Time Series Analysis and Forecasting

Problem set 2: Dynamic Models for Prediction

Emilie Krutnes Engen

Master in Big Data Analytics
Carlos III University of Madrid

Universidad
Carlos III de Madrid

Exercise 1

In this exercise we are examining the daily sales of paperback books at a particular book store. The aim is to investigate the characteristics of the simple exponential smoothing method by performing a four days' sales forecast.

Exponential smoothing is a forecasting method where future observations are calculated based on the weighted averages of past observations, with the weights decaying exponentially the further back the observations go. This imply that more recent observation have higher associated weights. Mathematically, the forecasts are defines according to the following equations.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots + \alpha(1-\alpha)^{T-1} y_1 \tag{1}$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. The one-step forecast at time $T+1$ is the weighted average of all the observations $y_1, \ldots, y_T$. The decreasing rate of the weights are determined by the parameter α.

(a) Time series plot

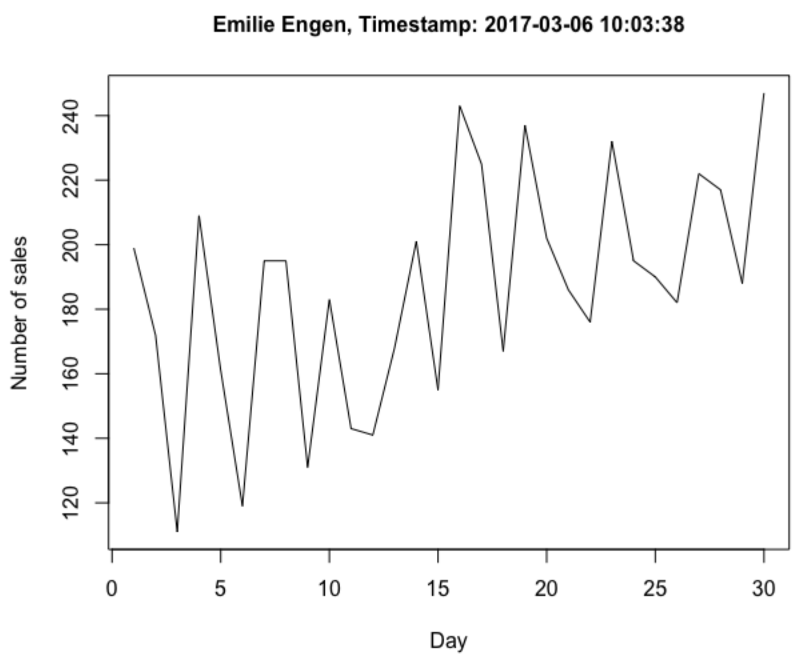

Figure 1: A time series plot of paperback sales in a given store for a period of 30 days

Before selecting a forecasting method we plot the series data to get a closer look at the main features of the data set. The following R command provides the time series plot.

```
plot(books[,1], xlab = 'Day', ylab = 'Number of sales', main = '')
```

The plot in Figure 1 display a clear trend and seasonal pattern.

(b) Simple exponential smoothing

Although the time series plot suggest appearance of both trend and seasonal pattern, we apply the simple exponential smoothing (SES) method to investigate the impact of the smoothing parameter, α. We select six different values for the smoothing parameter within the range $\alpha \in [0, 1]$. The differente forecasts are plotted by applying the following R code.

```
# Define paperback books
paperback <- books[,1]

# Fit a forecasting model with exponential smoothing using different alphas
fit1 <- ses(paperback, alpha = 0.0, initial = 'simple', h = 4)
fit2 <- ses(paperback, alpha = 0.2, initial = 'simple', h = 4)
fit3 <- ses(paperback, alpha = 0.4, initial = 'simple', h = 4)
fit4 <- ses(paperback, alpha = 0.6, initial = 'simple', h = 4)
fit5 <- ses(paperback, alpha = 0.8, initial = 'simple', h = 4)
fit6 <- ses(paperback, alpha = 1.0, initial = 'simple', h = 4)

# Plot the model
plot(fit1, plot.conf=FALSE, ylab='Number of sales', xlab='Day', main='',
    fcol='white', type='o')
lines(fitted(fit1), col=5, type='o')
lines(fitted(fit2), col=3, type='o')
lines(fitted(fit3), col=4, type='o')
lines(fitted(fit4), col=2, type='o')
lines(fitted(fit5), col=6, type='o')
lines(fitted(fit6), col=7, type='o')

# Plot the forecasts
lines(fit1$mean, col=5, type='o')
lines(fit2$mean, col=3, type='o')
lines(fit3$mean, col=4, type='o')
lines(fit4$mean, col=2, type='o')
lines(fit5$mean, col=6, type='o')
lines(fit6$mean, col=7, type='o')

legend('topleft',lty=1, col=c(1,5,3,4,2,6,7),
        c('data', expression(alpha == 0.0), expression(alpha == 0.2),
        expression(alpha == 0.4), expression(alpha == 0.6),
        expression(alpha == 0.8), expression(alpha == 1)), pch=1, cex=0.8)
```

The forecasts are presented in Figure 2. From the plot it is clear that for $\alpha = 0$ the method is simply the same as the average method, because each observation is effectively given the same weight. On the contrary, for $\alpha = 1$ the method corresponds to the naïve method, where all forecasts are equal to the last observed value of the series. Following this argument, if α

is small more weight is given to the observations from the distance past, while for a large α more weight is given to recent observations.

