Ecological factors influencing primate vision conformation: a phylogenetic regression workflow for the *mmodely* R-package (Version 0.2.2)

David M. Schruth
dschruth@anthropoidea.org

November 5, 2022

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 [15] to targeting in hunting [2] or locomotion [3]. Here, using the *mmodely* package on primate data for locomotion [14] and vision [4], I demonstrate how the origins of primate cranial morphology can be elucidated via several ecological variables from numerous datasets [12]. Model averaging [MA] [8] and model selection [MS] [9] results primarily highlight arboreal locomotor targeting and trophic security [15] 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 [13] 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 
-30.495  -5.996  17.023  34.907  54.965 

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 3 and 16 DF,  p-value: 0.02045 

> 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
-30.495  -5.996  17.023  34.907  54.965

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 3 and 16 DF,  p-value: 0.02045
[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 1.4591898 1.2008103
distributions of optimization parameters:
       n           n.1          q              rwGsm           model.no          R2
 Min.   :63   Min.   :63   Min.   :2.000   Min.   :117909   Min.   : 1.0   Min.   :0.04241
```

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

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 [16] 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 1.4591898 1.2008103

>
```

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. By default *mmodely* uses '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' 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 1.4591898 1.2008103
distributions of optimization parameters:
       n              n.1              q              rwGsm           model.no
 Min.   :53.00   Min.   :53.00   Min.   :2.000   Min.   : 97766   Min.   : 1
 1st Qu.:55.00   1st Qu.:55.00   1st Qu.:2.000   1st Qu.:101697   1st Qu.:15
 Median :56.00   Median :56.00   Median :3.000   Median :104196   Median :29
 Mean   :57.11   Mean   :57.11   Mean   :3.263   Mean   :106224   Mean   :29
 3rd Qu.:60.00   3rd Qu.:60.00   3rd Qu.:4.000   3rd Qu.:111627   3rd Qu.:43
 Max.   :61.00   Max.   :61.00   Max.   :6.000   Max.   :114595   Max.   :57
       R2                R2.adj              AIC             AICc            AICw
 Min.   :0.003598   Min.   :-0.03076   Min.   :336.1   Min.   :338.0   Min.   :0.0000000
 1st Qu.:0.076617   1st Qu.: 0.03391   1st Qu.:351.2   1st Qu.:352.4   1st Qu.:0.0000000
 Median :0.115857   Median : 0.06376   Median :363.4   Median :364.2   Median :0.0000013
 Mean   :0.122808   Mean   : 0.06855   Mean   :365.4   Mean   :366.3   Mean   :0.0175439
 3rd Qu.:0.159438   3rd Qu.: 0.10326   3rd Qu.:379.9   3rd Qu.:380.5   3rd Qu.:0.0004650
 Max.   :0.236042   Max.   : 0.15704   Max.   :395.0   Max.   :395.4   Max.   :0.6366126
```

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' as mentioned in the preceding section.

```
> w.means.pds <- average.fit.models(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim, by='rwGsm')
> #
> 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 
  0.18050188   0.10764875   5.79281563   0.56623437  -0.04558871  -2.60964312 

