Chapter 15 Forecasting

예측에는 qualitative forecasting 과 quantitative forecasting 이 있는데 qualitative forecasting 은 앞으로 발생될 사건을 경험으로 부터 얻은 직감에 따른 예측을 하는 주관적인 방법론이고 quantitative forecasting 은 과거의 수치 데이타와 수학을 통해서 예측을 하는 것을 말합니다. 이 장에는 다룰 내용은 quantitative forecasting 에 대한 기술적인 내용들이 됩니다.

지금까지 배운 예측 위한 근본은 correlation 의 존재 입니다. Correlation 의 요점은 정보간의 상호 관련이 있다면 correlation 이 형성이 되기 때문에 어떤 값들만 알면 다른 값을 예측할 수 있다는 것을 의미합니다. 이처럼 예측이 가능한 경우들은 우리가 실제 경험하지 않아도 앞으로 발생될 사건을 미리 인지할 수 있습니다. 하지만 간단한 linear regression model 로 예측을 하는 것은 전체적인 흐름만 이해를 돕지 주기적으로 발생되는 사건에 대해서는 예측을 할 수 없기 때문에 이것만으로는 미래를 예측하기에는 부족합니다.

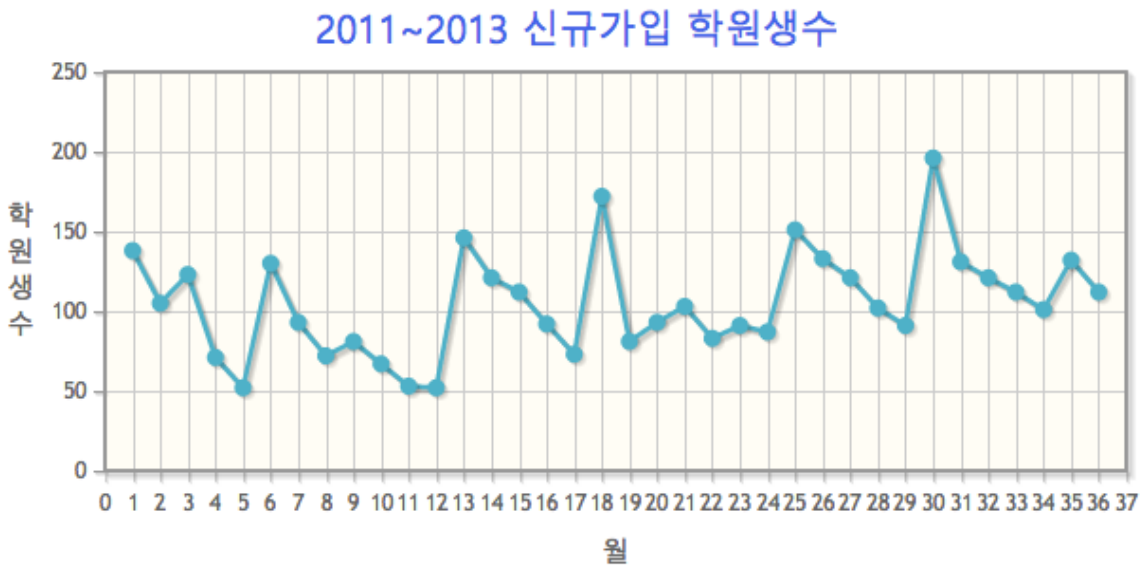
예를 들어 빙수를 판매하는 상점에서 상점 매출액이 계속 증가하는 경우 시간과 매출액의 선형 관계로 간단한 linear regression 만으로도 예측이 가능하지만 매출액의 변화를 월별로 보게 된다면 여름철에는 매출액이 늘고 가을 겨울로 가면서 매출액이 작아지는 형태로 나타난다면 간단한 linear regression 는 이러한 계절별 변화를 예측하지는 못합니다. 이러한 단점을 극복하기 위해서 예측값의 주기적인 특성을 해부하여 예측을 성공적으로 하는 방법을 배우는 것이 이 장에서 배울 내용입니다.

이번 장에서 다루게 되는 예제는 다음과 같은 3 년간 성인 대상 어학원에서 신규 수강생들의 정보입니다.

달	1	2	3	4	5	6	7	8	9	10	11	12
2011	138	105	123	71	52	130	93	72	81	67	53	52
2012	146	121	112	92	73	172	81	93	103	83	91	87
2013	151	133	121	102	91	196	131	121	112	101	132	112

단위: 인원수

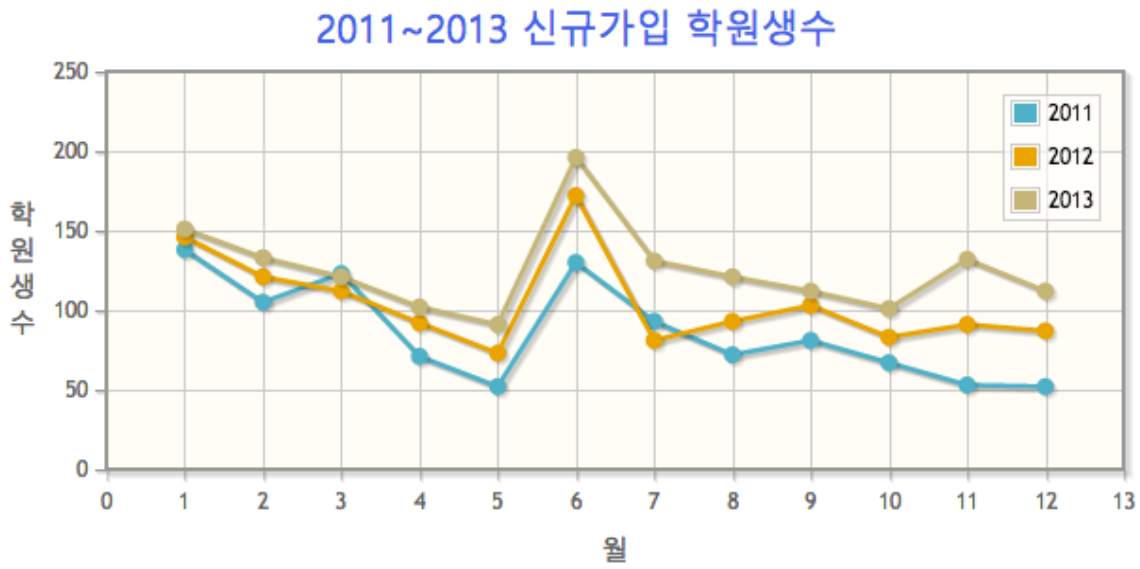
이 처럼 기간동안에 발생되는 값들이 나열된 것을 time series 이라 합니다. 값들을 그래프로 그려보고 데이타를 요소별로 분리하여 시간에 따른 특징들을 발견하는 것이 quantitative forecasting 에서 학습할 내용입니다.



3 년동안 신규가입 학원 수강생 변동의 내면을 살펴보면 다음의 4 가지 항목이 포함되어 있습니다.

1) Trend
2) Seasonal
3) Cyclical
4) Random

첫번째 Trend 요소는 데이타가 전체적으로 위로 올라가는지 아님 아래로 내려가는지를 알려주는 정보로 그림을 보면 신규가입 학생수는 점점 증가 추세를 나타납니다.



두번째 Seasonal 요소는 시간대, 월, 분기등과 같이 구분하여 특징을 나타나내는 것으로 신규 가입 수강생 수를 보면 매년 1 월과 6 월에 가입자가 많고 5 월과 11 월이 작은 것으로 나타납니다.

다음으로 Cyclical 요소는 수년 동안 측정을 하여 자료를 보았을 때 데이타의 전체적인 흐름이 아래로 떨어지는 구간인자 아니면 올라가고 있는 구간인지를 나타내는 것으로 예를들어 경기가 좋은 때 소매자 지출금액은 상승하지만 경기가 나쁠때는 지출금액이 낮아지게 됩니다. 이렇듯 상승과 하락을 반복하는 가정하에 상승 구간에 있는것과 하락 구간에 있는 것을 나타내는 것입니다.

