

# 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

May 16, 2023

## 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
-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 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
-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 2 and 16 DF, p-value: 0.03136

[1] "AIC = 134.2"

Analysis of Variance Table

Sequential SS for pglms: 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:

pglms(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 <- pgl.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

> pgl.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	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. : 98548	Min. : 1	Min. :0.003291
1st Qu.:55.00	1st Qu.:2.000	1st Qu.:102163	1st Qu.:15	1st Qu.:0.071943
Median :56.00	Median :3.000	Median :104187	Median :29	Median :0.102991
Mean :57.11	Mean :3.263	Mean :106649	Mean :29	Mean :0.095448
3rd Qu.:60.00	3rd Qu.:4.000	3rd Qu.:112457	3rd Qu.:43	3rd Qu.:0.120155
Max. :61.00	Max. :6.000	Max. :114586	Max. :57	Max. :0.160824
R2.adj	AIC	AICc	BIC	AICw
Min. : -0.03108	Min. :346.1	Min. :347.9	Min. :357.9	Min. :0.0000000
1st Qu.: 0.02269	1st Qu.:356.1	1st Qu.:357.4	1st Qu.:365.7	1st Qu.:0.0000000
Median : 0.04119	Median :363.8	Median :364.6	Median :371.9	Median :0.0001183
Mean : 0.03914	Mean :370.1	Mean :371.0	Mean :378.8	Mean :0.0175439
3rd Qu.: 0.05701	3rd Qu.:385.7	3rd Qu.:386.1	3rd Qu.:392.8	3rd Qu.:0.0044554
Max. : 0.08353	Max. :395.7	Max. :396.1	Max. :402.0	Max. :0.5017428

```

BICw
Min.   :0.0000000
1st Qu.:0.0000000
Median :0.0003486
Mean   :0.0175439
3rd Qu.:0.0080507
Max.   :0.3872977

```

## 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
0.1414613    0.0941625    6.1122000    0.7850438  -0.1429545  -2.6086294

```

```

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

```

```

      mass.Kg  group.size infant.carry  arboreal  DPL.km  nocturnal
98548  0.14751    0.09144    5.67384  2.36161 -0.17835  -2.43590
100270   NaN    0.10454    5.78165  1.36574 -0.31332  -2.45082
100336  0.12388    0.07604    5.30488   NaN -0.67883  -2.37865
100375  0.14699    0.09222    6.26345  2.22419 -0.11414   NaN
100572  0.16315   NaN    6.31558  1.52091  0.65187  -2.55368
102058   NaN    0.09321    5.31146   NaN -0.56941  -2.45679
102097   NaN    0.10523    6.38743  1.23979 -0.23555   NaN
102163  0.12554    0.07710    5.40560   NaN -0.55361   NaN
102294   NaN   NaN    6.58847  0.28159  0.64606  -2.60200
102360  0.14470   NaN    6.03704   NaN  0.13621  -2.51757
102399  0.16361   NaN    6.93719  1.34568  0.61014   NaN
103885   NaN    0.09469    5.42293   NaN -0.44990   NaN
104082   NaN   NaN    6.24963   NaN  0.41166  -2.66989
104121   NaN   NaN    7.22962  0.10093  0.59726   NaN
104187  0.14804   NaN    6.14270   NaN  0.17639   NaN
105909   NaN   NaN    6.74373   NaN  0.44273   NaN
108947  0.13141    0.08934   NaN  1.12782 -0.62157  -2.72678
110669   NaN    0.10171   NaN  0.38122 -0.63281  -2.77108
110735  0.11931    0.08599   NaN   NaN -0.86123  -2.38330
110774  0.13228    0.09055   NaN  0.99704 -0.55612   NaN
110971  0.15029   NaN   NaN  0.39647  0.07588  -2.94042
112457   NaN    0.10101   NaN   NaN -0.73305  -2.47211
112496   NaN    0.10307   NaN  0.24755 -0.56803   NaN
112562  0.12090    0.08647   NaN   NaN -0.74936   NaN
112693   NaN   NaN   NaN -0.58996  0.15663  -3.03979
112759  0.14380   NaN   NaN   NaN -0.13872  -2.58536
112798  0.15255   NaN   NaN  0.22487  0.06928   NaN
114284   NaN    0.11399   NaN   NaN -0.62643   NaN
114481   NaN   NaN   NaN   NaN  0.12437  -2.75393
114520   NaN   NaN   NaN -0.66475  0.14188   NaN
114586  0.14942   NaN   NaN   NaN -0.09152   NaN

```

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

