Nonlinear Regression

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

Nonlinear Regression

Introduction

- A mean function may not necessarily be a linear combination of terms. Some examples:

$$E(Y|X = x) = \theta_1 + \theta_2(1 - \exp(-\theta_3 x)) \tag{1}$$

$$E(Y|X = x) = \beta_0 + \beta_1 \psi_S(x, \lambda) \tag{2}$$

where $\psi_S(x, \lambda)$ is the scaled power transformation, defined as

$$\psi_S(X, \lambda) = \begin{cases} (X^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(X) & \text{if } \lambda = 0 \end{cases} \tag{3}$$

- Once λ has been chosen, the function becomes, in the sense we have been describing, linear in its terms, just as a quadratic in X can be viewed as linear in X and X^2.

Introduction

- Nonlinear mean functions often arise in practice when we have special information about the processes we are modeling.
- For example, consider again the function $E(Y|X = x) = \theta_1 + \theta_2(1 - \exp(-\theta_3 x))$. As X increases, assuming $\theta_3 > 0$, the function approaches $\theta_1 + \theta_2$. When $X = 0$, the function value is θ_1, representing the average growth with no supplementation. θ_3 is a rate parameter.

Estimation for Nonlinear Mean Functions

- Our general notational setup is straightforward. We say that

$$E(Y|X = \mathbf{x}) = \mathbf{m}(\mathbf{x}, \boldsymbol{\theta}) \tag{4}$$

  where $\mathbf{m}$ is a *kernel mean function*
- The variance function is

$$\text{Var}(Y|X = \mathbf{x}_i) = \sigma^2/w_i \tag{5}$$

  where the w_i are known positive weights, and σ^2 is an unknown positive number.

Estimation for Nonlinear Mean Functions

Iterative Estimation Technique

- Nonlinear regression is a complex topic.
- For example, the classic book by Seber and Wild (1989) is over 700 pages.
- We will discuss the Gauss-Newton algorithm without going into the mathematics in detail.
- On the next slide, I discuss how the algorithm works, as described by Weisberg.
- His account is somewhat abbreviated.

Estimation for Nonlinear Mean Functions

Iterative Estimation Technique

- We need to minimize

$$RSS(\boldsymbol{\theta}) = \sum_{i=1}^{n} w_i(y_i - \mathbf{m}(\mathbf{x}_i, \boldsymbol{\theta}))^2 \tag{6}$$

- This is done using *Gauss-Newton iteration*, with the following algorithm.
  1. Choose starting values $\boldsymbol{\theta}^{(0)}$ for $\boldsymbol{\theta}$, and compute $RSS(\boldsymbol{\theta}^{(0)})$.
  2. Set the iteration counter at $j = 0$.
  3. Compute $\mathbf{U}(\boldsymbol{\theta}^{(j)})$ and $\hat{\mathbf{e}}^{(j)}$ with ith element equal to $y_i - \mathbf{m}(\mathbf{x}_i, \boldsymbol{\theta}^{(j)})$.
  4. Compute the new estimate as

$$\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} + [\mathbf{U}(\boldsymbol{\theta}^{(j)})'\mathbf{W}\mathbf{U}(\boldsymbol{\theta}^{(j)})]^{-1}\mathbf{U}(\boldsymbol{\theta}^{(j)})'\mathbf{W}\hat{\mathbf{e}}^{(j)} \tag{7}$$

  5. Compute $RSS(\boldsymbol{\theta}^{(j+1)})$.
  6. If $RSS(\boldsymbol{\theta}^{(j)}) - RSS(\boldsymbol{\theta}^{(j+1)}) > tol_1$, and $j \leq itmax$ and $RSS(\boldsymbol{\theta}^{(j+1)}) > tol_2$, go to step 3, else stop.
- $\mathbf{U}(\boldsymbol{\theta})$ is a matrix of derivatives known as the *score matrix*. If $\boldsymbol{\theta}$ has k elements, then the $n \times k$ matrix $\mathbf{U}$ has element $u_{ij} = \partial \mathbf{m}(\mathbf{x}_i, \boldsymbol{\theta})/\partial\theta_j$ evaluated at the current estimate of $\boldsymbol{\theta}$.

