Covariate Plots

September 27, 2012

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] "/data/metrumrg/inst/example/project/script"
```

Listing 2:

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

```
  X tool run parameter   moment            value
1 1  nm7   1       ofv  minimum 2459.17577365853
2 2  nm7   1    THETA1 estimate          9.90635
3 3  nm7   1    THETA1     prse             <NA>
4 4  nm7   1    THETA1       se             <NA>
5 5  nm7   1    THETA2 estimate          21.8851
6 6  nm7   1    THETA2     prse             <NA>
```

Listing 3:

```
> 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" "SIGMA2.1" "SIGMA2.2" "cov"
[19] "prob"     "min"      "data"
```

Listing 4:

```
> 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 2.1 2.2  NA  NA
[20]  NA  NA
```

Listing 5:

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

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

Listing 6:

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

```
  tool run parameter   moment   value
2  nm7   1    THETA1 estimate 9.90635
3  nm7   1    THETA1     prse    <NA>
4  nm7   1    THETA1       se    <NA>
5  nm7   1    THETA2 estimate 21.8851
6  nm7   1    THETA2     prse    <NA>
7  nm7   1    THETA2       se    <NA>
```

Listing 7:

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

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

Listing 8:

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

```
[1] NA
```

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

Listing 9:

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

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

Listing 10:

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

```
   run parameter     value
2    1    THETA1   9.90635
5    1    THETA2   21.8851
8    1    THETA3 0.0708169
11   1    THETA4   3.36905
14   1    THETA5   94.6453
17   1    THETA6  0.972457
```

Listing 11:

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

```
 [1] "0.118667"    "0.00244158"  "-0.0290774"  "0.126793"    "0.00496435"
 [6] "-0.034875"   "0.0793793"   "0.012632"    "-0.0254671"  "0.0930842"
[11] "-0.0080005"  "-0.0604624"  "0.0776865"   "-0.0332067"  "-0.0431813"
```

```
 [16] "0.103263"     "-0.00113551"  "-0.0399993"   "0.124331"     "-0.00239167"
 [21] "-0.0292839"   "0.092983"     "0.00605732"   "-0.0318721"   "0.127233"
 [26] "0.0107016"    "-0.0244607"   "0.112805"     "0.0269113"    "-0.00833729"
 [31] "0.089758"     "0.00378936"   "-0.0420002"   "0.145258"     "-0.0511887"
 [36] "-0.0348086"   "0.123499"     "0.0100471"    "-0.0206122"   "0.0876026"
 [41] "-0.0100166"   "-0.0246651"   "0.085265"     "-0.00160533"  "-0.0344936"
 [46] "0.129989"     "0.0285799"    "-0.0412457"   "0.0885718"    "-0.00652974"
 [51] "-0.0477025"   "0.12808"      "-0.043111"    "-0.0414077"   "0.0643076"
 [56] "-0.0278958"   "-0.0369337"   "0.190188"     "-0.0205091"   "-0.0254164"
 [61] "0.118583"     "-0.00752829"  "-0.0254237"   "0.098401"     "-0.0268501"
 [66] "-0.0508131"   "0.128193"     "0.0232698"    "-0.0236511"   "0.167149"
 [71] "-0.0217309"   "-0.0381066"   "0.165549"     "0.00262881"   "-0.0201107"
 [76] "0.0947926"    "-0.0169354"   "-0.0396989"   "0.0463036"    "-0.00588823"
 [81] "-0.0567601"   "0.194389"     "-0.0168312"   "-0.0245029"   "0.104523"
 [86] "0.00451309"   "-0.0224615"   "0.106569"     "-0.0108574"   "-0.025071"
 [91] "0.108902"     "-0.0111868"   "-0.0269203"   "0.0997164"    "-0.0395324"
 [96] "-0.0396857"   "0.0850939"    "-0.0237444"   "-0.040846"    "0.118172"
[101] "-0.035136"    "-0.0617901"   "0.112738"     "-0.0257007"   "-0.0452832"
[106] "0.238888"     "0.0421279"    "-0.0113241"   "0.142488"     "-0.0102775"
[111] "-0.0246247"   "0.177368"     "0.0528058"    "0.00956807"   "0.106916"
[116] "0.00847416"   "-0.0370728"   "0.0610856"    "-0.0328229"   "-0.0478418"
[121] "0.144276"     "0.00445123"   "-0.0430469"   "0.132412"     "-0.00550047"
[126] "-0.0287152"   "0.0982564"    "-0.000317119" "-0.00174368"  "0.171034"
[131] "0.0245745"    "-0.0006445"   "0.0966462"    "-0.0427955"   "-0.0422816"
[136] "0.104502"     "-0.00684753"  "-0.02414"     "0.0483272"    "-0.0160977"
[141] "-0.0432605"   "0.10326"      "0.00876874"   "-0.0425961"   "0.0835958"
[146] "-0.000351788" "-0.0448004"   "0.112751"     "0.00295718"   "-0.0384557"
[151] "0.179579"     "0.0253308"    "-0.0173371"   "0.0567283"    "0.00398104"
[156] "-0.0299791"   "0.18087"      "-0.00186198"  "-0.0249436"   "0.117224"
[161] "0.0146449"    "-0.026458"    "0.0867064"    "-0.0341617"   "-0.0468784"
[166] "0.161089"     "0.0163142"    "0.00365988"   "0.110382"     "-0.0199146"
[171] "-0.0610093"   "0.093385"     "0.00430594"   "-0.0585277"   "0.131611"
[176] "-0.0273368"   "-0.0414609"   "0.0740755"    "-0.0393714"   "-0.0532818"
[181] "0.114813"     "0.00049734"   "-0.0327192"   "0.166116"     "0.0260548"
```