```
      mass.Kg  group.size infant.carry  arboreal  DPL.km  nocturnal
```



```

98548 0.00052 0.00244 0.07850 0.08104 0 0.03144
100270 NA 0.00023 0.00942 0.00895 0 0.00260
100336 0.00055 0.00192 0.07454 NA 0 0.01848
100375 0.00010 0.00027 0.00521 0.00807 0 NA
100572 0.00004 NA 0.03413 0.03761 0 0.00587
102058 NA 0.00012 0.00452 NA 0 0.00120
102097 NA 0.00003 0.00089 0.00102 0 NA
102163 0.00007 0.00023 0.00866 NA 0 NA
102294 NA NA 0.00116 0.00127 0 0.00020
102360 0.00001 NA 0.00201 NA 0 0.00021
102399 0.00001 NA 0.00409 0.00481 0 NA
103885 NA 0.00001 0.00045 NA 0 NA
104082 NA NA 0.00019 NA 0 0.00005
104121 NA NA 0.00012 0.00014 0 NA
104187 0.00000 NA 0.00030 NA 0 NA
105909 NA NA 0.00000 NA 0 NA
108947 0.00000 0.00000 NA 0.00000 0 0.00000
110669 NA 0.00000 NA 0.00000 0 0.00000
110735 0.00000 0.00000 NA NA 0 0.00000
110774 0.00000 0.00000 NA 0.00000 0 NA
110971 0.00000 NA NA 0.00000 0 0.00000
112457 NA 0.00000 NA NA 0 0.00000
112496 NA 0.00000 NA 0.00000 0 NA
112562 0.00000 0.00000 NA NA 0 NA
112693 NA NA NA 0.00000 0 0.00000
112759 0.00000 NA NA NA 0 0.00000
112798 0.00000 NA NA 0.00000 0 NA
114284 NA 0.00000 NA NA 0 NA
114481 NA NA NA NA 0 0.00000
114520 NA NA NA 0.00000 0 NA
114586 0.00000 NA 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  0.7955031  0.4193548  0.0000000

```

