MIfuns Sample Script

Covariate Plots

June 8, 2011

Tim Bergsma

1 Purpose

This script picks up after model.Rnw to process bootstrap results and make covariate plots.

1.1 Summarize bootstrap models.

Listing 1:

```
> #wait for bootstraps to finish
> getwd()

[1] "/Users/timb/project/metrum/inst/sample/script"
```

Listing 2:

```
> require(MIfuns)

MIfuns 4.3.2
```

Listing 3:

```
> boot <- read.csv('../nonmem/1005.boot/log.csv',as.is=TRUE)
> head(boot)

  X tool run parameter   moment            value
1 1  nm7   1       ofv  minimum 2292.12105209072
2 2  nm7   1    THETA1 estimate          7.69331
3 3  nm7   1    THETA1     prse             <NA>
4 4  nm7   1    THETA1       se             <NA>
5 5  nm7   1    THETA2 estimate          17.0917
6 6  nm7   1    THETA2     prse             <NA>
```

Listing 4:

```
> unique(boot$parameter)

 [1] "ofv"       "THETA1"    "THETA2"    "THETA3"    "THETA4"    "THETA5"
 [7] "THETA6"    "THETA7"    "OMEGA1.1"  "OMEGA2.1"  "OMEGA2.2"  "OMEGA3.1"
[13] "OMEGA3.2"  "OMEGA3.3"  "SIGMA1.1"  "cov"       "prob"      "min"
[19] "data"
```

Listing 5:

```
> text2decimal(unique(boot$parameter))

 [1]  NA 1.0 2.0 3.0 4.0 5.0 6.0 7.0 1.1 2.1 2.2 3.1 3.2 3.3 1.1  NA  NA  NA  NA
```

Listing 6:

```
> boot$X <- NULL
```

It looks like we have 14 estimated parameters. We will map them to the original control stream.

Listing 7:

```
> boot <- boot[!is.na(text2decimal(boot$parameter)),]
> head(boot)
```

```
  tool run parameter   moment    value
2  nm7   1    THETA1 estimate 7.69331
3  nm7   1    THETA1     prse     <NA>
4  nm7   1    THETA1       se     <NA>
5  nm7   1    THETA2 estimate 17.0917
6  nm7   1    THETA2     prse     <NA>
7  nm7   1    THETA2       se     <NA>
```

Listing 8:

```
> unique(boot$moment)
```

```
[1] "estimate" "prse"     "se"
```

Listing 9:

```
> unique(boot$value[boot$moment=='prse'])
```

```
[1] NA
```

prse, and therefore moment, is noninformative for these bootstraps.

Listing 10:

```
> boot <- boot[boot$moment=='estimate',]
> boot$moment <- NULL
> unique(boot$tool)
```

```
[1] "nm7"
```

Listing 11:

```
> boot$tool <- NULL
> head(boot)
```

```
   run parameter     value
2    1    THETA1   7.69331
5    1    THETA2   17.0917
8    1    THETA3 0.0644799
11   1    THETA4   3.30549
14   1    THETA5   120.258
17   1    THETA6   1.01506
```

Listing 12:

```
> unique(boot$value[boot$parameter %in% c('OMEGA2.1','OMEGA3.1','OMEGA3.2')])
```

```
[1] "0"
```

Listing 13:

```
> unique(boot$parameter[boot$value=='0'])
```

```
[1] "OMEGA2.1" "OMEGA3.1" "OMEGA3.2"
```

Off-diagonals (and only off-diagonals) are noninformative.

Listing 14:

```
> boot <- boot[!boot$value=='0',]
> any(is.na(as.numeric(boot$value)))
```

```
[1] FALSE
```

Listing 15:

```
> boot$value <- as.numeric(boot$value)
> head(boot)
```

```
   run parameter       value
2    1    THETA1   7.6933100
5    1    THETA2  17.0917000
8    1    THETA3   0.0644799
11   1    THETA4   3.3054900
14   1    THETA5 120.2580000
17   1    THETA6   1.0150600
```

1.2 Restrict data to 95 percentiles.

We did 300 runs. Min and max are strongly dependent on number of runs, since with an unbounded distribution, (almost) any value is possible with enough sampling. We clip to the 95 percentiles, to give distributions that are somewhat more scale independent.

