

Covariate Plots

May 21, 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/project/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	2467.30236668031
2	2	nm7	1	THETA1	estimate	10.2175
3	3	nm7	1	THETA1	prse	<NA>
4	4	nm7	1	THETA1	se	<NA>
5	5	nm7	1	THETA2	estimate	24.3171
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	10.2175
3	nm7	1	THETA1	prse	<NA>
4	nm7	1	THETA1	se	<NA>
5	nm7	1	THETA2	estimate	24.3171
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	10.2175
5	1	THETA2	24.3171
8	1	THETA3	0.0689622
11	1	THETA4	3.75759
14	1	THETA5	180.927
17	1	THETA6	0.972631

Listing 11:

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

```
[1] "0.119051"      "-0.000180592"  "-0.0351459"    "0.160918"      "-0.0468652"  
[6] "-0.0247546"    "0.233729"      "0.00135439"    "-0.0240276"    "0.122195"  
[11] "-0.0131397"    "-0.0437089"    "0.171962"      "-0.0349359"    "-0.0448369"
```

[16]	"0.107582"	"-0.0563908"	"-0.046028"	"0.0682629"	"-0.0383106"
[21]	"-0.0297646"	"0.122878"	"-0.0100309"	"-0.033563"	"0.12321"
[26]	"-0.0322801"	"-0.0500608"	"0.152055"	"-0.00892538"	"-0.00387561"
[31]	"0.0923754"	"-0.00999975"	"-0.0436539"	"0.0918378"	"-0.0620811"
[36]	"-0.0579281"	"0.112107"	"0.000269946"	"-0.0344167"	"0.108072"
[41]	"-0.013151"	"-0.0477776"	"0.135537"	"-0.00999128"	"-0.0505017"
[46]	"0.125247"	"-0.0339363"	"-0.0527939"	"0.0978788"	"-0.0479414"
[51]	"-0.0589469"	"0.112061"	"0.00413274"	"-0.00811303"	"0.148301"
[56]	"-0.0414546"	"-0.0389744"	"0.118716"	"-0.0191499"	"-0.0375006"
[61]	"0.119471"	"0.0143039"	"-0.0213005"	"0.0846409"	"-0.0346299"
[66]	"-0.0494923"	"0.125292"	"-0.0780693"	"-0.0554245"	"0.12932"
[71]	"-0.020177"	"-0.0404945"	"0.101297"	"0.0223134"	"-0.0369759"
[76]	"0.148803"	"-0.00290912"	"-0.0374595"	"0.167643"	"0.0119986"
[81]	"-0.0133665"	"0.120326"	"-0.0184556"	"-0.0379953"	"0.0567403"
[86]	"0.0217977"	"-0.0363667"	"0.0503781"	"-0.0292836"	"-0.0384853"
[91]	"0.1023"	"-0.0389275"	"-0.0368114"	"0.174382"	"-0.0260395"
[96]	"-0.0307639"	"0.165545"	"-0.0301535"	"-0.0435811"	"0.046039"
[101]	"-0.0333873"	"-0.0582988"	"0.103495"	"-0.016847"	"-0.0275739"
[106]	"0.158218"	"-0.0164931"	"-0.0412208"	"0.0630831"	"-0.0353153"
[111]	"-0.0575685"	"0.098713"	"-0.0669281"	"-0.0636005"	"0.150992"
[116]	"-0.0553599"	"-0.05384"	"0.128492"	"0.0184929"	"-0.0213927"
[121]	"0.0890435"	"-0.0427296"	"-0.0452522"	"0.09416"	"-0.00426671"
[126]	"-0.0586933"	"0.111846"	"-0.0134841"	"-0.0435239"	"0.205476"
[131]	"0.0327581"	"-0.00748341"	"0.0647775"	"-0.0227439"	"-0.0480752"
[136]	"0.0836974"	"-0.0121487"	"-0.0572991"	"0.0912959"	"-0.0436056"
[141]	"-0.0359514"	"0.0770188"	"-0.0241433"	"-0.0390269"	"0.0862018"
[146]	"-0.0040588"	"-0.0532202"	"0.0999502"	"-0.0253497"	"-0.0428425"
[151]	"0.12044"	"-0.00615923"	"-0.0264054"	"0.128415"	"-0.04583"
[156]	"-0.0505547"	"0.0769227"	"-0.00728049"	"-0.0386362"	"0.111309"
[161]	"0.00815661"	"-0.0123266"	"0.0565713"	"-0.039307"	"-0.0433977"
[166]	"0.122463"	"-0.017611"	"-0.0427158"	"0.0914727"	"-0.0373163"
[171]	"-0.0428836"	"0.115118"	"-0.00756888"	"-0.00727903"	"0.125168"
[176]	"-0.023686"	"-0.0423907"	"0.132452"	"-0.00273136"	"-0.0309037"
[181]	"0.116617"	"0.0108582"	"-0.0306744"	"0.116024"	"0.00874218"