```
[186] "-0.013542"    "0.202128"     "0.0177431"    "-0.0210056"   "0.0910116"
[191] "0.0151579"    "-0.0408482"   "0.0869701"    "0.0132548"    "-0.0369328"
[196] "0.121663"     "-0.0174076"   "-0.0312722"   "0.117321"     "-0.00248588"
[201] "-0.0311927"   "0.0696887"    "-0.0238136"   "-0.0435333"   "0.157211"
[206] "0.0276358"    "-0.0167381"   "0.103759"     "-0.0320846"   "-0.0491522"
[211] "0.127103"     "0.00962958"   "-0.0315438"   "0.109678"     "-0.00299232"
[216] "-0.0269879"   "0.163873"     "-0.0222193"   "-0.0279459"   "0.149759"
[221] "-0.0606377"   "-0.0582299"   "0.156683"     "-0.00684372"  "-0.0128815"
[226] "0.132928"     "0.0117905"    "-0.0325876"   "0.0667082"    "-0.0396458"
[231] "-0.0444948"   "0.164516"     "0.00957041"   "-0.0156377"   "0.0973463"
[236] "-0.00795969"  "-0.0377035"   "0.114301"     "-0.00647026"  "-0.0362529"
[241] "0.13034"      "-0.0293763"   "-0.0610235"   "0.146626"     "-0.000397635"
[246] "-0.0189132"   "0.137892"     "0.000286094"  "-0.0289521"   "0.0894692"
[251] "-0.0458925"   "-0.0433733"   "0.146661"     "0.0142578"    "-0.00460059"
[256] "0.128869"     "0.00757751"   "-0.0270152"   "0.173982"     "0.0191435"
[261] "-0.0231225"   "0.105146"     "-0.0288077"   "-0.0461907"   "0.174007"
[266] "-0.0250103"   "-0.0154689"   "0.157435"     "-0.0242077"   "-0.0433651"
[271] "0.112829"     "-0.0196418"   "-0.0358258"   "0.110401"     "-0.0343584"
[276] "-0.062192"    "0.119419"     "0.000857924"  "-0.0184118"   "0.0933218"
[281] "-0.0145935"   "-0.0412315"   "0.116967"     "-0.0102656"   "-0.0421786"
[286] "0.121022"     "-0.0340938"   "-0.0461654"   "0.204826"     "0.00484302"
[291] "-0.0163292"   "0.102246"     "-0.0446805"   "-0.0417689"   "0.100399"
[296] "-0.018729"    "-0.0527321"   "0.105438"     "-0.0330354"   "-0.0412064"
[301] "0.133183"     "-0.0168392"   "-0.0265834"   "0.0945209"    "-0.0238335"
[306] "-0.0466818"   "0.115866"     "-0.017409"    "-0.0383749"   "0.151987"
[311] "-0.00223927"  "-0.0378265"   "0.111782"     "-0.0362034"   "-0.0341998"
[316] "0.115669"     "-0.0487444"   "-0.0605145"   "0.0491949"    "-0.0400172"
[321] "-0.0576977"   "0.092407"     "-0.00301386"  "-0.0217215"   "0.120699"
[326] "-0.0180292"   "-0.0419035"   "0.0841477"    "-0.0272911"   "-0.0373338"
[331] "0.139443"     "-0.0562124"   "-0.0628544"   "0.133833"     "-0.00586636"
[336] "-0.046539"    "0.117263"     "0.00586304"   "-0.0212893"   "0.141696"
[341] "-0.0128159"   "-0.0454871"   "0.0762937"    "-0.0419307"   "-0.0446012"
[346] "0.115726"     "-0.0270751"   "-0.0334352"   "0.13825"      "0.0159639"
[351] "-0.0182246"   "0.153665"     "-0.0133607"   "-0.0312656"   "0.129178"
```