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

       mass.Kg group.size infant.carry arboreal   DPL.km nocturnal
97766  0.19055    0.10661      5.27955  2.14497  0.17321  -2.39842
99479  0.19291        NaN      5.93242  1.15125  0.82431  -2.56628
99515  0.19149    0.10508      5.50533  2.02454  0.27554       NaN
99948      NaN    0.11290      5.43312  1.43427 -0.20284  -2.69142
100265 0.17121    0.09306      5.15198      NaN -0.31424  -2.26522
101228 0.19403        NaN      6.19077  1.07344  0.91101       NaN
101661     NaN        NaN      6.07733  0.24428  0.55188  -2.86270
101697     NaN    0.11192      5.68042  1.30012 -0.08366       NaN
101978 0.18122        NaN      5.81765      NaN  0.38797  -2.47447
102014 0.17310    0.09240      5.36372      NaN -0.20310       NaN
102447     NaN    0.10224      5.37470      NaN -0.49337  -2.56634
103410     NaN        NaN      6.34930  0.13231  0.65686       NaN
103727 0.18338        NaN      6.04901      NaN  0.48008       NaN
104160     NaN        NaN      6.13867      NaN  0.35098  -2.79525
104196     NaN    0.10217      5.61760      NaN -0.36599       NaN
105909     NaN        NaN      6.72348      NaN  0.46017       NaN
108165 0.17288    0.11027          NaN  0.76684 -0.59839  -2.43827
109878 0.17993        NaN          NaN -0.19812  0.00003  -2.63726
109914 0.17409    0.10921          NaN  0.64746 -0.51530       NaN
110347     NaN    0.11443          NaN  0.27282 -0.70639  -2.71439
110664 0.16686    0.10608          NaN      NaN -0.68986  -2.34011
111627 0.18143        NaN          NaN -0.31944  0.07689       NaN
112060     NaN        NaN          NaN -0.81907 -0.06140  -2.91252
112096     NaN    0.11376          NaN  0.14293 -0.61509       NaN
112377 0.18233        NaN          NaN      NaN  0.00485  -2.57899
112413 0.16885    0.10551          NaN      NaN -0.59595       NaN
112846     NaN    0.11264          NaN      NaN -0.70910  -2.63068
113809     NaN        NaN          NaN -0.93885  0.02519       NaN
114126 0.18377        NaN          NaN      NaN  0.09167       NaN
114559     NaN        NaN          NaN      NaN  0.07534  -2.88197
114595     NaN    0.12410          NaN      NaN -0.60455       NaN
```

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 n.1 q  qXn  rwGsm model.no          R2       R2.adj      AIC     AICc         AICw
97766  53  53 5 5X53  97766        3 0.235638877  0.154323864 336.1358 337.9619 6.366126e-01
100265 54  54 4 4X54 100265       10 0.220662154  0.157042738 341.2041 342.4541 6.735960e-02
99515  54  54 4 4X54  99515        8 0.210940483  0.146527461 341.4751 342.7251 5.882441e-02
