



# Bayesian Workshop: How to use Bayesian methods in Pumas

Jose Storopoli and Mohamed Tarek {jose.storopoli,mohamed}@pumas.ai  
PumasAI

# Outline

- 1. Pumas
- 2. Bayesian Statistics
- 3. Priors
- 4. Pumas Set-up
- 5. Bayesian Pharmacokinetic Modeling
- 6. Bayesian Logistic Regression
- 7. Hierarchical Models
- 8. Bayesian Population Pharmacokinetic Modeling
- 9. Bayesian Population Pharmacodynamic Modeling
- 10. Posterior Queries and Post-processing
- 11. Markov Chain Monte Carlo (MCMC) and Model Metrics
- 12. Marginal MCMC for Hierarchical Models
- 13. Convergence Diagnostics
- 14. Model Comparison

# Pumas

# What is Pumas?

Pumas (**PharmaceUtical Modeling And Simulation**) (Rackauckas et al., 2020) is a suite of tools to perform quantitative analytics of various kinds across the horizontal of pharmaceutical drug development. The purpose of this framework is to bring efficient implementations of all aspects of the analytics in this domain under one cohesive package.

# Pumas Features

Pumas 2.3 currently includes:

- Non-compartmental Analysis
- Specification of Nonlinear Mixed Effects (NLME) Models
- Simulation of NLME model using differential equations or analytical solutions
- Deep control over the differential equation solvers for high efficiency
- Estimation of NLME parameters via Maximum Likelihood, Expectation Maximization and Bayesian methods
- Parallelization capabilities for both simulation and estimation
- Mixed analytical and numerical problems
- Simulation and estimation diagnostics for model post-processing
- Interactive model exploration and diagnostics tools through webapps
- Automated report generation for models and non-compartmental analysis
- Global and local sensitivity analysis routines for multi-scale models
- Bioequivalence analysis
- Optimal design of experiments

# Bayesian Statistics

- 2.1 Recommended References
- 2.2 What is Bayesian Statistics?
- 2.3 Frequentist versus Bayesian
- 2.4 Bayesian Statistics

# Bayesian Statistics - Recommended References

- Gelman et al. (2013b) - Chapter 1: Probability and inference
- McElreath (2020) - Chapter 1: The Golem of Prague
- Gelman, Hill, and Vehtari (2020) - Chapter 3: Some basic methods in mathematics and probability
- Khan and Rue (2021)
- **Probability:**
  - A great textbook - Bertsekas and Tsitsiklis (2008)
  - Also a great textbook (skip the frequentist part)- Dekking et al. (2010)
  - Bayesian point-of-view and also a philosophical approach- Jaynes (2003)
  - Bayesian point-of-view with a simple and playful approach - Kurt (2019)
  - Philosophical approach not so focused on mathematical rigor - Diaconis and Skyrms (2019)

# What is Bayesian Statistics?

Bayesian statistics is a **data analysis approach based on Bayes' theorem** where available knowledge about the parameters of a statistical model is updated with the information of observed data. (Gelman et al., 2013b). Previous knowledge is expressed as a **prior** distribution and combined with the observed data in the form of a **likelihood** function to generate a **posterior** distribution. The posterior can also be used to make predictions about future events.

# What changes from Frequentist Statistics?

- **Domain knowledge:**
  - You can incorporate knowledge and insights from previous studies using prior distributions on parameters
- **Epistemic uncertainty:**
  - You can quantify the epistemic uncertainty in the model parameters' values
  - Model identifiability not necessary
  - Works for small and large sample sizes
  - No Gaussian assumptions
- **Conceptually simpler and more general:**
  - Uses probability theory instead of *ad-hoc* methods
  - No *p*-values, *p*-hacking and *ad-hoc* assumptions in hypothesis tests

# A little bit more formal

- Bayesian Statistics uses probabilistic statements:
  - one or more parameters  $\theta$
  - unobserved data  $\tilde{y}$
- These statements are conditioned on the observed values of  $y$ :
  - $P(\theta | y)$
  - $P(\tilde{y} | y)$
- We also, implicitly, condition on the observed data from any covariate  $x$
- Generally, we are interested in:
  - expected response of a new subject to a drug, e.g.  $E[\hat{y} | y]$
  - the probability of drug effect is higher than zero, e.g.  $P(\theta > 0 | y) \geq 0.95$

# Definition of Bayesian Statistics

## Definition (Bayesian Statistics)

*The use of Bayes theorem as the procedure to **estimate parameters of interest**  $\theta$  or **unobserved data**  $\tilde{y}$ . (Gelman et al., 2013b)*

# Probability Interpretations

- **Objective** - frequency in the long run for an event:

- $P(\text{rain}) = \frac{\text{days that rained}}{\text{total days}}$

- $P(\text{me being elected president}) = 0$  (never occurred)

- **Subjective** - degrees of belief in an event:

- $P(\text{rain}) = \text{degree of belief that will rain}$

- $P(\text{me being elected president}) = 10^{-10}$  (highly unlikely)

# What is Probability?

## Definition (Probability)

We define  $A$  is an event and  $P(A)$  the probability of event  $A$ .  $P(A)$  has to be between 0 and 1, where higher values defines higher probability of  $A$  happening.

$$P(A) \in \mathbb{R}$$

$$P(A) \in [0, 1]$$

$$0 \leq P(A) \leq 1$$

# Probability Axioms<sup>i</sup>

- **Non-negativity:** For every  $A$ :

$$P(A) \geq 0$$

- **Additivity:** For every two *mutually exclusive*  $A$  and  $B$ :

$$P(A \cup B) = P(A) + P(B)$$

- **Normalization:** The probability of all possible *mutually exclusive* events  $A_1, A_2, \dots$  must sum up to 1:

$$\sum_{n \in \mathbb{N}} P(A_n) = 1$$



---

<sup>i</sup>Kolmogorov (1933)

# Sample Space<sup>ii</sup>

- Discrete

$$\Theta = \{1, 2, \dots\}$$

- Continuous

$$\Theta \in (-\infty, \infty)$$

---

<sup>ii</sup> $\theta$  domain can be general, not restricted to these domains.

# Discrete Sample Space

8 planets in our solar system:

- Mercury - ♀
- Venus - ♀
- Earth - ♂
- Mars ♂
- Jupiter - ♁
- Saturn ♃
- Uranus - ♂
- Neptune ♀

# Discrete Sample Space<sup>iii</sup>

The planet has a magnetic field



The planet has moon(s)



The planet has a magnetic field *and* moon(s)



The planet has a magnetic field *or* moon(s)



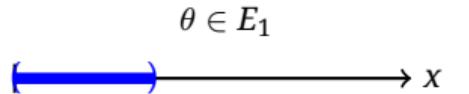
The planet does *not* have a magnetic field



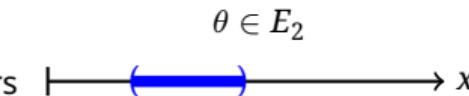
<sup>iii</sup>figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# Continuous Sample Space<sup>iv</sup>

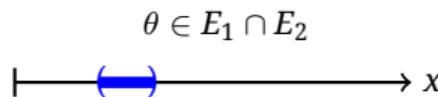
The distance is less than five centimeters



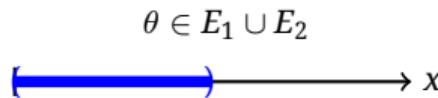
The distance is between three and seven centimeters



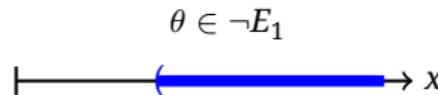
The distance is less than five centimeters  
and between three and seven centimeters



The distance is less than five centimeters  
or between three and seven centimeters



The distance is *not* less than five centimeters



---

<sup>iv</sup>figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# Discrete versus Continuous Parameters

Parameters can be continuous, such as: age, height, weight etc.

All probability rules and axioms are valid also for continuous parameters.

The only thing we have to do is to change all sums  $\sum$  for integrals  $\int$  and some probabilities  $P$  with probability density (probability mass per unit measure)  $p$ , e.g.:

$$p(A) \geq 0$$

$$p(A) \in \mathbb{R}$$

$$\int p(A)dA = 1$$

# Conditional Probability

## Definition (Conditional Probability)

*Probability of an event occurring in case another has occurred or not.*

*The notation we use is  $P(A | B)$ , that read as “the probability of observing A given that we already observed B”.*

$$P(A | B) = \frac{\text{number of elements in } A \text{ and } B}{\text{number of elements in } B}$$

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

*assuming that  $P(B) > 0$ .*

# Caution! Not always $P(A | B) = P(B | A)$

In some cases, we have the symmetry  $P(A | K) = P(K | A)$ , **but not always this is true**<sup>v</sup>

## Example (Pumas in Clinical Research)

- $P(\text{pumas})$ : Probability of some random person using Pumas, something really small, 2 thousand in 8 billion ( $\frac{2}{8 \cdot 10^6}$ )
- $P(\text{clinical research})$ : Probability of some random person does clinical research, 50 thousand in 8 billion ( $\frac{50}{8 \cdot 10^6}$ )
- $P(\text{clinical research} | \text{Pumas})$ : Probability of doing clinical research, given that you are a Pumas user, ( $\frac{9}{10} = 0.9$ )
- $P(\text{Pumas} | \text{clinical research})$ : Probability of being a Pumas user, given that you are a doing clinical research ( $\frac{1}{25}$ )
- **Hence:**  $P(\text{clinical research} | \text{Pumas}) \neq P(\text{Pumas} | \text{clinical research})$

# Joint Probability

## Definition (Joint Probability)

*Probability of two or more events occurring.*

*The notation we use  
is  $P(A, B)$ , that read as “the probability of observing A and also observing B”.*

