Time Series Analysis - 478 - HW #4

Alex Towell (atowell@siue.edu)

Problem 1

Suppose that simple exponential smoothing is being used to forecast a process, $y_t = \mu + \epsilon_t$. However, at the start of period t^*, the mean of the process shifts to a new mean level $\mu + \delta$. The mean remains at this new level for subsequent time periods. Calculate the expected value of the simple exponential moving average. (Note: you need to discuss cases for $T > t^*$ and $T < t^*$.)

In the simple exponential moving average, we model the mean of the time series with

$$\hat{y}_t = \frac{1-\theta}{1-\theta^t} \sum_{j=0}^{t-1} \theta^j y_{t-j}.$$

The expected value is

$$\mathrm{E}(\hat{y}_t) = \frac{1-\theta}{1-\theta^t} \sum_{j=0}^{t-1} \theta^j \, \mathrm{E}(y_{t-j}).$$

Given $T < t^*$, the process is $y_T = \mu + \epsilon_T$ and given $T \geq t^*$, the process is $y_T = \mu + \delta + \epsilon_T$.

Letting $T = t - j$ and solving for j in $t - j < t^*$, we see that we may partition the summation over $j \geq t - t^*$ and $j < t - t^*$,

$$\begin{aligned}
\mathrm{E}(\hat{y}_t) &= \frac{1-\theta}{1-\theta^t} \left\{ \sum_{j=0}^{t-t^*-1} \theta^j (\mu + \delta) + \sum_{j=t-t^*}^{t-1} \theta^j \mu \right\} \\
&= \frac{1-\theta}{1-\theta^t} \left\{ \sum_{j=0}^{t-t^*-1} \theta^j \delta + \sum_{j=0}^{t-1} \theta^j \mu \right\}.
\end{aligned}$$

Expanding the product,

$$\begin{aligned}
\mathrm{E}(\hat{y}_t) &= \frac{1-\theta}{1-\theta^t} \delta \sum_{j=0}^{t-t^*-1} \theta^j + \mu \\
&= \frac{1-\theta}{1-\theta^t} \delta \frac{1-\theta^{t-t^*}}{1-\theta} + \mu \\
&= \frac{1-\theta^{t-t^*}}{1-\theta^t} \delta + \mu \\
&= \left(\frac{1}{1-\theta^t} - \frac{\theta^{t-t^*}}{1-\theta^t} \right) \delta + \mu.
\end{aligned}$$

For large t, $\theta^t \approx 0$, and so we simplify the above to

$$\mathrm{E}(\hat{y}_t) = \left(1 - \theta^{t-t^*} \right) \delta + \mu.$$

We consider two degenerate cases given by $t^* = t$ and $t^* = 0$ which respectively have expectations given by $\mathrm{E}(\hat{y}_t) = \mu$ and $\mathrm{E}(\hat{y}_t) = \sigma + \mu$ (when we use the assumption that $\theta_t \approx 0$).

Problem 2

Consider the time series $\{y_T\} = (14, 19, 18, 22, 17, 28, 43, 45, 62, 60)$.

Parts (a)-(d)

We use the following R code to generate and populate the table.

```
discount <- 0.2
tsdata <- ts(c(14,19,18,22,17,28,43,45,62,60))

# part (a): calculate the simple moving average with span=3.
ma3 <- MA(tsdata,3)

# part (b): calculate the simple (1st order) exponential moving average using
# discount lambda = 0.2 and initial value y1.
#ema1 <- EMA(tsdata,tsdata[1],discount)
ema1 <- EMA(tsdata,14,discount)

# part (c): calculate the 2nd order exponential moving average using discount
# lambda = 0.2 and initial values y1.
ema2_data <- EMA2(tsdata,tsdata[1],tsdata[1],discount,discount)
#ema2_data <- EMA2(tsdata,tsdata[1],tsdata[1],discount,discount)
ema2 <- as.numeric(unlist(ema2_data[2]))

# part (d): calculate the (unbiased) linear trend estimator
yhat <- as.numeric(unlist(ema2_data[3]))

prob2 <- data.frame(
   t = 1:length(tsdata),
   yt = tsdata,
   ma = ma3,
   ema1 = ema1,
   ema2 = ema2,
   yhat = yhat,
   delta = tsdata-yhat)

# parts (a)-(d): generate the table with the previously computed values
knitr::kable(prob2,caption="Parts (a)-(d)",padding=-1L,
             col.names =c("time",
                          "$y_T$",
                          "MA","$\\hat{y}_T^{(1)}$",
                          "$\\hat{y}_T^{(2)}$",
                          "$\\hat{y}_T$",
                          "$y_T-\\hat{y}_T$"))
```

