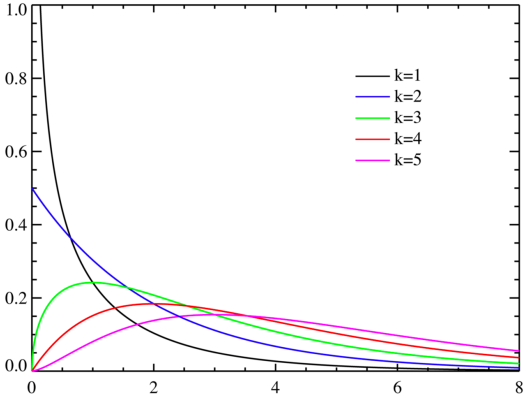
Parameter - related distributions



Normal Distribution N~(mean, sd)
The mother of all distributions!

↓ $Z = (x - M)/D$

Z Distribution N~(0,1)
standard normal distribution

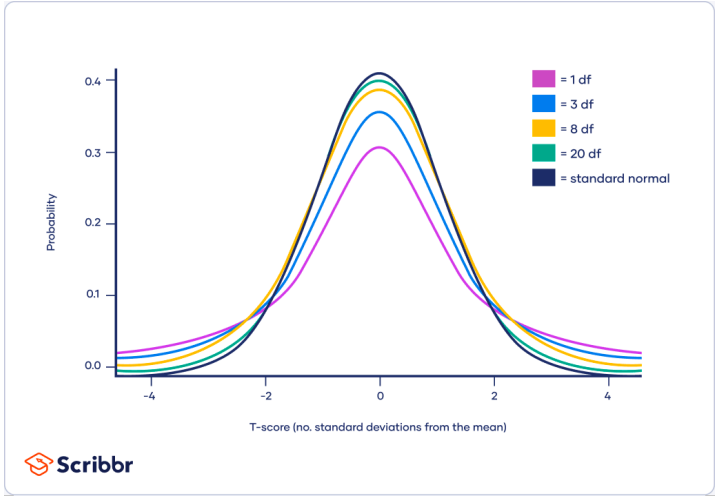
→ **Chi-squared Distribution $Q \sim X^2(df)$**

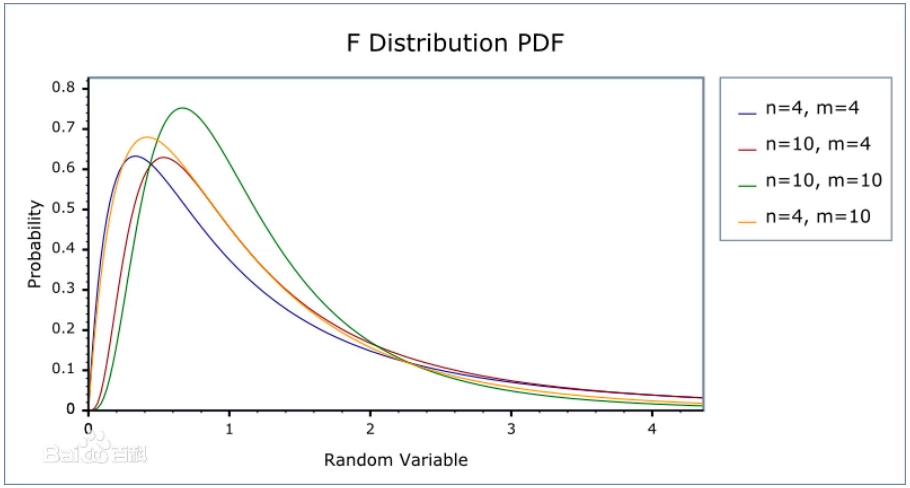
↓ $T = (m - M)/\{d/\mathrm{sqrt}(\mathbf{n})\}$

t Distribution t~(df)
smaller sample size

↓

F Distribution F ~ F(df1, df2)



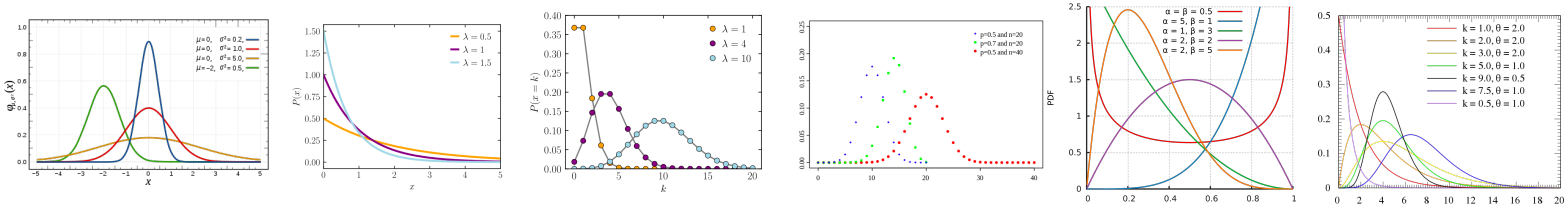


Variable-related distributions

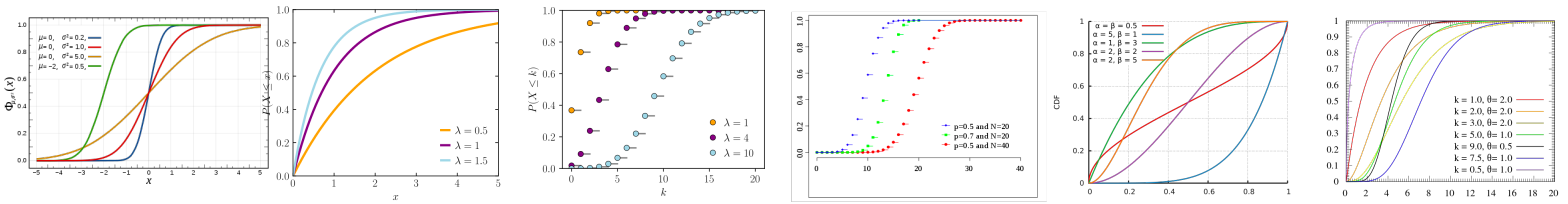
Distribution	Data Type	Range	Example	Parameters
Normal	Continuous	[-infinite, infinite]	“normal” data	mean, sd
Exponential	Continuous	(0, infinite]	growth	rate
Poisson	Discrete	[0, infinite]	count data	lambda
Binomial	Binary	0/1	success/ failure	n, rate
Uniform	Continuous	[-infinite, infinite]	unloaded dice	min, max
Beta	Continuous	[0, 1]	ratio	shape1, shape2
Gamma	Continuous	[0, infinite]	rainfalls	shape, scale