99479  54  54 4 4X54  99479       15 0.159437870  0.090820553 344.7822 346.0322 1.125714e-02
102014 55  55 3 3X55 102014       23 0.199073532  0.151960211 346.3426 347.1426 6.461097e-03
101978 55  55 3 3X55 101978       29 0.157261880  0.107689049 349.0357 349.8357 1.680756e-03
101228 55  55 3 3X55 101228       27 0.136145599  0.085330634 349.9976 350.7976 1.039042e-03
99948  54  54 4 4X54  99948       19 0.139274356  0.069011039 349.8797 351.1297 8.800952e-04
103727 56  56 2 2X56 103727       44 0.135863948  0.103255040 354.1488 354.6104 1.544188e-04
102447 55  55 3 3X55 102447       35 0.132566639  0.081541148 354.4851 355.2851 1.102008e-04
101697 55  55 3 3X55 101697       33 0.115768848  0.063755251 355.1295 355.9295 7.984646e-05
101661 55  55 3 3X55 101661       40 0.075663838  0.021291123 357.4003 358.2003 2.565367e-05
104196 56  56 2 2X56 104196       48 0.111577420  0.078052039 359.5935 360.0550 1.014876e-05
104160 56  56 2 2X56 104160       54 0.076617203  0.041772569 361.5871 362.0486 3.745508e-06
103410 56  56 2 2X56 103410       52 0.052861706  0.017120639 362.5986 363.0602 2.258673e-06
108165 58  58 4 4X58 108165       12 0.172692586  0.110254291 366.7325 367.8864 2.022363e-07
105909 57  57 2 2X57 105909       53 0.059217511  0.024373715 368.4775 368.9304 1.199919e-07
110664 59  59 3 3X59 110664       26 0.167774419  0.122380297 371.1545 371.8953 2.724807e-08
109914 59  59 3 3X59 109914       24 0.149301004  0.102899241 372.2453 372.9860 1.579370e-08
109878 59  59 3 3X59 109878       31 0.097025764  0.047772624 375.5944 376.3352 2.959539e-09
112413 60  60 2 2X60 112413       43 0.146292678  0.116338035 376.5542 376.9828 2.140873e-09
112377 60  60 2 2X60 112377       47 0.095321550  0.063578446 379.8656 380.2942 4.088274e-10
110347 59  59 3 3X59 110347       37 0.089066266  0.039378972 379.8072 380.5479 3.601133e-10
111627 60  60 2 2X60 111627       45 0.074955994  0.042498310 380.9956 381.4242 2.323562e-10
112846 60  60 2 2X60 112846       51 0.086873908  0.054834396 384.1559 384.5844 4.785392e-11
112096 60  60 2 2X60 112096       49 0.066657278  0.033908410 385.2425 385.6711 2.779406e-11
114126 61  61 2 2X61 114126       46 0.074302260  0.042381648 387.2179 387.6389 1.039057e-11
112060 60  60 2 2X60 112060       56 0.025363707 -0.008834058 387.6229 388.0514 8.454054e-12
114595 61  61 2 2X61 114595       50 0.072964399  0.040997654 391.0541 391.4752 1.526189e-12
114559 61  61 2 2X61 114559       57 0.021834406 -0.011895442 394.1127 394.5338 3.307112e-13
113809 61  61 2 2X61 113809       55 0.003597818 -0.030760878 395.0118 395.4329 2.109658e-13
```

