Dose-Time Response Modeling of Rat RBC AChE Activity: Propoxur Gavage Dosing

June 24, 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 gavage data for this analysis are in `PadillaData`. The following code prints out documentation for the datasets in use:

```
> printDataDoc(PadillaData)
```

```
----------------------------------------
Data set: PadillaData

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-time response for rat RBC. via gavage dosing. It includes an acute study that includes a time course at one dose and a dose-response at around the time of peak effect.

All the data used for this DR model are in `PadillaData`. The propoxur data need to be extracted, then several variables renamed (cheact = RBC.R, tmpstds == time) added to the dataset.

Set up the analysis dataset.

```
> dta <- subset(PadillaData, chemical %in% "propoxur", select = c("dose",
+     "RBC.R", "TMPSTDS"))
> names(dta) <- c("dose", "cheact", "tmpstds")
> dta <- CleanUp(subset(dta, !is.na(cheact)))
```

Summary of this dataset:

```
> summary(dta)

      dose           cheact          tmpstds
 Min.   : 0.00   Min.   : 33.10   Min.   : 0.2500
 1st Qu.: 0.65   1st Qu.:202.42   1st Qu.: 0.6667
 Median :10.00   Median :312.97   Median : 0.6667
 Mean   :11.04   Mean   :285.95   Mean   : 2.9576
 3rd Qu.:20.00   3rd Qu.:361.98   3rd Qu.: 1.0000
 Max.   :20.00   Max.   :540.75   Max.   :24.0000

> with(dta, table(dose, tmpstds))
```

```
     tmpstds
dose  0.25 0.5 0.666666666666667 1 4 24
  0   1    1   5                 1 1 1
  0.3 0    0   5                 0 0 0
  1   0    0   5                 0 0 0
  3   0    0   5                 0 0 0
  10  0    0   5                 0 0 0
  20  5    5   5                 5 5 4
```

Padilla's data are from an acute study, with multiple doses at 0.67 hours, and multiple timepoints for 20 mg/kg.

There is a control animal at each time point, so set up a factor with a level for each time point, to allow the control to vary from time to time. We need to check whether these controls are heterogeneous.

```
> dta$ftime <- factor(dta$tmpstds)
```

2 Dose-Response Modeling

2.1 strategy

Use the model with simple exponential recovery (`tcmfn4()`), first. It looks as if the time to peak effect for all these chemicals is likely to be less than a half-hour (the earliest time point in this dataset), so the exponential recovery model is probably indistinguishable from the one with the more complex time course.

Fitting the model will follow these steps:

1. First, use `GetInitialValues()` to get starting values for the model against these data, and determine if we can estimate `lg` and `tz` of the dose-response parameters.
2. Next, fit `tcmfn4()` using the parameterizations determined in the previous step, and a constant variance model.
3. Try fitting with other variance models the same model for fixed and random effects, using AIC to identify the best variance model.
4. Are the control values heterogeneous?

2.2 Initial Values

Save the initial values so that we do not need to go through all this to re-run the analysis. Also, set the argument `delta` to 0.25, the earliest non-zero time point.

```
> formals(tcmfn4)$delta <- min(dta$tmpstds[dta$tmpstds > 0])
> initfile <- paste("initvals-RBC-DR-1.RData", sep = "")
> if (!file.exists(initfile)) {
+     lA.start <- lm(I(log(cheact)) ~ ftime - 1, data = CleanUp(subset(dta,
+         dose %in% 0)))
+     Start <- c(coef(lA.start), log(0.5), -2, log(1), log(1.5))
+     init1 <- GetInitialValues(cheact ~ tcmfn4(dose, tmpstds,
+         lA = lA, tz = tz, lD = lD, lg = lg, lTr = lTr), data = dta,
+         params = list(lA ~ ftime - 1, lD ~ 1, tz ~ 1, lg ~ 1,
+             lTr ~ 1), start = Start, weights = varPower(value = 1))
+     save(init1, file = initfile)
+ } else load(initfile)
> tmp <- t(init1$Redundancy[[1]]$Eigens)
> tmp <- tmp[-grep("^lA", rownames(tmp)), ]
> round(tmp[, 1:5], digits = 2)
```