마지막으로 random 요소는 위의 3 가지로 설명이 되지 않는 것들을 모두 포함하고 있으며 다른 말로 noise 라고도 합니다.

1. Smoothing Forecasting

예측을 할 수 없는 random 요소의 특징은 이 값이 정말로 noise 로만 구성이 되어 있다면 평균값이 0 이 되는 것입니다. 이러한 특성을 이용하여 예측을 하는 방식들을 설명하겠습니다.

1.1.Simple Moving Average (SMA) Forecast

예를 들어서 2011 년 4 월 신규가입 수강생 수를 예측하기 위해서 2011 년 1 월, 2 월, 3 월의 평균값으로 예측값을 만드는 방식입니다.

$$SMA_4 = \frac{138 + 105 + 123}{3} = 122$$

이러한 방식으로 계산된 예측값에 실제값을 뺀 절대값들의 합의 평균을 Mean absolute deviation(MAD)라고 합니다.

$$MAD = \frac{\sum_{i=p+1}^{n} |SMA_i - A_i|}{n - p} \quad (15.1)$$

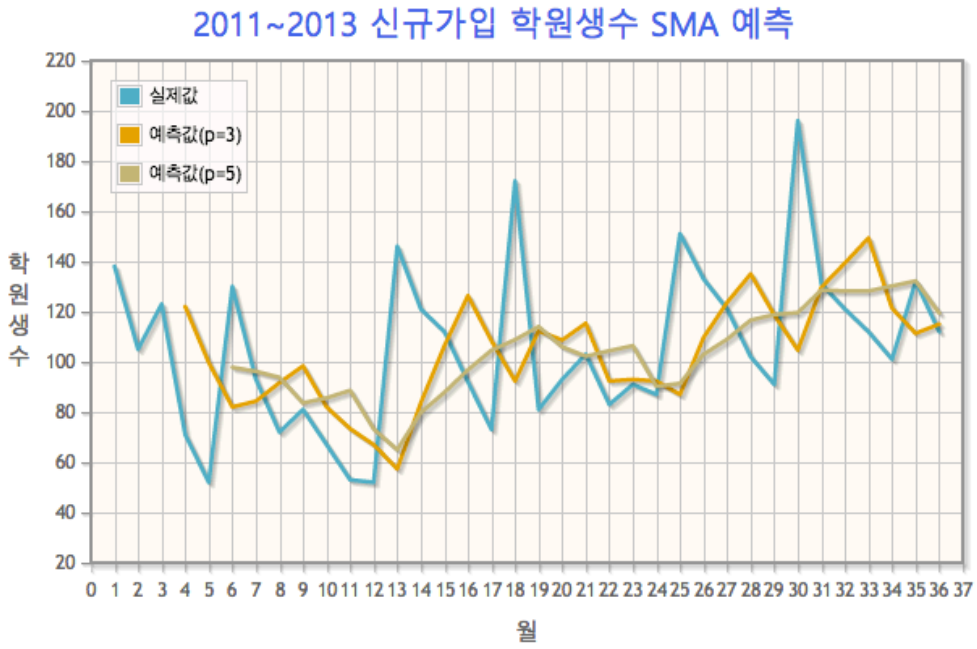
n: sample 의 개수
p: SMA 를 위해 사용된 sample 의 크기
A:실제값

이 수식을 신규가입 수강생 자료값 p=3 과 p=5 으로 적용하면 다음과 같은 결과를 얻게 됩니다.

년/월	실제값	P=3 예측값	P=5 예측값	P=3 MAD	P=5 MAD
2011/01	138				
2011/02	105				

2011/03	123				
2011/04	71	122.00		51.00	
2011/05	52	99.67		47.67	
2011/06	130	82.00	97.80	48.00	32.2
2011/07	93	84.33	96.20	8.67	3.2
…	…	…	…	…	…
2013/12	112	115.00	118.17	3.00	7.4
MAD				28.57	24.23

MAD 값의 의미는 예측값의 평균 오차값으로 이 예제에서 보면 p=5 개씩 잡아서 예측값을 계산할 때가 p=3 보다 오차값이 작게 나타나지만 p 를 늘린다고 예측 error 가 꼭 줄어들지는 않습니다. 하지만 p 값이 클 수록 예측값에 대한 값의 변동은 더욱 부드러워지게 됩니다.



jMath.prototype.forecast('sma', p)

```
var ts = [
     jMath([[1,138],[2,105],[3,123],[4,71],[5,52],[6,130]
           ,[7,93],[8,72],[9,81],[10,67],[11,53],[12,52]]),
     jMath([[1,146],[2,121],[3,112],[4,92],[5,73],[6,172]
           ,[7,81],[8,93],[9,103],[10,83],[11,91],[12,87]]),
     jMath([[1,151],[2,133],[3,121],[4,102],[5,91],[6,196]
          ,[7,131],[8,121],[9,112],[10,101],[11,132],[12,112]])
];
var list = jMath.joinByRow(ts);
for ( var i = 0 ; i < list.rows ; i++ )
{
    list[i][0] = i+1;
}
var fcast3 = list.forecast('sma',3);
console.log(fcast3);
```

```
4       122
5       99.66666666666667
6       82
7       84.33333333333333
8       91.66666666666667
9       98.33333333333333
10      82
11      73.33333333333333
12      67
13      57.333333333333336
14      83.66666666666667
15      106.33333333333333
...
rows    33
cols    2
mad     28.56565656565657
```

변수 ts 는 3 개의 jMath object 를 갖고있는 array 로 각 jMath object 는 [월,수강생]을 하나의 row 로 구성하여 12x2 크기의 matrix 를 형성합니다. 하지만 우리가 SMA 를 실행하기 위해서 필요한 것은 이 3 개의 jMath object 를 하나로 묶는 것이기 때문에 jMath.joinByRow(ts)로 3 개의 jMath object 를 36x2 크기로 된 하나의 jMath object 로 만듭니다.

다음 SMA 계산을 위해서 [월,수강생]값에서 월을 1 년 단위가 아닌 총 경과된 월로 수정하기 위해서 for 문으로 값을 수정합니다. 결과로 나타난 값은 각 경과된 월에 해당하는 예측값들입니다.

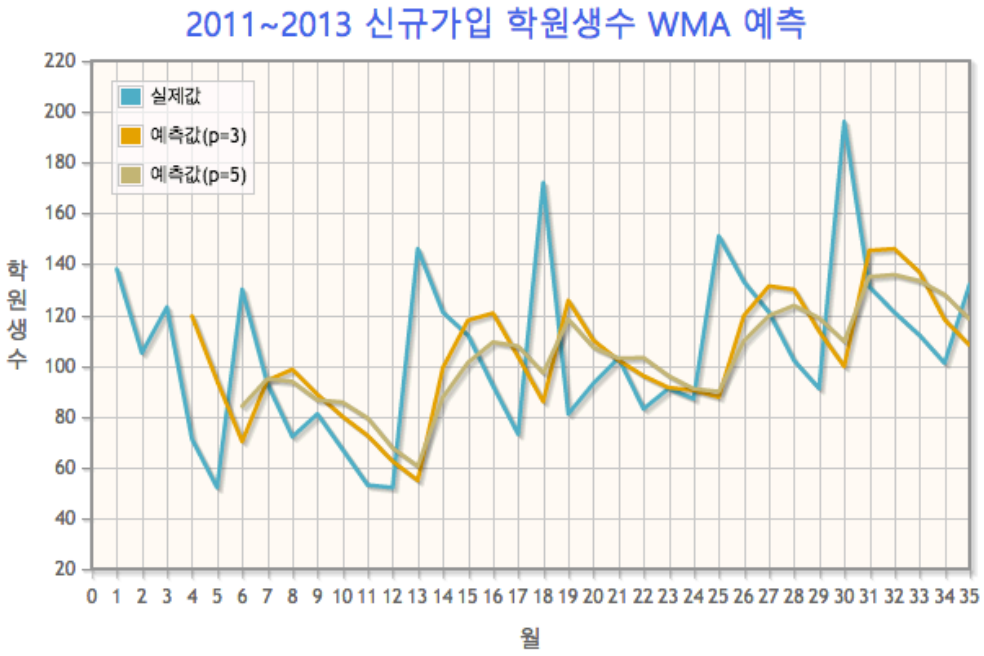
1.2. Weighted Moving Average (WMA) Forecast

