

Lab 3:  
Parasite  
Specificity +  
Host Range



# Today's Lab

1. Go over basics of data manipulation in R
2. Code along example working with GMPD
3. Set you free to answer your own question



# What is R and why should I care about knowing how to use it?

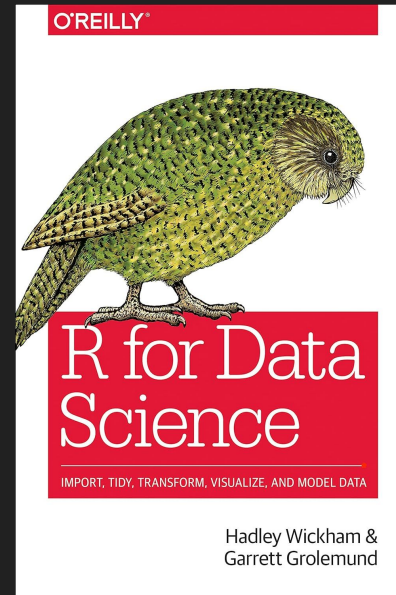
Statistical programming language

Based on precursor language called S

Good at working with data

- Manipulation
- Analysis
- Visualization

Free, Open-Sourced, Reproducible!



# Basic Terminology

**R** is the language, we'll be interacting with it through **RStudio** (an IDE: Integrated Development Environment)

The image shows a screenshot of the RStudio IDE interface. The main window is divided into several panes:

- Source Editor (R script):** Contains R code for biomass calculation per tree across various plots. The code includes comments and function calls like `plot(kalimantan$dbh, kalimantan$points)`. A white box labeled "R script" is overlaid on this pane.
- Environment:** Shows the current R environment with variables like `h1.trees` (716 obs. of 23 variables), `kal.plot` (94 obs. of 18 variables), `kalimantan` (1993 obs. of 44 variables), and `lst.plots` (59 obs. of 19 variables). A white box labeled "R environment" is overlaid on this pane.
- Console (R console):** Shows the execution of R commands. The first command is `kal.plot <- merge(kalimantan, dmed, dmed.plot, by = "plot")`. Subsequent commands calculate the square root of biomass and write the results to a CSV file. A white box labeled "R console" is overlaid on this pane.
- Plots:** Displays a box plot titled "Biomass estimation per plot with different models". The y-axis is labeled "Biomass (Mg/ha<sup>1</sup>)" and ranges from 100 to 500. The plot shows biomass distribution for various plots, with a white box labeled "Graphical output" overlaid on the plot area.

# Basic Terminology

**R Scripts:** The file where your code lives! Creating scripts allows you to know exactly what you did, and reproduce it every time.

The image shows a screenshot of the RStudio interface. A red arrow points from the 'R Scripts' text to the script editor. The script editor contains R code for biomass calculation and plotting. The R environment pane shows a list of objects. The R console shows the execution of R commands. The graphical output pane shows a box plot titled 'Biomass estimation per plot with different models'.

**R script**

```
200 # Biomass calculation per tree
201 kalimantansw.brown<-brown.moist.d(kalimantansdbh)
202 kalimantansw.yamakura<-yamakura.stem(kalimantansdbh, kalimantansh)+yamakura.branch(yamakura.stem(k
203 kalimantansw.basuki<-basuki.mixed.d(kalimantansdbh)
204 kalimantansw.samalca<-samalca.d(kalimantansdbh)
205 kalimantansw.hashimoto<-hashimoto.d(kalimantansdbh)
206 kalimantansw.kenzo<-kenzo.d(kalimantansdbh)
207 kalimantansw.forda<-forda.d(kalimantansdbh)
208 kalimantansw.jaya<-jaya.d(kalimantansdbh)
209 kalimantansw.novita<-novita.d(kalimantansdbh)
210 kalimantansw.nugroho.d<-nugroho.d(kalimantansdbh)
211 kalimantansw.nugroho.d.h<-nugroho.d.h(kalimantansdbh)
212
213
214 plot(kalimantansdbh, kalimantansh, xlab="DBH", ylab="Biomass (Mg/ha)", col=1:11)
215 points(kalimantansdbh, kalimantansw.brown, col=1)
216 points(kalimantansdbh, kalimantansw.yamakura, col=2)
217 points(kalimantansdbh, kalimantansw.basuki, col=3)
218 points(kalimantansdbh, kalimantansw.samalca, col=4)
219 points(kalimantansdbh, kalimantansw.hashimoto, col=5)
220 points(kalimantansdbh, kalimantansw.kenzo, col=6)
221 points(kalimantansdbh, kalimantansw.forda, col=7)
222 points(kalimantansdbh, kalimantansw.jaya, col=8)
223 points(kalimantansdbh, kalimantansw.novita, col=9)
224 points(kalimantansdbh, kalimantansw.nugroho.d, col=10)
225 points(kalimantansdbh, kalimantansw.nugroho.d.h, col=11)
226
227 legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "Kenzo", "Forda", "Jaya",
228
229 # Summing all values per plot and nested plot
230 bio.plot.brown<-as.data.frame(tapply(kalimantansw.brown, list(kalimantansplot_id, kalimantanssubpl
231
310:1 (Untitled) R Script
```

