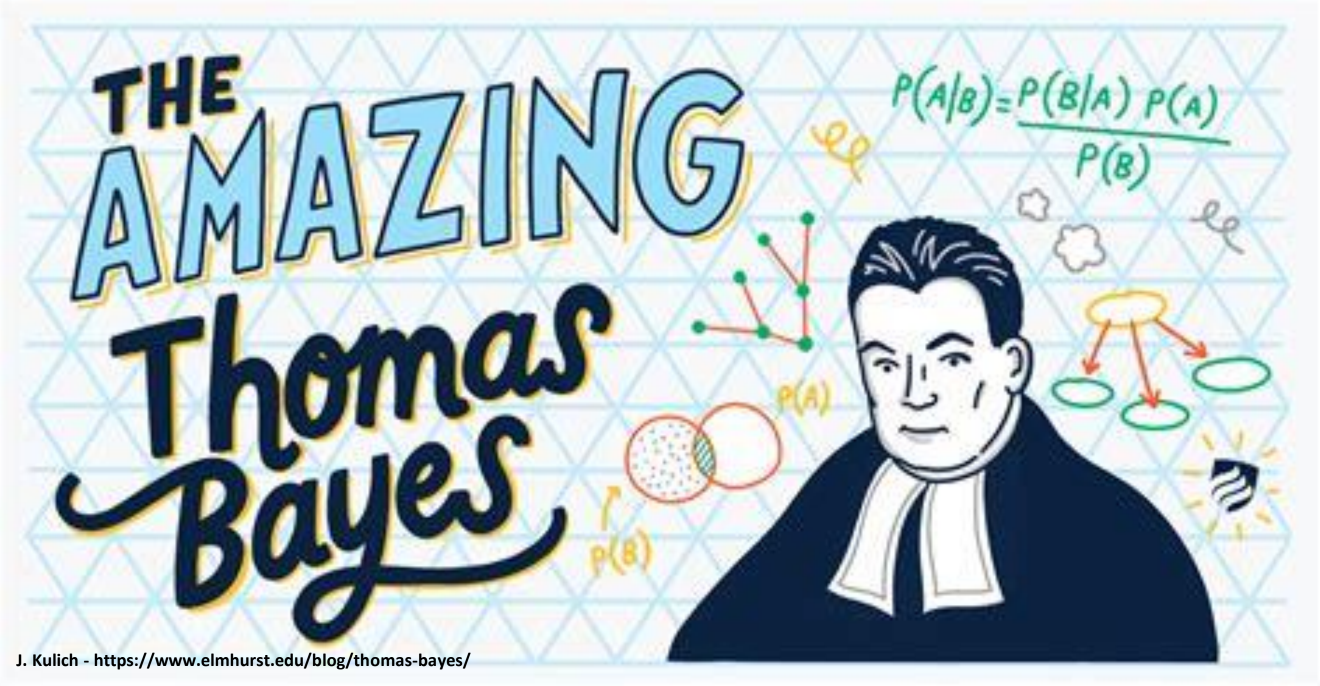
Bayesian stats with `Stan` and `brms()`

Béatrice Capolla & Florent Déry



J. Kulich - https://www.elmhurst.edu/blog/thomas-bayes/

An immersion into british philosophy

1711 — 1776

David Hume: a great skeptic

Criticizes Descartes' rationalism:
our ideas cannot be confirmed by immediate perceptions

Hume's fork:

Demonstrative statement – irrefutable (a priori knowledge), 2 + 2 = 4

Probable statement – refutable , needs empirical proof

Georgios Magkakis.

- A Treatise of Human Nature: Being an Attempt to introduce the experimental Method of Reasoning into Moral Subjects (1739)
- Enquiries concerning Human Understanding (1748)

Concludes that belief is at the center of rationalism, not reason

Concludes that reports of miracles change nothing regarding our understanding of human existence → Clergy opposes

Thomas Bayes' theorem might be seen as a rebuttal of David Hume's work

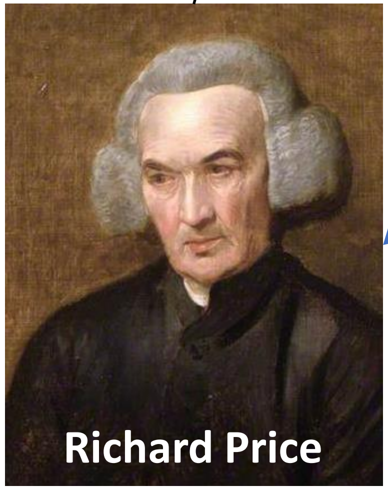
Thomas Bayes :
Highly skilled amateur mathematician & clergyman.
Started working on his theorem shortly after Hume's conclusion on miracles

1702 — 1761

1723 — 1791

An Essay towards solving a Problem in the Doctrine of Chances (1763)

Compiles and complete abandoned notes and essay



Richard Price :
Clergyman and pioneer insurance statistician.

