



PAPER #803

Exploration of Power Transformations with Monotonic Curvilinear Responses

Soon Hong, hongs@gvsu.edu
Grand Valley State University

Casey Jelsema, jelsemadebater@gmail.com
Grand Valley State University

Key Words: Statistical simulation; Curve straightening; Newton-Raphson optimization; Power transformation; SAS Macro; Coefficient of determination

Abstract

This article introduces a computational method to straighten monotonic curvilinear responses such as growth curves. This method includes estimating the index of the power transformation with the Newton-Raphson optimization technique using a SAS Macro. The properties of the estimate are examined using Monte Carlo simulation. Application of the method is discussed with examples.

1. Introduction

A simple linear regression model is most commonly used to examine the relationship between two quantitative variables. However, data analysts often struggle with fitting a straight-line regression model when the model assumptions are not met. For example, the scatter plot may strongly suggest a non-constant variance of the dependent variable along the regression line. A typical case would be that the variance of the dependent variable increases as the independent variable increases. Box and Cox (1964) suggested a possible remedy by transforming the dependent variable with a power transformation so that the transformed data would correspond more closely to an equal variance assumption at all values of the independent variable.

The power transformation discussed by Box and Cox is:

$$g(y) = \begin{cases} \log(y) & \text{if } p = 0 \\ y^p & \text{if } p > 0 \end{cases}.$$

Another example would be that the scatter plot shows a curvilinear pattern rather than a straight-line pattern. There are many mathematical functions that may depict curvilinear patterns. A special case is that the curvilinear pattern is monotonic increasing or decreasing such as growth curves or efficacy of a drug over time.

Our research is to explore effects of this power transformation on the dependent variable in such cases. Specifically, one of the research objectives is to find out if an optimal power transformation exists in a way that a best straight-line relationship can be achieved. If it does, how do we estimate the power transformation index to best straighten curvilinear data. How good is the estimation method? The properties of the estimate will be examined using Monte Carlo simulation. Ultimately, this computational method can be utilized for data analysis practically as an alternative or prior to fitting other mathematical functions. Our idea is quite intuitive. Consider a scatter plot showing a monotonically increasing or decreasing curvilinear pattern. Then think of raising the dependent variable to a power in an attempt to straighten the curve. While it is possible to subjectively assess the effect of transforming data with a scatter plot, it is preferred to use a numerical measure such as the coefficient of determination, R^2 , since R^2 is the proportion of the sample variance that is accounted for by the variance of the fitted values. According to the discussion of Kvalseth (1985), the coefficient of determination is a measure of straightness of the data in a simple linear regression model. The higher the R^2 , the closer the observed values are to the predicted values which are on the straight line. Therefore, the power which results in the highest R^2 will be the optimal power transformation index to best straighten the data. Hong (1996) previously investigated a rather simple maximization algorithm using a PASCAL program in a different context. We will discuss how we developed a SAS Macro procedure so that one can easily obtain a very precise estimate of the optimal power transformation index.

2. Methods

When we fit a simple linear regression model to a bivariate dataset, we often use R^2 as a measure of how well the regression line fits the data. Thus, we first investigated the effects of the power transformation p on the dependent variable with respect to the R-squared value $R^2(p)$. To do this, we wrote a SAS program that creates a linear bivariate dataset with a random error using the formula $Y = \beta_0 + \beta_1 X + \varepsilon$, where the error term ε was independently and randomly generated from a normal distribution $N(0, \sigma)$. This Y variable can then be raised to a power to create monotonically increasing or decreasing response data over the independent variable. Since we specified the power used to create a monotonically increasing or decreasing function, we knew that the power transformation to reverse the effect of the power would be its inverse. For example, if we cubed the Y variable, the optimal power transformation index to best straighten the data would be $1/3$. When the optimal power transformation index is unknown, we need to estimate the index. Our investigation to develop an estimation method will be illustrated in several phases as follows.

