Stat 511 Final Exam

May 8, 2003 Prof. Vardeman

1. A classic data set of Wishart (later analyzed by Ostle and then Stapleton) concerns weight gains of young pigs in a comparative feeding trial. 3 different foods were fed to pigs of both sexes. 5 male pigs and 5 female pigs were assigned to each food (making a total of 30 pigs in the study). (There were also 5 pens involved in this study, but that is a detail we shall initially ignore.) With

$y_{ijk}$ = the weight gain of the $k$th pig of sex $j$ fed food $i$

$x_{ijk}$ = the initial weight of the $k$th pig of sex $j$ fed food $i$

first consider the model

$$y_{ijk} = m + a_i + b_j + ab_{ij} + q x_{ijk} + e_{ijk} \qquad (*)$$

where constants $a_i, b_j, ab_{ij}$ for $i = 1,2,3$ and $j = 1,2$ are subject to the sum restrictions, $m$ and $q$ are constants, and the $e_{ijk}$ are iid $\mathrm{N}\left(0, s^2\right)$. (On **Printout #1** Sex is coded "1" for Male and "2" for Female.)

a) Under model (*) give 95% confidence limits for the standard deviation of weight gain for a pig of fixed sex and weight fed a fixed food. (Plug in numbers, but don't do arithmetic.)

b) In this model (that contains both continuous and categorical predictor variables for the balanced data set) what is the expected value of $\bar{y}_{1..} - \bar{y}_{2..}$ ? Is it $a_1 - a_2$ ?

c) Interpret (in words) $b_1 - b_2$. This quantity is estimable. Name, for example, a linear combination of $\bar{y}_{.1.}, \bar{y}_{.2.}, y_{111}$, and $y_{112}$ that has this expected value.

d) Make and interpret 95% confidence limits for $q$ .

e) Show how to compute 95% confidence limits for $a_1 - a_2$. (Plug in, but you don't need to do arithmetic.)

f) Find 95% prediction limits for the weight gain of an additional 38 lb male pig fed feed #1. (Plug in but you don't need to do arithmetic.)

As it turns out, the pigs in this study were raised in 5 pens (each of which had one pig of each Food$\times$Sex combination in it). With $g_1, g_2, g_3, g_4, g_5$ iid $\mathrm{N}\left(0, s_g^2\right)$ independent of the $e_{ijk}$ and

$$l(ijk) = \text{the pen number } (l, \text{ from 1 to 5}) \text{ for pig } ijk$$

consider the mixed model

$$y_{ijk} = m + a_i + b_j + ab_{ij} + q x_{ijk} + g_{l(ijk)} + e_{ijk} \qquad (**)$$

and an analysis of weight gain data based on it. (Pigs in the same pen share a random effect in (**).)

g) In the mixed effects model (**), weight gains for pigs in the same pen are correlated. What is this correlation (in terms of the model parameters)?

h) Give approximate 95% confidence limits for the standard deviations $s$ and $s_g$ in (**).

For $s$ :

For $s_g$ :

i) The quantity $m + a_1 + b_1 + ab_{11} + 38q$ is what the model (**) gives as the mean (across all possible pens) weight gain for a 38 lb male pig fed food 1. (Note that the first pig in the data set was a male of this size fed food 1.) Give an estimate of this quantity and show how to compute a standard error. Notice that there is an estimated variance-covariance matrix for the estimates of fixed effects printed on the output. You don't have to copy it onto this page. Just call it what it is called on the printout, and write an (otherwise numerical) expression involving it.

Estimate:

Standard Error:

j) If $g^*$ is a new random effect for a new pen and $e^*$ is a new random error associated with a new male pig that happens to weigh 38 lbs and is to be fed food 1, and these are independent of each other and all of the $e_{ijk}$ and $h_l$ for the data in hand, one might wish to predict $m + a_1 + b_1 + ab_{11} + 38q + g^* + e^*$. Give a sensible standard error of prediction for this. (If it's relevant, you may abbreviate your answer to part i) as "SE".)

