Practical Statistics

Part 2: Building a toolkit

Intro

- **Henry Häkkinen**
  - Post-Doc studying the effects of climate change on seabirds and how to plan effective conservation. Latterly branched out into impacts of renewable energy on biodiversity.

PART 0: Recap of Session 1

- We covered basic statistical theory
  - Normal distribution
  - Student's t-distribution
  - Running t-tests, chi-squared tests, linear regression
  - Residuals and model fit
  - Understanding and interpreting output (estimates, p-values, etc.)

In this session…

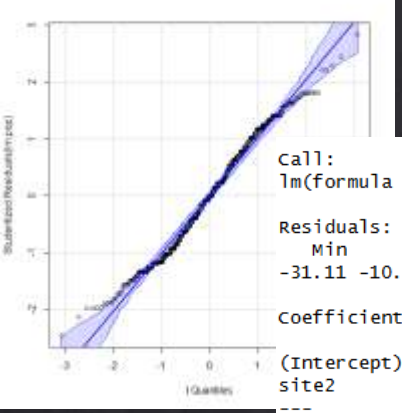
- Building on the skills from part 1
- Creating a statistical toolkit
  - More distributions and GLMs
  - Mixed effects, and other common model extensions
  - Non-parametric models
  - Model fit and model comparisons
- Expanding horizons: Other frameworks for looking at problems
- Work through examples in R.

What is not in this session…

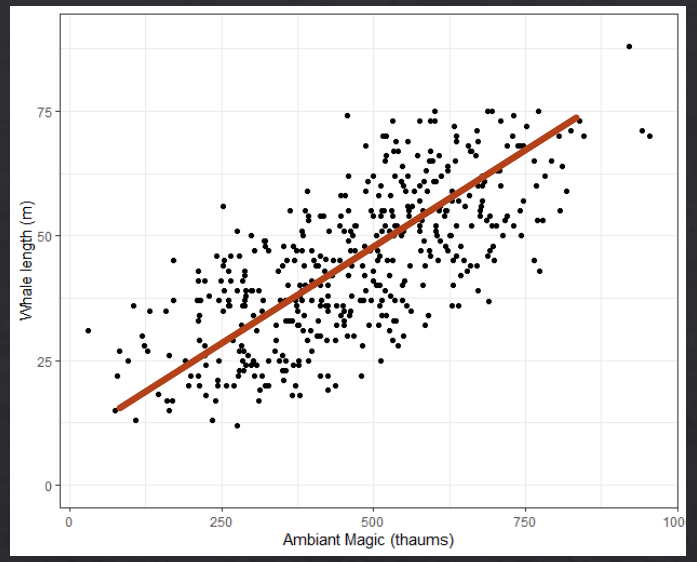
- Lots of mathematics
- Detail on lots of individual tests
- Comprehensive coverage of all statistical concepts.

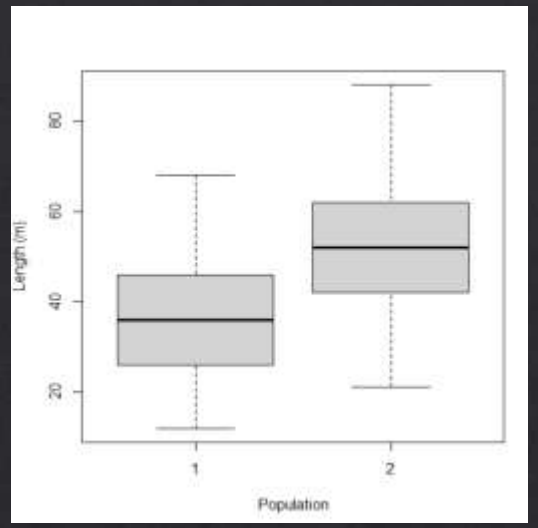
PART 1: Generalised Linear Models

Source: https://www.kristakingmath.com/



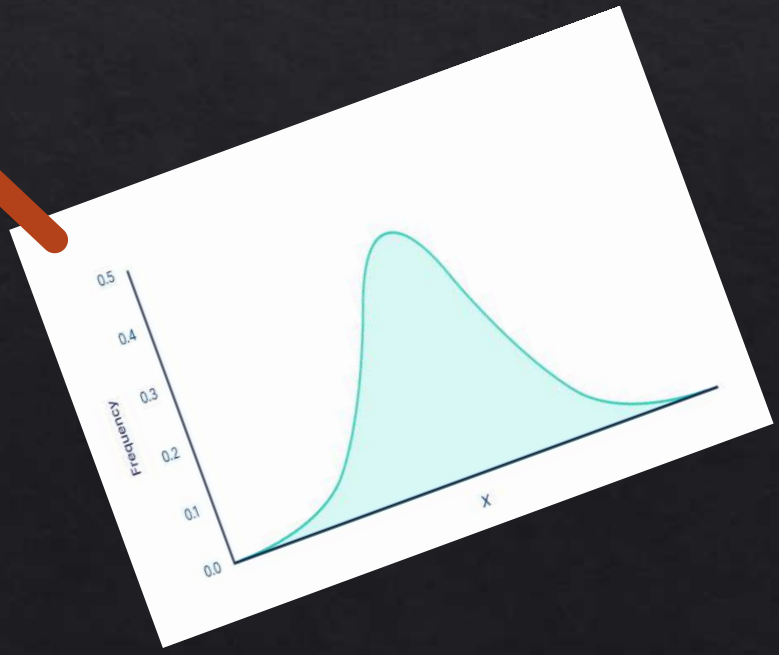
```
Call:
lm(formula = length_m ~ site, data = df1)

Residuals:
    Min      1Q  Median      3Q     Max
-31.11 -10.93  -0.11    9.89  35.89

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.9288     0.8605   42.92   <2e-16 ***
site2        15.1811     1.2169   12.48   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.82 on 442 degrees of freedom
Multiple R-squared:  0.2604,    Adjusted R-squared:  0.2587
F-statistic: 155.6 on 1 and 442 DF,  p-value: < 2.2e-16
```





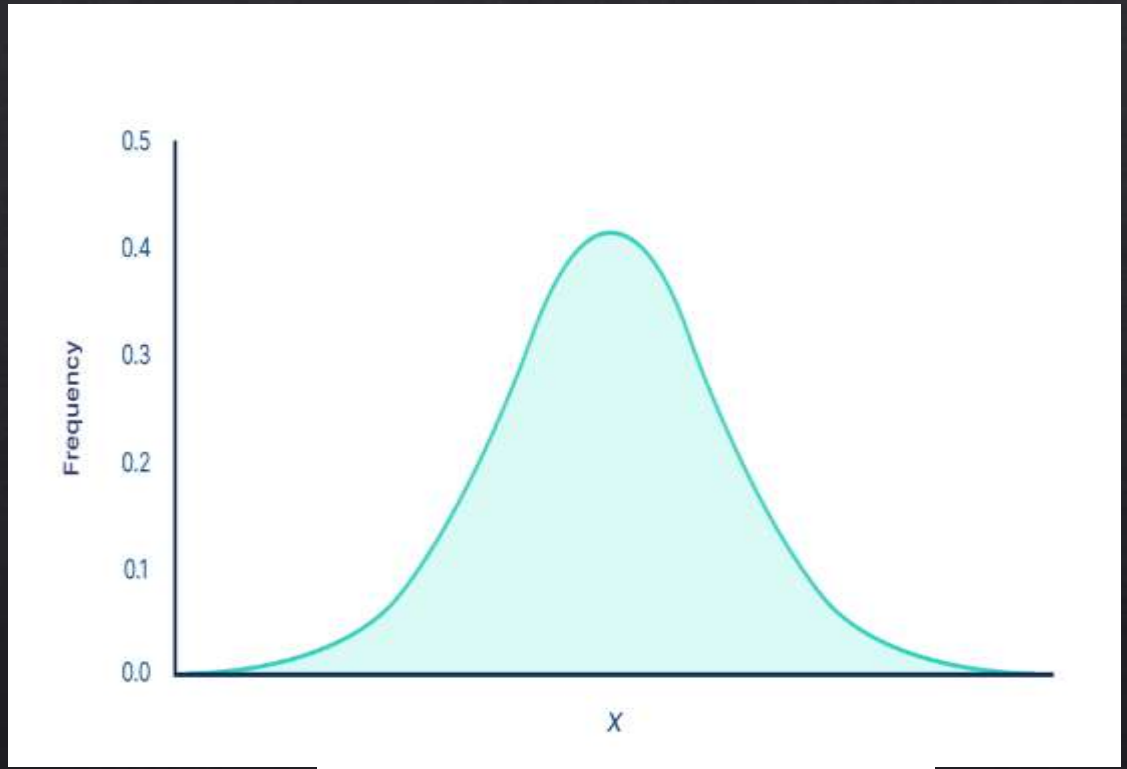
	A	B
1	weight (g)	feed
2	179	horsebean
3	160	horsebean
4	136	horsebean
5	227	horsebean
6	217	horsebean
7	168	horsebean
8	108	horsebean
9	124	horsebean
10	143	horsebean
11	140	horsebean
12	309	horsebean
13	229	horsebean
14	181	linseed
15	141	linseed
16	260	linseed
17	203	linseed
18	148	linseed
19	169	linseed
20	213	linseed
21	257	linseed
22	244	linseed
23	271	linseed
24	243	linseed
25	230	linseed
26	248	soybean
27	327	soybean
28	329	soybean
29	250	soybean
		soybean
		soybean

- Formula: y ~ x + group
- Null: “there is no difference between different groups/sites/treatments” – “there is no significant correlation between x and y”



Data distributions: normal

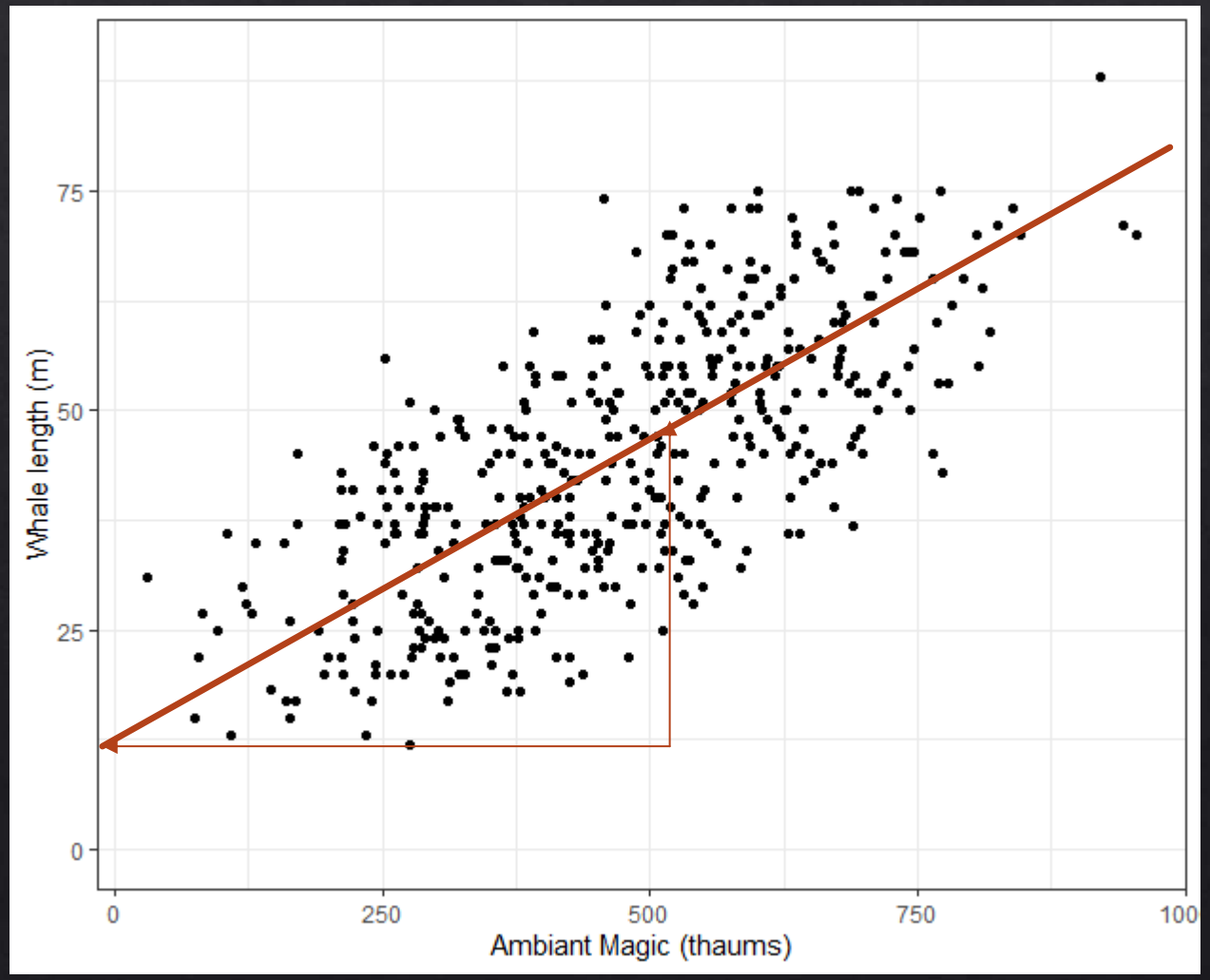
- **Normal (aka Gaussian) distribution**
- Continuous response variable:
  - Weight of population
  - Time of migration arrival
  - Intelligence
- Two parameters: mean (μ) and standard deviation (σ)
- Link function: none (identity)
  - $y_i \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$



$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Data distributions: normal

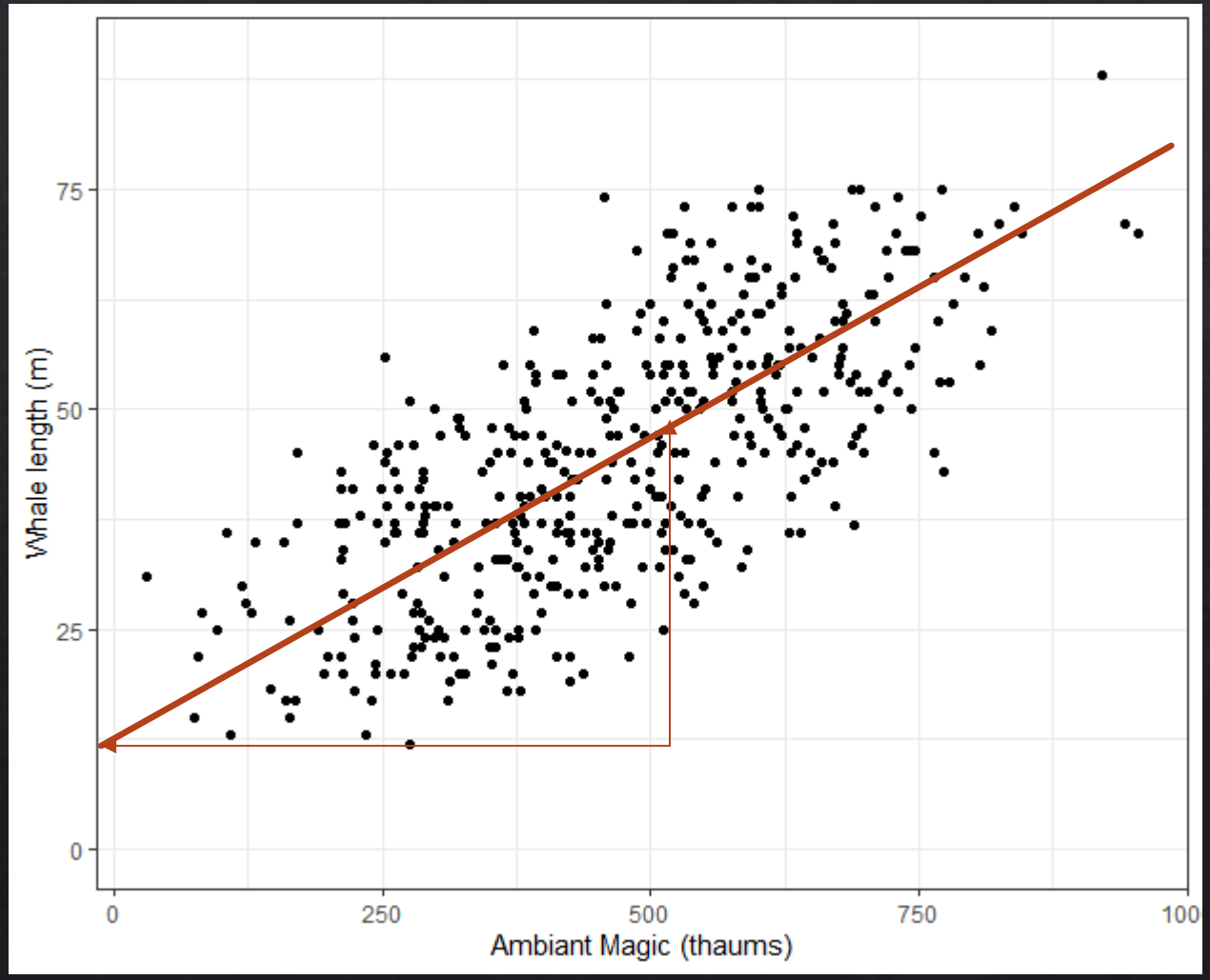
- **Normal (aka Gaussian) distribution**
- Two parameters: mean (μ) and standard deviation (σ)
- Link function: none (identity)
  - $y_i \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$



Data distributions: normal

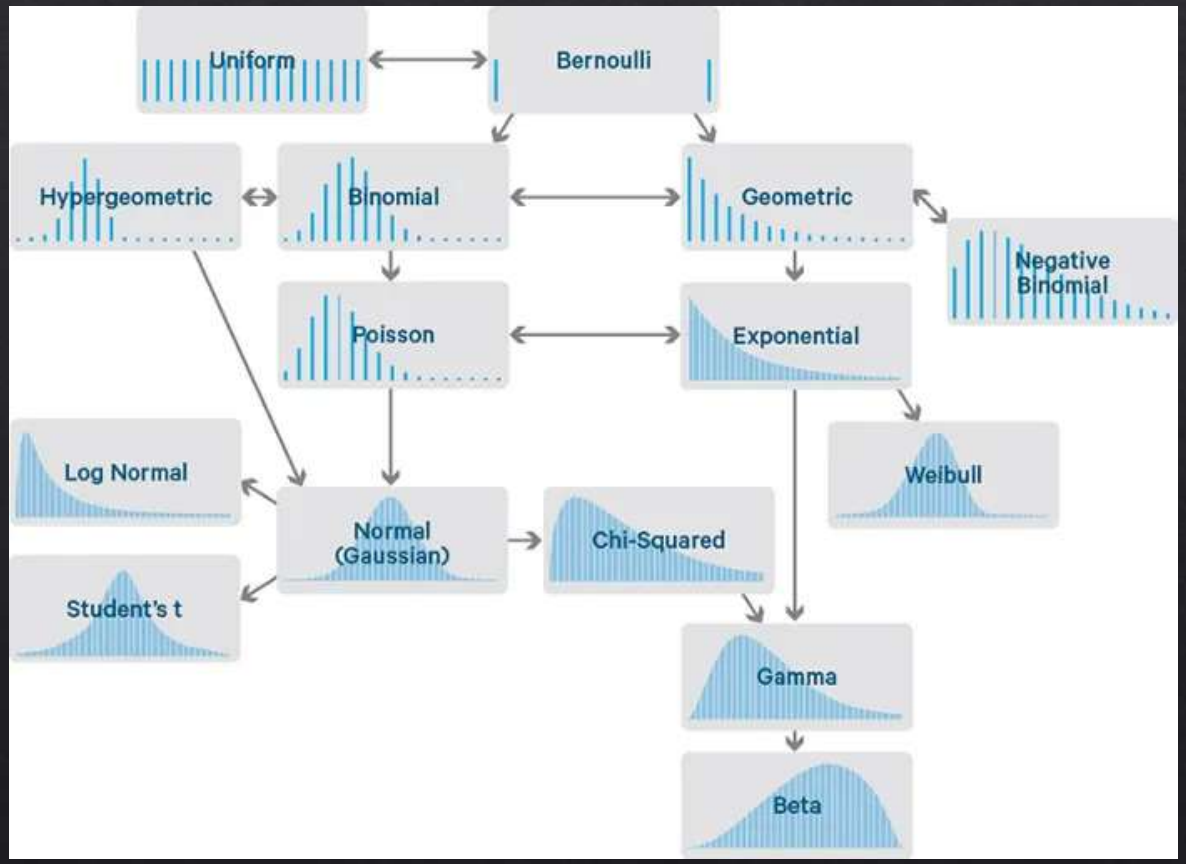
- **Normal (aka Gaussian) distribution**
- Two parameters: mean (μ) and standard deviation (σ)
- Link function: none (identity)
  - $y_i \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$

```
Call:
lm(formula = length_m ~ magic_thaums, data = df1)

Residuals:
     Min       1Q   Median       3Q      Max
-23.1046  -8.3972   0.1768   7.8501  30.2264

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  15.737262   1.462982   10.76   <2e-16 ***
magic_thaums  0.061293   0.002927   20.94   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.56 on 442 degrees of freedom
Multiple R-squared:  0.498,     Adjusted R-squared:  0.4969
F-statistic: 438.5 on 1 and 442 DF,  p-value: < 2.2e-16
```



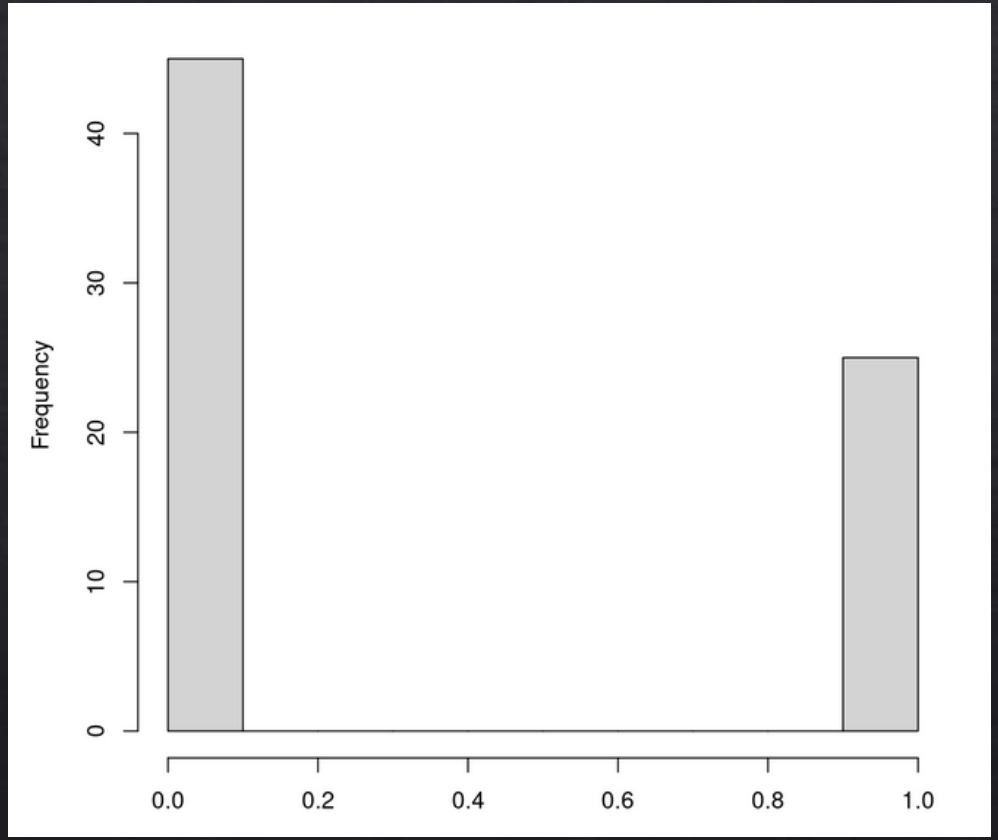
What is a Generalised Linear Model

- Previously we have spoken about tests and their differences.
- Different types of response variables sometimes require different distributions
  - Continuous
  - Count
  - Categorical (ranked/unranked)
  - Binary
- Luckily, there is a general solution, named GLMS.
- All GLMs work the same way, they just have different underlying distributions!



