Dose-Time Response Modeling of Rat RBC AChE Activity: Propoxur Inhalation Exposure

June 23, 2005

1 Preamble

Here is some code to set up the analysis: loading required libraries and datasets, and defining some functions.

First, `CarbamateData` loads the full dataset for this risk assessment, and causes the library `DRUtils` to be loaded.

```
> library(CarbamateData)

Loading required package: DRUtils
Loading required package: nlme
Loading required package: tcltk
```

Set up lattice to use B&W instead of color:

```
> library(lattice)
> ltheme <- canonical.theme(color = FALSE)
> ltheme$strip.background$col <- "transparent"
> lattice.options(default.theme = ltheme)
```

Use package `Hmisc` for some formatting support.

```
> library(Hmisc)

Hmisc library by Frank E Harrell Jr

Type library(help='Hmisc'), ?Overview, or ?Hmisc.Overview')
to see overall documentation.

NOTE:Hmisc no longer redefines [.factor to drop unused levels when
subsetting.  To get the old behavior of Hmisc type dropUnusedLevels().

Attaching package 'Hmisc':


        The following object(s) are masked from package:stats :

         ecdf
```

The rat inhalation data for this analysis are in `InhalationData`. The following code prints out documentation for the datasets in use:

```
> printDataDoc(InhalationData)
```

```
----------------------------------------
Data set: InhalationData

Dataset creation date: Wed Jun 22 12:07:28 2005
Script name: /home/setzer/tasks/CarbamateCRA/EmpiricalDoseResponsesFinal/Data/getdata.R
Script last modified: 2005-06-22 12:05:47
sysname: Linux
release: 2.4.24-openmosix2
version: #1 Wed Jul 28 14:33:52 CEST 2004
nodename: node24.cluster2.org
machine: i686
login: unknown
user: setzer
----------------------------------------
```

The following function turns out to be quite useful on subsetted dataframes. It just eliminates unused levels of all factors in the data frame:

```
> CleanUp <- function(x) {
+     for (nm in names(x)) {
+         if (is.factor(x[, nm]))
+             x[, nm] <- factor(x[, nm])
+     }
+     x
+ }
```

To get starting values, we often have to extract values from a previously fit model. The following function simplifies that. The argument `what` is a regular expression:

```
> getParms <- function(what, Par) {
+     Par[grep(what, names(Par))]
+ }
```

This script is for modeling the dose-response for rat RBC. via inhalation exposure.

Set up the analysis dataset.

```
> dta <- CleanUp(subset(InhalationData, chemical %in% "propoxur" &
+     !is.na(cheact)))
```

Summary of this dataset:

```
> summary(dta)

     chemical        mrid                id        sex          dose
 propoxur:381   Min.   :42648001   Min.   :  1.0   F:190   Min.   : 0.00
                1st Qu.:42648001   1st Qu.: 70.0   M:191   1st Qu.: 0.00
                Median :42648001   Median :241.0           Median :10.40
                Mean   :42648001   Mean   :217.1           Mean   :16.09
                3rd Qu.:42648001   3rd Qu.:361.0           3rd Qu.:50.50
                Max.   :42648001   Max.   :430.0           Max.   :50.50
     cheact         tmonstdy       tmunit
 Min.   :1.960   Min.   : 12.00   week:381
 1st Qu.:2.630   1st Qu.: 25.00
 Median :2.880   Median : 51.00
 Mean   :2.921   Mean   : 51.31
 3rd Qu.:3.180   3rd Qu.: 77.00
 Max.   :4.510   Max.   :102.00
```

```
                                    notes        tmpstds
                                       :377   Min.   :0.75
 approx 45 minutes after a 5hr exposure:  1   1st Qu.:0.75
 blood samples for ChE were taken      :  1   Median :0.75
 Chronic inhalation Rat                :  1   Mean   :0.75
 whole body                            :  1   3rd Qu.:0.75
                                              Max.   :0.75

> with(dta, table(dose, tmonstdy, sex))

, , sex = F

      tmonstdy
dose   12 25 51 77 102
  0    10 10 10 10  8
  2    10 10  9  9  7
  10.4 10 10 10 10  9
  50.5 10 10 10 10  8

, , sex = M

      tmonstdy
dose   12 25 51 77 102
  0    10 10 10 10  8
  2    10 10 10  9  7
  10.4 10 10 10  9  8
  50.5 10 10 10 10 10
```

