Dose-Time Response Modeling of Rat RBC AChE Activity: Carbaryl Dermal Dosing

June 28, 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
```

All the rat gavage data for this analysis are in `repeated_dermal`. The following code prints out documentation for the datasets in use:

```
> printDataDoc(repeated_dermal)
```

```
----------------------------------------
Data set: repeated_dermal

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 red blood cells. via dermal dosing. It includes a single subchronic study.

Set up the analysis dataset.

```
> dta <- CleanUp(subset(repeated_dermal, chemical %in% "Carbaryl" &
+     !is.na(cheact), select = c("cheact", "dose", "time", "sex",
+     "id", "mrid")))
> names(dta)[grep("time", names(dta))] <- "tmonstdy"
```

Summary of the relevant variables in this dataset:

```
> summary(dta)

     cheact          dose          tmonstdy     sex           id
 Min.   : 582   Min.   :  0.0   Min.   :-4.00   F:360   65271  :  9
 1st Qu.:1088   1st Qu.: 15.0   1st Qu.: 5.00   M:360   65272  :  9
 Median :1230   Median : 35.0   Median :12.00           65273  :  9
 Mean   :1215   Mean   : 42.5   Mean   :11.56           65274  :  9
 3rd Qu.:1349   3rd Qu.: 62.5   3rd Qu.:19.00           65275  :  9
 Max.   :2024   Max.   :100.0   Max.   :26.00           65276  :  9
                                                        (Other):666
       mrid
 45630601:720
```

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

     sex
dose  F  M
  0   90 90
  20  90 90
  50  90 90
  100 90 90

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

, , sex = F

     tmonstdy
dose  -4 1  5  8  12 15 19 22 26
  0   10 10 10 10 10 10 10 10 10
  20  10 10 10 10 10 10 10 10 10
  50  10 10 10 10 10 10 10 10 10
  100 10 10 10 10 10 10 10 10 10

, , sex = M

     tmonstdy
dose  -4 1  5  8  12 15 19 22 26
  0   10 10 10 10 10 10 10 10 10
  20  10 10 10 10 10 10 10 10 10
  50  10 10 10 10 10 10 10 10 10
  100 10 10 10 10 10 10 10 10 10
```

Set up a new dose variable that is 0 for pre-dosing measurements.

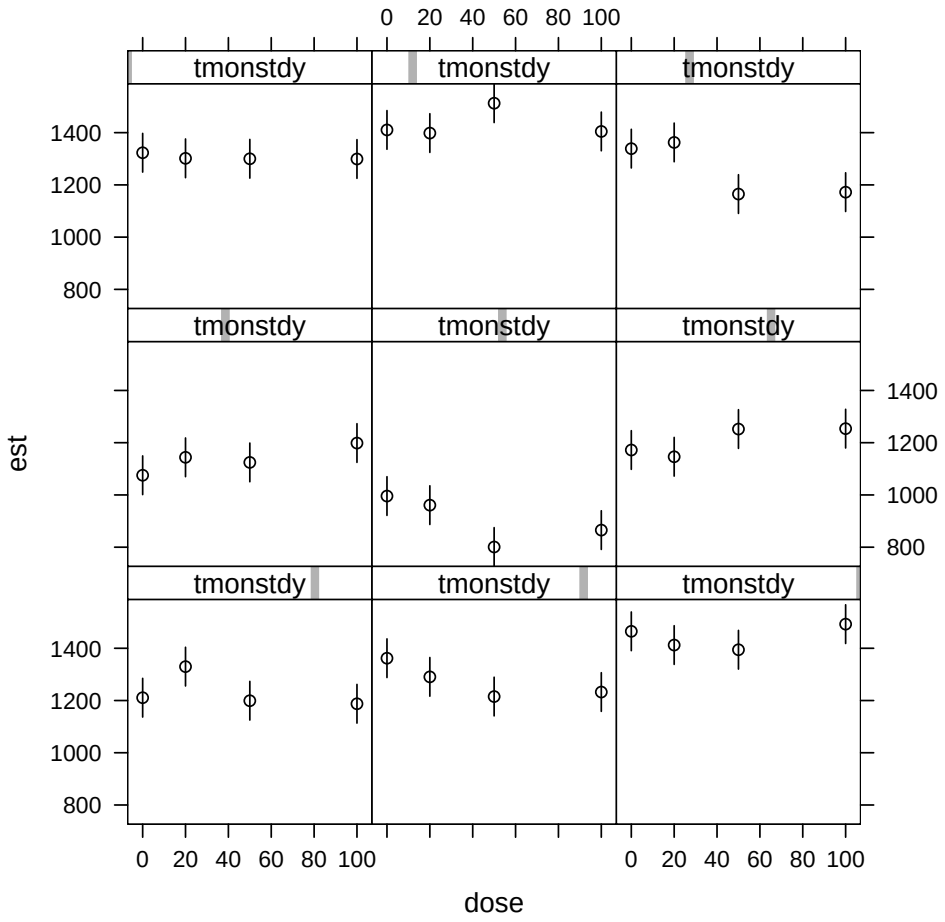
```
> dta$dose2 <- ifelse(dta$tmonstdy < 0, 0, dta$dose)
```

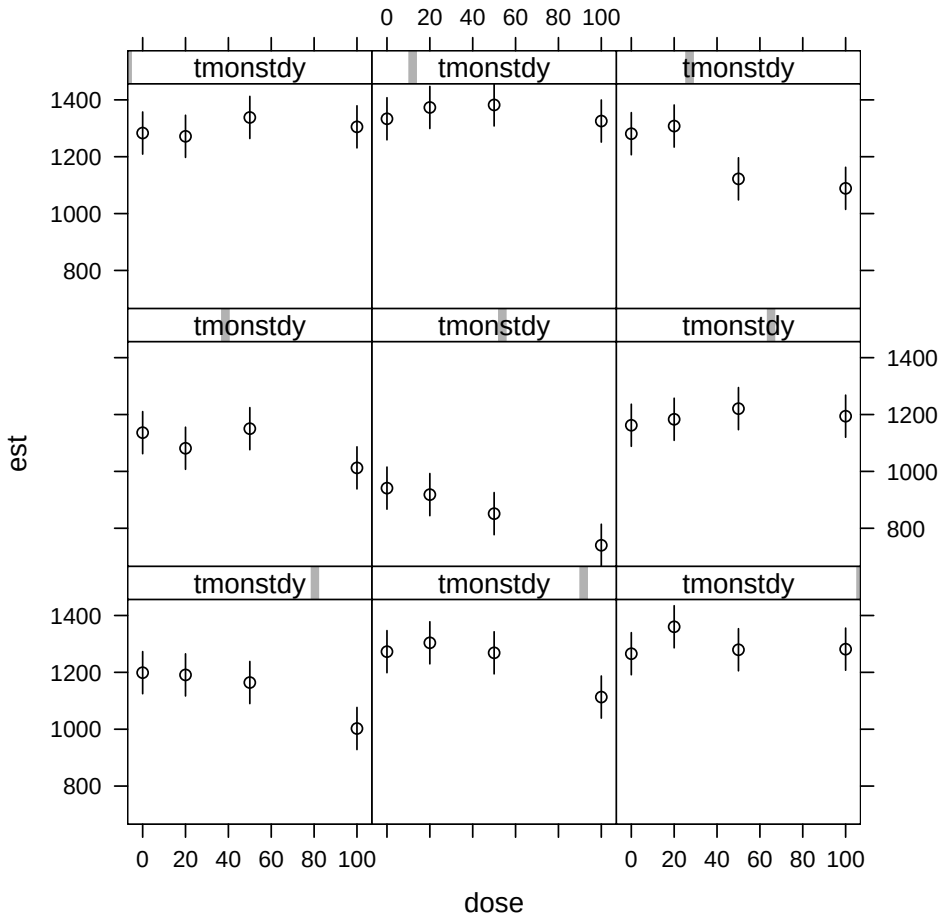
2 Dose-Response Modeling

2.1 strategy

There are only four doses (including control) here, so the dose-response model needs to be simple. The main complications are the repeated measures design, and the multiple time points. The following figures show the simple dose-means as a function of dose at each time point:

