

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

September 8, 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 = 1, 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 = 1, 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 iteration run should now generate more diverse output upon which the *mmodely* can demonstrate its 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 <- pglis.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.

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

```
models: 57
```

```
dimensions of sub-datasets:
```

q	n	qXn	rwGsm
5	8	18	23

	q	n
Min.	2.000000	54.00000
1st Qu.	2.000000	55.00000
Median	3.000000	57.00000
Mean	3.263158	57.38596
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. :54.00	Min. :2.000	Min. :100635	Min. : 1	Min. :0.003291
1st Qu.:55.00	1st Qu.:2.000	1st Qu.:102590	1st Qu.:15	1st Qu.:0.077285
Median :57.00	Median :3.000	Median :105909	Median :29	Median :0.098268
Mean :57.39	Mean :3.263	Mean :107090	Mean :29	Mean :0.100854
3rd Qu.:60.00	3rd Qu.:4.000	3rd Qu.:111717	3rd Qu.:43	3rd Qu.:0.136401
Max. :61.00	Max. :6.000	Max. :114777	Max. :57	Max. :0.164412
R2.adj	AIC	AICc	BIC	AICw
Min. : -0.03108	Min. :348.4	Min. :349.7	Min. :358.4	Min. :0.0000000
1st Qu.: 0.03022	1st Qu.:357.1	1st Qu.:358.3	1st Qu.:366.5	1st Qu.:0.0000000
Median : 0.04885	Median :367.4	Median :367.9	Median :374.6	Median :0.0000448
Mean : 0.04528	Mean :370.1	Mean :371.0	Mean :378.8	Mean :0.0175439
3rd Qu.: 0.06831	3rd Qu.:383.9	3rd Qu.:384.4	3rd Qu.:391.0	3rd Qu.:0.0053895
Max. : 0.08871	Max. :395.7	Max. :396.1	Max. :402.0	Max. :0.3968045

```

BICw
Min.   :0.0000000
1st Qu.:0.0000000
Median :0.0001259
Mean   :0.0175439
3rd Qu.:0.0073734
Max.   :0.4205973

```

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 multicollinearity, as denominators of regression coefficients change across models, it is possible to rescale these using 'standardize' [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=TRUE)
> #
> 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
0.1842242	0.0974225	6.2634475	0.5925500	-0.2471874	-2.6774837

```

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

```

	mass.Kg	group.size	infant.carry	arboreal	DPL.km	nocturnal
100635	0.19161	0.08175	6.26523	2.46357	-0.31234	-2.56085
102166	0.24375	NaN	6.63566	1.30648	0.36208	-2.59529
102423	0.13365	0.07425	5.61638	NaN	-0.68583	-2.50235
102590	NaN	0.11473	6.43453	1.21951	0.01318	-2.44344
103954	0.22096	NaN	6.07110	NaN	-0.01067	-2.55059
104121	NaN	NaN	7.03175	0.14109	0.55688	-2.60370
104378	NaN	0.10450	5.64919	NaN	-0.24079	-2.45362
105909	NaN	NaN	6.40374	NaN	0.39520	-2.67168
108398	0.15120	0.08256	NaN	1.17702	-0.77476	-2.93071
109929	0.22109	NaN	NaN	0.37822	-0.21896	-3.01372
110186	0.12306	0.08360	NaN	NaN	-0.98273	-2.58580
110353	NaN	0.11049	NaN	0.34349	-0.45816	-2.84288
111034	0.14476	0.08655	NaN	0.96328	-0.63786	NaN
111717	0.21187	NaN	NaN	NaN	-0.36439	-2.65797
111884	NaN	NaN	NaN	-0.56767	0.01832	-3.07758
112141	NaN	0.11000	NaN	NaN	-0.57849	-2.53363
112565	0.22501	NaN	NaN	0.18164	-0.09680	NaN
112822	0.12040	0.08649	NaN	NaN	-0.81017	NaN
112989	NaN	0.11296	NaN	0.17466	-0.35424	NaN
113672	NaN	NaN	NaN	NaN	0.01002	-2.81593
114353	0.22333	NaN	NaN	NaN	-0.21582	NaN
114520	NaN	NaN	NaN	-0.67069	0.14111	NaN
114777	NaN	0.12119	NaN	NaN	-0.44009	NaN

```
attr(,"MSE")
```

