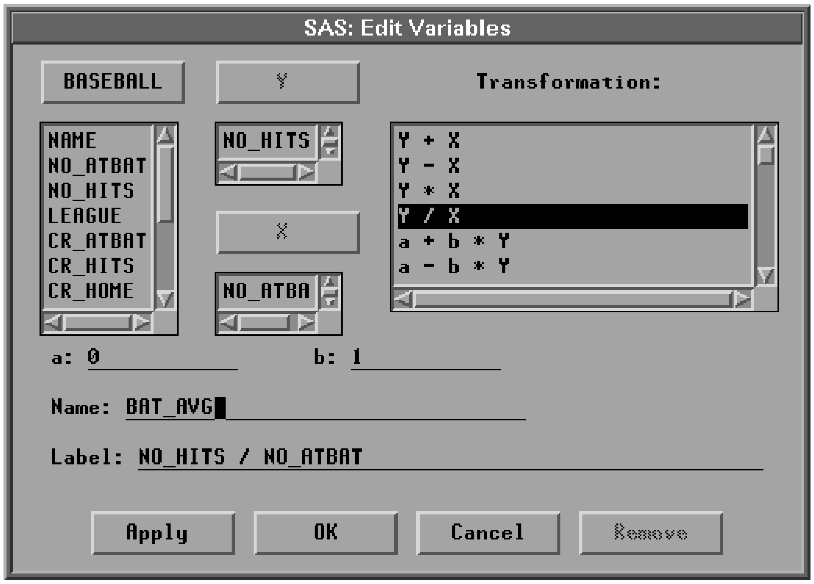
Chapter 20
Transforming Variables

Chapter Table of Contents

Chapter 20
Transforming Variables

A *transformation* generates a new variable from existing variables according to a mathematical formula. SAS/INSIGHT software provides a variety of variable transformations. The most commonly used transformations are available from the **Edit:Variables** menu. You can perform other more complex transformations using the Edit Variables dialog.



Y: NO_HITS

X: NO_ATBA

a: 0

b: 1

Name: BAT_AVG

Label: NO_HITS / NO_ATBAT

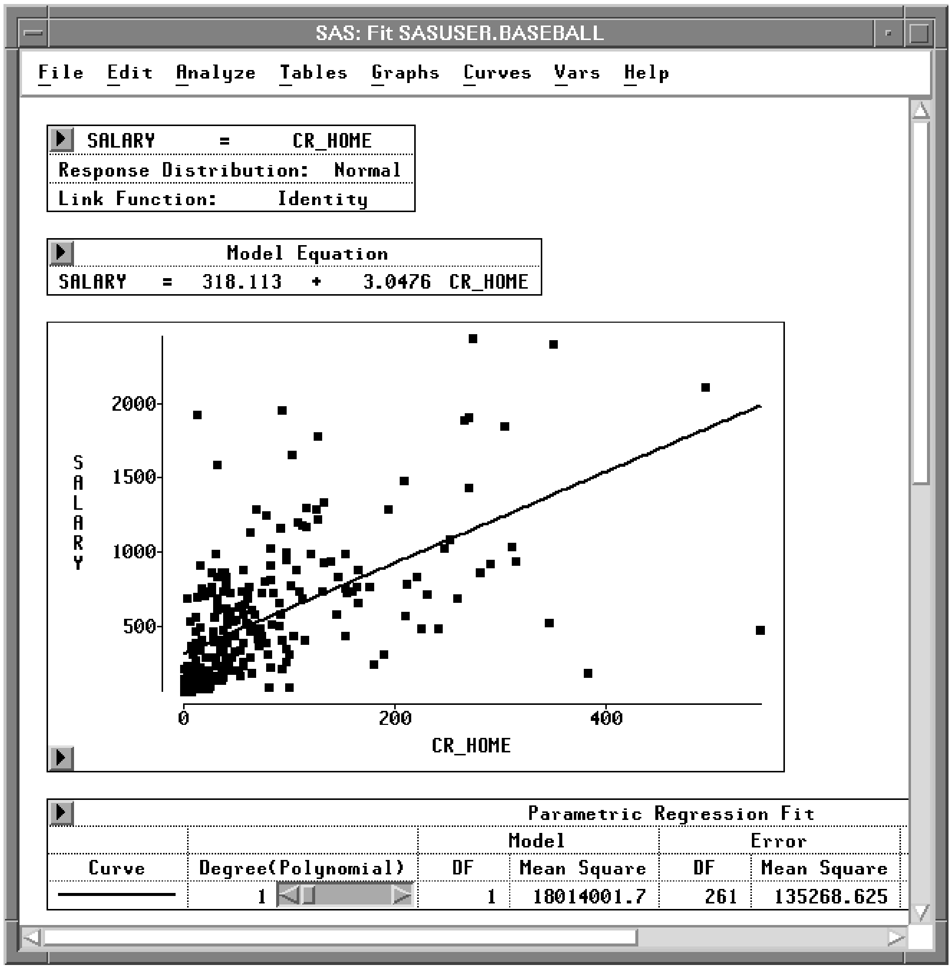
Figure 20.1. Edit Variables Dialog

Common Transformations

The most common transformations are available in the **Edit:Variables** menu. For example, log transformations are commonly used to linearize relationships, stabilize variances, or reduce skewness. Perform a log transformation in a fit window by following these steps:

