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:

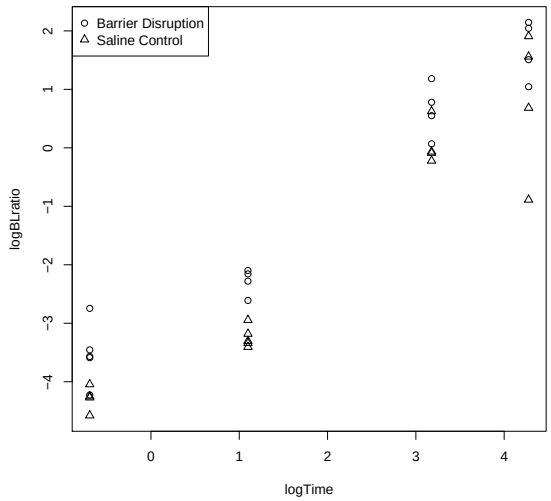
```
bb = read.csv("case1102.csv")
dim(bb)
sapply(bb, class)

# 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 inoculation" was similar for the two treatments. Check the help text for prop.table() if you are unfamiliar with this function.

```
with(bb, table(treat,sex))
#     sex
#treat  F  M
#   BD 13  4
#   NS 13  4
prop.table(with(bb, table(treat,days)), 1)
#     days
#treat          9         10         11
#   BD 0.05882353 0.82352941 0.11764706
#   NS 0.11764706 0.76470588 0.11764706
```

   (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.

```
# Plot key variables
with(bb, plot(logBLratio ~ logTime, pch=as.numeric(bb$treat)))
levels(bb$treat)
# [1] "BD" "NS"
with(bb, table(treat,as.numeric(treat)))
#treat  1  2
#   BD 17  0
#   NS  0 17
legend("topleft", legend=c("Barrier Disruption","Saline Control"), pch=1:2)
```



(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.

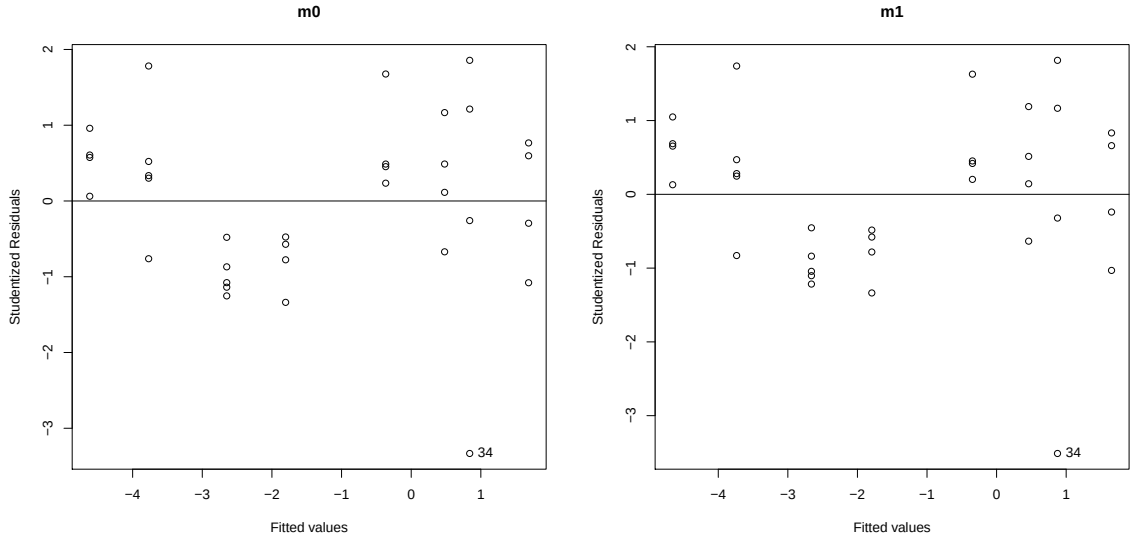
```
m0 = lm(logBLratio ~ logTime + treat, bb)
summary(m0)
#Coefficients:
#            Estimate Std. Error t value Pr(>|t|)
#(Intercept) -3.00928    0.18400 -16.355  < 2e-16 ***
#logTime      1.09784    0.05654  19.416  < 2e-16 ***
#treatNS     -0.84579    0.21640  -3.908 0.000471 ***
```

(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.

