지수 평활법 (exponential smoothing)

* 직관적이어서 이해하기 쉽고 (ARIMA 대비 상대적으로요)
* 시계열 자료의 구성요소가 변동이 느리거나 매우 규칙적(periodic)일 때 예측 정확도도 높아서 시계열 자료 예측에 실무적으로 많이 사용

기본 개념 : 최근 관측한 값에 높은 가중치를 주고, 먼 과거의 관측치에는 낮은 관측치를 주어 가중 평균값을 구한는 것

가중치는 현재로부터 과거로 갈수록 지수적으로 감소시켜서 차등적으로 주는 평활 함수(smoothing function)을 사용하는 것에서 "지수(exponential)" 이름을 따왔습니다. 그리고 여러개의 관측치를 모아서 가중 평균을 사용하기 때문에 "평활(smoothing)"되는 효과가 있음

[ 시계열 특성에 따른 지수평활법 기법 결정 ]

(1) 시계열 자료에 추세(Trend)가 있는지, 추세가 있다면 1차 선형인지 아니면 비선형 인가?

- 추세 없음 (No Trend) --> Simple Exponential Smoothing

- 1차 선형 추세 (Linear Trend) --> Two Parameter Exponential Smoothing

- 2차 비선형 추세 (Quadratic Trend) --> Three Parameter Smoothing

(2) 시계열 자료에 계절성(Seasonality)이 있는가?

- 계절성 없음 (No Seasonality)

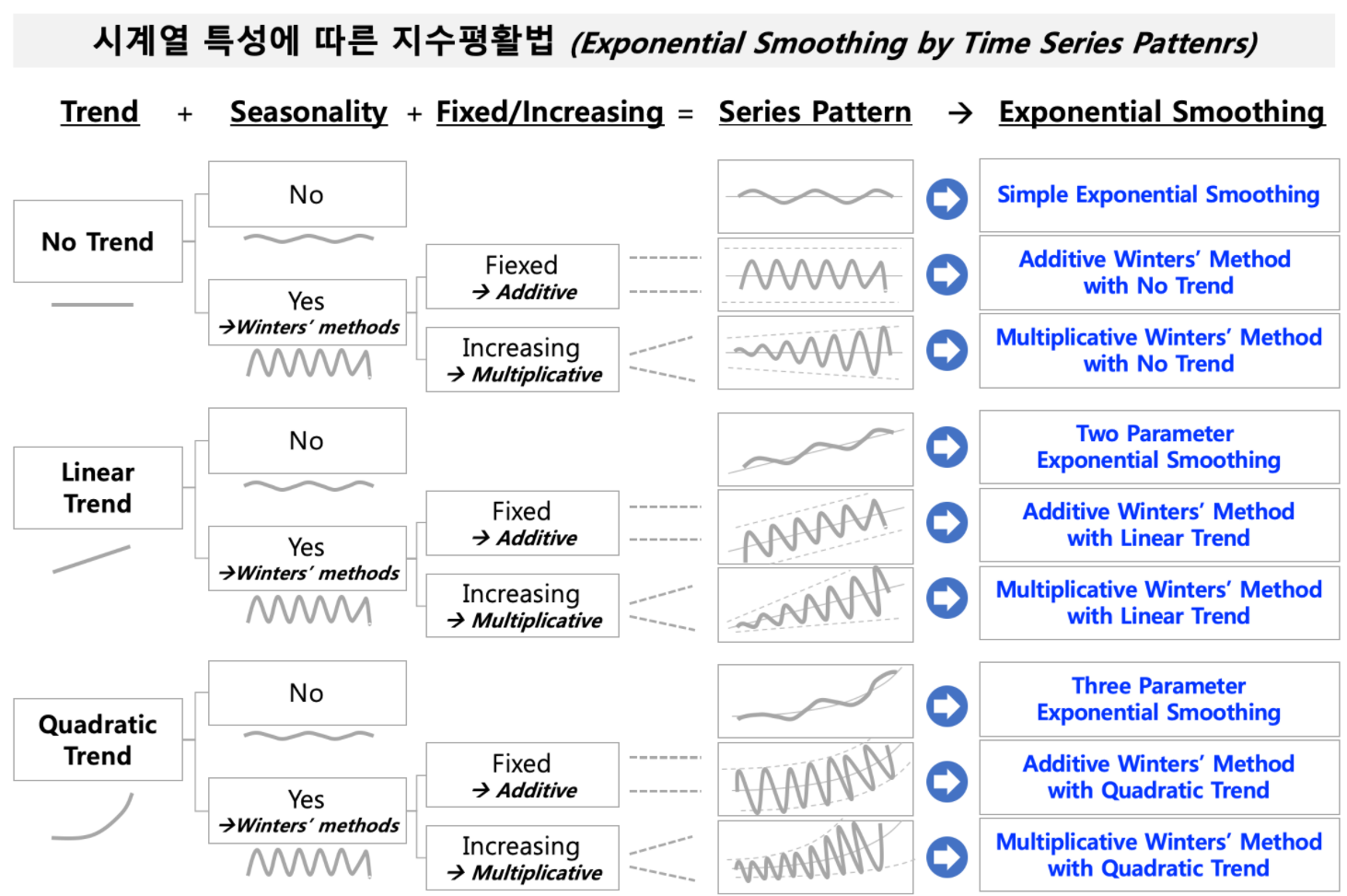
- 계정성 있음 (with Seasonality) --> Winters' Method

