Naïve methods, Exponential Smoothing models and the Theta method

Week 2: Statistical Forecasting Methods

UNIVERSITY *of* NICOSIA

Naïve methods

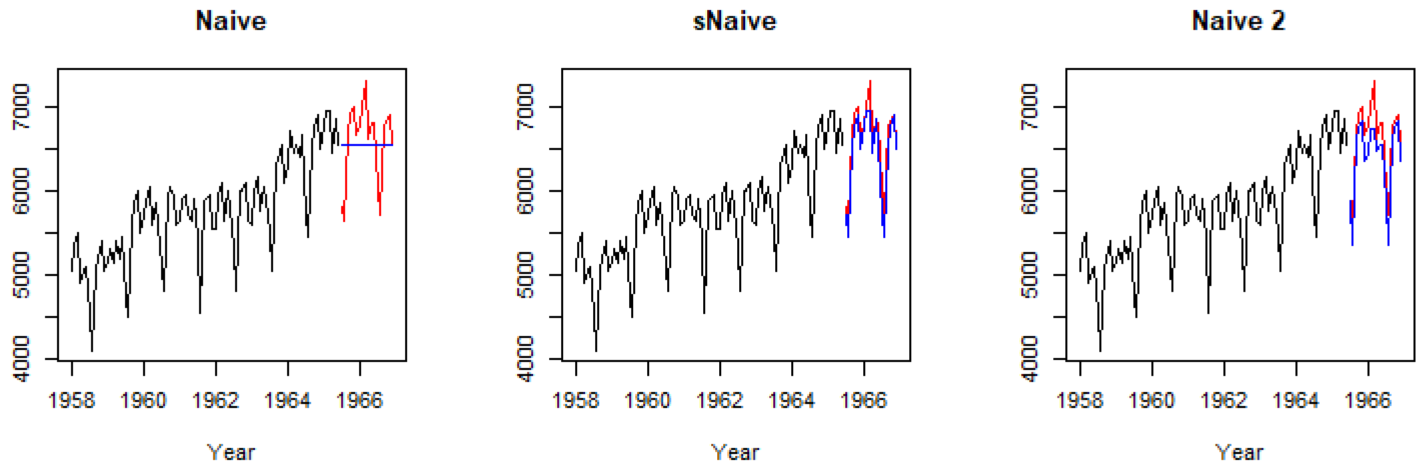
- **Naïve**: The last known observation is used to produce forecasts
- **sNaive**: The last known observation of the same period is used to produce forecasts
- **Naïve 2**: The last known observation is used to produce forecasts, adjusted for seasonality
- **Naïve with drift**: The last known observation and running trend is used to produce forecasts
- **Naïve 2 with drift**: Like Naïve 2, but with drift

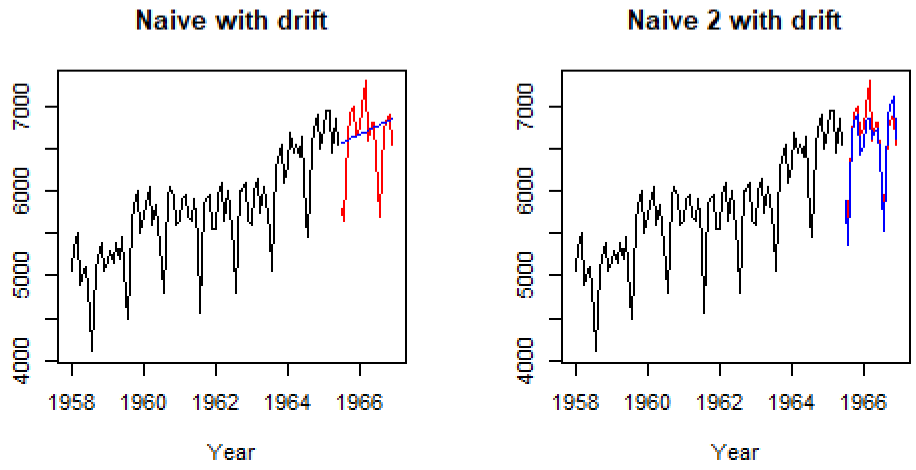
UNIC

Naïve methods

```
library(forecast)
library(Mcomp)

time_series <- subset(M3, 12)[[1110]]
insample <- subset(M3, 12)[[1110]]$x
outsample <- subset(M3, 12)[[1110]]$xx

par(mfrow=c(2,3))

Naive <- naive(insample, h=18)$mean
plot(time_series, main="Naive")
lines(Naive, col="blue")
accuracy(outsample,Naive)[5]

sNaive <- snaive(insample, h=18)$mean
plot(time_series, main="sNaive")
lines(sNaive, col="blue")
accuracy(outsample,sNaive)[5]

SI <- decompose(insample, type="multiplicative")$seasonal
Naive2 <- naive(insample/SI, h=18)$mean*as.numeric(head(tail(SI,12*2),18))
plot(time_series, main="Naive 2")
lines(Naive2, col="blue")
accuracy(outsample,Naive2)[5]

NaiveD <- rwf(insample, h=18, drift=T)$mean
plot(time_series, main="Naive with drift")
lines(NaiveD, col="blue")
accuracy(outsample,NaiveD)[5]

NaiveD2 <- rwf(insample/SI, h=18,
drift=T)$mean*as.numeric(head(tail(SI,12*2),18))
plot(time_series, main="Naive 2 with drift")
lines(NaiveD2, col="blue")
accuracy(outsample,NaiveD2)[5]
```

Naïve methods





- The MAPEs of the Naïve models examined are 5.73, 2.36, 3.41, 4.88 and **2.16**, respectively
- So, adequately capturing seasonality and trend gives us a forecasting boost of 59% and 15%, respectively.
- By considering both trend and seasonality, accuracy is improved by 62%

Moving Averages

- The last *k* known observations are used to produce forecasts
- *k* is the order of the MA model:
  - ✓ When *k* is large, forecasts become smooth and account for long term patterns
  - ✓ When *k* is small, forecasts become more volatile and account for recent level changes

✓ All *k* observations are equally weighted and important for computing the forecasts

$$F_{t+1} = \frac{1}{k} \sum_{i=t-k+1}^{t} Y_i$$

UNIC

MOFC

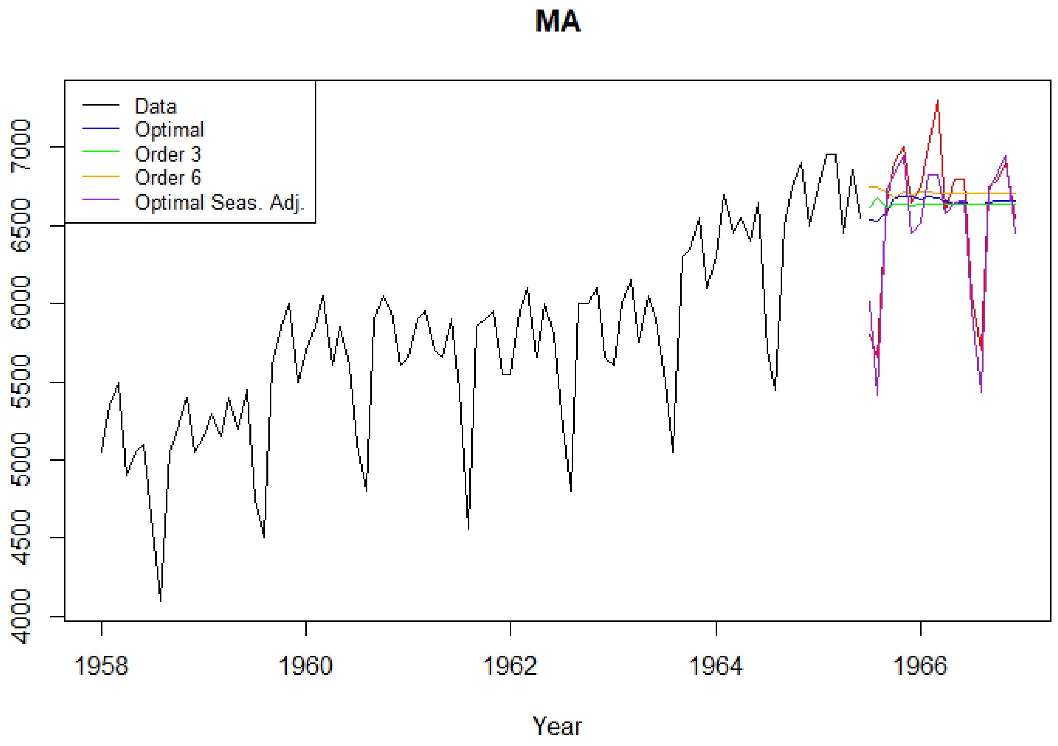
Moving Averages

Optimal estimation of order

```
library(smooth)
MA <- forecast(sma(insample), h=18)$mean
plot(time_series, main="MA")
lines(MA, col="blue")
accuracy(outsample,MA)[5]
```