Data distributions: binomial

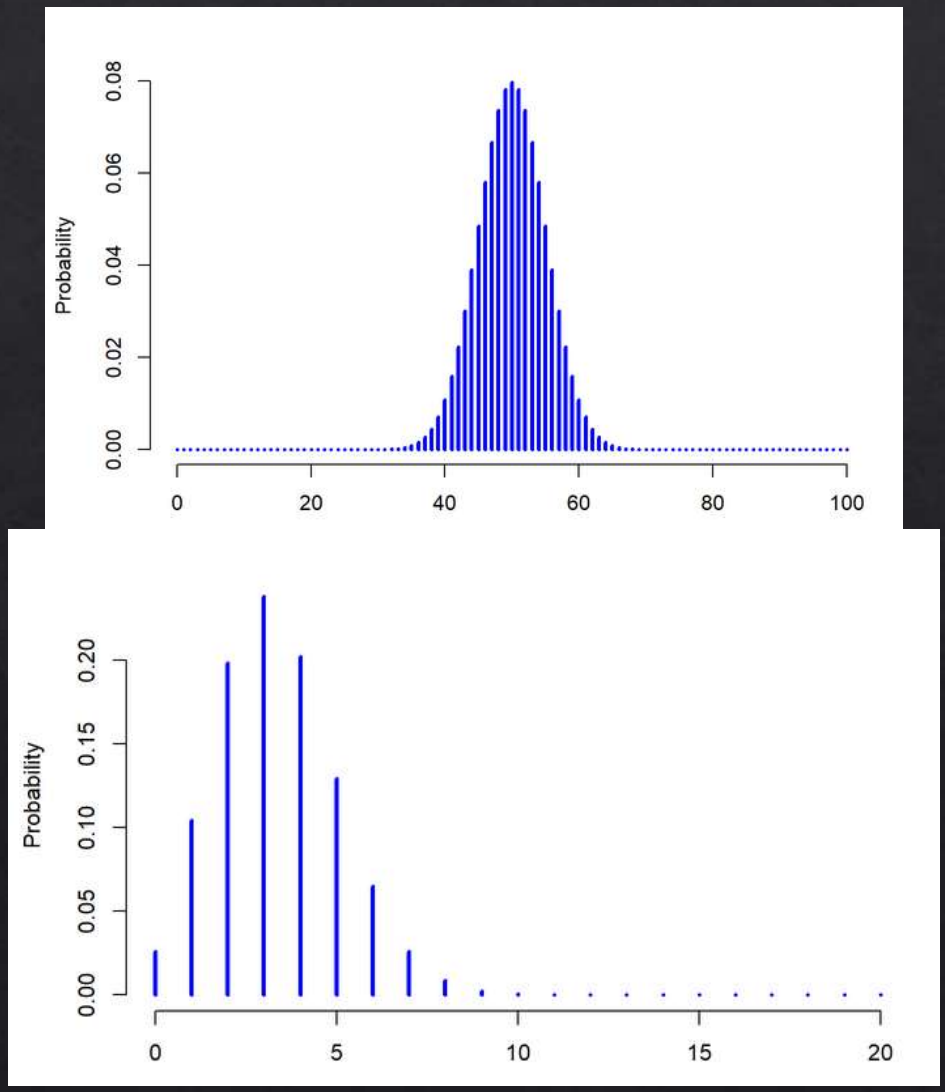
- **Binomial distribution**
- Binary response variable (either 0 or 1):
  - Species presence/absence
  - Survival
  - Disease occurrence
- Two parameters: number of trials (n) and probability of success (p).
- Link function: logit or log
  - $\text{logit}(p_i) \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$



$$f(k,n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Data distributions: binomial

- **Binomial distribution**
- Data example: Does local availability of water and topography predict presence of mites?
- Response variable: mites (0/1)
- Predictor variables:
  - Water availability (continuous)
  - Topography (categorical)



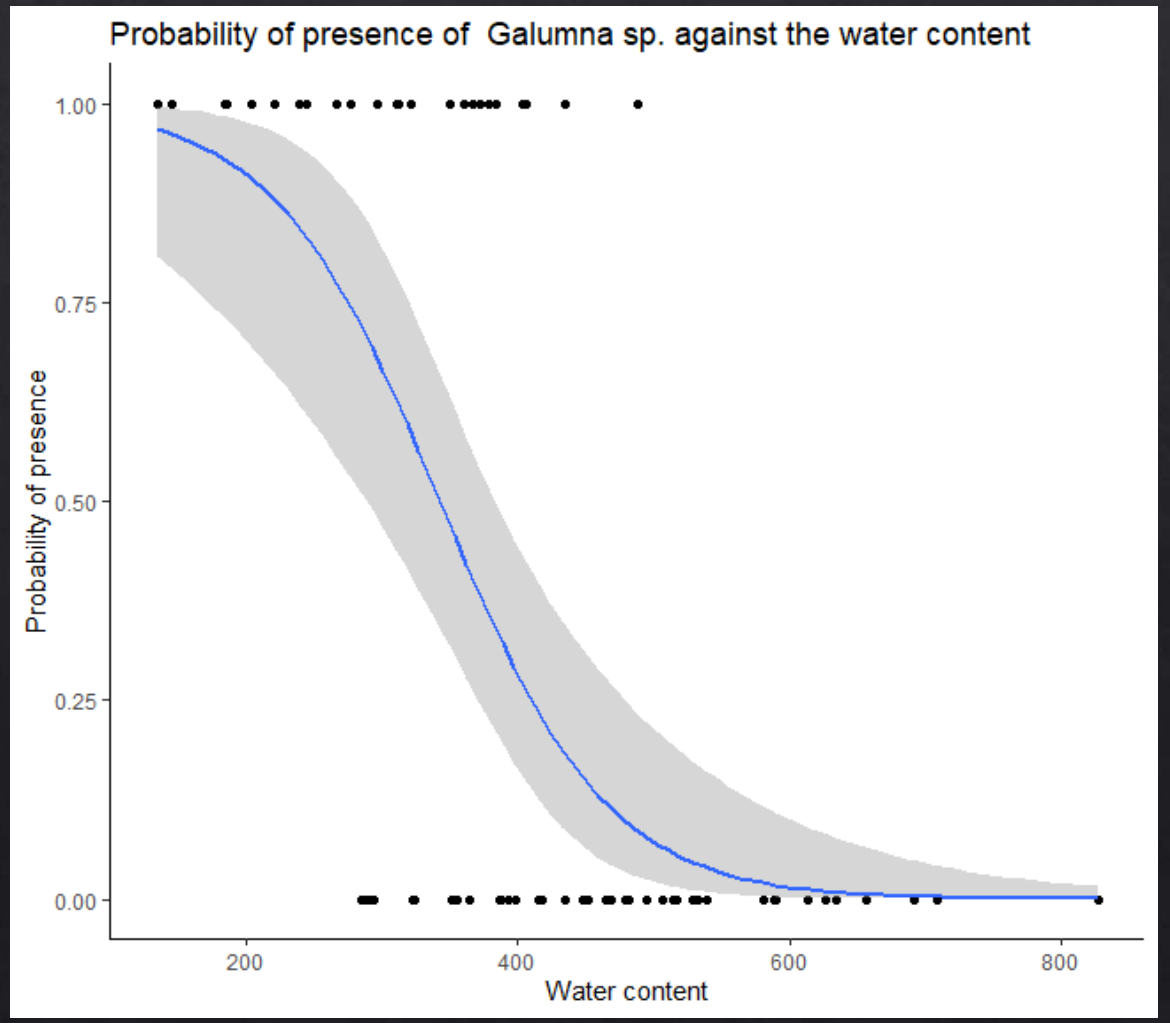
Data distributions: binomial

```
Call:
glm(formula = pa ~ WatrCont_s + Topo, family = binomial(link = "logit"),
    data = mite1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.0288     0.5639  -3.598 0.000321 ***
WatrCont_s   -2.2511     0.6456  -3.487 0.000489 ***
TopoHummock   2.0908     0.7353   2.843 0.004466 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 91.246  on 69  degrees of freedom
Residual deviance: 48.762  on 67  degrees of freedom
AIC: 54.762

Number of Fisher Scoring iterations: 6
```

- Link function: logit or log $\text{logit}(p_i) \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$

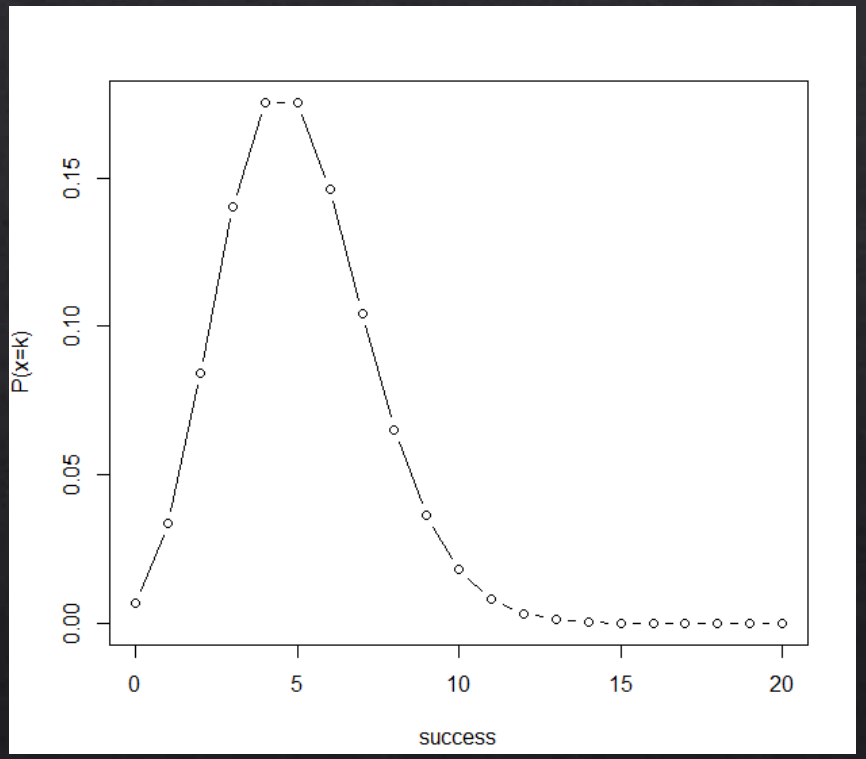
Data distributions: binomial

- Many GLMS can be difficult to interpret just based on raw output.
- Always remember estimates are on the link function scale!
- Plotting is the easiest way to work out what is going on



Data distributions: count data

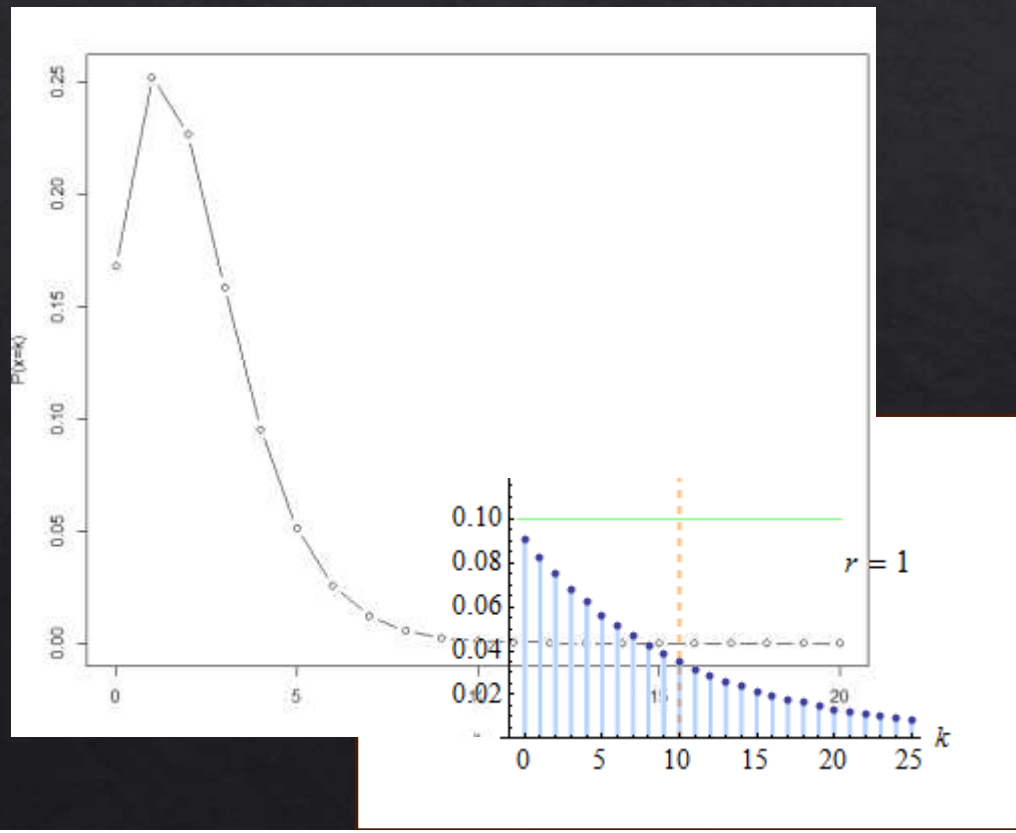
- **Poisson distribution**
- Count response variable (where events are relatively rare):
  - Number of sightings
  - Number of mutations on a strand of DNA
  - Number of extreme climate events per year
- One parameter: Number of events that occur in a fixed time (λ)
- Link function: log or logit
  - $\log(\lambda_i) \sim a + b_1 x_{1,i} + b_2 x_{2,i} \ldots$
- Very simple, but very restrictive! Often fails to fit data



$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

Data distributions: count data

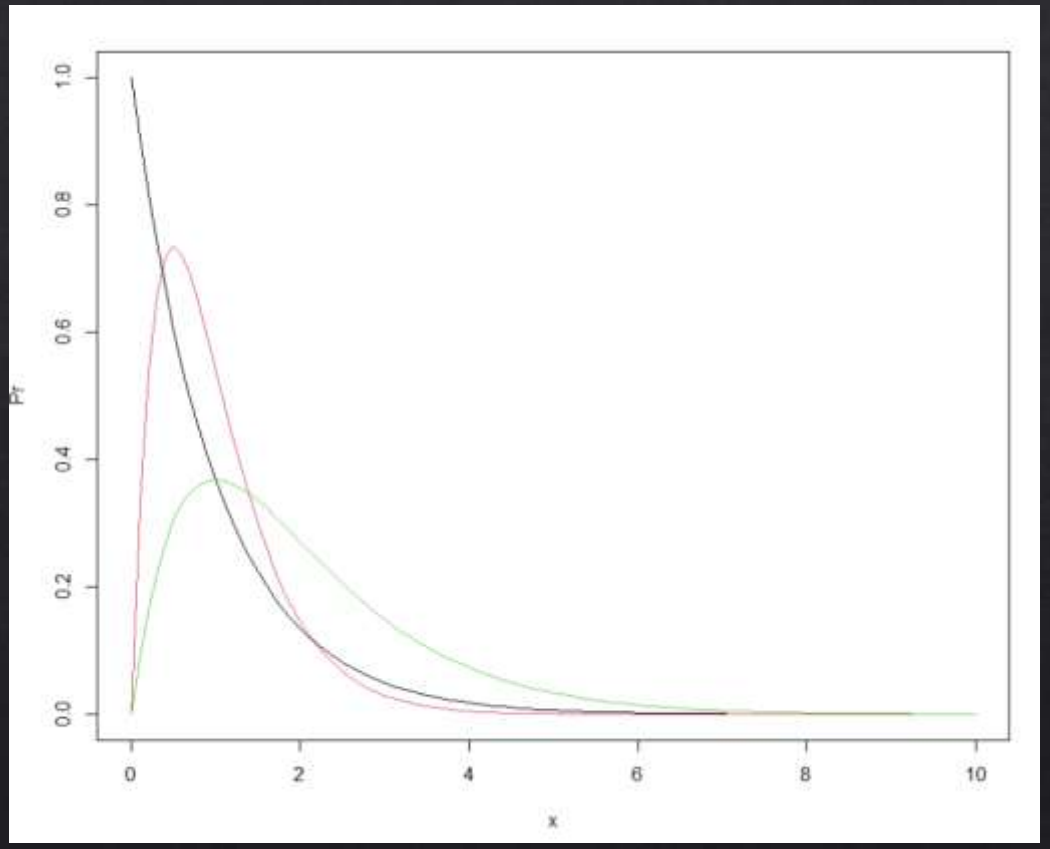
- **Negative Binomial distribution**
- Count response variable:
  - Species abundance
  - Parasite load
  - Number of precipitation events
- Two parameter: Number of events (r), and probability of event occurring (p)
- Link function: log or logit
  - $\log(y) \sim a + b_1x + b_2x \ldots$
- Much more flexible than Poisson!



$$f(k;r,p) \equiv \Pr(X = k) = \binom{k+r-1}{k}(1-p)^k p^r$$

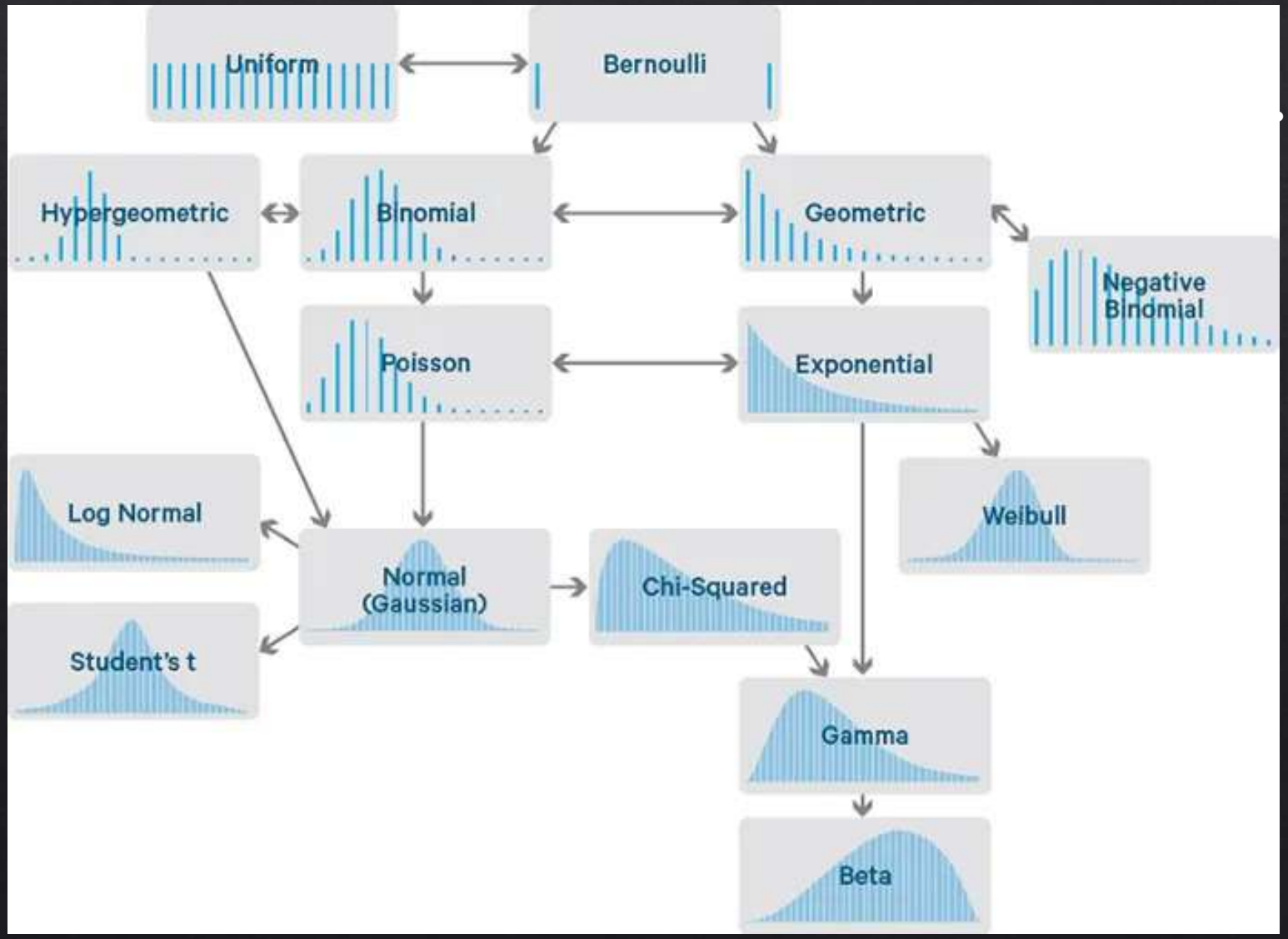
Data distributions: gamma

- **Gamma distribution**
- Used for positive continuous data, but with a skew
  - Basically, anything that you would use a normal distribution for, but with a strong skew!
- Two parameters: shape (α) and scale (θ). 1/θ is known as the rate (λ)
- Link function: negative reciprocal
  - $-y^{-1} \sim a + b_1x + b_2x \ldots + \epsilon$
- Drawback: non-intuitive and difficult to interpret residuals



$$f(x;\alpha,\lambda) = \frac{x^{\alpha-1}e^{-\lambda x}\lambda^{\alpha}}{\Gamma(\alpha)} \quad \text{for } x > 0 \quad \alpha,\lambda > 0,$$