Estimation for Nonlinear Mean Functions

Iterative Estimation Technique

- Generally, the Gauss-Newton algorithm will converge to a solution as long as you start reasonably close to the solution, and certain other problems do not occur.
- However, problems do occur, and Weisberg's account does not deal with them.
- In particular, in some cases, the *step* will be too long, and the function will not decrease because you have stepped over the point where the minimum occurs.
- You can tell this has happened, because the vector of derivatives of the function with respect to the values of θ will not be near zero, even though the function has increased.
- You have moved "in the right direction," but too far. A solution to this is called "backtracking." You simply multiply the step by a constant less than one, and try again with the reduced step that is going in the same direction, but not as far.
- Many unsophisticated algorithms use what is called "step-halving." Each time you backtrack, the initial step is multiplied by 1/2 and the iteration is retried. This keeps going on until the function is reduced, or a maximum number of step-halves has occurred.

Large Sample Inference

- Under certain regularity conditions, the final estimate $\hat{\boldsymbol{\theta}}$ will be approximately normally distributed,

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}^*, \sigma^2[\mathbf{U}(\boldsymbol{\theta}^*)'\mathbf{W}\mathbf{U}(\boldsymbol{\theta}^*)]^{-1}) \tag{8}$$

- We can obtain a consistent estimate of the covariance matrix of the estimates by substituting the estimates $\hat{\boldsymbol{\theta}}$ in place of the true minimizing values (that we would obtain if we had the population at hand instead of the sample) in the above formula. Thus,

$$\widehat{\text{Var}}(\hat{\boldsymbol{\theta}}) = \hat{\sigma}^2[\mathbf{U}(\hat{\boldsymbol{\theta}})'\mathbf{W}\mathbf{U}(\hat{\boldsymbol{\theta}})]^{-1} \tag{9}$$

where

$$\hat{\sigma}^2 = \frac{RSS(\hat{\boldsymbol{\theta}})}{n-k} \tag{10}$$

and k is the number of parameters estimated in the mean function.

Large Sample Inference

- Weisberg is careful to stress that, *in small samples, large-sample inferences may be inaccurate.*
- He then goes on to investigate some examples of nonlinear regression in practice, using data on turkey growth.
- Let's start with a little artificial example of our own.
- Suppose Y and X fit the model

$$E(Y|X = x) = X^2 - 1 \tag{11}$$

with $\text{Var}(Y|X = x) = \sigma^2$

- We can easily create some artificial data satisfying that model prescription.

An Artificial Example

- The following R code generates data fitting the model

```
> x <- 1:100/10
> y <- x^2 - 1 + rnorm(100,0,.25)
```

- Now, suppose we suspect that the model is of the form $E(Y|X = x) = x^{\theta_1} - \theta_2$, but we don't know the values for the θs.
- We can use `nls` as follows

```
> m1 <- nls(y~x^theta1 - theta2,start=list(theta1=.5,theta2=.5))
> summary(m1)

Formula: y ~ x^theta1 - theta2

Parameters:
        Estimate Std. Error t value Pr(>|t|)
theta1 2.0002158  0.0004066 4919.15   <2e-16 ***
theta2 0.9531194  0.0389960   24.44   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2797 on 98 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 1.409e-06
```

- `nls` did quite well at estimating the correct structure.

An Artificial Example

- Note that the Gauss-Newton algorithm requires derivatives, and if these derivatives do not exist or do not have real values, the method will fail.
- Try repeating the previous example, but with values of X extended into the negative values.

```
> x <- -1:100/10
> y <- x^2 - 1 + rnorm(102,0,.25)
> m2 <- nls(y~x^theta1 - theta2,
+   start=list(theta1=.5,theta2=.5))
```

- The rather cryptic error message results from an inability to calculate the derivative, i.e.

$$\frac{\partial X^{\theta_1} + \theta_2}{\partial \theta_1} = X^{\theta_1} \log(X) \tag{12}$$

- The derivative is $-\infty$ when $X = 0$, and takes on imaginary values for negative values of X.