**R environment**

Object	Value
Global Environment +	
hil.trees	716 obs. of 23 variables
kal.plot	94 obs. of 18 variables
kalimantan	1993 obs. of 44 variables
lst.plots	59 obs. of 19 variables
lst	
pub	
wet	
valu	
EF	12.43397 992.96783
EFm	49.7359197162173
EFS	198.943678864869
N_tot	2696.5863280181

**R console**

```
> kal.plot<-merge(kal.plot, dmed, dmed.plot, by="Plot")
>
> # calculating the
> kal.plot$dg<-sqrt((4*kal.plot$
>
> write.csv(kal.plot, "kalimanta
>
```

**Graphical output**

Biomass estimation per plot with different models

# Basic Terminology

**R Environment:** List of all the objects you're currently working with, loaded into your memory. Data objects, custom functions, etc.

The image shows a screenshot of the RStudio interface. The main window displays an R script with code for biomass calculation and plotting. The console shows the execution of R commands. The environment pane lists objects in the global environment, including 'hi1.trees', 'kal.plot', 'kalimantan', and 'lst.plots'. The graphical output pane shows a box plot titled 'Biomass estimation per plot with different models'.

**R script**

```
201 # biomass calculation per tree
202 kalimantan$w.brown~brown.moist.d(kalimantansdbh)
203 kalimantan$w.yamakura~yamakura.stem(kalimantansdbh, kalimantan$w.yamakura.branch(yamakura.stem(k
204 kalimantan$w.basuki~basuki.mixed.d(kalimantansdbh)
205 kalimantan$w.samalca~samalca.d(kalimantansdbh)
206 kalimantan$w.hashimoto~hashimoto.d(kalimantansdbh)
207 kalimantan$w.kenzo~kenzo.d(kalimantansdbh)
208 kalimantan$w.forda~forda.d(kalimantansdbh)
209 kalimantan$w.jaya~jaya.d(kalimantansdbh)
210 kalimantan$w.novita~novita.d(kalimantansdbh)
211 kalimantan$w.nugroho.d~nugroho.d(kalimantansdbh)
212 kalimantan$w.nugroho.d.h~nugroho.d.h(kalimantansdbh)
213
214 plot(kalimantansdbh, kalimantan$w.brown, col=1)
215 points(kalimantansdbh, kalimantan$w.yamakura, col=2)
216 points(kalimantansdbh, kalimantan$w.basuki, col=3)
217 points(kalimantansdbh, kalimantan$w.samalca, col=4)
218 points(kalimantansdbh, kalimantan$w.hashimoto, col=5)
219 points(kalimantansdbh, kalimantan$w.kenzo, col=6)
220 points(kalimantansdbh, kalimantan$w.forda, col=7)
221 points(kalimantansdbh, kalimantan$w.jaya, col=8)
222 points(kalimantansdbh, kalimantan$w.novita, col=9)
223 points(kalimantansdbh, kalimantan$w.nugroho.d, col=10)
224 points(kalimantansdbh, kalimantan$w.nugroho.d.h, col=11)
225
226 legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "Kenzo", "Forda", "Jaya",
227
228
229 # Summing all values per plot and nested plot
230 bio.plot.brown~as.data.frame(tapply(kalimantan$w.brown, list(kalimantan$plot_id, kalimantansubpl
231
310:1 [Untitled] R Script
```