And so on…



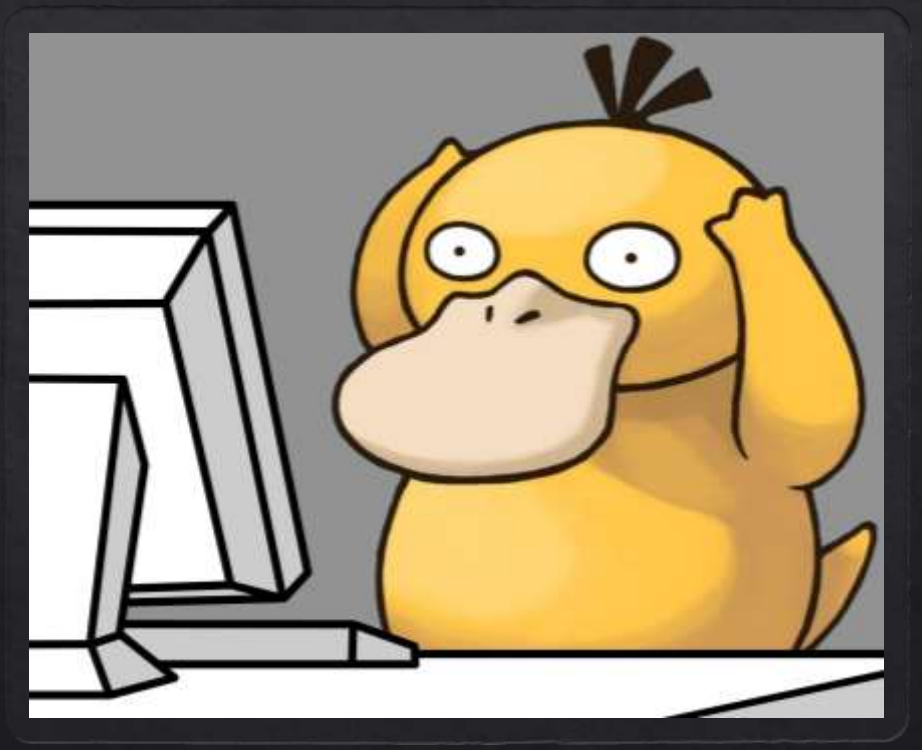
- + Modifications!
  - Quasi-
  - Zero-inflated
  - Inverse
  - Mixture distributions

Source: https://awstip.com/statistical-probability-distributions-89398c4b68c7

Statistical Models: General Approach

- Steps to making a statistical test:
  1. Think about our study system, identify our response and our predictors and make a formula
  2. Make plots of our data (especially our response variable), and select a distribution.
     i. Check our data fulfils the assumptions of the distribution
  3. Run chosen statistical test (or possibly several from a shortlist)
  4. Check the diagnostics and residuals
     i. Consider alternative distributions if necessary
  5. Interpret our output

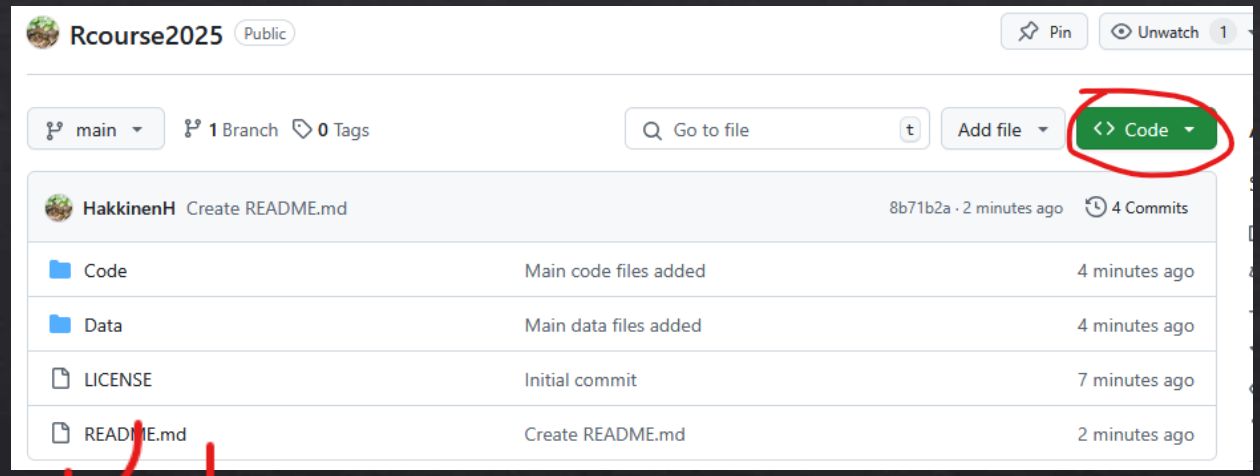
Questions

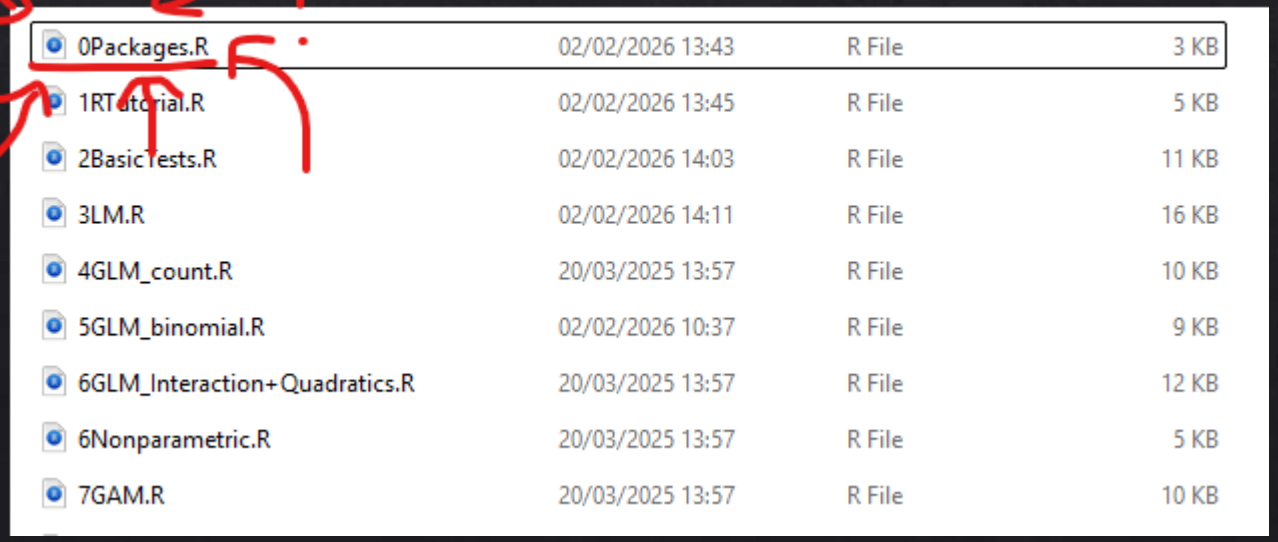


The Practical: Part 1

https://github.com/HakkinenH/RStatsCourse2026

- Code -> download ZIP -> Unzip and put somewhere useful
- Run 0Packages.R first ((if you haven't already) to install packages
- There are a number of example GLMs and exercises, work through them in whatever order you like:
  - 3LM.R (GLM with normal distribution)
  - 4GLM_count.R (Poisson/Quasipoisson/NB)
  - 5GLM_binomial (binomial)





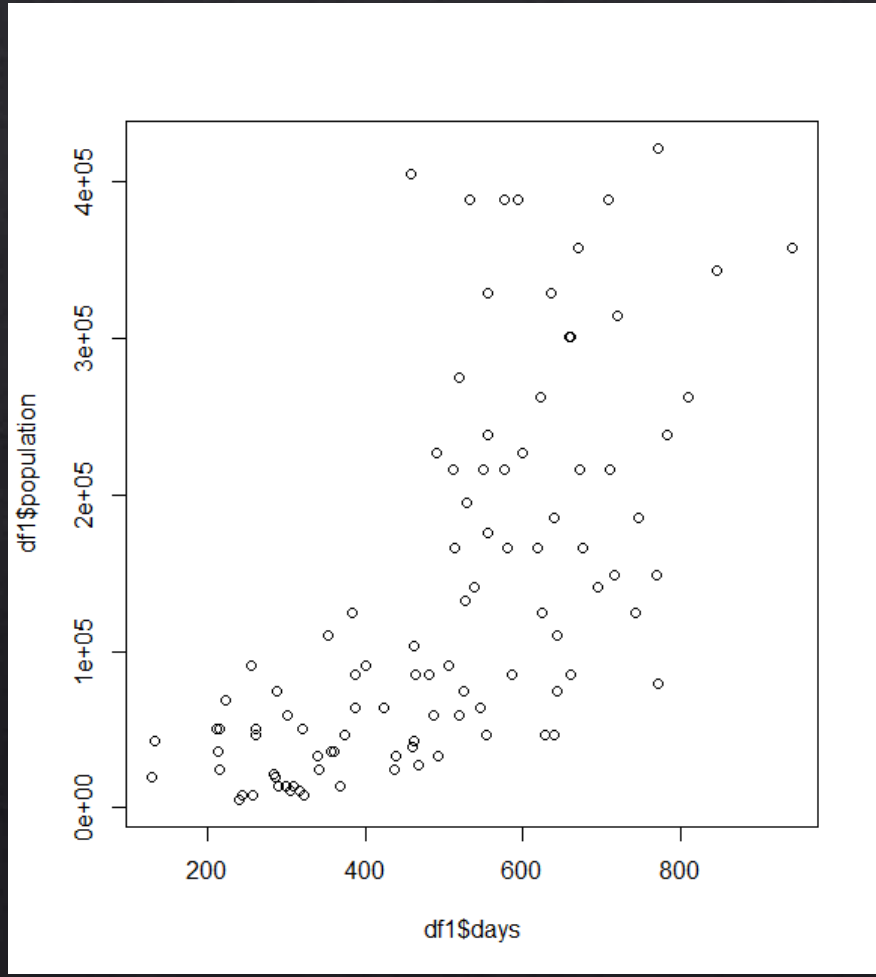
Part 3: GLM extensions

Statistical Models: Bells and Whistles

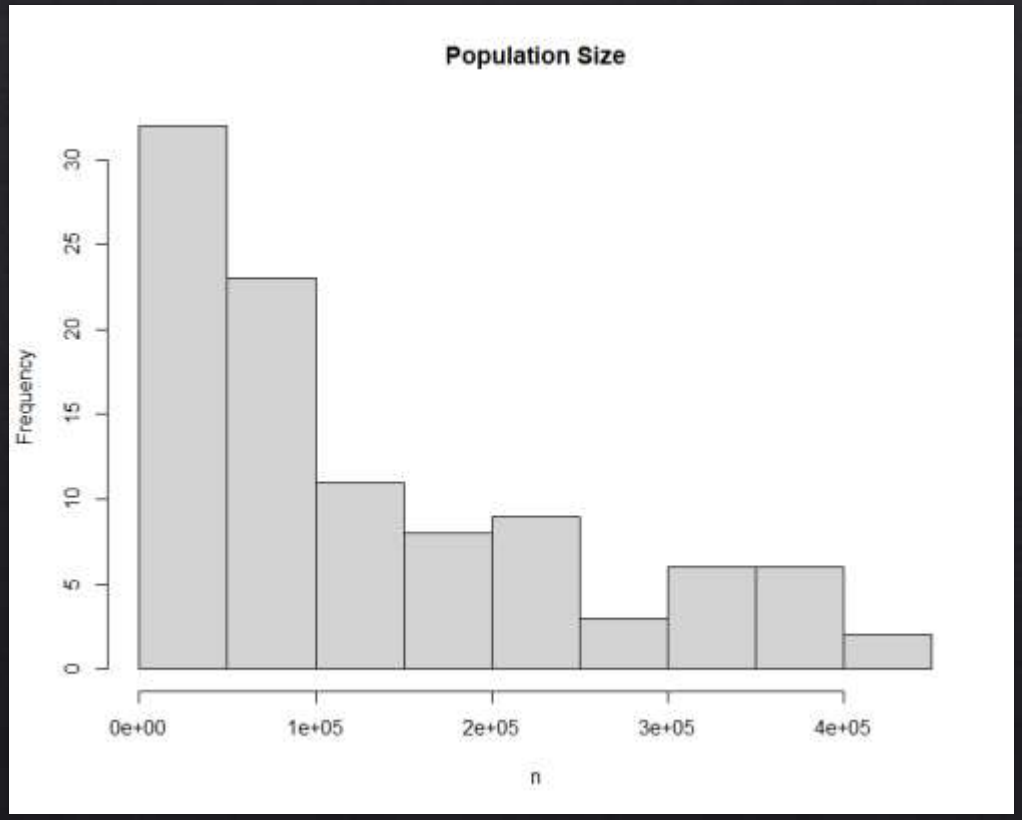
- Sometimes additional features are needed to adequately model your data
- These are *optional extensions*, you may never need to use them but it is useful to know they exist
- They do NOT change any of the principles we have already spoken about.
- I often build a basic model, find the right distribution, test it, THEN add additional complications
- Complexity does not necessarily mean the model is good! If in doubt, keep it simple!

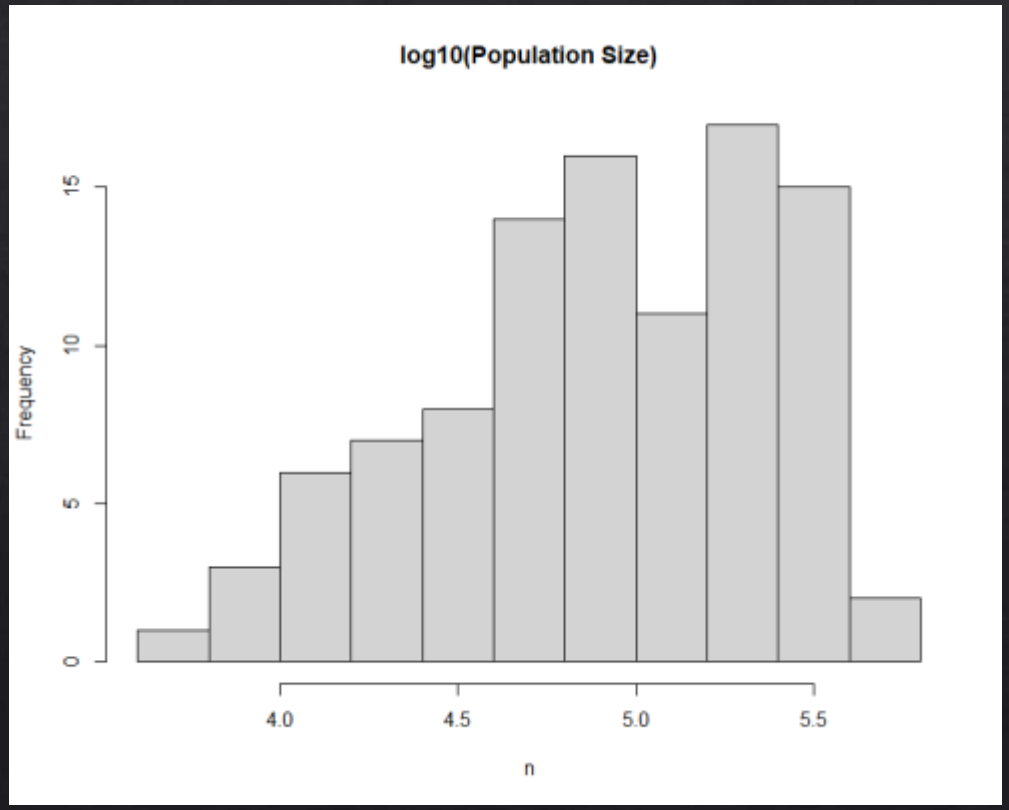
Extension 1: Transformations

- Non-linear data can sometimes be transformed to fit a normal distribution
- If measurements show extreme skew, or show extreme heteroskedasticity (variance increases or decreases over x), then sometimes transformation can be the easiest solution
- Examples include:
  - Logarithms
  - Square root
  - Inverse (1/x)



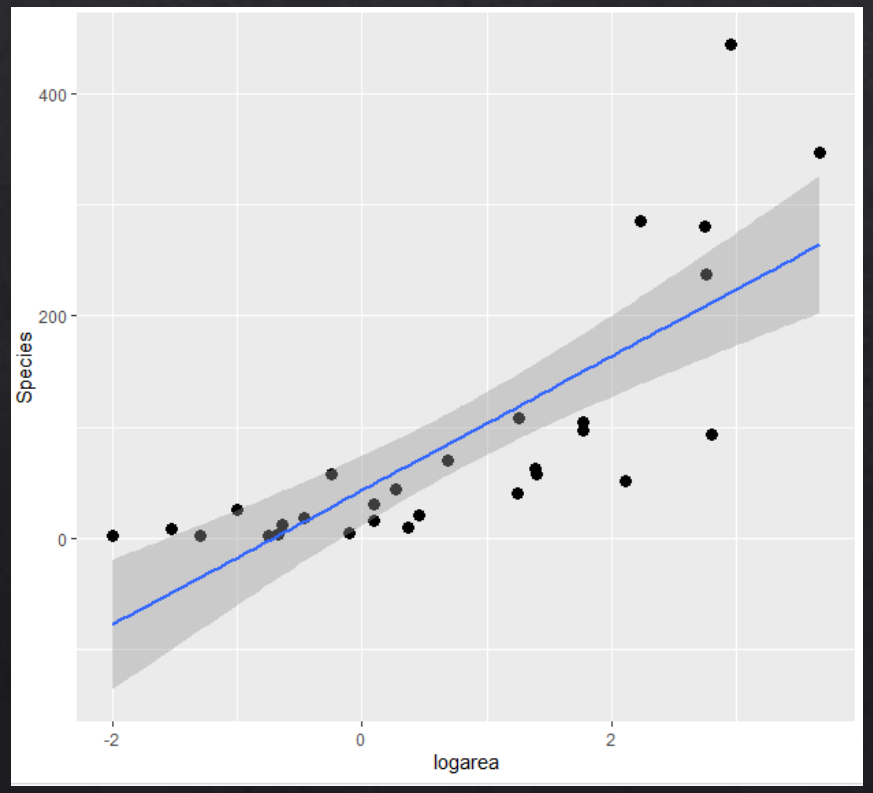
Extension 1: Transformations





Extension 2: Quadratics

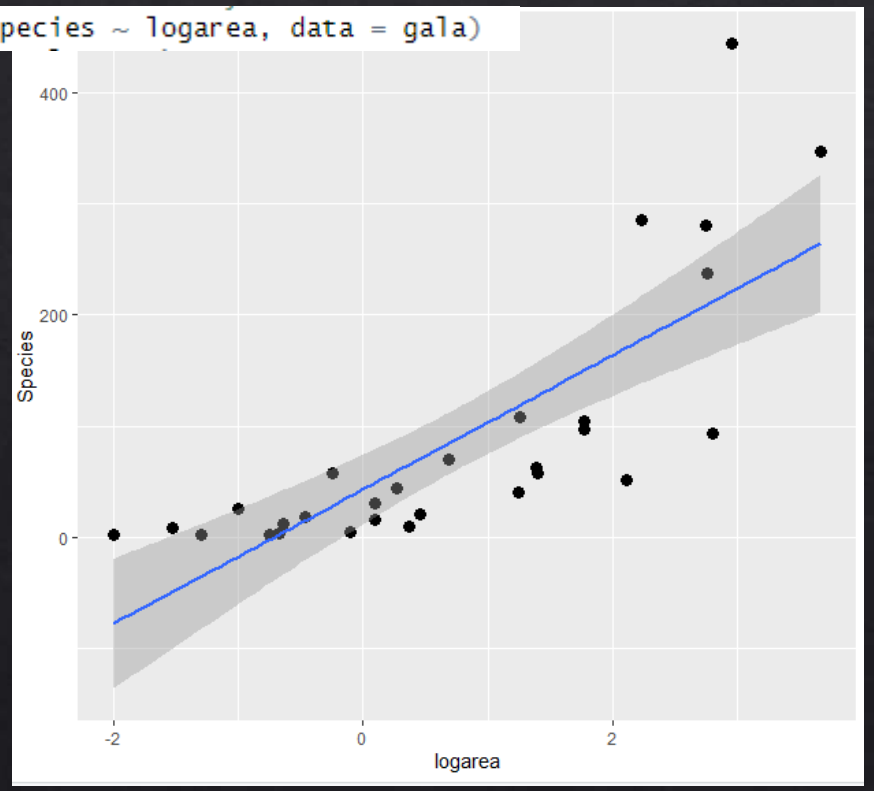
- Sometimes a relationship is not linear, it is curved. What can we do in this case?



