Course EPIB-621 - Data Analysis for the Health Sciences

Assignment 3 - Solutions

1. Consider the data set called assign3num1.txt on the course web page.

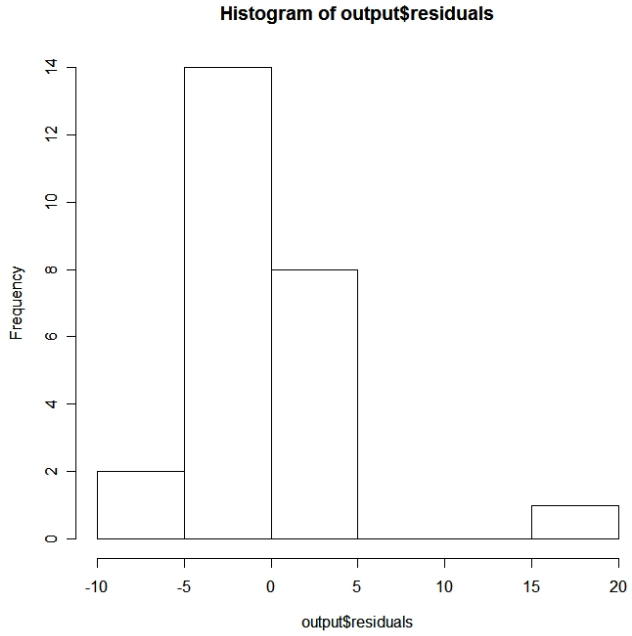
(a) Run a linear regression of y on x1 and x2. Report the usual summary statistics, including the R^2 value and confidence intervals.

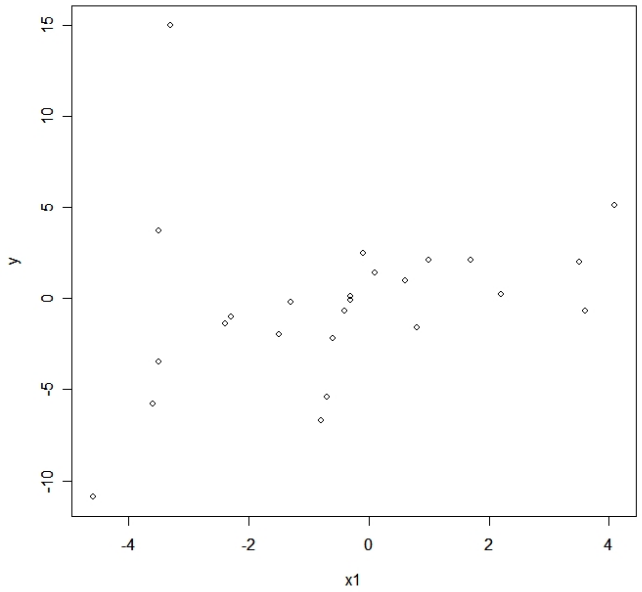
```
#  Run the regression with output and Confidence intervals
> output<- lm( y ~ x1 + x2)

> summary(output)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max
-5.9427 -1.9614 -0.6522  0.9763 16.2161

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.4267     0.9081   0.470   0.6430
x1            0.6095     0.3783   1.611   0.1214
x2            0.9215     0.4158   2.216   0.0373 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.328 on 22 degrees of freedom
Multiple R-Squared: 0.2353,     Adjusted R-squared: 0.1658
F-statistic: 3.384 on 2 and 22 DF,  p-value: 0.0523

> confint(output)
                  2.5 %   97.5 %
(Intercept) -1.45647959 2.309912
x1          -0.17498207 1.394034
x2           0.05924673 1.783781
```

```
#  Note the relatively poor R2 value.
```

