Ecological factors influencing primate vision conformation: a phylogenetic regression workflow for the *mmodely* R-package (Version 0.2.5)

David M. Schruth
dschruth@anthropoidea.org

November 7, 2024

1 Introduction

Traits evolve in the context of historically and ecologically complex arrangements that can present difficulty for researchers attempting to uncover causal relationships [5] or draw inference to larger populations [7]. Primates represent a manageably diverse clade of mammals who exhibit a wide range of behaviors and morphology conducive to revealing evolutionary processes. Ideas on the origins on primates range from predation detection [1] or deterrence [17] to targeting in hunting [2] or locomotion [3]. Here, using the *mmodely* package on primate data for locomotion [16] and vision [4], I demonstrate how the origins of primate cranial morphology can be elucidated via several ecological variables from numerous datasets [14]. Model averaging [MA] [8] and model selection [MS] [9] results primarily highlight arboreal locomotor targeting and trophic security [17] variables (such as stature or group size) as playing key roles in determining convergence of primate orbits. The *mmodely* package enables implementation of a combination of phylogenetic controlled regression [10] and information theoretic [11] (MA and MS) examination to simultaneously compare (weighted) predictor coefficients across the numerous sub-datasets generated during the exploration of all possible model combinations.

2 Licensing

The *mmodely* package is licensed under the Apache License v2.0: it is therefore free to use and redistribute, however, we, the copyright holders, wish to maintain primary control over any further development. Please be sure to cite *mmodely* if you use the package in presentations or work leading to publication.

3 Installation

This package largely depends upon the *caper* package, but most functions do not require any particular library. It is recommended that you have *caper*, *ape*, and the *caroline* package installed as a minimum.

```
> # wget https://cran.r-project.org/src/contrib/Archive/caroline/caroline_0.8.0.tar.gz
> # wget https://cran.r-project.org/src/contrib/Archive/caper/caper_0.5.tar.gz
> # wget https://cran.r-project.org/src/contrib/Archive/ape/ape_3.0-5.tar.gz
> # R CMD INSTALL caroline_0.8.0.tar.gz
> # R CMD INSTALL caper_0.5.tar.gz
> # R CMD INSTALL ape_3.0-5.tar.gz
```

Building the *mmodely* package from source requires that you have the proper dependency packages, *caroline*, installed from CRAN. This can typically be accomplished via the following commands from within the R command line environment:

```
install.packages(c('caroline','ape','caper'))
```

After a successful installation the *mmodely* package can be loaded in the normal way: by starting R and invoking the following `library` command:

```
> library(caper)
> library(mmodely)
```

4 Reading in Data

Read in the tree [15] and datasets then merge them together.

```
> data.path <- system.file("extdata","primate-example.data.csv", package="mmodely")
> data <- read.csv(data.path, row.names=1)
> data$gn_sp <- rownames(data)
> # merge data sets here if applicable
>
> tree.path <- system.file("extdata","primate-springer.2012.tre", package="mmodely")
> phyl <- ape::read.tree(tree.path)[[5]]
> #5. RAxML phylogram based on the 61199 bp concatenation of 69 nuclear and ten mitochondrial genes.
>
> phyl <- trim.phylo(phylo=phyl, gs.vect=data$gn_sp) # prune unused nodes and branches
> comp <- comp.data(phylo=phyl, df=data)
```

Typically there will be some missing data (species) in certain sources that do not occur in others. A merge of these will result in NA values for some cells. The more missing cells and merges there are, the more sub-datasets will be possible, due to case-wise deletion in the process of combinatorics underlying model iteration, averaging, and selection. The above example has little if any missing data, but the examples below introduce some artificially.

5 Basic Reporting

First, for illustration purposes, we perform a simple analysis of a single model using 'pgls' directly from the *caper* package, then show-off the 'pgls.report' functionality of the *mmodely* package. ANOVA, AIC, and one-line model reports can be output via this function.

```
> model <- as.formula('OC ~ mass.Kg + group.size')
> fit <- caper::pgls(formula=model, data=comp)
> summary(fit)

Call:
caper::pgls(formula = model, data = comp)

Residuals:
    Min      1Q  Median      3Q     Max 
-55.382 -12.734  -4.075  27.512  57.101 

Branch length transformations:

kappa  [Fix]  : 1.000
lambda [Fix]  : 1.000
delta  [Fix]  : 1.000

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 66.037399   8.409402  7.8528 7.05e-07 ***
mass.Kg      0.153139   0.060894  2.5148  0.02298 *  
group.size  -0.030092   0.107433 -0.2801  0.78299    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.2 on 16 degrees of freedom
Multiple R-squared: 0.3513,	Adjusted R-squared: 0.2702 
F-statistic: 4.332 on 2 and 16 DF,  p-value: 0.03136 

> pgls.report(comp, f=model, anova=TRUE, QC.plot=TRUE)

Call:
pgls(formula = f, data = cd, lambda = l, kappa = k, delta = d, 
    bounds = bounds)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-55.382 -12.734  -4.075  27.512  57.101

Branch length transformations:

kappa  [Fix]  : 1.000
lambda [Fix]  : 1.000
delta  [Fix]  : 1.000

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 66.037399   8.409402  7.8528 7.05e-07 ***
mass.Kg      0.153139   0.060894  2.5148  0.02298 *
group.size  -0.030092   0.107433 -0.2801  0.78299
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.2 on 16 degrees of freedom
Multiple R-squared: 0.3513,        Adjusted R-squared: 0.2702
F-statistic: 4.332 on 2 and 16 DF,  p-value: 0.03136
[1] "AIC = 134.2"
Analysis of Variance Table
Sequential SS for pgls: lambda = 1.00, delta = 1.00, kappa = 1.00

Response: OC
           Df  Sum Sq Mean Sq F value   Pr(>F)
mass.Kg     1 10043.9 10043.9  8.5862 0.009806 **
group.size  1    91.8    91.8  0.0785 0.782991
Residuals  16 18716.2  1169.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
 ++mass(0.023)  |  group(0.783)

Call:
pgls(formula = f, data = cd, lambda = l, kappa = k, delta = d,
    bounds = bounds)

Coefficients:
(Intercept)      mass.Kg   group.size
   66.03740      0.15314     -0.03009
```