```

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

```

```

      mass.Kg  group.size infant.carry  arboreal  DPL.km  nocturnal
98548      1          1          1  1.00000  0          0
100270    NaN          1          1  1.00000  0          0
100336      1          1          1      NaN  0          0
100375      1          1          1  1.00000  0      NaN
100572      1      NaN          1  1.00000  1          0
102058    NaN          1          1      NaN  0          0
102097    NaN          1          1  1.00000  0      NaN
102163      1          1          1      NaN  0      NaN
102294    NaN      NaN          1  1.00000  1          0
102360      1      NaN          1      NaN  1          0
102399      1      NaN          1  1.00000  1      NaN
103885    NaN          1          1      NaN  0      NaN
104082    NaN      NaN          1      NaN  1          0
104121    NaN      NaN          1  0.26827  1      NaN
104187      1      NaN          1      NaN  1      NaN
105909    NaN      NaN          1      NaN  1      NaN
108947      1          1      NaN  1.00000  0          0
110669    NaN          1      NaN  0.73206  0          0
110735      1          1      NaN      NaN  0          0

```

110774	1	1	NaN	1.00000	0	NaN
110971	1	NaN	NaN	1.00000	1	0
112457	NaN	1	NaN	NaN	0	0
112496	NaN	1	NaN	0.72772	0	NaN
112562	1	1	NaN	NaN	0	NaN
112693	NaN	NaN	NaN	0.00000	1	0
112759	1	NaN	NaN	NaN	0	0
112798	1	NaN	NaN	1.00000	1	NaN
114284	NaN	1	NaN	NaN	0	NaN
114481	NaN	NaN	NaN	NaN	1	0
114520	NaN	NaN	NaN	0.00000	1	NaN
114586	1	NaN	NaN	NaN	0	NaN

attr(,"MSE")

	mass.Kg	group.size	infant.carry	arboreal	DPL.km	nocturnal
98548	0	0	0	0.00000	0	0
100270	NA	0	0	0.00000	0	0
100336	0	0	0	NA	0	0
100375	0	0	0	0.00000	0	NA
100572	0	NA	0	0.00000	0	0
102058	NA	0	0	NA	0	0
102097	NA	0	0	0.00000	0	NA
102163	0	0	0	NA	0	NA
102294	NA	NA	0	0.00000	0	0
102360	0	NA	0	NA	0	0
102399	0	NA	0	0.00000	0	NA
103885	NA	0	0	NA	0	NA
104082	NA	NA	0	NA	0	0
104121	NA	NA	0	0.00019	0	NA
104187	0	NA	0	NA	0	NA
105909	NA	NA	0	NA	0	NA
108947	0	0	NA	0.00000	0	0
110669	NA	0	NA	0.00000	0	0
110735	0	0	NA	NA	0	0
110774	0	0	NA	0.00000	0	NA
110971	0	NA	NA	0.00000	0	0
112457	NA	0	NA	NA	0	0
112496	NA	0	NA	0.00000	0	NA
112562	0	0	NA	NA	0	NA
112693	NA	NA	NA	0.00000	0	0
112759	0	NA	NA	NA	0	0
112798	0	NA	NA	0.00000	0	NA
114284	NA	0	NA	NA	0	NA
114481	NA	NA	NA	NA	0	0
114520	NA	NA	NA	0.00000	0	NA
114586	0	NA	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	98548	3	0.16048264	0.07117229	346.0969	347.9230	357.9187	5.017428e-01
54	54	4	4X54	100336	10	0.14913518	0.07967683	350.2650	351.5150	360.2100	8.326947e-02
55	55	3	3X55	102163	23	0.13444779	0.08353296	354.4627	355.2627	362.4921	1.278441e-02
56	56	2	2X56	104187	44	0.10825218	0.07460132	359.5895	360.0510	365.6656	1.166579e-03
57	57	2	2X57	105909	53	0.05921713	0.02437332	368.4776	368.9304	374.6067	1.376543e-05
58	58	4	4X58	108947	12	0.11461615	0.04779473	376.2135	377.3673	386.5157	2.026416e-07
59	59	3	3X59	110735	26	0.10578641	0.05701112	380.2823	381.0230	388.5925	3.257626e-08
60	60	2	2X60	112562	43	0.09182925	0.05996361	384.4738	384.9023	390.7568	4.682973e-09
61	61	2	2X61	114284	50	0.07037884	0.03832294	391.4801	391.9012	397.8127	1.414965e-10