```
[356] "-0.00427368" "-0.0375721"  "0.0784119"   "-0.0189957"   "-0.0278067"
[361] "0.0859116"   "-0.0112815"  "-0.0467892"  "0.152549"     "-0.0117118"
[366] "-0.0259305"  "0.146448"    "-0.00831176" "-0.0340455"   "0.117957"
[371] "-0.0228686"  "-0.0302882"  "0.0997908"   "-0.00566569"  "-0.0270352"
[376] "0.148161"    "-0.0358147"  "-0.0466065"  "0.154804"     "-0.0038829"
[381] "-0.034445"   "0.0821751"   "0.0179158"   "-0.0208937"   "0.15992"
[386] "-0.00843196" "-0.0361845"  "0.154324"    "-0.0204424"   "-0.031392"
[391] "0.0876009"   "0.0185985"   "-0.0384597"  "0.145694"     "-0.0513708"
[396] "-0.0353267"  "0.0961088"   "-0.0153156"  "-0.0325847"   "0.113948"
[401] "-0.034475"   "-0.0391479"  "0.120384"    "-0.0235307"   "-0.0403034"
[406] "0.146402"    "-0.00909605" "-0.0229485"  "0.0978063"    "-0.0228648"
[411] "-0.0477661"  "0.0527554"   "-0.0401577"  "-0.0404197"   "0.191297"
[416] "0.0233098"   "0.00229656"  "0.0966464"   "-0.0101051"   "-0.0304403"
[421] "0.102137"    "-0.0675559"  "-0.0322084"  "0.0669484"    "-0.00414386"
[426] "-0.0350418"  "0.117324"    "0.0193679"   "-0.0293469"   "0.0433609"
[431] "-0.0378898"  "-0.0554593"  "0.116679"    "-0.0318529"   "-0.0605824"
[436] "0.0694299"   "-0.0246664"  "-0.054557"   "0.0898861"    "-0.0190008"
[441] "-0.0526564"  "0.115292"    "-0.0447852"  "-0.0434396"   "0.121013"
[446] "-0.00118378" "-0.0408613"  "0.0741351"   "-0.0189214"   "-0.0253848"
[451] "0.10436"     "-0.00162131" "-0.0200163"  "0.157001"     "-0.00523825"
[456] "-0.0247979"  "0.351512"    "0.0448057"   "0.0022718"    "0.118066"
[461] "-0.0221414"  "-0.0276653"  "0.114707"    "-0.00405176"  "-0.0277663"
[466] "0.125921"    "-0.0129464"  "-0.0347535"  "0.0982235"    "0.0112565"
[471] "-0.0208746"  "0.0690502"   "-0.0578277"  "-0.0478438"   "0.116005"
[476] "-0.0531912"  "-0.0461021"  "0.18995"     "0.0234269"    "-0.00411539"
[481] "0.0874318"   "-0.0666249"  "-0.0453449"  "0.250372"     "0.00769703"
[486] "-0.0208782"  "0.167599"    "0.0451788"   "-0.000658281" "0.102172"
[491] "-0.0143271"  "-0.0314074"  "0.089999"    "-0.0435918"   "-0.0577448"
[496] "0.0724952"   "-0.0250449"  "-0.0245537"  "0.105756"     "-0.0395233"
[501] "-0.031799"   "0.113584"    "0.019943"    "-0.0149399"   "0.074473"
[506] "-0.0676744"  "-0.0450858"  "0.0890917"   "-0.0412407"   "-0.0493289"
[511] "0.114196"    "-0.0385613"  "-0.0429864"  "0.0887943"    "-0.0233526"
[516] "-0.052806"   "0.0437427"   "-0.0220758"  "-0.03631"     "0.108754"
[521] "-0.00844638" "-0.0437149"  "0.0888674"   "-0.0272032"   "-0.0455592"
```

