Covariate Plots

April 3, 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/1005.bootlog.csv',as.is=TRUE)
> head(boot)
```

```
  X tool run parameter   moment            value
1 1  nm7   1       ofv  minimum 2353.21240698929
2 2  nm7   1    THETA1 estimate          8.64914
3 3  nm7   1    THETA1     prse             <NA>
4 4  nm7   1    THETA1       se             <NA>
5 5  nm7   1    THETA2 estimate          21.5594
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 8.64914
3  nm7   1    THETA1     prse    <NA>
4  nm7   1    THETA1       se    <NA>
5  nm7   1    THETA2 estimate 21.5594
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   8.64914
5    1    THETA2   21.5594
8    1    THETA3 0.0766718
11   1    THETA4   4.05759
14   1    THETA5   106.257
17   1    THETA6   1.07562
```

Listing 11:

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

```
  [1] "0.104011"    "0.00913698"  "-0.0372135"  "0.120086"    "-0.0471432"
  [6] "-0.0433544"  "0.238156"    "0.0317947"   "-0.0116794"  "0.0868132"
 [11] "-0.0301877"  "-0.0257547"  "0.172754"    "-0.00561372" "-0.0255986"
```

```
 [16] "0.141958"    "-0.00461459" "-0.0327725"  "0.145157"    "-0.0372318" 
 [21] "-0.052796"   "0.128854"    "0.0129519"   "-0.0247666"  "0.0909884"  
 [26] "0.0193092"   "-0.00895973" "0.147129"    "-0.0208962"  "-0.030639"  
 [31] "0.126885"    "-0.0132187"  "-0.039874"   "0.111968"    "-0.0584352" 
 [36] "-0.0565045"  "0.279561"    "0.0256141"   "-0.026671"   "0.0945945"  
 [41] "0.0161412"   "-0.0296876"  "0.110733"    "-0.00118301" "-0.0471514" 
 [46] "0.122448"    "0.0296459"   "-0.0283253"  "0.138172"    "-0.00870162"
 [51] "-0.0229121"  "0.21043"     "0.0241651"   "-0.0135542"  "0.0916718"  
 [56] "-0.0207034"  "-0.0464984"  "0.133865"    "-0.000638375" "-0.0216873"
 [61] "0.117777"    "-0.0105039"  "-0.0277175"  "0.113716"    "-0.0320301" 
 [66] "-0.0475784"  "0.13642"     "-0.0178311"  "-0.0544559"  "0.0838039"  
 [71] "-0.0369852"  "-0.0676137"  "0.137901"    "-0.0194737"  "-0.0517899" 
 [76] "0.152181"    "0.0128349"   "-0.00772528" "0.123007"    "-0.0108739" 
 [81] "-0.0475914"  "0.10136"     "-0.0233726"  "-0.0327004"  "0.13657"    
 [86] "-0.014055"   "-0.0375471"  "0.202238"    "0.0120881"   "-0.0164129" 
 [91] "0.103806"    "-0.00915871" "-0.0352087"  "0.139801"    "0.00780125" 
 [96] "-0.0185142"  "0.0701158"   "-0.0303831"  "-0.0583481"  "0.0972232"  
[101] "-0.00612742" "-0.0476269"  "0.0824277"   "0.0100558"   "-0.025531"  
[106] "0.135292"    "0.00602963"  "-0.0143407"  "0.0770001"   "0.0132279"  
[111] "-0.0382334"  "0.0983815"   "-0.022497"   "-0.0421142"  "0.120734"   
[116] "-0.0372276"  "-0.0476385"  "0.132248"    "-0.014428"   "-0.0553789" 
[121] "0.0911876"   "-0.0610659"  "-0.037353"   "0.0759784"   "-0.0316384" 
[126] "-0.0448702"  "0.100121"    "-0.027525"   "-0.0452165"  "0.0722031"  