Listing 16:

```
> boot <- inner(
+       boot,
+       preserve='run',
+       id.var='parameter',
+       measure.var='value'
+ )
> head(boot)
```

```
  run parameter       value
1   1    THETA1   7.6933100
2   1    THETA2          NA
3   1    THETA3   0.0644799
4   1    THETA4   3.3054900
5   1    THETA5 120.2580000
6   1    THETA6   1.0150600
```

Listing 17:

```
> any(is.na(boot$value))
```

```
[1] TRUE
```

Listing 18:

```
> boot <- boot[!is.na(boot$value),]
```

1.3 Recover parameter metadata from a specially-marked control stream.

We want meaningful names for our parameters. Harvest these from a reviewed control stream.

Listing 19:

```
> wiki <- wikitab(1005,'../nonmem')
> wiki
```

```
   parameter                                     description
1     THETA1                         apparent oral clearance
2     THETA2                  central volume of distribution
3     THETA3                        absorption rate constant
4     THETA4                    intercompartmental clearance
5     THETA5               peripheral volume of distribution
6     THETA6                        male effect on clearance
7     THETA7                      weight effect on clearance
8   OMEGA1.1          interindividual variability of clearance
9   OMEGA2.2 interindividual variability of central volume
10  OMEGA3.3               interindividual variability of K_a
11  SIGMA1.1                              proportional error
                                                           model tool  run
1  CL/F (L/h) ~ theta_1 *  theta_6 ^MALE * (WT/70)^theta_7  * e^eta_1  nm7 1005
2                          V_c /F (L) ~ theta_2 * (WT/70)^1 * e^eta_2  nm7 1005
3                                     K_a (h^-1 ) ~ theta_3 * e^eta_3  nm7 1005
4                                                Q/F (L/h) ~ theta_4  nm7 1005
5                                               V_p /F (L) ~ theta_5  nm7 1005
6                                               MALE_CL/F ~ theta_6  nm7 1005
7                                                 WT_CL/F ~ theta_7  nm7 1005
8                                             IIV_CL/F ~ Omega_1.1  nm7 1005
9                                           IIV_V_c /F ~ Omega_2.2  nm7 1005
10                                            IIV_K_a  ~ Omega_3.3  nm7 1005
11                                            err_prop ~ Sigma_1.1  nm7 1005
    estimate prse        se
1    8.57997 9.51  0.815572
2    21.6409 9.33   2.02017
3  0.0684281 8.04  0.005504
4    3.78411 13.5  0.510932
5    107.375 15.7   16.8257
6   0.998986 13.7    0.1364
```

```
7    1.67117 21.9  0.366424
8   0.195776 23.1 0.0451412
9   0.128574 30.4 0.0391464
10  0.106528 25.2 0.0268981
11  0.067111 11.4 0.0076591
```

Listing 20:

```
> wiki$name <- nospace(noUnits(lhs(wiki$model)))
> wiki$estimate <- as.numeric(wiki$estimate)
> unique(wiki$parameter)
```

```
 [1] "THETA1"   "THETA2"   "THETA3"   "THETA4"   "THETA5"   "THETA6"
 [7] "THETA7"   "OMEGA1.1" "OMEGA2.2" "OMEGA3.3" "SIGMA1.1"
```

Listing 21:

```
> unique(boot$parameter)
```

```
 [1] "THETA1"   "THETA3"   "THETA4"   "THETA5"   "THETA6"   "THETA7"
 [7] "OMEGA1.1" "OMEGA2.2" "OMEGA3.3" "SIGMA1.1" "THETA2"
```

Listing 22:

```
> boot <- stableMerge(boot, wiki[,c('parameter','name')])
> head(boot)
```

```
  run parameter       value       name
1   1    THETA1   7.6933100       CL/F
3   1    THETA3   0.0644799        K_a
4   1    THETA4   3.3054900        Q/F
5   1    THETA5 120.2580000      V_p/F
6   1    THETA6   1.0150600 MALE_CL/F
7   1    THETA7   1.4394100    WT_CL/F
```

1.4 Create covariate plot.

Now we make a covariate plot for clearance. We will normalize clearance by its median (we also could have used the model estimate). We need to take cuts of weight, since we can only really show categorically-constrained distributions. Male effect is already categorical. I.e, the reference individual has median clearance, is female, and has median weight.