Table 1: Parts (a)-(d)

time	y_T	MA	$\hat{y}_T^{(1)}$	$\hat{y}_T^{(2)}$	$\hat{y}_T$	$y_T - \hat{y}_T$
1	14	NA	14.00000	14.00000	14.00000	0.000000
2	19	NA	15.00000	14.20000	15.80000	3.200000
3	18	17.00000	15.60000	14.48000	16.72000	1.280000
4	22	19.66667	16.88000	14.96000	18.80000	3.200000
5	17	19.00000	16.90400	15.34880	18.45920	-1.459200

time	y_T	MA	$\hat{y}_T^{(1)}$	$\hat{y}_T^{(2)}$	$\hat{y}_T$	$y_T - \hat{y}_T$
6	28	22.33333	19.12320	16.10368	22.14272	5.857280
7	43	29.33333	23.89856	17.66266	30.13446	12.865536
8	45	38.66667	28.11885	19.75389	36.48380	8.516198
9	62	50.00000	34.89508	22.78213	47.00803	14.991974
10	60	55.66667	39.91606	26.20892	53.62321	6.376792

Part (e)

Finish the table by calculate the errors. What is the SSE?

The sum of squared error (SSE) is computed and displayed by the following R code:

```
sse = sum((tsdata-yhat)^2)
cat("The sum of squared error is ", sse, ".")
```

```
## The sum of squared error is  562.0258 .
```

Note that this is not the forecast one-step sum of squared error. I believe you would specifically ask for that if that is what you wanted, and the computation of the one-step ahead SSE seemed a bit tedious.

Part (g)

Make a one-step-ahead forecast for Time=11 based on the linear trend process. If the true observation is 72, what's your prediction error? Also provide the prediction interval.

```
# here's how we can compute the one-step-ahead forecast.
# i decided to do this computation manually, rather than using the forecasting
# library.
b1hat <- (ema1[10] - ema2[10]) * discount / (1-discount)
y10_one_step <- yhat[10] + b1hat
print(y10_one_step)
```

```
## [1] 57.04999
```

We see that $\hat{y}_{11}(10) \approx 57$. If the true observation $y_{11} = 72$, then the 1-step forecast error is

$$e_{10}(1) = y_{11} - \hat{y}_{11}(10) = 72 - 57 = 15.$$

To compute the intervals, we use the forecast library.

```
library("forecast")
dEMA <- holt(tsdata,h=1,level=c(95),initial="simple",alpha=discount,beta=discount)
summary(dEMA)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = tsdata, h = 1, level = c(95), initial = "simple", alpha = discount,
##
##  Call:
```

```
##     beta = discount)
##
##   Smoothing parameters:
##     alpha = 0.2
##     beta  = 0.2
##
##   Initial states:
##     l = 14
##     b = 5
##
##   sigma:  9.2384
## Error measures:
##                      ME     RMSE     MAE       MPE    MAPE     MASE      ACF1
## Training set -0.5745289 9.238406 8.18886 -16.84568 31.0149 1.188706 0.5912006
##
## Forecasts:
##    Point Forecast    Lo 95    Hi 95
## 11       58.88679 40.77985 76.99373
```

```
# we got this estimate of one-step-ahead variance of
# the forecast errors from the holt procedure
rmse <- 9.238
lo <- y10_one_step - 1.96 * rmse
hi <- y10_one_step + 1.96 * rmse
print(y10_one_step)
```

```
## [1] 57.04999
```

```
print(lo)
```

```
## [1] 38.94351
```

```
print(hi)
```

```
## [1] 75.15647
```