An Artificial Example

Turkey Growth Example

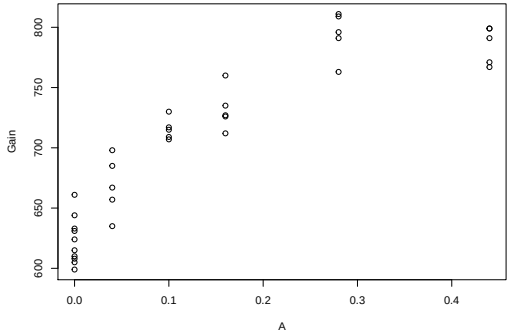
- An experiment was conducted to study the effects on turkey growth of different amounts A of methionine, ranging from a control with no supplementation to 0.44% of the total diet.
- The experimental unit was a pen of young turkeys, and treatments were assigned to pens at random so that 10 pens get the control (no supplementation) and 5 pens received each of the other five amounts used in the experiment, for a total of 35 pens.
- Pen weights, the average weight of the turkeys in the pen, were obtained at the beginning and the end of the experiment three weeks later. The response variable is *Gain*, the average weight gain in grams per turkey in a pen. The weight gains are in the `turk0` data set.
- The primary goal of this experiment is to understand how expected weight gain $E(Gain|A)$ changes as A is varied.

An Artificial Example

Turkey Growth Example

- Here is what a plot reveals.

```
> data(turk0)
> attach(turk0)
> plot(A,Gain)
```



An Artificial Example

Turkey Growth Example

- We can see what appears to be an exponential function with an asymptote at around 810.
- A versatile asymptotic function is the two-parameter exponential augmented with an intercept, i.e.

$$E(Gain|A) = \theta_1 + \theta_2(1 - \exp(-\theta_3 A)) \tag{13}$$

- It helps to have starting values for the parameters, so let's examine the behavior of this function.
- The function takes on a value of θ_1 at $A = 0$, so θ_1 is clearly the intercept, which, we can see from the plot, is roughly 620.
- When $A = \infty$, the function has an asymptote at $\theta_1 + \theta_2$, so θ_2 is the difference between the asymptote and θ_1. A reasonable estimate is $800 - 620 = 180$.

An Artificial Example

Turkey Growth Example

- Getting an estimate for θ_3 is more involved. One approach is to solve equations for a subset of the data.
- Looking at the plot, when $A = .16$, *Gain* is approximately 750, so plugging in these values along with our prior estimates for θ_1 and θ_2 gives

$$
\begin{aligned}
750 &= 620 + 180(1 - \exp(-\theta_3(.16))) && (14)\\
130 &= 180(1 - \exp(-\theta_3(.16))) && (15)\\
130/180 - 1 &= -\exp(-\theta_3(.16)) && (16)\\
\log(50/180) &= -\theta_3(.16) && (17)\\
-\log(50/180)/.16 &= \theta_3 && (18)
\end{aligned}
$$

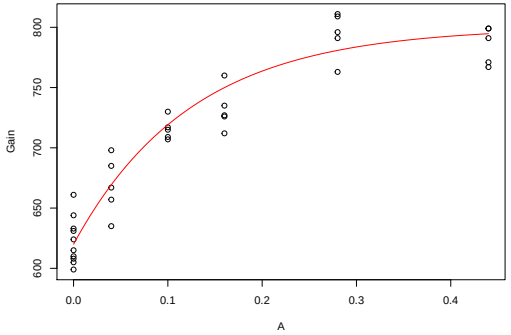
This evaluates to 8.005837 in R.

An Artificial Example

Turkey Growth Example

- We can check out how this approximation looks by adding our curve to the plot of the data.

```
> plot(A,Gain)
> curve(620+180*(1-exp(-8*x)),add=T,col="red")
```



An Artificial Example

Turkey Growth Example

- The fit looks good with the starting values, so we should be able to get convergence with `nls`

