Heteroscedastic Errors

- Sometimes plots and/or tests show that the error variances $\sigma_i^2 = Var(\epsilon_i)$ depend on i
- Several standard approaches to fixing the problem, depending on the nature of the dependence.
  - Weighted Least Squares.
  - Transformation of the response.
  - Generalized Linear Models.

Weighted Least Squares

- Suppose variances are known except for a constant factor.
- That is, $\sigma_i^2 = \sigma^2/w_i$.
- Use **weighted least squares**. (See Chapter 10 in the text.)
- This usually arises realistically in the following situations:
  - Y_i is an average of n_i measurements where you know n_i. Then $w_i = n_i$.
  - Plots suggest that σ_i^2 might be proportional to some power of some covariate: $\sigma_i^2 = kx_i^\gamma$. Then $w_i = x_i^{-\gamma}$.

Variances depending on (mean of) Y

- Two standard approaches are available:
  - Older approach is **transformation**.
  - Newer approach is use of **generalized linear model**; see STAT 402.

Transformation

- Compute $Y_i^* = g(Y_i)$ for some function g like logarithm or square root.
- Then regress Y_i^* on the covariates.
- This approach sometimes works for skewed response variables like income;
- after transformation we occasionally find the errors are more nearly normal, more homoscedastic and that the model is simpler.
- See page 130ff and check under transformations and Box-Cox in the index.

Generalized Linear Models

- Transformation uses the model

$$E(g(Y_i)) = x_i^T \beta$$

while generalized linear models use

$$g(E(Y_i)) = x_i^T \beta$$

- Generally latter approach offers more flexibility.
- Then model variance as a general function of the mean.
- For transformation followed by ordinary least squares the transformed data must follow a *homoscedastic* linear model.
- Hybrid approach also possible: parameters *estimated* by least squares but inference (estimation of SEs, testing, confidence intervals) based on model in which errors may be heteroscedastic.

Weighted Least Squares

- Suppose
$$E(Y_i) = x_i^T\beta$$
and
$$Var(Y_i) = \sigma^2/w_i$$
Suppose *errors are independent with normal distributions*.
- Likelihood (product of normal densities) is
$$\prod_{i=1}^n \frac{\sqrt{w_i}}{\sqrt{2\pi}\sigma}\exp\left[-\frac{w_i}{2\sigma^2}(Y_i - x_i^T\beta)^2\right]$$
- Choose β to maximize this likelihood.
- Minimize
$$\sum_{i=1}^n w_i(Y_i - x_i^T\beta)^2\,.$$
- Process is called weighted least squares.

- Do minimization algebraically.
- Quantity to be minimized is

$$\sum_{i=1}^{n} \left[w_i^{1/2} Y_i - (w_i^{1/2} x_i)^T \beta \right]^2 .$$

- Just an ordinary least squares problem with response variable being

$$Y_i^* = w_i^{1/2} Y_i$$

  and the covariates being

$$x_i^* = w_i^{1/2} x_i .$$

- Calculation can be written in matrix form.

Matrix Formulation

- Let $W^{1/2}$ be diagonal matrix with $w_i^{1/2}$ in ith diagonal position. Put $Y^* = W^{1/2}Y$ and $X^* = W^{1/2}X$. Then

$$Y = X\beta + \epsilon \quad \text{becomes} \quad Y^* = X^*\beta + W^{1/2}\epsilon$$

- If ϵ had mean 0, independent entries and $Var(\epsilon_i) = \sigma^2/w_i$ then $\epsilon^* = W^{1/2}\epsilon$ has mean 0, independent entries $\epsilon_i^* = w_i^{1/2}\epsilon$ and $Var(\epsilon_i^*) = \sigma^2$.
- So ordinary multiple regression theory applies.
- The estimate of β is

$$\hat{\beta}_w = \left[(X^*)^T X^*\right]^{-1}(X^*)^T Y^* = (X^T W X)^{-1} X^T W Y$$