Distributions recap

Probability density/mass function



Cumulative distribution function



R function on distribution recap

notation	function	Input	Output	Example
d	density	x	height	d(0) = 0.4
p	probability	x	area under curve till x	p(0)=0.5
q	quantile	area under curve till x	x	q(0.5)=0
r	generator	n	n datapoints	r(10) =

Lesson 10:
The animal model

The limits of GLS

- We have learnt how to run linear models with complex autocorrelation structures (GLS)

- Unfortunately, the method has limits:

1. It assumes the **response data is normally distributed**
2. It cannot handle **replication of individuals in species**

So how can we deal with repetition and response data with non-normal distributions?

The animal model

- The problems described in the previous slide can be dealt with using an extension of mixed models and thus extending the covariance matrix **V**

- Recall, the general likelihood solution of the coefficient estimates in linear models is:

$$\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}y$$

- In the case of ordinary linear models,

$$y = X\beta + \varepsilon \quad \text{where} \quad \varepsilon \sim N\left(0, I\sigma_R^2\right), \quad \text{so } V = I\sigma_R^2$$

- Hence:

$$\hat{\beta} = (X'X)^{-1}X'y \qquad (\because V^{-1} = I^{-1} = I)$$

The animal model

- In mixed models with grouping random effects,

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Zb} + \boldsymbol{\varepsilon} \quad \text{where} \quad \boldsymbol{\varepsilon} \sim \boldsymbol{N}\left(\boldsymbol{0}, \boldsymbol{Z}\sigma_B^2\right), \; \boldsymbol{\varepsilon} \sim \boldsymbol{N}\left(\boldsymbol{0}, \boldsymbol{I}\sigma_R^2\right),$$

- as random effects are simply variance terms,

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Z}\sigma_B^2 + \boldsymbol{I}\sigma_R^2, \qquad \text{so } \boldsymbol{V} \sim \boldsymbol{Z}\sigma_B^2 + \boldsymbol{I}\sigma_R^2$$

- Phylo GLS is a special case where the residual errors are distributed according to the **phylogeny covariance matrix A**

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon} \quad \text{where} \quad \boldsymbol{\varepsilon} \sim \boldsymbol{N}\left(\boldsymbol{0}, \boldsymbol{A}\sigma_R^2\right), \quad \text{so } \boldsymbol{V} \sim \boldsymbol{A}\sigma_R^2$$

- It is easy to see that phylogenetic covariance can instead be included as a separate random term,

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Z}\sigma_B^2 + \boldsymbol{A}\sigma_a^2 + \boldsymbol{I}\sigma_R^2, \qquad \text{so: } \boldsymbol{V} \sim \boldsymbol{Z}\sigma_B^2 + \boldsymbol{A}\sigma_a^2 + \boldsymbol{I}\sigma_R^2$$

The animal model

- The expression:

A: additive effect (breeding value)

$$y = X\beta + Z\sigma_B^2 + A\sigma_a^2 + I\sigma_R^2$$

- is known as the animal model because it was first developed in quantitative genetics research for animal breeding, where ancestry was an important consideration for breed traits. Phylogenies are similar to ancestries, so model applicable in comparative biology

- The elegance of using the phylogenetic data as an additive term in the model means that we can now simply use the mixed model framework to combine multiple individuals per species and non-normal response data with phylogenetic information in GLMM

Solving the Animal Model

1. Since early 1980s, animal breeders have successfully used a frequentist approach with restricted maximum likelihood (REML), to for example increase meat or milk yield in cattle (Simm 1998). However, inference with REML is not trivial for GLMM models. Models for non-Gaussian traits especially are challenging in regard to uncertainty in breeding values and other parameter estimates (Tempelman and Gianola 1994; Sorensen and Gianola 2002; Bolker *et al.* 2009; Fong *et al.* 2010).

2. A popular approach is best linear unbiased prediction (BLUP) (Henderson 1950) for calculating breeding values (Wilson *et al.* 2009). However, BLUP ignores all the uncertainty associated with the estimation and are not suitable for hypothesis testing in evolutionary questions (Postma 2006; Wilson *et al.* 2009; Hadfield *et al.* 2010).

3. Another approach is to perform modeling in a Bayesian framework.

A crisis of code

- The only package I am aware of that runs the animal model using the frequentist statistical framework is *asreml*. Unfortunately we have to pay to use this software!!!

- By contrast there are several packages for handling the animal model using the Bayesian statistical framework. Therefore we will run the animal model using Bayesian packages in R. I am aware of two packages that do this well: *MCMCglmm* and *brms*. We will use *brms* here.

- The linear function in brms has been set up to mimic glmer() so is relatively easy to use

Frequentism vs Bayesianism

- Before going any further, its important that we say something very briefly about the main differences between frequentist versus bayesian approaches

- It is necessary to understand this to understand the basics of the code

Frequentism

- Frequentists use only the data collected in the present experiment to draw inferences.

- Thus the likelihood function for a normally distributed response variable uses only the present information

$$L(\boldsymbol{y}, \boldsymbol{X}, \boxed{?}, \sigma^2) = \frac{1}{(2\pi)^{n/2}\sigma^n} exp\left(-\frac{1}{2\sigma^2}(y - \boxed{?})^2\right)$$

- $\boldsymbol{\mu}$ is representative for the betas here

Bayesianism

- Bayesianists assume that there is pre-existing data describing the parameter/s ($\boldsymbol{\mu}$,$\boldsymbol{\sigma}$) that can be combined with the present information to draw inferences. This information is known as the prior information

- Thus the likelihood function uses only present information

$$L(y, X, \square, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(y - \square)^2\right)$$

- The Prior is earlier information describing the parameters.

$$P\left(\square, \square^{\,2}\right) = \frac{1}{\sqrt{2\pi\square^{\,2}}} exp\left(-\frac{1}{2\square^{\,2}}(\square - M)^2\right)$$

- Where M and $\boldsymbol{\tau}$ are estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ from earlier experiments/ knowledge.