Arbitrary selections

```
MA3 <- forecast(sma(insample, order=3), h=18)$mean
MA6 <- forecast(sma(insample, order=6), h=18)$mean
lines(MA3, col="green")
lines(MA6, col="orange")
accuracy(outsample,MA3)[5]
accuracy(outsample,MA6)[5]

MA_sa <- forecast(sma(insample/SI),
h=18)$mean*as.numeric(head(tail(SI,12*2),18))
lines(MA_sa, col="purple")
accuracy(outsample,MA_sa)[5]

legend("topleft", legend=c("Data","Optimal", "Order 3", "Order 6", "Optimal
Seas. Adj."),
    col=c("black","blue", "green", "orange", "purple"), lty=1, cex=0.8)
```



- The MAPEs of the MA models examined are 4.83, 5.18, 4.97 and **2.31**, respectively
- MA improve Naïve by 16% when seasonality is not considered and by 2% otherwise

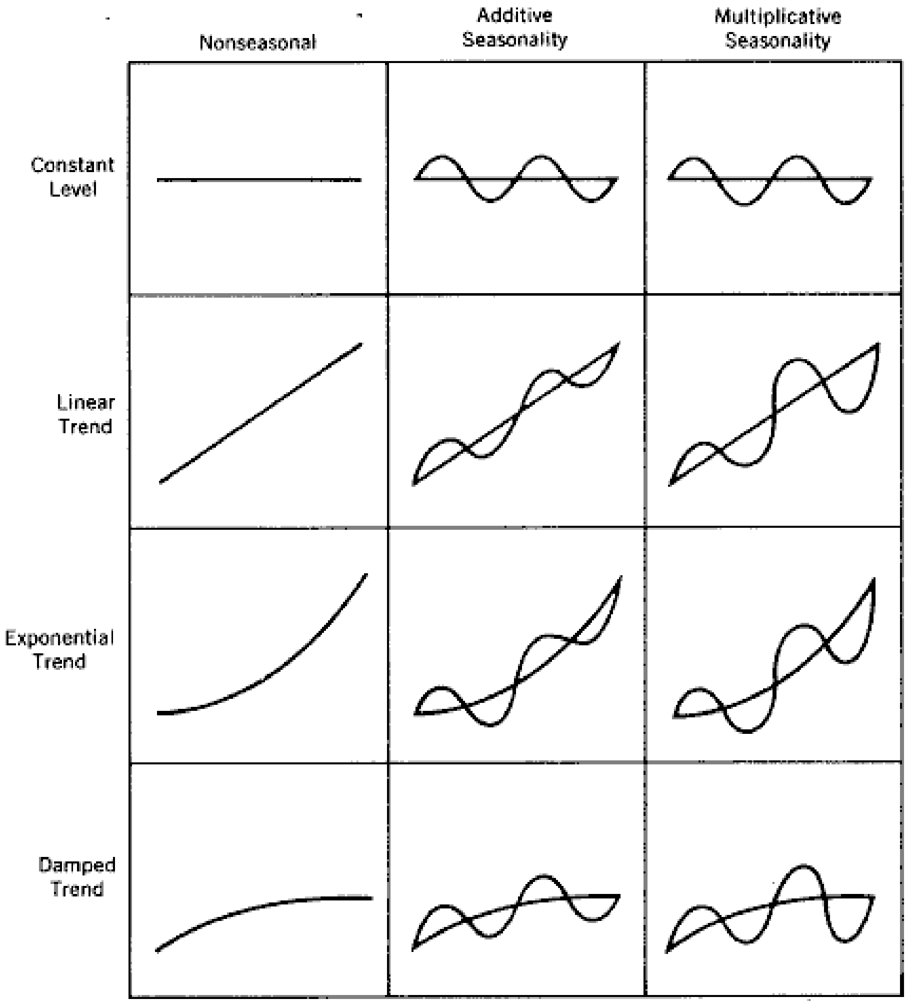
UNIC

MOFC

Exponential Smoothing

- Forecasting through data smoothing
- Past observations are **averaged** to predict the next one, using **exponential** weights
- **Recent** observations display **higher** weights than the old ones
- By smoothing the series, **randomness is removed** and the true signal becomes clear
- Typically used for **short-term forecasting**
- **Easy** and **fast** to use
- Little requirements in terms of data availability

Exponential Smoothing models



- **Simple – Constant level**
  - ✓ One-step-ahead forecasts
- **Holt – Linear trend**
  - ✓ Short-term forecasting
- **Holt – Exponential trend**
- **Damped – Damped trend**
  - ✓ Mid-term forecasting

Simple Exponential Smoothing

Definition and equations

$$e_t = Y_t - F_t$$

$$S_t = S_{t-1} + \alpha \cdot e_t$$

$$F_{t+1} = S_t$$

$$e_{t-1} = Y_{t-1} - F_{t-1}$$

$$F_t = F_{t-1} + \alpha e_{t-1}$$

$$F_t = F_{t-1} + \alpha(Y_{t-1} - F_{t-1})$$

$$F_t = \alpha Y_{t-1} + (1-\alpha)F_{t-1}$$

Simple Exponential Smoothing

$$F_t = \alpha Y_{t-1} + (1-\alpha)F_{t-1}$$

$$e_{t-1} = Y_{t-1} - F_{t-1}$$

$$F_{t+1} = \alpha Y_t + (1-\alpha)F_t$$

$$F_{t+1} = \alpha Y_t + \alpha(1-\alpha)\, Y_{t-1} + (1-\alpha)^2\, F_{t-1}$$

$$F_{t+1} = \alpha Y_t + \alpha(1-\alpha)\, Y_{t-1} + \alpha(1-\alpha)^2\, Y_{t-2} + \alpha(1-\alpha)^3\, Y_{t-3}\, \alpha(1-\alpha)^4\, Y_{t-4} + \ldots$$
$$\ldots + \alpha(1-\alpha)^{t-1}\, Y_1 + (1-\alpha)^t F_1$$

Simple Exponential Smoothing

- Weights are exponentially increased through time

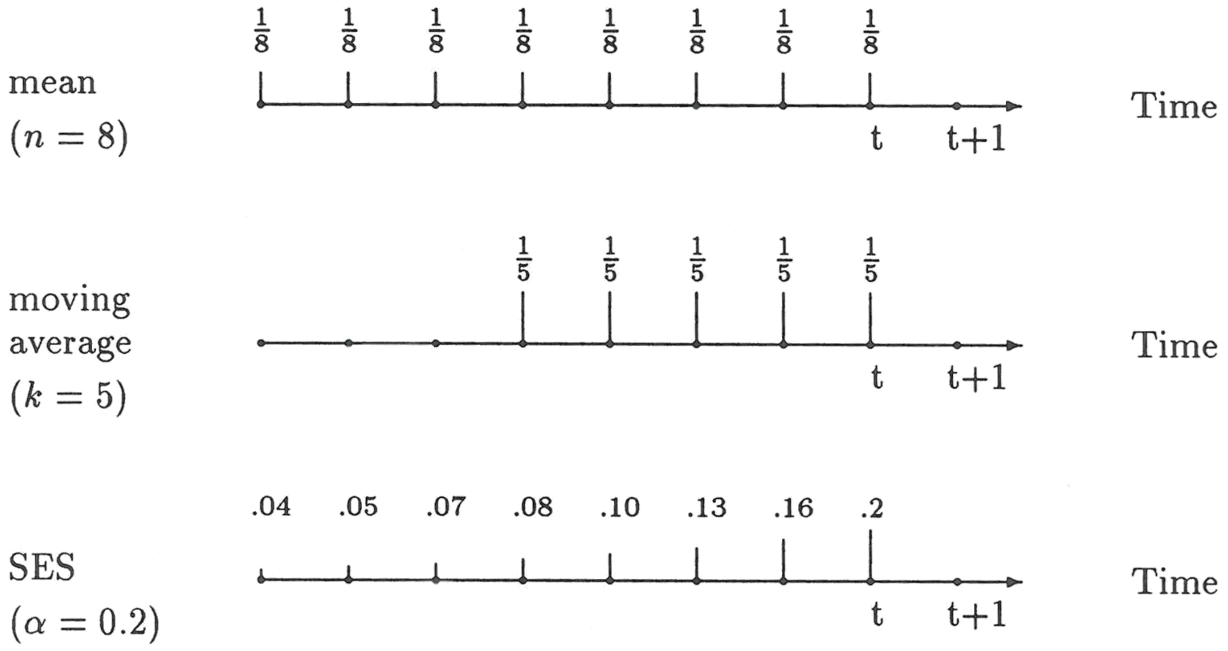
- All observations are used for producing the forecasts, but with different weights

- Small values of *a* indicate that past observations will be heavier weighted and vice-versa

- For *a*=1 SES is equivalent to the Naïve method, while for *a*=0, the forecasts are equal to S_0 (initial level).