Phase I

Does an optimal power transformation exist? Numerous bivariate data sets were generated with various parameters $(\beta_0, \beta_1, \sigma)$, and transformed with the power $(1/p)$. For each data set, the Y variable was raised to a power p in a certain interval and a simple linear regression was fitted. At various values of p the coefficient of determination was calculated. Numerous scatter plots of $R^2(p)$ versus p showed a smooth curve opening downwards, indicating that there was indeed a maximum $R^2(p)$ with respect to p , and therefore the existence of an optimal power transformation.

For example, a bivariate data set was generated with parameters $(\beta_0 = 1, \beta_1 = 2, \sigma = 3)$ and the Y variable was raised to a power of three as shown in the SAS code below.

```
DATA dataset;
  DO X=1 TO 30;
    CALL STREAMINIT( TIME()*1000 );
    y1 = 1 + 2*X + RAND( 'NORMAL' , 0 , 3 );
    y = y1**3;
  OUTPUT;
END;
RUN;
```

Figure 1 shows the scatter plot of Y versus X . Figure 2 shows the scatter plot of $R^2(p)$ versus p at various values of p in $[0, 1]$.

Figure 1. Plot of Y versus X .

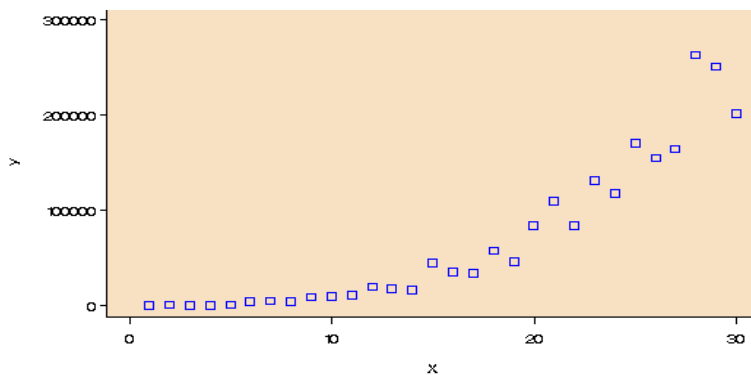
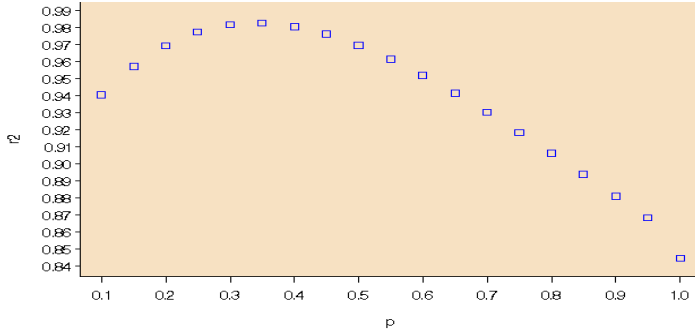


Figure 2. Plot of $R^2(p)$ versus p .



Phase II

How do we estimate the power transformation index to best straighten curvilinear data numerically? We need to find p such that $R^2(p)$ is maximized where $-\infty < p < +\infty$.

The power transformation we propose is:

$$g(y) = \begin{cases} \log(y) & \text{if } p = 0 \\ y^p & \text{if } p \neq 0 \end{cases}.$$

While we used an incremental method to locate a power index p that maximizes the function $R^2(p)$ in phase I, this method would not result in accurate and efficient estimates. So we decided to utilize an efficient algorithm such as the Newton-Raphson optimization procedure PROC NLP in SAS. The NLP procedure (Non Linear Programming) offers a set of optimization techniques for minimizing or maximizing a continuous nonlinear function $g(x)$ of decision variable x with lower and upper bound. This procedure utilizes the Taylor series expansion to find the solution x such that $g'(x) = f(x) = 0$.

An initial value x_1 within bounds $[-4, 4]$ is used to find the next value x_2 in the formula below:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

This iterative algorithm continues until $|x_{n+1} - x_n| < \text{almost } 0 \text{ but positive}$.