Ex: A Patient Tested for a Disease

Data:
The patient was tested positive

Question:
What's the probability that this patient has disease?

A Patient Tested for a Disease

Data:

1. The patient was tested positive
2. The accuracy of the test is 95%

Question:

What's the probability that this patient has disease?

A Patient Tested for a Disease

Data:

1. The patient was tested positive
2. The accuracy of the test is 95%
3. This disease affects 1% of the population

Question:

What's the probability that this patient has disease?

Bayesian theorem

贝叶斯定理

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Bayesian theorem

贝叶斯定理

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Bayesian theorem
贝叶斯定理

$$P(A\ ,\ B) = P(A \mid B) \cdot P(B) = P(B \mid A) \cdot P(A)$$

A Patient Tested for a Disease

Data:

1. The patient was tested positive $P(T)$
2. The accuracy of the test is 95%

$P(T|D) = 0.95, P(T'|D') = 0.95$

3. This disease affects 1% of the population

$P(D) = 0.01$

Question:

What's the probability that this patient has disease?

$P(D|T) = P(T|D) \cdot P(D)/P(T)$

A Patient Tested for a Disease

Data:

1. The patient was tested positive

$P(T) = P(T|D) \cdot P(D) + P(T|D') \cdot P(D') = 6\%$

2. The accuracy of the test is 95%

$P(T|D) = 0.95, P(T'|D') = 0.95$

3. This disease affects 1% of the population

$P(D) = 0.01$

Question:

What's the probability that this patient has disease?

$P(D|T) = P(T|D) \cdot P(D)/P(T)$

A Patient Tested for a Disease

Question:
What's the probability that this patient has disease?

Data:

1. The patient was tested positive

1

2. The accuracy of the test is 95%

0.95

3. This disease affects 1% of the population

A Patient Tested for a Disease

Question:
What's the probability that this patient has disease?

Data:

1. The patient was tested positive

1

2. The accuracy of the test is 95%

0.95

3. This disease affects 1% of the population

0.16

4. This patient came for the test because he does not feel well.

Bayesian Thinking
贝叶斯逻辑

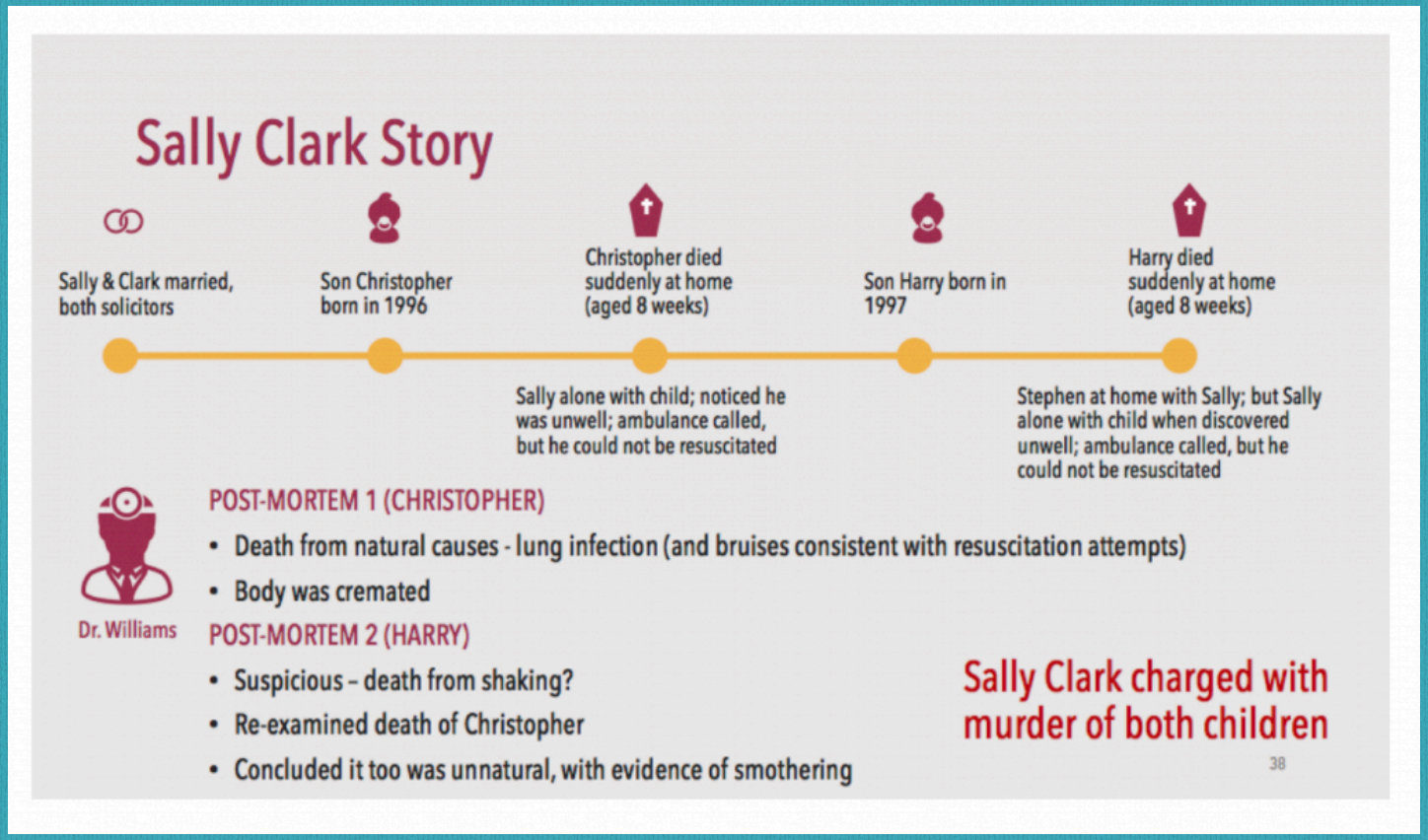
1. Estimates the statistical probability of something being true