**R console**

```
> kal.plot<-merge(kal.plot, dmed.hmed.plot, by="Plot")
>
> # calculating the
> kal.plot$dg<-sqrt((4*kal.plot$
>
> write.csv(kal.plot, "kalimanta
>
```

**R environment**

Object	Obs	Vars
hi1.trees	716	23
kal.plot	94	18
kalimantan	1993	44
lst.plots	59	19

**Graphical output**

Biomass estimation per plot with different models

Biomass (Mg?ha<sup>-1</sup>)

# Basic Terminology

**R Console:** Shows the most recent code and commands you've run. You can also type a run code here, but it's not recorded like in a script.

The image shows a screenshot of the RStudio interface. A red arrow points from the R Console to the R script editor. The R script editor contains R code for biomass calculation and plotting. The R Console shows the execution of the code. The R Environment pane shows the objects created in the environment. The Graphical Output pane shows a box plot of biomass estimation per plot with different models.

**R script**

```
201 # biomass calculation per tree
202 kalimantan$w.brown~brown.moist.d(kalimantansdbh)
203 kalimantan$w.yamakura~yamakura.stem(kalimantansdbh, kalimantansh)+yamakura.branch(yamakura.stem(k
204 kalimantan$w.basuki~basuki.mixed.d(kalimantansdbh)
205 kalimantan$w.samalca~samalca.d(kalimantansdbh)
206 kalimantan$w.hashimoto~hashimoto.d(kalimantansdbh)
207 kalimantan$w.kenzo~kenzo.d(kalimantansdbh)
208 kalimantan$w.forda~forda.d(kalimantansdbh)
209 kalimantan$w.jaya~jaya.d(kalimantansdbh)
210 kalimantan$w.novita~novita.d(kalimantansdbh)
211 kalimantan$w.nugroho.d~nugroho.d(kalimantansdbh)
212 kalimantan$w.nugroho.d.h~nugroho.d.h(kalimantansdbh)
213
214 plot(kalimantansdbh, kalimantansh, main="Biomass estimation per plot with different models", xlab="DBH
215 points(kalimantansdbh, kalimantansh, col="brown", size=10)
216 points(kalimantansdbh, kalimantansh, col="darkred", size=10)
217 points(kalimantansdbh, kalimantansh, col="darkgreen", size=10)
218 points(kalimantansdbh, kalimantansh, col="darkblue", size=10)
219 points(kalimantansdbh, kalimantansh, col="darkmagenta", size=10)
220 points(kalimantansdbh, kalimantansh, col="darkcyan", size=10)
221 points(kalimantansdbh, kalimantansh, col="darkorange", size=10)
222 points(kalimantansdbh, kalimantansh, col="darkgrey", size=10)
223 points(kalimantansdbh, kalimantansh, col="black", size=10)
224 points(kalimantansdbh, kalimantansh, col="white", size=10)
225
226 legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "kenzo", "Forda", "Jaya",
227
228
229 # Summing all values per plot and nested plot
230 bio.plot.brown~as.data.frame(tapply(kalimantans$w.brown, list(kalimantansplot_id, kalimantanssubpl
231
310:1 (Untitled) : R Script
```

**R console**

```
> kal.plot<-merge(kal.plot, Dmed.Hmed.plot, by="Plot")
> # calculating the
> kal.plot$dg<-sqrt((4*kal.plot$
>
> write.csv(kal.plot, "kalimanta
>
```

**R environment**

Object	Size
h1.trees	716 obs. of 23 variables
kal.plot	94 obs. of 18 variables
kalimantan	1993 obs. of 44 variables
lst.plots	59 obs. of 19 variables

**Graphical output**

Biomass estimation per plot with different models

# Basic Terminology

**Other Stuff:** There are a few tabs here, all of them useful. Most of the names are self-explanatory, and we'll go over them in class

The image shows a screenshot of the RStudio interface. The main window is divided into four panes:

- R script:** The top-left pane shows R code for biomass calculation per tree, including data loading and plotting functions.
- R console:** The bottom-left pane shows the execution of R commands, such as `kal.plot<-merge(kal.plot, dmed, dmed.plot, by="Plot")` and `write.csv(kal.plot, "Kalimantan")`.
- R environment:** The top-right pane shows the current environment with variables like `hi1.trees` (716 obs. of 23 variables), `kal.plot` (94 obs. of 18 variables), `kalimantan` (1993 obs. of 44 variables), and `ts1.plots` (59 obs. of 19 variables).
- Graphical output:** The bottom-right pane shows a box plot titled "Biomass estimation per plot with different models". The y-axis is labeled "Biomass (Mg/ha<sup>-1</sup>)" and ranges from 100 to 500. The plot displays several box plots representing different models, with whiskers extending to the minimum and maximum values.




# Our Dataset

**ECOLOGY**  
ECOLOGICAL SOCIETY OF AMERICA

Data Papers |  Free Access

## Global Mammal Parasite Database version 2.0

Patrick R. Stephens  Paula Pappalardo, Shan Huang, James E. Byers, Maxwell J. Farrell, Alyssa Gehman, Ria R. Ghai, Sarah E. Haas, Barbara Han, Andrew W. Park, John P. Schmidt ... [See all authors](#) ▾

First published: 08 March 2017 | <https://doi.org/10.1002/ecy.1799> | Citations: 79

[Find Text](#)

Corresponding Editor: William K. Michener.

<https://parasites.nunn-lab.org/>

Hosts: Ungulates,  
Carnivores, and  
Primates





# Our Dataset

**ECOLOGY**  
ECOLOGICAL SOCIETY OF AMERICA

Data Papers |  Free Access

## Global Mammal Parasite Database version 2.0

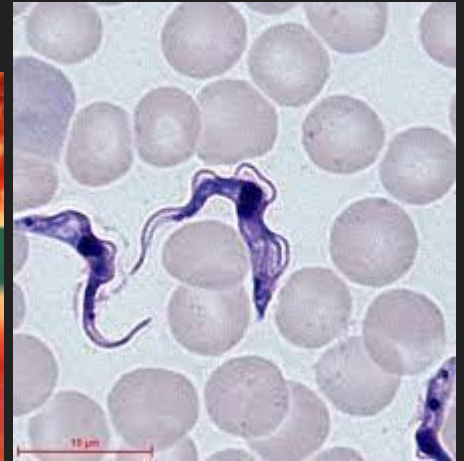
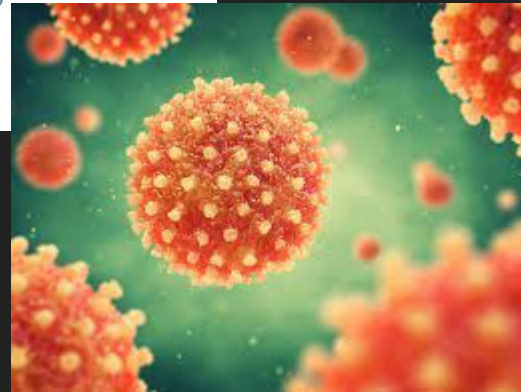
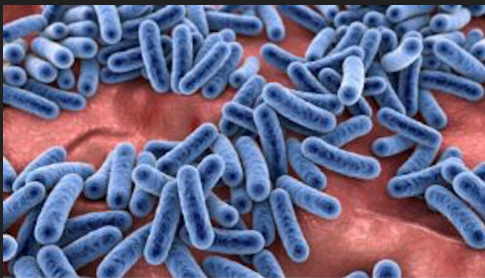
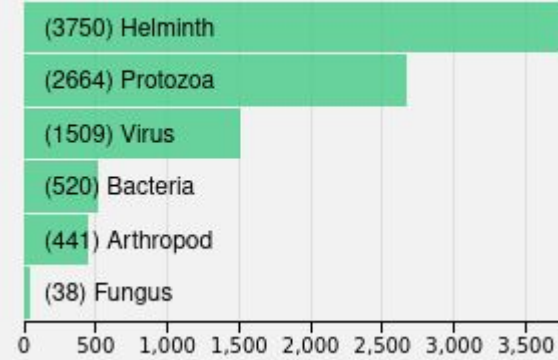
Patrick R. Stephens  Paula Pappalardo, Shan Huang, James E. Byers, Maxwell J. Farrell, Alyssa Gehman, Ria R. Ghai, Sarah E. Haas, Barbara Han, Andrew W. Park, John P. Schmidt ... [See all authors](#) 