Simple Exponential Smoothing

Weight assigned to:	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
Y_t	0.2	0.4	0.6	0.8
Y_{t-1}	0.16	0.24	0.24	0.16
Y_{t-2}	0.128	0.144	0.096	0.032
Y_{t-3}	0.1024	0.0864	0.0384	0.0064
Y_{t-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$



UNIC

MOFC

Simple Exponential Smoothing

Level initialization

- Average of series
- Average of first three observations
- First observation

Parameter selection

- *a* values range in **[0,1]**
- Minimize in-sample, **one-step-ahead MSE**, MAE or MAPE
- Different criteria – different optimal values

Simple Exponential Smoothing

(Y1+Y2)/2

t	Y	F	e	S = F + a*e	S = a*Y + (1-a)*F	S
0						167,5
1	200	167,5	32,5	167,5 + **0.2** * 32,5	0.2 * 200 + 0.8 * 167,5	174,0
2	135	174,0	-39,0	174 + 0.2 * -39	0.2 * 135 + 0.8 * 174	166,2
3	195	166,2	28,8	166,2 + 0.2 * 28,8	0.2 * 195 + 0.8 * 166,2	172,0
4	197,5	172,0	25,5	172 + 0.2 * 25,5	0.2 * 197,5 + 0.8 * 172	177,1
5	310	177,1	132,9	177,1 + 0.2 * 132,9	0.2 * 310 + 0.8 * 177,1	203,7
6	175	203,7	-28,7	203,7 + 0.2 * -28,7	0.2 * 175 + 0.8 * 203,7	197,9
7	155	197,9	-42,9	197,9 + 0.2 * -42,9	0.2 * 155 + 0.8 * 197,9	189,3
8	130	189,3	-59,3	189,3 + 0.2 * -59,3	0.2 * 130 + 0.8 * 189,3	177,5
9	220	177,5	42,5	177,5 + 0.2 * 42,5	0.2 * 220 + 0.8 * 177,5	186,0
10	277,5	186,0	91,5	186 + 0.2 * 91,5	0.2 * 277,5 + 0.8 * 186	204,3
11	235	204,3	30,7	204,3 + 0.2 * 30,7	0.2 * 235 + 0.8 * 204,3	210,4
12	???	**210,4**				

forecasts are the same for all forecasting horizons

UNIC

MOFC

Simple Exponential Smoothing

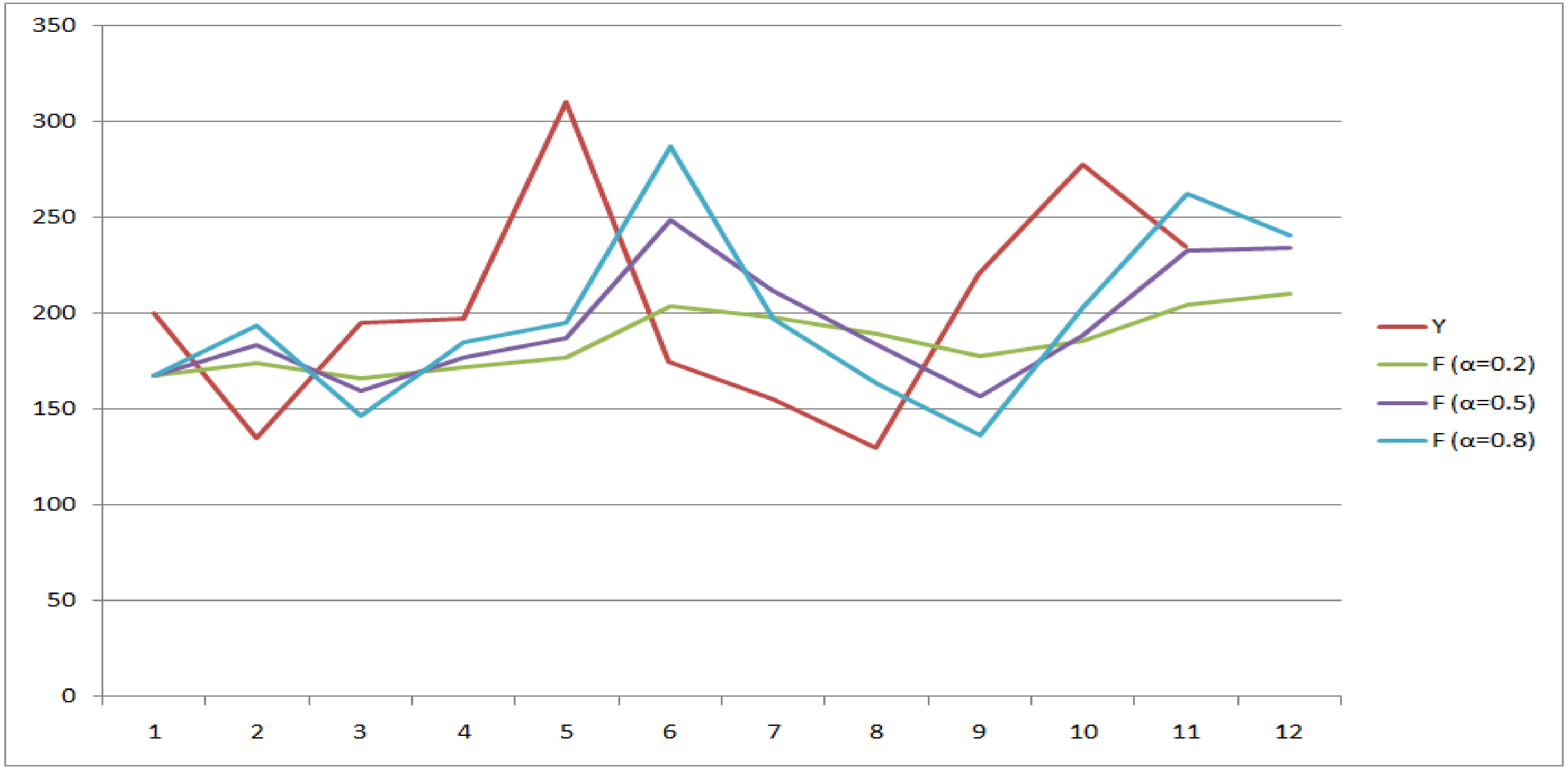
t	Y	F (α=0.2)	E	AE	APE	SAPE
1	200	167,5	32,50	32,50	0,163	0,177
2	135	174,0	-39,00	39,00	0,289	0,252
3	195	166,2	28,80	28,80	0,148	0,159
4	197,5	172,0	25,54	25,54	0,129	0,138
5	310	177,1	132,93	132,93	0,429	0,546
6	175	203,7	-28,65	28,65	0,164	0,151
7	155	197,9	-42,92	42,92	0,277	0,243
8	130	189,3	-59,34	59,34	0,456	0,372
9	220	177,5	42,53	42,53	0,193	0,214
10	277,5	186,0	91,52	91,52	0,330	0,395
11	235	204,3	30,72	30,72	0,131	0,140
12		210,4				
		α=0.2	19,51	**50,41**	**0,25**	**0,25**
		α=0.5	12,08	54,39	0,27	0,27
		α=0.8	**8,30**	58,13	0,29	0,29

Bias ← **Accuracy** →

UNIC

MOFC

Simple Exponential Smoothing



Holt Exponential Smoothing

$$e_t = Y_t - F_t$$
$$S_t = S_{t-1} + T_{t-1} + a * e_t$$
$$T_t = T_{t-1} + a * b * e_t$$
$$\mathrm{F}_{t+m} = S_t + mT_t$$

The forecast is linearly adjusted at every step

- 0<α<1 and 0<β<a

Initial level

- Average of first few observations

Initial trend

- First, first difference
- Average of first differences

Damped Exponential Smoothing

$$e_t = Y_t - F_t$$
$$S_t = S_{t-1} + \varphi T_{t-1} + a * e_t$$
$$T_t = \varphi T_{t-1} + a * b * e_t$$

$$\mathrm{F}_{t+m} = S_t + \sum_{i=1}^{m} \varphi^i \, T_t$$

The forecast is non-linearly adjusted at every step

Damped can be used to express the rest of the exponential smoothing models

- φ=0 - SES
- φ<1 – damped trend
- φ=1 – Holt
- φ>1 – exponential trend

UNIC

MOFC

Seasonal exponential smoothing models

Seasonal Models

$$e_t = Y_t - F_t$$

$$S_t = S_{t-1} + \frac{\alpha \cdot e_t}{I_{t-p}}$$

$$I_t = I_{t-p} + \frac{\gamma \cdot e_t}{S_t}$$

$$F_{t+m} = S_t \cdot I_{t-p+m}$$

Seasonal Adjustments

- Remove **additive** seasonality

  actual data **-** index = deseasonalized data

- Remove **multiplicative** seasonality

  actual data **/** index = deseasonalized data

Forecasts are **reseasonalized** using the opposite trasformation

State-space exponential smoothing models

ETS (*Error*, *Trend*, *Seasonal*)