2. A Ni-Cad battery manufacturer was interested in finding what set of process conditions produces the smallest fraction of cells with "shorts." For $2 \times 2 \times 2 = 8$ different process set-ups, counts were made of batteries with shorts. The sample sizes varied set-up to set-up (from 88 to 100). Factors and their levels in the study were

| | |
|---|---|
| A- Nylon Sleeve | 1-no vs 2-yes |
| B- Rolling Direction | 1-lower edge first vs 2-upper edge first |
| C- Rolling Order | 1-negative first vs 2-positive first |

**Printout #2** summarizes two (binomial) GLM analyses of the manufacturer's data. (The data matrix `M` has counts of shorts in column 1 and counts of nonshorts in column 2.) The first of the two analyses was done using a logit link and the second used a probit link.

a) Does it appear from these data and analyses that any of these 3 factors influence the rate of short production? Is it plausible that the differences between observed rates is "just noise"? Explain.

b) Notice that there were 4 of 8 different process set-ups which produced no observed shorts. Which of these would you recommend for future running of the production process? Why?

c) Give approximate 95% confidence limits for the rate of shorts associated with your choice from part b). Do the estimation first based on the logit, then based on the probit link. How much difference is there between the two set of limits?

3. Two different processes (laser drilling and electrical discharge machining) were used to make holes in miniature high precision metal parts. These were supposed to be at a $45^\circ$ angle with the surface of the part. Measured angles for 13 parts drilled by each process are analyzed on **Printout #3**.

a) Under the assumption that these are two independent samples from normal distributions of angles, standard facts about sample variances imply that

$$\left(\frac{12s^2_{\text{laser}}}{s^2_{\text{laser}}}\right) \text{ and } \left(\frac{12s^2_{\text{edm}}}{s^2_{\text{edm}}}\right)$$

are independent $c^2_{12}$ random variables. Use this fact and percentage points of the $F$ distribution to make 90% confidence limits for $s_{\text{laser}}/s_{\text{edm}}$ (which could be used to compare precisions of the two drilling methods). (Plug in, but you need not simplify.)

Examination of plots for the two samples suggests that normal distribution model assumptions may not be appropriate. So, consider instead an analysis of based on bootstrap ideas.

b) What is the value of a "bootstrap standard error" for the statistic $s_{\text{laser}}/s_{\text{edm}}$?

c) What value would you subtract from the observed ratio $s_{\text{laser}}/s_{\text{edm}}$ in order to try and correct for bias in estimating $s_{\text{laser}}/s_{\text{edm}}$?

d) What are (uncorrected) 90% percentile bootstrap confidence limits for $s_{\text{laser}}/s_{\text{edm}}$? Explain.

4. Fitting sinusoids to noisy data can be done via nonlinear least squares. **Printout #4** concerns doing just this. 41 data pairs $(x_i, y_i)$ have a sinusoidal pattern and R's `nle` has been used to fit the model

$$y_i = A\sin\left(k\left(x_i - x_0\right)\right) + e_i$$

to them by estimation of parameters $A, k, x_0$ (and $s$ ).

a) A different set of starting values not shown on the printout led to a fit with $\hat{A} = -2.76987, \hat{k} = 6.39168,$ and $\hat{x}_0 = 0.04980$ and $SSE = 20.94946$. This final error sum of squares is exactly as for the fit on Printout #4, while the estimates for $A$ and $x_0$ are quite different. This shows that the function that `nle` optimizes has multiple local optima. Why should this have been obvious before beginning in this situation?

(In light of the above point, in order to talk rationally about inference for parameters, we would really need to do something like require that $A > 0$ and $k > 0$ and put some restriction on $x_0$. We'll assume that has been done and that the algorithm has converged to a global optimum subject to these.)

b) What are approximate 95% confidence limits for $A\sin\left(k\left(1 - x_0\right)\right)$, the mean response at $x = 1$? (Plug in, but don't try to simplify.)

c) Give approximate 95% confidence limits for the period of the sinusoid, $\dfrac{k}{2p}$.