First published: 08 March 2017 | <https://doi.org/10.1002/ecy.1799> | Citations: 79

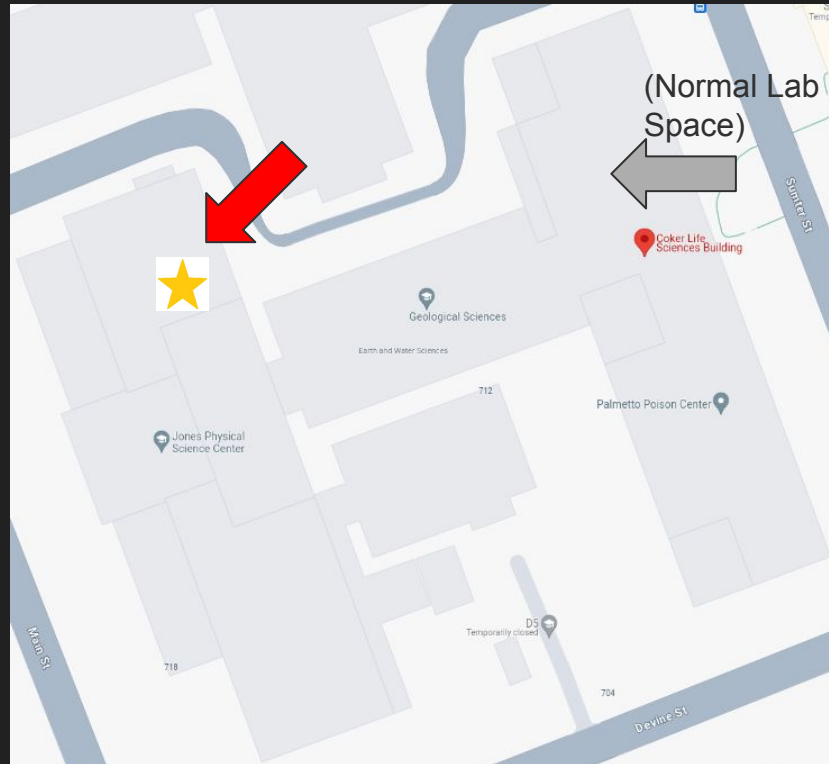
Find Text

Corresponding Editor: William K. Michener.

### Parasite Type



# Where is lab today?



The Question:

What are predictors of parasite richness across mammalian hosts?

# Data I've given you

Host traits: extracted from PanTHERIA

Morphometric data: Adult body mass, neonatal body mass, adult forearm length, basal metabolic rate...

Pace of Life data: Gestation length, litter size, age at eyes opening, age at first reproduction...

Life-History/ecological data: Habitat breadth, population density, trophic level

Geographic Range: Total range size, mean latitude, max/min latitude, temperature or PET across range...

**Read the MetaData!!!**

## **PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals**

Ecological Archives E090-184

Kate E. Jones [✉](#) Jon Bielby, Marcel Cardillo, Susanne A. Fritz, Justin O'Dell, C. David L. Orme, Kamran Safi, Wes Sechrest, Elizabeth H. Boakes, Chris Carbone, Christina Connolly, Michael J. Cutts, Janine K. Foster, Richard Grenyer, Michael Habib, Christopher A. Plaster, Samantha A. Price



# Data I've given you

## Host phylogenetic information

Based on Phylogenetic Tree From Upham, Esselstyn, and Jetz (2019).

How do we convert this to something we can use for prediction?