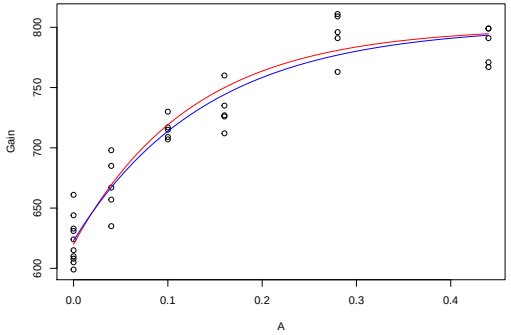
```
> m1 <- nls(Gain ~ theta1 + theta2 *
+ (1 - exp(-theta3 * A)),
+ start=list(theta1=620,theta2=180,theta3=8))
> summary(m1)

Formula: Gain ~ theta1 + theta2 * (1 - exp(-theta3 * A))

Parameters:
       Estimate Std. Error t value Pr(>|t|)
theta1  622.958      5.901  105.57  < 2e-16 ***
theta2  178.252     11.636   15.32 2.74e-16 ***
theta3    7.122      1.205    5.91 1.41e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.66 on 32 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 6.736e-06
```

An Artificial Example

Turkey Growth Example

- Plotting the final fitted function shows only minor change from our starting values.

```
> plot(A,Gain)
> curve(620+180*(1-exp(-8*x)),add=T,col="red")
> curve(622.958+178.252*(1-exp(-7.122*x)),col="blue" ,add=T)
```



An Artificial Example

Turkey Growth Example

- Using the repeated observations at each level of A, we can perform a lack-of-fit test for the mean function.
- The idea, as you recall from Weisberg section 5.3, is to compare the nonlinear fit to the one-way analysis of variance, using the levels of the predictor as a grouping variable.
- The residual variance in ANOVA is computed and pooled strictly within-group, and consequently is a measure of error variance that does not depend on the model we fit.
- That estimate of variance is compared to the estimate obtained from fitting our exponential model.
- As the logic goes, failure to reject supports the idea that the model is reasonable.

An Artificial Example

Turkey Growth Example

```
> p1 <- lm(Gain~as.factor(A),turk0)
> xtablenew(anova(m1,p1))
```

	Res.Df	Res.Sum Sq	Df	Sum Sq	F value	Pr(>F)
1	32	12367.42				
2	29	9823.60	3	2543.82	2.50	0.0789

which $F = 2.50$ with $(3, 29)$ df, for a significance level of 0.08, so we cannot reject the notion that the fit appears adequate.

An Artificial Example

Three Sources of Methionine

- The complete turkey experiment, with data in the file `turkey`, actually investigated 3 sources of methionine, which we might call S_1, S_2, S_3.
- We wish to fit response curves separately for the 3 sources, and test whether they are different, and how well they fit.
- We quickly realize that, for $A = 0$, it doesn't matter what the source was, so the expected response is the same at $A = 0$ for all 3 sources.
- Treating the S_i as dummy variables, we may write

$$\begin{aligned} E(Gain|A = a, S_1, S_2, S_3) &= \theta_1 + S_1[\theta_{21}(1 - \exp(-\theta_{31}a))] \\ &+ S_2[\theta_{22}(1 - \exp(-\theta_{32}a))] \\ &+ S_3[\theta_{23}(1 - \exp(-\theta_{33}a))] \end{aligned}$$

An Artificial Example

Three Sources of Methionine

- Another reasonable function has common intercepts and asymptotes, but separate rate parameters:

$$\begin{aligned} E(Gain|A = a, S_1, S_2, S_3) &= \theta_1 + \theta_2\{S_1[1 - \exp(-\theta_{31}a)] \\ &+ S_2[1 - \exp(-\theta_{32}a)] \\ &+ S_3[1 - \exp(-\theta_{33}a)]\} \end{aligned}$$

- Even more restricted is the model that specifies a common exponential function for all 3 sources:

$$E(Gain|A = a, S_1, S_2, S_3) = \theta_1 + \theta_2(1 - \exp(-\theta_3 A)) \quad (19)$$

