

SAS JMP를 이용한 S형 소프트웨어 신뢰도 성장모델에서의 모수 추정에 관한 연구

문숙경

목원대학교 응용통계학과

A study on the parameter estimation of S- Shaped Software Reliability Growth Models Using SAS JMP

Moon sook-kyung

Dept. of Applied Statistics, Mokwon University

Abstract

Studies present a guide to parameter estimation of software reliability models using SAS JMP. In this paper, we consider only software reliability growth model(SRGM), where mean value function has a S-shaped growth curve, such as Yamada et al. model, and ohba inflection model. Besides these stochastic SRGM, deterministic SRGM's, by fitting Logistic and Gompertz growth curve, have been widely used to estimate the error content of software systems. Introductions or guide lines of JMP are concerned. Estimation of parameters of Yamada et al. model and Logistic model is accomplished by using JMP. The differences between Yamada et al. model and Logistic model is discussed, along with the variability in the estimates or error sum of squares. This paper have shown that JMP can be an effective tool in these research.

1. 서론

최근 사회 각 분야에서 시스템의 기능과 구조가 날로 복잡해지고 다기능화 되어짐에 따라 이를 제어하는 소프트웨어의 역할이 날로 증대되고 있는 실정이다. 흔히 신뢰성 연구라 하면 국내에선 대체로 하드웨어에 관한 연구들이 대부분이고, 소프트웨어에 대한 연구는 그다지 활발한 편이 아니다. 미국, 일본을 중심으로 한 외국에서는 지난 20여년 동안 많은 연구논문들이 제안되었었고 최근까지도 실제적인 연구결과들이 많이 발표되어지고 있다.

일반적으로 소프트웨어는 몇 단계의 개발과정 즉, 계획, 설계, 코딩(coding) 및 시험 단계를 거쳐 최종시스템의 방출(release)이 이루어지는데, 에러들을 찾아내고 제거시키는 주된 작업을 수행하는 시험단계는 전체 개발비용의 50%이상을 차지한다. 이처럼 개발 과정 중 큰 비중을 차지하는 시험단계에서는 주로 소프트웨어에 잔존하는 에러를 찾아 수정시키는 일을 한다. 시험을 많이 할수록 개발과정중 소프트웨어에 잔존하는 에러들이 많이 걸러내게 되어지나 경제적, 시간적 제약이 있으므로 어느 시점에서 중단하고 시스템을 방출하여야 한다. 이 때 만약 시험기간 중 미처 발견되지 못한 에러들은 시스템 운용중에 고장으로 발생됨으로 보전비용(maintenance cost) 증가, 회사의 명예 실추 및 나아가 치명적인 인명 피해를 가져다 줄수 있다.

이러한 시험기간 중 발견된 에러를 수정할 시 새로운 에러를 발생하지 않는다고 하면 시험 시간이 증가함에 따라 발견되는 에러들은 점점 많아질 뿐 아니라, 시스템 신뢰도도 증가되므로 이런 상황에 적용할 수 있는 신뢰도 모델들을 신뢰도 성장 모델(SRGM)이라 한다. 시험기간 중 특정시점 t 시간까지 발견되어진 누적에러수 ($N(t)$)가 nonhomogenous Poisson process(NHPP)를 따른다는 가정하에 평균값 함수(mean value function or cumulative failure rate)가 취하는 형태에 따라 Musa exponential 모델[2], Goel and Okumoto 모델[1], Yamada et al.[6] 및 Ohby[8] 모델 등을 들 수 있으며 이중 전자의 2개 모델은 지수형을 후자에 속하는 2개의 모델은 S형을 나타낸다. 그 밖에도 비선형 성장 곡선으로 S형을 나타내는 Logistic 모델 및 Gompertz 모델등도 열거할 수 있겠다. 그런데 이러한 모델들을 사용할 경우 가장 큰 문제점은 각 모델에 사용된 모수(parameters)를 추정하여야 한다는 것이다.

본고에서는 실제 적용력이 뛰어난 S형 신뢰도 성장 모델 중에서 Yamada et al.과 Logistic모델을 사용하여 실제 주어진 시험데이터를 가지고 SAS JMP를 이용하여 간편하게 모수를 추정할 수 있는 방법을 소개하려 한다. 통계전문가용인 SAS package와 달리 SAS JMP는 학생들과 통계전문가가 아닌 비통계인들에게 통계분석을 용이하게 도와주는 도구인데 맥킨토시버전과 window버전 두 종류가 있으며, 디스켓 4장 분량의 소규모 패키지나, 본고에서 사용할 'Nonlinear Fit'나 'Survival Analysis'등과 같은 내용은 고급단계에 속하는 내용이다.

