Covariate Plots

October 24, 2014

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 2459.17577212358
2 2  nm7   1    THETA1 estimate          9.90624
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 9.90624
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   9.90624
5    1    THETA2   21.8851
8    1    THETA3 0.0708172
11   1    THETA4   3.36908
14   1    THETA5   94.6441
17   1    THETA6  0.972458
```

Listing 11:

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

```
  [1] "0.118664"     "0.00243896"   "-0.0290797"   "0.126793"     "0.00496537"
  [6] "-0.0348756"   "0.0793852"    "0.0126321"    "-0.0254622"   "0.0930784"
 [11] "-0.00800534"  "-0.0604644"   "0.0776862"    "-0.0332063"   "-0.0431811"
 [16] "0.103248"     "-0.00113366"  "-0.0399984"   "0.124331"     "-0.00239167"
 [21] "-0.029284"    "0.0929795"    "0.0060518"    "-0.0318701"   "0.127233"
 [26] "0.0107017"    "-0.0244607"   "0.112813"     "0.0269052"    "-0.00833897"
 [31] "0.089781"     "0.00380984"   "-0.0419745"   "0.145258"     "-0.0511888"
 [36] "-0.034809"    "0.123498"     "0.0100472"    "-0.0206121"   "0.0876049"
 [41] "-0.0100154"   "-0.0246587"   "0.0852641"    "-0.00160618"  "-0.0344951"
 [46] "0.129994"     "0.0285775"    "-0.0412475"   "0.0885414"    "-0.00653592"
 [51] "-0.0477025"   "0.128111"     "-0.0431012"   "-0.0414133"   "0.0643106"
 [56] "-0.0278942"   "-0.0369338"   "0.190189"     "-0.0205082"   "-0.0254159"
 [61] "0.118579"     "-0.00753156"  "-0.0254262"   "0.0984033"    "-0.0268537"
 [66] "-0.0508149"   "0.128197"     "0.0232717"    "-0.0236485"   "0.167175"
 [71] "-0.0217408"   "-0.0381045"   "0.165601"     "0.00264623"   "-0.0201151"
 [76] "0.0947895"    "-0.0169357"   "-0.0396992"   "0.0463236"    "-0.00590113"
 [81] "-0.0567564"   "0.194381"     "-0.016843"    "-0.0245055"   "0.104538"
 [86] "0.00451804"   "-0.0224571"   "0.106584"     "-0.0108647"   "-0.0250814"
 [91] "0.108904"     "-0.0111865"   "-0.02692"     "0.099795"     "-0.0395158"
 [96] "-0.0396872"   "0.0850947"    "-0.0237443"   "-0.0408458"   "0.118172"
[101] "-0.035141"    "-0.0617929"   "0.11275"      "-0.0256919"   "-0.0452782"
[106] "0.238867"     "0.0421172"    "-0.0113253"   "0.14246"      "-0.0102746"
[111] "-0.0246251"   "0.17737"      "0.0528248"    "0.00957745"   "0.106911"
[116] "0.00847151"   "-0.0370734"   "0.0610825"    "-0.0328265"   "-0.0478436"
[121] "0.144272"     "0.00444813"   "-0.0430471"   "0.132424"     "-0.00549816"
[126] "-0.0287111"   "0.0982603"    "-0.000319021" "-0.0017437"   "0.171037"
[131] "0.0245734"    "-0.00064495"  "0.0966426"    "-0.0427972"   "-0.0422852"
[136] "0.104497"     "-0.00685034"  "-0.0241402"   "0.0483264"    "-0.0161017"
[141] "-0.0432612"   "0.10326"      "0.0087696"    "-0.0425963"   "0.0835945"
[146] "-0.000345655" "-0.0447935"   "0.112744"     "0.00295219"   "-0.0384519"
[151] "0.179545"     "0.0253152"    "-0.017339"    "0.0567219"    "0.00398271"
[156] "-0.0299789"   "0.180876"     "-0.00185966"  "-0.0249431"   "0.117255"