2. updates that probability as new evidence appears

3. approaching the truth without achieving absolute certainty

人：什么是机器学习?
机器：问我一个问题
人：3乘以7等于多少?
机器：7
人：连三七二十一都不知道
机器：21

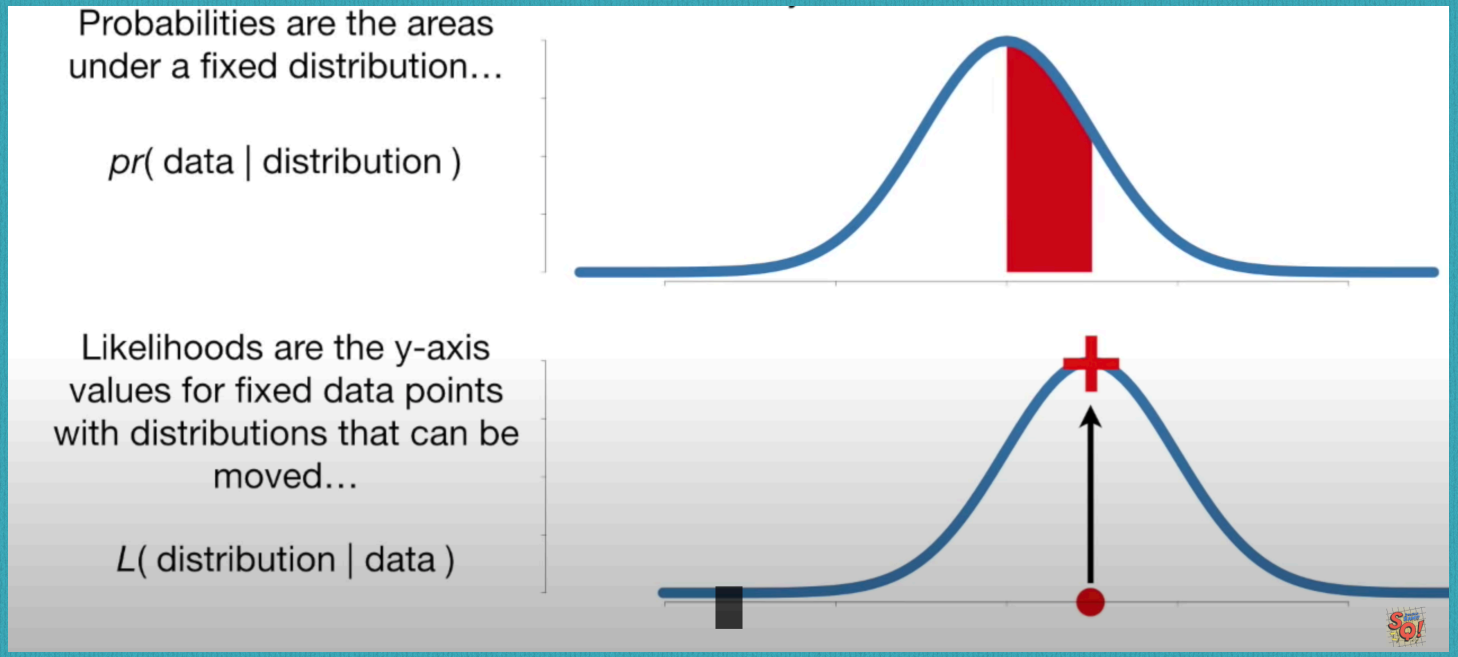
Prosecutor's fallacy



Bayesianism

- Posterior = Likelihood × Prior
- $$P(\square, \sigma^2) \propto \frac{1}{\sqrt{\square^{\,2}\sigma^2}}\, exp\left(\frac{-(\square - M)^2}{2\square^{\,2}} + \frac{-(y - XB)^2}{2\sigma^2}\right)$$
- Then takes logs and solve posterior likelihood as before
- -log Posterior = -log Likelihood + -log Prior

“Probability vs. Likelihood”



https://www.youtube.com/watch?v=pYxNSUDSFH4

Prior and Posterior

先验与后验

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

P(A) = Having disease: 1%

P(A | B): Having disease given tested positive

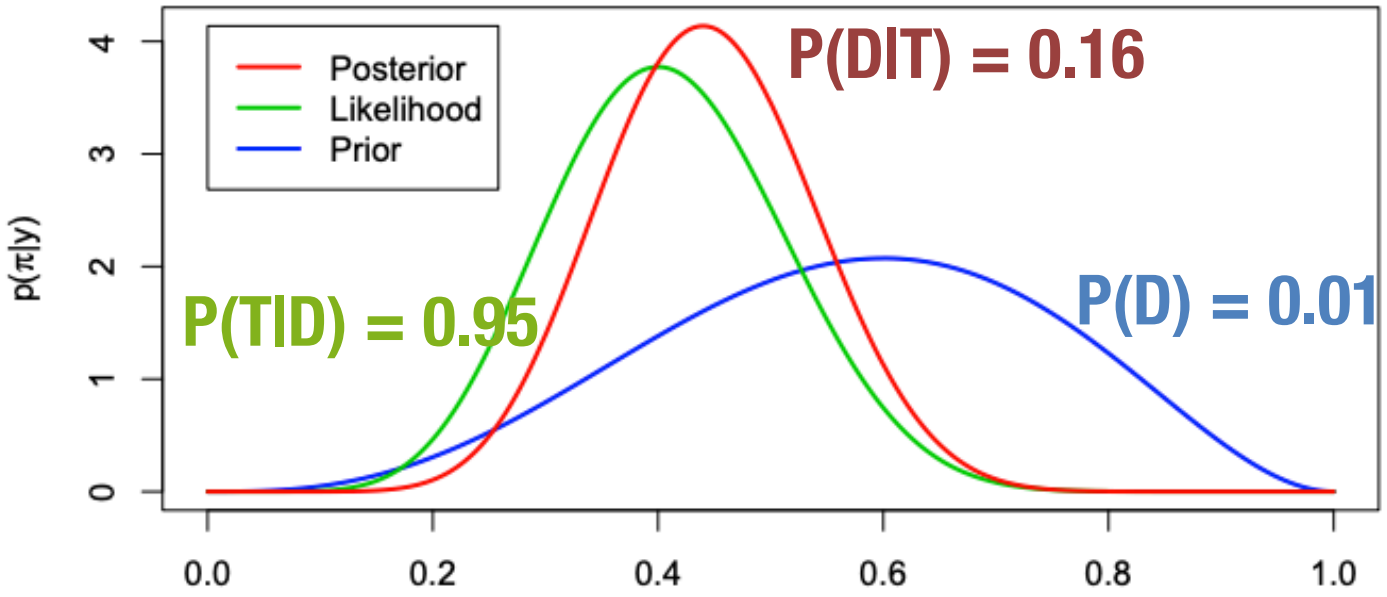
P(B | A): Tested positive given having the disease: 0.95