Simple Moving Average 방식은 과거의 p 개의 자료값의 영향을 균등하게 하여 예측값을 생성했지만, 만일 예측을 위해 최근의 자료에 더 많은 영향을 주고 과거의 자료일 수록 영향을 작게 주는 방식으로 하려면 각 값에 weight 을 다른게 주어 계산을 해야 합니다. 예를 들어 신규가입 수강생수를 계산할 때 다음과 같이 예측값을 계산 할 수 있습니다.

$$WMA_4 = \frac{1 \times 138 + 2 \times 105 + 3 \times 123}{1 + 2 + 3} = 119.5$$

이 수식은 2011 년 4 월의 신규가입 수를 예측하기 위해서 3 월의 자료값에 더 영향력을 주고 1 월에 영향력을 작게 주는 방식입니다. 이와 같은 방식을 적용한 결과는 다음과 같습니다.

년/월	실제값	P=3 예측값	P=5 예측값	P=3 MAD	P=5 MAD
2011/01	138				
2011/02	105				
2011/03	123				
2011/04	71	119.50		48.50	
2011/05	52	94.00		42.00	
2011/06	130	70.17	84.07	59.83	45.93
2011/07	93	94.17	94.80	1.17	1.80
…	…	…	…	…	…
2013/12	112	118.33	118.20	6.33	6.2
MAD				27.8	25.21



결과를 보면 weight 으로 1, 2, 3 으로 했을 경우 MAD 값 27.8 은 SMA 로 28.57 보다 더 작게 나타납니다. 이는 weight 으로 사용되는 값에 따라서 다른 결과를 나타날 것입니다

두 결과를 자세히 보면 예측값의 흐름은 실제값보다 늦게 올라가거나 내려갑니다. 이유는 time series 에 trend 와 seasonal 요소가 있기 때문입니다.

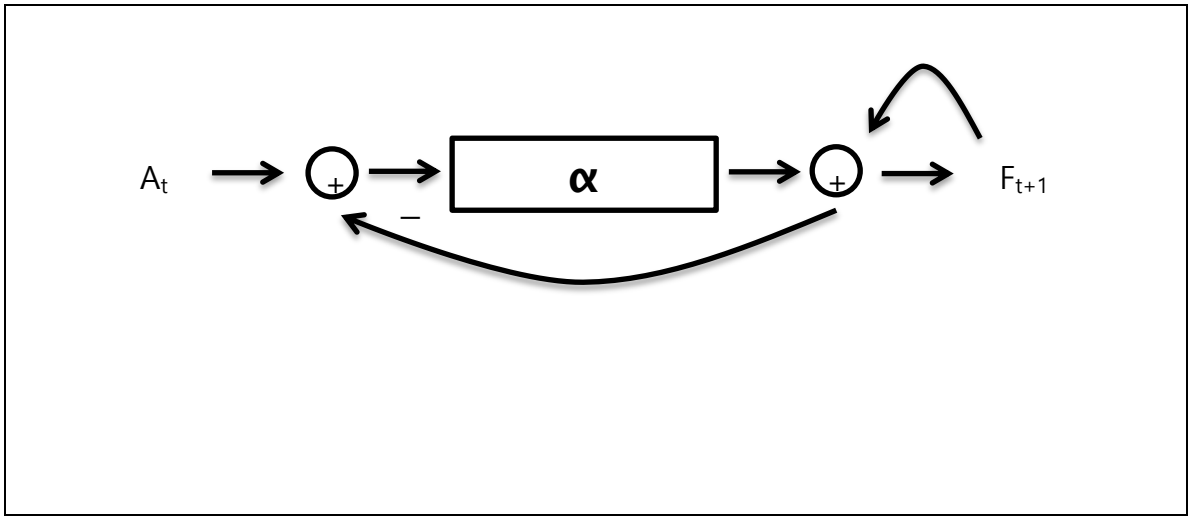
```
jMath.prototype.forecast('wma', weights)
```

```
var ts = [
     jMath([[1,138],[2,105],[3,123],[4,71],[5,52],[6,130]
           ,[7,93],[8,72],[9,81],[10,67],[11,53],[12,52]]),
     jMath([[1,146],[2,121],[3,112],[4,92],[5,73],[6,172]
           ,[7,81],[8,93],[9,103],[10,83],[11,91],[12,87]]),
     jMath([[1,151],[2,133],[3,121],[4,102],[5,91],[6,196]
          ,[7,131],[8,121],[9,112],[10,101],[11,132],[12,112]])
];
var list = jMath.joinByRow(ts);
for ( var i = 0 ; i < list.rows ; i++ )
{
    list[i][0] = i+1;
}
var fcast3 = list.forecast('wma',[3,2,1]);
console.log(fcast3.mad);

27.79797979797981
```

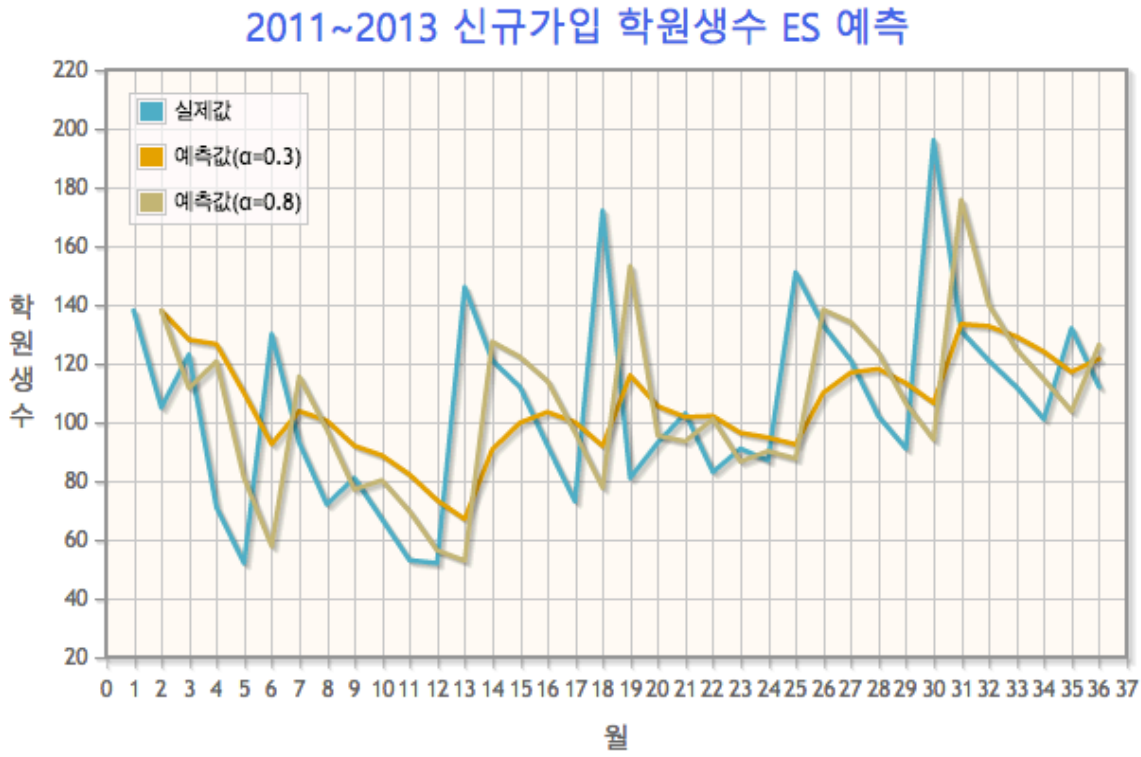
1.3.Exponential Smoothing(ES) Forecast

예측하기 위해서 전 예측값과 실제값의 오류값을 기반으로 예측값을 만들어 나가는 방식입니다.