(b) Plot a histogram of residuals, and report any outliers. Identify the outlier by plotting x1 versus the residuals.

```
> hist(output$residuals)
> plot(x1, output$residuals)
```






```
#  The outlier occurs at fifth lowest x1 value:

> sort(x1)
 [1] -4.6 -3.6 -3.5 -3.5 -3.3 -2.4 -2.3 -1.5 -1.3 -0.8 -0.7 -0.6 -0.4
[14] -0.3 -0.3 -0.1  0.1  0.6  0.8  1.0  1.7  2.2  3.5  3.6  4.1

# So when x = -3.3

> x1
 [1] -0.6 -4.6 -0.4 -1.3  1.0  4.1  1.7 -0.3  0.6 -0.3 -2.3 -0.7 -3.3
[14]  0.8 -3.6 -0.1 -0.8 -3.5 -1.5 -3.5  2.2  3.6  3.5  0.1 -2.4
```

```
#  So outlier occurs at the 13th data point.
```

(c) Delete that outlier from the data set, and rerun the linear regression. By how much do the parameter estimates and their confidence intervals change? How much does the R^2 value change by?

```
#  Rerun regression without the 13th data point

> output2<- lm( y[-13] ~ x1[-13] + x2[-13])
> summary(output2)

Call:
lm(formula = y[-13] ~ x1[-13] + x2[-13])

Residuals:
     Min       1Q   Median       3Q      Max
-3.87999 -1.63591  0.07775  1.08741  4.33537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1650     0.5058  -0.326 0.747452
x1[-13]       0.9862     0.2144   4.600 0.000155 ***
x2[-13]       0.8281     0.2289   3.618 0.001613 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.379 on 21 degrees of freedom
Multiple R-Squared: 0.5982,     Adjusted R-squared: 0.5599
F-statistic: 15.63 on 2 and 21 DF,  p-value: 6.951e-05

> confint(output2)
                 2.5 %    97.5 %
(Intercept) -1.2168577 0.8868163
x1[-13]      0.5403211 1.4319815
x2[-13]      0.3521316 1.3040780

#  Note the very large influence of this outlier, R2 now
#  much higher (more than double previous value), and beta
#  parameters much more accurately estimated (standard errors cut
#  by half, approximately.
```

2. The FIM is a measure of functional independence, often used in the emergency room following an accident or trauma. Higher values of the FIM indicate better functioning, with 126 being considered as "normal". In this question we will consider the effect of alcohol on the FIM. This is an interesting question, since it is not clear whether alcohol will have a positive (e.g., more relaxed during an accident may lead to less injuries) or negative (e.g., worse accidents if under the influence) effect on the FIM, on average.

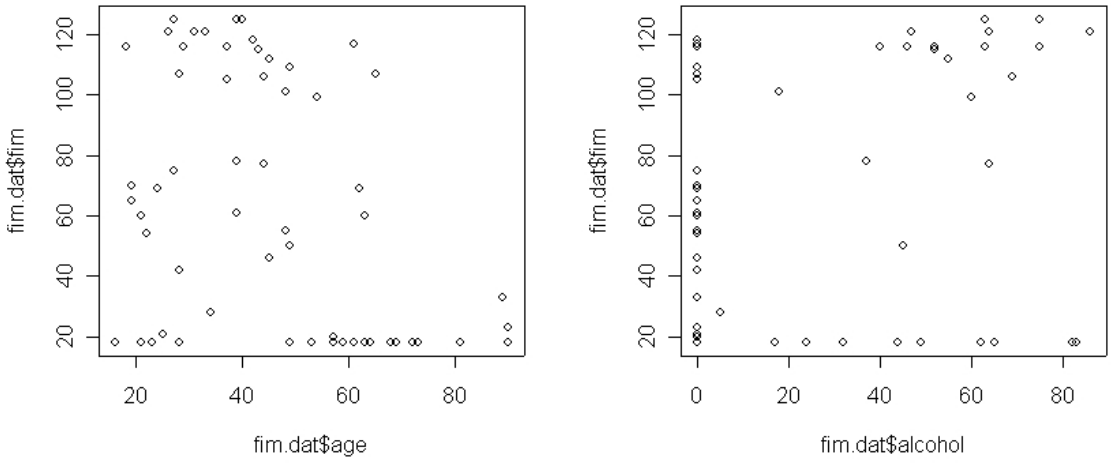
(a) The course web site has a data set called fimdata.txt. Download this file somewhere to your hard disk. Open R and read in the data set (using a command such as

```
fim.dat <-read.table(file, header=T),
```

where file points to where you stored the data set, such as "c://temp//fimdata.txt"). The variables include age in years, sex (1=male, 0=female), fim, and alcohol level on arrival at the emergency room. Print out some descriptive statistics for these variables, for example using the R command summary(fim.dat). Print out plots of age versus fim and alcohol versus fim. Do you see any trends?