## Printout #1

```
> PEN<-as.factor(pen)
> FOOD<-as.factor(food)
> SEX<-as.factor<-(sex)
> options(contrasts=c("contr.sum","contr.sum"))
> pigs<-cbind(FOOD,SEX,y,x,PEN)
> pigs
      FOOD SEX     y  x PEN
 [1,]    1   1  9.52 38   1
 [2,]    1   2  9.94 48   1
 [3,]    2   1  8.51 39   1
 [4,]    2   2 10.00 48   1
 [5,]    3   1  9.11 48   1
 [6,]    3   2  9.75 48   1
 [7,]    1   1  8.21 35   2
 [8,]    1   2  9.48 32   2
 [9,]    2   1  9.95 38   2
[10,]    2   2  9.24 32   2
[11,]    3   1  8.50 37   2
[12,]    3   2  8.66 28   2
[13,]    1   1  9.32 41   3
[14,]    1   2  9.32 35   3
[15,]    2   1  8.43 46   3
[16,]    2   2  9.34 41   3
[17,]    3   1  8.90 42   3
[18,]    3   2  7.63 33   3
[19,]    1   1 10.56 48   4
[20,]    1   2 10.90 46   4
[21,]    2   1  8.86 40   4
[22,]    2   2  9.68 46   4
[23,]    3   1  9.51 42   4
[24,]    3   2 10.37 50   4
[25,]    1   1 10.42 43   5
[26,]    1   2  8.82 32   5
[27,]    2   1  9.20 40   5
[28,]    2   2  9.67 37   5
[29,]    3   1  8.76 40   5
[30,]    3   2  8.57 30   5

> lmout.1<-lm(y~1+FOOD*SEX+x,model=TRUE)
> summary(lmout.1)

Call:
lm(formula = y ~ 1 + FOOD * SEX + x, model = TRUE)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.00320 -0.29578 -0.06553  0.35820  1.15652 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.27343    0.69904   8.974 5.65e-09 ***
FOOD1        0.36734    0.14430   2.546 0.018071 *  
FOOD2       -0.06168    0.14458  -0.427 0.673605    
SEX1        -0.19844    0.10352  -1.917 0.067746 .  
x            0.07558    0.01725   4.383 0.000217 ***
FOOD1:SEX1   0.06474    0.14424   0.449 0.657761    
FOOD2:SEX1  -0.09201    0.14553  -0.632 0.533473    
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
Residual standard error: 0.5585 on 23 degrees of freedom
Multiple R-Squared: 0.5611,     Adjusted R-squared: 0.4466
F-statistic:   4.9 on 6 and 23 DF,  p-value: 0.002322
> anova(lmout.1)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
FOOD       2 2.2686  1.1343  3.6364 0.0424387 *
SEX        1 0.4344  0.4344  1.3926 0.2500268
x          1 6.3357  6.3357 20.3108 0.0001592 ***
FOOD:SEX   2 0.1321  0.0661  0.2118 0.8106986
Residuals 23 7.1745  0.3119
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

> predict.lm(lmout.1,interval = "confidence",level=.95)
         fit      lwr       upr
1   9.379249 8.851585  9.906914
2  10.402485 9.786501 11.018470
3   8.869066 8.349226  9.388906
4  10.130201 9.553178 10.707224
5   9.424618 8.862568  9.986668
6   9.766952 9.134976 10.398928
7   9.152499 8.593218  9.711780
8   9.193149 8.625331  9.760967
9   8.793483 8.268526  9.318439
10  8.920865 8.316267  9.525463
11  8.593199 8.048865  9.137533
12  8.255281 7.631413  8.879150
13  9.606000 9.089303 10.122697
14  9.419899 8.887480  9.952319
15  9.398151 8.846708  9.949594
16  9.601117 9.084371 10.117863
17  8.971117 8.454371  9.487863
18  8.633199 8.088865  9.177533
19 10.135085 9.561202 10.708967
20 10.251318 9.671084 10.831553
21  8.944650 8.427510  9.461790
22  9.979034 9.430043 10.528025
23  8.971117 8.454371  9.487863
24  9.918119 9.242535 10.593703
25  9.757167 9.235567 10.278767
26  9.193149 8.625331  9.760967
27  8.944650 8.427510  9.461790
28  9.298783 8.764597  9.832968
29  8.819950 8.299278  9.340621
30  8.406449 7.819584  8.993313
```

```
> X<-model.matrix(lmout.1)
> X
   (Intercept) FOOD1 FOOD2 SEX1  x FOOD1:SEX1 FOOD2:SEX1
1            1     1     0    1 38          1          0
2            1     1     0   -1 48         -1          0
3            1     0     1    1 39          0          1
4            1     0     1   -1 48          0         -1
5            1    -1    -1    1 48         -1         -1
