Tool Use Analysis

Jonathan Pertile

```r
#load packages
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.0      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.0.0 --
v broom        1.0.1     v rsample      1.1.0
v dials        1.0.0     v tune         1.0.0
v infer        1.0.3     v workflows    1.0.0
v modeldata    1.0.0     v workflowsets 1.0.0
v parsnip      1.0.1     v yardstick    1.0.0
v recipes      1.0.1
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Use tidymodels_prefer() to resolve common conflicts.
```

```r
library(knitr)
library(ape)
```

```
Attaching package: 'ape'

The following object is masked from 'package:rsample':

    complement

The following object is masked from 'package:dials':

    degree
```

```r
library(caper)
```

```
Loading required package: MASS

Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select

Loading required package: mvtnorm
```

```r
library(phytools)
```

```
Loading required package: maps

Attaching package: 'maps'

The following object is masked from 'package:purrr':

    map
```

```r
library(nlme)
```

```

Attaching package: 'nlme'

The following object is masked from 'package:dplyr':

    collapse
```

```r
library(gee)
library(TreeTools)
```

```

Attaching package: 'TreeTools'

The following object is masked from 'package:phytools':

    as.multiPhylo
```

```r
library(patchwork)
```

```

Attaching package: 'patchwork'

The following object is masked from 'package:MASS':

    area
```

```r
library(phylolm)
library(broom)
library(readxl)
library(rms)
```

```
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula

Attaching package: 'Hmisc'
```

```
The following object is masked from 'package:ape':

    zoom

The following object is masked from 'package:parsnip':

    translate

The following objects are masked from 'package:dplyr':

    src, summarize

The following objects are masked from 'package:base':

    format.pval, units

Loading required package: SparseM

Attaching package: 'SparseM'

The following object is masked from 'package:base':

    backsolve
```

```r
library(MuMIn)
```

```
Registered S3 methods overwritten by 'MuMIn':
  method         from
  nobs.pgls      caper
  nobs.phylolm   phylolm
  logLik.phylolm phylolm
```

```r
library(kableExtra)
```

```

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
#load data and phylogenetic tree
tool_use_expanded <- read_xlsx("PrimateToolUseDataset.xlsx", sheet = "Dataset")
tooluse.tree <- read.nexus("ToolUsePhylo54.nex")
```

```r
#this ensures that phylogeny and data have species in the same order
species <- tibble(TipLabels(tooluse.tree, single = TRUE)) |>
  rename("Species" = 1)

tool_use_expanded2 <- merge(species, tool_use_expanded, sort = FALSE)

#this turns dummy variables into factors
tool_use_expanded2$Omnivore <- as.factor(tool_use_expanded2$Omnivore)
tool_use_expanded2$Folivore <- as.factor(tool_use_expanded2$Folivore)
tool_use_expanded2$Frugivore <- as.factor(tool_use_expanded2$Frugivore)
```

```r
row.names(tool_use_expanded2) <- tool_use_expanded2$Species
rownames(tool_use_expanded2) == tooluse.tree$tip.label
```

```
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```r
#row names in the database matchup with tree tip labels ??
```

```r
set.seed(1)
global_model <- phyloglm(Tool_Use ~ log(Brain_Weight_grams, 2) +
               log(Group_Size, 2) + Extractive_Foraging + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20, #the limit had been reached
         log.alpha.bound = 8) #increased so alpha doesn't reach bound

summary(global_model)
```

```
Call:
phyloglm(formula = Tool_Use ~ log(Brain_Weight_grams, 2) + log(Group_Size,
    2) + Extractive_Foraging + Dexterity_Index, data = tool_use_expanded2,
    phy = tooluse.tree, method = c("logistic_IG10"), btol = 20,
    log.alpha.bound = 8)
       AIC     logLik Pen.logLik
     68.93     -28.46     -22.55

Method: logistic_IG10
Mean tip height: 73.00302
Parameter estimate(s):
alpha: 17.44132

Coefficients:
                           Estimate   StdErr z.value  p.value
(Intercept)                -3.55206  1.17525 -3.0224 0.002508 **
log(Brain_Weight_grams, 2)  0.41804  0.20777  2.0121 0.044212 *
log(Group_Size, 2)          0.14137  0.21595  0.6547 0.512689
Extractive_Foraging         0.90305  0.65429  1.3802 0.167527
Dexterity_Index            -0.15469  0.33410 -0.4630 0.643354
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Note: Wald-type p-values for coefficients, conditional on alpha=17.44132
```

```r
vif(global_model)
```

```
log(Brain_Weight_grams, 2)         log(Group_Size, 2)
                  1.269393                   1.222721
       Extractive_Foraging            Dexterity_Index
                  1.048763                   1.154054
```

```r
#making modelitos
set.seed(1)

#removing 1 category
modelito2 <- phyloglm(Tool_Use ~
               log(Group_Size, 2) + Extractive_Foraging + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)
```

```
modelito3 <- phyloglm(Tool_Use ~  log(Brain_Weight_grams, 2) +
              Extractive_Foraging + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito4 <- phyloglm(Tool_Use ~  log(Brain_Weight_grams, 2) +
               log(Group_Size, 2) + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito5 <- phyloglm(Tool_Use ~  log(Brain_Weight_grams, 2) +
               log(Group_Size, 2) + Extractive_Foraging,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

#removing 2 categories
modelito6 <- phyloglm(Tool_Use ~ Extractive_Foraging + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito7 <- phyloglm(Tool_Use ~
               log(Group_Size, 2) + Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito8 <- phyloglm(Tool_Use ~
               log(Group_Size, 2) + Extractive_Foraging,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito9 <- phyloglm(Tool_Use ~  log(Brain_Weight_grams, 2) +
              Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito10 <- phyloglm(Tool_Use ~ log(Brain_Weight_grams, 2) +
                Extractive_Foraging,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito11 <- phyloglm(Tool_Use ~ log(Brain_Weight_grams, 2) +
               log(Group_Size, 2),
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

#removing 3 categories
modelito12 <- phyloglm(Tool_Use ~ Dexterity_Index,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito13 <- phyloglm(Tool_Use ~ Extractive_Foraging,
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito14 <- phyloglm(Tool_Use ~ log(Group_Size, 2),
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)

modelito15 <- phyloglm(Tool_Use ~ log(Brain_Weight_grams, 2),
         data = tool_use_expanded2,
         phy = tooluse.tree,
         method = c("logistic_IG10"),
         btol = 20,
         log.alpha.bound = 8)
```

```
aic_table <- rbind(global_model$aic) |>
  rbind(modelito2$aic) |>
  rbind(modelito3$aic) |>
  rbind(modelito4$aic) |>
  rbind(modelito5$aic) |>
  rbind(modelito6$aic) |>
  rbind(modelito7$aic) |>
  rbind(modelito8$aic) |>
  rbind(modelito9$aic) |>
  rbind(modelito10$aic) |>
  rbind(modelito11$aic) |>
  rbind(modelito12$aic) |>
  rbind(modelito13$aic) |>
  rbind(modelito14$aic) |>
  rbind(modelito15$aic)

modelito_aics <- cbind(tibble(1:15), "AIC" = aic_table) |>
  rename("modelito_number" = 1)
```

```
modelito_aics[order(modelito_aics$AIC),]
```

```
 modelito_number      AIC
```

```
10                10 65.50130
15                15 65.87442
5                  5 67.10437
3                  3 67.34915
11                11 67.35071
9                  9 67.89266
1                  1 68.92896
4                  4 69.24813
13                13 71.24784
14                14 71.95937
6                  6 73.16221
7                  7 73.90343
12                12 76.59433
8                  8 79.39972
2                  2 81.79290
```

```
aics <- modelito_aics[order(modelito_aics$AIC),]
for(i in 1:dim(aics)[1]){
aics$diff[i] <- aics$AIC[i]-aics$AIC[1]}

aics$wi <- exp(-0.5*aics$diff)

aics$aic.weights <- aics$wi/sum(aics$wi)
aics <- aics |> dplyr::select(-4)

kable(aics, digits = 4)
```