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(0.04) +group(0.088) +infan(0.133) arbore(0.365) DPL(0.877)  |  noctur(0.329) R2adj: 0.136 AICc: 340.6
2 ++mass(0.038) +group(0.096) +infan(0.111) arbore(0.373) DPL(0.804)  |   R2adj: 0.13 AICc: 345.19
3 ++mass(0.038) +group(0.057) +infan(0.116) arbore(0.343)  |  -noctu(0.313) R2adj: 0.154 AICc: 337.96
4 +mass(0.054) +group(0.096) +infan(0.164)  |  DPL(0.75) noctur(0.33) R2adj: 0.141 AICc: 344.88
5 ++grou(0.042) +mass(0.056) arbore(0.868)  |  DPL(0.529) -noctu(0.294) R2adj: 0.1 AICc: 369.93
6 ++mass(0.039) +infan(0.073) DPL(0.443) arbore(0.461)  |  noctur(0.334) R2adj: 0.083 AICc: 347.9
7 +group(0.076) +infan(0.18) arbore(0.631)  |  DPL(0.866) -noctu(0.298) R2adj: 0.05 AICc: 353.63
8 ++mass(0.037) +group(0.058) +infan(0.103) arbore(0.377)  |   R2adj: 0.147 AICc: 342.73
9 ++mass(0.051) +group(0.105) +infan(0.137)  |  DPL(0.835) R2adj: 0.136 AICc: 349.52
10 +mass(0.053) +group(0.085) +infan(0.112)  |  noctur(0.342) R2adj: 0.157 AICc: 342.45
11 ++grou(0.046) +mass(0.055) arbore(0.892)  |  DPL(0.587) R2adj: 0.091 AICc: 375.05
12 +group(0.051) +mass(0.056) arbore(0.665)  |  -noctu(0.313) R2adj: 0.11 AICc: 367.89
```

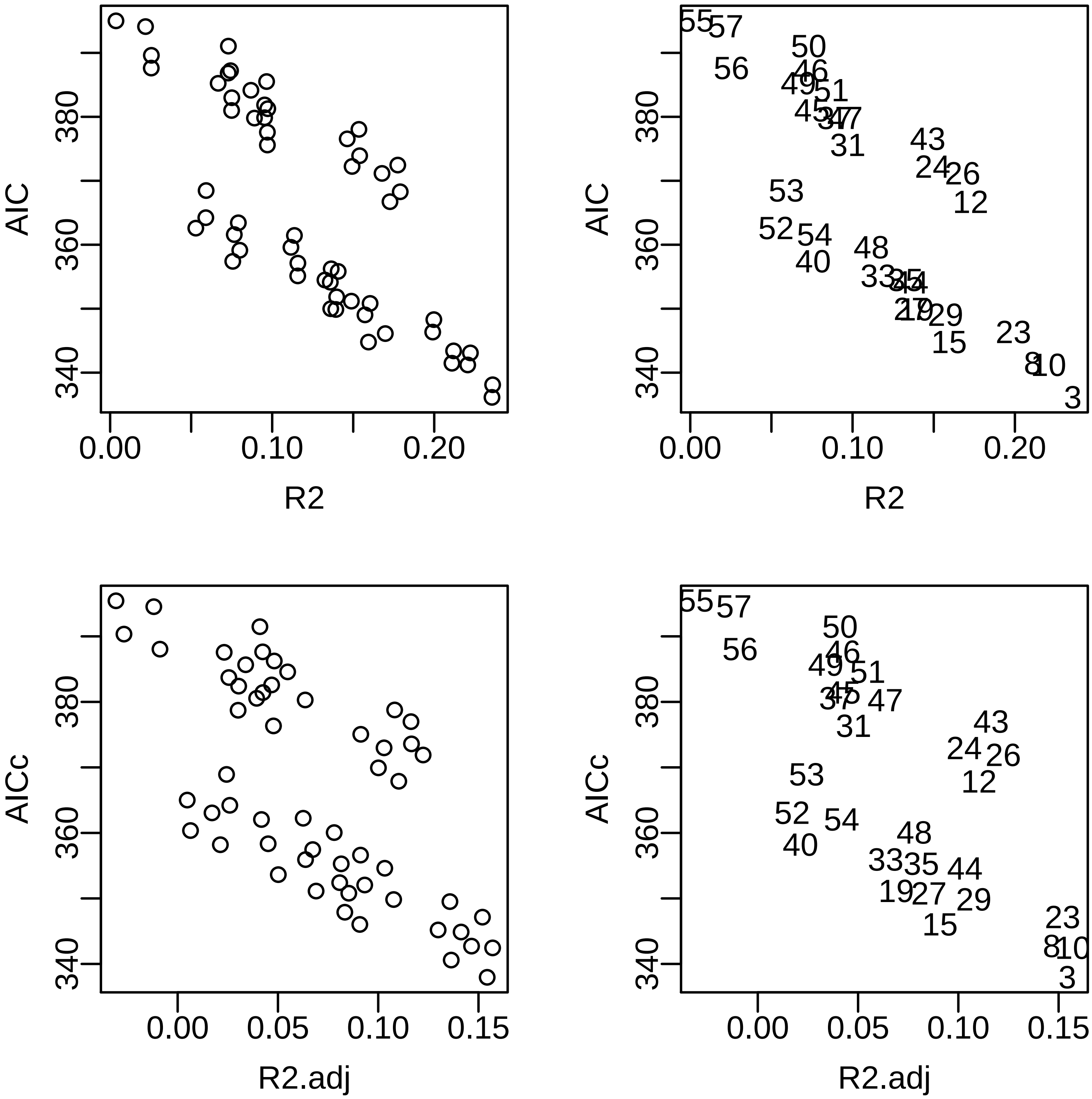

Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of 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.