본고에서는 2절에 우선 S형 신뢰도 성장 모델 및 모수추정에 대하여 3절에 SAS JMP 소개 및 사용법을 제시하였고 끝으로 실제 자료를 갖고 SAS JMP를 이용하여 모델에서의 모수추정을 실시하였다.

2. S형 신뢰도 성장 모델

2.1 모델소개

신뢰도 성장 모델 중에서도 NHPP 모형을 사용하는 것과 비선형 성장곡선, 즉 Gompertz, Logistic 모델 등을 사용하는 방법도 있다. 전자의 경우, t 시간까지 발생된 누적에러 수 ($N(t), t > 0$)가 NHPP를 따른다는 가정을 하는 것으로서, t 시간까지 발생되리라는 누적에러수의 기대치 즉, 평균값 함수 ($M(t) = E[N(t)]$)가 취하는 형태에 따라 각기 다른 모델로 불려진다. 예컨대, 평균값 함수가 지수형태를 취하면 지수형, S형태를 취하면 S형이라 불려지는데, S형에 속하는 대표적 모델이 Yamada et al.이 제안한 delayed S형 모델과 ohba가 제안한 inflection S형 신뢰도 성장모델이며 이들의 평균값 함수($M(t)$)는 각각 다음과 같다.

$$M(t) = a[1 - (1 + bt) \exp(-bt)], b > 0,$$

, 여기에서 a 는 총 에러수를 b 는 에러발생률을 나타냄.

$$M(t) = a[1 - \exp(-bt)]/[1 - c \cdot \exp(-bt)], b > 0, c > 0$$

, 여기에서 a 는 총에러수를, b 는 에러발생률을 그리고 c 는 inflection 계수를 나타냄.

위의 경우처럼 stochastic 모델을 사용하는 것과는 달리 Logistic, Gompertz와 같은 비선형 성장곡선을 소프트웨어내에 잔존하는 에러수를 예측하는데 많이 쓰여지고 있다. 실제로 일본의 소프트웨어 하우스에서는 이들 모형을 사용하여 에러수를 예측하는 경우가 많다.[7] 이들 모형들은 원래 수요 예측이나 경제성장 혹은 미래 인구수 예측을 위해 고안되었던 모형들이었으며, 이들 모형은 각각 다음과 같다.

$$\text{Logistic 모형} : Y_t = a/[1 + b \exp(ct)] + \epsilon_t$$

위 모형은 펄 성장곡선 모형(Pearl Growth Curve Model) 이라고 한다.

$$\text{Gompertz 모형} : Y_t = a \cdot \exp[-b \exp(-ct)] + \epsilon_t$$

위의 두 모형에서 Y_t 는 t 시간까지 발생된 에러수를, a, b, c 는 추정하여야할 모수(parameters)를 $\epsilon_t \sim i.i.d. N(0, \sigma^2)$ 을 가정한다.

2.2 모수추정

NHPP를 이용한 신뢰도 성장모델에서의 모수 추정은 대체로 최우추정법이 사용되어지며, Yamada et al. 모델일 경우 구하는 절차는 다음과 같다. 흔히 t_i 시간까지 발

견된 누적에러수를 y_i 라 할 경우, n 개의 데이터 즉, $(t_k, y_k), k=1, 2, \dots, n$ 가 주어졌다는 가정하에 우도함수 $L(a, b)$ 는 다음 (1)식 같이 표현 될 수 있다.

$$L(a, b) = \prod_{k=1}^n \frac{\{M(t_k) - M(t_{k-1})\}^{y_k - y_{k-1}}}{(y_k - y_{k-1})} \cdot \exp[-\{M(t_k) - M(t_{k-1})\}] \quad (1)$$

, 여기서 $t_0 = 0, y_0 = 0$

(1)식에 로그를 취하면 아래(2)식을 얻는다.

$$\begin{aligned} \ln L(a, b) &= \sum_k (y_k - y_{k-1}) \ln a \\ &+ \sum_k (y_k - y_{k-1}) \ln [(1 + bt_{k-1})e^{-bt_{k-1}} - (1 + bt_k)e^{-bt_k}] \\ &- \sum_k \ln [(y_k - y_{k-1})!] - a(1 - (1 + bt_n)e^{-bt_n}) \end{aligned} \quad (2)$$

