Analysing Bergman's rule with TaxMeta

Philipp Neubauer

2016-01-19

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

Estimating Bergman's rule for mammals

Using the supplied Bergman dataset:

```
data('Bergman')
head(Bergman)
```

```
##                     species       genus  corr  N  z_response        se
## 1              Alces alces       Alces  0.35  8  0.36544375 0.4472136
## 2          Anoura cultrata      Anoura  0.95  4  1.83178082 1.0000000
## 3       Blarina brevicauda     Blarina  0.01  6  0.01000033 0.5773503
## 4 Chaetodipus penicillatus Chaetodipus  0.10 16  0.10033535 0.2773501
## 5         Dipodomys agilis   Dipodomys  0.57  4  0.64752284 1.0000000
## 6   Dipodomys californicus   Dipodomys -0.33 38 -0.34282825 0.1690309
```

`?Bergman` gives info about the data. First, to apply the hierarchical model, we need to get the taxonomy using the taxize package:

```
taxonomy = c('species','genus','family','order')
Bergman_full <- get_tax(Bergman, taxonomy)
```

We can now apply the meta-analysis:

```
bergman_res_full <- TaxMeta(Bergman_full,
                        'z_response',
                        distribution = 'norm',
                        study_epsilon = 'se',
                        taxonomy = taxonomy,
                        type='full',
                        loo_waic = T,
                        n.chains = 3,
                        n.iter = 150e3,
                        n.thin=3,
                        n.burnin=25e3,
                        return_MCMC=T)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 40
##    Unobserved stochastic nodes: 178
```

```
##    Total graph size: 939
##
## Initializing model
```

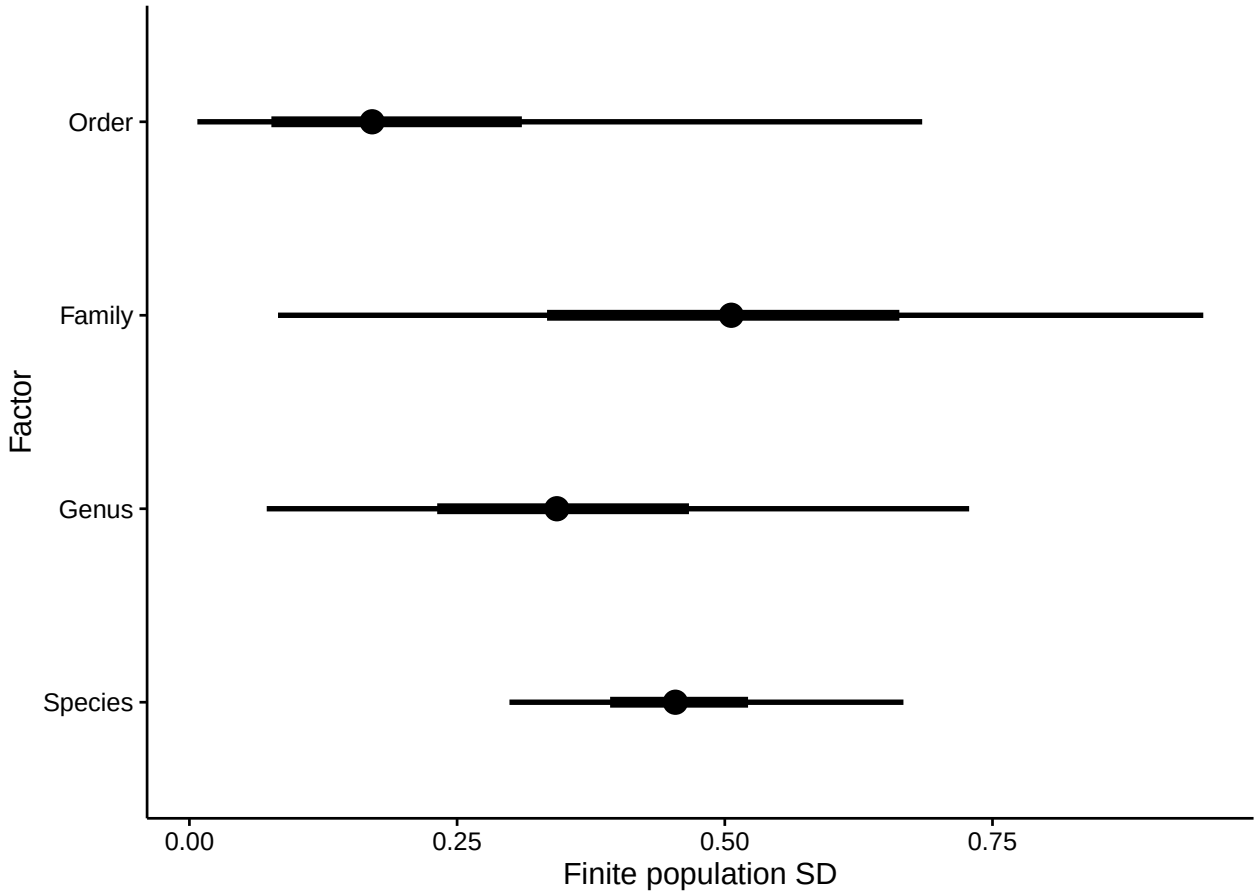
```
bergman_res_full
```

```