A program is below, with results and graphs:

```
> fim.dat<-read.table(file="c://temp//fimdata.txt", header =T)
> summary(fim.dat)
      sex              age              fim            alcohol
 Min.   :0.0000   Min.   :16.00   Min.   : 18.00   Min.   : 0.00
 1st Qu.:1.0000   1st Qu.:28.00   1st Qu.: 18.00   1st Qu.: 0.00
 Median :1.0000   Median :42.50   Median : 63.00   Median : 0.00
 Mean   :0.8167   Mean   :44.52   Mean   : 66.53   Mean   :25.62
 3rd Qu.:1.0000   3rd Qu.:59.50   3rd Qu.:112.75   3rd Qu.:52.75
 Max.   :1.0000   Max.   :90.00   Max.   :125.00   Max.   :86.00
> plot(fim.dat$age,fim.dat$fim)
> plot(fim.dat$alcohol,fim.dat$fim)
```



No extremely strong trends, but it looks like FIM may decrease with increasing age, affect of alcohol not clear.

(b) To use the data set in WinBUGS, you will need to add []'s after each variable name. Open WinBUGS, and cut and paste in the data set you saved. Change the first line to

```
sex[]  age[] fim[]  alcohol[]
```

Run a simple linear regression in WinBUGS, with fim as the dependent variable, and age, sex, and alcohol level. Report your results, and provide an interpretation for each beta coefficient.

Program is below, and results follow.

```
model
{
for (i in 1:60) {
fim.mean[i] <- alpha + beta.sex*sex[i] + beta.age*age[i]
           + beta.alc*alcohol[i];
fim[i]   ~ dnorm(fim.mean[i],tau);
}
    alpha    ~ dnorm(0.0,1.0E-4);
    beta.sex     ~ dnorm(0.0,0.001);
    beta.age     ~ dnorm(0.0,0.001);
    beta.alc     ~ dnorm(0.0,0.001);
    tau      <- 1/(sigma*sigma)
    sigma   ~ dunif(0.001, 100)
 }
```

node	mean	sd	2.5%	median	97.5%	start	sample
alpha	109.0	17.61	73.45	109.2	143.1	1001	10000
beta.age	-0.8959	0.2556	-1.404	-0.8969	-0.391	1001	10000
beta.alc	0.5127	0.1716	0.1701	0.5128	0.8503	1001	10000
beta.sex	-19.68	12.75	-45.02	-19.8	6.134	1001	10000
sigma	37.15	3.586	30.98	36.87	44.87	1001	10000
tau	7.443E-4	1.405E-4	4.967E-4	7.356E-4	0.001043	1001	10000

FIM decreases with age, as expected, by about 1 point on the FIM scale for each year increase. Alcohol increases the FIM by about a half point for each increase of one unit of the alcohol sclae, while males seem to be, on average, about 20 points lower than females. The latter result, however, is not estimated accurately with these data.

(c) It is easy to make predictions for any variable combination using WinBUGS, simply by adding a single line to the program for each prediction you want to make. For example, if you want to make a fim prediction for a male aged 50 with an alcohol level of 80, add a line like:

```
pred.fim.male.50.80 <- alpha + beta.sex*1 + beta.age*50
      + beta.alc*80
```

and then monitor the new parameter you created, pred.fim.male.50.80. Create predictions for males aged 30 and alcohol level of 85 and for females aged 50 with alcohol level of zero. Create another variable that monitors the difference between these two groups, with a command such as

```
diff <- pred.fim.male.30.85 - pred.fim.female.50.0
```

Run a program with these additional lines, report the outcomes, and comment of the difference between the two sets of predictions you made.