(2)식을 각각 모수 a, b 에 대해 편미분하면 (3),(4)식을 얻을 수 있다.

$$y_n = a[1 - (1 + bt_n)e^{-bt_n}] \quad (3)$$

$$at_n^2 e^{-bt_n} = \sum_k \frac{(y_k - y_{k-1})(t_k^2 e^{-bt_k} - t_{k-1}^2 e^{-bt_{k-1}})}{[(1 - bt_{k-1})e^{-bt_{k-1}} - (1 - bt_k)e^{-bt_k}]} \quad (4)$$

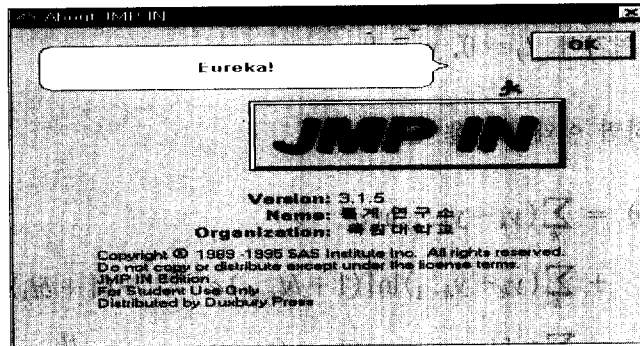
(3), (4)에서 a, b 에 대한 근을 구할 시 복잡한 수치해석적 방법이 사용되어진다. 그러므로 본 고에서는 이들 모형들도 Gompertz같은 비 선형 회귀모형(Nonlinear regression)으로 간주하고 최소 자승법(least squares method)에 의해 모수추정을 하려한다. 이를 위해 SASJMP의 Nonlinear Fit 분석도구를 이용할 것이다.

3. SAS JMP 소개 및 사용법

3.1 SAS JMP소개

SAS JMP는 SAS사가 1980년 말경에 Macintosh용으로 처음 개발된 것으로 95년 IBM용 JMP가 release되었고, 현재는 Version 3.1이 사용되어지고 있으며, 본 논문에서도 이 버전으로 실행함을 밝힌다. SAS와 달리 사용기간 제한이 없으며, 패키지 가

격도 저렴하며 SAS처럼 복잡한 명령어를 사용하지 않고 모든 것이 메뉴로 구성되어 명령어를 몰라도 사용할 수 있는 편리함과 함께 통계량과 그래픽을 동시에 출력시켜 주는 큰 장점들이 있다. SASJMP는 디스켓 4장으로 되어있어 첫 번째 디스켓을 넣고 installation.exe를 실행시키면 SASJMP가 설치되며, 프로그램 install을 끝내고 시작메뉴를 클릭하면 JMP가 시작되며 다음 <그림 1>과 같은 화면이 펼쳐진다.



< 그림 1 > SASJMP 시작 화면

3.2 데이터 입력

Excel과 같은 spreadsheet로 되어있어 자료입력이 간편한데, 최신 출시된 버전에는 SAS데이터 (*.sd2)를 읽어올 수 있으나 논문에서 사용한 버전으로는 안 된다. 열을 두 번 클릭하면 새로운 데이터열을 만들 수 있으며 변수명을 줄 수 있고 기존 데이터 열을 변형할 수 있는 향이 열린다. 참고로 ../data directory에 있는 예제자료 중 animals.jmp를 불러오면 <그림 2>와 같다.

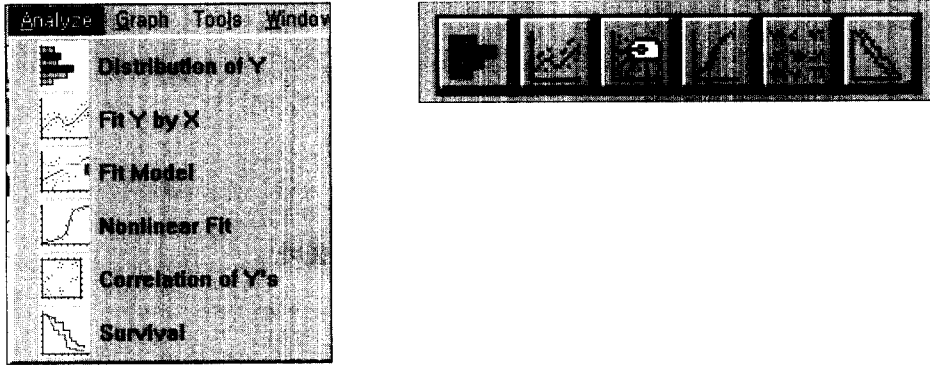
Animals		4 Cols			
		<input checked="" type="checkbox"/> N	<input type="checkbox"/>	<input checked="" type="checkbox"/> C	<input type="checkbox"/> N
24 Rows		species	subject	miles	season
1	FOX		1	0	fall
2	FOX		1	0	winter
3	FOX		1	5	spring

