Problems with the Error Solutions

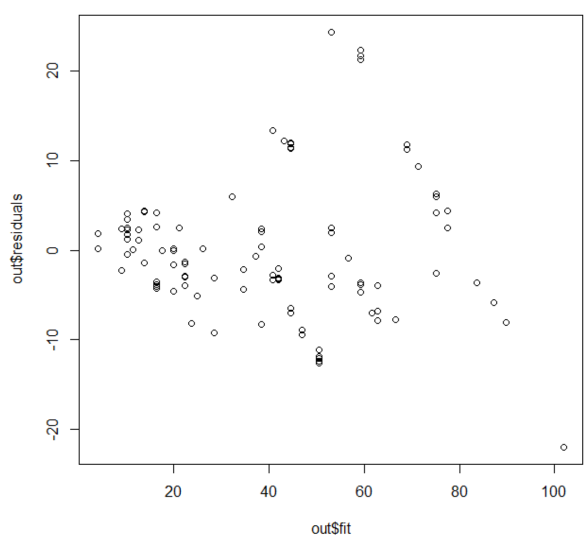
1. Here we go…

(a) With R,

```
> out <- lm(Lab ~ Field, pipeline)
> summary(out)

Call:
lm(formula = Lab ~ Field, data = pipeline)

Residuals:
    Min      1Q  Median      3Q     Max 
-21.985  -4.072  -1.431   2.504  24.334 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.96750    1.57479  -1.249    0.214    
Field        1.22297    0.04107  29.778   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.865 on 105 degrees of freedom
Multiple R-squared:  0.8941,    Adjusted R-squared:  0.8931 
F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

By non-constant variance, Faraway means non-constant variance of the residuals:

```
> plot(out$residuals ~ out$fit)
```



Holy $#@! This certainly looks like a non-constant variance situation. Just for fun, I checked for serial correlation of the residuals and got nothing:

```
> cor(residuals(out)[-1], residuals(out)[-length(residuals(out))])
[1] 0.08754773
```

(b) Following Faraway's instructions…

```
> i <- order(pipeline$Field)  ##  This extracts the row numbers from the
> npipe <- pipeline[i,]   ###### pipeline data and puts them in order of increasing
> ff <- gl(12, 9)[-108]   ###### values of the Field variable value
> meanfield <- unlist(lapply(split(npipe$Field, ff), mean))
> varlab <- unlist(lapply(split(npipe$Lab, ff), var))
> varlab
        1          2          3          4          5          6          7
13.611111   9.034444  14.848611   8.670000  11.528611  17.193611  78.442500
        8          9         10         11         12
63.721111 131.972500 188.493611 156.602500   9.184107
```

This is certainly consistent with our residuals vs. fits plot from before; it seems the variance increases in the Field variable.

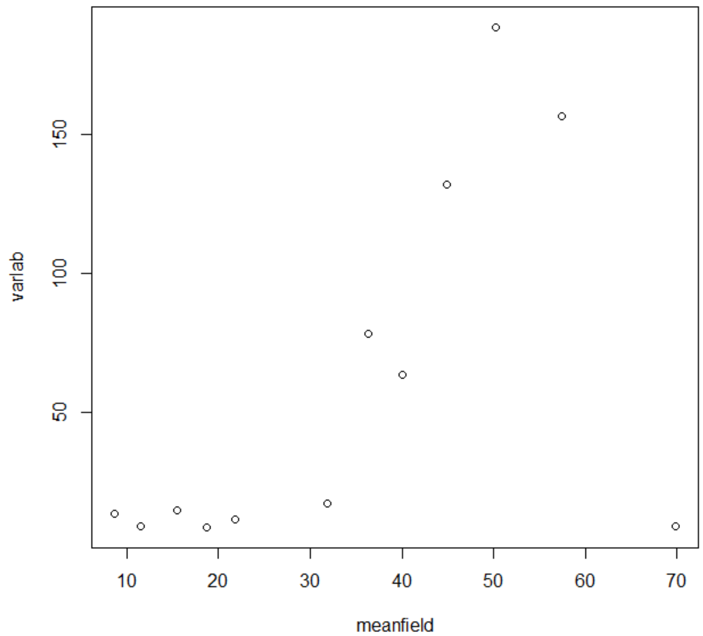
Next, we are to assume that var(Lab) = a_0 Field ^ a_1. Taking logs of both sides gives

$$\log(\text{var(Lab)}) = \log(a_0) + a_1 * \log(\text{Field}).$$

Then we can regress like Faraway suggests using simple linear regression to estimate a_0 and a_1:

```
> logs.reg <- lm(log(varlab) ~ log(meanfield))
> summary(logs.reg)

Call:
lm(formula = log(varlab) ~ log(meanfield))

Residuals:
    Min      1Q  Median      3Q     Max
-2.2038 -0.6729  0.1656  0.7205  1.1891

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.3538     1.5715  -0.225   0.8264
log(meanfield)   1.1244     0.4617   2.435   0.0351 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.018 on 10 degrees of freedom
Multiple R-squared:  0.3723,    Adjusted R-squared:  0.3095
F-statistic: 5.931 on 1 and 10 DF,  p-value: 0.03513
```

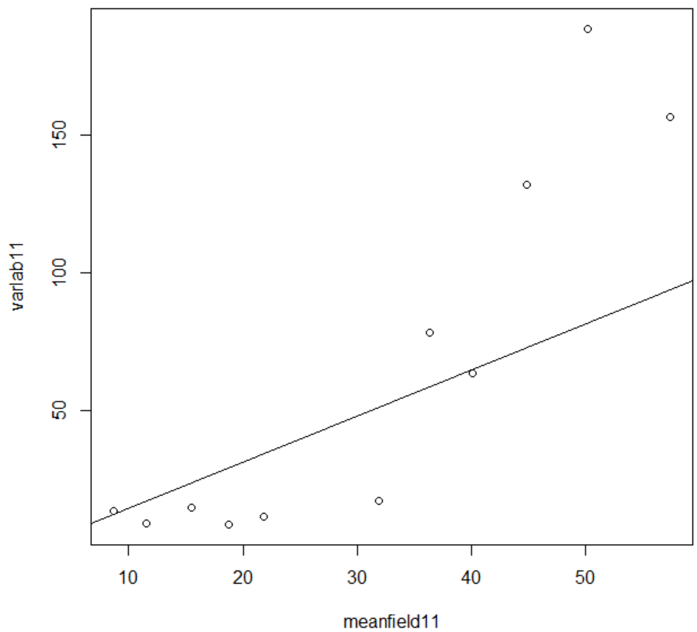
The output suggests estimating a_0 with e ^ -.3538 = .7020 and estimating a_1 with 1.1244. I kept the last point. However, as Faraway suggests, you ~~might~~ ***probably*** want to exclude it because



So I'll throw out the 12th point and redo…

```
> varlab <- data.frame(varlab)
> meanfield <- data.frame(meanfield)
> varlab11 <- varlab[1:11,]
> meanfield11 <- meanfield[1:11,]
> logs11.reg <- lm(log(varlab11) ~ log(meanfield11))
> summary(logs11.reg)

Call:
lm(formula = log(varlab11) ~ log(meanfield11))

Residuals:
     Min       1Q   Median       3Q      Max 
-1.00477 -0.42268  0.05989  0.37854  0.93815 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.9352     1.0929  -1.771 0.110403
log(meanfield11)   1.6707     0.3296   5.070 0.000672 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.657 on 9 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7118
F-statistic:  25.7 on 1 and 9 DF,  p-value: 0.0006723