However, we need to supply a continuous function $R^2(p)$ to PROC NLP. So we worked on finding various linear or non-linear functions to fit the data ($R^2(p)$ vs. p) such as polynomial functions and gamma density functions. This task was achieved by using the SAS procedure PROC NLIN. PROC NLIN uses one of the iterative methods to fit a linear or non-linear function producing the smallest sum of squared of errors. After we spent much time on this investigation, we concluded that a fifth degree polynomial function was well fitted around the solution and resulted in very accurate (unbiased) estimates while a gamma density function resulted in more precise (smaller variance) but less accurate estimates.

Phase III

We realized that these functions were reasonably well fitted to the data within a small range of p and performance of the optimal power index estimator was excellent. Further, a polynomial function within a small range produced a very accurate optimal power index estimate. So we developed a two-step strategy to find the solution. First, use an incremental method with a wide range, typically $[-4, 4]$ to find an initial optimal power index, p_0 . Second, recalculate $R^2(p)$ within a small range $[p_0 - 0.2, p_0 + 0.2]$ with a typical increment of 80 and fit the data ($R^2(p)$ vs. p) to a fifth degree polynomial function. Then, find a solution that maximizes $R^2(p)$. The SAS MACRO (FindBestPower) program is included in the Appendix. Example SAS code to run the Macro FindBestPower will be given in Example 4.

Phase IV

How good is the estimate? The properties of the estimate will be examined using Monte Carlo simulation. The performance of our method will be shown for various values of the parameters $(\beta_0, \beta_1, \sigma, p)$. One thousand simulations were performed for each combination of parameter values and $\beta_0 = 1$. The summary statistics of the estimates are provided in Tables 1 through 11 in the Appendix.

3. Discussion

This SAS macro procedure is easy to use. This procedure may be used when the scatter plot shows a monotonic increasing or decreasing curvilinear pattern. However, it was often observed that the transformed data would correspond more closely to an equal variance assumption of the simple linear regression model than the original data. Further research is needed to investigate the relationship between Box and Cox's and Hong and Jelsema's power transformation. Some interesting findings as well as recommendations from the simulation study are summarized below.

1. The power index estimator seems unbiased.
2. The standard error seems to increase proportionally as σ increases.
3. The standard error seems to decrease as $|p|$ decreases.
4. A typical bound of $[-4, 4]$ seems to work pretty well. If the solution is outside the bounds, one may change the bounds covering the solution and rerun the procedure.
5. It is recommended to use the increment of 80 for the initial estimate even though a smaller increment may work well.
6. It is recommended to eliminate outliers prior to application.

When the Y -variable has non-positive values, it is necessary to add some positive constant to the Y -variable to make all values positive. The Macro FindBestPower will delete the observation when its y -value is non-positive.

4. Examples

An optimal power index estimate was obtained by running the SAS Macro FindBestPower with three data sets. The following figures show scatter plots of the original data set (Before) and the transformed data set (After).

Example 1: Miles per gallon and horse power of cars

Figure 3 shows a scatter plot of miles per gallon (MPG) versus the horse power (HP) of vehicles. Figures 4 and 5 show scatter plots of the transformed miles per gallon (TMPG) versus the horse power (HP) of vehicles. The estimate of power index was -1.0225 with an outlier and -1.1759 without an outlier. An example of SAS code to find the power index is as follows.

```
%include 'f:\find_bestpower.sas' ;  
%FindBestPower (Dataset=cars, Yvar=mpg, Xvar=hp, Lb=-4, Ub=4, increment=80);
```

The SAS code above assumes the following.

The file that contains the SAS Macro entitled find_bestpower.sas is in the F-drive.

The name of the SAS data set in the Work library = cars.

The name of the Macro = FindBestPower.

The response variable = mpg.

The explanatory variable = hp.

The lower bound of the power index = -4 .

The upper bound of the power index = $+4$.

The increment for the initial search = 80.

Figure 3. Before (Scatter Plot: MPG vs. HP).