```
13 ++grou(0.038) +mass(0.052)  |  DPL(0.428) -noctu(0.299) R2adj: 0.117 AICc: 373.59
14 ++mass(0.037) +infan(0.059) DPL(0.391) arbore(0.465)  |   R2adj: 0.081 AICc: 352.41
15 ++mass(0.039) +infan(0.099) arbore(0.694)  |  -noctu(0.286) R2adj: 0.091 AICc: 346.03
16 ++mass(0.048) +infan(0.083) DPL(0.666)  |  noctur(0.335) R2adj: 0.093 AICc: 352.05
17 +mass(0.052) DPL(1)  |  arbore(0.932) -noctu(0.291) R2adj: 0.03 AICc: 378.73
18 +group(0.082) +infan(0.15) arbore(0.64)  |  DPL(0.944) R2adj: 0.045 AICc: 358.35
19 +group(0.064) +infan(0.128) arbore(0.525)  |  -noctu(0.299) R2adj: 0.069 AICc: 351.13
20 +group(0.077) +infan(0.187)  |  DPL(0.639) -noctu(0.3) R2adj: 0.067 AICc: 357.46
21 ++grou(0.044)  |  arbore(0.939) DPL(0.486) -noctu(0.274) R2adj: 0.03 AICc: 382.4
22 +infan(0.096) DPL(0.629) arbore(0.772)  |  -noctu(0.303) R2adj: 0.006 AICc: 360.37
23 ++mass(0.05) +group(0.084) +infan(0.099)  |   R2adj: 0.152 AICc: 347.14
24 +group(0.052) +mass(0.054) arbore(0.714)  |   R2adj: 0.103 AICc: 372.99
25 ++grou(0.041) ++mass(0.05)  |  DPL(0.492) R2adj: 0.108 AICc: 378.77
26 +group(0.053) +mass(0.058)  |  noctur(0.33) R2adj: 0.122 AICc: 371.9
27 ++mass(0.037) +infan(0.087) arbore(0.742)  |   R2adj: 0.085 AICc: 350.8
28 ++mass(0.046) +infan(0.068) DPL(0.589)  |   R2adj: 0.091 AICc: 356.62
29 ++mass(0.038) +infan(0.09)  |  -noctu(0.3) R2adj: 0.108 AICc: 349.84
30 ++mass(0.05) DPL(0.934)  |  arbore(0.913) R2adj: 0.026 AICc: 383.72
31 ++mass(0.05)  |  arbore(0.924) -noctu(0.285) R2adj: 0.048 AICc: 376.34
32 ++mass(0.042) DPL(0.995)  |  -noctu(0.295) R2adj: 0.047 AICc: 382.59
33 +group(0.063) +infan(0.112) arbore(0.569)  |   R2adj: 0.064 AICc: 355.93
34 +group(0.083) +infan(0.155)  |  DPL(0.725) R2adj: 0.063 AICc: 362.24
35 +group(0.076) +infan(0.119)  |  -noctu(0.319) R2adj: 0.082 AICc: 355.29
36 ++grou(0.047)  |  arbore(0.913) DPL(0.543) R2adj: 0.023 AICc: 387.57
37 +group(0.056) arbore(0.836)  |  -noctu(0.294) R2adj: 0.039 AICc: 380.55
38 ++grou(0.036)  |  DPL(0.442) -noctu(0.275) R2adj: 0.048 AICc: 386.24
39 +infan(0.078) DPL(0.561) arbore(0.777)  |   R2adj: 0.005 AICc: 365.02
40 +infan(0.107) arbore(0.977)  |  -noctu(0.271) R2adj: 0.021 AICc: 358.2
41 +infan(0.088) DPL(0.708)  |  -noctu(0.304) R2adj: 0.026 AICc: 364.22
42   |  DPL(0.95) arbore(0.718) -noctu(0.268) R2adj: -0.027 AICc: 390.35
43 +group(0.052) +mass(0.055)  |   R2adj: 0.116 AICc: 376.98
44 ++mass(0.034) +infan(0.078)  |   R2adj: 0.103 AICc: 354.61
45 ++mass(0.047)  |  arbore(0.87) R2adj: 0.042 AICc: 381.42
46 ++mass(0.04) DPL(0.912)  |   R2adj: 0.042 AICc: 387.64
47 ++mass(0.037)  |  -noctu(0.287) R2adj: 0.064 AICc: 380.29
48 +group(0.072) +infan(0.105)  |   R2adj: 0.078 AICc: 360.06
49 +group(0.056) arbore(0.891)  |   R2adj: 0.034 AICc: 385.67
50 ++grou(0.038)  |  DPL(0.51) R2adj: 0.041 AICc: 391.48
51 ++grou(0.049)  |  -noctu(0.305) R2adj: 0.055 AICc: 384.58
52 +infan(0.093)  |  arbore(0.963) R2adj: 0.017 AICc: 363.06
53 +infan(0.071) DPL(0.619)  |   R2adj: 0.024 AICc: 368.93
54 +infan(0.092)  |  -noctu(0.273) R2adj: 0.042 AICc: 362.05
55 DPL(0.98)  |  arbore(0.697) R2adj: -0.031 AICc: 395.43
56   |  arbore(0.705) -noctu(0.265) R2adj: -0.009 AICc: 388.05
57 DPL(0.93)  |  -noctu(0.269) R2adj: -0.012 AICc: 394.53
```