< 그림 2 > animals.jmp

3.3 분석방법

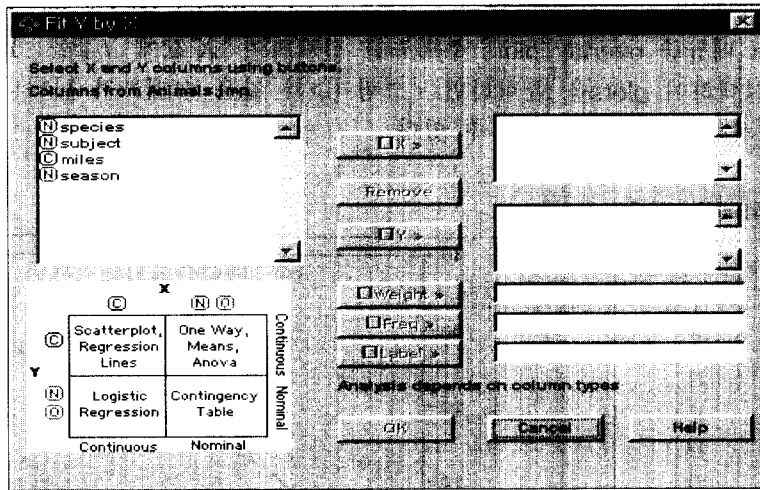
SAS JMP에서 제공해 주는 분석방법들(시작메뉴의 Analyze참조)은 Univariate, Contingency Table, Paired T-test, Regression Analysis, ANOVA, Principal Component, Logistic Regression, Time Series, Survival Analysis, Quality Control, Nonlinear Regression 등이 있다.

메뉴바에 있는 Analyze를 누르면 <그림 3>과 같은 부메뉴가 펼쳐지며 툴바에 있는 그림들을 클릭해도 된다.



< 그림 3 > Analye 부메뉴 및 툴바

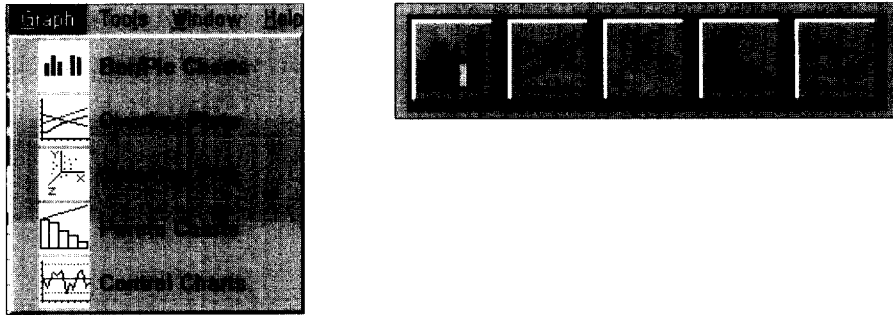
참고로 부메뉴중 Fit Y by X를 선택하면 <그림 4>와 같이 나타나는데, 창의 왼쪽 아래 화면에는 변수의 속성에 따라 분석방법도 표시되어져 있다.



< 그림 4 > Fit Y by X 화면

3.4 통계 그래픽스

메뉴바의 Graph 메뉴를 선택하여 <그림 5>와 같은 부메뉴가 나타난다. 부메뉴에 표시된 그래프들을 모두 제공한다.



