Time Series (TS) Forecasting
Differencing & Smoothing

Nagiza F. Samatova, samatova@csc.ncsu.edu
Professor, Department of Computer Science
North Carolina State University

Senior Scientist, Computer Science & Mathematics Division
Oak Ridge National Laboratory

Outline

- Data-driven vs. Model-based Methods
  - Local vs. Global Patterns
- Differencing Methods
  - De-trending
  - De-seasonalizing
- Smoothing Methods
  - Moving Averages
  - Exponential Smoothing
  - ets() function for TS forecasting

Basic Notation

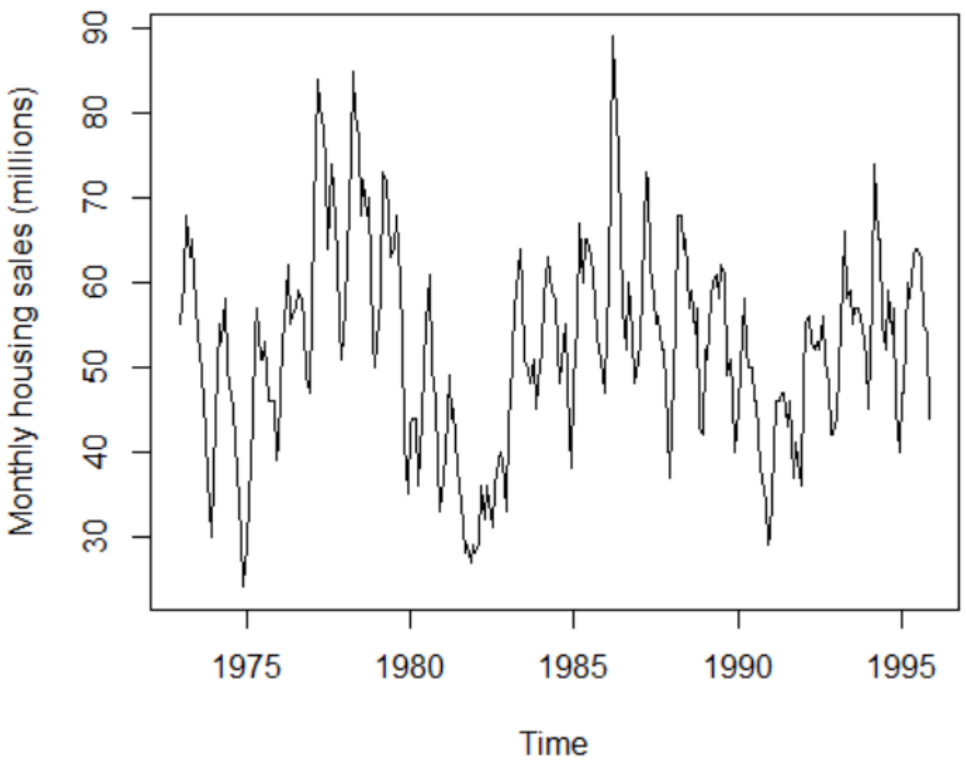
Symbol	Definition
$\boldsymbol{t = 1, 2, 3, \ldots,}$	An index for the time period of interest; e.g., for a *daily* time period, $t = 1$ means day 1, $t = 2$ means day 2, etc.
$\boldsymbol{y_1, y_2, \ldots, y_T}$	A series of T values measure over T time periods; e.g., for the annual average stock price, y_1 denotes the price for year 1, y_2 denotes the price for year 2, etc.
$\boldsymbol{F_t}$ or $\boldsymbol{\hat{y}_t}$	The forecast value for time period t
$\boldsymbol{F_{t+k}}$ or $\boldsymbol{\widehat{y_{t+k}}}$	The k-step-ahead forecast when forecasting time is t; e.g., F_{t+1} is the forecast for time period $(t + 1)$ made during the time period t
$\boldsymbol{e_t = y_t - F_t}$	The forecast error for time period t

TS Parts: Systematic vs Non-systematic

TS Part	Definition	Detection	How to deal w/
Level	Average value of ts		
Trend	Long-term increase decrease in the data	lag.plot	De-trend via lag-1 differencing
Seasonality	Variations occurring during known periods of the year (monthly, quarterly, holidays)	lag.plot, Acf plots	De-seasonalize via lag-k differencing
Cycles	Other oscillating patterns about the trend (e.g., business or economic conditions)		
Auto-correlation	Correlation between neighboring points in ts	Acf, lag.plot	
Noise	Residuals after level, trend, seasonality, and cycles are removed	Normality tests	

Monthly Housing Sales

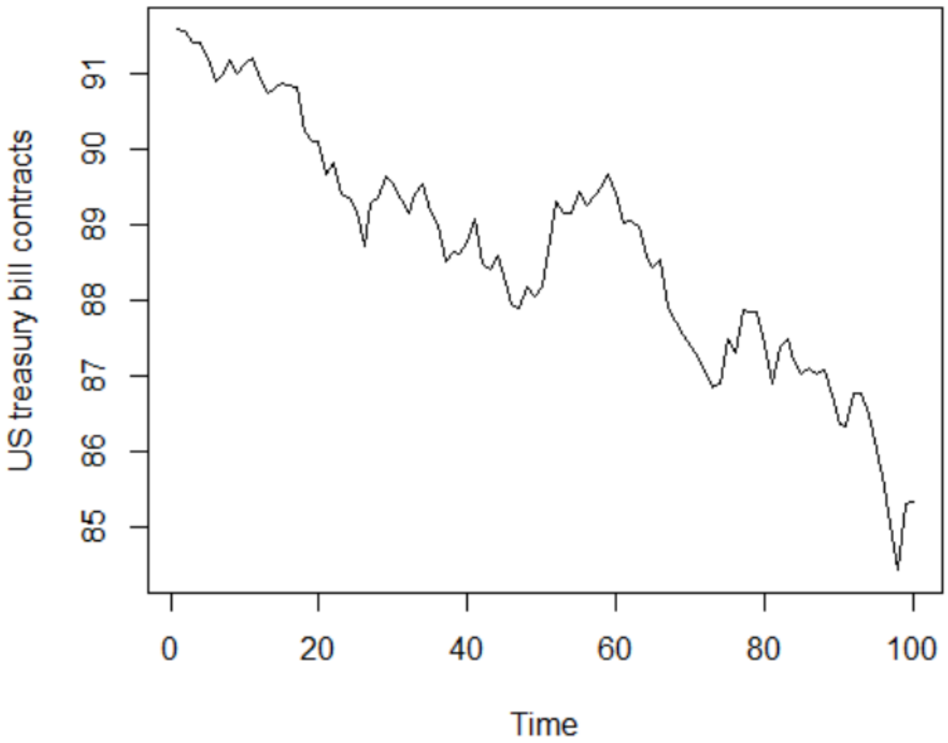
```
plot(hsales, ylab="Monthly housing sales (millions)")
```



- Strong seasonality within each year
- Cyclic behavior 6-10 years
- No trend over this period

US Treasury Bill Contracts

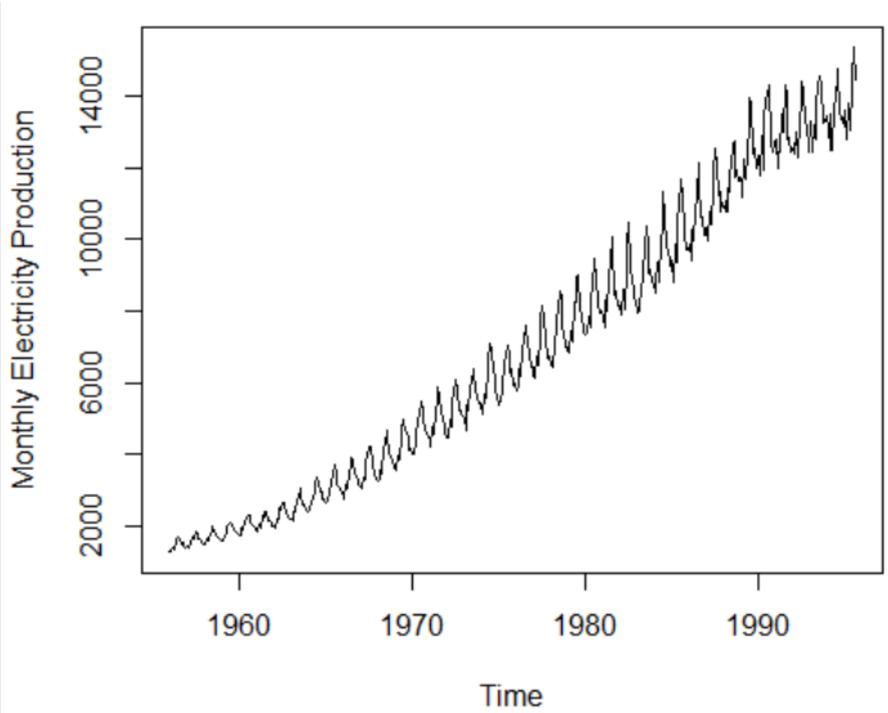
```
plot(ustreas, ylab="US treasury bill contracts")
```



