Parameter Table

April 3, 2012

Tim Bergsma

1 Purpose

This script picks up after model.Rnw to process bootstrap results and make a parameter table. It assumes the current working directory is the script directory containing this file.

1.1 Package

Listing 1:

```
> library(metrumrg)
```

2 inputs

'wikitab' gives us a quick synthesis of 'rlog' and the 'lookup' of wiki notation in 1005.ctl. We do some science on the result first, and then some aesthetics for printing in a LaTeXtable. Table 1.

Listing 2:

```
> tab <- wikitab(1005,'../nonmem')
> tab$estimate <- signif(as.numeric(tab$estimate),3)
> tab$tool <- NULL
> tab$run <- NULL
> tab$se <- NULL
> tab
```

```
  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	estimate
1	CL/F (L/h)	~	theta_1 *	theta_6 ^MALE * (WT/70)^theta_7	*	e^eta_1		9.5100
2				V_c /F (L) ~ theta_2 * (WT/70)^1	*	e^eta_2		22.8000
3				K_a (h^-1) ~ theta_3	*	e^eta_3		0.0714
4				Q/F (L/h) ~ theta_4				3.4700
5				V_p /F (L) ~ theta_5				113.0000
6				MALE_CL/F ~ theta_6				1.0200
7				WT_CL/F ~ theta_7				1.1900
8				IIV_CL/F ~ Omega_1.1				0.2140
9				cov_CL,V ~ Omega_2.1				0.1210
10				IIV_V_c /F ~ Omega_2.2				0.0945
11				cov_CL,Ka ~ Omega_3.1				-0.0116
12				cov_V,Ka ~ Omega_3.2				-0.0372
13				IIV_K_a ~ Omega_3.3				0.0466
14				err_prop ~ Sigma_1.1				0.0492
15				err_add ~ Sigma_2.2				0.2020

	prse
1	9.84
2	9.56
3	7.35
4	15.4
5	21
6	11.2
7	28.4

```
8  22.8
9  26.4
10 33.2
11  173
12 36.1
13 34.7
14 10.9
15 33.5
```

Now we can extract some information from the model statements.

Listing 3:

```
> tab$units <- justUnits(tab$model)
> tab$model <- noUnits(tab$model)
> tab$name <- with(tab, wiki2label(model))
> tab[c('model','units','name')]
```

```
                                                          model units
1  CL/F  ~ theta_1 *  theta_6 ^MALE * (WT/70)^theta_7  * e^eta_1   L/h
2                       V_c /F  ~ theta_2 * (WT/70)^1 * e^eta_2     L
3                                  K_a  ~ theta_3 * e^eta_3 h^-1
4                                            Q/F  ~ theta_4   L/h
5                                          V_p /F  ~ theta_5     L
6                                       MALE_CL/F ~ theta_6
7                                         WT_CL/F ~ theta_7
8                                       IIV_CL/F ~ Omega_1.1
9                                       cov_CL,V ~ Omega_2.1
10                                    IIV_V_c /F ~ Omega_2.2
11                                    cov_CL,Ka  ~ Omega_3.1
12                                     cov_V,Ka  ~ Omega_3.2
13                                      IIV_K_a  ~ Omega_3.3
14                                      err_prop ~ Sigma_1.1
15                                       err_add ~ Sigma_2.2
        name
```

```
1       CL/F
2      V_c/F
3        K_a
4        Q/F
5      V_p/F
6  MALE_CL/F
7    WT_CL/F
8   IIV_CL/F
9   cov_CL,V
10 IIV_V_c/F
11 cov_CL,Ka
12  cov_V,Ka
13   IIV_K_a
14  err_prop
15   err_add
```

3 variance

The estimates for the matrix diagonals are variances, and their square roots have special meaning. In model 1005, interindividual variability was modelled exponentially, in which case square root of variance gives an approximate CV; alternatively, and exact CV can be calculated. For proportional error terms like ERR1, square root gives an exact CV. For additive error terms like ERR2, square root gives standard deviation.

We can use functions of 'parameter' to sort out the various error components, as they are used in this model.

3.1 exponential

Listing 4:

```
> expo <- is.iiv(tab$parameter) & is.diagonal(tab$parameter)
> tab$parameter[expo]
```

```
[1] "OMEGA1.1" "OMEGA2.2" "OMEGA3.3"
```

Listing 5:

```
> tab$cv[expo] <- cvLognormal(tab$estimate[expo])
> tab[,c('parameter','name','estimate','cv')]
```

	parameter	name	estimate	cv
1	THETA1	CL/F	9.5100	NA
2	THETA2	V_c/F	22.8000	NA
3	THETA3	K_a	0.0714	NA
4	THETA4	Q/F	3.4700	NA
5	THETA5	V_p/F	113.0000	NA
6	THETA6	MALE_CL/F	1.0200	NA
7	THETA7	WT_CL/F	1.1900	NA
8	OMEGA1.1	IIV_CL/F	0.2140	0.4884902
9	OMEGA2.1	cov_CL,V	0.1210	NA
10	OMEGA2.2	IIV_V_c/F	0.0945	0.3148161
11	OMEGA3.1	cov_CL,Ka	-0.0116	NA
12	OMEGA3.2	cov_V,Ka	-0.0372	NA
13	OMEGA3.3	IIV_K_a	0.0466	0.2184098
14	SIGMA1.1	err_prop	0.0492	NA
15	SIGMA2.2	err_add	0.2020	NA

3.2 proportional

Listing 6:

```
> writeLines(read.nmctl('../nonmem/ctl/1005.ctl')$err)
```

```
 Y=F*(1+ERR(1)) + ERR(2)
 IPRE=F
;<doc>
```

Listing 7:

```
> prop <- is.random(tab$parameter) & tab$name %contains% 'prop'
> tab$parameter[prop]
```

```
[1] "SIGMA1.1"
```

Listing 8:

```
> tab$cv[prop] <- sqrt(tab$estimate[prop])
> tab[,c('parameter','name','estimate','cv')]
```

```
   parameter      name estimate        cv
1     THETA1      CL/F   9.5100        NA
2     THETA2     V_c/F  22.8000        NA
3     THETA3       K_a   0.0714        NA
4     THETA4       Q/F   3.4700        NA
5     THETA5     V_p/F 113.0000        NA
6     THETA6 MALE_CL/F   1.0200        NA
7     THETA7   WT_CL/F   1.1900        NA
8   OMEGA1.1  IIV_CL/F   0.2140 0.4884902
9   OMEGA2.1  cov_CL,V   0.1210        NA
10  OMEGA2.2 IIV_V_c/F   0.0945 0.3148161
11  OMEGA3.1 cov_CL,Ka  -0.0116        NA
12  OMEGA3.2  cov_V,Ka  -0.0372        NA
13  OMEGA3.3   IIV_K_a   0.0466 0.2184098
14  SIGMA1.1  err_prop   0.0492 0.2218107
15  SIGMA2.2   err_add   0.2020        NA
```

3.3 additive

Listing 9:

```
> add <- is.residual(tab$parameter) & tab$name %contains% 'add'
> tab$parameter[add]
```

```
[1] "SIGMA2.2"
```

Listing 10:

```
> tab$sd[add] <- sqrt(tab$estimate[add])
> tab[,c('parameter','name','estimate','cv','sd')]
```

```
   parameter      name estimate        cv        sd
1     THETA1      CL/F   9.5100        NA        NA
2     THETA2     V_c/F  22.8000        NA        NA
3     THETA3       K_a   0.0714        NA        NA
4     THETA4       Q/F   3.4700        NA        NA
5     THETA5     V_p/F 113.0000        NA        NA
6     THETA6 MALE_CL/F   1.0200        NA        NA
7     THETA7   WT_CL/F   1.1900        NA        NA
8   OMEGA1.1  IIV_CL/F   0.2140 0.4884902        NA
9   OMEGA2.1  cov_CL,V   0.1210        NA        NA
10  OMEGA2.2 IIV_V_c/F   0.0945 0.3148161        NA
11  OMEGA3.1 cov_CL,Ka  -0.0116        NA        NA
12  OMEGA3.2  cov_V,Ka  -0.0372        NA        NA
13  OMEGA3.3   IIV_K_a   0.0466 0.2184098        NA
14  SIGMA1.1  err_prop   0.0492 0.2218107        NA
15  SIGMA2.2   err_add   0.2020        NA 0.4494441
```

4 covariance

The estimates of matrix off-diagonals are covariances, and are more useful if transformed to correlations. We could extract the matrices manually, or use package shortcuts.

Listing 11:

```
> cor <- omegacor(run=1005,project='../nonmem')
> cor
```

```
           [,1]       [,2]       [,3]
[1,]  1.0000000  0.8492811 -0.1163229
[2,]  0.8492811  1.0000000 -0.5607054
[3,] -0.1163229 -0.5607054  1.0000000
```

Listing 12:

```
> half(cor)
```

```
       1.1        2.1        2.2        3.1        3.2        3.3
 1.0000000  0.8492811  1.0000000 -0.1163229 -0.5607054  1.0000000
```

Listing 13:

```
> offdiag(half(cor))
```

```
       2.1        3.1        3.2
 0.8492811 -0.1163229 -0.5607054
```

Listing 14:

```
> off <- is.iiv(tab$parameter) & is.offdiagonal(tab$parameter)
> tab$parameter[off]
```

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

Listing 15:

```
> tab$cor[off] <- offdiag(half(cor))
> tab[,c('parameter','name','estimate','cv','sd','cor')]
```

```
   parameter      name estimate        cv        sd        cor
1     THETA1      CL/F   9.5100        NA        NA         NA
2     THETA2     V_c/F  22.8000        NA        NA         NA
3     THETA3       K_a   0.0714        NA        NA         NA
```

```
4     THETA4       Q/F   3.4700        NA        NA         NA
5     THETA5     V_p/F 113.0000        NA        NA         NA
6     THETA6 MALE_CL/F   1.0200        NA        NA         NA
7     THETA7    WT_CL/F   1.1900        NA        NA         NA
8   OMEGA1.1   IIV_CL/F   0.2140 0.4884902        NA         NA
9   OMEGA2.1   cov_CL,V   0.1210        NA        NA  0.8492811
10  OMEGA2.2 IIV_V_c/F   0.0945 0.3148161        NA         NA
11  OMEGA3.1 cov_CL,Ka  -0.0116        NA        NA -0.1163229
12  OMEGA3.2  cov_V,Ka  -0.0372        NA        NA -0.5607054
13  OMEGA3.3    IIV_K_a   0.0466 0.2184098        NA         NA
14  SIGMA1.1  err_prop   0.0492 0.2218107        NA         NA
15  SIGMA2.2   err_add   0.2020        NA 0.4494441         NA
```

5 confidence interval

We wish to include 95 percentiles in our table as confidence intervals.

Listing 16:

```
> 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 17:

```
> boot <- boot[boot$moment=='estimate',]
```

```
> boot <- data.frame(cast(boot,... ~ moment))
> head(boot)
```

```
   X tool run parameter  estimate
1  2  nm7   1    THETA1   8.64914
2  5  nm7   1    THETA2   21.5594
3  8  nm7   1    THETA3 0.0766718
4 11  nm7   1    THETA4   4.05759
5 14  nm7   1    THETA5   106.257
6 17  nm7   1    THETA6   1.07562
```

Listing 18:

```
> boot <- boot[,c('run','parameter','estimate')]
> sapply(boot,class)
```

```
        run   parameter    estimate
  "integer" "character"    "factor"
```

Listing 19:

```
> boot$estimate <- as.numeric(as.character(boot$estimate))
> 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" "SIGMA2.1" "SIGMA2.2"
```

Listing 20:

```
> quan <- function(x,probs)as.character(signif(quantile(x,probs=probs,na.rm=TRUE),3))
> boot$lo <- with(boot, reapply(estimate,parameter,quan,probs=.05))
> boot$hi <- with(boot, reapply(estimate,parameter,quan,probs=.95))
> head(boot)
```

```
  run parameter    estimate     lo     hi
1   1    THETA1   8.6491400   6.19     11
2   1    THETA2  21.5594000   19.1   26.4
3   1    THETA3   0.0766718 0.0607 0.0801
4   1    THETA4   4.0575900   2.82   5.06
5   1    THETA5 106.2570000   87.9    930
6   1    THETA6   1.0756200   0.86   1.29
```

Listing 21:

```
> boot <- unique(boot[,c('parameter','lo','hi')])
> boot
```

```
   parameter      lo       hi
1     THETA1    6.19       11
2     THETA2    19.1     26.4
3     THETA3  0.0607   0.0801
4     THETA4    2.82     5.06
5     THETA5    87.9      930
6     THETA6    0.86     1.29
7     THETA7   0.571     1.87
8   OMEGA1.1   0.132    0.431
9   OMEGA2.1  0.0663    0.222
10  OMEGA2.2  0.0511    0.162
11  OMEGA3.1  -0.046   0.0242
12  OMEGA3.2 -0.0565 -0.00863
13  OMEGA3.3  0.0231   0.0738
14  SIGMA1.1  0.0403   0.0592
15  SIGMA2.1       0        0
16  SIGMA2.2  0.0645    0.304
```