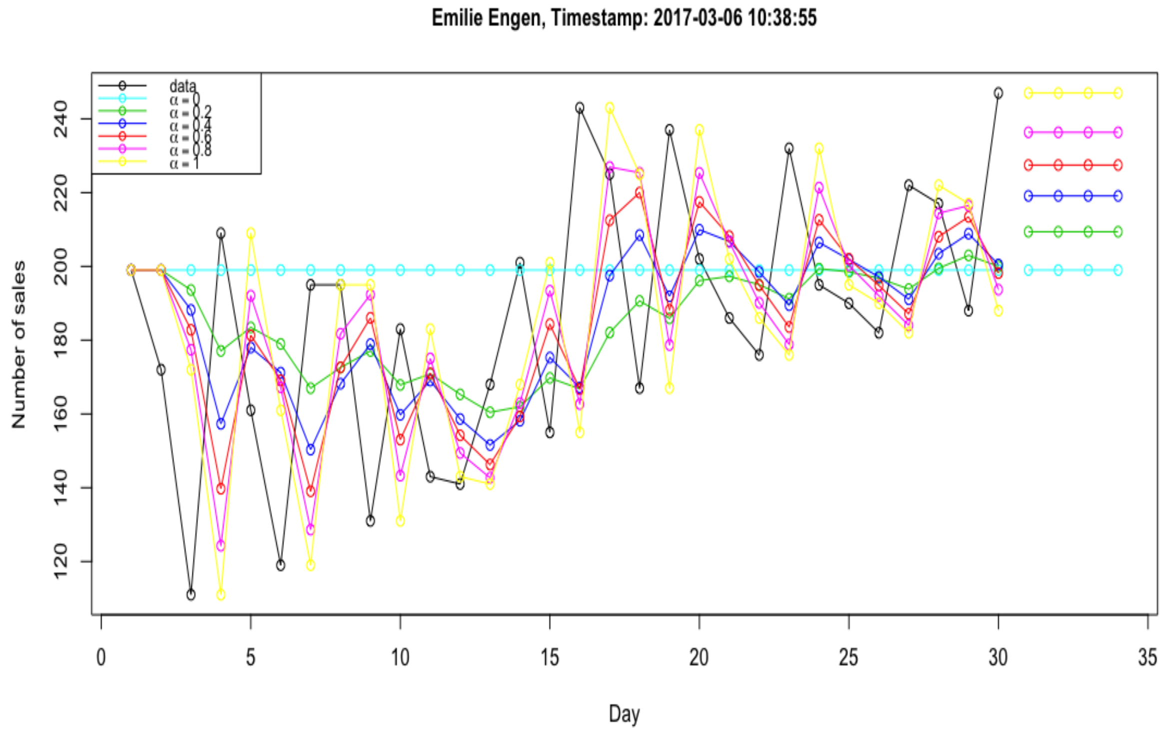

Figure 2: A four days forecast of paperback sales using simple exponential smoothing with different values for the smoothing parameter

We evaluate the accuracy associated with each of the selected values for α by recording the Sum of Squares Error (SSE). The SSE is computed and plotted by running the following R commands.

```
# Compute the SSEs
sse1 = fit1$model$SSE;sse1
sse2 = fit2$model$SSE;sse2
sse3 = fit3$model$SSE;sse3
sse4 = fit4$model$SSE;sse4
sse5 = fit5$model$SSE;sse5
sse6 = fit6$model$SSE;sse6

sse_matrix = c(sse1,sse2,sse3,sse4,sse5,sse6)
alpha_matrix = c(0,0.2,0.4,0.6,0.8,1)

# Plot the SSE values with respect to alpha
plot(alpha_matrix,sse_matrix, ylab='SSE',
    xlab='The smoothing parameter, alpha', col='blue', pch=21, bg='blue')
```

The plot in Figure 3 show the SSE for the different values of α. We optimize the simple exponential smoothing by minimizing the SSE with respect to the α parameter. From the plot the optimal value for α is close to 0.2.

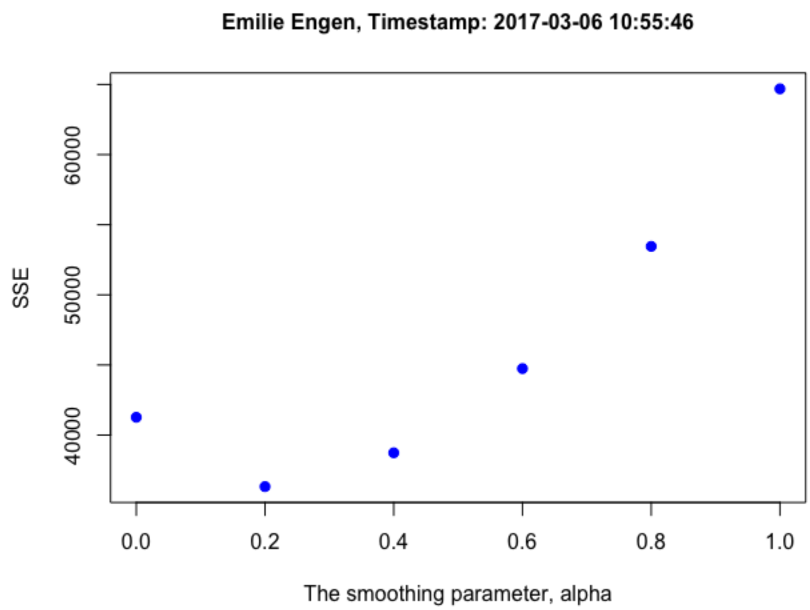

Figure 3: Plot of the Sum of Squares Error (SSE) for different values of the alpha parameter

(c) Optimal parameter selection and forecast generating

Instead of searching for the optimal α through trial and error we can instead let the *ses* function in R select the optimal α. The corresponding R code is provided below.

```
fit_opt <- ses(paperback, initial = 'simple', h = 4)
```

When removing the declaration of *alpha* we are simply asking the function to select the α that minimize the SSE. This value is obtained by running the following R code.

```
summary(fit_opt)
```

The forecast for the optimal smoothing parameter, $\alpha = 0.2125$ is presented in Figure 4. This plot is produced by the following R code.

```
plot(fit1, plot.conf=FALSE, ylab='Number of sales', xlab='Day', main='',
    fcol='white', type='o')
lines(fitted(fit_opt), col=2, type='o')
lines(fit_opt$mean, col=2, type='o')
legend('topleft',lty=1, col=c(1,2), c('data', expression(alpha == 0.2125)),
        pch=1, cex=0.6)
```