- No seasonality
- Downward trend
- No cyclic behavior over this period

Monthly Electricity Production

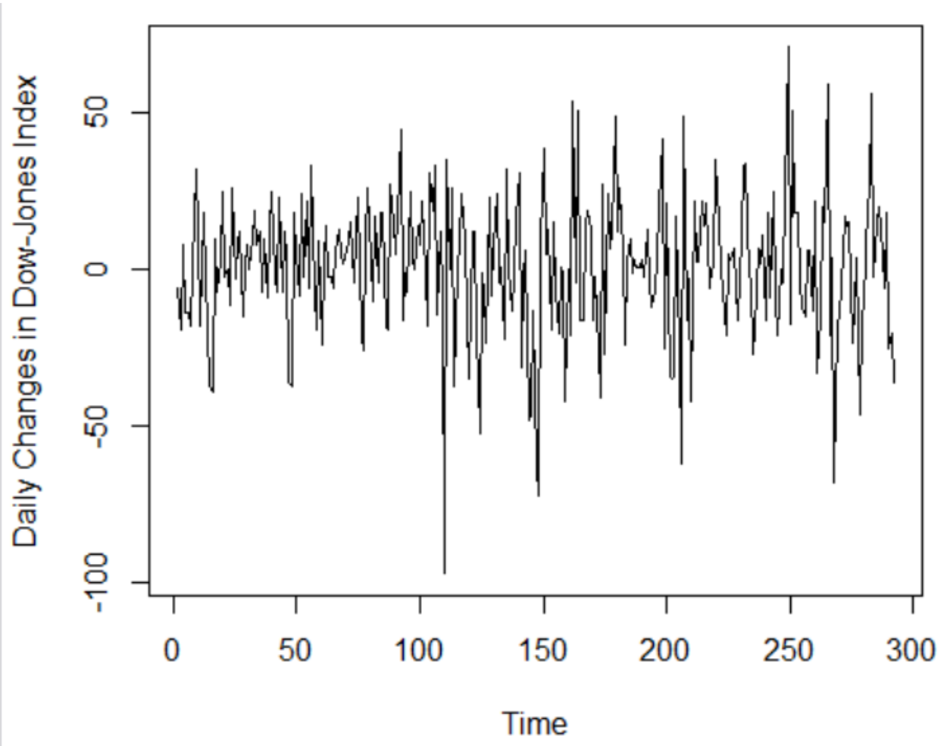
```
plot(elec, ylab="Monthly Electricity Production")
```



- Strong seasonality
- Increasing trend
- No cyclic behavior over this period

Daily Changes in Dow-Jones Index

```
plot(diff(dj,1), ylab="Daily Changes in Dow-Jones Index")
```



- No seasonality
- No trend
- No cyclic behavior over this period
- Random, unpredictable fluctuations, like white noise

Additive and Multiplicative TS Components

- A time series with additive components can be modeled as:

$$y_t = Level + Trend/Cycles + Seasonality + Noise$$

- A time series with multiplicative components is modeled as:

$$y_t = Level \times Trend/Cycles \times Seasonality \times Noise$$

- Forecasting methods attempt to isolate the systematic part and quantify the noise level.
  - The systematic part is used for generating point forecasts
  - The level of noise helps assess the uncertainty associated with the point forecasts

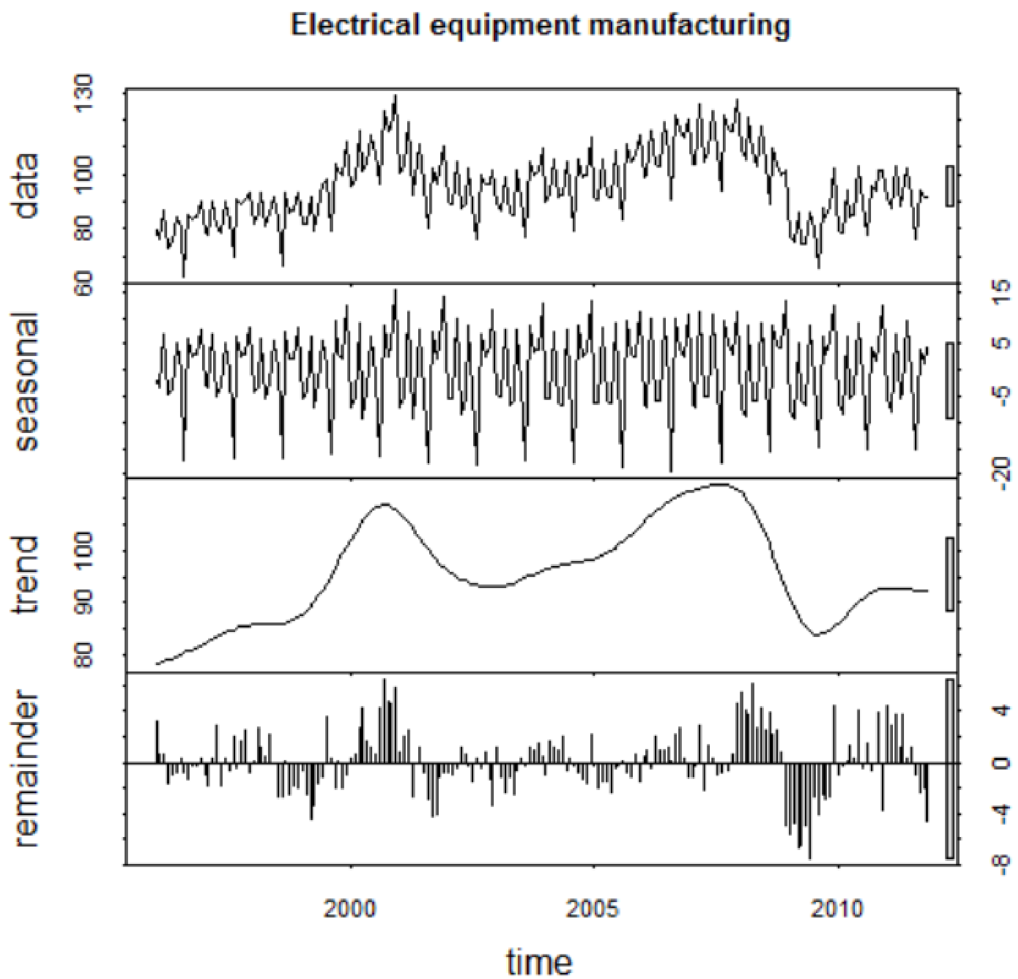
Additive vs Multiplicative Decomposition

- **Additive model** is appropriate of the magnitude of the seasonal fluctuations or **the variation around the trend-cycle does NOT vary with the level of the time series:**
  - **decompose()**
  - **stl(): for time series with additive seasonality**

- **Multiplicative model** is appropriate of the magnitude of the seasonal fluctuations or **the variation around the trend-cycle appears to be proportional to the level of the time series**
  - Rather than building a multiplicative model, transform the time series to stabilize the variance over time and then use the additive model

Example: TS Decomposition with STL

```
fit <- stl ( elecequip, s.window=5 )
plot(fit, main="Electrical equipment manufacturing")
```



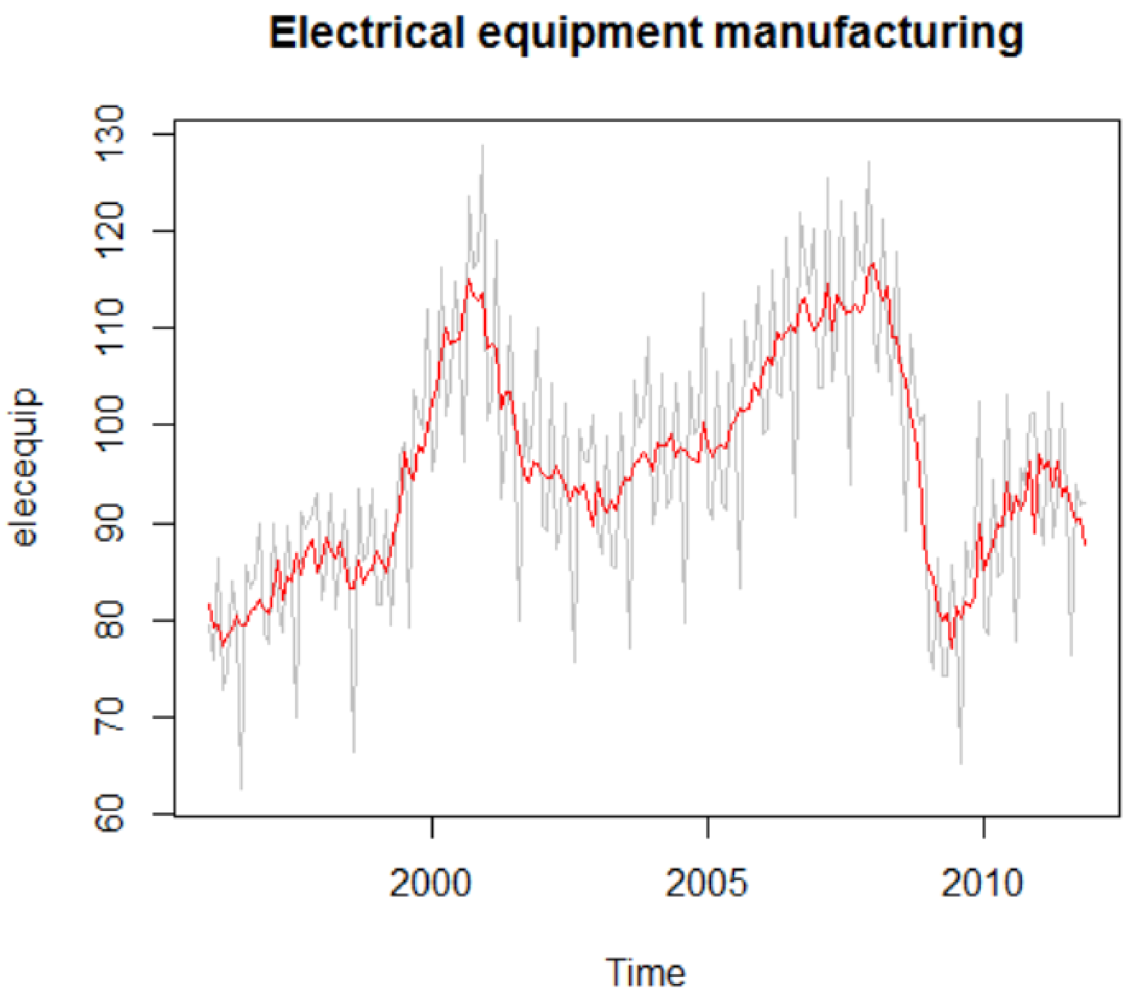
Seasonality- vs. Trend-adjusted Time Series