- $W = W^{1/2}W^{1/2}$ is diagonal matrix with w_i on diagonal.
- Estimate is unbiased. Variance covariance is

$$\sigma^2\left[(X^*)^T X^*\right]^{-1} = \sigma^2(X^T W X)^{-1}.$$

Example

- Can do weighted least squares in SAS
- Example: use SENIC data set taking variance of RISK to be proportional to 1/CENSUS.
- Motivation: RISK is an estimated proportion;
- Variance of a Binomial proportion is inversely proportional to the sample size.
- This makes weight CENSUS.

```
proc reg  data=scenic;
  model Risk = Culture Stay Nratio Chest Facil;
  weight Census;
run ;
```

Edited Output

```
                 Sum of     Mean
Source     DF   Squares  Square F Value Prob>F
Model       5  12876.94 2575.39  17.819 0.0001
Error     107  15464.47  144.53
C Total   112  28341.41
 Root MSE       12.02197     R-square   0.4544
 Dep Mean        4.76215     Adj R-sq   0.4289
              Parameter Estimates
               Par     Std    T for H0:
Variable   DF   Est    Error  Par=0   Prob > |T|
INTERCEP    1  0.4681 0.6239  0.750     0.4547
CULTURE     1  0.0300 0.0089  3.365     0.0011
STAY        1  0.2374 0.0444  5.342     0.0001
NRATIO      1  0.6239 0.3480  1.793     0.0759
CHEST       1  0.0035 0.0044  0.799     0.4263
FACIL       1  0.0089 0.0060  1.467     0.1452
```

Edited output for unweighted case

```
               Sum of   Mean
Source    DF  Squares Square F Value Prob>F
Model      5  108.33  21.67  24.913 0.0001
Error    107   93.05   0.87
C Total 112  201.38
  Root MSE  0.93255   R-square   0.5379
  Dep Mean  4.35487   Adj R-sq   0.5163
         Parameter Estimates
                Par      Std  T for H0:
Variable  DF   Est     Error  Par=0 Prob > |T|
INTERCEP   1 -0.7680 0.61022 -1.259  0.2109
CULTURE    1  0.0432 0.00985  4.385  0.0001
STAY       1  0.2339 0.05741  4.075  0.0001
NRATIO     1  0.6724 0.29931  2.246  0.0267
CHEST      1  0.0092 0.00541  1.698  0.0925
FACIL      1  0.0184 0.00630  2.928  0.0042
```

Discussion

- Notice many changes in significance levels.
- Weighted model would fail diagnostic tests – it would be clearly heteroscedastic.
- Can compute standardized residuals and so on from starred variables as usual.

Transformation

- Sometimes response variable has distribution which makes it likely that the errors will be not very normal and that the errors will not be homoscedastic. Typical examples:
    - **Binary Response Data**: the Y_i are either just Bernoulli variables (0 or 1) or Binomial variables.
    - Example: For each of the doses $d_1, \ldots, d_p$ a number of animals $n_1, \ldots, n_p$ are treated with the corresponding dose of some drug.
    - The number, Y, dying at dose d is Binomial with parameter $h(d)$.
    - **Count Data**: the Y_i are counts of the number of times something happens such as the number of traffic accidents at a corner, or cases of leukemia in a region.
    - Typically we suppose Y_i to have Poisson distributions.
    - Skewed continuous data: the Y_i seem to come from some skewed continuous distribution — times to recurrence of a disease after surgery might be an example.