```

      BICw
53 3.872977e-01
54 1.231693e-01
55 3.935009e-02
56 8.050673e-03
57 9.210568e-05
58 2.389353e-07
59 8.459012e-08
60 2.866395e-08
61 8.417041e-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(0.136)	+group(0.159)	+infan(0.261)	arbore(0.422)		DPL(0.892)	noctur(0.355)	R2adj: 0.051	AICc: 350.5
2	+mass(0.126)	+group(0.139)	+infan(0.184)	arbore(0.419)		DPL(0.925)	R2adj: 0.057	AICc: 354.06	
3	+mass(0.13)	+group(0.132)	+infan(0.214)	+arbor(0.315)		noctur(0.354)	R2adj: 0.071	AICc: 347.92	
4	+group(0.163)	+mass(0.169)	+infan(0.245)		DPL(0.54)	-noctur(0.32)	R2adj: 0.068	AICc: 353.63	
5	+group(0.106)	+mass(0.149)	arbore(0.775)		DPL(0.57)	-noctur(0.28)	R2adj: 0.036	AICc: 379.5	
6	+mass(0.092)	+infan(0.149)	arbore(0.428)	DPL(0.577)		noctur(0.353)	R2adj: 0.04	AICc: 355.15	
7	+group(0.098)	+infan(0.257)	arbore(0.672)		DPL(0.809)	noctur(0.344)	R2adj: 0.03	AICc: 356.14	
8	+group(0.116)	+mass(0.121)	+infan(0.153)	arbore(0.331)		R2adj: 0.076	AICc: 351.53		
9	+group(0.152)	+mass(0.154)	+infan(0.203)		DPL(0.594)	R2adj: 0.071	AICc: 357.37		
10	+infan(0.141)	+mass(0.181)	+group(0.203)		noctur(0.349)	R2adj: 0.08	AICc: 351.52		
11	+group(0.095)	+mass(0.137)	arbore(0.793)		DPL(0.592)	R2adj: 0.034	AICc: 383.51		
12	+group(0.127)	+mass(0.148)	arbore(0.567)		-noctur(0.287)	R2adj: 0.048	AICc: 377.37		
13	+group(0.085)	+mass(0.151)		DPL(0.363)	-noctur(0.308)	R2adj: 0.054	AICc: 382.5		
14	+mass(0.082)	+infan(0.103)	arbore(0.461)	DPL(0.582)		R2adj: 0.041	AICc: 359.02		
15	+mass(0.088)	+infan(0.175)	arbore(0.558)		-noctur(0.316)	R2adj: 0.053	AICc: 352.96		
16	+infan(0.109)	+mass(0.114)	DPL(0.885)		-noctur(0.318)	R2adj: 0.057	AICc: 358.33		
17	+mass(0.096)	arbore(0.849)	DPL(0.94)		-noctur(0.25)	R2adj: 0.01	AICc: 385.14		
18	+group(0.084)	+infan(0.178)	arbore(0.676)		DPL(0.844)	R2adj: 0.034	AICc: 359.84		
19	+group(0.084)	+infan(0.198)	arbore(0.531)		noctur(0.349)	R2adj: 0.048	AICc: 353.67		
20	+group(0.097)	+infan(0.227)		DPL(0.601)	-noctur(0.302)	R2adj: 0.054	AICc: 358.85		
21	+group(0.065)		arbore(0.984)	DPL(0.557)	-noctur(0.267)	R2adj: 0.019	AICc: 384.73		
22	+infan(0.133)	DPL(0.58)	arbore(0.733)		noctur(0.343)	R2adj: 0.001	AICc: 361.83		
23	+infan(0.126)	+mass(0.162)	+group(0.176)		R2adj: 0.084	AICc: 355.26			
24	+group(0.11)	+mass(0.136)	arbore(0.602)		R2adj: 0.047	AICc: 381.43			