- If the variation due to seasonality is not of primary interest then generate seasonally adjusted time series for subsequent analysis:
  - Additive model: $y_t - S_t$, where S_t is the additive seasonal component
  - Multiplicative model: y_t/S_t, where S_t is the multiplicative seasonal component

- If the purpose is to analyze and interpret turning points, then it is better to use trend-cycle component rather than seasonally adjusted data

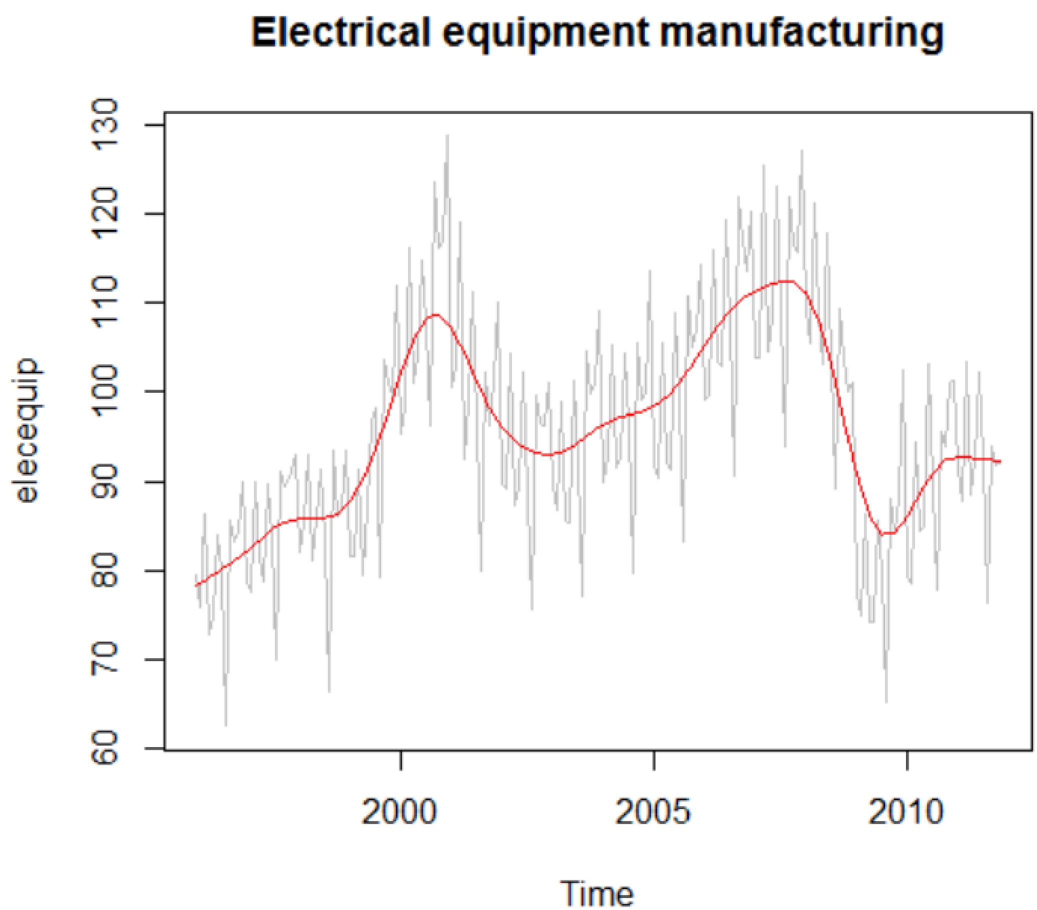
Example: Seasonally-adjusted TS

```
fit <- stl ( elecequip, s.window=5 )
plot(elecequip, col="grey", main="Electrical equipment manufacturing")
lines(seasadj (fit), col="red", ylab="Seasonally adjusted")
```



Example: Trend/Cycle Component in TS

```
fit <- stl ( elecequip, s.window=5 )
plot(elecequip, col="grey", main="Electrical equipment manufacturing")
lines(fit$time.series[ , 2], col="red", ylab="Trend")
```



Global vs Local Patterns in TS Data

- **Global patterns** are the patterns in the TS data that extend throughout the TS period used for training and forecasting:
  - Trends: persistent upward or persistent downward trend with a non-changing slope

- **Local patterns** change over-time in a hard-to-predict manner

How to Detect, Extract & Quantify TS Parts?

- The answer to this question depends a number of assumptions:
  - Whether the forecasting model to learn is persistent, i.e., does not / will not change over time
  - Whether TS exhibits long temporal scale patterns (i.e., **global patterns**) or short temporal scale patterns (i.e., **local patterns**)
  - How frequent do the short-scale patterns change over time?

TS Data Analysis Methods

TS Data Analysis & Forecasting

Data-Driven

Data-driven methods are used when model assumptions are likely to be violated, or when the structure of time series changes over time.

- **Baseline**: average, naive, seasonal naive, drift
- **Differencing**
- **Smoothing**: moving average, exponential smoothing

Model-based

Training data is used to estimate model parameters, and then the model with these parameters is used to generate forecasts.

- **ARIMA**
- **Linear Regression**
- **Logistic Regression**
- **Neural Networks**

Data-driven vs. Model-Based Methods

- **Model-based methods are generally preferable for forecasting series with global patterns; they use all the data to estimate the global patterns.**
  - For a local pattern, a model would require specifying how and when the patterns change, which is usually impractical and often unknown.
  - Model-based methods such as neural networks, regression trees, etc. are also used for TS forecasting for **incorporating external information into forecasts**.

- **Data-driven methods are preferable for forecasting series with local patterns. Such methods "learn" patterns from the data, and their memory length can be set to best adapt to the rate of change in the series.**
  - Patterns that change quickly warrant a "short memory," whereas patterns that change slowly warrant a "long memory"

TS Data Analysis Methods

TS Data Analysis & Forecasting

Data-Driven

Data-driven methods are used when model assumptions are likely to be violated, or when the structure of time series changes over time.

- **Baseline**: average, naive, seasonal naive, drift
- **Differencing**
- **Smoothing**: moving average, exponential smoothing

Model-based

Training data is used to estimate model parameters, and then the model with these parameters is used to generate forecasts.

- **ARIMA**
- **Linear Regression**
- **Logistic Regression**
- **Neural Networks**

Baseline: Simple Forecasting Methods

- **Average**: meanf (ts.data, h=20)
  - Forecast of all future values is the mean of historical data $\{y_1, \dots, y_T\}$
  - $F_{T+h} = \hat{y}_{T+h} = \bar{y} = (y_1 + \cdots + y_T)/T$

- **Naive:** naive (ts.data, h=20) or rwf (ts.data, h=20)
  - Forecast is equal to the last observed value
  - $F_{T+h|T} = \hat{y}_{T+h|T} = y_T$

- **Seasonal naive:** snaive (ts.data, h=20)
  - Forecast is equal to the last value from the same season
  - $\hat{y}_{T+h|T} = y_{T+h-km}$, where m is the seasonal period and $k = round\left(\frac{h-1}{m}\right) + 1$

- **Drift:** rwf (ts.data, drift=TRUE, h=20)
  - Forecast is equal to the last value plus the average change
  - Equivalent to extrapolating a line between the first and last observation
  - $F_{T+h|T} = \hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + \frac{h}{T-1}(y_T - y_1)$