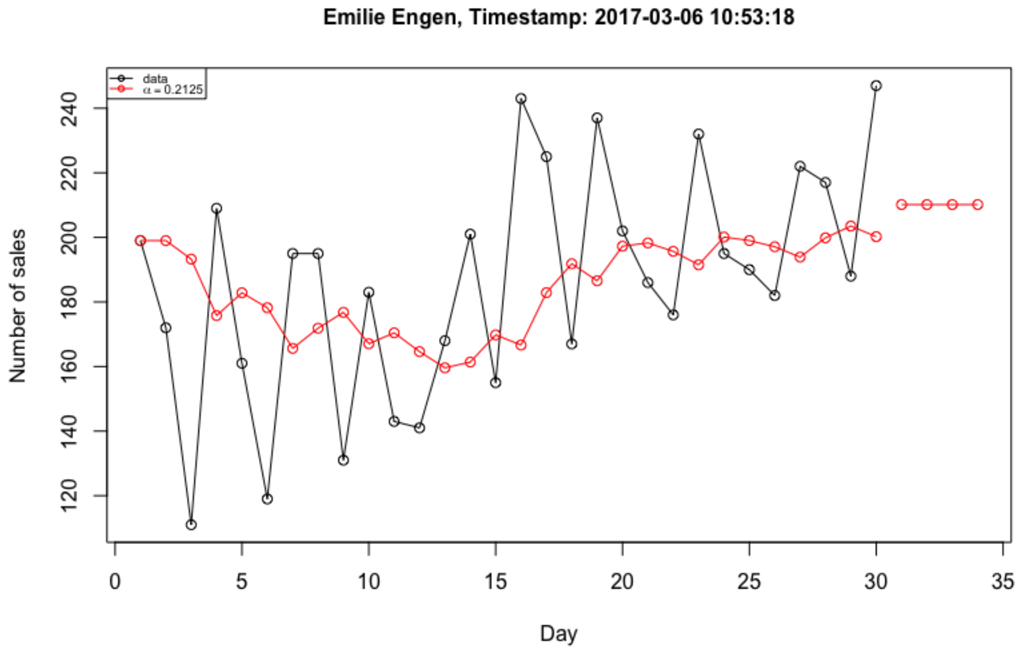

Figure 4: A four days forecast of paperback sales using simple exponential with optimal value for the alpha parameter

The plot in Figure 5 show the SSE for different values of α, where the red point is the optimal α. As previously suspected the optimal value is close to 0.2. The new point is included by running the following R code.

```
sse_opt = fit_opt$model$SSE;sse_opt
plot(alpha_matrix, sse_matrix, ylab='SSE',
     xlab='The smoothing parameter, alpha',col='blue', pch=21, bg='blue')
points(0.2125,sse_opt,pch=21,col=2,bg=2)
```

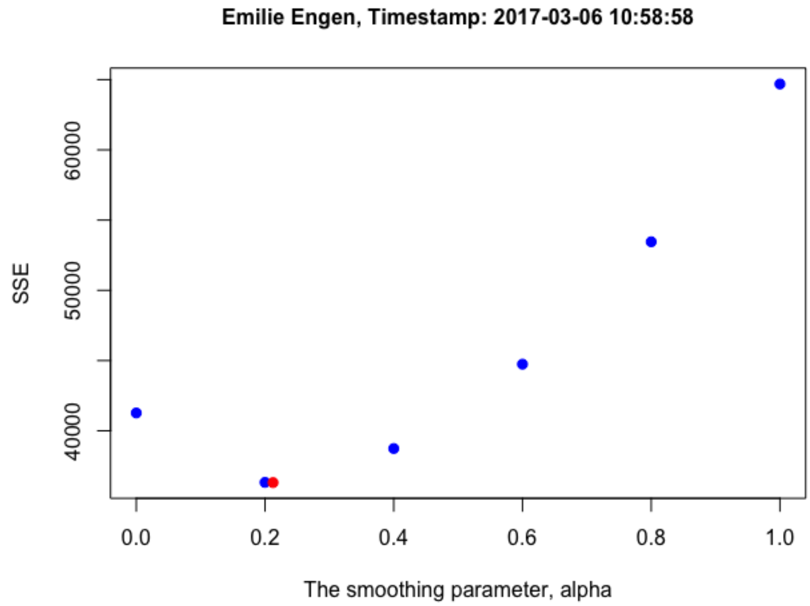

Figure 5: Plot of the Sum of Squares Error (SSE) for different alphas where the red point is the optimal alpha

Exercise 2

In this exercise we are analyzing the monthly short-term overseas visitors in Australia. The data is collected between May 1985 and April 2005.

(a) Time series plot

Before selecting a forecasting method we plot the series data to get a closer look at the main characteristics of the data set. The following R command provides the time series- and seasonal plot, respectively.

```
plot(visitors, main='', ylab='Thousands of people', xlab='Year')
seasonplot(visitors,main='', ylab='Thousands of people', xlab='Month',
            year.labels=TRUE, year.labels.left=TRUE, col=1:7, pch=19)
```

The plots indicate both a trend and seasonality. The Holt-Winters' method is therefore considered appropriate for forecasting the future number of visitors.

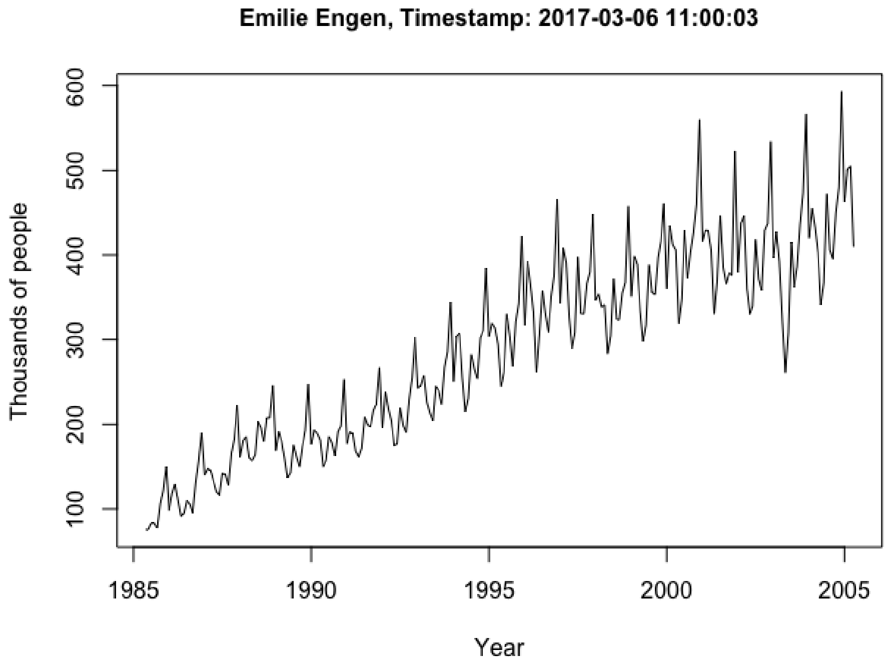

Figure 6: A time series plot of monthly visitors to Australia between 1985 and 2005