```
[526] "0.109084"     "0.0282791"    "-0.0144868"   "0.129467"     "-0.00760179"
[531] "-0.0198443"   "0.124011"     "0.0141874"    "-0.0382784"   "0.058798"
[536] "-0.0244619"   "-0.0366656"   "0.151256"     "-0.00470712"  "-0.0293704"
[541] "0.174942"     "-0.00865067"  "-0.0339619"   "0.156372"     "-0.0134531"
[546] "-0.0319259"   "0.146135"     "-0.0145825"   "-0.0205733"   "0.146592"
[551] "-0.0146933"   "-0.0412467"   "0.164537"     "-0.0107485"   "-0.0207015"
[556] "0.0803491"    "-0.0214791"   "-0.0432422"   "0.112331"     "-0.0225237"
[561] "-0.0453018"   "0.182547"     "-0.0240038"   "-0.0307109"   "0.14805"
[566] "-0.00531694"  "-0.0421742"   "0.104717"     "0.00909885"   "-0.0103995"
[571] "0.141556"     "-0.0117496"   "-0.0268157"   "0.0559191"    "-0.0145175"
[576] "-0.0399342"   "0.285597"     "0.0645653"    "0.008981"     "0.226193"
[581] "0.046553"     "-0.0167025"   "0.0863958"    "-0.0242859"   "-0.0445802"
[586] "0.106771"     "0.00712704"   "-0.0384483"   "0.128781"     "0.00935036"
[591] "-0.0255267"   "0.151834"     "0.0441309"    "0.00135338"   "0.112869"
[596] "-0.00344943"  "-0.0223531"   "0.0481188"    "-0.0179503"   "-0.0554499"
[601] "0.0818385"    "-0.025363"    "-0.0342881"   "0.0964024"    "-0.00882523"
[606] "-0.0304101"   "0.139398"     "-0.0187498"   "-0.0402843"   "0.155415"
[611] "-0.0104276"   "-0.0216436"   "0.0635677"    "-0.00394499"  "-0.0362467"
[616] "0.134238"     "0.00363447"   "-0.00675624"  "0.0945888"    "-0.0698271"
[621] "-0.0602748"   "0.0923153"    "-0.0150987"   "-0.0350403"   "0.0816814"
[626] "-0.00441326"  "-0.0490816"   "0.128442"     "-0.0261726"   "-0.0399655"
[631] "0.109772"     "-0.026388"    "-0.0386704"   "0.0884206"    "0.0352567"
[636] "-0.0224689"   "0.125035"     "-0.0162191"   "-0.018687"    "0.0837287"
[641] "0.00446699"   "-0.0381742"   "0.113758"     "0.0275214"    "-0.00948809"
[646] "0.0651329"    "-0.028727"    "-0.059332"    "0.129258"     "-0.0386893"
[651] "-0.0235989"   "0.141795"     "0.00185141"   "-0.0213213"   "0.11366"
[656] "-0.0188672"   "-0.0347942"   "0.0657828"    "-0.0261557"   "-0.0511753"
[661] "0.119634"     "-0.0109633"   "-0.0345759"   "0.107459"     "-0.0279097"
[666] "-0.0412287"   "0.128838"     "-0.00841188"  "-0.0275271"   "0.0641964"
[671] "-0.0448836"   "-0.0548619"   "0.105469"     "-0.00757213"  "-0.0405844"
[676] "0.171145"     "0.00200576"   "-0.0121884"   "0.0862844"    "-0.0229536"
[681] "-0.0273754"   "0.183262"     "0.00836226"   "-0.0156567"   "0.0791053"
[686] "-0.0363684"   "-0.0454922"   "0.233884"     "0.00371482"   "-0.0186512"
[691] "0.142954"     "-0.00156277"  "-0.0336849"   "0.0595535"    "-0.0238319"
```