6            1    -1    -1   -1 48          1          1
7            1     1     0    1 35          1          0
8            1     1     0   -1 32         -1          0
9            1     0     1    1 38          0          1
10           1     0     1   -1 32          0         -1
11           1    -1    -1    1 37         -1         -1
12           1    -1    -1   -1 28          1          1
13           1     1     0    1 41          1          0
14           1     1     0   -1 35         -1          0
15           1     0     1    1 46          0          1
16           1     0     1   -1 41          0         -1
17           1    -1    -1    1 42         -1         -1
18           1    -1    -1   -1 33          1          1
19           1     1     0    1 48          1          0
20           1     1     0   -1 46         -1          0
21           1     0     1    1 40          0          1
22           1     0     1   -1 46          0         -1
23           1    -1    -1    1 42         -1         -1
24           1    -1    -1   -1 50          1          1
25           1     1     0    1 43          1          0
26           1     1     0   -1 32         -1          0
27           1     0     1    1 40          0          1
28           1     0     1   -1 37          0         -1
29           1    -1    -1    1 40         -1         -1
30           1    -1    -1   -1 30          1          1
attr(,"assign")
[1] 0 1 1 2 3 4 4
attr(,"contrasts")
attr(,"contrasts")$FOOD
[1] "contr.sum"

attr(,"contrasts")$SEX
[1] "contr.sum"

> t(X)%*%X

              (Intercept) FOOD1 FOOD2 SEX1     x FOOD1:SEX1 FOOD2:SEX1
  (Intercept)          30     0     0    0  1203          0          0
  FOOD1                 0    20    10    0     0          0          0
  FOOD2                 0    10    20    0     9          0          0
  SEX1                  0     0     0   30    31          0          0
  x                  1203     0     9   31 49349         -8        -21
  FOOD1:SEX1            0     0     0    0    -8         20         10
  FOOD2:SEX1            0     0     0    0   -21         10         20
```

```
> ginv(t(X)%*%X)
             [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.566523646 -1.147025e-02  2.294050e-02  0.0395086448 -0.0382341724
[2,] -0.011470252  6.675248e-02 -3.350496e-02 -0.0002955759  0.0002860412
[3,]  0.022940503 -3.350496e-02  6.700992e-02  0.0005911518 -0.0005720824
[4,]  0.039508645 -2.955759e-04  5.911518e-04  0.0343514281 -0.0009852530
[5,] -0.038234172  2.860412e-04 -5.720824e-04 -0.0009852530  0.0009534706
[6,]  0.006372362 -4.767353e-05  9.534706e-05  0.0001642088 -0.0001589118
[7,] -0.043332062  3.241800e-04 -6.483600e-04 -0.0011166201  0.0010806001
              [,6]        [,7]
[1,]  6.372362e-03 -0.04333206
[2,] -4.767353e-05  0.00032418
[3,]  9.534706e-05 -0.00064836
[4,]  1.642088e-04 -0.00111662
[5,] -1.589118e-04  0.00108060
[6,]  6.669315e-02 -0.03351343
[7,] -3.351343e-02  0.06789135

> lmeout.1<-lme(y~1+FOOD*SEX+x,random=~1|PEN)
> summary(lmeout.1)
Linear mixed-effects model fit by REML
 Data: NULL
       AIC     BIC    logLik
  79.94085 90.1603 -30.97043

Random effects:
 Formula: ~1 | PEN
        (Intercept)  Residual
StdDev:   0.2568625 0.5003566

Fixed effects: y ~ 1 + FOOD * SEX + x
                Value Std.Error DF   t-value p-value
(Intercept)  6.059989 0.7846109 19  7.723559  <.0001
FOOD1        0.368939 0.1293201 19  2.852909  0.0102
FOOD2       -0.064877 0.1297052 19 -0.500189  0.6227
SEX1        -0.203937 0.0934864 19 -2.181456  0.0419
x            0.080906 0.0192210 19  4.209272  0.0005
FOOD1:SEX1   0.063849 0.1292312 19  0.494067  0.6269