Completed and published Bayes works after his death

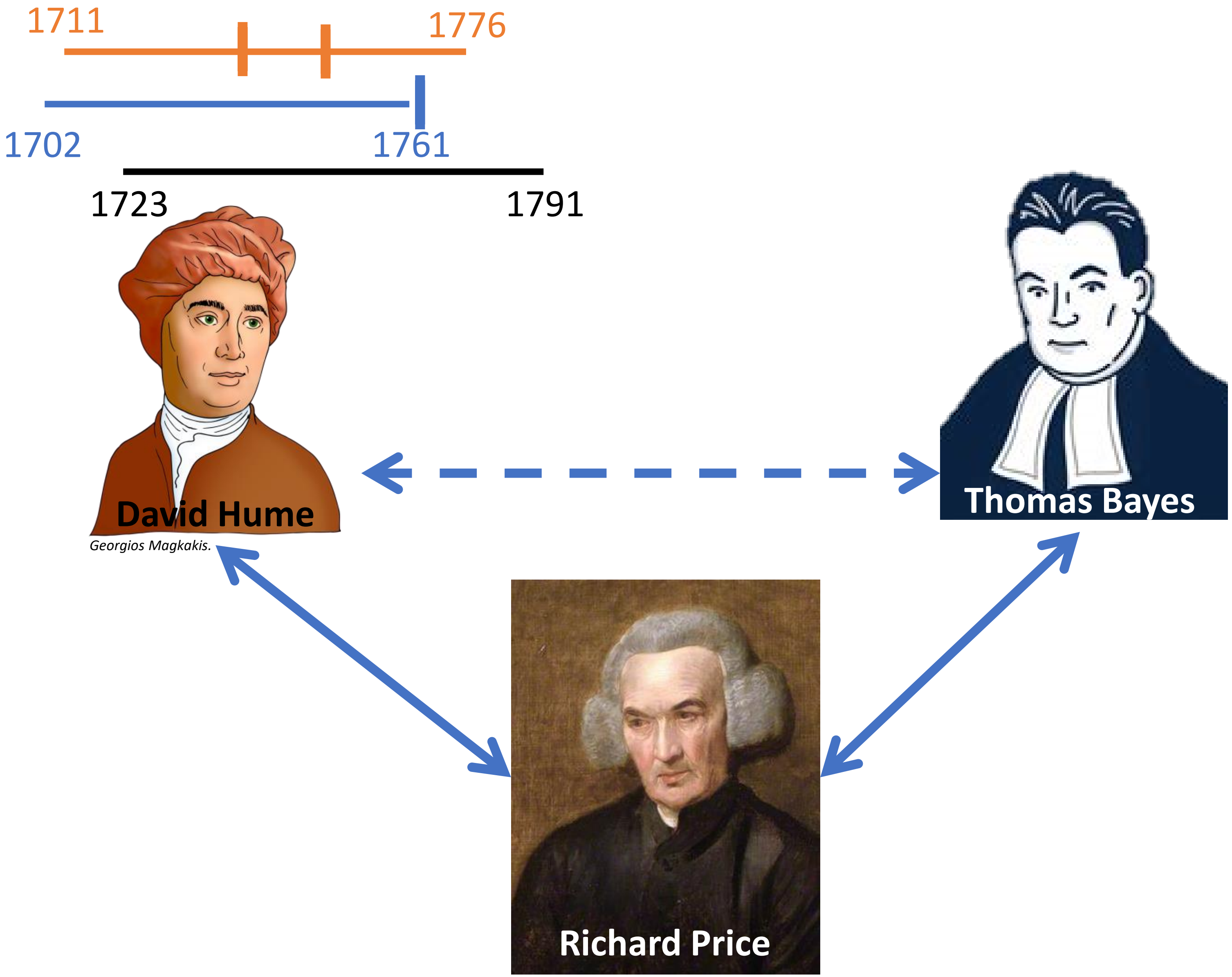
Thomas Bayes' theorem might be seen as a rebuttal of David Hume's work

Bayes theorem combines
Hume's fork:

Demonstrative statement
–
irrefutable
(a priori knowledge)

Probable statement
–
refutable ,
needs
empirical proof

Price uses **Bayes' theorem** to explain miracles (improbable events)



https://mindmatters.ai/2020/12/how-bayes-math-rule-can-counter-unreasonable-skepticism/

Did you say probable(-ility)?