⟹ **Open the BASEBALL data set.**

⟹ **Create a fit analysis of SALARY versus CR_HOME.**



SALARY	=	CR_HOME
Response Distribution:		Normal
Link Function:		Identity

Model Equation

SALARY = 318.113 + 3.0476 CR_HOME

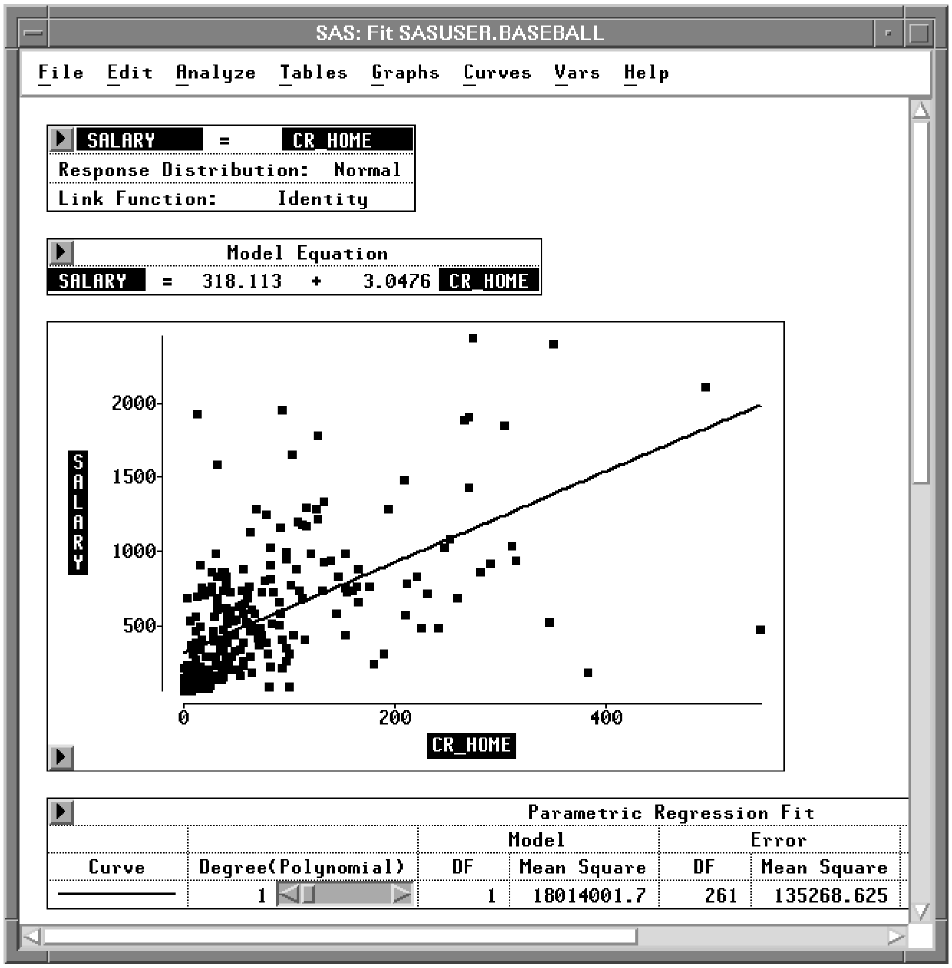
Parametric Regression Fit

		Model		Error	
Curve	Degree(Polynomial)	DF	Mean Square	DF	Mean Square
—	1	1	18014001.7	261	135268.625

Figure 20.2. Fit Analysis of **SALARY** versus **CR_HOME**

You might expect players who hit many home runs to receive high salaries. However, most players do not hit many home runs, and most do not have high salaries. This obscures the relationship between **SALARY** and **CR_HOME**. Most of the observations appear in the lower left corner of the scatter plot, and the regression line does not fit the data well. To make the relationship clearer, apply a logarithmic transformation.

⟹ **Select both variables in the scatter plot.**
Use your host's method for noncontiguous selection.



SAS: Fit SASUSER.BASEBALL

SALARY = CR_HOME
Response Distribution: Normal
Link Function: Identity

Model Equation
SALARY = 318.113 + 3.0476 CR_HOME

Parametric Regression Fit

		Model		Error	
Curve	Degree(Polynomial)	DF	Mean Square	DF	Mean Square
———	1	1	18014001.7	261	135268.625

Figure 20.3. **SALARY** and **CR_HOME** Selected

⟹ **Choose Edit:Variables:log(Y).**

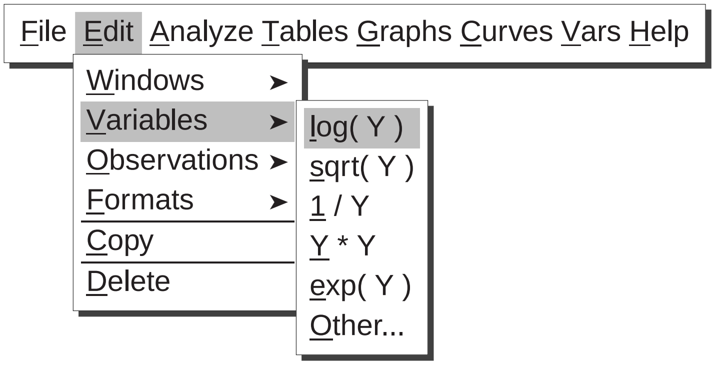
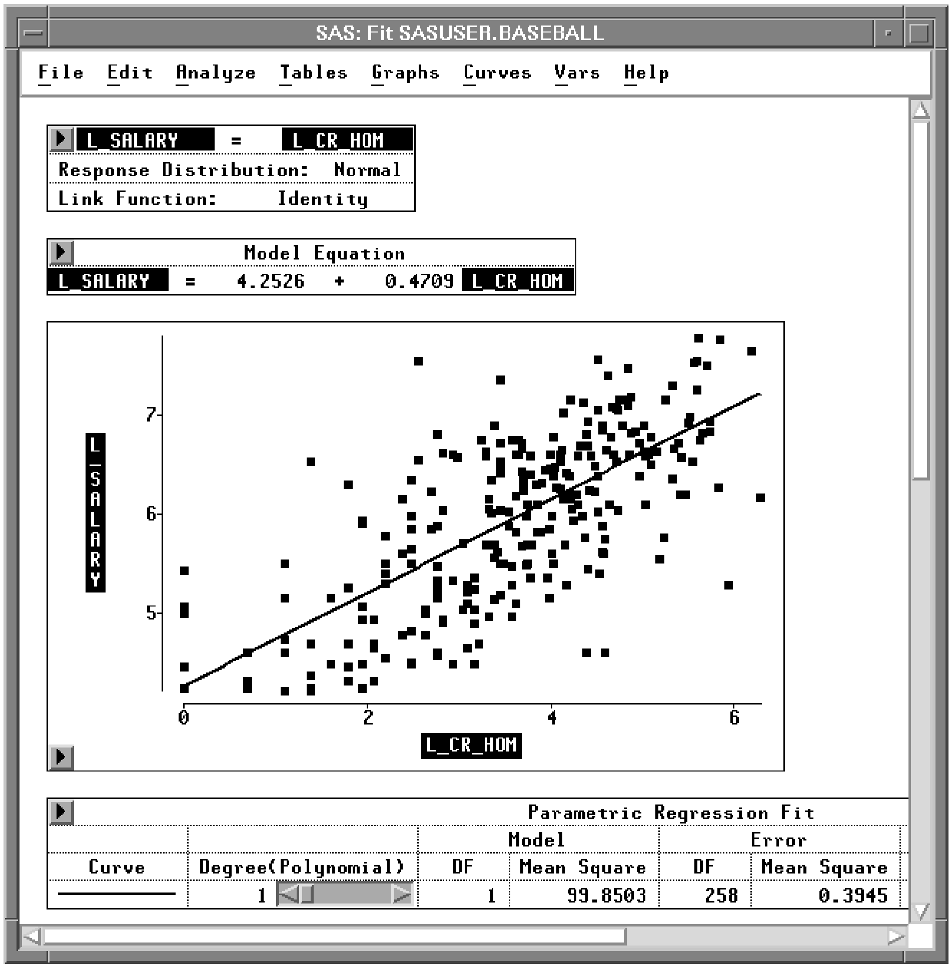