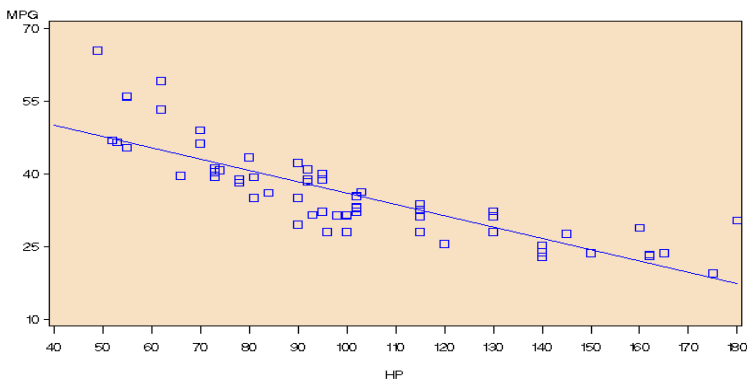


Figure 4. After (Scatter Plot: TMPG vs. HP with power = -1.0225).

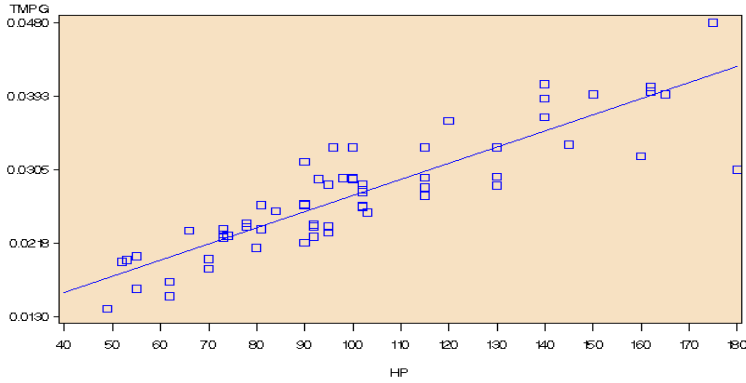
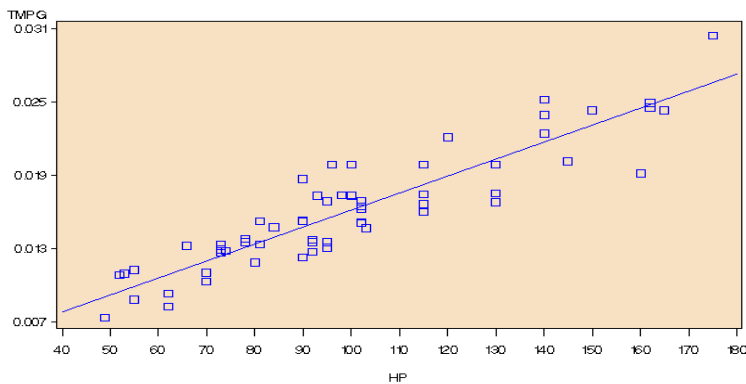


Figure 5. After (Scatter Plot: TMPG vs. HP with power = -1.1759 without an outlier).



Example 2: Bank employee salary and education level

Figure 6 shows a scatter plot of the current salary of bank employees (SAL) versus their education level in years (EDU). Figure 7 shows a scatter plot of the transformed salary (TSAL) versus the education level (EDU). The estimate of the power index was 0.0632. In this case, one may use the natural log transformation since 0.0632 is very close to zero.

Figure 6. Before (Scatter Plot: Salary vs. Education).