6 Multivariate Combinatoric Iteration

The *mmodely* package's chief contribution is enabling approaches that utilize multi-model iteration averaging. Using a smaller subset of variables can speed up the (slower) maximum likelihood computation step and still achieve the desired result of fixed tree transformation parameters.

```
> pv0 <- c("mass.Kg","group.size","arboreal","nocturnal") #"swing.pct"
> est.mods <- get.model.combos(predictor.vars=pv0, outcome.var='OC', min.q=2)
> ps <- get.phylo.stats(phylo=phyl, data=data, trait.clmn='OC');

$lambda
[1] 0.8133958

$logL
[1] -204.9656

$P
[1] 0

$K
[1] 0.7251836

$P
[1] 0.001

> lambda <- ps$lambda$lambda ; print(lambda)

[1] 0.8133958

> PGLSi <- pgls.iter(models=est.mods, phylo=phyl, df=data, l=lambda, k='ML', d='ML')

1 OC~mass.Kg+group.size+arboreal+nocturnal
2 OC~mass.Kg+group.size+arboreal
3 OC~mass.Kg+group.size+nocturnal
4 OC~mass.Kg+arboreal+nocturnal
5 OC~group.size+arboreal+nocturnal
6 OC~mass.Kg+group.size
7 OC~mass.Kg+arboreal
8 OC~mass.Kg+nocturnal
9 OC~group.size+arboreal
10 OC~group.size+nocturnal
11 OC~arboreal+nocturnal

> pgls.iter.stats(PGLSi) # check run, especially to see how few sub-datasets exist

models: 11
dimensions of sub-datasets:
    q     n   qXn rwGsm
    3     1     3     1
               q  n
Min.    2.000000 63
1st Qu. 2.000000 63
Median  2.000000 63
Mean    2.545455 63
3rd Qu. 3.000000 63
Max.    4.000000 63
tree transformation parameter averages:
        l         k         d
0.8133958 1.4591957 1.2008048
distributions of optimization parameters:
       n            q             rwGsm           model.no          R2
 Min.   :63   Min.   :2.000   Min.   :117909   Min.   : 1.0   Min.   :0.04241
```

```
 1st Qu.:63   1st Qu.:2.000   1st Qu.:117909   1st Qu.: 3.5   1st Qu.:0.05945
 Median :63   Median :2.000   Median :117909   Median : 6.0   Median :0.20084
 Mean   :63   Mean   :2.545   Mean   :117909   Mean   : 6.0   Mean   :0.15453
 3rd Qu.:63   3rd Qu.:3.000   3rd Qu.:117909   3rd Qu.: 8.5   3rd Qu.:0.21835
 Max.   :63   Max.   :4.000   Max.   :117909   Max.   :11.0   Max.   :0.23433
     R2.adj              AIC             AICc             BIC             AICw
 Min.   :0.01049   Min.   :400.1   Min.   :400.5   Min.   :406.5   Min.   :0.0003387
 1st Qu.:0.02002   1st Qu.:401.5   1st Qu.:402.1   1st Qu.:409.3   1st Qu.:0.0006806
 Median :0.16042   Median :402.6   Median :403.7   Median :412.0   Median :0.0654038
 Mean   :0.11865   Mean   :405.8   Mean   :406.4   Mean   :413.4   Mean   :0.0909091
 3rd Qu.:0.18158   3rd Qu.:412.4   3rd Qu.:412.8   3rd Qu.:418.8   3rd Qu.:0.1460563
 Max.   :0.19116   Max.   :413.5   Max.   :414.2   Max.   :422.1   Max.   :0.3197358
      BICw
 Min.   :0.0001827
 1st Qu.:0.0009305
 Median :0.0287186
 Mean   :0.0909091
 3rd Qu.:0.1117665
 Max.   :0.4371383
```

7 Tree Transformation Averaging and Re-iteration

After running PGLS on a test-subset of predictor-variable combinations using maximum likelihood, we can average the tree transformation parameters [18] to obtain fixed values going forward. This approach can speed up computations for larger sets of modeling data and variable combinations.

```
> tt.avgs <- apply(PGLSi$params, 2, mean, na.rm=TRUE) # tree transformation averages
> print(tt.avgs)
```

```
        l         k         d
0.8133958 1.4591957 1.2008048
```

Next we use the full set of variables and our tree transform averages. For demonstration, we sprinkle in some missing values to our dataset so as to artificially boost the number of sub-datasets. The subsequent fixed tree parameter itteration run should now generate more diverse output upon which the *mmodely* can demonstrate it unique model averaging and model selection functionality.

```
> pvs <- c("mass.Kg","group.size","infant.carry","arboreal","DPL.km","nocturnal")
> all.mods <- get.model.combos(predictor.vars=pvs, outcome.var='OC', min.q=2)
> data <- subset(data,!grepl(rownames(data),pattern='gorilla')) # remove an OC measurement outlier
> # randomly sprinkle in some missing values (for more interesting for model selection)
> missing.value.ct <- 1
> for(pv in pv0){ data[sample(x=1:nrow(data),size=missing.value.ct),pv] <- NA}
> PGLSi <- pgls.iter(models=all.mods, phylo=phyl, df=data, l=lambda, k=tt.avgs['k'], d=tt.avgs['d'])
```

```
1 OC~mass.Kg+group.size+infant.carry+arboreal+DPL.km+nocturnal
2 OC~mass.Kg+group.size+infant.carry+arboreal+DPL.km
3 OC~mass.Kg+group.size+infant.carry+arboreal+nocturnal
4 OC~mass.Kg+group.size+infant.carry+DPL.km+nocturnal
5 OC~mass.Kg+group.size+arboreal+DPL.km+nocturnal
6 OC~mass.Kg+infant.carry+arboreal+DPL.km+nocturnal
7 OC~group.size+infant.carry+arboreal+DPL.km+nocturnal
8 OC~mass.Kg+group.size+infant.carry+arboreal
9 OC~mass.Kg+group.size+infant.carry+DPL.km
10 OC~mass.Kg+group.size+infant.carry+nocturnal
11 OC~mass.Kg+group.size+arboreal+DPL.km
12 OC~mass.Kg+group.size+arboreal+nocturnal
13 OC~mass.Kg+group.size+DPL.km+nocturnal
14 OC~mass.Kg+infant.carry+arboreal+DPL.km
15 OC~mass.Kg+infant.carry+arboreal+nocturnal
16 OC~mass.Kg+infant.carry+DPL.km+nocturnal
17 OC~mass.Kg+arboreal+DPL.km+nocturnal
18 OC~group.size+infant.carry+arboreal+DPL.km
19 OC~group.size+infant.carry+arboreal+nocturnal
20 OC~group.size+infant.carry+DPL.km+nocturnal
21 OC~group.size+arboreal+DPL.km+nocturnal
22 OC~infant.carry+arboreal+DPL.km+nocturnal
23 OC~mass.Kg+group.size+infant.carry
24 OC~mass.Kg+group.size+arboreal
25 OC~mass.Kg+group.size+DPL.km
26 OC~mass.Kg+group.size+nocturnal
27 OC~mass.Kg+infant.carry+arboreal
28 OC~mass.Kg+infant.carry+DPL.km
29 OC~mass.Kg+infant.carry+nocturnal
30 OC~mass.Kg+arboreal+DPL.km
31 OC~mass.Kg+arboreal+nocturnal
32 OC~mass.Kg+DPL.km+nocturnal
33 OC~group.size+infant.carry+arboreal
34 OC~group.size+infant.carry+DPL.km
35 OC~group.size+infant.carry+nocturnal
36 OC~group.size+arboreal+DPL.km
37 OC~group.size+arboreal+nocturnal
```

```
38 OC~group.size+DPL.km+nocturnal
39 OC~infant.carry+arboreal+DPL.km
40 OC~infant.carry+arboreal+nocturnal
41 OC~infant.carry+DPL.km+nocturnal
42 OC~arboreal+DPL.km+nocturnal
43 OC~mass.Kg+group.size
44 OC~mass.Kg+infant.carry
45 OC~mass.Kg+arboreal
46 OC~mass.Kg+DPL.km
47 OC~mass.Kg+nocturnal
48 OC~group.size+infant.carry
49 OC~group.size+arboreal
50 OC~group.size+DPL.km
51 OC~group.size+nocturnal
52 OC~infant.carry+arboreal
53 OC~infant.carry+DPL.km
54 OC~infant.carry+nocturnal
55 OC~arboreal+DPL.km
56 OC~arboreal+nocturnal
57 OC~DPL.km+nocturnal
```