Listing 22:

```
> boot$ci <- with(boot, parens(glue(lo,',',hi)))
> boot
```

	parameter	lo	hi	ci
1	THETA1	6.19	11	(6.19,11)
2	THETA2	19.1	26.4	(19.1,26.4)
3	THETA3	0.0607	0.0801	(0.0607,0.0801)
4	THETA4	2.82	5.06	(2.82,5.06)
5	THETA5	87.9	930	(87.9,930)
6	THETA6	0.86	1.29	(0.86,1.29)
7	THETA7	0.571	1.87	(0.571,1.87)
8	OMEGA1.1	0.132	0.431	(0.132,0.431)
9	OMEGA2.1	0.0663	0.222	(0.0663,0.222)
10	OMEGA2.2	0.0511	0.162	(0.0511,0.162)
11	OMEGA3.1	-0.046	0.0242	(-0.046,0.0242)
12	OMEGA3.2	-0.0565	-0.00863	(-0.0565,-0.00863)
13	OMEGA3.3	0.0231	0.0738	(0.0231,0.0738)
14	SIGMA1.1	0.0403	0.0592	(0.0403,0.0592)
15	SIGMA2.1	0	0	(0,0)
16	SIGMA2.2	0.0645	0.304	(0.0645,0.304)

Listing 23:

```
> tab <- stableMerge(tab,boot[,c('parameter','ci')])
> tab
```

	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 estimate prse
1  CL/F  ~ theta_1 *  theta_6 ^MALE * (WT/70)^theta_7  * e^eta_1   9.5100 9.84
2                        V_c /F  ~ theta_2 * (WT/70)^1 * e^eta_2  22.8000 9.56
3                                    K_a  ~ theta_3 * e^eta_3   0.0714 7.35
4                                              Q/F  ~ theta_4   3.4700 15.4
5                                            V_p /F  ~ theta_5 113.0000   21
6                                         MALE_CL/F ~ theta_6   1.0200 11.2
7                                           WT_CL/F ~ theta_7   1.1900 28.4
8                                         IIV_CL/F ~ Omega_1.1   0.2140 22.8
9                                         cov_CL,V ~ Omega_2.1   0.1210 26.4
10                                      IIV_V_c /F ~ Omega_2.2   0.0945 33.2
11                                      cov_CL,Ka  ~ Omega_3.1  -0.0116  173
12                                       cov_V,Ka  ~ Omega_3.2  -0.0372 36.1
13                                        IIV_K_a  ~ Omega_3.3   0.0466 34.7
14                                        err_prop ~ Sigma_1.1   0.0492 10.9
15                                         err_add ~ Sigma_2.2   0.2020 33.5
   units      name        cv        sd        cor                ci
1    L/h      CL/F        NA        NA         NA          (6.19,11)
2      L     V_c/F        NA        NA         NA        (19.1,26.4)
3  h^-1       K_a        NA        NA         NA    (0.0607,0.0801)
4    L/h       Q/F        NA        NA         NA        (2.82,5.06)
5      L     V_p/F        NA        NA         NA         (87.9,930)
6        MALE_CL/F        NA        NA         NA        (0.86,1.29)
7          WT_CL/F        NA        NA         NA       (0.571,1.87)
8         IIV_CL/F 0.4884902        NA         NA      (0.132,0.431)
9         cov_CL,V        NA        NA  0.8492811     (0.0663,0.222)
10       IIV_V_c/F 0.3148161        NA         NA     (0.0511,0.162)
11       cov_CL,Ka        NA        NA -0.1163229    (-0.046,0.0242)
12        cov_V,Ka        NA        NA -0.5607054 (-0.0565,-0.00863)
13         IIV_K_a 0.2184098        NA         NA    (0.0231,0.0738)
```

```
14        err_prop 0.2218107        NA         NA    (0.0403,0.0592)
15         err_add        NA 0.4494441         NA     (0.0645,0.304)
```

6 aesthetics

Here we format the table for printing.

Listing 24:

```
> tab$name <- NULL
> tab$parameter <- NULL
> tab$model <- wiki2latex(tab$model)
> tab$estimate <- as.character(tab$estimate)
> tab$estimate <- paste(tab$estimate,'$', tab$units,'$')
> tab$units <- NULL
```

Note that no parameter defines more than one of CV, SD,and COR. We could collapse these into a single column, and add a descriptive flag.

Listing 25:

```
> m <- as.matrix(tab[,c('cv','sd','cor')])
> tab$variability <- suppressWarnings(apply(m,1,max,na.rm=TRUE))
> tab$variability[is.infinite(tab$variability)] <- NA
> i <- !is.na(m)
> i[!i] <- NA
> tab$statistic <- apply(i,1,function(x){
+   p <- colnames(i)[x]
+   ifelse(all(is.na(p)),NA,p[!is.na(p)])
+ })
> toPercent <- with(tab, !is.na(statistic) & statistic=='cv')
> tab$variability[toPercent] <- percent(tab$variability[toPercent])
> tab$variability <- as.character(signif(tab$variability,3))
> tab$statistic <- map(tab$statistic,from=c(NA,'cv','cor','sd'),to=c(NA,'\\%CV','CORR','SD'))
```

```
> tab$variability <- paste(tab$statistic,tab$variability,sep=' = ')
> tab$variability[is.na(tab$statistic)] <- NA
> tab$statistic <- NULL
> tab$cv <- NULL
> tab$sd <- NULL
> tab$cor <- NULL
```

Table 1: Parameter Estimates from Population Pharmacokinetic Model Run 1005

description	model	estimate	prse	ci	variability
apparent oral clearance	$\mathrm{CL/F} \sim \theta_1 \cdot \theta_6^{\mathrm{MALE}} \cdot (\mathrm{WT}/70)^{\theta_7} \cdot \mathrm{e}^{\eta_1}$	9.51 L/h	9.84	(6.19,11)	
central volume of distribution	$\mathrm{V_c/F} \sim \theta_2 \cdot (\mathrm{WT}/70)^1 \cdot \mathrm{e}^{\eta_2}$	22.8 L	9.56	(19.1,26.4)	
absorption rate constant	$\mathrm{K_a} \sim \theta_3 \cdot \mathrm{e}^{\eta_3}$	0.0714 h^-1	7.35	(0.0607,0.0801)	
intercompartmental clearance	$\mathrm{Q/F} \sim \theta_4$	3.47 L/h	15.4	(2.82,5.06)	
peripheral volume of distribution	$\mathrm{V_p/F} \sim \theta_5$	113 L	21	(87.9,930)	
male effect on clearance	$\mathrm{MALE_{CL/F}} \sim \theta_6$	1.02	11.2	(0.86,1.29)	
weight effect on clearance	$\mathrm{WT_{CL/F}} \sim \theta_7$	1.19	28.4	(0.571,1.87)	
interindividual variability of clearance	$\mathrm{IIV_{CL/F}} \sim \Omega_{1.1}$	0.214	22.8	(0.132,0.431)	%CV = 48.8
interindividual clearance-volume covariance	$\mathrm{cov_{CL,V}} \sim \Omega_{2.1}$	0.121	26.4	(0.0663,0.222)	CORR = 0.849
interindividual variability of central volume	$\mathrm{IIV_{V_c/F}} \sim \Omega_{2.2}$	0.0945	33.2	(0.0511,0.162)	%CV = 31.5
interindividual clearance-Ka covariance	$\mathrm{cov_{CL,Ka}} \sim \Omega_{3.1}$	-0.0116	173	(-0.046,0.0242)	CORR = -0.116
interindividual volume-Ka covariance	$\mathrm{cov_{V,Ka}} \sim \Omega_{3.2}$	-0.0372	36.1	(-0.0565,-0.00863)	CORR = -0.561
interindividual variability of Ka	$\mathrm{IIV_{K_a}} \sim \Omega_{3.3}$	0.0466	34.7	(0.0231,0.0738)	%CV = 21.8
proportional error	$\mathrm{err_{prop}} \sim \Sigma_{1.1}$	0.0492	10.9	(0.0403,0.0592)	%CV = 22.2
additive error	$\mathrm{err_{add}} \sim \Sigma_{2.2}$	0.202	33.5	(0.0645,0.304)	SD = 0.449

7 simple parameter table

We can make a quick parameter table that does not use wikitab markup. Table 2.

Listing 26:

```
> tab <- rlog(1005,'../nonmem',tool='nm7',file=NULL)
> head(tab)
```

```
  tool  run parameter   moment            value
1  nm7 1005       ofv  minimum 2405.91625845151
2  nm7 1005    THETA1 estimate          9.50754
3  nm7 1005    THETA1     prse             9.84
4  nm7 1005    THETA1       se         0.935942
5  nm7 1005    THETA2 estimate          22.7907
6  nm7 1005    THETA2     prse             9.56
```

Listing 27:

```
> tab$tool <- NULL
> tab$run <- NULL
> tab <- tab[tab$moment %in% c('estimate','prse'),]
> unique(tab$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" "SIGMA2.1" "SIGMA2.2"
```

Listing 28:

```
> tab$value <- signif(as.numeric(tab$value),3)
> tab$parameter <- factor(tab$parameter,levels=unique(tab$parameter))#to preserve row order during cast
> tab <- cast(tab,parameter ~ moment)
> tab
```

```
   parameter estimate   prse
1     THETA1   9.5100   9.84
2     THETA2  22.8000   9.56
3     THETA3   0.0714   7.35
4     THETA4   3.4700  15.40
5     THETA5 113.0000  21.00
6     THETA6   1.0200  11.20
7     THETA7   1.1900  28.40
8   OMEGA1.1   0.2140  22.80
9   OMEGA2.1   0.1210  26.40
10  OMEGA2.2   0.0945  33.20
11  OMEGA3.1  -0.0116 173.00
12  OMEGA3.2  -0.0372  36.10
13  OMEGA3.3   0.0466  34.70
14  SIGMA1.1   0.0492  10.90
15  SIGMA2.1   0.0000    Inf
16  SIGMA2.2   0.2020  33.50
```

Listing 29:

```
> tab$parameter <- parameter2wiki(tab$parameter)
> tab
```

```
    parameter estimate   prse
1    theta_1    9.5100   9.84
2    theta_2   22.8000   9.56
3    theta_3    0.0714   7.35
4    theta_4    3.4700  15.40
5    theta_5  113.0000  21.00
6    theta_6    1.0200  11.20
7    theta_7    1.1900  28.40
8  Omega_1.1    0.2140  22.80
9  Omega_2.1    0.1210  26.40
10 Omega_2.2    0.0945  33.20
11 Omega_3.1   -0.0116 173.00
```

```
12 Omega_3.2   -0.0372  36.10
13 Omega_3.3    0.0466  34.70
14 Sigma_1.1    0.0492  10.90
15 Sigma_2.1    0.0000    Inf
16 Sigma_2.2    0.2020  33.50
```

Listing 30:

```
> tab$parameter <- wiki2latex(tab$parameter)
> tab
```

```
                  parameter estimate   prse
1    $\\mathrm{\\theta_{1}}$   9.5100   9.84
2    $\\mathrm{\\theta_{2}}$  22.8000   9.56
3    $\\mathrm{\\theta_{3}}$   0.0714   7.35
4    $\\mathrm{\\theta_{4}}$   3.4700  15.40
5    $\\mathrm{\\theta_{5}}$ 113.0000  21.00
6    $\\mathrm{\\theta_{6}}$   1.0200  11.20
7    $\\mathrm{\\theta_{7}}$   1.1900  28.40
8  $\\mathrm{\\Omega_{1.1}}$   0.2140  22.80
9  $\\mathrm{\\Omega_{2.1}}$   0.1210  26.40
10 $\\mathrm{\\Omega_{2.2}}$   0.0945  33.20
11 $\\mathrm{\\Omega_{3.1}}$  -0.0116 173.00
12 $\\mathrm{\\Omega_{3.2}}$  -0.0372  36.10
13 $\\mathrm{\\Omega_{3.3}}$   0.0466  34.70
14 $\\mathrm{\\Sigma_{1.1}}$   0.0492  10.90
15 $\\mathrm{\\Sigma_{2.1}}$   0.0000    Inf
16 $\\mathrm{\\Sigma_{2.2}}$   0.2020  33.50
```

Table 2: Simple Parameter Table

parameter	estimate	prse
θ_1	9.5100	9.84
θ_2	22.8000	9.56
θ_3	0.0714	7.35
θ_4	3.4700	15.40
θ_5	113.0000	21.00
θ_6	1.0200	11.20
θ_7	1.1900	28.40
$\Omega_{1,1}$	0.2140	22.80
$\Omega_{2,1}$	0.1210	26.40
$\Omega_{2,2}$	0.0945	33.20
$\Omega_{3,1}$	-0.0116	173.00
$\Omega_{3,2}$	-0.0372	36.10
$\Omega_{3,3}$	0.0466	34.70
$\Sigma_{1,1}$	0.0492	10.90
$\Sigma_{2,1}$	0.0000	Inf
$\Sigma_{2,2}$	0.2020	33.50