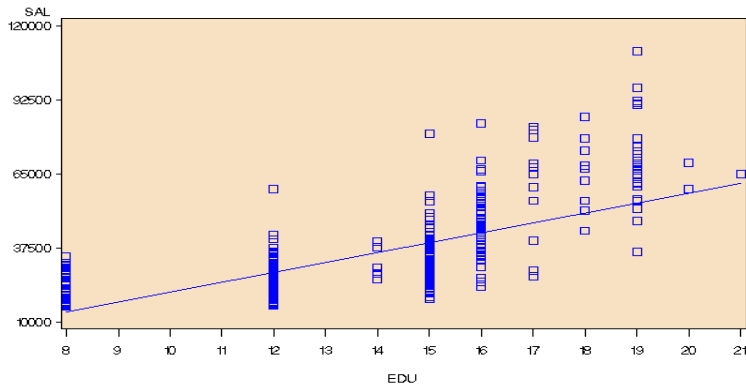
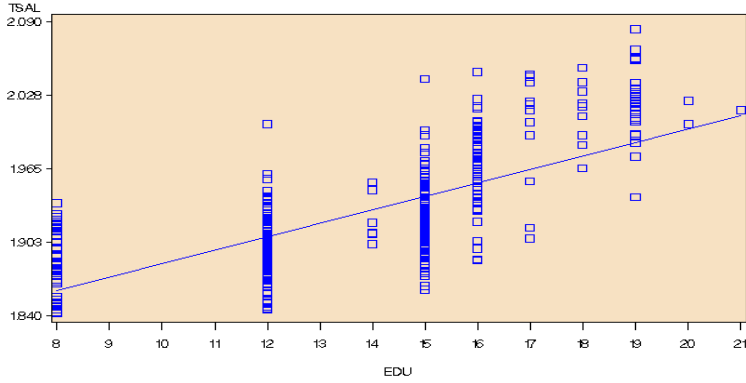


Figure 7. After (Scatter Plot: Salary vs. Education with power = 0.0632).



Example 3: Tumor growth over time

Example 3 represents tumor sizes at selected times for three groups of 10 rats subjected to different immunotherapies. An outlier was removed due to nonstandard growth patterns caused by the remission of tumors prior to application of the method. Figure 8 shows a scatter plot of the mean tumor size (cell means) versus the day of observation (day). Figure 9 shows a scatter plot of the transformed mean tumor size (cell means) versus the day of observation (day). The estimate of power index was 0.2906.

Figure 8. Before (Scatter Plot: Mean Tumor Size vs. Day).

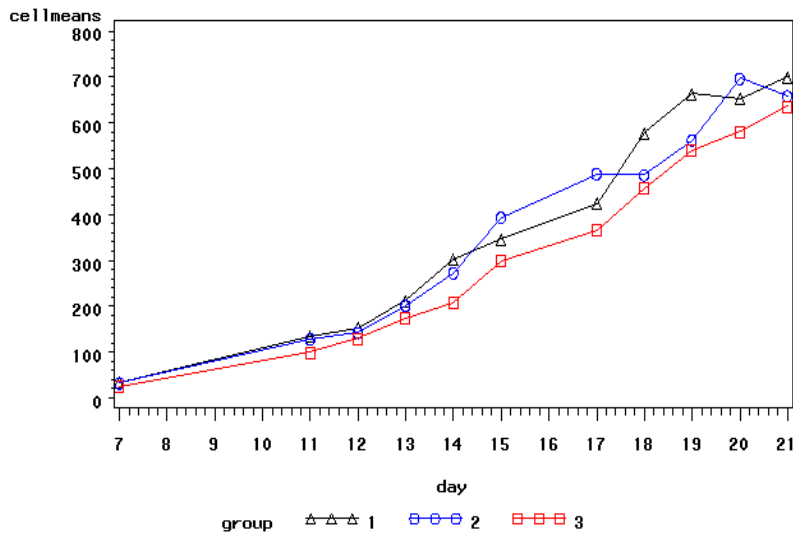
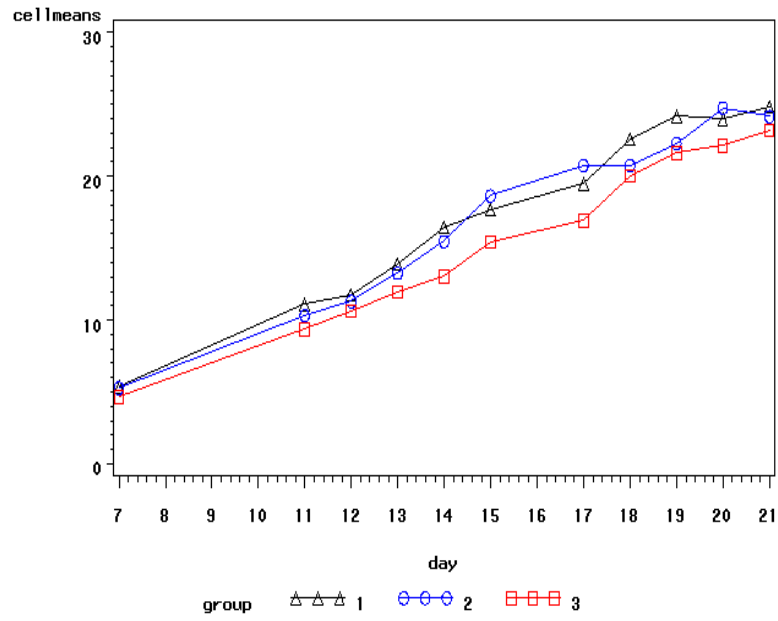