FOOD2:SEX1  -0.085973 0.1310152 19 -0.656205  0.5196
 Correlation:
           (Intr) FOOD1  FOOD2  SEX1   x      FOOD1:
FOOD1      -0.044
FOOD2       0.087 -0.501
SEX1        0.209 -0.009  0.019
x          -0.982  0.045 -0.089 -0.212
FOOD1:SEX1  0.024 -0.001  0.002  0.005 -0.025
FOOD2:SEX1 -0.163  0.007 -0.015 -0.035  0.166 -0.497

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.0926377 -0.4351479  0.1515717  0.4655641  2.0662327

Number of Observations: 30
Number of Groups: 5

> fixed.effects(lmeout.1)
(Intercept)       FOOD1       FOOD2        SEX1           x  FOOD1:SEX1
 6.05998854  0.36893857 -0.06487715 -0.20393657  0.08090635  0.06384894
 FOOD2:SEX1
-0.08597280
```

```
> random.effects(lmeout.1)
  (Intercept)
1  -0.1320881
2   0.1365033
3  -0.2731774
4   0.1545292
5   0.1142330

> predict(lmeout.1,level=0:1)
   PEN predict.fixed predict.PEN
1    1      9.363281    9.231193
2    1     10.452520   10.320432
3    1      8.860550    8.728462
4    1     10.168526   10.036438
5    1      9.457619    9.325531
6    1      9.821245    9.689157
7    2      9.120562    9.257065
8    2      9.158018    9.294521
9    2      8.779643    8.916147
10   2      8.874024    9.010527
11   2      8.567650    8.704153
12   2      8.203118    8.339621
13   3      9.606000    9.332823
14   3      9.400737    9.127560
15   3      9.426894    9.153717
16   3      9.602181    9.329004
17   3      8.972181    8.699004
18   3      8.607650    8.334472
19   4     10.172344   10.326874
20   4     10.290707   10.445236
21   4      8.941456    9.095985
22   4     10.006713   10.161242
23   4      8.972181    9.126710
24   4      9.983058   10.137587
25   5      9.767813    9.882046
26   5      9.158018    9.272251
27   5      8.941456    9.055689
28   5      9.278556    9.392789
29   5      8.810369    8.924602
30   5      8.364930    8.479163

> W<-lmeout.1$varFix
> W
             (Intercept)         FOOD1         FOOD2          SEX1
(Intercept)  0.615614240 -4.444439e-03  8.888878e-03  1.530862e-02
FOOD1       -0.004444439  1.672370e-02 -8.411726e-03 -1.145284e-04
FOOD2        0.008888878 -8.411726e-03  1.682345e-02  2.290567e-04
SEX1         0.015308623 -1.145284e-04  2.290567e-04  8.739712e-03
x           -0.014814797  1.108339e-04 -2.216678e-04 -3.817612e-04
FOOD1:SEX1   0.002469133 -1.847231e-05  3.694463e-05  6.362686e-05
FOOD2:SEX1  -0.016790103  1.256117e-04 -2.512235e-04 -4.326627e-04
                        x    FOOD1:SEX1    FOOD2:SEX1
(Intercept) -1.481480e-02  2.469133e-03 -0.0167901028
FOOD1        1.108339e-04 -1.847231e-05  0.0001256117
FOOD2       -2.216678e-04  3.694463e-05 -0.0002512235
SEX1        -3.817612e-04  6.362686e-05 -0.0004326627
x            3.694463e-04 -6.157438e-05  0.0004187058
FOOD1:SEX1  -6.157438e-05  1.670071e-02 -0.0084150097
FOOD2:SEX1   4.187058e-04 -8.415010e-03  0.0171649840
```

```
> cc<-c(1,1,0,1,38,1,0)
> t(cc)%*%W%*%cc

             [,1]
  [1,] 0.06659204

> intervals(lmeout.1)
Approximate 95% confidence intervals

 Fixed effects:
                  lower        est.        upper
(Intercept)  4.41777908  6.05998854  7.702197990
FOOD1        0.09826839  0.36893857  0.639608756
FOOD2       -0.33635335 -0.06487715  0.206599059
SEX1        -0.39960591 -0.20393657 -0.008267226
x            0.04067637  0.08090635  0.121136340
FOOD1:SEX1  -0.20663515  0.06384894  0.334333033
FOOD2:SEX1  -0.36019077 -0.08597280  0.188245177
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: PEN
                     lower      est.     upper
sd((Intercept)) 0.07845735 0.2568625 0.8409456

 Within-group standard error:
    lower      est.     upper
0.3653562 0.5003566 0.6852403
```