$P(A, B) = \text{number of elements in } A \text{ or } B$

$P(A, B) = P(A \cup B)$

$P(A, B) = P(B, A)$

# Product Rule of Probability<sup>vi</sup>

## Definition (Product Rule)

*We can decompose a joint probability  $P(A, B)$  into the product of two probabilities:*

$$P(A, B) = P(B, A)$$

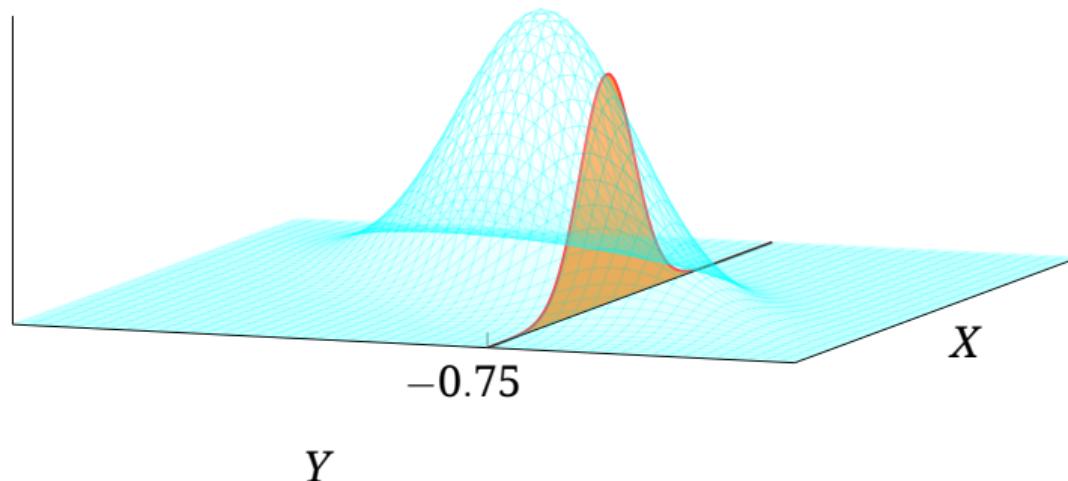
$$P(A) \cdot P(B | A) = P(B) \cdot P(A | B)$$

---

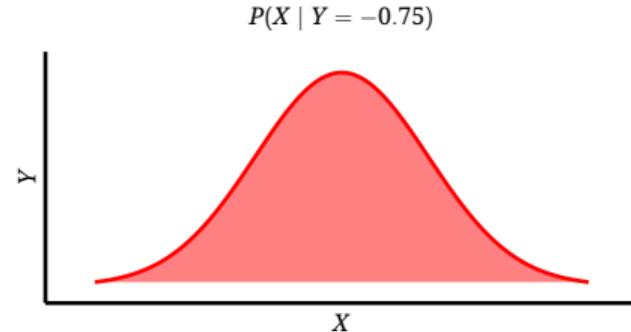
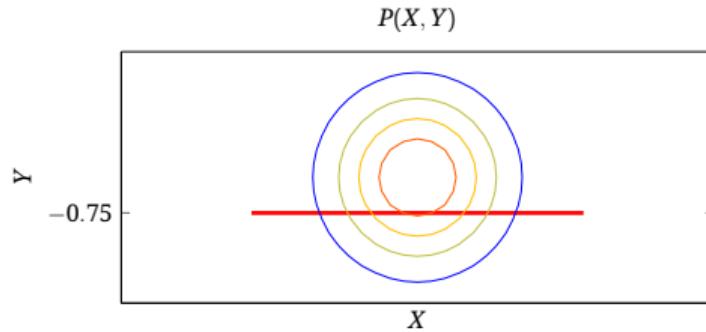
<sup>vi</sup>also called the Product Rule of Probability.

# Visualization of Joint Probability versus Conditional Probability

$P(X, Y)$  versus  $P(X \mid Y = -0.75)$



# Visualization of Joint Probability versus Conditional Probability



# Who was Thomas Bayes?

- **Thomas Bayes** (1701 - 1761) was a statistician, philosopher and Presbyterian minister who is known for formulating a specific case of the theorem that bears his name: Bayes' theorem.
- Bayes never published what would become his most famous accomplishment; his notes were edited and published posthumously by his friend **Richard Price**.
- The theorem official name is **Bayes-Price-Laplace**, because **Bayes** was the first to discover, **Price** got his notes, transcribed into mathematical notation, and read to the Royal Society of London, and **Laplace** independently rediscovered the theorem without having previous contact in the end of the XVIII century in France while using probability for statistical inference with census data in the Napoleonic era.



# Bayes Theorem

## Theorem (Bayes)

*Tells us how to “invert” conditional probability:*

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

# Bayes' Theorem Proof

Remember the following probability identity:

$$P(A, B) = P(B, A)$$

$$P(A) \cdot P(B | A) = P(B) \cdot P(A | B)$$

OK, now divide everything by  $P(B)$ :

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

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

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

# Another Probability Textbook Classic<sup>vii</sup>

## Example (Breast Cancer)

How accurate is a **breast cancer** test?

- 1% of women have **breast cancer** (Prevalence)
- 80% of mammograms detect **breast cancer** (True Positive)
- 9.6% of mammograms detect **breast cancer** when there is no incidence (False Positive)

$$P(C | +) = \frac{P(+ | C) \cdot P(C)}{P(+)}$$

$$P(C | +) = \frac{P(+ | C) \cdot P(C)}{P(+ | C) \cdot P(C) + P(+ | \neg C) \cdot P(\neg C)}$$

$$P(C | +) = \frac{0.8 \cdot 0.01}{0.8 \cdot 0.01 + 0.096 \cdot 0.99}$$

$$P(C | +) \approx 0.0776$$

---

<sup>vii</sup> Adapted from: [Yudkowsky - An Intuitive Explanation of Bayes' Theorem](#).

# Why Bayes' Theorem is Important?

Idea (We can Invert the Conditional Probability)

$$P(\text{hypothesis} \mid \text{data}) = \frac{P(\text{data} \mid \text{hypothesis}) \cdot P(\text{hypothesis})}{P(\text{data})}$$

But isn't this the *p*-value? **NO!**

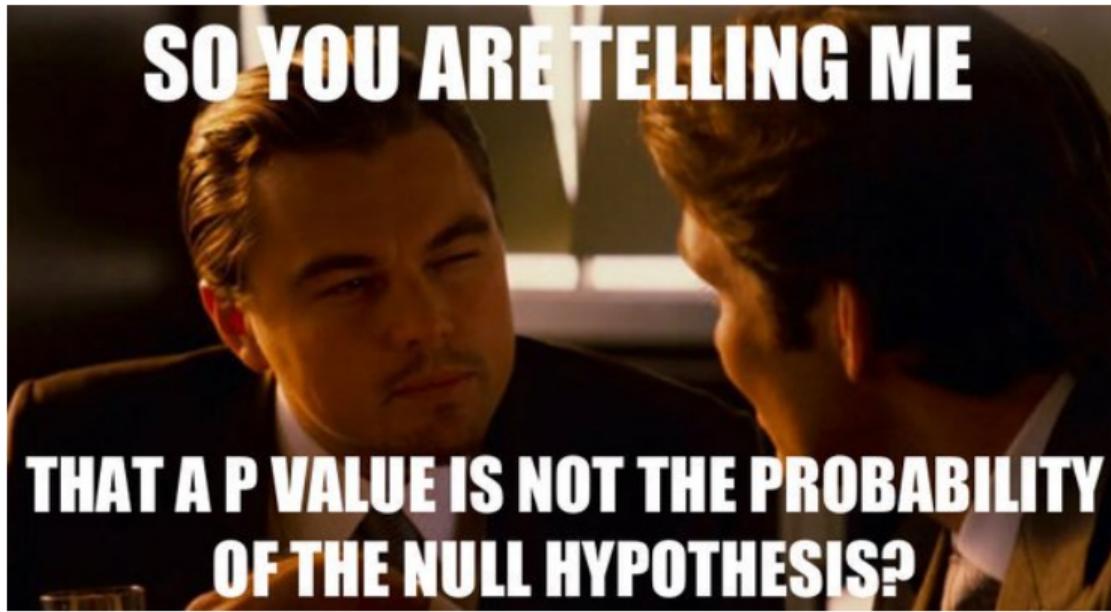
# What are *p*-values?

## Definition (*p*-value)

*p*-value is the probability of obtaining results at least as extreme as the observed, given that the null hypothesis  $H_0$  is true:

$$P(D \mid H_0)$$

# What $p$ -value is **not**!



# What *p*-value is **not**!

- ***p*-value is not the probability of the null hypothesis** - Infamous confusion between  $P(D | H_0)$  and  $P(H_0 | D)$ . To get  $P(H_0 | D)$  you need Bayesian statistics.
- ***p*-value is not the probability of data being generated at random** - **No!** We haven't stated anything about randomness.
- ***p*-value measures the effect size of a statistical test** - Also **no...** *p*-value does not say anything about effect sizes. Just about if the observed data diverge from the expected under the null hypothesis. Besides, *p*-values can be hacked in several ways (Head et al., 2015).