- **Conditional probability:** If y and x are two random variables

conditional probability = $p(y|x)$;
the probability of y for a given value of x (or **the probability of y knowing x**).

- **Joint probability**: probability of obtaining both a certain value of x and a certain value of y, noted $p(x,y)$, can be calculated in two ways:

$$p(x,y) = p(x)p(y|x) = p(y)p(x|y)$$

probability of getting x *
probability of getting y knowing x

probability of getting y *
probability of getting x knowing y

- **Marginal probability**: probability of a variable y, $p(y)$, is its probability if we ignore the value of the other variables. If we do not know $p(y)$ directly, but we know $p(y,x)$ for each possible value of another variable x, then $p(y)$ corresponds to the sum of the joint probabilities of x and y for each value of x.

$$p(y) = \sum_x p(y,x) = \sum_x p(y|x)p(x)$$

Bayes' theorem

Frequentist approach !

Likelihood function

Prior distribution

Posterior distribution

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}$$

Probability of data

Marginal probability

constant often omitted as it does not change the outcomes

Bayes' theorem

What we know
A Priori

The model

Parameters
estimates

$$p(x|Data) = \frac{p(x)p(Data|x)}{p(Data)}$$

Probability
of data

Bayes' theorem

Joint probability can becomes Bayes theorem
by splitting the two right parts by $p(y)$;

$$p(x,y) = p(x)p(y|x) = p(y)p(x|y)$$

becomes

Prior distribution

Posterior distribution

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}$$

Likelihood function (model)

Probability of data

We can then calculate the probability distribution of x conditional of y if we know:

- the probability distribution of y conditional of x, and;
- the marginal probability distribution of x.

As for the denominator $p(y)$, this can be obtained by taking the sum (or the integral) of $p(x)p(y|x)$ over the set of possible values of x.

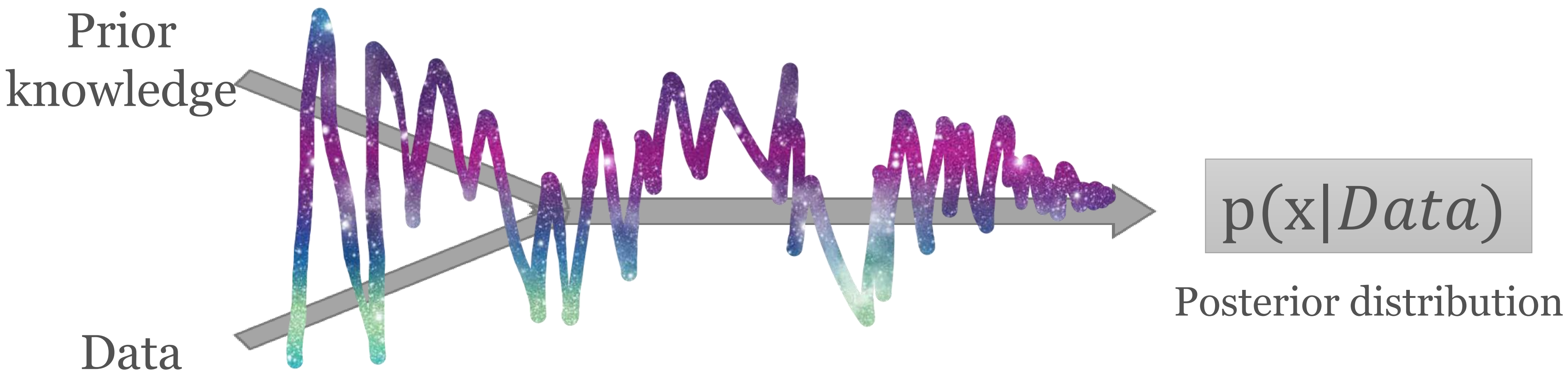
So, how do we solve Bayes' theorem?