Adding the lines as suggested above (reproduced below) to the program and rerunning, we have:

```
pred.fim.male.30.85 <- alpha + beta.sex*1 + beta.age*30 + beta.alc*85
pred.fim.female.50.0 <- alpha + beta.sex*0 + beta.age*50 + beta.alc*0
diff <-  pred.fim.male.30.85 - pred.fim.female.50.0
```

node	mean	sd	2.5%	median	97.5%	start	sample
diff	41.82	15.52	11.93	41.6	72.54	1001	10000
pred.fim.female.50.0	64.25	10.61	43.25	64.38	84.84	1001	10000
pred.fim.male.30.85	106.1	11.43	83.2	106.2	129.0	1001	10000

Thus, the younger male drinkers group are clearly expected to have higher FIM values compared to the older females non-drinkers.

3. In this question we again will use the bicreg program to examine various models relating to predicting IQ.

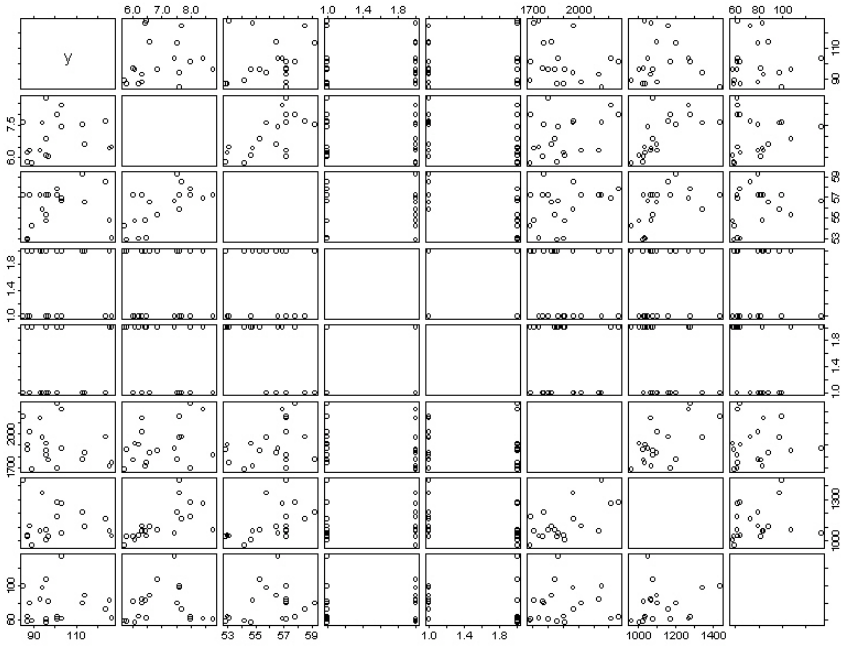
(a) Obtain the data file "brain.data.txt" from the course web page. Variables used are:

CCMIDSA:	Corpus Collasum Surface Area (cm2)
IQ:	Intelligence Quotient
HC:	Head Circumference (cm)
ORDER:	Birth Order
SEX:	Sex (1=Male 2=Female)
TOTSA:	Total Surface Area (cm2)
TOTVOL:	Total Brain Volume (cm3)
WEIGHT:	Body Weight (kg)

Read the data set into R (for example, using a "read.table" command, note that a header is present, so use "header=T"), and use the "pairs" command to produce a pairwise plot. Comment on which variables seem to be related or not, just looking at the graphs alone.

The graph is below. Examining it, there is clearly a strong correlation between between CCMIDSA and TOTVOL, and a weaker negative correlation between IQ

and TOTSA. Hard to see much else aside from these, so expect TOTSA to possibly enter the model for IQ, but unclear if other variables are important, or how strong this relationship is.



(b) Use the bicreg program to perform model selection on this data set. There are instructions for how to use this function in the first few lines of the function itself. Provide the output in your answer.