> plot(varlab11 ~ meanfield11, xlab="meanfield11", ylab="varlab11")
> abline(logs11.reg)
>
```



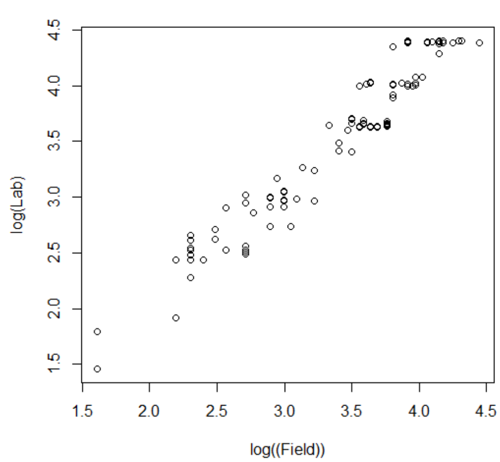
Even though the trend does not appear linear, this clearly gives a better approximation for the a_0 and a_1: a_0 is approximately e^-1.9352 = 0.1444 and a_1 is approximately 1.6707. Now it seems appropriate to set the WLS regression weights to [a_0 * Field ^ a_1] ^ -1:

```
> a0 = .1444
> a1 = 1.6707
> w <- 1/(a0 * pipeline$Field ^ a1)
> wlmod <- lm(Lab ~ Field, data = pipeline, weights=w)
> summary(wlmod)

Call:
lm(formula = Lab ~ Field, data = pipeline, weights = w)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-1.7433 -0.6719 -0.2493  0.5967  2.7277

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.05531    0.69766  -1.513    0.133
Field        1.18963    0.03401  34.984   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9846 on 105 degrees of freedom
Multiple R-squared:  0.921,    Adjusted R-squared:  0.9202
F-statistic:  1224 on 1 and 105 DF,  p-value: < 2.2e-16
```

(c) Out of all of these, I think taking logs of both variables worked out nicely:

```
> outloglog <- lm(log(Lab) ~ log((Field)))
> summary(outloglog)

Call:
lm(formula = log(Lab) ~ log((Field)))

Residuals:
     Min       1Q   Median       3Q      Max
-0.40212 -0.11853 -0.03092  0.13424  0.40209

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.06849    0.09305  -0.736    0.463
log((Field))  1.05483    0.02743  38.457   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1837 on 105 degrees of freedom
Multiple R-squared:  0.9337,    Adjusted R-squared:  0.9331
F-statistic:  1479 on 1 and 105 DF,  p-value: < 2.2e-16

> plot(log(Lab) ~ log(Field))
```



2. Here we go!

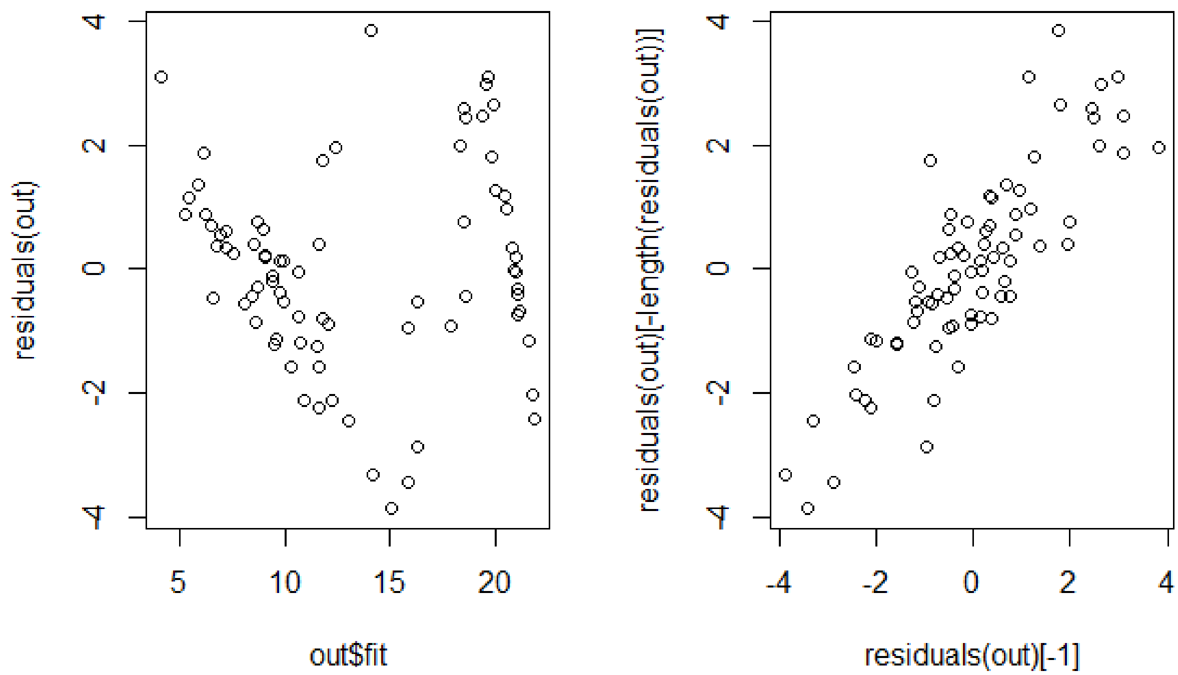
```
> attach(divusa)
> out <- lm(divorce ~ unemployed + femlab + marriage + birth + military)
> summary(out)