P(B) = Tested positive

Existing/easy to collect info

Data collected

Frequentism vs Bayesianism



Prior + Data $\xrightarrow{\pi}$ **Posterior**

pre-existing data + present information -> inferences

Prior + Data —?—> Posterior

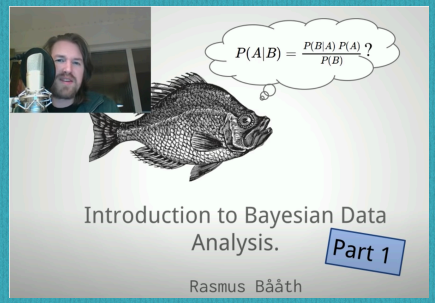
pre-existing data + present information -> inferences

生成模型

“Generative model:

a story of how your data came to be.”

–Rasmus Baath



Linear model

X → Model → New shadehouse

Light Damage + LMM = Growth

Generative model: 熟悉的陌生人

Parameters → Model → + Some Randomness → Data

10, 0, 1 + rnorm() rbinom() rpois() rbeta() =

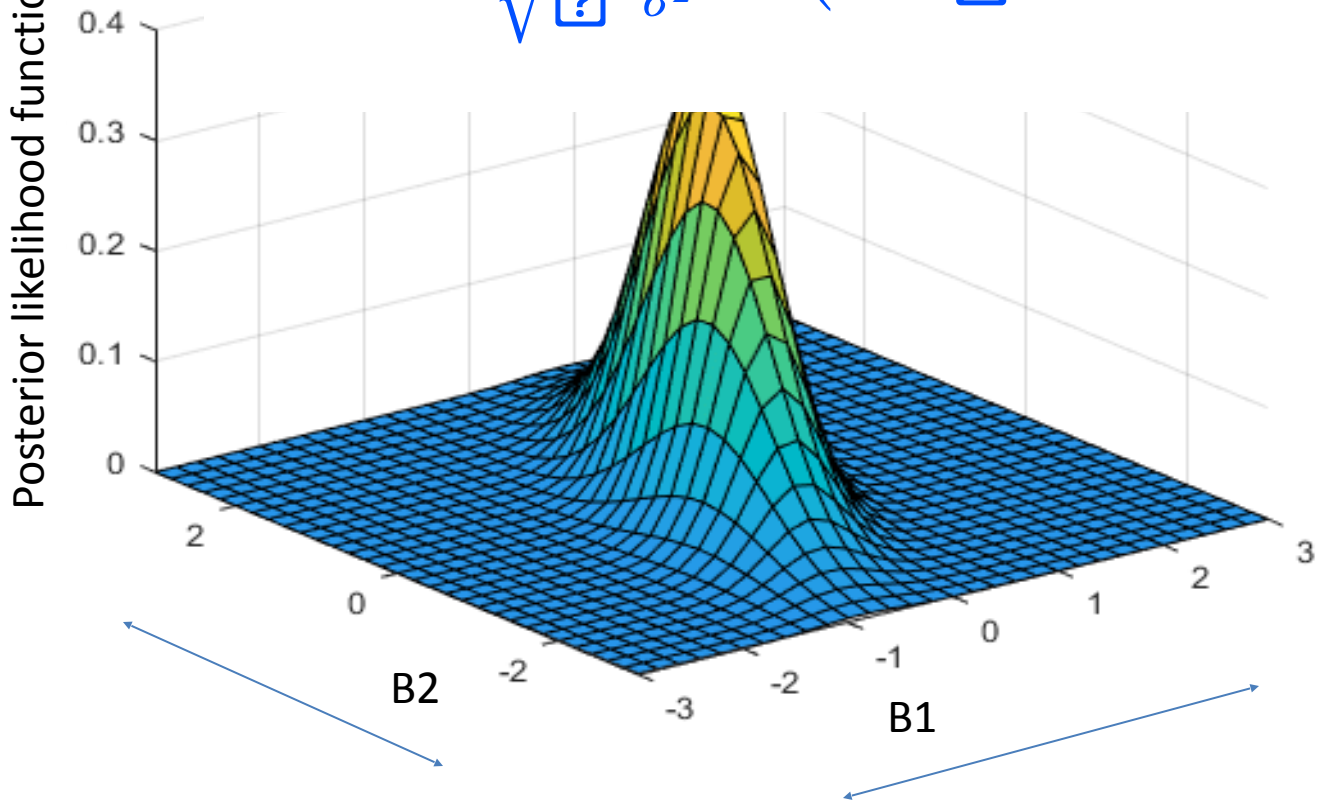
0.22 -0.19 1.31
-1.67 -0.43 0.70
0.19 1.45 -0.31 0.64

From Yesterday

```
#now lets make some modeled response values
b0 <- 0
sd.b <- 1   #across sites variation: random effect residual
sd.e <- .15  #within sites variation, model residual

for(i in d$site){
        dd <- d[d$site == i,]
        nn <- nrow(dd)
        #introduce between site variation
        b1.site <- 0 + rnorm(n=1, mean=0, sd=sd.b) #site mean
        #introduce within site variation
        d$x[d$site == i] <- b0 + b1.site + rnorm(n=nn, sd=sd.e)
}
```

Parameter drawing in Bayesian models

$$P(?, \sigma^2) \propto \frac{1}{\sqrt{?^2 \sigma^2}} exp\left(\frac{-(? - M)^2}{2?^2} + \frac{-(y - XB)^2}{2\sigma^2} \right)$$

