# Simulating with Parameter Uncertainty

March 20, 2013

Bill Knebel
Tim Bergsma

# 1 Purpose

This script shows how to conduct a simulation that considers uncertainty in the parameter estimates.

# 2 Data

Here we load metrumrg and read in the data to be used for simulations.

Listing 1:

```
> library(metrumrg)
> data <- read.csv("../data/derived/phase1.csv")
> head(data)
```

```
  C ID TIME SEQ EVID  AMT    DV SUBJ HOUR TAFD  TAD LDOS MDV HEIGHT WEIGHT SEX
1 C  1 0.00   0    0    .     0    1 0.00 0.00    .    .   0    174   74.2   0
2 .  1 0.00   1    1 1000     .    1 0.00 0.00    0 1000   1    174   74.2   0
3 .  1 0.25   0    0    . 0.363    1 0.25 0.25 0.25 1000   0    174   74.2   0
4 .  1 0.50   0    0    . 0.914    1 0.50 0.50  0.5 1000   0    174   74.2   0
5 .  1 1.00   0    0    .  1.12    1 1.00 1.00    1 1000   0    174   74.2   0
6 .  1 2.00   0    0    .  2.28    1 2.00 2.00    2 1000   0    174   74.2   0
   AGE DOSE FED SMK DS CRCN predose zerodv
1 29.1 1000   1   0  0 83.5       1      0
2 29.1 1000   1   0  0 83.5       0      0
3 29.1 1000   1   0  0 83.5       0      0
4 29.1 1000   1   0  0 83.5       0      0
5 29.1 1000   1   0  0 83.5       0      0
6 29.1 1000   1   0  0 83.5       0      0
```

We use NONMEM output from a simple two compartment model to generate parameters. We use 1005.lst and 1005.cov output from NM7 to populate a call to metrumrg::simpar().

Listing 2:

```
> cov <- read.table("../nonmem/1005/1005.cov", skip=1, header=T)
> head(cov)
```

```
    NAME      THETA1      THETA2       THETA3      THETA4      THETA5
1 THETA1  0.85349000  0.78471700  1.02964e-03  0.06201550  -1.2885700
2 THETA2  0.78471700  4.74387000  6.65868e-03  0.89539600   5.5877600
3 THETA3  0.00102964  0.00665868  2.75169e-05  0.00221641  -0.0298637
4 THETA4  0.06201550  0.89539600  2.21641e-03  0.28656000   0.2410890
5 THETA5 -1.28857000  5.58776000 -2.98637e-02  0.24108900 559.0090000
6 THETA6 -0.03952260 -0.02453050 -1.02177e-04 -0.01047580   0.7350690
        THETA6       THETA7   SIGMA.1.1. SIGMA.2.1.   SIGMA.2.2.  OMEGA.1.1.
1 -0.039522600 -0.176224000 -4.89162e-04          0  2.04096e-02 6.28811e-03
2 -0.024530500  0.068529700 -3.11007e-03          0  1.89401e-02 5.84996e-03
3 -0.000102177 -0.000132916 -1.02493e-05          0  5.86438e-05 3.24081e-06
4 -0.010475800  0.015606300 -6.27671e-04          0  2.50369e-03 4.31368e-03
```

```
5  0.735069000 -0.684622000  4.52242e-02          0 -4.20659e-01 2.73881e-01
6  0.012748500  0.000415439  1.17741e-04          0 -1.03450e-03 1.63668e-03
    OMEGA.2.1.   OMEGA.2.2.   OMEGA.3.1.   OMEGA.3.2.   OMEGA.3.3.
1 -1.59957e-04 -4.31064e-03 -5.37918e-03 -2.56445e-03 -3.38999e-03
2 -2.19085e-02 -2.43988e-02 -1.95676e-02 -1.11920e-02  4.75058e-03
3 -6.49265e-05 -7.78059e-05 -6.74428e-05 -2.74920e-05  2.82116e-05
4 -6.19519e-03 -7.76509e-03 -4.54515e-03 -2.24113e-03  3.06880e-03
5  1.59962e-01  2.51679e-02 -7.08665e-03  7.40212e-02 -3.34805e-02
6  2.98890e-04  5.89470e-04 -5.36299e-04 -5.60638e-05 -3.30708e-04
```

We are interested in theta covariance, so we remove extra columns and rows.