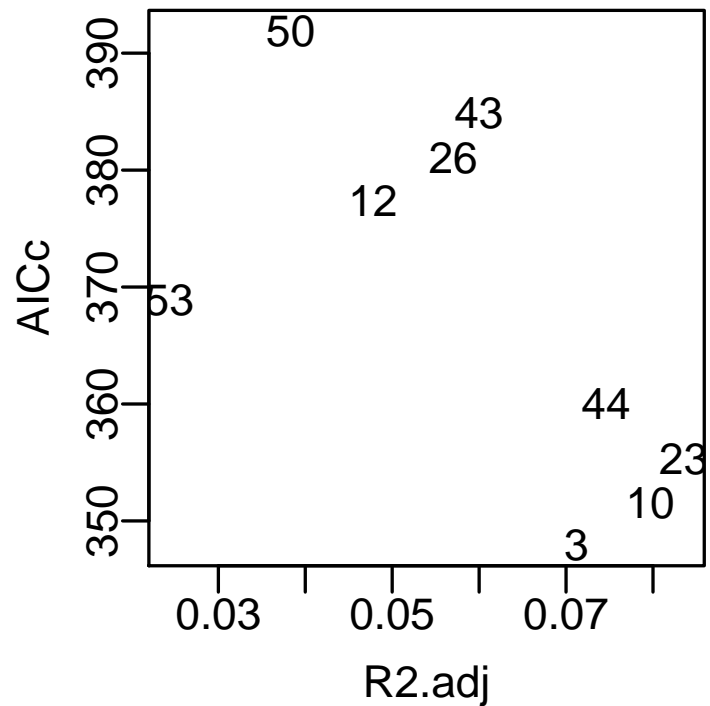
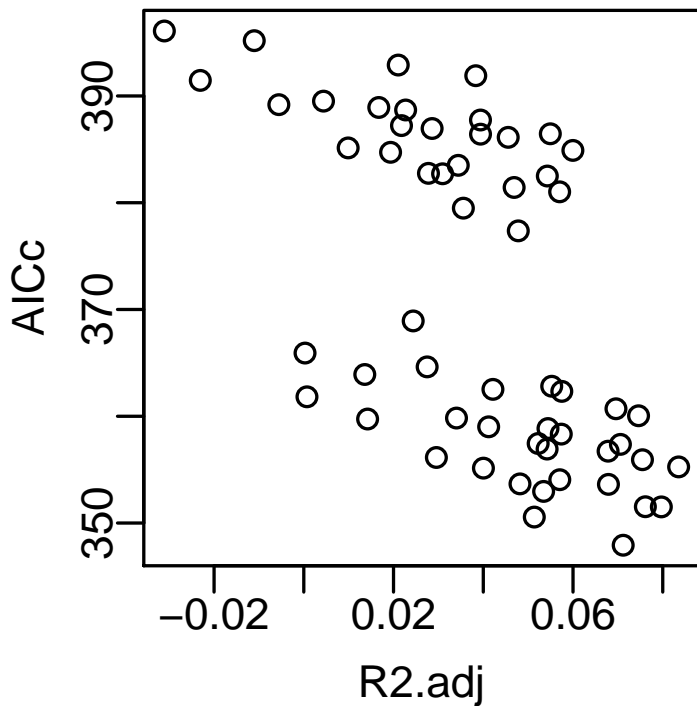
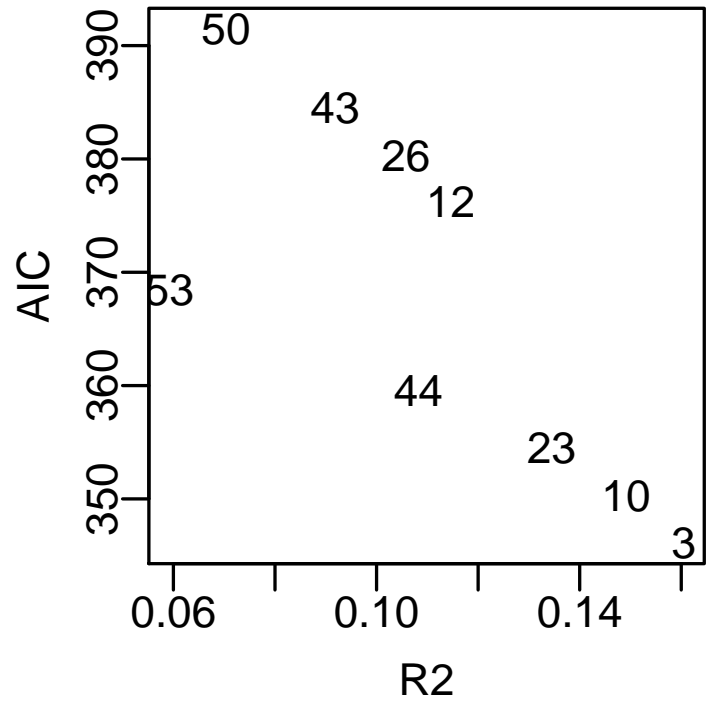
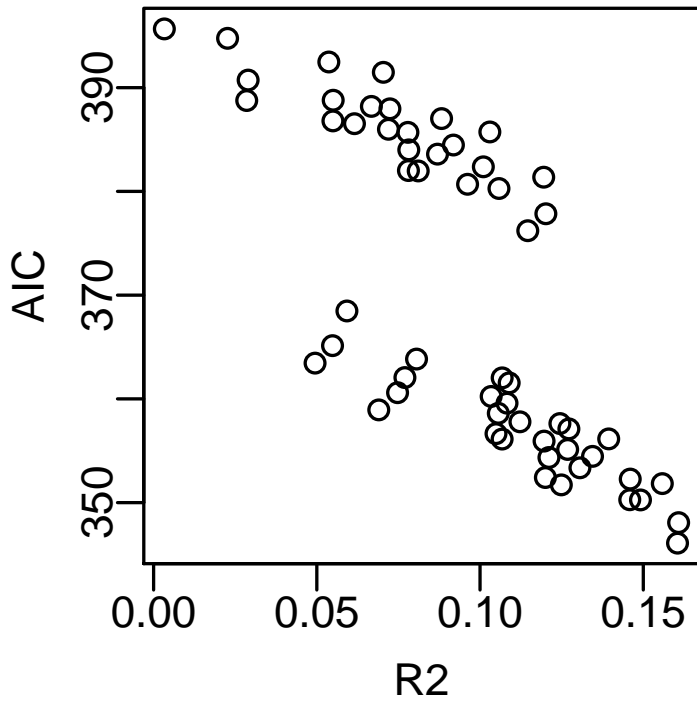


Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of 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(0.079) +mass(0.138) | DPL(0.407) R2adj: 0.055 AICc: 386.46
26 +group(0.137) +mass(0.176) | noctur(0.344) R2adj: 0.057 AICc: 381.02
27 +mass(0.079) +infan(0.119) arbore(0.597) | R2adj: 0.054 AICc: 356.93
28 +infan(0.092) +mass(0.098) DPL(0.846) | R2adj: 0.057 AICc: 362.33
29 +mass(0.1) +infan(0.101) | -noctu(0.3) R2adj: 0.075 AICc: 355.93
30 +mass(0.084) arbore(0.904) DPL(0.943) | R2adj: 0.004 AICc: 389.52
31 +mass(0.091) arbore(0.859) | -noctu(0.243) R2adj: 0.028 AICc: 382.75
32 +mass(0.09) | DPL(0.871) -noctu(0.288) R2adj: 0.023 AICc: 388.69
33 +group(0.072) +infan(0.14) arbore(0.56) | R2adj: 0.052 AICc: 357.45
34 +group(0.086) +infan(0.188) | DPL(0.659) R2adj: 0.055 AICc: 362.82
35 +group(0.101) +infan(0.138) | noctur(0.325) R2adj: 0.068 AICc: 356.73
36 +group(0.056) | arbore(0.955) DPL(0.579) R2adj: 0.017 AICc: 388.92