# The relationship between $p$ -value and $H_0$

To find out about any  $p$ -value, **find out what  $H_0$  is behind it**. Its definition will never change, since it is always  $P(D | H_0)$ :

- ***t-test***:  $P(D | \text{the difference between the groups is zero})$
- ***ANOVA***:  $P(D | \text{there is no difference between groups})$
- ***Regression***:  $P(D | \text{coefficient has a null value})$
- ***Shapiro-Wilk***:  
 $P(D | \text{population is distributed as a Normal distribution})$

# What are Confidence Intervals?

## Definition (Confidence Intervals)

*A confidence interval of X% for a parameter is an interval  $(a, b)$  generated by a repeated sampling procedure has probability X% of containing the true value of the parameter, for all possible values of the parameter.*



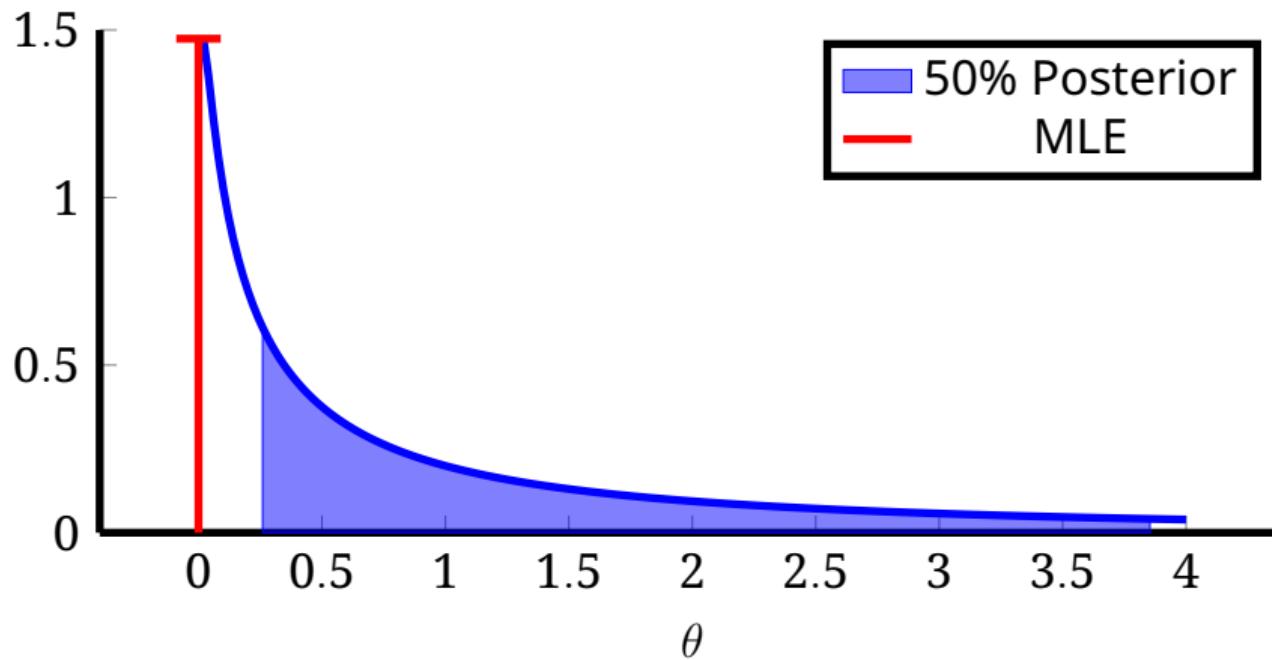
Neyman (1937) (the “father” of confidence intervals)

# What are Confidence Intervals?

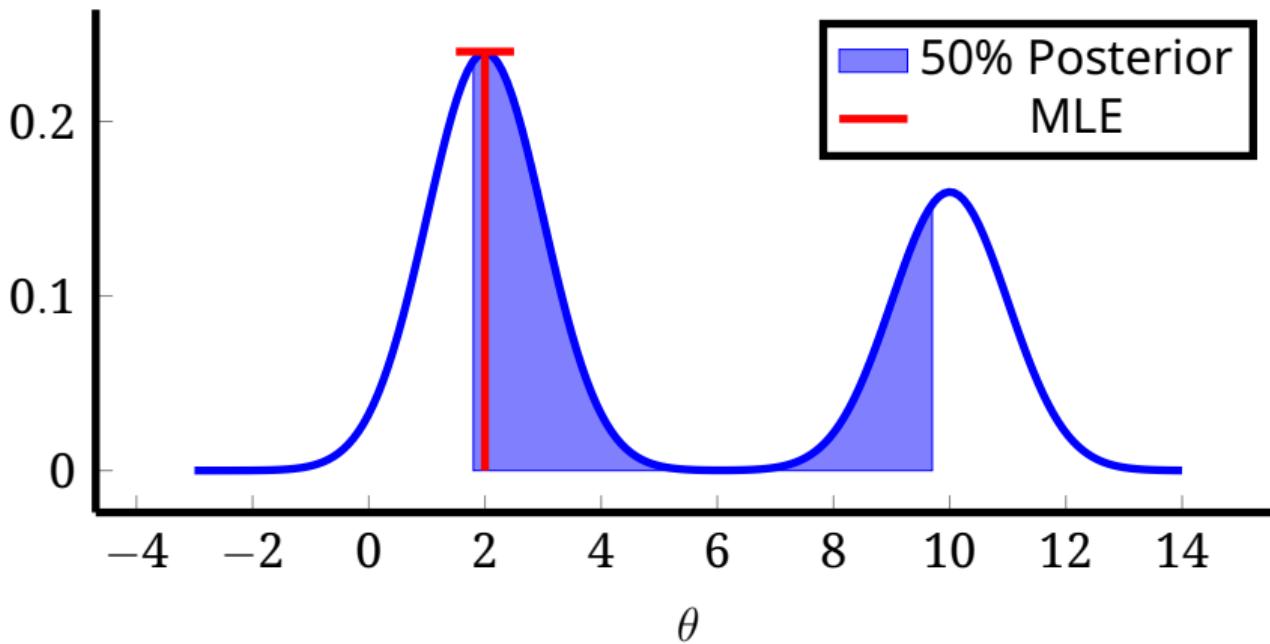
## Example (Confidence Intervals of a Public Policy Analysis)

Say you performed a statistical analysis to compare the efficacy of a public policy between two groups and you obtain a difference between the mean of these groups. You can express this difference as a confidence interval. Often we choose 95% confidence. In other words, 95% is *not* the probability of obtaining data such that the estimate of the true parameter is contained in the interval that we obtained, it is the **probability of obtaining data such that, if we compute another confidence interval in the same way, it contains the true parameter**. The interval that we got in this particular instance is irrelevant and might as well be thrown away.

# Confidence Intervals versus Posterior Intervals



# Confidence Intervals versus Posterior Intervals



# But why I never see stats without *p*-values?

We cannot understand *p*-values if we do not comprehend its origins and historical trajectory. The first mention of *p*-values was made by the statistician Ronald Fischer in 1925 (Fisher, 1925):

*[p-value is a] measure of evidence against the null hypothesis*

- To quantify the strength of the evidence against the null hypothesis, Fisher defended " $p < 0.05$  as the standard level to conclude that there is evidence against the tested hypothesis"
- "We should not be off-track if we draw a conventional line at 0.05"



$$p = 0.06$$

- Since  $p$ -value is a probability, it is also a continuous measure.
- There is no reason for us to differentiate  $p = 0.049$  against  $p = 0.051$ .
- Robert Rosenthal, a psychologist said “surely, God loves the .06 nearly as much as the .05” (Rosnow & Rosenthal, 1989).

# But why I never heard about Bayesian statistics?<sup>viii</sup>

*... it will be sufficient ... to reaffirm my personal conviction ... that the theory of inverse probability is founded upon an error, and must be wholly rejected.*

Fisher (1925)



---

<sup>viii</sup>inverse probability was how Bayes' theorem was called in the beginning of the 20th century

# Inside every non-Bayesian, there is a Bayesian struggling to get out<sup>ix</sup>

- In his final year of life, Fisher published a paper (Fisher, 1962) examining the possibilities of Bayesian methods, but with the prior probabilities being determined experimentally.
- Some authors speculate (Jaynes, 2003) that if Fisher were alive today, he would probably be a Bayesian.



---

<sup>ix</sup>quote from Dennis Lindley

# Bayes' Theorem as an Inference Engine

Now that you know what is probability and Bayes' theorem, I will propose the following:

$$\underbrace{P(\theta | y)}_{\text{Posterior}} = \frac{\overbrace{P(y | \theta) \cdot P(\theta)}^{\text{Likelihood Prior}}}{\underbrace{P(y)}_{\text{Normalizing Constant}}}$$

- $\theta$  – parameter(s) of interest
- $y$  – observed data
- **Prior:** prior probability of the parameter(s) value(s)
- **Likelihood:** probability of the observed data given the parameter(s) value(s)
- **Posterior:** posterior probability of the parameter(s) value(s) after we observed data  $y$
- **Normalizing Constant<sup>x</sup>:**  $P(y)$  gives a measure of the likelihood of the entire model class which is useful when considering multiple models. This probability is transformed and can be interpreted as something that only exists so that the result  $P(y | \theta)P(\theta)$  be constrained between 0 e 1 – a valid probability.

---

<sup>x</sup>sometimes also called *evidence*.

# Bayes' Theorem as an Inference Engine

Bayesian statistics allows us to **quantify directly the uncertainty** related to the value of one or more parameters of our model given the observed data. This is the **main feature** of Bayesian statistics, since we are estimating directly  $P(\theta | y)$  using Bayes' theorem. The resulting estimate is totally intuitive: simply quantifies the uncertainty that we have about the value of one or more parameters given the data, model assumptions (likelihood) and the prior probability of these parameter's values.

# Priors

- 3.1 Recommended References
- 3.2 The subjectivity of the Prior
- 3.3 Types of Priors

# Priors and Posteriors - Recommended References

- Gelman et al. (2013b):
  - Chapter 2: Single-parameter models
  - Chapter 3: Introduction to multiparameter models
- McElreath (2020) - Chapter 4: Geocentric Models
- Gelman, Hill, and Vehtari (2020):
  - Chapter 9, Section 9.3: Prior information and Bayesian synthesis
  - Chapter 9, Section 9.5: Uniform, weakly informative, and informative priors in regression
- van de Schoot et al. (2021)

# Prior Probability

Bayesian statistics is characterized by the use of prior information as the prior probability  $P(\theta)$ , often just prior:

$$\underbrace{P(\theta | y)}_{\text{Posterior}} = \frac{\overbrace{P(y | \theta) \cdot P(\theta)}^{\text{Likelihood Prior}}}{\underbrace{P(y)}_{\text{Normalizing Constant}}}$$

# The subjectivity of the Prior

- Many criticisms to Bayesian statistics are due the subjectivity in eliciting priors probability on certain hypothesis or model parameter's values.
- Subjectivity is something unwanted in the ideal picture of the scientist and the scientific method.

## Counter-arguments:

- Very weak priors avoid subjectivity.
- Anything that involves human action will never be free from subjectivity. We have subjectivity in everything and science is **no** exception.
- Frequentist statistics is also subjective, since there is **A LOT** of subjectivity in choosing the model and target p-value (Jaynes, 2003; van de Schoot et al., 2021)

# Types of Priors

In general, we can have 3 main types of priors in a Bayesian approach  
(Gelman et al., 2013b; McElreath, 2020; van de Schoot et al., 2021):

- **uniform (flat)**: not recommended.
- **weakly informative**: little domain knowledge added.
- **informative**: medium to high domain knowledge.