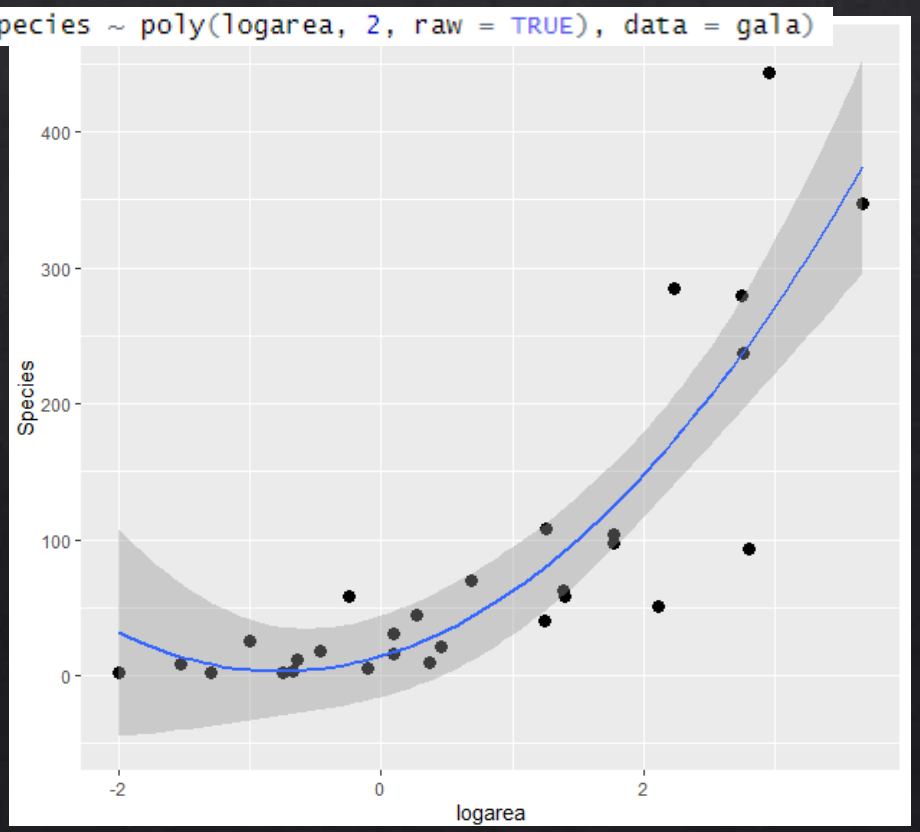
Extension 2: Quadratics

- Sometimes a relationship is not linear, it is curved. In such cases we add polynomials (such as quadratics)

```
lm.raw <- lm(Species ~ logarea, data = gala)
```

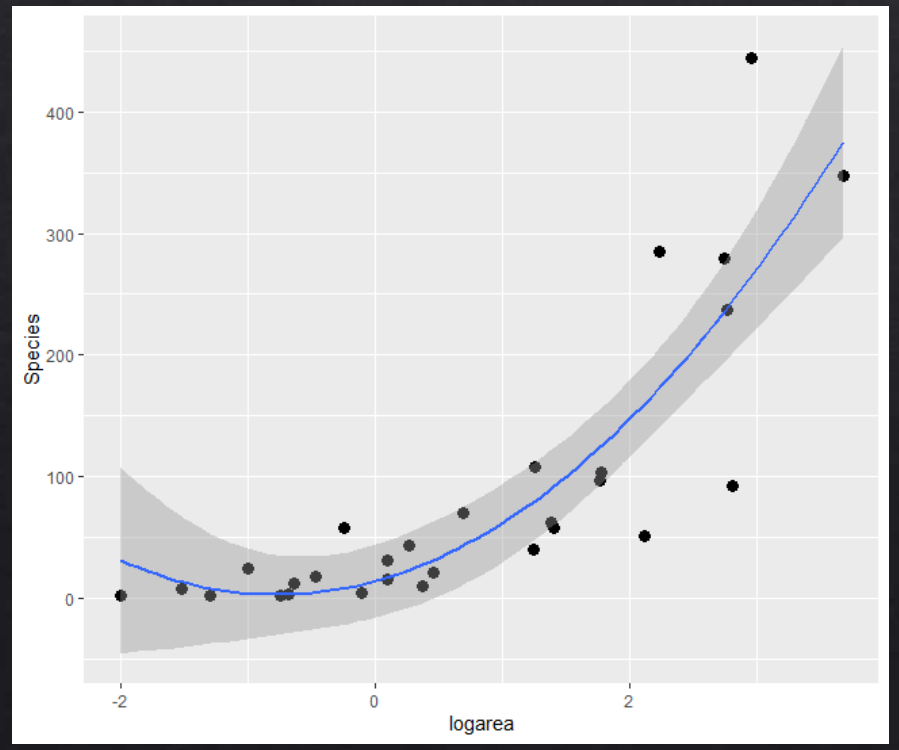



```
lm.poly1.raw <- lm(Species ~ poly(logarea, 2, raw = TRUE), data = gala)
```



Extension 2: Quadratics

- Example of quadratic output:

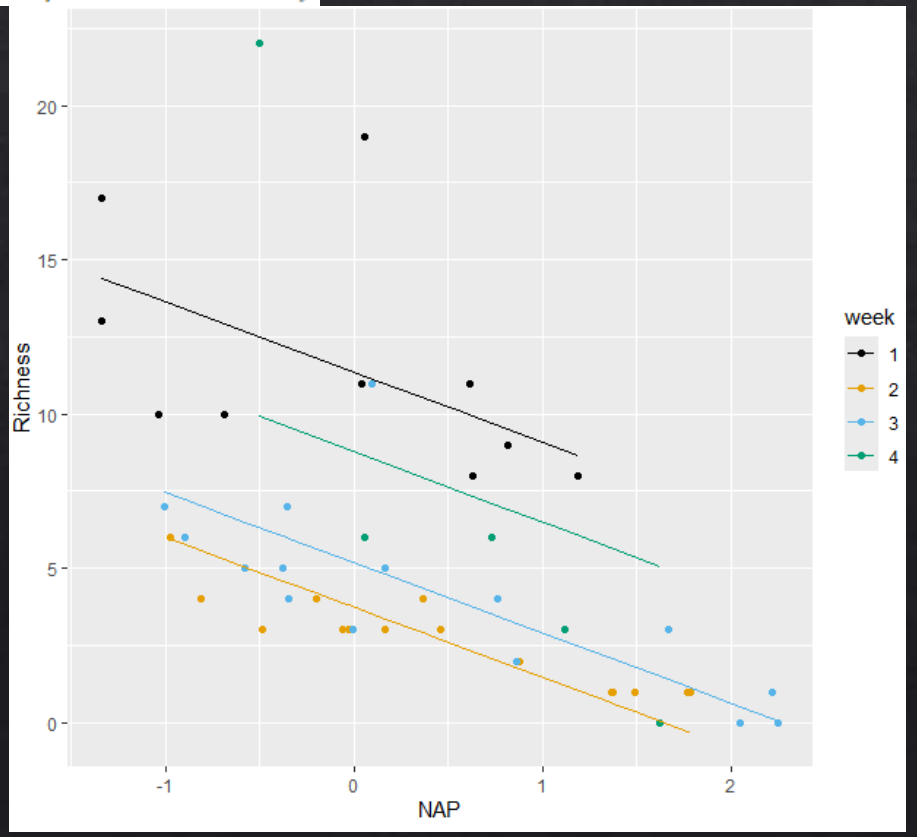



```
Call:
lm(formula = Species ~ poly(logarea, 2, raw = TRUE), data = gala)

Residuals:
     Min       1Q   Median       3Q      Max
-151.009  -27.361   -1.033   20.825  178.805

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                       14.153     14.561   0.972 0.340010
poly(logarea, 2, raw = TRUE)1     29.065     11.194   2.596 0.015293 *
poly(logarea, 2, raw = TRUE)2     18.896      5.008   3.773 0.000842 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.88 on 26 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7338
F-statistic:  39.6 on 2 and 26 DF,  p-value: 1.285e-08
```

Extension 3: Interactions

- When a model has a continuous and categorical predictor, we estimate the slope and the intercept separately. I.e. the slope is consistent across all categories, but the intercepts vary

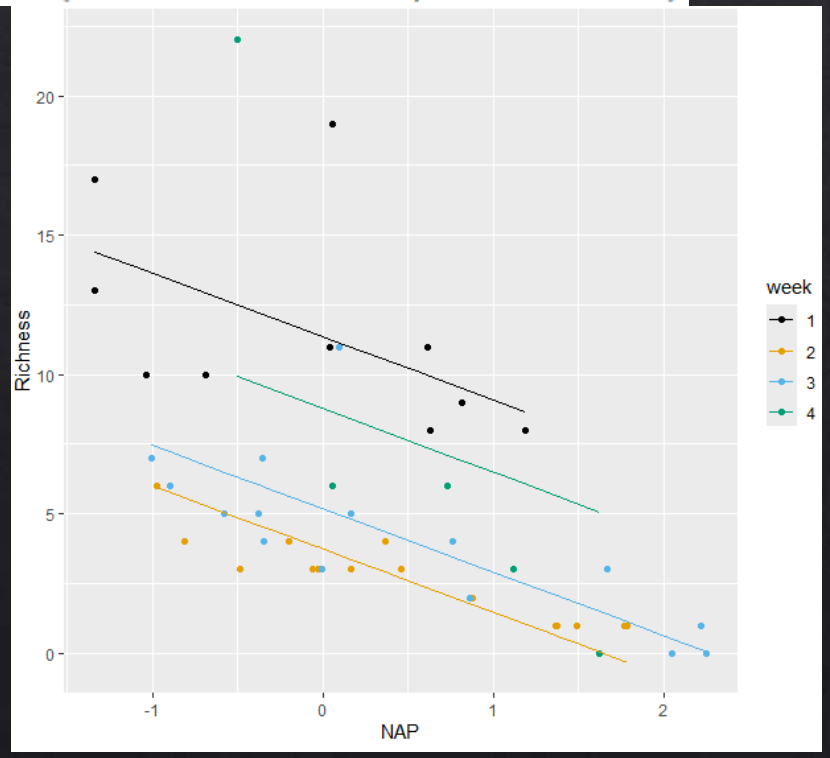
```
lm1 <- lm(Richness ~ NAP + week, data = RIKZdat)
```



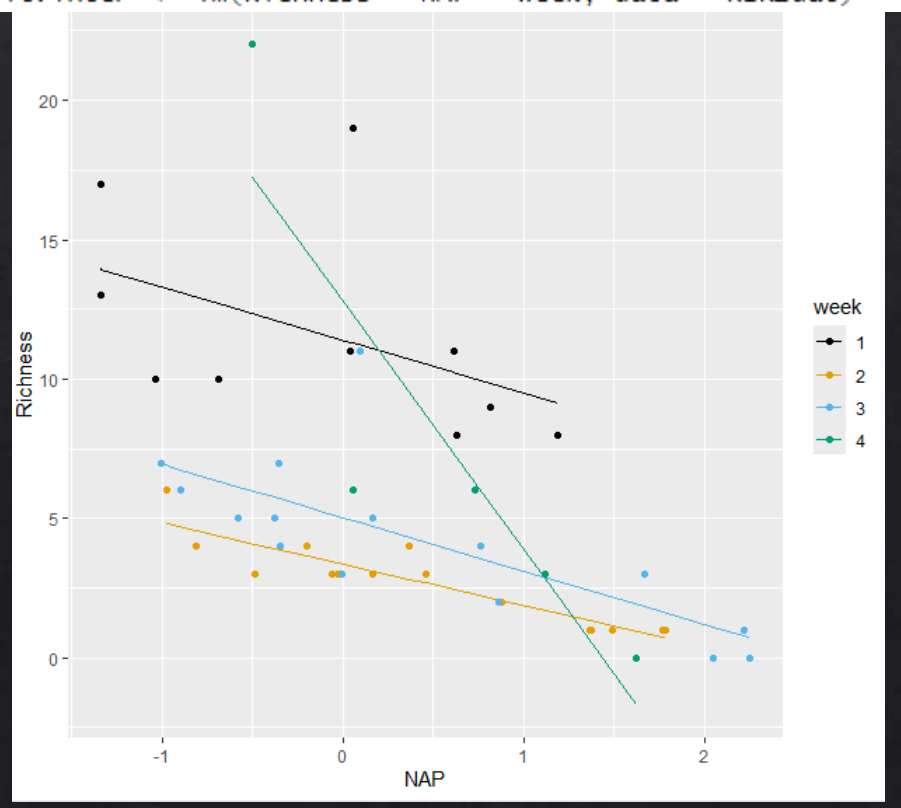
Extension 3: Interactions

- Sometimes the effect of one variable *is dependent* on the value of another
- In this case we add an interaction term to account for this

```
lm1 <- lm(Richness ~ NAP + week, data = RIKZdat)
```

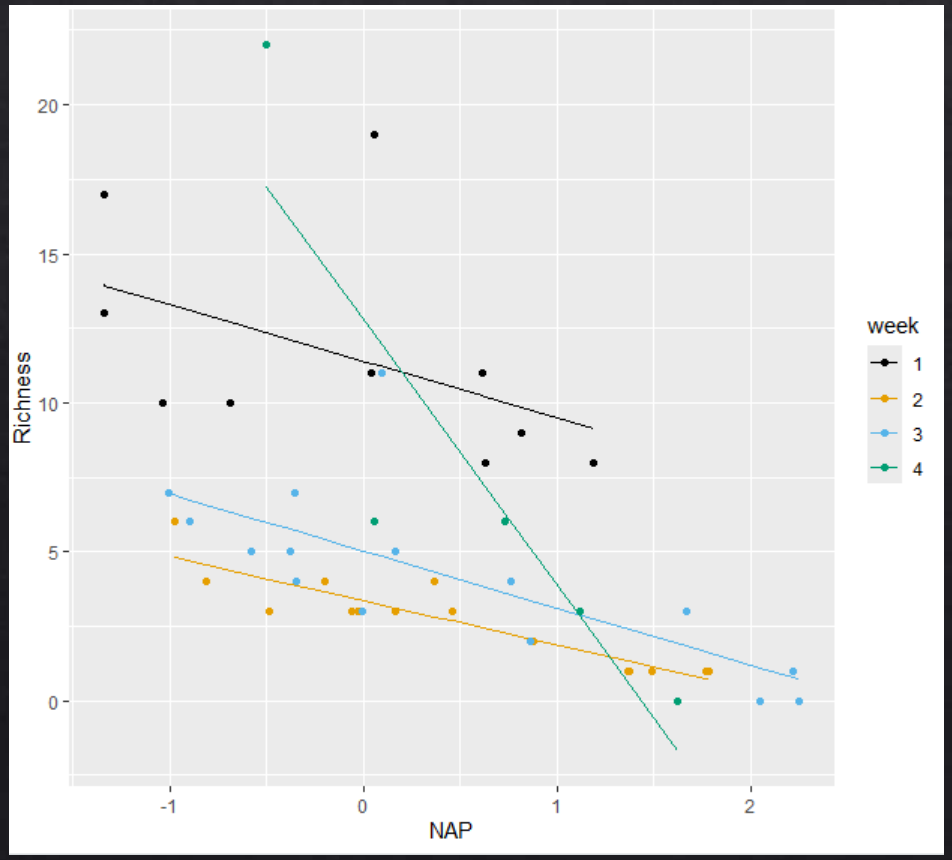


```
lmfit.inter <- lm(Richness ~ NAP * week, data = RIKZdat)
```



Extension 3: Interactions

- Example of interaction output:




```
Residuals:
    Min      1Q  Median      3Q     Max
-6.3022 -0.9442 -0.2946  0.3383  7.7103

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.40561    0.77730  14.673  < 2e-16 ***
NAP         -1.90016    0.87000  -2.184 0.035369 *
week2       -8.04029    1.05519  -7.620 4.30e-09 ***
week3       -6.37154    1.03168  -6.176 3.63e-07 ***
week4        1.37721    1.60036   0.861 0.395020
NAP:week2    0.42558    1.12008   0.380 0.706152
NAP:week3   -0.01344    1.04246  -0.013 0.989782
NAP:week4   -7.00002    1.68721  -4.149 0.000188 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.442 on 37 degrees of freedom
Multiple R-squared:  0.7997,    Adjusted R-squared:  0.7618
F-statistic: 21.11 on 7 and 37 DF,  p-value: 3.935e-11
```

Extension 4: Random Effects

- In ecology, sometimes our observations are not independent.
- For example:
  - You want to test whether canopy cover reduces salamander larval density. You survey 10 ponds, 5 quadrats in each. The problem is each pond likely has unmeasured features (e.g., water chemistry, predators), and you have repeatedly measured in each pond. This is a form of pseudo-replication, and introduces various issues of bias and inaccuracy.
- The solution to uneven or unstructured measurements are *random effects models (*aka GLMM)

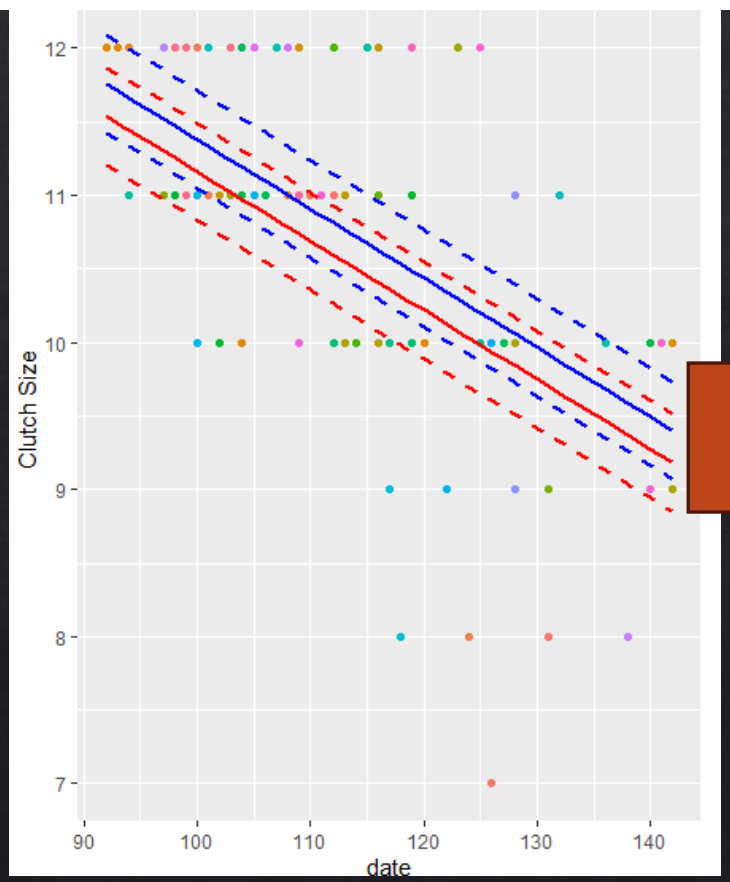
Extension 4: Random Effects

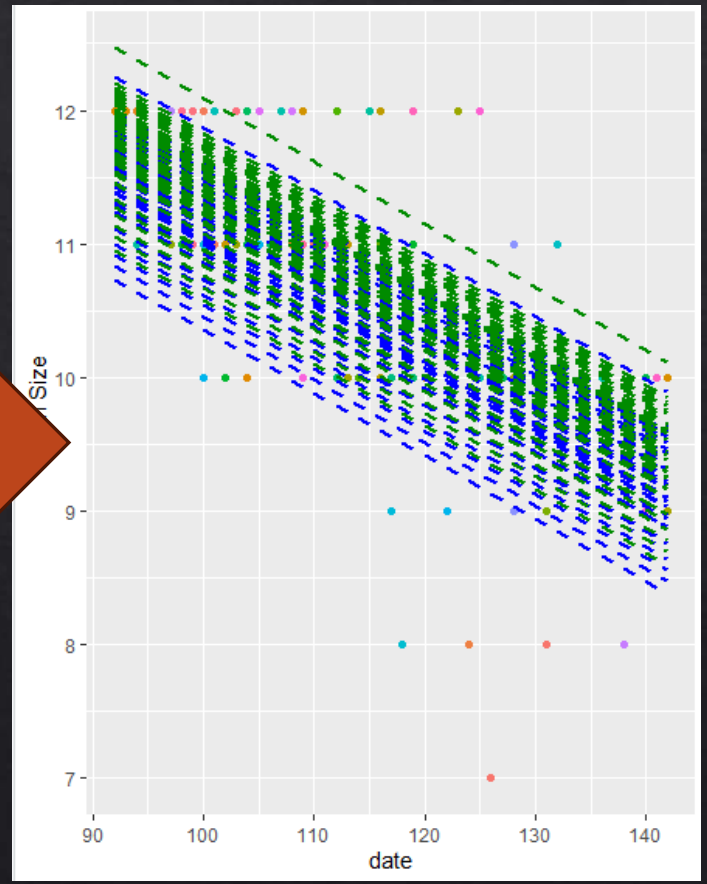
- When should you use a mixed effect models (i.e. a model with fixed and random effects)?
- You don’t have to! Often there are other solutions
- But they are useful when:
  - there are repeated measures on the same sample units (same site, same animal, same lake)
  - the data are naturally clustered or hierarchical in nature (e.g., you are collecting data on individuals that live in packs within multiple populations).
  - you are interested in quantifying variability of a response across different levels of replication (e.g., among and within lakes, among and within sites, etc)
- Mixed effect models can be complex, if they are a total headache consider finding other solutions (e.g. try a Bayesian approach)

Extension 4: Random Effects

- On the left we see average predictions across two categories. This is good, but some of these observations are repeated from the same sites. This is unstructured and we can't include it as a fixed effect variable. To account for this noise, we will add site as a random effect (right)
- We see a lot of signal in the data is due to the random effect of site. Happily, even after controlling for it, we still see a significant slope.

```
clutch.fix <- lm(CLUTCH ~ date + Ideply, data=clutch)
```






```
clutch.ri <- lmer(CLUTCH ~ date + Ideply + (1 | strtno), data=clutch, REML=T)
```

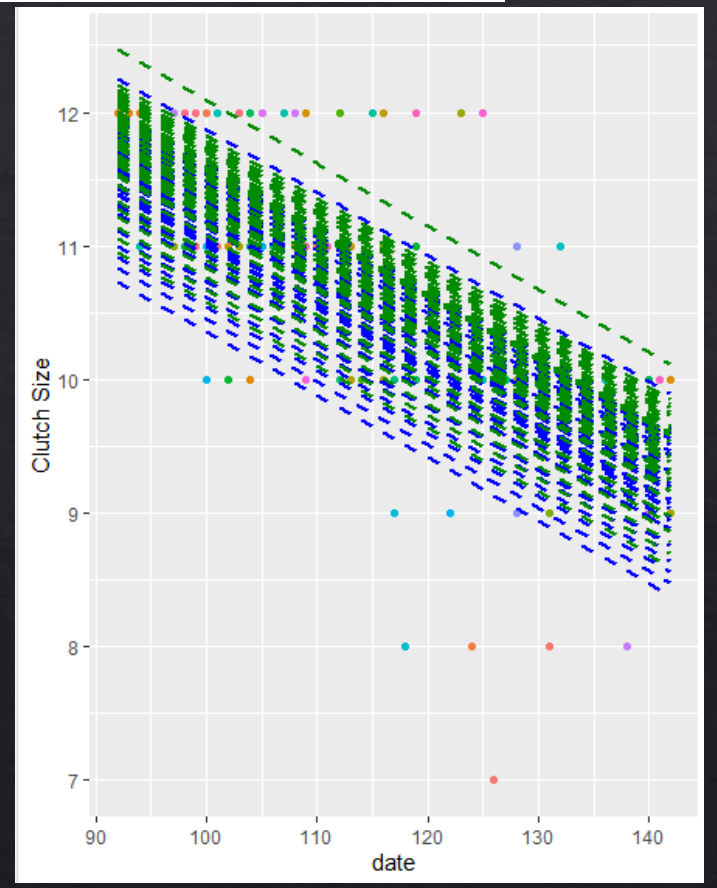
Extension 4: Random Effects

- On the left we see average predictions across two categories. This is good, but some of these observations are repeated from the same sites. This is unstructured and we can't include it as a fixed effect variable. To account for this noise, we will add site as a random effect (right)
- We see a lot of signal in the data is due to the random effect of site. Happily, even after controlling for it, we still see a significant slope.

```
clutch.ri <- lmer(CLUTCH ~ date + Ideply + (1 | strtno), data=clutch, REML=T)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: CLUTCH ~ date + Ideply + (1 | strtno)
   Data: clutch

REML criterion at convergence: 283.2

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.81168 -0.44368 -0.03218  0.55879  2.14321

Random effects:
 Groups   Name        Variance Std.Dev.
 strtno   (Intercept) 0.2532   0.5032
 Residual             0.5755   0.7586
Number of obs: 106, groups:  strtno, 52

Fixed effects:
            Estimate Std. Error t value
(Intercept) 16.093267   0.785919  20.477
date        -0.047134   0.007023  -6.712
IdeplyTRUE  -0.219184   0.217833  -1.006

Correlation of Fixed Effects:
           (Intr) date
date       -0.978
IdeplyTRUE  0.053 -0.214
```

```
                    2.5 %      97.5 %
.sig01                 NA          NA
.sigma                 NA          NA
(Intercept) 14.55289488 17.63363881
date        -0.06089867 -0.03337013
IdeplyTRUE  -0.64612823  0.20776081
```