- Error: {A,M}
- Trend:{N, A, A_d, M, M_d}
- Seasonal:{N, A, M}

(N,N)	=	simple exponential smoothing
(A,N)	=	Holt's linear method
(M,N)	=	Exponential trend method
(A_d,N)	=	additive damped trend method
(A,A)	=	additive Holt-Winters method
(A_d,M)	=	Holt-Winters damped method

A total of 30 state space models

	Seasonal Component		
Trend	*N*	*A*	*M*
Component	*(None)*	*(Additive)*	*(Multiplicative)*
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)
M (Multiplicative)	(M,N)	(M,A)	(M,M)
M_d (Multiplicative damped)	(M_d,N)	(M_d,A)	(M_d,M)

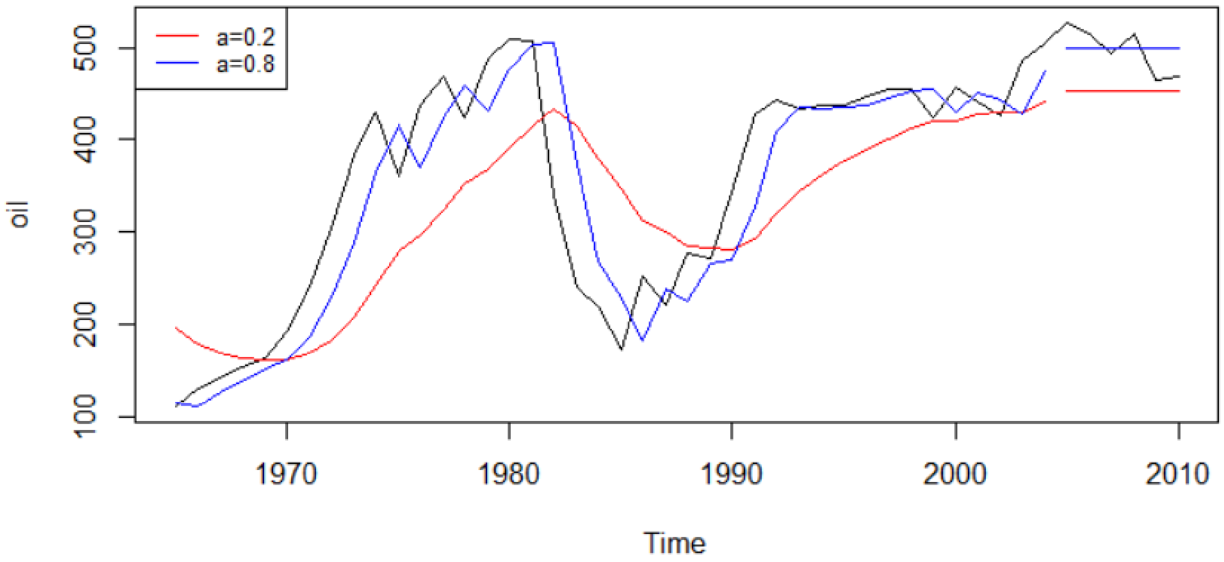
UNIC

MOFC

Exponential Smoothing in R

```
library(forecast)
library(fpp)
oildata_train <- window(oil, start = 1965, end = 2004)
oildata_test <- window(oil, start = 2005, end = 2010)

fit1 <- ses(oildata_train, alpha = 0.2, h=6)
fit2 <- ses(oildata_train, alpha = 0.8, h=6)

plot(oil, type="l")
lines(fit1$fitted, col="red")
lines(fit1$mean, col="red")
lines(fit2$fitted, col="blue")
lines(fit2$mean, col="blue")
legend("topleft", legend=c("a=0.2", "a=0.8"),
       col=c("red", "blue"), lty=1, cex=0.8)
```

Fit and evaluate SES for various smoothing parameters (a)



UNIC

MOFC

Exponential Smoothing in R

0.2 0.8

```
#Insample accuracy
mse_in1 <- mean((oildata_train - fit1$fitted)^2)
mse_in2 <- mean((oildata_train - fit2$fitted)^2)
#Outsample accuracy
mse_out1 <- mean((oildata_test - fit1$mean)^2)
mse_out2 <- mean((oildata_test - fit2$mean)^2)

c(mse_in1, mse_in2)
c(mse_out1, mse_out2)
```

```
> c(mse_in1, mse_in2)
[1] 9104.339 3006.520
> c(mse_out1, mse_out2)
[1] 2457.2304  576.7129
```

Although this is not always the case, the model that produced the most accurate forecasts when trained, is also the most accurate model in predicting the future

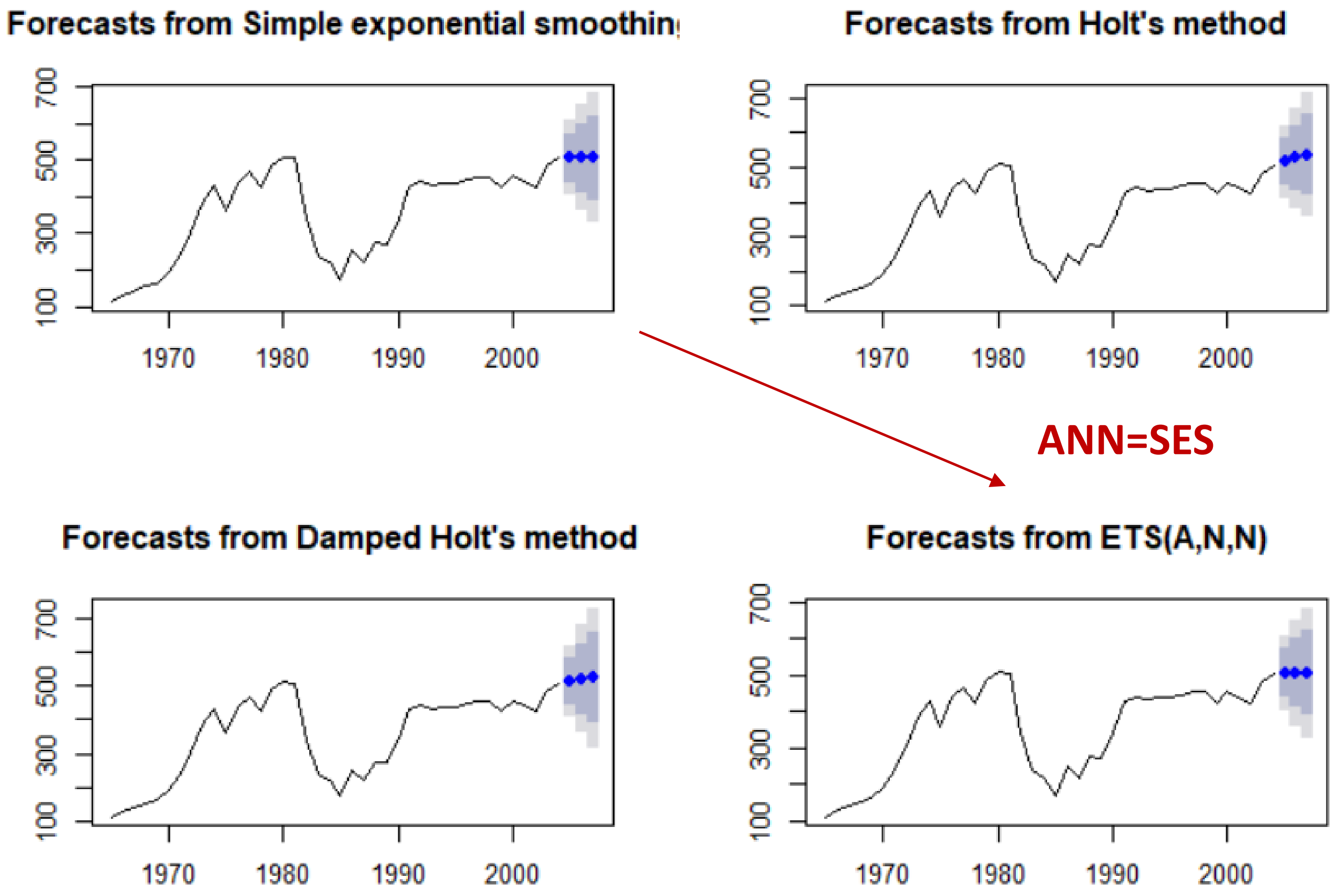
Fitting the "optimal" SES model

```
#Optimal parameters
model <- ses(oildata_train)
model$model
```

```
Simple exponential smoothing

Call:
 ses(y = oildata_train)

  Smoothing parameters:
    alpha = 0.9999

  Initial states:
    l = 110.8832

  sigma:  52.6202

     AIC     AICc      BIC
468.5515 469.2182 473.6181
```