Running the bicreg program, the commands and results are given below:

```
> brain.data <- read.table("c://lawrence//work//courses//677//assignments//
                              brain.data.txt", header=T)
> attach(brain.data)
> y<-IQ
> x<-matrix(c(CCMIDSA, HC, ORDER, SEX, TOTSA, TOTVOL, WEIGHT), byrow=F, ncol=7)
> out.bic<-bicreg(x,y)
> out.bic
Call: bicreg(x = x, y = y)
 Posterior probabilities(%):
  X1   X2   X3   X4   X5   X6   X7
21.3 21.3  9.3 14.5 44.4 13.6  9.2

 Coefficient posterior expected values:
```

```
(Intercept)   X1           X2           X3           X4           X5

96.0795127    0.7700018    0.4153930    0.0390453   -0.5641523   -0.0119250

X6           X7
-0.0007185   -0.0001642

> summary(out.bic)

Call: bicreg(x = x, y = y)
 27  models were selected
 Best  5  models (cumulative posterior probability =  0.471 ):

     p!=0   model 1    model 2    model 3    model 4    model 5
Int  100.0  101.00000  142.96344   85.21792  127.90852   45.04984
X1   21.3      .          .        2.25700    4.14751      .
X2   21.3      .          .          .          .        0.99688
X3   9.3       .          .          .          .          .
X4   14.5      .          .          .          .          .
X5   44.4      .       -0.02201      .       -0.02933      .
X6   13.6      .          .          .          .          .
X7   9.2       .          .          .          .          .

nVar          0          1          1          2          1
r2            0.000      0.085      0.024      0.158      0.019
BIC           0.00000    1.22195    2.50148    2.55338    2.61208
post prob     0.198      0.107      0.057      0.055      0.054
```

(c) Look at the best model. How much better is it than the second best model?

Looking at the results provided in (b), we find that the posterior probability of the best model, which is simply the null model is about 0.198, while the probability of the second best model, which contains the variable TOTSA alone is about 0.107. Thus, the first model is about 2 times more likely to be correct that the second. Note that none of the variables reach 50% probability of being in a model (closest was at 45%).

4. Average blood mercury levels vary from region to region, because of different degrees of fish consumption, different origins for available fish, and other factors. The data set called fish.txt contains the mercury concentrations (in μ grams per liter) for 20 randomly selected persons from each of five different regions. Assume that within each region, mercury levels vary randomly around the region specific mean, and that the region specific means also vary randomly around an overall

mean value. Run a random effects model in R, and report the parameter estimates (and confidence intervals) for both region specific and overall mercury levels.

```
#  Read in the data set

> fish <- read.table(file="g:\\fish.txt", header=T)

#  Create a new data frame suitable for random effects models

#  First denote the region variable to be a factor

>  fish$region <- as.factor(fish$region)

#  Create a new data frame with this factor variable, suitable for
#  random effects models

#  Be sure to load the nlme package first!!

> fish.grouped <- grouped.Data(mercury ~ 1 | region, data = fish)

#  Look at result

> fish.grouped
Grouped Data: mercury ~ 1 | region
    mercury region
1       1.1      1
2       1.5      1
3       2.4      1
4       0.5      1
5       0.6      1
6       1.3      1
7       2.8      1
8       0.7      1
....etc........
97      3.5      5
98      4.5      5
99      6.0      5
100     3.9      5

#  Now run nlme function

> output <- lme(mercury ~ 1, data=fish.grouped)
```

```
> summary(output)
Linear mixed-effects model fit by REML
 Data: fish.grouped
       AIC      BIC    logLik
  281.6555 289.4409 -137.8278

Random effects:
 Formula: ~1 | region
        (Intercept)  Residual
StdDev:    1.376096 0.8789408

Fixed effects: mercury ~ 1
             Value Std.Error DF  t-value p-value
(Intercept) 2.874 0.6216538 95 4.623152       0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.29944906 -0.81477919 -0.01444134  0.57894388  2.03553066

Number of Observations: 100
Number of Groups: 5

#  Look at overall mean coefficients

> output$coefficients
$fixed
(Intercept)
      2.874

$random
$random$region
  (Intercept)
1 -1.49843465
2 -1.12603101
3 -0.05292052
4  0.93688916
5  1.74049701

#  Note overall mean of about 3, but much regional variation.
```