	mass.Kg	group.size	infant.carry	arboreal	DPL.km	nocturnal
100635	0.00393	0.00326	0.45680	0.15991	0.01960	0.02476
102166	0.00009	NA	0.03581	0.01898	0.00133	0.00075
102423	0.00252	0.00125	0.07358	NA	0.00418	0.00862
102590	NA	0.00004	0.01855	0.00284	0.00081	0.00037
103954	0.00005	NA	0.00199	NA	0.00031	0.00021
104121	NA	NA	0.00048	0.00034	0.00002	0.00002
104378	NA	0.00004	0.00177	NA	0.00021	0.00027
105909	NA	NA	0.00004	NA	0.00000	0.00000


```

108398 0.00000 0.00000 NA 0.00000 0.00000 0.00000
109929 0.00000 NA NA 0.00000 0.00000 0.00000
110186 0.00000 0.00000 NA NA 0.00000 0.00000
110353 NA 0.00000 NA 0.00000 0.00000 0.00000
111034 0.00000 0.00000 NA 0.00000 0.00000 NA
111717 0.00000 NA NA NA 0.00000 0.00000
111884 NA NA NA 0.00000 0.00000 0.00000
112141 NA 0.00000 NA NA 0.00000 0.00000
112565 0.00000 NA NA 0.00000 0.00000 NA
112822 0.00000 0.00000 NA NA 0.00000 NA
112989 NA 0.00000 NA 0.00000 0.00000 NA
113672 NA NA NA NA 0.00000 0.00000
114353 0.00000 NA NA NA 0.00000 NA
114520 NA NA NA 0.00000 0.00000 NA
114777 NA 0.00000 NA NA 0.00000 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  0.7726650  0.3183165  0.0000000

```

```

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

```

```

      mass.Kg  group.size infant.carry  arboreal  DPL.km  nocturnal
100635      1          1          1  1.00000  0.00000      0
102166      1         NaN          1  1.00000  1.00000      0
102423      1          1          1      NaN  0.00000      0
102590     NaN          1          1  1.00000  0.67970      0
103954      1         NaN          1      NaN  0.64158      0
104121     NaN         NaN          1  0.52499  1.00000      0
104378     NaN          1          1      NaN  0.00000      0
105909     NaN         NaN          1      NaN  1.00000      0
108398      1          1         NaN  1.00000  0.00000      0
109929      1         NaN         NaN  1.00000  0.00000      0
110186      1          1         NaN      NaN  0.00000      0
110353     NaN          1         NaN  1.00000  0.00000      0
111034      1          1         NaN  1.00000  0.00000     NaN
111717      1         NaN         NaN      NaN  0.00000      0
111884     NaN         NaN         NaN  0.00000  1.00000      0
112141     NaN          1         NaN      NaN  0.00000      0
112565      1         NaN         NaN  1.00000  0.00000     NaN
112822      1          1         NaN      NaN  0.00000     NaN
112989     NaN          1         NaN  0.74699  0.00000     NaN
113672     NaN         NaN         NaN      NaN  1.00000      0
114353      1         NaN         NaN      NaN  0.00000     NaN
114520     NaN         NaN         NaN  0.00000  1.00000     NaN
114777     NaN          1         NaN      NaN  0.00000     NaN

```

```
attr(,"MSE")
```

```

      mass.Kg  group.size infant.carry  arboreal  DPL.km  nocturnal
100635      0          0          0  0.00000  0.00000      0
102166      0         NA          0  0.00000  0.00000      0
102423      0          0          0      NA  0.00000      0
102590     NA          0          0  0.00000  0.00667      0
103954      0         NA          0      NA  0.00249      0
104121     NA         NA          0  0.00048  0.00000      0
104378     NA          0          0      NA  0.00000      0
105909     NA         NA          0      NA  0.00000      0
108398      0          0         NA  0.00000  0.00000      0
109929      0         NA         NA  0.00000  0.00000      0