Call:
lm(formula = divorce ~ unemployed + femlab + marriage + birth +
    military)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8611 -0.8916 -0.0496  0.8650  3.8300

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.48784    3.39378   0.733   0.4659
unemployed  -0.11125    0.05592  -1.989   0.0505 .
femlab       0.38365    0.03059  12.543  < 2e-16 ***
marriage     0.11867    0.02441   4.861 6.77e-06 ***
birth       -0.12996    0.01560  -8.333 4.03e-12 ***
military    -0.02673    0.01425  -1.876   0.0647 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.65 on 71 degrees of freedom
Multiple R-squared:  0.9208,    Adjusted R-squared:  0.9152
F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

Checking for serial correlation…

```
> cor(residuals(out)[-1], residuals(out)[-length(residuals(out))])
[1] 0.8469792
```

There certainly seems to be some serial correlation. Let's look at a couple of plots.

```
> par(mfrow=c(1,2))
> plot(residuals(out) ~ out$fit)
> plot(residuals(out)[-1], residuals(out)[-length(residuals(out))])
```



The first plot on the left indicates a sort-of down, up, down pattern of the residuals with the fits. The second plot indicates serial correlation in the residuals. In fact, I computed this correlation previously to be 0.8469792 (see above).

(b)
```
> glmod <- gls(divorce ~ unemployed + femlab + marriage + birth + military, method= "ML",
correlation=corAR1(form= ~year), data = na.omit(divusa))
> summary(glmod)
Generalized least squares fit by maximum likelihood
  Model: divorce ~ unemployed + femlab + marriage + birth + military
  Data: na.omit(divusa)
       AIC      BIC    logLik
  179.9523 198.7027 -81.97613

Correlation Structure: AR(1)
 Formula: ~year
 Parameter estimate(s):
      Phi
0.9715486

Coefficients:
                Value Std.Error   t-value p-value
(Intercept) -7.059682  5.547193 -1.272658  0.2073
unemployed   0.107643  0.045915  2.344395  0.0219
```

```
femlab      0.312085  0.095151  3.279878  0.0016
marriage    0.164326  0.022897  7.176766  0.0000
birth      -0.049909  0.022012 -2.267345  0.0264
military    0.017946  0.014271  1.257544  0.2127

 Correlation: 
          (Intr) unmply femlab marrig birth 
unemployed -0.420                           
femlab     -0.802  0.240                    
marriage   -0.516  0.607  0.307             
birth      -0.379  0.041  0.066 -0.094      
military   -0.036  0.436 -0.311  0.530  0.128

Standardized residuals:
       Min         Q1        Med         Q3        Max 
-1.4509327 -0.9760939 -0.6164694  1.1375377  2.1593261 

Residual standard error: 2.907665 
Degrees of freedom: 77 total; 71 residual
> intervals(glmod, which="var-cov")
Approximate 95% confidence intervals

 Correlation structure:
        lower      est.     upper
Phi 0.6528097 0.9715486 0.9980192
attr(,"label")
[1] "Correlation structure:"

 Residual standard error:
     lower       est.      upper 
 0.7974404  2.9076645 10.6020628 
```

