Stat 511 Final Exam

May 5, 2004 **Prof. Vardeman**

1. In a study meant to determine the variability of diameters of widgets produced on a manufacturing line, an engineer measures $m = 10$ widgets produced on the line once each. Then the engineer measures the diameter of an 11^{th} widget $n = 8$ times. Suppose one models a measured widget diameter, y, as

$$y = x + \varepsilon$$

where x is the true diameter of the particular widget and ε is measurement error, for $x \sim \text{N}\left(0, \sigma_x^2\right)$ independent of $\varepsilon \sim \text{N}\left(0, \sigma^2\right)$. With s_y^2 the sample variance of the measurements on the 10 widgets, $\text{E}s_y^2 = \sigma_x^2 + \sigma^2$, and with s^2 the sample variance of the repeat measurements on the 11^{th} widget, $\text{E}\,s^2 = \sigma^2$. If the engineer observes $s_y = .05 \text{ mm}$ and $s = .01 \text{ mm}$, find approximate 90% confidence limits for σ_x. (Hint: Cochran-Satterthwaite.)

2. Printout #1 concerns the analysis of data taken from an article of Chowdhury and Mitra on a study intended to reduce defects produced by a wave soldering machine in an electronics plant. Numbers of "shorts" and numbers of "dry solder" defects were counted on a standard unit of product, as levels of 10 two-level factors (that we here simply call A through J) were changed. The data on the printout concern counts for 32 (out of $2^{10} = 1024$ possible) combinations of levels of these and a GLM analysis using a Poisson model and the canonical link function $h(\mu) = \ln(\mu)$.

a) The first two models fit (to "shorts" and "dry solder") include main effects for the 10 two-level factors A through J. Why would it be hopeless to try to fit a model with main effects and all 2 factor interactions?

Based on the results for the first two models, the smaller 3^{rd} and 4^{th} models were fit. Use these smaller models to answer the rest of the questions about this scenario.

b) Which factor seems to have the biggest effect on shorts? Which factor seems to have the biggest effect on dry solder defects?

biggest effect on shorts:__________ biggest effect on dry solder defects:__________

c) For the 3rd model (the smaller model fit to “shorts”) the first fitted/predicted value is 7.968834. Show how this is obtained from the vector of estimated coefficients $\boldsymbol{\beta}$. (Show calculation of this value.)

d) Some R code (and what is returned by the program) in this context is below. Use the result and show why a sensible `se.fit` for the first fitted value is 0.9388530. (Hints: “delta method” and “link function.”)

```
> t(c(1,-1,1,-1))%*%vcov(glm.out3)%*%c(1,-1,1,-1)

            [,1]
 [1,] 0.01388052
```

e) As it turns out, only factors A through I were factors whose levels could be set on the wave soldering machine. (Factor J was a factor whose levels fluctuated beyond the control of the process operators.) What levels of Factors C,D,E, and H do you recommend for future operation of the machine, if one wishes to minimize the total of shorts and dry solder defects at the worst level of J? What do you estimate to be the mean total under the conditions you recommend?

C level __________ D level __________ E level __________ H level __________

estimated mean $shorts + dry\ solder\ defects$ at the worst level of J ___________________________

3. Several nominally identical bolts are used to hold face-plates on a model of transmission manufactured by an industrial concern. Some testing was done to determine the torque required to loosen bolts number 3 and 4 on 34 transmissions. Since the bolts are tightened simultaneously by two heads of a pneumatic wrench fed from a single compressed air line, it is natural to expect the torques to be correlated. Printout #2 concerns the estimation of the correlation based on these 34 pairs.

a) Give a point estimate of the “population” correlation between Bolt 3 torque and Bolt 4 torque, a bootstrap standard error for that estimate, and a “bias-corrected” version of the estimate.

estimate __________ standard error __________ bias-corrected estimate __________

b) What are 90% bootstrap percentile confidence limits for the population correlation? (Report two numbers.)

lower limit __________ upper limit __________

c) As it turns out, the "acceleration factor" for the bootstrap samples represented on the printout is $\hat{a} = -.02290961$. The lower limit of a 90% BC_a confidence interval for the population correlation is at approximately which percentile of the bootstrapped correlations? (Show your work.)