##
## Iterations = 26003:176000
## Thinning interval = 3
## Number of chains = 3
## Sample size per chain = 50000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean       SD  Naive SE Time-series SE
## Intercept     0.3087  0.28361 0.0007323       0.005524
## family_scale  9.3461 24.52737 0.0633294       0.344576
## genus_scale   8.1521 20.12186 0.0519544       0.337556
## hyper_scale   0.7988  1.04108 0.0026881       0.014159
## order_scale   8.5506 21.92537 0.0566111       0.316305
## sd.family     0.5018  0.22946 0.0005925       0.002326
## sd.genus      0.3571  0.17118 0.0004420       0.001700
## sd.order      0.2180  0.18641 0.0004813       0.002227
## sd.species    0.4616  0.09516 0.0002457       0.000624
## species_scale 9.0767 28.15061 0.0726846       0.355813
##
## 2. Quantiles for each variable:
##
##                    2.5%     25%    50%    75%   97.5%
## Intercept     -0.234897 0.13720 0.3111 0.4850  0.8523
## family_scale   0.421807 1.81810 4.0042 9.0323 49.6240
## genus_scale    0.546535 1.87578 3.7752 8.0972 40.9338
## hyper_scale    0.044082 0.22301 0.4781 0.9746  3.5039
## order_scale    0.353079 1.62921 3.6392 8.3352 45.6604
## sd.family      0.082755 0.33400 0.5060 0.6630  0.9468
## sd.genus       0.072224 0.23141 0.3432 0.4665  0.7282
## sd.order       0.007446 0.07656 0.1707 0.3105  0.6843
## sd.species     0.298900 0.39290 0.4539 0.5217  0.6668
## species_scale  0.226921 1.36539 3.3743 8.2622 51.0307
##
## Potential scale reduction factors:
##
##               Point est. Upper C.I.
## Intercept           1.00       1.01
## family_scale        1.01       1.02
## genus_scale         1.00       1.00
## hyper_scale         1.00       1.00
## order_scale         1.01       1.01
## sd.family           1.00       1.00
## sd.genus            1.00       1.00
## sd.order            1.00       1.00
## sd.species          1.00       1.00
## species_scale       1.01       1.01
```

```
##
## Multivariate psrf
##
## 1
##
##
##  waic 41.30119 waic SE 8.436146
##
##
##  loo 68.31956 loo SE 8.512628
```