# Prior Selection

| Support         | Distributions  |
|-----------------|--|
| (0, 1)          | Beta, KSOOneSided, NoncentralBeta, LogitNormal   |
| $\mathcal{R}^+$ | BetaPrime, Chi, Chisq, Erlang, Exponential, FDist, Frechet, <b>Gamma</b> , InverseGamma, InverseGaussian, Kolmogorov, LogNormal, NoncentralChisq, NoncentralF, Rayleigh, Weibull |
| $\mathcal{R}$   | Cauchy, Gumbel, Laplace, Logistic, Normal, NormalCanon, NormalInverseGaussian, PGeneralizedGaussian, TDist   |

# Prior Selection

| Support                 | Distributions   |
|-------------------------|---|
| $\mathcal{R}$ vectors   | MvNormal  |
| $\mathcal{R}^+$ vectors | MvLogNormal   |
| PD mats                 | Wishart, InverseWishart   |
| Corr mats               | LKJ, LKJCholesky  |
| Other                   | Constrained, truncated, LocationScale, Uniform, Arcsine, Biweight, Cosine, Epanechnikov, Semicircle, SymTriangularDist, Triweight, Pareto, GeneralizedPareto, GeneralizedExtremeValue, Levy |

# Prior Selection

Check **similar** models from literature. If you really have to choose a new prior, follow the following process:

- Decide the **support** of the prior. The support of the prior distribution must match the domain of the parameter.
- Decide the **center** of the prior, e.g. mean, median or mode.
- Decide the **strength** of the prior. This is often controlled by a standard deviation or scale parameter in the distribution constructor.
- Decide the **shape** of the probability density function (PDF) of the prior. Left skewed, right skewed, centered, heavy tail, etc.

# Prior Selection

A prior too strong around the wrong parameter values can negatively hurt your study.

It will then require many more observations to infer a posterior distribution centered around the true data generating parameter values.

A prior too weak can often hinder the performance of the inference.

# Priors for Covariance Matrices

We can specify a prior for a covariance matrix  $\Sigma$ .

For computational efficiency, we can make the covariance matrix  $\Sigma$  into a correlation matrix. Every covariance matrix can be decomposed into:

$$\Sigma = \text{diag}_{\text{matrix}}(\tau) \cdot \mathbf{C} \cdot \text{diag}_{\text{matrix}}(\tau)$$

where  $\mathbf{C}$  is a correlation matrix with 1s in the diagonal and the off-diagonal elements between -1 and 1  $\rho \in (-1, 1)$ .  $\tau$  is a vector composed of the variables' variances from  $\Sigma$  (is is the  $\Sigma$ 's diagonal).

# Priors for Covariance Matrices

Additionally, the correlation matrix  $\mathbf{C}$  can be decomposed once more for greater computational efficiency. Since all correlations matrices are symmetric and positive definite (all of its eigenvalues are real numbers  $\mathbb{R}$  and positive  $> 0$ ), we can use the [Cholesky Decomposition](#) to decompose it into a triangular matrix (which is much more computational efficient to handle):

$$\mathbf{C} = \mathbf{L}_\mathbf{C} \mathbf{L}_\mathbf{C}^T$$

where  $\mathbf{L}_\mathbf{C}$  is a lower-triangular matrix.

What we are missing is to define a prior for the correlation matrix  $\mathbf{C}$ .

## Simulating from the prior

It can be useful sometimes to simulate from the model propagating the uncertainty from the prior distributions to the model predictions.

Predictions from the model can further be plotted against the observations to get a feel for the behavior of the prior model.

# Pumas Set-up

# Setting-up Pumas

Now let's learn how to set-up and use Pumas.

# Bayesian Pharmacokinetic Modeling

- 5.1 Recommended References
- 5.2 Pharmacokinetics
- 5.3 Compartment Models
- 5.4 Bayesian PK Models
- 5.5 Bayesian PK Models in Pumas

# Bayesian Pharmacokinetic Modeling - Recommended References

- Gabrielsson et al. (2006):
  - Chapter 1: General Principles
  - Chapter 2: Pharmacokinetic Concepts
- Owen and Fiedler-Kelly (2014):
  - Chapter 10: PK/PD Models
- Bonate (2011):
  - Chapter 10: Bayesian Modeling regression
- Margossian et al. (2022)

# Pharmacokinetics

## Definition (Pharmacokinetics)

*Pharmacokinetics is the study of the time course of drug concentration in different body spaces such as plasma, blood, urine, cerebrospinal fluid, and tissues, and the relationship between concentration and the time course of drug action such as onset, intensity, and duration of action.*

*Gabrielsson et al. (2006, p. 13)*

# Pharmacokinetics

Pharmacokinetics is generally represented as "**PK compartments**" in a model.

They can be either:

- **Pharmacokinetic** (PK) models with only PK compartments
- **Pharmacokinetic-Pharmacodynamic<sup>xi</sup>** (PKPD) models with both PK compartments and PD compartments

---

<sup>xi</sup>more about these later on.

# 1-Compartment Model

$$\text{Central}' = -\frac{CL}{V_c} \cdot \text{Central}$$

where:

- $CL$  is elimination clearance from the Central compartment
- $V_c$  is volume of the Central compartment

# 1-compartment Model with First-Order Absorption

$$\text{Depot}' = -Ka \cdot \text{Depot}$$

$$\text{Central}' = Ka \cdot \text{Depot} - \frac{CL}{V_C} \cdot \text{Central}$$

where:

- $CL$  is elimination clearance from the Central compartment
- $V_C$  is volume of the Central compartment
- $Ka$  is absorption rate constant

## 2-Compartment Model

$$\text{Central}' = -\frac{(CL + Q)}{V_C} \cdot \text{Central} + \frac{Q}{V_P} \cdot \text{Peripheral}$$

$$\text{Peripheral}' = \frac{Q}{V_C} \cdot \text{Central} - \frac{Q}{V_P} \cdot \text{Peripheral}$$

where:

- $CL$  is elimination clearance from the Central compartment
- $Q$  is the intercompartmental clearance
- $V_C$  is volume of the Central compartment
- $V_P$  is the volume of the Peripheral compartment

## 2-Compartment Model with First-Order Absorption

$$\text{Depot}' = -Ka \cdot \text{Depot}$$

$$\text{Central}' = Ka \cdot \text{Depot} - \frac{(CL + Q)}{V_C} \cdot \text{Central} + \frac{Q}{V_P} \cdot \text{Peripheral}$$

$$\text{Peripheral}' = \frac{Q}{V_C} \cdot \text{Central} - \frac{Q}{V_P} \cdot \text{Peripheral}$$

where:

- $CL$  is elimination clearance from the Central compartment
- $Q$  is the intercompartmental clearance
- $V_C$  is volume of the Central compartment
- $V_P$  is the volume of the Peripheral compartment
- $Ka$  is absorption rate constant

# How to make it Bayesian?

Just put **priors** in all parameters:

$$CL \sim \text{LogNormal}(\log \mu_{CL}, \sigma_{CL})$$

$$Q \sim \text{LogNormal}(\log \mu_Q, \sigma_Q)$$

$$V_C \sim \text{LogNormal}(\log \mu_{V_C}, \sigma_{V_C})$$

$$V_P \sim \text{LogNormal}(\log \mu_{V_P}, \sigma_{V_P})$$

$$Ka \sim \text{LogNormal}(\log \mu_{Ka}, \sigma_{Ka})$$

# Bayesian PK model in Pumas

```
pk2cpt = @model begin
    @param begin
        tvcl ~ LogNormal(log(10), 0.25) # CL
        tvq ~ LogNormal(log(15), 0.5)    # Q
        tvvc ~ LogNormal(log(35), 0.25) # V1
        tvvp ~ LogNormal(log(105), 0.5) # V2
        tvka ~ LogNormal(log(2.5), 1)   # ka
        σ ~ truncated(Cauchy(), 0, Inf) # sigma
    end
    @pre begin
        CL = tvcl
        Vc = tvvc
        Q = tvq
        Vp = tvvp
        Ka = tvka
    end
    @dynamics Depots1Central1Periph1
    @derived begin
        cp := @. Central/Vc
        dv ~ @. LogNormal(log(cp), σ)
    end
end
```

# Bayesian Logistic Regression

- 6.1 Recommended References
- 6.2 Binary Data
- 6.3 What is Logistic Regression
- 6.4 Comparison with Linear Regression
- 6.5 Logistic Regression Specification
- 6.6 Bayesian Logistic Regression in Pumas

# Bayesian Logistic Regression - Recommended References

- Gelman et al. (2013b) - Chapter 16: Generalized linear models
- McElreath (2020):
  - Chapter 10: Big Entropy and the Generalized Linear Model
  - Chapter 11, Section 11.1: Binomial regression
- Gelman, Hill, and Vehtari (2020):
  - Chapter 13: Logistic regression
  - Chapter 14: Working with logistic regression
  - Chapter 15, Section 15.3: Logistic-binomial model
  - Chapter 15, Section 15.4: Probit regression

## Binary Data<sup>xii</sup>

We use logistic regression when our dependent variable is **binary**. It only takes two distinct values, usually coded as 0 and 1.