4. Printout #3 concerns the analysis of some very old data of Rumford taken to study the cooling of a large cannon barrel that had been heated to a uniform (high) temperature by friction. Newton's Law of cooling predicted that when the source of heating was removed, the temperature would decline exponentially to the ambient temperature (that was apparently about 60°F on the day these data were collected). The model fit to the temperatures (in degrees F) and times (in minutes from the cessation of heating) is

$$y_i = \alpha + \beta \exp\left(-\theta t_i\right) + \varepsilon_i$$

a) Give approximate 90% prediction limits for an unrecorded temperature at 45 minutes based on this model. (Plug numbers in everywhere, but you need not simplify.)

b) Notice from the plot on the printout, that over the first 45 minutes or so of cooling the fitted curve does a nice job of tracking the plotted points. There is, however, clear statistical evidence that this trend can not continue. Explain carefully/statistically. (The real temperature series had to eventually decline to about 60°F … is that consistent with what has thus far been observed?)

5. Printout #4 concerns the analysis of some data from a test of thermal battery lives. Batteries using 4 different Electrolytes were tested at 3 different Temperatures, in 9 different Batches/Tests, over a period of 3 Days (3 Batches per day). Each batch/test was run at a single temperature and included one battery of each electrolyte. Let

y_{ij} = the life of the battery with electrolyte i in test/batch j

$d(j)$ = the number of the day on which test/batch j is run

$t(j)$ = the level of temperature at which test/batch j is run

Consider fixed effects $\mu, \eta_i, \tau_t, \eta\tau_{it}$ (say, satisfying the sum restrictions) where the η's represent Electrolyte main effects, the τ's represent Temperature main effects, and the $\eta\tau$'s the Electrolyte by Temperature interactions. Suppose that $\beta_1, \beta_2, \ldots, \beta_9$ are iid $\mathrm{N}(0, \sigma_\beta^2)$, independent of $\delta_1, \delta_2, \delta_3$ that are iid $\mathrm{N}(0, \sigma_\delta^2)$, independent of ε_{ij}'s that are iid $\mathrm{N}(0, \sigma^2)$. We will work under the model

$$y_{ij} = \mu + \eta_i + \tau_{t(j)} + \eta\tau_{it(j)} + \beta_j + \delta_{d(j)} + \varepsilon_{ij}$$

(In addition to the fixed effects, there are random day and batch/test effects, along with the basic experimental error, ε.)

a) Give approximate 95% confidence intervals for σ and σ_β.

for σ ________________________ for σ_β ________________________

b) How is it obvious from the printout that σ_δ is poorly determined? In retrospect, why should that be no surprise?

c) A bit of the printout is duplicated below. This gives 3 kinds of predictions for the first case in the data set. Show how to get each of these 3 predictions from other things on the printout. (Show sums that give these values.)

```
> predict(battlife,level=0:2)
   Day Test predict.fixed predict.Day predict.Test
1    1  1/1      1.890000    1.890073     2.128209
```

d) A 10th batch/test will be run tomorrow. Temperature 1 will be used. What do you predict for a battery life for the battery with electrolyte 1 and what is a standard error of prediction?

prediction ____________________ standard error of prediction _____________________

6. Miscellaneous Linear Models

a) Write a matrix $\mathbf{C}$ so that with $\boldsymbol{\beta} = (\mu_{11}, \mu_{12}, \mu_{13}, \mu_{21}, \mu_{22}, \mu_{23}, \mu_{31}, \mu_{32}, \mu_{33})$ the testable hypothesis $\mathrm{H}_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{0}$ for (the cell means version of) the 3×3 factorial is the hypothesis that $\alpha_1 = \alpha_2$ and simultaneously $\beta_1 = \beta_3$.

b) Carefully convince me using a simple "column space and projection" argument that simple linear regression for n points (x_i, y_i) gives the same the same predicted values as using the n points $(x_i - \overline{x}, y_i)$ (for $\overline{x}$ the sample mean of the x_i).