```

110186	0	0	NA	NA	0.00000	0
110353	NA	0	NA	0.00000	0.00000	0
111034	0	0	NA	0.00000	0.00000	NA
111717	0	NA	NA	NA	0.00000	0
111884	NA	NA	NA	0.00000	0.00000	0
112141	NA	0	NA	NA	0.00000	0
112565	0	NA	NA	0.00000	0.00000	NA
112822	0	0	NA	NA	0.00000	NA
112989	NA	0	NA	0.00000	0.00000	NA
113672	NA	NA	NA	NA	0.00000	0
114353	0	NA	NA	NA	0.00000	NA
114520	NA	NA	NA	0.00000	0.00000	NA
114777	NA	0	NA	NA	0.00000	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
54 54 4 4X54 100635      8 0.14488625 0.07508105 348.4342 349.6842 358.3791 3.968045e-01
55 55 3 3X55 102423     23 0.13358768 0.08262225 352.6542 353.4542 360.6835 6.024640e-02
56 56 2 2X56 103954     44 0.11910224 0.08586082 357.6070 358.0685 363.6831 5.997010e-03
57 57 2 2X57 105909     54 0.07667079 0.04247342 367.4101 367.8630 373.5393 4.478175e-05
58 58 4 4X58 108398     12 0.10853590 0.04125559 374.4853 375.6392 384.7876 9.173156e-07
59 59 3 3X59 110186     26 0.09938903 0.05026480 378.6234 379.3641 386.9335 1.424506e-07
60 60 2 2X60 112822     43 0.08340822 0.05124711 382.8597 383.2883 389.1428 2.002324e-08
61 61 2 2X61 114777     50 0.08055197 0.04884686 390.1596 390.5806 396.4922 5.224275e-10

      BICw
54 4.205973e-01
55 1.328830e-01
56 2.965691e-02
57 2.147192e-04
58 7.750748e-07
59 2.650670e-07
60 8.782575e-08
61 2.226963e-09
```

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]], rtn.line=FALSE, mn=i)

1 +mass(0.195) +group(0.196) +infan(0.252) arbore(0.41) | DPL(0.737) -noctu(0.312) R2adj: 0.058 AICc: 353.62
2 +infan(0.173) +group(0.195) +mass(0.199) arbore(0.413) | DPL(0.821) R2adj: 0.057 AICc: 352.16
3 +infan(0.19) +mass(0.201) +group(0.207) +arbor(0.296) | -noctu(0.322) R2adj: 0.075 AICc: 351.11
4 +group(0.193) +infan(0.216) +mass(0.267) | DPL(0.468) -noctu(0.282) R2adj: 0.074 AICc: 356.69
5 +group(0.151) +mass(0.251) arbore(0.764) | DPL(0.462) -noctu(0.239) R2adj: 0.033 AICc: 377.52
6 +mass(0.062) +infan(0.166) arbore(0.524) DPL(0.796) | -noctu(0.321) R2adj: 0.05 AICc: 358.66
7 +group(0.069) +infan(0.208) arbore(0.627) | DPL(0.954) noctur(0.343) R2adj: 0.049 AICc: 359.79
8 +infan(0.134) +group(0.194) +mass(0.2) +arbor(0.32) | R2adj: 0.075 AICc: 349.68
9 +infan(0.183) +group(0.194) +mass(0.265) | DPL(0.548) R2adj: 0.071 AICc: 355.48
10 +infan(0.116) +group(0.254) mass(0.344) | -noctu(0.316) R2adj: 0.083 AICc: 354.76
11 +group(0.13) +mass(0.268) arbore(0.807) | DPL(0.534) R2adj: 0.026 AICc: 381.78
12 +group(0.189) +mass(0.296) arbore(0.569) | -noctu(0.253) R2adj: 0.041 AICc: 375.64
13 +group(0.126) +mass(0.257) | -DPL(0.302) -noctu(0.265) R2adj: 0.052 AICc: 380.58
14 +mass(0.059) +infan(0.109) arbore(0.521) DPL(0.711) | R2adj: 0.05 AICc: 357.25
15 +mass(0.06) +infan(0.168) arbore(0.551) | -noctu(0.3) R2adj: 0.067 AICc: 356.21
16 +mass(0.071) +infan(0.115) | DPL(0.92) -noctu(0.288) R2adj: 0.071 AICc: 361.55
17 +mass(0.072) arbore(0.936) | DPL(0.822) -noctu(0.228) R2adj: 0.02 AICc: 383.03
18 +group(0.069) +infan(0.143) arbore(0.629) DPL(0.965) | R2adj: 0.05 AICc: 358.28
19 +group(0.053) +infan(0.179) arbore(0.561) | noctur(0.338) R2adj: 0.067 AICc: 357.27
20 +group(0.071) +infan(0.173) | DPL(0.748) -noctu(0.302) R2adj: 0.072 AICc: 362.54
21 ++grou(0.049) arbore(0.988) | DPL(0.658) -noctu(0.254) R2adj: 0.031 AICc: 383.39
22 +infan(0.139) DPL(0.666) arbore(0.794) | noctur(0.334) R2adj: -0.001 AICc: 367.3
23 +infan(0.105) +group(0.238) mass(0.323) | R2adj: 0.083 AICc: 353.45
24 +group(0.156) +mass(0.303) arbore(0.632) | R2adj: 0.037 AICc: 379.82
25 +group(0.11) +mass(0.268) | DPL(0.379) R2adj: 0.048 AICc: 384.75
26 +group(0.183) mass(0.387) | -noctu(0.307) R2adj: 0.05 AICc: 379.36
```