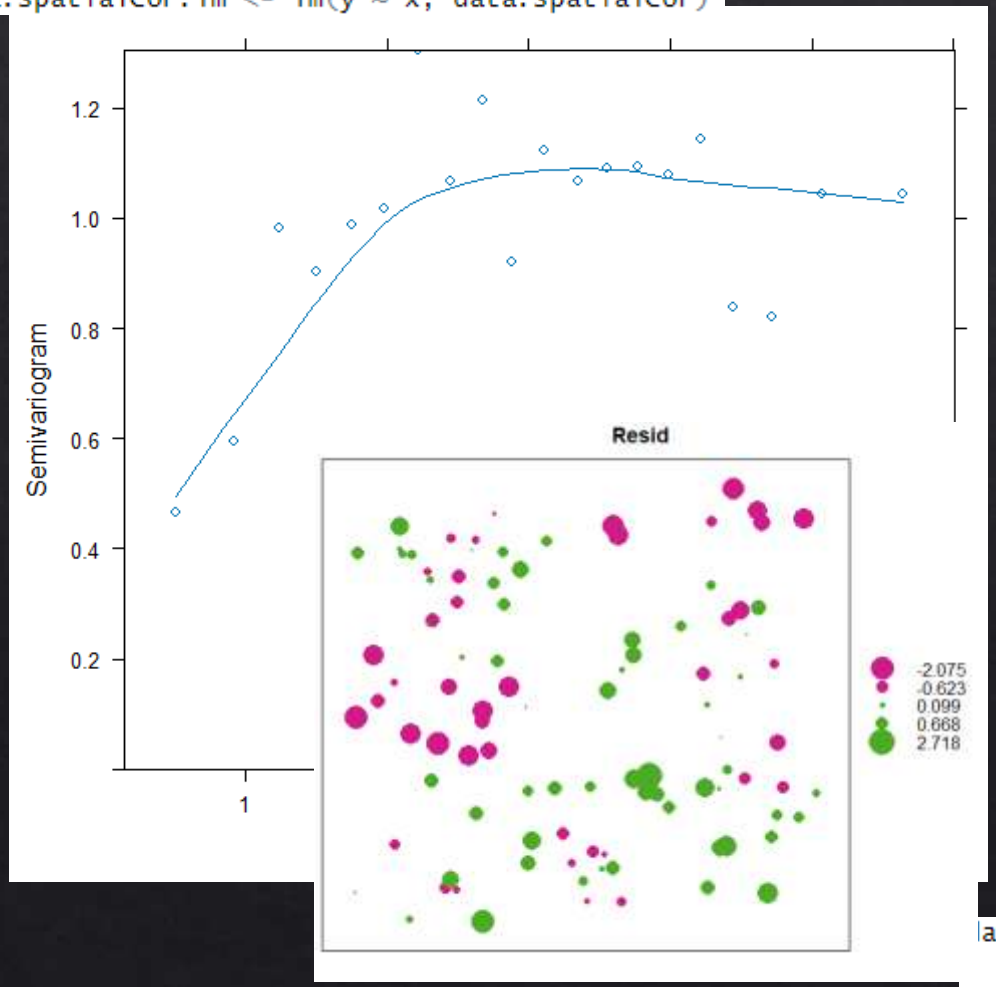


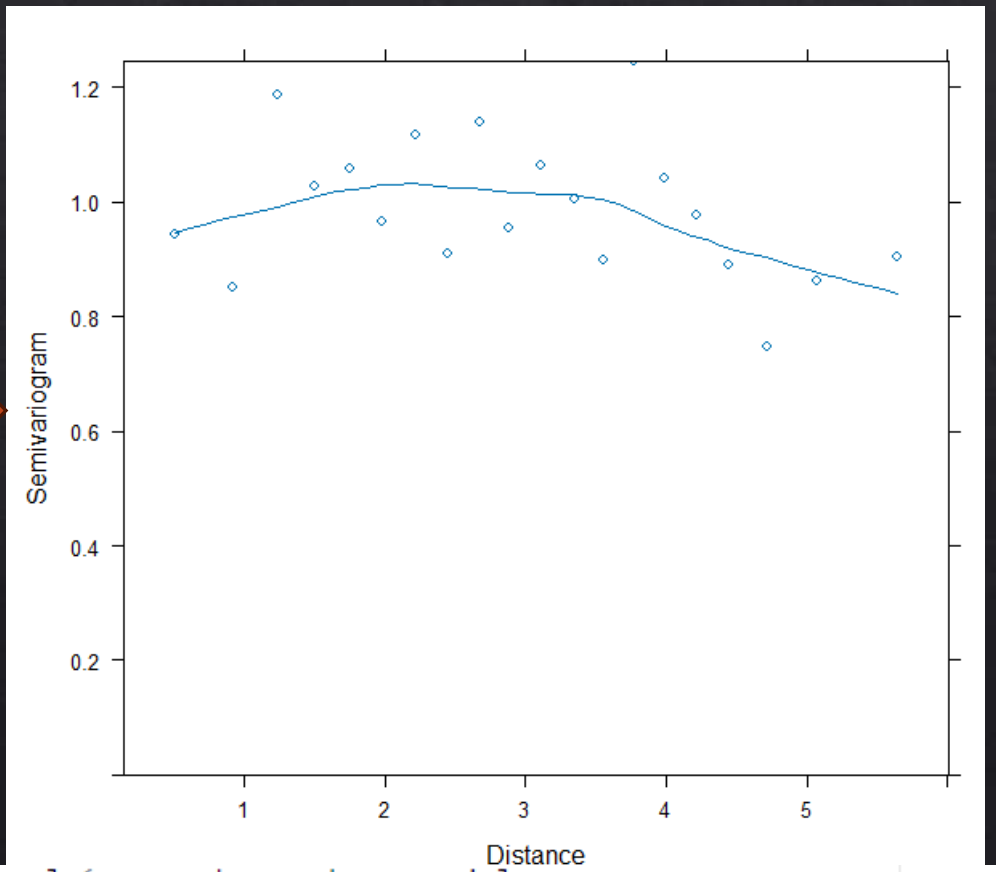
Extension 5: Spatial Autocorrelation

- When you have spatial data, linear models assume that observations are independent
- With autocorrelation, sites that are close together are more likely to be similar
  - This breaks a key assumption of our model
- We can control for this by adding a spatial term to our model
  - Conceptually similar to random effects, we are controlling for non-independence in observations
- This is complex and should only be done if we think we need it
- Luckily we can test for spatial autocorrelation by looking at our model residuals with a variogram

Extension 5: Spatial Autocorrelation

```
data.spatialCor.lm <- lm(y ~ x, data.spatialCor)
```





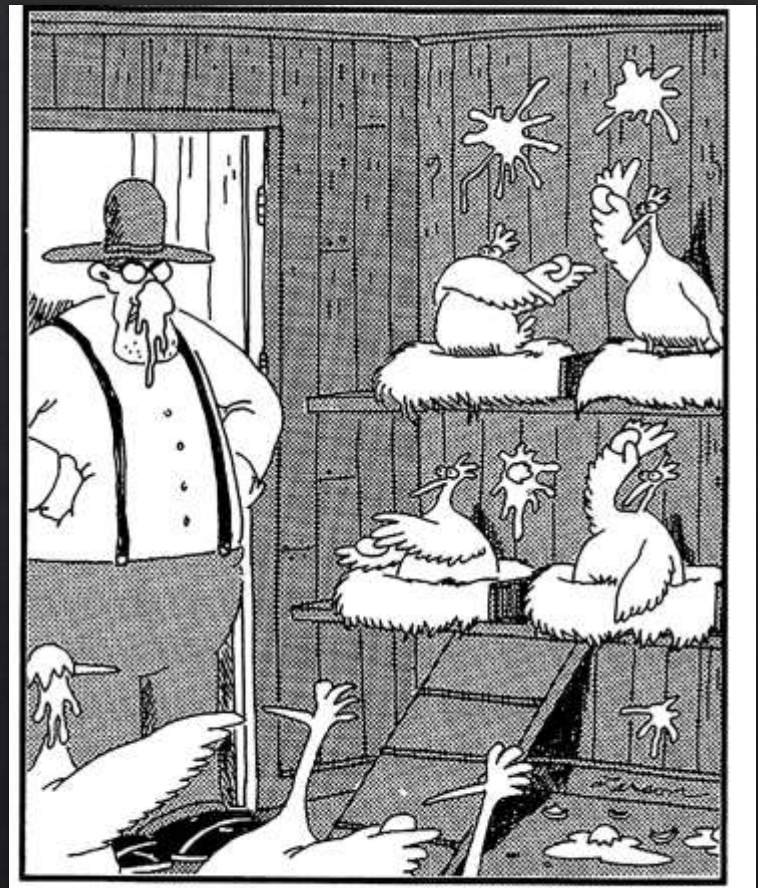

```
data.spatialCor.glsExp <- gls(y ~ x, data = data.spatialCor,
                              correlation = corExp(form = ~LAT + LONG, nugget = TRUE),
                              method = "REML")
```

And so on…

- Implementing each one of these is simply learning one more “add-on” for your model. They do not substantially change how they work.

Questions?

PART 3: Moving past GLMs



As quickly as it had started, the egg fight was over.

Statistical Models

- Question: what do we do if no model fits properly?

Statistical Models

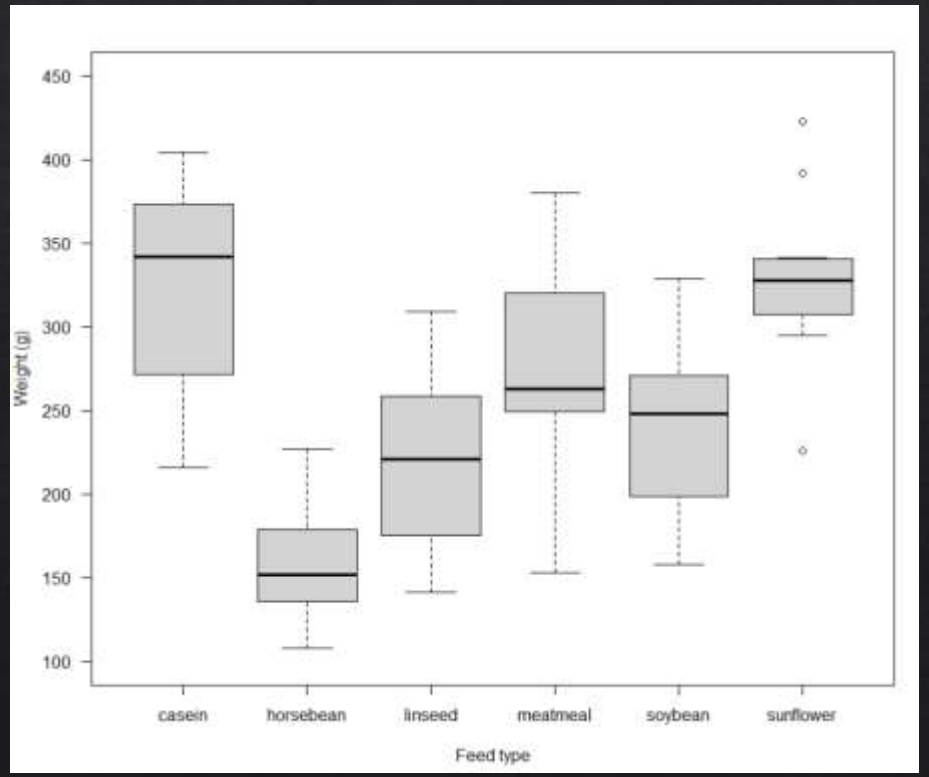
- Question: what do we do if no model fits properly?
  - We have a lot of options!
  - Look again. R has a lot of distributions implemented, one of them might work
  - Try a data transformation? Could a square root, a log, a logit transformation help (note: use this as a last resort, it's better to use native data)
  - Get more complicated. Some issues can be solved by adding more parameters and corrections (e.g. quadratic parameters), or by bootstrapping.
  - **Use a non- or semi-parametric approach!**

Non-parametric model

- A very useful alternative when no other approach works.
  - Requires no distribution, makes no assumptions about data
  - BUT less powerful, cannot model interactions or other complex parameters.
- Every parametric test has a non-parametric equivalent. But they all basically work the same way. For some reason they all have complicated names.
  - Paired t-test → Signed-rank test
  - 2-sample t-test -> Mann-Whitney test
  - ANOVA → Kruskal-Wallis test
  - Linear regression → Spearman rank correlation (in some cases)

Non-parametric models

- Example: for this dataset we could use a linear model (ANOVA), as we have a continuous response and categorical predictor. But if it's not normally distributed, we can try the equivalent non-parametric test.

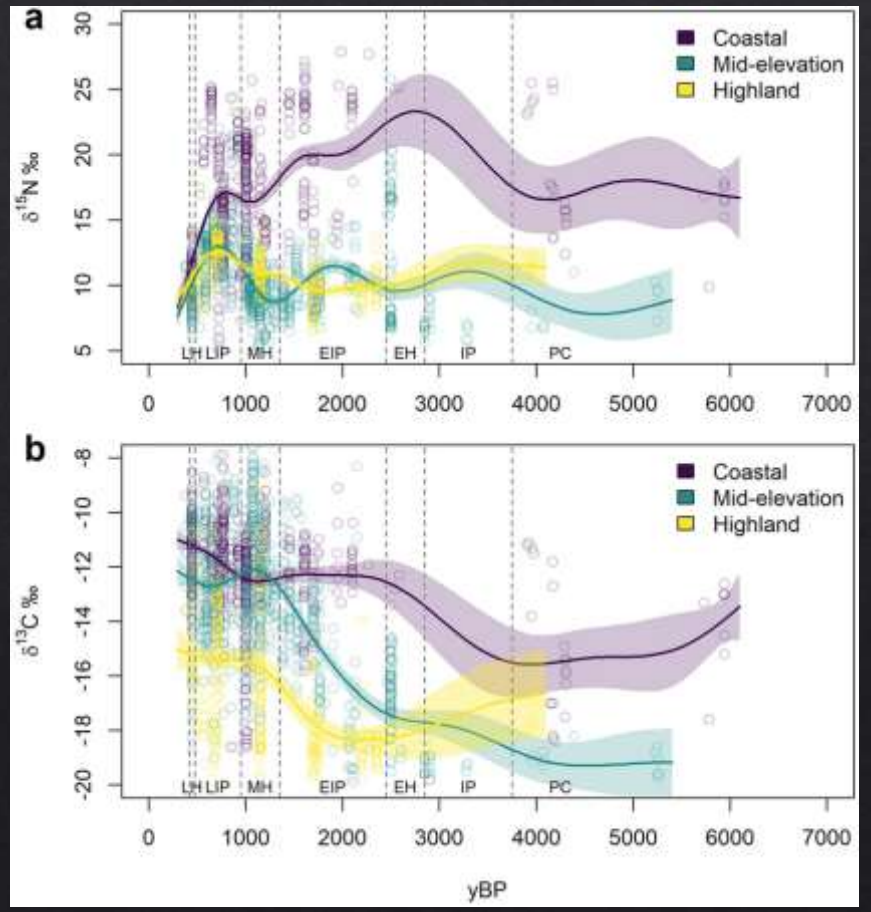



```
        Kruskal-Wallis rank sum test

data:  weight by feed
Kruskal-Wallis chi-squared = 37.343, df = 5, p-value = 5.113e-07
```

```
        Pairwise comparisons using Wilcoxon rank sum exact test

ata:  chickwts$weight and chickwts$feed

          casein  horsebean linseed meatmeal soybean
orsebean 0.00016 -         -       -        -
inseed   0.00305 0.01191   -       -        -
eatmeal  0.11355 0.00096   0.05451 -        -
oybean   0.01110 0.00227   0.27306 0.28035  -
unflower 1.00000 9.3e-05   0.00025 0.09384  0.00334

 value adjustment method: BH
```

Semi-parametric models:

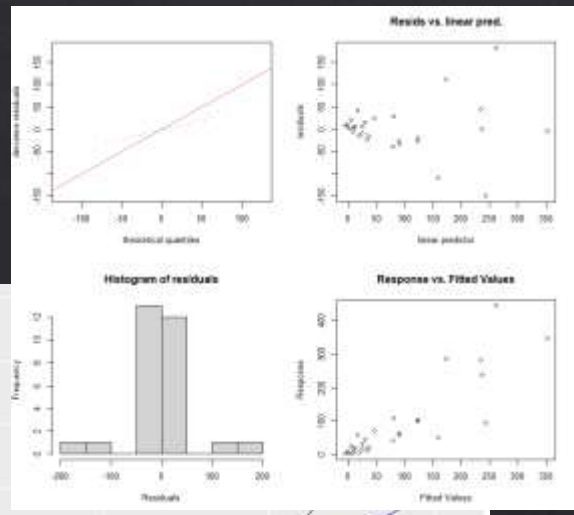
- Combines properties of parametric and non-parametric models
- Most commonly used are Generalised Additive Models
- Very flexible, very useful for non-linear relationships
  - Especially for those that don't fit into a standard distribution
- BUT predictions are difficult and very prone to over-fitting
- Interpretation is also difficult, as few concrete parameters are returned
- Helpfully if data fits a parametric distribution, a GAM will just resolve to that.