Figure 20.4. Edit:Variables Menu

This performs a log transformation on both **SALARY** and **CR_HOME** and transforms the scatter plot to a log-log plot. Now the regression fit is improved, and the relationship between salary and home run production is clearer.



L_SALARY = L_CR_HOM
Response Distribution: Normal
Link Function: Identity

Model Equation
L_SALARY = 4.2526 + 0.4709 L_CR_HOM

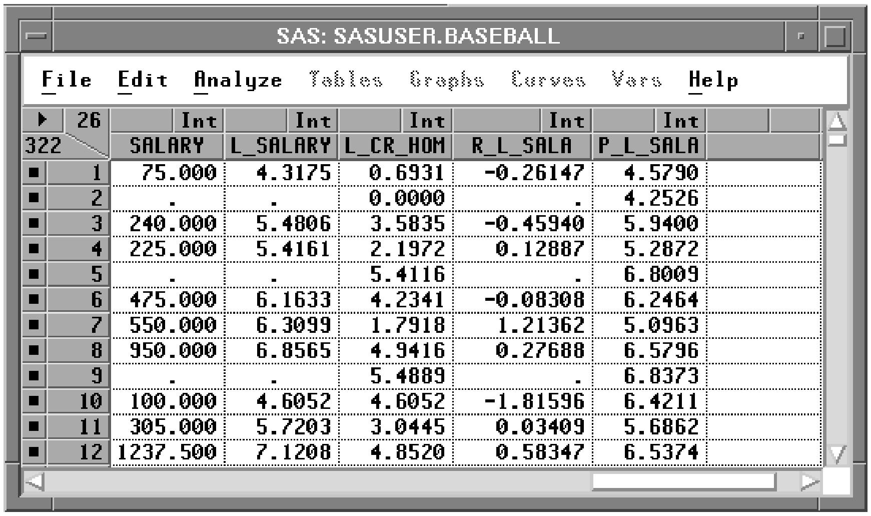
Parametric Regression Fit

		Model		Error	
Curve	Degree(Polynomial)	DF	Mean Square	DF	Mean Square
—	1	1	99.8503	258	0.3945

Figure 20.5. Fit Analysis of **L_SALARY** versus **L_CR_HOM**

The degrees of freedom (**DF**) is reduced from 261 to 258. This is due to missing values resulting from the log transformation, described in the following step.

⟹ **Scroll the data window to display the last four variables.**
Notice that in addition to residual and predicted values from the regression, the log transformations created two new variables: **L_SALARY** and **L_CR_HOM**.

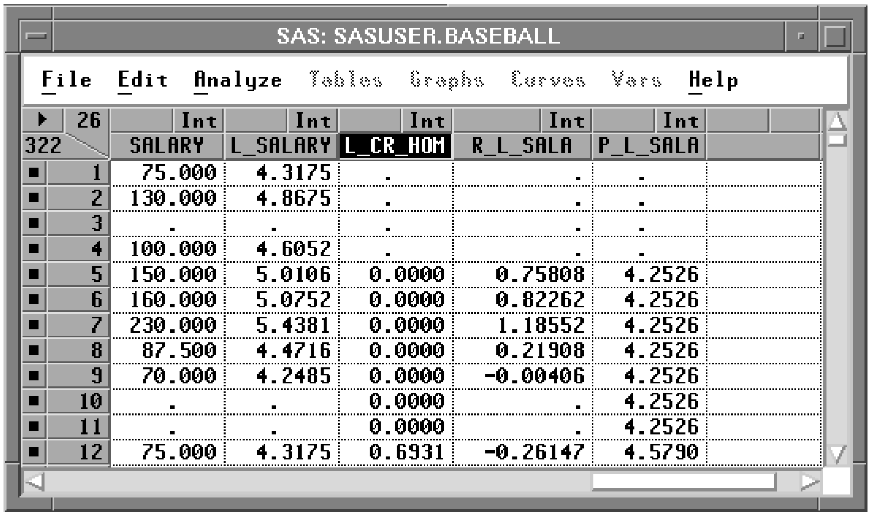


	26	Int	Int	Int	Int	Int
	322	SALARY	L_SALARY	L_CR_HOM	R_L_SALA	P_L_SALA
■	1	75.000	4.3175	0.6931	-0.26147	4.5790
■	2	.	.	0.0000	.	4.2526
■	3	240.000	5.4806	3.5835	-0.45940	5.9400
■	4	225.000	5.4161	2.1972	0.12887	5.2872
■	5	.	.	5.4116	.	6.8009
■	6	475.000	6.1633	4.2341	-0.08308	6.2464
■	7	550.000	6.3099	1.7918	1.21362	5.0963
■	8	950.000	6.8565	4.9416	0.27688	6.5796
■	9	.	.	5.4889	.	6.8373
■	10	100.000	4.6052	4.6052	-1.81596	6.4211
■	11	305.000	5.7203	3.0445	0.03409	5.6862
■	12	1237.500	7.1208	4.8520	0.58347	6.5374

Figure 20.6. New Variables

The log transformation is useful in many cases. However, the result of **log(Y)** is undefined where **Y** is less than or equal to 0. In such cases, SAS/INSIGHT software cannot transform the value, so a missing value (.) is generated. To see this, sort the data in the data window.