```
m1 = lm(logBLratio ~ logTime*treat, bb)
summary(m1)
#Coefficients:
#                Estimate Std. Error t value Pr(>|t|)
#(Intercept)     -2.98457    0.21145  -14.11 8.74e-15 ***
#logTime          1.08417    0.07933   13.67 2.02e-14 ***
#treatNS         -0.89928    0.30693   -2.93  0.00642 **
#logTime:treatNS  0.02870    0.11497    0.25  0.80458
```

(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.

```
source("http://www.stat.cmu.edu/~hseltman/402/R/rp.R")
rp(m0, identify=TRUE)
rp(m1, identify=TRUE)
```



There is clear evidence of a negative residual outlier on rat 34 near that has a high predicted (fitted) value for the logBLratio outcome. There is also clear evidence of non-linearity (smile pattern). It is easy to over-read the indicators of unequal error variability on a plot like this with a fairly small sample size; I don't think there is good evidence of violation of the constant error variance assumption. You cannot see whether or not the Normal error distribution assumption, the fixed-x assumption or the independent errors assumption is violated on a residual vs. fit plot.

(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*.

We do not have sufficient evidence to reject the null hypothesis that the effects of treatment on logBLratio is constant across time. An equivalent statement is that we do not have sufficient evidence to reject the null hypothesis that the slopes of the logBLratio on logTime plots are parallel for the two treatments. Just saying that we retain the null of "no interaction" would not be clear to most researchers.

Extra comments: This leaves us with two possibilities: there is no interaction or there is an interaction and we missed it (made a type two error). We can split the latter into two possibilities: not enough power vs. good power couple with bad luck (inevitable chance that, say, an outcome with 20% probability will occur 20% of the time). One way to get part way around this limitation is to use confidence intervals. The approximate 95% CI on the slope difference is 0.029 +/- 2(0.115) = [-0.201, 0.259]. Because logTime=0 and logTime=1 are will in the range of logTime (-0.69 to 4.28), this can be interpreted as being 95% confident that the mean rise in outcome over that time interval is less than 4.28 more for treated than control rats, compared to the best guess of 1.08 for the control rats. Subject matter experts are needed to judge the importance of this, but it seems to me that this last statement describes a possible large interaction effect, i.e., the power to detect interaction was insufficient.

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

```
summary(influence.measures(m1))
#Potentially influential observations of
#         lm(formula = logBLratio ~ logTime * treat, data = bb) :
#
#   dfb.1_ dfb.lgTm dfb.trNS dfb.lT:N dffit   cov.r   cook.d hat
#34  0.00   0.00     0.14    -0.85    -1.49_*  0.33_*  0.40   0.15
```

Because Cook's D is not flagged for any data point, and there are no high leverage points the outlier will likely not affect the results drastically. One flagged effect is DFFITS=-1.49, which suggests that the fitted value of case 34 will change a lot if that case is dropped. We can see this effect (compared to the minor effect of removing, say, case 33) as follows:

```
# predict(m1, newdata=bb[34,])
# 0.8755268
predict(update(m1,subset=-34),newdata= bb[34,])
# 1.194230
predict(m1, newdata=bb[33,])
# 0.8755268
predict(update(m1,subset=-33), newdata=bb[33,])
# 0.9102933
```

We also see that the confidence intervals change a lot when case 34 is dropped:

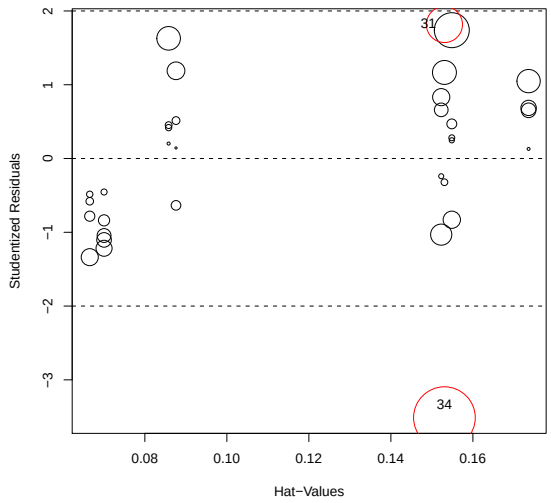
```
apply(confint(m1),1,diff)
#    (Intercept)         logTime         treatNS logTime:treatNS
#      0.8636959       0.3240190       1.2536878       0.4695868
apply(confint(update(m1,subset=-34)),1,diff)
#    (Intercept)         logTime         treatNS logTime:treatNS
#      0.7368614       0.2764365       1.0704263       0.4121385
apply(confint(update(m1,subset=-33)),1,diff)
#    (Intercept)         logTime         treatNS logTime:treatNS
#      0.8781695       0.3294488       1.2757022       0.4911744
```

(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.

```
summary(m2)
#Coefficients:
#                Estimate Std. Error t value Pr(>|t|)
#(Intercept)     -2.98457    0.18014 -16.568 2.52e-16 ***
#logTime          1.08417    0.06758  16.043 5.86e-16 ***
#treatNS         -0.93577    0.26169  -3.576  0.00125 **
#logTime:treatNS  0.11175    0.10076   1.109  0.27649
#
#Residual standard error: 0.5456 on 29 degrees of freedom
#Multiple R-squared: 0.9482,     Adjusted R-squared: 0.9428
#F-statistic: 176.8 on 3 and 29 DF,  p-value: < 2.2e-16
```

There is no effect on the estimate of the intercept or slope for logTime, but their estimated standard errors are smaller when the outlier is removed. The estimated slope for treatNS (intercept difference for barrier treatment compared to control) is larger in magnitude when the outlier is dropped and the interaction coefficient is also larger. Both have smaller standard errors when the outlier is dropped. (These smaller standard errors lead to larger (absolute value) t-values and hence smaller p-values.) R-squared is also somewhat better (bigger) when the outlier is dropped, and the residual standard error is smaller.

(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.



This plot matches the influence.measures() in that it shows one negative residual outlier that does not have undue leverage. Using a different cutoff values for flagging Cook's distance, this plot shows flags (in red) both rats 31 and 34 as potentially having a meaningful effect on the location of the "predicted means surface" and the values of the estimated coefficients.

(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*.

This would indicate that dropping subject 34 is unduly lowering the estimate of the logTime slope coefficient (and dropping that point would result in a higher estimate).

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")
```

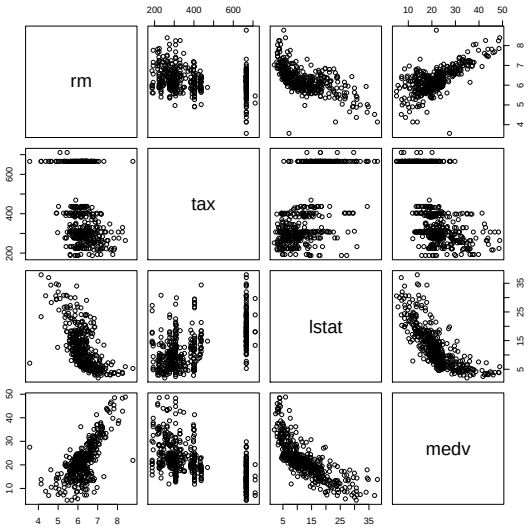
(a) (10 points) Explore the missing value pattern of the data. Turn in R code, your results, and a brief summary of the missingness in complete sentences.

```
sapply(bost,function(x)mean(is.na(x)))
#       rm       tax     lstat      medv
```

```
#0.1326531 0.0000000 0.1265306 0.0000000
table(apply(bost,1,function(x)sum(is.na(x))))
#  0   1   2
#373 107  10
pattern=apply(bost,1,function(x)paste(c("P","N")[1+is.na(x)],collapse=""))
table(pattern)
#NPNP NPPP PPNP PPPP
#  10   55   52  373
prop.table(table(pattern))
#      NPNP       NPPP       PPNP       PPPP
#0.02040816 0.11224490 0.10612245 0.76122449
```