---

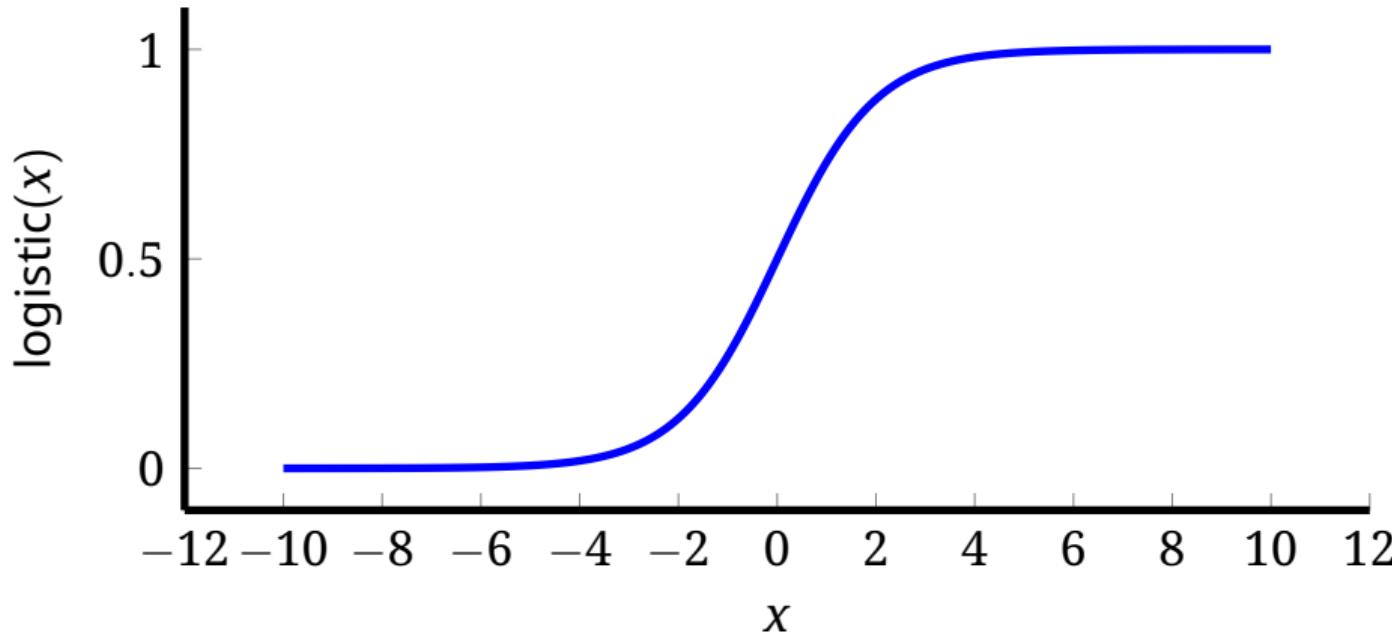
<sup>xii</sup>also known as dichotomous, dummy, indicator variable, etc.

# What is Logistic Regression

Logistic regression behaves exactly as a linear model: it makes a prediction by simply computing a weighted sum of the independent variables  $\mathbf{X}$  using the estimated coefficients  $\beta$ , along with a constant term  $\alpha$ . However, instead of outputting a continuous value  $y$ , it returns the **logistic function** of this value:

$$\text{logistic}(x) = \frac{1}{1 + e^{-x}}$$

# Logistic Function

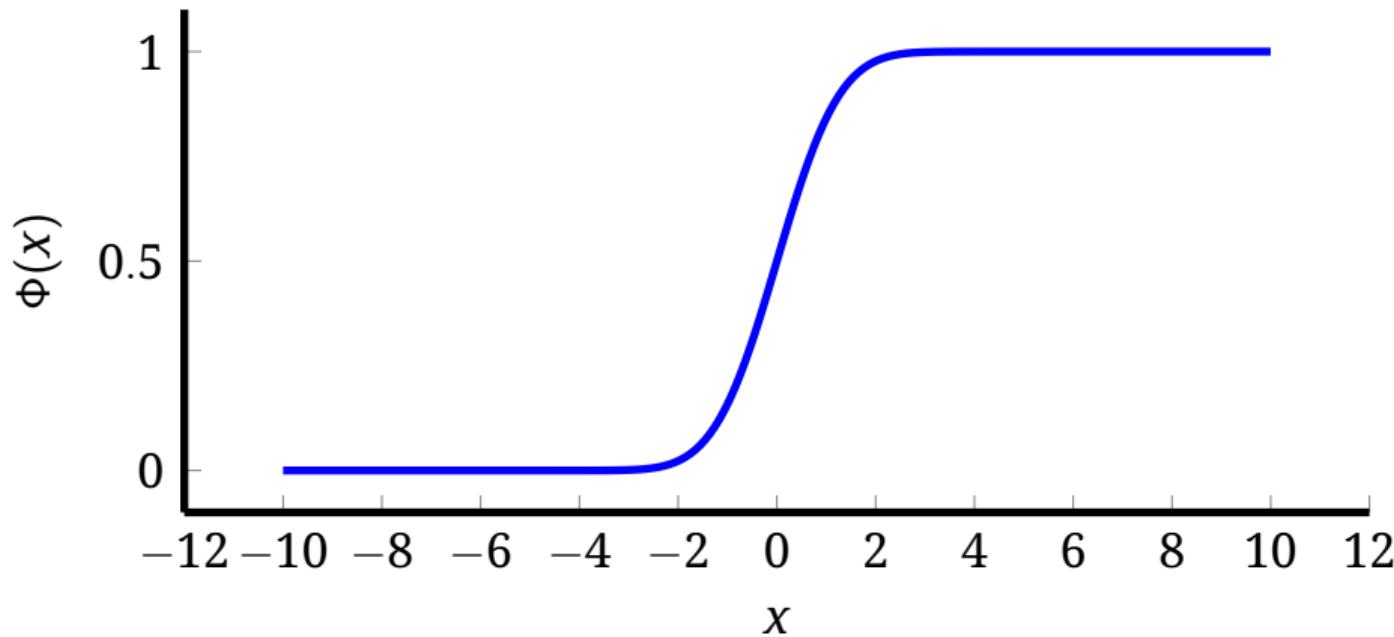


# Probit Function

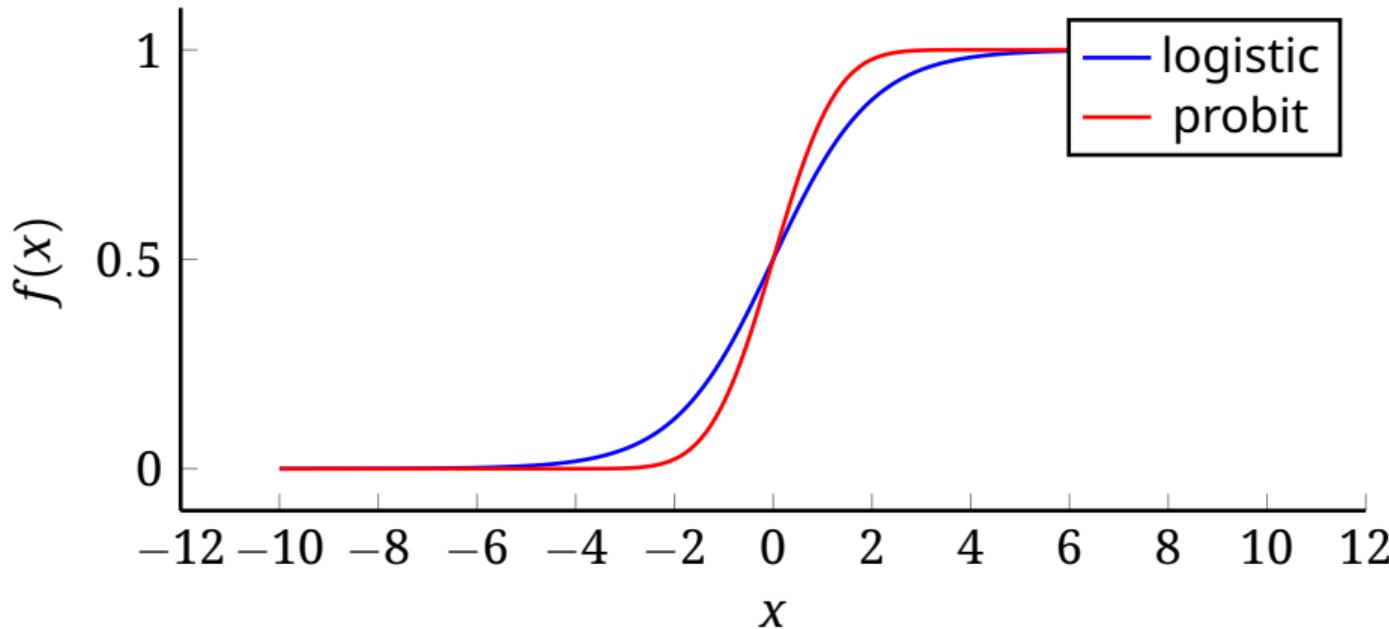
We can also opt to choose to use the **probit function** (usually represented by the Greek letter  $\Phi$ ) which is the CDF of a normal distribution:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$$

# Probit Function



# Logistic Function versus Probit Function



# Comparison with Linear Regression

Linear regression follows the following mathematical expression:

$$\text{linear} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

- $\alpha$  - intercept.
- $\beta = \beta_1, \beta_2, \dots, \beta_k$  - independent variables'  $x_1, x_2, \dots, x_k$  coefficients.
- $k$  - number of independent variables.

If you implement a small mathematical transformation, you'll have **logistic regression**:

- $\hat{p} = \text{logistic}(\text{linear}) = \frac{1}{1+e^{-\text{linear}}}$  - probability of an observation taking value 1.
- $\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$  - y's predicted binary value.

# Logistic Regression Specification

We can model logistic regression using two approaches:

- **Bernoulli likelihood – binary** dependent variable  $\mathbf{y}$  which results from a Bernoulli trial with some probability  $p$
- **binomial likelihood – discrete and positive** dependent variable  $\mathbf{y}$  which results from  $k$  successes in  $n$  independent Bernoulli trials.

# Bernoulli Likelihood

$$\mathbf{y} \sim \text{Bernoulli}(p)$$

$$p = \text{logistic/probit}(\alpha + \mathbf{X}\beta)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

where:

- **y - dependent binary variable.**
- $p$  - probability of y taking value of 1 – success in an independent Bernoulli trial.
- logistic/probit – logistic or probit function.
- $\alpha$  – intercept (also called constant).
- $\beta$  – coefficient vector.
- $\mathbf{X}$  – data matrix.