[186]	"-0.0296331"	"0.138759"	"-0.0683127"	"-0.0438555"	"0.189366"
[191]	"0.0151989"	"-0.00926847"	"0.121471"	"-0.0239017"	"-0.0398277"
[196]	"0.09268"	"0.0039945"	"-0.0193032"	"0.0591578"	"-0.0149362"
[201]	"-0.044323"	"0.146896"	"-0.0324002"	"-0.0402981"	"0.110027"
[206]	"-0.0192553"	"-0.0548452"	"0.132528"	"-0.0207119"	"-0.0277938"
[211]	"0.127069"	"0.0105913"	"-0.0239616"	"0.126773"	"-0.0387041"
[216]	"-0.035095"	"0.0905951"	"0.00257814"	"-0.0266208"	"0.124992"
[221]	"0.00727043"	"-0.0131874"	"0.0996276"	"-0.0208238"	"-0.039991"
[226]	"0.118728"	"-0.00815922"	"-0.00711044"	"0.102443"	"-0.0234218"
[231]	"-0.0618831"	"0.120803"	"0.00425309"	"-0.0275259"	"0.119351"
[236]	"0.0151113"	"-0.0294246"	"0.0731433"	"-0.0345354"	"-0.0409155"
[241]	"0.0950186"	"-0.00192872"	"-0.0281212"	"0.106594"	"-0.00635315"
[246]	"-0.018829"	"0.121685"	"0.0171865"	"-0.0266126"	"0.113849"
[251]	"-0.000574062"	"-0.0178876"	"0.120099"	"-0.0232324"	"-0.0536901"
[256]	"0.0900706"	"0.00933255"	"-0.0195814"	"0.084177"	"0.00453573"
[261]	"-0.0260064"	"0.119259"	"-0.0402363"	"-0.0439391"	"0.056326"
[266]	"-0.0265285"	"-0.0306197"	"0.111643"	"-0.0265237"	"-0.0339386"
[271]	"0.153309"	"-0.0192626"	"-0.0466527"	"0.11408"	"0.0223671"
[276]	"-0.0241304"	"0.128141"	"0.0239277"	"-0.0118655"	"0.164084"
[281]	"-0.00478854"	"-0.0269591"	"0.114523"	"-0.00688562"	"-0.0250792"
[286]	"0.097906"	"-0.00513483"	"-0.0372698"	"0.0770277"	"-0.0126065"
[291]	"-0.03349"	"0.106434"	"-0.00622938"	"-0.0176153"	"0.099618"
[296]	"-0.0360498"	"-0.0396552"	"0.0794731"	"-0.0469355"	"-0.0416642"
[301]	"0.0887753"	"0.0154474"	"-0.0309217"	"0.0803378"	"0.00526525"
[306]	"-0.0396822"	"0.141655"	"-0.0505842"	"-0.0425454"	"0.0778128"
[311]	"-0.0200901"	"-0.0442403"	"0.100253"	"-0.0195989"	"-0.0382479"
[316]	"0.113288"	"-0.0183501"	"-0.0251172"	"0.153398"	"-0.0166292"
[321]	"-0.0473002"	"0.0935705"	"-0.0469584"	"-0.0539268"	"0.166873"
[326]	"0.00194385"	"-0.0332657"	"0.138673"	"0.00829756"	"-0.0284688"
[331]	"0.175101"	"0.00132006"	"-0.013095"	"0.107681"	"-0.0258877"
[336]	"-0.0246556"	"0.0866981"	"-0.0223248"	"-0.0393533"	"0.167534"
[341]	"-0.00549327"	"-0.0321292"	"0.106566"	"-0.00954148"	"-0.0403308"
[346]	"0.154392"	"-0.00321983"	"-0.0372449"	"0.0917156"	"0.013052"
[351]	"-0.0337191"	"0.0539502"	"-0.0241872"	"-0.0341665"	"0.12145"