< 그림 5 > 그래프 부메뉴 및 툴바

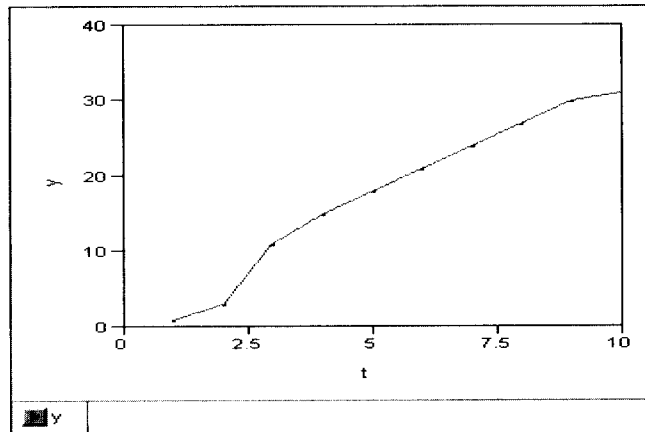
4. 적용예

4.1 적용데이터 소개

본 절에서 사용되는 시험 데이터는 (<표 1>참조) 일본에 상주하는 IBM 품질인증 실험실에서 베이직, 어셈블리 언어로 된 약 3000라인 정도 분량의 매크로 프로그램으로써 10일단의 기능 실험을 실시하였을 때 얻어진 자료인데 약 31개의 에러가 발견되어진다. 우선 이 데이터가 취하는 형태가 S형인지를 확인하는 절차가 꼭 필요하다. 이를 위해 Graph 메뉴중 overlay plots 부메뉴를 선택하여 X 값에 t 값을 Y 값에 에러 누적 개수를 선택하여 plots을 클릭하면 <그림 6>과 같은 그래프가 나타나며 그림에서 보듯이 S형을 나타내는 것을 볼 수 있다.

< 표 1 > 누적에러 개수

Time(t)	누적에러 갯수
1	1
2	3
3	11
4	15
5	18
6	21
7	24
8	27
9	30
10	31



< 그림 6 > 누적에러 개수 그래프

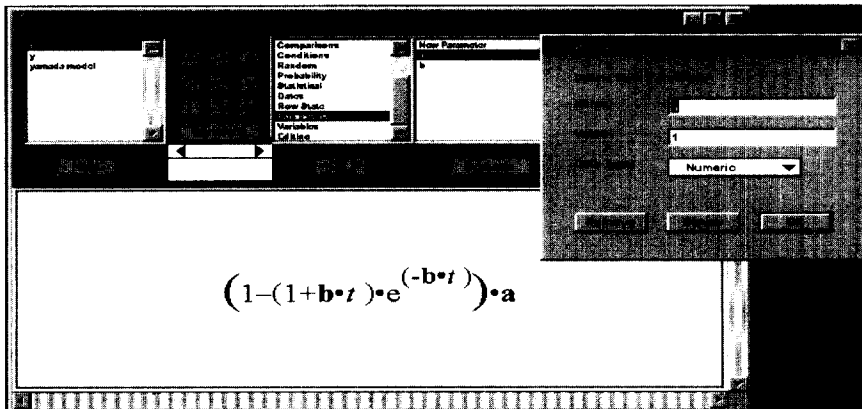
4.2 SAS JMP에서 비선형회귀(Nonlinear Regression)분석 절차

4.2.1 데이터 생성

데이터 파일의 첫 번째 열(Column)에 <표 1>의 시간 t 를, 2번째열에 누적에러 개수 y 를 각각 입력시키고 난 후 모형식(formula in SAS JMP)을 지닌 3번째 열을 만든다. 예컨대, Yamada et al. 모형인 경우는 $a/[1-(1+bt) \exp(-bt)]$ 를 Logistic 모형인 경우는 $a/[1+b \exp(ct)]$ 식을 계산기(calculator in SAS JMP)를(<그림 7-2> 참조) 이용하여 입력시킨다. 단, 이때 추정하여야 할 모수 a, b, c 등은 계산기창 우편의 Parameter function에서 New Parameter을 클릭하여 미리 만들어 놓아야 하며 모형식에 입력 시킬 때 볼드체(bold type)로 표시되어진다.