# Binomial Likelihood

$$\mathbf{y} \sim \text{Binomial}(n, p)$$

$$p = \text{logistic/probit}(\alpha + \mathbf{X}\beta)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

where:

- **y - discrete positive variable** –  $k$  successes of  $n$  independent Bernoulli trials.
- $n$  - number of independent Bernoulli trials.
- $p$  - probability of  $y$  taking value of 1 – success in an independent Bernoulli trial.
- logistic/probit – logistic or probit function.
- $\alpha$  – intercept (also called constant).
- $\beta$  – coefficient vector.
- $\mathbf{X}$  – data matrix.

# Bayesian Logistic Regression in Pumas

```
● ● ●  
logistic_model = @model begin  
    @param begin  
        α ~ Normal(0, 2.5)  
        βAUC ~ Normal(0, 2.5)  
        βisF ~ Normal(0, 2.5)  
        βRACE_Caucasian ~ Normal(0, 2.5)  
    end  
    @covariates begin  
        AUC  
        isF  
        RACE  
    end  
    @pre begin  
        linear_pred =  
            α +  
            AUC * βAUC +  
            isF * βisF +  
            (RACE == "Caucasian") * βRACE_Caucasian  
    end  
    @derived begin  
        NAUSEA ~ @. Bernoulli(logistic(linear_pred))  
    end  
end
```

# Hierarchical Models

- 7.1 Hierarchical Models - Recommended References
- 7.2 What are hierarchical models?
- 7.3 When to Use Hierarchical Models?
- 7.4 Hyperprior
- 7.5 Mathematical Specification of Hierarchical Models

# Hierarchical Models - Recommended References

- Gelman et al. (2013b):
  - Chapter 5: Hierarchical models
  - Chapter 15: Hierarchical linear models
- McElreath (2020):
  - Chapter 13: Models With Memory
  - Chapter 14: Adventures in Covariance
- Gelman and Hill (2007)
- Michael Betancourt's case study on [Hierarchical modeling](#)
- Kruschke and Vanpaemel (2015)

# I have many names...

Hierarchical models are also known for several names<sup>xiii</sup>:

- Hierarchical Models
- Random Effects Models
- Mixed Effects Models
- Cross-Sectional Models
- Nested Data Models

---

<sup>xiii</sup>for the whole full list [check here](#).

# What are hierarchical models?

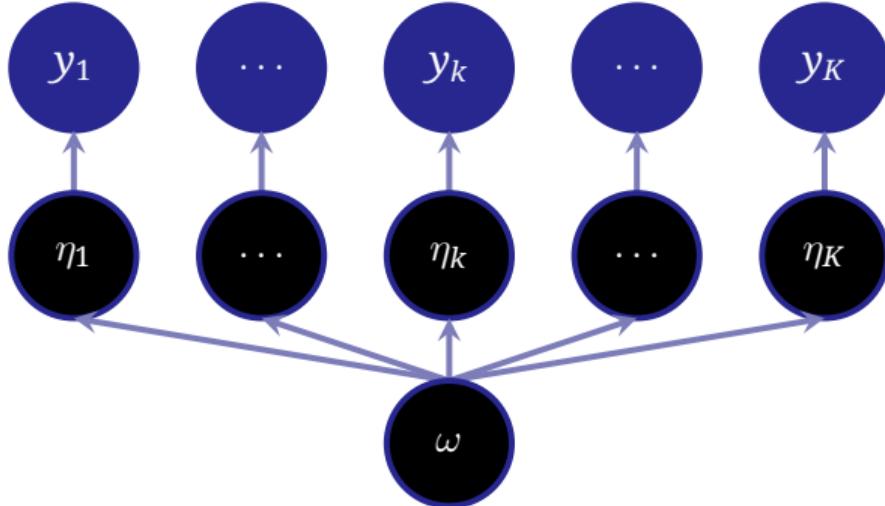
## Definition (Hierarchical Model)

*Statistical model specified in multiple levels that estimates parameters from the posterior distribution using a Bayesian approach. The sub-models inside the model combines to form a hierarchical model, and Bayes' theorem is used to integrate it to observed data and account for all uncertain.*

Hierarchical models are mathematical descriptions that involves several parameters, where some parameters' estimates depend on another parameters' values.

# What are Hierarchical Models?<sup>xiv</sup>

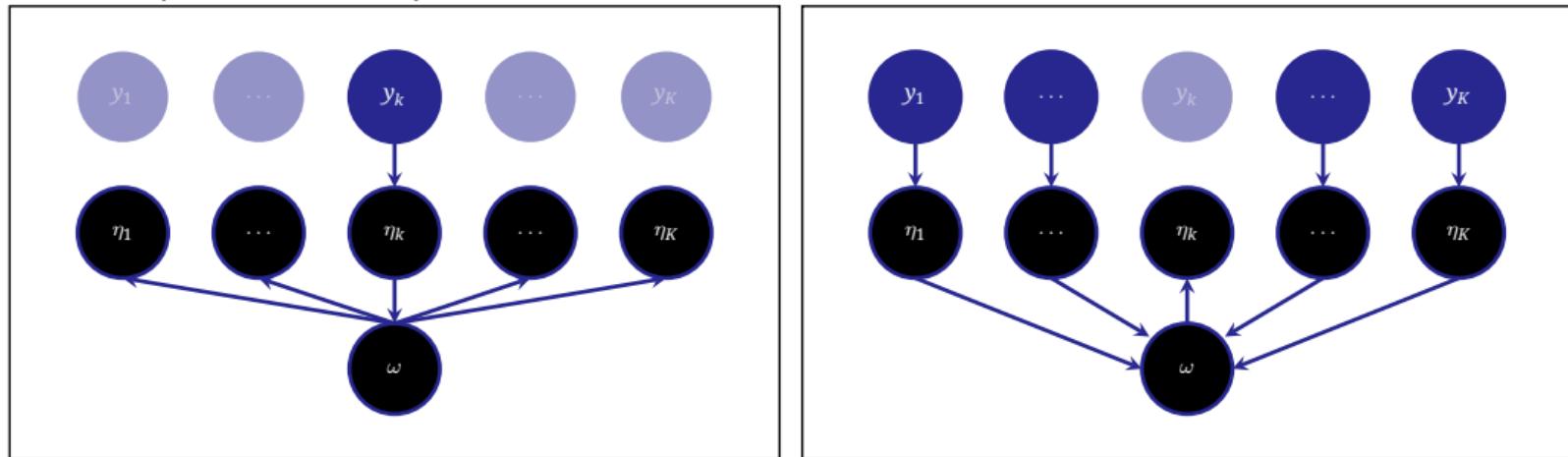
Hyperparameter  $\omega$  that parameterizes  $\eta_1, \eta_2, \dots, \eta_K$ , that are used to define the distribution of some observations  $\mathbf{y} = y_1, y_2, \dots, y_K$



<sup>xiv</sup>figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# What are Hierarchical Models?<sup>xv</sup>

Even that the observations directly inform only a single set of parameters, a hierarchical model couples individual parameters, and provides a “backdoor” for information flow.



For example, the observations from the  $k$ th group,  $y_k$ , informs directly the parameters that quantify the  $k$ th group's behavior,  $\eta_k$ . These parameters, however, inform directly the population-level parameters,  $\omega$ , that, in turn, informs others group-level parameters. In the same manner, observations that informs directly other group's parameters also provide indirectly information to population-level parameters, which then informs other group-level parameters, and so on...

<sup>xv</sup>figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# When to Use Hierarchical Models?

**Hierarchical models** are used when information is available in **several levels of units of observation**. The hierarchical structure of analysis and organization assists in the understanding of **multiparameter problems**, while also performing a crucial role in the development of **computational strategies**.

# When to Use Hierarchical Models?

Hierarchical models are particularly appropriate for research projects where participant data can be organized in more than one level<sup>xvi</sup>. The units of analysis are generally individuals that are nested inside contextual/aggregate units (groups).

An example is when we measure individual performance and we have additional information about distinct group membership such as:

- sex
- age group
- income level
- education level
- state/province of residence

---

<sup>xvi</sup>also known as nested data.

# When to Use Hierarchical Models?

Another good use case is **big data** (Gelman et al., 2013b).

- simple nonhierarchical models are usually inappropriate for hierarchical data: with few parameters, they generally *cannot* fit large datasets accurately.
- whereas with many parameters, they tend to **overfit**.
- hierarchical models can have enough parameters to fit the data well, while using a population distribution to structure some dependence into the parameters, thereby **avoiding problems of overfitting**.

# When to Use Hierarchical Models?

Most important is **not to violate** the **exchangeability assumption** (de Finetti, 1974).

This assumption stems from the principle that **groups are exchangeable**.

# Exchangeability (de Finetti, 1974)<sup>xvii</sup>



<sup>xvii</sup>figures adapted from Michael Betancourt (CC-BY-SA-4.0).

# Exchangeability (de Finetti, 1974)<sup>xviii</sup>



<sup>xviii</sup>figures adapted from Michael Betancourt (CC-BY-SA-4.0).

# Hyperprior

In hierarchical models, we have a hyperprior, which is a prior's prior:

$$\mathbf{y} \sim \text{Normal}(10, \boldsymbol{\eta})$$

$$\boldsymbol{\eta} \sim \text{Normal}(0, \omega)$$

$$\omega \sim \text{Normal}^+(1)$$

Here  $\mathbf{y}$  is a variable of interest that belongs to distinct groups.  $\boldsymbol{\eta}$ , a prior for  $\mathbf{y}$ , is a vector of group-level parameters with their own prior (which becomes a hyperprior)  $\omega$ .

# Mathematical Specification of Hierarchical Models

We have  $N$  observations organized in  $J$  groups with  $K$  covariates.

$$\mathbf{y} \sim \text{Normal}(\mathbf{X}\boldsymbol{\eta}_j, \sigma)$$

$$\boldsymbol{\eta}_j \sim \text{Multivariate Normal}(\mathbf{0}, \boldsymbol{\Omega}) \quad \text{for } j \in \{1, \dots, J\}$$

$$\boldsymbol{\Omega} = \text{Diagonal}(\boldsymbol{\omega}) \cdot \mathbf{C} \cdot \text{Diagonal}(\boldsymbol{\omega})$$

$$\boldsymbol{\omega} \sim \text{Normal}(0, 0.4)$$

$$\mathbf{C} \sim \text{LKJ}(\eta)$$

$$\sigma \sim \text{Exponential}(1)$$

Each coefficient vector  $\boldsymbol{\eta}_j$  represents the model columns  $\mathbf{X}$  coefficients for every group  $j \in J$ .