Traditional Analysis: Transformation

- For Binomial Y_i use *arc sin* transformation:

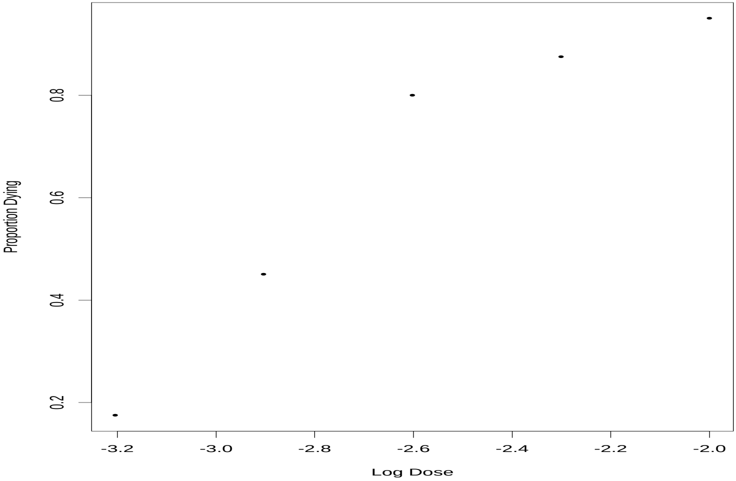
$$Y_i^* = 2\arcsin\sqrt{Y_i/n_i}$$

- For Poisson Y_i take square roots $Y_i^* = \sqrt{Y_i}$.
- Appropriate whenever we think σ_i^2 is proportional to $\mu_i = E(Y_i)$.
- For data such as money where percentage changes might be a sensible way to think about the variable take logarithms, $Y_i^* = \log Y_i$.
- Useful if σ_i is proportional to μ_i.
- Look up Box-Cox transformation to **estimate** the transformation.
- Problem: If the model was linear before transformation then it will not be linear after transformation.

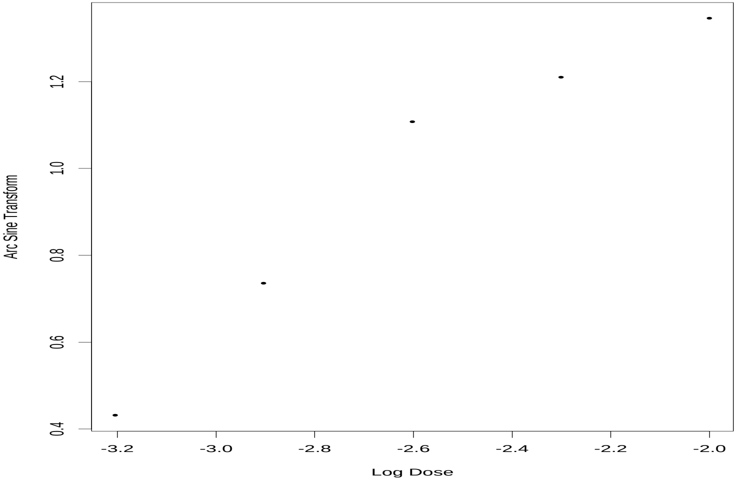
Transformation versus GLIM

- At each of 5 doses of some drug 40 animals were tested.
- Number surviving, Y, recorded for each dose.
- The log doses are -3.204, -2.903, -2.602, -2.301, and -2.000 and the numbers surviving are 7, 18, 32, 35, 38.

Plot of Y_i/n_i versus log Dose



After the arcsine transform



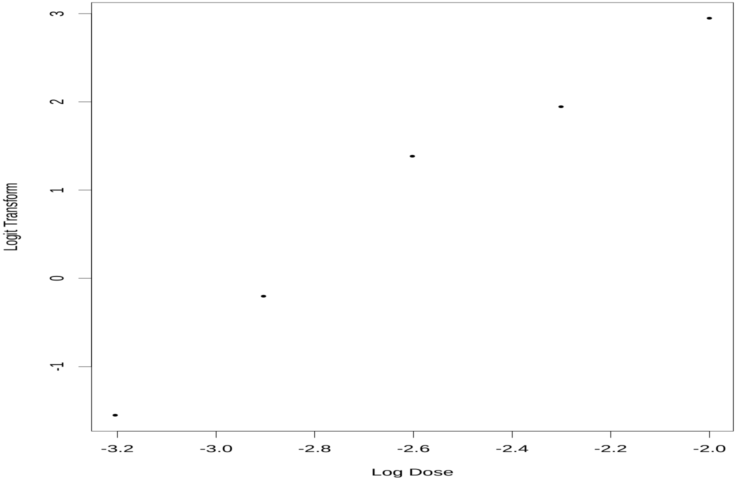
Logistic Transformation

- More standard transformation for this problem is the **logit** or **logistic**

$$Y_i^* = \log[Y_i/(n_i - Y_i)]$$

- Notice chaos if $Y_i = 0$ or $Y_i = n_i$.