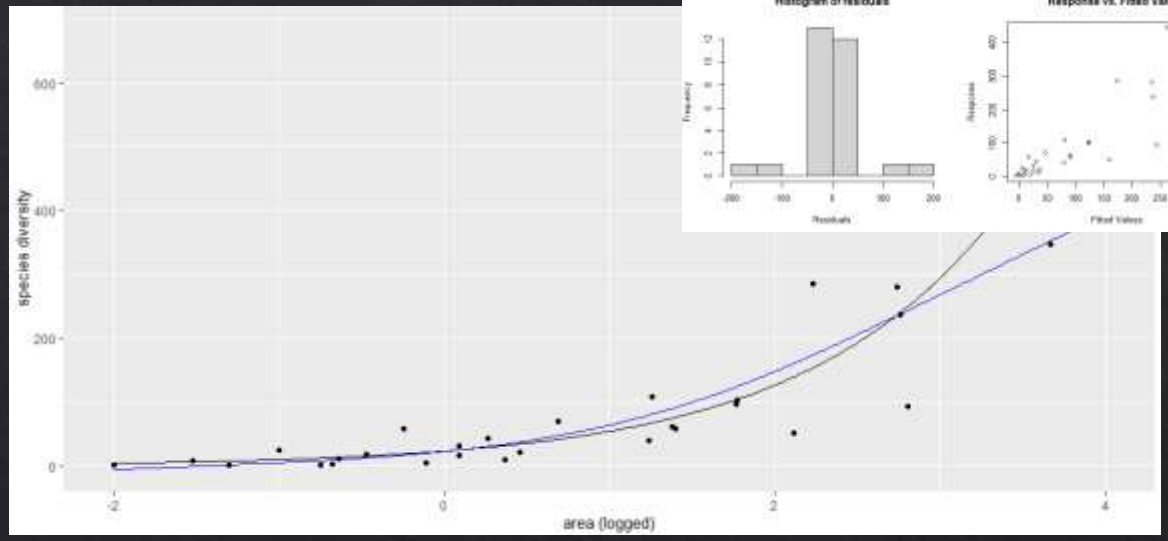


Source: https://www.nature.com/articles/s41598-022-05774-y

Generalised Additive Models

- Based on “smoothing splines”, with two parameters
  - f’’(x) is the second derivative of f(x) (the main expression) and describes how smooth the curve is (high values are indicative of a lot of ‘wigglyness’),
  - λ is a tuning parameter that determines the penalty for `smoothness’. As λ→∞, the line will approach being completely straight
- Still produce residuals, which should be checked in the same way as for other models
- P-values should be interpreted with caution! They tend to be under-estimated






```
Formula:
Species ~ s(logarea)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    87.34      11.11   7.864 2.76e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
             edf Ref.df     F p-value
s(logarea) 2.465  3.107 24.86  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Summary: Parametric versus Non-parametric

- Parametric models require you to make a choice on how the system works. Good practice as it makes you really think about the data.
- Parametric models have more constraints, are more complicated, and more prone to error
- BUT when they work, they are more powerful, and much more useful for building predictions

- Non/semi-parametric models nearly always work, they are very flexible
- They have many fewer assumptions, and it is difficult to generate errors with them
- BUT they are less powerful, prone to overfitting, and can be very difficult to interpret or predict with.

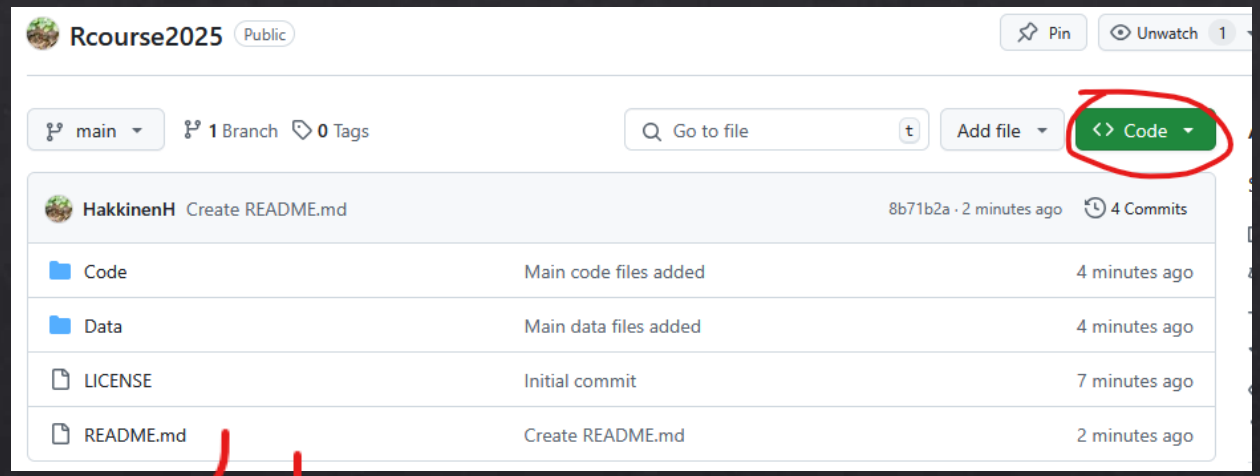
- *A dark secret method:* run both and make sure they give you a similar answer. If they don't then something is likely wrong.

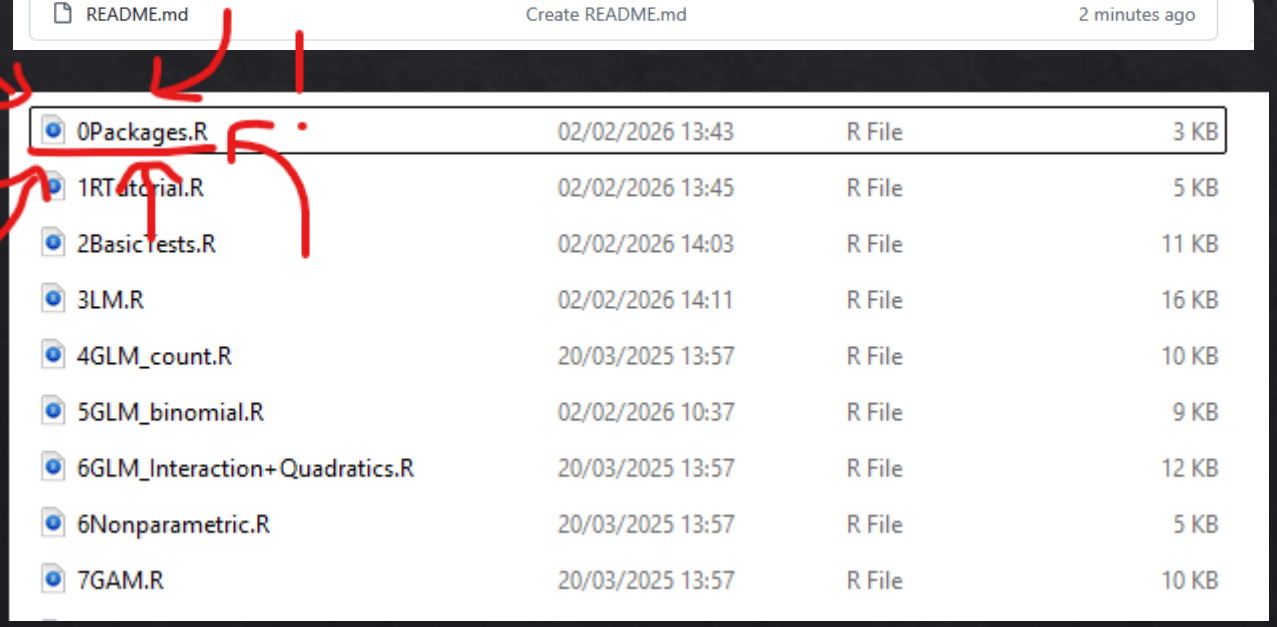
Questions?

The Practical: Part 2

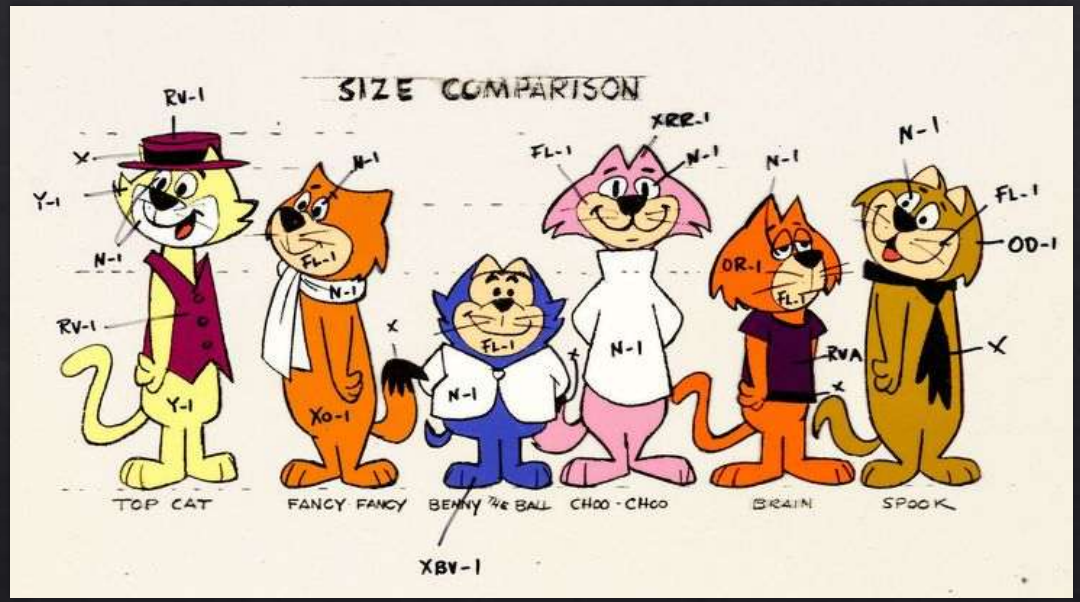
https://github.com/HakkinenH/RStatsCourse2026

- Code -> download ZIP -> Unzip and put somewhere useful
- Run 0Packages.R first ((if you haven't already) to install packages
- Work through files 6 to 11, in whatever order you like!





PART 5: Model fit and model comparisons



Model fit and model comparisons

- It is possible you will end up with some competing models at the end of your analysis.
  - E.g. some might have interactions, some might not. Some might have quadratics, some might not.
- How do you decide which one is “best” is they all fit ok (i.e. residuals look fine)?
- This is where model fit and model comparison come in

Option 1: put everything in the model

- Include every potential predictor variable, every interaction, every quadratic function you can think of in a model
- ISSUES:
  - Interpretation. With so many variables, how can you make useful predictions? What do all those interactions mean?
  - Overfitting. Models need data to fit correctly, they need flexibility to select the best output (referred to as degrees of freedom). Having too many parameters negates this process, producing a poor model
  - As a general rule of thumb, you need 10-20 data points for every “event” per variable.

Option 2: Comparing models

- You can compare models by using likelihood and information tests.
  - ANOVA
  - Likelihood-ratio tests
  - AIC/AICc/BIC etc.

Option 2: Comparing models

- I ran two models, one is a normal linear model, one is a normal linear model but with a quadratic term
- They use exactly the same data (otherwise I couldn't compare them).
- Which is better?
  - I normally use AIC. But for GAMs its functional equivalent is GCV. Treat is as the same, it penalises complexity and rewards lower residuals and higher likelihood
- Are the models significantly different?
  - We can compare directly using an anova(). Do they give substantially different results?

Linear model:

```
   GCV.Cp
1504.496
```

GAM:

```
   GCV.Cp
1053.727
```

```
anova(mod_lm, mod_gam1, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: Overall ~ Income
Model 2: Overall ~ s(Income, bs = "cr")
  Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
1    52.000      75336
2    45.259      41479 6.7411    33857 2.778e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Option 2a: Step-wise regression

- This approach can be used to eliminate all “useless” variables from a model, or to build up a model from a basic set of variables
- This is an approach commonly taught in courses up until recently
- THIS APPROACH IS NOT RECOMMENDED
- Why? It is multiple testing, it gives false positives, and it is no longer seen as acceptable by most statisticians

```
Step:  AIC=61.31
mpg ~ wt + qsec + am

       Df Sum of Sq    RSS    AIC
<none>              169.29 61.307
+ hp    1     9.219 160.07 61.515
+ carb  1     8.036 161.25 61.751
+ disp  1     3.276 166.01 62.682
+ cyl   1     1.501 167.78 63.022
+ drat  1     1.400 167.89 63.042
+ gear  1     0.123 169.16 63.284
+ vs    1     0.000 169.29 63.307
- am    1    26.178 195.46 63.908
- qsec  1   109.034 278.32 75.217
- wt    1   183.347 352.63 82.790

Call:
lm(formula = mpg ~ wt + qsec + am, data = mtcars)

Coefficients:
(Intercept)           wt         qsec           am
      9.618       -3.917        1.226        2.936
```

Option 3: Go Complicated: Regularization using penalization

- LASSO and Ridge regression
- Essentially penalises parameters that add nothing to the model and “shrinks” them towards 0.
- Shrinkage can be viewed as a method for reducing model complexity; shrinkage reduces variability in parameter estimates at the expense of introducing some bias
- Useful if you’re building predictions, or have too many correlating parameters, but otherwise quite niche

Option 4: Your Own Judgment

- Be critical. Can you justify all the parameters being in there?
  - This includes interactions
- If you did find something was significant, can you explain why this pattern exists? Is it causation or just correlation?
- If you can't interpret it, it's basically useless.
- If in doubt, favour a simple model
- If in doubt, plot both models. Sometimes a "significant" interaction has no practical effort on predictions or outputs. In which case it is adding little information and can be ignored.
- If you have a few mutually exclusive, but perfectly valid, models THEN you can compare with AIC or a LR-test or similar.

Questions

Quick review

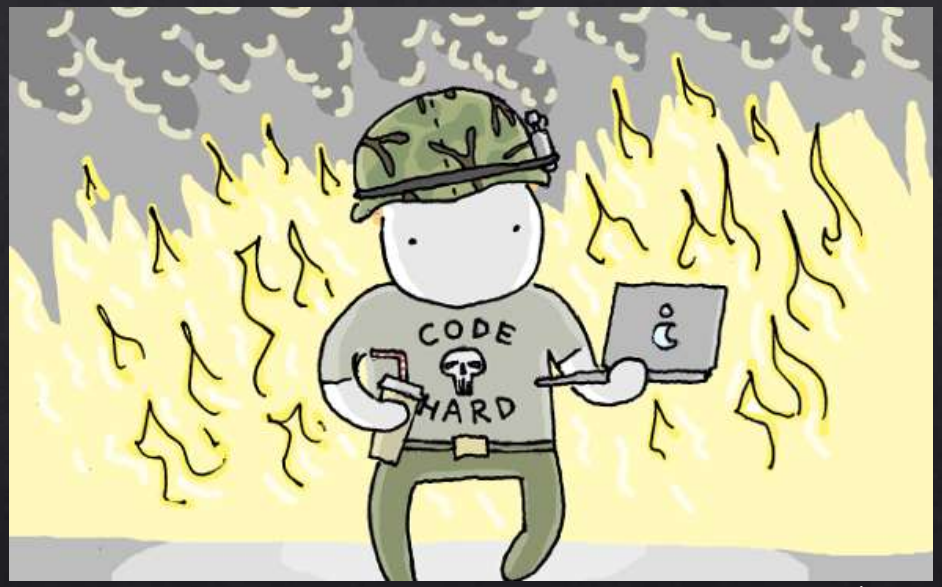
- Go through all the steps of running a simple statistical model
- Creating a statistical toolkit
  - Distributions, residuals and model fit
  - Fixed effects and interactions
  - Mixed effects
  - Model fit and model comparisons
- Work through examples in R.

Quick review

- This covers a lot of classical statistics
- The bad news: there is a wider world out there
- The good news: a lot of the principles we've revised here are the same out there!

Summary

- ALL MODELS ARE WRONG (but some are useful)
  - Corollary: there is no such thing as a perfect model. It cannot be achieved
- We are not modelling reality, we are modelling *data.*
  - If we knew everything about our data and how they work, we wouldn't need statistics in the first place. We are trying to make sense of the world with imperfect data.
- Statistical models are tools, they cannot replace your brain!
- It is your choice on what model to run! R cannot tell you!
- Sometimes there is a "right" model, but usually multiple methods and approaches can be used. Most often they will get the same answer!



toggl.com

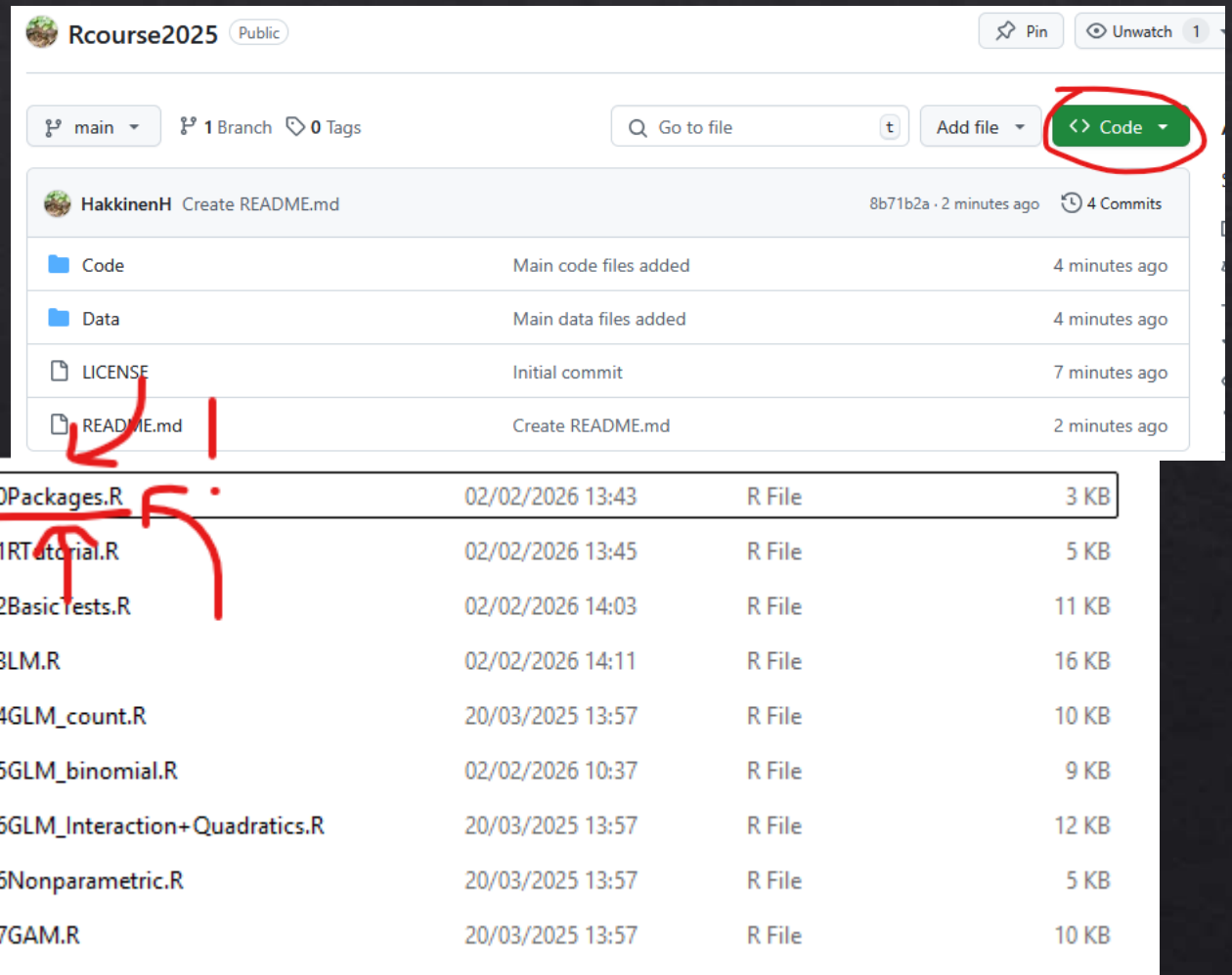
Statistical Models: General Approach

- Steps to making a statistical test:
  1. Think about our study system, identify our response and our predictors and make a formula
  2. Make plots of our data (especially our response variable), and select a distribution.
     i. Check our data fulfils the assumptions of the distribution
  3. Run chosen statistical test
  4. Check the diagnostics and residuals
     i. Consider alternative distributions if necessary
  5. If model fits badly consider how to fix.
     i. Options include: data transformations, alternative distributions, quadratics, random effects,
     ii. If all else fails, use non-parametric tests!
  6. Interpret our output
  7. If we have multiple, valid models, use model comparison to find the best one

The Practical: Part 3

https://github.com/HakkinenH/RStatsCourse2026

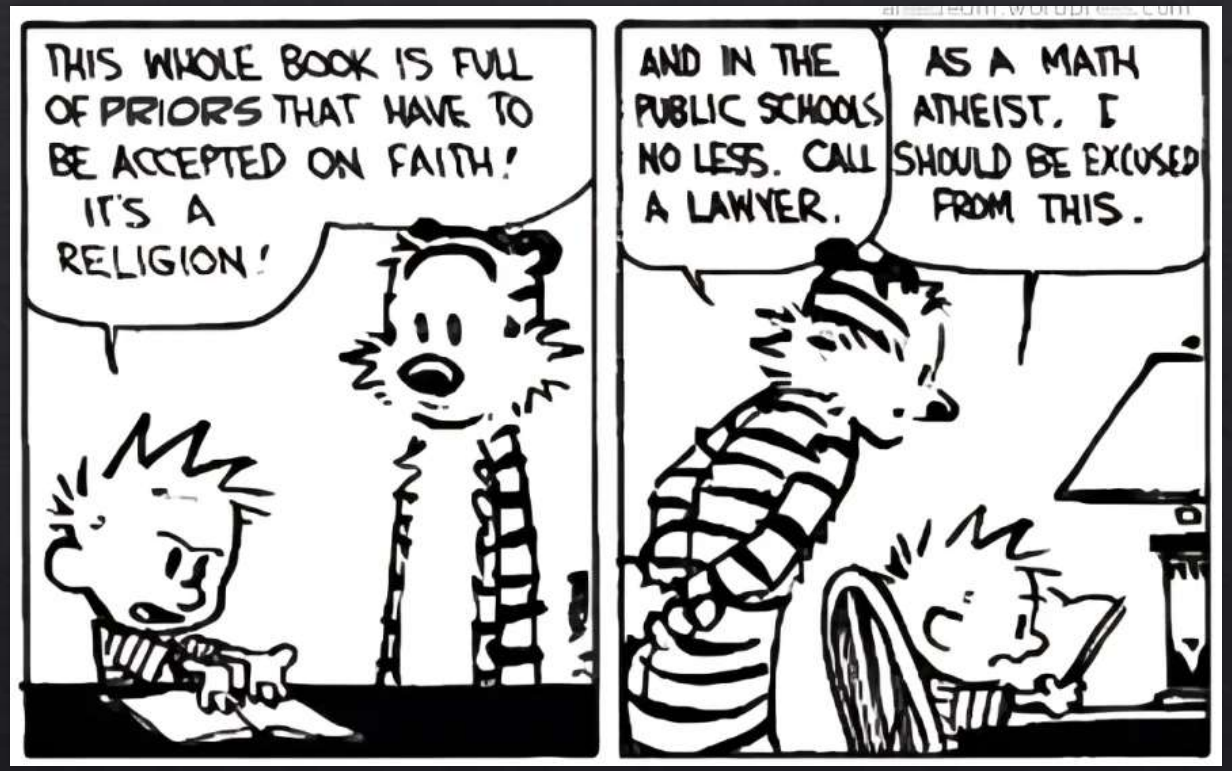
- Code -> download ZIP -> Unzip and put somewhere useful
- Run 0Packages.R first ((if you haven't already) to install packages
- Keep working through whatever examples you want! Ask for help if stuck



Further resources

- I used these heavily for examples and scripts in this course:
- https://statistics4ecologists-v3.netlify.app
- https://bookdown.org/ndphillips/YaRrr/
- https://r.qcbs.ca/workshop06/book-en/reviewing-linear-models.html
- https://saestatsteaching.tech/nonparametric-methods
- https://r.qcbs.ca/workshop07/book-en/mixed-model-protocol.html
- https://cran.r-project.org/web/packages/DHARMa/vignettes/DHARMa.html
- https://www.flutterbys.com.au/stats/tut/tut8.4a.html
- https://noamross.github.io/gams-in-r-course

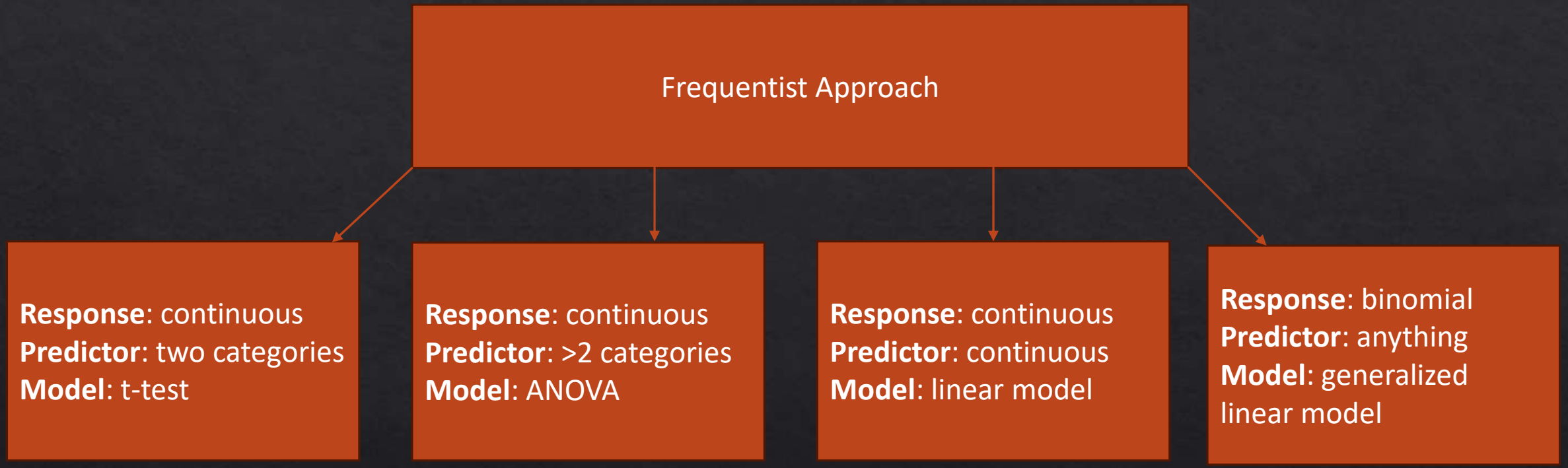
Cut Slides: Other Statistical Frameworks