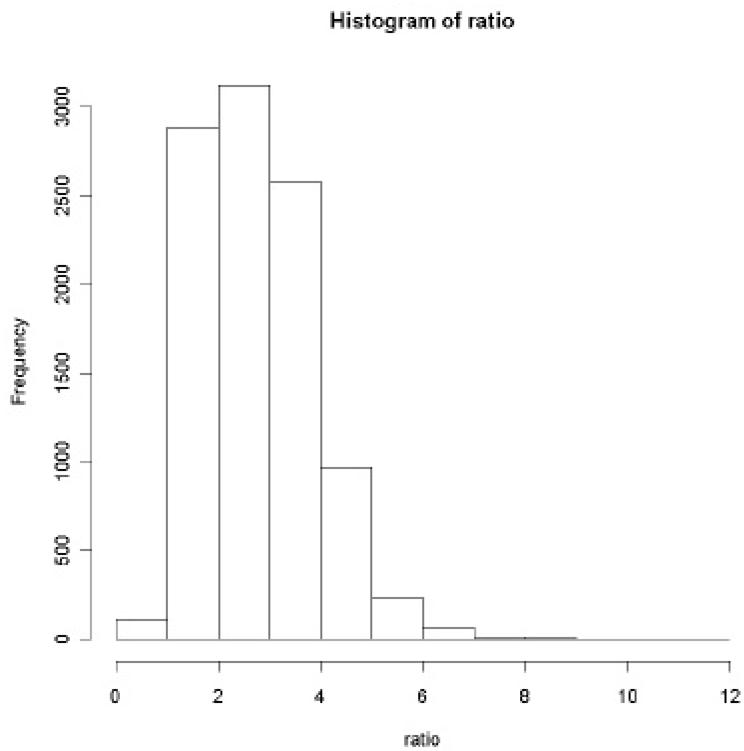
## Printout #2

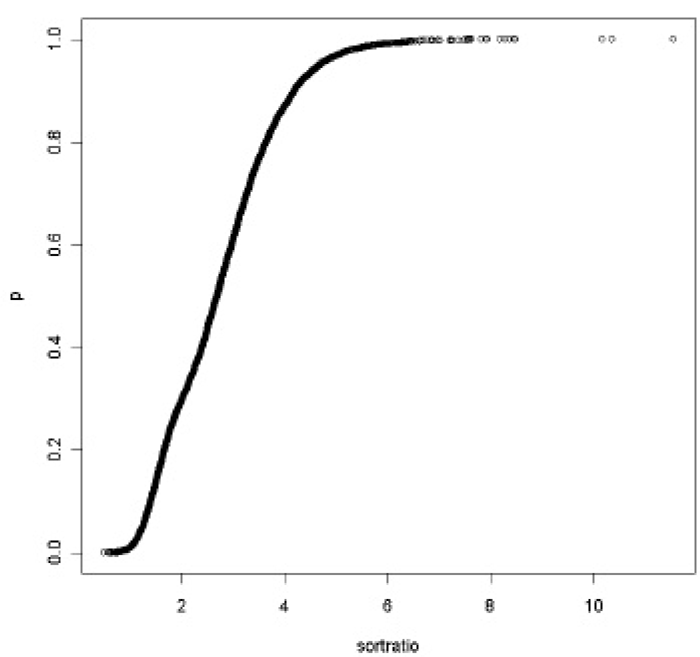
```
> A<-c(1,1,1,1,2,2,2,2)
> B<-c(1,1,2,2,1,1,2,2)
> C<-c(1,2,1,2,1,2,1,2)
> AA<-as.factor(A)
> BB<-as.factor(B)
> CC<-as.factor(C)
> shorts<-c(1,8,0,2,0,1,0,0)
> nonshorts<-c(79,80,90,98,90,89,90,90)
> M<-matrix(c(shorts,nonshorts),nrow=8)
> M
     [,1] [,2]
[1,]    1   79
[2,]    8   80
[3,]    0   90
[4,]    2   98
[5,]    0   90
[6,]    1   89
[7,]    0   90
[8,]    0   90
> contrasts=c("constr.sum","contr.sum")
> cbind(M,AA,BB,CC)
          AA BB CC
[1,] 1 79  1  1  1
[2,] 8 80  1  1  2
[3,] 0 90  1  2  1
[4,] 2 98  1  2  2
[5,] 0 90  2  1  1
[6,] 1 89  2  1  2
[7,] 0 90  2  2  1
[8,] 0 90  2  2  2

> glmout.1<-glm(M~1+AA+BB+CC,family=binomial)
> summary(glmout.1)

Call:
glm(formula = M ~ 1 + AA + BB + CC, family = binomial)

Deviance Residuals:
[1]   0.2516  -0.1364  -0.5382   0.2160  -0.3823   0.2350  -0.1556  -0.5117

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.5828     0.7340  -7.606 2.82e-14 ***
AA1           1.2410     0.5259   2.360   0.0183 *
BB1           0.8988     0.3927   2.289   0.0221 *
CC1          -1.1906     0.5259  -2.264   0.0236 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.03208  on 7  degrees of freedom
Residual deviance:  0.90572  on 4  degrees of freedom
AIC: 19.319

Number of Fisher Scoring iterations: 5

> predict.glm(glmout.1,type="response",se.fit=TRUE)
$fit
[1] 0.0096266101 0.0951457730 0.0016078817 0.0171233382 0.0008116713 0.0087110307 0.0001345714
[8] 0.0014538378

$se.fit
           1            2            3            4            5            6            7
```

```
0.0096840952 0.0307436541 0.0019331464 0.0121408795 0.0011299453 0.0087560684 0.0002086252
           8
0.0017574550

$residual.scale
[1] 1

> glmout.2<-glm(M~1+AA+BB+CC,family=binomial(link="probit"))
> summary(glmout.2)