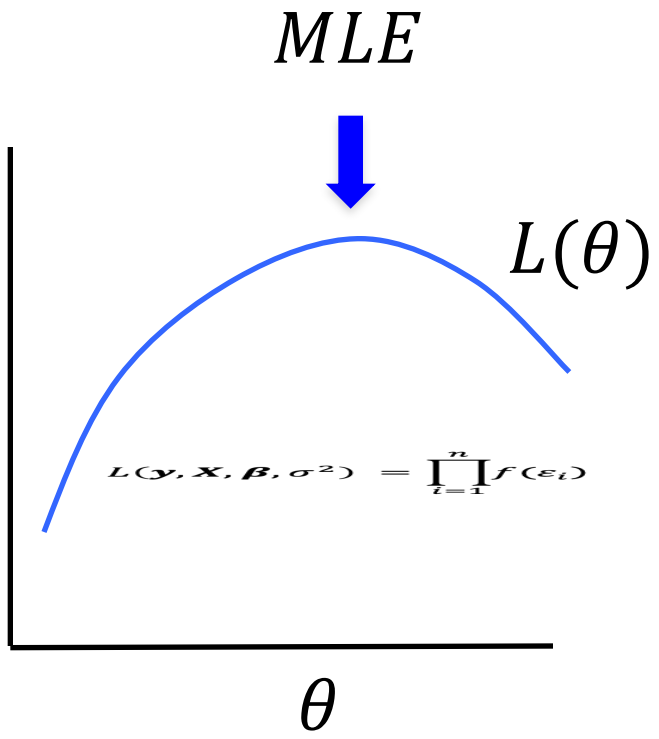


- Sample the predictor space and draw the distributions

Maximum Likelihood Estimation (MLE)

The MLE is the most likely value of the parameters given the data

- MLE uses:
  - the data information (**y**, **X**),
  - the model formulation (***y*** = ***Xβ*** + ***ε)***
  - a probability density function $f(e_i)$ (e.g. Normal, Binomial, Poisson)
- Then it constructs a product function for that data called a likelihood function L(θ), where θ are the parameters to be solved (e.g. β_0, β_1, σ)
- Then to solve for the L(θ), it looks for the maximum value of L(θ) (the turning point), using derivatives and numeric solutions



Methods so far for model fitting

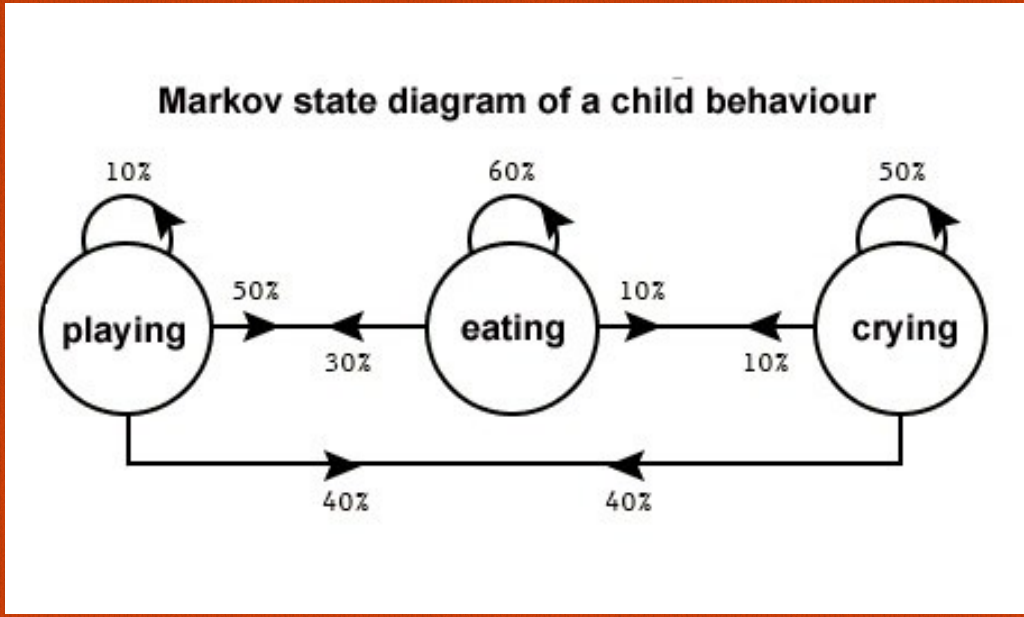
- Least squares (LM models): normal data
- Maximum Likelihood (GLM): not for random effect
- REML ()
- Bootstrapping (restricted to observed data)
- GLS: normal data

How to fit a Bayesian model

- Approximate bayesian computation (only keep the priors that generate the data) Slow!
- Faster methods:
    - instead of throwing away a lot of simulations, it calculates the probability (likelihood) of generating the data with that prior.
    - Sampling the parameter space smarter (instead of just exhaustive search through all prior, MCMC is smarter, Ridges!)
    - The faster methods just try to do a close-enough job as approximate bayesian computation, but faster.
- Key: The rule for deciding which parameters in the prior distribution to keep in the posterior

What is MCMC any way?