Listing 3:

```
> cov<- cov[1:7,c(2:8)]
```

# 3 Parameters

Now we generate 10 sets of population parameters based on the 1005.lst results.

Listing 4:

```
> set.seed(10)
> PKparms <- simpar(
+     nsim=10,
+     theta=c(8.58,21.6, 0.0684, 3.78, 107, 0.999, 1.67),
+     covar=cov,
+     omega=list(0.196, 0.129, 0.107),
+     odf=c(40,40,40),
+     sigma=list(0.0671),
+     sdf=c(200)
+ )
> PKparms
```

```
     TH.1  TH.2    TH.3  TH.4   TH.5   TH.6  TH.7  OM1.1   OM2.2   OM3.3
1   9.458 24.04 0.06312 3.509 106.50 1.0150 1.593 0.1847 0.15400 0.13630
2  10.720 22.98 0.06798 3.817 111.30 0.8489 1.301 0.2862 0.12000 0.16400
3   9.024 21.24 0.06630 3.969 139.40 1.0770 1.507 0.1647 0.12770 0.11300
4  10.670 23.61 0.07183 3.909 121.20 1.1500 1.013 0.1886 0.11460 0.08460
5  10.010 23.01 0.07001 3.573 100.00 0.9469 1.767 0.1526 0.08448 0.13140
6   8.952 21.73 0.06876 3.343  97.78 1.0610 1.809 0.2462 0.17640 0.08805
7   8.655 19.73 0.06614 3.380 135.60 0.9608 1.343 0.2221 0.14440 0.09957
8   9.214 21.24 0.06098 3.084 115.60 1.0420 1.753 0.2287 0.13820 0.06118
9   8.998 23.97 0.06985 4.114 145.40 0.9439 1.778 0.1765 0.12310 0.08504
10  8.938 22.73 0.06436 3.704 113.10 1.0610 1.582 0.2116 0.11940 0.09954
     SG1.1
1  0.06894
2  0.06099
3  0.06041
```

```
4  0.07700
5  0.06269
6  0.07274
7  0.06160
8  0.06692
9  0.06092
10 0.06269
```

# 4 Control Streams

We read in a control stream and clean out extra xml markup.

Listing 5:

```
> ctl <- as.nmctl(readLines("../nonmem/ctl/1005.ctl"))
> ctl[] <- lapply(ctl,function(rec)sub("<.*","",rec))
```

Now we iterate across the rows of PKparms, writing out a separate ctl for each.

Listing 6:

```
> dir.create('../nonmem/sim')
> set <- lapply(
+       rownames(PKparms),
+       function(row,params,ctl){
+               params <- as.character(PKparms[row,])
+               ctl$prob <- sub(1005,row,ctl$prob)
+               ctl$theta <- params[1:7]
+               ctl$omega <- params[8:10]
+               ctl$sigma <- params[11]
+               names(ctl)[names(ctl)=='estimation'] <- 'simulation'
+               ctl$simulation <- paste(
+                       '(',
+                       as.numeric(row) + 7995,
+                       'NEW) (',
+                       as.numeric(row) + 8996,
+                       'UNIFORM) ONLYSIMULATION'
+               )
+               ctl$cov <- NULL
+               ctl$table <- NULL
+               ctl$table <- NULL
+               ctl$table <- 'ID TIME DV WT SEX LDOS NOPRINT NOAPPEND FILE=sim.tab
    '
+               write.nmctl(ctl,file=file.path('../nonmem/sim',paste(sep='.',row,'
    ctl')))
+               return(ctl)
+       },
+       params=PKparms,
+       ctl=ctl
+ )
```

# 5 Simulation

Finally, we run NONMEM simulations using NONR.

Listing 7:

```
> NONR72(
+       run=1:10,
+       command="/opt/NONMEM/nm72/nmqual/autolog.pl",
+       project="../nonmem/sim",
+       diag=FALSE,
+       checkrunno=FALSE,
+       grid=TRUE
+ )
```