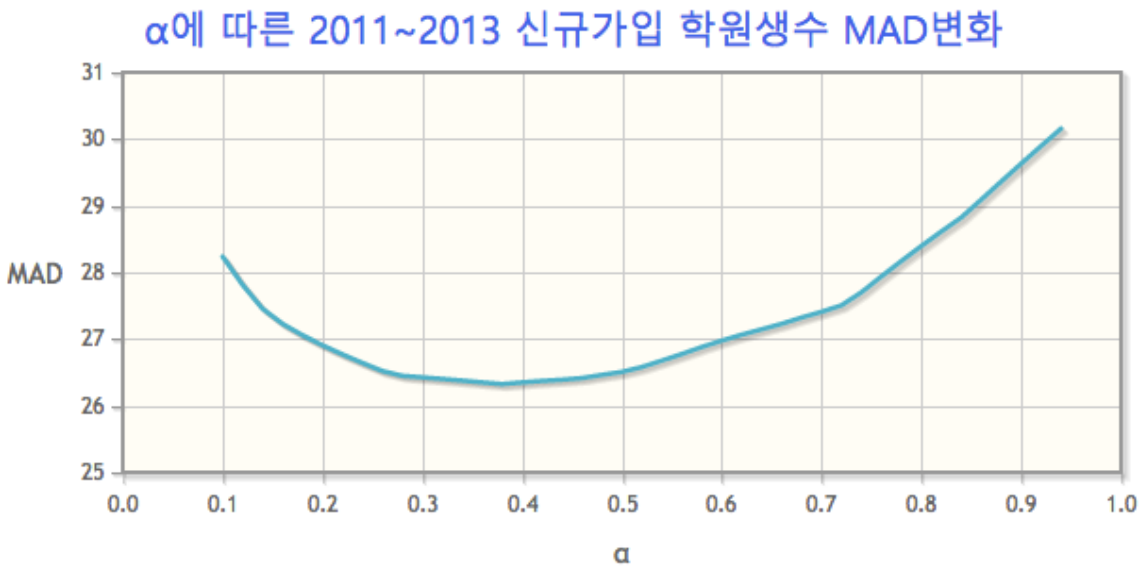


$$F_{t+1} = F_t + \alpha(A_t - F_t) \tag{15.2}$$

과거 값을 기반으로 자신이 스스로 오류를 수정하여 예측을 하는 방식으로 α값은 0~1 사이에 입니다. 만일 α가 크다면 과거 값에 민감하게 반응을 하고 그렇지 않으면 작게 반응을 합니다. 또한 전 오류값이 크면 예측값에 많은 변화가 나타나게 됩니다.



MAD 값은 α가 0.3 일 때 26.42, α가 0.8 일 때 28.4 로 즉 α값에 따라서 MAD 값이 다르기 때문에 최적의 값을 찾아야 합니다.



jMath.prototype.forecast('exp', alpha)

```
var ts = [
     jMath([[1,138],[2,105],[3,123],[4,71],[5,52],[6,130]
           ,[7,93],[8,72],[9,81],[10,67],[11,53],[12,52]]),
     jMath([[1,146],[2,121],[3,112],[4,92],[5,73],[6,172]
           ,[7,81],[8,93],[9,103],[10,83],[11,91],[12,87]]),
     jMath([[1,151],[2,133],[3,121],[4,102],[5,91],[6,196]
          ,[7,131],[8,121],[9,112],[10,101],[11,132],[12,112]])
];
var list = jMath.joinByRow(ts);
for ( var i = 0 ; i < list.rows ; i++ )
{
    list[i][0] = i+1;
}
var fcast3 = list.forecast('exp',0.3);
console.log(fcast3.mad);

26.425226956645734
```

1.4.Exponential Smoothing with Trend Adjustment(ESTA) Forecast

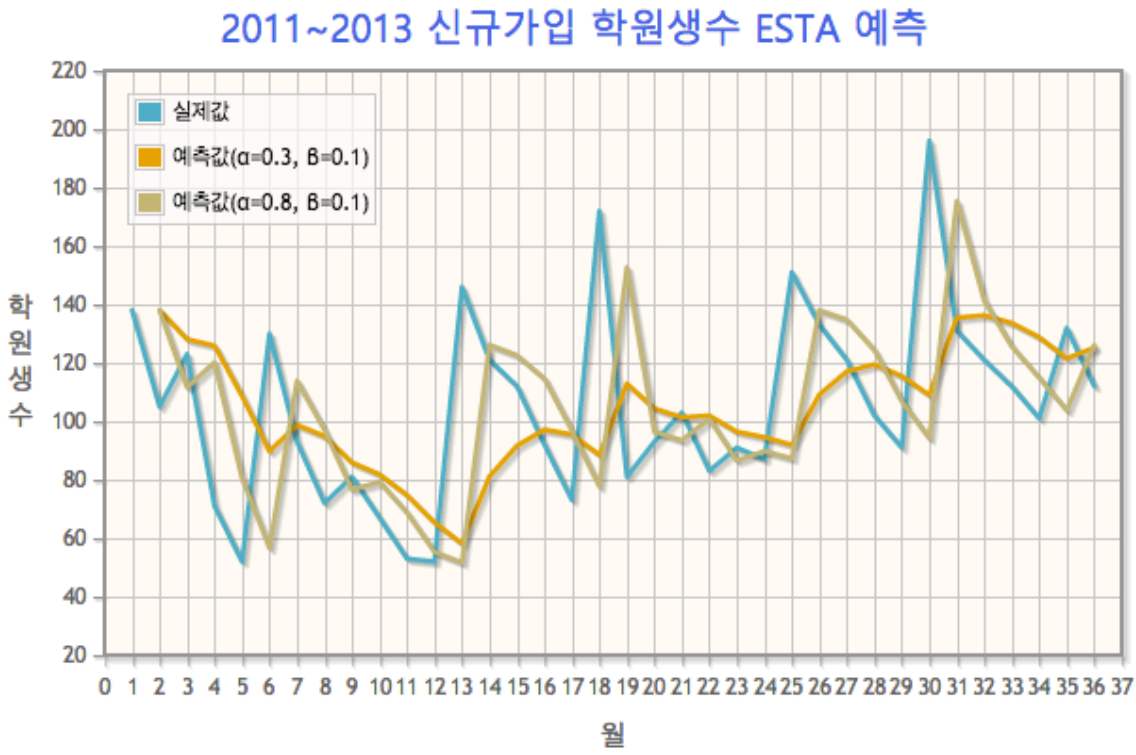
Moving average 와 exponential smoothing 방식에 문제점은 random 요소를 제거하는데 목적을 두고 설계된 방식이기 때문에 그 외의 요소인 trend, seasonal, cyclical 요소가 중요하게 작용하고 있다면 예측 결과가 좋지 못합니다. 예를 들어 신규가입 학생수를 보면

seasonal 요소로 신규가입 학원생의 수가 많은 달이 있고 적은 달이 있는데 앞에 소개드린 예측 방식을 적용한 결과를 보면 실제값 보다 늦게 값의 변화가 나타나는 현상을 보셨습니다.