```
[696] "-0.040877"    "0.0778213"   "-0.0396723"  "-0.0301191"  "0.0918914"
[701] "-0.0157761"   "-0.0291842"  "0.112114"    "0.0144103"   "-0.0306037"
[706] "0.138055"     "-0.0309847"  "-0.0432093"  "0.138142"    "0.00913342"
[711] "-0.0332078"   "0.138769"    "-0.0134422"  "-0.0507458"  "0.124471"
[716] "-0.0479363"   "-0.0479304"  "0.171498"    "-0.012174"   "-0.024206"
[721] "0.0539806"    "-0.0110509"  "-0.0497888"  "0.0957331"   "-0.0272111"
[726] "-0.0377325"   "0.105208"    "-0.0423641"  "-0.030917"   "0.0727409"
[731] "-0.00617633"  "-0.0425025"  "0.140177"    "-0.0588427"  "-0.0585425"
[736] "0.117713"     "-0.0278959"  "-0.0488723"  "0.141551"    "0.0282836"
[741] "-0.00360554"  "0.150674"    "0.00337359"  "-0.0222434"  "0.141123"
[746] "-0.0345793"   "-0.0358528"  "0.126258"    "0.00664547"  "-0.031716"
[751] "0.12749"      "-0.0124222"  "-0.0283894"  "0.131373"    "-0.0134413"
[756] "-0.0361807"   "0.14828"     "-0.0190469"  "-0.0179633"  "0.121132"
[761] "-0.0326461"   "-0.0519745"  "0.115293"    "-0.0400476"  "-0.0586148"
[766] "0.153755"     "-0.00780256" "-0.0310576"  "0.0721372"   "-0.0137677"
[771] "-0.0349868"   "0.106619"    "0.00161006"  "-0.045939"   "0.138163"
[776] "-0.0181831"   "-0.0264278"  "0.0938936"   "-0.0191982"  "-0.0385006"
[781] "0.146534"     "-0.0017786"  "-0.0262203"  "0.0941724"   "0.0024702"
[786] "-0.0389428"   "0.15367"     "0.0248957"   "0.00317056"  "0.134979"
[791] "-0.0159844"   "-0.0366245"  "0.150754"    "-0.0121233"  "-0.0210397"
[796] "0.10095"      "-0.0100419"  "-0.0380713"  "0.0781764"   "-0.0131158"
[801] "-0.0260255"   "0.183737"    "0.0471527"   "-0.00331539" "0.1228"
[806] "0.0128748"    "-0.0222088"  "0.0961972"   "0.00880841"  "-0.0339741"
[811] "0.0988057"    "0.0129825"   "-0.025063"   "0.106896"    "-0.0307457"
[816] "-0.0488858"   "0.199373"    "-0.00271342" "-0.0350051"  "0.103328"
[821] "0.0245543"    "-0.00192463" "0.106199"    "0.00493493"  "-0.0361124"
[826] "0.0844659"    "0.00496473"  "-0.0254264"  "0.0585897"   "-0.00588847"
[831] "-0.0442538"   "0.0701973"   "-0.00733007" "-0.0466282"  "0.0715264"
[836] "-0.0347379"   "-0.041555"   "0.0926782"   "-0.0344967"  "-0.0327237"
[841] "0.121334"     "-0.0321834"  "-0.0385032"  "0.0993592"   "0.000596698"
[846] "-0.0240696"   "0.149366"    "-0.0155156"  "-0.0419891"  "0.15887"
[851] "0.0105771"    "-0.00491576" "0.0673419"   "-0.0108881"  "-0.0470597"
[856] "0.127817"     "0.0066866"   "-0.0184135"  "0.148965"    "0.0134111"
[861] "-0.0248296"   "0.135588"    "0.0179441"   "-0.00795444" "0.0606021"
```