```
> out <- lme(cheact ~ sex:factor(tmonstdy):factor(dose) - 1, random = ~1 |
+     id, data = dta)
> ints <- intervals(out, which = "fixed")$fixed
> nm <- rownames(ints)
> pd <- data.frame(lc = ints[, "lower"], uc = ints[, "upper"],
+     est = ints[, "est."], sex = factor(sapply(strsplit(nm, ":"),
+         function(x) gsub("sex", "", x[1]))), tmonstdy = as.numeric(sapply(strsplit(nm,
+         ":"), function(x) gsub("factor\\(tmonstdy\\)", "", x[2]))),
+     dose = as.numeric(sapply(strsplit(nm, ":"), function(x) gsub("factor\\(dose\\)",
+         "", x[3]))))
> rownames(pd) <- 1:nrow(pd)
> pdM <- CleanUp(subset(pd, sex == "M"))
> pdF <- CleanUp(subset(pd, sex == "F"))
```


tmonstdy
est
dose
0
20
40
60
80
100
800
1000
1200
1400



Most of these curves look pretty flat. Try a simple exponential dose-response function. This is easily implemented by just taking the log of `cheact`:

```
> dta$ftmonstdy <- factor(dta$tmonstdy)
> lvls <- levels(dta$ftmonstdy)
> lvls[1] <- lvls[2]
> levels(dta$ftmonstdy) <- lvls
> drmod1 <- lme(I(log(cheact)) ~ sex:factor(tmonstdy) - 1 + sex:ftmonstdy:dose2,
+     data = dta, random = ~1 | id)
> mests <- getParms("dose", fixed.effects(drmod1))
> Fmests <- getParms("sexF", mests)
> Mmests <- getParms("sexM", mests)
```

Are the slope estimates different from each other (by sex)?

```
> NF <- length(Fmests)
> L <- diag(NF) - matrix(1, nrow = NF, ncol = NF)/NF
> L <- L[-NF, ]
> colnames(L) <- names(Fmests)
> anova(drmod1, L = L)

F-test for linear combination(s)
  sexF:ftmonstdy1:dose2 sexF:ftmonstdy5:dose2 sexF:ftmonstdy8:dose2
```

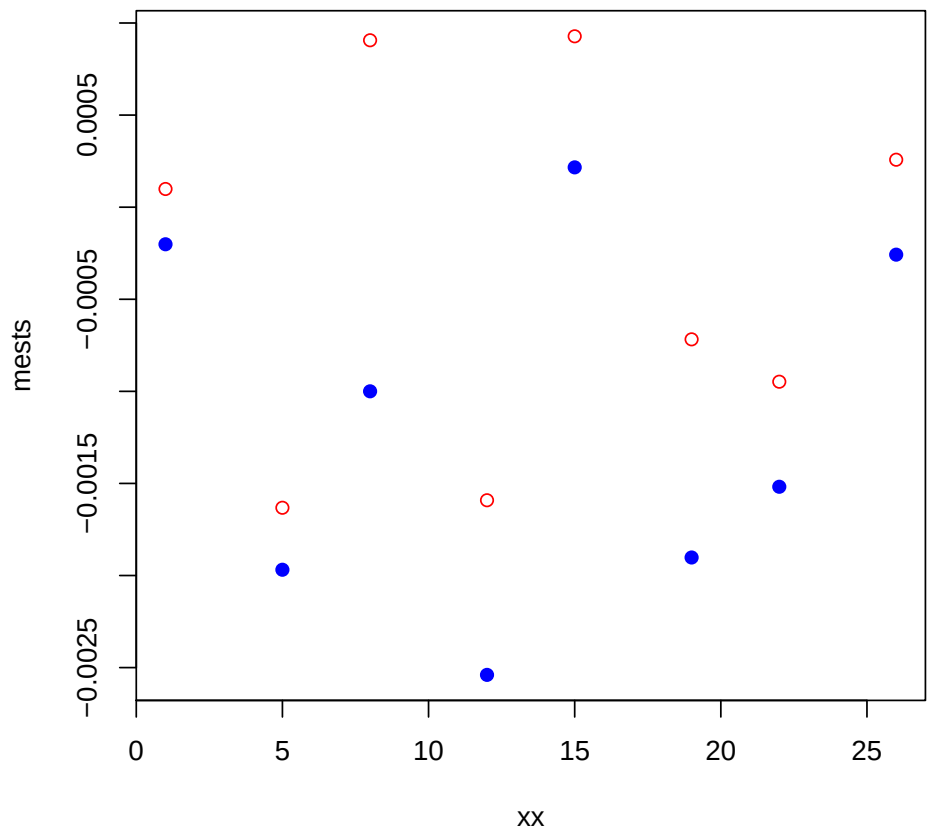
```
1                 0.875                -0.125                -0.125
2                -0.125                 0.875                -0.125
3                -0.125                -0.125                 0.875
4                -0.125                -0.125                -0.125
5                -0.125                -0.125                -0.125
6                -0.125                -0.125                -0.125
7                -0.125                -0.125                -0.125
  sexF:ftmonstdy12:dose2 sexF:ftmonstdy15:dose2 sexF:ftmonstdy19:dose2
1                 -0.125                 -0.125                 -0.125
2                 -0.125                 -0.125                 -0.125
3                 -0.125                 -0.125                 -0.125
4                  0.875                 -0.125                 -0.125
5                 -0.125                  0.875                 -0.125
6                 -0.125                 -0.125                  0.875
7                 -0.125                 -0.125                 -0.125
  sexF:ftmonstdy22:dose2 sexF:ftmonstdy26:dose2
1                 -0.125                 -0.125
2                 -0.125                 -0.125
3                 -0.125                 -0.125
4                 -0.125                 -0.125
5                 -0.125                 -0.125
6                 -0.125                 -0.125
7                  0.875                 -0.125
  numDF denDF  F-value p-value
1     7   607 8.190718  <.0001
```

```
> NM <- length(Mmests)
> L <- diag(NM) - matrix(1, nrow = NM, ncol = NM)/NM
> L <- L[-NM, ]
> colnames(L) <- names(Mmests)
> anova(drmod1, L = L)
```

```
F-test for linear combination(s)
  sexM:ftmonstdy1:dose2 sexM:ftmonstdy5:dose2 sexM:ftmonstdy8:dose2
1                 0.875                -0.125                -0.125
2                -0.125                 0.875                -0.125
3                -0.125                -0.125                 0.875
4                -0.125                -0.125                -0.125
5                -0.125                -0.125                -0.125
6                -0.125                -0.125                -0.125
7                -0.125                -0.125                -0.125
  sexM:ftmonstdy12:dose2 sexM:ftmonstdy15:dose2 sexM:ftmonstdy19:dose2
1                 -0.125                 -0.125                 -0.125
2                 -0.125                 -0.125                 -0.125
3                 -0.125                 -0.125                 -0.125
4                  0.875                 -0.125                 -0.125
5                 -0.125                  0.875                 -0.125
6                 -0.125                 -0.125                  0.875
7                 -0.125                 -0.125                 -0.125
  sexM:ftmonstdy22:dose2 sexM:ftmonstdy26:dose2
1                 -0.125                 -0.125
2                 -0.125                 -0.125
3                 -0.125                 -0.125
4                 -0.125                 -0.125
5                 -0.125                 -0.125
```

```
6                  -0.125                  -0.125
7                   0.875                  -0.125
  numDF denDF  F-value p-value
1     7   607 7.563892  <.0001
```

Yes. Both P-values are much less that 10^{-4}. Does there seem to be a pattern with respect to `tmonstdy`?



There is certainly no apparent pattern, though male and female slopes clearly covary across times. It would be nice to be able to treat `m` as a random effect, but R does not seem to have a function for fitting linear models with this kind of crossed nesting: subject crossed with times, in this case. Note also that a number of values of `m` are *positive*, which indicates an *increase* in AChE activity (negative inhibition) with increasing AChE dose. It may well be that some of the time points were victims of problems in implementation of the Ellman assay. If that were so, the average of the slope estimates would be biased towards smaller (absolute) values, and the more extreme (negative) estimates would be closer to the correct values. Here are all the slope estimates, with standard errors and Wald p-values (not adjusted for multiple comparisons).