8 Fixed iteration run statistics

We should briefly inspect how this fixed iteration run performed and how many sub-datasets we need to investigate. It is recommended to try *mmodely* using 'rwGsm.' This abbreviation stands for 'raw *Genus species* sums.' It represents a sum of the (concatenated) raw character values of all species constituting the underlying dataset (which has all rows with any missing data removed) for a particular combination of model predictor variables. While this default is preferred, the number of species 'n' [default] or number of model variables 'q' can also be used.

```
> pgls.iter.stats(PGLSi)

models: 57
dimensions of sub-datasets:
    q     n   qXn rwGsm
    5     9    16    31
               q        n
Min.    2.000000 53.00000
1st Qu. 2.000000 55.00000
Median  3.000000 56.00000
Mean    3.263158 57.10526
3rd Qu. 4.000000 60.00000
Max.    6.000000 61.00000
tree transformation parameter averages:
        l         k       d
0.8133958 1.4591957 1.2008048
distributions of optimization parameters:
       n              q             rwGsm           model.no        R2
 Min.   :53.00   Min.   :2.000   Min.   : 98558   Min.   : 1   Min.   :0.00362
 1st Qu.:55.00   1st Qu.:2.000   1st Qu.:102113   1st Qu.:15   1st Qu.:0.06607
 Median :56.00   Median :3.000   Median :104197   Median :29   Median :0.10661
 Mean   :57.11   Mean   :3.263   Mean   :106654   Mean   :29   Mean   :0.09621
 3rd Qu.:60.00   3rd Qu.:4.000   3rd Qu.:112386   3rd Qu.:43   3rd Qu.:0.11997
 Max.   :61.00   Max.   :6.000   Max.   :114596   Max.   :57   Max.   :0.16022
     R2.adj              AIC             AICc             BIC              AICw
 Min.   :-0.03074   Min.   :345.9   Min.   :347.8   Min.   :357.8   Min.   :0.0000000
 1st Qu.: 0.02009   1st Qu.:356.4   1st Qu.:357.4   1st Qu.:365.0   1st Qu.:0.0000000
 Median : 0.04045   Median :364.5   Median :365.3   Median :372.6   Median :0.0000738
 Mean   : 0.03995   Mean   :369.9   Mean   :370.8   Mean   :378.5   Mean   :0.0175439
 3rd Qu.: 0.06142   3rd Qu.:384.6   3rd Qu.:385.7   3rd Qu.:392.6   3rd Qu.:0.0038459
 Max.   : 0.09923   Max.   :395.4   Max.   :395.8   Max.   :401.7   Max.   :0.4671129
```

```
      BICw
 Min.   :0.0000000
 1st Qu.:0.0000000
 Median :0.0002093
 Mean   :0.0175439
 3rd Qu.:0.0092162
 Max.   :0.3472579
```

