Module 5 - Exponential Smoothing Methods

MATH1307 Forecasting

RMIT University, School of Science, Mathematical Sciences

All the contents in this presentation are mainly based on the textbook of MATH1307 Forecasting course 'Hyndman et al., Forecasting with Exponential Smoothing: The State Space Approach. Springer, 2008.' Other resources than the textbook are cited accordingly.

Introduction

Exponential smoothing methods are used to model time series data by fitting different trend and seasonality patterns.

They provide a class of alternative models for time series data.

State-space model approach makes the inclusion of additive and multiplicative error terms into the exponential smoothing models possible.

In this module, we will cover

- Assessing forecast accuracy
- Exponential smoothing methods
    - Simple exponential smoothing
    - Holt’s linear method
    - Holt-Winters’ trend and seasonality method

- State-space models
    - Additive errors
    - Multiplicative errors
- Practical applications

Assessing Forecast Accuracy

We can use different error measures to assess forecast accuracy.

Define the one-step-ahead forecast error as $e_t = Y_t - \hat{Y}_t$, then we have the following *scale-dpenedent* errors:

- Mean absolute error (MAE) = $mean(|e_t|)$.
- Mean squared error (MSE) = $mean(e_t^2)$.
- Root mean squared error (RMSE) = $\sqrt{mean(e_t^2)}$.

When comparing forecast methods on a single series, we prefer the MAE, which cannot be used to make comparisons between series as it makes no sense to compare accuracy on different scales.

The percentage error is defined by $p_t = 100e_t/Y_t$, which is scale-independent. We have the following percentage errors:

- Mean absolute percentage error (MAPE) = $mean(|p_t|)$.
- Symmetric mean absolute percentage error (sMAPE) = $mean(200|Y_t - \hat{Y}_t|/(Y_t + \hat{Y}_t))$.

Notice that because the value of sMAPE can be negative, it is not a measure of *absolute percentage errors* at all.

Mean absolute scaled error (MASE) is another generally applicable measure of forecast accuracy.

It is obtained by scaling the errors based on the in-sample MAE from the naiıve forecast method.

Thus, a scaled error is defined as

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n} |Y_i - Y_{i-1}|}, \tag{1}$$

which is independent of the scale of the data.

A scaled error is less than one if it arises from a better forecast than the average one-step naiıve forecast computed in-sample.

Conversely, it is greater than one if the forecast is worse than the average one-step na¨ive forecast computed in-sample. Then,

- Mean absolute scaled error (MASE) = $mean(|q_t|)$.

The MASE can be used to compare forecast methods on a single series, and to compare forecast accuracy between series as it is scale-free. It is the only available method which can be used in all circumstances.

Classification of Exponential Smoothing Methods

In exponential smoothing, we represent the trend component as a combination of a level term (ℓ) and a growth term (b).

The level and growth can be combined in several ways, giving five future trend types.

We will define the following trend types or growth patterns:

$$\begin{aligned}
\text{None:} \quad & T_h = \ell \\
\text{Additive:} \quad & T_h = \ell + bh \\
\text{Additive damped:} \quad & T_h = \ell + (\phi + \phi^2 + \cdots + \phi^h)b \\
\text{Multiplicative:} \quad & T_h = \ell b^h \\
\text{Multiplicative damped:} \quad & T_h = \ell b(\phi + \phi^2 + \cdots + \phi^h)
\end{aligned} \tag{2}$$

In the damped trend types, we dampen the trend as the length of the forecast horizon increases to improve the forecast accuracy, particularly at long lead times.

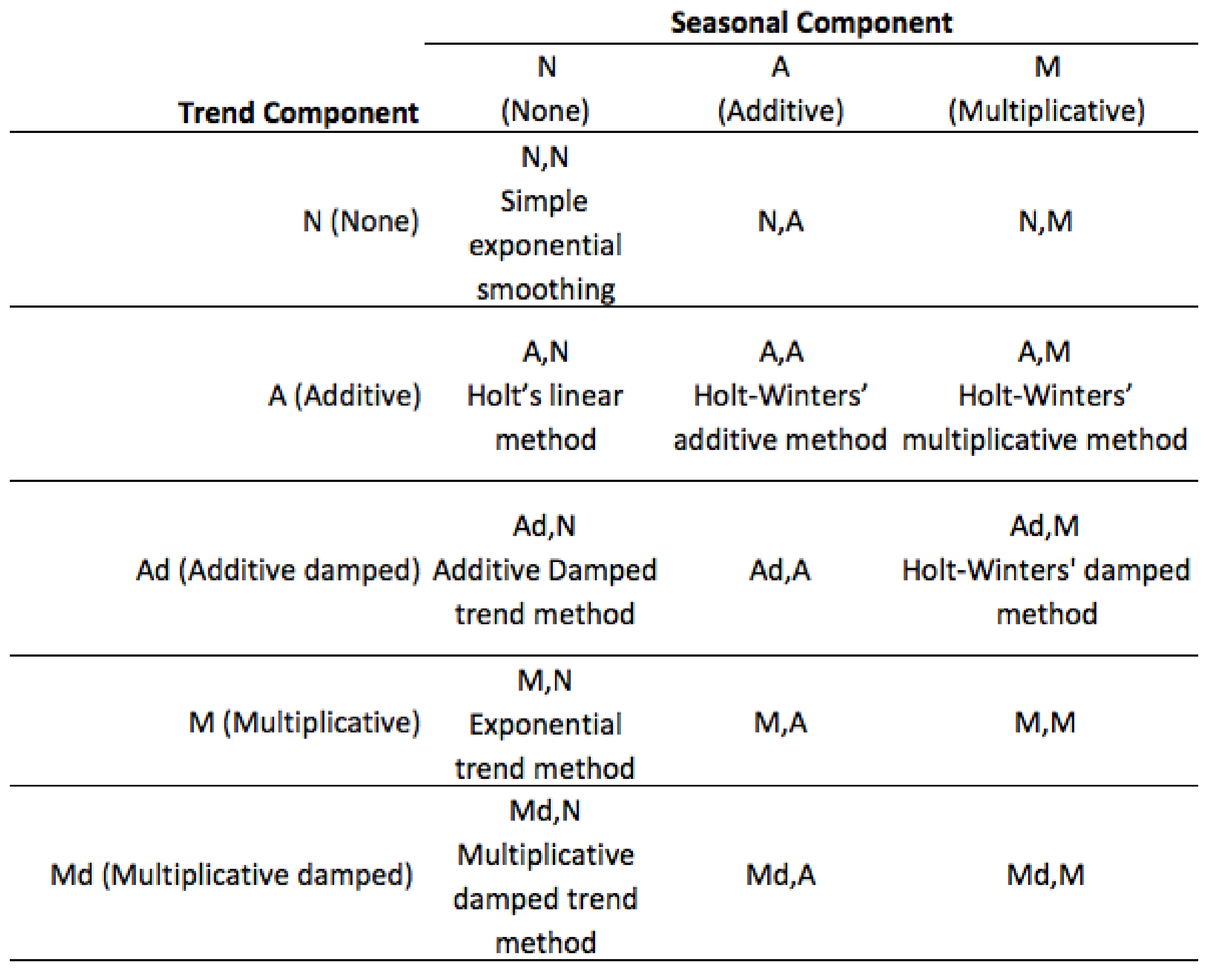
We can also include a seasonal component to the model either additively or multiplicatively.

And we include the error in the same way: either additively or multiplicatively.

In general, the nature of the error component has often been ignored, because the distinction between additive and multiplicative errors makes no difference to point forecasts.

When the type of error component is ignored, we count fifteen exponential smoothing methods given in the following table.

Also, relevant cells include the common cases of the methods.



	Seasonal Component		
	N	A	M
Trend Component	(None)	(Additive)	(Multiplicative)
N (None)	N,N Simple exponential smoothing	N,A	N,M
A (Additive)	A,N Holt's linear method	A,A Holt-Winters' additive method	A,M Holt-Winters' multiplicative method
Ad (Additive damped)	Ad,N Additive Damped trend method	Ad,A	Ad,M Holt-Winters' damped method
M (Multiplicative)	M,N Exponential trend method	M,A	M,M
Md (Multiplicative damped)	Md,N Multiplicative damped trend method	Md,A	Md,M

For each method in the here, there are two possible state-space models.

One is corresponding to a model with additive errors and the other is corresponding to a model with multiplicative errors.

If the same parameter values are used, these two models give equivalent point forecasts although different prediction intervals.

Thus, there are 30 potential models described in this classification.

An exponential smoothing method is an algorithm for producing point forecasts only.

In addition to the point forecasts, stochastic state-space models give a framework for computing prediction intervals and other properties.

Exponential Smoothing Methods

In this section, we will cover simple exponential smoothing (N,N), Holt’s linear method (A,N), the damped trend method (Ad,N) and Holt-Winters’ seasonal method (A,A and A,M).

Suppose that we have given the series $Y_t = Y_1, Y_2, \dots, Y_n$, then a forecast of Y_{t+h} given the observation up to time point t is denoted by $\hat{Y}_{t+h|t}$.

For one-step forecasts, we will use the notation Y_{t+1}.

Simple Exponential Smoothing

The method of simple exponential smoothing basically takes the forecast for the previous period (say for time t, $\hat{Y}_t$) and adjusts it using the forecast error $(Y_t - \hat{Y}_t)$ using the following expression:

$$\hat{Y}_{t+1} = \hat{Y}_t + \alpha(Y_t - \hat{Y}_t), \tag{3}$$

where α is a constant between 0 and 1.

The new forecast is just a function of the old forecast and its error with weight changing according to the value of α.

After simple manipulation we can write Eq. (2) as a **weighted average** of observation and forecast at time point t:

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t . \tag{4}$$

If we replace $\hat{Y}_t$ with its components and expand the Eq. (3), we get

$$\begin{aligned} \hat{Y}_{t+1} \quad &= \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}] \\ &= \alpha Y_t + \alpha(1-\alpha) Y_{t-1} + (1-\alpha)^2 \hat{Y}_{t-1} . \end{aligned} \tag{5}$$

If we keep replacing, we obtain the following general form:

$$
\begin{aligned}
\hat{Y}_{t+1} = \quad & \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} \\
& +\alpha(1-\alpha)^3 Y_{t-3} \\
& +\alpha(1-\alpha)^4 Y_{t-4} + \cdots + \alpha(1-\alpha)^{t-1} Y_1 \\
& +(1-\alpha)^t \hat{Y}_1 .
\end{aligned}
\tag{6}
$$

In this case, $\hat{Y}_{t+1}$ represents a weighted moving average of all past observations with the weights decreasing exponentially; hence, it’s called **exponential smoothing**.

An important problem with these models is called *initialization problem*.

When the value of α is small, the weight of $\hat{Y}_1$ would be quite large and this would cause problems with relatively short time series.

For longer range forecasts, it is assumed that the forecast function is *flat*: $\hat{Y}_{t+h|t} = \hat{Y}_{t+1}$ for $h = 2, 3, 4, \ldots$ because simple exponential smoothing works best for data that have no trend, seasonality, or other underlying patterns.

Notice that we need to specify $\hat{Y}_1$ and α for the simple exponential smoothing.

We will use the function `ses()` from the `forecast` package to fit exponential smoothing models.

The argument `alpha` is used to specify the value of α and if it is not specified α is estimated.

To apply Box-Cox transformation before the model is fitted, we use argument `lambda` which corresponds to the Box-Cox transformation parameter λ.

It is also possible to choose the approach to estimate initial values.

If the argument `initial` is set to `optimal`, the initial values are optimized along with the smoothing parameters.

If it is set to `simple`, the initial values are obtained using simple calculations on the first few observations.

To illustrate the contents, let’s apply the simple exponential smoothing to the annual US new freight cars series observed between 1947 and 1993.

```r
data("freight") # Available in forecast package
fit1 <- ses(freight, alpha=0.1, initial="simple", h=5) # Set alpha to a small value
summary(fit1)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = freight, h = 5, initial = "simple", alpha = 0.1)
##
##   Smoothing parameters:
##     alpha = 0.1
##
##   Initial states:
##     l = 4631.45
##
##   sigma:  1634.026
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -615.3614 1634.026 1339.442 -109.3975 125.4587 1.279911
##                   ACF1
## Training set 0.4752577
##
## Forecasts:
##      Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 1994       1739.251 -354.8369 3833.339 -1463.380 4941.883
## 1995       1739.251 -365.2813 3843.784 -1479.354 4957.856
## 1996       1739.251 -375.6741 3854.177 -1495.248 4973.751
## 1997       1739.251 -386.0161 3864.519 -1511.065 4989.567
## 1998       1739.251 -396.3080 3874.811 -1526.805 5005.307
```

```r
fit2 <- ses(freight, alpha=0.8, initial="simple", h=5) # Set alpha to a large value
summary(fit2)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = freight, h = 5, initial = "simple", alpha = 0.8)
##
##   Smoothing parameters:
##     alpha = 0.8
##
##   Initial states:
##     l = 4631.45
##
##   sigma:  1445.217
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -62.30317 1445.217 1034.509 -27.74406 59.45001 0.9885304
##                   ACF1
## Training set 0.02924579
##
## Forecasts:
##      Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## 1994       2288.851   436.73119 4140.971  -543.7218 5121.424
## 1995       2288.851   -83.01964 4660.722 -1338.6121 5916.314
## 1996       2288.851  -507.78857 5085.491 -1988.2401 6565.942
## 1997       2288.851  -876.05270 5453.755 -2551.4515 7129.153
## 1998       2288.851 -1205.72164 5783.424 -3055.6366 7633.339
```

```r
fit3 <- ses(freight, h=5) # Let the software estimate alpha
summary(fit3)
```

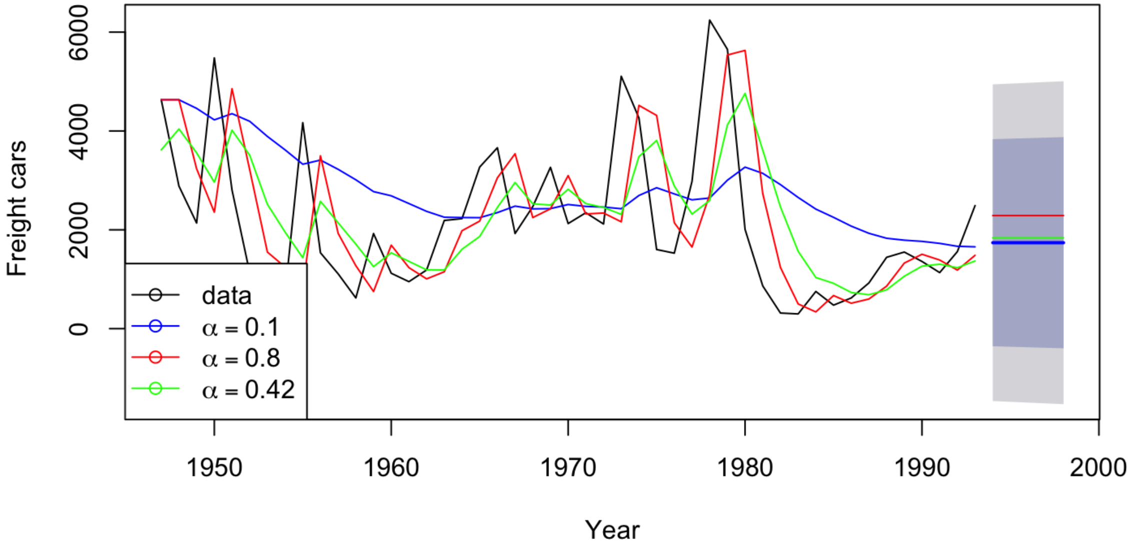
```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = freight, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.4179
##
##   Initial states:
##     l = 3616.2556
##
##   sigma:  1451.738
##
##      AIC     AICc      BIC
## 869.2817 869.8398 874.8321
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -90.54977 1420.514 1123.457 -51.68427 81.36899 1.073525
##                   ACF1
## Training set 0.2185174
##
## Forecasts:
##      Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## 1994       1837.766  -22.71028 3698.243 -1007.587 4683.120
## 1995       1837.766 -178.62927 3854.162 -1246.045 4921.578
## 1996       1837.766 -323.32813 3998.861 -1467.342 5142.875
## 1997       1837.766 -458.92853 4134.461 -1674.725 5350.258
## 1998       1837.766 -586.95743 4262.490 -1870.529 5546.062
```

The output of `ses()` function mainly shows the measures of model fit and the forecasts.

For some of the error measures, the model with $\alpha = 0.8$ gives better results than the one with the estimated α.

The following code chunk displays the fitted models, observed series and forecasts for 5-time points ahead.

```r
#main = "Exponential smoothing fits and forecasts for annual US new freight cars series."
plot(fit1, plot.conf=FALSE, ylab=" Freight cars", xlab="Year", main="",
     type="l")
lines(fitted(fit1), col="blue", type="l")
lines(fitted(fit2), col="red", type="l")
lines(fitted(fit3), col="green", type="l")
lines(fit1$mean, col="blue", type="l")
lines(fit2$mean, col="red", type="l")
lines(fit3$mean, col="green", type="l")
legend("bottomleft",lty=1, col=c(1,"blue","red","green"),
       c("data", expression(alpha == 0.1),
         expression(alpha == 0.8), expression(alpha == 0.42)),pch=1)
```


Freight cars
0
2000
4000
6000
1950
1960
1970
1980
1990
2000
Year
data
α = 0.1
α = 0.8
α = 0.42

Holt’s Linear Method

Holt’s linear method extends simple exponential smoothing to linear exponential smoothing to be able **include trends to the forecasting model**.

In this model, we have two smoothing constants α and β^*. Both constants are in between 0 and 1.

The forecasting model for Holt’s linear method is composed of two components.

$$\begin{aligned} \text{Level:} \quad && \ell_t &= \alpha Y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}), \\ \text{Growth:} \quad && b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}), \\ \text{Forecast:} \quad && \hat{Y}_{t+h|t} &= \ell_t + b_t h. \end{aligned} \tag{7}$$

Here ℓ_t denotes an estimate of the level of the series at time t and b_t denotes an estimate of the slope (or growth) of the series at time t.

The constant b_t is a weighted average of the previous growth b_{t-1} and an estimate of growth based on the difference between successive levels.

When $\alpha = \beta^*$, Holt’s method is equivalent to *Brown’s double exponential smoothing*.

Another special case occurs when $\beta^* = 0$, which makes $b_t = b_{t-1} = b$.

For this case, we have the following forecasting model:

$$\begin{aligned} \text{Level:} \qquad \ell_t &= \alpha Y_t + (1-\alpha)(\ell_{t-1} + b), \\ \text{Forecast:} \qquad \hat{Y}_{t+h|t} &= \ell_t + bh. \end{aligned} \tag{8}$$

This method is known as **simple exponential smoothing with drift**.

Damped Trend Method

This is a modification of Holt's linear method.

The forecasting equations include another parameter ϕ to create a damped trend.

The forecast equations of this model are as follows:

$$\begin{aligned} \text{Level:} \qquad \ell_t &= \alpha Y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}), \\ \text{Growth:} \qquad b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}), \\ \text{Forecast:} \qquad \hat{Y}_{t+h|t} &= \ell_t + (\phi + \phi^2 + \cdots + \phi^h) b_t. \end{aligned} \tag{9}$$

The dampening factor ϕ is in between 0 and 1. When $\phi = 1$, this method gives the same forecasts as Holt's linear method.

For $0 < \phi < 1$, as $h \rightarrow \infty$ the forecasts approach an asymptotic value given by $\ell_t + \phi b_t/(1 - \phi)$.

Exponential Trend Method

A variation from Holt's linear trend method is the one with the level and the slope to be multiplied rather than added Hyndman and Athanasopoulos, 2014:

$$\begin{aligned} \text{Level:} \qquad \ell_t &= \alpha Y_t + (1-\alpha)(\ell_{t-1} b_{t-1}), \\ \text{Growth:} \qquad b_t &= \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}), \\ \text{Forecast:} \qquad \hat{Y}_{t+h|t} &= \ell_t b_t^h. \end{aligned} \tag{10}$$

where b_t now represents an estimated growth rate which is multiplied rather than added to the estimated level.

The trend in the forecast function is now exponential rather than linear so that the forecasts project a constant growth rate rather than a constant slope.

To fit Holt's linear, exponential, and dampened trend models, we will use the function `holt()` from the package `forecast`.

We can fit an exponential trend instead of a linear trend setting by the argument `exponential` to `TRUE`.

Also, we can fit a damped trend by setting `damped = TRUE`. If the arguments `alpha` and `beta` are not specified, they are estimated from the data.