# Mathematical Specification of Hierarchical Models

If you need to extend to more than one group, such as  $J_1, J_2, \dots$ :

$$\mathbf{y} \sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\eta}_{j1} + \mathbf{X}\boldsymbol{\eta}_{j2}, \sigma)$$

$$\boldsymbol{\eta}_{j1} \sim \text{Multivariate Normal}(\mathbf{0}, \boldsymbol{\Omega}_1) \quad \text{for } j_1 \in \{1, \dots, J_1\}$$

$$\boldsymbol{\eta}_{j2} \sim \text{Multivariate Normal}(\mathbf{0}, \boldsymbol{\Omega}_2) \quad \text{for } j_2 \in \{1, \dots, J_2\}$$

$$\boldsymbol{\Omega}_1 = \text{Diagonal}(\boldsymbol{\omega}_1) \cdot \mathbf{C}_1 \cdot \text{Diagonal}(\boldsymbol{\omega}_2)$$

$$\boldsymbol{\Omega}_2 = \text{Diagonal}(\boldsymbol{\omega}_1) \cdot \mathbf{C}_2 \cdot \text{Diagonal}(\boldsymbol{\omega}_2)$$

$$\mathbf{C}_1 \sim \text{LKJ}(\eta_1)$$

$$\mathbf{C}_2 \sim \text{LKJ}(\eta_2)$$

$$\sigma \sim \text{Exponential}(1)$$

# Bayesian Population Pharmacokinetic Modeling

- 8.1 Recommended References
- 8.2 Population Pharmacokinetic Models
- 8.3 Bayesian PopPK Models

# Bayesian Population Pharmacokinetic Modeling - Recommended References

- Gabrielsson et al. (2006):
  - Chapter 1: General Principles
  - Chapter 2: Pharmacokinetic Concepts
- Owen and Fiedler-Kelly (2014):
  - Chapter 10: PK/PD Models
- Bonate (2011):
  - Chapter 10: Bayesian Modeling regression
- Margossian et al. (2022)

# Population Pharmacokinetic Models

Most of the Pharmacokinetic data comes from multiple subjects

How do we incorporate **between-subject variability (BSV)** into our model?

Answer: **Hierarchical** Models

# Adding Between-Subject Variability

Here we introduce a subject-specific parameter,  $\eta_i$ , for each subject  $i$  to capture the heterogeneity between subjects while recognizing similarities.

This is sometimes called a “random-effect” to contrast it to the “fixed-effects” which are the population-level parameters.

This nomenclature is inherited from the nonlinear mixed effects literature.

However in Bayesian, this is a misnomer since all the population and subject-specific parameters are modelled as random variables.

# 1-Compartment Model with Between-Subject Variability

$$\text{Central}' = -\frac{CL}{V_c} \cdot \text{Central}$$

where:

- $CL_i = \theta_{CL} * e^{\eta_{CL,i}}$  is elimination clearance from the Central compartment, decomposed onto:
  - $\theta_{CL}$  the typical value (population value) of the clearance parameter
  - $\eta_{CL,i}$  subject  $i$ 's subject-specific deviation from the population value
- $V_{C,i} = \theta_{V_C} * e^{\eta_{V_C,i}}$  is volume of the Central compartment, decomposed onto:
  - $\theta_{V_C}$  the typical value (population value) of the volume parameter
  - $\eta_{V_C,i}$  subject  $i$ 's subject-specific deviation from the population value

# How to make it Bayesian?

Just put **priors** on all parameters, e.g.:

$$\theta \sim \text{Normal}(0, 2.5)$$

$$\omega \sim \text{Normal}^+(0, 2.5)$$

$$\eta_i \sim \text{Normal}(0, \omega)$$

where each subject  $i$  has its own  $\eta_i$ .

# Hands-on

# Bayesian Population Pharmacodynamic Modeling

- 9.1 Recommended References
- 9.2 Pharmacodynamics
- 9.3 Compartment Models
- 9.4 Bayesian PD Models
- 9.5 Bayesian PopPD Models

# Bayesian Population Pharmacokinetic Modeling - Recommended References

- Gabrielsson et al. (2006):
  - Chapter 1: General Principles
  - Chapter 3: Pharmacodynamic Concepts
- Owen and Fiedler-Kelly (2014):
  - Chapter 10: PK/PD Models
- Bonate (2011):
  - Chapter 10: Bayesian Modeling regression
- Margossian et al. (2022)

# Pharmacodynamics

## Definition (Pharmacodynamics)

*Pharmacodynamics can be defined as the study of the time course of the biological effects of drugs, the relationship of the effects to drug exposure, and the mechanisms of drug action.*

*Gabrielsson et al. (2006, p. 199)*

# Pharmacodynamics

Pharmacodynamics is generally represented as “**PD compartments**” in a model.

They can be either:

- **Pharmacodynamic** (PD) models with only PD compartments
- **Pharmacokinetic-Pharmacodynamic** (PKPD) models with both PK compartments and PD compartments

# PD Compartment Models

There are several ways to specify (and characterize) PD compartmental models.

Here we'll use the following characterization for the PD compartments:

- **Impact:**
  - Production Rate
  - Degradation Rate
- **Rate:**
  - Stimulation
  - Inhibition
- **Effect:**
  - Emax
  - Linear
  - Sigmoid
  - ...

# PD Compartment Models

$$\text{Resp}' = \text{Resp}_0 \cdot k_{\text{out}} \cdot \text{Impact}(\text{Rate}, \text{Effect}, \text{PK compartment}) - k_{\text{out}} \cdot \text{Resp}$$

Example with Production Rate, Stimulation, Emax and Central PK compartment:

$$\text{Resp}' = \text{Resp}_0 \cdot k_{\text{out}} \cdot \left( 1 + \frac{\text{Emax} * \text{Central}}{\text{EC}_{50} + \text{Central}} \right) - k_{\text{out}} \cdot \text{Resp}$$

# How to make it Bayesian?

Just put **priors** in all parameters:

$$k_{\text{out}} \sim \text{LogNormal}(\log \mu_{k_{\text{out}}}, \sigma_{k_{\text{out}}})$$
$$\text{Resp}_0 \sim \text{LogNormal}(\log \mu_{\text{Resp}_0}, \sigma_{\text{Resp}_0})$$

...

# How to make it Bayesian?

Just put **priors** in all parameters:

$$\theta \sim \text{Normal}(0, 2.5)$$

$$\omega \sim \text{Normal}^+(0, 2.5)$$

$$\eta_i \sim \text{Normal}(0, \omega)$$

where each subject  $i$  has its own  $\eta_i$ .

# Hands-on

# Posterior Queries and Post-processing

10.1 Posterior Queries

10.2 Visual Predictive Check

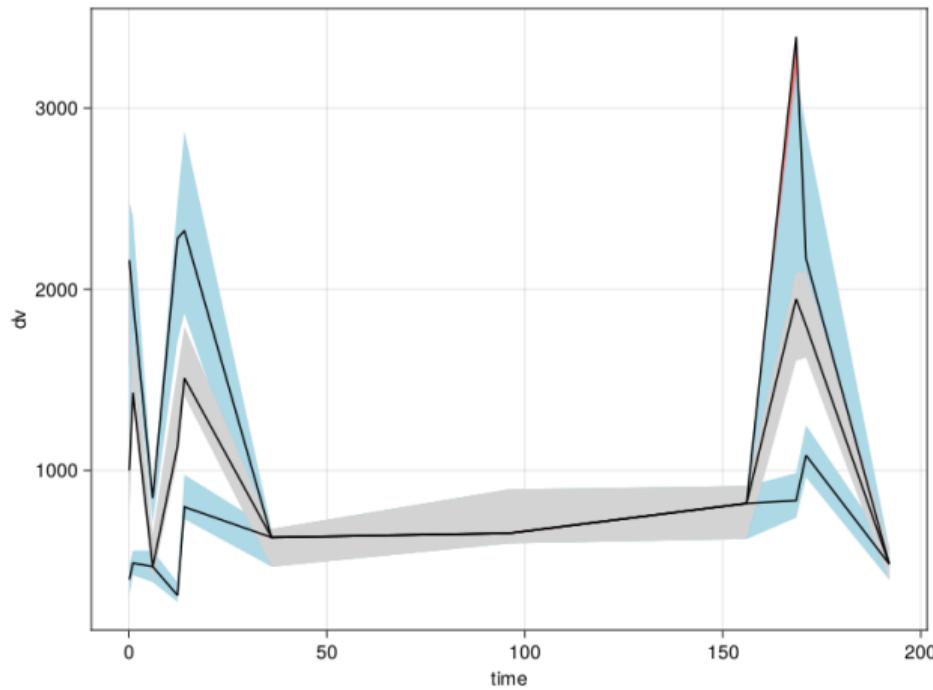
10.3 Hands-On

# Posterior Queries

$$E(\theta > 0 | \text{data})$$

$$E(y(\theta) | \text{data})$$

# Visual Predictive Check



# Hands-On

# Markov Chain Monte Carlo (MCMC) and Model Metrics

- 11.1 Markov Chain Monte Carlo (MCMC) and Model Metrics - Recommended References
- 11.2 Why Do We Need MCMC?
- 11.3 MCMC Algorithms
- 11.4 More Intuition

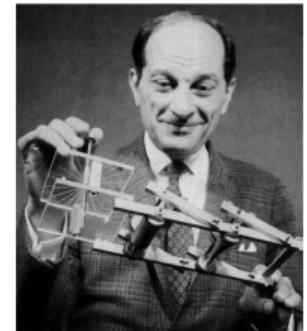
# Markov Chain Monte Carlo (MCMC) and Model Metrics

## - Recommended References

- Gelman et al. (2013b)
  - Chapter 10: Introduction to Bayesian computation
  - Chapter 11: Basics of Markov chain simulation
  - Chapter 12: Computationally efficient Markov chain simulation
- McElreath (2020) - Chapter 9: Markov Chain Monte Carlo
- Neal (2011)
- Betancourt (2017)
- Gelman, Hill, and Vehtari (2020) - Chapter 22, Section 22.8:  
Computational efficiency
- Chib and Greenberg (1995)
- Casella and George (1992)

# History Behind the Monte Carlo Methods<sup>xix</sup>

- The idea came when Ulam was playing Solitaire while recovering from surgery. Ulam was trying to calculate the deterministic, i.e. analytical solution, of the probability of being dealt an already-won game. The calculations were almost impossible. So, he thought that he could play hundreds of games to statistically estimate, i.e. numerical solution, the probability of this result.
- Ulam described the idea to John von Neumann in 1946.