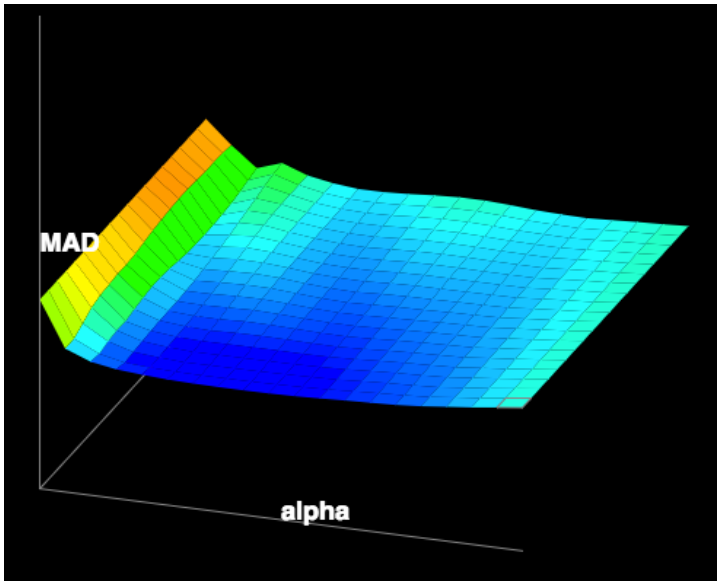
이를 보안하기 위한 방안으로 exponential smoothing 기법에 data 에서 발견되는 trend 를 보상하는 방식을 이용하여 해결을 할 수 있습니다.

$$\begin{aligned} FIT_t &= F_t + T_t \\ F_{t+1} &= FIT_t + \alpha(A_t - FIT_t) \\ T_{t+1} &= \beta(F_{t+1} - F_t) + (1-\beta)T_t \end{aligned} \tag{15.2}$$

FIT 는 수식에서 보듯이 trend 가 포함된 예측값이고 F 는 exponentially smooth 된 예측값, T 는 exponentially smooth 된 trend 입니다. 여기서 T 값을 계산하기 위한 β 값을 보게 되면 이 값이 1 에 가까울 수록 최근에 예측한 trend 가 영향을 강하게 주고, 반대로 0 에 가까우면 전체 trend 에 영향을 더 많이 받게 됩니다.



MAD 값은 α가 0.3 일 때 26.235, α가 0.8 일 때 28.4 로 전보다는 약간 향상이 되었습니다. 이 방법역시 Exponential Smoothing 방식과 같이 MAD 가 최적이 되는 α와 β값을 찾아야 합니다.



여기서 파랑색 영역이 낮은 값을 갖는 영역으로 MAD 값의 최저는 β는 0 이고 α값이 0.45 근방에서 26.35 정도가 되는데 지금까지의 모든 smoothing 방식을 보면 SMA 방식이 가장 낮은 MAD 값을 얻는 결과를 만들니다. 하지만 항상 SMA 가 좋은 결과를 내는것은 아니기 때문에 SMA 를 예측을 위해 항상 사용하지는 못합니다.

jMath.prototype.forecast('expta', alpha, beta)

```
var ts = [
     jMath([[1,138],[2,105],[3,123],[4,71],[5,52],[6,130]
           ,[7,93],[8,72],[9,81],[10,67],[11,53],[12,52]]),
     jMath([[1,146],[2,121],[3,112],[4,92],[5,73],[6,172]
           ,[7,81],[8,93],[9,103],[10,83],[11,91],[12,87]]),
     jMath([[1,151],[2,133],[3,121],[4,102],[5,91],[6,196]
          ,[7,131],[8,121],[9,112],[10,101],[11,132],[12,112]])
];
var list = jMath.joinByRow(ts);
for ( var i = 0 ; i < list.rows ; i++ )
{
    list[i][0] = i+1;
}
var fcast3 = list.forecast('expta',0.3,0.1);
console.log(fcast3.mad);

26.2350362457034
```

2. Regression 분석을 이용한 Forecast

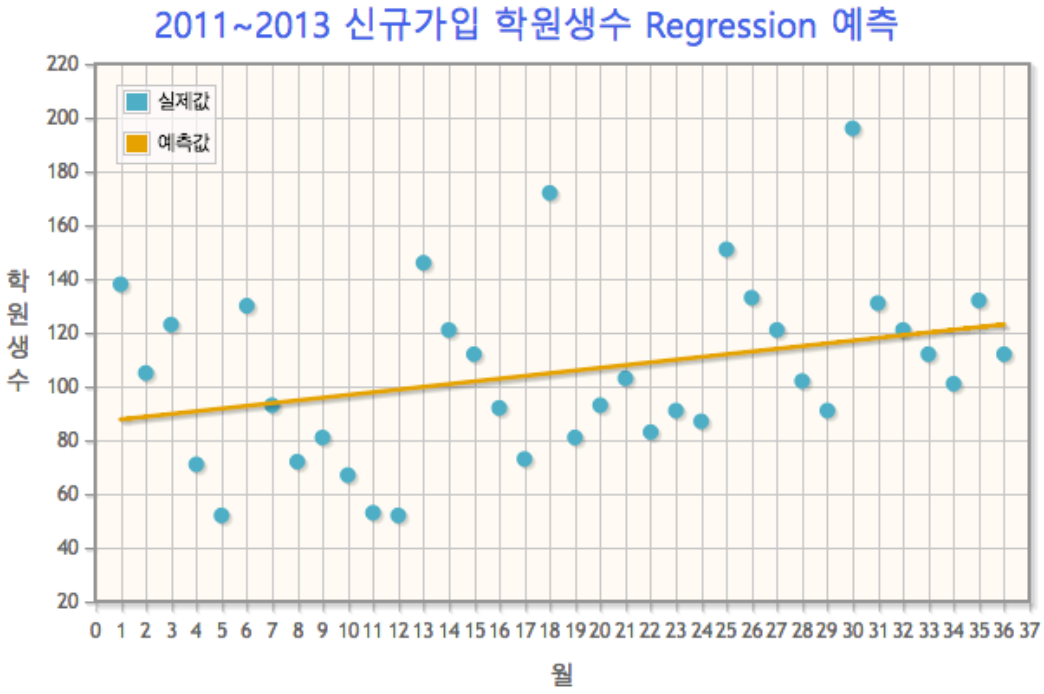
X 값은 시간이 되고 Y 값은 이에 해당하는 값으로 이러한 방법은 simple regression 분석시 년도별 모바일 게임 매출액에 대한 regression 분석을 하면서 이미 이 방법을 이용해 보았습니다. Regression 분석의 특징은 time series 에 가장 적합한 수식을 찾는 즉 trend 를 찾아 주어 미래에 값의 예측하는 방법으로 trend projection 이라고 합니다.

Smoothing 방법와 같이 random 요소를 제거하는 목적으로 trend 와 seasonal 요소로 발생으로 예측을 제대로 하지 못할 경우 더 좋은 결과를 만들어 줍니다. 주의 사항은 먼 미래일 수록 예측된 값의 정확도는 떨어지게 됩니다.

Regression 을 위한 방법은 동일하기 때문에 생략을 하고 MAD 값 계산은 다음과 같습니다.

$$MAD = E(|e|) = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{15.3}$$

Residue 의 평균값이 MAD 가 됩니다.



$$\hat{y} = 86.7 + 1.01x$$

	Coefficient	SE	T statistic	p-value
Intercept	86.7	10.7	8.1	1.91e-9
월(x)	10.1	0.5	2	0.0532

MAD 값은 25.036 로 SMA 분석시 5 개로 평균으로 처리한 결과의 MAD 값 24.23 보다는 큰 값으로 나타내지만, 다른 분석 방법보다는 성능이 좋게 나타났습니다.

Linear regression 을 사용하기 위한 전제 조건은 residue 값들의 서로 독립적어야 합니다. 이를 위해서 autocorrelation 값을 측정을 하면 되는데 즉 자기자신의 값들에 대한 correlation 을 측정하므로써 서로가 independent 인지 아닌지를 알 수 있습니다. 이를 위해 사용되는 statistic 으로 Durbin-Watson statistic 을 이용하면 됩니다.