```
[866] "0.00193795"   "-0.0211145"   "0.0592927"    "-0.0327239"   "-0.0356361"
[871] "0.136619"     "-0.0223624"   "-0.0262953"   "0.10639"      "-0.0196667"
[876] "-0.0533358"   "0.0742856"    "-0.00833166"  "-0.0373464"   "0.0998201"
[881] "-0.00384369"  "-0.0251428"   "0.170603"     "-0.0143744"   "-0.0394314"
[886] "0.0868016"    "-0.028705"    "-0.0297085"   "0.100436"     "0.00791463"
[891] "-0.0297892"   "0.0597656"    "-0.039134"    "-0.0377067"   "0.112923"
[896] "0.00219277"   "-0.0172659"   "0.174119"     "0.0131603"    "-0.0141507"
```

Listing 12:

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

```
[1] "SIGMA2.1"
```

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

Listing 13:

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

```
[1] FALSE
```

Listing 14:

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

```
   run parameter      value
2    1    THETA1  9.9063500
5    1    THETA2 21.8851000
8    1    THETA3  0.0708169
11   1    THETA4  3.3690500
14   1    THETA5 94.6453000
17   1    THETA6  0.9724570
```

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 15:

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

```
  run parameter       value
1   1    THETA1  9.9063500
2   1    THETA2 21.8851000
3   1    THETA3  0.0708169
4   1    THETA4  3.3690500
5   1    THETA5 94.6453000
6   1    THETA6  0.9724570
```

Listing 16:

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

```
[1] TRUE
```

Listing 17:

```
> 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 18:

```
> 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   OMEGA2.1       interindividual clearance-volume covariance
10  OMEGA2.2 interindividual variability of central volume
11  OMEGA3.1           interindividual clearance-Ka covariance
12  OMEGA3.2              interindividual volume-Ka covariance
13  OMEGA3.3                 interindividual variability of Ka
14  SIGMA1.1                               proportional error
15  SIGMA2.2                                   additive 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                                             cov_CL,V ~ Omega_2.1  nm7 1005
```

```
10                                               IIV_V_c /F ~ Omega_2.2  nm7 1005
11                                               cov_CL,Ka  ~ Omega_3.1  nm7 1005
12                                                cov_V,Ka  ~ Omega_3.2  nm7 1005
13                                                 IIV_K_a  ~ Omega_3.3  nm7 1005
14                                                err_prop  ~ Sigma_1.1  nm7 1005
15                                                 err_add  ~ Sigma_2.2  nm7 1005
     estimate prse         se
1     9.50754 9.84   0.935942
2     22.7907 9.56    2.17864
3   0.0714314 7.35 0.00525212
4     3.47438 15.4   0.535659
5     113.269   21     23.793
6     1.02439 11.2   0.114304
7     1.19226 28.4   0.338587
8    0.213813 22.8  0.0488382
9    0.120739 26.4  0.0319111
10  0.0945275 33.2  0.0313504
11 -0.0116063  173  0.0200793
12 -0.0371985 36.1   0.013426
13  0.0465611 34.7  0.0161799
14  0.0491683 10.9 0.00538067
15   0.201814 33.5  0.0676412
```