1.4.1 Recover original covariates for guidance.

Listing 23:

```
> covariates <- read.csv('../data/derived/phase1.csv',na.strings='.')
> head(covariates)
```

```
     C ID TIME SEQ EVID  AMT    DV SUBJ HOUR TAFD  TAD LDOS MDV HEIGHT WEIGHT
1    C  1 0.00   0    0   NA 0.000    1 0.00 0.00   NA   NA   0    174   74.2
2 <NA>  1 0.00   1    1 1000    NA    1 0.00 0.00 0.00 1000   1    174   74.2
3 <NA>  1 0.25   0    0   NA 0.363    1 0.25 0.25 0.25 1000   0    174   74.2
4 <NA>  1 0.50   0    0   NA 0.914    1 0.50 0.50 0.50 1000   0    174   74.2
5 <NA>  1 1.00   0    0   NA 1.120    1 1.00 1.00 1.00 1000   0    174   74.2
6 <NA>  1 2.00   0    0   NA 2.280    1 2.00 2.00 2.00 1000   0    174   74.2
  SEX  AGE DOSE FED SMK DS CRCN predose zerodv
1   0 29.1 1000   1   0  0 83.5       1      1
2   0 29.1 1000   1   0  0 83.5       0      0
3   0 29.1 1000   1   0  0 83.5       0      0
4   0 29.1 1000   1   0  0 83.5       0      0
5   0 29.1 1000   1   0  0 83.5       0      0
6   0 29.1 1000   1   0  0 83.5       0      0
```

Listing 24:

```
> with(covariates,constant(WEIGHT,within=ID))
```

```
[1] TRUE
```

Listing 25:

```
> covariates <- unique(covariates[,c('ID','WEIGHT')])
> head(covariates)
```

```
   ID WEIGHT
1   1   74.2
16  2   80.3
31  3   94.2
46  4   85.2
61  5   82.8
76  6   63.9
```

Listing 26:

```
> covariates$WT <- as.numeric(covariates$WEIGHT)
> wt <- median(covariates$WT)
> wt
```

```
[1] 81
```

Listing 27:

```
> range(covariates$WT)
```

```
[1]  61 117
```

1.4.2 Reproduce the control stream submodel for selective cuts of a continuous covariate.

In the model we normalized by 70 kg, so that cut will have null effect. Let's try 65, 75, and 85 kg. We have to make a separate column for each cut, which is a bit of work. Basically, we make two more copies of our weight effect columns, and raise our normalized cuts to those powers, effectively reproducing the submodel from the control stream.

Listing 28:

```
> head(boot)

  run parameter       value      name
1   1     THETA1   7.6933100      CL/F
3   1     THETA3   0.0644799       K_a
4   1     THETA4   3.3054900       Q/F
5   1     THETA5 120.2580000     V_p/F
6   1     THETA6   1.0150600 MALE_CL/F
7   1     THETA7   1.4394100   WT_CL/F
```

Listing 29:

```
> unique(boot$name)

 [1] "CL/F"      "K_a"       "Q/F"       "V_p/F"     "MALE_CL/F" "WT_CL/F"
 [7] "IIV_CL/F"  "IIV_V_c/F" "IIV_K_a"   "err_prop"  "V_c/F"
```

Listing 30:

```
> clearance <- boot[boot$name %in% c('CL/F','WT_CL/F','MALE_CL/F'),]
> head(clearance)

   run parameter    value      name
1    1     THETA1 7.693310      CL/F
6    1     THETA6 1.015060 MALE_CL/F
7    1     THETA7 1.439410   WT_CL/F
12   2     THETA1 9.225660      CL/F
17   2     THETA6 0.951979 MALE_CL/F
18   2     THETA7 1.909720   WT_CL/F
```