To plot the contributions of different taxonomic levels, simply plot the result:

```
plot(bergman_res_full)
```



To generate predictions at other taxonomic levels, simply add some bogus data to the data. Say, we need an estimate for a species for which the genus was not in the orginal dataset, but the family was part of the dataset:

```
Bergman_pred <- Bergman_full
Bergman_pred[length(Bergman_pred)+1,] <- Bergman_pred[11,]
Bergman_pred[length(Bergman_pred)+2,] <- Bergman_pred[11,]

Bergman_pred[length(Bergman_full)+1,'genus'] <- 'bogus_genus'
Bergman_pred[length(Bergman_full)+1,'species'] <- 'Bogus species'
```

```
Bergman_pred[length(Bergman_full)+1,'z_response'] <- NA
Bergman_pred[length(Bergman_full)+2,'species'] <- 'Bogus species'
Bergman_pred[length(Bergman_full)+2,'z_response'] <- NA

bergman_res_pred <- TaxMeta(Bergman_pred,
                        'z_response',
                        distribution = 'norm',
                        study_epsilon = 'se',
                        taxonomy = taxonomy,
                        type='full',
                        loo_waic = T,
                        n.chains = 3,
                        n.iter = 150e3,
                        n.thin=30,
                        n.burnin=25e3,
                        return_MCMC=T)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 38
##    Unobserved stochastic nodes: 180
##    Total graph size: 942
##
## Initializing model
```

```
bergman_res_pred
```

```
##
## Iterations = 26030:176000
## Thinning interval = 30
## Number of chains = 3
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                   Mean      SD  Naive SE Time-series SE
## Intercept      0.33677  0.2831 0.0023118      0.0053266
## family_scale  10.03887 25.0154 0.2042497      0.5407525
## genus_scale    9.27884 28.6032 0.2335438      0.6920907
## hyper_scale    0.73569  0.9382 0.0076605      0.0140853
## order_scale    9.43829 36.4366 0.2975035      0.5898891
## pred[1]        0.00645  0.8176 0.0066760      0.0068823
## pred[2]       -0.11394  0.6758 0.0055180      0.0055179
## sd.family      0.52492  0.2232 0.0018223      0.0024998
## sd.genus       0.35853  0.1735 0.0014165      0.0020685
## sd.order       0.21703  0.1843 0.0015046      0.0023249
## sd.species     0.43642  0.1067 0.0008712      0.0009835
## species_scale 10.05286 35.8544 0.2927498      0.6322678
##
## 2. Quantiles for each variable:
```

```
##
##                    2.5%      25%      50%    75%   97.5%
## Intercept     -0.212000  0.16193  0.33685 0.5126  0.8774
## family_scale   0.466772  2.00368  4.35525 9.8490 51.1339
## genus_scale    0.583186  2.05407  4.20043 8.9619 47.0983
## hyper_scale    0.040627  0.20640  0.43891 0.8915  3.2648
## order_scale    0.376918  1.74971  3.90966 8.8528 48.0171
## pred[1]       -1.578032 -0.46169 -0.01657 0.4603  1.6729
## pred[2]       -1.449752 -0.46143 -0.12484 0.2305  1.2738
## sd.family      0.098188  0.36928  0.53246 0.6799  0.9510
## sd.genus       0.074464  0.22882  0.34084 0.4717  0.7367
## sd.order       0.007081  0.07719  0.17014 0.3083  0.6849
## sd.species     0.257955  0.36065  0.42686 0.4999  0.6701
## species_scale  0.238670  1.46074  3.66265 9.0405 53.8473
##
## Potential scale reduction factors:
##
##               Point est. Upper C.I.
## Intercept           1.00       1.01
## family_scale        1.11       1.12
## genus_scale         1.23       1.27
## hyper_scale         1.00       1.01
## order_scale         1.22       1.24
## pred[1]             1.00       1.00
## pred[2]             1.00       1.00
## sd.family           1.00       1.00
## sd.genus            1.00       1.00
## sd.order            1.00       1.00
## sd.species          1.00       1.00
## species_scale       1.13       1.13
##
## Multivariate psrf
##
## 1
##
##
##  waic 44.45657 waic SE 8.712652
##
##
##  loo 66.54654 loo SE 8.362529
```

We can also compare alternative model formulations:

```
bergman_res_fixed <- TaxMeta(Bergman_full,
                        'z_response',
                        distribution = 'norm',
                        study_epsilon = 'se',
                        taxonomy = taxonomy,
                        type='fixed',
                        scale=0.1,
                        loo_waic = T,
                        n.chains = 3,
                        n.iter = 150e3,
                        n.thin=3,
```

```
                        n.burnin=20e3,
                        return_MCMC=F)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 40
##    Unobserved stochastic nodes: 92
##    Total graph size: 656
##
## Initializing model
```

```
bergman_res_gamma <- TaxMeta(Bergman_full,
                        'z_response',
                        distribution = 'norm',
                        study_epsilon = 'se',
                        taxonomy = taxonomy,
                        type='gamma',
                        loo_waic = T,
                        n.chains = 3,
                        n.iter = 150e3,
                        n.thin=3,
                        n.burnin=25e3,
                        return_MCMC=F)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 40
##    Unobserved stochastic nodes: 93
##    Total graph size: 658
##
## Initializing model
```

```
bergman_res_unif <- TaxMeta(Bergman_full,
                        'z_response',
                        distribution = 'norm',
                        study_epsilon = 'se',
                        taxonomy = taxonomy,
                        type='uniform',
                        loo_waic = T,
                        n.chains = 3,
                        n.iter = 150e3,
                        n.thin=3,
                        n.burnin=25e3,
                        return_MCMC=F)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 40
##    Unobserved stochastic nodes: 93
##    Total graph size: 658
##
## Initializing model
```

