Jean-Paul Nduwayo-Ntore

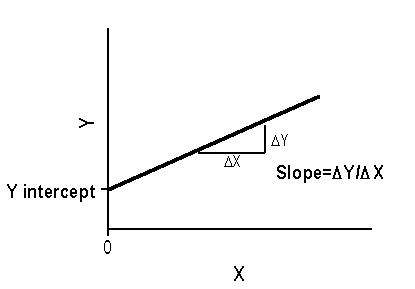
Stat. 550: Regression

Theory: Transforming Nonlinearity

Spring, 2011.

**TRANSFORMING NONLINEARITY**

Nonlinearity is easily understood when we define first what is linearity or a linear relationship. So, a **Linear** Relationshipis a statistical term used to describe the relationship between a variable and a constant. Linear relationships can be expressed in a graphical format where the variable and the constant are connected via a straight line or in a mathematical format where the independent variable is multiplied by the slope coefficient, added by a constant, which determines the dependent variable.



**Why do we want things to be linear?**

* Linear relationships are simple, and there is nice statistical theory for these models.

Thus, an important use of transformations is to ‘straighten’ the relationship between two variables so that they can have a linear relationship.

* This is possible only when the nonlinear relationship is simple and monotone

– Simple implies that the curvature does not change—there is one curve

– Monotone implies that the curve is always positiveor alway*s* negative

NOTE: When transformations fail to remedy these problems, another option is to use nonparametric regression, which makes fewer assumptions about the data.

**The ‘Bulging Rule’ for transformations:**

* Tukey and Mosteller’s rule provides a starting point for possible transformations to correct nonlinearity.
* Normally we should try to transform explanatory variables rather than the response variable Y since a transformation of Y will affect the relationship of Y with all Xs not just the one with the nonlinear relationship.
* If, however, the response variable is highly skewed, it makes sense to transform it instead.

The Bulging Rule states the following:



1. If the data have a shape similar to that shown in the first quadrant, then the data analyst tries re-expressing by going up-ladder for X, Y or both.
2. If the data have a shape similar to that shown in the second quadrant, then the data analyst tries re-expressing by going the down-ladder for X, and/or up-ladder for Y.
3. If the data have a shape similar to that shown in the third quadrant, then the data analyst tries re-expressing by going down-ladder for X, Y or both.
4. If the data have a shape similar to that shown in the fourth quadrant, then the data analyst tries re-expressing by going the up-ladder for X, and/or down-ladder for Y.

**EXAMPLE:**

> set.seed(10000) # To set a seed for the creation of a random numbers

> X<-rnorm(200,2,1) # Generate 200 random numbers, of mean 2 and standard deviation of 1.

> E <- rnorm(200,0,1) # Errors

> Y <- 0.6 \* X^3 + 80 + E

> plot(X,Y)

fit.linear <- lm(Y ~ X)

> plot(X,residuals(fit.linear))

> abline(0,0,lty = 2, col = 'red')



> min(X)

[1] -0.3913438

> X.new1 <- (X + 1)^2 **#INCREASE X**

plot(X.new1, Y)

fit.linear1 <- lm(Y ~ X.new1)

> plot(X.new1,residuals(fit.linear1))

> abline(0,0,lty = 2, col = 'red')



> X.new2 <- (X + 1)^3 **#INCREASE X MORE**

> plot(X.new2, Y)

> fit.linear2 <- lm(Y ~ X.new2)

> plot(X.new2,residuals(fit.linear2))

> abline(0,0,lty = 2, col = 'red')



> X.new3 <- (X + 1)^4 **# FINAL INCREASE FOR X**

> plot(X.new3, Y)

> fit.linear3 <- lm(Y ~ X.new3)

> plot(X.new3,residuals(fit.linear3))

> abline(0,0,lty = 2, col = 'red')