```
> tmp <- summary(drmod1)$tTable[grep("dose", names(fixed.effects(drmod1))),
+     ]
> rownames(tmp) <- paste(sapply(strsplit(rownames(tmp), ":"), function(x) gsub("sex",
+     "", x[1])), sapply(strsplit(rownames(tmp), ":"), function(x) sprintf("%02d",
+     as.integer(gsub("ftmonstdy", "", x[2])))), sep = " : ")
> indx <- order(rownames(tmp))
> tmp <- tmp[indx, ]
> tmp
```

```
               Value    Std.Error  DF    t-value      p-value
F : 01  9.892982e-05 0.0004073544 607  0.2428594 8.081964e-01
F : 05 -1.631968e-03 0.0004073544 607 -4.0062617 6.934678e-05
F : 08  9.064844e-04 0.0004073544 607  2.2252968 2.642797e-02
F : 12 -1.591139e-03 0.0004073544 607 -3.9060306 1.043623e-04
F : 15  9.279727e-04 0.0004073544 607  2.2780476 2.307047e-02
F : 19 -7.174972e-04 0.0004073544 607 -1.7613588 7.868112e-02
F : 22 -9.476064e-04 0.0004073544 607 -2.3262457 2.033380e-02
F : 26  2.577731e-04 0.0004073544 607  0.6327982 5.271037e-01
M : 01 -2.012355e-04 0.0004073544 607 -0.4940059 6.214807e-01
M : 05 -1.968472e-03 0.0004073544 607 -4.8323330 1.711262e-06
M : 08 -9.996785e-04 0.0004073544 607 -2.4540757 1.440453e-02
M : 12 -2.539418e-03 0.0004073544 607 -6.2339286 8.518392e-10
M : 15  2.162864e-04 0.0004073544 607  0.5309540 5.956449e-01
M : 19 -1.901993e-03 0.0004073544 607 -4.6691368 3.725988e-06
M : 22 -1.517900e-03 0.0004073544 607 -3.7262398 2.125044e-04
M : 26 -2.576439e-04 0.0004073544 607 -0.6324809 5.273109e-01
```

The row labels indicate the sex and the time point (in days). Note that a few, the 8 and 15 day timepoints in females are significantly *positive*. Now, sort out the estimates that are significantly *negative* (one-sided tests):

```
> signif <- tmp[, "p-value"] < 0.1 & tmp[, "Value"] < 0
> tmp2 <- tmp[signif, ]
> tmp2
```

```
               Value    Std.Error  DF   t-value      p-value
F : 05 -0.0016319683 0.0004073544 607 -4.006262 6.934678e-05
F : 12 -0.0015911387 0.0004073544 607 -3.906031 1.043623e-04
F : 19 -0.0007174972 0.0004073544 607 -1.761359 7.868112e-02
F : 22 -0.0009476064 0.0004073544 607 -2.326246 2.033380e-02
M : 05 -0.0019684721 0.0004073544 607 -4.832333 1.711262e-06
M : 08 -0.0009996785 0.0004073544 607 -2.454076 1.440453e-02
M : 12 -0.0025394182 0.0004073544 607 -6.233929 8.518392e-10
M : 19 -0.0019019934 0.0004073544 607 -4.669137 3.725988e-06
M : 22 -0.0015179001 0.0004073544 607 -3.726240 2.125044e-04
```

Note that with one exception, the same time points are significantly negative in males and females.

Now, we need to convert these slopes to BMDs:

In this model, the BMD for BMR $\times$ 100 % inhibition is

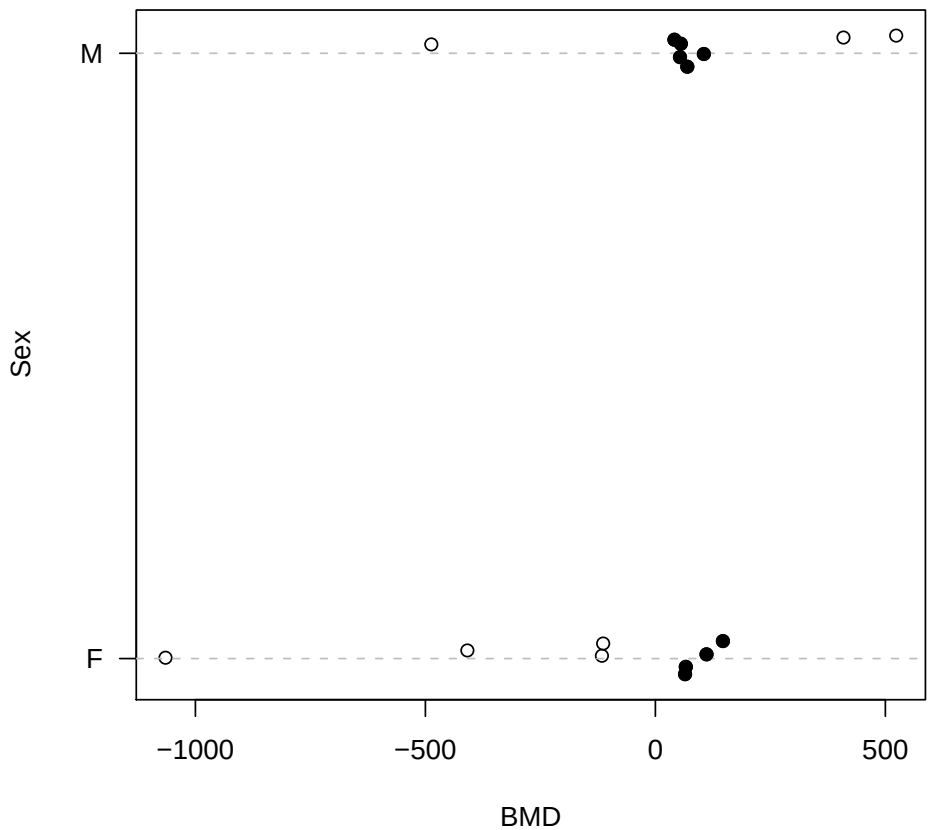
$$\frac{log(1 - BMR)}{\beta}$$

where β is `m`, the coefficient of `dose2` in the linear model. Here are the corresponding estimates of BMD:

```
> log(0.9)/tmp2[, "Value", drop = FALSE]
```

```
           Value
F : 05  64.56039
F : 12  66.21705
F : 19 146.84449
F : 22 111.18595
M : 05  53.52401
M : 08 105.39440
M : 12  41.49002
M : 19  55.39479
M : 22  69.41202
```

The following figure shows the BMD values for males and females plotted, to show the clustering around about 65 (values have been jittered around the horizontal lines so that idividual values are visible). Note that negative BMD values are not meaningful. They should, perhaps, be thought of as residing at $+\infty$. The BMDs corresponding to significantly negative slopes are the darkened points.



The mean of the female values is 97.2, with standard deviation 39.5, and for males is 65, with standard deviation 24.6.

It is probably interesting to note the corresponding estimates for `m`: Females: -0.00122, with standard deviation 0.00046, and for males: -0.00179, with standard deviation 0.000571.

Note that the estimated standard deviations are not much greater than the estimated standard error of about 0.00041 (see the table, above). Based on this, I will refit the model, parameterizing tmonstdy so that for females, times 5, 12, 19, and 22 days contribute to the BMD estimate, and for males, times 5, 8, 12, 19, and 22 days. That estimate will be the basis for the BMD estimate. We need an estimate of log(BMD) and its standard error to use in calculating RPFs. Taking the logarithm of the above formula:

$$lBMD = lD = log(-log(1-(BMR))) - log(-\beta)$$

where we replace β with the estimate of `m` from the linear model. To get an approximate standard error, use the delta method, which gives:

$$sd(lD) = sd(log(\hat{\beta})) = \frac{sd(\hat{\beta})}{-\hat{\beta}}$$

```
> sf <- interaction(dta$sex, dta$ftmonstdy, drop = TRUE, sep = ":")
> levels(sf)[c(4, 8, 12, 14)] <- "mF"
> levels(sf)[c(3, 5, 7, 10, 11)] <- "mM"
> drmod2 <- lme(I(log(cheact)) ~ sex:factor(tmonstdy) - 1 + sf:dose2,
+     data = dta, random = ~1 | id)
> tTab <- summary(drmod2)$tTable
> mM <- tTab["sfmM:dose2", "Value"]
> mM.se <- tTab["sfmM:dose2", "Std.Error"]
> mF <- tTab["sfmF:dose2", "Value"]
> mF.se <- tTab["sfmF:dose2", "Std.Error"]
> lD.M <- log(-log(0.9)) - log(-mM)
> lD.M.se <- mM.se/(-mM)
> lD.F <- log(-log(0.9)) - log(-mF)
> lD.F.se <- mF.se/(-mF)
> BMD.M <- exp(lD.M)
> BMD.M.CL <- exp(c(lD.M + qnorm(c(0.025, 0.975)) * lD.M.se))
> BMDL.M <- exp(lD.M + qnorm(0.05) * lD.M.se)
> BMD.F <- exp(lD.F)
> BMD.F.CL <- exp(c(lD.F + qnorm(c(0.025, 0.975)) * lD.F.se))
> BMDL.F <- exp(lD.F + qnorm(0.05) * lD.F.se)
```

Summary of this fit:

```
> summary(drmod2)

Linear mixed-effects model fit by REML
 Data: dta
        AIC      BIC   logLik
  -1087.659 -955.969 572.8294

Random effects:
 Formula: ~1 | id
        (Intercept)   Residual
StdDev:  0.05291511 0.08646488