Logistic Transformation Plot



Transformation Analysis: logistic

Dose	Dead	Tested
-3.204	7	40
-2.903	18	40
-2.602	32	40
-2.301	35	40
-2.000	38	40

SPlus Code

```
postscript("logist.ps",
               onefile=F,horizontal=F)
plot(Dose, Dead/Tested,xlab="Log Dose",
     ylab="Proportion Dying")
dev.off()
postscript("arc_logist.ps",
      onefile=F,horizontal=F)
plot(Dose, Dead/Tested,xlab="Log Dose",
     ylab="Arc Sine Transform")
dev.off()
```

```
postscript("logit_logist.ps",
     onefile=F,horizontal=F)
plot(Dose,log(Dead/(Tested-Dead)),
     xlab="Log Dose", ylab="Logit Transform")
dev.off()
linfit <- lm( log(Dead/(Tested-Dead)) ~ Dose,
        data=dead)
summary(linfit)
glmfit <- glm( cbind(Dead,Tested-Dead) ~ Dose,
     data=dead, family=binomial)
summary(glmfit)
```

SPlus Plotting Code

```
dead <- read.table("data", header = T)
postscript("logist_plus_curve.ps",
               onefile=F,horizontal=F)
plot(Dose, Dead/Tested,xlab="Log Dose",
     ylab="Proportion Dying")
d <- seq(-3.3,-1.9,length=200)
etalin <- coef(linfit)[1] + d*coef(linfit)[2]
p  <- exp(etalin)/(1+exp(etalin))
lines(d,p)
etaglm <- coef(glmfit)[1] + d*coef(glmfit)[2]
p  <- exp(etaglm)/(1+exp(etaglm))
lines(d,p,lty=2)
dev.off()
```

Output: 4 graphs and following

```
S-PLUS : Copyright (c) 1988, 1996 MathSoft, Inc.
S : Copyright AT&T.
Version 3.4 Release 1 for Sun SPARC, SunOS 5.3 : 1996
Working data will be in .Data
> dead <- read.table("data", header = T)
#
# Read in data.  Columns are named by words
# read off line 1 because of header=T bit.
#
> attach(dead)
#
# Makes variables which are columns of dead
# accessible to the plotting rooutines
#
```

SPlus Plotting Code

```
> postscript("logist.ps",onefile=F,horizontal=F)
#
#    Declares that the next graph should be put
#    in a postscript file called logist.ps.  The
#    file should be encapsulated postscript and in
#    portrait orientation.
#
> plot(Dose, Dead/Tested,xlab="Log Dose",
    ylab="Proportion Dying")
#
#    Plot Proportion dying on the y axis against
#    Dose and label the axes
#
> dev.off()
```

SPlus Plotting Code

```
#
#    Finish up the postscript file
#
Starting to make postscript file.
Finished postscript file,
     executing command "lpr -h logist.ps &".
 null device
           1
> postscript("arc_logist.ps",onefile=F,horizontal=F)
> plot(Dose, asin(sqrt(Dead/Tested)),xlab="Log Dose",
     ylab="Arc Sine Transform")
> dev.off()
```

SPlus Plotting Code

```
> postscript("logit_logist.ps",
       onefile=F,horizontal=F)
> plot(Dose, log(Dead/(Tested-Dead)),
     xlab="Log Dose",
     ylab="Logit Transform")
> dev.off()
> linfit <- lm( log(Dead/(Tested-Dead))~Dose,
          data=dead)
#
#   Regress log(Y/(n-Y)) on Dose
#
```