Removing Trend & Seasonality

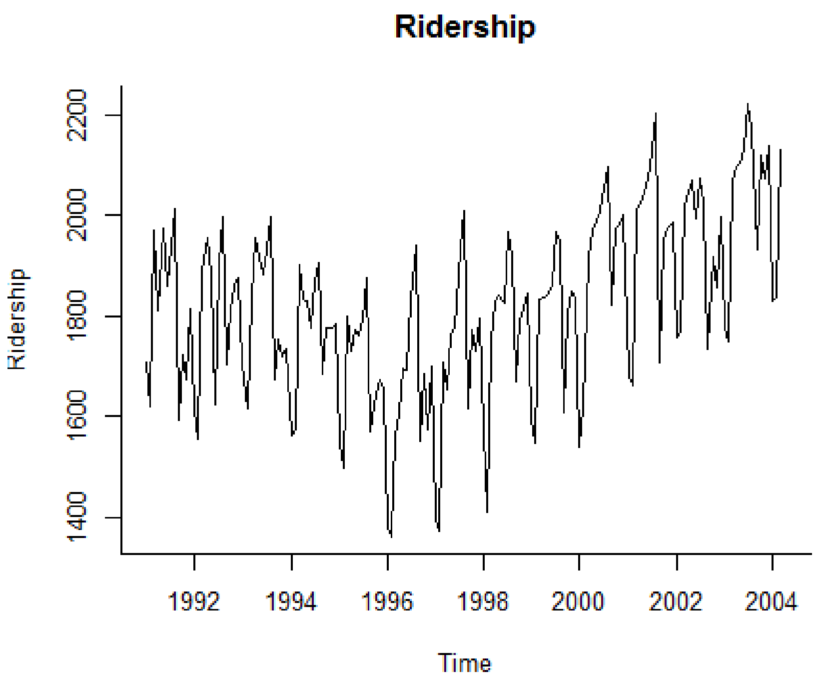
DIFFERENCING

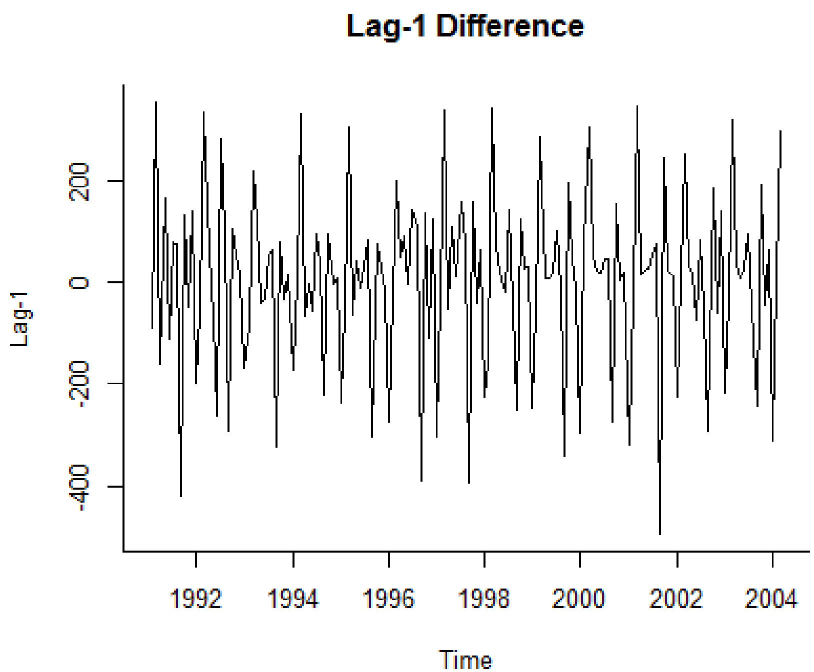
Differencing

- **Differencing: take the difference between two time series.**
  - Used for removing a trend and/or a seasonal pattern
  - **De-trend**: to remove the trend in a ts
  - **De-seasonalize**: to remove seasonality in a ts
- **De-trend:**
  - A **lag-1** difference $(y_t - y_{t-1})$ means taking the difference between every two consecutive values in a series.
  - lag-1 differencing results in series that measure the changes form one period to the next
- **De-seasonalize:**
  - Differencing at lag-k means subtracting the value from k periods back $(y_t - y_{t-k})$
  - Example: lag-12 difference of the Amtrak ridership to remove annual seasonality: `diff(ridership.ts, lag = 12)`
- **Double Differencing:**
  - To remove both trend and seasonality
  - To remove higher-order trends

Differencing for de-trending

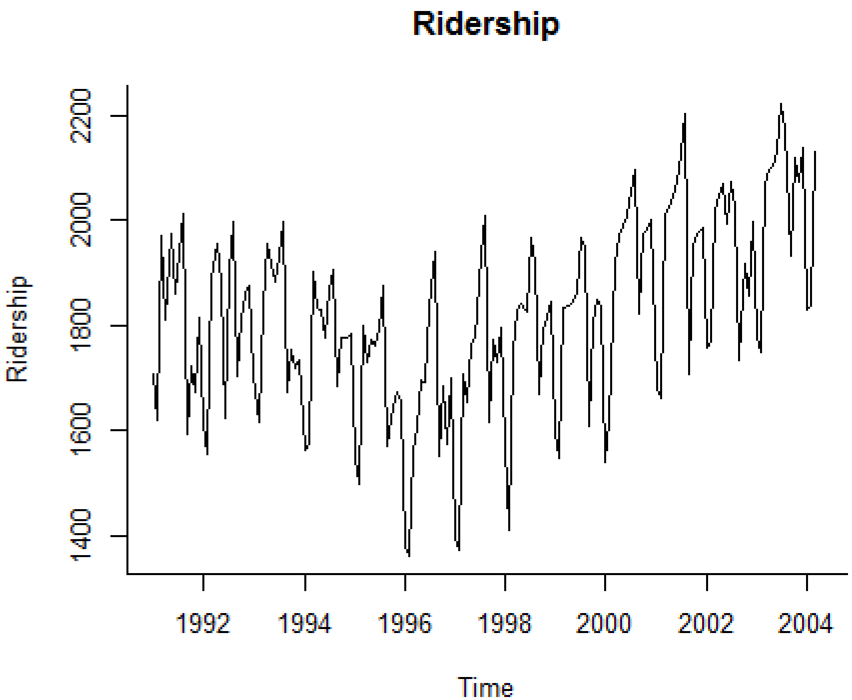
- **lag-1 differencing is useful for removing a trend**
- **De-trending with differencing vs. regression:**
  - differencing does not assume that the trend is global, i.e., that the trend is fixed throughout the entire period.
- **For *quadratic* and *exponential* trends, often another round of lag-1 differencing must be applied in order to remove the trend**

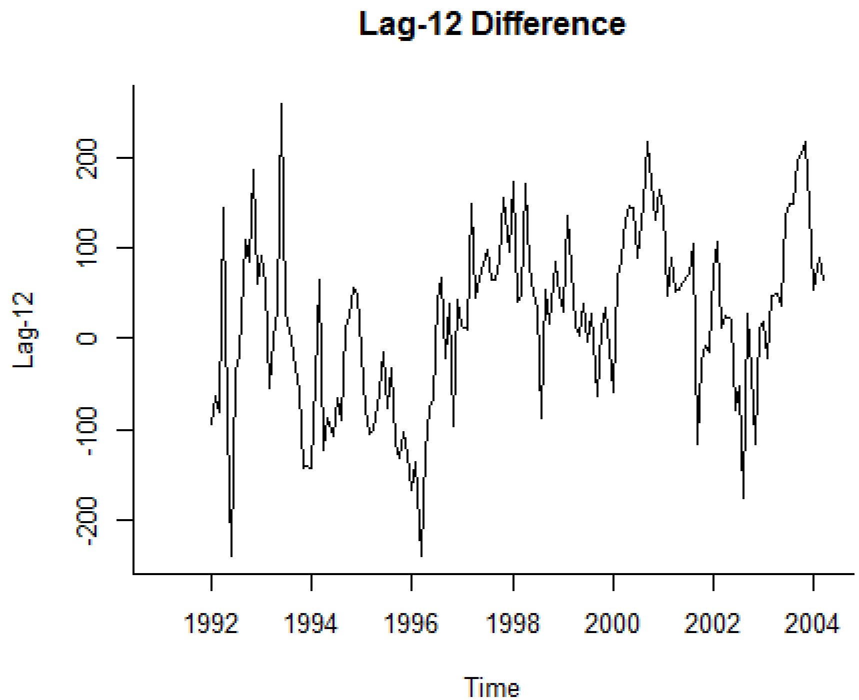




Differencing for Removing Seasonality

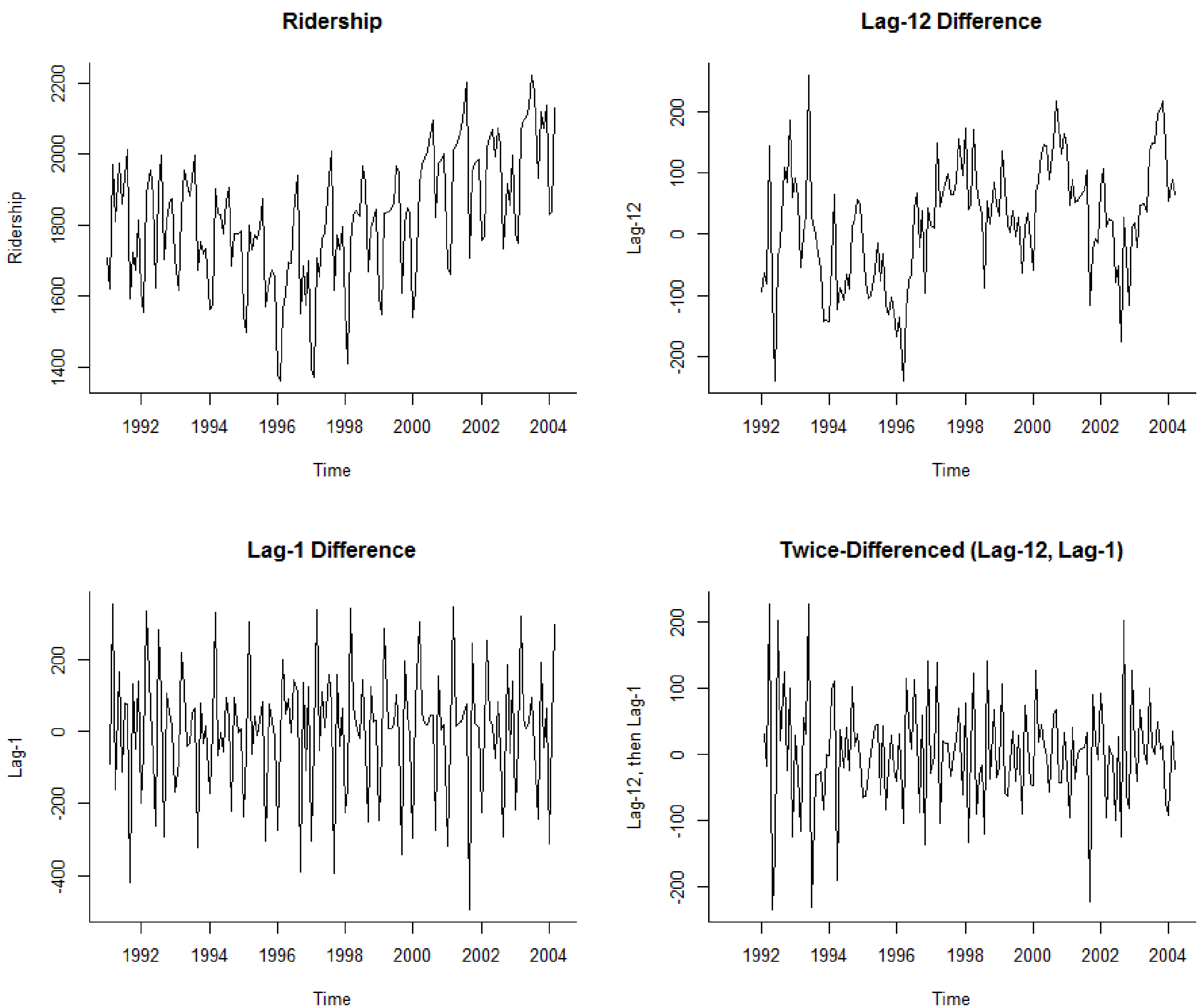
- **To remove a seasonal pattern with M seasons, difference at lag M.**
  - to remove a day-of-week pattern in daily data, take lag-7 differences
- **Example, lag-12 applied to Amtrak monthly data removes monthly pattern:** `diff(ridership.ts, lag = 12)`