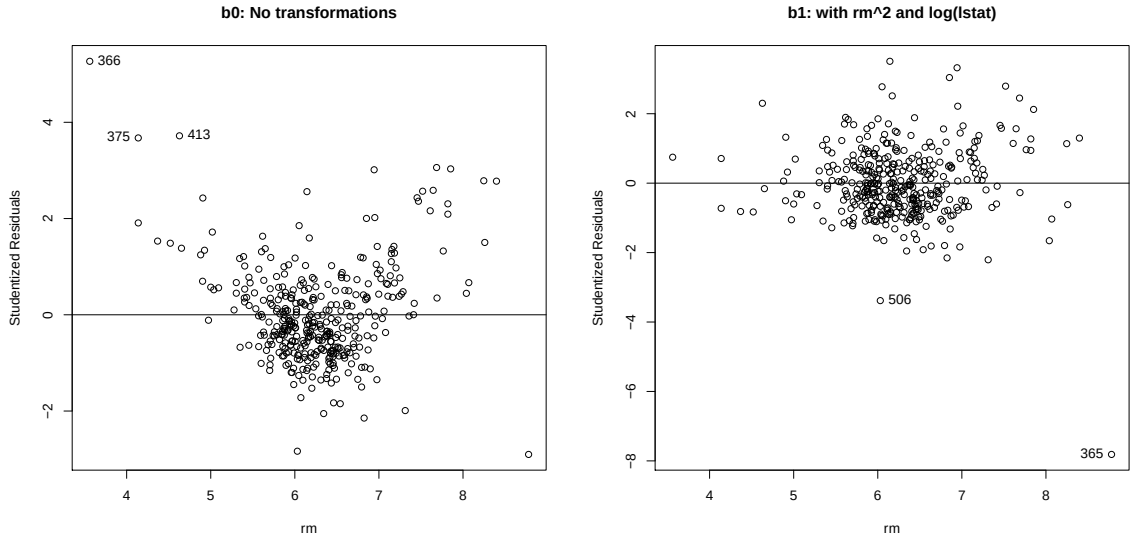
There is missing data only from rooms and percent lower status, both around 13%. Around 3/4 of housing tracts have no missing data, and only 10 tracts (2%) have both missing, while about 22% are missing just one or the other.

(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.



Rooms is directly correlated with value with a lot of low values at the low end of rooms. Tax is negatively correlated with value with a lot of scatter but no evidence of nonlinearity. Lower status percent is negatively associated with value, but in a curved relationship.

(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").



The plot on the right shows that the row names of the worst outliers are 506 and 365. The return value of `rp(b1, "rm", main="b1: with rm^2 and log(lstat)", identify=TRUE)` shows the row numbers are 279 and 373.

(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.

```
i1=influence.measures(b1)
# Count how many observations are ‘‘flagged’’ for a
# particular influence measure:
sum(i1$is.inf[,"cook.d"])
# [1] 1
sum(i1$is.inf[,"hat"])
# [1] 13

# Show all influence measures for those observations
# that were ‘‘flagged’’ for a particular influence measure:
round(i1$infmat[i1$is.inf[,"cook.d"],,drop=F],2)
#     dfb.1_  dfb.rm dfb.I(^2 dfb.tax dfb.lg() dffit cov.r cook.d  hat
# 365  -2.13    2.58    -2.84   -0.43    -0.62 -3.57  0.57   2.19 0.17
```

```
# See which of the above were flagged:
i1$is.inf[i1$is.inf[,"cook.d"],,drop=F]
#    dfb.1_  dfb.rm dfb.I(^2 dfb.tax dfb.lg() dffit cov.r cook.d  hat
#365   TRUE    TRUE     TRUE   FALSE    FALSE  TRUE  TRUE   TRUE TRUE

#### >>> The missing data messed up the code I originally provided.
#### >>> This is the correct code:
# Show all data for those observations that were ‘‘flagged’’
# for a particular influence measure:
which(i1$is.inf[,"cook.d"])
# 365
# 279
bost[rownames(bost)==365, ]
#       rm tax lstat medv
# 365 8.78 666  5.29 21.9
# Put value in perspective as Z-scores
scale(bost)[rownames(bost)==365, ,drop=FALSE]
#           rm      tax     lstat       medv
# 365 3.954214 1.535106 -1.052622 0.03357553

summary(i1)
# The output has 31 flagged observations out of 490 tracts.
```