Printout #1

```
> options(contrasts=c("contr.sum","contr.sum"))
> getOption("contrasts")
[1] "contr.sum" "contr.sum"
> Data
   shorts drysolder AA BB CC DD EE FF GG HH II JJ
1       7        36  1  1  2  1  2  1  1  2  2  1
2       8        68  1  2  2  1  2  2  1  1  1  1
3      10        26  2  1  1  1  1  1  1  2  1  1
4      12        68  2  2  1  1  1  2  1  1  2  1
5      13        33  1  2  2  1  1  1  2  2  2  1
6       6        53  1  1  2  1  1  2  2  1  1  1
7       6        45  2  2  1  1  2  1  2  2  1  1
8       5        42  2  1  1  1  2  2  2  1  2  1
9       5        44  1  2  1  2  2  1  1  1  2  1
10     12        42  1  1  1  2  2  2  1  2  1  1
11     17        42  2  2  2  2  1  1  1  1  1  1
12     15        44  2  1  2  2  1  2  1  2  2  1
13     13        47  1  1  1  2  1  1  2  1  2  1
14     11        37  1  2  1  2  1  2  2  2  1  1
15     15        43  2  1  2  2  2  1  2  1  1  1
16     10        29  2  2  2  2  2  2  2  2  2  1
17     12        57  1  1  2  1  2  1  1  2  2  2
18     12        43  1  2  2  1  2  2  1  1  1  2
19      8        58  2  1  1  1  1  1  1  2  1  2
20      8        58  2  2  1  1  1  2  1  1  2  2
21      4        42  1  2  2  1  1  1  2  2  2  2
22      6        57  1  1  2  1  1  2  2  1  1  2
23     11        41  2  2  1  1  2  1  2  2  1  2
24      1        60  2  1  1  1  2  2  2  1  2  2
25      2        38  1  2  1  2  2  1  1  1  2  2
26     13        32  1  1  1  2  2  2  1  2  1  2
27     13        48  2  2  2  2  1  1  1  1  1  2
28     17        32  2  1  2  2  1  2  1  2  2  2
29     16        46  1  1  1  2  1  1  2  1  2  2
30     16        51  1  2  1  2  1  2  2  2  1  2
31     15        60  2  1  2  2  2  1  2  1  1  2
32     15        50  2  2  2  2  2  2  2  2  2  2
> glm.out1<-glm(shorts~1+AA+BB+CC+DD+EE+FF+GG+HH+II+JJ,family=poisson)
> summary(glm.out1)

Call:
glm(formula = shorts ~ 1 + AA + BB + CC + DD + EE + FF + GG +
    HH + II + JJ, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6370  -0.7082   0.0594   0.6169   1.8354

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.29921    0.05727  40.145  < 2e-16 ***
AA1         -0.04092    0.05697  -0.718   0.4726
BB1          0.02435    0.05494   0.443   0.6577
CC1         -0.10581    0.05692  -1.859   0.0630 .
DD1         -0.22890    0.05672  -4.036 5.45e-05 ***
EE1          0.10759    0.05516   1.950   0.0511 .
FF1          0.01564    0.05665   0.276   0.7824
GG1          0.01521    0.05534   0.275   0.7834
HH1         -0.09044    0.05697  -1.588   0.1124
II1          0.08194    0.05555   1.475   0.1401
JJ1         -0.01198    0.05472  -0.219   0.8268
```

```
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 68.867  on 31  degrees of freedom
Residual deviance: 38.403  on 21  degrees of freedom
AIC: 190.49

Number of Fisher Scoring iterations: 5

> glm.out2<-glm(drysolder~1+AA+BB+CC+DD+EE+FF+GG+HH+II+JJ,family=poisson)
> summary(glm.out2)

Call:
glm(formula = drysolder ~ 1 + AA + BB + CC + DD + EE + FF + GG +
    HH + II + JJ, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5852  -0.7493  -0.1851   0.8154   2.0756

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.8183218  0.0263279 145.029  < 2e-16 ***
AA1         -0.0142206  0.0261481  -0.544   0.5865
BB1         -0.0013831  0.0262236  -0.053   0.9579
CC1         -0.0007949  0.0262847  -0.030   0.9759
DD1          0.0657825  0.0262916   2.502   0.0123 *
EE1          0.0081552  0.0260676   0.313   0.7544
FF1         -0.0338992  0.0262943  -1.289   0.1973
GG1         -0.0002630  0.0262269  -0.010   0.9920
HH1          0.1082786  0.0262915   4.118 3.82e-05 ***
II1          0.0140835  0.0262437   0.537   0.5915
JJ1         -0.0503142  0.0260973  -1.928   0.0539 .
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 78.250  on 31  degrees of freedom
Residual deviance: 47.278  on 21  degrees of freedom
AIC: 249.86

Number of Fisher Scoring iterations: 4

> glm.out3<-glm(shorts~1+CC+DD+EE,family=poisson)
> summary(glm.out3)

Call:
glm(formula = shorts ~ 1 + CC + DD + EE, family = poisson)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.14357  -0.61397   0.05257   0.72906   1.63986

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.30714    0.05681  40.609  < 2e-16 ***
CC1         -0.10820    0.05504  -1.966   0.0493 *
DD1         -0.23160    0.05619  -4.122 3.76e-05 ***
EE1          0.10820    0.05504   1.966   0.0493 *
```