Markov chain + **Monte Carlo**

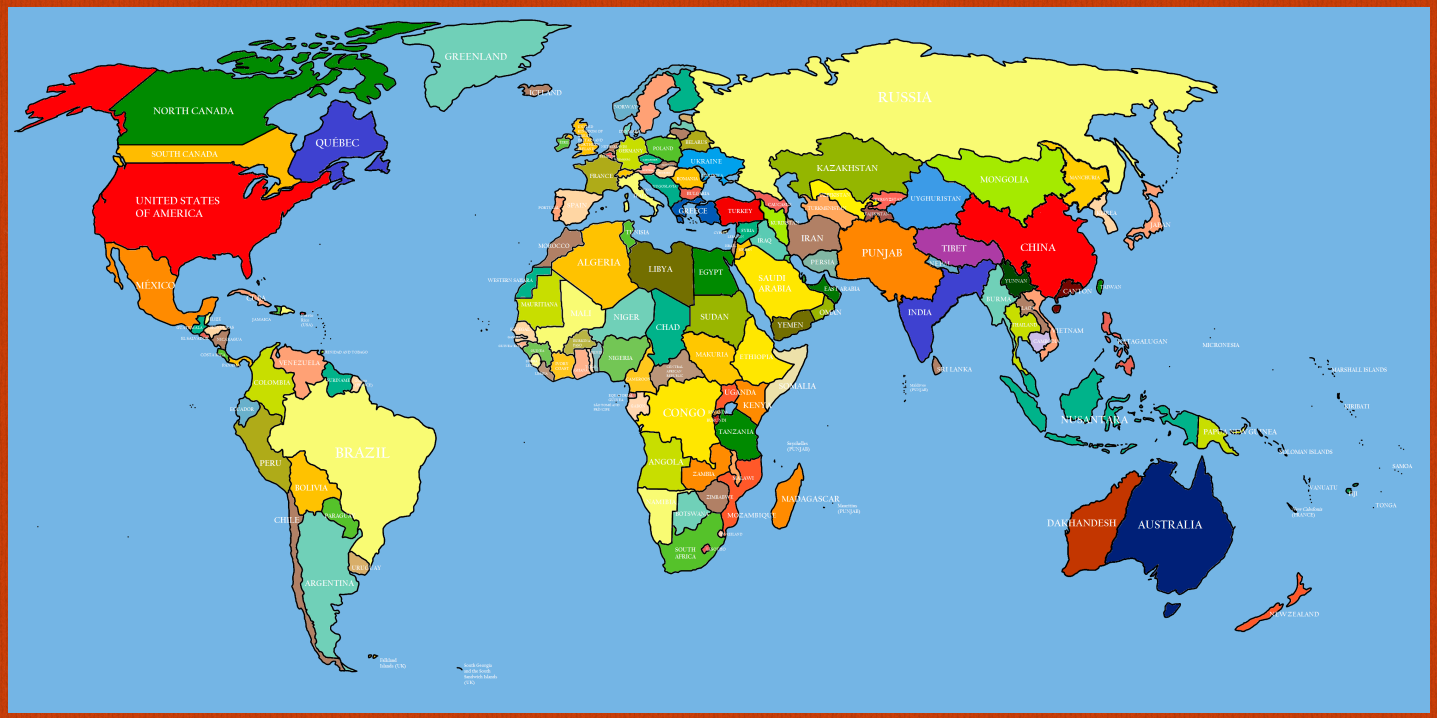


Markov Chain

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

Monte Carlo

computational algorithms that rely on repeated **random sampling** to obtain numerical results

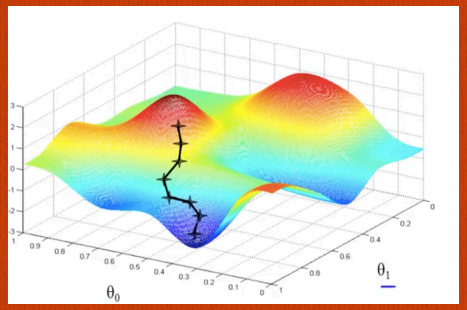


% of land on earth

Markov chain + Monte Carlo

computational algorithms that does repeated sampling based on the previous sampling with some randomness

1. Initial draw from prior (v1)
2. Take a step (s ~ N(0, sd) and propose another position in parameter space to add to the posterior (v2 = v1+ s)
3. Is my data more likely with v1 or v2?
    A. More likely with v2 (L (data | v2) > L (data | v1)):
        - add it to posterior, v2 become the new v1
    B. Less likely with v2 (L (data | v2) < L (data | v1)):
        - draw a random probability p from uniform(0,1)
        - If L(data | v2)/L (data | v1) > p, same as A.
        - If L(data | v2)/L (data | v1) < p, make a copy of v1 in the posterior.



Markov Chain Monte Carlo

Metropolis algorithm

```
# Xiaobai is 14kg, what is the mean of the weight of 2.5-year-old?
samples<-numeric(100) #save posterior
### Initial guess: 10kg
samples[1]<-10
for(i in 2:100){
   step = 0.5
   proposal<- samples[i-1]+ rnorm(1, 0, 0.2)
   # assume sd = 2 (we can estimate it while fixing the mean)
   if ((dnorm(14,proposal, 2)/dnorm(14, samples[i-1], 2))>runif(1))
      samples[i]<-proposal
   else (samples[i]<-samples[i-1])
}
plot(samples,type = “b")
```

Talk is cheap, show me the code

MCMC

- The reason that Bayesian analysis became popular again.
- Every step counts!
- Different tastes
    - Metropolis-hastings
    - Gibbs sampling
    - Hit-n-Run, the T-walk, particle monte carlo, etc.
    - One that Stan uses: Hamiltonian Monte Carlo, No U Turns (NUTS)

“Let’s take a break.”

– And Dance!