Exponential Smoothing in R

```
#Exponential smoothing models
model1 <- ses(oildata_train, h=3)
model2 <- holt(oildata_train, h=3)
model3 <- holt(oildata_train, damped = TRUE, h=3)
model4 <- forecast(ets(oildata_train), h=3)
par(mfrow=c(2,2))
plot(model1) ; plot(model2)
plot(model3) ; plot(model4)
```



Fit and evaluate different exponential smoothing models

Exponential Smoothing in R

```
#Insample accuracy
mse_in1 <- mean((oildata_train - model1$fitted)^2)
mse_in2 <- mean((oildata_train - model2$fitted)^2)
mse_in3 <- mean((oildata_train - model3$fitted)^2)
mse_in4 <- mean((oildata_train - model4$fitted)^2)
#Outsample accuracy
mse_out1 <- mean((oildata_test - model1$mean)^2)
mse_out2 <- mean((oildata_test - model2$mean)^2)
mse_out3 <- mean((oildata_test - model3$mean)^2)
mse_out4 <- mean((oildata_test - model4$mean)^2)

c(mse_in1, mse_in2, mse_in3, mse_in4)
c(mse_out1, mse_out2, mse_out3, mse_out4)
```

SES Holt Damped ETS

```
> c(mse_in1, mse_in2, mse_in3, mse_in4)
[1] 2630.445 2557.511 2563.338 2630.445
> c(mse_out1, mse_out2, mse_out3, mse_out4)
[1] 212.6920 665.8960 348.9897 212.6920
```

- Holt is considered the most accurate model (based on in-sample data) but SES works much better for the out-of-sample data

- SES is correctly identified by ETS as the most appropriate model for automatic forecasting

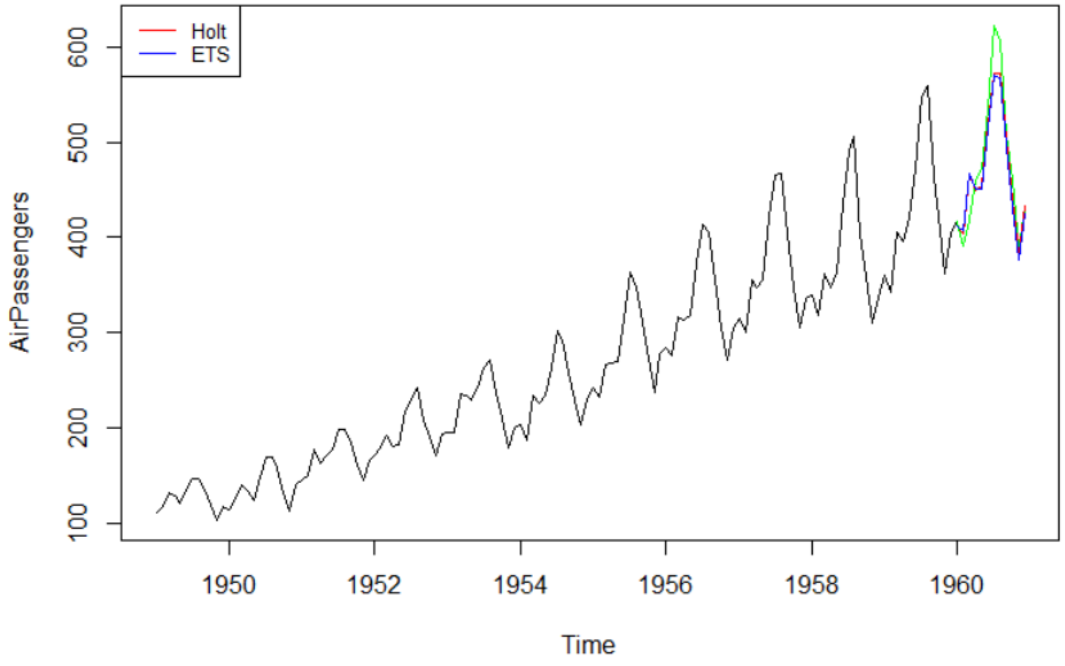
UNIC

MOFC

Exponential Smoothing in R

Exponential Smoothing for Seasonal Time Series

```
#Seasonal ts
insample <- window(AirPassengers, start=c(1949,1) , end=c(1959,12))
outsample <- window(AirPassengers, start=c(1960,1) , end=c(1960,12))

SI <- decompose(insample, type="multiplicative")$seasonal
Dy <- insample/SI
frc1 <- holt(Dy, h=12)$mean*as.numeric(tail(SI,12))

frc2 <- forecast(ets(insample), h=12)$mean

plot(AirPassengers)
lines(outsample, col="green")
lines(frc1, col="red")
lines(frc2, col="blue")
legend("topleft", legend=c("Holt", "ETS"),
       col=c("red", "blue"), lty=1, cex=0.8)

mean((outsample - frc1)^2)
mean((outsample - frc2)^2)
```




```
> mean((outsample - frc1)^2)
[1] 618.7616
> mean((outsample - frc2)^2)
[1] 750.6526
```

UNIC

MOFC

The Theta method

- Introduced by Assimakopoulos & Nikolopoulos (2000)
- The **winner of the M3 competition** and, for more than 20 years, an unbeatable benchmark
- Probably **the most accurate statistical forecasting method** till now
- A **univariate** forecasting method based on time series decomposition (*θ transformation*)
- Theta transformation can be used to create various **Theta lines** of the same trend but different curvature to that of the original data
- Different Theta lines can be used to highlight different time series characteristics, like **level** and **trend**.
- Also known as “SES with drift”*

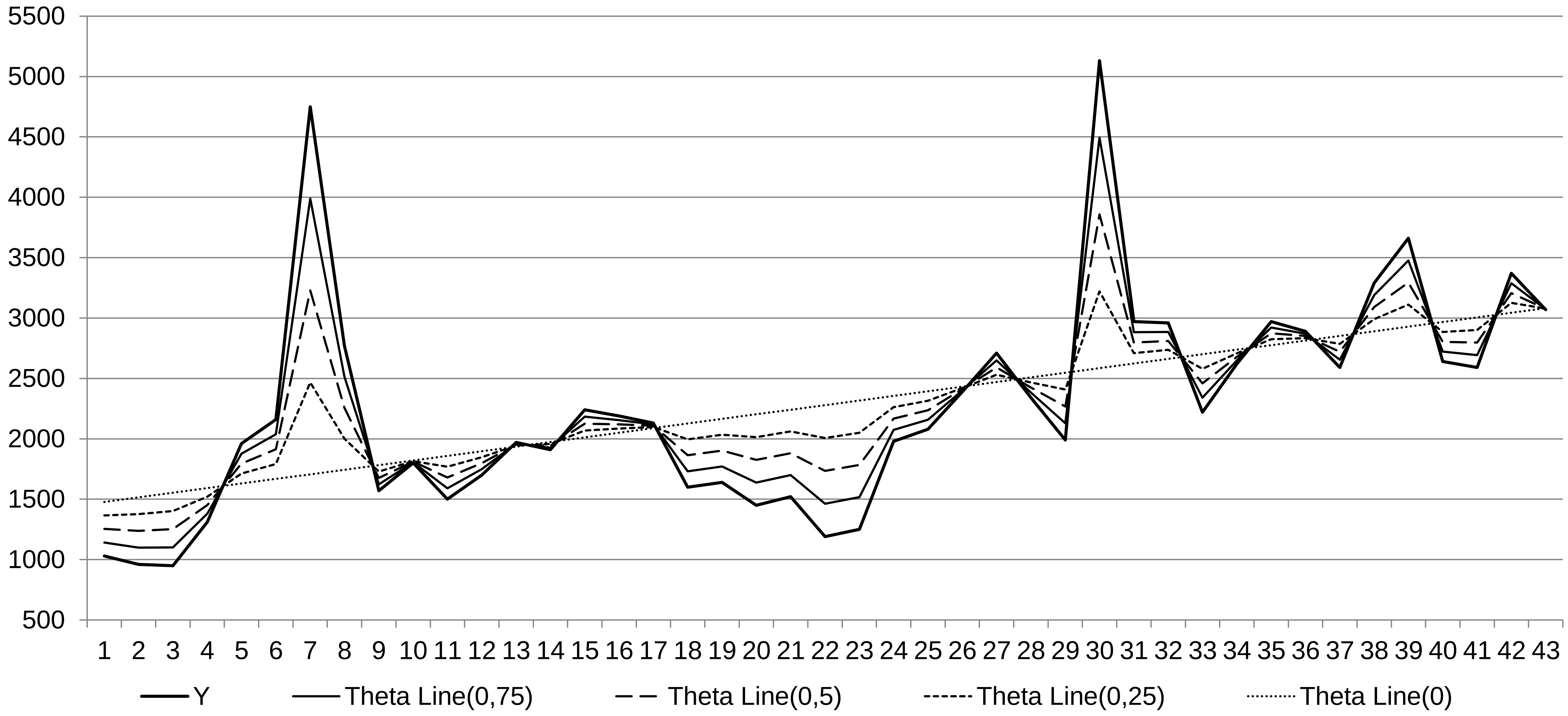
UNIC

*Hyndman & Billah (2003). Unmasking the Theta method. International Journal of Forecasting, 19, 287-290