```
           [,1]  [,2] [,3] [,4] [,5]
CondIndex 40.95 10.91 4.32 2.40 1.96
mu         0.05  0.18 0.45 0.82 1.00
lD         0.87  0.12 0.00 0.00 0.00
tz         0.76  0.24 0.00 0.00 0.00
lg         1.00  0.00 0.00 0.00 0.00
lTr        0.02  0.03 0.89 0.01 0.00
```

The largest condition index is associated with a linear combination of `lD`, `tz`, and `lg`. There is a very good chance that at least one of these will need to be fixed, and not estimated. Preliminary optimization forces `tz` to a very small value (-9.9), indicating that there is no need to force a larger horizontal asymptote. For other chemicals and endpoints, we have been setting `tz` to -10 in this case, and will do so here.There are no repeated measures in this study, so we can use gnls to do the entire analysis:

```
> Start <- init1$start$beta
> Start <- Start[-grep("tz", names(Start))]
> formals(tcmfn4)$tz <- -10
> drmod1 <- gnls(cheact ~ tcmfn4(dose, tmpstds, lA = lA, lD = lD,
+     lg = lg, lTr = lTr), data = dta, params = list(lA ~ ftime -
+     1, lD ~ 1, lg ~ 1, lTr ~ 1), start = Start)
```

Now, plot the residuals, to see if we need to try to add `varPower()` to the `weights`:

```
> print(plot(drmod1))
```

This looks pretty good as is. There are three large residuals, in three different time and dose groups. What happens if we delete those observations and refit?

```
> dta2 <- dta[-which(resid(drmod1, type = "p") > 2), ]
> Start <- coef(drmod1)
> drmod2 <- gnls(cheact ~ tcmfn4(dose, tmpstds, lA = lA, lD = lD,
+     lg = lg, lTr = lTr), data = dta2, params = list(lA ~ ftime -
+     1, lD ~ 1, lg ~ 1, lTr ~ 1), start = Start)
```

Given the width of the confidence intervals, the difference is not very important:

```
> intervals(drmod1)[[1]][-grep("^lA", names(coef(drmod1))), ]
```

```
         lower       est.       upper
lD  -1.6175831  0.4292008  2.47598467
lg  -0.7058106  0.1127963  0.93140327
lTr -1.1251837 -0.6010059 -0.07682799
```