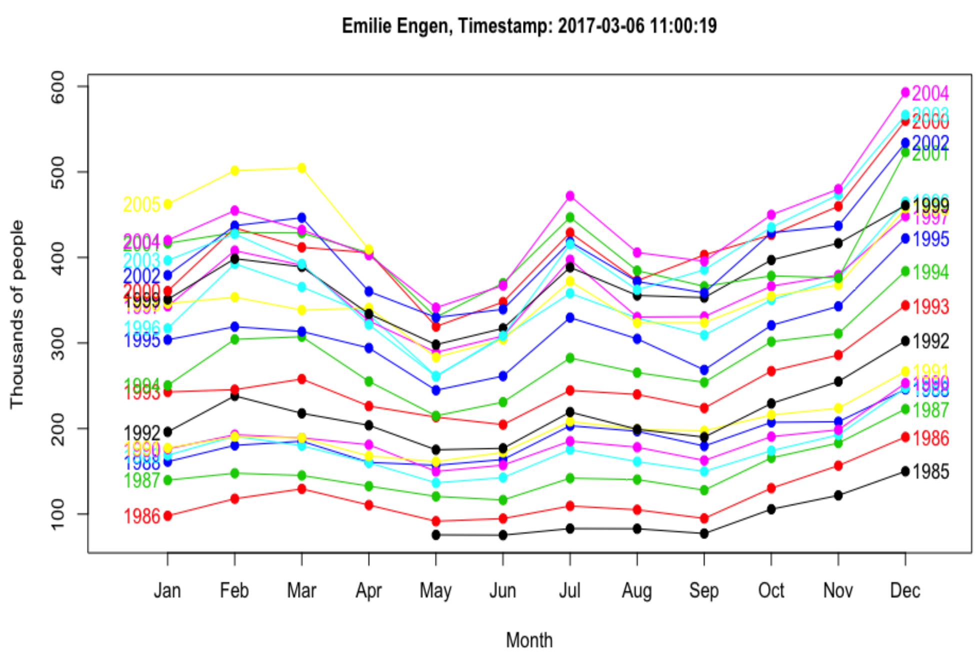

Figure 7: A seasonal plot of monthly visitors to Australia between 1985 and 2005

(b) Forecasting with Holt-Winters' multiplicative method

Holt-Winters' method in an extension of Holt's method that account for both trend and seasonality. The forecast equation in Holt-Winters' include three smoothing equations: one for the level ℓ_t, one for trend b_t, and one for the seasonal component s_t. Each equation have an associated smoothing parameter, denoted α, β^* and γ, respectively.

There are two variations of the Holt-Winters' method: the additive and the multiplicative method. The two methods differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant over time, while the multiplicative method capture proportional changes in the seasonal variations with respect to time. The multiplicative method can be expressed in component form by the following equations.

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t-m+h_m^+} \tag{2}$$

$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{3}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{4}$$

$$s_t = \gamma \frac{y_t}{\ell_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m} \tag{5}$$

where m is the number of seasons and $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$, ensuring that the estimates of the seasonal indices come from the final year of the sample. We apply and plot the two-year forecast with Holt-Winters' multiplicative method by using the following R code.

```
# Fit a forecasting model with HW multiplicative method
fit1 <- hw(visitors,seasonal='multiplicative')

# Plot the model
plot(fit1, plot.conf=FALSE, ylab='Thousands of people', xlab='Year', main='',
    fcol='white')
lines(fitted(fit1), col=4, lty=2)
lines(fit1$mean, col=4)
legend('topleft', lty=1, col=c(1,4), c('data', 'Holt-Winters Multiplicative'),
        pch=1, cex=0.8)
```

The two-year forecast of monthly visitors in Australia is presented in Figure 8.

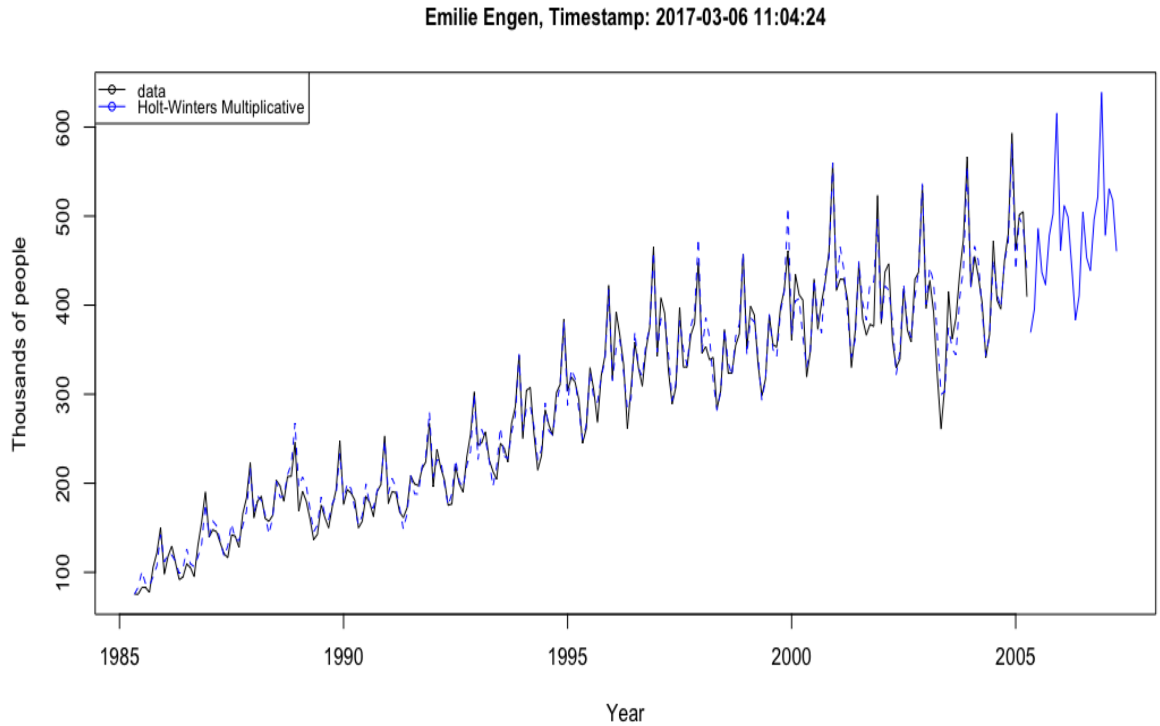

Figure 8: A two-year forecast of monthly visitors to Australia using Holt-Winters' multiplicative model

(c) Multiplicative seasonality

The multiplicative method is, as previously described, preferred when the time series show increasing changes in the seasonal variations as the level of the series increases. The plot in Figure 8 indicate an increasing seasonal variation over the time series. As a result, the Holt-Winters' multiplicative method is considered more appropriate as it provided a better fit for the data than the additive method.