MOFC

Theta lines (1/3)

Emphasize **long-term** characteristics for **0<θ<1**

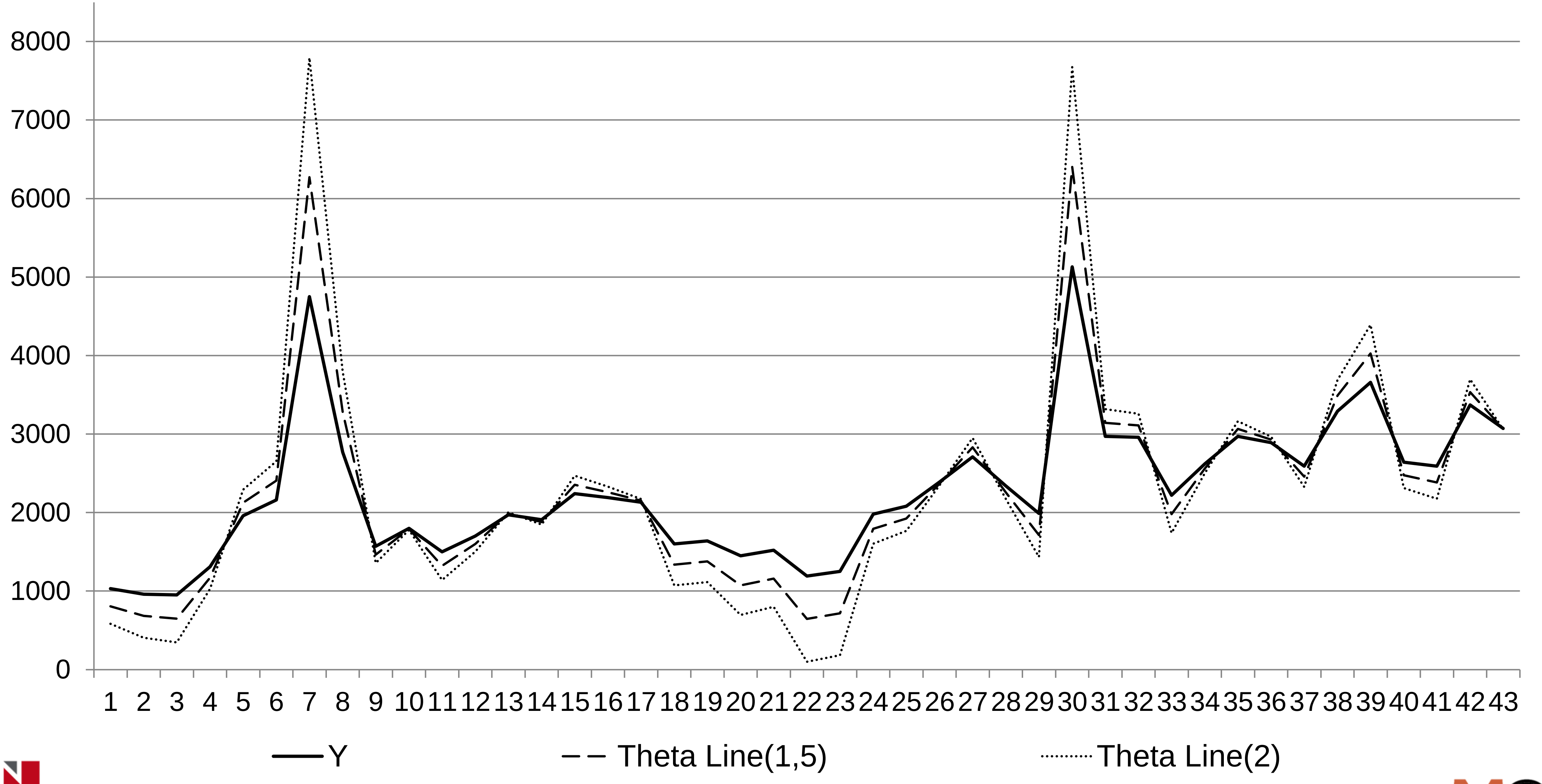


UNIC

MOFC

Theta lines (2/3)

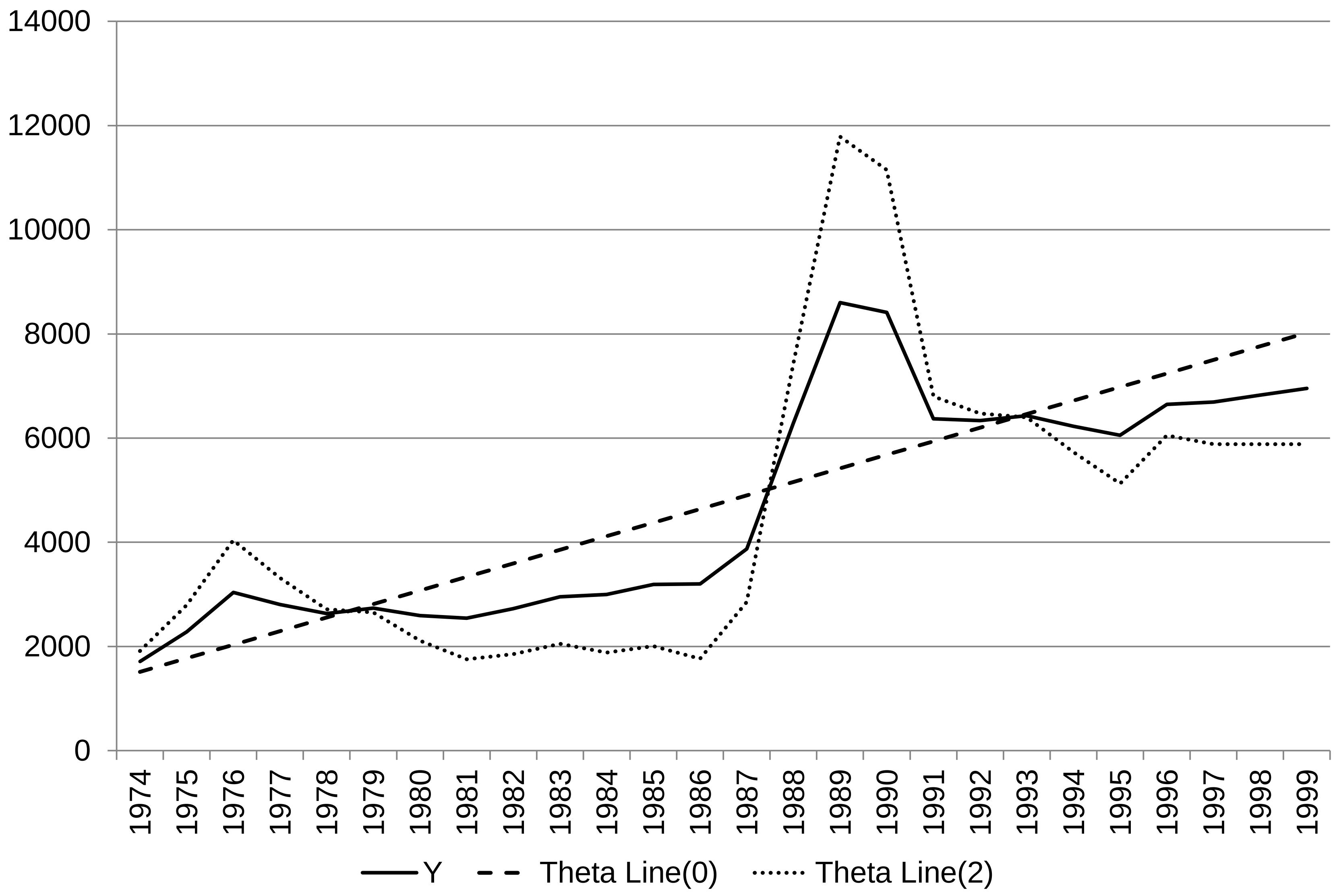
Emphasize **short-term** characteristics for **θ>1**