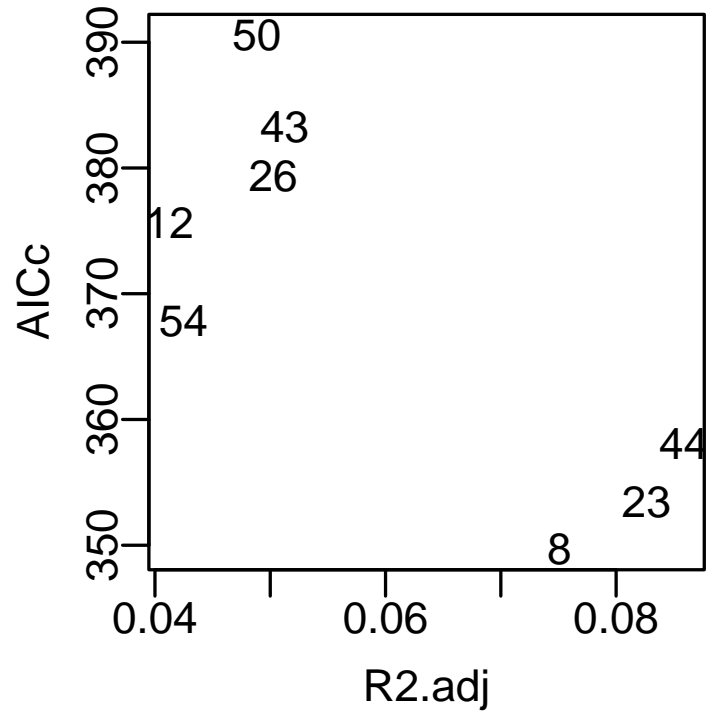
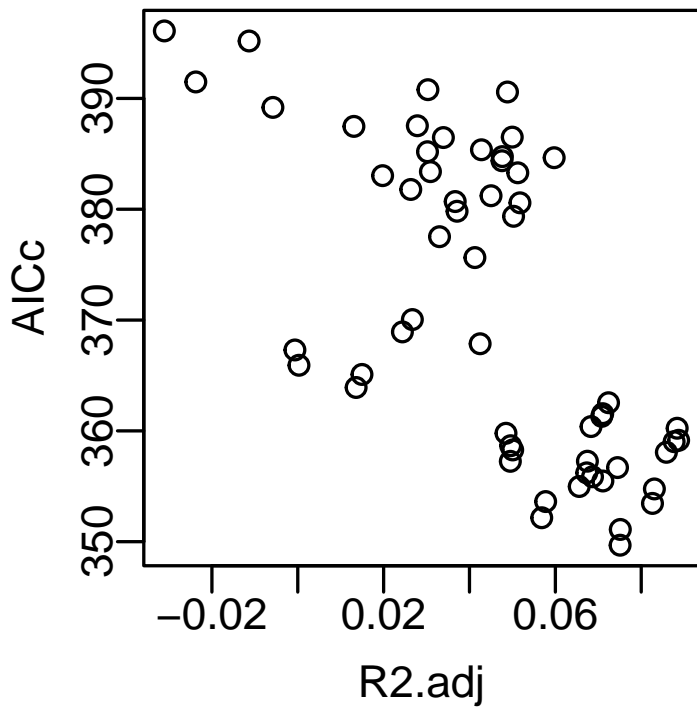