The 95% prediction interval computed from the *forecast* library is

$$[40.78, 76.99].$$

As a slight twist, I thought I would use the estimate of the variance of the forecast errors from the *holt* procedure's output and apply it to the previous approach, resulting in the interval

$$[38.94, 75.16].$$

Observe that these prediction interval estimates differ, primarily due to the fact that the *holt* procedure uses a different set of initial values, but they are reasonably close.

Problem 3

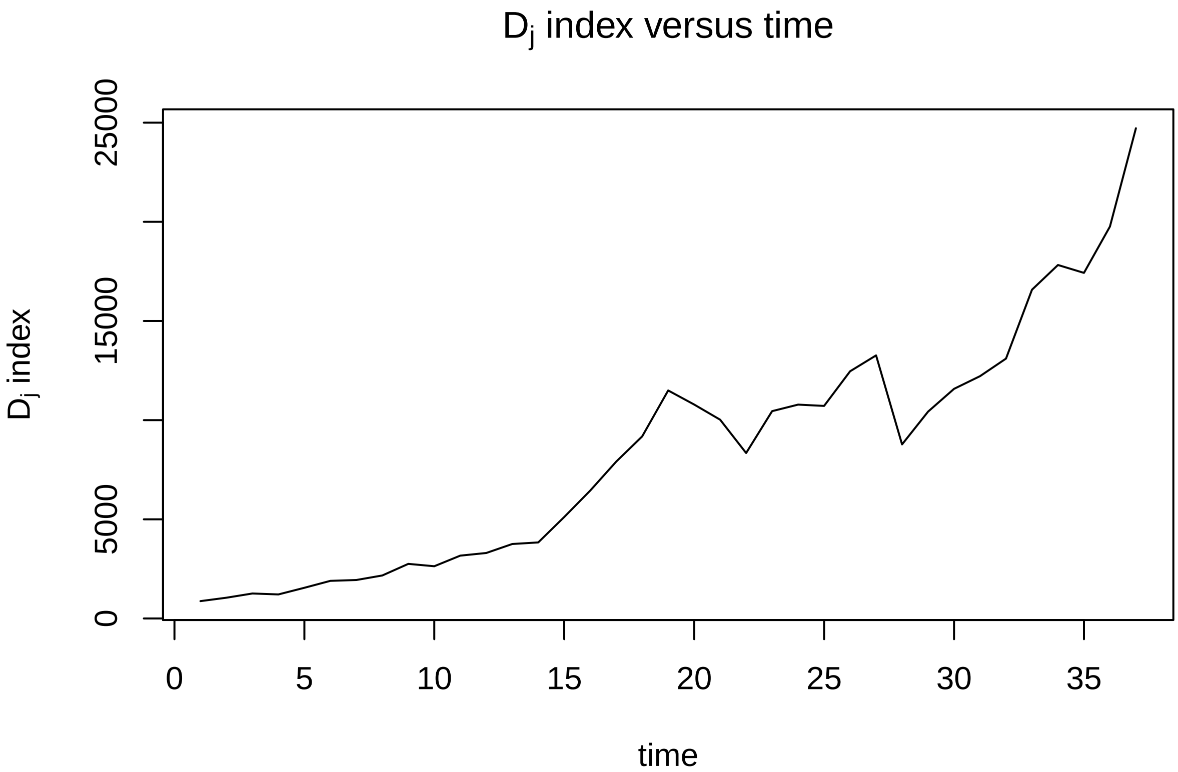
Consider the Dow Jones Index data on Blackboard. The dataset contains yearly Dow Jones closing index from year 1981 to 2016.