9 Model Averaging

Now we can estimate the predictor variable parameters by averaging over all possible fixed PGLS runs, using the AICc differences (from the lowest AICc) as weights. By default this AICw weighted average is performed per sub-dataset using 'rwGsm' or 'n' [default] as mentioned in the preceding section. While model averaging is not recommended under high multicolinearity, as denominators of regression coefficients change across models, it is possible to rescale these using 'standarize' [13]. A slightly more conservative alternative to MA uses 'model importance' which is equivalent to an AIC-weighted MA of binary indicators of presence or absence of covariate model inclusion [12].

```
> w.means.pds <- average.fit.models(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim, by='rwGsm', standardize=TRU
> #
> apply(w.means.pds, 2, mean, na.rm=T) #average of weighted means over all sub-datasets

     mass.Kg   group.size infant.carry     arboreal       DPL.km    nocturnal
  0.14808937   0.09737625   5.42210437   0.58122187  -0.14692097  -1.97411938

> w.means.pds                                         # weighted means    per   sub-dataset

       mass.Kg group.size infant.carry arboreal   DPL.km nocturnal
98558  0.15703    0.09432      5.10026  2.15505 -0.10861  -1.44005
100270 0.17071        NaN      5.56540  1.20656  0.55895  -1.62289
100275 0.13427    0.08808      4.60875      NaN -0.57430  -1.24644
100396 0.15618    0.09391      5.16899  2.15042 -0.01737       NaN
100642     NaN    0.09954      5.46251  1.34996 -0.23355  -2.61006
101987 0.15675        NaN      5.19131      NaN  0.15665  -1.52027
102108 0.17006        NaN      5.61600  1.18237  0.63972       NaN
102113 0.13306    0.08664      4.75636      NaN -0.51235       NaN
102354     NaN        NaN      5.93049  0.22829  0.51250  -2.68039
102359     NaN    0.09643      5.09970      NaN -0.52894  -2.48492
102480     NaN    0.09957      5.60673  1.25806 -0.08883       NaN
103825 0.15623        NaN      5.27714      NaN  0.21657       NaN
104071     NaN        NaN      5.75197      NaN  0.31787  -2.66272
104192     NaN        NaN      6.07194  0.12960  0.63980       NaN
104197     NaN    0.09671      5.29904      NaN -0.40059       NaN
105909     NaN        NaN      6.24708      NaN  0.44010       NaN
108957 0.13540    0.09444          NaN  0.66988 -0.66177  -1.16128
110669 0.15206        NaN          NaN -0.18125 -0.09096  -1.36769
110674 0.12731    0.09724          NaN      NaN -0.72107  -1.08671
110795 0.13532    0.09349          NaN  0.79417 -0.61195       NaN
111041     NaN    0.09968          NaN  0.11684 -0.71865  -2.51637
112386 0.15303        NaN          NaN      NaN -0.06473  -1.40119
112507 0.15207        NaN          NaN -0.08382 -0.04130       NaN
112512 0.12620    0.09537          NaN      NaN -0.69933       NaN
112753     NaN        NaN          NaN -0.90045 -0.08380  -2.61947
112758     NaN    0.10529          NaN      NaN -0.68907  -2.47073
112879     NaN    0.09962          NaN  0.13621 -0.62789       NaN
114224 0.15375        NaN          NaN      NaN -0.03589       NaN
114470     NaN        NaN          NaN      NaN  0.08042  -2.69473
114591     NaN        NaN          NaN -0.91234  0.00561       NaN
114596     NaN    0.11769          NaN      NaN -0.61179       NaN
attr(,"MSE")
       mass.Kg group.size infant.carry arboreal DPL.km nocturnal
```

```
98558  0.00057    0.00163      0.03826  0.04211      0   0.01898
100270 0.00006         NA      0.02798  0.02942      0   0.00602
100275 0.00039    0.00151      0.05094       NA      0   0.01721
100396 0.00021    0.00018      0.00430  0.00059      0        NA
100642      NA    0.00012      0.00462  0.00434      0   0.00152
101987 0.00001         NA      0.00136       NA      0   0.00039
102108 0.00001         NA      0.00792  0.00861      0        NA
102113 0.00010    0.00035      0.01155       NA      0        NA
102354      NA         NA      0.00088  0.00090      0   0.00014
102359      NA    0.00007      0.00228       NA      0   0.00075
102480      NA    0.00001      0.00017  0.00026      0        NA
103825 0.00000         NA      0.00042       NA      0        NA
104071      NA         NA      0.00008       NA      0   0.00002
104192      NA         NA      0.00017  0.00018      0        NA
104197      NA    0.00001      0.00027       NA      0        NA
105909      NA         NA      0.00000       NA      0        NA
108957 0.00000    0.00000           NA  0.00000      0   0.00000
110669 0.00000         NA           NA  0.00000      0   0.00000
110674 0.00000    0.00000           NA       NA      0   0.00000
110795 0.00000    0.00000           NA  0.00000      0        NA
111041      NA    0.00000           NA  0.00000      0   0.00000
112386 0.00000         NA           NA       NA      0   0.00000
112507 0.00000         NA           NA  0.00000      0        NA
112512 0.00000    0.00000           NA       NA      0        NA
112753      NA         NA           NA  0.00000      0   0.00000
112758      NA    0.00000           NA       NA      0   0.00000
112879      NA    0.00000           NA  0.00000      0        NA
114224 0.00000         NA           NA       NA      0        NA
114470      NA         NA           NA       NA      0   0.00000
114591      NA         NA           NA  0.00000      0        NA
114596      NA    0.00000           NA       NA      0        NA

> w.import.pds <- variable.importance(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim, by='rwGsm')
> #
> apply(w.import.pds, 2, mean, na.rm=T) #average of weighted means over all sub-datasets

     mass.Kg   group.size infant.carry     arboreal       DPL.km    nocturnal
   1.0000000    1.0000000    1.0000000    0.6693031    0.3225806    0.0000000

> w.import.pds                                    # weighted means    per   sub-dataset

       mass.Kg group.size infant.carry arboreal DPL.km nocturnal
98558        1          1            1  1.00000      0         0
100270       1        NaN            1  1.00000      1         0
100275       1          1            1      NaN      0         0
100396       1          1            1  1.00000      0       NaN
100642     NaN          1            1  1.00000      0         0
101987       1        NaN            1      NaN      1         0
102108       1        NaN            1  1.00000      1       NaN
102113       1          1            1      NaN      0       NaN
102354     NaN        NaN            1  1.00000      1         0
102359     NaN          1            1      NaN      0         0
102480     NaN          1            1  1.00000      0       NaN
103825       1        NaN            1      NaN      1       NaN
104071     NaN        NaN            1      NaN      1         0
104192     NaN        NaN            1  0.27372      1       NaN
104197     NaN          1            1      NaN      0       NaN
105909     NaN        NaN            1      NaN      1       NaN
108957       1          1          NaN  1.00000      0         0
110669       1        NaN          NaN  0.00000      0         0
110674       1          1          NaN      NaN      0         0
```

```
110795       1          1          NaN  1.00000      0       NaN
111041     NaN          1          NaN  0.71607      0         0
112386       1        NaN          NaN      NaN      0         0
112507       1        NaN          NaN  0.00000      0       NaN
112512       1          1          NaN      NaN      0       NaN
112753     NaN        NaN          NaN  0.00000      0         0
112758     NaN          1          NaN      NaN      0         0
112879     NaN          1          NaN  0.71906      0       NaN
114224       1        NaN          NaN      NaN      0       NaN
114470     NaN        NaN          NaN      NaN      1         0
114591     NaN        NaN          NaN  0.00000      1       NaN
114596     NaN          1          NaN      NaN      0       NaN
attr(,"MSE")
       mass.Kg group.size infant.carry arboreal DPL.km nocturnal
98558        0          0            0  0.00000      0         0
100270       0         NA            0  0.00000      0         0
100275       0          0            0       NA      0         0
100396       0          0            0  0.00000      0        NA
100642      NA          0            0  0.00000      0         0
101987       0         NA            0       NA      0         0
102108       0         NA            0  0.00000      0        NA
102113       0          0            0       NA      0        NA
102354      NA         NA            0  0.00000      0         0
102359      NA          0            0       NA      0         0
102480      NA          0            0  0.00000      0        NA
103825       0         NA            0       NA      0        NA
104071      NA         NA            0       NA      0         0
104192      NA         NA            0  0.00022      0        NA
104197      NA          0            0       NA      0        NA
105909      NA         NA            0       NA      0        NA
108957       0          0           NA  0.00000      0         0
110669       0         NA           NA  0.00000      0         0
110674       0          0           NA       NA      0         0
110795       0          0           NA  0.00000      0        NA
111041      NA          0           NA  0.00000      0         0
112386       0         NA           NA       NA      0         0
112507       0         NA           NA  0.00000      0        NA
112512       0          0           NA       NA      0        NA
112753      NA         NA           NA  0.00000      0         0
112758      NA          0           NA       NA      0         0
112879      NA          0           NA  0.00000      0        NA
114224       0         NA           NA       NA      0        NA
114470      NA         NA           NA       NA      0         0
114591      NA         NA           NA  0.00000      0        NA
114596      NA          0           NA       NA      0        NA
```