[131] "-0.00160219" "-0.0498296"  "0.0811049"   "0.01685"     "-0.0284518" 
[136] "0.0585185"   "-0.0130713"  "-0.0516862"  "0.158004"    "-0.00211818"
[141] "-0.0381416"  "0.127999"    "-0.00914735" "-0.0561452"  "0.1329"     
[146] "0.0147834"   "-0.0250157"  "0.0951388"   "-0.0119525"  "-0.0233392" 
[151] "0.0871447"   "-0.0129608"  "-0.0370327"  "0.0961099"   "-0.00934003"
[156] "-0.0435359"  "0.110851"    "-0.0253761"  "-0.0360567"  "0.109055"   
[161] "0.0211067"   "-0.0234625"  "0.156331"    "-0.00326409" "-0.0273039" 
[166] "0.0966882"   "-0.0108823"  "-0.0535592"  "0.216562"    "-0.0282568" 
[171] "-0.0334737"  "0.0831576"   "-0.0132692"  "-0.0357714"  "0.0895132"  
[176] "0.011181"    "-0.0240417"  "0.144076"    "-0.00203105" "-0.0341183" 
[181] "0.173604"    "-0.0562174"  "-0.0367228"  "0.158231"    "-0.0106107" 
```

```
[186] "-0.0305045"  "0.122245"    "0.00298379"  "-0.0374551"  "0.0789944"
[191] "-0.00269587" "-0.0325763"  "0.0953883"   "0.00805023"  "-0.0378012"
[196] "0.108036"    "-0.00742945" "-0.037653"   "0.110399"    "-0.0283661"
[201] "-0.069566"   "0.108719"    "-0.0166837"  "-0.0282772"  "0.145469"
[206] "0.0197014"   "-0.0185515"  "0.138225"    "-0.00741411" "-0.0318173"
[211] "0.10142"     "-0.0445175"  "-0.0356729"  "0.125758"    "-0.0047929"
[216] "-0.0407196"  "0.0920434"   "-0.00850728" "-0.0385727"  "0.159856"
[221] "-0.0106127"  "-0.0189822"  "0.100593"    "0.028353"    "-0.0188271"
[226] "0.222256"    "-0.0373153"  "-0.0350459"  "0.112554"    "-0.00708125"
[231] "-0.0408384"  "0.116183"    "0.0188421"   "-0.0272687"  "0.0930931"
[236] "-0.0190976"  "-0.0265887"  "0.104984"    "-0.0395479"  "-0.0440541"
[241] "0.11286"     "-0.0149657"  "-0.0401269"  "0.0807202"   "0.0102482"
[246] "-0.0394861"  "0.0754772"   "-0.0326841"  "-0.0588911"  "0.117272"
[251] "-0.0268556"  "-0.0356572"  "0.144632"    "0.00805324"  "-0.0309206"
[256] "0.103054"    "-0.013012"   "-0.0306872"  "0.12777"     "-0.0196015"
[261] "-0.0387179"  "0.155761"    "-0.00476701" "-0.0287921"  "0.106573"
[266] "-0.018925"   "-0.0485841"  "0.15746"     "-0.010334"   "-0.0366549"
[271] "0.11261"     "-0.0254729"  "-0.0453578"  "0.112289"    "-0.00865634"
[276] "-0.0379589"  "0.119916"    "0.00816334"  "-0.0371269"  "0.124079"
[281] "-0.00896936" "-0.0376234"  "0.174149"    "0.00603724"  "-0.0133605"
[286] "0.0777343"   "-0.03327"    "-0.0433517"  "0.121953"    "0.00542045"
[291] "-0.0129371"  "0.101135"    "-0.0191816"  "-0.033636"   "0.305688"
[296] "0.0807639"   "0.00314576"  "0.154303"    "-0.0151509"  "-0.0175936"
[301] "0.248084"    "0.0131583"   "-0.021273"   "0.0967497"   "-0.00614986"
[306] "-0.00428826" "0.0727331"   "-0.0213507"  "-0.0444937"  "0.107167"
[311] "-0.041371"   "-0.0554941"  "0.0995659"   "0.0157866"   "-0.0265188"
[316] "0.102372"    "-0.0286341"  "-0.056387"   "0.160824"    "-0.0147961"
[321] "-0.0247954"  "0.0996811"   "-0.0408084"  "-0.0628177"  "0.144187"