37 +group(0.077) arbore(0.802) | -noctu(0.274) R2adj: 0.031 AICc: 382.72
38 ++grou(0.047) | DPL(0.429) -noctu(0.287) R2adj: 0.039 AICc: 387.75
39 +infan(0.09) DPL(0.59) arbore(0.796) | R2adj: 0 AICc: 365.92
40 +infan(0.156) arbore(0.976) | -noctu(0.307) R2adj: 0.014 AICc: 359.74
41 +infan(0.085) DPL(0.656) | -noctu(0.301) R2adj: 0.027 AICc: 364.63
42 DPL(0.876) | arbore(0.849) -noctu(0.234) R2adj: -0.023 AICc: 391.46
43 +group(0.117) +mass(0.158) | R2adj: 0.06 AICc: 384.9
44 +mass(0.084) +infan(0.087) | R2adj: 0.075 AICc: 360.05
45 +mass(0.08) arbore(0.92) | R2adj: 0.022 AICc: 387.23
46 +mass(0.077) | DPL(0.912) R2adj: 0.021 AICc: 392.9
47 +mass(0.087) | -noctu(0.288) R2adj: 0.039 AICc: 386.42
48 +group(0.082) +infan(0.122) | R2adj: 0.07 AICc: 360.7
49 +group(0.065) arbore(0.852) | R2adj: 0.029 AICc: 386.96
50 ++grou(0.041) | DPL(0.481) R2adj: 0.038 AICc: 391.9