⟹ **Select L_CR_HOM in the data window, and choose Sort from the data pop-up menu.**

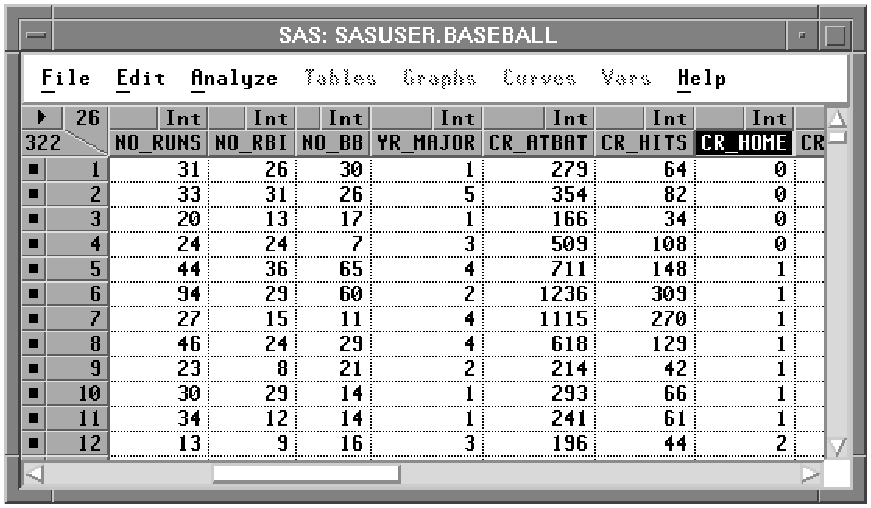


	26	Int	Int	Int	Int	Int
	322	SALARY	L_SALARY	L_CR_HOM	R_L_SALA	P_L_SALA
■	1	75.000	4.3175	.	.	.
■	2	130.000	4.8675	.	.	.
■	3	.	.	.	.	.
■	4	100.000	4.6052	.	.	.
■	5	150.000	5.0106	0.0000	0.75808	4.2526
■	6	160.000	5.0752	0.0000	0.82262	4.2526
■	7	230.000	5.4381	0.0000	1.18552	4.2526
■	8	87.500	4.4716	0.0000	0.21908	4.2526
■	9	70.000	4.2485	0.0000	-0.00406	4.2526
■	10	.	.	0.0000	.	4.2526
■	11	.	.	0.0000	.	4.2526
■	12	75.000	4.3175	0.6931	-0.26147	4.5790

Figure 20.7. Missing Values in Log Transformation

Missing values in the SAS System are considered to be less than any other value, so they appear first in the sorted variable. These values represent players who have never hit home runs. Their value for **CR_HOME** is 0, so the log of this value cannot be calculated. This means the log transformation has removed data from the fit analysis. The following steps circumvent this problem.

⟹ **Select CR_HOME in the data window.**



▶ 26 322	Int NO_RUNS	Int NO_RBI	Int NO_BB	Int YR_MAJOR	Int CR_ATBAT	Int CR_HITS	Int CR_HOME	CR
■ 1	31	26	30	1	279	64	0	
■ 2	33	31	26	5	354	82	0	
■ 3	20	13	17	1	166	34	0	
■ 4	24	24	7	3	509	108	0	
■ 5	44	36	65	4	711	148	1	
■ 6	94	29	60	2	1236	309	1	
■ 7	27	15	11	4	1115	270	1	
■ 8	46	24	29	4	618	129	1	
■ 9	23	8	21	2	214	42	1	
■ 10	30	29	14	1	293	66	1	
■ 11	34	12	14	1	241	61	1	
■ 12	13	9	16	3	196	44	2	

Figure 20.8. **CR_HOME** Selected

⟹ **Choose Edit:Variables:Other.**

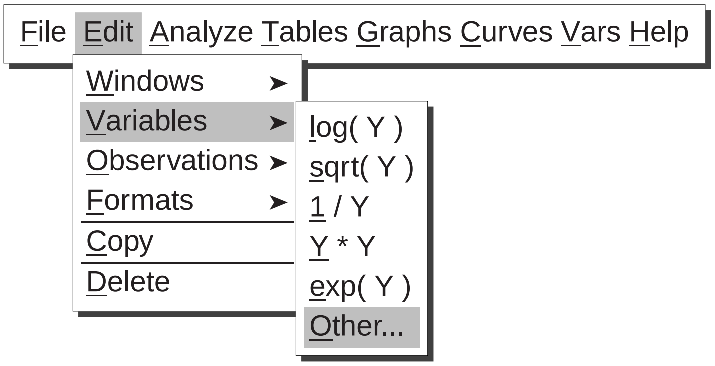
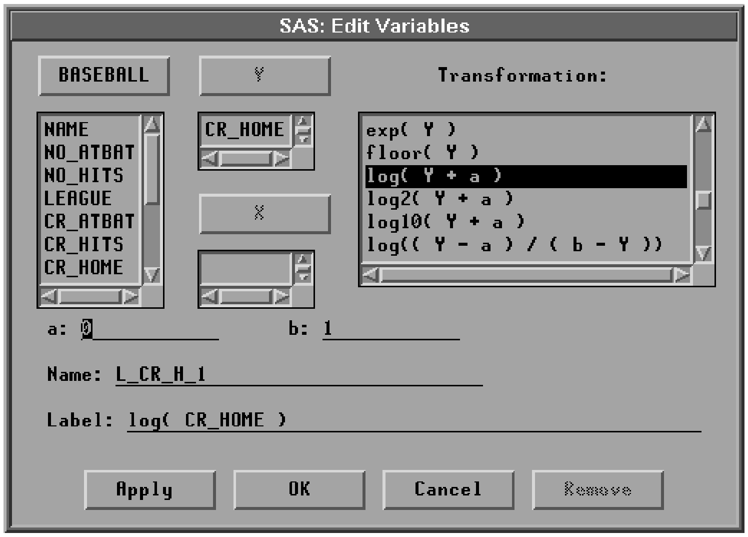


Figure 20.9. Edit:Variables Menu

This displays the Edit Variables dialog shown in Figure 20.10. In the dialog you can see that the variable **CR_HOME** is already assigned as the **Y** variable.