Untitled 1					
4 Cols		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10 Rows		t	y	yamada model	logistic model
1	1	1	1	1.595837	3.235388
2	2	2	3	5.180765	5.47048
3	3	3	11	9.521411	8.764499
4	4	4	15	13.91758	13.03446
5	5	5	18	18.00081	17.73514
6	6	6	21	21.6038	22.06411
7	7	7	24	24.67689	25.44277
8	8	8	27	27.23546	27.75498
9	9	9	30	29.32749	29.19862
10	10	10	31	31.01424	30.04892

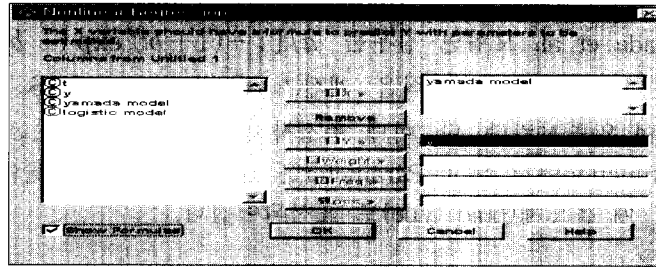
< 그림 7-1 > SASJMP 계산기 계산 결과



< 그림 7-2 > SASJMP 계산기

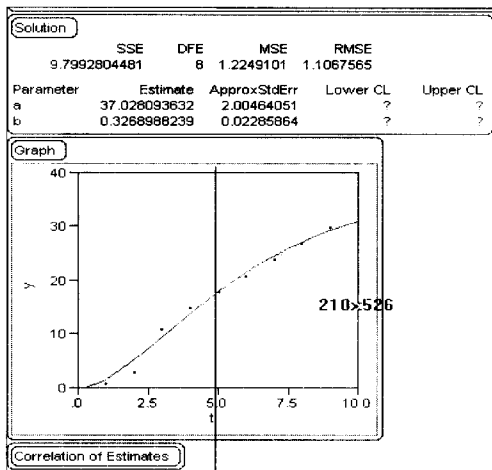
4.2.2 비선형 적합

모형식을 포함한 열을 만든후 Analyze 메뉴에서 Nonlinear Fit을 클릭하면 <그림 8>과 같은 대화창이 열린다.



< 그림 8 > 대화창

여기에서 새로만든 모형식을 포함한 열을 X로, 누적에러수를 Y로 선택하고 OK를 클릭하면, Nonline Fitting Control Panel 창이 열린다. 단, 이때 손실함수를 정의할 경우 Loss를 클릭하여 손실함수를 정의한다. 여기에서 Go를 클릭하면 최대반복횟수 (Iteration Limit) 60번 및 허용 오차(O, D, G, CL criterion) 중 한 조건이라도 만족하면 중지되거나 error sum of squares(SSE)가 크면 Reset을 클릭하고 다시한번 Go를 클릭하면 적합이 더욱 잘된다. 단, 이때 SSE값이 유난히 커지거나 Custom Loss Function을 정의했을 경우 옵션 중에 second Deriv.method를 선택하고 Go를 클릭하면 적합이 더욱 정교해지며 <그림 9-1>, <그림 9-2>와 같은 모수 추정값 및 그래프가 최종 출력된다.



Formula:

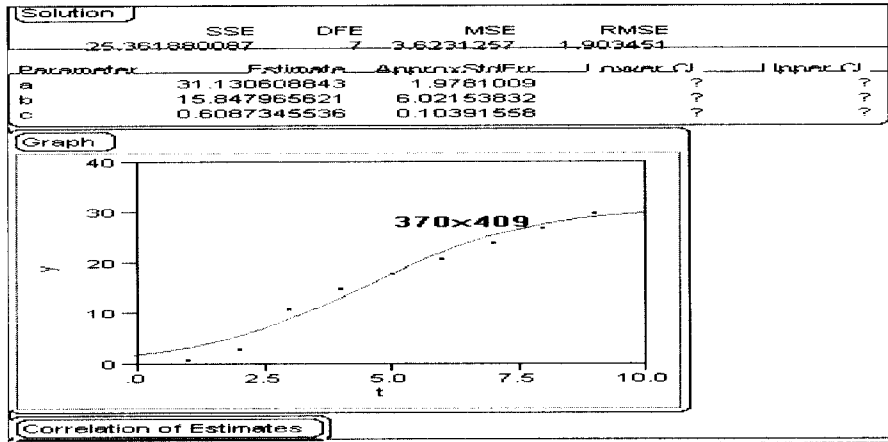
$$\left(1 - (1 + b \cdot t) \cdot e^{-b \cdot t}\right) \cdot a \quad \text{Model}$$

$$\left(1 - (1 + b \cdot t) \cdot e^{-b \cdot t}\right)^a$$

$$- \left((t \cdot e^{-b \cdot t}) - (1 + b \cdot t) \cdot t \cdot e^{-b \cdot t} \right)$$