Figure 9. After (Scatter Plot: Mean Tumor Size vs. Day with power = 0.2906).



Appendix

SAS Macro program to find an optimal power index

```
/* Written by Casey Jelsema and Soon Hong in 2008 */
/* Dept of Statistics, Grand Valley State University */
/* This macro will find a best power transformation of */
/* the dependent variable that results in the maximum */
/* coefficient of determination. */
/***** Macro Parameters *****/
/* dataset = name of SAS data set. */
/* Yvar = name of dependent variable. */
/* Xvar = name of independent variable. */
/* Lb = lower bound for power estimate = typically -4. */
/* Ub = upper bound for power estimate = typically +4. */
/* increment = number of increments to be used to pick */
/* an initial power estimate = typically 80. */
%MACRO FindBestPower(dataset,Yvar, Xvar, Lb,Ub,increment);
DATA temporary; RUN;
%LET step = %EVAL((%EVAL(&ub - &lb))+1);
/*This DO-loop iterates through power transformations and
creates a dataset of powers and corresponding r2 values.*/
%DO j=1*&step*&lb %TO 1*&step*&ub;
/*Performs the power transformation, creating variable y*/
DATA temp;
SET &dataset;
if &yvar le 0 then delete;
y = &yvar**(&j/&step);
RUN;
/*Computes regression stats and stores them in set temp2*/
PROC REG DATA=temp NOPRINT OUTEST=temp2 EDF;
MODEL y=&xvar;
RUN;
DATA temp2;
SET temp;
r2 = _RSQ_;
p = (&j/&step);
RUN;
/*Adds latest values to the last step dataset*/
DATA temporary;
SET temporary temp2;
IF r2=. THEN DELETE;
KEEP p r2 ;
RUN;
%END;
PROC SORT DATA=temporary OUT=power_index;
BY DESCENDING r2;
DATA power_index;
SET power_index;
IF _N_ =1 ;
PROC SQL NOPRINT;
SELECT p INTO :init_est SEPARATED BY " " FROM Power_index;
RUN; QUIT;
%LET a2 = %SYSEVALF(&init_est - .2);
%LET b2 = %SYSEVALF(&init_est + .2);
```

```

PROC DATASETS;
    delete temporary;          RUN;
/*This DO-loop iterates through power transformations around the
initial estimate and creates a dataset of powers and corresponding r2
values.*/
%DO j=1 %TO 80;
/*Performs the power transformation, creating variable y*/
    DATA temp;
        SET &dataset;
        y = &yvar**(&a2 + &j*.005);
    RUN;
/*Computes regression stats and stores them in set temp2*/
    PROC REG DATA=temp NOPRINT OUTEST=temp2 EDF;
        MODEL y=&xvar;
    RUN;
    DATA temp2;
        SET temp;
        r2 = _RSQ_;
        p = (&a2 + &j*.005);
    RUN;
/*Adds latest values to the last step dataset*/
/*Deletes unnecessary items from temp dataset*/
    DATA temporary;
        SET temporary temp2;
        IF r2=. THEN DELETE;
        KEEP p r2 ;
    RUN;
%END;
/*NEWTON-RHAPSON OPTIMIZATION - POLYNOMIAL*/
/*Use non-linear regression to generate a function which
uses the power as an input and outputs the r2 value*/
PROC NLIN DATA=temporary NOPRINT;
    PARAMETERS a=0 b=0 c=0 d=0 e=0 g=0 h=0 i=0 j=0 k=0;
    MODEL r2 =
a+(b*p)+(c*(p+d)**2)+(e*(p+g)**3)+(h*(p+i)**4)+(j*(p+k)**5);
    OUTPUT OUT=poly PARMS= a b c d e g h i j k;
    RUN;
/*Uses Newton-Rhapson method (with line search) to find maximum of a
function f. f is defined by the parameters generated by the previous
PROC NLIN.*/
PROC NLP DATA=poly TECH=NEWRAP OUTEST=power_index NOPRINT;
    BOUNDS &a2 < phat , phat < &b2;
    MAX f;
    DECVAR phat;
    f = a+(b*phat)+(c*((phat+d)**2))+(e*(phat+g)**3)+(h*(phat+i)**4)+
(j*(phat+k)**5);
    RUN;
/*END OF NEWTON-RHAPSON OPTIMIZATION - POLYNOMIAL*/
DATA power_index;
    SET power_index;
    IF _TYPE_ = "PARMS";
    Method="Newton-Raphson" ;
    KEEP phat method ;
    RUN;
PROC PRINT DATA=power_index;
    RUN;
%MEND;