⟹ **Scroll down the transformation window, and select log(Y + a).**



Y: CR_HOME

Transformation: log(Y + a)

a: 0

b: 1

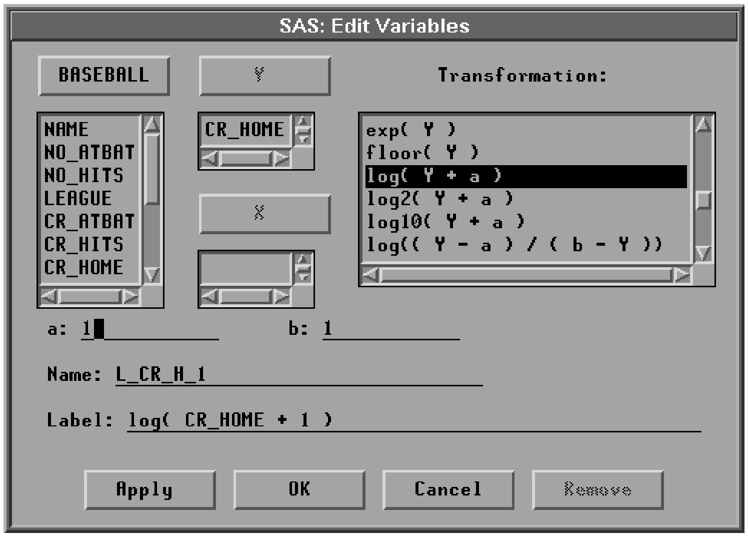
Name: L_CR_H_1

Label: log(CR_HOME)

Figure 20.10. Edit Variables Dialog

⟹ **In the field for a enter the value 1, then press the Return key.**

Notice that the **Label** value changes from **log(CR_HOME)** to **log(CR_HOME + 1)** to reflect the new value of **a**. Setting **a** to **1** avoids the problem of generating missing values because **(CR_HOME + 1)** is greater than zero in all cases for this data.



Y: CR_HOME

Transformation: log(Y + a)

a: 1

b: 1

Name: L_CR_H_1

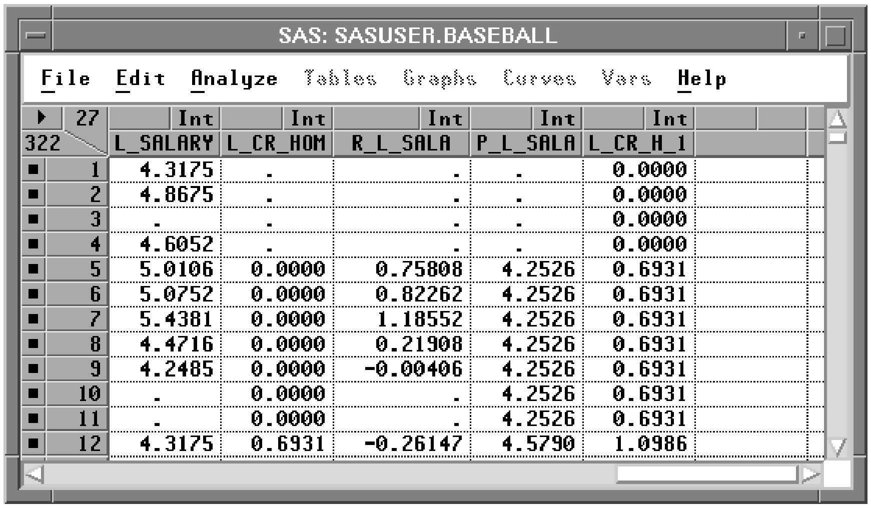
Label: log(CR_HOME + 1)

Figure 20.11. Edit Variables Dialog

⟹ **Click OK to perform the transformation.**

⟹ **Scroll all the way to the right to see the new variable, L_CR_H_1.**

Notice that the new variable contains no missing values.