Part (a)

Read the data into R. Then construct a time plot (D_j index v.s. time).

The following R code is used to read the data file and generate the time series plot:

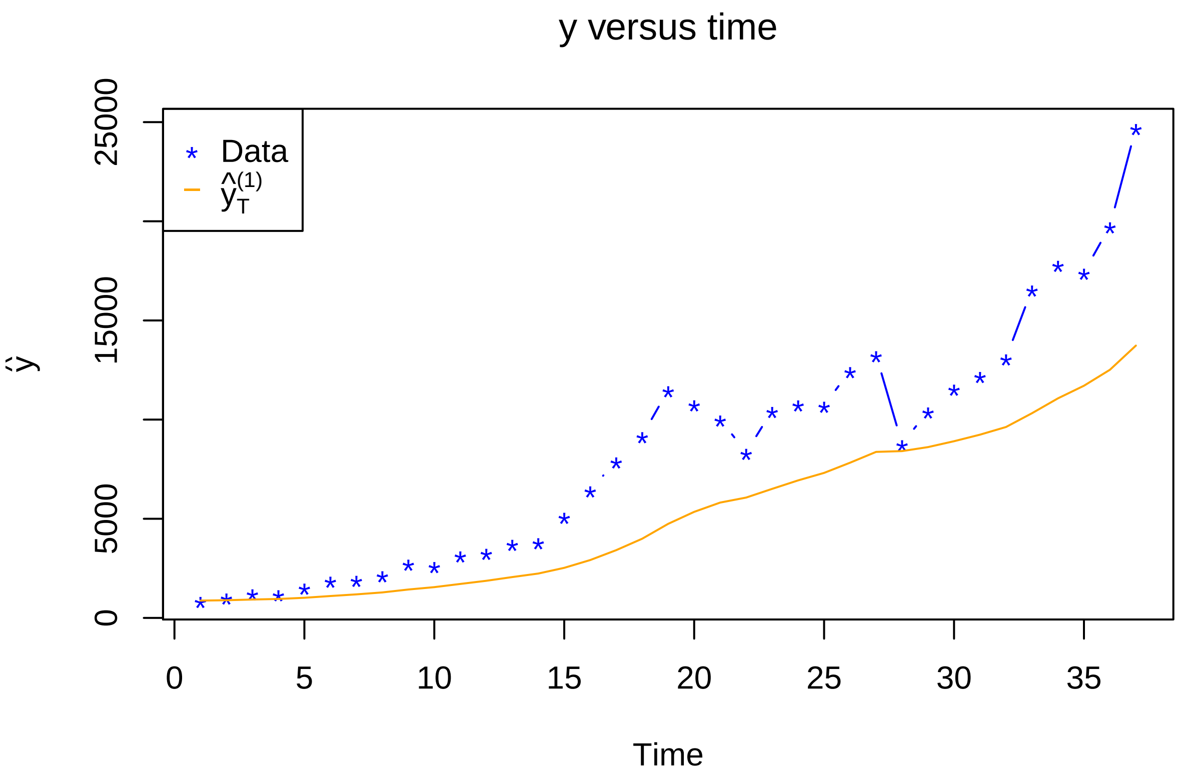
```
library(latex2exp)
data <- read.table("./DJI_yearly.txt", header=TRUE)
tsdata <- ts(data[,2])
plot(tsdata, main=TeX("$D_j$ index versus time"), xlab="time", ylab=TeX("$D_j$ index"))
```



Part (b)

Calculate the simple exponential moving average of the data using $\lambda = 0.1$, and initial value equals the 1st observation. Impose the 1st order EMA on the time plot. Comment on what you observe.

```
discount <- 0.1
ema1 <- EMA(tsdata,tsdata[1],discount)
plot(tsdata,type="b",col="blue",pch='*',ylab=TeX('$\\hat{y}$'), main=TeX('$y$ versus time'))
lines(ema1,type="l",col="orange")
legend("topleft",
  legend = c("Data", TeX("$\\hat{y}_T^{(1)}$")),
  col = c(
     "blue",
     "orange"),
  pch = c('*','-'))
```



The first-order EMA $\hat{y}_T^{(1)}$ has similiar growth to the time series data, but it consistently estimates values below it.