Listing 31:

```
> frozen <- data.frame(cast(clearance,run ~ name),check.names=FALSE)
> head(frozen)

  run    CL/F MALE_CL/F WT_CL/F
1   1 7.69331  1.015060 1.43941
2   2 9.22566  0.951979 1.90972
3   3 9.07645  0.998521 1.74639
4   4 8.89737  0.842595 1.63451
5   5 8.06114  0.971182 1.29660
6   6 8.82893  0.984982 1.93506
```

Listing 32:

```
> frozen$`WT_CL/F:65` <- (65/70)**frozen$`WT_CL/F`
> frozen$`WT_CL/F:75` <- (75/70)**frozen$`WT_CL/F`
> frozen$`WT_CL/F:85` <- (85/70)**frozen$`WT_CL/F`
```

1.4.3 Normalize key parameter

Listing 33:

```
> #cl <- median(boot$value[boot$name=='CL/F'])
> cl <- with(wiki, estimate[name=='CL/F'])
> cl
```

```
[1] 8.57997
```

Listing 34:

```
> head(frozen)
```

```
  run    CL/F MALE_CL/F WT_CL/F WT_CL/F:65 WT_CL/F:75 WT_CL/F:85
1   1 7.69331  1.015060 1.43941  0.8988207   1.104408   1.322429
2   2 9.22566  0.951979 1.90972  0.8680331   1.140831   1.448870
3   3 9.07645  0.998521 1.74639  0.8786036   1.128048   1.403645
4   4 8.89737  0.842595 1.63451  0.8859186   1.119374   1.373483
5   5 8.06114  0.971182 1.29660  0.9083837   1.093579   1.286265
6   6 8.82893  0.984982 1.93506  0.8664045   1.142827   1.456015
```

Listing 35:

```
> frozen[['CL/F']] <- frozen[['CL/F']]/cl
> head(frozen)
```

```
  run      CL/F MALE_CL/F WT_CL/F WT_CL/F:65 WT_CL/F:75 WT_CL/F:85
1   1 0.8966593  1.015060 1.43941  0.8988207   1.104408   1.322429
2   2 1.0752555  0.951979 1.90972  0.8680331   1.140831   1.448870
3   3 1.0578650  0.998521 1.74639  0.8786036   1.128048   1.403645
4   4 1.0369931  0.842595 1.63451  0.8859186   1.119374   1.373483
5   5 0.9395301  0.971182 1.29660  0.9083837   1.093579   1.286265
6   6 1.0290164  0.984982 1.93506  0.8664045   1.142827   1.456015
```

Listing 36:

```
> frozen$`WT_CL/F` <- NULL
> molten <- melt(frozen,id.var='run',na.rm=TRUE)
> head(molten)
```

```
  run variable     value
1   1     CL/F 0.8966593
2   2     CL/F 1.0752555
3   3     CL/F 1.0578650
4   4     CL/F 1.0369931
5   5     CL/F 0.9395301
6   6     CL/F 1.0290164
```

1.4.4 Plot.

Now we plot. We reverse the variable factor to give us top-down layout of strips.