(d) Trend characteristics

We now investigate further extensions of the model by either an exponential or damped trend, or both. The model is fitted and plotted by apply the following R commands.

```
fit1 <- hw(visitors,seasonal='multiplicative')
fit2 <- hw(visitors,seasonal='multiplicative',exponential=TRUE)
fit3 <- hw(visitors,seasonal='multiplicative',damped=TRUE)
fit4 <- hw(visitors,seasonal='multiplicative',exponential=TRUE,damped=TRUE)

plot(fit1, plot.conf=FALSE, ylab='Thousands of people', xlab='Year', main='',
    fcol='white')
lines(fitted(fit1), col=4, lty=2)
lines(fitted(fit2), col=5, lty=2)
lines(fitted(fit3), col=3, lty=2)
lines(fitted(fit4), col=2, lty=2)

lines(fit1$mean, col=4)
lines(fit2$mean, col=5)
lines(fit3$mean, col=3)
lines(fit4$mean, col=2)
legend('topleft',lty=1, col=c(1,4,5,3,2), c('data','Holt-Winters
        Multiplicative','Exponential Trend','Damped Trend','Exp and Damped
        Trend'), pch=1, cex=0.8)
```

Here *fit1* is the original multiplicative Holt-Winters' model, while *fit2* consider an exponential trend, *fit3* a damped trend, and *fit4* both an exponential and damped trend. The forecasts are presented in Figure 9.

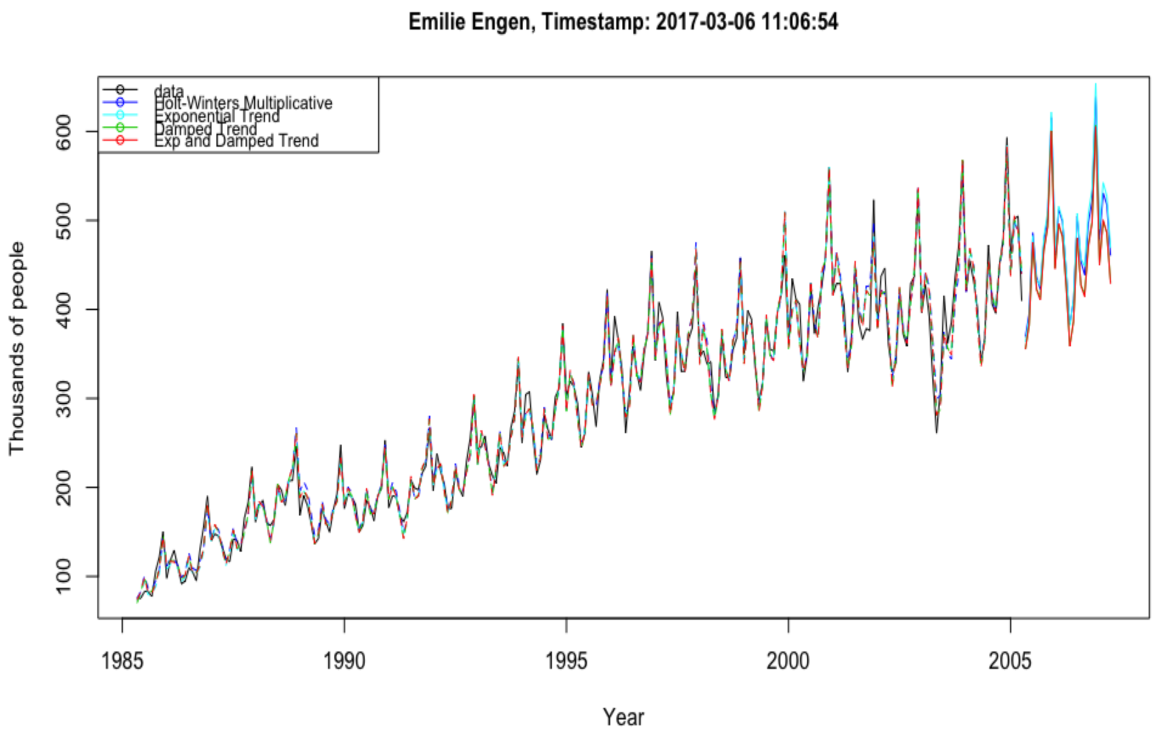

Figure 9: A two year forecast of monthly visitors to Australia using Holt-Winters' multiplicative model including exponential and damped trend

The plot presented in Figure 9 show that the methods provide similar forecast. We proceed by analyzing the accuracy of the different methods, given by the following R code.

```
a1<-accuracy(fit1);a1
a2<-accuracy(fit2);a2
a3<-accuracy(fit3);a3
a4<-accuracy(fit4);a4
```

The error measures for the different methods are presented in Table 1. The different methods all show very similar accuracy. Depending on the error measure used different methods provide the highest accuracy. For instance Holt-Winters' multiplicative method with damped trend provide a slightly lower Root Mean Squared Error (RMSE) than the other methods, but the Holt-Winters' with damped trend have a slightly higher accuracy when considering the Mean Average Precentage Error (MAPE).