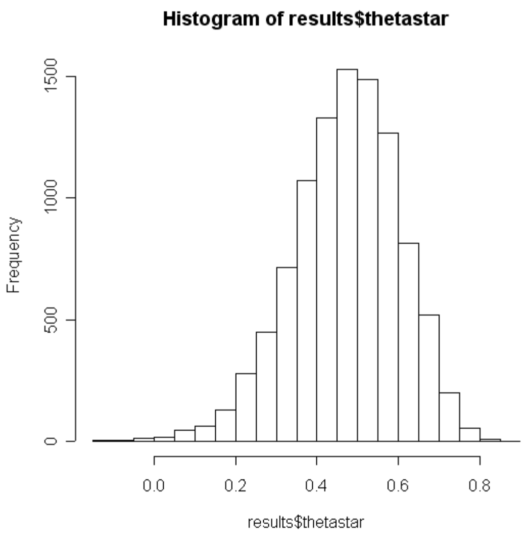
```
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 68.867  on 31  degrees of freedom
Residual deviance: 43.646  on 28  degrees of freedom
AIC: 181.74

Number of Fisher Scoring iterations: 4

> predict.glm(glm.out3,type="response",se.fit=TRUE)
$fit
 [1]  7.968834  7.968834  7.968834  7.968834  9.894190  9.894190  6.418142
 [8]  6.418142 10.199373 10.199373 15.723325 15.723325 12.663651 12.663651
[15] 12.663651 12.663651  7.968834  7.968834  7.968834  7.968834  9.894190
[22]  9.894190  6.418142  6.418142 10.199373 10.199373 15.723325 15.723325
[29] 12.663651 12.663651 12.663651 12.663651

$se.fit
        1         2         3         4         5         6         7         8 
0.9388530 0.9388530 0.9388533 0.9388533 1.1095008 1.1095008 0.7909284 0.7909284 
        9        10        11        12        13        14        15        16 
1.1317461 1.1317461 1.5486336 1.5486336 1.3285814 1.3285814 1.3285808 1.3285808 
       17        18        19        20        21        22        23        24 
0.9388530 0.9388530 0.9388533 0.9388533 1.1095008 1.1095008 0.7909284 0.7909284 
       25        26        27        28        29        30        31        32 
1.1317461 1.1317461 1.5486336 1.5486336 1.3285814 1.3285814 1.3285808 1.3285808 

$residual.scale
[1] 1

> vcov(glm.out3)
              (Intercept)           CC1           DD1           EE1
(Intercept)  0.0032278312  3.264778e-04  7.184481e-04 -3.264753e-04
CC1          0.0003264778  3.029169e-03 -7.864334e-20  2.849094e-19
DD1          0.0007184481 -7.864334e-20  3.157457e-03 -1.302888e-20
EE1         -0.0003264753  2.849094e-19 -1.302888e-20  3.029168e-03

> glm.out4<-glm(drysolder~1+DD+HH+JJ,family=poisson)
> summary(glm.out4)

Call:
glm(formula = drysolder ~ 1 + DD + HH + JJ, family = poisson)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5962  -0.9731  -0.1077   1.0208   2.1385  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  3.81888    0.02632 145.101  < 2e-16 ***
DD1          0.06940    0.02613   2.656   0.0079 ** 
HH1          0.11050    0.02622   4.214 2.51e-05 ***
JJ1         -0.05031    0.02610  -1.928   0.0539 .  
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 78.250  on 31  degrees of freedom
Residual deviance: 49.589  on 28  degrees of freedom
```

```
AIC: 238.17

Number of Fisher Scoring iterations: 4

> predict.glm(glm.out4,type="response",se.fit=TRUE)
$fit
 [1] 41.57360 51.85592 41.57360 51.85592 41.57360 51.85592 41.57360 51.85592
 [9] 45.13508 36.18540 45.13508 36.18540 45.13508 36.18540 45.13508 36.18540
[17] 45.97481 57.34568 45.97481 57.34568 45.97481 57.34568 45.97481 57.34568
[25] 49.91333 40.01619 49.91333 40.01619 49.91333 40.01619 49.91333 40.01619

$se.fit
       1        2        3        4        5        6        7        8 
2.226919 2.627083 2.226919 2.627083 2.226919 2.627083 2.226919 2.627083 
       9       10       11       12       13       14       15       16 
2.369383 2.001184 2.369383 2.001184 2.369383 2.001184 2.369383 2.001184 
      17       18       19       20       21       22       23       24 
2.403181 2.826626 2.403181 2.826626 2.403181 2.826626 2.403181 2.826626 
      25       26       27       28       29       30       31       32 
2.554280 2.162925 2.554280 2.162925 2.554280 2.162925 2.554280 2.162925 

$residual.scale
[1] 1

> vcov(glm.out4)
              (Intercept)           DD1           HH1          JJ1
(Intercept)  6.926758e-04 -4.730150e-05 -7.568183e-05 3.423852e-05
DD1         -4.730150e-05  6.826255e-04 -4.051033e-20 4.616354e-20
HH1         -7.568183e-05 -4.051033e-20  6.876769e-04 3.284898e-20
JJ1          3.423852e-05  4.616354e-20  3.284898e-20 6.810690e-04
```