Call:
glm(formula = M ~ 1 + AA + BB + CC, family = binomial(link = "probit"))

Deviance Residuals:
[1]   0.17283  -0.13068  -0.41706   0.16617  -0.27212   0.15695  -0.05632  -0.39104

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.7237     0.2754  -9.890  < 2e-16 ***
AA1           0.5181     0.1974   2.625  0.00868 **
BB1           0.3958     0.1610   2.458  0.01396 *
CC1          -0.4991     0.1987  -2.512  0.01202 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.03208  on 7  degrees of freedom
Residual deviance:  0.50326  on 4  degrees of freedom
AIC: 18.916

Number of Fisher Scoring iterations: 5

> predict.glm(glmout.2,type="response",se.fit=TRUE)
$fit
[1] 1.047295e-02 9.496660e-02 9.658639e-04 1.776125e-02 4.112885e-04 9.465107e-03 1.762000e-05
[8] 8.491411e-04

$se.fit
           1            2            3            4            5            6            7
0.0102756572 0.0303632127 0.0014956150 0.0123386316 0.0007669500 0.0092984797 0.0000467015
           8
0.0013301563

$residual.scale
[1] 1
```

# Printout #3

```
> laser<-c(42.8,42.2,42.7,43.1,40.0,43.5,42.3,40.3,41.3,48.5,39.5,41.1,42.1)
> edm<-c(46.1,45.3,45.3,44.7,44.2,44.6,43.4,44.6,44.6,45.5,44.4,44.0,43.2)
> B<-10000
> sd(laser)
[1] 2.245937
> sd(edm)
[1] 0.8169675
> bootstrap(laser,20,sd)
$thetastar
 [1] 3.471828 2.138745 2.076394 2.359243 2.394251 1.510434 2.208260 1.242568
 [9] 2.937119 1.339537 1.988493 3.391033 1.226523 3.633904 2.148255 1.923038
[17] 1.154312 2.467169 1.064461 2.001474

$func.thetastar
NULL

$jack.boot.val
NULL

$jack.boot.se
NULL

$call
bootstrap(x = laser, nboot = 20, theta = sd)

> laser.boot<-bootstrap(laser,B,sd)
> edm.boot<-bootstrap(edm,B,sd)
> ratio<-laser.boot$thetastar/edm.boot$thetastar
> mean(ratio)
[1] 2.746609
> sd(ratio)
[1] 1.117361
> sortratio<-sort(ratio)
> number<-1:B
> p<-(number-.5)/B
> hist(ratio)
> plot(sortratio,p)
```