[356]	"0.024811"	"-0.00789897"	"0.102479"	"-0.00375084"	"-0.0268973"
[361]	"0.168735"	"-0.0122592"	"-0.0239245"	"0.105811"	"-0.0100921"
[366]	"-0.0402121"	"0.121078"	"0.0188064"	"-0.0104297"	"0.125318"
[371]	"-0.00623269"	"-0.0412824"	"0.141397"	"-0.0100839"	"-0.0394641"
[376]	"0.196902"	"-0.0117792"	"-0.0246135"	"0.103343"	"-0.0192354"
[381]	"-0.0342719"	"0.119019"	"-0.00377079"	"-0.0428051"	"0.212707"
[386]	"-0.00182089"	"-0.00852011"	"0.108921"	"0.0308685"	"-0.0181617"
[391]	"0.078967"	"-0.0263966"	"-0.0647335"	"0.149222"	"-0.00466303"
[396]	"-0.0315204"	"0.166565"	"0.00174848"	"-0.0275089"	"0.11657"
[401]	"-0.0170873"	"-0.00149147"	"0.0996214"	"-0.0171483"	"-0.0473605"
[406]	"0.167097"	"-0.00449225"	"-0.0290761"	"0.111971"	"0.0118674"
[411]	"-0.028269"	"0.183087"	"-0.00310802"	"-0.0281029"	"0.205429"
[416]	"0.0184022"	"-0.0227506"	"0.100862"	"-0.0367313"	"-0.0382668"
[421]	"0.0914832"	"-0.0256853"	"-0.033501"	"0.143749"	"-0.00746803"
[426]	"-0.0158685"	"0.127771"	"1.22118e-05"	"-0.0219342"	"0.121675"
[431]	"0.00404428"	"-0.0306099"	"0.0874249"	"0.00708025"	"-0.0444552"
[436]	"0.0762115"	"-0.0260912"	"-0.0490801"	"0.147446"	"0.0143244"
[441]	"-0.00021218"	"0.145266"	"-0.0300578"	"-0.0412663"	"0.0938826"
[446]	"-0.00510198"	"-0.0162865"	"0.0565924"	"-0.029033"	"-0.0437357"
[451]	"0.130052"	"-0.0438933"	"-0.04623"	"0.0815868"	"-0.00845643"
[456]	"-0.0485005"	"0.0714602"	"-0.0458659"	"-0.0417598"	"0.127839"
[461]	"-0.0396236"	"-0.0315442"	"0.159601"	"0.000628593"	"-0.0205316"
[466]	"0.113267"	"-0.0671687"	"-0.0569785"	"0.115182"	"-0.00734051"
[471]	"-0.0331347"	"0.142631"	"-0.0243046"	"-0.0259693"	"0.128869"
[476]	"-0.0189191"	"-0.0317327"	"0.121122"	"-0.0271371"	"-0.0356392"
[481]	"0.120737"	"0.0220296"	"-0.0286796"	"0.120278"	"-0.0102112"
[486]	"-0.0207366"	"0.0859358"	"0.0242897"	"-0.0305029"	"0.0977561"
[491]	"-0.0217894"	"-0.0476644"	"0.141661"	"0.0113669"	"-0.00318492"
[496]	"0.15114"	"0.0161736"	"-0.0141236"	"0.0616868"	"-0.0504549"
[501]	"-0.0590268"	"0.0480405"	"-0.0110037"	"-0.0535986"	"0.169312"
[506]	"-0.0202765"	"-0.0334601"	"0.136846"	"-0.00992388"	"-0.0239469"
[511]	"0.144067"	"0.0236586"	"-0.0173407"	"0.0711178"	"-0.075803"
[516]	"-0.0653104"	"0.113578"	"-0.00930896"	"-0.0216874"	"0.103513"
[521]	"-0.0341801"	"-0.0568593"	"0.0832939"	"0.0224872"	"-0.0399623"