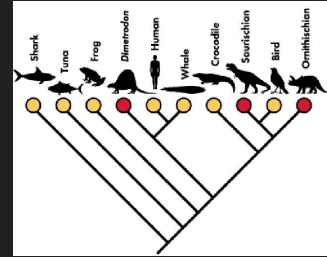
Different approaches, but the one I chose was eigenvalue decomposition

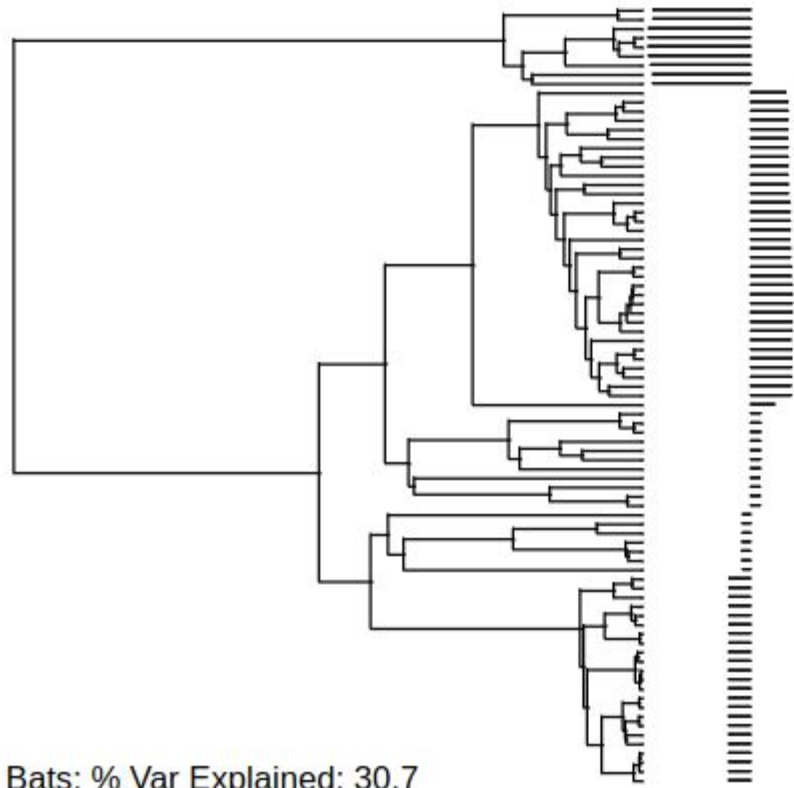
Long story short, a way of condensing a complicated phylogeny into a series of continuous axis. You lose information doing this, but what's left can still be quite useful!

## Inferring the mammal tree: Species-level sets of phylogenies for questions in ecology, evolution, and conservation

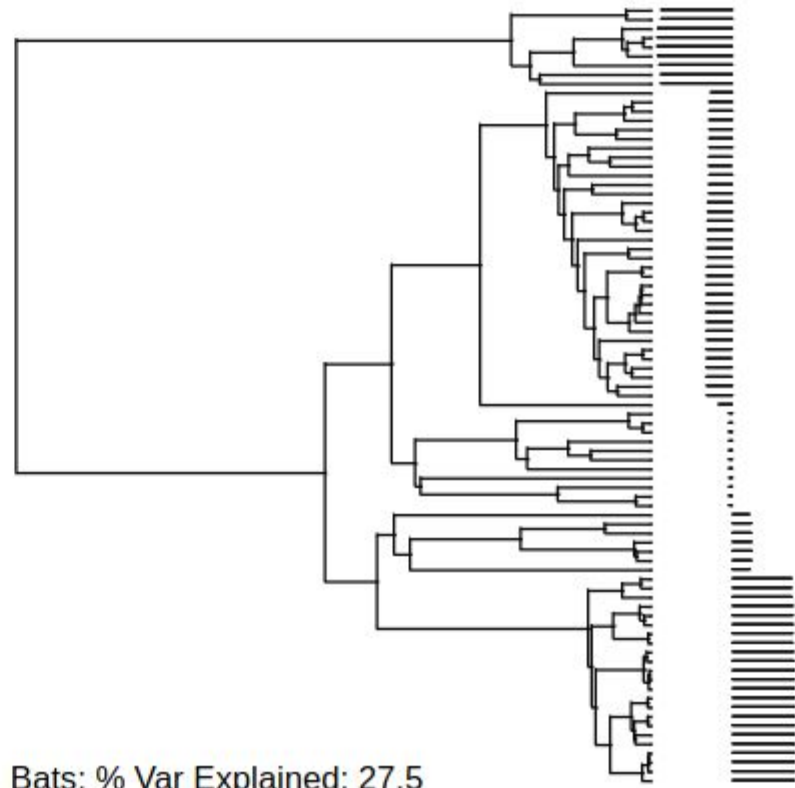
Nathan S. Upham , Jacob A. Esselstyn, Walter Jetz 