[326] "-0.00670838" "-0.0329585"  "0.134171"    "-0.00127687" "-0.0277096"
[331] "0.299174"    "0.0344797"   "-0.00868071" "0.140386"    "0.0101187"
[336] "-0.012055"   "0.125188"    "-0.026049"   "-0.0388588"  "0.145022"
[341] "0.00327066"  "-0.0280941"  "0.0674323"   "-0.0511751"  "-0.0551048"
[346] "0.145631"    "0.0188765"   "-0.00392401" "0.154201"    "-0.0140666"
[351] "-0.0342735"  "0.166203"    "-0.00102814" "-0.0137689"  "0.142006"
```

```
[356] "0.00486449"   "-0.0115718"   "0.2267"       "-0.0395758"   "-0.0357008"
[361] "0.11422"      "0.00737285"   "-0.0415097"   "0.12635"      "0.0151095"
[366] "-0.0257725"   "0.0985544"    "-0.0113973"   "-0.0356029"   "0.073256"
[371] "-0.0507985"   "-0.0592862"   "0.158047"     "0.0198305"    "-0.00539255"
[376] "0.131727"     "-0.0100876"   "-0.0372"      "0.132284"     "-0.0292671"
[381] "-0.0280004"   "0.111457"     "0.00176474"   "-0.0306441"   "0.0783869"
[386] "-0.00800962"  "-0.0516039"   "0.0699839"    "0.00900772"   "-0.0341619"
[391] "0.0780957"    "-0.0348528"   "-0.0475667"   "0.181723"     "-0.0386095"
[396] "-0.0474309"   "0.132333"     "0.0132769"    "-0.0224961"   "0.0953981"
[401] "-0.000313327" "-0.0414034"   "0.113903"     "-0.100806"    "-0.0600464"
[406] "0.239589"     "0.0253211"    "-0.00021077"  "0.134746"     "-0.00280059"
[411] "-0.0187158"   "0.0791351"    "-0.0365909"   "-0.0378554"   "0.0900315"
[416] "-0.0332589"   "-0.038724"    "0.0162428"    "-0.0300598"   "0.053535"
[421] "-0.00816293"  "-0.0364476"   "0.256276"     "0.0318442"    "-0.0153879"
[426] "0.237328"     "-0.0152552"   "-0.0330509"   "0.081732"     "-0.0260483"
[431] "-0.0416431"   "0.108835"     "-0.00922453"  "-0.0255428"   "0.25614"
[436] "-0.0263969"   "-0.0346336"   "0.120812"     "-0.0402302"   "-0.0414771"
[441] "0.145762"     "-0.00574315"  "-0.0127784"   "0.126006"     "0.021455"
[446] "-0.00605063"  "0.139934"     "-0.00331317"  "-0.0293562"   "0.0707318"
[451] "-0.0615362"   "-0.0512842"   "0.171448"     "-0.00672612"  "-0.0250951"
[456] "0.0937763"    "-0.00826427"  "-0.0271846"   "0.180796"     "-0.0192484"
[461] "-0.0429829"   "0.133553"     "0.0194617"    "-0.0130215"   "0.0555605"
[466] "-0.00310581"  "-0.0303713"   "0.116832"     "-0.0292164"   "-0.0302653"
[471] "0.141538"     "0.022605"     "-0.0168009"   "0.13919"      "0.0290067"
[476] "-0.0183177"   "0.116014"     "-0.0157294"   "-0.0191159"   "0.114408"
[481] "-0.0114867"   "-0.046426"    "0.11318"      "-0.00209735"  "-0.0201756"
[486] "0.0885676"    "-0.0163522"   "-0.0470814"   "0.0638346"    "-0.0324004"
[491] "-0.0450172"   "0.0600393"    "-0.00326473"  "-0.0515564"   "0.0836378"
[496] "-0.0303393"   "-0.0621737"   "0.13985"      "0.00622292"   "-0.0270494"
[501] "0.128134"     "-0.0131302"   "-0.0364662"   "0.135931"     "0.0238204"
[506] "-0.00250665"  "0.161623"     "-0.0467674"   "-0.0493532"   "0.0600108"