SPlus Plotting Code

```
> summary(linfit)
#
# Print out a summary of the regression results.
#
Call: lm(formula = log(Dead/(Tested-Dead))~Dose,
     data = dead)
Residuals:
       1       2      3        4      5
 -0.2283 0.00792 0.4812 -0.07283 -0.188

Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept) 10.5322  0.9109    11.5626  0.0014
       Dose  3.6999  0.3455    10.7095  0.0017
```

SPlus Plotting Code

```
Residual std error: 0.3288 on 3 df
Multiple R-Sq: 0.9745
F-: 114.7 on 1 and 3 df, p-value is 0.001741
Correlation of Coefficients:
     (Intercept)
Dose 0.9869
> glmfit <- glm(cbind(Dead,Tested-Dead)~Dose,
     data=dead, family=binomial)
#   Fits the model that log(E(Y)/(n-E(Y))
#   is a linear function of Dose
> summary(glmfit)
Call:glm(formula=cbind(Dead,Tested-Dead)~Dose,
     family = binomial, data = dead)
Deviance Residuals:
       1       2      3       4       5
 -0.4319 -0.0356 1.0264 -0.4763 -0.5454
```

SPlus Plotting Code

```
Coefficients:
                Value Std. Error  t value
(Intercept) 11.238232  1.5651480 7.180300
       Dose  3.936472  0.5604985 7.023162
(Dispersion Parameter for Binomial family
     taken to be 1 )
Null Deviance: 80.77441 on 4 df

Residual Deviance: 1.76566 on 3 df

Number of Fisher Scoring Iterations: 3

Correlation of Coefficients:
     (Intercept)
Dose 0.9929832
```

SPlus Plotting Code

```
> postscript("logist_plus_curve.ps",onefile=F,
     horizontal=F)
> plot(Dose, Dead/Tested,xlab="Log Dose",
     ylab="Proportion Dying")
> d <- seq(-3.3,-1.9,length=200)
> etalin <- coef(linfit)[1]+d*coef(linfit)[2]
> p  <- exp(etalin)/(1+exp(etalin))
#
#  For each dose in d, (list of 200 numbers
#  running from -3.3 to -1.9) compute fitted
#  probability according to logit model:
#  if log(x/(1-x))=p then p=exp(x)/(1+exp(x))
#
> lines(d,p)
# Plot the fitted curve for the least squares
# method on the graph of the data
```

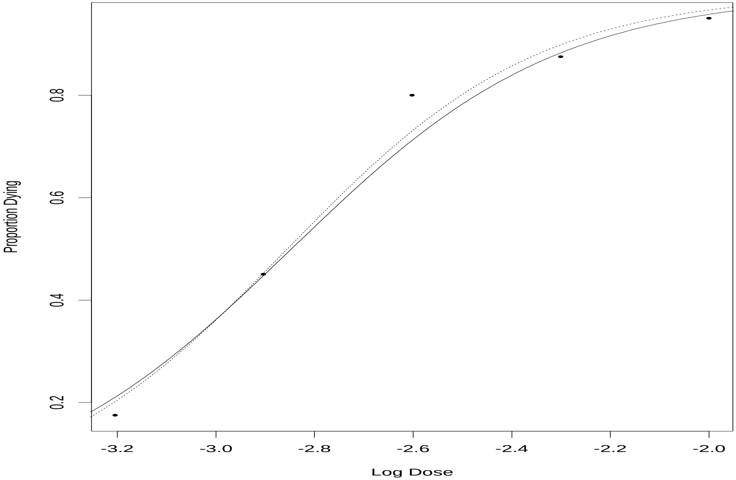
SPlus Plotting Code

```
> etaglm <- coef(glmfit)[1]+d*coef(glmfit)[2]
> p  <- exp(etaglm)/(1+exp(etaglm))
#
# Do same for generalized model fit
#
> lines(d,p,lty=2)
#
# Plot fitted curve for glm method on the
# graph of data. Use dashed (lty=2) line.
#
> dev.off()
> q() # end-of-file
```

Comparison

- Slopes of two fits differ by about one half of one standard error.
- Function `glm` fits a generalized linear model, using *maximum likelihood* methods for a binomial model for the number of dead animals at each dose.
- Standard errors produced by `glm` are more appropriate and larger.
- Linear model fit assumes homoscedasticity which is definitely wrong for binomial data.
- The two fitted curves are plotted along with the data in the last set of lines.