5. Repeat question 4, but use a Bayesian hierarchical model. Provide your program

in WinBUGS, and monitor and report on all unknown parameters from the model.

```
WinBUGS program and results are given below

model
{
    for( i in 1:20)  # Loop over 20 subjects from region 1
{
   mercury[i] ~ dnorm(mu[1] , tau.w) # Data are normally distributed
}
    for( i in 21:40)  # Loop over 20 subjects from region 2
{
   mercury[i] ~ dnorm(mu[2] , tau.w) # Data are normally distributed
}
    for( i in 41:60)  # Loop over 20 subjects from region 3
{
   mercury[i] ~ dnorm(mu[3] , tau.w) # Data are normally distributed
}
    for( i in 61:80)  # Loop over 20 subjects from region 4
{
   mercury[i] ~ dnorm(mu[4] , tau.w) # Data are normally distributed
}
    for( i in 81:100)  # Loop over 20 subjects from region 5
{
   mercury[i] ~ dnorm(mu[5] , tau.w) # Data are normally distributed
}

    for (i in 1:5)          #  Loop over five regions
     {
       mu[i] ~ dnorm(merc.m, tau.b) # Means are normally distributed
     }
   merc.m ~ dnorm(0, .001) # Prior for mean mercury level, very wide
   tau.w <- 1/(sigma.w*sigma.w) # Prior for within SD (precision)
   sigma.w ~ dunif(0,100)
   tau.b <- 1/(sigma.b*sigma.b) # Prior for between SD (precision)
   sigma.b ~ dunif(0,100)

# Inits

list(mu=c(2,2,2,2,2), merc.m = 2, sigma.w = 1, sigma.b = 1)

#  Data
```

```
list(mercury=c(1.1, 1.5, 2.4, 0.5, 0.6, 1.3, 2.8, 0.7, 0.5, 0.5, 1.7,
1.5, 0.9, 2.1, 0.9, 1.0, 3.0, 1.6, 1.7, 0.6, 3.2, 2.2, 1.6, 0.5, 2.8,
1.7, 1.8, 1.0, 1.2, 3.3, 0.9, 2.0, 1.4, 1.9, 1.0, 0.5, 1.7, 0.5, 2.8,
2.5, 4.3, 1.9, 2.8, 3.1, 3.7, 3.6, 3.1, 2.1, 2.9, 3.4, 2.1, 3.0, 3.2,
0.8, 2.6, 3.3, 2.2, 3.1, 3.3, 1.9, 2.6, 4.6, 2.5, 2.7, 2.6, 3.6, 3.3,
3.8, 3.0, 5.6, 4.1, 5.0, 4.7, 2.6, 2.9, 5.6, 4.0, 3.4, 4.6, 5.4, 5.0,
4.8, 6.3, 4.2, 4.1, 4.7, 5.8, 5.4, 4.8, 4.6, 2.9, 4.1, 5.3, 3.9, 4.1,
5.1, 3.5, 4.5, 6.0, 3.9))

#  Results

node    mean    sd      MC error   2.5%     median  97.5%   start   sample
merc.m  2.855   1.36    0.008828    0.5098  2.863   5.111   1001    20000
mu[1]   1.367   0.1988  0.001446    0.9793  1.365   1.759   1001    20000
mu[2]   1.741   0.2004  0.001435    1.346   1.742   2.128   1001    20000
mu[3]   2.822   0.2     0.00144     2.43    2.822   3.217   1001    20000
mu[4]   3.816   0.1986  0.001385    3.426   3.816   4.208   1001    20000
mu[5]   4.622   0.2009  0.001447    4.231   4.622   5.019   1001    20000
sigma.b 2.226   1.895   0.03096     0.888   1.794   6.033   1001    20000
sigma.w 0.8912  0.06579 4.972E-4     0.7744  0.8873  1.033   1001    20000

#  Near identical results to frequentist method, but with interval estimates.
```