< 그림 9-1 > 모수 추정값 및 그래프 출력 결과

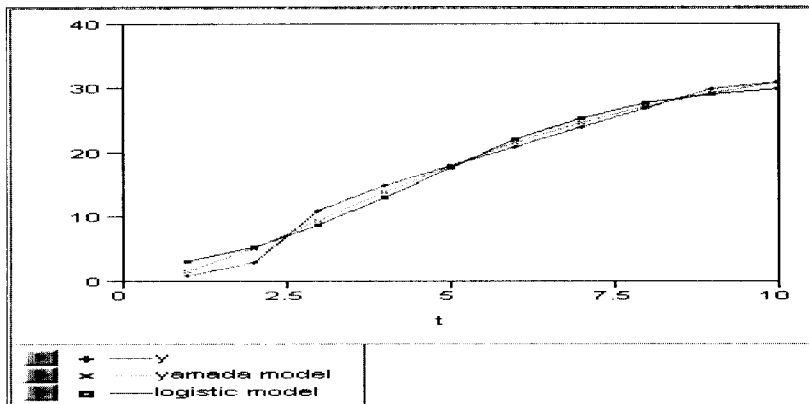
<그림 9-1>은 <표 1>의 데이터를 Yamada et al. 모델에 <그림 9-2>는 Logistic 모델에 각각 적합시킨 결과이다.



< 그림 9-2 > 모수 추정값 및 그래프 출력 결과

5. 토의 및 결론

S형 SRGM 모델 종류 및 기존 모수추정 방법을 그리고 SAS JMP에 대한 소개 및 사용법을 간략히 살펴보고 SAS JMP의 Nonlinear fit을 이용하여 실제 시험 데이터를 Yamada et al. 모델과 Logistic 모델에 적용시켜 보았다. <그림 10>은 이들 두 모델을 적합시킨 결과를 실제 누적데이터 (Y)와 함께 나타낸 것이다.



< 그림 10 > 누적갯수 실제 측정치와 모델 예측치 비교 그래프

그림에서 보듯이 Yamada et al. 모델이 Logistic 모델보다 더 적합이 잘 된 것 같으며, 실제 SSE 값을 비교해 보면 각각 9.8 25.4에서 알 수 있듯이 Yamada et al. 모델이 더 작다. 참고문헌[5]에서 같은 데이터에 의해 구한 a, b 의 MLE가 각각 37.9 0312로 나타났으며 본 논문에서 구한 값과 거의 같음을(<그림 9-1>참조) 알 수 있었다. 본 고에서는 JMP의 Nonlinea Fit 기능 중 가장 단순한 과장만 익혔지만 차후 custom loss function, weighted least squares 등 다양하고 powerful한 기능등을 익혀 여러 형태의 비 선형 곡선을 적합시킬수 있어야 할 것이다.

참고문헌

- [1] Goel A.L. and Okumoto K., "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans. Rel.* vol. R-28, pp. 206-211, Sep. 1979.
- [2] Musa J.D. "A thoery of software reliability and its application," *IEEE Trans. Software Eng.*, vol. SE-1, pp. 312-327, Sep. 1975.
- [3] John Sall and Ann Lehman, SAS Institute, "JMP start Statistics," *Duxbury Press.*
- [4] SAS Institute, "JMP User's Manual."
- [5] Yamada S., Ohba M., and Osaki S., "S-shaped reliability growth modeling for softwareerr or detection," *IEEE Trans. Rel.*, vol. R-32 pp. 475-484, Dec. 1983.
- [6] Yamada S., Ohba M., and Osaki S., "S-shaped software reliability growth models and their application," *IEEE Trans. Rel.*, vol. R-33 pp. 289-292, Oct. 1984.
- [7] Yamada S. and Osaki S., "Software Reliability Growth Modeling: Models and Applications," *IEEE Trans. on software engineering*, vol. SE-11, No. 12, Dec. 1985.
- [8] Lynch T., Pham H. and Way Kuo, "Modeling Software-Reliability with Multiple Failure-Types and Imperfect Debugging," 1994 PROCEEDINGS Annual RELIABILITY and MAINTAINABILITY Symposium, pp. 235-240.
- [9] Khoshgoftaar T.M., Bhattacharyya B.B. and Richardson Gary D., "Predicting Software Errors, During Development, Using Nonlinear Regression Models: A Comparative Study," *IEEE Trans. on Rel.*, vol. 41 No.3, Sep. 1992.