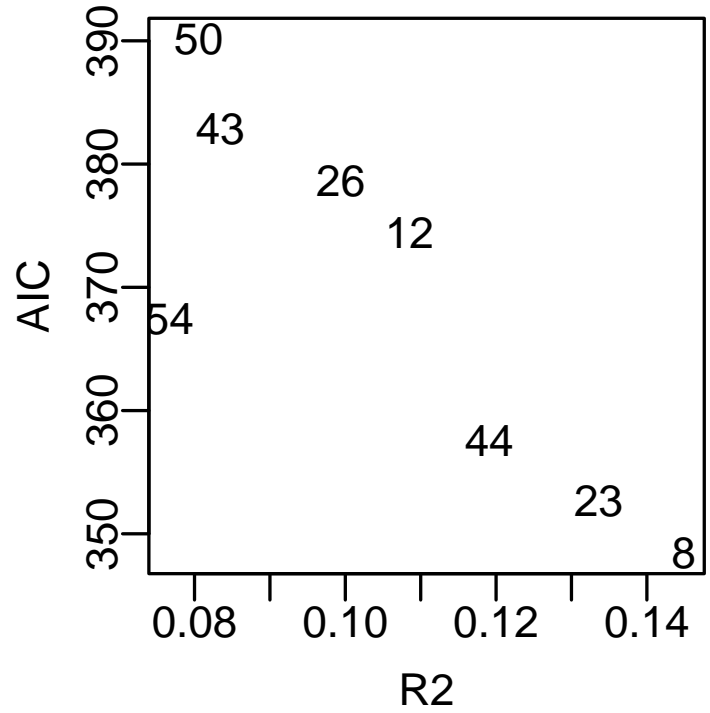
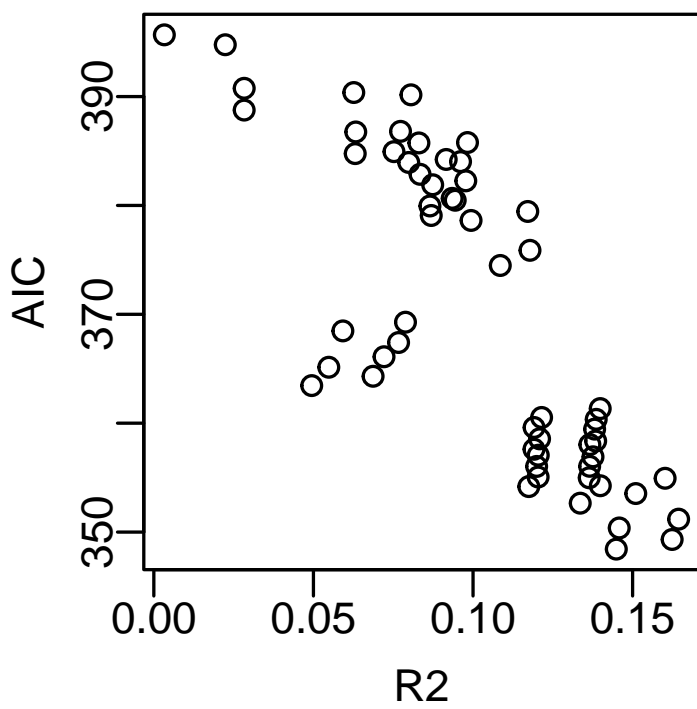


Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of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

```

27 +mass(0.056) +infan(0.115) arbore(0.596) | R2adj: 0.066 AICc: 354.98
28 +mass(0.066) +infan(0.095) DPL(0.97) | R2adj: 0.068 AICc: 360.39
29 +mass(0.067) +infan(0.095) | -noctu(0.285) R2adj: 0.089 AICc: 359.15
30 +mass(0.063) arbore(0.966) | DPL(0.919) R2adj: 0.013 AICc: 387.48
31 +mass(0.071) arbore(0.835) | -noctu(0.229) R2adj: 0.037 AICc: 380.69
32 +mass(0.063) | DPL(0.661) -noctu(0.26) R2adj: 0.034 AICc: 386.47
33 ++grou(0.048) +infan(0.127) arbore(0.599) | R2adj: 0.069 AICc: 355.86
34 +group(0.071) +infan(0.145) | DPL(0.848) R2adj: 0.071 AICc: 361.31
35 +group(0.06) +infan(0.12) | -noctu(0.312) R2adj: 0.088 AICc: 360.24
36 ++grou(0.043) | arbore(0.977) DPL(0.725) R2adj: 0.028 AICc: 387.53
37 +group(0.052) arbore(0.835) | -noctu(0.26) R2adj: 0.045 AICc: 381.22
38 ++grou(0.035) | DPL(0.522) -noctu(0.278) R2adj: 0.05 AICc: 386.51
39 +infan(0.09) DPL(0.59) arbore(0.796) | R2adj: 0 AICc: 365.92
40 +infan(0.153) arbore(0.977) | -noctu(0.305) R2adj: 0.015 AICc: 365.09
41 +infan(0.087) DPL(0.726) | -noctu(0.293) R2adj: 0.027 AICc: 370.05
42 DPL(0.985) | arbore(0.815) -noctu(0.229) R2adj: -0.024 AICc: 391.49
43 +group(0.152) mass(0.372) | R2adj: 0.051 AICc: 383.29
44 +mass(0.059) +infan(0.084) | R2adj: 0.086 AICc: 358.07
45 +mass(0.061) arbore(0.919) | R2adj: 0.03 AICc: 385.19
46 +mass(0.054) | DPL(0.789) R2adj: 0.03 AICc: 390.79
47 +mass(0.068) | -noctu(0.274) R2adj: 0.048 AICc: 384.38
48 +group(0.051) +infan(0.108) | R2adj: 0.088 AICc: 359.03
49 ++grou(0.042) arbore(0.902) | R2adj: 0.043 AICc: 385.37
50 ++grou(0.031) | DPL(0.616) R2adj: 0.049 AICc: 390.58
51 ++grou(0.042) | -noctu(0.302) R2adj: 0.06 AICc: 384.65
52 +infan(0.104) | arbore(0.961) R2adj: 0.014 AICc: 363.91
53 +infan(0.071) DPL(0.619) | R2adj: 0.024 AICc: 368.93
54 +infan(0.091) | -noctu(0.264) R2adj: 0.042 AICc: 367.86
55 DPL(0.884) | arbore(0.773) R2adj: -0.031 AICc: 396.09
56 | arbore(0.78) -noctu(0.223) R2adj: -0.006 AICc: 389.19
57 DPL(0.99) | -noctu(0.258) R2adj: -0.011 AICc: 395.18

> 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 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')