[161] "0.0146557"    "-0.0264507"   "0.0867032"    "-0.0341645"   "-0.0468786"
[166] "0.161076"     "0.0163088"    "0.00365636"   "0.110393"     "-0.0199049"
[171] "-0.0610041"   "0.0933731"    "0.00429236"   "-0.0585371"   "0.131606"
[176] "-0.0273357"   "-0.0414518"   "0.0740837"    "-0.0393725"   "-0.0532824"
[181] "0.114814"     "0.000498372"  "-0.0327205"   "0.166113"     "0.0260557"
[186] "-0.013542"    "0.202145"     "0.0177434"    "-0.0210069"   "0.0910233"
[191] "0.0151667"    "-0.0408356"   "0.0869729"    "0.0132574"    "-0.0369298"
[196] "0.121655"     "-0.0173966"   "-0.0312672"   "0.117305"     "-0.00249383"
[201] "-0.0312059"   "0.069709"     "-0.0238348"   "-0.0435522"   "0.157213"
[206] "0.0276325"    "-0.0167408"   "0.103765"     "-0.0320893"   "-0.0491547"
[211] "0.127115"     "0.00963332"   "-0.0315349"   "0.109701"     "-0.00298643"
[216] "-0.0269827"   "0.163874"     "-0.0222174"   "-0.0279429"   "0.149759"
[221] "-0.0606384"   "-0.0582304"   "0.156683"     "-0.00684463"  "-0.0128832"
[226] "0.132937"     "0.0117909"    "-0.0325853"   "0.0667211"    "-0.0396385"
[231] "-0.0444916"   "0.16451"      "0.00956835"   "-0.0156386"   "0.0973435"
[236] "-0.00795893"  "-0.0376994"   "0.1143"       "-0.00646968"  "-0.0362551"
[241] "0.130343"     "-0.0293751"   "-0.0610221"   "0.146619"     "-0.000407164"
[246] "-0.0189185"   "0.137894"     "0.000294066"  "-0.0289474"   "0.0894661"
```

```
[251] "-0.0458925"  "-0.0433672"  "0.146665"    "0.0142544"   "-0.00460381"
[256] "0.128807"    "0.00755358"  "-0.0270419"  "0.173962"    "0.0191587"
[261] "-0.0230961"  "0.105145"    "-0.0287821"  "-0.0461986"  "0.174007"
[266] "-0.0250103"  "-0.0154687"  "0.157457"    "-0.024208"   "-0.043364"
[271] "0.11283"     "-0.0196416"  "-0.035826"   "0.110426"    "-0.0343319"
[276] "-0.0621871"  "0.119436"    "0.000846538" "-0.0184177"  "0.0932987"
[281] "-0.0145868"  "-0.0412257"  "0.116972"    "-0.0102762"  "-0.0421894"
[286] "0.12102"     "-0.0340955"  "-0.0461667"  "0.20483"     "0.00482516"
[291] "-0.0163381"  "0.102248"    "-0.0446729"  "-0.0417648"  "0.100401"
[296] "-0.0187281"  "-0.0527303"  "0.105437"    "-0.0330351"  "-0.0412061"
[301] "0.133189"    "-0.0168328"  "-0.0265733"  "0.0945628"   "-0.023821"
[306] "-0.046713"   "0.115873"    "-0.0174054"  "-0.0383742"  "0.151988"
[311] "-0.00223515" "-0.0378195"  "0.111794"    "-0.0362"     "-0.0342003"
[316] "0.115687"    "-0.0487321"  "-0.0605172"  "0.0491989"   "-0.0400207"
[321] "-0.0576997"  "0.0924036"   "-0.00301072" "-0.0217227"  "0.120697"
[326] "-0.0180288"  "-0.0419027"  "0.0841434"   "-0.0272731"  "-0.0373285"
[331] "0.139445"    "-0.0562158"  "-0.0628585"  "0.133842"    "-0.0058623"
[336] "-0.0465414"  "0.117257"    "0.00585463"  "-0.0212939"  "0.141695"
[341] "-0.0128165"  "-0.0454878"  "0.0762859"   "-0.0419356"  "-0.0446045"
[346] "0.115748"    "-0.0270666"  "-0.0334317"  "0.13824"     "0.0159619"
[351] "-0.0182228"  "0.153652"    "-0.0133617"  "-0.0312735"  "0.129189"
[356] "-0.00427276" "-0.0375778"  "0.0784215"   "-0.0189919"  "-0.0278138"