Histogram of ratio
Frequency
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
ratio


p
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
sortratio

## Printout #4

```
> x
 [1] 1.00 1.05 1.10 1.15 1.20 1.25 1.30 1.35 1.40 1.45 1.50 1.55 1.60 1.65 1.70
[16] 1.75 1.80 1.85 1.90 1.95 2.00 2.05 2.10 2.15 2.20 2.25 2.30 2.35 2.40 2.45
[31] 2.50 2.55 2.60 2.65 2.70 2.75 2.80 2.85 2.90 2.95 3.00
> y
 [1] -0.26232650 -1.10885623 -0.93172175 -2.08970403 -3.07915638 -3.46467589
 [7] -2.44028167 -1.96107298 -2.83569242 -2.18375270 -1.29994042  0.34442342
[13]  1.55725364  1.58448747  3.95302468  2.76315046  1.53574872  3.23472845
[19]  2.09476030  0.64154217  0.82649241  0.83010341 -1.12225588 -2.12193283
[25] -2.32102729 -2.81875419 -1.35762260 -2.21964804 -1.72421575 -1.73203445
[31] -0.50204286  2.14766524  2.05368006  2.14626108  2.22802343  3.17616529
[37]  2.41612346  2.17823690 -0.06156593  0.83626934  0.78952921

> nlsfit.1<-nls(formula=y~A*sin(k*(x-x0)),start=c(A=3,k=6,x0=.5),trace=T)
46.0056 :  3.0 6.0 0.5
22.68993 :  2.5463625 6.2993888 0.5293745
20.95141 :  2.7601609 6.3924310 0.5414934
20.94946 :  2.7698732 6.3916728 0.5413062
20.94946 :  2.7698742 6.3916832 0.5413134
> summary(nlsfit.1)

Formula: y ~ A * sin(k * (x - x0))

Parameters:
   Estimate Std. Error t value Pr(>|t|)
A   2.76987    0.16818   16.47   <2e-16 ***
k   6.39168    0.09408   67.94   <2e-16 ***
x0  0.54131    0.02314   23.40   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.7425 on 38 degrees of freedom

Correlation of Parameter Estimates:
        A      k
k  0.1034
x0 0.0920 0.9196

> vcov(nlsfit.1)
             A           k           x0
A  0.0282848655 0.001636502 0.0003579508
k  0.0016365023 0.008850525 0.0020015831
x0 0.0003579508 0.002001583 0.0005352606
> coefficients(nlsfit.1)
        A         k        x0
2.7698742 6.3916832 0.5413134
> confint(nlsfit.1,level=.90)
Waiting for profiling to be done...
21.18238 :  6.3916832 0.5413134
21.17991 :  6.3851322 0.5398804
21.17990 :  6.3852631 0.5399496
21.17990 :  6.3852323 0.5399402
21.87146 :  6.3787817 0.5385672
21.87146 :  6.3786360 0.5385309
21.92093 :  2.779055 6.504811
23.12721 :  2.780710 6.559517
```

```
.
.
.

23.12642 :  2.774893 6.560042
23.12642 :  2.774898 6.560063
24.80465 :  2.770824 6.614209
24.80383 :  2.764828 6.614718
24.80383 :  2.764833 6.614737
26.94822 :  2.754928 6.668537
26.94738 :  2.748802 6.669037
26.94738 :  2.748806 6.669053
          5%       95%
A  2.4864217 3.0535622
k  6.2288065 6.5487028
x0 0.5014983 0.5783315
> sinusoid<-coef(nlsfit.1)[1]*sin(coef(nlsfit.1)[2]*(x-coef(nlsfit.1)[3]))
> plot(x,sinusoid)
> plot(c(.5,-4),c(3.5,4),type="n",xlab="x",ylab="y")
> points(x,y)
> lines(x,sinusoid)
```