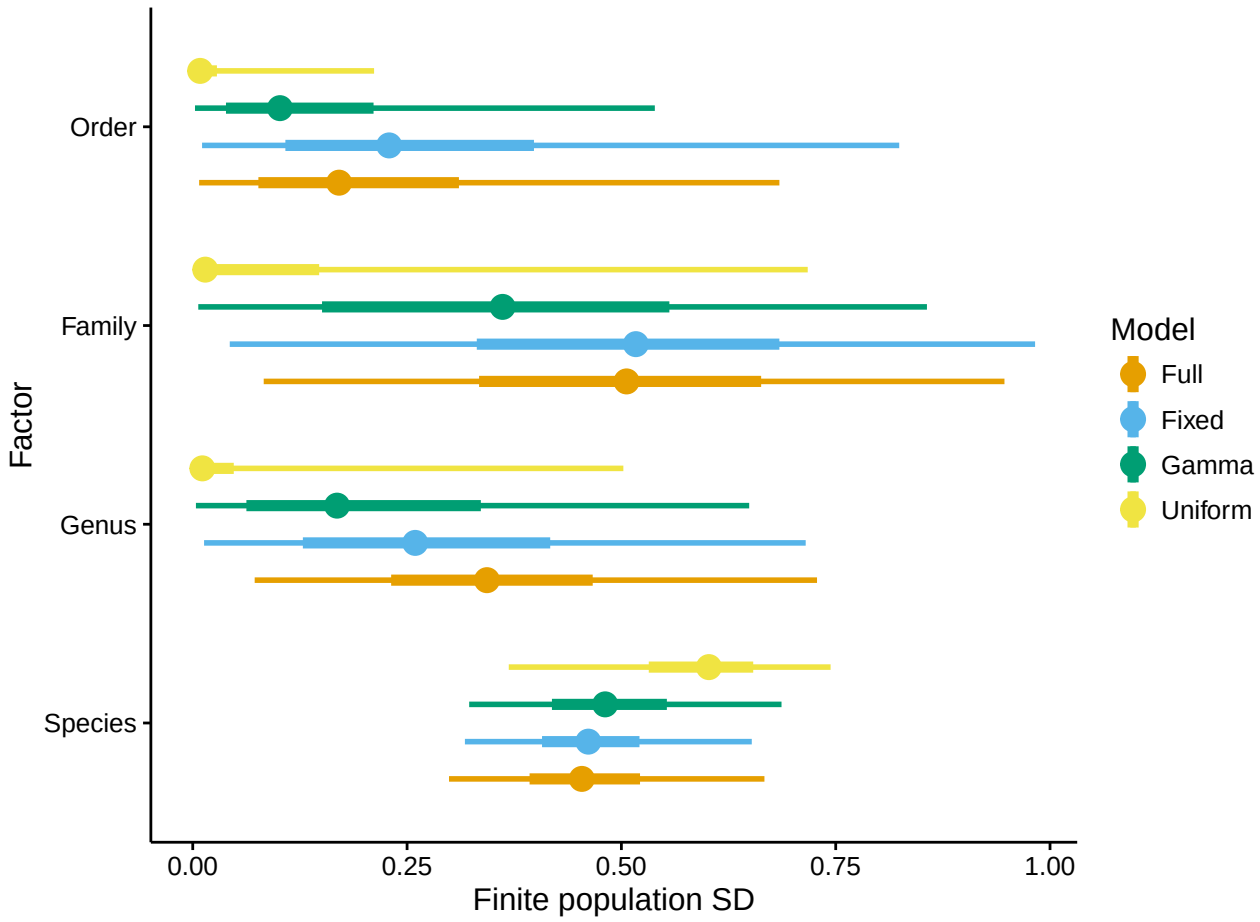
And compare the results:

```
loo_comp <- loo::compare(bergman_res_fixed$loo,
             bergman_res_gamma$loo,
             bergman_res_unif$loo,
             bergman_res_full$loo)

loo_comp <- data.frame(loo_comp)
loo_comp$model <- do.call('rbind',strsplit(rownames(loo_comp),split = '\\$'))[,1]
loo_comp$loo_weight = loo_comp$weight

waic_comp <- loo::compare(bergman_res_fixed$waic,
             bergman_res_gamma$waic,
             bergman_res_unif$waic,
             bergman_res_full$waic)

waic_comp <- data.frame(waic_comp)
waic_comp$model <- do.call('rbind',strsplit(rownames(waic_comp),split = '\\$'))[,1]
waic_comp$waic_weight = waic_comp$weight

comp_tab <- dplyr::inner_join(loo_comp,waic_comp,by='model')

knitr::kable(comp_tab[,c('model','looic','se_looic','loo_weight','waic','se_waic','waic_weight')],
             row.names=F,
             digits=c(0,1,1,2,1,1,2),
             align=c('l','r','r','r','r','r','r'))
```

model	looic	se_looic	loo_weight	waic	se_waic	waic_weight
bergman_res_fixed	65.6	7.9	0.58	39.4	8.1	0.46
bergman_res_gamma	67.8	8.3	0.19	40.9	8.4	0.22
bergman_res_full	68.3	8.5	0.15	41.3	8.4	0.18
bergman_res_unif	69.3	8.5	0.09	41.8	8.6	0.14

This comparison does not give a good idea of model performance, so lets try a plot:

```
plot(Full=bergman_res_full,
     Fixed =bergman_res_fixed,
     Gamma =bergman_res_gamma,
     Uniform=bergman_res_unif)
```


Order
Family
Genus
Species
Factor
0.00
0.25
0.50
0.75
1.00
Finite population SD
Model
Full
Fixed
Gamma
Uniform