(3) 시계열 자료의 계절성이 시간이 지남에 따라 고정(fixed)되어 있는지 아니면 확산(increasing)되는가?

- 고정(상수) 계절 변동 (fixed seasonal variation) --> Additive Model

- 확산 계절 변동 (increasing seasonal variation) --> Multiplicative Model

위의 설명을 각 시계열 자료 패턴별로 시도표 예시와 함께 decision tree 형태로 구분해서 지수 평활법 (exponential smoothing) 기법을 짝지어보면 아래와 같습니다.



추세와 계절성이 모두 없는 단순 지수 평활법 (Simple Exponential Smoothing)

추세는 없고 고정계절변동이 있는 가법 윈터스 방법 (Additive Winters' Method Exponential Smoothing with No Trend)

추세는 없고 승법 윈터스 방법 (Multiplicative Winters' Method Exponential Smoothing with No Trend)

1차 선형 추세는 있고 계절성은 없는 이중 지수 평활법 (Two Parameter Exponential Smoothing)

1차 선형 추세와 고정계절변동이 있는 가법 윈터스 지수 평활법 (Additive Winters' Method Exponential Smoothing with Linear Trend)

1차 선형 추세와 확산계절변동이 있는 승법 원터스 지수 평활법 (Multiplicative Winters' Method Exponential Smoothing with Linear Trend)

2차 비선형 추세는 있고 계절성은 없는 삼중 지수 평활법 (Three Parameter Exponential Smoothing)

2차 비선형 추세와 고정계절변동이 있는 가법 윈터스 지수 평활법 (Additive Winters' Method Exponential Smoothing with Quadratic Trend)

2차 비선형 추세와 확산계절변동이 있는 승법 윈터스 지수 평활법 (Multiplicative Winters' Method Exponential Smoothing with Quadratic Trend)

https://t1.daumcdn.net/cfile/tistory/999B593B5E117A002F를 지수 평활법으로 예측하고자 하는 t 시점의 시계열

https://t1.daumcdn.net/cfile/tistory/999E69335E117C3F2B를 t 시점의 계절 요소값

https://t1.daumcdn.net/cfile/tistory/9919F4345E117C7330를 t 시점에서의 오차

출처: https://rfriend.tistory.com/511 [R, Python 분석과 프로그래밍의 친구 (by R Friend)]