Printout #2

```
> pairs<-data.frame(Bolt3,Bolt4)
> pairs
   Bolt3 Bolt4
1     16    16
2     15    16
3     15    17
4     15    16
5     20    20
6     19    16
7     19    20
8     17    19
9     15    15
10    11    15
11    17    19
12    18    17
13    18    14
14    15    15
15    18    17
16    15    17
17    18    20
18    15    14
19    17    17
20    14    16
21    17    18
22    19    16
23    19    18
24    19    20
25    15    15
26    12    15
27    18    20
```

```
28    13    18
29    14    18
30    18    18
31    18    14
32    15    13
33    16    17
34    16    16

> cor(Bolt3,Bolt4)
[1] 0.4701989
> B<-10000
> theta<-function(x,xdata)
+ {
+ cor(xdata[x,1],xdata[x,2])
+ }
> results<-bootstrap(1:34,B,theta,pairs)
> hist(results$thetastar)
```



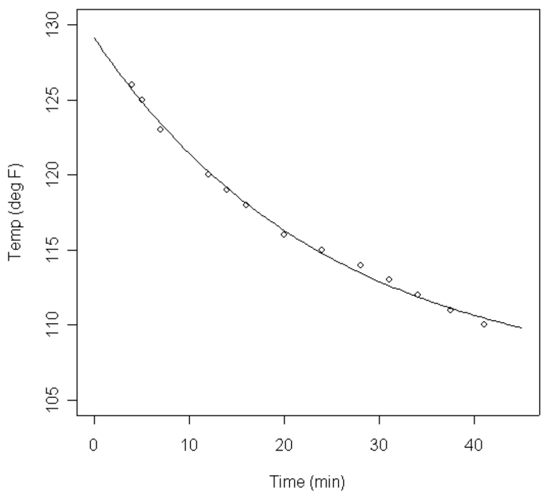

```
> mean(results$thetastar)
[1] 0.4711684
> sd(results$thetastar)
[1] 0.1314702
> quantile(results$thetastar,seq(0,1,.01))
        0%         1%         2%         3%         4%         5%         6% 
-0.1479573  0.1228274  0.1789409  0.2073450  0.2266627  0.2422192  0.2579794 
        7%         8%         9%        10%        11%        12%        13% 
 0.2708504  0.2809813  0.2907166  0.3002330  0.3079947  0.3160773  0.3238752 
       14%        15%        16%        17%        18%        19%        20% 
 0.3306443  0.3365023  0.3429050  0.3492155  0.3543904  0.3594075  0.3648174 
       21%        22%        23%        24%        25%        26%        27% 
 0.3700242  0.3750038  0.3797068  0.3842541  0.3882326  0.3928402  0.3966370 
       28%        29%        30%        31%        32%        33%        34% 
 0.4007967  0.4046489  0.4086629  0.4127554  0.4165107  0.4204644  0.4246587 
       35%        36%        37%        38%        39%        40%        41% 
 0.4283454  0.4313706  0.4347199  0.4378982  0.4419792  0.4462897  0.4493470 
       42%        43%        44%        45%        46%        47%        48% 
 0.4526056  0.4557312  0.4587347  0.4621118  0.4654594  0.4692503  0.4725353 
       49%        50%        51%        52%        53%        54%        55% 
 0.4755727  0.4787026  0.4820999  0.4855461  0.4890941  0.4919085  0.4950555 
       56%        57%        58%        59%        60%        61%        62% 
 0.4983513  0.5014691  0.5048364  0.5092485  0.5122951  0.5155306  0.5186822 
       63%        64%        65%        66%        67%        68%        69% 
 0.5217270  0.5254693  0.5282805  0.5319152  0.5350800  0.5383860  0.5423168 
```

```
      70%        71%        72%        73%        74%        75%        76% 
 0.5453013  0.5490111  0.5525533  0.5560782  0.5596810  0.5632250  0.5671693 
      77%        78%        79%        80%        81%        82%        83% 
 0.5717659  0.5756376  0.5798177  0.5834216  0.5876983  0.5918712  0.5961320 
      84%        85%        86%        87%        88%        89%        90% 
 0.5999433  0.6052300  0.6101329  0.6149759  0.6207049  0.6271412  0.6334970 
      91%        92%        93%        94%        95%        96%        97% 
 0.6411559  0.6488240  0.6571211  0.6657643  0.6732457  0.6839240  0.6948381 
      98%        99%       100% 
 0.7113090  0.7358604  0.8573627 
```