Fixed effects: I(log(cheact)) ~ sex:factor(tmonstdy) - 1 + sf:dose2
                            Value   Std.Error  DF  t-value p-value
sexF:factor(tmonstdy)-4  7.171793 0.016028244 614 447.4472  0.0000
sexM:factor(tmonstdy)-4  7.167649 0.016028244 614 447.1887  0.0000
sexF:factor(tmonstdy)1   7.259195 0.023657856 614 306.8408  0.0000
sexM:factor(tmonstdy)1   7.215841 0.023657856 614 305.0083  0.0000
sexF:factor(tmonstdy)5   7.182495 0.019525094 614 367.8597  0.0000
sexM:factor(tmonstdy)5   7.160208 0.019218003 614 372.5782  0.0000
sexF:factor(tmonstdy)8   6.989468 0.023657856 614 295.4396  0.0000
sexM:factor(tmonstdy)8   7.069286 0.019218003 614 367.8471  0.0000
sexF:factor(tmonstdy)12  6.851063 0.019525094 614 350.8850  0.0000
sexM:factor(tmonstdy)12  6.825458 0.019218003 614 355.1596  0.0000
sexF:factor(tmonstdy)15  7.052315 0.023657856 614 298.0961  0.0000
sexM:factor(tmonstdy)15  7.068170 0.023657856 614 298.7663  0.0000
sexF:factor(tmonstdy)19  7.159787 0.019525094 614 366.6967  0.0000
sexM:factor(tmonstdy)19  7.106046 0.019218003 614 369.7599  0.0000
sexF:factor(tmonstdy)22  7.197217 0.019525094 614 368.6137  0.0000
sexM:factor(tmonstdy)22  7.193124 0.019218003 614 374.2909  0.0000
sexF:factor(tmonstdy)26  7.256209 0.023657856 614 306.7145  0.0000
sexM:factor(tmonstdy)26  7.172639 0.023657856 614 303.1821  0.0000
```

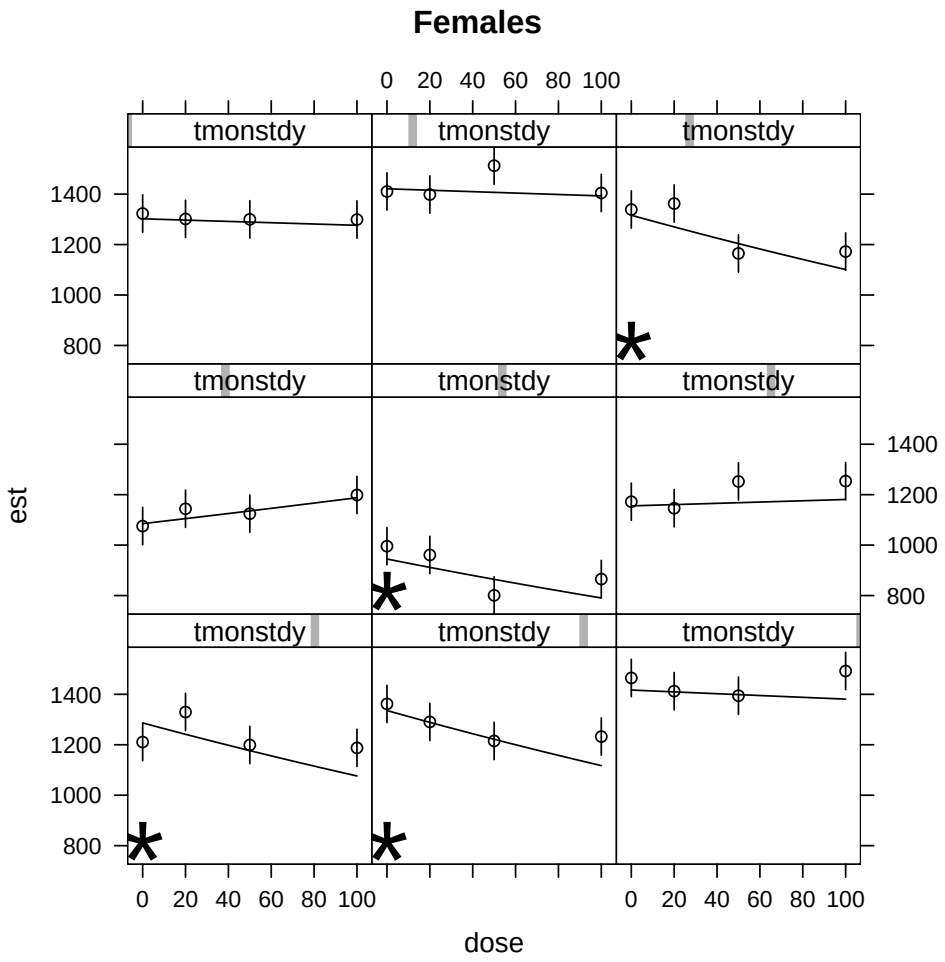
```
sfM:1:dose2              -0.000200 0.000409432 614  -0.4895  0.6247
sfF:1:dose2               0.000098 0.000409432 614   0.2405  0.8100
sfmM:dose2               -0.001785 0.000249486 614  -7.1533  0.0000
sfmF:dose2               -0.001223 0.000262355 614  -4.6598  0.0000
sfF:8:dose2               0.000906 0.000409432 614   2.2128  0.0273
sfM:15:dose2              0.000217 0.000409432 614   0.5303  0.5961
sfF:15:dose2              0.000927 0.000409432 614   2.2653  0.0238
sfM:26:dose2             -0.000257 0.000409432 614  -0.6272  0.5307
sfF:26:dose2              0.000257 0.000409432 614   0.6284  0.5300
 Correlation:
                         sF:()- sM:()- sxF:()1 sxM:()1 sF:()5 sM:()5 sF:()8
sexM:factor(tmonstdy)-4  0.000
sexF:factor(tmonstdy)1   0.185  0.000
sexM:factor(tmonstdy)1   0.000  0.185  0.000
sexF:factor(tmonstdy)5   0.224  0.000  0.292   0.000
sexM:factor(tmonstdy)5   0.000  0.227  0.000   0.297   0.000
sexF:factor(tmonstdy)8   0.185  0.000  0.241   0.000   0.292  0.000
sexM:factor(tmonstdy)8   0.000  0.227  0.000   0.297   0.000  0.494  0.000
sexF:factor(tmonstdy)12  0.224  0.000  0.292   0.000   0.510  0.000  0.292
sexM:factor(tmonstdy)12  0.000  0.227  0.000   0.297   0.000  0.494  0.000
sexF:factor(tmonstdy)15  0.185  0.000  0.241   0.000   0.292  0.000  0.241
sexM:factor(tmonstdy)15  0.000  0.185  0.000   0.241   0.000  0.297  0.000
sexF:factor(tmonstdy)19  0.224  0.000  0.292   0.000   0.510  0.000  0.292
sexM:factor(tmonstdy)19  0.000  0.227  0.000   0.297   0.000  0.494  0.000
sexF:factor(tmonstdy)22  0.224  0.000  0.292   0.000   0.510  0.000  0.292
sexM:factor(tmonstdy)22  0.000  0.227  0.000   0.297   0.000  0.494  0.000
sexF:factor(tmonstdy)26  0.185  0.000  0.241   0.000   0.292  0.000  0.241
sexM:factor(tmonstdy)26  0.000  0.185  0.000   0.241   0.000  0.297  0.000
sfM:1:dose2              0.000  0.000  0.000  -0.736   0.000 -0.194  0.000
sfF:1:dose2              0.000  0.000 -0.736   0.000  -0.191  0.000 -0.157
sfmM:dose2               0.000  0.000  0.000  -0.258   0.000 -0.552  0.000
sfmF:dose2               0.000  0.000 -0.246   0.000  -0.571  0.000 -0.246
sfF:8:dose2              0.000  0.000 -0.157   0.000  -0.191  0.000 -0.736
sfM:15:dose2             0.000  0.000  0.000  -0.157   0.000 -0.194  0.000
sfF:15:dose2             0.000  0.000 -0.157   0.000  -0.191  0.000 -0.157
