Covariate Plots

September 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] "/home/timb/metrumrg/inst/sample/script"
```

Listing 2:

```
> require(metrumrg)
```

```
metrumrg 5.0
```

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 2522.77057359274
2 2  nm7   1    THETA1 estimate           8.0323
3 3  nm7   1    THETA1     prse             <NA>
4 4  nm7   1    THETA1       se             <NA>
5 5  nm7   1    THETA2 estimate          23.1126
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  8.0323
3  nm7   1    THETA1     prse    <NA>
4  nm7   1    THETA1       se    <NA>
5  nm7   1    THETA2 estimate 23.1126
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   8.0323
5    1    THETA2  23.1126
8    1    THETA3 0.074152
11   1    THETA4  4.00584
14   1    THETA5   106.33
17   1    THETA6  1.11708
```

Listing 12:

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

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

Listing 13:

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

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

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

Listing 14:

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

```
[1] TRUE
```

Listing 15:

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

```
   run parameter      value
2    1    THETA1   8.032300
5    1    THETA2  23.112600
8    1    THETA3   0.074152
11   1    THETA4   4.005840
14   1    THETA5 106.330000
17   1    THETA6   1.117080
```

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   8.032300
2   1    THETA2  23.112600
3   1    THETA3   0.074152
4   1    THETA4   4.005840
5   1    THETA5 106.330000
6   1    THETA6   1.117080
```

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 Ka
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	9.53	0.817948
2	21.6409	9.34	2.02094
3	0.0684281	8.04	0.00550178
4	3.78411	13.5	0.511271
5	107.376	15.7	16.8344
6	0.998986	14.8	0.148279
7	1.67117	21.7	0.363297
8	0.195776	23	0.0450967
9	0.128574	30.4	0.0391104

```
10  0.106527 25.3  0.0268981
11  0.067111 11.4 0.00766169
```

Listing 20:

```
> wiki$name <- wiki2label(wiki$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"   "THETA2"   "THETA3"   "THETA4"   "THETA5"   "THETA6"
 [7] "THETA7"   "OMEGA1.1" "OMEGA2.2" "SIGMA1.1" "OMEGA3.3"
```

Listing 22:

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

```
  run parameter      value      name
1   1    THETA1   8.032300      CL/F
2   1    THETA2  23.112600     V_c/F
3   1    THETA3   0.074152       K_a
4   1    THETA4   4.005840       Q/F
5   1    THETA5 106.330000     V_p/F
6   1    THETA6   1.117080 MALE_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      0
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   8.032300      CL/F
2   1    THETA2  23.112600     V_c/F
3   1    THETA3   0.074152       K_a
4   1    THETA4   4.005840       Q/F
5   1    THETA5 106.330000     V_p/F
6   1    THETA6   1.117080 MALE_CL/F
```

Listing 29:

```
> unique(boot$name)
```

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

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 8.032300      CL/F
6    1    THETA6 1.117080 MALE_CL/F
7    1    THETA7 1.593720   WT_CL/F
12   2    THETA1 8.610640      CL/F
17   2    THETA6 0.988349 MALE_CL/F
18   2    THETA7 1.473480   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 8.03230  1.117080 1.59372
```

```
2    2 8.61064  0.988349 1.47348
3    3 8.57952  0.966976 1.66538
4    4 8.08375  1.035090 2.09828
5    5 9.84918  0.775953 1.70033
6    6 8.34069  0.943343 1.77161
```

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 8.03230  1.117080 1.59372  0.8886006   1.116228   1.362649
2   2 8.61064  0.988349 1.47348  0.8965541   1.107007   1.331206
3   3 8.57952  0.966976 1.66538  0.8838942   1.121761   1.381740
4   4 8.08375  1.035090 2.09828  0.8559877   1.155770   1.502896
5   5 9.84918  0.775953 1.70033  0.8816078   1.124469   1.391148
6   6 8.34069  0.943343 1.77161  0.8769630   1.130012   1.410535
```

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 0.9361688  1.117080 1.59372  0.8886006   1.116228   1.362649
2   2 1.0035746  0.988349 1.47348  0.8965541   1.107007   1.331206
3   3 0.9999476  0.966976 1.66538  0.8838942   1.121761   1.381740
4   4 0.9421653  1.035090 2.09828  0.8559877   1.155770   1.502896
5   5 1.1479271  0.775953 1.70033  0.8816078   1.124469   1.391148
6   6 0.9721118  0.943343 1.77161  0.8769630   1.130012   1.410535
```