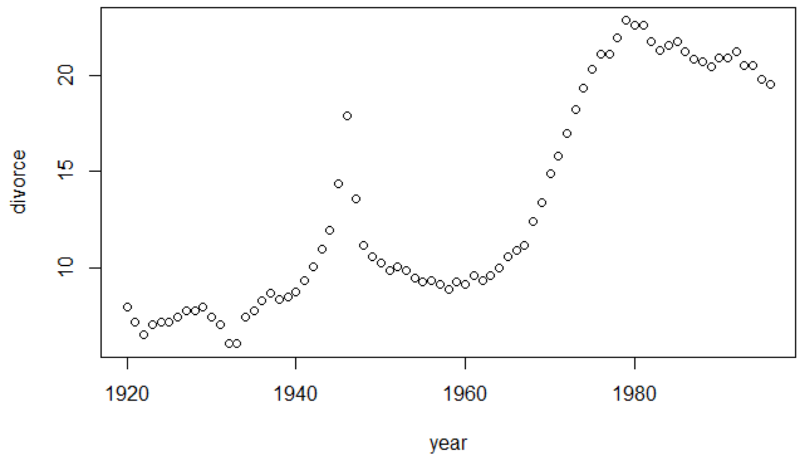
So it seems the phi value is very likely greater than 0, and probably around .97. That is, there is a high degree of serial correlation in the residuals.

The GLS model is pretty consistent with the lm() model. All the predictors are statistically significant with the possible exception of military. The intercept is likely not needed also.

(c) Why might there be correlation in the errors? Check this out:

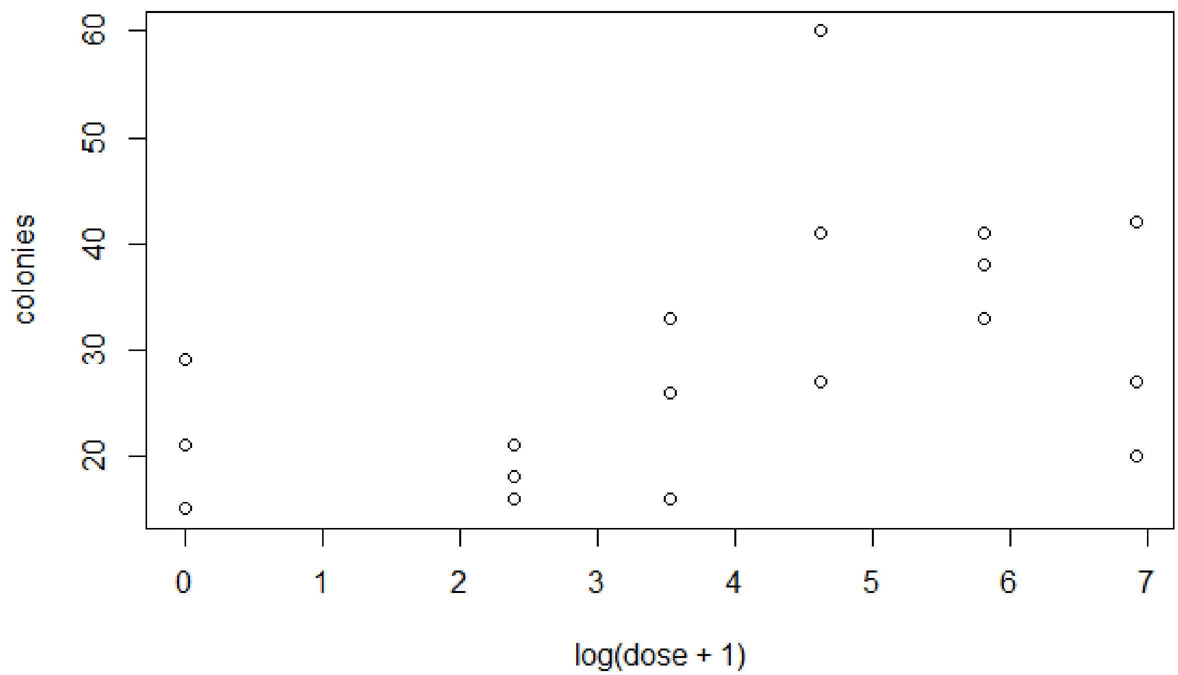
```
> plot(divorce ~ year)
```

Note the divorce rate shot up in the mid-1940s, perhaps having to do with husbands going away to war. It then fell back down through the 1950s, and began increasing again in the 1960s, when we know that divorce began to be more common.


divorce
10
15
20
1920
1940
1960
1980
year

3. Salmonella!

```
> attach(salmonella)
> out <- lm(colonies ~ log(dose+1))
> summary(out)

Call:
lm(formula = colonies ~ log(dose + 1))

Residuals:
    Min      1Q  Median      3Q     Max 
-16.376  -6.882  -1.509   5.400  29.119 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)   
(Intercept)     19.823      5.064   3.915  0.00123 **
log(dose + 1)    2.396      1.128   2.125  0.04955 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.84 on 16 degrees of freedom
Multiple R-squared:  0.2201,    Adjusted R-squared:  0.1713 
F-statistic: 4.514 on 1 and 16 DF,  p-value: 0.04955

> plot(colonies ~ log(dose+1))
```



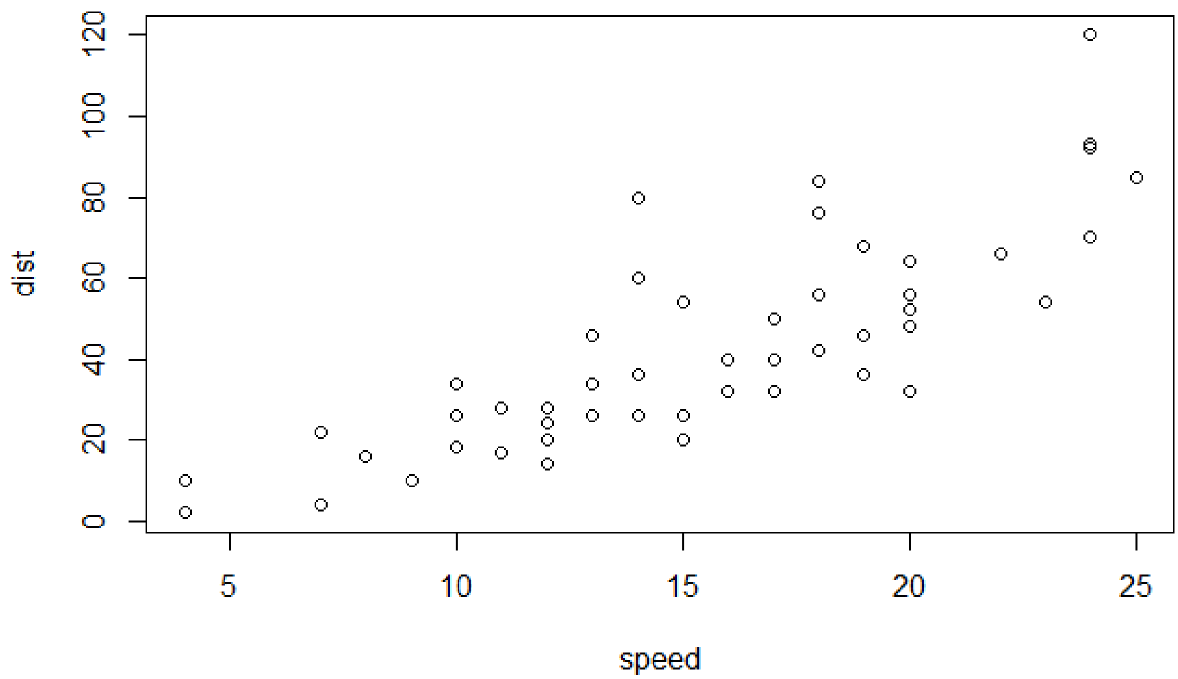
There could be a lack of linear fit here. Let's see:

```
> pureErrorAnova(out)
Analysis of Variance Table

Response: colonies
             Df  Sum Sq Mean Sq F value  Pr(>F)
log(dose + 1)  1  530.71  530.71  5.8356 0.03257 *
Residuals     16 1881.06  117.57
 Lack of fit   4  789.73  197.43  2.1709 0.13420
 Pure Error   12 1091.33   90.94
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lack-of-linear fit p-value is .13420, and so I would conclude there is insufficient evidence to conclude there exists a lack of linear fit.

4. Cars!!!