We can apply the Box-Cox transformation before fitting the model by setting the argument `lambda` to the value of λ of the Box-Cox transformation. The `initial` argument specifies the approach to estimates the initial values as before.

To illustrate the implementation of the methods, let’s consider the annual US net electricity generation series.

```r
# Set alpha and beta
fit1 <- holt(usnetelec, alpha=0.8, beta=0.1, initial="simple", h=5)
# Let the software estimate both alpha and beta
fit2 <- holt(usnetelec, initial="simple", h=5)
# Brown’s double exponential smoothing
fit3 <- holt(usnetelec, alpha=0.2, beta=0.2, initial="simple", h=5)
# Simple exponential smoothing with drift
fit4 <- holt(usnetelec,  beta=0, initial="simple", h=5)
# Fit with exponential trend
fit5 <- holt(usnetelec, alpha=0.1, beta=0.8, initial="simple",
             exponential=TRUE, h=5)
# Fit with additive damped trend
fit6 <- holt(usnetelec, alpha=0.8, beta=0.2, damped=TRUE,
             initial="simple", h=5)
```

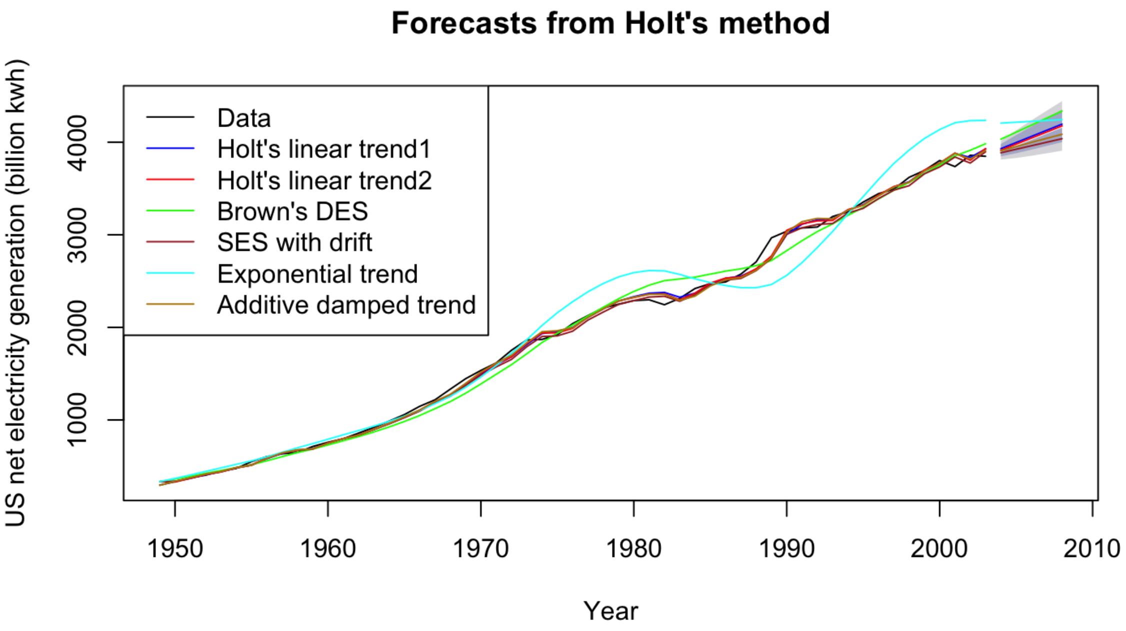
```r
summary(fit2) # Which has the smallest RMSE
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = usnetelec, h = 5, initial = "simple")
##
##   Smoothing parameters:
##     alpha = 1
##     beta  = 0.0875
##
##   Initial states:
##     l = 296.1
##     b = 38
##
##   sigma:  51.6245
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 5.733357 51.62446 37.64055 0.4248986 2.265907 0.5333341
##                    ACF1
## Training set 0.08139712
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2004       3913.583 3847.423 3979.742 3812.401 4014.765
## 2005       3979.165 3881.424 4076.907 3829.683 4128.648
## 2006       4044.748 3919.865 4169.631 3853.755 4235.741
## 2007       4110.331 3960.093 4260.569 3880.561 4340.100
## 2008       4175.913 4001.135 4350.692 3908.613 4443.214
```

We obtained the smallest RMSE with Holt’s linear trend model when the constants are estimated from the data.

The following code chunk displays the fitted models, observed series and forecasts for 5-time points ahead.

```r
plot(fit2, type="l", ylab="US net electricity generation (billion kwh)", xlab="Year",
     fcol="white", plot.conf=FALSE)
lines(fitted(fit1), col="blue")
lines(fitted(fit2), col="red")
lines(fitted(fit3), col="green")
lines(fitted(fit4), col="brown")
lines(fitted(fit5), col="cyan")
lines(fitted(fit6), col="darkgoldenrod")
lines(fit1$mean, col="blue", type="l")
lines(fit2$mean, col="red", type="l")
lines(fit3$mean, col="green", type="l")
lines(fit4$mean, col="brown", type="l")
lines(fit5$mean, col="cyan", type="l")
lines(fit6$mean, col="darkgoldenrod", type="l")
legend("topleft", lty=1, col=c("black","blue","red","green","brown",
                               "cyan","darkgoldenrod"),
       c("Data","Holt's linear trend1","Holt's linear trend2", "Brown's DES",
         "SES with drift" , "Exponential trend","Additive damped trend"))
```


Forecasts from Holt's method
US net electricity generation (billion kwh)
1000
2000
3000
4000
1950
1960
1970
1980
1990
2000
2010
Year
Data
Holt's linear trend1
Holt's linear trend2
Brown's DES
SES with drift
Exponential trend
Additive damped trend

Holt-Winters’ Trend and Seasonality Method

If there is a seasonality exists in the series, Holt’s-Winter method provides a solution to the model fitting problem. Holt-Winters’ method has the smoothing equations for the level, for trend, and for seasonality.

There are two different Holt-Winters’ methods, depending on whether seasonality is modelled in an additive or multiplicative way.

Multiplicative seasonality case

The basic equations for Holt-Winters' multiplicative method are as follows:

$$\begin{aligned}
\text{Level:} \qquad & \ell_t && = \alpha \frac{Y_t}{s_{t-m}} \\
& && +(1-\alpha)(\ell_{t-1} + b_{t-1}), \\
\text{Growth:} \qquad & b_t && = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}), \\
\text{Seasonal:} \qquad & S_t && = \gamma Y_t/(\ell_{t-1} + b_{t-1}) + (1-\gamma)s_{t-m} \\
& && +(1-\gamma)s_{t-m} \\
\text{Forecast:} \qquad & \hat{Y}_{t+h|t} && = (\ell_t + b_t h)s_{t-m+h_m^+},
\end{aligned} \tag{11}$$

Here m is the length of seasonality like the number of months or quarters in a year, ℓ_t represents the level of the series, b_t denotes the growth, s_t is the seasonal component, $\hat{Y}_{t+h|t}$ is the forecast for h periods ahead, and $h_m^+ = [(h-1) \mod m] + 1$. The parameters α, β^*, and γ are all restricted in between 0 and 1.

Additive seasonality case

The seasonal component in Holt-Winters' method may also be treated additively, although in practice this seems to be less commonly used.

The basic equations for Holt-Winters' additive method are as follows:

$$\begin{aligned}
\text{Level:} \quad && \ell_t &= \alpha(Y_t - s_{t-m}) \\
&& &+(1-\alpha)(\ell_{t-1} + b_{t-1}), \\
\text{Growth:} \quad && b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}), \\
\text{Seasonal:} \quad && S_t &= \gamma(Y_t - \ell_{t-1} - b_{t-1}) \\
\text{Forecast:} \quad && \hat{Y}_{t+h|t} &= (\ell_t + b_t h)s_{t-m+h_m^+},
\end{aligned} \tag{12}$$

The only differences in the other equations are that the seasonal indices are now added and subtracted instead of taking products and ratios.

To implement the Holt-Winters method, we will use `hw()` function from the package `forecast`.

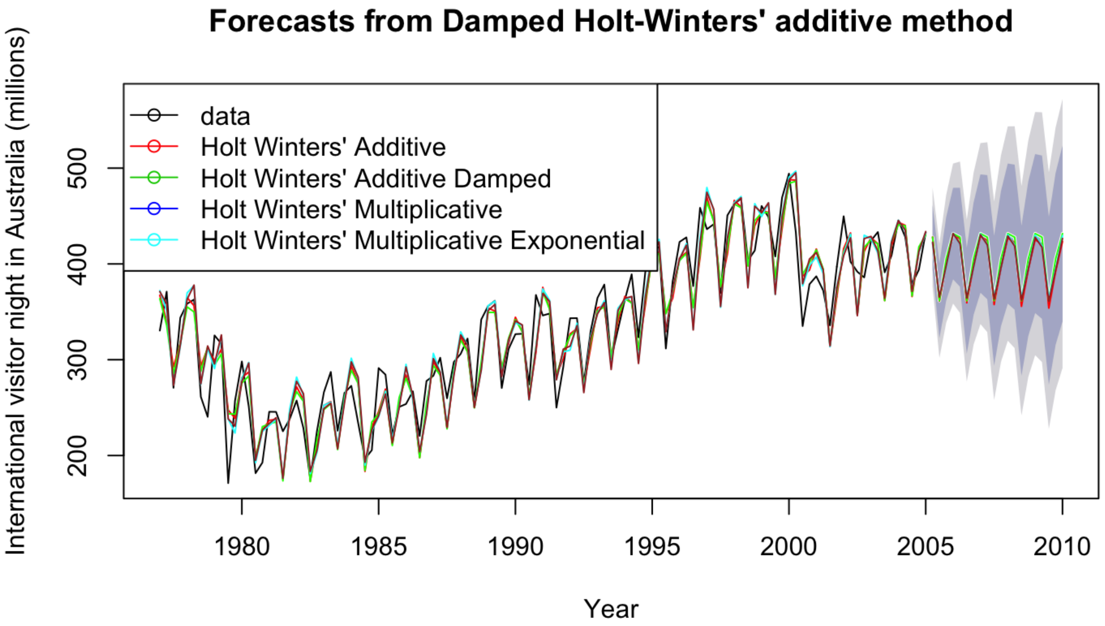
In addition to the parameter explained for `ses()` and `holt()` functions, we use the argument `seasonal=c("additive","multiplicative")` to set the type of model.

Let's revisit the quarterly motor vehicle production in the UK (thousands of cars) for the first quarter of 1977 through the first quarter of 2005 and fit various Holt-Winter models using the following code chunk.

```r
data("ukcars")
fit1 <- hw(ukcars,seasonal="additive", h=5*frequency(ukcars))
fit2 <- hw(ukcars,seasonal="additive",damped = TRUE, h=5*frequency(ukcars))
fit3 <- hw(ukcars,seasonal="multiplicative", h=5*frequency(ukcars))
fit4 <- hw(ukcars,seasonal="multiplicative",exponential = TRUE,
           h=5*frequency(ukcars))
```

The additive model with damped trend gives the smallest RMSE value within the set of considered models. The following code chunk displays the fitted models, observed series and forecasts for 5-time points ahead.

```
plot(fit2,ylab="International visitor night in Australia (millions)",
     plot.conf=FALSE, type="l", fcol="white", xlab="Year")
lines(fitted(fit1), col="red", lty=1)
lines(fitted(fit2), col="green", lty=1)
lines(fitted(fit3), col="cyan", lty=1)
lines(fitted(fit4), col="brown", lty=1)
lines(fit1$mean, type="l", col="red")
lines(fit2$mean, type="l", col="green")
lines(fit3$mean, type="l", col="cyan")
lines(fit4$mean, type="l", col="brown")
legend("topleft",lty=1, pch=1, col=1:5,
       c("data","Holt Winters' Additive", "Holt Winters' Additive Damped",
         "Holt Winters' Multiplicative", "Holt Winters' Multiplicative Exponential"))
```

Forecasts from Damped Holt-Winters' additive method
data
Holt Winters' Additive
Holt Winters' Additive Damped
Holt Winters' Multiplicative
Holt Winters' Multiplicative Exponential
International visitor night in Australia (millions)
200
300
400
500
1980
1985
1990
1995
2000
2005
2010
Year

For all the methods discussed above, we can get some special cases for extreme values of parameters. For example,

- if $\alpha = 0$, the level is constant over time,
- if $\beta^* = 0$, the slope is constant over time, and
- if $\gamma = 0$, the seasonal pattern is constant over time,
- if $\alpha = 1$, naive forecasts are obtained using the simple exponential smoothing method, and
- the additive and multiplicative trend methods are special cases of their damped counterparts obtained by letting $\phi = 1$.

State Space Models

State-space models allow considerable flexibility in the specification of the parametric structure. The textbook uses the *innovations* formulation of the model. Let Y_t denote the observation at time t, and let $\boldsymbol{X}_t$ be a **state vector** containing unobserved components that describe the level, trend and seasonality of the series. Then a linear innovations state-space model is written as

$$\begin{aligned} Y_t &= \boldsymbol{\omega}'\boldsymbol{X}_{t-1} + \epsilon_t, \\ \boldsymbol{X}_t &= \boldsymbol{F}\boldsymbol{X}_{t-1} + \boldsymbol{g}\epsilon_t, \end{aligned} \tag{13}$$

where $\{\epsilon_t\}$ is a white noise series and $\boldsymbol{F}, \boldsymbol{g}$, and $\boldsymbol{\omega}$ are coefficients.

The first line of Eq. (13) is known as the *measurement* (or observation) equation; it describes the relationship between the unobserved states $\boldsymbol{X}_{t-1}$ and the observation Y_t.

The second line is known as the *transition* (or state) equation; it describes the evolution of the states over time.

The use of identical errors (or innovations) in these two equations makes it an **innovations** state-space model.

Nonlinear state-space models are also possible. One form that we will use is

$$\begin{aligned} Y_t &= w(\boldsymbol{X}_{t-1}) + r(\boldsymbol{X}_{t-1})\epsilon_t, \\ \boldsymbol{X}_t &= f(\boldsymbol{X}_{t-1}) + g(\boldsymbol{X}_{t-1})\epsilon_t. \end{aligned} \tag{14}$$

where $\{\epsilon_t\}$ is a white noise process with $N(o, \sigma^2)$.

For each exponential smoothing method, there are two corresponding state-space models according to their error terms (one additive and one multiplicative).

When the same parameters values are used the point forecasts for both types of model are identical.

We will use the triplet to denote the state-space model correspondents of exponential smoothing models.

Each triplet will include an additional letter either A or M to denote the type of error terms and the remaining two words will have the same interpretation as those in Table 1.

The triplet (E,T,S) refers to the three components: error, trend and seasonality.

For example, ETS(A,A,N) means that the corresponding model has additive errors, additive trend and no seasonality.

From Table 1, this is Holt’s linear method with additive errors. ETS(M,Md,M) refers to a model with multiplicative errors, a damped multiplicative trend and multiplicative seasonality.

ETS can also be considered an abbreviation of **ExponenTial Smoothing**.

We denote the conditional mean of a future observation given knowledge of the past by $E(Y_{t+h}|X_t)$, where X_t contains the unobserved components such as ℓ_t, b_t, and s_t.

For $h = 1$, we will use $\mu_{t+1} = \mu_{t+1|t}$. For most of the models, $\mu_{t+h|t} = \hat{Y}_{t+h|t}$.

However, for the models with the multiplicative trend or multiplicative seasonality, the conditional mean and the point forecast will differ slightly for $h \geq 2$.

Now, we will proceed onto the specific state-space models corresponding to the exponential smoothing models.

State-space models for Holt's linear method

We will illustrate the additive and multiplicative error models.

Additive error model: ETS(A,A,N)

Let $\hat{\mu}_t = \hat{Y}_t = \ell_{t-1} + b_{t-1}$ denote he one-step forecast of Y_t assuming we know the values of all parameters. And let $\epsilon_t = Y_t - \mu_t$ denote one-step ahead forecast error at time t. We have

$$Y_t = \ell_{t-1} + b_{t-1} + \epsilon_t. \tag{15}$$

Using the level and growth equations of Eq. (6), we get

$$\begin{aligned} \ell_t &= \ell_{t-1} + b_{t-1} + \alpha\epsilon_t, \\ b_t &= b_{t-1} + \beta^*(\ell_t - \ell_{t-1} - b_{t-1}) + \alpha\beta^*\epsilon_t. \end{aligned} \tag{16}$$

If we simplify the last equation in (15) by setting $\beta = \alpha\beta^*$, equations in (14) and (15) constitute a state-space model underlying Holt's method.

We can write it in standard state space formulation state vector as $\boldsymbol{X}_t = (\ell_t, b_t)'$ such as

$$\begin{aligned} Y_t &= \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{X}_{t-1} + \epsilon_t \\ \boldsymbol{X}_t &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{X}_{t-1} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \epsilon_t. \end{aligned} \tag{17}$$

The model is fully specified once we state the distribution of the error term ϵ_t. Usually we assume that $\epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

Multiplicative error model: ETS(M,A,N)

We will a similar way to derive the model with multiplicative error terms. First we set $\epsilon_t = (Y_t - \mu_t)/\mu_t$ to define ϵ_t as a relative error. Then, we find

$$\begin{aligned} Y_t &= (\ell_{t-1} + b_{t-1})(1 + \epsilon_t) \\ \ell_t &= (\ell_{t-1} + b_{t-1})(1 + \alpha\epsilon_t), \\ b_t &= b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\epsilon_t, \end{aligned} \tag{18}$$

or in a state space model form

$$\begin{aligned} Y_t &= \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{X}_{t-1}(1+\epsilon_t), \\ \boldsymbol{X}_t &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{X}_{t-1} + \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{X}_{t-1} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \epsilon_t. \end{aligned} \tag{19}$$

Again, we have the assumption $\epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

This kind of nonlinear state-space models are usually considered difficult to handle in estimating and forecasting.

However, we can still compute forecasts, the likelihood and prediction intervals for this nonlinear model with no more effort than is required for the additive error model.

We use the function `ets()` from the `forecast` package to fit state-space models corresponding to the exponential smoothing methods.

We need to specify the considered model using the argument `model`, which get a three-character string.

The characters and their meanings are `"N"=none`, `"A"=additive`, `"M"=multiplicative`, and `"Z"=automatically`.

The first letter denotes the error type ("A", "M" or "Z"); the second letter denotes the trend type ("N","A","M" or "Z"); and the third letter denotes the season type ("N","A","M" or "Z").

So, for example, "ANN" is a simple exponential smoothing with additive errors, "MAM" is multiplicative Holt-Winters' method with multiplicative errors, and so on.

The arguments `alpha`, `beta`, and `gamma` are for the corresponding constants and it is possible to apply a Box-Cox transformation using the argument `lambda` before fitting the model. There are also some other useful arguments with this function.

Let’s illustrate the application of the models over the annual US net electricity generation series, which we have fitted Holt’s linear method.

```r
fit.AAN = ets(usnetelec, model="AAN")
summary(fit.AAN)
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = usnetelec, model = "AAN")
##
##   Smoothing parameters:
##     alpha = 0.9985
##     beta  = 1e-04
##
##   Initial states:
##     l = 244.2092
##     b = 65.5768
##
##   sigma:  51.8636
##
##      AIC     AICc      BIC
## 660.5982 661.8227 670.6349
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.04060982 49.94203 36.44997 -0.8111995 2.440206 0.5164647
##                   ACF1
## Training set 0.1263413
```

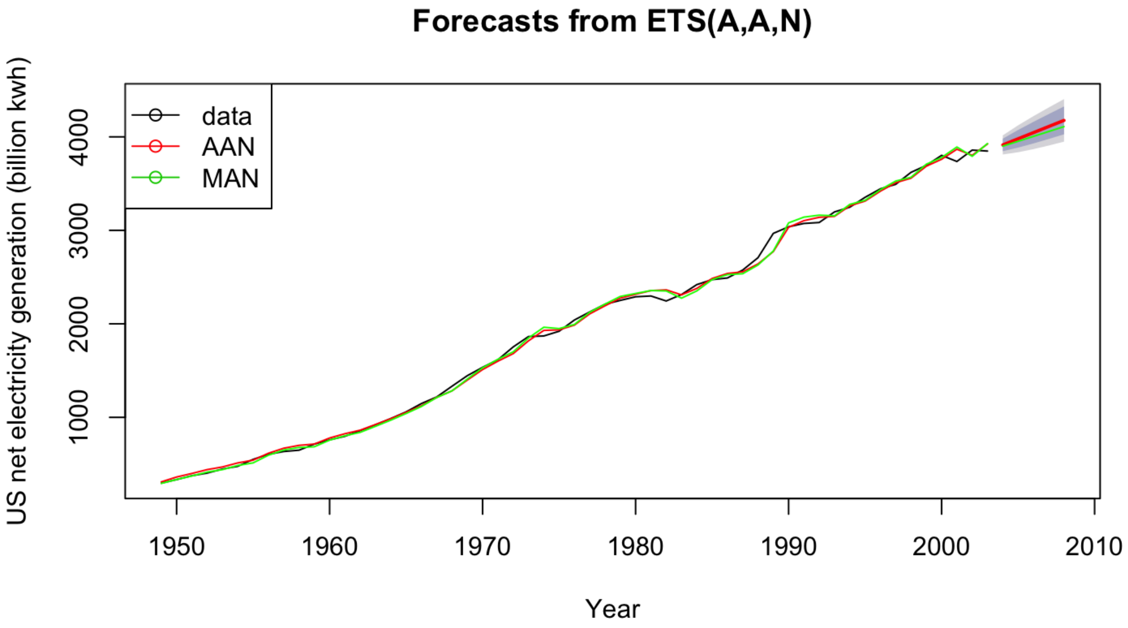
```r
fit.MAN = ets(usnetelec, model="MAN")
summary(fit.MAN)
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = usnetelec, model = "MAN")
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.2191
##
##   Initial states:
##     l = 254.9338
##     b = 38.3125
##
##   sigma:  0.0259
##
##      AIC     AICc      BIC
## 634.0437 635.2682 644.0803
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 1.162583 52.00363 36.77721 0.2629582 1.942062 0.5211014
##                     ACF1
## Training set 0.006113498
```

The model with additive error terms gives better fits and forecasts in terms of AIC and RMSE.