An Artificial Example

Three Sources of Methionine

- We use weighted least squares nonlinear regression.

```
> data(turkey)
> tdata <- turkey
> tdata <- turkey
> # create the indicators for the categories of S
> tdata$S1 <- tdata$S2 <- tdata$S3 <- rep(0,dim(tdata)[1])
> tdata$S1[tdata$S==1] <- 1
> tdata$S2[tdata$S==2] <- 1
> tdata$S3[tdata$S==3] <- 1
> m4a <- nls( Gain ~ th1 + th2*(1-exp(-th3*A)),weights=m,
+           data=tdata,start=list(th1=620,th2=200,th3=10))
> m3a <- nls( Gain ~ th1 + th2 *(
+                     S1*(1-exp(-th31*A))+
+                     S2*(1-exp(-th32*A))+
+                     S3*(1-exp(-th33*A))),weights=m,
+           data=tdata,start= list(th1=620, th2=200, th31=10,th32=10,th33=10))
> m2a <- nls(Gain ~ th1 +
+                    S1*(th21*(1-exp(-th31*A)))+
+                    S2*(th22*(1-exp(-th32*A)))+
+                    S3*(th23*(1-exp(-th33*A))),weights=m,
+           data=tdata,start= list(th1=620,
+                                  th21=200,th22=200,th23=200,
+                                  th31=10,th32=10,th33=10))
> m1a <- nls( Gain ~ S1*(th11 + th21*(1-exp(-th31*A)))+
+                     S2*(th12 + th22*(1-exp(-th32*A)))+
+                     S3*(th13 + th23*(1-exp(-th33*A))),weights=m,
+            data=tdata,start= list(th11=620,th12=620,th13=620,
+                                   th21=200,th22=200,th23=200,
+                                   th31=10,th32=10,th33=10))
```

An Artificial Example

Three Sources of Methionine

- Reproducing the ANOVA table, we see that the relaxed models don't appear to improve significantly on the most restricted model.

```
> xtablenew(anova(m4a,m3a,m2a,m1a))
```

	Res.Df	Res.Sum Sq	Df	Sum Sq	F value	Pr(>F)
1	10	4326.08				
2	8	2568.39	2	1757.69	2.74	0.1242
3	6	2040.01	2	528.38	0.78	0.5011
4	4	1151.15	2	888.85	1.54	0.3184

- Note that the ANOVA method reports F values (shown in the above table) that disagree slightly with the calculations (even with the errata) in chapter 11.
- This is because these ANOVAs use as an estimate of error variance the model-derived estimate from the model with the smaller residual sum of squares.

An Artificial Example

Three Sources of Methionine

- The calculation shown in the book uses $\hat{\sigma}^2_{pe}$, the "model free" estimate obtained by pooling within-group variances.

```
> sspe <- sum(tdata$SD^2*(tdata$m-1))
> dfpe <- sum(tdata$m-1)
> s2pe <- sspe/dfpe
> sspe; dfpe; s2pe
[1] 19916
[1] 57
[1] 349.4035
```

An Artificial Example

Three Sources of Methionine

- A test of model fit is not rejected, even for the most restricted model. The corrected calculations are shown below.

```
> F = (4326.1/10)/s2pe
> F
[1] 1.238139
> 1-pf(F,10,57)
[1] 0.2874172
```

Bootstrap Inference

- Inference methods based on large samples depends on the rate of convergence to the asumptotic result.
- Large sample inference depends for its accuracy on a host of factors, including the way the mean function was parameterized.
- Weisberg suggests bootstrapping as a way to alert oneself to situations where the large sample theory may not be working well.

Bootstrap Inference

The Temperature Example

- The data set `segreg` contains data on electricity consumption in KWH and mean temperature in degrees F for one building on the University of Minnesota's Twin Cities campus for 39 months in 1988–1992.
- As usual, higher temperature should lead to higher consumption. (Steam heating simplifies things by essentially eliminating the use of electricity for heating.)
- The mean function plotted to the data is

$$E(C|Temp) = \begin{cases} \theta_0 & Temp \leq \gamma \\ \theta_0 + \theta_1(Temp - \gamma) & Temp > \gamma \end{cases}$$

- The interpretation of this is pretty straightforward. What do the parameters mean? (C.P.)