Bayes theorem: can be hand solved for simple cases, but not for most of modern day use. --> Need a powerful computer.

Instead of resolving complex integrals, it is estimated by simulating many, many, many posterior distributions

Widely used algorithms:

`WinBUGS, BUGS, JAGS, STAN, Nimble`


Prior knowledge
Data
p(x|Data)
Posterior distribution

Frequentist

Parameter is a known value for a given population. The sample serves to estimate this parameter.

Sample is one of many sample possibles from a population.

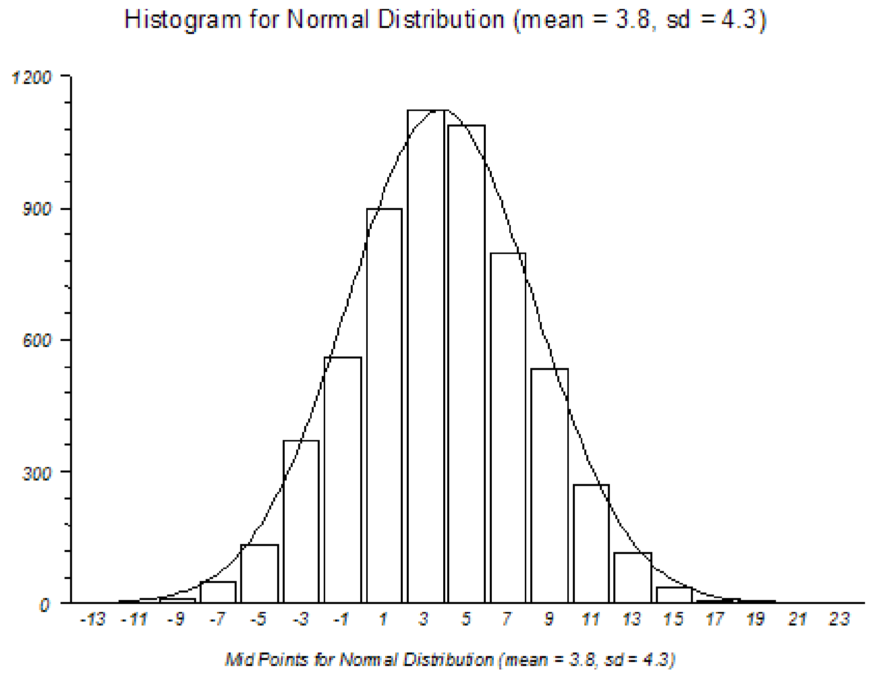
95% confidence intervals: 95% of possible samples of x would produce an interval containing the said value.

Bayesian

Parameter is a random value drawn from a distribution

Sample represents truth (what we actually know)

95% confidence intervals: Probability to observe the parameter within the interval's boundary. → credible interval.



Stan… who?