The following display shows the original series, fits and forecasts from both models.

```r
frc.AAN = forecast(fit.AAN, h =5) # Produce forecasts for AAN model
frc.MAN = forecast(fit.MAN, h =5) # Produce forecasts for MAN model
plot(frc.AAN, ylab="US net electricity generation (billion kwh)",
     plot.conf=FALSE, type="l", fcol="red", xlab="Year")
lines(fitted(fit.AAN), col="red", lty=1)
lines(fitted(fit.MAN), col="green", lty=1)
lines(frc.MAN$mean,col="green", type="l")
legend("topleft",lty=1, pch=1, col=1:3, c("data","AAN", "MAN"))
```


Forecasts from ETS(A,A,N)
data
AAN
MAN
US net electricity generation (billion kwh)
1000
2000
3000
4000
1950
1960
1970
1980
1990
2000
2010
Year

As another illustration, let’s focus on the quarterly motor vehicle production in the UK (thousands of cars) series, which we have fitted a bunch of different seasonal models with different trend structures.

Let’s fit Holt-Winters’ additive method with damped trend and additive and multiplicative errors using a state-space approach.

Recall that the additive model with damped trend was giving the smallest RMSE value for this series.

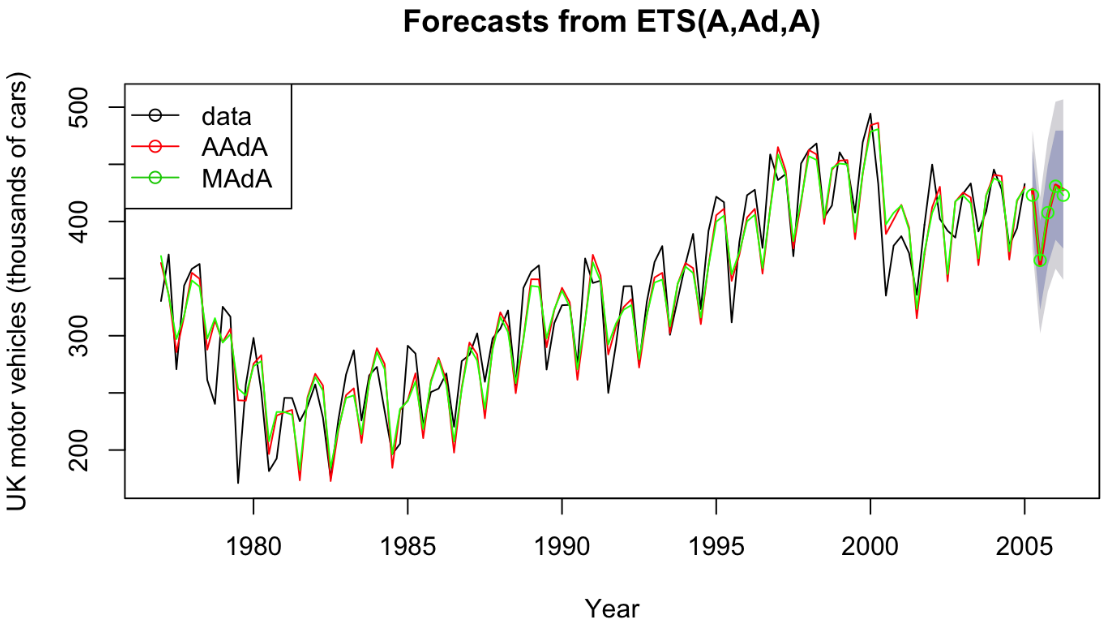
```r
fit.AAdA = ets(ukcars, model = "AAA", damped = TRUE)
summary(fit.AAdA)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = ukcars, model = "AAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.5814
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9284
##
##   Initial states:
##     l = 343.6012
##     b = -5.3444
##     s = -1.1652 -45.1153 21.2507 25.0298
##
##   sigma:  26.2512
##
##      AIC     AICc      BIC
## 1283.319 1285.476 1310.593
##
## Training set error measures:
##                    ME     RMSE      MAE     MPE     MAPE      MASE
## Training set 2.009896 25.18409 20.44382 0.10939 6.683841 0.6662543
##                    ACF1
## Training set 0.03323651
```

```r
fit.MAdA = ets(ukcars, model = "MAA", damped = TRUE)
summary(fit.MAdA)
```

```
## ETS(M,Ad,A)
##
## Call:
##  ets(y = ukcars, model = "MAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.5271
##     beta  = 1e-04
##     gamma = 0.0142
##     phi   = 0.8859
##
##   Initial states:
##     l = 353.5972
##     b = -7.1641
##     s = 1.1374 -37.8311 14.1752 22.5184
##
##   sigma:  0.0895
##
##      AIC     AICc      BIC
## 1305.171 1307.328 1332.445
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.812806 26.02203 20.95131 -0.1646796 6.833628 0.6827932
##                     ACF1
## Training set 0.06436705
```

```
frc.AAdA = forecast(fit.AAdA, h =5)
frc.MAdA = forecast(fit.MAdA, h =5)
plot(frc.AAdA, ylab="UK motor vehicles (thousands of cars)",
     plot.conf=FALSE, type="l", fcol="red", xlab="Year")
lines(fitted(fit.AAdA), col="red", lty=1)
lines(fitted(fit.MAdA), col="green", lty=1)
lines(frc.MAdA$mean,col="green", type="o")
legend("topleft",lty=1, pch=1, col=1:3, c("data","AAdA ", "MAdA "))
```



We get the better fit and forecasts from additive damped trend Holt-Winters’ model with the additive errors state-space model.

Practical applications

New Zealand annual number of births

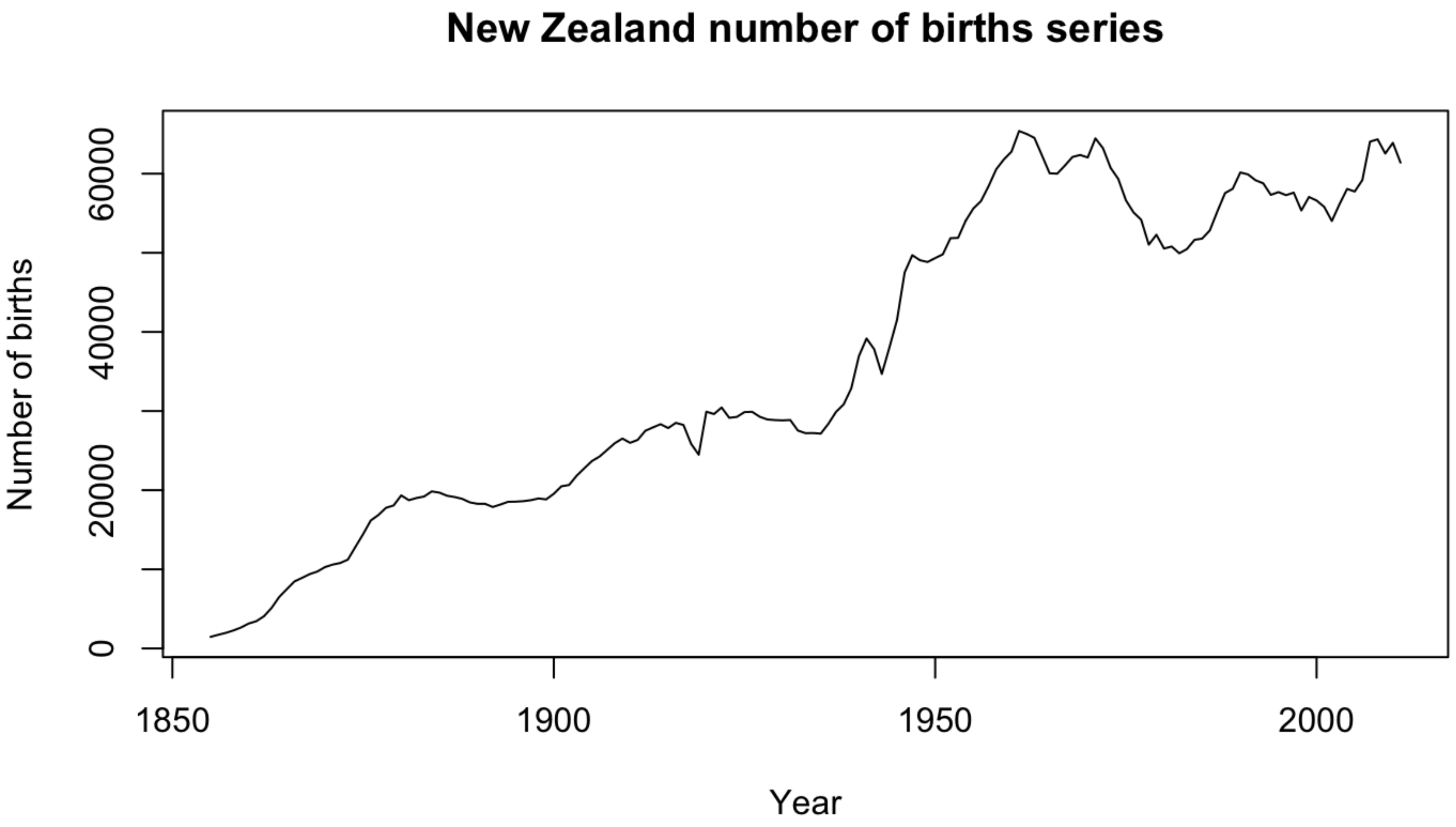
In the first application, we will focus on the annual number of births in New Zealand between 1855 and 2011.

First, we will read and convert the data set into `ts` object.

```
birth <- read.csv("~/Documents/MATH1307_Forecasting/presentations/Module 5/birth.csv")
birth = ts(birth[,"BirthNum"], start = 1855)
```

The time series plot of data is as follows:

```
plot(birth, ylab="Number of births", xlab = "Year", main = "New Zealand number of births series")
```

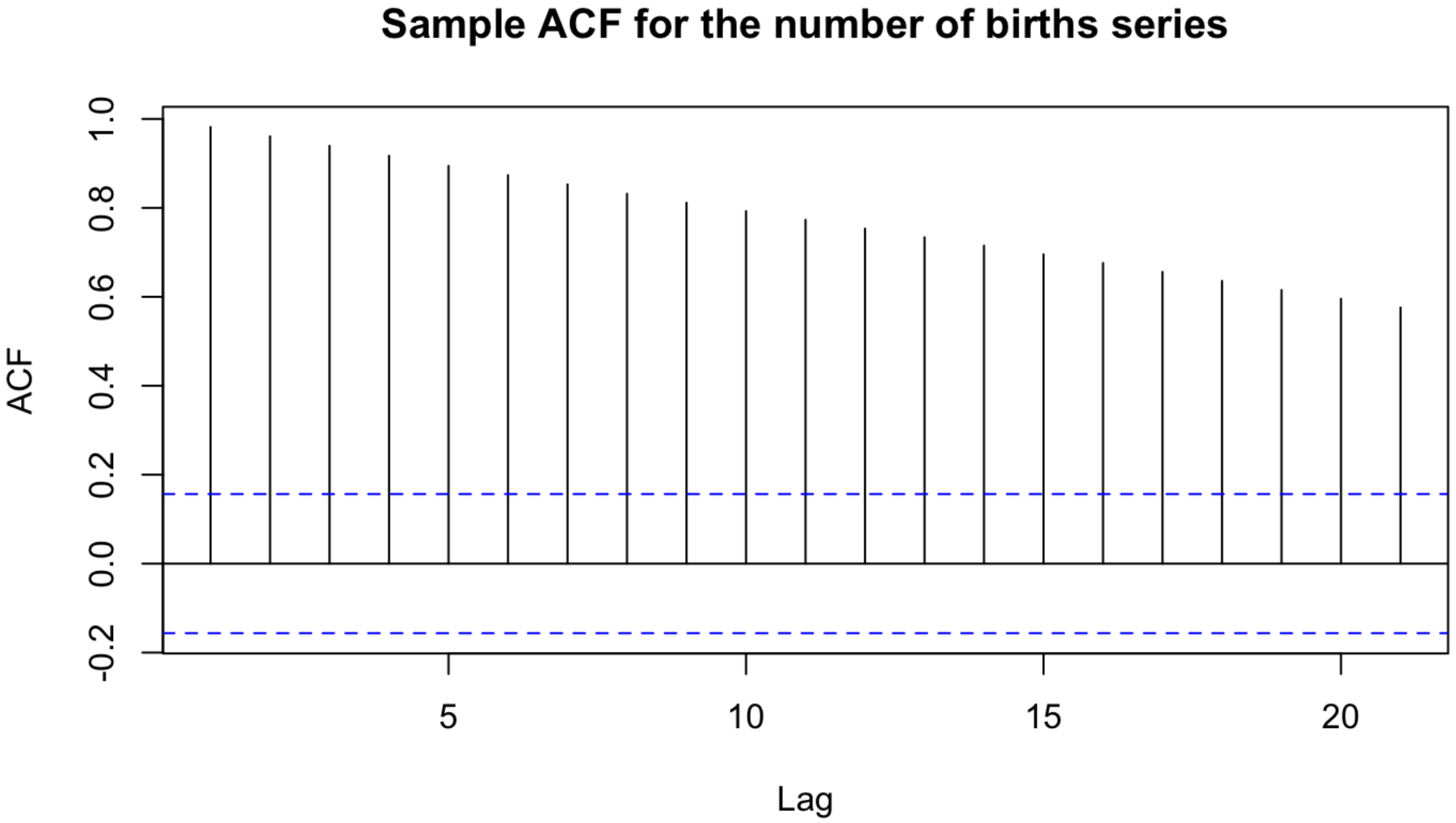


We can infer the following from the time series plot:

- There is an obvious trend in the series.
- There is no sign of seasonality.
- There is not an apparent changing variance.
- Because of the succeeding time points, we can consider an autoregressive behaviour for this series.
- There is no obvious sign of an intervention.

Next, let’s display the sample ACF function.

```
acf(birth, main = "Sample ACF for the number of births series")
```



From the ACF plot, we can observe the slowly decaying pattern implying the existence of a trend in the series.

Considering the existence of the trend, we can fit

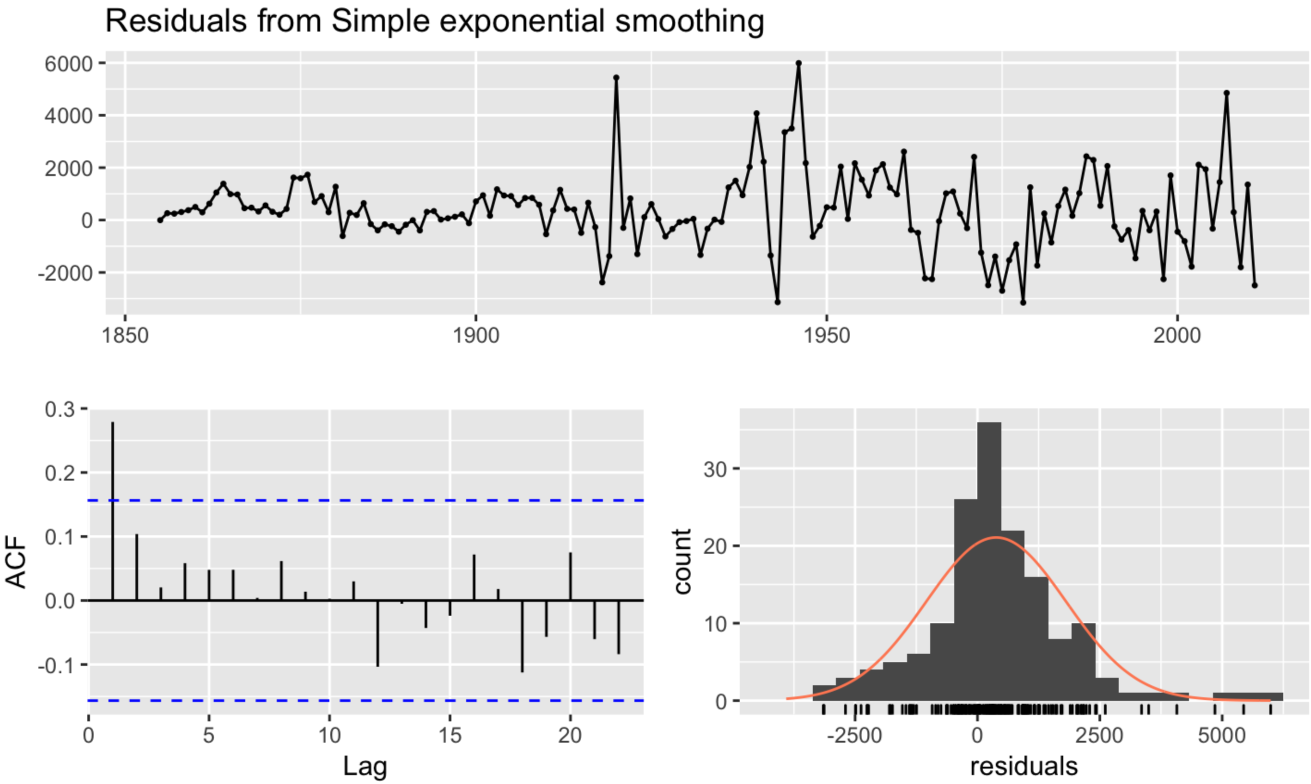
- simple exponential smoothing and
- Holt’s
  - linear,
  - damped, and
  - exponential trend models

to this series.

```r
#Simple exponential smoothing method
fit1.ses <- ses(birth, h=5)
summary(fit1.ses)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = birth, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 1463.9233
##
##   sigma:  1484.906
##
##      AIC     AICc      BIC
## 3090.993 3091.150 3100.162
##
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 381.8173 1475.418 1048.896 2.207143 3.800072 0.9936889
##                  ACF1
## Training set 0.278954
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2012       61403.25 59500.27 63306.23 58492.89 64313.61
## 2013       61403.25 58712.16 64094.34 57287.58 65518.92
## 2014       61403.25 58107.40 64699.09 56362.69 66443.81
## 2015       61403.25 57597.57 65208.93 55582.96 67223.54
## 2016       61403.25 57148.39 65658.11 54896.00 67910.50
```

```
checkresiduals(fit1.ses)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Simple exponential smoothing
## Q* = 16.24, df = 8, p-value = 0.03907
##
## Model df: 2.   Total lags used: 10
```

MASE of this model is obtained as 0.99.

Because of the nature of the model all forecast are equal to each other.

There is a changing variance in the residuals and hence the histogram of residuals has a long tail on both sides.

