Simple Regression Case Study

For the first example, suppose we have data on 25 healthy children and we wish to examine the effects of age (X) on the plasma level of a polyamine (Y). I'm not going to give you the data, but the scatterplot is below.
Our first step is to fit a straight line to the data:

```
> (def fit1 (regression-model age plasma))

Least Squares Estimates:

Constant                  13.4752      (0.637862)
Variable 0               -2.18200      (0.260406)

R Squared:               0.753249
Sigma hat:                1.84135
Number of cases:               25
Degrees of freedom:            23

FIT1
```

The Number after "Constant" is the estimate for the intercept, and the number after Variable 0 is the estimate for the slope. We can easily compute t-tests and p-values for the null hypotheses a) intercept =0 and b) slope = 0. Recall that the values of these test statistics are just $\hat{\beta}_i / s_{\hat{\beta}_i}$.

```
> (def t-stats (/ (send fit1 :coef-estimates) (send fit1 :coef-standard-errors)))
T-STATS
> (def p-values (- 1 (t-cdf (abs t-stats) (send fit1 :df))))
P-VALUES
> t-stats
(21.125566672046684 -8.379218569099329)
> p-values
(1.1102230246251565E-16 9.600853334568171E-9)
```

(We'll soon have a better way of doing this. I'll keep you posted.) From the very small p-values we conclude that both terms are significantly different from 0. We also see from the initial output that we have a suitably large value for R-squared. But before we conclude this is a good fit, we must check the residuals.

```
> (def resids (send fit1 :residuals))
RESIDS
```

```
> (plot-points age resids)
> (def expected (rseq (/ 1 26) (/ 25 26) 25))
EXPECTED
> (plot-points (normal-quant expected) (sort-data resids))
```

That last pair of commands generated a qq-plot, compared with a normal distribution.
Just for your own information, this is what the output from the Splus package looks like:

```
Residual Standard Error = 1.8413,  Multiple R-Square = 0.7532
N = 25,  F-statistic = 70.2113 on 1 and 23 df, p-value = 0

              coef std.err   t.stat p.value
Intercept  13.4752  0.6379  21.1256       0
        X  -2.1820  0.2604  -8.3792       0
```

Now try to imrove those residuals by fitting the line $\sqrt(y) = a + bx$.

```
> (def fit2 (regression-model age (sqrt plasma)))

Least Squares Estimates:

Constant                   3.67536      (8.607200E-2)
Variable 0               -0.354140      (3.513875E-2)

R Squared:                0.815369
Sigma hat:                0.248468
Number of cases:                25
Degrees of freedom:             23

FIT2
```

Well, we improved R Squared. We should now check the significance of the coefficients, but for now I'm going to skip this. Instead, let's look at the residuals.

```
> (plot-points age (send fit2 :residuals))
#<Object: 1413480, prototype = SCATTERPLOT-PROTO>
```

They're better, but let's see what happens if we transform to the log of y.

```
> (def fit3 (regression-model age (log plasma))
)

Least Squares Estimates:
```

```
Constant                  2.61302      (4.983179E-2)
Variable 0               -0.235516     (2.034374E-2)

R Squared:               0.853524
Sigma hat:               0.143852
Number of cases:               25
Degrees of freedom:            23

FIT3
> (def fit4 (regression-model age (/ 1 plasma)))

Least Squares Estimates:

Constant                  6.921464E-2  (5.799346E-3)
Variable 0                2.784904E-2  (2.367573E-3)

R Squared:               0.857462
Sigma hat:                1.674127E-2
Number of cases:               25
Degrees of freedom:            23

FIT4
```

I threw in the fit with $1/y$ as well. The R square value changed a fair amount from the square root transformation to the log transformation, but hardly changed from the log transformation to the inverse transformation. This sometimes indicates that we just made an effort to change something with little to show for it. We can compare the residual plots to see if that is the case.
Its a bit of a judgement call, but the residuals look a little more uniformly spread out for the log transform than for the inverse. Let's stick with the log model.