10 Model Selection

We can select the best model by sorting each subset (e.g. by AICc) or by using visualization methods.

```
> select.best.models(PGLSi, using='AICc')

    n q  qXn  rwGsm model.no         R2     R2.adj      AIC     AICc      BIC         AICw
53 53 5 5X53  98558        3 0.16007974 0.07072652 345.9441 347.7702 357.7659 4.671129e-01
54 54 4 4X54 100396        8 0.15679638 0.08796344 349.0633 350.3133 359.0083 1.309743e-01
55 55 3 3X55 102113       23 0.14927592 0.09923333 353.2706 354.0706 361.2999 2.001305e-02
56 56 2 2X56 103825       44 0.11006887 0.07648657 358.7657 359.2273 364.8418 1.518986e-03
57 57 2 2X57 105909       53 0.05921713 0.02437332 368.4776 368.9304 374.6067 1.187259e-05
58 58 4 4X58 108957       12 0.10716908 0.03978561 376.1636 377.3174 386.4658 1.791920e-07
59 59 3 3X59 110795       24 0.10155643 0.05255042 379.4725 380.2132 387.7826 4.212260e-08
60 60 2 2X60 112512       43 0.10755127 0.07623728 382.9089 383.3374 389.1919 8.832764e-09
61 61 2 2X61 114224       46 0.05993341 0.02751732 391.0582 391.4792 397.3908 1.507064e-10
           BICw
53 3.472579e-01
54 1.865819e-01
55 5.932599e-02
56 1.009574e-02
57 7.650832e-05
58 2.034870e-07
59 1.053416e-07
60 5.206880e-08
61 8.633999e-10
```

Plotting the coefficients of determination versus the AIC values allows selection of high-performing models for inspection and reporting.

```
> plot.pgls.iters(PGLSi)
```

```
> sdevs.objs <- get.pgls.coefs(PGLSi$fits, est='t value')
> coefs.objs <- get.pgls.coefs(PGLSi$fits, est='Estimate')
```

```
> report.vect <- sapply(1:length(PGLSi$fits), function(i) fit.1ln.rprt(PGLSi$fits[[i]], rtrn.line=FALSE, mn=i)

1 +mass(0.108) +group(0.162) +infan(0.204) arbore(0.452)  |  DPL(0.93) noctur(0.598) R2adj: 0.051 AICc: 350.42
2 +mass(0.097) +group(0.156) +infan(0.168) arbore(0.412)  |  DPL(0.988) R2adj: 0.069 AICc: 352.85
3 +mass(0.102) +group(0.133) +infan(0.153) arbore(0.369)  |  noctur(0.598) R2adj: 0.071 AICc: 347.77
4 +group(0.125) +mass(0.134) +infan(0.261)  |  noctur(0.604) DPL(0.583) R2adj: 0.071 AICc: 353.53
5 +group(0.095) +mass(0.133) arbore(0.917)  |  noctur(0.636) DPL(0.514) R2adj: 0.029 AICc: 379.33
6 +mass(0.072) +infan(0.11) arbore(0.516) DPL(0.612)  |  noctur(0.581) R2adj: 0.032 AICc: 355.07
7 +group(0.131) +infan(0.181) arbore(0.66)  |  DPL(0.851) -noctu(0.31) R2adj: 0.031 AICc: 356.65
8 +mass(0.093) +group(0.12) +infan(0.131) arbore(0.348)  |   R2adj: 0.088 AICc: 350.31
9 +group(0.123) +mass(0.125) +infan(0.219)  |  DPL(0.614) R2adj: 0.086 AICc: 356.21
10 +mass(0.14) +group(0.144) +infan(0.17)  |  noctur(0.649) R2adj: 0.084 AICc: 351.34
11 +group(0.091) +mass(0.121) arbore(0.861)  |  DPL(0.536) R2adj: 0.042 AICc: 382.18
12 +group(0.12) +mass(0.131) arbore(0.714)  |  noctur(0.677) R2adj: 0.04 AICc: 377.32
13 +group(0.064) +mass(0.124)  |  noctur(0.631) DPL(0.427) R2adj: 0.06 AICc: 381.97
14 +mass(0.064) +infan(0.089) arbore(0.485) DPL(0.549)  |   R2adj: 0.05 AICc: 357.63
15 +mass(0.069) +infan(0.126) arbore(0.649)  |  noctur(0.534) R2adj: 0.046 AICc: 352.83
16 +mass(0.082) +infan(0.135) DPL(0.863)  |  noctur(0.579) R2adj: 0.044 AICc: 358.72
17 +mass(0.086)  |  DPL(0.923) arbore(0.914) noctur(0.605) R2adj: -0.004 AICc: 385.02
18 +group(0.13) +infan(0.142) arbore(0.64)  |  DPL(0.942) R2adj: 0.033 AICc: 360.08
19 +group(0.113) +infan(0.123) arbore(0.551)  |  -noctu(0.312) R2adj: 0.05 AICc: 354.16
20 +group(0.095) +infan(0.203)  |  DPL(0.616) -noctu(0.302) R2adj: 0.058 AICc: 359.51
21 +group(0.077)  |  arbore(0.891) DPL(0.484) -noctu(0.305) R2adj: 0.011 AICc: 385.75
22 +infan(0.093) DPL(0.647) arbore(0.784)  |  -noctu(0.315) R2adj: 0.005 AICc: 361.81
23 +mass(0.13) +group(0.134) +infan(0.141)  |   R2adj: 0.099 AICc: 354.07
24 +group(0.113) +mass(0.121) arbore(0.662)  |   R2adj: 0.053 AICc: 380.21
```