Error Measures				
	H-W	Exp. H-W	Dmp. H-W	Exp. & Dmp. H-W
ME	-0.95	0.64	0.91	0.72
RMSE	14.83	14.49	14.45	14.46
MAE	10.97	10.63	10.65	10.73
MPE	-0.82	0.26	0.07	0.04
MAPE	4.27	4.03	4.06	4.09
MASE	0.41	0.39	0.39	0.40
ACF1	0.22	0.08	0.02	0.01

Table 1: Error measures for the multiplicative Holt-Winters' method with different trend equations

(e) Fitting forecasting models

We now proceed our analysis of the visitors data by considering three different approaches: an ETS, an Additive ETS applied to a Box-Cox transformed series STL decomposition applied to the Box-Cox transformed data followed by an ETS model applied to the seasonally adjusted data. Fitting these three models are performed by the provided R code below.

```
# (1) an ETS model
fit5 <- ets(visitors)

# (2) an additive ETS model applied to a Box-Cox transformed series
lambda <- BoxCox.lambda(visitors)
fit6 <- ets(visitors, model='AAZ', lambda=lambda)

# (3) an STL decomposition applied to the Box-Cox transformed data followed by
    an ETS
# model applied to the seasonally adjusted (transformed) data.
stl <- stlm(visitors, robust=TRUE, method=c('ets'), lambda=lambda)
fit7 <- forecast(stl, lambda=stl$lambda)
```

(f) Plotting the forecasting models

The forecasts using these three methods are plotted by applying the following R code.

```
plot(forecast(fit5, h=24), plot.conf=FALSE, ylab='Thousands of people', xlab='
   Year', main='', fcol='white')
lines(fitted(fit5), col=4, lty=2)
lines(fitted(fit6), col=5, lty=2)
lines(fitted(fit7), col=2, lty=2)

lines(forecast(fit5)$mean, col=4)
lines(forecast(fit6)$mean, col=5)
lines(fit7$mean, col=2)
legend('topleft',lty=1, col=c(1,4,5,2), c('data','ETS','Additive ETS','STL and
    ETS'),pch=1,cex=0.8)
```

The forecasts are presented in Figure 10

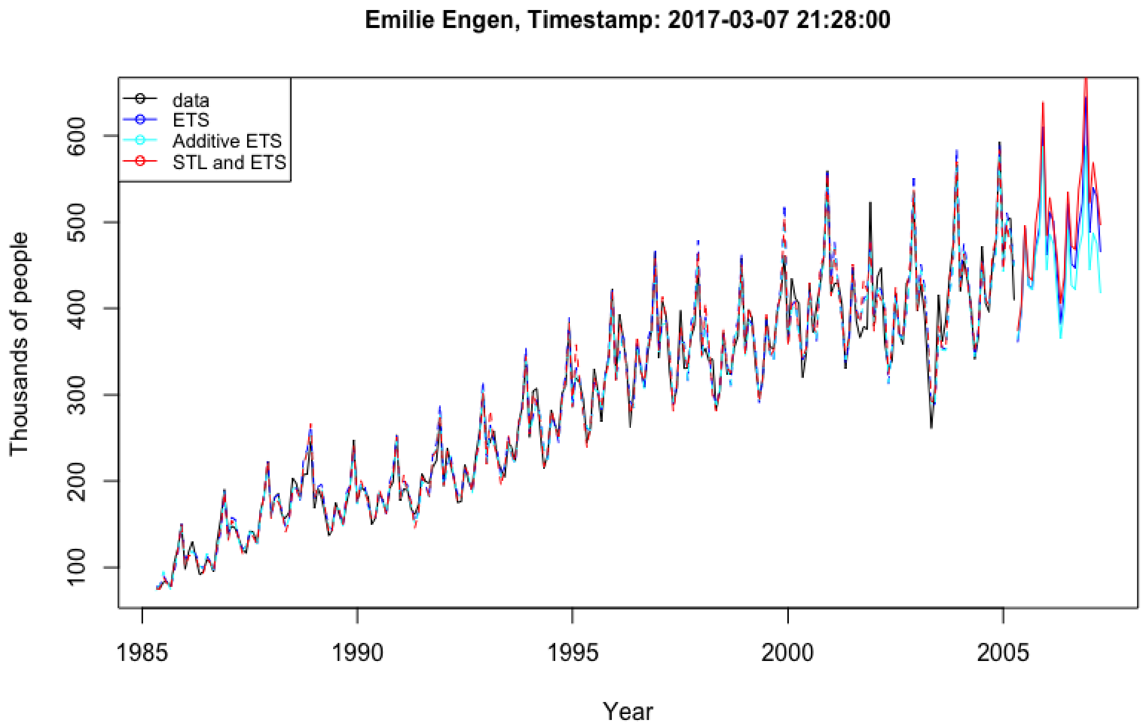

Figure 10: A two year forecast of monthly visitors to Australia using ETS, additive ETS on Box-Cox transformed data and including STL decomposition

(e) Residual diagnostics

For each of the proposed models, we investigate the residual diagnostics and compare the forecasts for the next two years. The residual plots for the ETS model are presented in Figure 11, for the additive ETS in Figure 12 and for the STL decomposition with ETS in Figure 13. The residuals, ACF and PACF are plotted for each method by applying the following R code.

```
# (1) Residuals plot, ACF an PACF for ETS
res5 <- residuals(fit5);res5
tsdisplay(res5, main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)
```