sfM:26:dose2             0.000  0.000  0.000  -0.157   0.000 -0.194  0.000
sfF:26:dose2             0.000  0.000 -0.157   0.000  -0.191  0.000 -0.157
                         sM:()8 sF:()12 sM:()12 sF:()15 sM:()15 sF:()19 sM:()19
sexM:factor(tmonstdy)-4
sexF:factor(tmonstdy)1
sexM:factor(tmonstdy)1
sexF:factor(tmonstdy)5
sexM:factor(tmonstdy)5
sexF:factor(tmonstdy)8
sexM:factor(tmonstdy)8
sexF:factor(tmonstdy)12  0.000
sexM:factor(tmonstdy)12  0.494  0.000
sexF:factor(tmonstdy)15  0.000  0.292   0.000
sexM:factor(tmonstdy)15  0.297  0.000   0.297   0.000
sexF:factor(tmonstdy)19  0.000  0.510   0.000   0.292   0.000
sexM:factor(tmonstdy)19  0.494  0.000   0.494   0.000   0.297   0.000
sexF:factor(tmonstdy)22  0.000  0.510   0.000   0.292   0.000   0.510   0.000
sexM:factor(tmonstdy)22  0.494  0.000   0.494   0.000   0.297   0.000   0.494
sexF:factor(tmonstdy)26  0.000  0.292   0.000   0.241   0.000   0.292   0.000
```

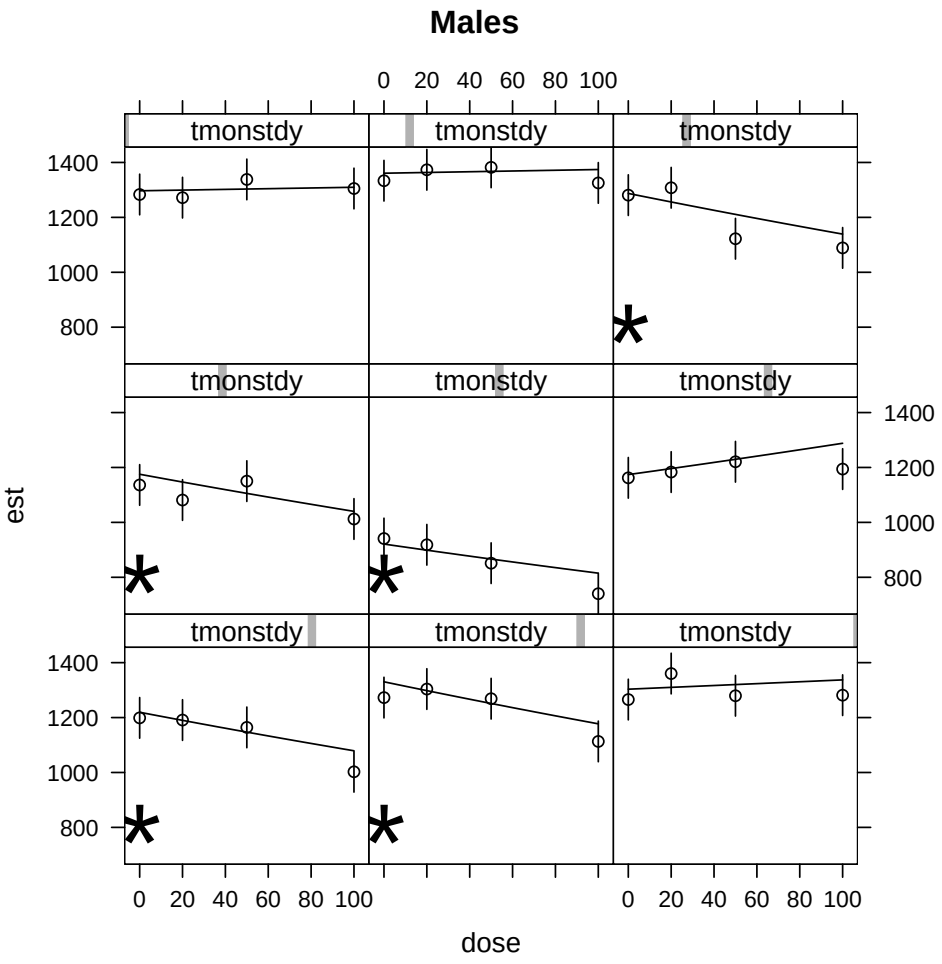
```
sexM:factor(tmonstdy)26  0.297  0.000   0.297   0.000   0.241   0.000   0.297
sfM:1:dose2             -0.194  0.000  -0.194   0.000  -0.157   0.000  -0.194
sfF:1:dose2              0.000 -0.191   0.000  -0.157   0.000  -0.191   0.000
sfmM:dose2              -0.552  0.000  -0.552   0.000  -0.258   0.000  -0.552
sfmF:dose2               0.000 -0.571   0.000  -0.246   0.000  -0.571   0.000
sfF:8:dose2              0.000 -0.191   0.000  -0.157   0.000  -0.191   0.000
sfM:15:dose2            -0.194  0.000  -0.194   0.000  -0.736   0.000  -0.194
sfF:15:dose2             0.000 -0.191   0.000  -0.736   0.000  -0.191   0.000
sfM:26:dose2            -0.194  0.000  -0.194   0.000  -0.157   0.000  -0.194
sfF:26:dose2             0.000 -0.191   0.000  -0.157   0.000  -0.191   0.000
                        sF:()22 sM:()22 sF:()26 sM:()26 sM:1:2 sF:1:2 sfmM:2
sexM:factor(tmonstdy)-4
sexF:factor(tmonstdy)1
sexM:factor(tmonstdy)1
sexF:factor(tmonstdy)5
sexM:factor(tmonstdy)5
sexF:factor(tmonstdy)8
sexM:factor(tmonstdy)8
sexF:factor(tmonstdy)12
sexM:factor(tmonstdy)12
sexF:factor(tmonstdy)15
sexM:factor(tmonstdy)15
sexF:factor(tmonstdy)19
sexM:factor(tmonstdy)19
sexF:factor(tmonstdy)22
sexM:factor(tmonstdy)22  0.000
sexF:factor(tmonstdy)26  0.292   0.000
sexM:factor(tmonstdy)26  0.000   0.297   0.000
sfM:1:dose2              0.000  -0.194   0.000  -0.157
sfF:1:dose2             -0.191   0.000  -0.157   0.000   0.000
sfmM:dose2               0.000  -0.552   0.000  -0.258   0.351  0.000
sfmF:dose2              -0.571   0.000  -0.246   0.000   0.000  0.334  0.000
sfF:8:dose2             -0.191   0.000  -0.157   0.000   0.000  0.214  0.000
sfM:15:dose2             0.000  -0.194   0.000  -0.157   0.214  0.000  0.351
sfF:15:dose2            -0.191   0.000  -0.157   0.000   0.000  0.214  0.000
sfM:26:dose2             0.000  -0.194   0.000  -0.736   0.214  0.000  0.351
sfF:26:dose2            -0.191   0.000  -0.736   0.000   0.000  0.214  0.000
                        sfmF:2 sF:8:2 sM:15: sF:15: sM:26:
sexM:factor(tmonstdy)-4
sexF:factor(tmonstdy)1
sexM:factor(tmonstdy)1
sexF:factor(tmonstdy)5
sexM:factor(tmonstdy)5
sexF:factor(tmonstdy)8
sexM:factor(tmonstdy)8
sexF:factor(tmonstdy)12
sexM:factor(tmonstdy)12
sexF:factor(tmonstdy)15
sexM:factor(tmonstdy)15
sexF:factor(tmonstdy)19
sexM:factor(tmonstdy)19
sexF:factor(tmonstdy)22
sexM:factor(tmonstdy)22
sexF:factor(tmonstdy)26
```

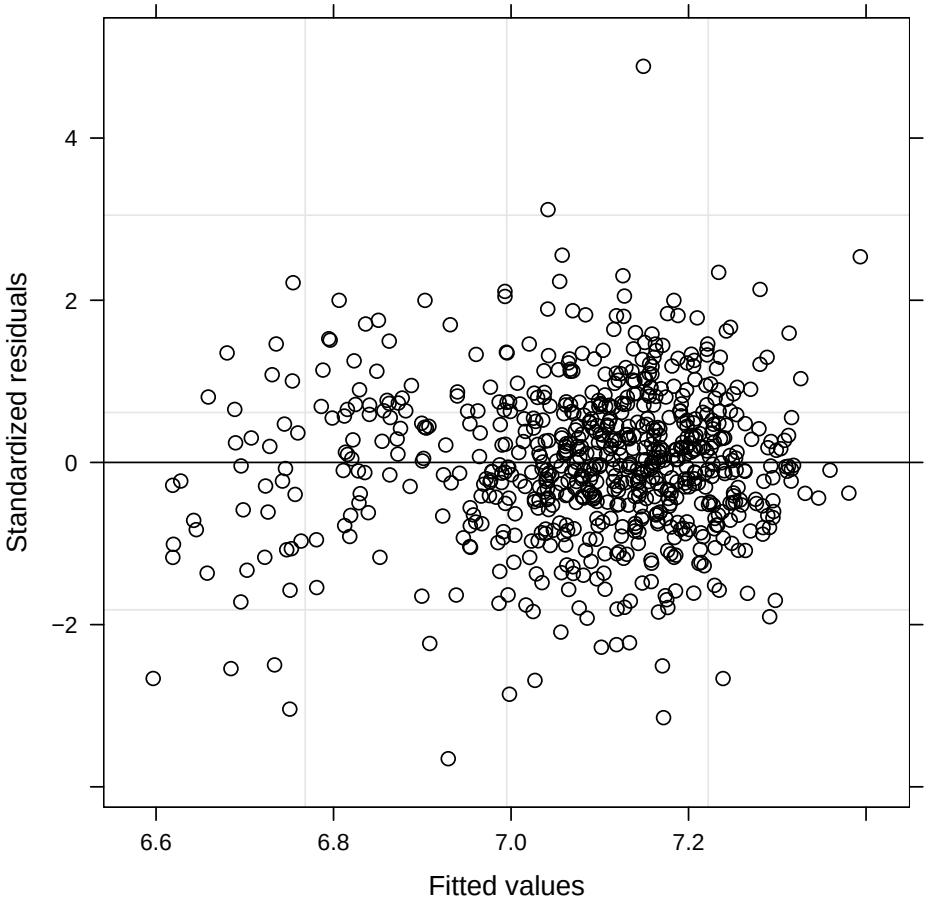
```
sexM:factor(tmonstdy)26
sfM:1:dose2
sfF:1:dose2
sfmM:dose2
sfmF:dose2
sfF:8:dose2              0.334
sfM:15:dose2             0.000  0.000
sfF:15:dose2             0.334  0.214  0.000
sfM:26:dose2             0.000  0.000  0.214  0.000
sfF:26:dose2             0.334  0.214  0.000  0.214  0.000