```
> intervals(drmod2)[[1]][-grep("^lA", names(coef(drmod1))), ]
```

```
         lower       est.      upper
lD  -0.8320133  0.6208227  2.0736586
lg  -0.4546040  0.1701693  0.7949427
lTr -0.9596003 -0.5405218 -0.1214432
```

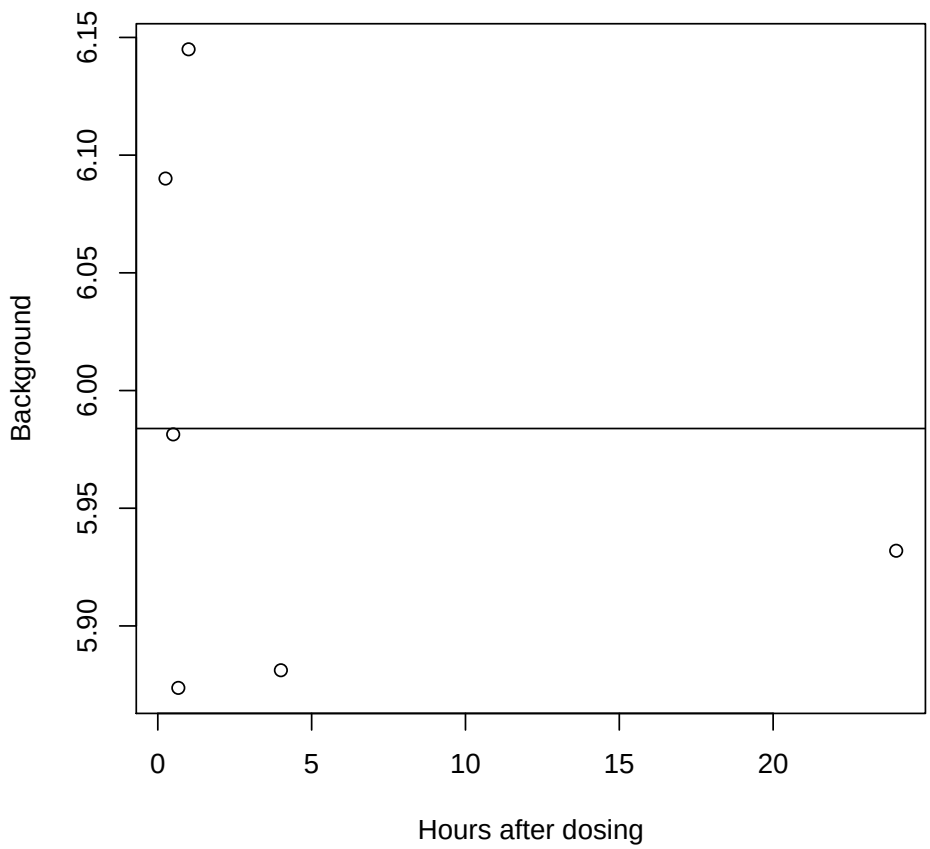
Dropping the three extreme points results in tightening the confidence bounds a bit, and raising the estimate of both `lD` and `lTr` a bit. With no objective reason for dropping the three extreme points, and because they are within range of the data, we go with the original fit and all the data.

Are the control values heterogeneous?

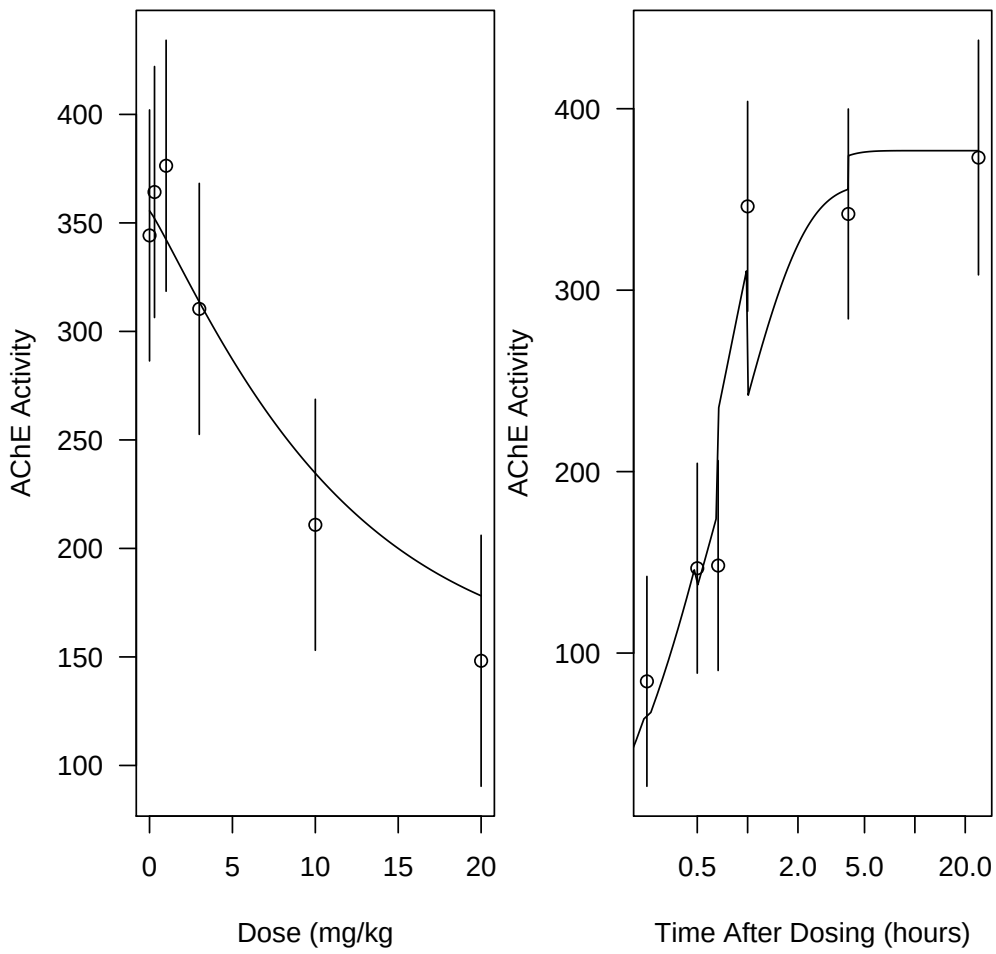
```
> nms <- names(getParms("^lA", coef(drmod1)))
> L <- diag(length(nms)) - matrix(1, nrow = length(nms), ncol = length(nms))/length(nms)
> L <- L[-nrow(L), ]
> colnames(L) <- nms
> anova(drmod1, L = L)
```