[511] "-0.00969187"  "-0.0596594"   "0.145024"     "-0.0376974"   "-0.0484321"
[516] "0.0565368"    "-0.0130286"   "-0.0271526"   "0.154523"     "-0.0131807"
[521] "-0.0451755"   "0.113626"     "-0.0383441"   "-0.044223"    "0.0817473"
```

```
[526] "0.012281"     "-0.0392017"   "0.096867"     "-0.0236485"   "-0.0602265"
[531] "0.135283"     "-0.0547812"   "-0.0596493"   "0.0978303"    "-0.0133438"
[536] "-0.0452658"   "0.151355"     "-0.0103895"   "-0.0179634"   "0.202614"
[541] "-0.0133574"   "-0.0175481"   "0.130016"     "-0.0325299"   "-0.0465055"
[546] "0.102502"     "-0.01108"     "-0.0234563"   "0.111016"     "-0.00780772"
[551] "-0.0495888"   "0.0692309"    "-0.00952532"  "-0.0452825"   "0.172192"
[556] "-0.00985301"  "-0.0359613"   "0.182642"     "-0.0225145"   "-0.0416749"
[561] "0.0962406"    "-0.00927879"  "-0.0384552"   "0.0785632"    "-0.0472972"
[566] "-0.0590447"   "0.137003"     "-0.0122332"   "-0.025317"    "0.216779"
[571] "0.0154012"    "-0.0297259"   "0.0664252"    "-0.0222947"   "-0.0369511"
[576] "0.107636"     "-0.0149354"   "-0.033219"    "0.0498347"    "-0.00565523"
[581] "-0.0433327"   "0.156404"     "-0.006563"    "-0.0303408"   "0.137037"
[586] "-0.0043323"   "-0.0262894"   "0.128957"     "-0.011593"    "-0.0349489"
[591] "0.153385"     "0.00142098"   "-0.0097036"   "0.15617"      "-0.0362487"
[596] "-0.049046"    "0.141879"     "0.00173955"   "-0.0237131"   "0.113199"
[601] "-0.0122241"   "-0.028493"    "0.160358"     "-0.0217398"   "-0.0636935"
[606] "0.150557"     "-0.0168562"   "-0.0316124"   "0.0885225"    "-0.0176003"
[611] "-0.0375996"   "0.18138"      "0.0054587"    "-0.0128398"   "0.119523"
[616] "0.00138007"   "-0.013513"    "0.206691"     "-0.0107246"   "-0.0320579"
[621] "0.110431"     "-0.0140905"   "-0.0296777"   "0.195532"     "0.000832394"
[626] "-0.0366896"   "0.0804631"    "0.00320842"   "-0.048028"    "0.0834082"
[631] "-0.0154136"   "-0.0405693"   "0.0786511"    "0.0114445"    "-0.0311136"
[636] "0.0703189"    "-0.0519029"   "-0.0571781"   "0.141083"     "7.60126e-05"
[641] "-0.00525737"  "0.147303"     "-0.0121747"   "-0.0208039"   "0.105252"
[646] "-0.0254071"   "-0.0545692"   "0.132921"     "-0.014789"    "-0.0353072"
[651] "0.113532"     "-0.00181898"  "-0.0357145"   "0.0594148"    "-0.0177323"
[656] "-0.0487605"   "0.146029"     "0.0123034"    "-0.012518"    "0.125668"
[661] "-0.0405879"   "-0.0469012"   "0.0525411"    "-0.00558911"  "-0.0443793"
[666] "0.0964632"    "-0.00993485"  "-0.0469478"   "0.113991"     "-0.0475557"
[671] "-0.0641161"   "0.165193"     "-0.0376724"   "-0.0303257"   "0.0825636"
[676] "-0.0321436"   "-0.0362561"   "0.111854"     "-0.0253142"   "-0.0333199"
[681] "0.191328"     "-0.0388725"   "-0.0471375"   "0.109017"     "-0.0313411"
[686] "-0.0343265"   "0.142782"     "-0.00212741"  "-0.0271065"   "0.316778"
[691] "0.0672675"    "0.0114322"    "0.2448"       "0.00188158"   "-0.00662405"