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.65286455 -0.52437395 -0.02485574  0.57701135  4.88477109

Number of Observations: 720
Number of Groups: 80
```

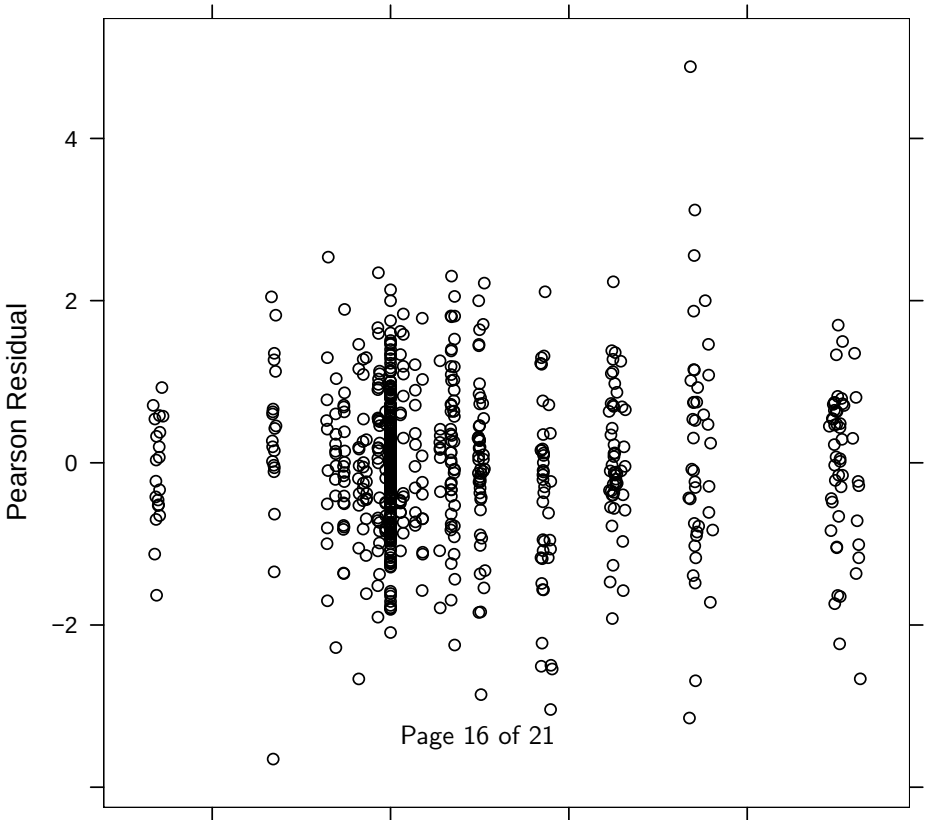
Diagnostic plots for this model. First, means and confidence intervals with fitted curve superimposed. Times that contribute to the calculation of the BMD for each sex are marked with an asterisk ("*").

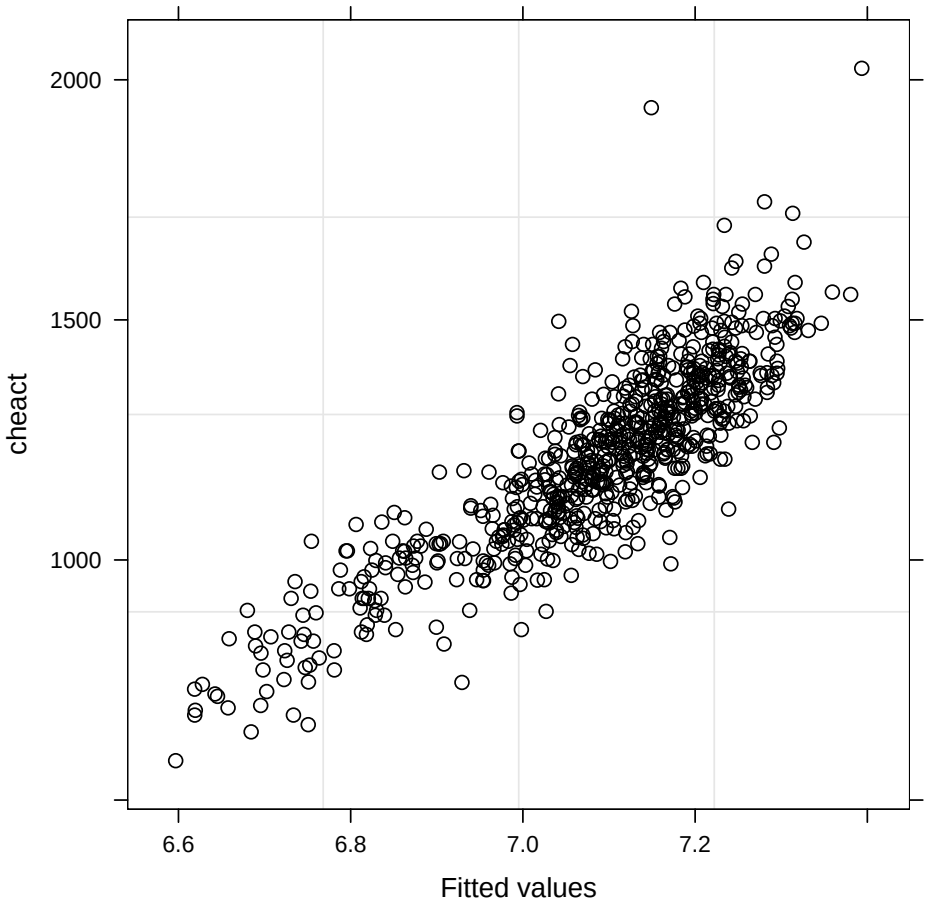
```
> ndta <- expand.grid(dose2 = seq(0, max(dta$dose2), length = 101),
+     sex = levels(dta$sex), tmonstdy = sort(unique(dta$tmonstdy)))
> ndta$ftmonstdy <- factor(ndta$tmonstdy)
> lvls <- levels(ndta$ftmonstdy)
> lvls[1] <- lvls[2]
> levels(ndta$ftmonstdy) <- lvls
> ndta$sf <- interaction(ndta$sex, ndta$ftmonstdy, drop = TRUE,
+     sep = ":")
> levels(ndta$sf)[c(3, 7, 11, 13)] <- "mF"
> levels(ndta$sf)[c(4, 6, 7, 10, 11)] <- "mM"
> ndta$pred <- predict(drmod2, newdata = ndta, level = 0)
```

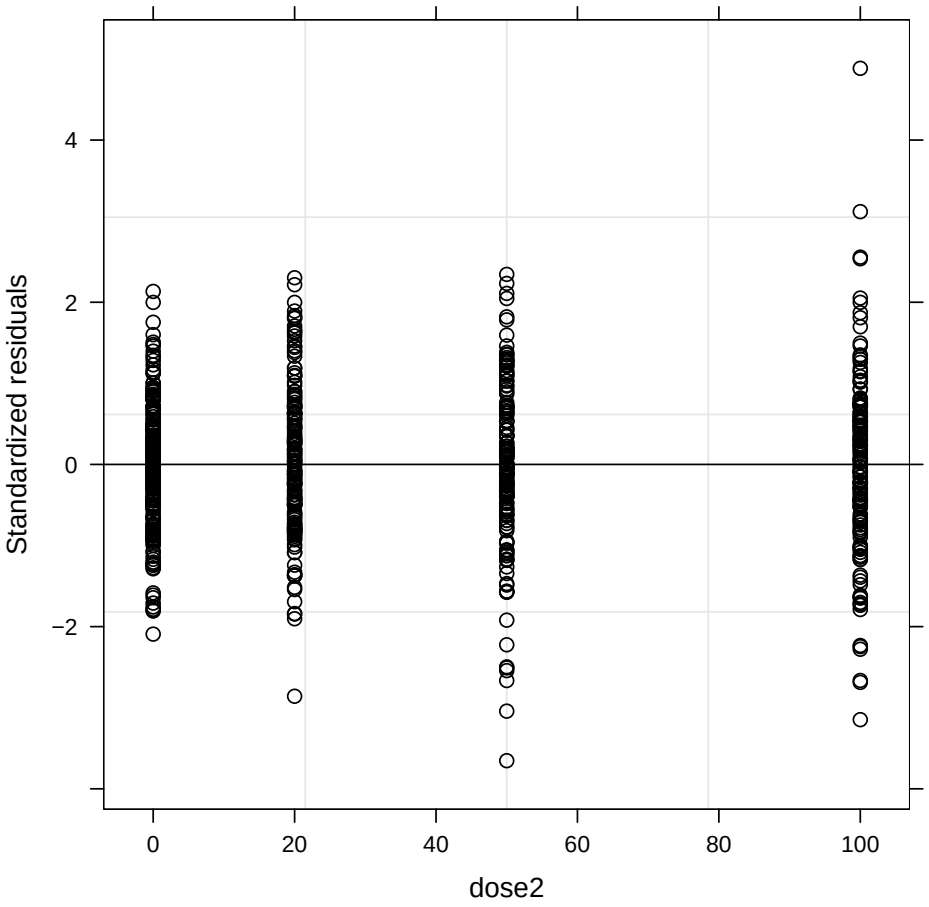

Females
tmonstdy
est
dose
0 20 40 60 80 100
800
1000
1200
1400
*

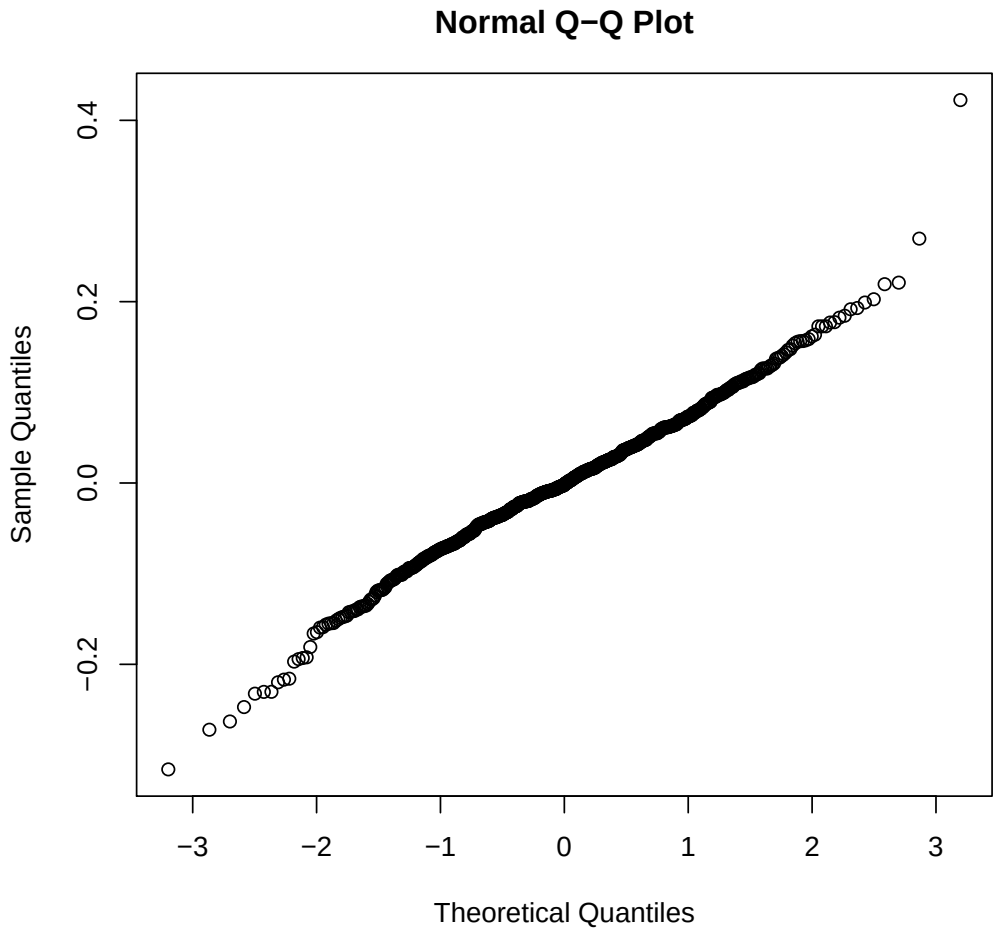

Males
tmonstdy
est
dose
0 20 40 60 80 100
800
1000
1200
1400
*


Standardized residuals
Fitted values
4
2
0
−2
6.6
6.8
7.0
7.2

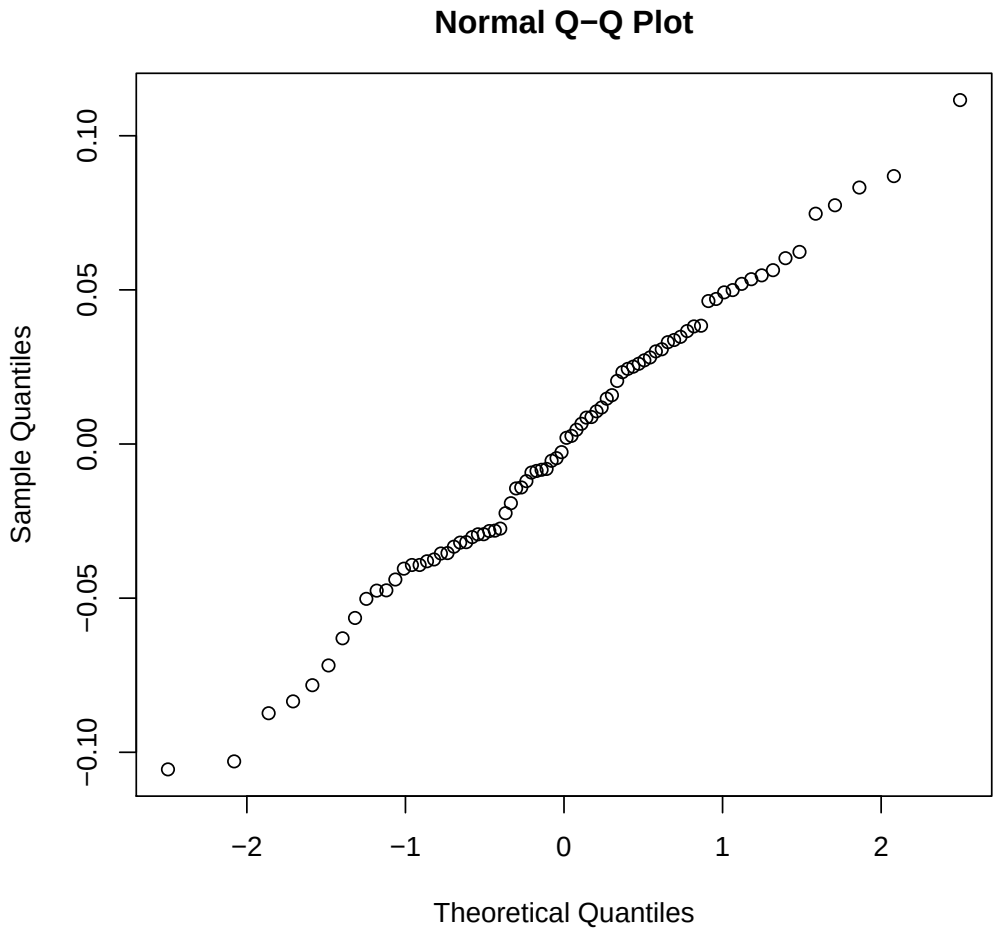