Compare Fits



Poisson Regression: Count Data

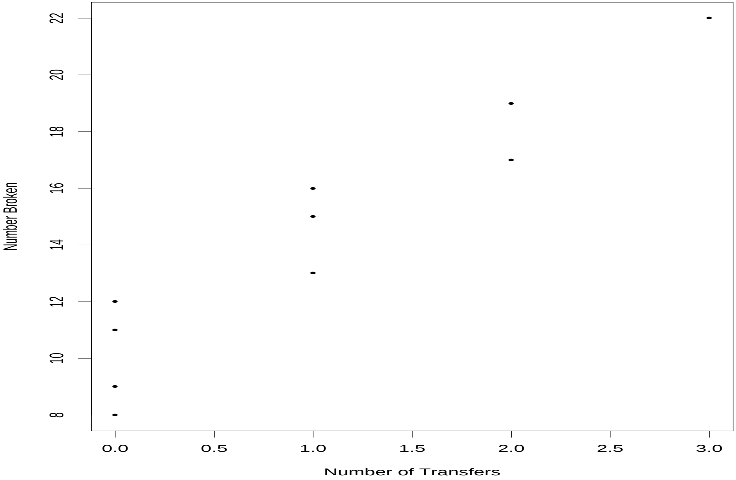
- First row below is the number of times a carton of glass objects was transferred from one aircraft to another during shipping.
- second row is the number of broken objects.

i:	1	2	3	4	5	6	7	8	9	10
X_i	1	0	2	0	3	1	0	1	2	0
Y_i	16	9	17	12	22	13	8	15	19	11

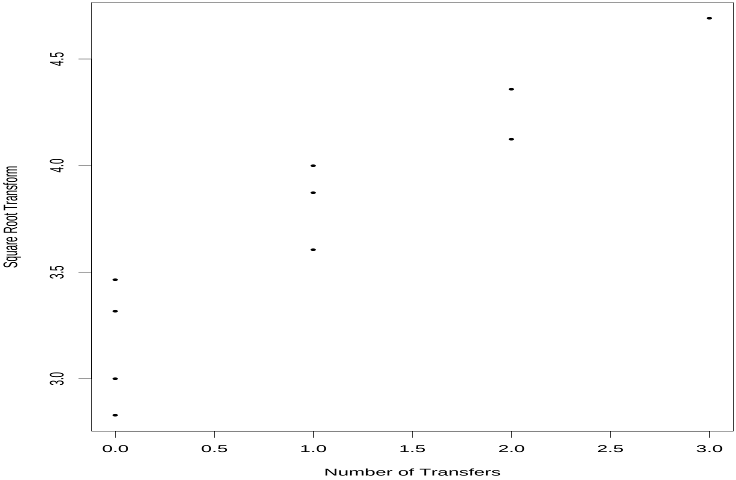
Modelling

- A reasonable model is that Y_i has a Poisson distribution with mean μ_i which depends in some way on X_i.
- We fit 3 models:
  1. The ordinary linear regression model
  $$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$
  2. The transformed regression model
  $$\sqrt{Y_i} = \beta_0 + \beta_1 X_i + \epsilon_i$$
  3. The Poisson regression model in which Y_i has a Poisson(μ_i) distribution and
  $$\log \mu_i = \beta_0 + \beta_1 X_i$$
  or equivalently
  $$\mu_i = \exp(\beta_0 + \beta_1 X_i)$$