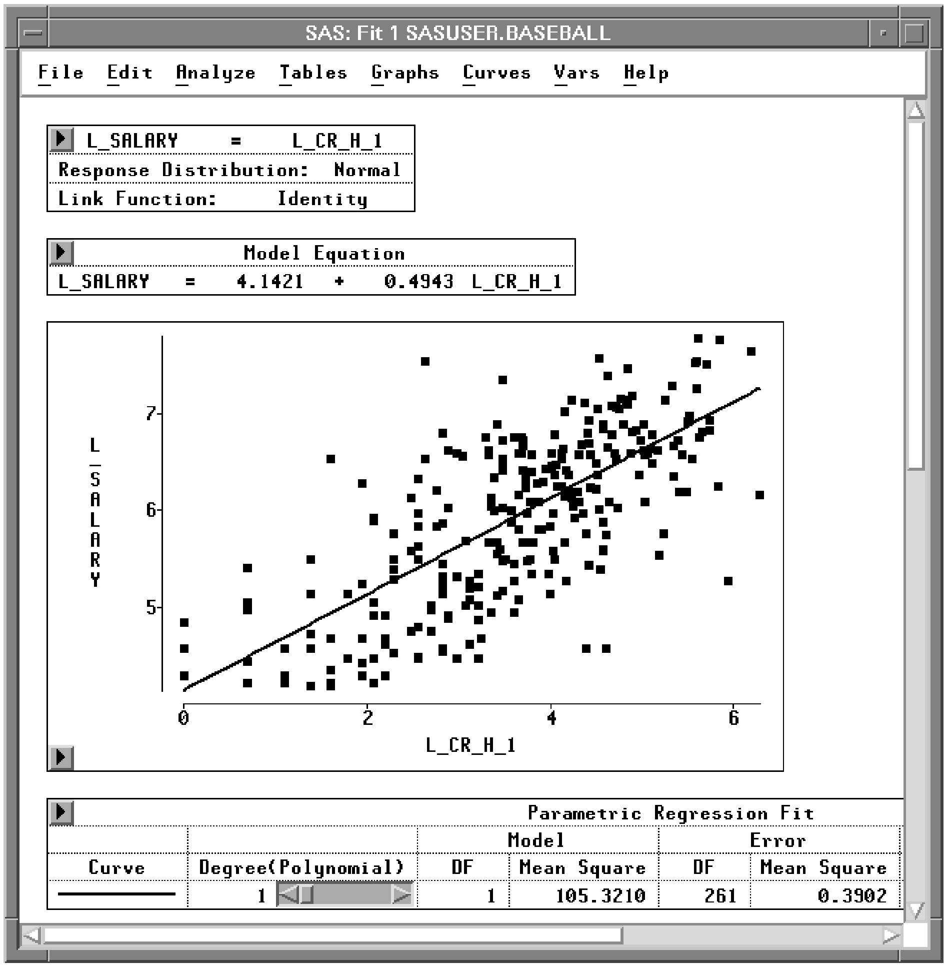


▶ 27	Int	Int	Int	Int	Int
322	L_SALARY	L_CR_HOM	R_L_SALA	P_L_SALA	L_CR_H_1
■ 1	4.3175	.	.	.	0.0000
■ 2	4.8675	.	.	.	0.0000
■ 3	.	.	.	.	0.0000
■ 4	4.6052	.	.	.	0.0000
■ 5	5.0106	0.0000	0.75808	4.2526	0.6931
■ 6	5.0752	0.0000	0.82262	4.2526	0.6931
■ 7	5.4381	0.0000	1.18552	4.2526	0.6931
■ 8	4.4716	0.0000	0.21908	4.2526	0.6931
■ 9	4.2485	0.0000	-0.00406	4.2526	0.6931
■ 10	.	0.0000	.	4.2526	0.6931
■ 11	.	0.0000	.	4.2526	0.6931
■ 12	4.3175	0.6931	-0.26147	4.5790	1.0986

Figure 20.12. New Variable

⟹ **Select L_SALARY and L_CR_H_1, then choose Analyze:Fit (Y X).**

At the lower left corner of the scatter plot, you can see observations that were not used in the previous fit analysis. Also note that the degrees of freedom (**DF**) is back to 261.


SAS: Fit 1 SASUSER.BASEBALL
File Edit Analyze Tables Graphs Curves Vars Help
L_SALARY
7
6
5
0
2
4
6
L_CR_H_1

L_SALARY = L_CR_H_1	
Response Distribution:	Normal
Link Function:	Identity

Model Equation

L_SALARY = 4.1421 + 0.4943 L_CR_H_1

Parametric Regression Fit

		Model		Error	
Curve	Degree(Polynomial)	DF	Mean Square	DF	Mean Square
—	1	1	105.3210	261	0.3902

Figure 20.13. New Fit Analysis

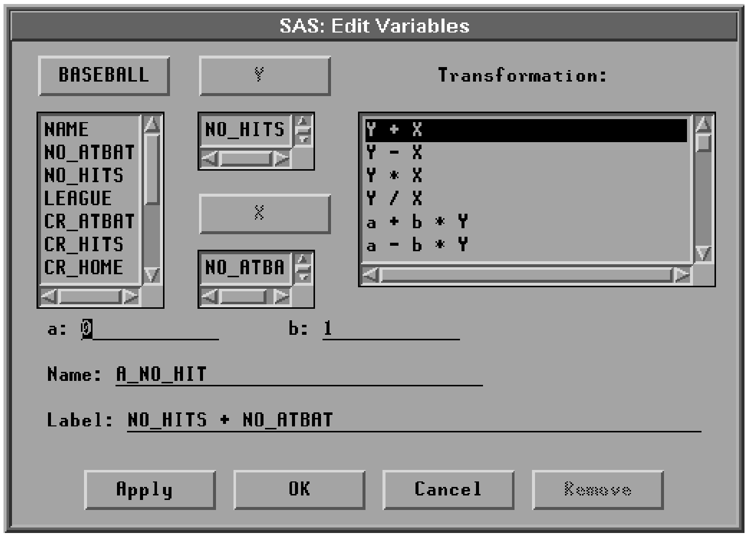
⊕ **Related Reading:** Linear Models, Chapter 39.

Other Transformations

You can use the Edit Variables dialog to create other types of transformations. Most transformations require one selected variable, as in the previous example. Here is an example using two variables. Suppose you are interested in batting averages, that is, the number of hits per batting opportunity. Calculate batting averages by following these steps.

⟹ **Choose Edit:Variables:Other to display the Edit Variables dialog.**

⟹ **Assign NO_HITS the Y role and NO_ATBAT the X role.**



a: 0 b: 1

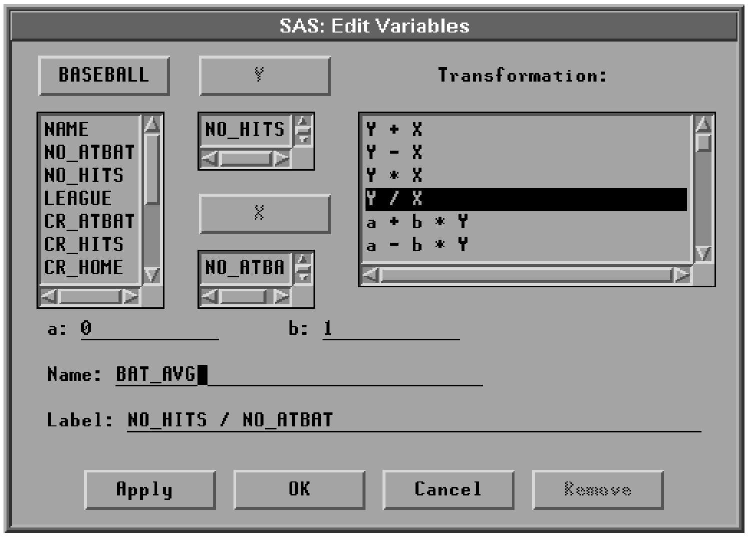
Name: A_NO_HIT

Label: NO_HITS + NO_ATBAT

Figure 20.14. Edit Variables Dialog

⟹ **Click on the Y / X transformation.**
Notice that the **Label** value is now **NO_HITS / NO_ATBAT**. You might want to enter a more mnemonic value for **Name**.