[361] "0.0859133"   "-0.0112831"  "-0.0467855"  "0.152543"    "-0.0117078"
[366] "-0.0259284"  "0.146406"    "-0.00833782" "-0.0340645"  "0.117956"
[371] "-0.0228683"  "-0.0302881"  "0.0998222"   "-0.0056598"  "-0.0270215"
[376] "0.148125"    "-0.035818"   "-0.0466027"  "0.154802"    "-0.00387403"
[381] "-0.0344275"  "0.0821857"   "0.0179231"   "-0.0208862"  "0.159922"
[386] "-0.00843247" "-0.0361851"  "0.154316"    "-0.0204364"  "-0.0313654"
[391] "0.0876008"   "0.0186172"   "-0.0384452"  "0.145706"    "-0.0513642"
[396] "-0.0353288"  "0.0960684"   "-0.0153065"  "-0.0325897"  "0.113952"
[401] "-0.034477"   "-0.0391484"  "0.120386"    "-0.0235295"  "-0.040302"
[406] "0.146426"    "-0.00909298" "-0.0229452"  "0.097815"    "-0.0228671"
[411] "-0.0477668"  "0.0527434"   "-0.0401562"  "-0.0404198"  "0.191286"
[416] "0.0233172"   "0.00230177"  "0.0966339"   "-0.010117"   "-0.0304394"
[421] "0.102042"    "-0.0675102"  "-0.0323489"  "0.0669474"   "-0.00414405"
[426] "-0.0350421"  "0.117324"    "0.019366"    "-0.0293495"  "0.043366"
[431] "-0.037891"   "-0.0554599"  "0.116669"    "-0.0318554"  "-0.0605897"
[436] "0.0694246"   "-0.0246743"  "-0.0545532"  "0.0898996"   "-0.0190038"
[441] "-0.0526655"  "0.115315"    "-0.0448101"  "-0.0434573"  "0.121016"
[446] "-0.00117652" "-0.040854"   "0.0741172"   "-0.0189367"  "-0.0253948"
[451] "0.104378"    "-0.00161245" "-0.02001"    "0.157005"    "-0.00523799"
[456] "-0.0247991"  "0.351464"    "0.0448184"   "0.0023031"   "0.118066"
[461] "-0.0221416"  "-0.0276645"  "0.114711"    "-0.00405511" "-0.0277706"
[466] "0.125923"    "-0.0129499"  "-0.0347455"  "0.0982356"   "0.0112521"
[471] "-0.0208778"  "0.069048"    "-0.0578171"  "-0.0478397"  "0.116005"
[476] "-0.0531913"  "-0.0461022"  "0.189958"    "0.023422"    "-0.00411683"
[481] "0.0874007"   "-0.0666566"  "-0.0453463"  "0.250447"    "0.00770081"
[486] "-0.0208701"  "0.167599"    "0.0451788"   "-0.00065829" "0.102168"
[491] "-0.0143335"  "-0.0314068"  "0.089994"    "-0.0436014"  "-0.0577496"
[496] "0.0724951"   "-0.0250448"  "-0.0245528"  "0.105756"    "-0.0395233"
```

```
[501] "-0.031799"    "0.113582"    "0.0199422"   "-0.0149443"   "0.0744757"
[506] "-0.0676757"   "-0.045086"   "0.0890981"   "-0.0412376"   "-0.0493254"
[511] "0.114201"     "-0.0385651"  "-0.0429911"  "0.0888071"    "-0.0233529"
[516] "-0.0528072"   "0.043756"    "-0.0220733"  "-0.0363111"   "0.108755"
[521] "-0.00844895"  "-0.0437119"  "0.0888473"   "-0.0272006"   "-0.0455575"
[526] "0.109073"     "0.0282737"   "-0.0144904"  "0.129467"     "-0.00760703"
[531] "-0.0198483"   "0.124011"    "0.0141876"   "-0.0382787"   "0.0587984"
[536] "-0.0244563"   "-0.0366547"  "0.151269"    "-0.00472419"  "-0.029383"
[541] "0.174937"     "-0.00865366" "-0.0339614"  "0.156336"     "-0.0134474"
[546] "-0.0319209"   "0.146132"    "-0.0145849"  "-0.0205749"   "0.146571"
[551] "-0.014698"    "-0.0412586"  "0.164571"    "-0.0107431"   "-0.0206866"
[556] "0.0803535"    "-0.0214819"  "-0.0432427"  "0.112315"     "-0.0225172"
[561] "-0.0452995"   "0.182547"    "-0.0240036"  "-0.0307118"   "0.148057"