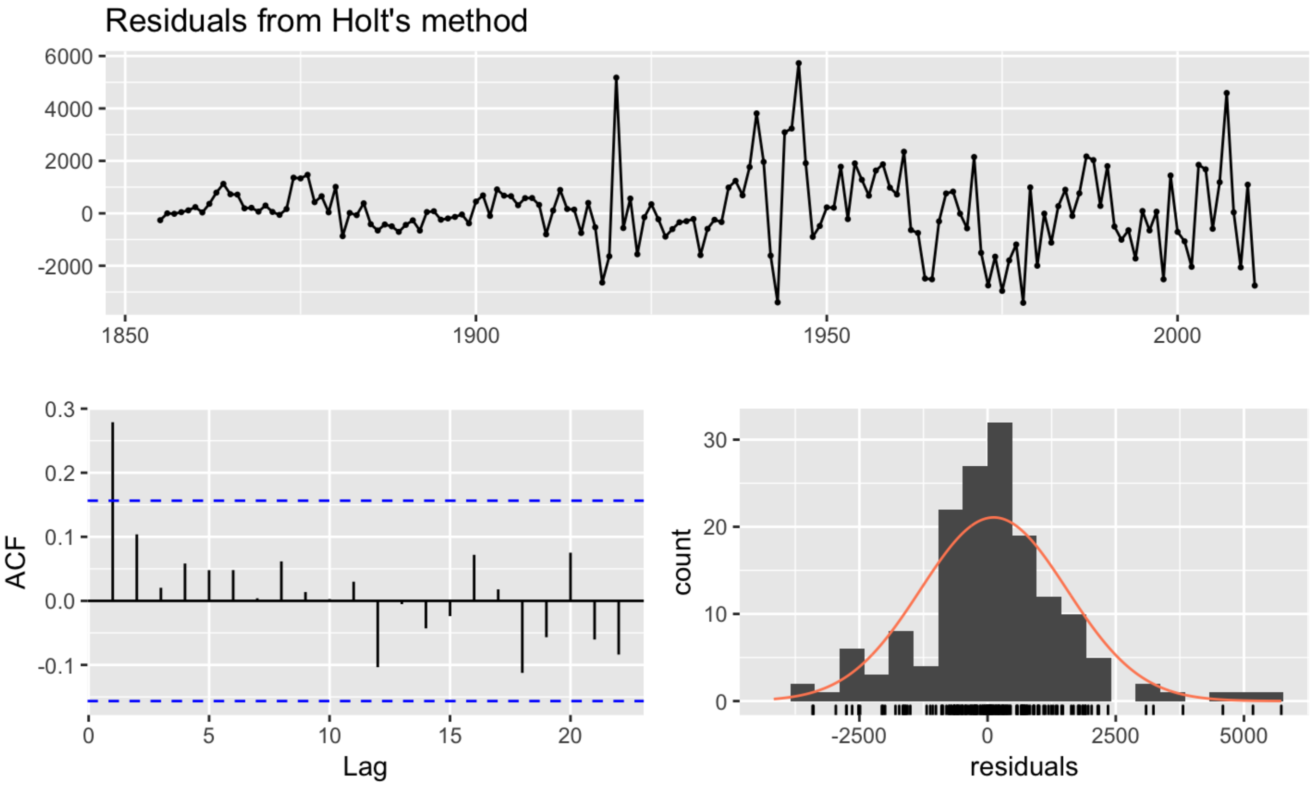
ACF of residuals shows the existence of a very significant first lag correlation. This implies that there is still uncaptured serial correlation left in the residuals.

Let’s go on with Holt’s models.

```r
fit1.holt <- holt(birth, initial="simple", h=5)
summary(fit1.holt)
```

```
## 
## Forecast method: Holt's method
## 
## Model Information:
## Holt's method 
## 
## Call:
##  holt(y = birth, h = 5, initial = "simple") 
## 
##   Smoothing parameters:
##     alpha = 1 
##     beta  = 0 
## 
##   Initial states:
##     l = 1460 
##     b = 262 
## 
##   sigma:  1430.138
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 119.8025 1430.138 999.8408 0.7065726 3.176657 0.9472156
##                   ACF1
## Training set 0.2788627
## 
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2012          61665 59832.20 63497.80 58861.98 64468.02
## 2013          61927 59335.04 64518.96 57962.93 65891.07
## 2014          62189 59014.51 65363.49 57334.03 67043.97
## 2015          62451 58785.41 66116.59 56844.96 68057.04
## 2016          62713 58614.75 66811.25 56445.26 68980.74
```

```
checkresiduals(fit1.holt)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 16.231, df = 6, p-value = 0.01257
##
## Model df: 4.   Total lags used: 10
```

MASE of this model is decreased to 0.93.

Now, we have forecasts reflecting the trend behaviour in the series with an increasing pattern.

There is still significant changing variance in the residuals and hence the histogram of residuals has a long tail on both sides.

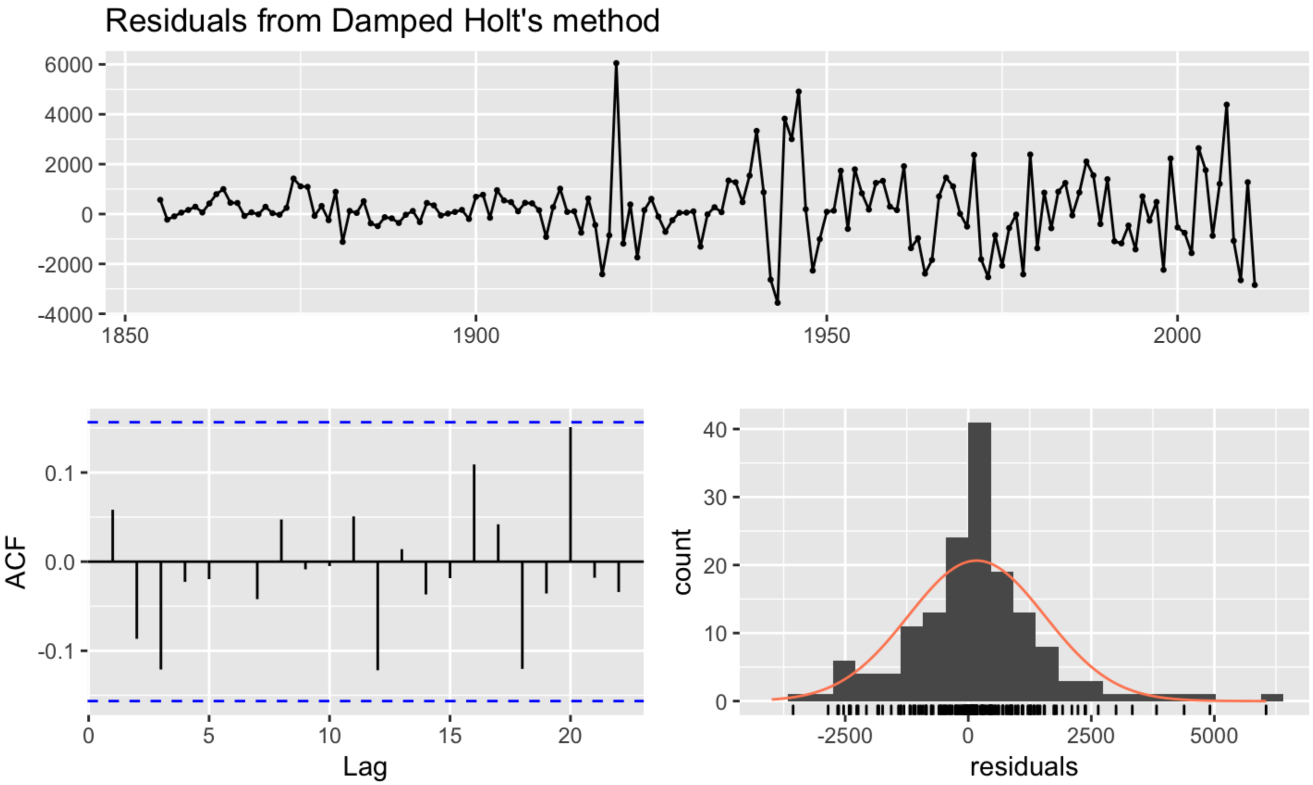
ACF of residuals shows the existence of a slightly significant first lag correlation and an insignificant Bg test result. This implies that there is no uncaptured serial correlation left in the residuals.

Now, we will fit the Holt’s models with the damped trend.

```r
fit2.holt <- holt(birth, initial="simple", h=5 , damped = TRUE)
summary(fit2.holt)
```

```
## 
## Forecast method: Damped Holt's method
## 
## Model Information:
## Damped Holt's method 
## 
## Call:
##  holt(y = birth, h = 5, damped = TRUE, initial = "simple") 
## 
##   Smoothing parameters:
##     alpha = 0.9999 
##     beta  = 0.2851 
##     phi   = 0.8 
## 
##   Initial states:
##     l = 446.2892 
##     b = 556.721 
## 
##   sigma:  1416.325
## 
##      AIC     AICc      BIC 
## 3079.077 3079.637 3097.414 
## 
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 169.3208 1393.589 953.9268 1.106952 3.216281 0.9037183
##                    ACF1
## Training set 0.05820576
## 
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2012       61036.45 59221.35 62851.54 58260.50 63812.39
## 2013       60742.98 57868.52 63617.43 56346.87 65139.08
## 2014       60508.20 56658.98 64357.42 54621.32 66395.07
## 2015       60320.38 55545.73 65095.02 53018.19 67622.56
## 2016       60170.12 54511.46 65828.78 51515.94 68824.30
```

```
checkresiduals(fit2.holt)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt's method
## Q* = 4.9558, df = 5, p-value = 0.4213
##
## Model df: 5.   Total lags used: 10
```

MASE of this model is decreased to 0.90.

The forecasts arising from the damped trend model tend to be steady around a mean level.

There is still significant changing variance in the residuals and hence the histogram of residuals has a long tail on both sides.

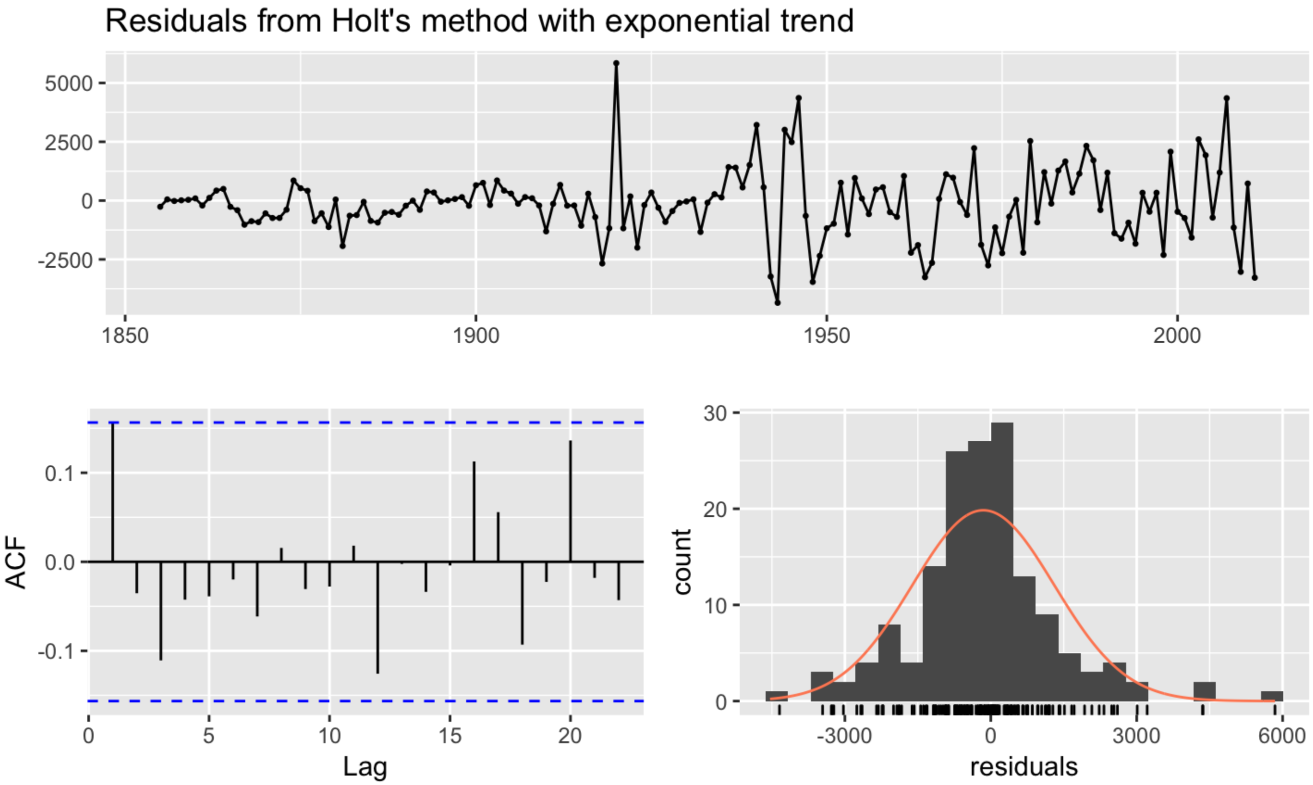
There is no significant serial correlation in ACF and the Bg test supports this result.

Next, we will fit the Holt’s models with the exponential trend.

```r
fit3.holt <- holt(birth, initial="simple", h=5 , exponential = TRUE)
summary(fit3.holt)
```

```
## 
## Forecast method: Holt's method with exponential trend
## 
## Model Information:
## Holt's method with exponential trend 
## 
## Call:
##  holt(y = birth, h = 5, initial = "simple", exponential = TRUE) 
## 
##   Smoothing parameters:
##     alpha = 1 
##     beta  = 0.1926 
## 
##   Initial states:
##     l = 1460 
##     b = 1.1795 
## 
##   sigma:  0.045
## Error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -153.8011 1464.987 1030.702 -0.7196593 3.227847 0.9764525
##                   ACF1
## Training set 0.1567659
## 
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2012       61550.21 58003.21 65077.46 55996.67 67055.97
## 2013       61697.78 56244.51 67428.16 53469.51 70346.86
## 2014       61845.70 54603.61 69415.71 51268.62 73468.61
## 2015       61993.97 52973.42 71827.65 48781.46 77419.06
## 2016       62142.60 51295.35 74171.98 46260.39 81138.34
```

```
checkresiduals(fit3.holt)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method with exponential trend
## Q* = 7.6855, df = 6, p-value = 0.2621
##
## Model df: 4.   Total lags used: 10
```

With the exponential model, MASE is increased to 0.97.

We have very similar residual diagnostics as the Holt’s damped trend model.

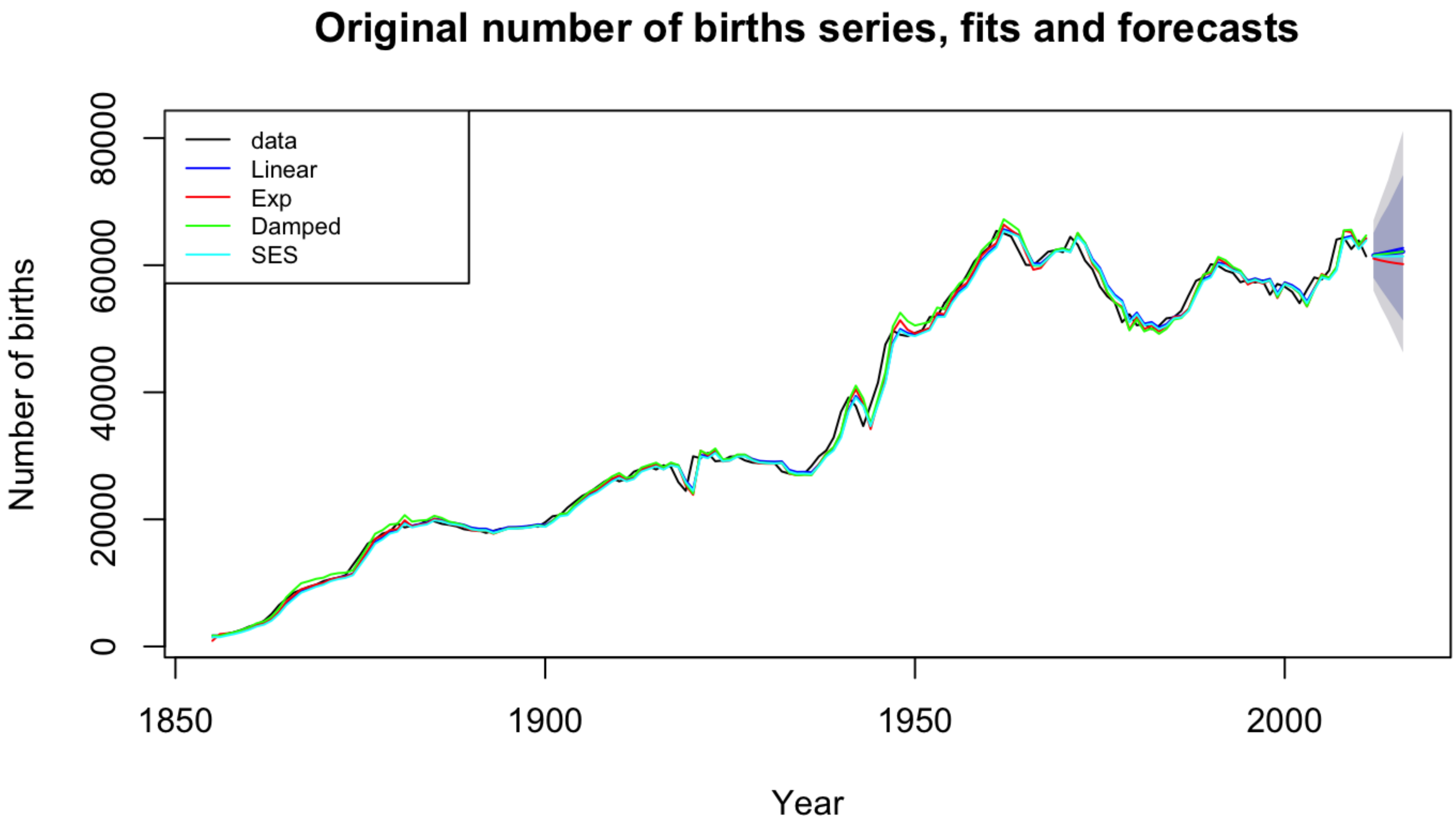
The forecasts show a slowly increasing trend.

Next plot shows the forecasts from these models.

Because the smallest MASE comes from the Holt’s damped trend model, forecast from this model can be used for further purposes.