[526]	"0.0859065"	"0.00715787"	"-0.0486096"	"0.12984"	"-0.0424162"
[531]	"-0.0489495"	"0.124499"	"0.0104711"	"-0.025529"	"0.0602257"
[536]	"0.00874295"	"-0.0406506"	"0.165542"	"-0.0204549"	"-0.0496066"
[541]	"0.0745452"	"-0.0265083"	"-0.053669"	"0.154701"	"0.0025018"
[546]	"-0.0195319"	"0.123926"	"-0.0250167"	"-0.0199301"	"0.110462"
[551]	"-0.00972275"	"-0.0377979"	"0.0939223"	"-0.0694852"	"-0.0596595"
[556]	"0.121445"	"0.028588"	"-0.00765232"	"0.12779"	"-0.0123621"
[561]	"-0.0288411"	"0.181718"	"-0.00689684"	"-0.0208526"	"0.0992612"
[566]	"-0.00203196"	"-0.023401"	"0.202127"	"-0.0520711"	"-0.052738"
[571]	"0.101145"	"-0.00782279"	"-0.0382739"	"0.133395"	"-0.0305058"
[576]	"-0.0466035"	"0.167235"	"-0.0150321"	"-0.0213539"	"0.0938163"
[581]	"0.009329"	"-0.0416772"	"0.117827"	"-0.0139281"	"-0.0343238"
[586]	"0.093711"	"-0.0190713"	"-0.0612229"	"0.125046"	"0.0120298"
[591]	"-0.0316022"	"0.163622"	"-0.000561501"	"-0.019679"	"0.0800932"
[596]	"0.0237109"	"-0.0406148"	"0.0747374"	"-0.0255327"	"-0.041617"
[601]	"0.0997792"	"0.0225686"	"-0.0249263"	"0.122458"	"-0.0095075"
[606]	"-0.0389327"	"0.119309"	"-0.0175739"	"-0.0205478"	"0.151788"
[611]	"0.0106983"	"-0.0187007"	"0.125639"	"-0.0177143"	"-0.0334653"
[616]	"0.137314"	"0.00735405"	"-0.0215206"	"0.19345"	"-0.00584656"
[621]	"-0.0135711"	"0.170215"	"-0.0107052"	"-0.0413287"	"0.101739"
[626]	"-0.032947"	"-0.0442447"	"0.0799393"	"-0.00660673"	"-0.0242625"
[631]	"0.0842766"	"-0.0487907"	"-0.0524578"	"0.110874"	"-0.01377"
[636]	"-0.0408812"	"0.132308"	"-0.00921443"	"-0.0340802"	"0.0931245"
[641]	"-0.00248746"	"-0.0286588"	"0.111315"	"-0.0120779"	"-0.0252895"
[646]	"0.123242"	"-0.0615834"	"-0.0468585"	"0.125662"	"-0.0269262"
[651]	"-0.0453195"	"0.115024"	"-0.0112525"	"-0.0375151"	"0.0684176"
[656]	"-0.00333729"	"-0.0485283"	"0.112122"	"-0.0254754"	"-0.0420037"
[661]	"0.117305"	"-0.00606851"	"-0.0283436"	"0.134787"	"0.0177798"
[666]	"-0.0195284"	"0.0892367"	"-0.0414485"	"-0.0486363"	"0.154688"
[671]	"0.0199821"	"-0.00463761"	"0.094692"	"-0.0385498"	"-0.0498494"
[676]	"0.107747"	"-0.0440315"	"-0.0518639"	"0.114158"	"-5.7712e-05"
[681]	"-0.0447193"	"0.0749854"	"-0.0298321"	"-0.0160654"	"0.149517"
[686]	"-0.00102999"	"-0.00897201"	"0.0967314"	"-0.00996676"	"-0.0279194"
[691]	"0.110555"	"0.00425955"	"-0.0461396"	"0.0824248"	"-0.0138764"