[566] "-0.00531293"  "-0.0421697"  "0.10471"     "0.00909561"   "-0.0103992"
[571] "0.141531"     "-0.0117441"  "-0.0268305"  "0.055915"     "-0.0145141"
[576] "-0.0399355"   "0.2861"      "0.0647719"   "0.00905442"   "0.226185"
[581] "0.0465552"    "-0.0167005"  "0.0863951"   "-0.0242882"   "-0.0445673"
[586] "0.106754"     "0.00710941"  "-0.0384524"  "0.128791"     "0.00935985"
[591] "-0.0255152"   "0.151828"    "0.0441336"   "0.00135239"   "0.112871"
[596] "-0.00344835"  "-0.022351"   "0.0481443"   "-0.0179547"   "-0.055449"
[601] "0.0818499"    "-0.0253572"  "-0.0342841"  "0.0963881"    "-0.00883748"
[606] "-0.0304162"   "0.139391"    "-0.0187507"  "-0.0402836"   "0.155407"
[611] "-0.0104272"   "-0.0216455"  "0.0635618"   "-0.00394322"  "-0.0362427"
[616] "0.134227"     "0.00362554"  "-0.00676369" "0.0945227"    "-0.0698679"
[621] "-0.0602625"   "0.0923166"   "-0.0150987"  "-0.0350389"   "0.081674"
[626] "-0.00441006"  "-0.0490822"  "0.128433"    "-0.0261758"   "-0.0399649"
[631] "0.109765"     "-0.0263731"  "-0.0386598"  "0.0884195"    "0.0352562"
[636] "-0.0224681"   "0.12504"     "-0.016216"   "-0.0186849"   "0.0836959"
[641] "0.00447469"   "-0.0381655"  "0.113755"    "0.0275129"    "-0.00949459"
[646] "0.0651385"    "-0.0287313"  "-0.0593346"  "0.12926"      "-0.0386841"
[651] "-0.0235969"   "0.141795"    "0.00184889"  "-0.0213231"   "0.113659"
[656] "-0.0188672"   "-0.0347941"  "0.0657835"   "-0.0261609"   "-0.051177"
[661] "0.119641"     "-0.010961"   "-0.0345783"  "0.107459"     "-0.0279097"
[666] "-0.0412287"   "0.128838"    "-0.00840944" "-0.0275247"   "0.0641978"
[671] "-0.0448826"   "-0.0548623"  "0.105479"    "-0.00756974"  "-0.0405811"
[676] "0.171146"     "0.00200264"  "-0.01219"    "0.0862845"    "-0.0229536"
[681] "-0.0273753"   "0.183248"    "0.00835915"  "-0.0156605"   "0.0791216"
[686] "-0.0363752"   "-0.0454898"  "0.233876"    "0.00372023"   "-0.0186535"
[691] "0.142954"     "-0.00156208" "-0.0336852"  "0.0595711"    "-0.023845"
[696] "-0.0408747"   "0.0778225"   "-0.0396712"  "-0.0301178"   "0.0918891"
[701] "-0.0157744"   "-0.0291887"  "0.11211"     "0.0144046"    "-0.0306082"
[706] "0.138055"     "-0.0309795"  "-0.043204"   "0.138132"     "0.00912754"
[711] "-0.0332121"   "0.138756"    "-0.0134344"  "-0.0507371"   "0.124444"
[716] "-0.0479321"   "-0.0479316"  "0.171498"    "-0.0121693"   "-0.024209"
[721] "0.0540019"    "-0.0110472"  "-0.0497729"  "0.0957406"    "-0.0272068"
[726] "-0.0377253"   "0.105232"    "-0.0423657"  "-0.0309091"   "0.0727367"
[731] "-0.0061838"   "-0.0425191"  "0.14017"     "-0.0588466"   "-0.0585397"
[736] "0.117701"     "-0.0279007"  "-0.0488742"  "0.141549"     "0.0282864"
[741] "-0.00360357"  "0.150651"    "0.00336836"  "-0.0222498"   "0.141123"
[746] "-0.0345781"   "-0.0358519"  "0.126264"    "0.00663694"   "-0.0317072"
```

```
[751] "0.127508"     "-0.0124047"  "-0.0283794"  "0.131374"    "-0.0134399"
[756] "-0.0361739"   "0.148282"    "-0.0190484"  "-0.0179618"  "0.121144"