```
> out <- lm(dist ~ speed)
> plot(dist ~ speed)
```




```
> summary(out)

Call:
lm(formula = dist ~ speed)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.069  -9.525  -2.272   9.215  43.201 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -17.5791     6.7584  -2.601   0.0123 *  
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438
```

```
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

> pureErrorAnova(out)
Analysis of Variance Table

Response: dist
          Df  Sum Sq Mean Sq F value    Pr(>F)
speed        1 21185.5 21185.5 97.0836 4.558e-11 ***
Residuals    48 11353.5   236.5
 Lack of fit 17  4588.7   269.9  1.2369    0.2948
 Pure Error  31  6764.8   218.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

It doesn’t appear that there is much statistical evidence pointing to lack of linear fit.

5. Here is the least-squares output:

```
> lsout <- lm(stack.loss ~ ., stackloss)
> summary(lsout)

Call:
lm(formula = stack.loss ~ ., data = stackloss)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2377 -1.7117 -0.4551  2.3614  5.6978

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -39.9197    11.8960  -3.356  0.00375 **
Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
Water.Temp    1.2953     0.3680   3.520  0.00263 **
Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.243 on 17 degrees of freedom
Multiple R-squared:  0.9136,    Adjusted R-squared:  0.8983
F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

Next is the LAD output:

```
> require(quantreg)
> LADout <- rq(stack.loss ~., data = stackloss)
> summary(LADout)

Call: rq(formula = stack.loss ~ ., data = stackloss)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept) -39.68986    -41.61973 -29.67754
Air.Flow      0.83188      0.51278   1.14117
Water.Temp    0.57391      0.32182   1.41090
Acid.Conc.   -0.06087     -0.21348  -0.02891
```

Next is the output from the Huber method:

```
> require(MASS)
> Huberout <- rlm(stack.loss ~ ., data = stackloss)
> summary(Huberout)
```

```
Call: rlm(formula = stack.loss ~ ., data = stackloss)
Residuals:
     Min       1Q   Median       3Q      Max
-8.91753 -1.73127  0.06187  1.54306  6.50163

Coefficients:require(MASS)

            Value    Std. Error t value
(Intercept) -41.0265   9.8073    -4.1832
Air.Flow      0.8294   0.1112     7.4597
Water.Temp    0.9261   0.3034     3.0524
Acid.Conc.   -0.1278   0.1289    -0.9922

Residual standard error: 2.441 on 17 degrees of freedom
```

Finally, below is the output from the least trimmed squares method. I did an exhaustive search since the data set is not that large:

```
> LTSout <- ltsreg(stack.loss ~ ., data = stackloss, nsamp="exact")
> summary(LTSout)
              Length Class      Mode
crit           1     -none-     numeric
sing           1     -none-     character
coefficients   4     -none-     numeric
bestone        4     -none-     numeric
fitted.values 21     -none-     numeric
residuals     21     -none-     numeric
scale          2     -none-     numeric
terms          3     terms      call
call           5     -none-     call
xlevels        0     -none-     list
model          4     data.frame list
> coef(LTSout)
  (Intercept)      Air.Flow    Water.Temp    Acid.Conc.
-3.580556e+01  7.500000e-01  3.333333e-01  3.489094e-17
```

The Acid.Conc variable doesn’t seem very important.