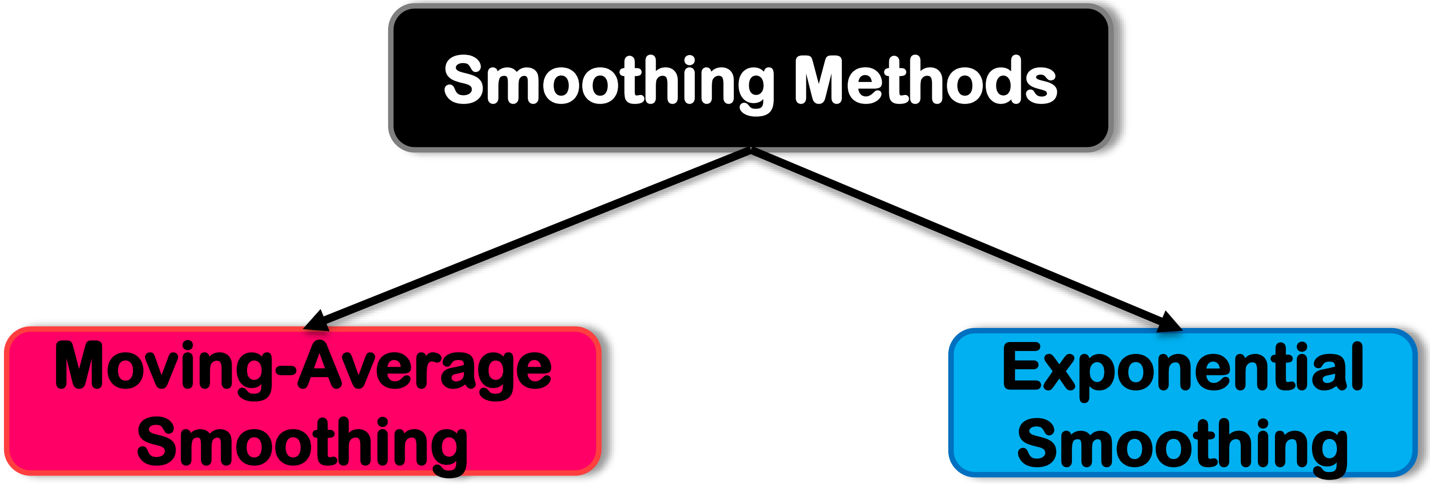
Removing Trend and Seasonality

- **When both trend and seasonality, apply differencing twice**



Smoothing Methods

MOVING AVERAGE



Smoothing Methods

- **Smoothing methods "smooth" out the noise in a series in an attempt to uncover the patterns.**
- **Smoothing is done by averaging over multiple observations**
- **Different smoothers differ by**
  - the number of observations averaged,
  - how the average is computed,
  - how many times averaging is performed

Moving-Average Smoothing

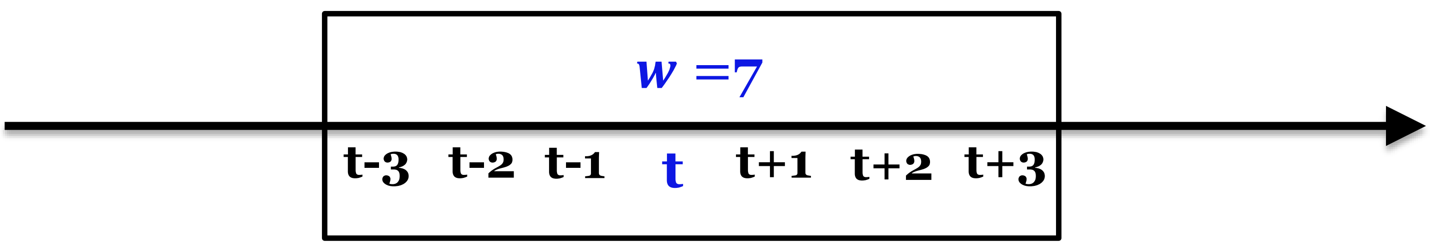
- **Moving average consists of averaging across a window of consecutive observations generating a series of averages**
- **Two types of moving averages:**
  - **a centered moving average**
  - **a trailing moving average**
- **Centered moving averages are used for visualizing trends:**
  - the averaging suppresses seasonality and noise, thus
  - making the trend more visible
- **Trailing moving averages are used for forecasting**

Use Trailing Moving-Average for forecasting ONLY after both trends and seasonality are removed.

Centered Moving-Average for Visualization

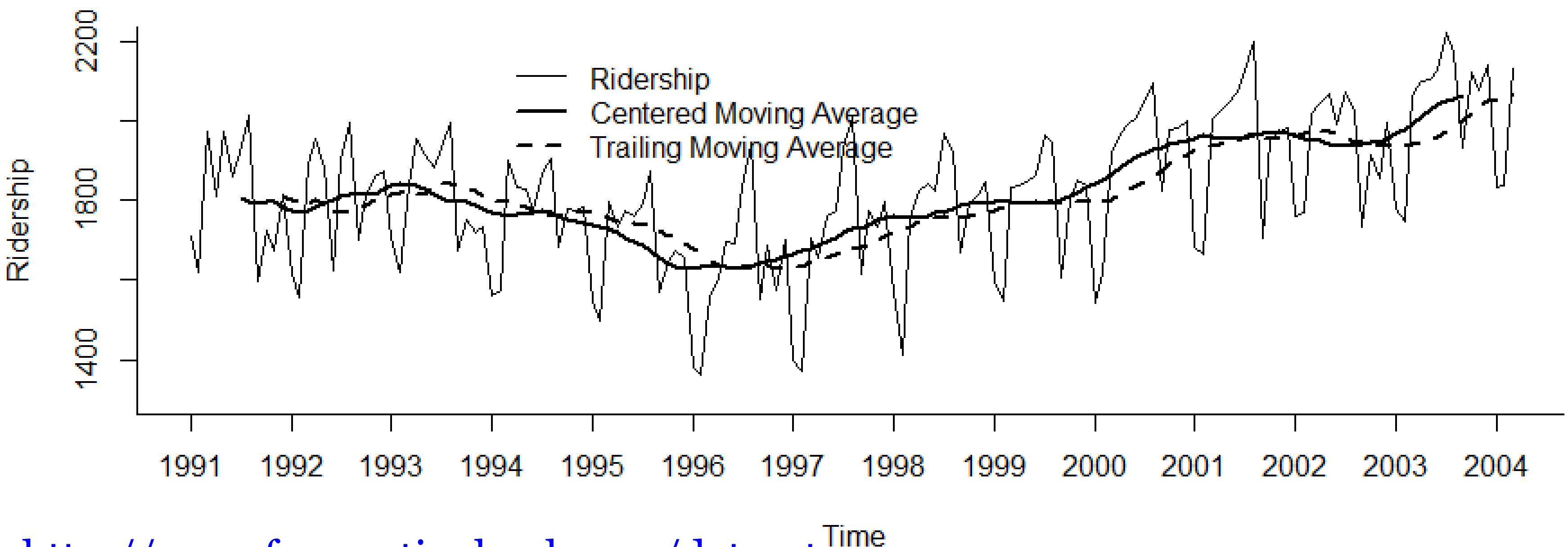
- In a **centered moving-average** the value of the **moving average at time t (MA_t)** is computed by centering the window around time t and averaging across the w values within the window

$$MA_t = \frac{y_{t-\frac{w-1}{2}} + \cdots y_{t-1} + y_t + y_{t+1} + \cdots + y_{t+\frac{w-1}{2}}}{w}$$



Window Width Selection

- **To choose the window width in a seasonal series:**
  - because the goal is to suppress seasonality for better visualizing the trend, the default choice is **the length of a seasonal cycle**
- **Example:**
  - Amtrak ridership data, the annual seasonality indicates $w = 12$

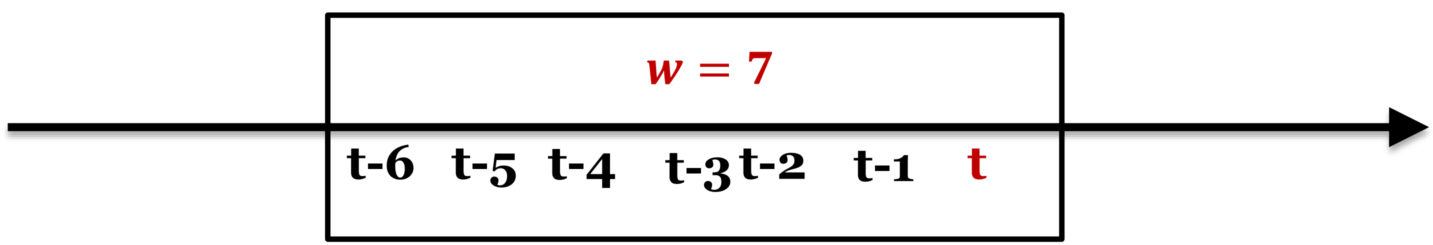


http://www.forecastingbook.com/datasets

Trailing Moving-Average for Forecasting

- **Centered moving averages are computed by averaging across data both in the past and future of a given time point.**
  - They cannot be used for forecasting because at the time of forecasting, the future is typically unknown
- **For the purposes of forecasting, use trailing moving averages:**
  - **where the window of width w is set on the most recent available w of the series.**
- **The k-step-ahead forecast F_{t+k} $(k = 1, 2, 3..)$ is then the average of the these w values**

$$\boxed{F_{t+k} = \frac{y_t + y_{t-1} + y_{t-2} + \cdots + y_{t-w+1}}{w}}$$



Example: Moving Average Smoothing

```
library("forecast")
library("zoo")
Amtrak.data <- read.csv("Amtrak data.csv")
ridership.ts <- ts(Amtrak.data$Ridership, start = c(1991, 1),
                                    end = c(2004, 3), freq = 12)

ma.trailing <- rollmean(ridership.ts, k = 12, align = "right")
ma.centered <- ma(ridership.ts, order = 12)
plot(ridership.ts, ylim = c(1300, 2200),  ylab = "Ridership",
                   xlab = "Time", bty = "l", xaxt = "n",
                   xlim = c(1991,2004.25), main = "")
axis(1, at = seq(1991, 2004.25, 1),
        labels = format(seq(1991, 2004.25, 1)))
lines(ma.centered, lwd = 2)
lines(ma.trailing, lwd = 2, lty = 2)
legend(1994,2200, c("Ridership","Centered Moving Average",
                    "Trailing Moving Average"),
                    lty=c(1,1,2), lwd=c(1,2,2), bty = "n")
```

http://www.forecastingbook.com/datasets

Notes: Lagging and De-trending

- **Moving average "lags behind," thereby it is over-forecasting in the presence of an increasing trend and under-forecasting in the presence of a decreasing trend:**
  - In general, the moving average can be used for forecasting only in a series that lacks seasonality and trend.