Part (c)

Calculate the sum of squared error (SSE) of your 1st order EMA.

The following R code is used to compute and display the SSE.

```
sse = sum((tsdata-ema1)^2)
cat("The sum of squared error is ", sse, ".")
```

```
## The sum of squared error is  588756025 .
```

Part (d)

Make a one-step-ahead forecast for year 2017 based on 1st order EMA. Also provide the prediction interval.

```
library("forecast")
sEMA=ses(tsdata, h=1, level=c(95), initial = "simple", alpha = discount)
summary(sEMA)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = tsdata, h = 1, level = c(95), initial = "simple", alpha = discount)
```

```
##
##   Smoothing parameters:
##     alpha = 0.1
##
##   Initial states:
##     l = 875
##
##   sigma:  4432.251
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE      ACF1
## Training set 3475.292 4432.251 3475.292 40.35451 40.35451 3.097379 0.6945515
##
## Forecasts:
##    Point Forecast   Lo 95    Hi 95
## 38       13733.58 5046.53 22420.63
```

First, note that the year 2017 corresponds to time unit 37 and the 1-step ahead forecast for 2018 is therefore time unit 38. Thus, we see that $\hat{y}_{38}^{(1)}(37) = 13733.58$. The 95% prediction interval is $[5046.53, 22420.63]$.

Part (e)

Fit a linear regression $y_t = \beta_0 + \beta_1 t + \epsilon_t$ with the data and report the OLS estimators $\hat{\beta}_0$ and $\hat{\beta}_1$.

We denote these estimators by $\hat{\beta}_{0,0}$ and $\hat{\beta}_{1,0}$, since they are from OLS given the data and the model $Y_t = \beta_0 + \beta_1 t + \epsilon_t$.

```
t <- 1:length(tsdata)
ols <- lm(tsdata~t)
b00=ols$coeff[1]
b10=ols$coeff[2]
print(ols)
```

```
##
## Call:
## lm(formula = tsdata ~ t)
##
## Coefficients:
## (Intercept)            t
##     -1620.7        527.4
```

We see that $\hat{\beta}_{0,0} = -1620.7$ and $\hat{\beta}_{1,0} = 527.4$.

Part (f)

Set the initial values for 2nd order smoothing based on part (e).

```
y01=b00-(1-discount)/discount*b10
y02=b00-2*(1-discount)/discount*b10
cat("y01 =", y01, "and y02 =", y02, ".")
```

```
## y01 = -6367.144 and y02 = -11113.57 .
```

We see that the initial values are given by $\hat{y}_0^{(1)} = -6367.144$ and $\hat{y}_0^{(2)} = -11113.57$.