[761] "-0.0326408"   "-0.051974"   "0.115299"    "-0.0400513"  "-0.0586101"
[766] "0.153749"     "-0.0078094"  "-0.0310534"  "0.072155"    "-0.0137717"
[771] "-0.0349942"   "0.106628"    "0.0016075"   "-0.0459419"  "0.13816"
[776] "-0.0181902"   "-0.0264274"  "0.0938884"   "-0.0191998"  "-0.0385028"
[781] "0.146527"     "-0.00176885" "-0.0262183"  "0.0941705"   "0.00247482"
[786] "-0.0389402"   "0.153674"    "0.0248971"   "0.0031693"   "0.135016"
[791] "-0.0159752"   "-0.0366186"  "0.150774"    "-0.0121317"  "-0.0210343"
[796] "0.100948"     "-0.0100324"  "-0.0380679"  "0.0781693"   "-0.0131155"
[801] "-0.0260249"   "0.183734"    "0.0471517"   "-0.00331566" "0.122793"
[806] "0.0128808"    "-0.022205"   "0.0961979"   "0.00881516"  "-0.0339731"
[811] "0.0988059"    "0.0129752"   "-0.0250672"  "0.106903"    "-0.0307499"
[816] "-0.0488798"   "0.199367"    "-0.00270252" "-0.034998"   "0.103325"
[821] "0.0245558"    "-0.00192005" "0.10619"     "0.00493672"  "-0.0361216"
[826] "0.0844764"    "0.00496451"  "-0.0254248"  "0.0585779"   "-0.00589244"
[831] "-0.0442521"   "0.0701998"   "-0.00732916" "-0.0466255"  "0.0715442"
[836] "-0.0347355"   "-0.0415529"  "0.0926787"   "-0.0344976"  "-0.0327243"
[841] "0.121283"     "-0.0321919"  "-0.0385139"  "0.099353"    "0.00059543"
[846] "-0.0240711"   "0.149382"    "-0.0155042"  "-0.0419845"  "0.158858"
[851] "0.0105719"    "-0.00492554" "0.067364"    "-0.0108857"  "-0.0470531"
[856] "0.127813"     "0.00668929"  "-0.0184073"  "0.148973"    "0.0134121"
[861] "-0.0248297"   "0.135644"    "0.0179563"   "-0.00793724" "0.0606008"
[866] "0.00193866"   "-0.0211141"  "0.0592926"   "-0.0327239"  "-0.0356362"
[871] "0.136618"     "-0.0223643"  "-0.0262967"  "0.106394"    "-0.0196676"
[876] "-0.0533358"   "0.0742905"   "-0.00833212" "-0.0373445"  "0.0998243"
[881] "-0.00384154"  "-0.0251419"  "0.170587"    "-0.0143729"  "-0.0394336"
[886] "0.0868"       "-0.0287053"  "-0.0297056"  "0.100429"    "0.00791036"
[891] "-0.0297891"   "0.0597762"   "-0.0391322"  "-0.03771"    "0.112944"
[896] "0.00219604"   "-0.017267"   "0.174094"    "0.0131618"   "-0.0141539"
```

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  9.9062400
5    1    THETA2 21.8851000
8    1    THETA3  0.0708172
11   1    THETA4  3.3690800
14   1    THETA5 94.6441000
17   1    THETA6  0.9724580
```

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  9.9062400
2   1    THETA2 21.8851000
3   1    THETA3  0.0708172
4   1    THETA4  3.3690800
5   1    THETA5 94.6441000
6   1    THETA6  0.9724580
```

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     9.50789 9.75    0.92708
2      22.791 9.55    2.17764
3   0.0714337 7.35 0.00525283
4     3.47451 15.4   0.535797
5     113.277   21    23.7452
6     1.02435 11.1   0.114056
7     1.19212 28.3    0.33679
8    0.213879 22.8  0.0488369
9     0.12077 26.4  0.0319144
10  0.0945105 33.2  0.0313616
11 -0.0116278  173  0.0200776
12 -0.0372064 36.1  0.0134244
13  0.0465631 34.8  0.0161816
14  0.0491707 10.9 0.00538135
15   0.201769 33.5  0.0676087
```

