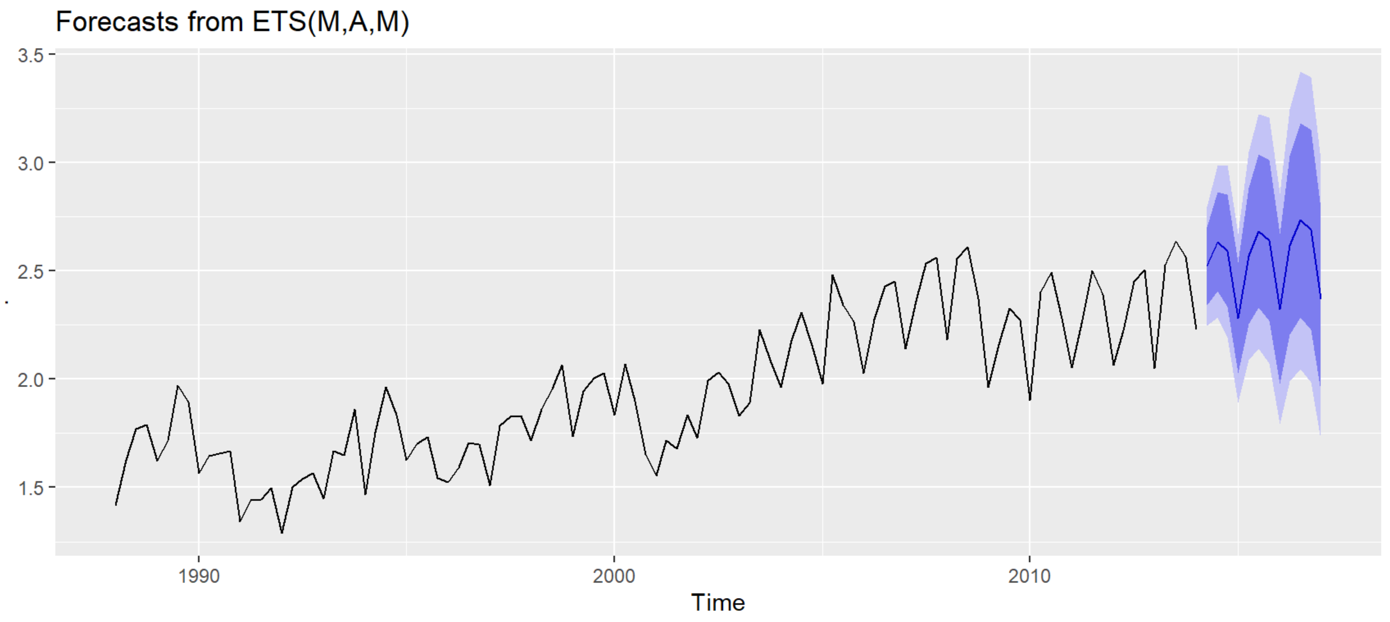
```
##                        ME      RMSE        MAE        MPE     MAPE
## Training set -0.006205705 0.1001195 0.07988903 -0.6704455 4.372443
## Test set     -0.158835253 0.1996098 0.16882205 -7.3332836 7.719241
##                   MASE        ACF1 Theil's U
## Training set 0.5458078 -0.01133907        NA
## Test set     1.1534049  0.29170452 0.7282225
```

```
fit.ets %>% forecast(h = 4*(2013-2007)+1) %>% accuracy(qcement)
```

```
##                       ME      RMSE        MAE        MPE     MAPE
## Training set  0.01406512 0.1022079 0.07958478  0.4938163 4.371823
## Test set     -0.13495515 0.1838791 0.15395141 -6.2508975 6.986077
##                   MASE        ACF1 Theil's U
## Training set 0.5437292 -0.03346295        NA
## Test set     1.0518075  0.53438371  0.680556
```

For most indicators in the test set, the ETS model seems to be the better model

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



Forecast combinations (Section 12.4)