Listing 19:

```
> 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" "OMEGA2.1" "OMEGA2.2" "OMEGA3.1" "OMEGA3.2"
[13] "OMEGA3.3" "SIGMA1.1" "SIGMA2.2"
```

Listing 20:

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

```
 [1] "THETA1"   "THETA2"   "THETA3"   "THETA4"   "THETA5"   "THETA6"
 [7] "THETA7"   "OMEGA1.1" "OMEGA2.1" "OMEGA2.2" "OMEGA3.1" "OMEGA3.2"
[13] "OMEGA3.3" "SIGMA1.1" "SIGMA2.2"
```

Listing 21:

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

```
  run parameter      value      name
1   1    THETA1  9.9063500      CL/F
2   1    THETA2 21.8851000     V_c/F
3   1    THETA3  0.0708169       K_a
4   1    THETA4  3.3690500       Q/F
5   1    THETA5 94.6453000     V_p/F
6   1    THETA6  0.9724570 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 22:

```
> 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 23:

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

```
[1] TRUE
```

Listing 24:

```
> 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 25:

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

```
[1] 81
```

Listing 26:

```
> 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 27:

```
> head(boot)
```

```
  run parameter      value      name
1   1    THETA1  9.9063500      CL/F
2   1    THETA2 21.8851000     V_c/F
3   1    THETA3  0.0708169       K_a
4   1    THETA4  3.3690500       Q/F
5   1    THETA5 94.6453000     V_p/F
6   1    THETA6  0.9724570 MALE_CL/F
```

Listing 28:

```
> 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"  "cov_CL,V"  "IIV_V_c/F" "cov_CL,Ka" "cov_V,Ka"
[13] "IIV_K_a"   "err_prop"  "err_add"
```

Listing 29:

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

```
   run parameter    value      name
1    1    THETA1 9.906350      CL/F
6    1    THETA6 0.972457 MALE_CL/F
7    1    THETA7 1.469340   WT_CL/F
16   2    THETA1 9.030580      CL/F
21   2    THETA6 1.038960 MALE_CL/F
22   2    THETA7 0.999503   WT_CL/F
```

Listing 30:

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

```
  run     CL/F MALE_CL/F  WT_CL/F
1   1  9.90635  0.972457 1.469340
2   2  9.03058  1.038960 0.999503
3   3  9.33141  0.846679 1.909750
4   4  9.25626  0.941014 1.697690
5   5 10.27090  1.252500 1.159260
6   6  9.41970  0.967219 1.484300
```

Listing 31:

```
> 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 32:

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

```
[1] 9.50754
```

Listing 33:

```
> 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  9.90635  0.972457 1.469340  0.8968292   1.106690   1.330136
2   2  9.03058  1.038960 0.999503  0.9286056   1.071392   1.214169
3   3  9.33141  0.846679 1.909750  0.8680311   1.140834   1.448878
4   4  9.25626  0.941014 1.697690  0.8817803   1.124264   1.390435
5   5 10.27090  1.252500 1.159260  0.9176764   1.083266   1.252420
6   6  9.41970  0.967219 1.484300  0.8958355   1.107833   1.334005
```

Listing 34:

```
> 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 1.0419467  0.972457 1.469340  0.8968292   1.106690   1.330136
2   2 0.9498335  1.038960 0.999503  0.9286056   1.071392   1.214169
3   3 0.9814747  0.846679 1.909750  0.8680311   1.140834   1.448878
4   4 0.9735705  0.941014 1.697690  0.8817803   1.124264   1.390435
5   5 1.0802900  1.252500 1.159260  0.9176764   1.083266   1.252420
6   6 0.9907610  0.967219 1.484300  0.8958355   1.107833   1.334005
```

Listing 35:

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