The one observation with high Cook's D is of most concern. It is tract 365 with a very large number of rooms, a moderately high tax rate, a moderately low percent lower status, and a typical median house value. From the pairs plot, you can see that a lot of rooms tends to be associated with a low tax rate, and a high value, so that explains in what way the tract is unusual.

(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.

Most of the imputed datasets have similar influence plots, but I see occasional single extra high leverage points which have high Cook's D and large (absolute value) residuals.

(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.

```
bost.lms = with(bost.mice, lm(medv ~ rm + I(rm^2) + tax + log(lstat)))
round(summary(pool(bost.lms)),digits=3)
#                est    se       t      df Pr(>|t|)
#(Intercept)  94.355 9.020  10.461 235.000        0
#rm          -21.615 2.875  -7.518 165.247        0
#I(rm^2)       2.061 0.232   8.873 136.259        0
#tax          -0.012 0.001  -8.694 167.340        0
#log(lstat)   -5.856 0.504 -11.617 204.257        0
#               lo 95   hi 95 missing   fmi
#(Intercept)  76.585 112.124      NA 0.131
#rm          -27.292 -15.939      65 0.179
#I(rm^2)       1.602   2.520      NA 0.208
#tax          -0.015  -0.009       0 0.178
#log(lstat)   -6.850  -4.862      NA 0.150
```

We see that all of the coefficients are highly statistically significant. The intercept means little. The rm coefficient means that more rooms are associated with a lower value, while the rm^2 coefficient means that that trend gets less steep as the number of rooms increases. A higher tax rate correlates with lower value. (Although the coefficient is small, the range of tax (187 to 711) means that the change in value from one end of the tax rate to the other is estimated at (711-187)(0.0118)=6.2 out of a median value IQR of 8.0.) Also, as percent lower status rises, median values fall, in a non-linear fashion. All of this could be a misinterpretation if there are important interactions, which we should also check.

(g) (5 points) Create "bostX" from "bost" by eliminating the single worst outlier. Repeat the multiple imputation 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.

```
bostX=bost[-354,]
bostX.mice= mice(bostX,10)
bostX.lms = with(bostX.mice, lm(medv ~ rm + I(rm^2) + tax + log(lstat)))
round(summary(pool(bostX.lms)),3)
#                est    se       t      df Pr(>|t|)   lo 95
#(Intercept) 113.612 8.425  13.484 310.649        0  97.034
```

```
#rm          -28.764 2.625 -10.958 377.148        0 -33.925
#I(rm^2)       2.683 0.212  12.678 372.484        0   2.267
#tax          -0.012 0.001  -9.470 242.133        0  -0.014
#log(lstat)   -5.490 0.513 -10.695  87.036        0  -6.510
#               hi 95 missing   fmi
#(Intercept) 130.190      NA 0.093
#rm          -23.602      65 0.064
#I(rm^2)       3.099      NA 0.066
#tax          -0.009       0 0.127
#log(lstat)   -4.469      NA 0.280
```

Dropping the single outlier had major effects on the coefficient estimates and moderate effects on the standard errors. I would use the model excluding tract 365 for either interpretation of the coefficients or prediction. It is important to realize that that model will not perform will for tracts like number 365 that are highly atypical.