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2} \tag{15.4}$$

e: residue 값

d 값	해석
0	완벽한 positive autocorrelation 이 존재
2	autocorrelation 은 없음
4	negative autocorrelation 이 존재

예를 들어 positive autocorrelation 으 존재는 전과 다음값에 차이가 일정하게 나타나 10, 10, 10 과 같이 되면 수식 15.4 의 분자가 0 이 되어 d 값이 0 이 됩니다. 반대로 negative autocorrelation 일 경우 10,-10,10,-10 과 같이 나타날 경우 3 이 됩니다.

이 수식을 사용하기 위한 조건으로 sample 크기는 15 개 이상이어야 하고 Durbin-Watson 검사의 critical value 에서 나타나는 값에 따라서 결정이 됩니다. 일반적으로 negative autocorrelation 은 잘 발생하지 않기 때문에 positive autocorrelation 이 존재하는 가만을 검사합니다.

H_0: positive autocorrelation 존재하지 않는다.

H_1: positive autocorrelation 존재한다

Durbin-Watson 검사결과 나타나는 값은 d_L ~ d_U 로 나타나 수식 15.4 의 d 값이 d_L 보다 작으면 null hypothesis 는 reject 이되어 positive autocorrelation 이 존재한는 근거가 되고 그렇지 않고 d_U 보다 크면 존재하지 않는 것으로 판단하면 됩니다. 그런데 만약 값이 이 사이에 있으면 결론을 내리지는 못합니다.

신규가입 수강생의 예제에서 d 값은 1.9 로 critical value 가 1.41 에서 1.52 이므로 residue 에 positive autocorrelation 이 존재하지 않는 것으로 판단이 됩니다. 만일 그렇지 않다면 regression 분석 결과로 나온 coefficient 값에 신뢰성이 사라지고 이를 해결하기 위해 실제 값에 변화를 설명해주는 다른 독립변수를 추가하여야 합니다.

3. Seasonality 를 이용한 Forecast

Time series 가 trend, seasonal, cyclic, random 요소들로 구성되어 있어 있을 수 있습니다. 이들 구성을 분해하여 예측하는 방법에 대해서 알아 보겠습니다.

Time series 를 trend, seasonal, random 요소로 구분하여 곱셈으로 표현한 multiplicative decomposition 모델은 다음과 같습니다.

$$y_i = T_i \times S_t \times R_t \tag{15.5}$$

여기서 T 는 trend 요소, S 는 seasonal 요소, R 은 random 요소입니다. Cyclic 요소가 생략된 이유는 오랜 기간의 데이타를 갖고 있어야 하는데 여기서 짧은 기간내에 변화를 예측하기 때문에 포함되지 않았습니다. 이러한 방법을 활용하기 위한 절차는

1) Seasonal 요소를 찾아냅니다.
2) 찾은 seasonal 요소를 최초의 데이타로 부터 제거시킵니다.
3) Seasonal 요소를 제거한 데이타로 부터 trend 요소를 찾아냅니다.
4) 2,3 에서 찾은 seasonal 요소와 trend 요소를 갖고 예측을 합니다.

Seasonal 요소는 기간으로 구분된 데이타에 일정한 패턴을 말하는 것입니다. 신규 가입 학생의 경우 1 년 단위로 12 달씩 끊어서 seasonal 요소를 찾아야 합니다. 이 과정으로 최종적으로 얻을 정보는 각 달마다 seasonal 요소가 얼마나 크게 작용하는 것입니다.
다음은 이해를 쉽게하기 위해서 3 년을 4 분기로 나뉘었을 때 상황입니다.

11	12	13	14	21	22	23	24	31	32	33	34

예를 들어 11 은 1 년차 1 분기, 34 는 3 년차 4 분기 를 말합니다.
이를 1 년 단위로 4 개씩 끊어어서 3x4 의 테이블에 평균값 만들게 되면 다음과 같이 채웁니다.

12.5	11	12	13	14	21	22	23	24	31	32	33	34

13.5	11	12	13	14	21	22	23	24	31	32	33	34

14.5	11	12	13	14	21	22	23	24	31	32	33	34

21.5	11	12	13	14	21	22	23	24	31	32	33	34

…

32.5	11	12	13	14	21	22	23	24	31	32	33	34

4 개씩 평균을 구하기 때문에 평균값의 위치는 두번째와 세번재 사이가 됩니다. 하지만 필요한 정보는 13 의 위치와 14 의 위치에 평균이기 때문에 평균을 얻는 과정을 수정을 해야 합니다.

13	11	12	13	14	21	22	23	24	31	32	33	34
	11	12	13	14	21	22	23	24	31	32	33	34

14	11	12	13	14	21	22	23	24	31	32	33	34
	11	12	13	14	21	22	23	24	31	32	33	34

…

32	11	12	13	14	21	22	23	24	31	32	33	34
	11	12	13	14	21	22	23	24	31	32	33	34

하지만 만약에 3 개씩 얻는다면 첫번째 평균만으로도 충분합니다.

12	11	12	13	14	21	22	23	24	31	32	33	34

13	11	12	13	14	21	22	23	24	31	32	33	34

14	11	12	13	14	21	22	23	24	31	32	33	34
21	11	12	13	14	21	22	23	24	31	32	33	34
	...											
33	11	12	13	14	21	22	23	24	31	32	33	34

이렇게 얻은 평균 값은 centered moving average(CMA)라고 합니다. 그럼 분기별 3 년치 데이타의 CMA 값들은 다음과 같은 table 로 형성이 됩니다.

		CMA_{13}	CMA_{14}
CMA_{21}	CMA_{22}	CMA_{23}	CMA_{24}
CMA_{31}	CMA_{32}		

평균때문에 random 요소는 제거가 되고 중앙에서의 4 분기값의 평균을 갖고 있기 때문에 seasonal 요소도 제거가 됩니다. 즉 CMA 값은 수식 15.5 에서 R, S 요소가 제거 되기 때문에 T 에 해당합니다.

$$RMA_i = S_i \times R_i = \frac{y_i}{T_i} = \frac{y_i}{CMA_i} \tag{15.6}$$

이 값을 ratio-to-moving average 라고 하고 이 값은 원래 데이타에서 seasonal 과 random 요소를 나타내는 값입니다. 이 값에서 seasonal 요소만을 추출하기 위해서 RMA 를 분기별로 평균을 한 seasonal factor(SF)값을 얻어 random 요소를 제거하고 이값을 normalization 을 한 normalized seasonal factor(NSF)를 얻게 되면 데이타에서 seasonal 요소(S)를 이해 할 수 있습니다.

	RMA_{21}	RMA_{22}	RMA_{13}	RMA_{14}
	RMA_{31}	RMA_{32}	RMA_{23}	RMA_{24}

SF	$(RMA_{21}+ RMA_{31})/2$	$(RMA_{22}+ RMA_{32})/2$	$(RMA_{13}+ RMA_{23})/2$	$(RMA_{14}+ RMA_{24})/2$

만약 분기별 데이타가 6 년치가 있다면 위의 각 SF 에 해당하는 CMA 는 5 개씩 있습니다.

다음 NSF 값들은

$$NSF_i = 4 \times \frac{SF_i}{\sum SF}$$

여기서 4 를 사용한 이유는 4 분기로 나뉘었기 때문입니다. 만일 월 단위로 seasonal 요소를 제거하려면 이 값은 12 가 됩니다.

이제 학습한 내용을 신규가입 수강생 예제에 적용하도록 하겠습니다. NSF 를 신규가입 학원생수 예제에 적용을 하기 위해서는 데이타가 12 개월씩 3 년치 데이타가 있고 분기별이 아닌 12 개월 단위로 일정한 특징이 있기 때문에 CMA 계산 법은 다음과 같아 합니다.

12 개월로 하기 때문에 1 ~ 12 월 달에 중간은 6.5 가 됩니다.

1	2	3	4	5	6	7	8	9	10	11	12

우리가 필요한 CMA 값들은 다음과 같습니다.