These forecasts are showing that the birth rate in New Zealand will go steady for the near future.

```
plot(fit3.holt, plot.conf=FALSE, ylab="Number of births",
     xlab="Year", main="", type="l")
lines(fitted(fit1.holt), col="blue", type="l")
lines(fitted(fit2.holt), col="red", type="l")
lines(fitted(fit3.holt), col="green", type="l")
lines(fitted(fit1.ses), col="cyan", type="l")
lines(fit1.holt$mean, col="blue", type="l")
lines(fit2.holt$mean, col="red", type="l")
lines(fit3.holt$mean, col="green", type="l")
lines(fit1.ses$mean, col="cyan", type="l")
legend("topleft",lty=1, text.width = 28, cex = 0.70,
       col=c(1,"blue","red","green","cyan"),
       c("data", "Linear", "Exp", "Damped", "SES"))
```

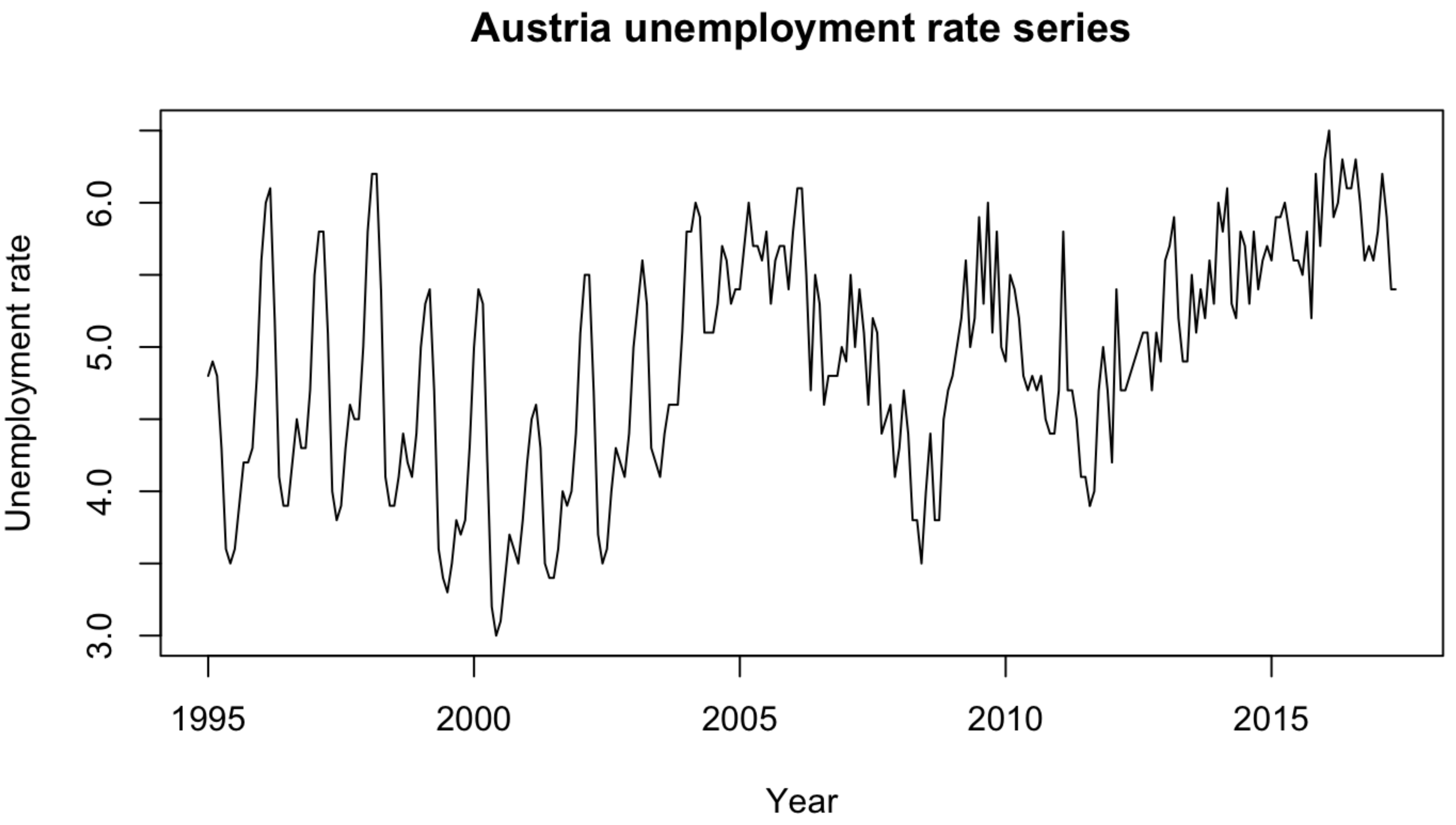

Original number of births series, fits and forecasts
Number of births
0
20000
40000
60000
80000
1850
1900
1950
2000
Year
data
Linear
Exp
Damped
SES

Unemployment rate series of Austria

In this application, we will revisit the monthly unemployment rate data of Austria between January 1995 and May 2017.

Recall that there was a changing pattern in seasonality in this series.

We will analyse this series from scratch starting from the time series plot.

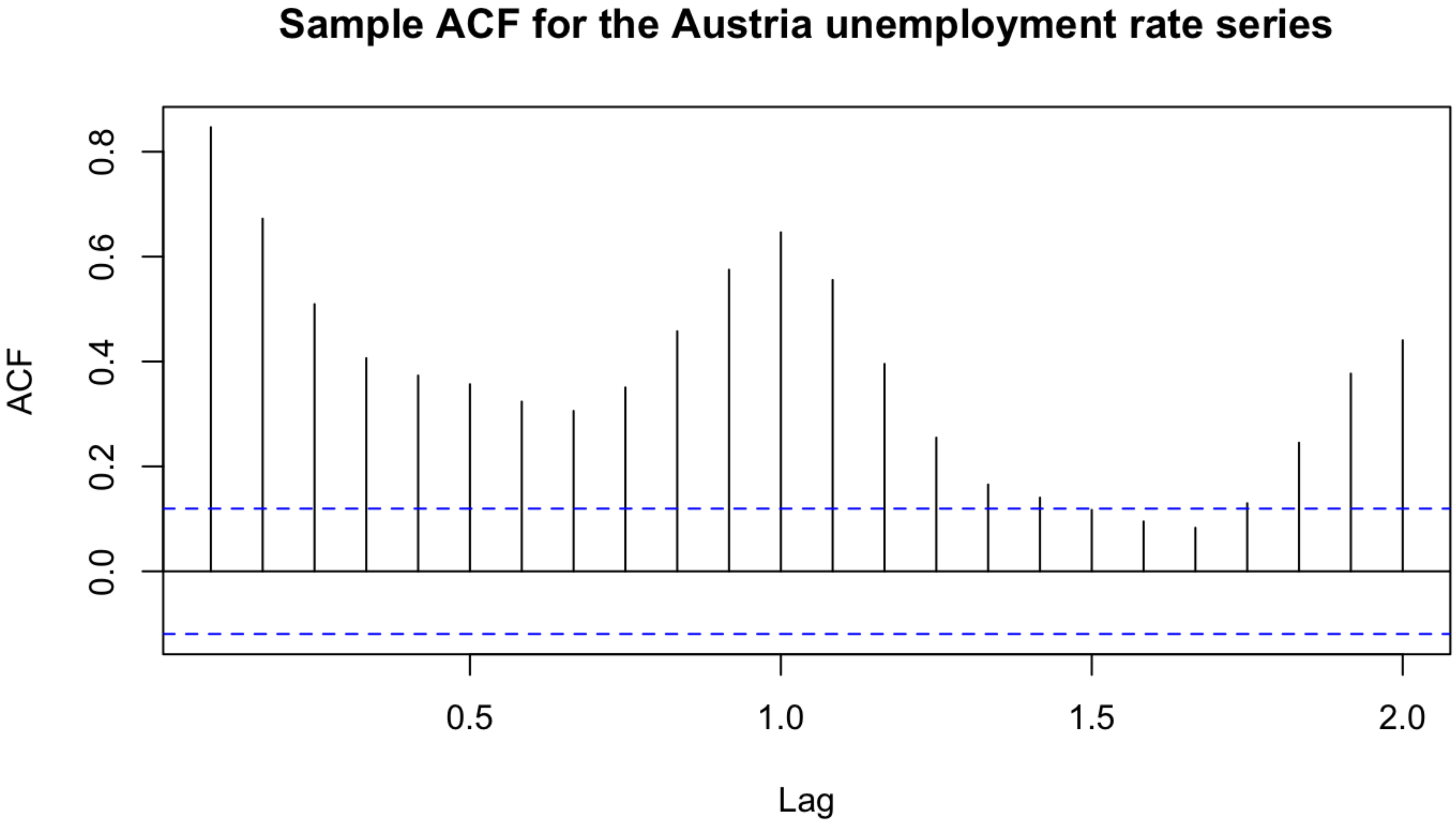

Austria unemployment rate series
Unemployment rate
3.0
4.0
5.0
6.0
1995
2000
2005
2010
2015
Year

We can infer the following from the time series plot:

- There are obvious trends in the series.
- Especially int he first part of the series there are repeating patterns due to the seasonality.
- There is an apparent changing variance.
- Because of bouncing observations around changing mean levels, we can consider a moving average behaviour for this series.
- The change in the seasonality can be perceived as an intervention point.

Next, let’s display the sample ACF function.

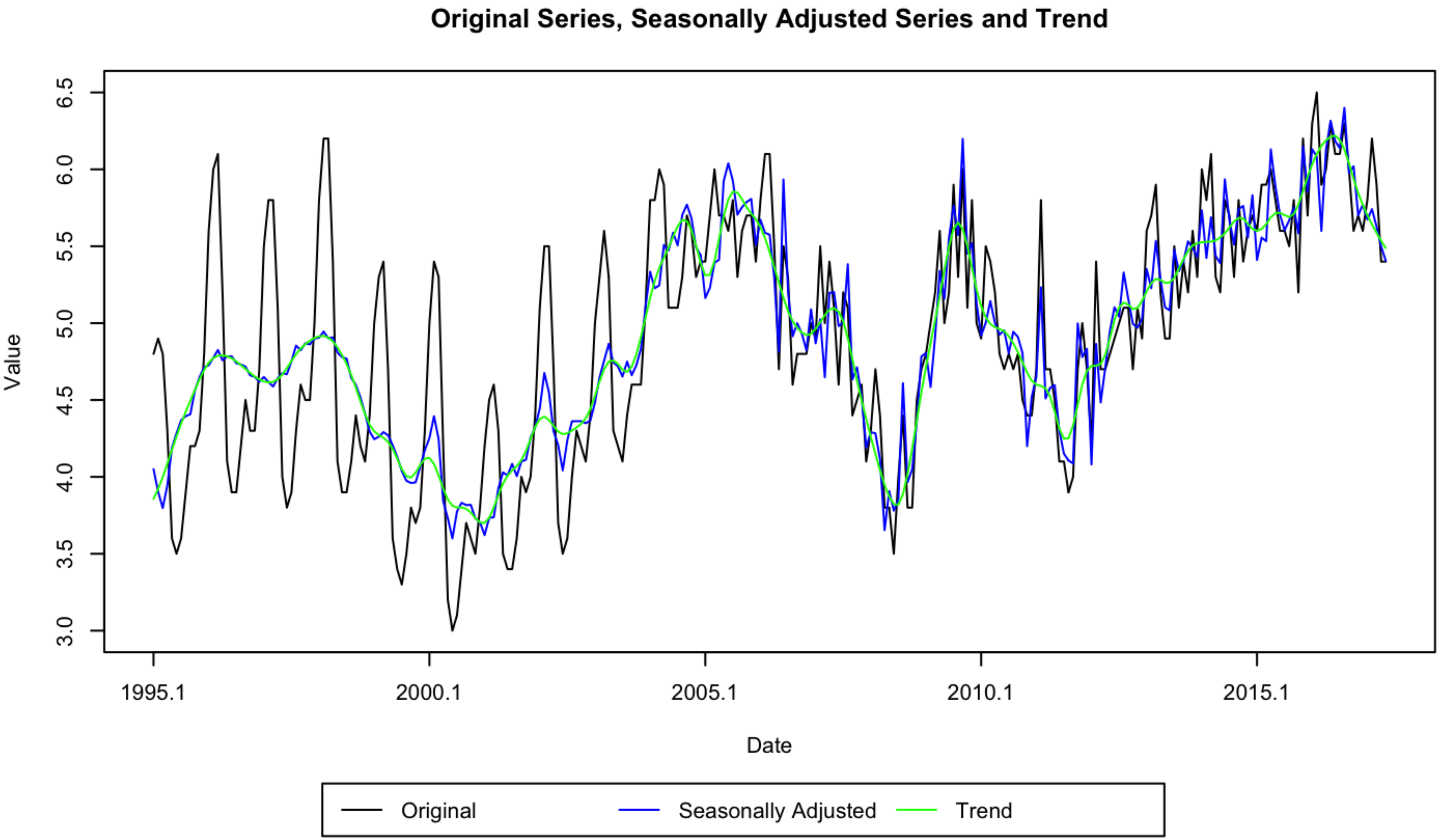
```r
acf(unemp.ts, main = "Sample ACF for the Austria unemployment rate series")
```



From the ACF plot, we can observe the slowly decaying pattern with the wave pattern on top of it implying the existence of a trend and seasonality in the series.

Let’s recall the x12 decomposition of this series.

```
fit.unemp.x12 = x12(unemp.ts)
plot(fit.unemp.x12 , sa=TRUE , trend=TRUE)
```


Original Series, Seasonally Adjusted Series and Trend
Value
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
1995.1
2000.1
2005.1
2010.1
2015.1
Date
Original
Seasonally Adjusted
Trend

In the period before 2005, seasonally adjusted series is in a good agreement with the trend component which implies that there is no other effect than the seasonality affecting the series.

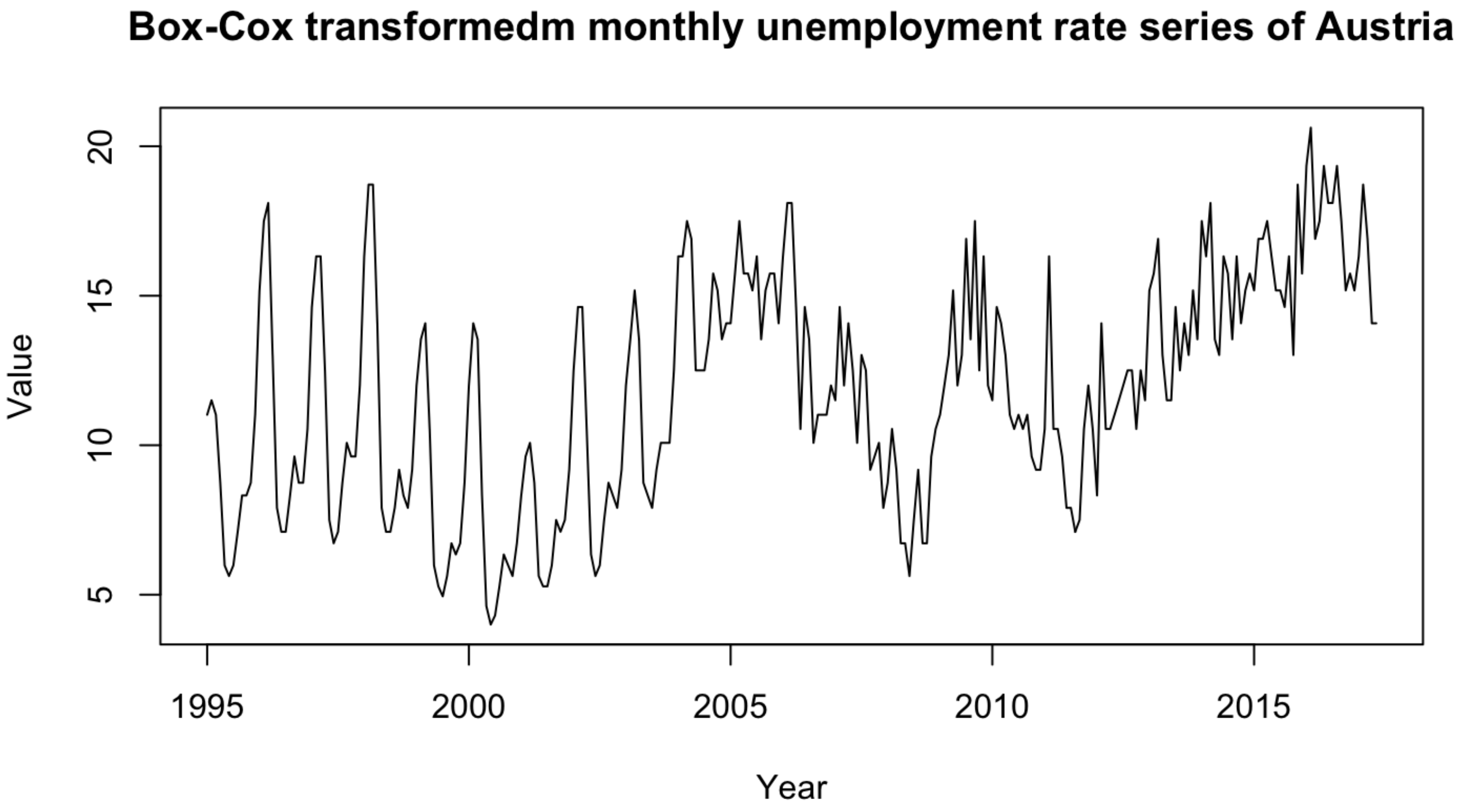
After 2005, the difference between seasonally adjusted series and the trend is due to some other effect.

From these inferences, the expectation is that the exponential smoothing models may not capture the seasonal effect in the whole series successfully.

We will apply, Holt-Winters’ model with additive and multiplicative seasonal component and damped and exponential trend components where appropriate.

We will start with the Box-Cox transformation to see if it helps the stabilise the variation in the series.

```
lambda = BoxCox.lambda(unemp.ts)
BC.unemp.ts = BoxCox(unemp.ts, lambda = lambda)
plot(BC.unemp.ts, ylab="Value",xlab="Year", main = "Box-Cox transformedm monthly unemployment rate series of Aus
```



From the time series plot of the transformed series, we cannot observe any significant improvement in the changing variance in the series.

Therefore, we will go on with the original series.

First, we will fit an additive seasonal model with a linear trend.

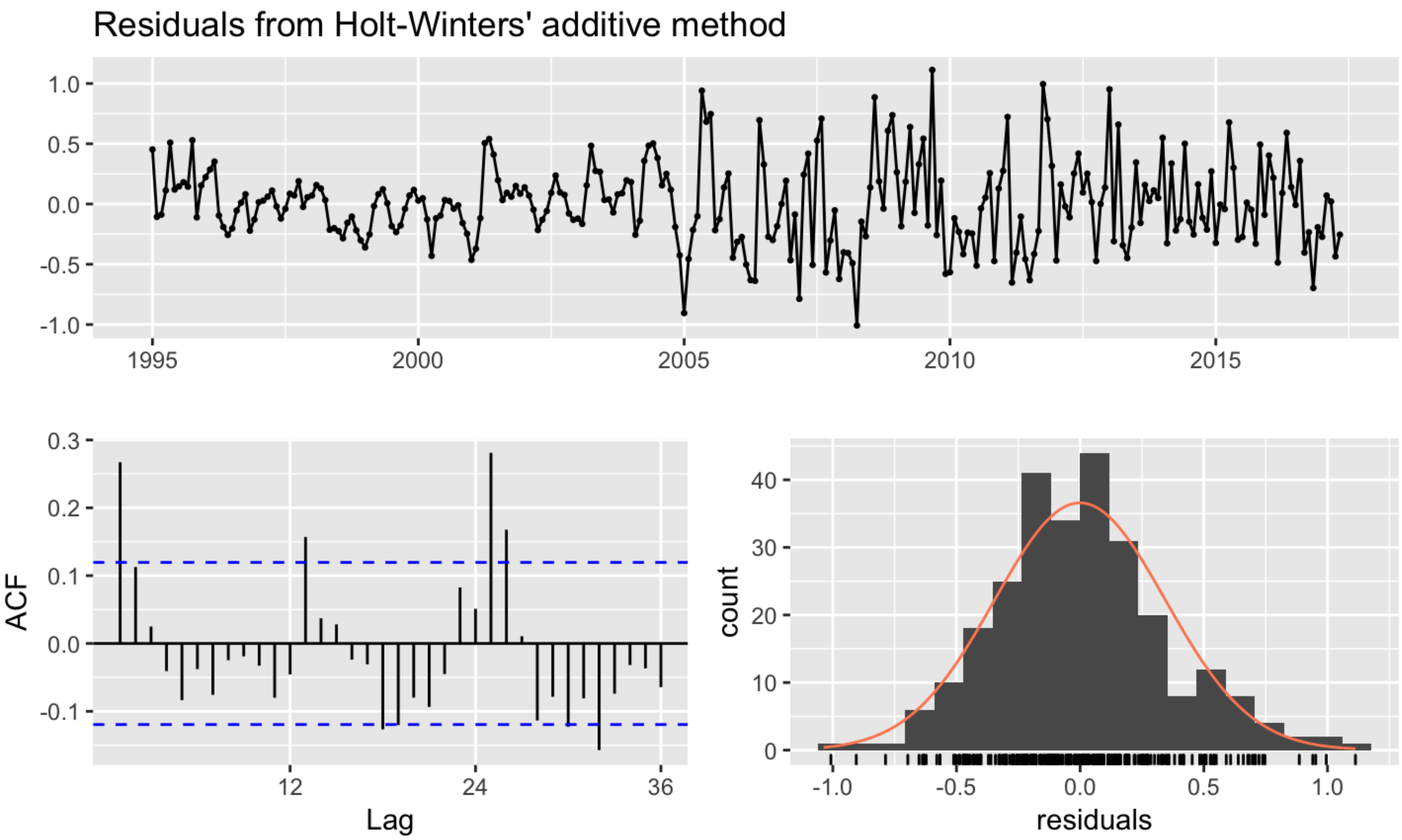
```r
fit1.hw <- hw(unemp.ts,seasonal="additive")
summary(fit1.hw)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = unemp.ts, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.3685
##     beta  = 1e-04
##     gamma = 0.4257
##
##   Initial states:
##     l = 3.684
##     b = 0.0069
##     s = 0.1821 -0.0873 -0.625 -0.1796 -0.443 -0.6476
##            -0.67 -0.7646 0.3805 1.056 1.1422 0.6562
##
##   sigma:  0.3556
##
##       AIC      AICc       BIC
##  966.2025  968.6408 1027.3126
##
## Error measures:
##                         ME      RMSE       MAE        MPE     MAPE
## Training set -0.0007182197 0.3448529 0.2658996 -0.1953569 5.436522
##                   MASE      ACF1
## Training set 0.5528819 0.2672687
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.400899 4.945192 5.856606 4.703955 6.097843
## Jul 2017       5.433947 4.948275 5.919619 4.691176 6.176718
## Aug 2017       5.426464 4.912556 5.940371 4.640510 6.212417
## Sep 2017       5.431546 4.890861 5.972230 4.604640 6.258451
## Oct 2017       5.082867 4.516657 5.649077 4.216923 5.948810
## Nov 2017       5.534646 4.944000 6.125293 4.631331 6.437962
## Dec 2017       5.401766 4.787642 6.015890 4.462545 6.340988
## Jan 2018       5.718365 5.081617 6.355114 4.744543 6.692188
## Feb 2018       6.021774 5.363166 6.680382 5.014520 7.029028
## Mar 2018       5.724340 5.044564 6.404116 4.684713 6.763968
## Apr 2018       5.478141 4.777826 6.178457 4.407101 6.549182
## May 2018       5.534426 4.814146 6.254706 4.432852 6.636000
## Jun 2018       5.483067 4.676738 6.289396 4.249894 6.716240
## Jul 2018       5.516115 4.692368 6.339862 4.256303 6.775927
## Aug 2018       5.508632 4.667819 6.349445 4.222719 6.794544
## Sep 2018       5.513714 4.656165 6.371263 4.202206 6.825222
## Oct 2018       5.165035 4.291062 6.039008 3.828408 6.501662
## Nov 2018       5.616815 4.726711 6.506918 4.255519 6.978110
```

```
forecast(fit1.hw)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.400899 4.945192 5.856606 4.703955 6.097843
## Jul 2017       5.433947 4.948275 5.919619 4.691176 6.176718
## Aug 2017       5.426464 4.912556 5.940371 4.640510 6.212417
## Sep 2017       5.431546 4.890861 5.972230 4.604640 6.258451
## Oct 2017       5.082867 4.516657 5.649077 4.216923 5.948810
## Nov 2017       5.534646 4.944000 6.125293 4.631331 6.437962
## Dec 2017       5.401766 4.787642 6.015890 4.462545 6.340988
## Jan 2018       5.718365 5.081617 6.355114 4.744543 6.692188
## Feb 2018       6.021774 5.363166 6.680382 5.014520 7.029028
## Mar 2018       5.724340 5.044564 6.404116 4.684713 6.763968
## Apr 2018       5.478141 4.777826 6.178457 4.407101 6.549182
## May 2018       5.534426 4.814146 6.254706 4.432852 6.636000
## Jun 2018       5.483067 4.676738 6.289396 4.249894 6.716240
## Jul 2018       5.516115 4.692368 6.339862 4.256303 6.775927
## Aug 2018       5.508632 4.667819 6.349445 4.222719 6.794544
## Sep 2018       5.513714 4.656165 6.371263 4.202206 6.825222
## Oct 2018       5.165035 4.291062 6.039008 3.828408 6.501662
## Nov 2018       5.616815 4.726711 6.506918 4.255519 6.978110
## Dec 2018       5.483934 4.577980 6.389889 4.098397 6.869472
## Jan 2019       5.800533 4.878992 6.722075 4.391157 7.209910
## Feb 2019       6.103942 5.167065 7.040820 4.671112 7.536773
## Mar 2019       5.806508 4.854534 6.758483 4.350589 7.262428
## Apr 2019       5.560310 4.593466 6.527154 4.081650 7.038970
## May 2019       5.616594 4.635098 6.598090 4.115526 7.117663
```

```
checkresiduals(fit1.hw)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' additive method
## Q* = 55.186, df = 8, p-value = 4.063e-09
##
## Model df: 16.   Total lags used: 24
```