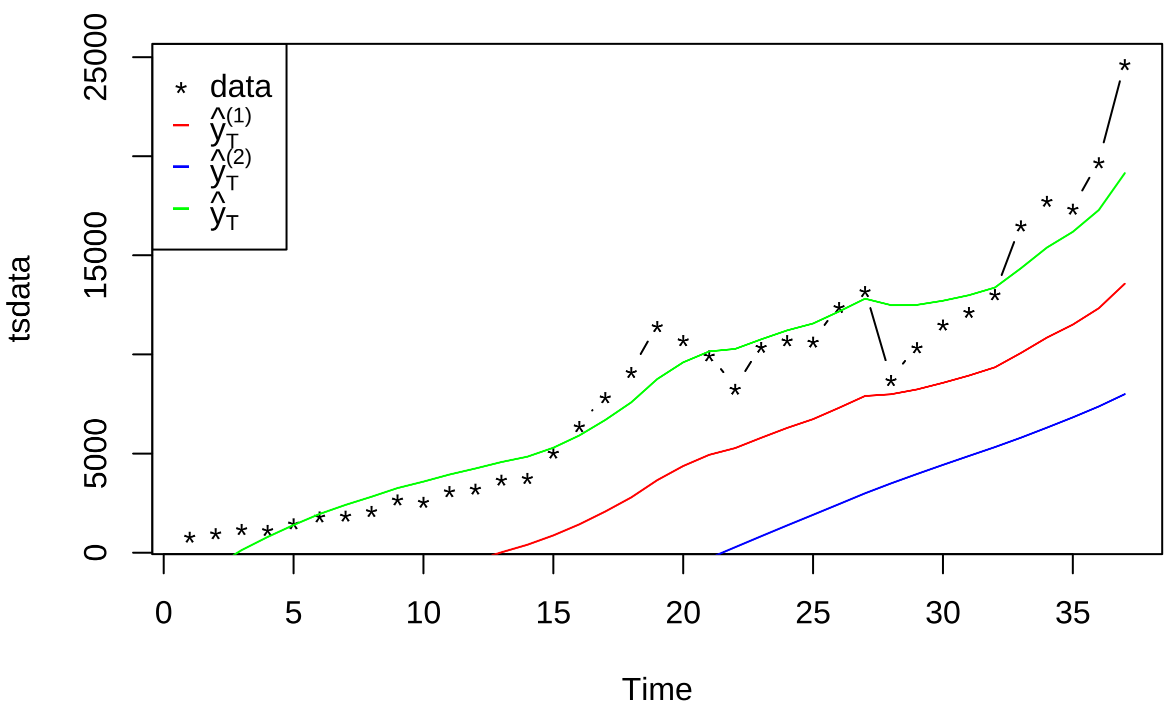
Part (g)

Calculate the 2nd order exponential moving average of the data using $\lambda = 0.1$, and use initial values in part (f). Calculate the unbiased estimates, by $\hat{y}_T = 2\hat{y}_T^{(1)} - \hat{y}_T^{(2)}$. Plot the estimates on the original time plot. Compare with the 1st order EMA approach. Comment on what you observe.

```
ema2 <- EMA2(tsdata,y01,y02,discount,discount)

plot(tsdata,col="black",type="b",pch='*')
lines(ema2$ema1,col="red")
lines(ema2$ema2,col="blue")
lines(ema2$yhat,col="green")

legend("topleft",
  legend = c(
     "data",
     TeX("$\\hat{y}_T^{(1)}$"),
     TeX("$\\hat{y}_T^{(2)}$"),
     TeX("$\\hat{y}_T$")),
  col = c(
     "black",
     "red",
     "blue",
     "green"),
  pch = c('*','-','-','-'))
```



We plot the data, a realization of $\{Y_t\}$, in black, $\hat{y}_T^{(1)}$ in red, $\hat{y}_T^{(2)}$ in blue, and $\hat{y}_T = 2\hat{y}_T^{(1)} - \hat{y}_T^{(2)}$ in green.

Each of the estimators exhibits a similiar trend. However, the most biased is the second-order smoother $\hat{y}_T^{(2)}$, $\hat{y}_T^{(1)}$ exhibits less bias, and $\hat{y}_T$ seems relatively unbiased, although the original time series may have a higher order component.

Part (h)

Calculate the sum of squared error (SSE) of your estimators in part (g).

```
print(sum((tsdata-ema2$ema1)^2))
```

```
## [1] 1063748664
```

```
print(sum((tsdata-ema2$ema2)^2))
```

```
## [1] 3721105725
```

```
print(sum((tsdata-ema2$yhat)^2))
```

```
## [1] 103638848
```

Note that I am assuming we are using the normal sum of squared errors, e.g., $\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_t)$.