```
Denom. DF: 50
 F-test for linear combination(s)
  lA.ftime0.25 lA.ftime0.5 lA.ftime0.666666666666667  lA.ftime1  lA.ftime4
1    0.8333333  -0.1666667                -0.1666667 -0.1666667 -0.1666667
2   -0.1666667   0.8333333                -0.1666667 -0.1666667 -0.1666667
3   -0.1666667  -0.1666667                 0.8333333 -0.1666667 -0.1666667
4   -0.1666667  -0.1666667                -0.1666667  0.8333333 -0.1666667
5   -0.1666667  -0.1666667                -0.1666667 -0.1666667  0.8333333
  lA.ftime24
1 -0.1666667
2 -0.1666667
3 -0.1666667
4 -0.1666667
5 -0.1666667
  numDF  F-value p-value
1     5 1.532958  0.1965
```

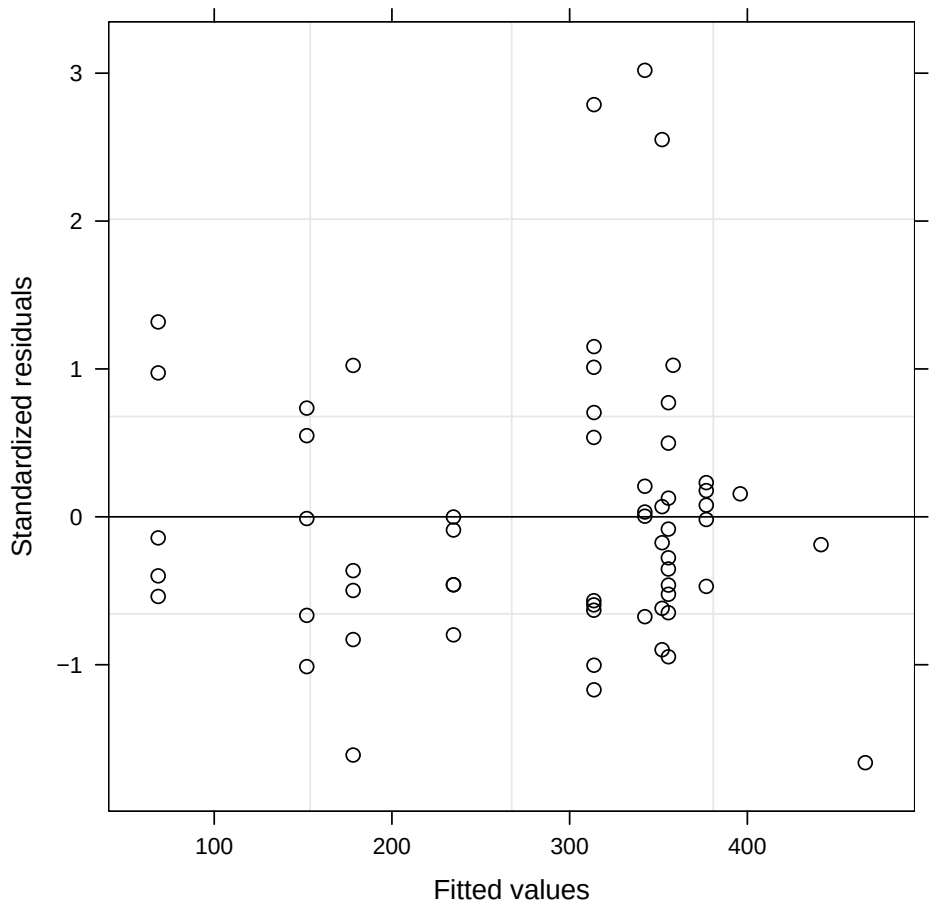
No. Calculate the deviations from their mean, and plot versus time:

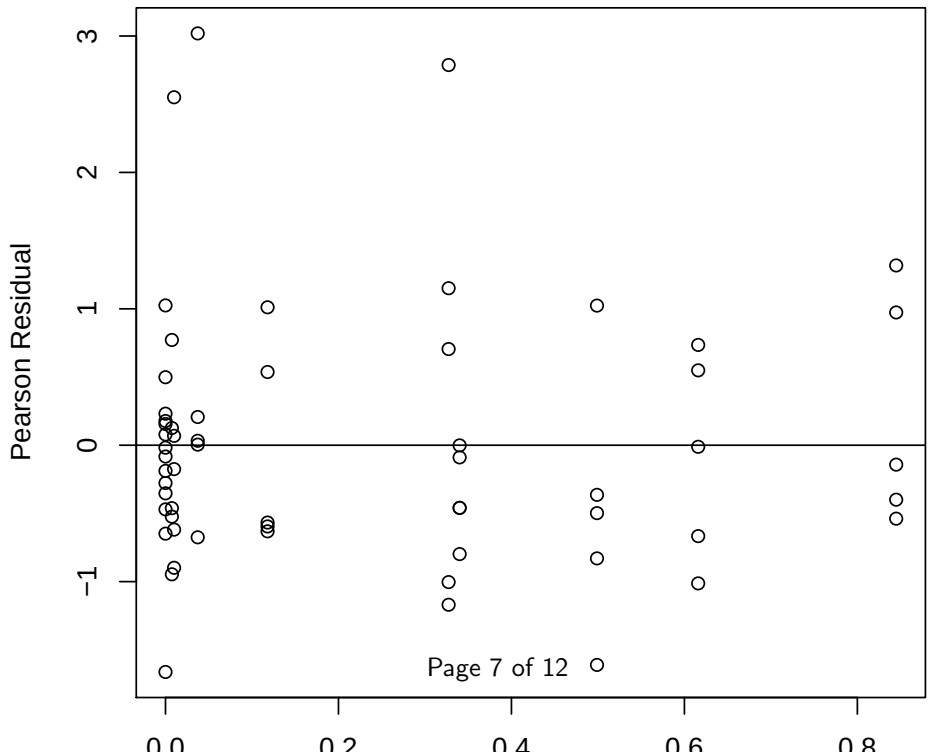


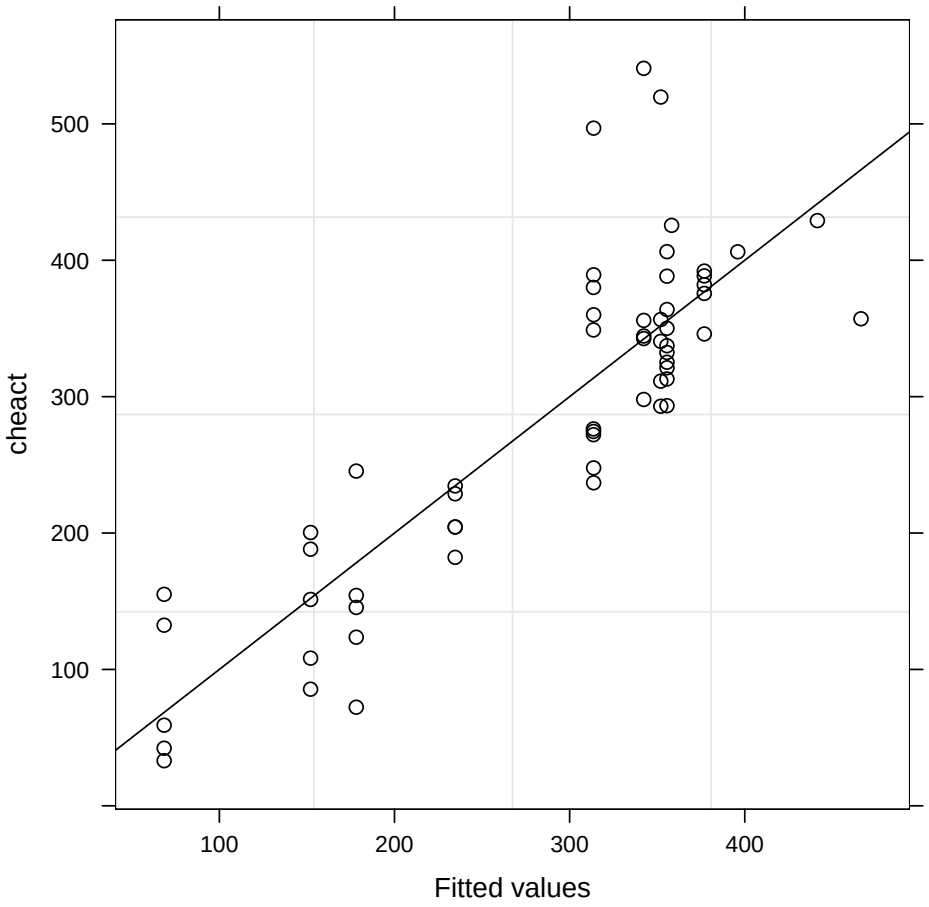
Diagnostic graphics for the model contained in `drmod1`: First, the dose-response and recovery curves with means and 95% confidence intervals from the data.

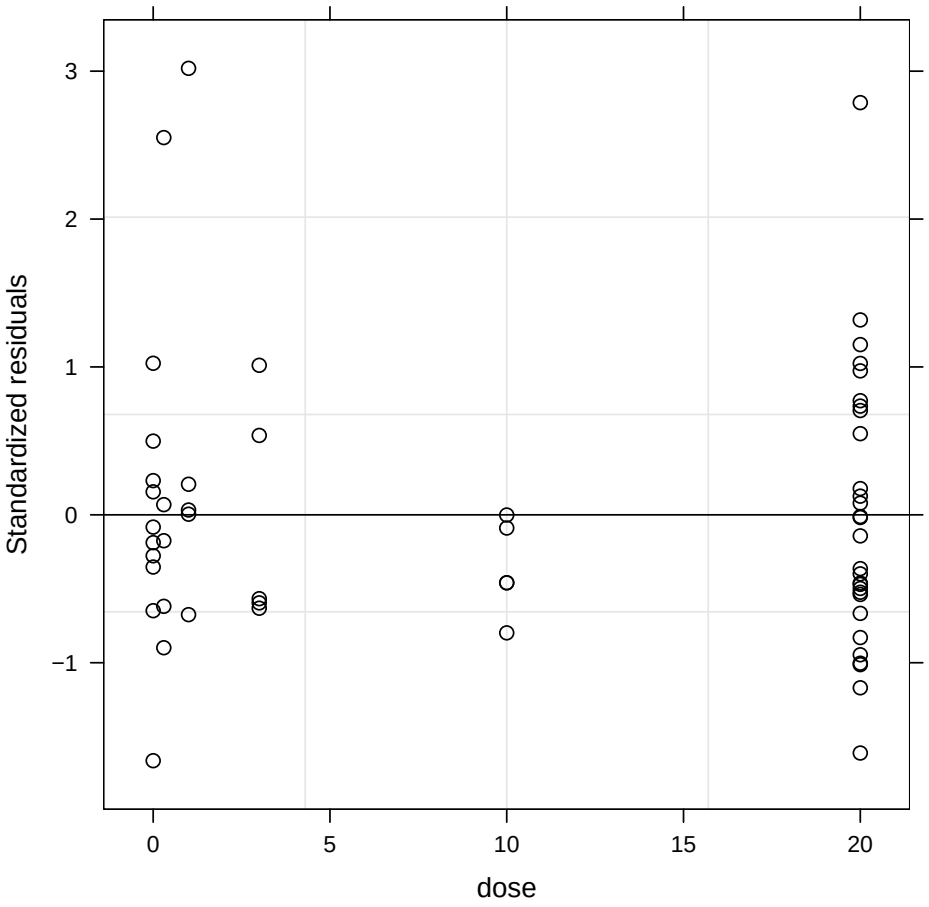


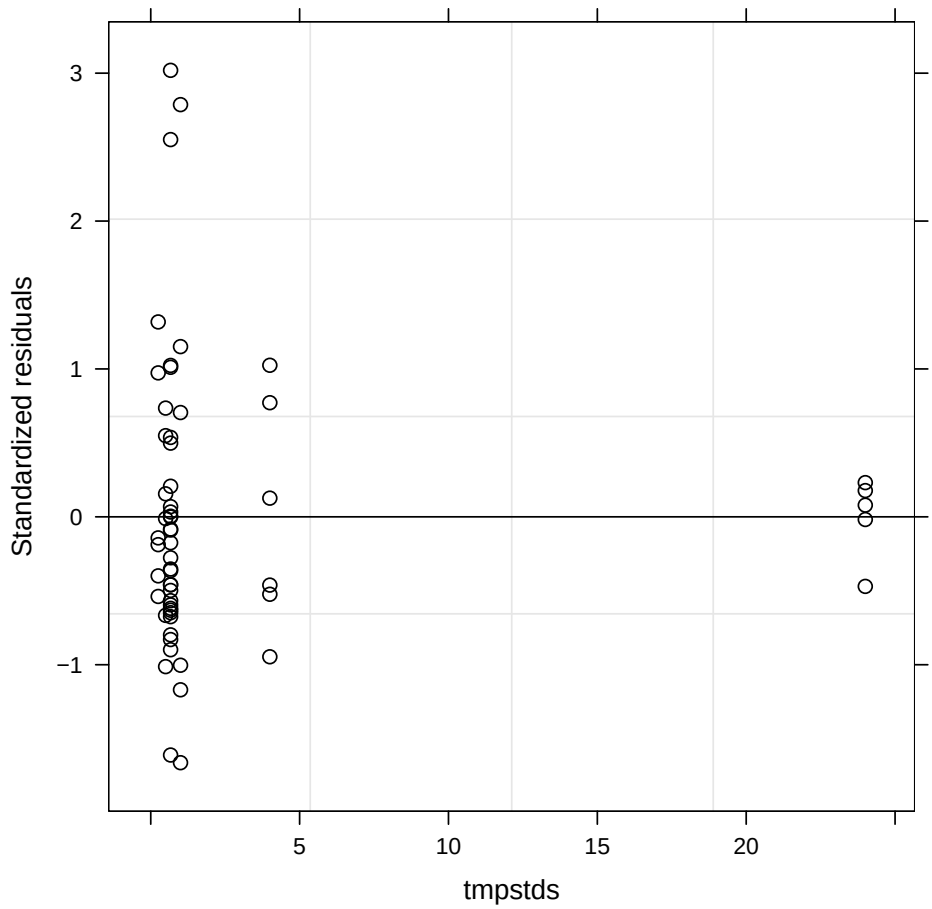
(The segmentation of the recovery curve is due to the fact that the background level varies across the recovery times).


Standardized residuals
Fitted values
3
2
1
0
−1
100
200
300
400


Pearson Residual
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8


cheact
500
400
300
200
100
100
200
300
400
Fitted values


Standardized residuals
3
2
1
0
−1
0
5
10
15
20
dose



Details of this final model fit:

```
> tTab <- summary(drmod1)$tTable
> Ints <- intervals(drmod1, which = "coef")$coef
> Ints90 <- intervals(drmod1, which = "coef", level = 0.9)$coef
> summary(drmod1)
```

```
Generalized nonlinear least squares fit
  Model: cheact ~ tcmfn4(dose, tmpstds, lA = lA, lD = lD, lg = lg, lTr = lTr)
  Data: dta
       AIC      BIC    logLik
  671.5511 692.3265 -325.7755

Coefficients:
                              Value Std.Error  t-value p-value
lA.ftime0.25               6.090061 0.1481569 41.10549  0.0000
lA.ftime0.5                5.981374 0.1353027 44.20735  0.0000
lA.ftime0.666666666666667  5.873691 0.0587778 99.93037  0.0000
lA.ftime1                  6.144966 0.1053726 58.31656  0.0000
lA.ftime4                  5.881187 0.0757194 77.67086  0.0000
lA.ftime24                 5.931948 0.0779858 76.06447  0.0000
lD                         0.429201 1.0190309  0.42119  0.6754
lg                         0.112796 0.4075593  0.27676  0.7831
```

```
lTr                       -0.601006 0.2609721 -2.30295  0.0255

 Correlation:
                          lA.0.2 lA.0.5 lA.0.6 lA.ft1 lA.ft4 lA.f24 lD
lA.ftime0.5                0.048
lA.ftime0.666666666666667 -0.012  0.051
lA.ftime1                 -0.069  0.178  0.118
lA.ftime4                 -0.017  0.019  0.016  0.063
lA.ftime24                 0.000  0.000  0.000  0.000  0.000
lD                         0.009  0.048 -0.690  0.078  0.009  0.000
lg                         0.071  0.078 -0.680  0.032 -0.002  0.000  0.980
lTr                       -0.188  0.176  0.161  0.640  0.097  0.000  0.088
                          lg
lA.ftime0.5
lA.ftime0.666666666666667
lA.ftime1
lA.ftime4
lA.ftime24
lD
lg
lTr                       -0.033

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.6623590 -0.5810053 -0.1421795  0.3649744  3.0193900

Residual standard error: 65.72229
Degrees of freedom: 59 total; 50 residual
```

3 Summary

The critical estimates from this analysis are listed below. They are printed with greater than usual precision, in case they are to be used in further computation. For reporting, round to two or three significant digits. BMD has units mg/kg, and times are in hours.

species RAT

mrid Padilla

`lD` **(standard error)** 0.429200793833359 (1.01903094000347)

BMD (95% CI) 1.53602942867161 (0.19837758165166, 11.8934124819013)

BMDL, the one-sided lower 95% CL 0.278427304362820

`lTr (standard error)` -0.601005853369822 (0.260972088434471)

Recovery Half-life (95% CI) 0.54825988959502 (0.324592827815927, 0.926049132266116)

Save everything:

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

Save the results for incorporating into a database:

```
> propoxur.oral.rbc <- list(mrid = "Padilla", species = "RAT",
+     BMDs = list(male = list(lD = tTab["lD", 1], lD.se = tTab["lD",
+         2], BMD = exp(Ints["lD", 2]), BMD.CI = exp(Ints["lD",
+         c(1, 3)]), BMDL = exp(Ints90["lD", 1]))), HalfLives = list(male = list(lTr = tTab["lTr",
+         "Value"], lTr.se = tTab["lTr", "Std.Error"], Tr = exp(Ints["lTr",
+         "est."]), Tr.CI = exp(Ints["lTr", c("lower", "upper")]))))
> save(propoxur.oral.rbc, file = file.path("..", "..", "01Summaries",
+     "propoxur.oral.rbc.RData"))
```