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  9.9062400      CL/F
2   1    THETA2 21.8851000     V_c/F
3   1    THETA3  0.0708172       K_a
4   1    THETA4  3.3690800       Q/F
5   1    THETA5 94.6441000     V_p/F
6   1    THETA6  0.9724580 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 HEIGHT WEIGHT SEX  AGE DOSE FED
1    C  1 0.00   0    0   NA 0.000    1 0.00    174   74.2   0 29.1 1000   1
2 <NA>  1 0.00   1    1 1000    NA    1 0.00    174   74.2   0 29.1 1000   1
3 <NA>  1 0.25   0    0   NA 0.363    1 0.25    174   74.2   0 29.1 1000   1
4 <NA>  1 0.50   0    0   NA 0.914    1 0.50    174   74.2   0 29.1 1000   1
5 <NA>  1 1.00   0    0   NA 1.120    1 1.00    174   74.2   0 29.1 1000   1
6 <NA>  1 2.00   0    0   NA 2.280    1 2.00    174   74.2   0 29.1 1000   1
  SMK DS CRCN TAFD  TAD LDOS MDV predose zerodv
1   0  0 83.5 0.00   NA   NA   0       1      0
2   0  0 83.5 0.00 0.00 1000   1       0      0
3   0  0 83.5 0.25 0.25 1000   0       0      0
4   0  0 83.5 0.50 0.50 1000   0       0      0
5   0  0 83.5 1.00 1.00 1000   0       0      0
6   0  0 83.5 2.00 2.00 1000   0       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  9.9062400       CL/F
2   1    THETA2 21.8851000      V_c/F
3   1    THETA3  0.0708172        K_a
4   1    THETA4  3.3690800        Q/F
5   1    THETA5 94.6441000      V_p/F
6   1    THETA6  0.9724580 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 9.906240      CL/F
6    1    THETA6 0.972458 MALE_CL/F
7    1    THETA7 1.469340   WT_CL/F
16   2    THETA1 9.030570      CL/F
21   2    THETA6 1.038960 MALE_CL/F
22   2    THETA7 0.999512   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  9.90624  0.972458 1.469340
2   2  9.03057  1.038960 0.999512
3   3  9.33170  0.846669 1.909640
4   4  9.25626  0.940994 1.697690
5   5 10.27090  1.252490 1.159250
6   6  9.42002  0.967179 1.484550
```

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] 9.50789
```

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  9.90624  0.972458 1.469340  0.8968292   1.106690   1.330136
2   2  9.03057  1.038960 0.999512  0.9286050   1.071392   1.214171
3   3  9.33170  0.846669 1.909640  0.8680382   1.140825   1.448847
4   4  9.25626  0.940994 1.697690  0.8817803   1.124264   1.390435
5   5 10.27090  1.252490 1.159250  0.9176771   1.083265   1.252417
6   6  9.42002  0.967179 1.484550  0.8958189   1.107852   1.334070
```

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 1.0418968  0.972458 1.469340  0.8968292   1.106690   1.330136
2   2 0.9497975  1.038960 0.999512  0.9286050   1.071392   1.214171
3   3 0.9814691  0.846669 1.909640  0.8680382   1.140825   1.448847
4   4 0.9735346  0.940994 1.697690  0.8817803   1.124264   1.390435
5   5 1.0802502  1.252490 1.159250  0.9176771   1.083265   1.252417
6   6 0.9907582  0.967179 1.484550  0.8958189   1.107852   1.334070
```

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 1.0418968
2   2     CL/F 0.9497975
3   3     CL/F 0.9814691
4   4     CL/F 0.9735346
5   5     CL/F 1.0802502
6   6     CL/F 0.9907582
```