```
> 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))
> distro.dots.modsel(sdevs.objs, R2x=7, xlab='t value')
> distro.dots.modsel(coefs.objs, R2x=7, xlab='Estimate')
```

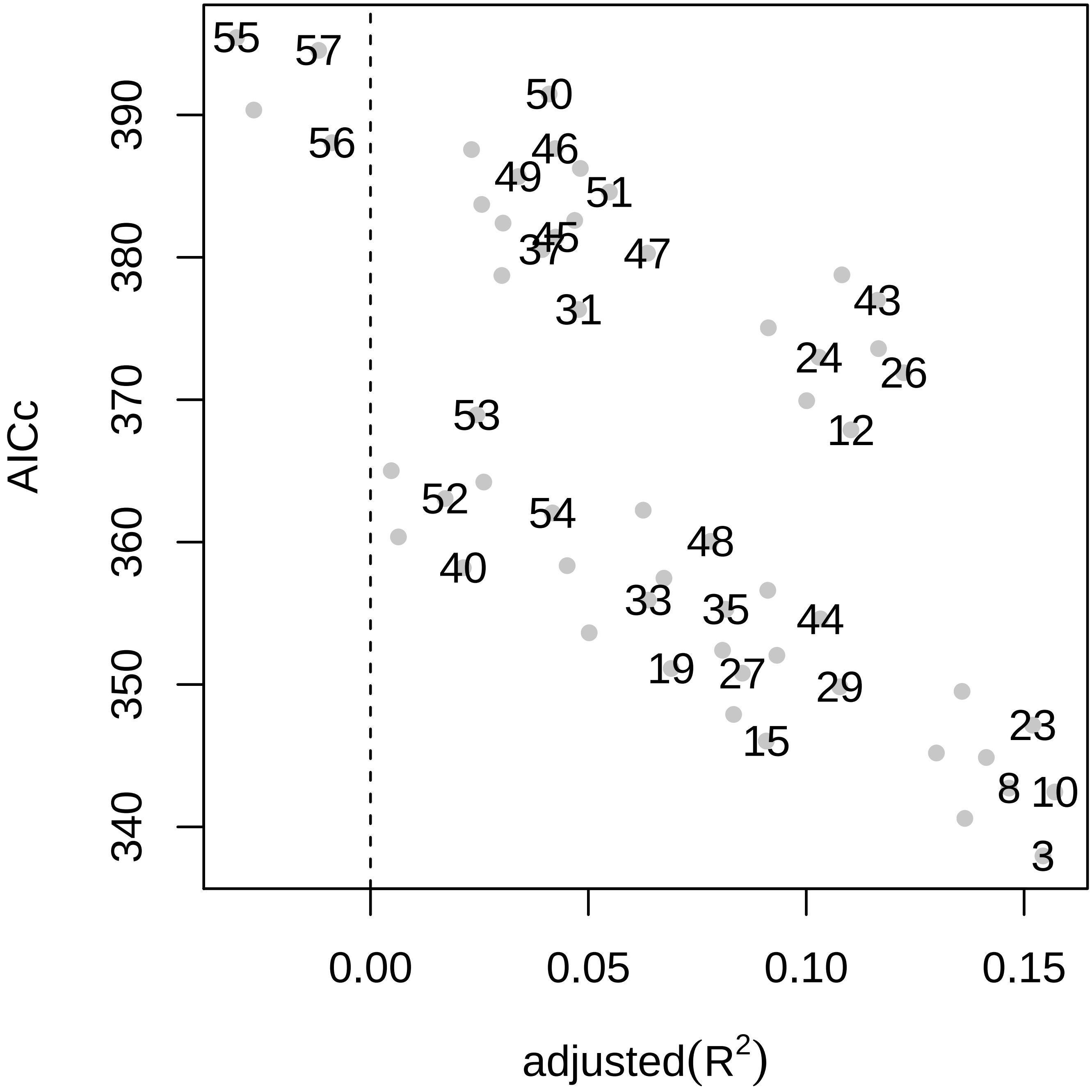

Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is recommended that all "best" models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

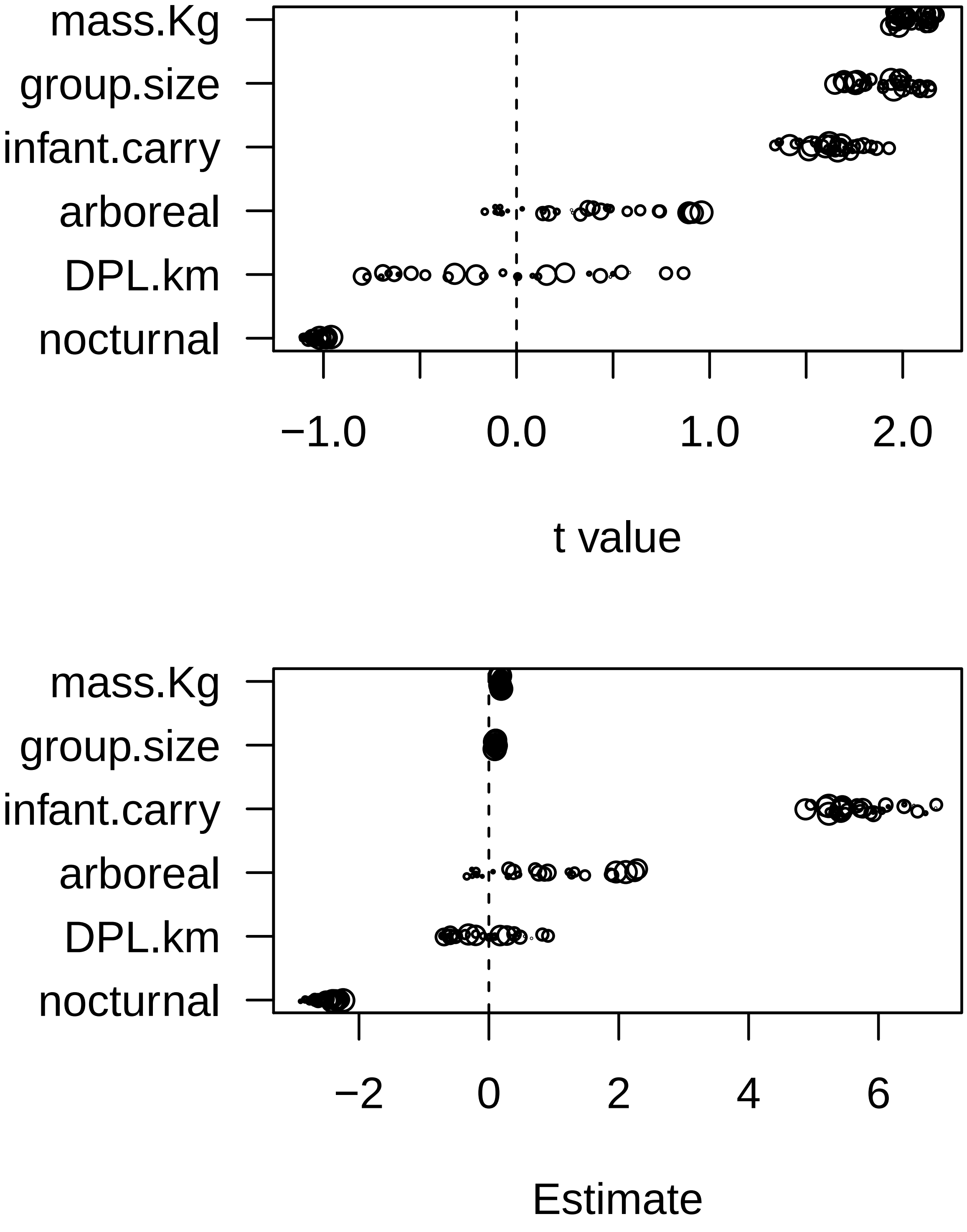

Figure 3: These horizontal parameter distribution dots plots demonstrate how the (t-values of) coefficients from all models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. Note that arboreal locomotor targeting and staturally 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] Rowe, N. and Meyers, M. (2017) *All the World's Primates*. Charlestown, RI: Pogonias Press.

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

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

[15] 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`.

[16] 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`.

[17] 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`.