```

Simulation results

Table 1. Simulation results of power index estimates when $p = 2.0$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	2.0105	0.0950
	1.0	2.0225	0.1823
	1.5	2.0160	0.2455
	2.0	1.9875	0.2920
1.0	0.5	2.0024	0.0467
	1.0	2.0086	0.0942
	1.5	2.0108	0.1283
	2.0	2.0001	0.1789
2.0	0.5	2.0011	0.0222
	1.0	2.0033	0.0450
	1.5	2.0050	0.0658
	2.0	2.0074	0.0833

Table 2. Simulation results of power index estimates when $p = 2/3$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	0.6693	0.0330
	1.0	0.6734	0.0585
	1.5	0.6713	0.0798
	2.0	0.6579	0.1011
1.0	0.5	0.6678	0.0150
	1.0	0.6700	0.0308
	1.5	0.6700	0.0437
	2.0	0.6680	0.0569
2.0	0.5	0.6671	0.0074
	1.0	0.6675	0.0149
	1.5	0.6673	0.0219
	2.0	0.6685	0.0295

Table 3. Simulation results of power index estimates when $p = 0.5$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	0.5035	0.0250
	1.0	0.5084	0.0435
	1.5	0.5105	0.0570
	2.0	0.5016	0.0688
1.0	0.5	0.5004	0.0118
	1.0	0.5020	0.0229
	1.5	0.5042	0.0328
	2.0	0.5050	0.0397
2.0	0.5	0.5000	0.0055
	1.0	0.5012	0.0111
	1.5	0.5012	0.0165
	2.0	0.5017	0.0218

Table 4. Simulation results of power index estimates when $p = 0.4$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	0.4019	0.0187
	1.0	0.4043	0.0336
	1.5	0.4019	0.0492
	2.0	0.3970	0.0583
1.0	0.5	0.4012	0.0090
	1.0	0.4021	0.0179
	1.5	0.4019	0.0257
	2.0	0.4017	0.0329
2.0	0.5	0.4002	0.0045
	1.0	0.4009	0.0089
	1.5	0.4012	0.0130
	2.0	0.4013	0.0170

Table 5. Simulation results of power index estimates when $p = 1/3$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	0.3356	0.0165
	1.0	0.3364	0.0300
	1.5	0.3331	0.0394
	2.0	0.3293	0.0490
1.0	0.5	0.3337	0.0076
	1.0	0.3361	0.0151
	1.5	0.3341	0.0209
	2.0	0.3359	0.0273
2.0	0.5	0.3334	0.0037
	1.0	0.3337	0.0073
	1.5	0.3340	0.0109
	2.0	0.3340	0.0151

Table 6. Simulation results of power index estimates when $p = 0$ ($Y^* = \ln(Y)$).