```

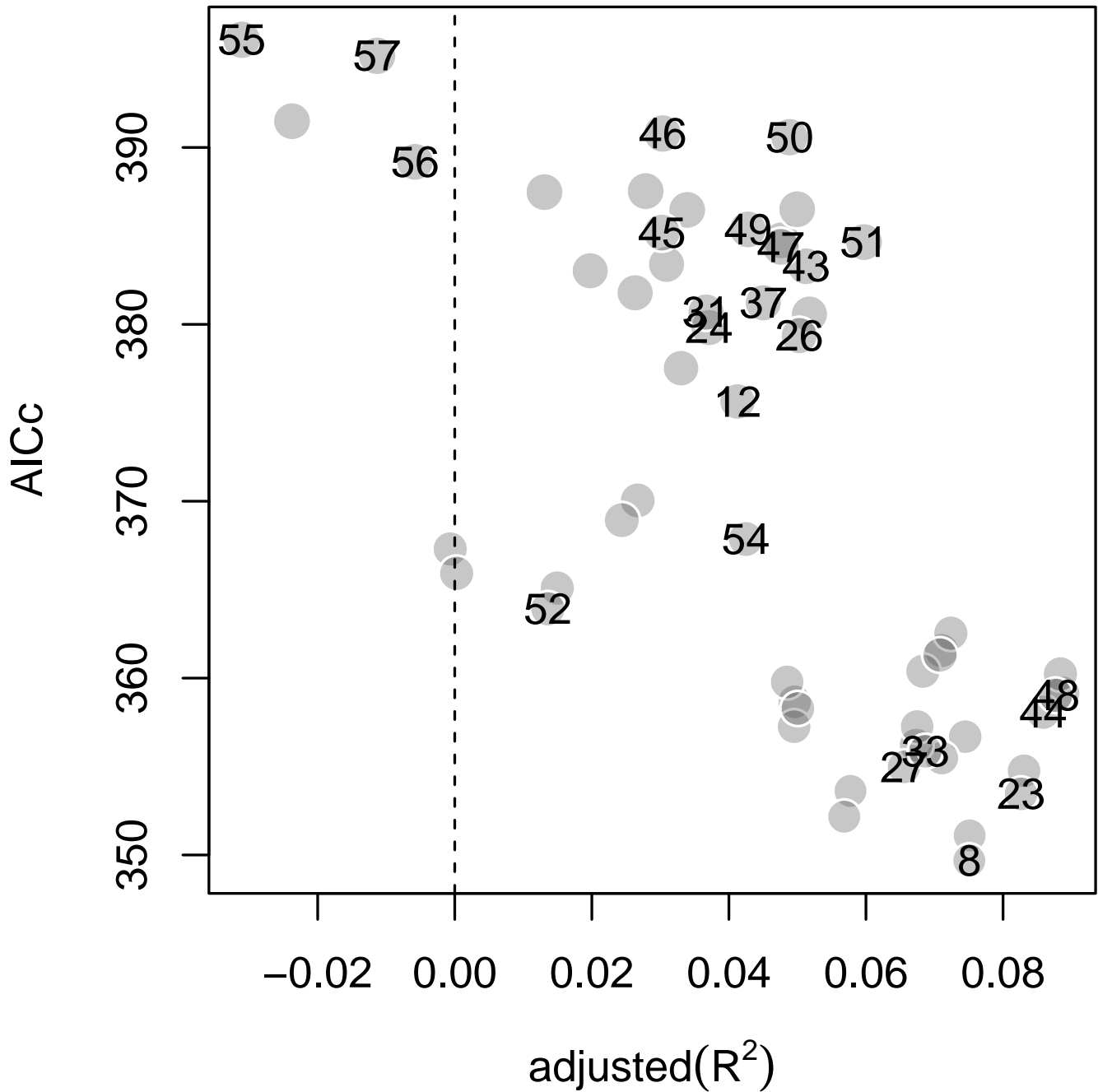


Figure 2: This is a one panel version of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is recommended that all "best" models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