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 0.9361688
2   2     CL/F 1.0035746
3   3     CL/F 0.9999476
4   4     CL/F 0.9421653
5   5     CL/F 1.1479271
6   6     CL/F 0.9721118
```

1.4.4 Plot.

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

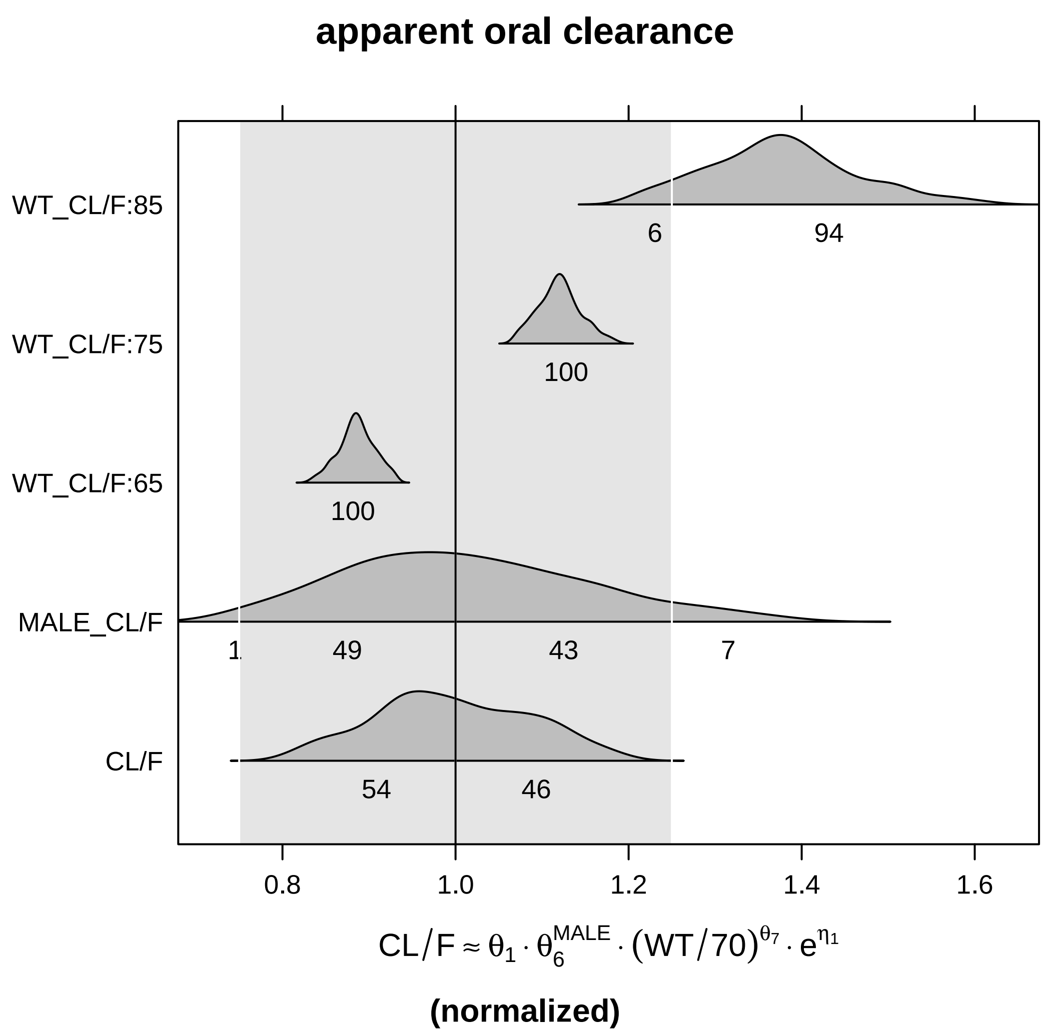
Listing 37:

```
> 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']),
+     sub=('(normalized)\n\n\n')
+   )
+ )
```


apparent oral clearance
WT_CL/F:85
6
94
WT_CL/F:75
100
WT_CL/F:65
100
MALE_CL/F
1
49
43
7
CL/F
54
46
0.8
1.0
1.2
1.4
1.6
$CL/F \approx \theta_1 \cdot \theta_6^{MALE} \cdot (WT/70)^{\theta_7} \cdot e^{\eta_1}$
(normalized)

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.