	modelito_number	AIC	diff	aic.weights
10	10	65.5013	0.0000	0.2600
15	15	65.8744	0.3731	0.2157
5	5	67.1044	1.6031	0.1166
3	3	67.3492	1.8479	0.1032
11	11	67.3507	1.8494	0.1031
9	9	67.8927	2.3914	0.0786
1	1	68.9290	3.4277	0.0468
4	4	69.2481	3.7468	0.0399
13	13	71.2478	5.7465	0.0147
14	14	71.9594	6.4581	0.0103
6	6	73.1622	7.6609	0.0056
7	7	73.9034	8.4021	0.0039
12	12	76.5943	11.0930	0.0010
8	8	79.3997	13.8984	0.0002
2	2	81.7929	16.2916	0.0001

```
summing_aic.w <- aics |> filter(diff < 2)

sum(summing_aic.w$aic.weights)
```

```
[1] 0.7987115
```

The top 12 models produce a summed aic weight above 0.95, so we will reject modelitos 2, 8, and 12. However, there is no aic weight above 0.9, so we will average our models using a global model average.

```
modelito1 <- global_model
```

```
INT_avg <- coefficients(modelito1)[["(Intercept)"]]*filter(aics,
                         modelito_number == 1)$aic.weight +
  coefficients(modelito2)[["(Intercept)"]]*filter(aics,
                         modelito_number == 2)$aic.weight +
  coefficients(modelito3)[["(Intercept)"]]*filter(aics,
                         modelito_number == 3)$aic.weight +
  coefficients(modelito4)[["(Intercept)"]]*filter(aics,
                         modelito_number == 4)$aic.weight +
  coefficients(modelito5)[["(Intercept)"]]*filter(aics,
                         modelito_number == 5)$aic.weight +
  coefficients(modelito6)[["(Intercept)"]]*filter(aics,
                         modelito_number == 6)$aic.weight +
  coefficients(modelito7)[["(Intercept)"]]*filter(aics,
                         modelito_number == 7)$aic.weight +
  coefficients(modelito8)[["(Intercept)"]]*filter(aics,
                         modelito_number == 8)$aic.weight +
  coefficients(modelito9)[["(Intercept)"]]*filter(aics,
                         modelito_number == 9)$aic.weight +
  coefficients(modelito10)[["(Intercept)"]]*filter(aics,
                         modelito_number == 10)$aic.weight +
  coefficients(modelito11)[["(Intercept)"]]*filter(aics,
                         modelito_number == 11)$aic.weight +
  coefficients(modelito12)[["(Intercept)"]]*filter(aics,
                         modelito_number == 12)$aic.weight +
  coefficients(modelito13)[["(Intercept)"]]*filter(aics,
                         modelito_number == 13)$aic.weight +
  coefficients(modelito14)[["(Intercept)"]]*filter(aics,
                         modelito_number == 14)$aic.weight +
  coefficients(modelito15)[["(Intercept)"]]*filter(aics,
                         modelito_number == 15)$aic.weight
```

```
BW_avg <- coefficients(modelito1)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 1)$aic.weight +
  coefficients(modelito3)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 3)$aic.weight +
  coefficients(modelito4)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 4)$aic.weight +
  coefficients(modelito5)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 5)$aic.weight +
  coefficients(modelito9)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 9)$aic.weight +
  coefficients(modelito10)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 10)$aic.weight +
  coefficients(modelito11)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 11)$aic.weight +
  coefficients(modelito15)[["log(Brain_Weight_grams, 2)"]]*filter(aics,
                         modelito_number == 15)$aic.weight
```

```
GS_avg <- coefficients(modelito1)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 1)$aic.weight +
  coefficients(modelito2)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 2)$aic.weight +
  coefficients(modelito4)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 4)$aic.weight +
  coefficients(modelito5)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 5)$aic.weight +
  coefficients(modelito7)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 7)$aic.weight +
  coefficients(modelito8)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 8)$aic.weight +
  coefficients(modelito11)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 11)$aic.weight +
  coefficients(modelito14)[["log(Group_Size, 2)"]]*filter(aics,
                         modelito_number == 14)$aic.weight
```

```
EF_avg <- coefficients(modelito1)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 1)$aic.weight +
  coefficients(modelito2)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 2)$aic.weight +
  coefficients(modelito3)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 3)$aic.weight +