⟹ **Enter BAT_AVG in the Name field.**



a: 0

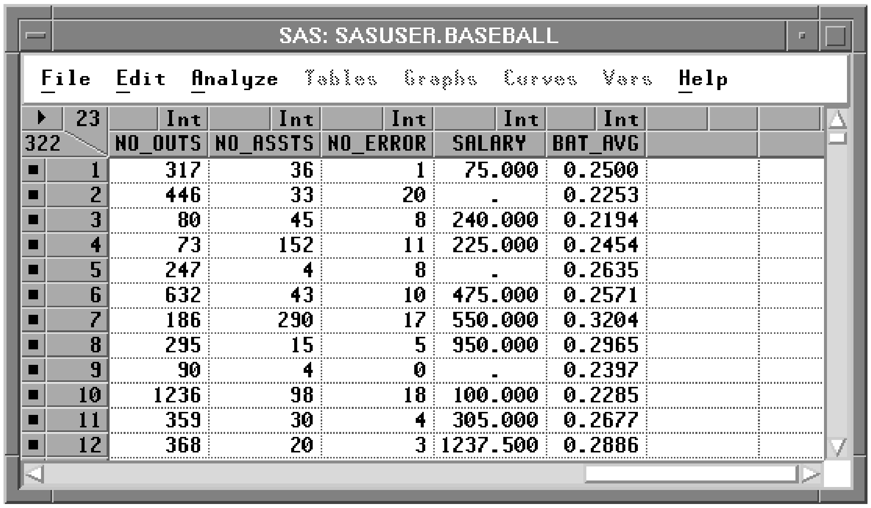
b: 1

Name: BAT_AVG

Label: NO_HITS / NO_ATBAT

Figure 20.15. Creating the Transformation

⟹ **Click the OK button to calculate the batting average.**
The new **BAT_AVG** variable appears at the last position in the data window.



23	Int	Int	Int	Int	Int
322	NO_OUTS	NO_ASSTS	NO_ERROR	SALARY	BAT_AVG
1	317	36	1	75.000	0.2500
2	446	33	20	.	0.2253
3	80	45	8	240.000	0.2194
4	73	152	11	225.000	0.2454
5	247	4	8	.	0.2635
6	632	43	10	475.000	0.2571
7	186	290	17	550.000	0.3204
8	295	15	5	950.000	0.2965
9	90	4	0	.	0.2397
10	1236	98	18	100.000	0.2285
11	359	30	4	305.000	0.2677
12	368	20	3	1237.500	0.2886

Figure 20.16. New **BAT_AVG** Variable

Now look at the distribution of batting averages for each league by creating a box plot.

⟹ **Choose Analyze:Box Plot/Mosaic Plot (Y).**
Specify **BAT_AVG** as the **Y** variable, **LEAGUE** as the **X** variable, and **NAME** for the **Label** role in the box plot variables dialog. Then click on **OK**.

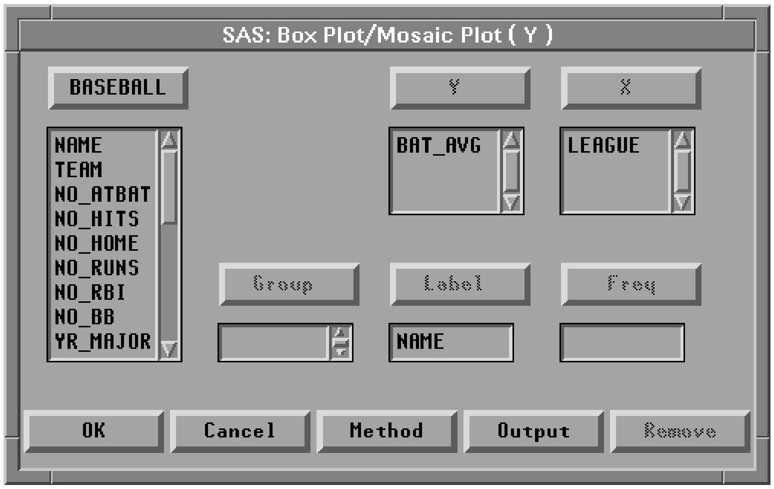


Figure 20.17. Box Plot Dialog

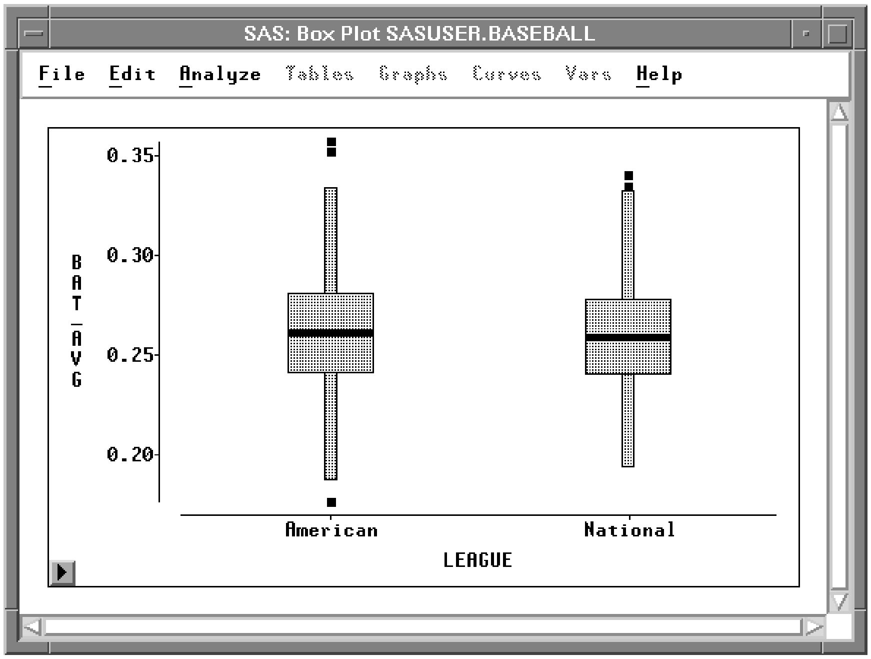


Figure 20.18. Box Plot of Batting Averages

Most players are batting between .200 and .300. There are, however, a few extreme observations.