```

```
[696] "0.157448"     "-0.0135308"   "-0.0220779"   "0.0448242"    "-0.0394565"
[701] "-0.0376439"   "0.0517516"    "-0.0165177"   "-0.0372831"   "0.15636"
[706] "-0.00136382"  "-0.0150206"   "0.145429"     "-0.0176364"   "-0.0264678"
[711] "0.0394156"    "0.0013846"    "-0.0425716"   "0.108247"     "-0.0190224"
[716] "-0.0269678"   "0.124904"     "-0.0146875"   "-0.0468655"   "0.130166"
[721] "0.01718"      "-0.0197003"   "0.113579"     "-0.0127142"   "-0.0317998"
[726] "0.0948791"    "0.00170895"   "-0.0311209"   "0.107903"     "0.00794105"
[731] "-0.0107732"   "0.0849295"    "-0.048675"    "-0.0464952"   "0.102294"
[736] "0.0108946"    "-0.0157981"   "0.135917"     "-0.0422271"   "-0.0563502"
[741] "0.133404"     "-0.000815716" "-0.0310407"   "0.110037"     "-0.0165128"
[746] "-0.0410405"   "0.154785"     "-0.0245455"   "-0.0424648"   "0.127039"
[751] "-0.0182603"   "-0.0343107"   "0.116968"     "-0.0217072"   "-0.0487972"
[756] "0.0984869"    "-0.0302001"   "-0.0426944"   "0.111688"     "-0.00441595"
[761] "-0.0368021"   "0.108995"     "-0.049961"    "-0.0398778"   "0.263623"
[766] "0.0370342"    "-0.0177461"   "0.117361"     "0.0288729"    "-0.00545463"
[771] "0.223527"     "0.0399772"    "-0.0054044"   "0.0898512"    "-0.0188695"
[776] "-0.0288428"   "0.118132"     "-0.0331514"   "-0.0398928"   "0.129312"
[781] "-0.00261119"  "-0.0194545"   "0.116613"     "0.0125516"    "-0.0323628"
[786] "0.145544"     "-0.0196772"   "-0.0168556"   "0.184315"     "0.00228708"
[791] "-0.0240086"   "0.105297"     "-0.00313683"  "-0.0268406"   "0.0965665"
[796] "-0.0279447"   "-0.0371252"   "0.266874"     "0.0468823"    "-0.000712649"
[801] "0.105379"     "-0.024423"    "-0.0398328"   "0.0759229"    "-0.03658"
[806] "-0.0380866"   "0.13549"      "-0.0121045"   "-0.016148"    "0.141384"
[811] "-0.0219627"   "-0.0475628"   "0.131304"     "0.000154671"  "-0.0338689"
[816] "0.114058"     "-0.00197001"  "-0.0274381"   "0.134066"     "0.0102423"
[821] "-0.0226342"   "0.109856"     "0.00247413"   "-0.0386299"   "0.114525"
[826] "0.00116333"   "-0.0323984"   "0.138583"     "-0.0147536"   "-0.0410308"
[831] "0.157734"     "-0.0188978"   "-0.0220031"   "0.0618972"    "-0.0141444"
[836] "-0.0324802"   "0.096893"     "-0.0200816"   "-0.0369858"   "0.0642268"
[841] "-0.037566"    "-0.0382265"   "0.102832"     "0.00411569"   "-0.0484505"
[846] "0.102461"     "0.0538331"    "0.0033227"    "0.201126"     "-0.0459456"
[851] "-0.0352084"   "0.137912"     "0.00586651"   "-0.00982986"  "0.132184"
[856] "0.00278613"   "-0.0303812"   "0.106092"     "0.0172967"    "-0.0226362"
[861] "0.153704"     "-0.0378785"   "-0.0473122"   "0.134259"     "-0.00751592"
```

```
[866] "-0.0527163"   "0.11659"      "0.0102108"    "-0.0411244"   "0.155539"
[871] "0.0143908"    "-0.0369336"   "0.0755889"    "0.0192428"    "-0.0176359"
[876] "0.0984393"    "-0.0407641"   "-0.0537309"   "0.132523"     "-0.0120527"