51 +group(0.065) | -noctu(0.318) R2adj: 0.046 AICc: 386.13
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(0.093) | -noctu(0.266) R2adj: 0.042 AICc: 362.52
55 DPL(0.884) | arbore(0.773) R2adj: -0.031 AICc: 396.09
56 | arbore(0.755) -noctu(0.225) R2adj: -0.006 AICc: 389.19
57 DPL(0.882) | -noctu(0.265) 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')

```

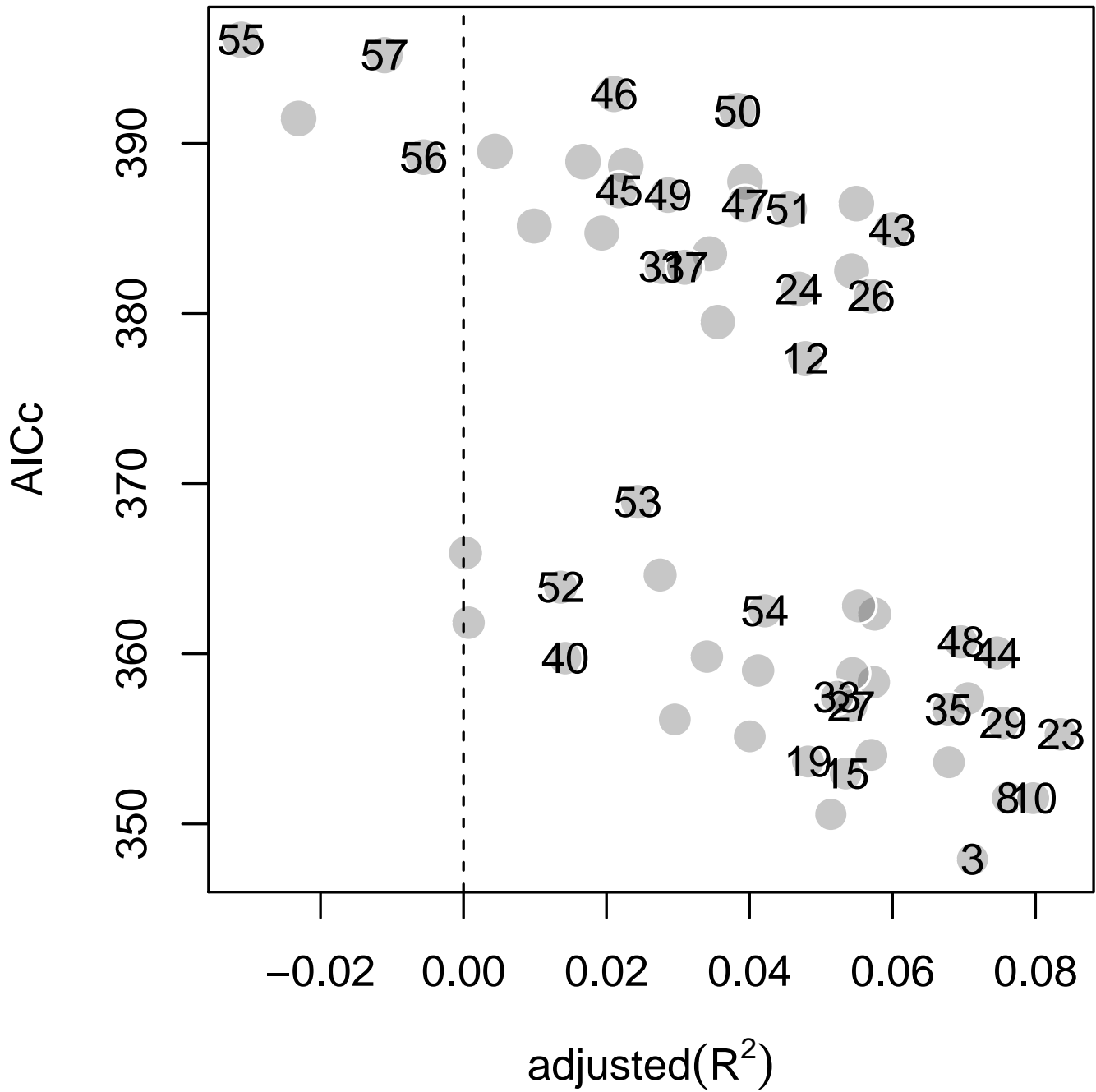


Figure 2: This is a one panel version of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is recommended that all "best" models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

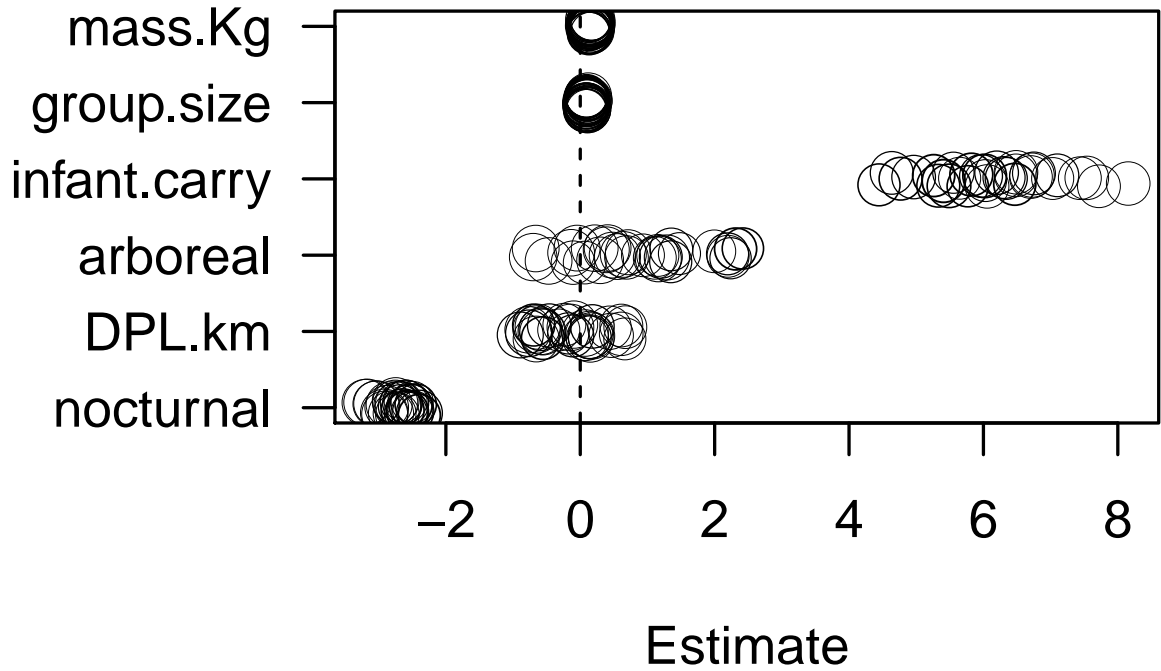
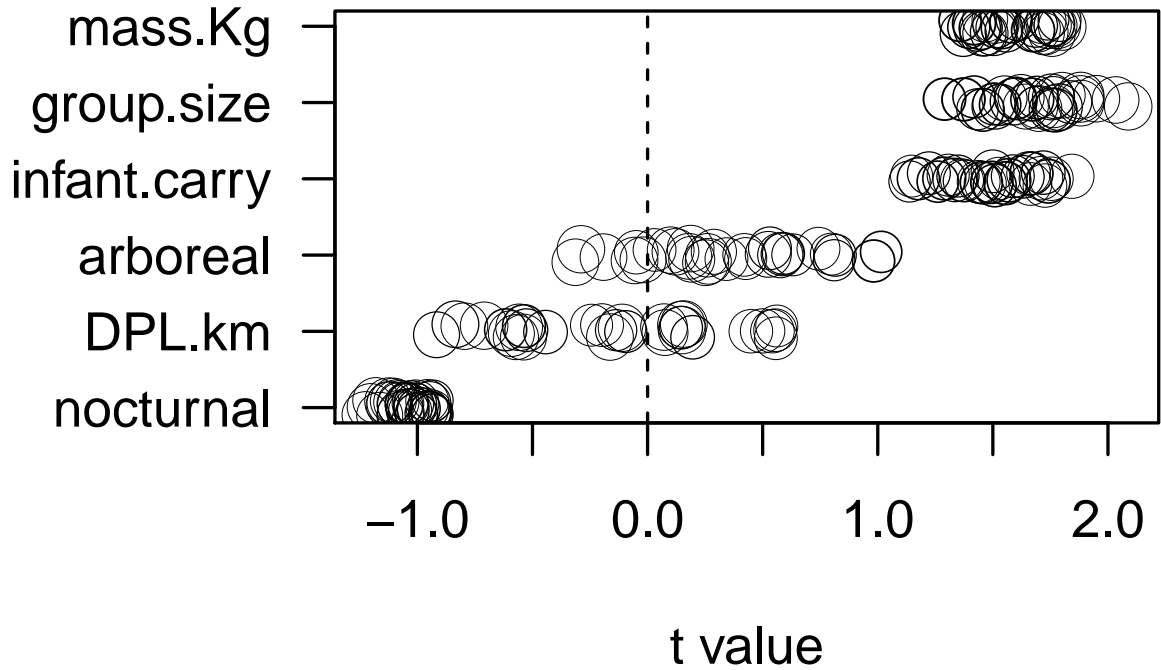


Figure 3: These horizontal parameter distribution dots plots demonstrate how the (t-values of) coefficients from all models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting of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