Printout #3

```
> t<-c(4,5,7,12,14,16,20,24,28,31,34,37.5,41)
> temp<-c(126,125,123,120,119,118,116,115,114,113,112,111,110)
> Rumford<-nls(formula=temp~(alpha+beta*exp(-
theta*t)),start=c(alpha=60,beta=70,theta=.01),trace=T)
51.0652 :  60.00 70.00  0.01 
33.76196 :  76.34815000 53.47123570  0.01255895 
31.03799 :  85.66467810 44.00552086  0.01514456 
29.10424 :  91.36651192 38.18158725  0.01764716 
26.1677 :  95.07653002 34.37239869  0.02000404 
24.695 :  100.15213752  29.13675125   0.02436665 
21.46191 :  105.02937480  24.06448952   0.03169066 
2.875144 :  106.7628246  22.2563831   0.0405277 
1.168267 :  106.18878603  22.91968177   0.04093712 
1.168227 :  106.19477292  22.91624738   0.04095288 
1.168227 :  106.19503706  22.91607849   0.04095397 
> summary(Rumford)

Formula: temp ~ (alpha + beta * exp(-theta * t))

Parameters:
       Estimate Std. Error t value Pr(>|t|)    
alpha 1.062e+02  1.015e+00  104.63  < 2e-16 ***
beta  2.292e+01  7.663e-01   29.91 4.09e-11 ***
theta 4.095e-02  4.073e-03   10.05 1.51e-06 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 

Residual standard error: 0.3418 on 10 degrees of freedom

Correlation of Parameter Estimates:
        alpha    beta   
beta  -0.9205         
theta  0.9752 -0.8288 

> vcov(Rumford)
             alpha         beta         theta
alpha  1.030060123 -0.715903260  4.031725e-03
beta  -0.715903260  0.587193174 -2.587124e-03
theta  0.004031725 -0.002587124  1.659288e-05

> predict(Rumford)
 [1] 125.6485 124.8679 123.3994 120.2137 119.1112 118.0955 116.2973 114.7708
 [9] 113.4750 112.6333 111.8890 111.1286 110.4698

> x<-seq(0,45,.5)
> temphat<-coef(Rumford)[1]+coef(Rumford)[2]*exp(-coef(Rumford)[3]*x)

> plot(c(0,45),c(105,130),type="n",xlab="Time (min)",ylab="Temp (deg F)")
> points(t, temp)
```

```
> lines(x,temphat)
```