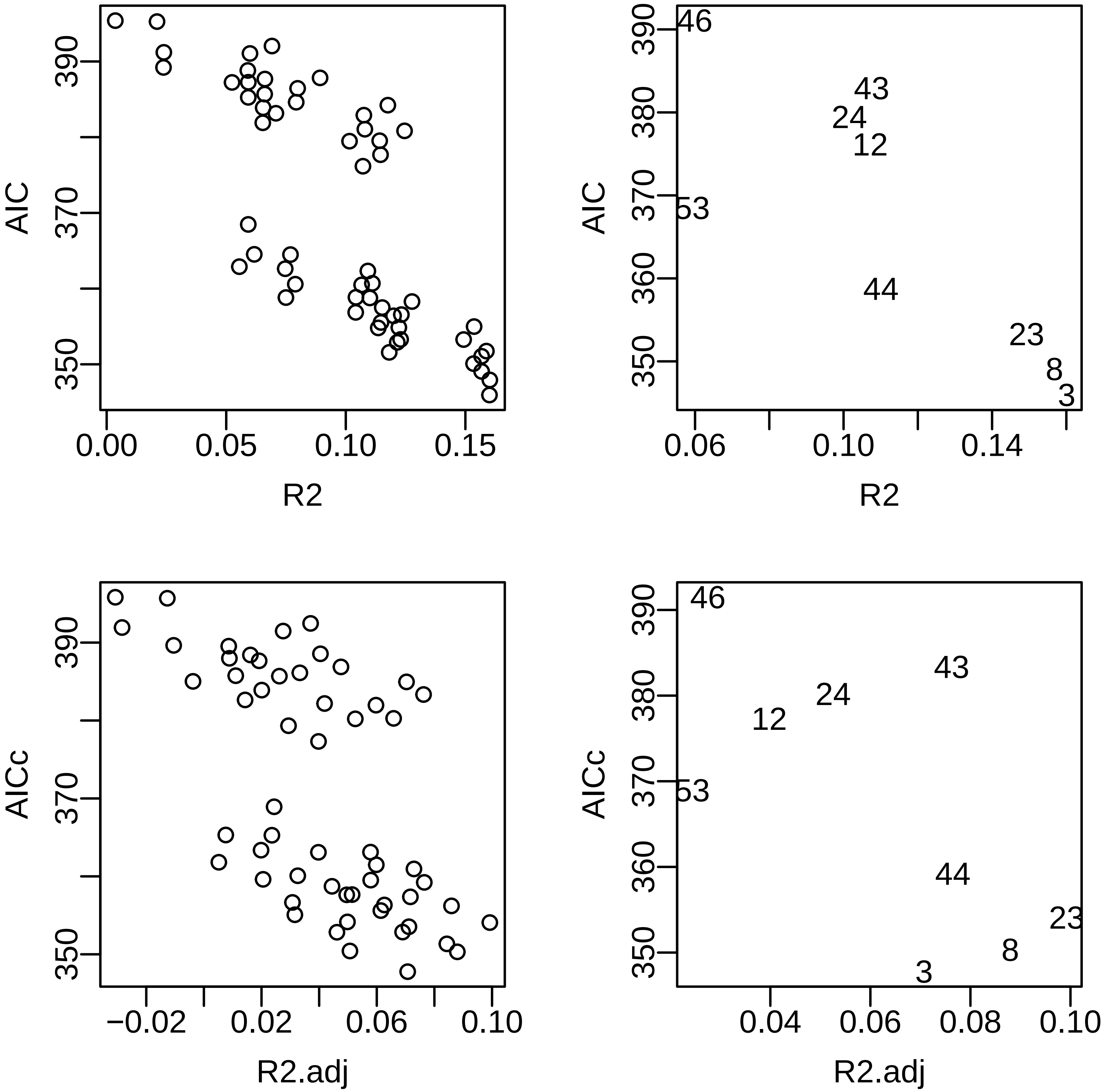

Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of each streak. Therefore, minimizing AIC tends to also maximize the coefficient of determination, but not necessarily. This four panel plot looks at correct and adjusted versions of each model assessment measure. All points are scaled by subdataset sample size by default if 'n' is used in grouping.

```
25 +group(0.062) +mass(0.117)  |  DPL(0.428) R2adj: 0.07 AICc: 384.96
26 +group(0.091) +mass(0.138)  |  noctur(0.695) R2adj: 0.066 AICc: 380.27
27 +mass(0.062) +infan(0.108) arbore(0.651)  |   R2adj: 0.061 AICc: 355.61
28 +mass(0.075) +infan(0.111) DPL(0.805)  |   R2adj: 0.06 AICc: 361.48
29 +mass(0.071) +infan(0.13)  |  noctur(0.552) R2adj: 0.063 AICc: 356.33
30 +mass(0.077)  |  DPL(0.964) arbore(0.959) R2adj: 0.009 AICc: 387.99
31 +mass(0.083)  |  arbore(0.941) noctur(0.607) R2adj: 0.014 AICc: 382.64
32 +mass(0.068)  |  DPL(0.938) noctur(0.585) R2adj: 0.016 AICc: 388.41
33 +group(0.102) +infan(0.103) arbore(0.569)  |   R2adj: 0.051 AICc: 357.66
34 +group(0.095) +infan(0.16)  |  DPL(0.698) R2adj: 0.058 AICc: 363.11
35 +group(0.099) +infan(0.129)  |  noctur(0.324) R2adj: 0.072 AICc: 357.37
36 +group(0.075)  |  arbore(0.912) DPL(0.534) R2adj: 0.009 AICc: 389.54
37 +group(0.101) arbore(0.894)  |  noctur(0.328) R2adj: 0.02 AICc: 383.9
38 ++grou(0.046)  |  DPL(0.451) -noctu(0.292) R2adj: 0.04 AICc: 388.56
39 +infan(0.071) DPL(0.559) arbore(0.775)  |   R2adj: 0.008 AICc: 365.31
40 +infan(0.102) arbore(0.976)  |  -noctu(0.285) R2adj: 0.021 AICc: 359.62
41 +infan(0.092) DPL(0.726)  |  -noctu(0.303) R2adj: 0.024 AICc: 365.28
42   |  DPL(0.93) arbore(0.678) -noctu(0.299) R2adj: -0.028 AICc: 391.93
43 +group(0.088) +mass(0.131)  |   R2adj: 0.076 AICc: 383.34
44 +mass(0.062) +infan(0.11)  |   R2adj: 0.076 AICc: 359.23
45 +mass(0.075)  |  arbore(0.971) R2adj: 0.026 AICc: 385.69
46 +mass(0.062)  |  DPL(0.964) R2adj: 0.028 AICc: 391.48
47 +mass(0.064)  |  noctur(0.586) R2adj: 0.033 AICc: 386.12
48 +group(0.086) +infan(0.105)  |   R2adj: 0.073 AICc: 360.95
49 +group(0.092) arbore(0.891)  |   R2adj: 0.019 AICc: 387.66
50 ++grou(0.044)  |  DPL(0.495) R2adj: 0.037 AICc: 392.46
51 +group(0.061)  |  noctur(0.324) R2adj: 0.048 AICc: 386.88
52 +infan(0.085)  |  arbore(0.964) R2adj: 0.02 AICc: 363.36
53 +infan(0.071) DPL(0.619)  |   R2adj: 0.024 AICc: 368.93
54 +infan(0.096)  |  -noctu(0.272) R2adj: 0.04 AICc: 363.09
55 DPL(0.995)  |  arbore(0.688) R2adj: -0.031 AICc: 395.82
56   |  arbore(0.671) -noctu(0.297) R2adj: -0.01 AICc: 389.64
57 DPL(0.923)  |  -noctu(0.279) R2adj: -0.013 AICc: 395.69
```

```
> par(mar=c(5,5,3,3))
> plot.pgls.R2AIC(PGLSi$optim)
```