Plot of Y versus x



Plot of $\sqrt{Y}$ versus x



SPlus Code

```
dat <- read.table("data", header = T)
attach(dat)
postscript("xyplot.ps",onefile=F,horizontal=F)
plot(Transfers,Broken,
     xlab="Number of Transfers",
     ylab="Number Broken")
dev.off()
postscript("xrootyplot.ps",onefile=F,horizontal=F)
plot(Transfers, sqrt(Broken),
     xlab="Number of Transfers",
     ylab="Square Root Transform")
dev.off()
#
#   Regress Number Broken on Number of Transfers
#
linfit <- lm( Broken ~ Transfers, data=dat)
```

SPlus Code — Continued

```
summary(linfit)
diag(linfit)
#
#   Regress Square Root of Number Broken
#     on Number of Transfers
#
rootlinfit <- lm( sqrt(Broken) ~ Transfers, data=dat)
summary(rootlinfit)
diag(rootlinfit)
#
#   The following fits log(E(Y)) is a linear function
#   of Dose and variance is equal to the mean
#
glmfit <- glm( Broken ~ Transfers, data=dat,
     family=Poisson)
summary(glmfit)
```

SPlus Code — Continued

```
postscript("points_plus_curve.ps",
            onefile=F,horizontal=F)
plot(Transfers,Broken,
 xlab="Number of Transfers",ylab="Number Broken")
d <- seq(0,4,length=200)
etalin <- coef(linfit)[1] + d*coef(linfit)[2]
lines(d,etalin)
etarootlin <- coef(rootlinfit)[1]
              + d*coef(rootlinfit)[2]
lines(d,etarootlin^2,lty=2)
etaglm <- coef(glmfit)[1] + d*coef(glmfit)[2]
p  <- exp(etaglm)
lines(d,p,lty=3)
legend(0,20,lty=1:3,
        legend=c("OLS","OLS on Root Y","GLM"))
dev.off()
```

SPlus Output — Edited

```
> summary(linfit)
Call: lm(formula=Broken~Transfers,data=dat)
Residuals:
  Min   1Q Median  3Q Max
 -2.2 -1.2    0.3 0.8 1.8
Coefficients:
             Value  Std. Err t value Pr(>|t|)
(Intercept) 10.2000  0.6633  15.3771  0.0000
  Transfers  4.0000  0.4690   8.5280  0.0000
Residual standard error: 1.483 on 8 df
Multiple R-Squared: 0.9009
F-statistic: 72.73 on 1 and 8 df
     the p-value is 2.749e-05
Correlation of Coefficients:
          (Intercept)
Transfers -0.7071
```

```
> summary(rootlinfit)
Call: lm(formula = sqrt(Broken)~Transfers,
           data = dat)
Residuals:
     Min      1Q  Median     3Q   Max
 -0.3722 -0.1263 0.01059 0.1392 0.274
Coefficients:
            Value  SE       t    Pr(>|t|)
(Intercept) 3.2006 0.1010 31.679  0.0000
  Transfers 0.5254 0.0714  7.354  0.0001
Residual standard error: 0.2259 on 8 df
Multiple R-Squared: 0.8711
F-statistic: 54.08 on 1 and 8 df
     the p-value is 7.965e-05
Correlation of Coefficients:
          (Intercept)
Transfers -0.7071
```

```
> summary(glmfit)

Call: glm(formula = Broken ~ Transfers,
     family = poisson, data = dat)
Deviance Residuals:
   Min        1Q    Median    3Q        Max
-0.81053 -0.23893 -0.02029 0.32991 0.60742

Coefficients:
                Value Std. Error   t value
(Intercept) 2.3529495  0.1317376 17.860883
  Transfers 0.2638422  0.0792345  3.329891
```

SPlus Output — Continued

```
(Dispersion Parameter for Poisson
       family taken to be 1 )

    Null Deviance: 12.56868 on 9 df

Residual Deviance: 1.813176 on 8 df

Number of Fisher Scoring Iterations: 3

Correlation of Coefficients:
          (Intercept)
Transfers -0.770864
```

The fits together