Printout #4

```
> options(contrasts=c("contr.sum","contr.sum"))
> getOption("contrasts")
[1] "contr.sum" "contr.sum"
> data
   Life TEMP ELECT DAY TEST
1  2.17    1     1   1    1
2  1.58    1     2   1    1
3  2.29    1     3   1    1
4  2.23    1     4   1    1
5  2.33    2     1   1    2
6  1.38    2     2   1    2
7  1.86    2     3   1    2
8  2.27    2     4   1    2
9  1.75    3     1   1    3
10 1.52    3     2   1    3
11 1.55    3     3   1    3
12 1.56    3     4   1    3
13 1.88    1     1   2    4
14 1.26    1     2   2    4
15 1.60    1     3   2    4
16 2.01    1     4   2    4
17 2.01    2     1   2    5
18 1.30    2     2   2    5
19 1.70    2     3   2    5
20 1.81    2     4   2    5
21 1.95    3     1   2    6
22 1.47    3     2   2    6
23 1.61    3     3   2    6
24 1.72    3     4   2    6
25 1.62    1     1   3    7
26 1.22    1     2   3    7
27 1.67    1     3   3    7
28 1.82    1     4   3    7
29 1.70    2     1   3    8
30 1.85    2     2   3    8
31 1.81    2     3   3    8
32 2.01    2     4   3    8
33 2.13    3     1   3    9
34 1.80    3     2   3    9
35 1.82    3     3   3    9
36 1.99    3     4   3    9
```

```
> battlife<-lme(Life~1+ELECT*TEMP,random=~1|Day/Test)
> summary(battlife)
Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  51.33544 69.00624 -10.66772

Random effects:
 Formula: ~1 | Day
        (Intercept)
StdDev: 0.003198544

 Formula: ~1 | Test %in% Day
        (Intercept)  Residual
StdDev:   0.1758407 0.1613474

Fixed effects: Life ~ 1 + ELECT * TEMP
                  Value  Std.Error DF   t-value p-value
(Intercept)   1.7847222 0.06451434 18 27.663964  0.0000
ELECT1        0.1641667 0.04657697 18  3.524632  0.0024
ELECT2       -0.2980556 0.04657697 18 -6.399205  0.0000
ELECT3       -0.0169444 0.04657697 18 -0.363794  0.7202
TEMP1        -0.0055556 0.09119966  4 -0.060916  0.9543
TEMP2         0.0511111 0.09119966  4  0.560431  0.6051
ELECT1:TEMP1 -0.0533333 0.06586978 18 -0.809678  0.4287
ELECT2:TEMP1 -0.1277778 0.06586978 18 -1.939854  0.0682
ELECT3:TEMP1  0.0911111 0.06586978 18  1.383200  0.1835
ELECT1:TEMP2  0.0133333 0.06586978 18  0.202420  0.8419
ELECT2:TEMP2 -0.0277778 0.06586978 18 -0.421707  0.6782
ELECT3:TEMP2 -0.0288889 0.06586978 18 -0.438576  0.6662
 Correlation:
             (Intr) ELECT1 ELECT2 ELECT3 TEMP1  TEMP2  ELECT1:TEMP1
ELECT1        0.000
ELECT2        0.000 -0.333
ELECT3        0.000 -0.333 -0.333
TEMP1         0.000  0.000  0.000  0.000
TEMP2         0.000  0.000  0.000  0.000 -0.500
ELECT1:TEMP1  0.000  0.000  0.000  0.000  0.000  0.000
ELECT2:TEMP1  0.000  0.000  0.000  0.000  0.000  0.000 -0.333
ELECT3:TEMP1  0.000  0.000  0.000  0.000  0.000  0.000 -0.333
ELECT1:TEMP2  0.000  0.000  0.000  0.000  0.000  0.000 -0.500
ELECT2:TEMP2  0.000  0.000  0.000  0.000  0.000  0.000  0.167
ELECT3:TEMP2  0.000  0.000  0.000  0.000  0.000  0.000  0.167
             ELECT2:TEMP1 ELECT3:TEMP1 ELECT1:TEMP2 ELECT2:TEMP2
ELECT1
ELECT2
ELECT3
TEMP1
TEMP2
ELECT1:TEMP1
ELECT2:TEMP1
ELECT3:TEMP1 -0.333
ELECT1:TEMP2  0.167        0.167
ELECT2:TEMP2 -0.500        0.167       -0.333
ELECT3:TEMP2  0.167       -0.500       -0.333       -0.333

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.97611566 -0.28950208  0.01316409  0.27366159  2.07311917

Number of Observations: 36
```

```
Number of Groups:
          Day Test %in% Day
            3             9
> intervals(battlife)
Approximate 95% confidence intervals

 Fixed effects:
                   lower         est.       upper
(Intercept)   1.64918263  1.784722222  1.92026181
ELECT1        0.06631208  0.164166667  0.26202125
ELECT2       -0.39591014 -0.298055556 -0.20020097
ELECT3       -0.11479903 -0.016944444  0.08091014