  coefficients(modelito5)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 5)$aic.weight +
  coefficients(modelito6)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 6)$aic.weight +
  coefficients(modelito8)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 8)$aic.weight +
  coefficients(modelito10)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 10)$aic.weight +
  coefficients(modelito13)[["Extractive_Foraging"]]*filter(aics,
                         modelito_number == 2)$aic.weight
```

```
DI_avg <- coefficients(modelito1)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 1)$aic.weight +
  coefficients(modelito2)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 2)$aic.weight +
  coefficients(modelito3)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 3)$aic.weight +
  coefficients(modelito4)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 4)$aic.weight +
  coefficients(modelito6)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 6)$aic.weight +
  coefficients(modelito7)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 7)$aic.weight +
  coefficients(modelito9)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 9)$aic.weight +
  coefficients(modelito12)[["Dexterity_Index"]]*filter(aics,
                         modelito_number == 12)$aic.weight
```

```
variables <- tibble(c("(Intercept)", "log(Brain_Weight_grams, 2)", "log(Group_Size, 2)",
  "Extractive_Foraging", "Dexterity_Index"))

estimates <- tibble(INT_avg) |> rbind(BW_avg) |> rbind(GS_avg) |> rbind(EF_avg) |> rbind(D
  rename(Weighted_Estimate = 1)

Weighted_Model_pre <- variables |> cbind(estimates) |> rename(
  Variable = 1
)
```

```
#calculate sum of weights for variables
INT_p <- NA

BW_p <- filter(aics, modelito_number == 1)$aic.weight +filter(aics,
                         modelito_number == 3)$aic.weight + filter(aics,
                         modelito_number == 4)$aic.weight + filter(aics,
                         modelito_number == 5)$aic.weight + filter(aics,
                         modelito_number == 9)$aic.weight + filter(aics,
                         modelito_number == 10)$aic.weight + filter(aics,
                         modelito_number == 11)$aic.weight + filter(aics,
                         modelito_number == 15)$aic.weight

GS_p <- filter(aics, modelito_number == 1)$aic.weight +filter(aics,
                         modelito_number == 2)$aic.weight + filter(aics,
                         modelito_number == 4)$aic.weight + filter(aics,
                         modelito_number == 5)$aic.weight + filter(aics,
                         modelito_number == 7)$aic.weight + filter(aics,
                         modelito_number == 8)$aic.weight + filter(aics,
                         modelito_number == 11)$aic.weight + filter(aics,
                         modelito_number == 14)$aic.weight

EF_p <- filter(aics, modelito_number == 1)$aic.weight +filter(aics,
                         modelito_number == 2)$aic.weight + filter(aics,
                         modelito_number == 3)$aic.weight + filter(aics,
                         modelito_number == 5)$aic.weight + filter(aics,
                         modelito_number == 6)$aic.weight + filter(aics,
                         modelito_number == 8)$aic.weight + filter(aics,
                         modelito_number == 10)$aic.weight + filter(aics,
                         modelito_number == 13)$aic.weight

DI_p <- filter(aics, modelito_number == 1)$aic.weight +filter(aics,
                         modelito_number == 2)$aic.weight + filter(aics,
                         modelito_number == 3)$aic.weight + filter(aics,
                         modelito_number == 4)$aic.weight + filter(aics,
                         modelito_number == 6)$aic.weight + filter(aics,
                         modelito_number == 7)$aic.weight + filter(aics,
                         modelito_number == 9)$aic.weight + filter(aics,
                         modelito_number == 12)$aic.weight
Akaike_variable_weight <- c(INT_p, BW_p, GS_p, EF_p, DI_p)

Weighted_Model <- Weighted_Model_pre |> cbind(Akaike_variable_weight)
```

```
Weighted_Model
```

```
                    Variable Weighted_Estimate Akaike_variable_weight
1                (Intercept)       -3.24262069                     NA
2 log(Brain_Weight_grams, 2)        0.44708985              0.9641368
3         log(Group_Size, 2)        0.04925934              0.3210610
4        Extractive_Foraging        0.50307780              0.5473453
5            Dexterity_Index       -0.02663981              0.2792573
```