Bootstrap Inference

The Temperature Example

- The mean function can be rewritten as

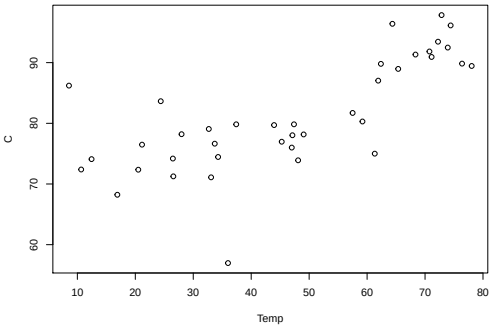
$$E(C|Temp) = \theta_0 + \theta_1(\max(0, Temp - \gamma)) \tag{20}$$

- Plotting C vs. $Temp$ (see next slide) suggests starting values of about 73,0.5, and 40 for θ_0, θ_1, and γ, respectively.

Bootstrap Inference

The Temperature Example

```
> data(segreg)
> attach(segreg)
> plot(Temp,C)
```



Bootstrap Inference

The Temperature Example

- This can be fit easily with `nls`

```
> m1 <- nls(C ~ th0 + th1*(pmax(0,Temp-gamma)),
+           data=segreg,start=list(th0=70,th1=.5,gamma=40))
> summary(m1)

Formula: C ~ th0 + th1 * (pmax(0, Temp - gamma))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
th0    74.6953     1.3433  55.607  < 2e-16 ***
th1     0.5674     0.1006   5.641 2.10e-06 ***
gamma  41.9512     4.6583   9.006 9.43e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.373 on 36 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.673e-08
```

- From the plot, one might get the impression that information about the knot is asymmetric: γ could be larger than 42 but is very unlikely to be muchless than 42.
- We might expect that, in this case, asymptotic normal theory might be a bad approximation.

Bootstrap Inference

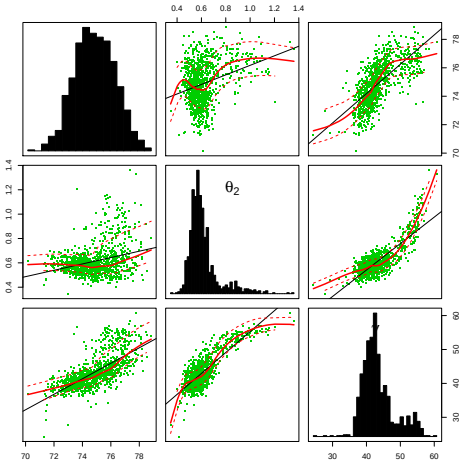
The Temperature Example

- We perform B=999 bootstrap replications, and display the scatterplot matrix of parameter estimates.
- We use the very valuable `boot.case` function.

```
> pdf("ALR_FIG1105.PDF", onefile=T)
> set.seed(10131985)
>   s1.boot <- boot.case(m1,B=999)
>   library(car)
> scatterplotMatrix(s1.boot,diagonal="histogram",
+   col=palette(),#[-1],
+   lwd=0.7,pch=".",
+   var.labels=c(expression(theta[1]),
+   expression(theta[2]),expression(gamma)),
+   ellipse=FALSE,smooth=TRUE,level=c(.90))
```

Bootstrap Inference

The Temperature Example



Bootstrap Inference

The Temperature Example

- The plot displays both the substantial nonnormality of the estimates of θ_2 and γ, but also the correlation between the various parameter estimates.
- ALR Table 11.5 compares the estimates and confidence intervals generated by the asymptotic normal theory and the bootstrap.
- There are some non-negligible differences.
- Weisberg also provides a scatterplot matrix for bootstrapped parameter estimates from the `turkey` data, demonstrating that the asymptotic normal theory is much more appropriate for those data than for the temperature data.

Bootstrap Inference

The Temperature Example

TABLE 11.5 Comparison of Large-Sample and Bootstrap Inference for the Segmented Regression Data

	Large Sample				Bootstrap		
	θ_0	θ_1	γ		θ_0	θ_1	γ
Estimate	74.70	0.57	41.95	Mean	74.92	0.62	43.60
SE	1.34	0.10	4.66	SD	1.47	0.13	4.81
2.5%	72.06	0.37	32.82	2.5%	71.96	0.47	37.16
97.5%	77.33	0.76	51.08	97.5%	77.60	0.99	55.59