[881] "-0.0253224"   "0.184773"     "0.00433056"   "-0.0433399"   "0.123165"
[886] "0.00728515"   "-0.0192025"   "0.0742313"    "0.00178023"   "-0.0299882"
[891] "0.18536"      "-0.0188647"   "-0.0370202"   "0.105274"     "0.00364877"
[896] "-0.00987043"  "0.122449"     "-0.0262211"   "-0.0292588"
```

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   8.6491400
5    1    THETA2  21.5594000
8    1    THETA3   0.0766718
11   1    THETA4   4.0575900
14   1    THETA5 106.2570000
17   1    THETA6   1.0756200
```

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   8.6491400
2   1    THETA2  21.5594000
3   1    THETA3   0.0766718
4   1    THETA4   4.0575900
5   1    THETA5 106.2570000
6   1    THETA6   1.0756200
```

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   8.6491400       CL/F
2   1    THETA2  21.5594000      V_c/F
3   1    THETA3   0.0766718        K_a
4   1    THETA4   4.0575900        Q/F
5   1    THETA5 106.2570000      V_p/F
6   1    THETA6   1.0756200 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   8.6491400      CL/F
2   1    THETA2  21.5594000     V_c/F
3   1    THETA3   0.0766718       K_a
4   1    THETA4   4.0575900       Q/F
5   1    THETA5 106.2570000     V_p/F
6   1    THETA6   1.0756200 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 8.649140      CL/F
6    1    THETA6 1.075620 MALE_CL/F
7    1    THETA7 1.443180   WT_CL/F
16   2    THETA1 9.550680      CL/F
21   2    THETA6 0.978555 MALE_CL/F
22   2    THETA7 0.427236   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  8.64914  1.075620 1.443180
2   2  9.55068  0.978555 0.427236
3   3       NA  1.280170 1.571080
4   4 10.65860  0.983133       NA
5   5  9.53191  0.996753 1.133910
6   6  9.95041  1.013770 0.571955
```

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  8.64914  1.075620 1.443180  0.8985696   1.104695   1.323397
2   2  9.55068  0.978555 0.427236  0.9688344   1.029915   1.086488
3   3       NA  1.280170 1.571080  0.8900928   1.114486   1.356672
4   4 10.65860  0.983133       NA         NA         NA         NA
5   5  9.53191  0.996753 1.133910  0.9194020   1.081373   1.246270
6   6  9.95041  1.013770 0.571955  0.9584993   1.040250   1.117449
```

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 0.9097138  1.075620 1.443180  0.8985696   1.104695   1.323397
2   2 1.0045375  0.978555 0.427236  0.9688344   1.029915   1.086488
3   3        NA  1.280170 1.571080  0.8900928   1.114486   1.356672
4   4 1.1210681  0.983133       NA         NA         NA         NA
5   5 1.0025632  0.996753 1.133910  0.9194020   1.081373   1.246270
6   6 1.0465809  1.013770 0.571955  0.9584993   1.040250   1.117449
```

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 0.9097138
2   2     CL/F 1.0045375
3   4     CL/F 1.1210681
4   5     CL/F 1.0025632
5   6     CL/F 1.0465809
6   7     CL/F 1.1491826
```