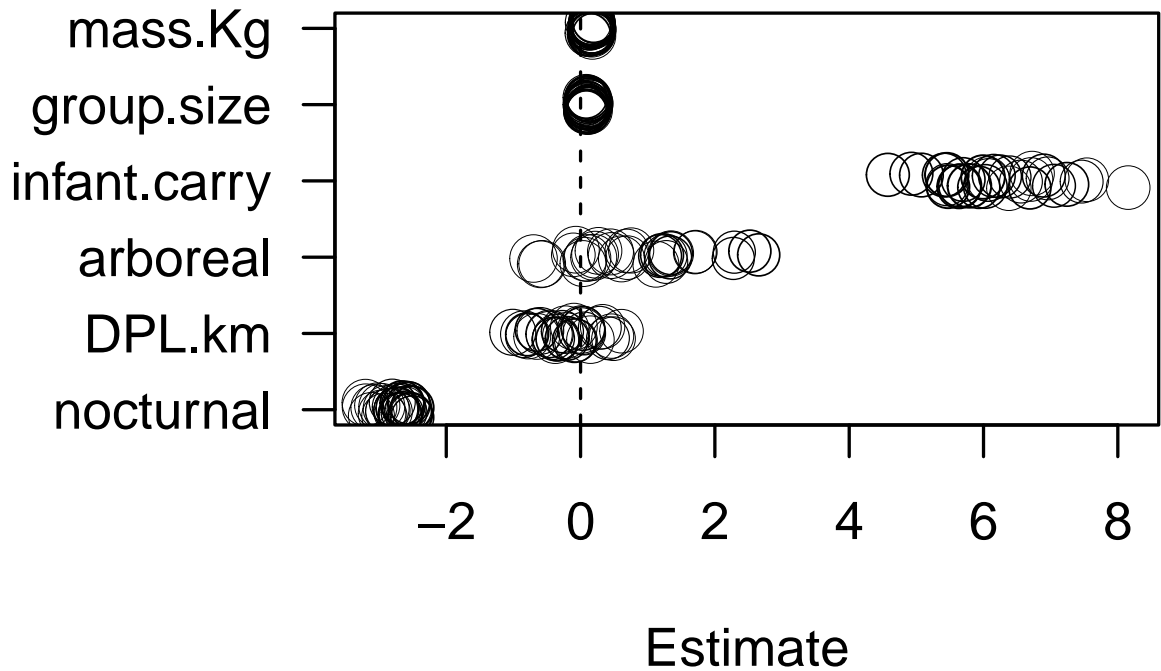
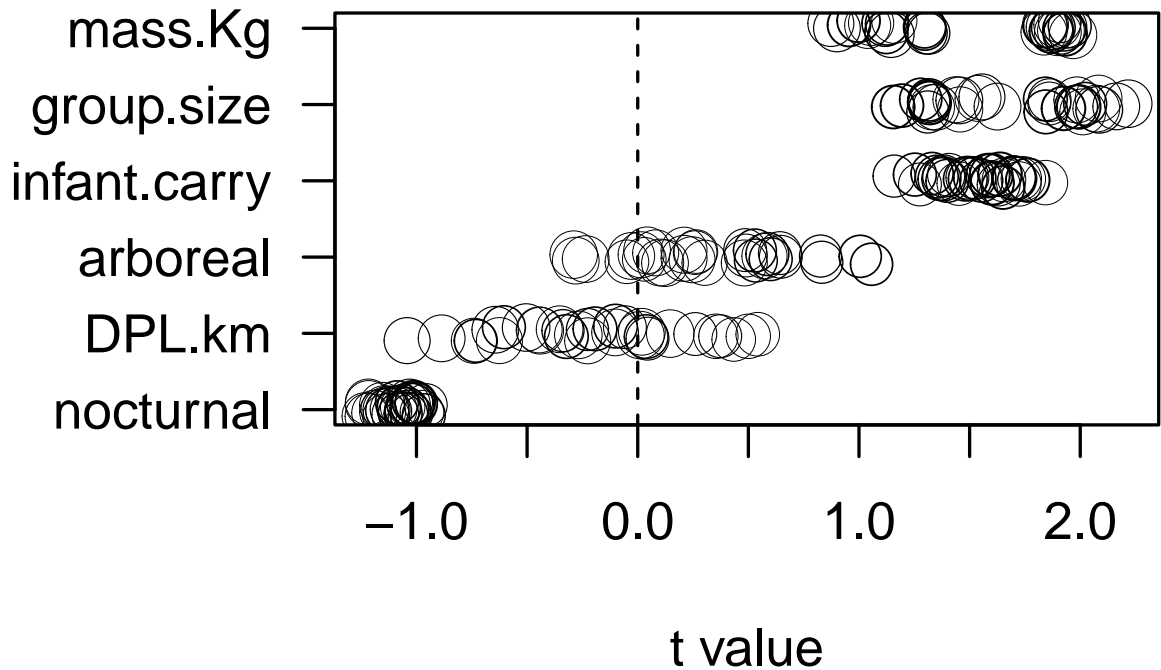


Figure 3: These horizontal parameter distribution dots plots demonstrate how the (t-values of) coefficients from all models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting of 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](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>.