MASE of this model is 0.56.

Changing variance is seen in the residuals.

We have a symmetric histogram for residuals.

The effect of seasonality is still seen in the ACF plot.

So, the Holt-Winters’ with additive seasonality and linear trend model is not capable of capturing the serial correlation structure in the series.

We will change the trend structure to the damped trend.

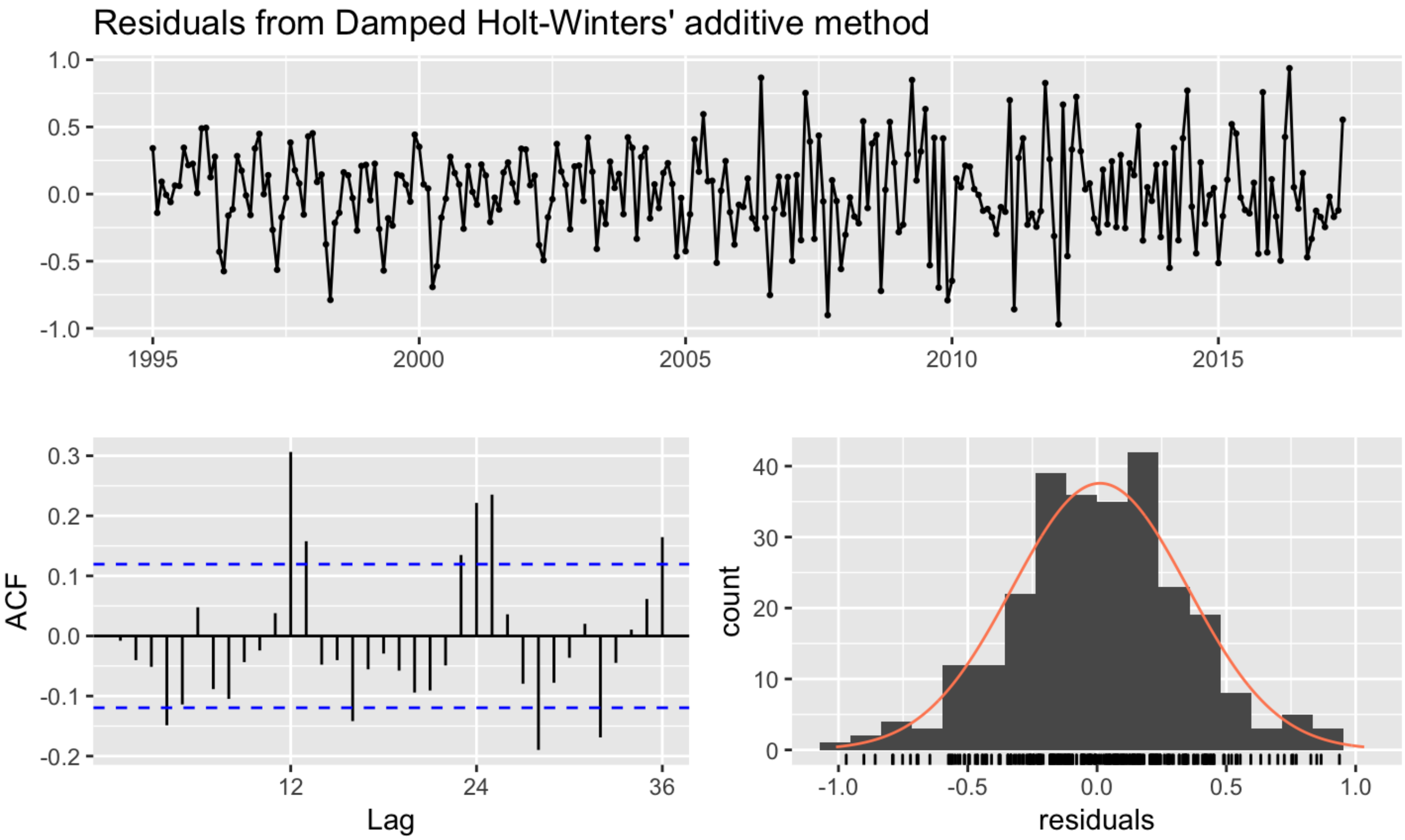
```r
fit2.hw <- hw(unemp.ts,seasonal="additive", damped = TRUE)
summary(fit2.hw)
```

```
##
## Forecast method: Damped Holt-Winters' additive method
##
## Model Information:
## Damped Holt-Winters' additive method
##
## Call:
##  hw(y = unemp.ts, seasonal = "additive", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.8403
##     beta  = 0.0054
##     gamma = 1e-04
##     phi   = 0.9308
##
##   Initial states:
##     l = 4.1813
##     b = -0.1045
##     s = -0.045 -0.097 -0.2684 -0.1247 -0.3162 -0.3386
##            -0.4503 -0.3446 0.225 0.6272 0.758 0.3746
##
##   sigma:  0.3512
##
##       AIC      AICc       BIC
##  960.5195  963.2555 1025.2243
##
## Error measures:
##                      ME      RMSE     MAE         MPE     MAPE      MASE
## Training set 0.01170264 0.3399632 0.26849 -0.02301827 5.547713 0.5582681
##                      ACF1
## Training set -0.007897814
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.205367 4.755230 5.655503 4.516943 5.893790
## Jul 2017       5.316518 4.727109 5.905928 4.415094 6.217943
## Aug 2017       5.338393 4.635692 6.041094 4.263704 6.413082
## Sep 2017       5.529445 4.728398 6.330492 4.304350 6.754541
## Oct 2017       5.385346 4.495982 6.274709 4.025182 6.745509
## Nov 2017       5.556353 4.586007 6.526699 4.072336 7.040369
## Dec 2017       5.608036 4.562377 6.653696 4.008838 7.207234
## Jan 2018       6.027274 4.910859 7.143689 4.319865 7.734683
## Feb 2018       6.410379 5.226981 7.593778 4.600527 8.220232
## Mar 2018       6.279417 5.032226 7.526607 4.372003 8.186830
## Apr 2018       5.876912 4.568680 7.185144 3.876145 7.877680
## May 2018       5.307134 3.940266 6.674002 3.216690 7.397578
## Jun 2018       5.201192 3.777803 6.624581 3.024306 7.378078
## Jul 2018       5.312633 3.834638 6.790628 3.052235 7.573031
## Aug 2018       5.334776 3.803891 6.865662 2.993489 7.676063
## Sep 2018       5.526078 3.943861 7.108296 3.106285 7.945871
## Oct 2018       5.382212 3.750087 7.014337 2.886092 7.878331
```

```
forecast(fit2.hw)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.205367 4.755230 5.655503 4.516943 5.893790
## Jul 2017       5.316518 4.727109 5.905928 4.415094 6.217943
## Aug 2017       5.338393 4.635692 6.041094 4.263704 6.413082
## Sep 2017       5.529445 4.728398 6.330492 4.304350 6.754541
## Oct 2017       5.385346 4.495982 6.274709 4.025182 6.745509
## Nov 2017       5.556353 4.586007 6.526699 4.072336 7.040369
## Dec 2017       5.608036 4.562377 6.653696 4.008838 7.207234
## Jan 2018       6.027274 4.910859 7.143689 4.319865 7.734683
## Feb 2018       6.410379 5.226981 7.593778 4.600527 8.220232
## Mar 2018       6.279417 5.032226 7.526607 4.372003 8.186830
## Apr 2018       5.876912 4.568680 7.185144 3.876145 7.877680
## May 2018       5.307134 3.940266 6.674002 3.216690 7.397578
## Jun 2018       5.201192 3.777803 6.624581 3.024306 7.378078
## Jul 2018       5.312633 3.834638 6.790628 3.052235 7.573031
## Aug 2018       5.334776 3.803891 6.865662 2.993489 7.676063
## Sep 2018       5.526078 3.943861 7.108296 3.106285 7.945871
## Oct 2018       5.382212 3.750087 7.014337 2.886092 7.878331
## Nov 2018       5.553436 3.872714 7.234158 2.982993 8.123878
## Dec 2018       5.605321 3.877213 7.333429 2.962408 8.248234
## Jan 2019       6.024747 4.250377 7.799116 3.311083 8.738411
## Feb 2019       6.408027 4.588443 8.227610 3.625214 9.190840
## Mar 2019       6.277227 4.413409 8.141044 3.426764 9.127690
## Apr 2019       5.874874 3.967742 7.782006 2.958167 8.791580
## May 2019       5.305237 3.355656 7.254817 2.323610 8.286863
```

```
checkresiduals(fit2.hw)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' additive method
## Q* = 86.265, df = 7, p-value = 7.772e-16
##
## Model df: 17.   Total lags used: 24
```

MASE of this first model is 0.55.

We have very similar residual diagnostics with the previous model.

The effect of seasonality is still seen in the ACF plot.

So, the Holt-Winters’ with additive seasonality and damped trend model is not capable of capturing the serial correlation structure in the series.

We will change the seasonality to multiplicative and fit the model with linear trend.

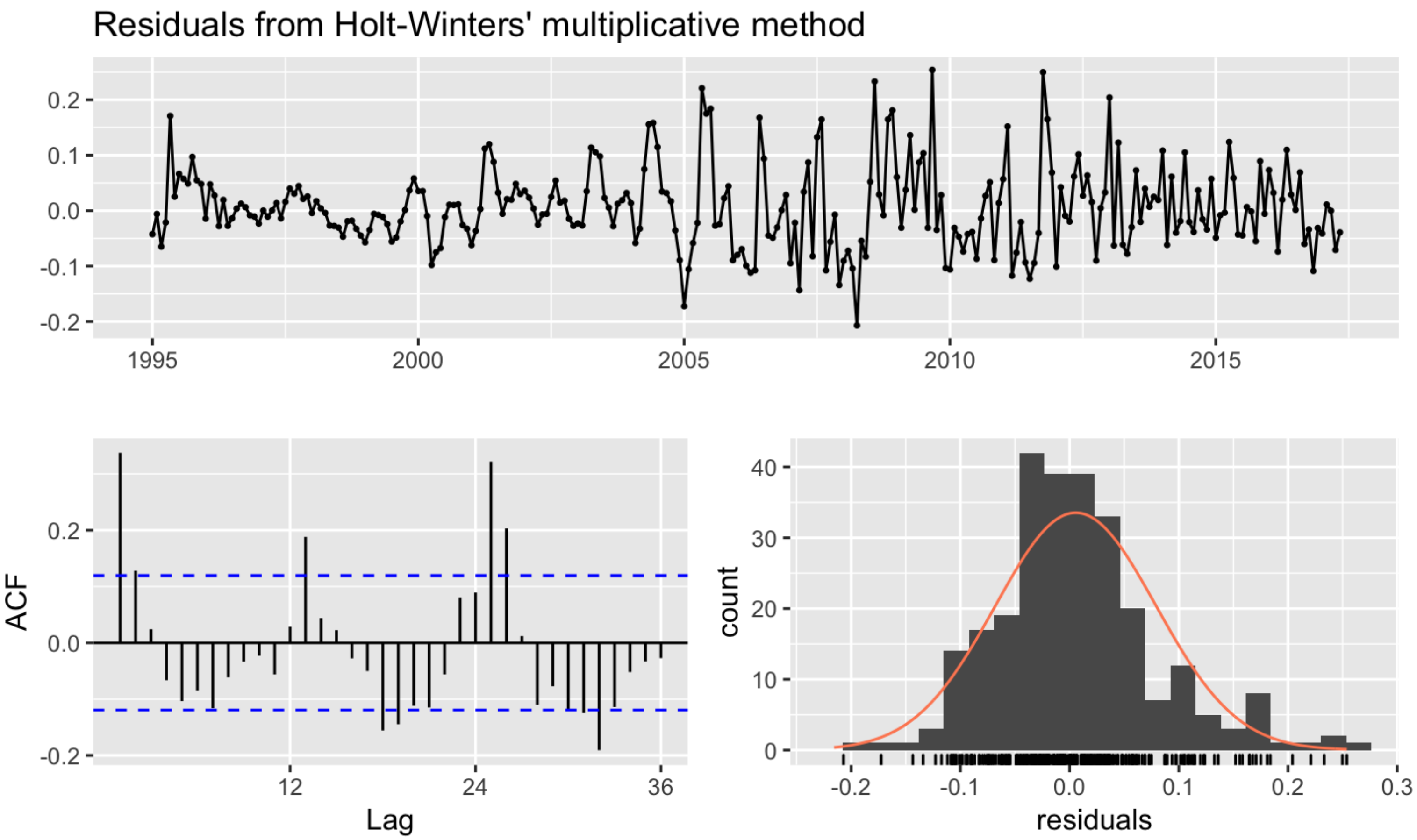
```r
fit3.hw <- hw(unemp.ts, seasonal = "multiplicative")
summary(fit3.hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = unemp.ts, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.3579
##     beta  = 1e-04
##     gamma = 0.3967
##
##   Initial states:
##     l = 4.0362
##     b = -0.0033
##     s = 1.0014 0.9083 0.8821 0.938 0.881 0.8248
##            0.841 0.8031 1.1378 1.2973 1.2422 1.243
##
##   sigma:  0.0761
##
##       AIC      AICc       BIC
##  982.9136  985.3519 1044.0237
##
## Error measures:
##                       ME      RMSE       MAE        MPE     MAPE       MASE
## Training set 0.009318358 0.3560515 0.2664244 0.03855288 5.391298 0.5539731
##                   ACF1
## Training set 0.3149543
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.396962 4.870519 5.923405 4.591836 6.202087
## Jul 2017       5.428501 4.865859 5.991144 4.568014 6.288988
## Aug 2017       5.389407 4.799690 5.979124 4.487513 6.291301
## Sep 2017       5.412722 4.790693 6.034751 4.461410 6.364034
## Oct 2017       5.075499 4.465529 5.685470 4.142630 6.008369
## Nov 2017       5.497619 4.809151 6.186088 4.444698 6.550541
## Dec 2017       5.336042 4.641841 6.030243 4.274354 6.397731
## Jan 2018       5.624334 4.866182 6.382487 4.464841 6.783828
## Feb 2018       5.908096 5.084787 6.731406 4.648953 7.167239
## Mar 2018       5.637762 4.827203 6.448322 4.398118 6.877406
## Apr 2018       5.396769 4.597658 6.195880 4.174634 6.618904
## May 2018       5.423342 4.597586 6.249098 4.160457 6.686227
## Jun 2018       5.370034 4.459947 6.280122 3.978176 6.761893
## Jul 2018       5.401401 4.466026 6.336777 3.970868 6.831934
## Aug 2018       5.362487 4.414390 6.310584 3.912498 6.812476
## Sep 2018       5.385670 4.414278 6.357063 3.900053 6.871287
## Oct 2018       5.050119 4.121554 5.978684 3.630001 6.470237
## Nov 2018       5.470112 4.445470 6.494755 3.903057 7.037167
```

```
forecast(fit3.hw)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.396962 4.870519 5.923405 4.591836 6.202087
## Jul 2017       5.428501 4.865859 5.991144 4.568014 6.288988
## Aug 2017       5.389407 4.799690 5.979124 4.487513 6.291301
## Sep 2017       5.412722 4.790693 6.034751 4.461410 6.364034
## Oct 2017       5.075499 4.465529 5.685470 4.142630 6.008369
## Nov 2017       5.497619 4.809151 6.186088 4.444698 6.550541
## Dec 2017       5.336042 4.641841 6.030243 4.274354 6.397731
## Jan 2018       5.624334 4.866182 6.382487 4.464841 6.783828
## Feb 2018       5.908096 5.084787 6.731406 4.648953 7.167239
## Mar 2018       5.637762 4.827203 6.448322 4.398118 6.877406
## Apr 2018       5.396769 4.597658 6.195880 4.174634 6.618904
## May 2018       5.423342 4.597586 6.249098 4.160457 6.686227
## Jun 2018       5.370034 4.459947 6.280122 3.978176 6.761893
## Jul 2018       5.401401 4.466026 6.336777 3.970868 6.831934
## Aug 2018       5.362487 4.414390 6.310584 3.912498 6.812476
## Sep 2018       5.385670 4.414278 6.357063 3.900053 6.871287
## Oct 2018       5.050119 4.121554 5.978684 3.630001 6.470237
## Nov 2018       5.470112 4.445470 6.494755 3.903057 7.037167
## Dec 2018       5.309329 4.296792 6.321865 3.760788 6.857869
## Jan 2019       5.596161 4.510226 6.682097 3.935366 7.256956
## Feb 2019       5.878485 4.718409 7.038560 4.104303 7.652667
## Mar 2019       5.609490 4.484289 6.734691 3.888643 7.330337
## Apr 2019       5.369690 4.275394 6.463985 3.696110 7.043270
## May 2019       5.396114 4.279380 6.512848 3.688217 7.104011
```

```
checkresiduals(fit3.hw)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 85.774, df = 8, p-value = 3.331e-15
##
## Model df: 16.   Total lags used: 24
```

MASE of this model increased to 0.57.

We have very similar residual diagnostics with the previous model.

The effect of seasonality is still seen in the ACF plot.

So, the Holt-Winters’ with multiplicative seasonality and linear trend model is not capable of capturing the serial correlation structure in the series.