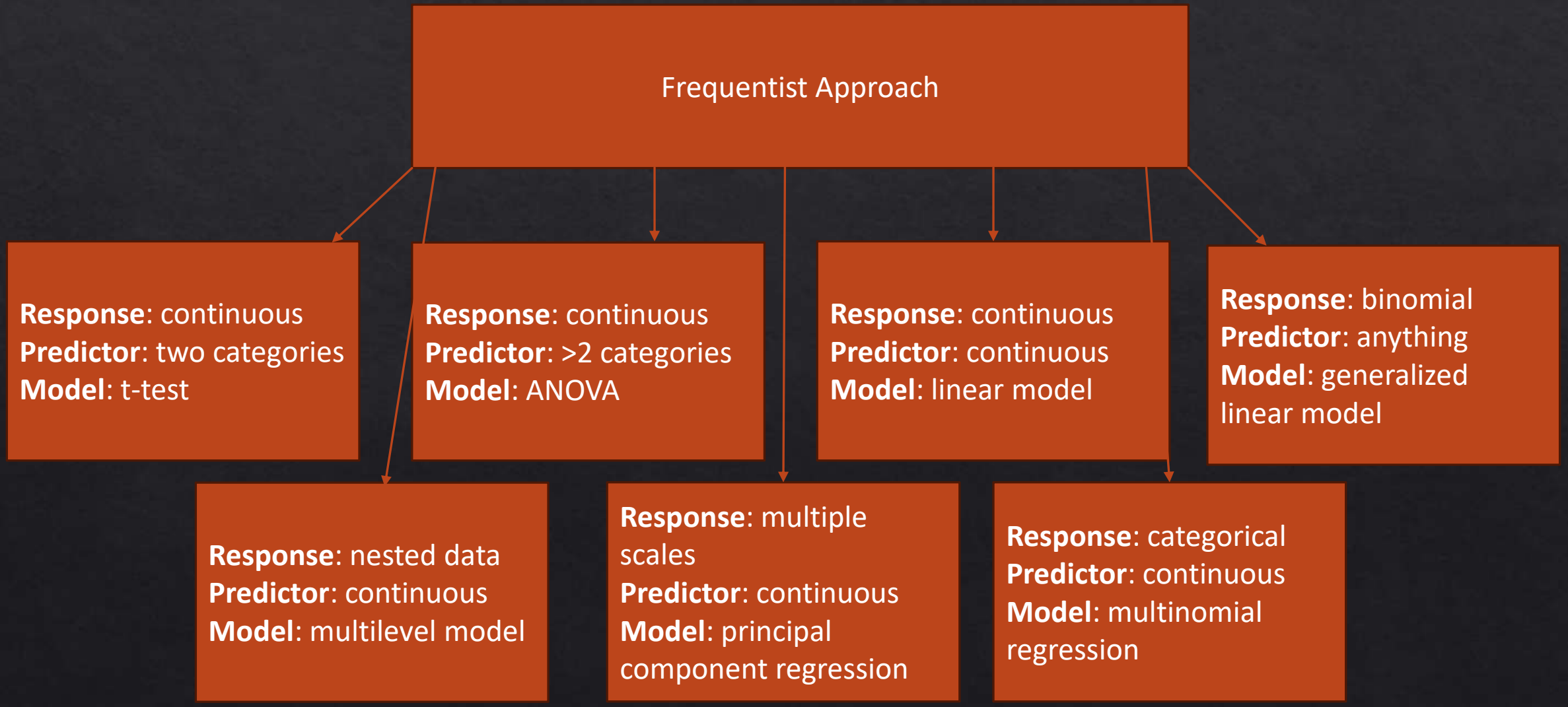
Expanding your horizons

- All the models we have looked at so far rely on p-values, null hypotheses, and model comparison
- This is actually a specific type of inferential statistics, there are more out there with other purposes…

Statistical Models



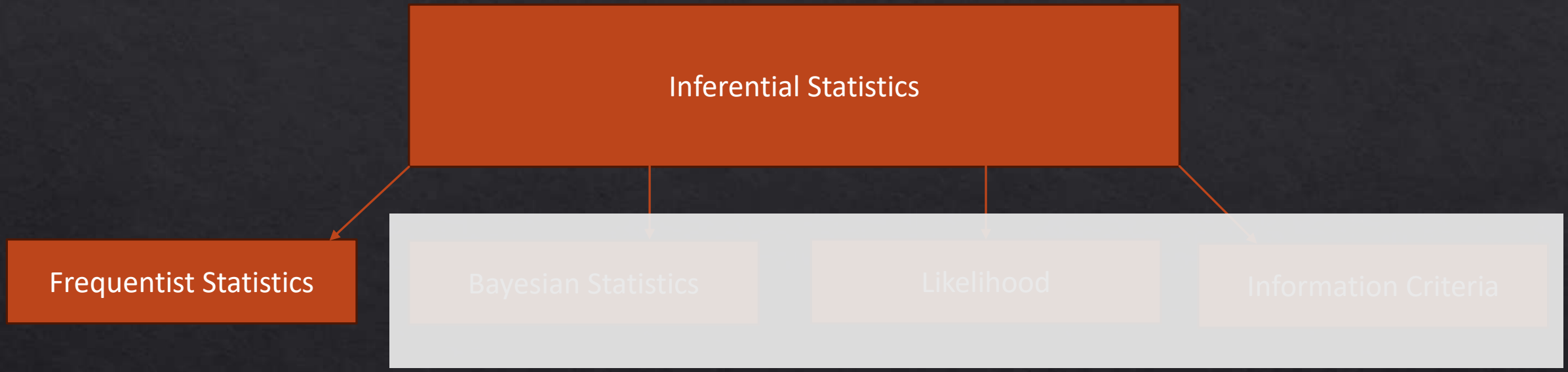
Statistical Models



Why would we ever consider using a different approach?

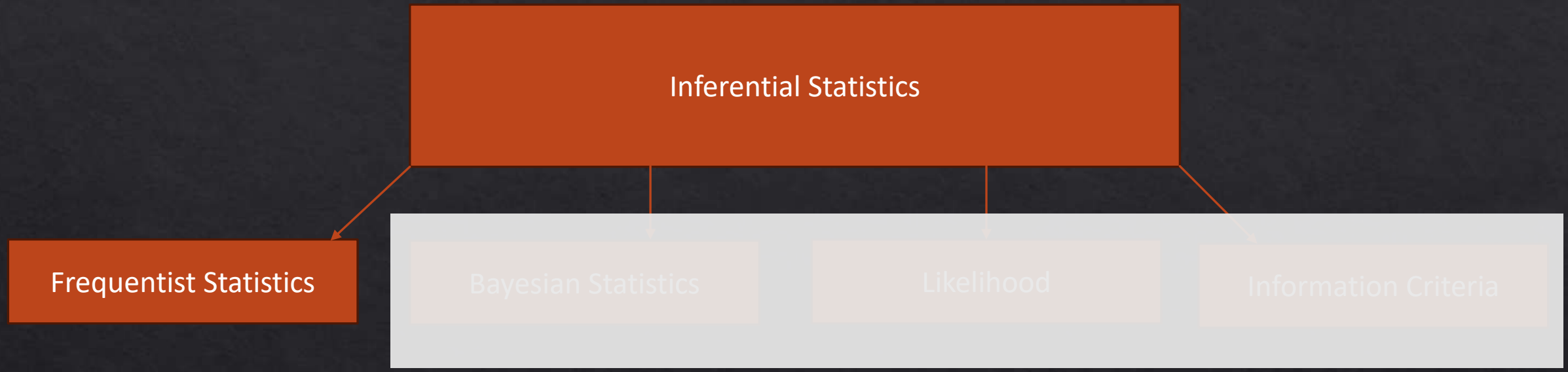
- P-values are often misinterpreted.
  - 0.05 is arbitrary!
  - They scale with sample size!
  - We often need to do multiple tests, which introduces errors!
- Null hypotheses are often meaningless. Instead, we should ask if effects are non-zero, or how large differences between populations are
- Parametric frequentist tests are often inflexible, sometimes other approaches are needed

Statistical Frameworks



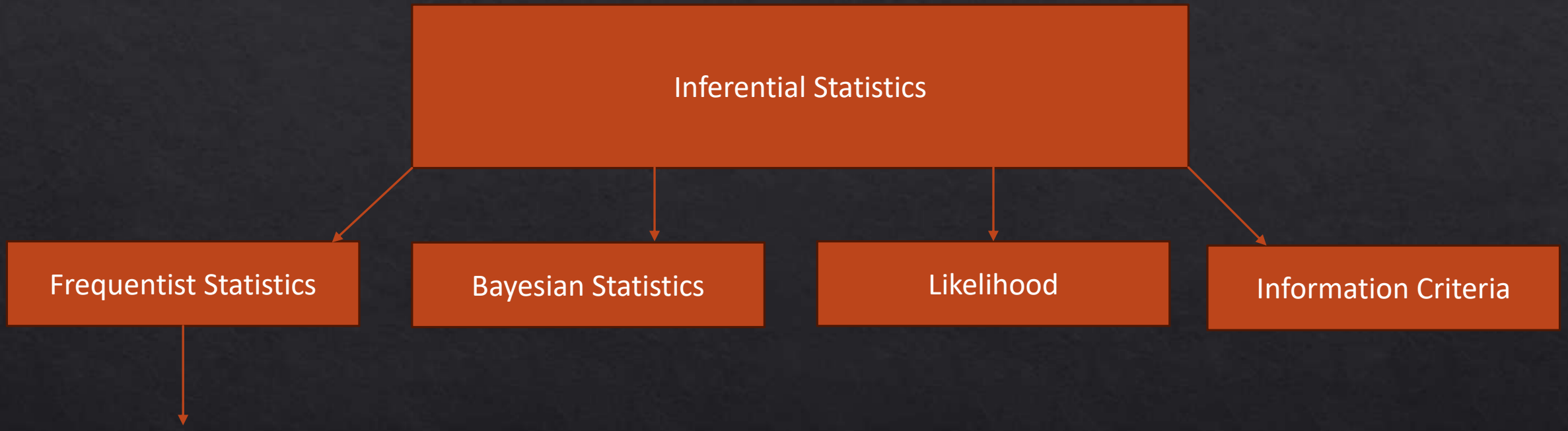
It is likely nearly every test you know is from a frequentist approach. Anything that uses a null hypothesis test or a p-value is a frequentist test.

Statistical Frameworks



These are all different ways to infer patterns in our data. They can tell us information about how well a model fits compared to other models, or about the "importance" of variables. They are not mutually exclusive, and often we use several together to get a better picture of what is going on.

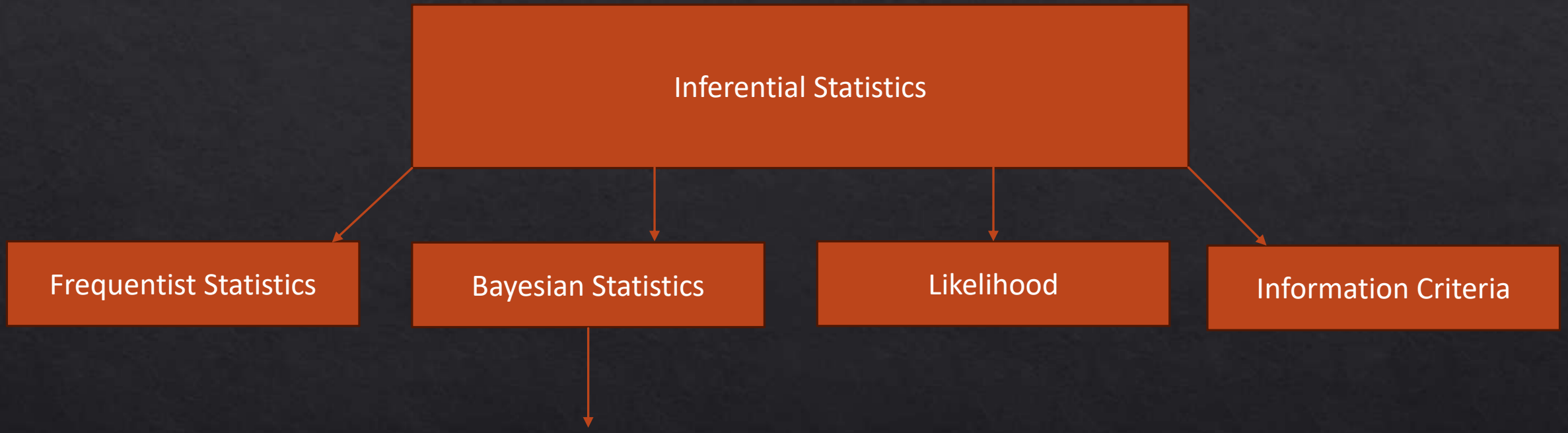
Statistical Frameworks



Common uses:

- p-values
- confidence intervals
- null hypothesis significance testing

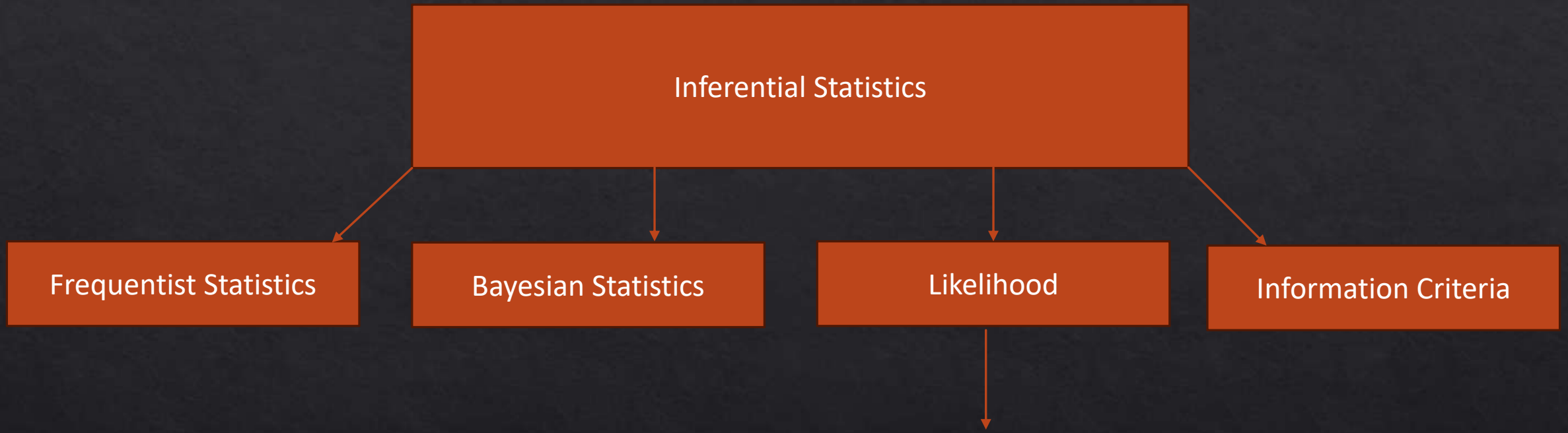
Statistical Frameworks



Common uses:

- Credible intervals
- Bayes factor (model comparison)

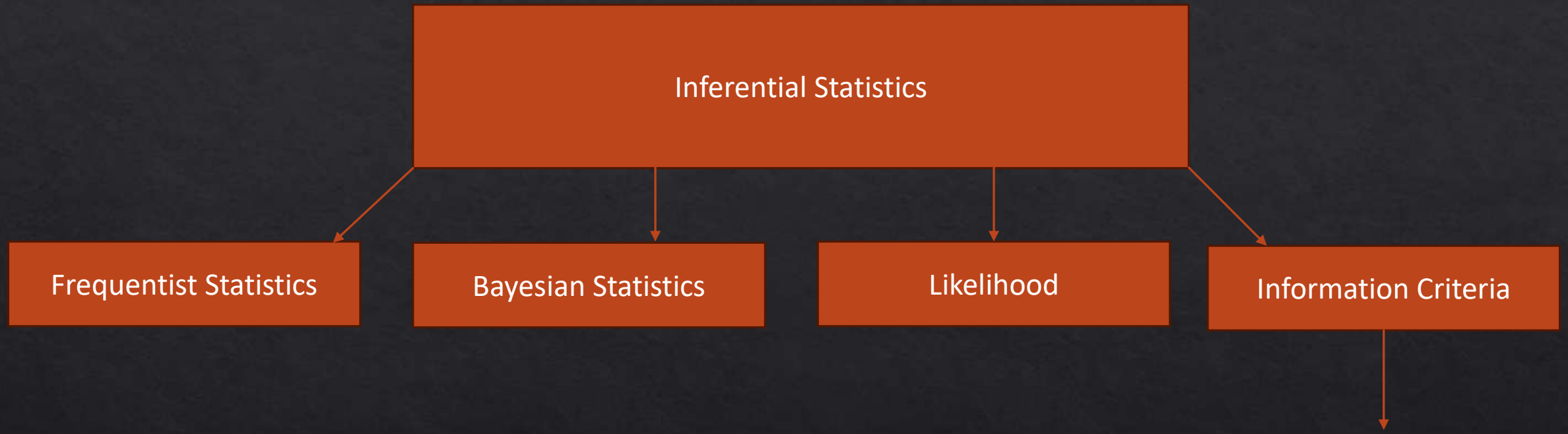
Statistical Frameworks



Common uses:

- Maximum Likelihood Estimation
- Likelihood-ratio tests (model comparison)

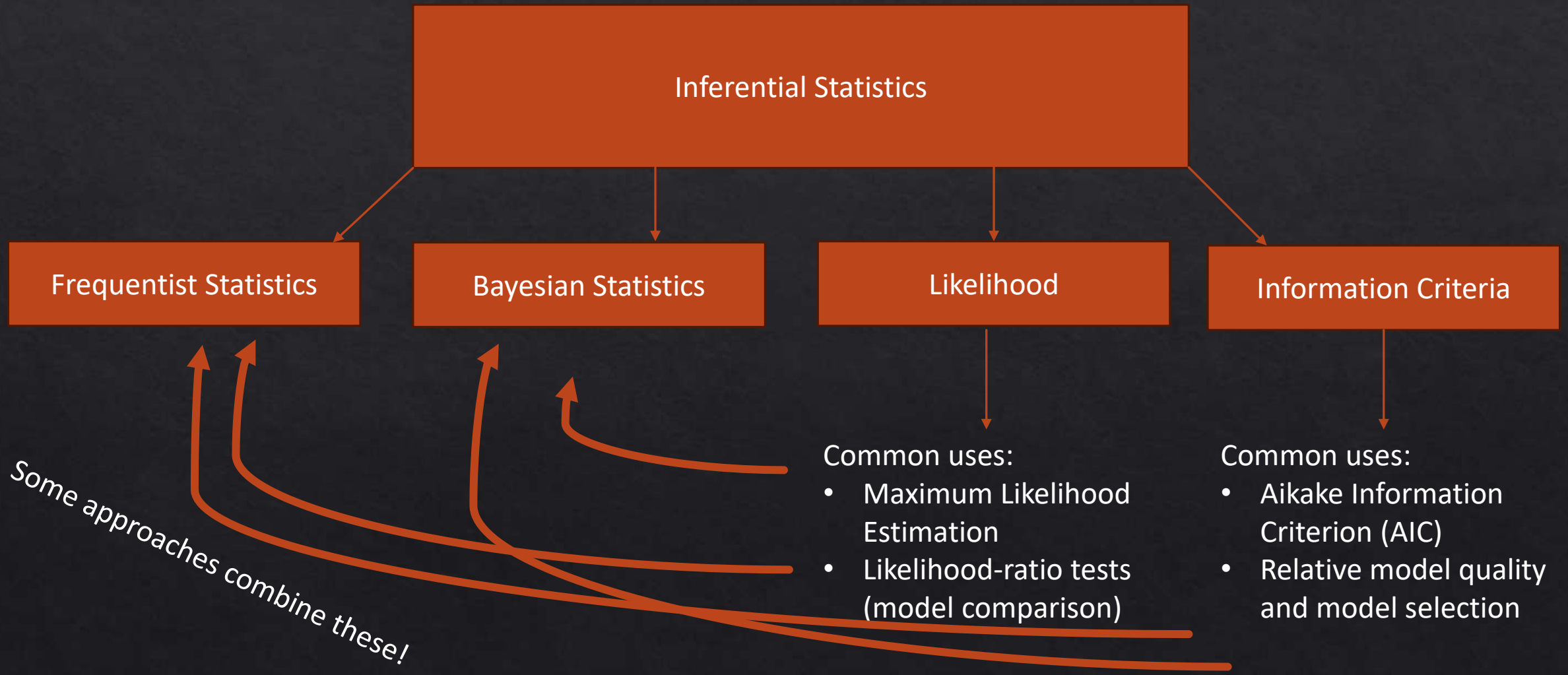
Statistical Frameworks



Common uses:

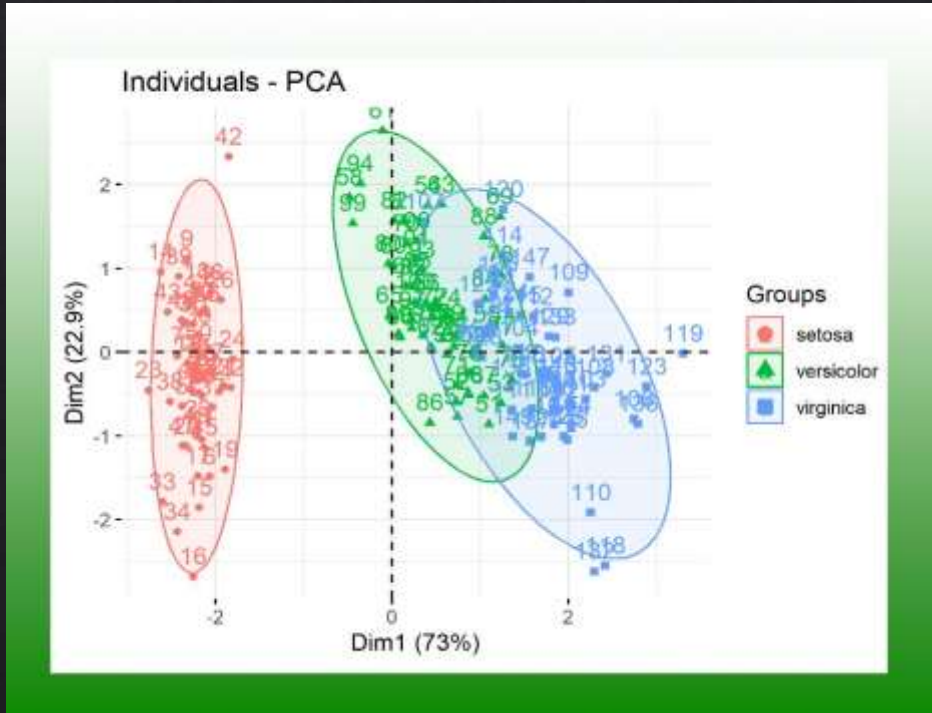
- Akaike Information Criterion (AIC)
- Relative model quality and model selection

Statistical Frameworks



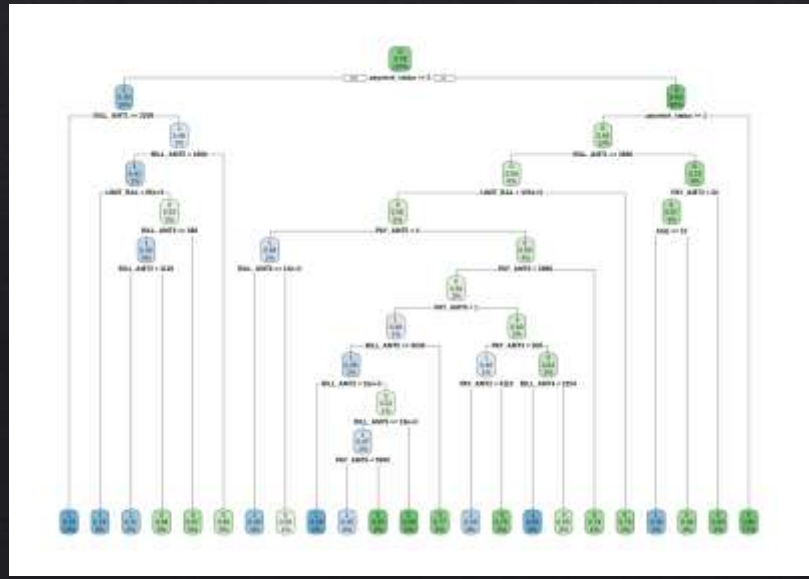
Wait, what about…

- Multivariate analyses
- We have looked at a lot of linear analyses, most of which have continuous or categorical predictors, and a single response variable. But what if you have a lot of variables, many of which correlate with each other? So many they can't be easily summarised? Then we use dimension reduction to create a set of uncorrelated variables. Useful for when we have a lot of inter-correlated variables!
- Key approaches include:
  - Principal Component Analysis (PCA)
  - Factor analysis
  - Cluster analysis
  - Canonical Correlation Analysis (CCA)
  - Correspondence Analysis (CA)



Wait, what about…

- **Classification model:** Most models we looked at have continuous response variables. But what if they are categorical? One option is to use variables and information to classify them into different groups. They often use machine learning to find the best solution by trial and error.
- **Machine learning:** this is an approach, not a statistical test. In a machine learning process the machine tries and learns to fit a model, typically using an Information Criterion. While making one is not simple, they follow all the same principles as model fitting and model comparison. In fact, R uses a lot of machine learning behind the scenes for some of the more complex tests we've done! multivariate analysis
- Useful for complex, hierarchical data, such as phylogenetics, population structures, population movement etc.
- Key approaches include:
  - Random forest
  - Gradient boost models
  - K-nearest neighbour classification
  - Decision tree analysis



Wait, what about…

- **Species Distribution Models:** Most commonly correlative models that identify what variables predict species presence or species abundance. This is not a statistical test. In fact they rely on several underlying approaches including
  - GLMs
  - GAMs
  - Machine learning algorithms
- Often we run several models and then use model average to reach an “ensemble” model.