- **To remove trends (de-trending) and to remove seasonality (de-seasonalizing) from a time series, use the following methods:**
  - regression models
  - advanced exponential smoothing methods
  - differencing

- **Use the moving average to forecast such de-trended and de-seasonalized series, and**

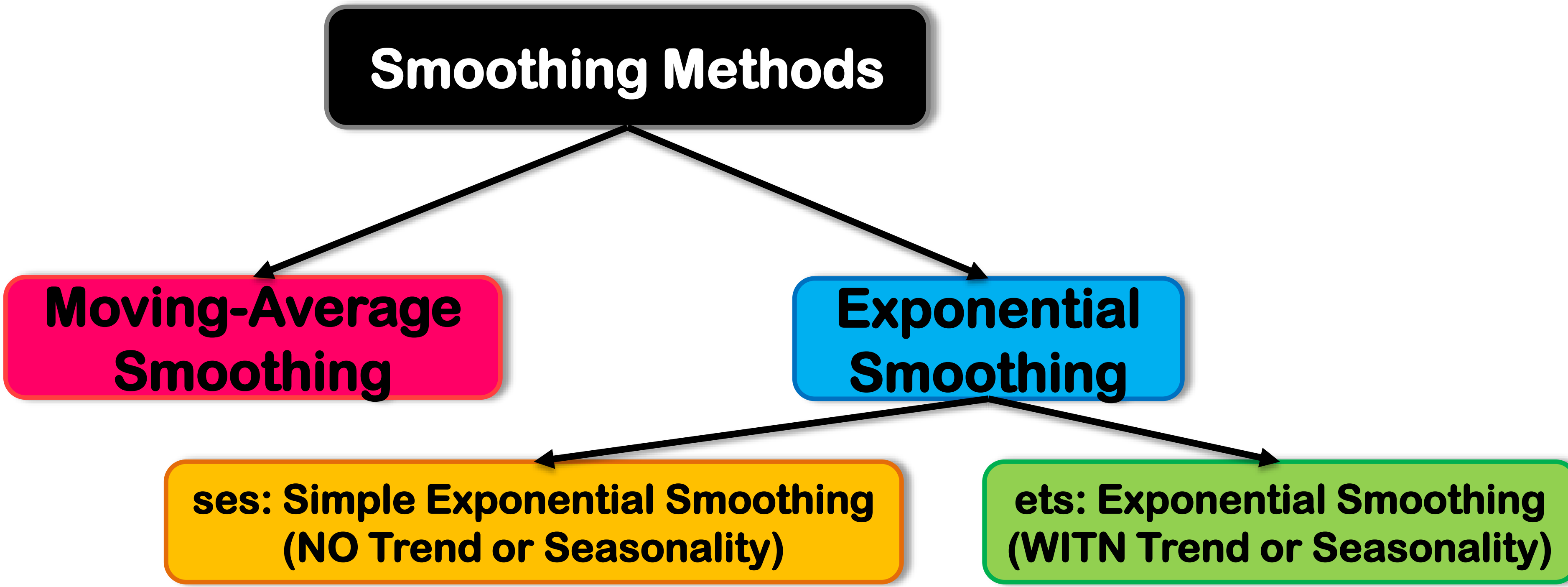
- **Finally, add the trend and seasonality back to the forecast**

Steps with Moving-Average Forecasting

1. **De-trend**
2. **De-seasonalize**
3. **Compute F_{t+k} trailing moving-average forecast**
4. **Add trend and seasonality back to produce the forecast, $\widehat{y_{t+k}}$:**

Smoothing Methods

EXPONENTIAL SMOOTHING



Simple Exponential Smoothing for Forecasting

- **Simple exponential smoothing (SES)** **is a** **weighted average of all past values**, **so that weights decrease exponentially into the past:**
  - unlike forecasting with a moving average of the w most recent values
  - it weights more recent values more heavily
- **Forecast with simple exponential smoothing only after removing trends or seasonality:**
  - **i.e.:** apply exponential smoothing **to the series of residuals**
- **Forecast at time $t+1$ (F_{t+1}) with exponential smother as follows:**

$$\boxed{F_t = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots}$$

$$\boxed{F_t = \alpha y_t + (1-\alpha) F_{t-1}}$$ **recursive definition**

where α is a constant between 0 and 1 called **smoothing constant**

Use Exponential Smoothing for forecasting ONLY <u>after</u> both trends and seasonality are removed.

Exponential Forecaster: Active Learner

- **The exponential smoother is a weighted average of all past observations, with exponentially decaying weights.**
- **Practically it is better to write it in another way:**

$$\boldsymbol{F_{t+1} = F_t + \alpha e_t}$$

  - where e_t is the forecast error at time t
- **This formulation presents the exponential forecaster as an "active learner"**
  - It looks at the previous forecast F_t and its distance from the actual value e_t and then corrects the next forecast based on that information.
    - If the forecast is too high in the last period, the next period is adjusted down.
    - The amount of correction depends on the value of the smoothing constant α

Active Learning Formulation

- **Formulation**

$$F_{t+1} = F_t + \alpha e_t$$

- **is advantageous in terms of data storage and computation time**
  - need to store only forecast and forecast error from the previous period, not entire series
  - Fro real-time forecasting, or forecasting many series in parallel and continuously, such savings are critical
- **The forecasting further into the future yields the same forecasts as a one-step-ahead forecast.**
  - Because series is assumed to lack trend and seasonality, forecasts into the future rely only on information that is available at the time of prediction

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

Choosing Smoothing Constant, α

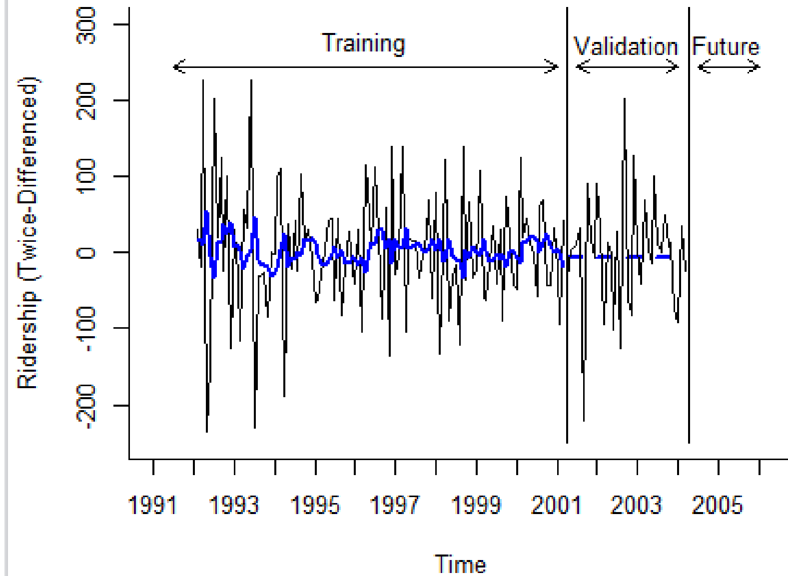
- **The smoothing constant α, which is set by the user, determines the rate of learning.**
  - A value close to 1 indicates fast learning
    - only the most recent observations have influence on forecasts
  - A value close to 0 indicates slow learning
    - past observations have a large influence on forecasts
  - Default values: in the range of 0.1-0.2.
- **Trial and error can also help in the choice of α**
  - Examine the time plot of the actual and predicted series, as well as predictive accuracy
  - MAPE or RMSE of the validation period
- **Finding the α value that optimizes one-step-ahead predictive accuracy over the training period can be used to determine the degree of *local vs. global nature* of the level.**
  - beware of choosing the "**best α**" for forecasting as this can lead to model **overfitting** and low predictive accuracy over in validation

ets() for Simple Exponential Smoothing in R

- In R, forecasting using simple exponential smoothing can be done via the ets() function
- ets stands for Error, Trend, and Seasonality
- The simple exponential smoothing model="ANN"
  - additive error (A)
  - no trend (N)
  - no seasonality (N)