UNIC

MOFC

Theta lines (3/3)



$$Y_t = \frac{1}{2} \cdot \left(Y_t^{\theta=0} + Y_t^{\theta=2}\right)$$

Classic Theta (1/2)

- Step 1. **Seasonality test**

90% confidence

- Step 2. **Deseasonalization** (if needed)
- Step 3. **Theta decomposition – Create Theta lines**

Two Theta lines of θ=0 and θ=2

- Step 4. **Forecasting**

Theta line (0) using linear regression in time and Theta line (2) using SES

- Step 5. **Combine**

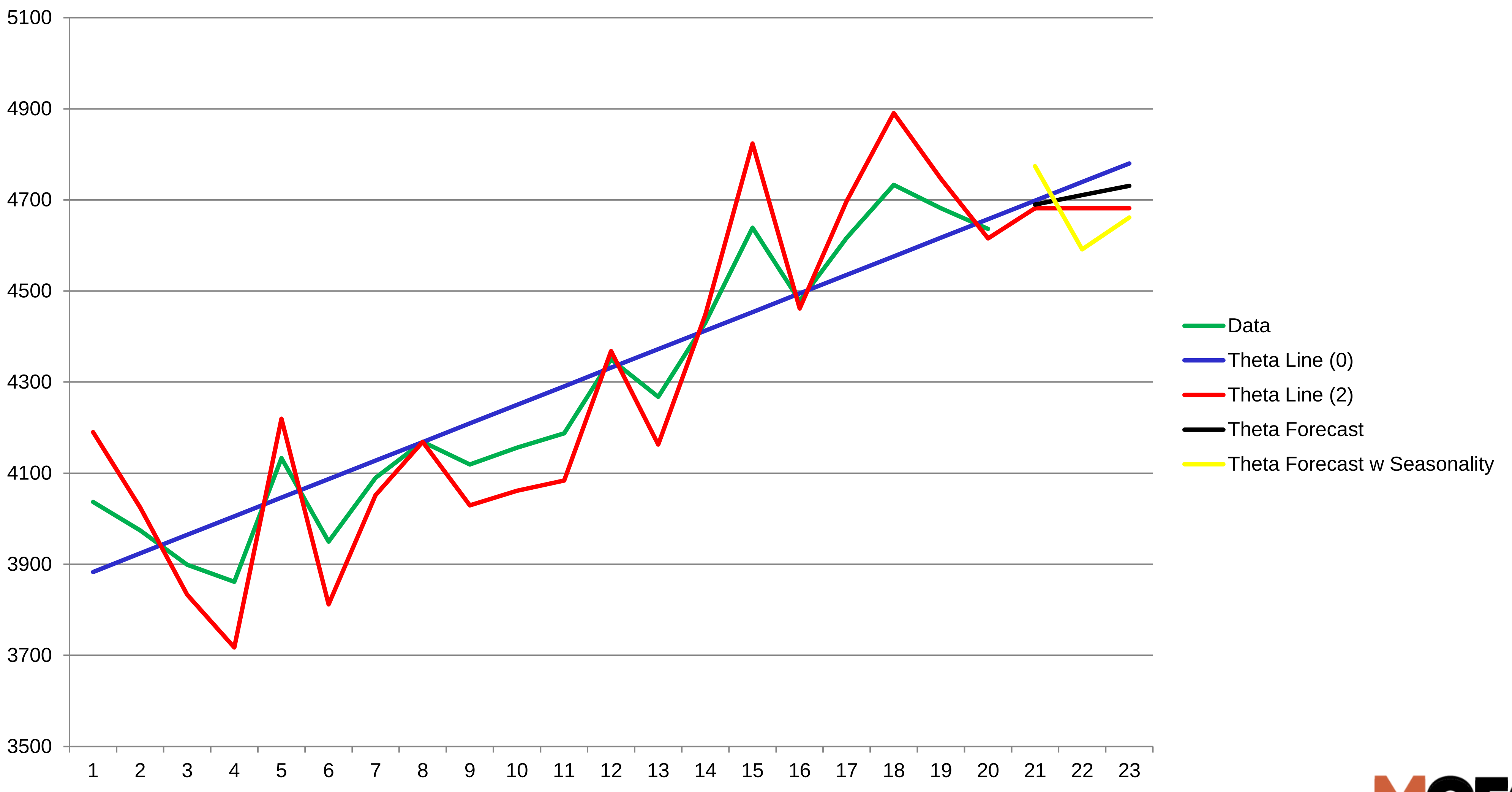
Equal weights

- Step 6. **Reseasonalization** (if needed)

Theta Line(θ) = θ x Data + (1-θ) x LRL

- Theta Line(0) = LRL
- Theta Line(2) = 2 x Data – LRL

Classic Theta (2/2)

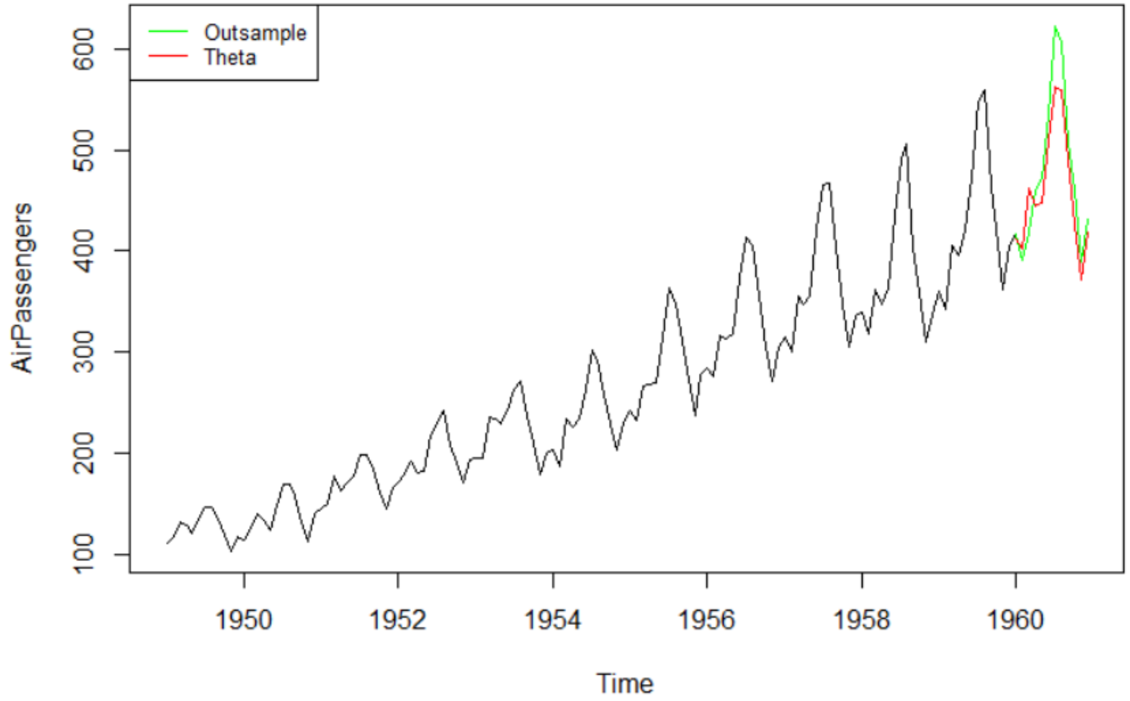


UNIC

MOFC

Theta in R

```
#Theta
insample <- window(AirPassengers, start=c(1949,1) , end=c(1959,12))
outsample <- window(AirPassengers, start=c(1960,1) , end=c(1960,12))
frc_theta <- thetaf(insample, h=12)$mean
plot(AirPassengers)
lines(outsample, col="green")
lines(frc_theta, col="red")
legend("topleft", legend=c("Outsample", "Theta"),
       col=c("green", "red"), lty=1, cex=0.8)
```