Pearson Residual
4
2
0
−2


2000
1500
1000
cheact
6.6
6.8
7.0
7.2
Fitted values


Standardized residuals
4
2
0
−2
0
20
40
60
80
100
dose2



QQ plot for residuals:



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. In these data, the BMD differs between sexes. BMD has units mg/kg, and times are in hours.

species RAT

mrid `[1] "45630601"`

Female `1D` **(se)** 4.45646776908981 (0.214600756546386)

Female BMD (95% CI) 86.1825541460728 (56.59145574268, 131.246537868068)

Female BMDL, the one-sided lower 95% CL 60.5506942907982

Male `1D` **(se)** 4.07816261138890 (0.139795145058822)

Male BMD (95% CI) 59.0368964135252 (44.8879400032506, 77.6456914237747)

Male BMDL, the one-sided lower 95% CL 46.9094927962633

Save the results:

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

Save the results for incorporating into a database:

```
> carbaryl.dermal.rbc <- list(mrid = levels(dta$mrid), species = "RAT",
+     BMD = list(female = list(lD = lD.F, lD.se = lD.F.se, BMD = BMD.F,
+         BMD.CI = BMD.F.CL, BMDL = BMDL.F), male = list(lD = lD.M,
+         lD.se = lD.M.se, BMD = BMD.M, BMD.CI = BMD.M.CL, BMDL = BMDL.M)))
> save(carbaryl.dermal.rbc, file = file.path("..", "..", "01Summaries",
+     "carbaryl.dermal.rbc.RData"))
```