Interpreting results:

```
logodds_INT <- -3.242
odds_INT <- exp(logodds_INT)

logodds_BW <- 0.447
odds_BW <- exp(logodds_BW)


logodds_GS <- 0.049
odds_GS <- exp(logodds_GS)
```

Variable	Weighted_Estimate	Weighted_Odds_Estimate	Akaike_variable_weight
(Intercept)	-3.243	0.039	NA
log(Brain_Weight_grams, 2)	0.447	1.564	0.964
log(Group_Size, 2)	0.049	1.050	0.321
Extractive_Foraging	0.503	1.654	0.547
Dexterity_Index	-0.027	0.973	0.279

```
logodds_EF <- 0.503
odds_EF <- exp(logodds_EF)


logodds_DI <- -0.027
odds_DI <- exp(logodds_DI)

Weighted_Estimate_as_odds <- tibble(c(odds_INT, odds_BW,
                                      odds_GS, odds_EF, odds_DI)) |>
  rename(Weighted_Odds_Estimate = 1)
Weighted_Model_post <- Weighted_Model |> cbind(Weighted_Estimate_as_odds)|>
  dplyr::select(1, 2, 4, 3)
kable(Weighted_Model_post, digits = 3) |>
  kable_styling(font_size = 10)
```

Holding all other variables constant, we expect that, on average, with every doubling in brain mass the odds of observed tool use behavior is higher by a factor of 1.564.

Holding all other variables constant, we expect that, on average, with every doubling in group size the odds of observed tool use behavior is higher by a factor of 1.050.

Holding all other variables constant, we expect that the odds of observed tool use behavior in primates that exhibit extractive foraging is higher than primates that do not by a factor of 1.654.

Holding all other variables constant, we expect that, on average, for every 1 point increase in dexterity index, the odds of observed tool use behavior are lower by a factor of 0.973.