Script to calculate power by simulation for the Farewell and Herzberg example

Initialize

```
library(knitr)
#knitr::spin("FHpower.r")
library(dae)
packageVersion("dae")
```

```
## [1] '3.1.38'
```

```
library(asreml)
packageVersion("asreml")
```

```
## [1] '4.1.0.130'
```

```
library(asremlPlus)
packageVersion("asremlPlus")
```

```
## [1] '4.2.32'
```

```
library(parallel)
library(foreach)
library(doParallel)
```

```
## Loading required package: iterators
```

```
options(width = 95)

source("globalPower.r")
asreml.options(ai.sing = TRUE)
#nsimul <- 5
```

Load the design

```
load("plaid.dat.rda")
```

Add factors for heterogeneous variances

```
plaid.dat <- within(plaid.dat,
                    {
                      MotionsExpress <- fac.combine(list(Motions, Expressiveness),
                                                    combine.levels = TRUE)
                      WMotionsExpress <- fac.nested(MotionsExpress)
                    })
plaid.dat <- with(plaid.dat, plaid.dat[order(MotionsExpress, WMotionsExpress), ])
sim.dat <- plaid.dat[, -match("Y", names(plaid.dat))]
n <- nrow(plaid.dat)
```

Generate the simulation data sets

```
FH.dat <- makeSimData(params = params, design = sim.dat, TME = TME, nsimul = nsimul)
```

Initialize parallel processing

```
cl <- makeCluster(ncores)
registerDoParallel(cl)

cat("\n\n#### Number of simulated data sets is ",nsimul,"\n")
```

```
##
##
## #### Number of simulated data sets is  1000
```

Analyze the simulated data for model d

```
FH.d.stats <- foreach(k = 1:nsimul,
                      .errorhandling = "pass",
                      .inorder=TRUE,
                      .packages = c("asreml", "asremlPlus", "dae"))  %dopar%
  {
    analPower(k, dat = FH.dat$data)
  }
FH.d.stats <- do.call(rbind, FH.d.stats)
head(FH.d.stats, n = 25)
```

```
##    ModelFit avPairVar         pTME   pTraindiff
## 1         F  1.800790 0.6986400246 2.155233e-01
## 2         F 10.040937 0.1289920545 2.110197e-04
## 3         F  5.903648 0.0006139683 2.130253e-05
## 4         F  2.937221 0.0093056319 6.874277e-04
## 5         F  8.874333 0.0387451703 3.646334e-03
## 6         F  4.285144 0.0609917555 1.667377e-05
## 7         F  5.883586 0.0288687220 3.245607e-03
## 8         F  4.614748 0.1066174547 2.153943e-01
## 9         F  2.556519 0.2408432243 1.825489e-01
## 10        F  8.239108 0.0166346805 7.574252e-04
## 11        F  6.246807 0.0020062233 1.362482e-06
## 12        F  1.993604 0.6767915569 2.984845e-02
## 13        F  7.100146 0.0018449361 3.290240e-04
## 14        F  6.796540 0.0170832905 5.249062e-03
## 15        F  3.658755 0.0623740183 1.901290e-02
## 16        F  2.900895 0.0055187552 3.291929e-04
## 17        F  5.821270 0.0267844652 3.805476e-02
## 18        F  6.093910 0.0182400719 1.165139e-05
## 19        F  1.753810 0.2680033139 8.022267e-01
## 20        F  3.334128 0.0156972494 2.753802e-04
## 21        F  2.515409 0.3487086032 3.871141e-03
## 22        F  5.931871 0.4689153367 7.617624e-01
## 23        F  4.396592 0.0010173661 1.361417e-03
## 24        F 10.063696 0.1569357037 2.924261e-04
## 25        F  3.305725 0.1630481913 1.044459e-04
```

```
summary(FH.d.stats)
```

```
##    ModelFit           avPairVar            pTME           pTraindiff
##  Length:1000        Min.   : 0.6681   Min.   :0.00000   Min.   :0.0000000
##  Class :character   1st Qu.: 3.2417   1st Qu.:0.01815   1st Qu.:0.0007096
```

```
##  Mode  :character   Median : 4.5915   Median :0.08881   Median :0.0080198
##                     Mean   : 5.1118   Mean   :0.20429   Mean   :0.0571874
##                     3rd Qu.: 6.4366   3rd Qu.:0.30340   3rd Qu.:0.0457283
##                     Max.   :22.8761   Max.   :0.99869   Max.   :0.9864421
```