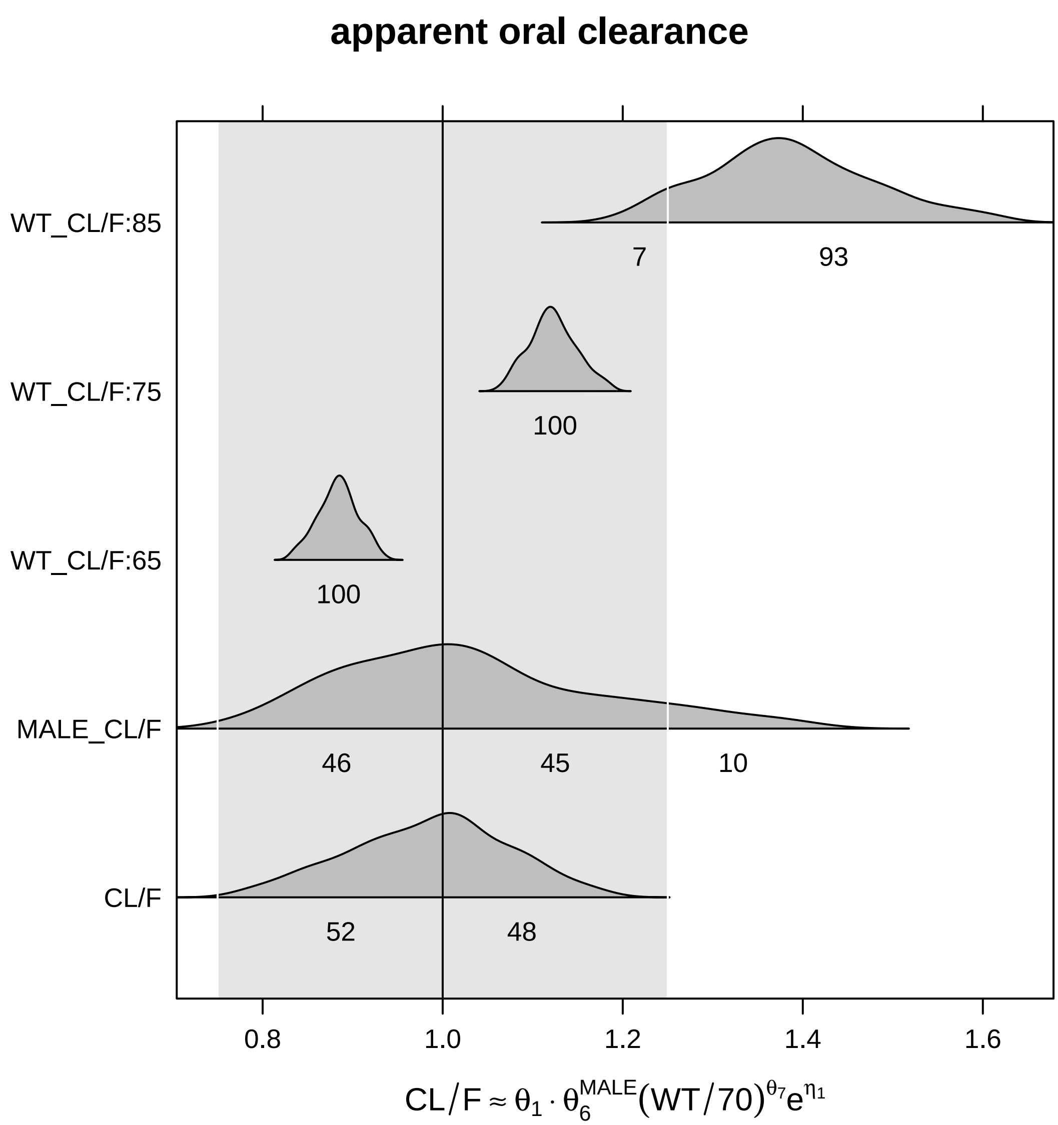
Listing 37:

```
> levels(molten$variable)

[1] "CL/F"        "MALE_CL/F"   "WT_CL/F:65" "WT_CL/F:75" "WT_CL/F:85"
```

Listing 38:

```
> molten$variable <- factor(molten$variable,levels=rev(levels(molten$variable)))
> print(
+   stripplot(
+     variable ~ value,
+     data=molten,
+     panel=panel.covplot,
+     xlab=parse(text=with(wiki,wiki2plotmath(noUnits(model[name=='CL/F'])))),
+     main=with(wiki,description[name=='CL/F'])
+   )
+ )
```



1.4.5 Summarize

We see that clearance is estimated with good precision. Ignoring outliers, there is not much effect on clearance of being male, relative to female. Increasing weight is associated with increasing clearance. There is a 93 percent probability that an 85 kg person will have at least 25 percent greater clearance than a 70 kg person.