1.4.4 Plot.

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

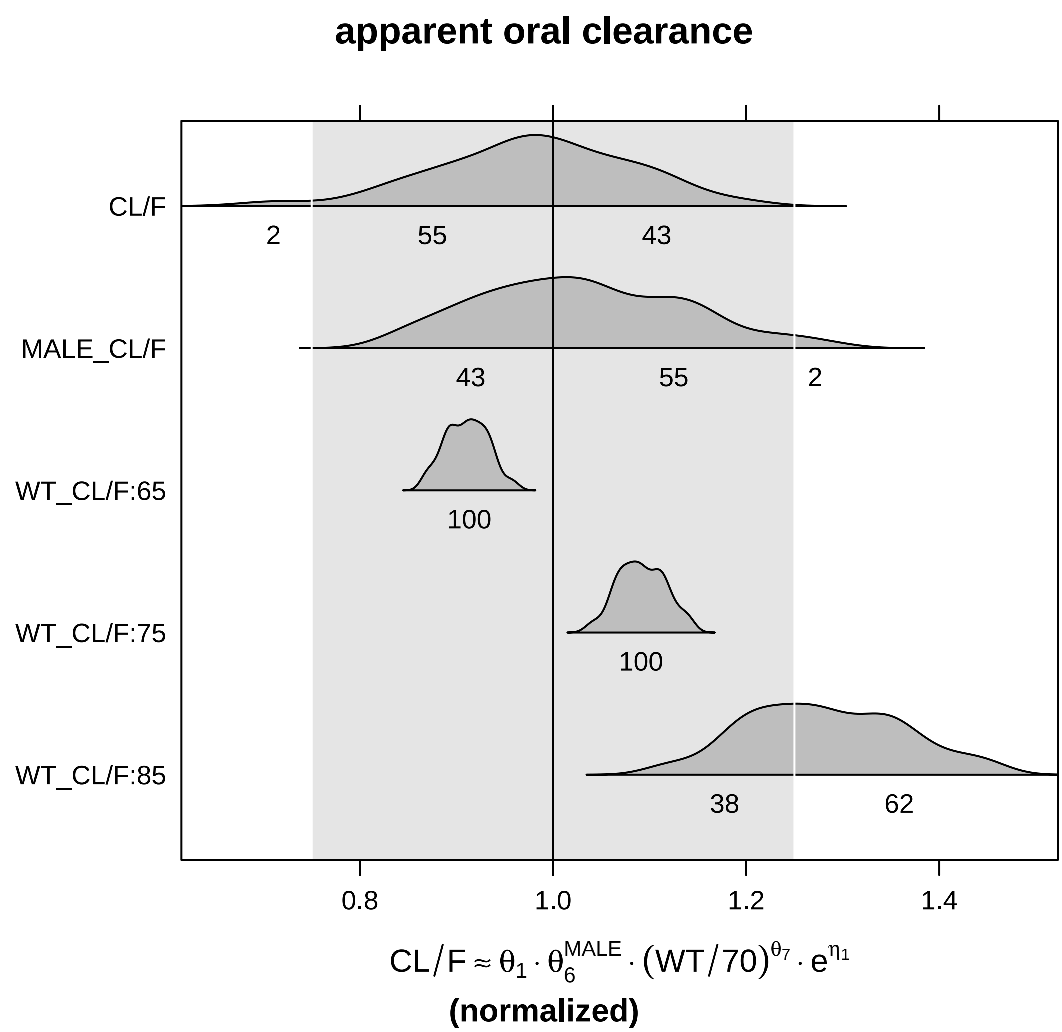
Listing 36:

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



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 some probability that an 85 kg person will have at least 25 percent greater clearance than a 70 kg person.