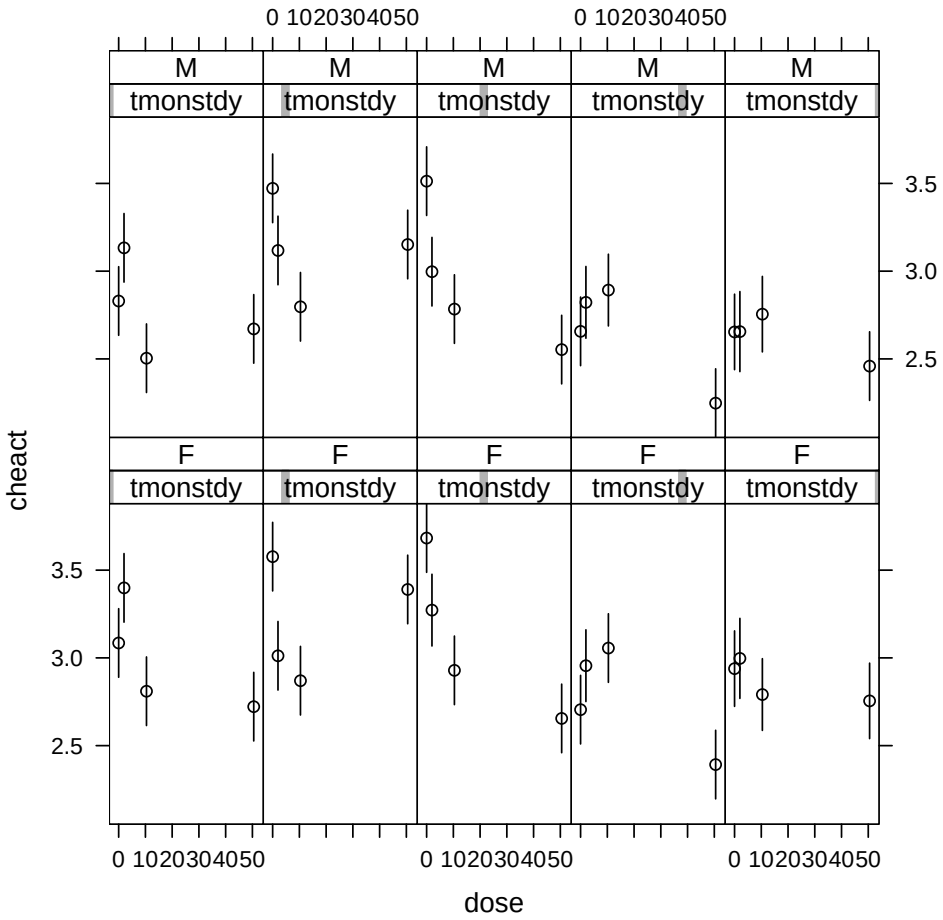
Since there are no pre-dosing values for any animal, there is no way to adjust out individual variability. Dose-response plots, by time on study and sex. First, just raw means and standard errors, calculated taking repeated measures into account:

```
> mns <- lme(cheact ~ sex:factor(tmonstdy):factor(dose) - 1, data = dta,
+     random = ~1 | mrid)
```