```
  run variable     value
1   1     CL/F 1.0419467
2   2     CL/F 0.9498335
3   3     CL/F 0.9814747
4   4     CL/F 0.9735705
5   5     CL/F 1.0802900
6   6     CL/F 0.9907610
```

1.4.4 Plot.

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

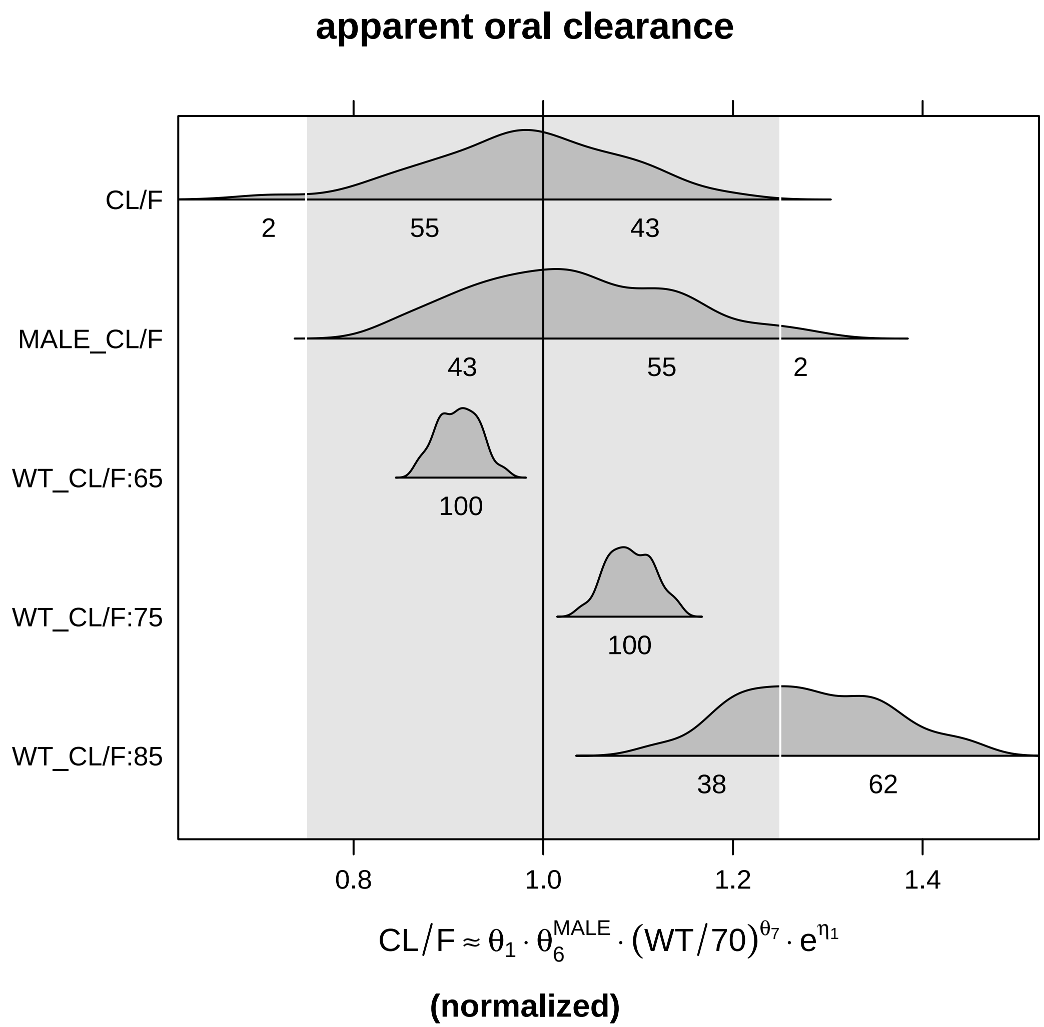
Listing 36:

```
> levels(molten$variable)
```

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

Listing 37:

```
> 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')
+   )
+ )
```



$$CL/F \approx \theta_1 \cdot \theta_6^{MALE} \cdot (WT/70)^{\theta_7} \cdot e^{\eta_1}$$

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.