36-402/608 Homework #3 due 10:30AM 2/5

1. Blood Brain Barrier (50 points)

   We will examine the blood-brain barrier experiment of Sleuth, chapter 11, pp. 307-310, using the data in case1102.csv. Start by running these R commands:

```
bb = read.csv("case1102.csv")
dim(bb)
sapply(bb, class)

# Simplify future typeing by changing names to lower case:
names(bb) = casefold(names(bb))
names(bb)

# Make new variables
bb$logBLratio = log(bb$brain/bb$liver)
bb$logTime = log(bb$time)
```

   (a) (5 points) Using `with()` and `table()` and `prop.table()` turn in R commands and output to show 1) whether the number of males and females is equal for the two treatments, and 2) whether the proportion of "days post innoculation" was similar for the two treatments. Check the help text for prop.table() if you are unfamiliar with this function.

   (b) (5 points) Make and turn in a plot of the outcome (logBLratio) vs. the log sacrifice time (logTime) as in Display 11.5 but showing only the log scale. Use meaningful text rather than variable names for the axis labels. Use the plot option `pch=as.numeric(treat)` to make different symbols for the two treatments. Use `with(bb, table(treat,as.numeric(treat)))` to figure out the correct text to fill into the quotes for making a legend with `legend("topleft", legend=c("","")`, `pch=1:2)`. Turn in the code and plot.

   (c) (5 points) Perform a regression to answer the question "Did the treatment cause a change in the outcome, correcting for the log of the sacrifice time?". For this part assume that the change is by a constant amount at each sacrifice time. Call your `lm()` result "m0" to match everyone to the same names. Turn in your R code and the coefficient table from the summary.

   (d) (5 points) Repeat the previous question, but now allowing a differential effect of treatment on the outcome that depends on the (log of the) sacrifice time. Call your `lm()` result "m1". Turn in your R code and the coefficient table from the summary.

(e) (5 points) Use `rp()` from http://www.stat.cmu.edu/~hseltman/402/R/rp.R (an improved version of what we saw in class) to make the residual vs. fit plots for both models. Turn in a few sentences describing any potential problems revealed by these plots.

(f) (5 points) Ignoring the potential problem(s), make a brief clear statement of the interpretation of the interaction line of the coefficient table for part *d*.

(g) (5 points) Turn in the `summary()` of the `influence.measures()` for the interaction model along with a brief interpretation.

(h) (5 points) Run the command `m2 = update(m1, subset=rownames(bb)!=34)` to remove the potentially troublesome observation. Summarize what changes to any substantial extent between models m1 and m2.

(i) (5 points) Run the `influencePlot()` from package "car" on model "m1". Turn in brief comments on what we can learn from examining the plot.

(j) (5 points) Explain what it would have indicated if we would have seen a large negative value for the logTime component of DFBETAS (dfb.lgTm) in the influence output for subject 34 in part *g*.

2. Boston Housing Data (50 points)

Here we will use a subset of the 1970 Boston housing data that explores the ability of various explanatory variables to help predict median house value (medv in thousands of dollars) at the level of "housing tract". (A housing tract is typically smaller than a "neighborhood".) We will include average number of rooms per dwelling (rm), property-tax rate per $10,000 (tax), and percent "lower status" of the tract (lstat). Load our version of the data using:

```
bost = read.csv("bost.csv")
```

(a) (10 points) Explore the missing value pattern of the data. Turn in R code, your results, and a brief summary of the missingness in complete sentences.

(b) (10 points) Examine the data using `pairs()` and comment on all of your important preliminary conclusions and concerns about the data, particularly with respect to consideration of use of multiple regression for making predictions.

(c) (5 points) Run `b0 = lm(medv ~ rm + tax + lstat, bost)`. Notice that `rp(b0)`, `rp(b0,"rm")`, and `rp(b0,"lstat")` all suggest non-linearity. Run `b1 = lm(medv ~ rm + I(rm^2) + tax + log(lstat), bost)`. Now notice that `rp(b1)`, `rp(b1,"rm")`, and `rp(b1,"log(lstat)")` no longer show problems with linearity, but now there are outliers. Use the `identify=TRUE` option of `rp()` to identify the two worst outliers. Turn in their row numbers in "bost" as well as their row names (which is what is added to the plot using "identify").

(d) (10 points) Use `summary(influence.measures())` and `influencePlot()` to explore the potential effects of these outliers on model "b1". Turn in a summary of your findings.

This code may be of help:

```
i1=influence.measures(b1)
# Count how many observations are ‘‘flagged’’ for a
# particular influence measure:
sum(i1$is.inf[,"cook.d"])
# Show all influence measures for those observations
# that were ‘‘flagged’’ for a particular influence measure:
i1$infmat[i1$is.inf[,"cook.d"],]
# Show all data for those observations that were ‘‘flagged’’
# for a particular influence measure:
bost[i1$is.inf[,"cook.d"],]
```

(e) (5 points) Use `bost.mice = mice(bost,10)` (after loading package "mice") to create 10 imputed datasets, i.e., filling in the missing values. Note that you can use `complete(foo, i)` to get the ith imputed dataset for mice result "foo". Using, e.g.,

```
i=1
influencePlot(lm(medv ~ rm + I(rm^2) + tax + log(lstat),
           data = complete(bost.mice, i)))
```

examine a few of the imputed datasets for influential points, and comment on what you find.

(f) (5 points) Run the ten `lm()` commands all at once using `with(bost.mice, ...)`. Note that you do *not* specify `data=` in your `lm()` command in this context. Use `summary(pool())` on your `with()` result to get your multiply imputed regression results. Summarize what the model says.

(g) (5 points) Create "bostX" from "bost" by eliminating the single worst outlier. Repeat the multiple impuation process, and comment on any important differences in the multiply imputed model. As a rule of thumb, a 5% change in a coefficient estimate or a 10% change in a standard error is likely to be of interest to users of a model.