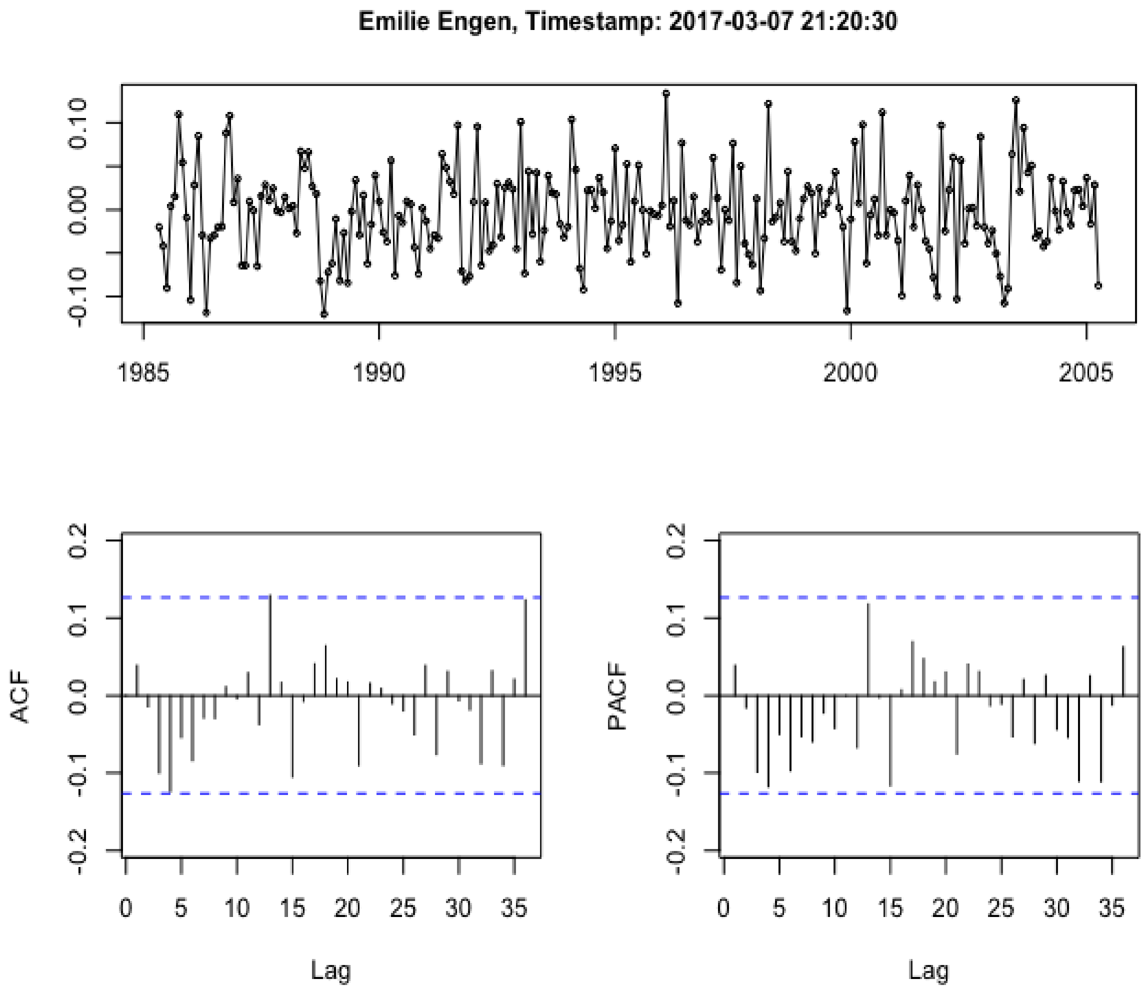

Figure 11: Residuals, ACF and PACF plots for the ETS model

```
# (2) Residuals plot, ACF an PACF from additive ETS
res6 <- residuals(fit6);res6
tsdisplay(res6, main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)
```

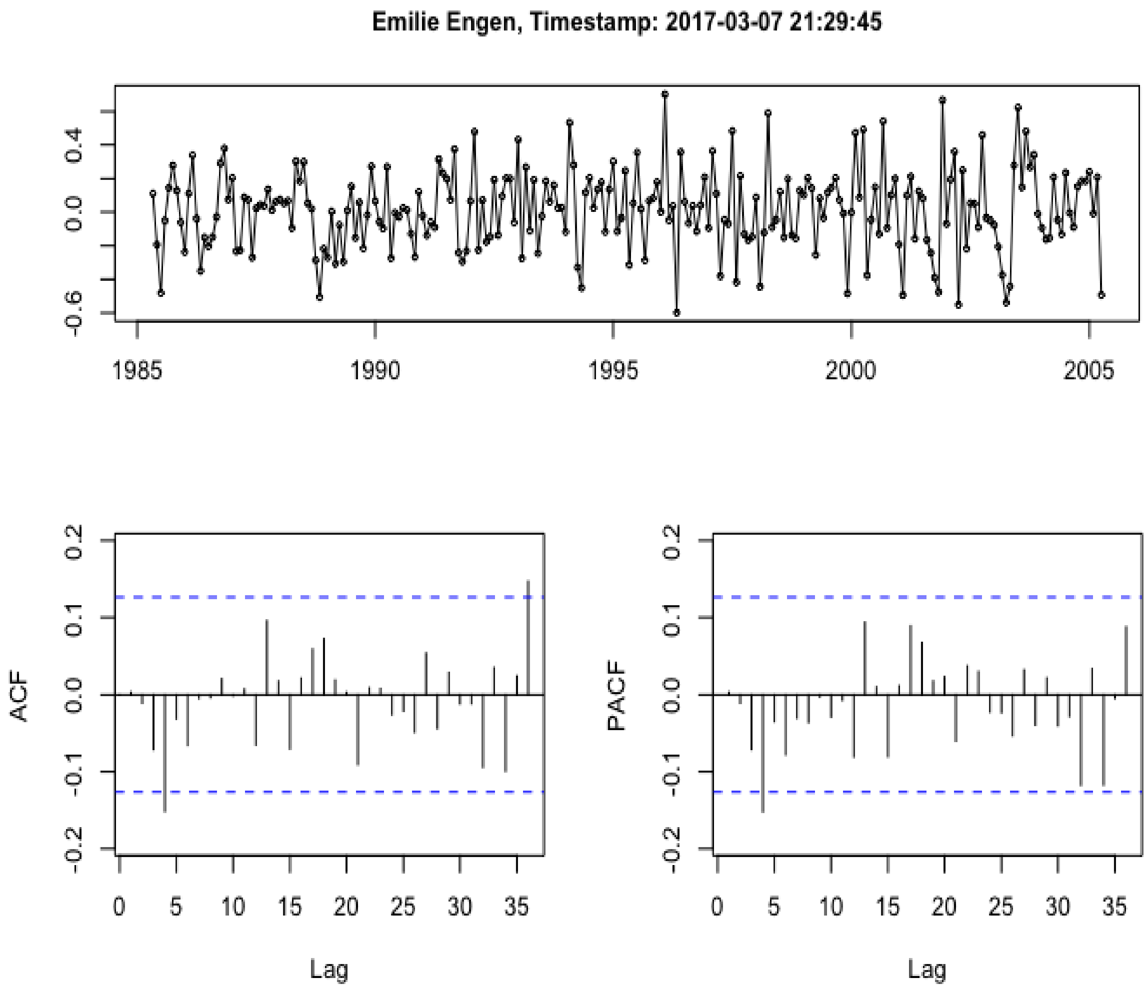

Figure 12: Residuals, ACF and PACF plots for the additive ETS model with Box-Cox transformed data

```
# (3) Residuals plot, ACF an PACF for STL and ETS
res7 <- residuals(fit7);res7
tsdisplay(res7, main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)
```