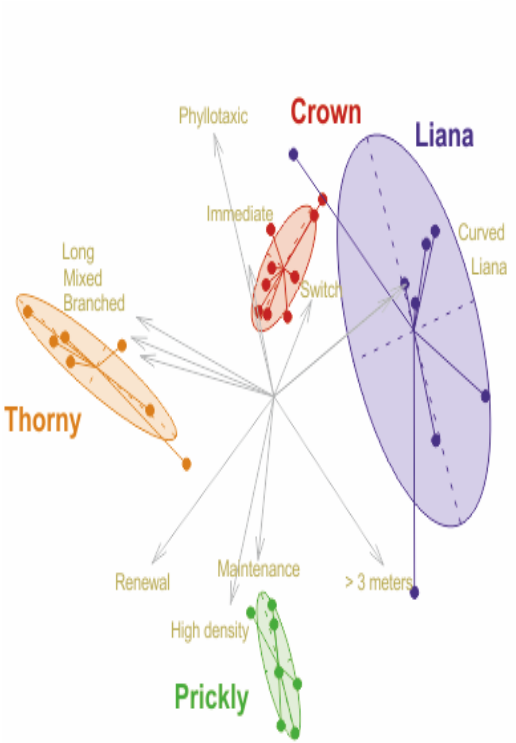
The brm() glmm function

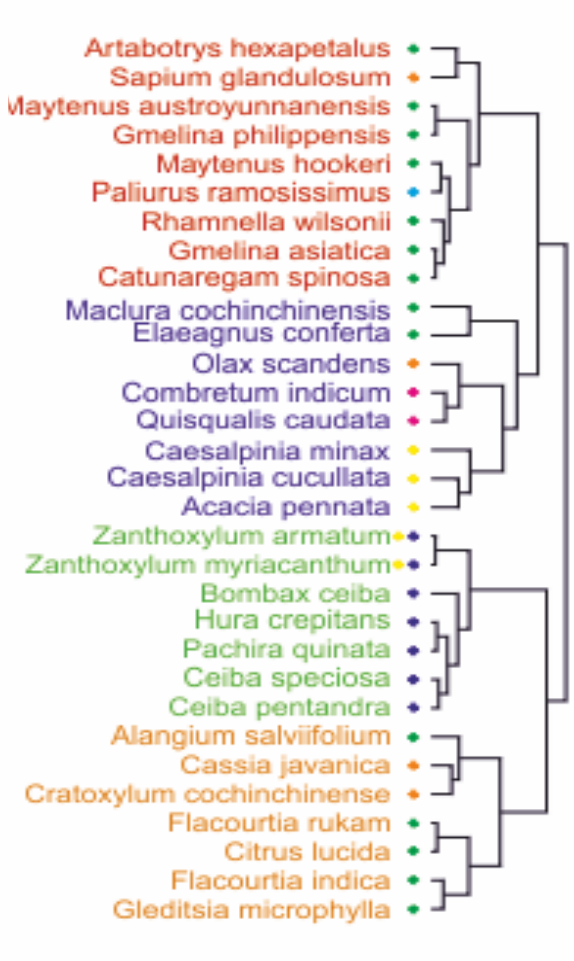
- ```
  mod <- brm(Dlog~x1+(1|phy)+(1|Sp), cov_ranef =
  list(phy = phylo_cor),data = bark2, family =
  gaussian(), sample_prior = TRUE,
      iter = 20000,warmup = 10000, chains = 4, cores =
4, thin = 10, save_all_pars = TRUE,seed=T,
control=list(adapt_delta=0.99))
  ```
- Priors can be “informative” or “non-informative:”
- These will be discussed in the next lecture
- For now we will accept the defaults in the brm() function.

1. Indicate if draws from priors should be drawn additionally to the posterior draws. Options are `"no"` (the default), `"yes"`, and `"only"`.
2. Among others, these draws can be used to calculate Bayes factors for point hypotheses via `hypothesis`. (Perform non-linear hypothesis testing for all model parameters, see code)
3. Please note that improper priors are not sampled, including the default improper priors used by `brm`.
4. If `sample_prior` is set to `"only"`, draws are drawn solely from the priors ignoring the likelihood, which allows among others to generate draws from the prior predictive distribution. In this case, all parameters must have proper priors.

ppcheck: Prior/Posterior Predictive Checks
  ```

Practice: ?set_prior {brms}

	Default	Proper	Example
“fixed” effects (continuous/ categorical)	improper flat prior	normal or t	"normal(0,5)", class = "b", coef = “x1") "student_t(10, 0, 1)", class = "b", coef = "x2")
sd of ‘random' effects			

The brm() glmm function

- ```
  mod <- brm(Dlog~x1+(1|phy)+(1|Sp), cov_ranef =
  list(phy = phylo_cor),data = bark2, family =
  gaussian(), sample_prior = TRUE,
      iter = 20000,warmup = 10000, chains = 4, cores =
  4, thin = 10, save_all_pars = TRUE,seed=T,
  control=list(adapt_delta=0.99))
  ```
- Priors can be “informative” or “non-informative:”
- These will be discussed in the next lecture
- For now we will accept the defaults in the brm() function.

1. Used in the No-U-Turn Sampler (NUTS), a variant of Hamiltonian Monte Carlo.
2. Is the target average proposal acceptance probability during Stan's adaptation period.
3. The default value of `adapt_delta` is 0.95, except when the prior for the regression coefficients is `R2`, `hs`, or `hs_plus`, in which case the default is 0.99.
4. The higher, the more conservative, the slower sampling speeds, and more robust to posterior distributionswith high curvature.
  ```

Spiny trunk data





- Different species with spiny trunks cluster out.
- Do they have different nutritional value and defence efficiency?

Code 10.1
Linear model with species repeats

Diagnostics

R-hat

- R-hat (sometimes also Rhat) convergence diagnostic compares the between- and within-chain estimates for model parameters and other univariate quantities of interest.
- If chains have not mixed well (so that between- and within-chain estimates don't agree), R-hat is larger than 1.
- We recommend running at least four chains by default and in general only fully trust the sample if R-hat is less than 1.01.
- In early workflow, R-hat below 1.1 is often sufficient.

Bulk and Tail ESS

- The effective sample size (ESS) of a quantity of interest captures how many independent draws contain the same amount of information as the dependent sample obtained by the MCMC algorithm.
- The higher the ESS the better.
- Stan uses R-hat adjustment to use the within- and between-chain information in computing the ESS.
- For example, in case of multimodal distributions with well-separated modes, this leads to an ESS estimate that is close to the number of distinct modes that are found.

pp_check: Graphical posterior predictive checking

Comparing observed data to simulated data from the posterior (or prior) predictive distribution.

If a model is a good fit, we can use it to generate data that looks a lot like the data we observed.

For each draw of the parameters from the posterior distribution θ, we generate an entire vector of outcomes (**Yrep**). The result is an *S x N* matrix of simulations, where *S* is the the size of the posterior sample (number of draws from the posterior distribution) and *N* is the number of data points in *y*. That is, each row of the matrix is an individual "replicated" dataset of *N* observations.

This is to update our beliefs about the unknown parameters θ in the model by how well it matches with the observed data.

hypothesis

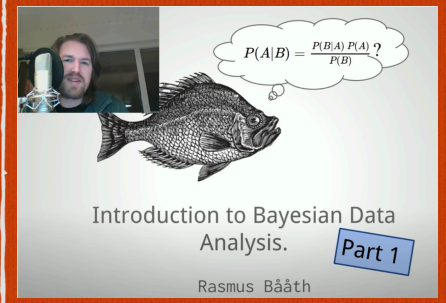
`hypothesis` computes an evidence ratio (`Evid.Ratio`) for each hypothesis.

For a one-sided hypothesis, this is just the posterior probability (`Post.Prob`) under the hypothesis against its alternative. E.g. when the hypothesis is `b1 > 0`, the evidence ratio is the ratio of the posterior probability of `b1 > 0` and the posterior probability of `b1 < 0`. Values greater than one indicate that the evidence in favor of `b1 > 0`.

For a two-sided (point) hypothesis, the evidence ratio is a Bayes factor between the hypothesis and its alternative computed via the Savage-Dickey density ratio method. That is the posterior density at the point of interest divided by the prior density at that point. Values greater than one indicate that evidence in favor of the point hypothesis has increased after seeing the data.

When interpreting Bayes factors, make sure that your priors are reasonable and carefully chosen, as the result will depend heavily on the priors. In particular, avoid using default priors.

Acknowledgement



混合效应模型的贝叶斯实现