Published: December 4, 2019 • <https://doi.org/10.1371/journal.pbio.3000494>





Bats; % Var Explained: 30.7



Bats; % Var Explained: 27.5

# Generalized Linear Models

Really a general term, describing a really simple framework

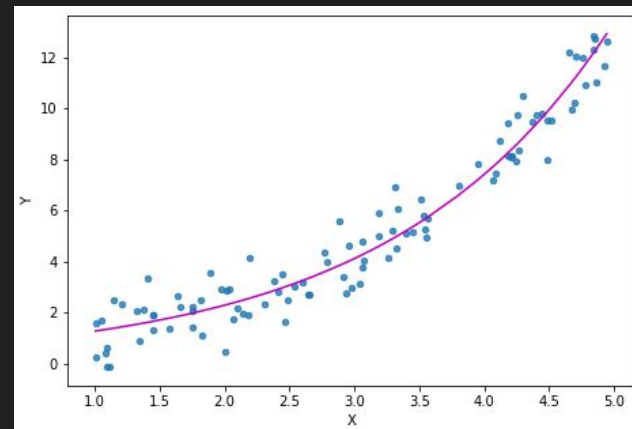
Data = Model + Error

$$\hat{Y} = \beta_0 + \beta_1 X$$

If this looks like linear regression, that's because it is!

The “Generalized” form however just means that we're specific about our assumptions about error distributions in a way that means we don't just have to use continuous numerical predictors.

R takes care of the majority of this for us, but we need to know how to interpret them





# Generalized Linear Models in R

In R, we can make a glm with the `glm()` function (go figure!)

?`glm()` will show you that there's many arguments, but the most important ones for our purposes are the "formula" and "data"

When you use `summary()` to look at a glm model, it'll look something like this...

This has a lot of information, but the ones I want you to look at for interpretation are...

```
Call:
glm(formula = richness ~ log(AdultBodyMass), family = "poisson",
     data = hostDat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.181  -4.222  -2.206   1.602  26.334

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.237456   0.054979   22.51  <2e-16 ***
log(AdultBodyMass) 0.181173   0.005309   34.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10478.6  on 372  degrees of freedom
Residual deviance:  9307.4  on 371  degrees of freedom
  (103 observations deleted due to missingness)
AIC: 10807

Number of Fisher Scoring iterations: 5
```

# Generalized Linear Models in R

## 1. The coefficient estimates

- These describe the relationships between your predictors and the response variable
- Can use these to understand whether the relationship is positive or negative (for our case—a poisson link function, mean  $y$  changes by  $\exp(\beta_1)$  per unit change of  $x$ )

```
Call:
glm(formula = richness ~ log(AdultBodyMass), family = "poisson",
     data = hostDat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.181  -4.222  -2.206   1.602  26.334

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.237456  0.054979   22.51  <2e-16 ***
log(AdultBodyMass) 0.181173  0.005309   34.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10478.6  on 372  degrees of freedom
Residual deviance:  9307.4  on 371  degrees of freedom
(103 observations deleted due to missingness)
AIC: 10807

Number of Fisher Scoring iterations: 5
```

# Generalized Linear Models in R

1. The coefficient estimates
  - a. These describe the relationships between your predictors and the response variable
  - b. Can use these to understand whether the relationship is positive or negative (for our case-a poisson link function, mean y changes by  $\exp(\beta_1)$  per unit change of x)
2. P-values; give us an idea of significance
  - a.  $P > 0.05$ , then the term is significant (and helps explain variation in Y).

```
Call:
glm(formula = richness ~ log(AdultBodyMass), family = "poisson",
     data = hostDat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.181  -4.222  -2.206   1.602  26.334

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.237456   0.054979   22.51  <2e-16 ***
log(AdultBodyMass) 0.181173   0.005309   34.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10478.6  on 372  degrees of freedom
Residual deviance:  9307.4  on 371  degrees of freedom
(103 observations deleted due to missingness)
AIC: 10807

Number of Fisher Scoring iterations: 5
```