1 월	2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월	12 월
						107	108	109	110	111	112
201	202	203	204	205	206	207	208	209	210	211	212
301	302	303	304	305	306						

107 은 1 년차 7 월달이라는 의미로 이 값은 다음과 같은 기간 동안에 평균값으로 얻을 수 있습니다.

1	2	3	4	5	6	7	8	9	10	11	12	
	2	3	4	5	6	7	8	9	10	11	12	13

이렇게 해서 얻은 CMA 값들은 다음과 같습니다.

1 월	2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월	12 월

						86.75	87.75	87.9583	88.375	90.125	92.75
94	94.375	96.167	97.75	100	103.0417	104.7083	105.417	106.2916	107.083	108.25	110
113.083	116.33	117.875	119	121.458	124.208	306					

예를 들어 107 위치에 있는 값은

$$\frac{138 + 2 \times (105 + 123 + 71 + 52 + 130 + 93 + 72 + 81 + 67 + 53 + 52) + 146}{24} = \frac{2082}{24} = 86.75$$

다음 RMA 값들은

1 월	2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월	12 월
0	0	0	0	0	0	1.072	0.8205	0.9209	0.7581	0.5881	0.5606
1.5532	1.2821	1.1646	0.9412	0.73	1.6692	0.7736	0.8822	0.969	0.7751	0.8406	0.7909
1.3353	1.1433	1.0265	0.8571	0.7492	1.578	0	0	0	0	0	0

1 년차 7 월의 RMA 계산은 다음과 같습니다.

$$RMA_{107} = \frac{93}{86.75} = 1.072$$

이 값을 통해서 random 요소를 제거한 12 개의 SF 값은 다음과 같습니다.

	1 월	2 월	3 월	4 월	5 워	6 월
SF	(201+301)/2	(202+302)/2	(203+303)/2	(204+304)/2	(205+305)/2	(206+306)/2
	1.4442	1.2127	1.0956	0.8992	0.7396	1.6236
	7 월	8 월	9 월	10 월	11 월	12 월
SF	(107+207)/2	(108+208)/2	(109+209)/2	(110+210)/2	(111+211)/2	(112+212)/2
	0.9228	0.8514	0.945	0.7666	0.7144	0.6758

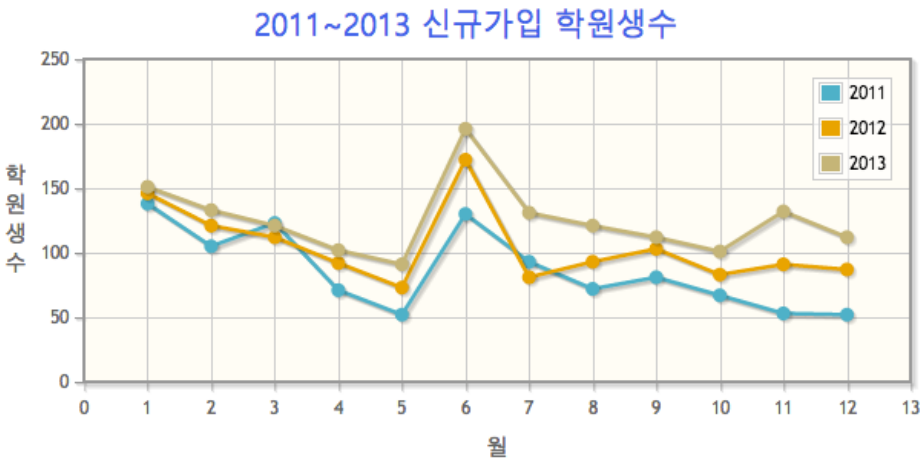
여기서 201 은 2 년차 1 월 RMA 값입니다.

예를 들어 1 월의 SF 값은

$$\frac{1.5532 + 1.3353}{2} = 1.4442$$

이렇게 해서 얻은 NSF 값은 다음과 같습니다.

월	1 월	2 월	3 월	4 월	5 월	6 월
NSF	1.4575	1.2238	1.1056	0.9074	0.7464	1.6385
월	7 월	8 월	9 월	10 월	11 월	12 월
NSF	0.9313	0.8592	0.9536	0.7737	0.7209	0.682



년도별 신규가입 학생수를 보면 6 월이 다른 달에 비해서 상대적으로 높게 나타나고 12 월이 적게 나타나는데 NSF 값을 보시면 이러한 패턴과 일치함을 확일 할 수 있습니다.

이제 S 를 얻었기 때문에 seasonal 요소를 제거한 trend 요소를 알아 보도록 하겠습니다. 다시 수식 15.5 를 통해서 얻을 수 있습니다.

$$T_i \times R_i = \frac{y_i}{S_i} = \frac{y_i}{NSF_i}$$

이렇게 얻은 값을 regression 분석으로 random 요소를 제거한 trend 요소인 T 값을 얻게 됩니다.

$$T_i = 72.243 + 1.8686x$$

	Coefficient	SE	T statistic	p-value
Intercept	72.243	5.055	14.29	6.66e-16
월	1.8686	0.238	7.842	3.954e-9

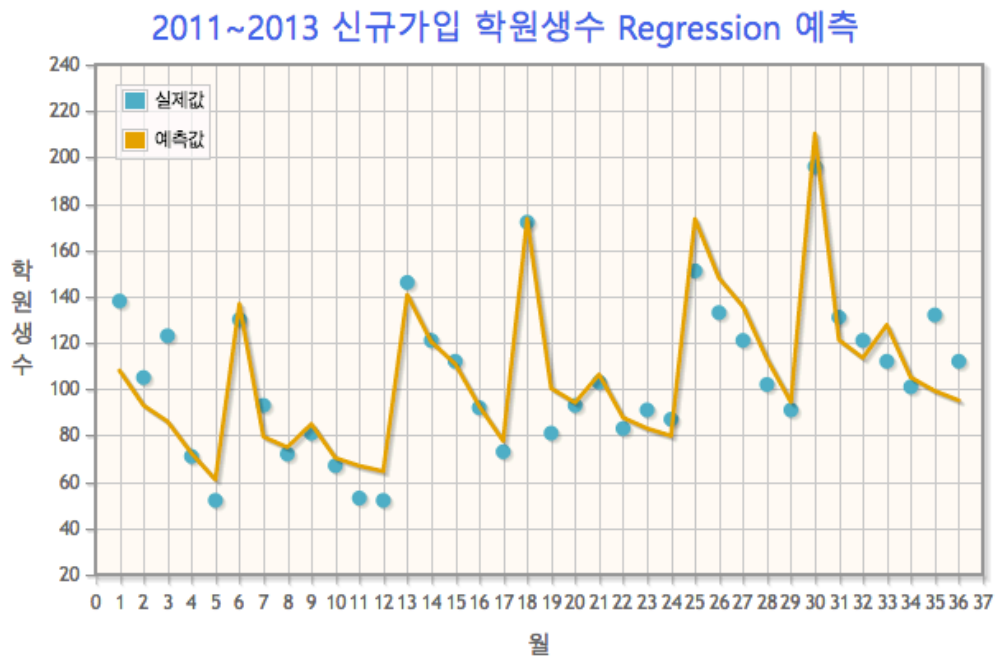
p-value 를 보면 월의 coefficient 는 통계적으로 중요한 값임을 알 수 있습니다. 그리고 residue 들이 positive autocorrelation 이 있는가를 위해 Durbin-Watson statistic 값이 1.263 으로 critical value 가 1.41 에서 1.52 이므로 residue 에 positive autocorrelation 이 존재하는 것는 것으로 나타납니다.