INSTALLATION DOCUMENTATION COMMUNITY ABOUT US YOUR SUPPORT SEARCH

RStan

the R interface to Stan

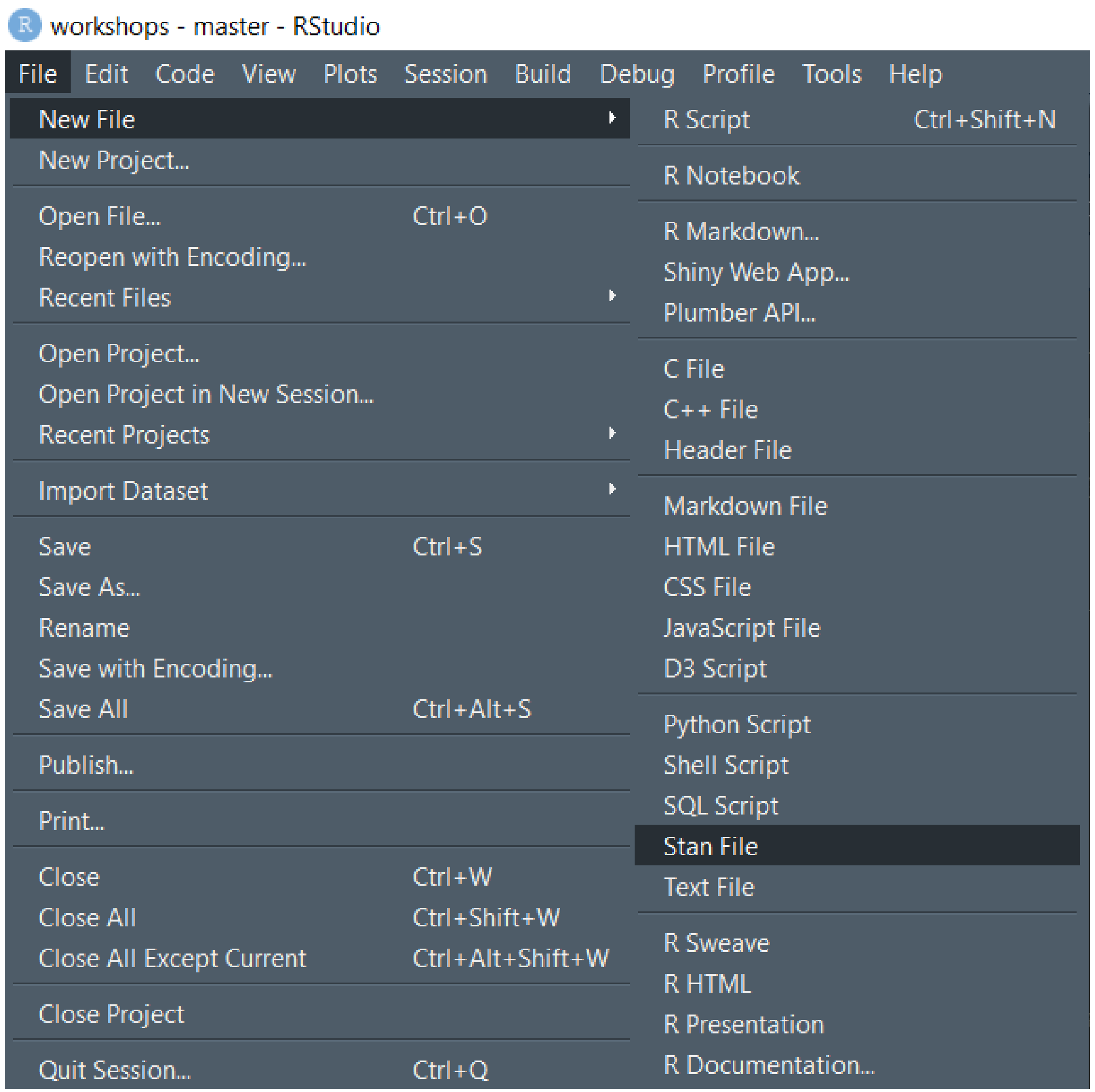
Download and Get Started

Instructions for downloading, installing, and getting started with RStan on all platforms.

- RStan Quick Start Guide (GitHub)

https://mc-stan.org/users/interfaces/rstan

To open a new Stan file



Empty Stan file

```
//
// This Stan program defines a simple model, with a
// vector of values 'y' modeled as normally distributed
// with mean 'mu' and standard deviation 'sigma'.
//
// Learn more about model development with Stan at:
//
//    http://mc-stan.org/users/interfaces/rstan.html
//    https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started
//

// The input data is a vector 'y' of length 'N'.
data {
  int<lower=0> N;
  vector[N] y;
}

// The parameters accepted by the model. Our model
// accepts two parameters 'mu' and 'sigma'.
parameters {
  real mu;
  real<lower=0> sigma;
}

// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  y ~ normal(mu, sigma);
}

```

Use two forward slashes instead of a pound (#) when you want to write text and not code

Where to click if you have trouble coding in Stan

Define your data

Define your parameters

Define your model

Our data : BTdata.txt

Tarsus length

Bird ID

One measure/ID

Mom ID

Bird sex

	tarsus	back	animal	dam	fosternest	hatchdate	sex
1	-1.89229718	1.146421150	R187142	R187557	F2102	-0.68740208	Fem
2	1.13610981	-0.759652092	R187154	R187559	F1902	-0.68740208	Male
3	0.98468946	0.144937259	R187341	R187568	A602	-0.42798142	Male
4	0.37900806	0.255584701	R046169	R187518	A1302	-1.46566405	Male
5	-0.07525299	-0.300699223	R046161	R187528	A2602	-1.46566405	Fem
6	-1.13519543	1.557721860	R187409	R187945	C2302	0.35028055	Fem
7	-1.13519543	-0.426824377	R187507	Fem3	C1902	-0.42798142	Male
8	1.89321156	-1.340762577	R187028	R187030	C1302	-0.94682274	Fem
9	-0.37809369	0.067481541	R046128	R187517	C602	-1.98450537	Fem