[696]	"-0.0399555"	"0.131041"	"-0.0533527"	"-0.0523496"	"0.132381"
[701]	"-0.00847989"	"-0.0339069"	"0.123183"	"0.0180261"	"-0.0153281"
[706]	"0.140781"	"-0.031946"	"-0.0263124"	"0.172881"	"-0.0218296"
[711]	"-0.022929"	"0.155953"	"0.0161524"	"0.0034478"	"0.13212"
[716]	"-0.030182"	"-0.0465635"	"0.0879679"	"0.0232893"	"-0.0219862"
[721]	"0.101942"	"-0.0286605"	"-0.0328508"	"0.110715"	"-0.0216954"
[726]	"-0.0161832"	"0.0939071"	"-0.0120504"	"-0.0335866"	"0.0785833"
[731]	"0.0172422"	"-0.0306775"	"0.174188"	"0.00767122"	"-0.0168081"
[736]	"0.128617"	"0.00409679"	"-0.0203337"	"0.14788"	"0.0141352"
[741]	"-0.0310049"	"0.0920424"	"-0.0317739"	"-0.0508649"	"0.113691"
[746]	"-0.0336232"	"-0.0435206"	"0.0908226"	"0.0141227"	"-0.0356563"
[751]	"0.151611"	"-0.0251462"	"-0.0392605"	"0.148354"	"0.0193436"
[756]	"-0.00507787"	"0.0745763"	"-0.0413064"	"-0.0343446"	"0.137556"
[761]	"-0.015931"	"-0.0358218"	"0.155282"	"-0.0081873"	"-0.0175252"
[766]	"0.0753223"	"0.00919663"	"-0.0285469"	"0.0838528"	"-0.0245324"
[771]	"-0.0456321"	"0.101281"	"-0.0577684"	"-0.0578549"	"0.132731"
[776]	"0.0302677"	"0.00794933"	"0.124703"	"0.00780066"	"-0.0108088"
[781]	"0.134473"	"-0.00920211"	"-0.0361238"	"0.1327"	"-0.0686589"
[786]	"-0.0527107"	"0.0797545"	"0.000304411"	"-0.0260952"	"0.0853618"
[791]	"-0.0388245"	"-0.0393543"	"0.0781255"	"-0.00640155"	"-0.0413997"
[796]	"0.111592"	"-0.00343729"	"-0.0497574"	"0.0966432"	"-0.00642743"
[801]	"-0.0481817"	"0.118091"	"-0.0344291"	"-0.0376505"	"0.0514939"
[806]	"-0.0330126"	"-0.0403231"	"0.0988183"	"-0.00400863"	"-0.0478751"
[811]	"0.0856974"	"-0.0150332"	"-0.0376421"	"0.0835704"	"0.00803595"
[816]	"-0.0297944"	"0.160558"	"-0.0369876"	"-0.0056921"	"0.138061"
[821]	"0.014991"	"-0.0181043"	"0.110175"	"0.0151729"	"-0.0232506"
[826]	"0.0904678"	"-0.0184155"	"-0.0314285"	"0.122581"	"-0.000255755"
[831]	"-0.0403649"	"0.114469"	"-0.0228913"	"-0.0419759"	"0.0795526"
[836]	"-0.0248807"	"-0.0350207"	"0.0960592"	"-0.0109763"	"-0.0484649"
[841]	"0.105196"	"-0.0459898"	"-0.0436156"	"0.104471"	"-0.00655441"
[846]	"-0.0505706"	"0.154361"	"0.0160661"	"-0.0085245"	"0.123968"
[851]	"-0.0328788"	"-0.0486597"	"0.127745"	"-0.0355985"	"-0.0396566"
[856]	"0.0855845"	"-0.0460008"	"-0.068768"	"0.143131"	"-0.0150595"
[861]	"-0.0450452"	"0.144031"	"-0.0257264"	"-0.0512494"	"0.0906441"

```
[866] "-0.0559069" "-0.0388699" "0.0754001" "-0.0187794" "-0.0305177"
[871] "0.148891" "-0.000973053" "-0.0249487" "0.140169" "-0.0317729"
[876] "-0.0439059" "0.0988978" "-0.0247639" "-0.0487974" "0.130069"
[881] "-0.0271039" "-0.0350573" "0.15454" "0.0134436" "-0.0141455"
[886] "0.14081" "-0.0311889" "-0.0621757" "0.103084" "-0.0068436"
[891] "-0.0129621" "0.107912" "-0.0140241" "-0.0316433" "0.188295"
[896] "0.0206126" "-0.00150471" "0.0996349" "0.00831133" "-0.0446597"
```