⟹ **Select the upper extreme observations for each league.**

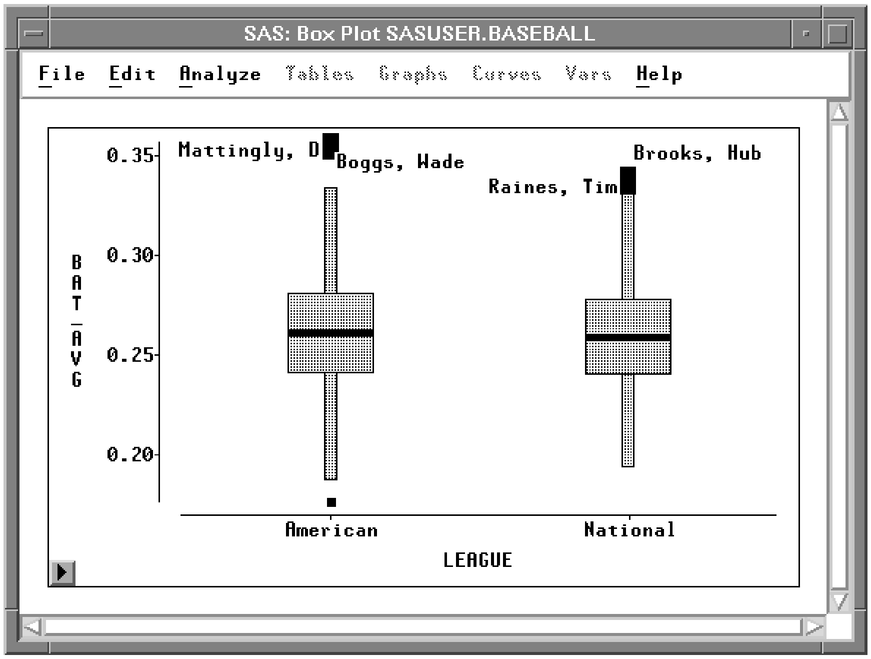


Figure 20.19. Examining the Extreme Observations

Don Mattingly and Wade Boggs led the American League in batting, while Tim Raines and Hubie Brooks led the National League.

The **Edit:Variables** menu and dialog offer many other transformations. Here is the complete list of transformations in the **Edit:Variables** menu:

log(Y)	calculates the natural logarithm of the **Y** variable.
sqrt(Y)	calculates the square root of the **Y** variable.
1 / Y	calculates the reciprocal of the **Y** variable.
Y * Y	calculates the square of the **Y** variable.
exp(Y)	raises e (2.718...) to the power given by the **Y** variable.

Here is the complete list of transformations in the **Edit:Variables** dialog:

Y + X **Y - X** **Y * X** **Y / X**	These four transformations perform addition, subtraction, multiplication, and division on the specified **Y** and **X** variables.
a + b * Y **a - b * Y** **a + b / Y** **a - b / Y**	These four transformations create linear transformations of the **Y** variable. Using the default values **a**=0 and **b**=1, the second and third transformations create additive and multiplicative inverses **-Y** and **1 / Y**.
Y ** b	is the power transform. **b** can be positive or negative.

((Y + a) ** b - 1) / b is the Box-Cox transformation. This transformation raises the sum of the **Y** variable plus **a** to the power **b**, then subtracts 1 and divides by **b**.

a <= Y <= b creates a variable with value 1 when the value of **Y** is between **a** and **b** inclusively, and value 0 for all other values of **Y**. Values for **a** and **b** can be character or numeric; character values should not be in quotations. You can use this transformation to create indicator variables for subsetting your data.

(Y - mean(Y)) / std(Y) standardizes the **Y** variable by subtracting its mean and dividing by its standard deviation. Standardizing changes the mean of the variable to 0 and its standard deviation to 1.

abs(Y) calculates the absolute value of **Y**.

arccos(Y) calculates the arccosine (inverse cosine) of **Y**. The value is returned in radians.

arcsin(Y) calculates the arcsine (inverse sine) of **Y**. The value is returned in radians.

arcsin(sqrt(Y)) calculates the arcsine of the square root of **Y**. The value is returned in radians.

arctan(Y) calculates the arctangent (inverse tangent) of **Y**. The value is returned in radians.

ceil(Y) calculates the smallest integer greater than or equal to **Y**.

cos(Y) calculates the cosine of **Y**.

exp(Y) raises e (2.718...) to the power given by the **Y** variable.

floor(Y) calculates the largest integer less than or equal to **Y**.

log(Y + a) calculates the natural logarithm of the **Y** variable plus an offset **a**.

log2(Y + a) calculates the logarithm base 2 of the **Y** variable plus an offset **a**.

log10(Y + a) calculates the logarithm base 10 of the **Y** variable plus an offset **a**.

log((Y - a) / (b - Y)) calculates the natural logarithm of the quotient of the **Y** variable minus **a** divided by **b** minus the **Y** variable. When **a** = 0 and **b** = 1, this is a logit transformation.

ranbin(a, b) generates a binomial random variable containing values either 0 or 1. **a** is the seed value for the random transformation. **b** is the probability that the generated value will be 1. If **a** is less than or equal to 0, the time of day is used. This is a special case of the SAS function RANBIN where *n*, the number of trials, is 1.

ranexp(a) generates a random variable from an exponential distribution. **a** is the seed value for the random transformation. If **a** is less than or equal to 0, the time of day is used.

rangam(a, b) generates a random variable from a gamma distribution. **a** is the seed value for the random transformation, and **b** is the shape parameter. If **a** is less than or equal to 0, the time of day is used.

rannor(a) generates a random variable from a normal distribution with mean 0 and variance 1. **a** is the seed value for the random transformation. If **a** is less than or equal to 0, the time of day is used.

ranpoi(a, b) generates a random variable from a Poisson distribution. **a** is the seed value for the random transformation, and **b** is the mean parameter. If **a** is less than or equal to 0, the time of day is used.

ranuni(a) generates a uniform random variable containing values between 0 and 1. **a** is the seed value for the random transformation. If **a** is less than or equal to 0, the time of day is used.

round(Y) calculates the nearest integer to **Y**.

sin(Y) calculates the sine of **Y**.

sqrt(Y + a) calculates the square root of the **Y** variable plus an offset **a**.

tan(Y) calculates the tangent of **Y**.

If your work requires other transformations that do not appear in the **Edit:Variables** menu or in the **Edit Variables** dialog, you can perform many kinds of transformations using the SAS DATA step. For more complete descriptions of the **ranbin**, **ranexp**, **rangam**, **rannor**, **ranpoi**, and **ranuni** transformations and for complete information on the DATA step, refer to *SAS Language Reference: Dictionary*.

References

Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. (1983), *Graphical Methods for Data Analysis*, Belmont, CA: Wadsworth International Group.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/INSIGHT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 752 pp.

SAS/INSIGHT User's Guide, Version 8

ISBN 1–58025–490–X

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.