We will change the trend component to exponential and fit the model.

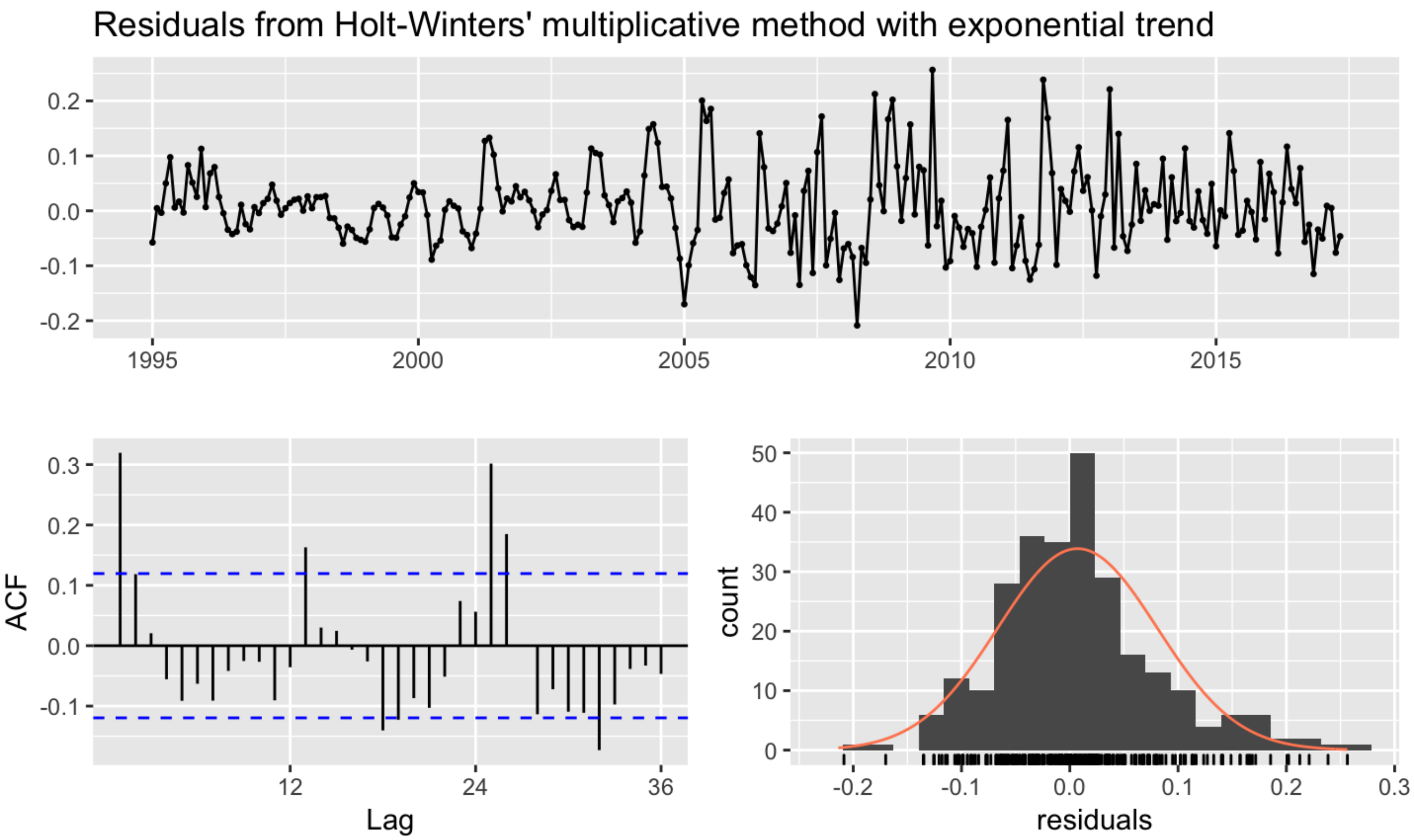
```r
fit4.hw <- hw(unemp.ts, seasonal = "multiplicative", exponential = TRUE)
summary(fit4.hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method with exponential trend
##
## Model Information:
## Holt-Winters' multiplicative method with exponential trend
##
## Call:
##  hw(y = unemp.ts, seasonal = "multiplicative", exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.353
##     beta  = 1e-04
##     gamma = 0.4806
##
##   Initial states:
##     l = 4.0562
##     b = 0.9986
##     s = 0.9846 0.9644 0.9341 0.932 0.9379 0.8526
##            0.8384 0.8165 1.036 1.2153 1.2307 1.2575
##
##   sigma:  0.0762
##
##       AIC      AICc       BIC
##  982.3427  984.7810 1043.4528
##
## Error measures:
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.01885068 0.3575806 0.2667201 0.1917535 5.376745 0.5545879
##                   ACF1
## Training set 0.2899992
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.368009 4.834529 5.880719 4.546691 6.154095
## Jul 2017       5.362422 4.815016 5.922546 4.546314 6.221078
## Aug 2017       5.342721 4.769037 5.935510 4.500977 6.262244
## Sep 2017       5.305099 4.707212 5.912338 4.395591 6.256037
## Oct 2017       4.965791 4.372807 5.595143 4.128926 5.918952
## Nov 2017       5.371220 4.710398 6.062912 4.365892 6.437751
## Dec 2017       5.247263 4.587958 5.944582 4.288639 6.347015
## Jan 2018       5.550475 4.822411 6.317428 4.477167 6.711412
## Feb 2018       5.851159 5.058298 6.683873 4.689915 7.163026
## Mar 2018       5.561548 4.800337 6.382940 4.403718 6.852474
## Apr 2018       5.312728 4.558694 6.120653 4.187473 6.596943
## May 2018       5.367345 4.590952 6.193973 4.223888 6.701613
## Jun 2018       5.290762 4.412125 6.236061 3.984217 6.824506
## Jul 2018       5.285255 4.390534 6.253687 3.963835 6.830222
## Aug 2018       5.265838 4.375428 6.273240 3.976363 6.832826
## Sep 2018       5.228757 4.348194 6.189963 3.971133 6.803567
## Oct 2018       4.894332 4.042950 5.845480 3.640541 6.423067
## Nov 2018       5.293927 4.341380 6.322050 3.916563 6.883603
```

```
forecast(fit4.hw)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2017       5.368009 4.834529 5.880719 4.546691 6.154095
## Jul 2017       5.362422 4.815016 5.922546 4.546314 6.221078
## Aug 2017       5.342721 4.769037 5.935510 4.500977 6.262244
## Sep 2017       5.305099 4.707212 5.912338 4.395591 6.256037
## Oct 2017       4.965791 4.372807 5.595143 4.128926 5.918952
## Nov 2017       5.371220 4.710398 6.062912 4.365892 6.437751
## Dec 2017       5.247263 4.587958 5.944582 4.288639 6.347015
## Jan 2018       5.550475 4.822411 6.317428 4.477167 6.711412
## Feb 2018       5.851159 5.058298 6.683873 4.689915 7.163026
## Mar 2018       5.561548 4.800337 6.382940 4.403718 6.852474
## Apr 2018       5.312728 4.558694 6.120653 4.187473 6.596943
## May 2018       5.367345 4.590952 6.193973 4.223888 6.701613
## Jun 2018       5.290762 4.412125 6.236061 3.984217 6.824506
## Jul 2018       5.285255 4.390534 6.253687 3.963835 6.830222
## Aug 2018       5.265838 4.375428 6.273240 3.976363 6.832826
## Sep 2018       5.228757 4.348194 6.189963 3.971133 6.803567
## Oct 2018       4.894332 4.042950 5.845480 3.640541 6.423067
## Nov 2018       5.293927 4.341380 6.322050 3.916563 6.883603
## Dec 2018       5.171753 4.248122 6.198563 3.831273 6.935760
## Jan 2019       5.470602 4.484828 6.595710 4.019166 7.223712
## Feb 2019       5.766959 4.682990 6.943164 4.233246 7.730652
## Mar 2019       5.481516 4.437273 6.637276 3.935483 7.349168
## Apr 2019       5.236276 4.236114 6.360686 3.765521 7.068449
## May 2019       5.290107 4.248577 6.404732 3.807723 7.214688
```

```
checkresiduals(fit4.hw)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method with exponential trend
## Q* = 68.769, df = 8, p-value = 8.636e-12
##
## Model df: 16.   Total lags used: 24
```

MASE of this model is 0.57.

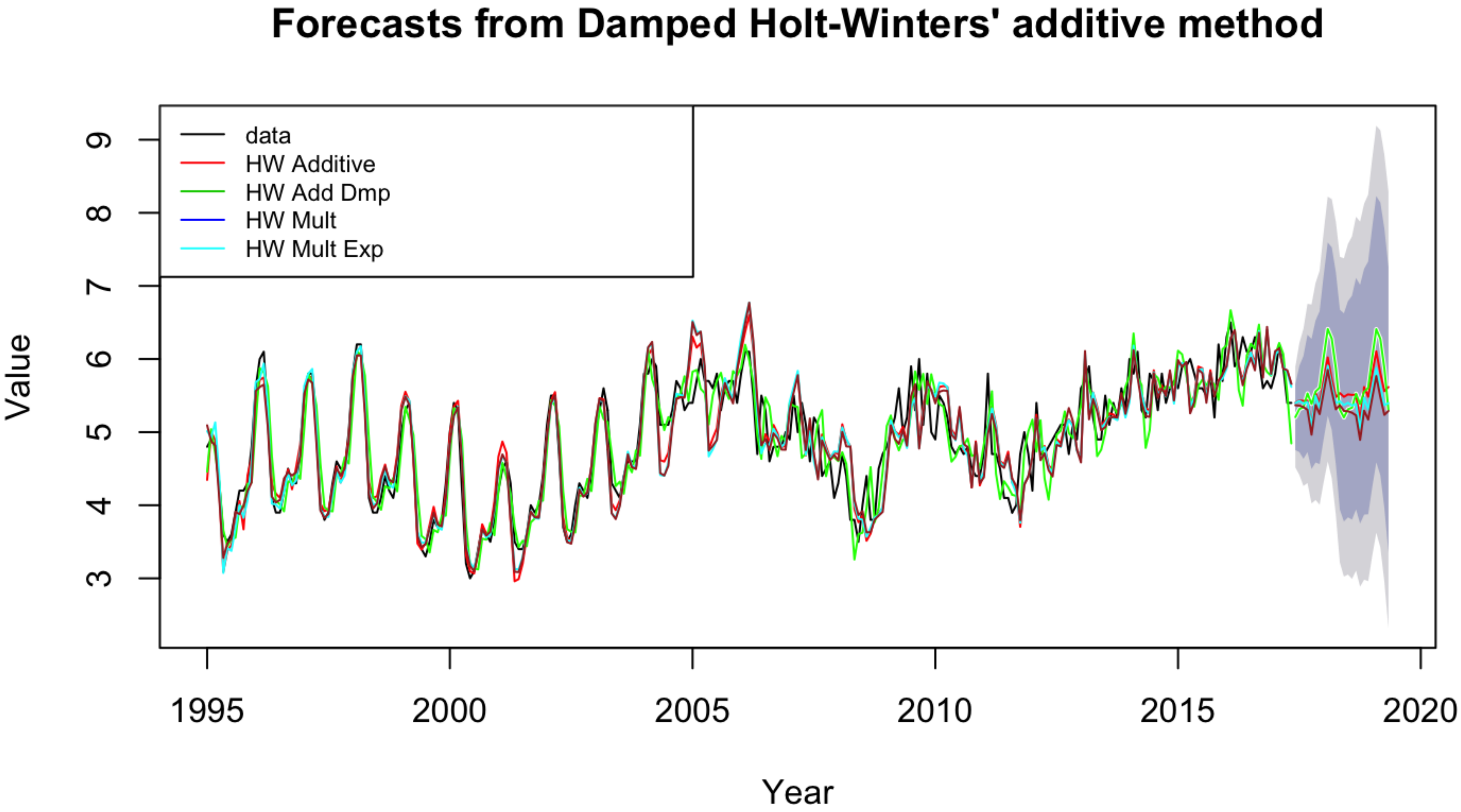
We have very similar residual diagnostics with the previous model.

The effect of seasonality is still seen in the ACF plot.

So, the Holt-Winters’ with multiplicative seasonality and exponential trend model is not capable of capturing the serial correlation structure in the series.

All the fitted models have similar goodness-of-fit performance. The following plot shows the forecasts.

```
plot(fit2.hw,ylab="Value", plot.conf=FALSE, type="l", fcol="white", xlab="Year")
lines(fitted(fit1.hw), col="red", lty=1)
lines(fitted(fit2.hw), col="green", lty=1)
lines(fitted(fit3.hw), col="cyan", lty=1)
lines(fitted(fit4.hw), col="brown", lty=1)
lines(fit1.hw$mean, type="l", col="red")
lines(fit2.hw$mean, type="l", col="green")
lines(fit3.hw$mean, type="l", col="cyan")
lines(fit4.hw$mean, type="l", col="brown")
legend("topleft",lty=1, col=1:5, text.width = 9, cex = 0.7,
       c("data","HW Additive", "HW Add Dmp", "HW Mult", "HW" ))
```


Forecasts from Damped Holt-Winters' additive method
data
HW Additive
HW Add Dmp
HW Mult
HW Mult Exp
Value
3
4
5
6
7
8
9
Year
1995
2000
2005
2010
2015
2020

There is a notable difference between the forecasts of the models with additive and multiplicative seasonal components.

This difference is due to the way the models handle seasonality.

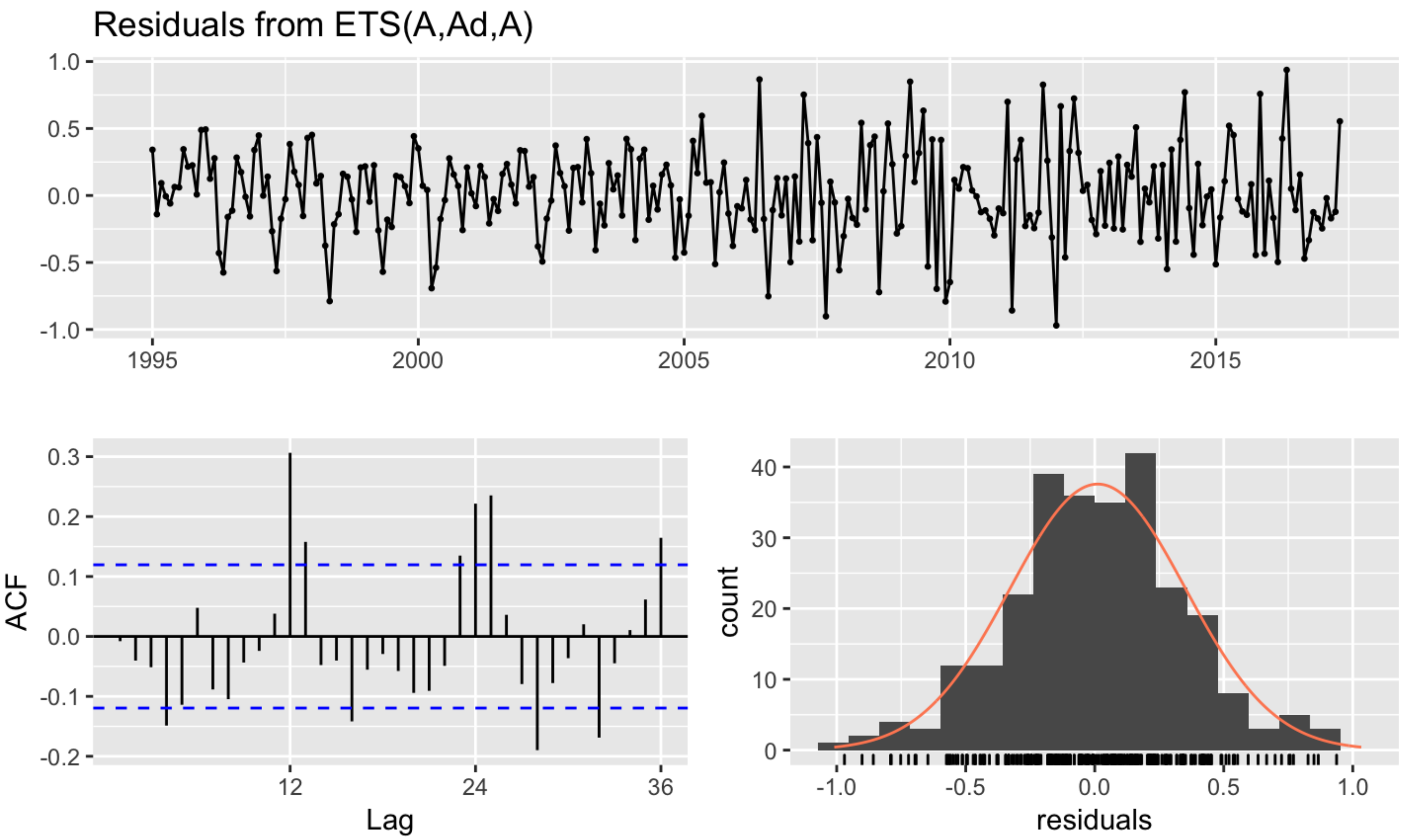
We will go on with fitting state-space counterparts of these models where appropriate.

The first state space model is with additive errors, damped additive trend and additive seasonality.

```r
fit.AAdA = ets(unemp.ts, model = "AAA", damped = TRUE)
summary(fit.AAdA)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = unemp.ts, model = "AAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.8403
##     beta  = 0.0054
##     gamma = 1e-04
##     phi   = 0.9308
##
##   Initial states:
##     l = 4.1813
##     b = -0.1045
##     s = -0.045 -0.097 -0.2684 -0.1247 -0.3162 -0.3386
##            -0.4503 -0.3446 0.225 0.6272 0.758 0.3746
##
##   sigma:  0.3512
##
##       AIC      AICc       BIC
##  960.5195  963.2555 1025.2243
##
## Training set error measures:
##                       ME      RMSE     MAE         MPE     MAPE      MASE
## Training set 0.01170264 0.3399632 0.26849 -0.02301827 5.547713 0.5582681
##                       ACF1
## Training set -0.007897814
```

```
checkresiduals(fit.AAdA)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 86.265, df = 7, p-value = 7.772e-16
##
## Model df: 17.   Total lags used: 24
```

With this ETS(A,Ad,A) model, MASE decreased to 0.56.

We have very slightly better residual diagnostics in terms of the time series plot of residuals with this model.

The effect of seasonality is still seen in the ACF plot.

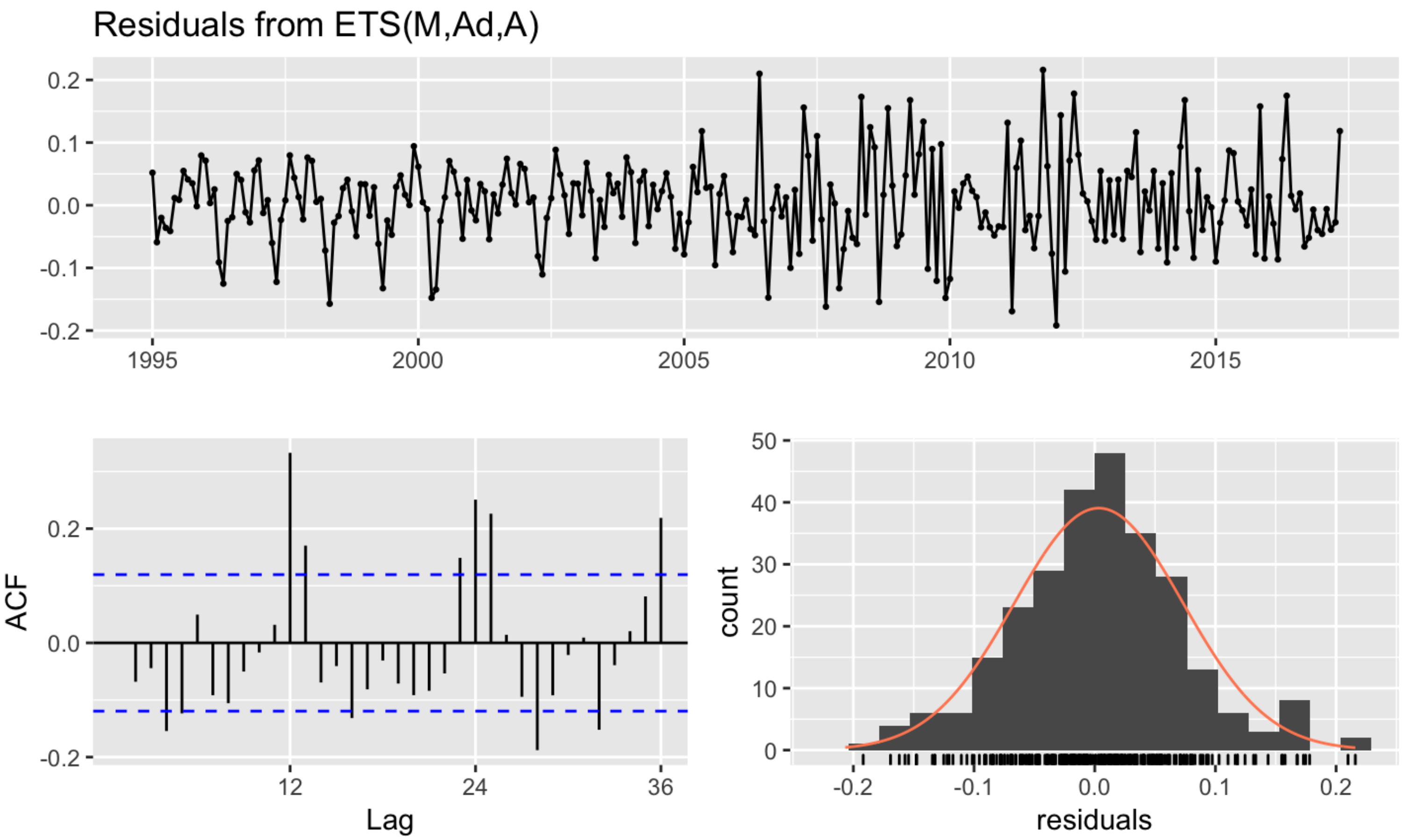
So, the ETS(A,Ad,A) model is also not capable of capturing the serial correlation structure in the series.