# Generalized Linear Models in R

1. The coefficient estimates
  - a. These describe the relationships between your predictors and the response variable
  - b. Can use these to understand whether the relationship is positive or negative (for our case-a poisson link function, mean  $y$  changes by  $\exp(\beta_1)$  per unit change of  $x$ )
2. P-values; give us an idea of significance
  - a.  $P > 0.05$ , then the term is significant (and helps explain variation in  $Y$ ).
3. Akaike Information Criterion (AIC)

```
Call:
glm(formula = richness ~ log(AdultBodyMass), family = "poisson",
     data = hostDat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.181  -4.222  -2.206   1.602  26.334

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.237456   0.054979   22.51  <2e-16 ***
log(AdultBodyMass) 0.181173   0.005309   34.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10478.6  on 372  degrees of freedom
Residual deviance:  9307.4  on 371  degrees of freedom
(103 observations deleted due to missingness)
AIC: 10807

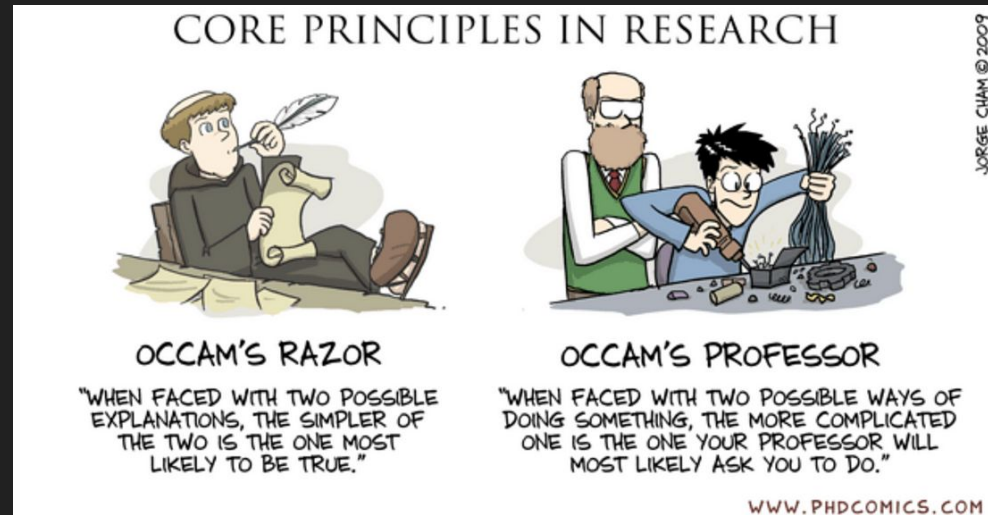
Number of Fisher Scoring iterations: 5
```

# Model Comparison through AIC

When we're comparing models, there's often a tradeoff between **goodness of fit** and **parsimony**.

The idea of **parsimony** stems from Occam's razor-when comparing among equally supported explanations, the one that requires the fewest assumptions is usually correct.

There are many ways you can evaluate model performance in light of these two ideas, but one of the most common is **AIC**



# Model Comparison through AIC

## AIC: Akaike Information Criterion

$$AIC = 2k - 2\ln(\hat{L})$$

$AIC$  = Akaike information criterion

$k$  = number of estimated parameters in the model

$\hat{L}$  = maximum value of the likelihood function for the model



The lower the AIC score, the better.

Models are penalized by the number of parameters ( $k$ ), but rewarded by goodness of fit ( $L$ )

Model with the lowest AIC score generally has most support given the data, though if the difference between models is  $<2$  you can't really distinguish between them

Note: You can only compare models trained on the same data! (Otherwise the comparison is meaningless!)

## Mystery Host(s)

How many parasites do you think they have?



We can make a guess using the `predict()` function!