TEMP1        -0.25876641 -0.005555556  0.24765530
TEMP2        -0.20209975  0.051111111  0.30432197
ELECT1:TEMP1 -0.19172061 -0.053333333  0.08505394
ELECT2:TEMP1 -0.26616505 -0.127777778  0.01060950
ELECT3:TEMP1 -0.04727617  0.091111111  0.22949839
ELECT1:TEMP2 -0.12505394  0.013333333  0.15172061
ELECT2:TEMP2 -0.16616505 -0.027777778  0.11060950
ELECT3:TEMP2 -0.16727617 -0.028888889  0.10949839
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: Day
                       lower        est.        upper
sd((Intercept)) 1.083016e-28 0.003198544 9.446475e+22
  Level: Test
                     lower      est.     upper
sd((Intercept)) 0.08833263 0.1758407 0.3500401

 Within-group standard error:
    lower      est.     upper
0.1163846 0.1613474 0.2236805

> random.effects(battlife)
Level: Day
    (Intercept)
1  7.328656e-05
2 -7.487975e-05
3  1.593186e-06

Level: Test %in% Day
    (Intercept)
1/1  0.23813565
1/2  0.10251527
1/3 -0.11915864
2/4 -0.07566525
2/5 -0.10802138
2/6 -0.04262070
3/7 -0.16247039
3/8  0.00550611
3/9  0.16177933

> predict(battlife)
     1/1      1/1      1/1      1/1      1/2      1/2      1/2      1/2
2.128209 1.591542 2.091542 2.258209 2.115922 1.612589 1.892589 2.132589
     1/3      1/3      1/3      1/3      2/4      2/4      2/4      2/4
1.824248 1.477581 1.540915 1.637581 1.814260 1.277593 1.777593 1.944260
     2/5      2/5      2/5      2/5      2/6      2/6      2/6      2/6
1.905237 1.401904 1.681904 1.921904 1.900638 1.553971 1.617304 1.713971
     3/7      3/7      3/7      3/7      3/8      3/8      3/8      3/8
```

```
1.727531 1.190865 1.690865 1.857531 2.018841 1.515508 1.795508 2.035508
     3/9      3/9      3/9      3/9
2.105114 1.758448 1.821781 1.918448
attr(,"label")
[1] "Fitted values"

> predict(battlife,level=0:2)
   Day Test predict.fixed predict.Day predict.Test
1    1  1/1      1.890000    1.890073     2.128209
2    1  1/1      1.353333    1.353407     1.591542
3    1  1/1      1.853333    1.853407     2.091542
4    1  1/1      2.020000    2.020073     2.258209
5    1  1/2      2.013333    2.013407     2.115922
6    1  1/2      1.510000    1.510073     1.612589
7    1  1/2      1.790000    1.790073     1.892589
8    1  1/2      2.030000    2.030073     2.132589
9    1  1/3      1.943333    1.943407     1.824248
10   1  1/3      1.596667    1.596740     1.477581
11   1  1/3      1.660000    1.660073     1.540915
12   1  1/3      1.756667    1.756740     1.637581
13   2  2/4      1.890000    1.889925     1.814260
14   2  2/4      1.353333    1.353258     1.277593
15   2  2/4      1.853333    1.853258     1.777593
16   2  2/4      2.020000    2.019925     1.944260
17   2  2/5      2.013333    2.013258     1.905237
18   2  2/5      1.510000    1.509925     1.401904
19   2  2/5      1.790000    1.789925     1.681904
20   2  2/5      2.030000    2.029925     1.921904
21   2  2/6      1.943333    1.943258     1.900638
22   2  2/6      1.596667    1.596592     1.553971
23   2  2/6      1.660000    1.659925     1.617304
24   2  2/6      1.756667    1.756592     1.713971
25   3  3/7      1.890000    1.890002     1.727531
26   3  3/7      1.353333    1.353335     1.190865
27   3  3/7      1.853333    1.853335     1.690865
28   3  3/7      2.020000    2.020002     1.857531
29   3  3/8      2.013333    2.013335     2.018841
30   3  3/8      1.510000    1.510002     1.515508
31   3  3/8      1.790000    1.790002     1.795508
32   3  3/8      2.030000    2.030002     2.035508
33   3  3/9      1.943333    1.943335     2.105114
34   3  3/9      1.596667    1.596668     1.758448
35   3  3/9      1.660000    1.660002     1.821781
36   3  3/9      1.756667    1.756668     1.918448
```