Optional: Bayesian Statistics

Bayesian Statistics

- Bayesian statistics use a different approach. There is no null model, there are no p-values, and no residuals (sort of). There is no theoretical “ideal” distribution. Instead, the data is considered to be “reality” and we just see how well we can describe it mathematically.
- Bayesian statistics rely heavily on priors. This is information we provide to the model about what we think reality is like
- This reliance on “Bayes theorem” is critical. We tell the model what we think reality is like, and then the model find parameters to describe it.
- This may seem odd, and subjective, but it is more important and appropriate than you think

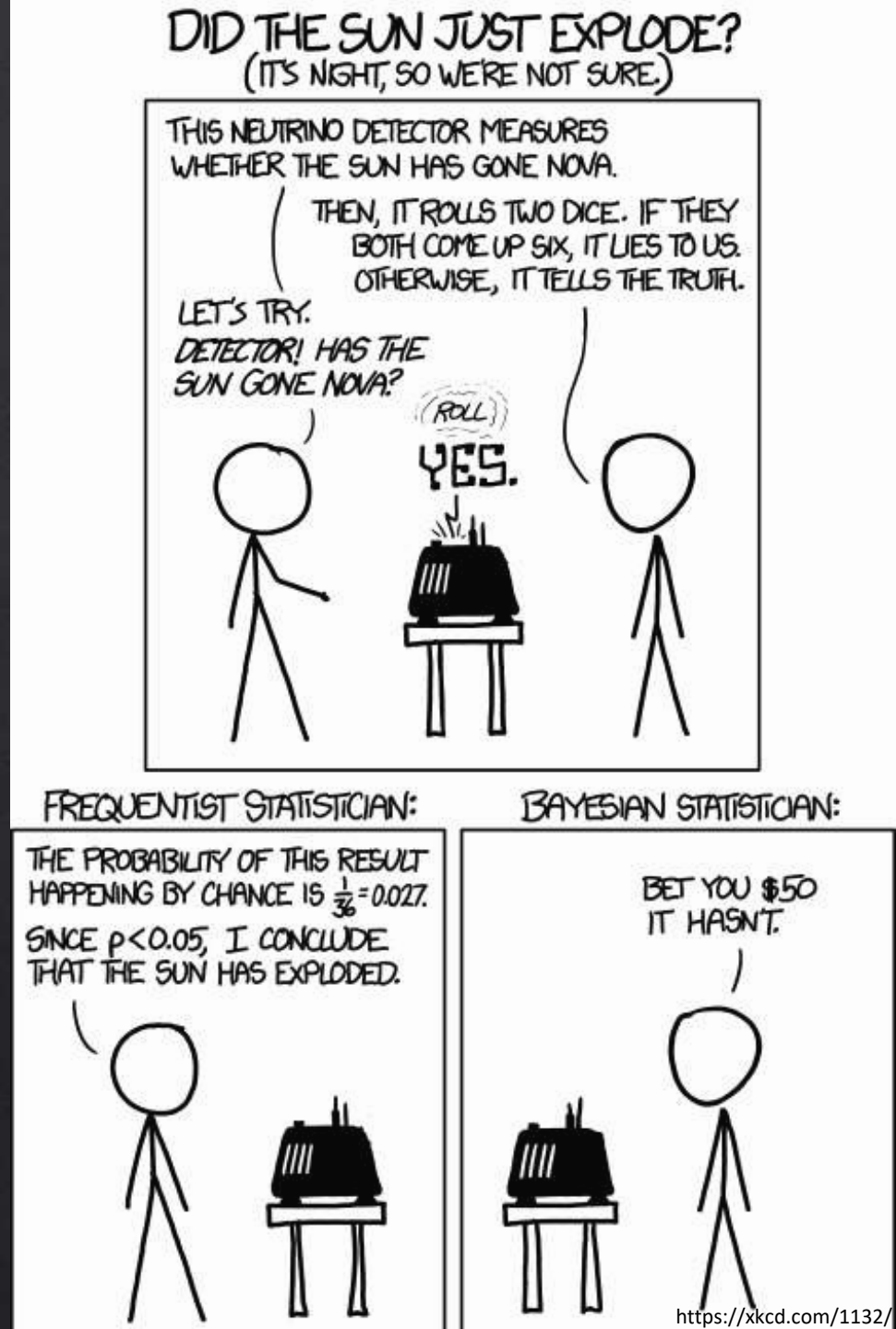
Bayesian Statistics

- Here's a famous case study
- *"The serum test screens pregnant women for babies with Down's syndrome. The test is a very good one, but not perfect. Roughly 1% of babies have Down's syndrome. If the baby has Down's syndrome, there is a 90% chance that the result will be positive. If the baby is unaffected, there is still a 1% chance that the result will be positive. A pregnant woman has been tested and the result is positive."*
- What is the chance that her baby actually has Down's syndrome?

- 48%! This is non-intuitive, but the knowledge of the general probability of prevalence has revealed some interesting information!
- Priors make modelling faster as it only trials the parameter estimates we tell it to. They also rule out physically impossible, but mathematically valid, answers. They also can account for uncertainty in observations

Bayesian Statistics

- The basic process for making a Bayesian model works like this:
  - define the model
  - define the priors
  - run the model
  - check model ran properly (convergence, fuzzy caterpillars etc)
  - check parameter output
  - make conclusions and plots



https://xkcd.com/1132/

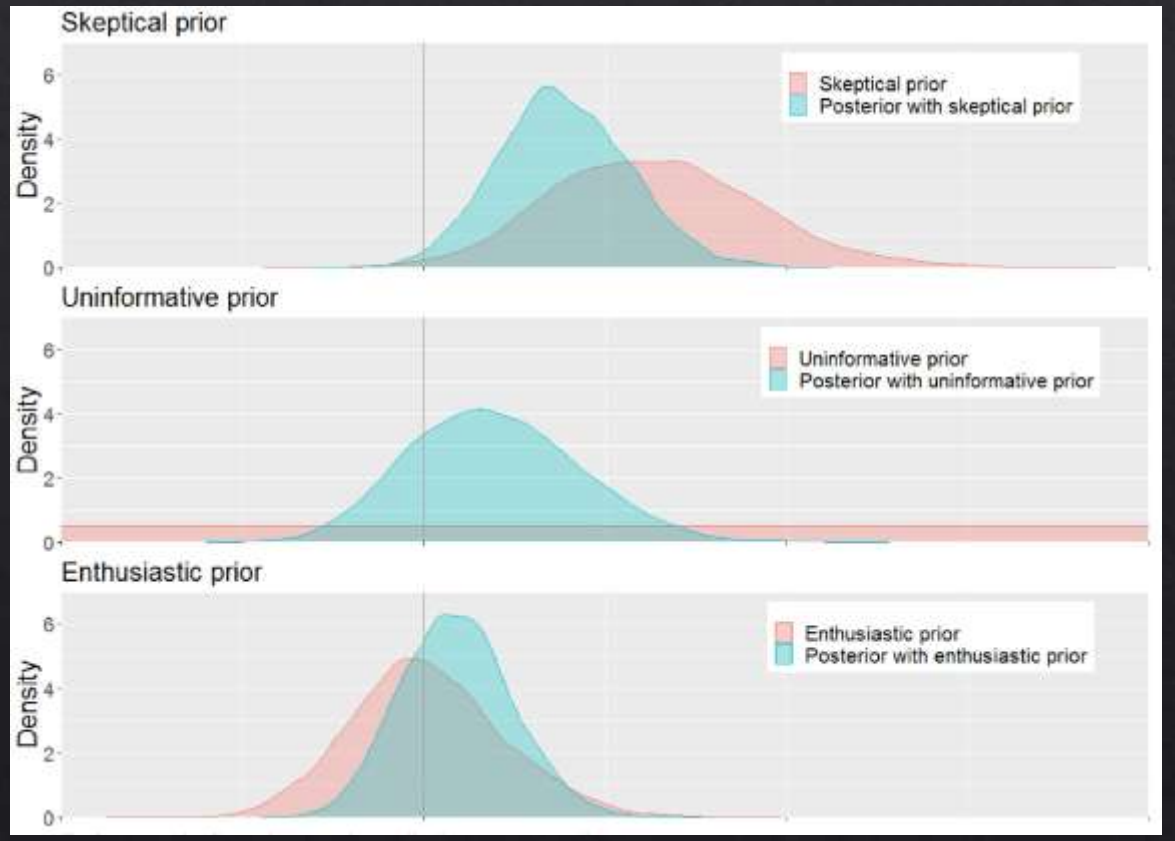
The Bayesian Approach: Example

- We are going to compare jaw length in male and female jackals.
- This is a normal distribution, that may vary between two categories (male and female)
- As it's a normal distribution it only has two parameters, the mean and the standard deviation
- We set the priors to be specific, but with some flexibility to change
- Here's the model:

```
jaw.mod<-function(){

  # Priors
    mu.male ~ dnorm(100, 0.001) # mean of male jaw lengths
    mu.female ~ dnorm(100, 0.001) # mean of female jaw lengths
    sigma ~ dunif(0, 30) # common sigma
    tau <- 1/(sigma*sigma) #precision

  # Likelihood (Y | mu.male, mu.female, sigma) = Normal(mu[sex], sigma^2)
    for(i in 1:nmales){
      males[i] ~ dnorm(mu.male, tau)
    }
    for(i in 1:nfemales){
      females[i] ~ dnorm(mu.female, tau)
    }

  # Derived quantities:  difference in means
    mu.diff <- mu.male - mu.female
}
```

The Bayesian Approach: Example

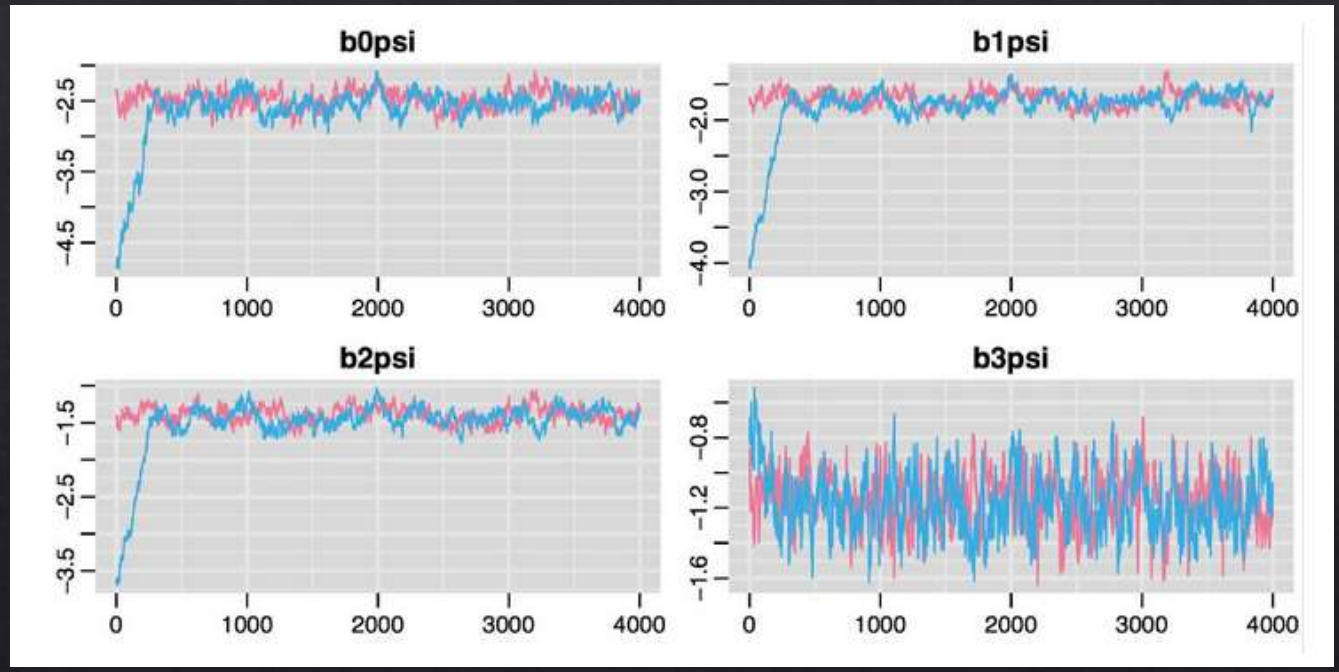
- It is up to us on how restrictive or flexible the priors are
- Many are tempted to use an “uninformative” prior, but these often bias models AND make them harder to converge.



https://www.redjournal.org/article/S0360-3016%2821%2903256-9/fulltext

The Bayesian Approach: Example

- We run the model and make sure it's converged. The model will iterate across the parameter space and try out all (or nearly all) possible solutions.
- It should resolve to the most likely solutions. This consistency is called "convergence".
- We can check for it by looking for "fuzzy caterpillars"



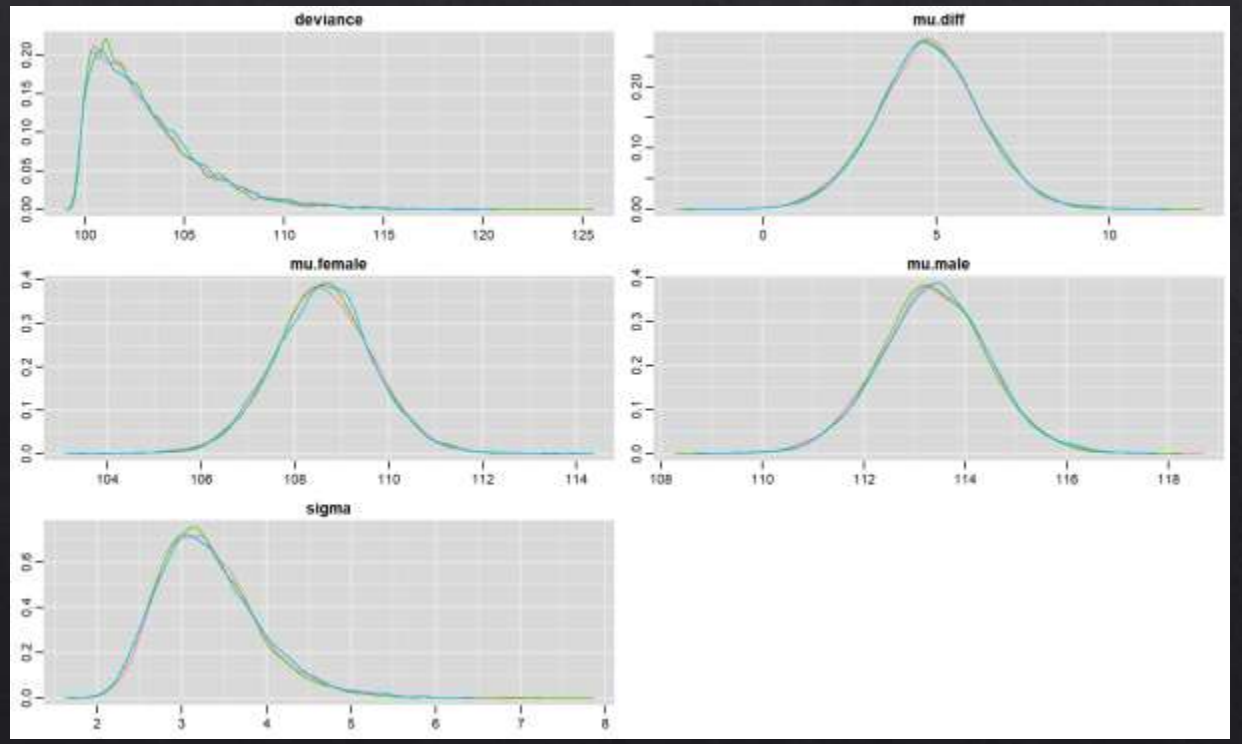
The Bayesian Approach: Example

- The output of a Bayesian model will tell us where the parameters are by the end
- This output is known as the “posterior” distribution
- These are equivalent to parameter estimates in normal GLMs, but intrinsically come with credible intervals.

```
Inference for Bugs model at "jaw.mod", fit using jags,
 3 chains, each with 10000 iterations (first 5000 discarded)
 n.sims = 15000 iterations saved. Running time = secs
          mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
mu.diff     4.779   1.508   1.775   3.797   4.772   5.760   7.740 1.001 15000
mu.female 108.602   1.064 106.501 107.912 108.598 109.284 110.683 1.001 15000
mu.male   113.381   1.069 111.262 112.688 113.375 114.082 115.511 1.001  9600
sigma       3.317   0.594   2.380   2.895   3.239   3.655   4.701 1.002  3200
deviance  103.075   2.744  99.892 101.036 102.395 104.379 110.156 1.001  4900

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule: pV = var(deviance)/2)
pV = 3.8 and DIC = 106.8
DIC is an estimate of expected predictive error (lower deviance is better).
```

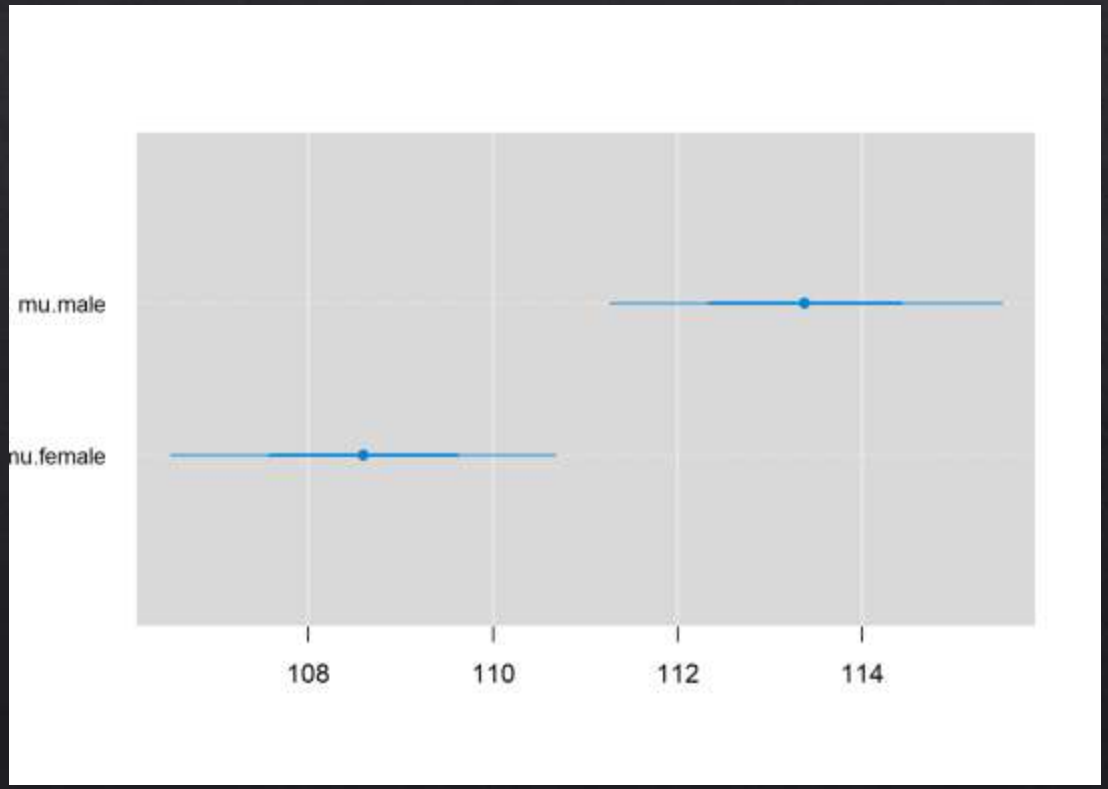
The Bayesian Approach: Example

- We can look at these estimates as distributions. Where is the mean estimate, how certain is the model that this is the “real” estimate?



The Bayesian Approach: Example

- We use credible intervals to check whether there is any overlaps or substantial differences. For example, is there any overlap in average female and average male jaw size at 90% CI?



The Bayesian Approach: When do I use it?

- If it makes sense for your data (heavy conditional probabilities)
- If standard frequentist methods don't work
  - Lots of random effects?
  - Highly hierarchical data?
  - No standard distributions work?
  - Other problems that can't be modelled easily in a frequentist approach (e.g. heteroskedasticity)?
- If the model you wish to build has too much complexity for the sample size. Bayesian approaches can compensate for small sample sizes to some extent, especially for categories, interactions and other terms that require a large number of degrees of freedom in a frequentist approach
- If you like the Bayesian approach and have a lot of time on your hands!
  - Makes you specify your model and really understand your data
  - Also takes time and experience to run them!