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	10.2175000
5	1	THETA2	24.3171000
8	1	THETA3	0.0689622
11	1	THETA4	3.7575900
14	1	THETA5	180.9270000
17	1	THETA6	0.9726310

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	10.2175000
2	1	THETA2	24.3171000
3	1	THETA3	0.0689622
4	1	THETA4	3.7575900
5	1	THETA5	180.9270000
6	1	THETA6	0.9726310

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	nm7	1005
2	V_c /F (L) ~ theta_2 * (WT/70)^1	nm7	1005
3	K_a (h^-1) ~ theta_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	10.2175000	CL/F
2	1	THETA2	24.3171000	V_c/F
3	1	THETA3	0.0689622	K_a
4	1	THETA4	3.7575900	Q/F
5	1	THETA5	180.9270000	V_p/F
6	1	THETA6	0.9726310	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		0
2	0	29.1	1000	1	0	0	83.5		0
3	0	29.1	1000	1	0	0	83.5		0
4	0	29.1	1000	1	0	0	83.5		0
5	0	29.1	1000	1	0	0	83.5		0
6	0	29.1	1000	1	0	0	83.5		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	10.2175000	CL/F
2	1	THETA2	24.3171000	V _c /F
3	1	THETA3	0.0689622	K _a
4	1	THETA4	3.7575900	Q/F
5	1	THETA5	180.9270000	V _p /F
6	1	THETA6	0.9726310	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	10.217500	CL/F
6	1	THETA6	0.972631	MALE_CL/F
7	1	THETA7	1.441780	WT_CL/F
16	2	THETA1	8.655780	CL/F
21	2	THETA6	1.106340	MALE_CL/F
22	2	THETA7	0.944065	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	10.21750	0.972631	1.441780
2	2	8.65578	1.106340	0.944065
3	3	8.54538	1.084290	0.984631
4	4	9.74759	0.870490	1.522740
5	5	9.27924	1.060020	1.185630
6	6	8.68753	1.039070	1.179110

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	10.21750	0.972631	1.441780	0.8986628	1.104588	1.323038
2	2	8.65578	1.106340	0.944065	0.9324286	1.067302	1.201170
3	3	8.54538	1.084290	0.984631	0.9296296	1.070293	1.210668
4	4	9.74759	0.870490	1.522740	0.8932872	1.110775	1.343999
5	5	9.27924	1.060020	1.185630	0.9158848	1.085239	1.258848
6	6	8.68753	1.039070	1.179110	0.9163275	1.084751	1.257256