1.4.4 Plot.

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

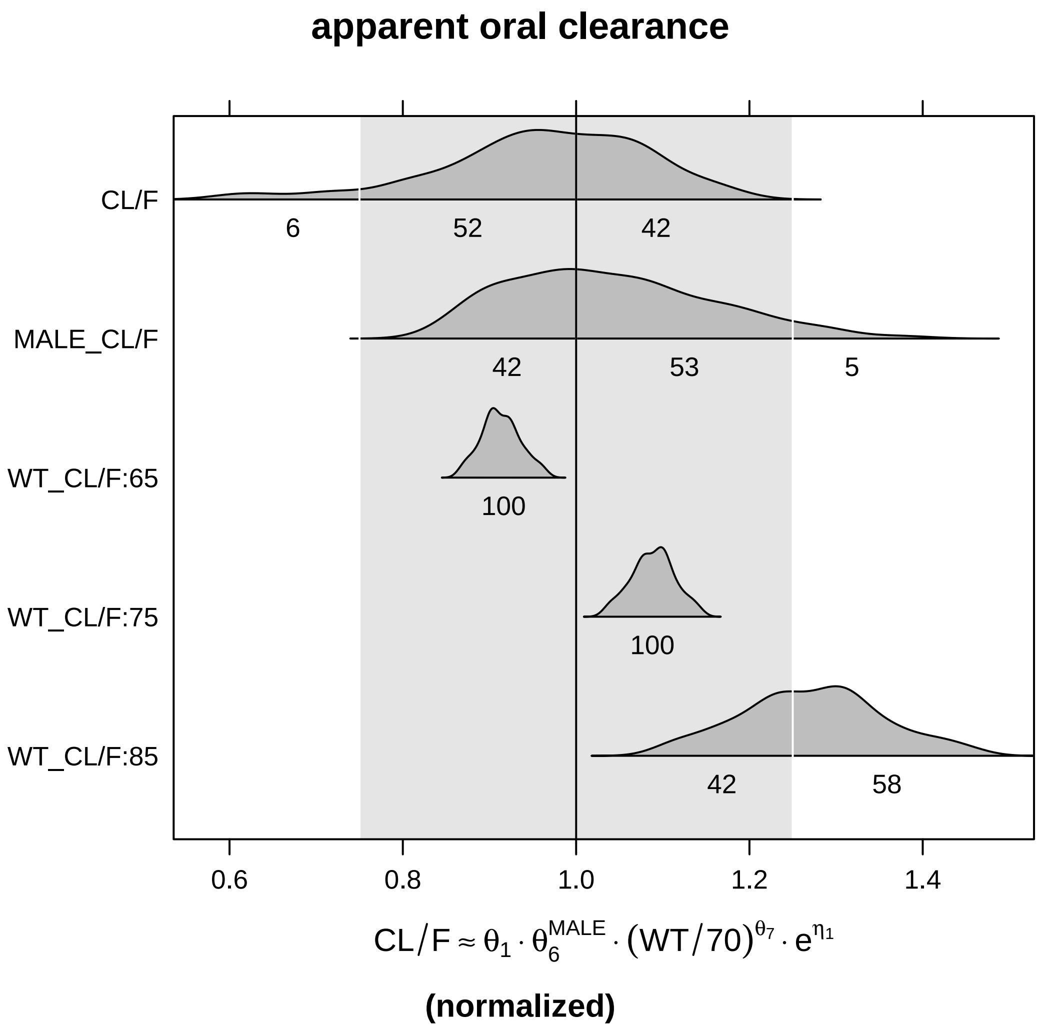
Listing 36:

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


apparent oral clearance
CL/F
6
52
42
MALE_CL/F
42
53
5
WT_CL/F:65
100
WT_CL/F:75
100
WT_CL/F:85
42
58
0.6
0.8
1.0
1.2
1.4
$CL/F \approx \theta_1 \cdot \theta_6^{MALE} \cdot (WT/70)^{\theta_7} \cdot e^{\eta_1}$
(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.