We will change the errors to the multiplicative.

```r
fit.MAdA = ets(unemp.ts, model = "MAA", damped = TRUE)
summary(fit.MAdA)
```

```
## ETS(M,Ad,A) 
## 
## Call:
##  ets(y = unemp.ts, model = "MAA", damped = TRUE) 
## 
##   Smoothing parameters:
##     alpha = 0.8838 
##     beta  = 1e-04 
##     gamma = 1e-04 
##     phi   = 0.9567 
## 
##   Initial states:
##     l = 4.1661 
##     b = 0.0315 
##     s = -0.0843 -0.2119 -0.3159 -0.1732 -0.3095 -0.3884
##            -0.4394 -0.2574 0.3317 0.7084 0.773 0.367
## 
##   sigma:  0.0723
## 
##       AIC      AICc       BIC 
##  957.1937  959.9297 1021.8985 
## 
## Training set error measures:
##                       ME    RMSE       MAE        MPE     MAPE      MASE
## Training set 0.003056647 0.34325 0.2634517 -0.1563431 5.403039 0.5477919
##                     ACF1
## Training set -0.04361398
```

```
checkresiduals(fit.MAdA)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,A)
## Q* = 100.56, df = 7, p-value < 2.2e-16
##
## Model df: 17.   Total lags used: 24
```

With this ETS(M,Ad,A) model, MASE decreased to 0.549.

We have very similar residual diagnostics as the ETS(A,Ad,A) model

The effect of seasonality is still seen in the ACF plot.

So, the ETS(M,Ad,A) model is also not capable of capturing the serial correlation structure in the series.

We had a slight improvement in the goodness-of-fit with the state space approach.

This is mainly because of the changing structure of seasonality in the series.

The change in the structure of seasonality can be perceived as a seasonal trend.

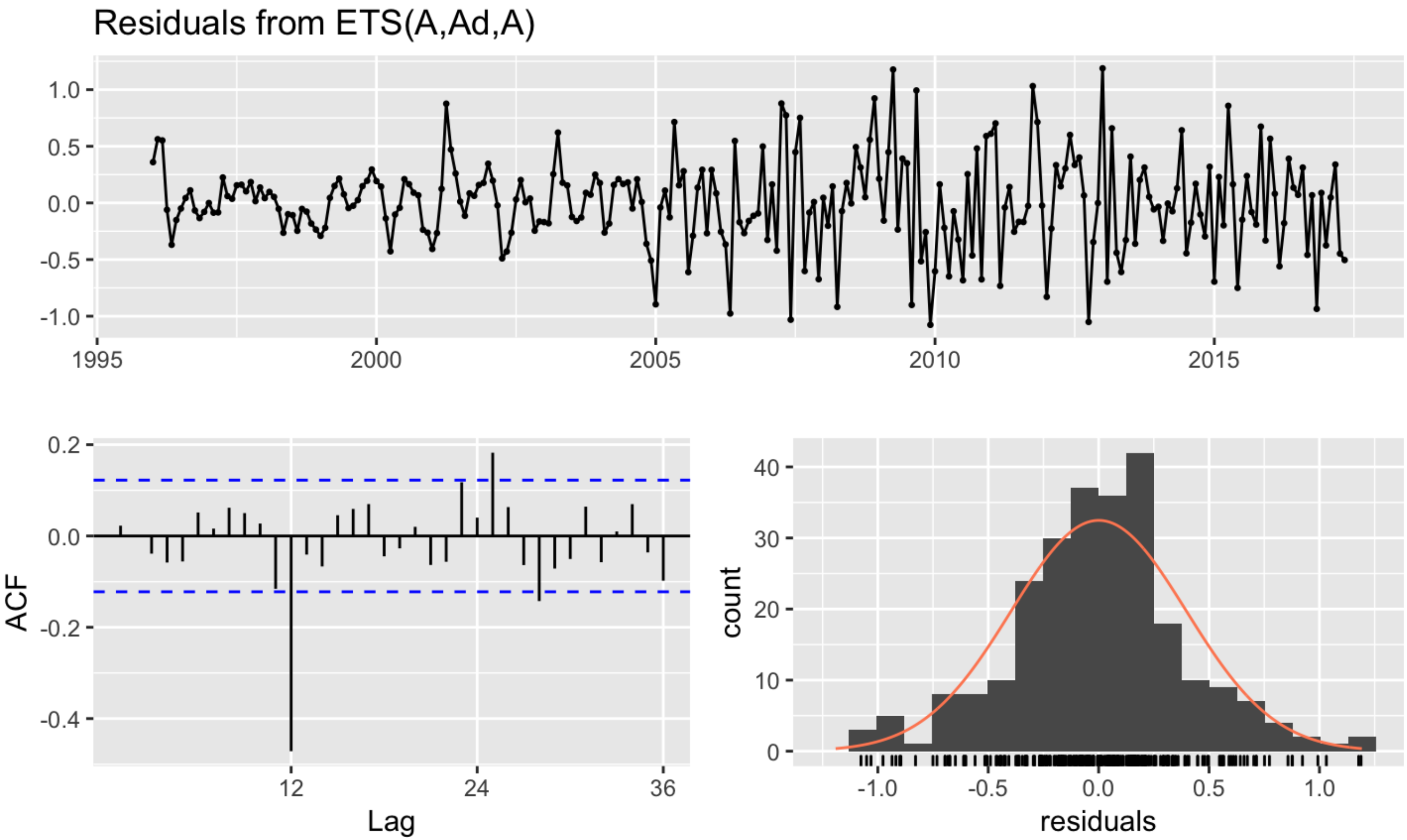
A seasonal difference might filtrate the effect of this changing seasonality.

Therefore, we will apply the first seasonal difference and fit the models to the first seasonal differenced series.

```r
diff.unemp.ts = diff(unemp.ts,lag = 12)
fit.AAdA.s = ets(diff.unemp.ts, model = "AAA", damped = TRUE)
summary(fit.AAdA.s)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = diff.unemp.ts, model = "AAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.6341
##     beta  = 0.0088
##     gamma = 1e-04
##     phi   = 0.8855
##
##   Initial states:
##     l = 0.6193
##     b = -0.1983
##     s = -0.0159 0.0086 -0.0508 -0.014 0.0172 0.0061
##            0.0303 0.0237 -0.0228 0.003 0.0182 -0.0036
##
##   sigma:  0.4102
##
##        AIC      AICc       BIC
##  986.4977  989.3716 1050.3810
##
## Training set error measures:
##                        ME      RMSE       MAE MPE MAPE      MASE
## Training set 0.0005270654 0.3964042 0.2977238 NaN  Inf 0.4116385
##                    ACF1
## Training set 0.02251933
```

```
checkresiduals(fit.AAdA.s)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 80.749, df = 7, p-value = 9.659e-15
##
## Model df: 17.   Total lags used: 24
```

The MASE of ETS(A,Ad,A) model with first seasonally differenced series decreased to 0.422, which is a significant improvement.

Although there is still some changing variance seen in the residuals, there is no repeating pattern in the time series plot.

There is no sign of seasonality in the ACF plot but a significant serial correlation at lag 12 indicating the effect of intervention effect in seasonality.

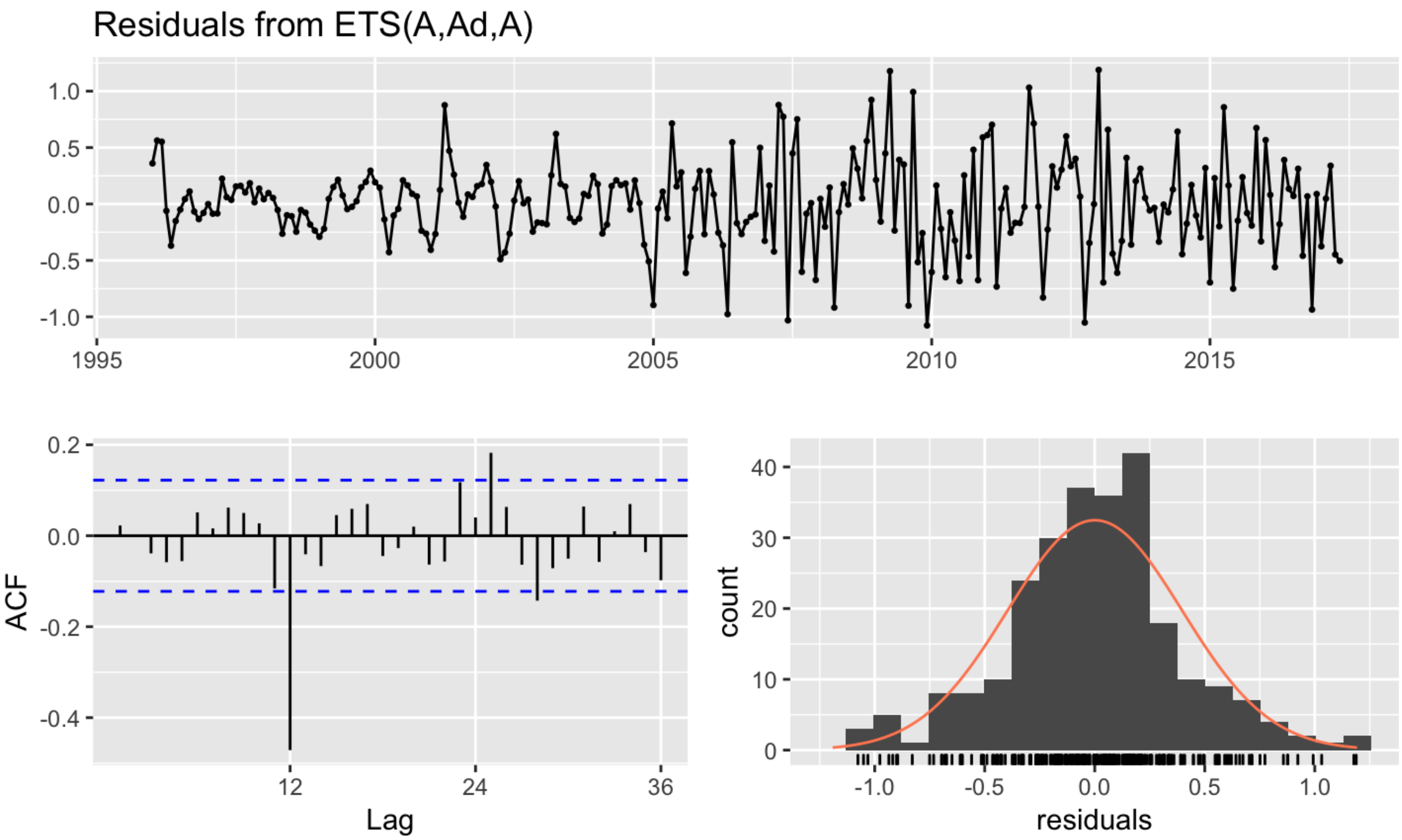
So, the ETS(A,Ad,A) model with the first seasonally differenced series performs better than the previous alternatives.

Lastly, we will fit ETS(M,Ad,A) model with first seasonally differenced series.

```
fit.AAA.s = ets(diff.unemp.ts, model = "AAA")
summary(fit.AAA.s)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = diff.unemp.ts, model = "AAA")
##
##   Smoothing parameters:
##     alpha = 0.6341
##     beta  = 0.0088
##     gamma = 1e-04
##     phi   = 0.8855
##
##   Initial states:
##     l = 0.6193
##     b = -0.1983
##     s = -0.0159 0.0086 -0.0508 -0.014 0.0172 0.0061
##            0.0303 0.0237 -0.0228 0.003 0.0182 -0.0036
##
##   sigma:  0.4102
##
##       AIC      AICc       BIC
##  986.4977  989.3716 1050.3810
##
## Training set error measures:
##                         ME      RMSE       MAE MPE MAPE      MASE
## Training set 0.0005270654 0.3964042 0.2977238 NaN  Inf 0.4116385
##                    ACF1
## Training set 0.02251933
```

```
checkresiduals(fit.AAA.s)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 80.749, df = 7, p-value = 9.659e-15
##
## Model df: 17.   Total lags used: 24
```

The MASE of ETS(A,A,A) model with first seasonally differenced series decreased to 0.416.

We have very similar residual diagnostics as the previous model.

There is no sign of seasonality in the ACF plot but a significant serial correlation at lag 12 indicating the effect of intervention effect in seasonality.

So, the ETS(A,A,A) model with the first seasonally differenced series performs better than the previous alternatives.

The following code chunk computes the forecasts for the models with original series.

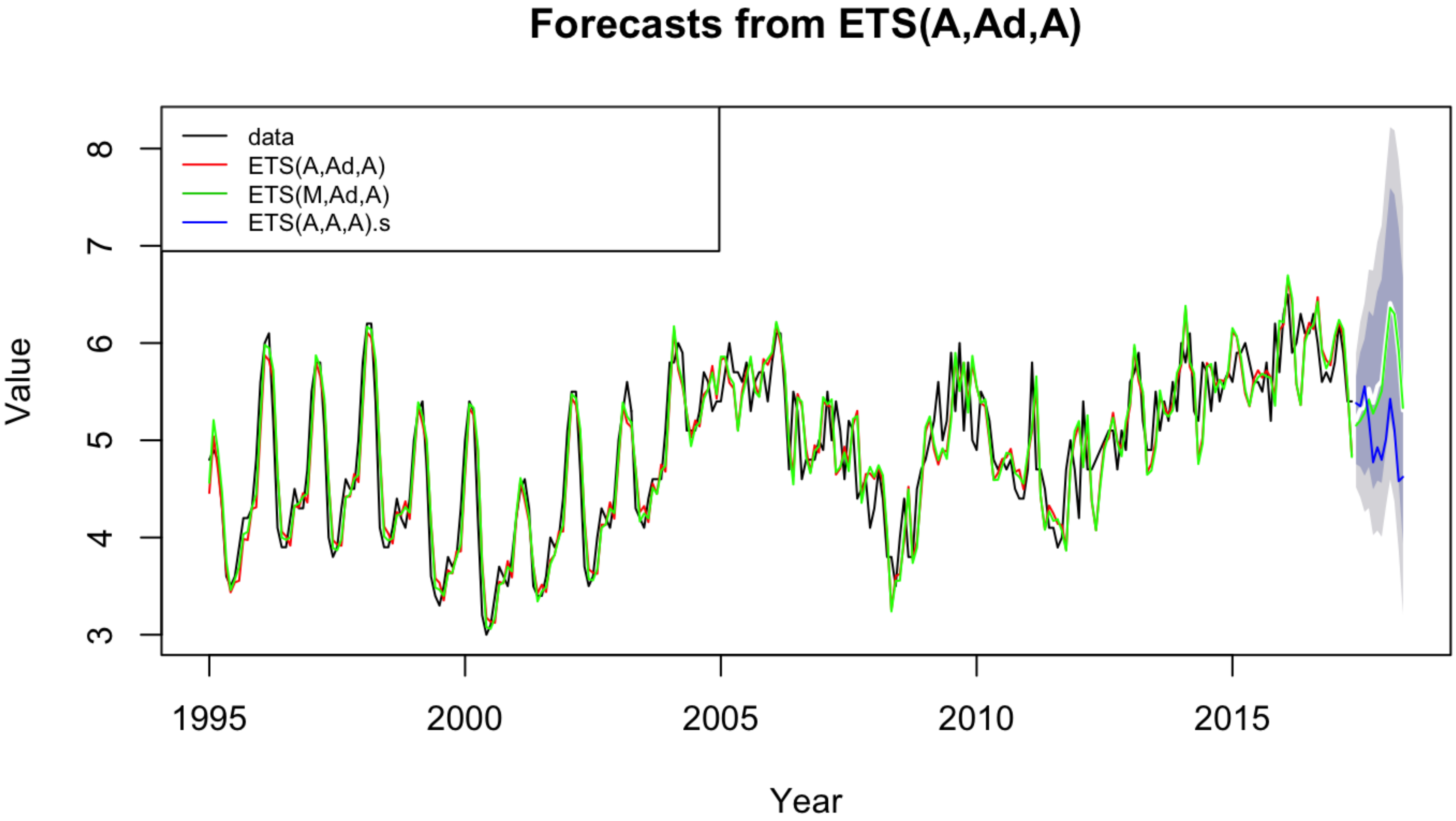
```
frc.AAdA = forecast(fit.AAdA , h = frequency(unemp.ts))
frc.MAdA = forecast(fit.MAdA , h = frequency(unemp.ts))
frc.AAdA.s = forecast(fit.AAdA.s , h = frequency(unemp.ts))
frc.AAA.s = forecast(fit.AAA.s , h = frequency(unemp.ts))
```

To find the forecasts with the difference series, we need to take the difference operation back.

Because we applied seasonal differencing, we need to give 12 observations to take the differencing back

```
back.series = diffinv(frc.AAA.s$mean,xi = unemp.ts[(21*12+6):(22*12+5)],
                      lag = 12)
# To extract forecasts get rid of the additional 12 elements
forecasts.AAA.s = back.series[(length(back.series)-11):length(back.series)]
forecasts.AAA.s = ts(forecasts.AAA.s, start = c(2017,6), frequency = 12)
```

```
plot(frc.AAdA, ylab="Value",plot.conf=FALSE, type="l", fcol="white",
     xlab="Year")
lines(fitted(fit.AAdA), col="red", lty=1)
lines(fitted(fit.MAdA), col="green", lty=1)
lines(fitted(fit.AAdA.s), col="blue", lty=1)
lines(frc.MAdA$mean,col="green", type="l")
lines(frc.AAdA.s$mean,col="red", type="l")
lines(forecasts.AAA.s,col="blue", type="l")
legend("topleft",lty=1, col=1:4, text.width = 9, cex = 0.7,
       c("data","ETS(A,Ad,A)", "ETS(M,Ad,A)", "ETS(A,A,A).s"))
```


Forecasts from ETS(A,Ad,A)
data
ETS(A,Ad,A)
ETS(M,Ad,A)
ETS(A,A,A).s
Value
3
4
5
6
7
8
Year
1995
2000
2005
2010
2015

Summary

In this module, we focused on exponential smoothing methods that provide a bunch of models to handle trend and seasonality in the series using additive and multiplicative seasonal and damped and exponential trend patterns. We considered 15 different exponential smoothing approaches.

Then, we moved on to state-space model formulation of the exponential smoothing models which enriches the class of exponential smoothing models by introducing additive and multiplicative error terms.

In total, we can fit any of 30 different exponential smoothing models by going over the state-space model formulation. Also, we considered auto-selection of the state space models for exponential smoothing.

References

Hyndman, R.J., Koehler, A.B., Ord, J.K., and Snyder, R.D. (2008). Forecasting with exponential smoothing: the state space approach, Springer-Verlag.

Hyndman, R.J., Athanasopoulos, G. (2014). *Forecasting: Principles and Practice*. OTEXTS.

Thanks for your attendance! Please use Socrative.com with room *FORECASTINGPG* to give feedback!