Example: Simple Exponential Smoothing

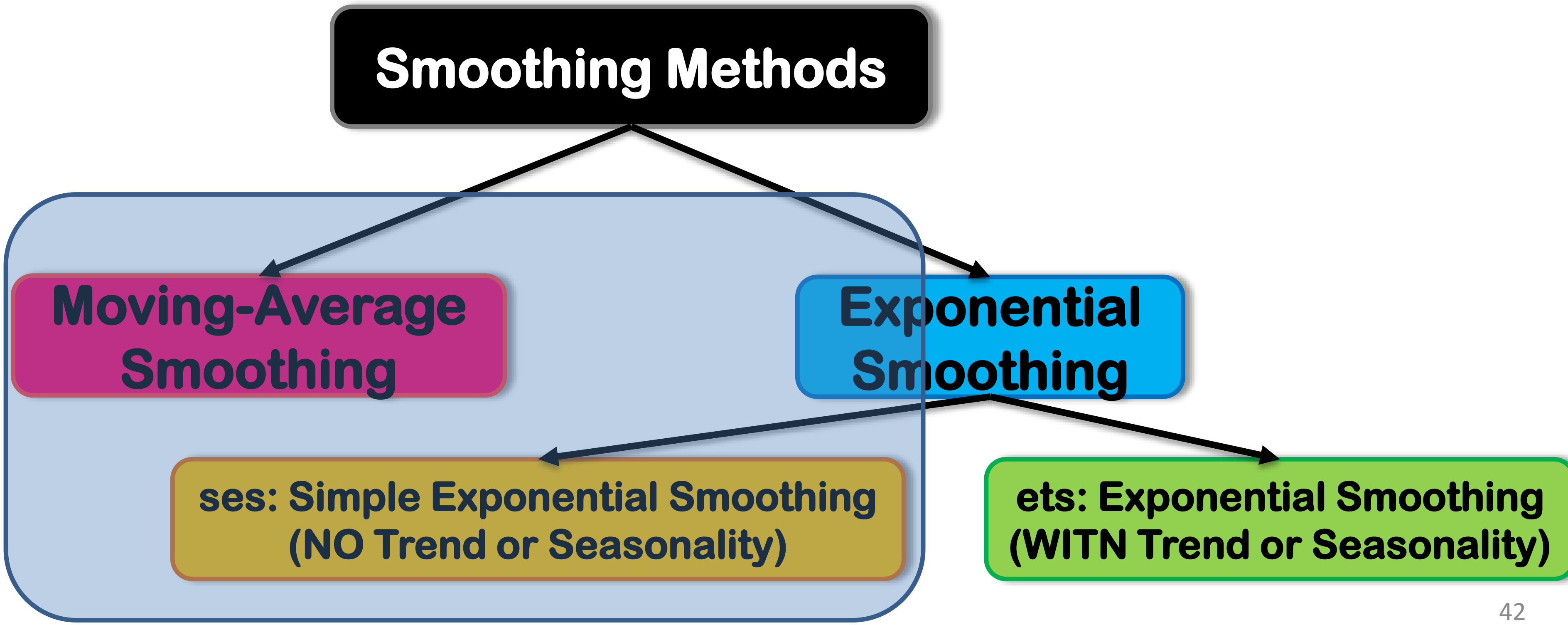
```
Amtrak.data <- read.csv("Amtrak data.csv")
ridership.ts <- ts(Amtrak.data$Ridership,
       start = c(1991, 1),
       end = c(2004, 3), freq = 12)
```




```
diff.twice.ts <- diff(diff(ridership.ts, lag = 12), lag = 1)
nValid <- 36
nTrain <- length(diff.twice.ts) - nValid
train.ts <- window(diff.twice.ts, start = c(1992, 2),
                   end = c(1992, nTrain + 1))
valid.ts <- window(diff.twice.ts, start = c(1992, nTrain + 2),
                   end = c(1992, nTrain + 1 + nValid))
ses <- ets(train.ts, model = "ANN", alpha = 0.2)
ses.pred <- forecast(ses, h = nValid, level = 0)
```

Smoothing Methods

EXPONENTIAL SMOOTHING



Advanced Exponential Smoothing

- **Both moving average and simple exponential smoothing should only be used for forecasting series with no trend or seasonality**
- **One approach is to remove trend and seasonality before smoothing**
- **Another approach is to use more sophisticated version of exponential smoothing that captures trend and/or seasonality**

Holt's Model: Series with an Additive Trend

- **Holt's linear trend model**
  - **For series that contain a trend**
  - **Uses double exponential smoothing**
  - **Trend is not assumed to be global, but can change over time**
- **Double Exponential Smoothing:**
  - the local trend is estimated from the data, and is updated as more data become available
  - the level of the series is also estimated from the data, and is updated as more data become available
- **The k-step-ahead forecast is given by combining the level estimate at time t (L_t) and the trend estimate (which is assumed to be additive) at time t (T_t)**

$$\boxed{F_{t+k} = L_t + kT_t}$$

k-step-ahead Forecast

$$\boxed{F_{t+k} = L_t + kT_t}$$

- **In the presence of a trend, one- two- three-step-ahead forecasts are no longer identical.**
- **Update the level and trend as follows:**

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

  - The level at time t is a weighted average of the actual value at time t and the level in the previous period, adjusted for the trend
  - The trend at time t is a weighted average of the trend in the previous period and the more recent information on the change in level.

- **Two smoothing constants α and β determine the rate of learning**

Two Types of Errors in Exponential Smoothing

- **Additive error**
  - y_{t+1} consists not only of level+trend but also an additional error:
    - $\boldsymbol{y_{t+1} = L_t + T_t + e_t}$
  - the errors are assumed to have a fixed magnitude, irrespective of the current level+trend of the series
- **Multiplicative error**
  - The size of the error grows as the level of the series increases
    - $\boldsymbol{y_{t+1} = (L_t + T_t) \times (1 + e_t)}$
  - Error acts as a percentage increase in the current level+trend

- **In R, etc() function includes either option:**
  - **model = "AAN" or**
  - **model = "MAN"**
  - The first letter represents **error type**

Holt-Winter's Model: Trend + Seasonality

- **The Holt-Winter's exponential smoothing method:**
  - **For series that contain both trend and seasonality**

- **In multiplicative seasonality:**
  - values on different seasons differ by percentage amounts
- **In additive seasonality:**
  - values on different seasons differ by a fixed amount
- **Assuming seasonality with M seasons, the forecast for an additive trend and multiplicative seasonality is given by:**

$$\boxed{F_{t+k} = (L_t + kT_k)S_{t+k-M}}$$

  - By forecasting time t, the series must include at least a full cycle of seasons in order to produce forecasts using this formula ($t > M$)

Update in Holt-Winter's Method

- Being an adaptive method, Holt-Winter's exponential smoothing allows the level, trend and seasonality patterns to change over time
- The three components are estimated and updated as new information arrives
- The updating equations for this additive trend and multiplicative seasonality are:

$$L_t = \frac{\alpha y_t}{S_{t-M}} + (1-\alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

$$S_t = \gamma\left(\frac{y_t}{L_t}\right) + (1-\gamma)S_{t-M}$$

Summary: Exponential Smoothing Methods

Method	Trend	Season	Example
Simple exponential smoothing	NO	NO	ses(x)
Holt's method	Linear	NO	holt(x)
Exponential trend	Exponential	NO	holt(x, exponential=TRUE)
Damped trend	Linear	NO	holt(x, damped=TRUE)
Damped exponential trend	Exponential	NO	holt(x, damped=TRUE, exponential=TRUE)
Holt and Winter's	YES	YES	hw(x)

Exponential Smoothing with ets() in R

- **2 error types** × **3 trend types** × **3 seasonality types**
- **Automated model selection**
  - by leaving out the model option altogether or by using model="**ZZZ**", the ets() function will fit several models and choose the "best" one
  - You can also automatically select a model from a subset of models by putting Z in one or two places in the model specification
    - e.g. model = "**A**ZZ" automatically looks for best **additive error** model
- **"Best" is by minimizing Akaike's Information Criterion (AIC)**
  - combines fit to the training data with a penalty for the number of smoothing parameters and initial values (L_0, T_0, etc) included in the model
- **Selection Metrics:**
  - $\boldsymbol{AIC = -\log(likelihood) + 2p}$ **(Akaike's Information Criterion)**
    - where likelihood is a measure of fit to the training period and p is the number of smoothing parameters and initial states in the model
  - $\boldsymbol{BIC = AIC + p(log(n) - 2)}$ **(Bayesian Information Criterion)**

Taxonomy of Exponential Smoothing Methods

Trend Component		**Seasonal Component** N (None)	A (Additive)	M (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

(N,N): Simple exponential smoothing
(A,N): Holt's linear method
(A,A): Additive Holt-Winters' method
(A,M): Multiplicative Holt-Winters' method
(A_d,M): Damped multiplicative Holt-Winters' method

Taxonomy of Exponential Smoothing Methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (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

R functions	Methods
ses()	(N,N)
holt()	(A,N), (A_d,N), (M,N), (M_d,N)
hw()	(A,A), (A_d,A), (A,M), (A_d,M), (M,M), (M_d,M)

Automated model selection in ets()

- **Here AIC difference of 9.139 strongly favors the MNA model**

```
> ets(train.ts)
ETS(M,N,A)

Call:
 ets(y = train.ts)

  Smoothing parameters:
    alpha = 0.5653
    gamma = 1e-04

  Initial states:
    l = 1899.2864
    s=25.8874 -10.448 -0.6701 -122.7463 199.0173 150.1639
           42.8476 82.9771 52.7806 48.1995 -254.7762 -213.2329

  sigma:  0.0316

     AIC     AICc      BIC
1608.976 1612.865 1648.347
>
```

AIC/BIC

- **Selection Metrics:**
  - $\boldsymbol{AIC = -\log(likelihood) + 2p}$ **(Akaike's Information Criterion)**
    - where likelihood is a measure of fit to the training period and p is the number of smoothing parameters and initial states in the model
  - $\boldsymbol{BIC = AIC + p(log(n) - 2)}$ **(Bayesian Information Criterion)**

- **How to interpret AIC/BIC:**
  - AIC/AICs/BIC does not have much meaning by itself.
  - Only useful in comparison to AIC value for another model fitted to *same data set*.
  - Consider several models with AIC values close to the minimum.
  - A difference in AIC values of 2 or less is not regarded as substantial → choose the simpler but non-optimal model.
  - AIC can be negative.