Stan hates factor… so make it integer!

```r
# Convert factor to integer ####
library(dplyr)

BTdata$sex_no <- as.integer((BTdata$sex))

BTdata$dam_no <- as.integer((BTdata$dam))
dam_ID <- distinct(BTdata, dam_no, dam) %>%
  arrange(dam_no)

```

Creates a DF indicating which dam_no corresponds to which dam ID

	tarsus	back	animal	dam	fosternest	hatchdate	sex	sex_no	dam_no
1	-1.89229718	1.146421150	R187142	R187557	F2102	-0.68740208	Fem	1	56
2	1.13610981	-0.759652092	R187154	R187559	F1902	-0.68740208	Male	2	57
3	0.98468946	0.144937259	R187341	R187568	A602	-0.42798142	Male	2	61
4	0.37900806	0.255584701	R046169	R187518	A1302	-1.46566405	Male	2	38
5	-0.07525299	-0.300699223	R046161	R187528	A2602	-1.46566405	Fem	1	43
6	-1.13519543	1.557721860	R187409	R187945	C2302	0.35028055	Fem	1	94
7	-1.13519543	-0.426824377	R187507	Fem3	C1902	-0.42798142	Male	2	3
8	1.89321156	-1.340762577	R187028	R187030	C1302	-0.94682274	Fem	1	23
9	-0.37809369	0.067481541	R046128	R187517	C602	-1.98450537	Fem	1	37

Our model

$$\mu_{ij} = \beta_{0j} + \beta_i x_{ij} + \varepsilon_{0ij}$$

Tarsus length

Intercept

Sex

Error

$i = observation\ number$

$j = group\ (mother\ ID)$

$\beta_{0j} = \beta_0 + \mu_{0j}$

$\mu_{0j} \sim N(0, \sigma^2_{\mu_0})$

$\varepsilon_{0ij} \sim N(0, \sigma^2_{e_0})$

Coding data in Stan

```stan
// The input data is a vector 'y' of length 'N'.
data {
  int<lower = 1> n_obs; // indicates my number of observation (length).

  int<lower = 1, upper = 3> sex_no [n_obs]; // a table containing for each observation (length) the corresponding sex. Female = 1, Male = 2,
                                            // Unknown = 3.

  vector[n_obs] tarsus; // The vector tarsus has a length of n_obs.

  int<lower = 1> n_dam; // indicates the number of dam (clusters) in the data frame. We could look for that information in the environement, but we are way too lazy for that.

  int<lower = 1, upper = n_dam> dam_no[n_obs]; // a table containing for each observation (length) the corresponding dam number.
}
```

Use chevrons to indicate boundaries!

Use brackets to indicate the length of your data!

Coding parameters in Stan

```
// The parameters accepted by the model. Our model
// accepts two parameters 'mu' and 'sigma'.
parameters {
  real b_0; // Population intercept.
  real u_dam[n_dam]; // Random effects.
  real<lower = 0> sigma_u_dam; // Standard deviation of random effects.
  real b_sex; // Population slope
  real<lower = 0> sigma; // Population standard deviation.
}

transformed parameters {
  real b_dam[n_dam]; // Varying intercepts (one per cluster)
  real mu[n_obs]; // Individual mean

  for (j in 1:n_dam){
    b_dam[n_dam] = b_0 + u_dam[n_dam];
  } // Varying intercepts definition

    for (i in 1:n_obs){
    mu[i] = b_dam[dam_no[i]] + b_sex * sex_no[i];
    }

  }

```

Coding models in Stan

intro_to_bayesian.R | bayesian_model.stan* | BTdata

Check on Save | Check

```
// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  b_0 ~ std_normal();
  u_dam ~ std_normal();
  sigma_u_dam ~ std_normal();
  b_sex ~ std_normal();
  sigma ~ std_normal();
  u_dam ~ normal(0, sigma_u_dam); // Random effects distribution.

  for (i in 1:n_obs){
    tarsus[i] ~ normal(mu[i], sigma); //
}

}

```

Run your Stan model in R