Analyze the simulated data for model c

```
FH.c.stats <- foreach(k = 1:nsimul,
                      .errorhandling = "pass",
                      .inorder=TRUE,
                      .packages = c("asreml", "asremlPlus", "dae"))  %dopar%
  {
    analPower(k, dat = FH.dat$data, model = "c")
  }
FH.c.stats <- do.call(rbind, FH.c.stats)
head(FH.c.stats, n = 25)
```

```
##    ModelFit avpairVar         pTME   pTraindiff
## 1         F  1.824276 0.7021381943 6.474865e-02
## 2         F 10.084816 0.1991660466 1.353359e-05
## 3         F  5.875360 0.0002774969 9.019313e-11
## 4         F  2.966012 0.0320801113 4.637979e-06
## 5         F  8.865617 0.0877922850 1.694764e-05
## 6         F  4.338831 0.2085886623 4.146236e-07
## 7         F  5.820711 0.1096556669 7.043364e-04
## 8         F  4.730611 0.1865748068 2.448451e-01
## 9         F  2.486760 0.2699100296 2.240038e-02
## 10        F  8.187588 0.0509778230 5.843177e-07
## 11        F  6.157004 0.0116136450 2.153652e-17
## 12        F  1.988488 0.6756911323 8.209220e-03
## 13        F  7.058306 0.0272118947 2.422354e-06
## 14        F  6.854632 0.0509919834 5.692206e-05
## 15        F  3.714792 0.0994882128 5.742454e-03
## 16        F  2.918926 0.0470053322 1.079110e-05
## 17        F  5.822584 0.0311289406 1.762585e-03
## 18        F  6.126270 0.1151129653 3.334271e-10
## 19        F  1.734193 0.3099355284 7.446848e-01
## 20        S        NA           NA           NA
## 21        F  2.507546 0.2408803337 8.106314e-05
## 22        F  5.920042 0.5143185851 6.783321e-01
## 23        F  4.345584 0.0253610472 6.045380e-06
## 24        F 10.114164 0.1687775445 5.778968e-08
## 25        F  3.259747 0.2393270752 1.398751e-06
```

```
summary(FH.c.stats)
```

```
##    ModelFit           avpairVar            pTME            pTraindiff
##  Length:1000        Min.   : 0.6572   Min.   :0.000277   Min.   :0.000000
##  Class :character   1st Qu.: 3.2692   1st Qu.:0.045732   1st Qu.:0.000002
##  Mode  :character   Median : 4.5955   Median :0.143314   Median :0.000307
##                     Mean   : 5.1165   Mean   :0.240130   Mean   :0.029630
##                     3rd Qu.: 6.4390   3rd Qu.:0.358297   3rd Qu.:0.006549
##                     Max.   :22.7083   Max.   :1.000000   Max.   :0.981704
##                     NA's   :16        NA's   :16         NA's   :16
```

Hand back clusters

```
stopCluster(cl)
```

Calculate power statistics

```
getStats <- function(stats)
{
  converged <- table(stats$ModelFit)["F"]
  power <- sum(stats$pTME <= 0.05, na.rm = TRUE)/as.numeric(table(stats$ModelFit)["F"])

  return(cbind(converged, power))
}
```

Power results

```
cat("\n\n#### Power statistics for model c)\n\n")
```

```
##
##
## #### Power statistics for model c)
```

```
getStats(FH.c.stats)
```

```
##   converged     power
## F       984 0.2621951
```

```
cat("\n\n#### Power statistics for model d)\n\n")
```

```
##
##
## #### Power statistics for model d)
```

```
getStats(FH.d.stats)
```

```
##   converged power
## F      1000 0.389
```

Save the workspace

```
save.image("FHpower.RData")
```