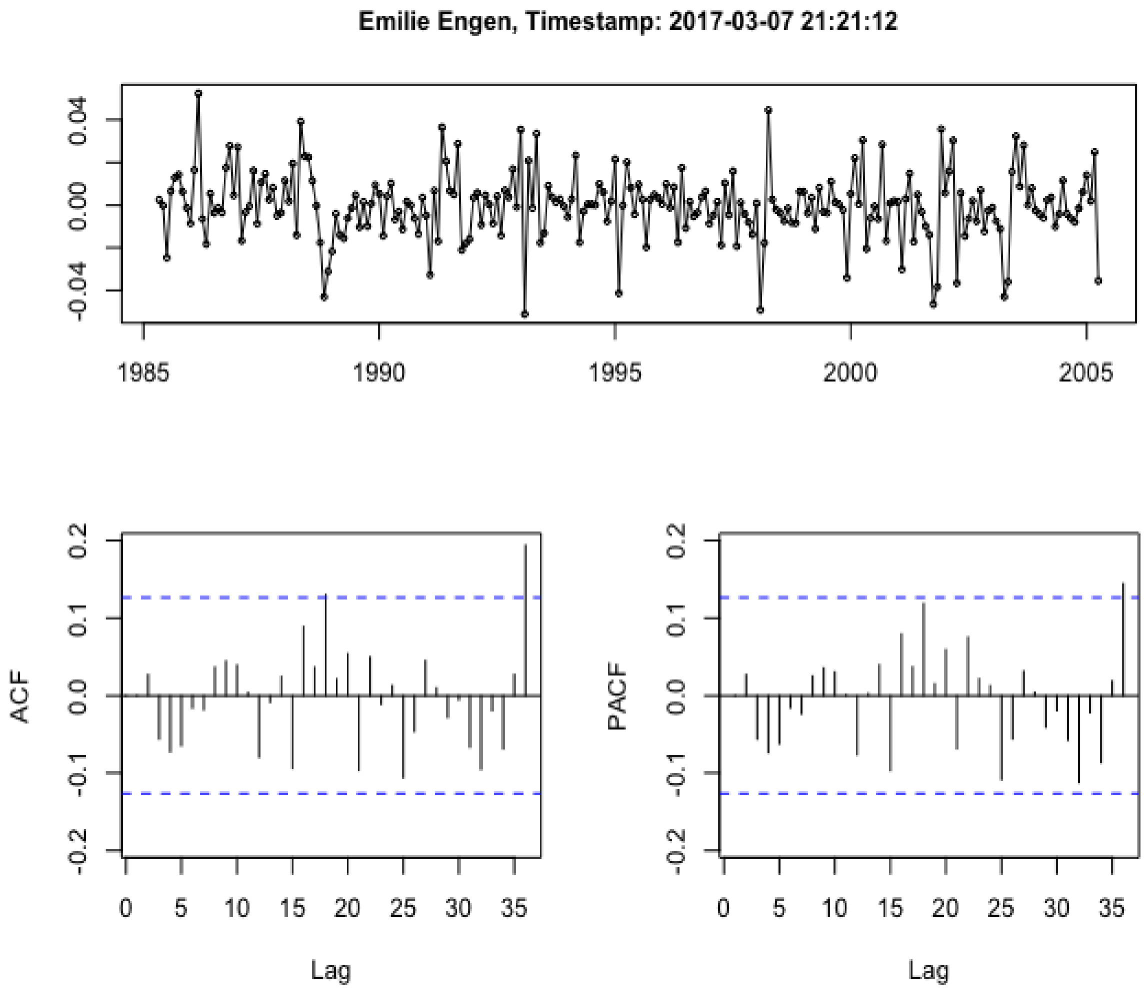

Figure 13: Residuals, ACF and PACF plots for the additive ETS model with STL decomposition and Box-Cox transformed data

From the residual diagnostic plots it seems as if the ETS model should be preferred as the residuals are unbiased and homoscedastic. The variation in the residuals are close to constant over the time series, while the STL model clearly does not have constant variation in the residuals. We further see from the histograms in Figure 14 that the two other methods seems to have residuals that are slightly biased.

We have also plotted the histograms for the three models by applying the following R code.

```
par(mfrow = c(1, 3))
# Histogram of residuals ETS
hist(res5, breaks=20, nclass='FD', main='',xlab='Residuals',col='light blue')
title(main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)

# Histogram of residuals additive ETS
hist(res6, breaks=15, nclass='FD', main='',xlab='Residuals',col='light blue')
title(main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)

# Histogram of residuals STL and ETS
hist(res7, breaks=15, nclass='FD', main='',xlab='Residuals',col='light blue')
title(main=paste('Emilie Engen, Timestamp:',Sys.time()),cex.main=1)
```

The histograms of the residuals for the three models are presented in Figure 14. Of the three models, we have that both the ETS model and the STL decomposed model show a close to normal distribution of the residuals. The distribution for the additive ETS model is slightly right skewed.

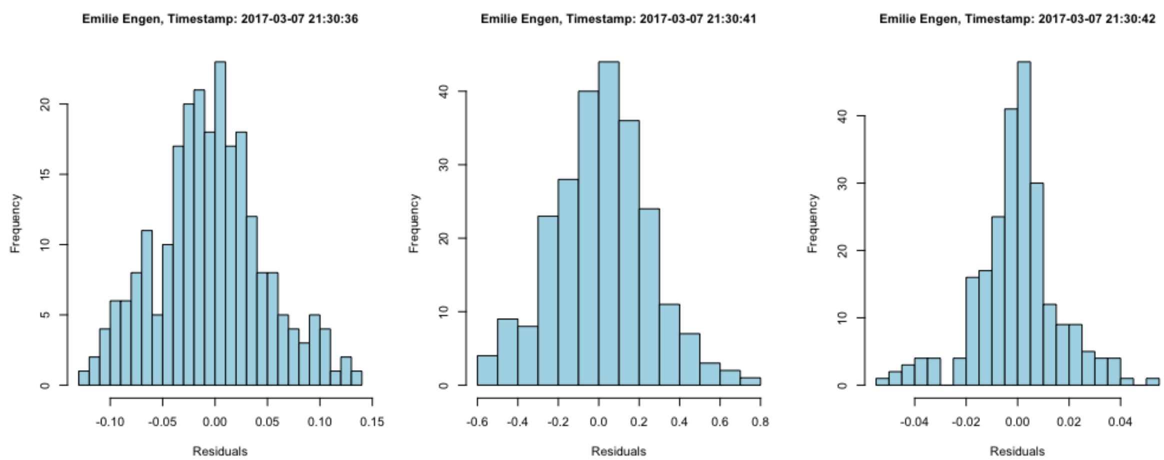

Figure 14: Histogram of residuals for ETS, additive ETS with Box-Cox transformed data and including STL decomposition