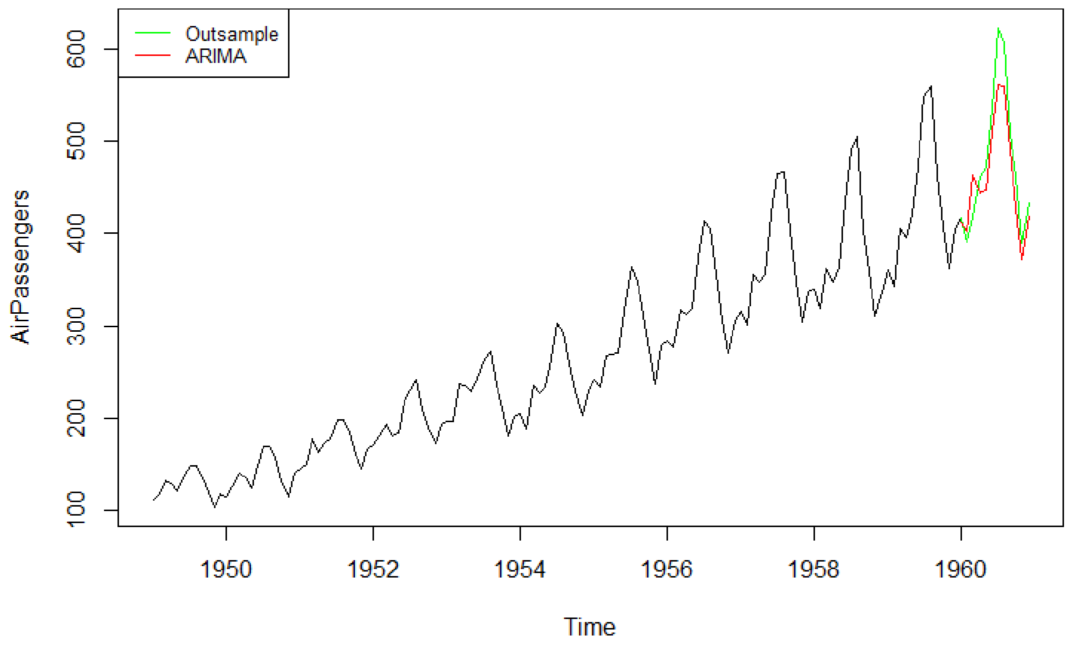
ARIMA models

- In auto-regression (**AR**) models, we forecast the variable of interest using a linear combination of past values of the variable. So, AR models perform a regression of the variable against itself.
- In moving averages (**MA**) models, forecast errors are used in a regression instead of the past values of the forecast variable
- AR and MA models can be effectively combined (ARMA models) to account both for auto-correlations and past errors
- Given that ARMA models do not capture seasonality and trend, first differences and seasonal differences can be used within ARMA models to account for such systematic variations. These are the ARIMA models

$$\underbrace{\left(1-\boldsymbol{\varphi}_1\boldsymbol{B}-\cdots-\boldsymbol{\varphi}_p\boldsymbol{B}^p\right)}_{\text{AR terms}}\underbrace{(1-\boldsymbol{B})^n}_{\text{Simple Differences}}\underbrace{(1-\boldsymbol{B}^m)^N}_{\text{Seasonal Differences}}\boldsymbol{y}_t = \boldsymbol{c}+\underbrace{\left(\boldsymbol{\theta}_1\boldsymbol{B}+\cdots+\boldsymbol{\theta}_q\boldsymbol{B}^q\right)}_{\text{MA terms}}\boldsymbol{e}_t$$

ARIMA models

```
arimaf<- forecast(auto.arima(insample), h=12)
plot(AirPassengers)
lines(outsample, col="green")
lines(frc_theta, col="red")
legend("topleft", legend=c("Outsample", "ARIMA"),
    col=c("green", "red"), lty=1, cex=0.8)

auto.arima(insample)
```



First differences

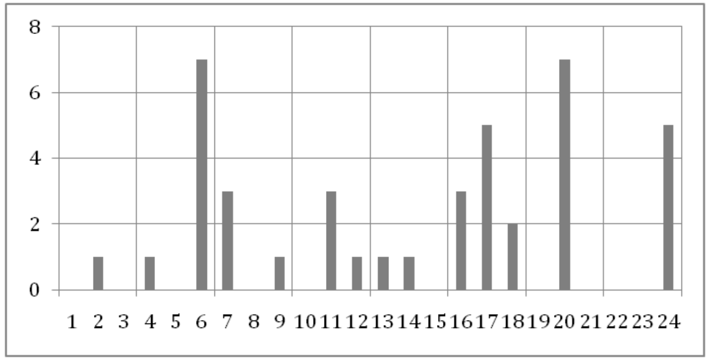
Seasonal differences

Forecasts based on AR(1)

```
Series: insample
ARIMA(1,1,0)(0,1,0)[12]

Coefficients:
          ar1
      -0.2431
s.e.   0.0894

sigma^2 estimated as 109.8:  log likelihood=-447.95
AIC=899.9   AICc=900.01   BIC=905.46
```

Croston's method

t	Y_t	t	Y_t
1	0	13	1
2	1	14	1
3	0	15	0
4	1	16	3
5	0	17	5
6	7	18	2
7	3	19	0
8	0	20	7
9	1	21	0
10	0	22	0
11	3	23	0
12	1	24	5



Each component is predicted using SES

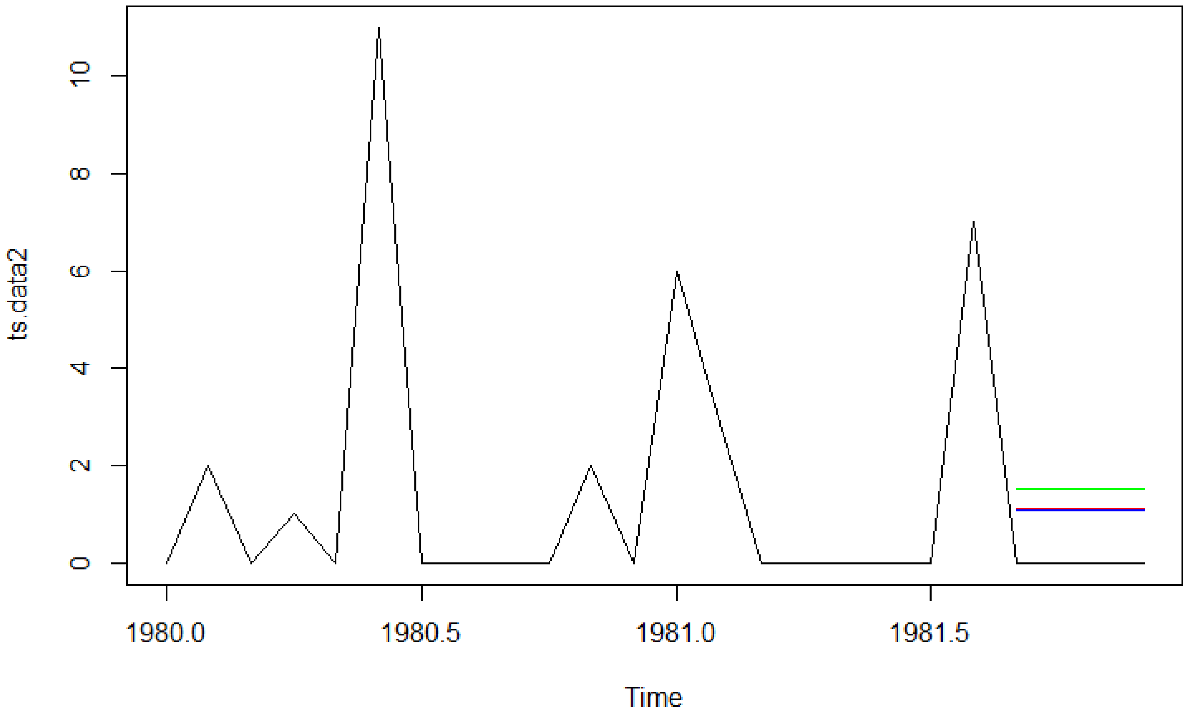
Demand	Interval
1	2
1	2
7	2
3	1
1	2
3	2
1	1
1	1
1	1
3	2
5	1
2	1
7	2
5	4

$$F = \frac{F_{Demand}}{F_{Interval}}$$

There are also many variants of the Croston's method, like SBA, SBJ and TSB, that mitigate the biases of the originally proposed method

Croston's method

```
library(tsintermittent)
insample <- head(ts.data2, 20)
outsample <- tail(ts.data2, 4)
Croston <- ts(crost(insample, type="croston",h=4)$frc.out, start = start(outsample), frequency = 12)
SBA <- ts(crost(insample, type="sba",h=4)$frc.out, start = start(outsample), frequency = 12)
TSB <- ts(tsb(insample,h=4)$frc.out, start = start(outsample), frequency = 12)

plot(ts.data2)
lines(Croston, col="blue")
lines(SBA, col="red")
lines(TSB, col="green")
```



UNIC

MOFC

UNIVERSITY *of* NICOSIA