11 Coefficient Plotting

Finally, the resulting model fits from the PGLS runs can be be plotted out horizontally as distributions so the influence of each ecological predictor variable can be compared.

```
> par.old <- par(mar=c(5,8,1,4),mfrow=c(2,1))
> sparge.modsel(sdevs.objs, R2x=7, xlab='t value')
> sparge.modsel(coefs.objs, R2x=7, xlab='Estimate')
```

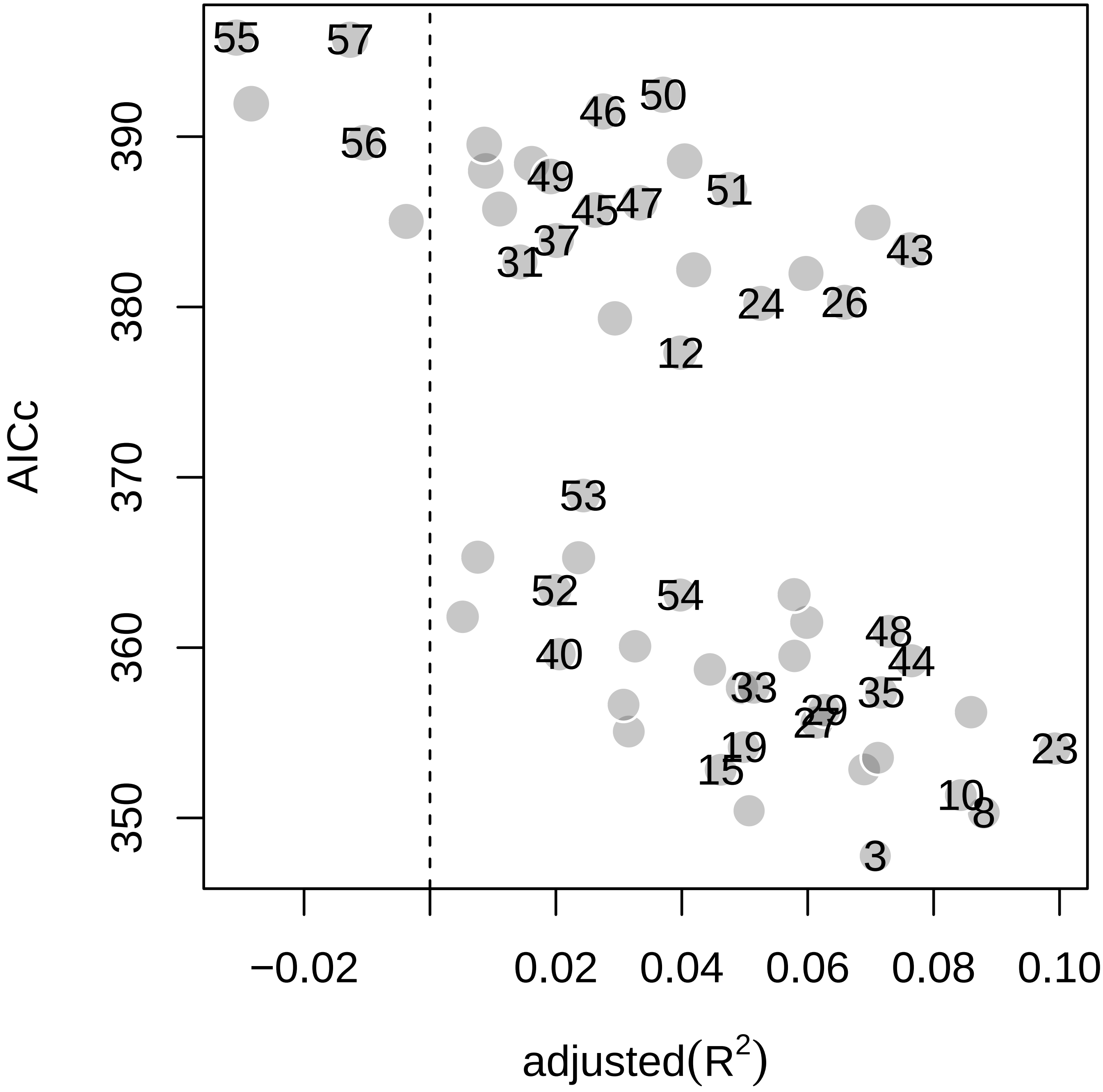

Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is recommended that all "best" models selected from each sub-dataset should be inspected or reported somehow, such as in the form of the sparge plot below.