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	0.0000	0.0048
	1.0	-0.0001	0.0091
	1.5	0.0000	0.0130
	2.0	-0.0000	0.0156
1.0	0.5	0.0001	0.0010
	1.0	0.0001	0.0020
	1.5	0.0000	0.0033
	2.0	-0.0000	0.0061
2.0	0.5	0.0011	0.0364
	1.0	0.0020	0.0391
	1.5	0.0001	0.0384
	2.0	-0.0000	0.0301

Table 7. Simulation results of power index estimates when $p = -1/3$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	-0.3350	0.0162
	1.0	-0.3371	0.0276
	1.5	-0.3348	0.0397
	2.0	-0.3331	0.0491
1.0	0.5	-0.3340	0.0080
	1.0	-0.3349	0.0146
	1.5	-0.3346	0.0219
	2.0	-0.3352	0.0271
2.0	0.5	-0.3335	0.0035
	1.0	-0.3339	0.0072
	1.5	-0.3337	0.0113
	2.0	-0.3342	0.0147

Table 8. Simulation results of power index estimates when $p = -0.4$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	-0.4011	0.0194
	1.0	-0.4058	0.0360
	1.5	-0.4021	0.0481
	2.0	-0.3982	0.0607
1.0	0.5	-0.4009	0.0092
	1.0	-0.4029	0.0176
	1.5	-0.4026	0.0256
	2.0	-0.4021	0.0329
2.0	0.5	-0.4002	0.0044
	1.0	-0.4003	0.0086
	1.5	-0.4006	0.0126
	2.0	-0.4019	0.0173

Table 9. Simulation results of power index estimates when $p = -0.5$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	-0.5037	0.0245
	1.0	-0.5078	0.0459
	1.5	-0.5048	0.0575
	2.0	-0.5010	0.0671
1.0	0.5	-0.5009	0.0112
	1.0	-0.5031	0.0225
	1.5	-0.5025	0.0328
	2.0	-0.5050	0.0400
2.0	0.5	-0.5002	0.0057
	1.0	-0.5009	0.0112
	1.5	-0.5014	0.0173
	2.0	-0.5015	0.0208

Table 10. Simulation results of power index estimates when $p = -2/3$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	-0.6699	0.0315
	1.0	-0.6770	0.0576
	1.5	-0.6718	0.0832
	2.0	-0.6615	0.0981
1.0	0.5	-0.6680	0.0155
	1.0	-0.6710	0.0312
	1.5	-0.6694	0.0448
	2.0	-0.6695	0.0567
2.0	0.5	-0.6668	0.0074
	1.0	-0.6680	0.0150
	1.5	-0.6685	0.0214
	2.0	-0.6686	0.0269

Table 11. Simulation results of power index estimates when $p = -2.0$.

Slope	Sigma	Mean	Std. Dev.
0.5	0.5	-2.0077	0.0985
	1.0	-2.0204	0.1844
	1.5	-2.0016	0.2415
	2.0	-1.9815	0.3205
1.0	0.5	-2.0020	0.0466
	1.0	-2.0127	0.0890
	1.5	-2.0093	0.1257
	2.0	-2.0100	0.1655
2.0	0.5	-2.0012	0.0228
	1.0	-2.0027	0.0434
	1.5	-2.0057	0.0628
	2.0	-2.0010	0.0901

Acknowledgments

This undergraduate research project was a collaborative effort between Casey Jelsema and Dr. Soon Hong as an independent study in the department of statistics from fall 2007 to winter 2008 at Grand Valley State University.

References

Box, G. E. P., & Cox, D. R. (1964), "Analysis of Transformations," *Journal of the Royal Statistical Society*, B-26, 211-243.

Hong, Soon B. & Koopmans, Lambert H. (1996), "Comparison Problems for Experiments with Curve Responses," *Institute of Mathematical Statistics, Lecture Notes-Monograph Series*, 30, 99-113.

Kvalseth, T.O. (1985), "Cautionary Note About R^2 ," *The American Statistician*, 39, 279-285.

"SAS OnlineDoc, SAS/OR User's Guide, Version 9.1: Mathematical Programming," Cary, NC: SAS Institute, 2003.

"SAS OnlineDoc, SAS/STAT User's Guide, Version 9.1," Cary, NC: SAS Institute, 2003.