We see that the SSE for $\hat{y}_T^{(1)}$ is 1063748664, the SSE for $\hat{y}_T^{(2)}$ is 3721105725, and the SSE for $\hat{y}_T$ is 103638848.

Part (i)

Make a one-step-ahead forecast for year 2017 based on 2nd order EMA. Also provide the prediction interval.

I'm going to use the *forecast* library, which will also compute a prediction interval.

```
step1=ses(tsdata, h=1, level=c(95), initial="simple", alpha=discount)
step2=ses(step1$fitted, h=1, level=c(95), initial="simple", alpha=discount)
summary(step2)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = step1$fitted, h = 1, level = c(95), initial = "simple",
##
##  Call:
##      alpha = discount)
##
##   Smoothing parameters:
##     alpha = 0.1
##
##   Initial states:
##     l = 875
##
##   sigma:  2527.742
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE      ACF1
## Training set 2009.604 2527.742 2009.604 35.28082 35.28082 6.216361 0.9268477
##
## Forecasts:
##    Point Forecast   Lo 95    Hi 95
## 38       8310.534 3356.25 13264.82
```

We see that $\hat{y}_{38}^{(2)}(37) \approx 8311$ with a 95% prediction interval $[3356.25, 13264.82]$.

Note that we get much better results if we use the *holt* procedure, which corrects for the bias.

```
improved=holt(tsdata, h=1, level=c(95), initial="simple", alpha=discount, beta=discount)
summary(improved)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = tsdata, h = 1, level = c(95), initial = "simple", alpha = discount,
##
##  Call:
##      beta = discount)
##
##   Smoothing parameters:
##     alpha = 0.1
##     beta  = 0.1
##
##   Initial states:
##     l = 875
##     b = 171.54
##
##   sigma:  2454.618
## Error measures:
##                    ME     RMSE      MAE      MPE    MAPE     MASE      ACF1
## Training set 1307.375 2454.618 1675.615 10.53202 17.8645 1.493404 0.6051192
##
## Forecasts:
##    Point Forecast    Lo 95    Hi 95
## 38       18369.85 13558.89 23180.81
```

We see that $\hat{y}_{38}(37) \approx 18370$ with a 95% prediction interval $[13558.89, 23180.81]$.

Part (j)

The true index for 2017 is 24719.22. What is your forecast errors in part (d) and (i). Is the 2nd order approach an apparent improvement over the use of simple exponential smoothing?

The forecast error for the simple exponential smoother $\hat{y}_38^{(1)}(37)$ is

$$e_37(1) = 24719.22 - 13733.58 \approx 10986$$

and the forecast error for the double exponential smoother $\hat{y}_38^{(2)}(37)$ is

$$e_37(1) = 24719.22 - 8311 \approx 16408.$$

The second-order smoother has significantly more forecast error.

If we use the results from the *holt* procedure, which corrects for the bias, we get a much better result,

$$e_37(1) = 24719.22 - 18370 \approx 6349.$$

Appendix

```r
# moving average
MA <- function(tsdata, span)
{
   ma=filter(tsdata, rep(1/span,span),side=1)
   return(ma)
}

# exponential moving average
EMA <- function(tsdata, start, discount)
{
   N=length(tsdata)
   ema=vector(length=N)
   ema[1]=start
   for (i in 2:N)
   {
       ema[i]=ema[i-1]*(1-discount)+tsdata[i]*discount
   }
   return(ema)
}

# 2nd-order exponential smoothing
EMA2 <- function(tsdata, start1, start2, discount1,discount2)
{
   ema1=EMA(tsdata, start1, discount1)
   ema2=EMA(ema1, start2, discount2)
   return(list(ema1=ema1,ema2=ema2,yhat=2*ema1-ema2))
}
```

Figure 1: Library of functions