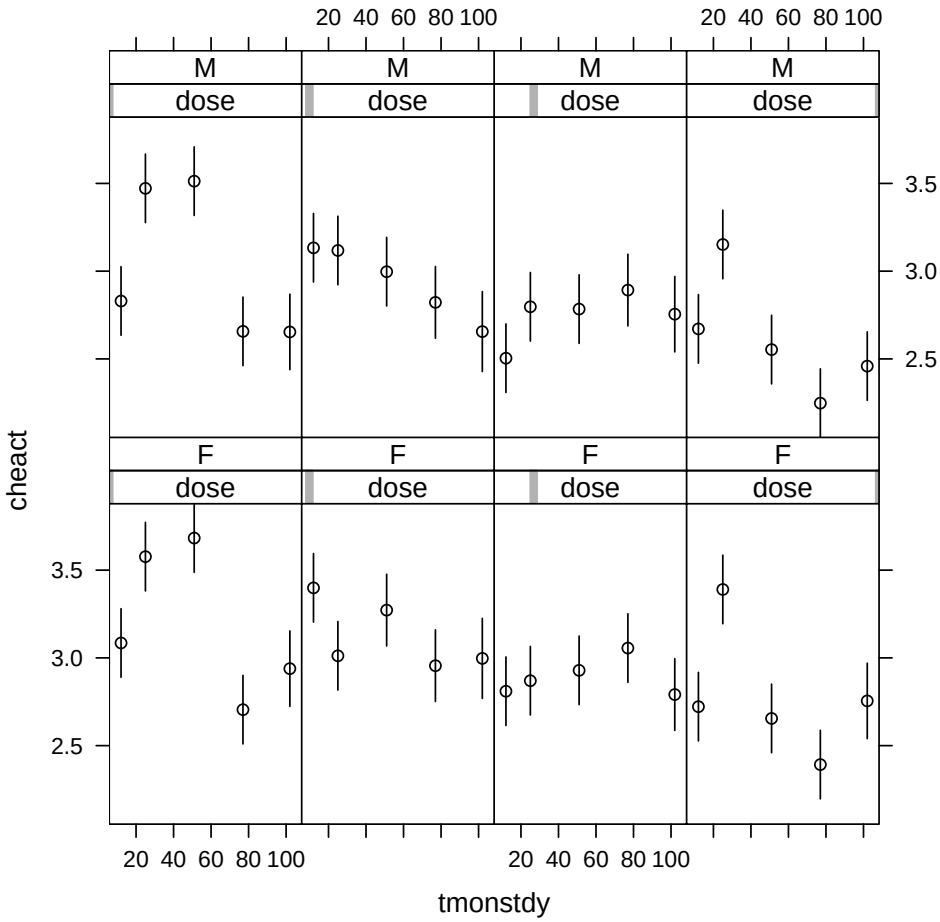
Next, extract from the effect names the values of `sex`, `tmonstdy`, and `dose`, as well as estimates and confidence intervals, into a dataframe, for plotting:

```
> Ints <- intervals(mns, which = "fixed")$fixed
> nms <- rownames(Ints)
> pdta <- data.frame(lower = Ints[, "lower"], cheact = Ints[, "est."],
+     upper = Ints[, "upper"], sex = factor(sapply(strsplit(nms,
+         ":"), function(x) gsub("sex", "", x[1]))), tmonstdy = as.numeric(sapply(strsplit(nms,
+         ":"), function(x) gsub("factor\\(tmonstdy\\)", "", x[2]))),
+     dose = as.numeric(sapply(strsplit(nms, ":"), function(x) gsub("factor\\(dose\\)",
+         "", x[3]))))
```

Now, plot it:

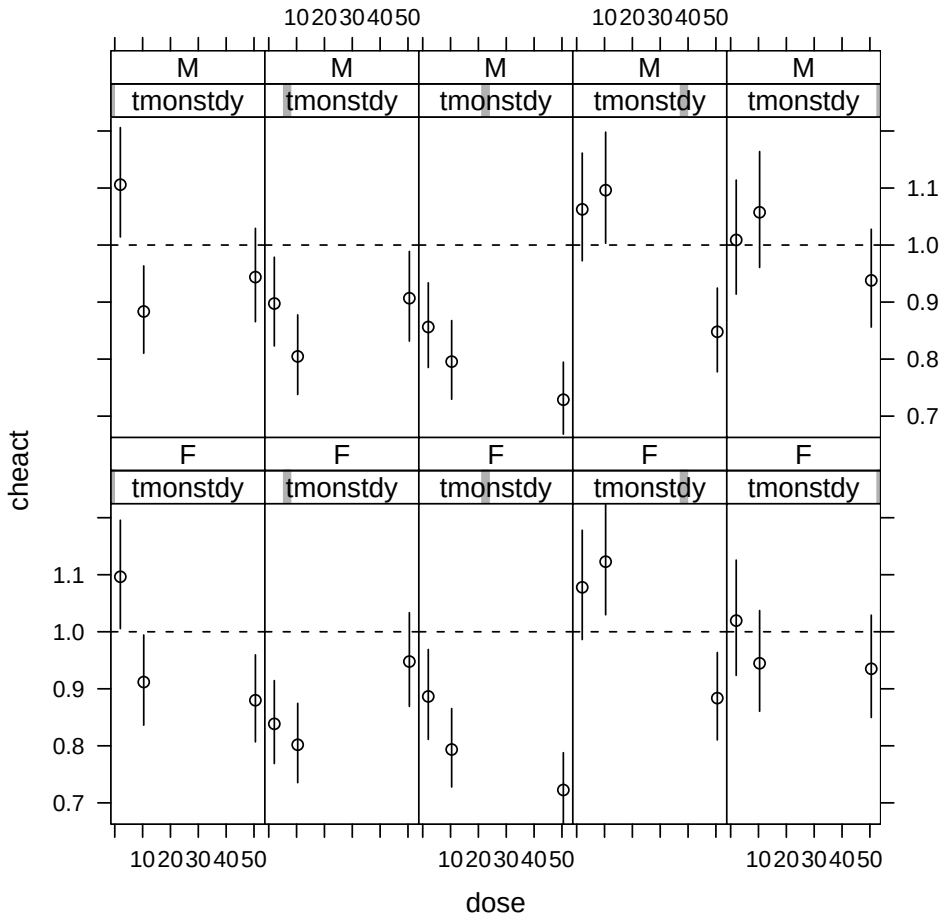


Plotting versus `tmonstdy`:



These data look pretty noisy. Does it help to normalize activity as a fraction of control? Model log(cheact):

```
> dta$lncheact <- log(dta$cheact)
> dta$sexXtmonstdy <- with(dta, interaction(sex, tmonstdy, drop = TRUE,
+     sep = ":"))
> dta$sexXtmonstdyXdose <- with(dta, interaction(sex, tmonstdy,
+     factor(dose), drop = TRUE, sep = ":"))
> lvls <- levels(dta$sexXtmonstdyXdose)
> lvls[grep(":0$", lvls)] <- "Control"
> levels(dta$sexXtmonstdyXdose) <- lvls
> drmod0 <- lme(lncheact ~ sexXtmonstdy + sexXtmonstdyXdose, data = dta,
+     random = ~1 | id)
> Ints <- exp(intervals(drmod0, which = "fixed")$fixed)
> nms <- rownames(Ints)
> indx <- grep("sexXtmonstdyXdose", nms)
> Ints <- Ints[indx, ]
> nms <- gsub("sexXtmonstdyXdose", "", nms[indx])
> pdta <- data.frame(lower = Ints[, "lower"], cheact = Ints[, "est."],
+     upper = Ints[, "upper"], sex = factor(sapply(strsplit(nms,
+         ":"), function(x) x[1])), tmonstdy = as.numeric(sapply(strsplit(nms,
+         ":"), function(x) x[2])), dose = as.numeric(sapply(strsplit(nms,
+         ":"), function(x) x[3])))
```



It is hard to see any consistent dose-response here. Stop without fitting any other model.

2 Summary

There is no apparent dose-response, so BMDs, etc. are not calculated.

species RAT

mrid `[1] "42648001"`

Save everything:

```
> save.image(file = "InhalationRBCDR.RData")
```

Save the results for incorporating into a database:

```
> propoxur.inhalation.rbc <- list(mrid = levels(dta$mrid), species = "RAT")
> save(propoxur.inhalation.rbc, file = file.path("..", "..", "01Summaries",
+     "propoxur.inhalation.rbc.RData"))
```