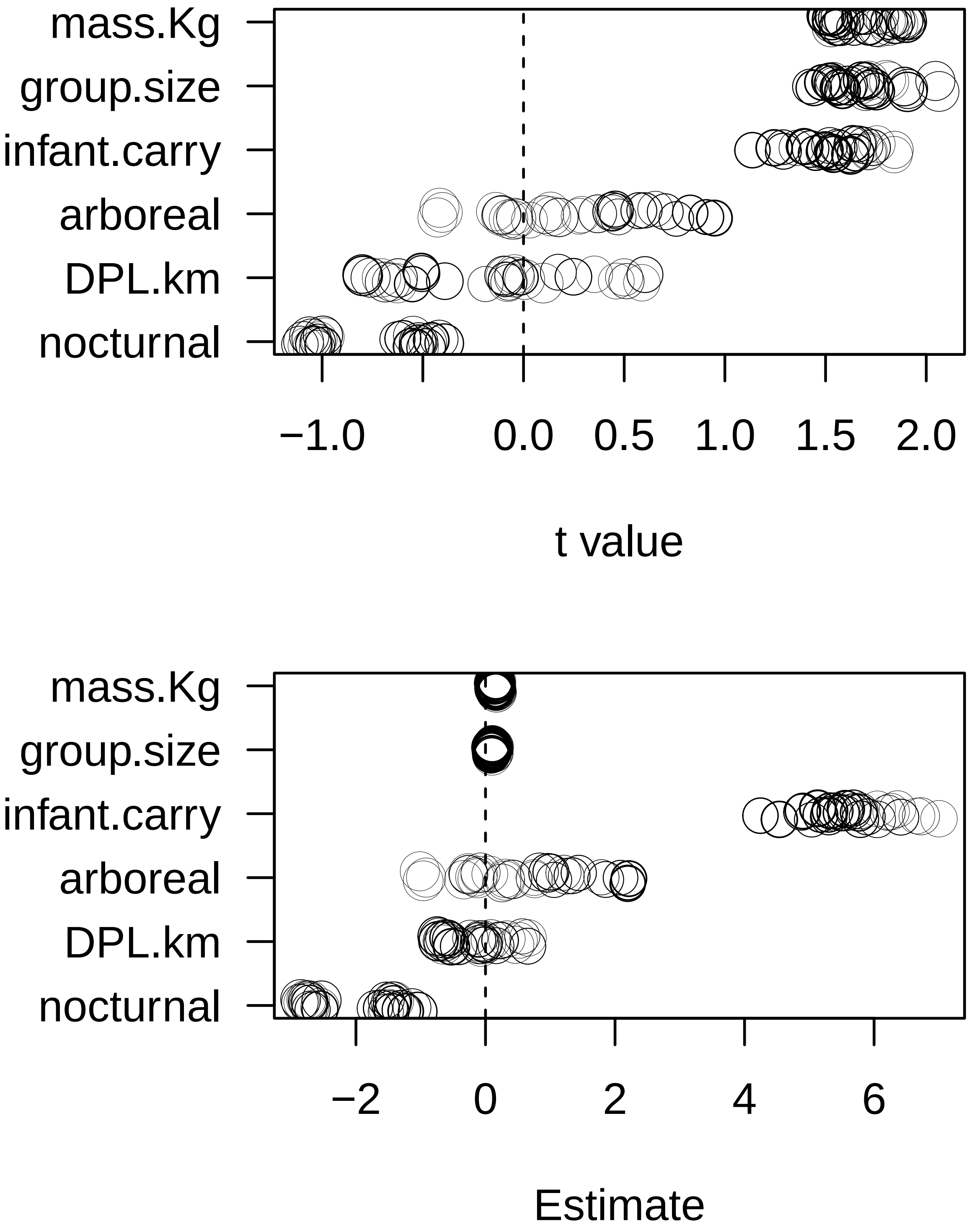

Figure 3: These horizontal parameter distribution dots plots demonstrate how the (t-values of) coefficients from all models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting of each model, point sizes represent underlying sample sizes and circle thickness corresponds to coefficient of determination values. Note that arboreal locomotor targeting and staturally protective (e.g. infant fur-cling carrying) factors play key roles driving orbital convergence.

References

[1] Isbell, L. (2006) ‘Snakes as agents of evolutionary change in primate brains’ *Journal of Human Evolution*, 51(1) pp. 1–35.

[2] Cartmill, M. (1972) ‘Arboreal Adaptations and the Origin of the Order Primates’ In *The Function and Evolutionary Biology of Primates*, pp. 97–122, Aldine-Atherton.

[3] Szalay, Frederick S., and Marian Dagosto. ‘Evolution of Hallucial Grasping in the Primates’ *Journal of Human Evolution*, 17(1–2) pp. 1–33. `https://doi.org/10.1016/0047-2484(88)90047-4`.

[4] Wheeler, B.C., B.J. Bradely, and J.M. Kamilar. (2011) ‘Predictors of Orbital Convergence in Primates: A Test of the Snake Detection Hypothesis of Primate Evolution’ *Journal of Human Evolution*, 61, pp. 233–42.

[5] Pagel, M. (1999) ‘Inferring the historical patterns of biological evolution’ *Nature*, 401, pp. 877–884. Available at: `https://doi.org/10.1038/44766`.

[6] Grueber, C.E., Nakagawa, .R.J., and Jamieson, I.G. (2011) ‘Multimodel inference in ecology and evolution: challenges and solutions’ *Journal of Evolutionary Biology*, 24, pp. 699-711. Available at: `https://doi.org/10.1111/j.1420-9101.2010.02210.x`

[7] Stephens, P.A., Buskirk, S.W., and del Rio, C.M. (2007) ‘Inference in ecology and evolution’ *Trends in Ecology and Evolution*, 22(4). Available at: `https://doi.org/10.1016/j.tree.2006.12.003`.

[8] Symonds, M.R.E., and Moussalli, A. (2011) ‘A brief guide to model selection, multimodel inference and model averaging in behavioral ecology using Akaike’s information criterion’ *Behavioral Ecology and Sociobiology*, 65, pp. 13–21. Available at: `https://doi.org/10.1007/s00265-010-1037-6`.

[9] Johnson, J.B., Omland, K.S. (2004) ‘Model selection in ecology and evolution’ *Trends in Ecology and Evolution*, 19(2). Available at: `https://doi.org/10.1016/j.tree.2003.10.013`.

[10] Revell, L.J. (2014) phytools: An R package for phylogenetic comparative biology (and other things). [CRAN]. Available at: `http://cran.r-project.org/package=phytools`.

[11] Garamszegi L.Z. (2011) ‘Information-theoretic approaches to statistical analysis in behavioral ecology: an introduction’ *Behavioral Ecology and Sociobiology*, 65, pp. 1–11. Available at: `https://doi.org/10.1007/s00265-010-1028-7`.

[12] Burnham, P.B. and Anderson, D.R. (2000) *‘Model Selection and Inference: A Practical Information-Theoretic Approach’*

[13] Cade, B.S. (2015) ‘Model averaging and muddled multimodel inferences’ *Ecology*, 96(9), pp. 2370–2382. Available at: `https://doi.org/10.1890/14-1639.1`.

[14] Rowe, N. and Meyers, M. (2017) *All the World’s Primates*. Charlestown, RI: Pogonias Press.

[15] Springer, M.S., et.al. (2012) ‘Re-evaluating primate monogamy’ *PLoS ONE* 7(11) p. e49521 `http://doi.org/10.1371/journal.pone.0049521`.

[16] Schruth, D.M. (2019) ‘Primate Locomotor Activity’ *The Center for Open Science*. Available at: `https://osf.io/cd68q/`.

[17] Schruth, D.M. (2021a) Arboreal locomotion and trophic security at the dawn of Euprimate vision. EcoEvoRxiv. Available at: `http://doi.org/10.32942/osf.io/d6wk2`.

[18] Schruth, D.M. (2021b) ‘A global variable-permutation based approach for estimating tree transformation parameters used in phylogenetically controlled multivariate regression’ *Protocols.io*. Available at: `http://doi.org/10.17504/protocols.io.bzdhp236`.

[19] Ross, C. (2001) ‘Park or Ride? Evolution of Infant Carrying in Primates’ *International Journal of Primatology*, 22(5), pp. 749–771. Available at: `https://doi.org/10.1023/A:1012065332758`.