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	1.0746734	0.972631	1.441780	0.8986628	1.104588	1.323038
2	2	0.9104122	1.106340	0.944065	0.9324286	1.067302	1.201170
3	3	0.8988003	1.084290	0.984631	0.9296296	1.070293	1.210668
4	4	1.0252484	0.870490	1.522740	0.8932872	1.110775	1.343999
5	5	0.9759875	1.060020	1.185630	0.9158848	1.085239	1.258848
6	6	0.9137516	1.039070	1.179110	0.9163275	1.084751	1.257256

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	1.0746734
2	2	CL/F	0.9104122
3	3	CL/F	0.8988003
4	4	CL/F	1.0252484
5	5	CL/F	0.9759875
6	6	CL/F	0.9137516

1.4.4 Plot.

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

Listing 36:

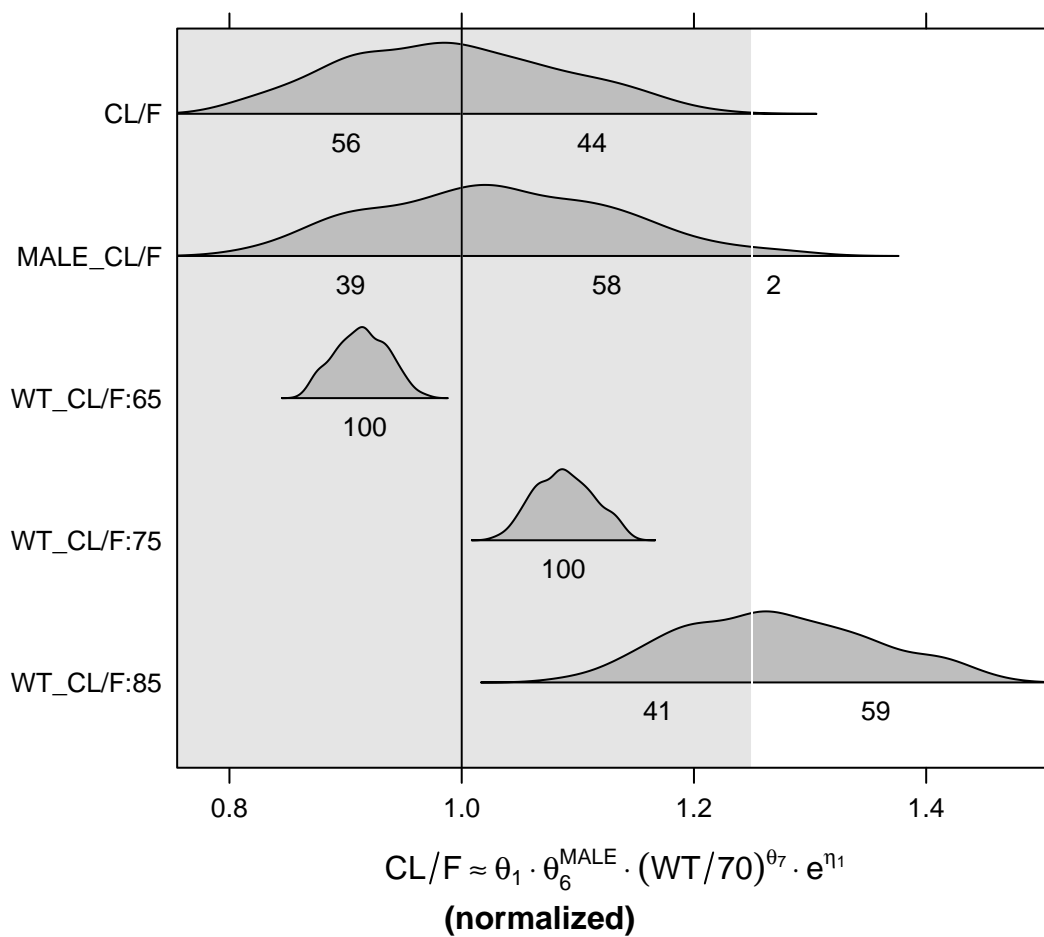
```
> levels(molten$variable)

[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



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.