Example: Using ets() Best Model Selection

```
fit <- ets(ausbeer)
fit2 <- ets(ausbeer,model="AAA",damped=FALSE)
fcast1 <- forecast(fit, h=20)
fcast2 <- forecast(fit2, h=20)
```

```
ets(y, model="ZZZ", damped=NULL, alpha=NULL,
    beta=NULL, gamma=NULL, phi=NULL,
    additive.only=FALSE,
    lower=c(rep(0.0001,3),0.80),
    upper=c(rep(0.9999,3),0.98),
    opt.crit=c("lik","amse","mse","sigma"), nmse=3,
    bounds=c("both","usual","admissible"),
    ic=c("aic","aicc","bic"), restrict=TRUE)
```

Example: ets() (cont.)

```
> fit
ETS(M,Md,M)

  Smoothing parameters:
    alpha = 0.1776
    beta  = 0.0454
    gamma = 0.1947
    phi   = 0.9549

  Initial states:
    l = 263.8531
    b = 0.9997
    s = 1.1856 0.9109 0.8612 1.0423

  sigma:  0.0356

     AIC     AICc      BIC
2272.549 2273.444 2302.715
```

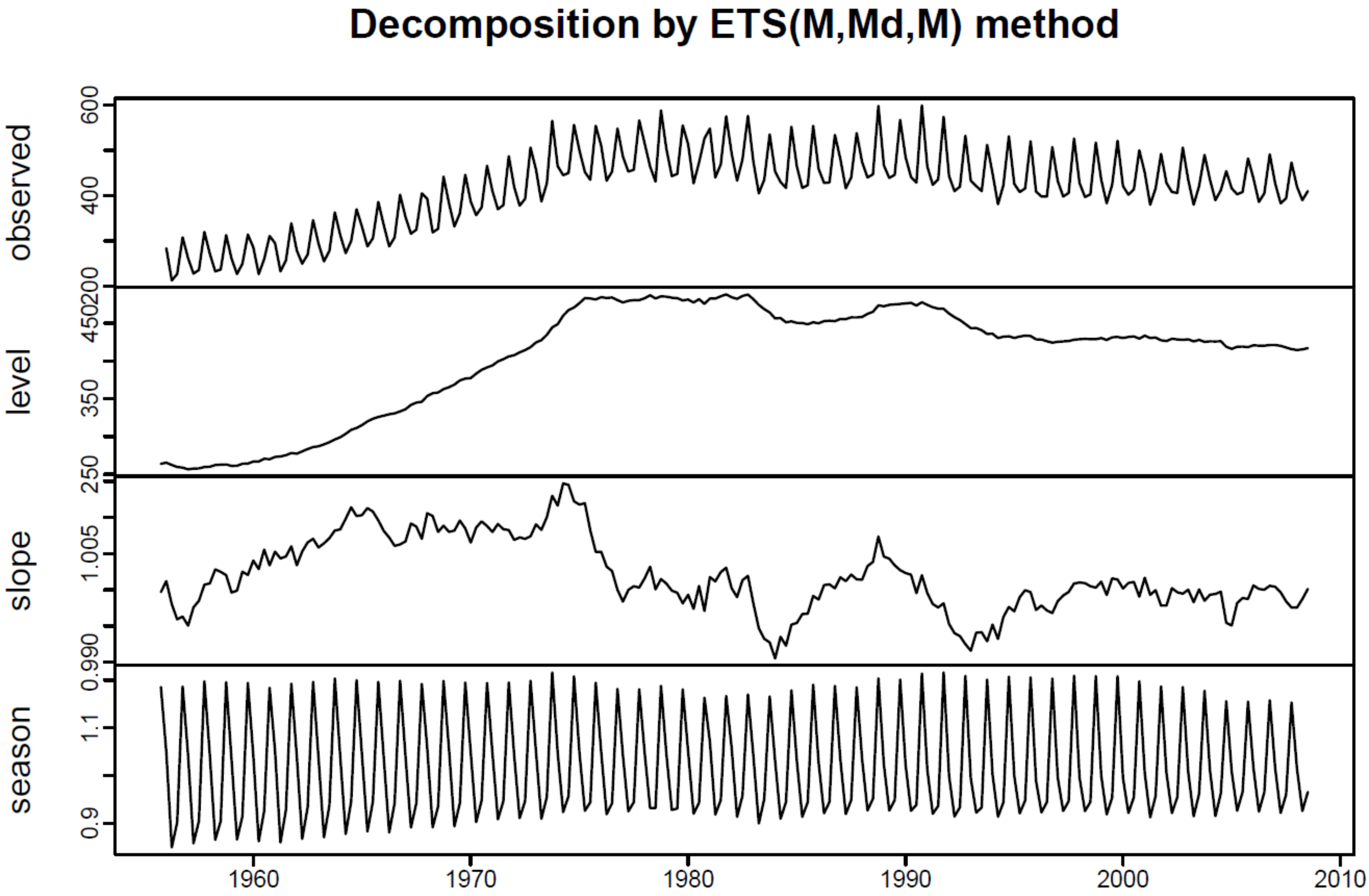
```
> fit2
ETS(A,A,A)

  Smoothing parameters:
    alpha = 0.2079
    beta  = 0.0304
    gamma = 0.2483

  Initial states:
    l = 255.6559
    b = 0.5687
    s = 52.3841 -27.1061 -37.6758 12.3978

  sigma:  15.9053

     AIC     AICc      BIC
2312.768 2313.481 2339.583
```

Summary of ets() function

ETS: **E**rror, **T**rend, **S**easonal

- **Automatically chooses** a model by default using the AIC/AICc/BIC
- Can handle any **combination of trend, seasonality and damping**
- Produces prediction intervals for every model
- Ensures the parameters are admissible
- In practice, the models work fine for short- to medium-term forecasts provided the data are strictly positive.
- Produces an object of class ets:
  - Methods: coef(), plot(), summary(), residuals(), fitted(), simulate(), and forecast()
  - plot() function shows time plots of the original time series along with the extracted components (level, growth and seasonal).

plot(fit)

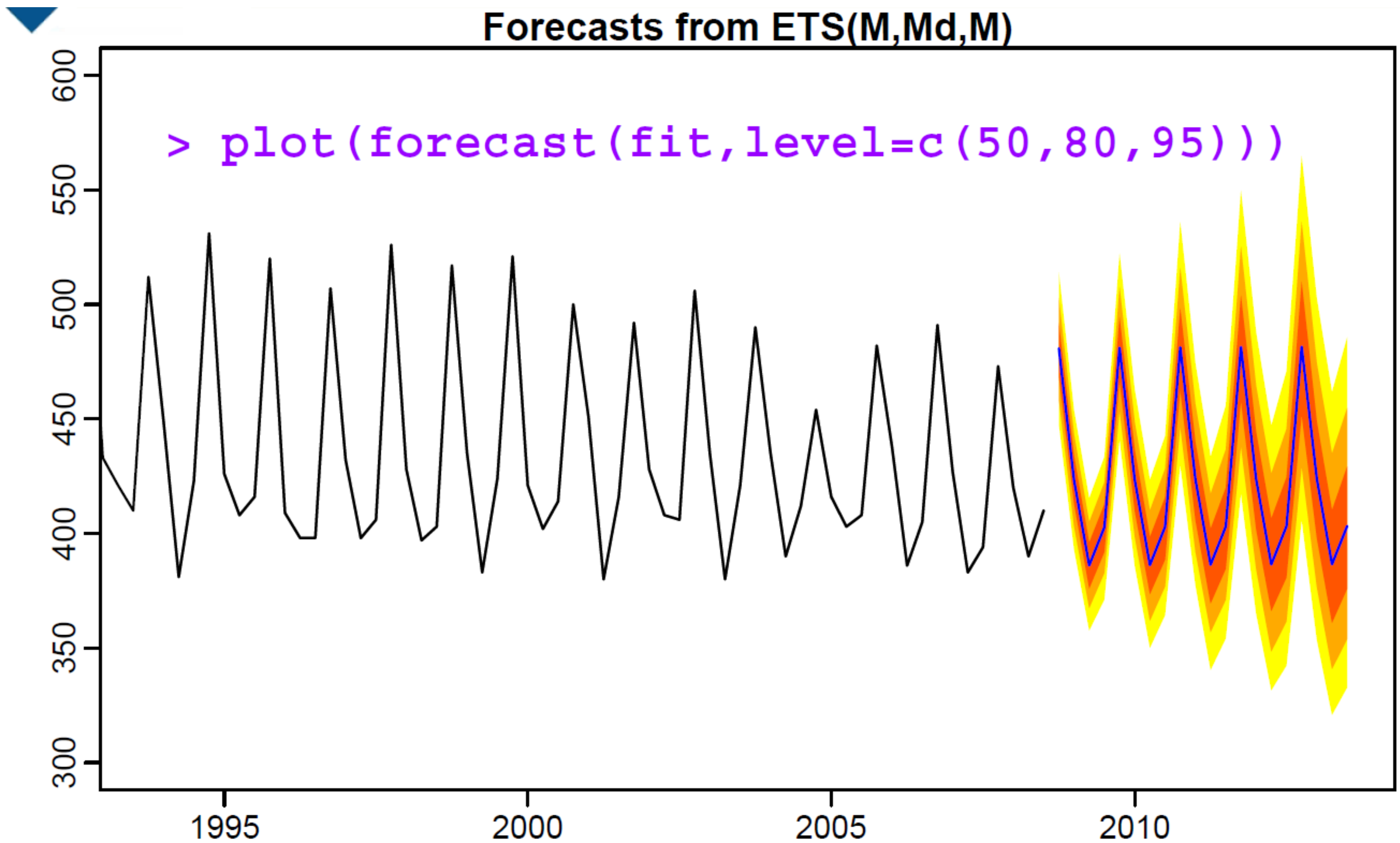


Decomposition by ETS(M,Md,M) method
observed
level
slope
season
600
400
200
450
350
250
1.005
0.990
1.1
0.9
1960
1970
1980
1990
2000
2010

Accuracy (more in the future lecture)

```
> accuracy(fit)
      ME      RMSE       MAE       MPE      MAPE      MASE
 0.17847 15.48781 11.77800  0.07204  2.81921  0.20705

> accuracy(fit2)
      ME      RMSE       MAE       MPE      MAPE      MASE
-0.11711 15.90526 12.18930 -0.03765  2.91255  0.21428
```

Forecast from ETS()



Acknowledgements

- Books
  - Free and online (otexts.com/fpp): Forecasting Principles & Practice by R. Hyndman, G. Athanasopoulos ← Excellent Book!!!
  - Practical Time Series Forecasting with R: A Hand-on Guide by Shmueli & Lichtendahl
- Packages
  - R: fpp (install.packages ("fpp", dependencies=TRUE))