마지막으로 예상값을 구하기 위해서 multiplicative decomposition 방법을 이용하여 다음의 수식을 이용하면 됩니다.

$$F_i = T_i \times S_i \tag{15.7}$$

	2011			2012			2013		
	T_i	S_i	F_i	T_i	S_i	F_i	T_i	S_i	F_i
1 월	74.11	1.4442	108.02	96.53	1.4442	140.70	118.96	1.4442	173.38
2 월	75.98	1.2127	92.99	98.40	1.2127	120.43	120.83	1.2127	147.87
3 월	77.85	1.0956	86.07	100.27	1.0956	110.86	122.69	1.0956	135.66
4 월	79.72	0.8992	72.34	102.14	0.8992	92.68	124.56	0.8992	113.03
5 월	81.59	0.7396	60.90	104.01	0.7396	77.63	126.43	0.7396	94.37
6 월	83.45	1.6236	136.74	105.88	1.6236	173.48	128.30	1.6236	210.22
7 월	85.32	0.9228	79.46	107.75	0.9228	100.34	130.17	0.9228	121.22
8 월	87.19	0.8514	74.91	109.61	0.8514	94.18	132.04	0.8514	113.44
9 월	89.06	0.945	84.93	111.48	0.945	106.31	133.91	0.945	127.70
10 월	90.93	0.7666	70.35	113.35	0.7666	87.70	135.77	0.7666	105.04
11 월	92.80	0.7144	66.90	115.22	0.7144	83.06	137.64	0.7144	99.23
12 월	94.67	0.6758	64.56	117.09	0.6758	79.85	139.51	0.6758	95.14

이 수식을 통해서 계산을 할 경우 주의 할 점은 먼 미래의 값을 얻을 수록 오류는 커지게 됩니다.



MAD 값을 특정해 보면 10.298 로 즉 평균 오류가 지금까지 학습한 예측방법들 중에서 제일 낮습니다. 하지만 무조건 복잡한 방법이 계산이 평균 오류를 낮게 하지는 못하는 것을 명심해야 합니다.

```
jMath.prototype.forecast('season', p)
```

여기서 p 는 seasonal 요소의 개수로 이 예제에서는 12 입니다.

```
var ts = [
     jMath([[1,138],[2,105],[3,123],[4,71],[5,52],[6,130]
           ,[7,93],[8,72],[9,81],[10,67],[11,53],[12,52]]),
     jMath([[1,146],[2,121],[3,112],[4,92],[5,73],[6,172]
           ,[7,81],[8,93],[9,103],[10,83],[11,91],[12,87]]),
     jMath([[1,151],[2,133],[3,121],[4,102],[5,91],[6,196]
          ,[7,131],[8,121],[9,112],[10,101],[11,132],[12,112]])
];
var list = jMath.joinByRow(ts);
for ( var i = 0 ; i < list.rows ; i++ )
{
    list[i][0] = i+1;
}
var fcast = list.forecast('season',12);
console.log(fcast.mad, fcast.durbinWatson);

10.298323399922936, 1.2633285760323967
```

마지막으로 seasonal 요소를 regression 분석에서 dummy 변수로 넣어서 best subset 기법으로 regression model 을 만들고 예측된 값을 분석하는 방법을 소개하겠습니다.

https://github.com/handuck/jMath

지금까지 다룬 데이타는 모두 x 로 월또는 분기와 같은 주기에 있는 값들로 이 주기에 값들을 dummy 변수로 추가 합니다. 예를 들어 분기 데이타의 경우 다음과 같습니다.

분기	Q1	Q2	Q3	실제값
1	0	0	0	Y_1
2	1	0	0	Y_2
3	0	1	0	Y_3
4	0	0	1	Y_4
5	0	0	0	Y_5
…	…	…	…	…
12	0	0	1	Y_{12}

1 년을 주기로 4 분기로 나누기 때문에 dummy 변수의 개수 4-1 개인 3 으로 1 분기는 모두 0, 2 분기는 Q1 만 1, 3 분기는 Q2 만 1, 4 분기는 Q3 만 1 로 하여 데이타를 변형을 합니다. 이렇게 추가된 독립변수 중 필요없는 것은 제거를 하기 위해서 regression modeling 을 사용하면 됩니다. 이 경우 독립변수 개수가 4 개이므로 best subset 방식으로 하면 됩니다. 만일 독립 변수개수가 5 개 이상이면 general stepwise 방식을 사용하면 됩니다. 이렇게 계산된 값을 갖고 MAD 를 측정해 보면 다른 예측 방법과 성능을 비교할 수 있습니다.

예를 들어 Apple 사의 2011~2013 년 동안 분기별 매출액 정보로 부터 예측을 하기 위해서 다음과 jMath 에서 다음과 같이 합니다.

```
jMath.prototype.forecast('dummpy', p)
```

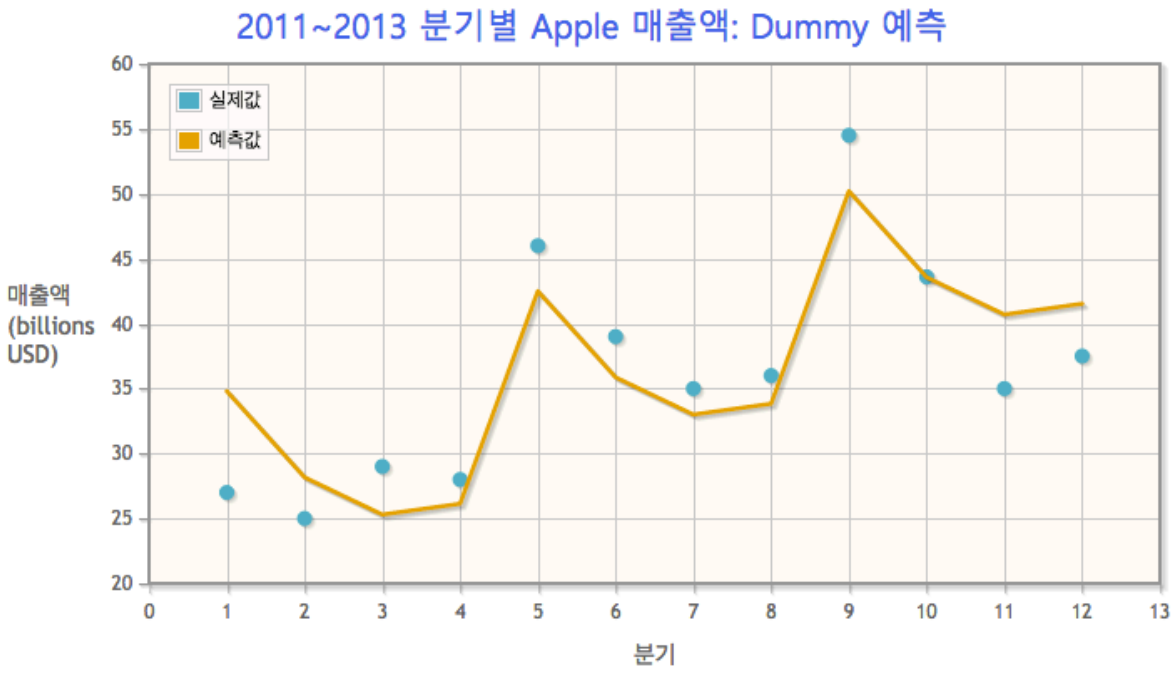
여기서 p 는 seasonal 요소의 개수로 이 예제에서는 4 입니다.

```
var list = jMath([ [1,27], [2,25], [3,29], [4,28], [5,46], [6,39],
                   [7,35], [8,36], [9,54.5], [10,43.6], [11,35], [12,37.5]]);
result = list.forecast('dummy', 4);
console.log(result.mad);

3.45
```

여기서 데이타 값 [x,y]에서 x 는 분기이고 y 는 매출액입니다.

$$F_t = 32.875 + 1.925t - 8.5583Q_1 - 13.35Q_2 - 14.44167Q_3$$


2011~2013 분기별 Apple 매출액: Dummy 예측
실제값
예측값
매출액 (billions USD)
분기
60
55
50
45
40
35
30
25
20
0
1
2
3
4
5
6
7
8
9
10
11
12
13