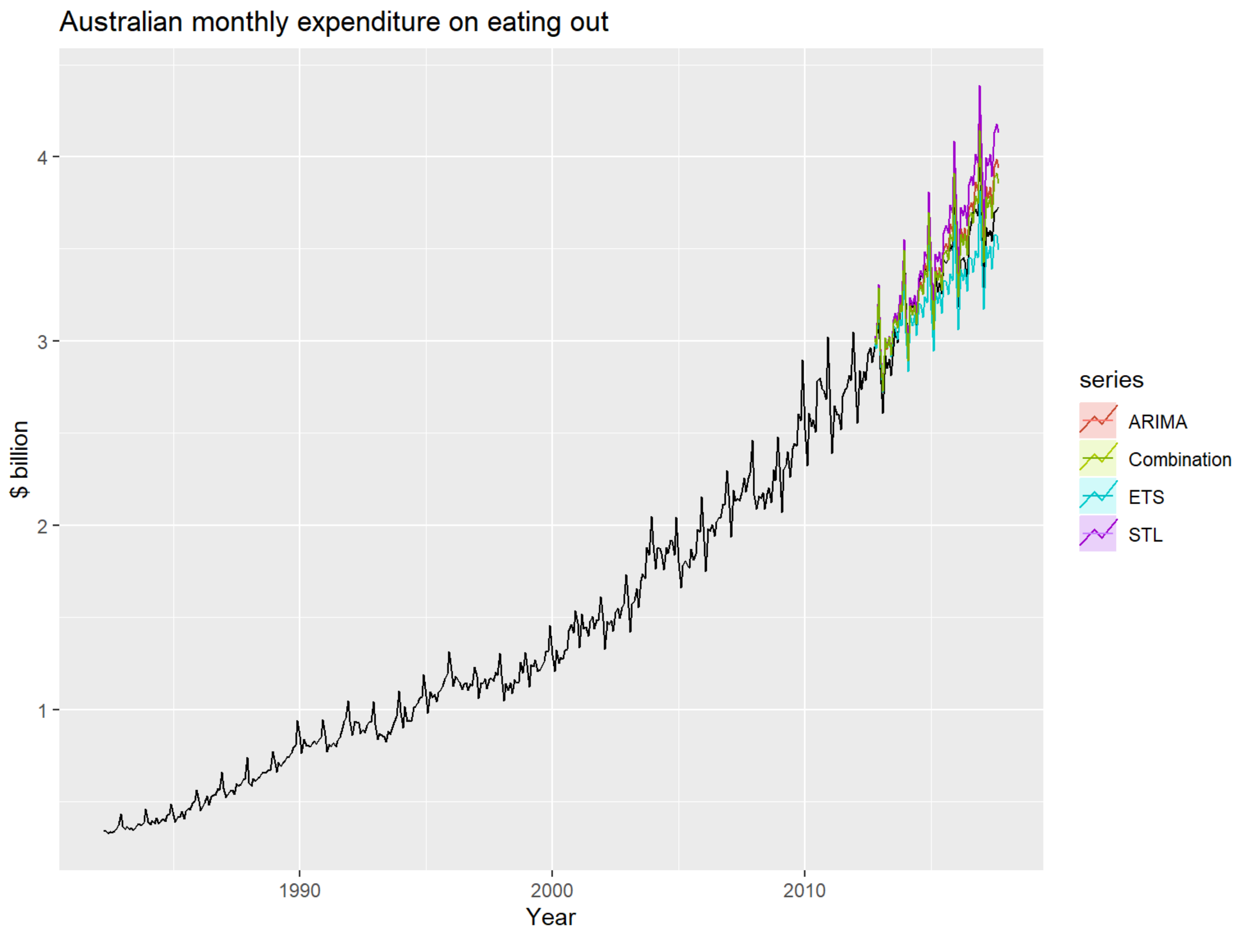
An easy way to improve forecast accuracy is to use several different methods on the same time series, and to average the resulting forecasts. Nearly 50 years ago, John Bates and Clive Granger wrote a famous paper (Bates & Granger, 1969), showing that combining forecasts often leads to better forecast accuracy. Twenty years later, Clemen (1989) wrote:

The results have been virtually unanimous: combining multiple forecasts leads to increased forecast accuracy. In many cases one can make dramatic performance improvements by simply averaging the forecasts.

Here is an example using monthly expenditure on eating out in Australia, from April 1982 to September 2017. We use forecasts from the following models: ETS, ARIMA and STL-ETS.

```r
train <- window(auscafe, end=c(2012,9))
h <- length(auscafe) - length(train)
ETS <- forecast(ets(train), h=h)
ARIMA <- forecast(auto.arima(train, lambda=0, biasadj=TRUE),
  h=h)
STL <- stlf(train, lambda=0, h=h, biasadj=TRUE)
Combination <- (ETS[["mean"]] + ARIMA[["mean"]] +
  STL[["mean"]])/3
```

```r
autoplot(auscafe) +
  autolayer(ETS, series="ETS", PI=FALSE) +
  autolayer(ARIMA, series="ARIMA", PI=FALSE) +
  autolayer(STL, series="STL", PI=FALSE) +
  autolayer(Combination, series="Combination") +
  xlab("Year") + ylab("$ billion") +
  ggtitle("Australian monthly expenditure on eating out")
```


Australian monthly expenditure on eating out
$ billion
1
2
3
4
1990
2000
2010
Year
series
ARIMA
Combination
ETS
STL

```r
c(ETS = accuracy(ETS, auscafe)["Test set","RMSE"],
  ARIMA = accuracy(ARIMA, auscafe)["Test set","RMSE"],
  `STL-ETS` = accuracy(STL, auscafe)["Test set","RMSE"],
  Combination =
    accuracy(Combination, auscafe)["Test set","RMSE"])
```

```
##         ETS       ARIMA     STL-ETS Combination
##  0.13699696  0.12146220  0.21446157  0.09325927
```

Here we see that the combination forecast better than any of the 3 models.

Thank you for your attention!