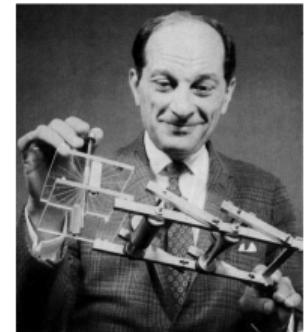


---

<sup>xix</sup>those who are interested, should read Eckhardt (1987).

# History Behind the Monte Carlo Methods<sup>xx</sup>

- Due to the secrecy, von Neumann and Ulam's work demanded a code name. Nicholas Metropolis suggested using "Monte Carlo", a homage to the "Casino Monte Carlo" in Monaco, where Ulam's uncle would ask relatives for money to play.



---

<sup>xx</sup>those who are interested, should read Eckhardt (1987).

# Why Do We Need MCMC?

The main computation barrier for Bayesian statistics is the denominator in Bayes' theorem,  $P(\text{data})$ :

$$P(\theta \mid \text{data}) = \frac{P(\theta) \cdot P(\text{data} \mid \theta)}{P(\text{data})}$$

In finite, discrete cases, we can turn the denominator into a sum over a finite number of combinations of parameters. Let  $N$  resemble the number of combinations of parameter values and let  $\theta_i$  resemble a particular vector of such combination. Using the **chain rule** of probability:

$$P(\text{data}, \theta_1) = P(\theta_1) \cdot P(\text{data} \mid \theta_1)$$

# Why Do We Need MCMC?

We can then **marginalize** the random variable  $\theta$  using:

$$P(\text{data}) = \sum_i^N P(\text{data}, \theta_i) = \sum_i^N P(\theta_i) \cdot P(\text{data} \mid \theta_i)$$

In the non-finite cases (domains are not compact, e.g. they go to  $\infty$  or  $-\infty$ ), this turns into an infinite series which can still be tractable (computationally feasible) sometimes.

In the continuous case, this turns into an integral over all values of  $\theta$ .

$$P(\text{data}) = \int_{\theta} P(\text{data} \mid \theta) \times P(\theta) d\theta$$

# Why Do We Need MCMC?

$$P(\text{data}) = \int_{\theta} P(\text{data} | \theta) \times P(\theta) d\theta$$

In many cases, e.g. if  $\theta$  is high dimensional, this integral (or sum in the discrete case) is intractable. That is, it is not feasible to deterministically evaluate or even approximate it to a known residual error. Therefore, we must find other ways to compute and use the posterior  $P(\theta | \text{data})$  without using the denominator  $P(\text{data})$ .

**This is where Monte Carlo methods come into play!**

# Why Do We Need the Denominator $P(\text{data})$ ?

To normalize the posterior with the intent of making it a **valid probability**. This means that the probability for all possible parameters' values must be 1:

- in the **discrete** case:

$$\sum_{\theta} P(\theta \mid \text{data}) = 1$$

- in the **continuous** case:

$$\int_{\theta} P(\theta \mid \text{data}) d\theta = 1$$

$P(\text{data})$  also gives us a measure of the total likelihood of a model class. This is useful when considering and comparing multiple models.

# Why Do We Need the Denominator $P(\text{data})$ ?

- After evaluating  $P(\text{data})$ , we still need to evaluate more integrals over  $\theta$ .
- A lot of useful questions can be formulated as such integrals.
- The probability  $P(\theta > 0 | \text{data})$  according to the posterior is:

$$\int_{\theta} 1_{\theta>0} P(\theta | \text{data}) d\theta$$

where  $1_{\theta>0}$  is 1 if  $\theta > 0$  and 0 otherwise.

- The expected prediction  $y(\theta)$  which is a function of the parameters  $\theta$  given the posterior distribution is:

$$\int_{\theta} y(\theta) P(\theta | \text{data}) d\theta$$

- All those integrals are typically intractable for high dimensional  $\theta$ .

# Sampling from the posterior

- What if we have a way to sample from the posterior distribution  $P(\theta | \text{data})$  without evaluating  $P(\text{data})$  or any of the above integrals?
- Can we answer our useful questions?
- Let the number of samples from the posterior be  $N$  and let  $\theta_i$  be the  $i^{\text{th}}$  sample from the posterior.

$$\int_{\theta} \mathbf{1}_{\theta > 0} P(\theta | \text{data}) d\theta \approx \frac{1}{N} \sum_i^N \mathbf{1}_{\theta_i > 0}$$

$$\int_{\theta} y(\theta) P(\theta | \text{data}) d\theta \approx \frac{1}{N} \sum_i^N y(\theta_i)$$

- These are all tractable sums!

# What If We Remove the Denominator $P(\text{data})$ ?

By removing the denominator  $P(\text{data})$ , we conclude that the posterior  $P(\theta | \text{data})$  is **proportional** to the product of the prior and the likelihood  $P(\theta) \cdot P(\text{data} | \theta)$ :

$$P(\theta | \text{data}) \propto P(\theta, \text{data}) = P(\theta) \cdot P(\text{data} | \theta)$$

# Markov Chain Monte Carlo (MCMC)

Here is where **Markov Chain Monte Carlo** comes in:

MCMC is an ample class of computational tools to approximate integrals and generate samples from a posterior probability (Brooks et al., 2011) using nothing but the un-normalized probability  $P(\theta, \text{data})$ .

MCMC is used when it is not possible to sample  $\theta$  directly from the posterior probability  $P(\theta | \text{data})$ . Instead, we collect samples in an iterative manner, where every step of the process we expect that the distribution which we are sampling from  $P^*(\theta^{(*)} | \text{data})$  becomes more similar in every iteration to the posterior  $P(\theta | \text{data})$ .

All of this is to **eliminate the evaluation** (often impossible) of the intractable integrals.

# Markov Chains

- We proceed by defining an **ergodic Markov chain<sup>xxi</sup>.** in which the set of possible states is the sample size and the stationary distribution is the distribution to be *approximated* (or *sampled*).
- Let  $X_0, X_1, \dots, X_n$  be a simulation of the chain. The Markov chain **converges to the stationary distribution from any initial state  $X_0$**  after a **sufficient large number of iterations  $r$** . The distribution of the state  $X_r$  will be similar to the stationary distribution, hence we can use it as a sample.



---

<sup>xxi</sup>meaning that there is an **unique stationary distribution**

# Markov Chains

- Markov chains have a property that the probability distribution of the next state **depends only on the current state and not in the sequence of events that preceded:**

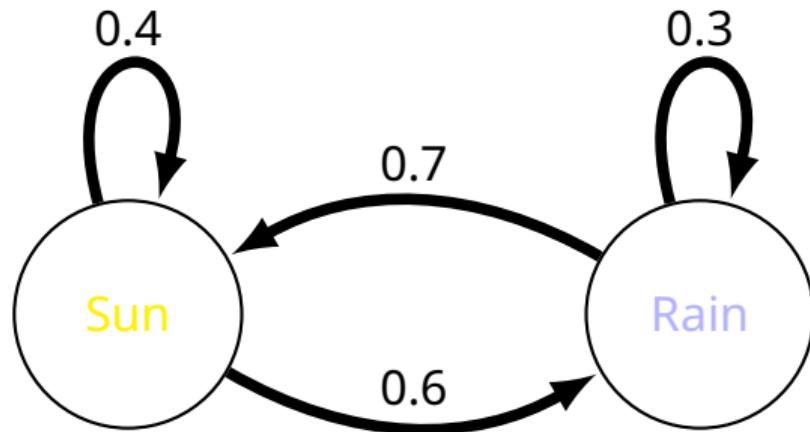
$$P(X_{n+1} = x \mid X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = x \mid X_n)$$

This property is called **Markovian**

- Similarly, using this argument with  $X_r$  as the initial state, we can use  $X_{2r}$  as a sample, and so on. We can use the sequence of states  $X_r, X_{2r}, X_{3r}, \dots$  as almost **independent samples** of Markov chain stationary distribution.



# Example of a Markov Chain



# Markov Chains

The efficacy of this approach depends on:

- **how big  $r$  must be** to guarantee an **adequate sample**.
- **computational power** required for every Markov chain iteration.

Besides, it is custom to discard the first iterations of the algorithm because they are usually non-representative of the underlying stationary distribution to be approximate. In the initial iterations of MCMC algorithms, often the Markov chain is in a “warm-up”<sup>xxii</sup> process, and its state is very far away from an ideal one to begin a trustworthy sampling.

Generally, it is recommended to **discard the first half iterations** (Gelman et al., 2013a).

---

<sup>xxii</sup>Some references call this “burnin”.

# MCMC Algorithms

MCMC algorithms<sup>xxiii</sup> use a Markov chain that's guaranteed to converge to the target posterior distribution  $P(\theta | \text{data})$  using nothing but the un-normalized probability  $P(\theta, \text{data})$ .

We have **TONS** of MCMC algorithms. Here we are going to cover two classes of MCMC algorithms:

- Metropolis-Hastings (Hastings, 1970; Metropolis et al., 1953).
- Hamiltonian Monte Carlo<sup>xxiv</sup> (Betancourt, 2017; Neal, 2011).

---

<sup>xxiii</sup>see the [Wikipedia page](#) for a full list.

<sup>xxiv</sup>sometimes called Hybrid Monte Carlo, specially in the physics literature.

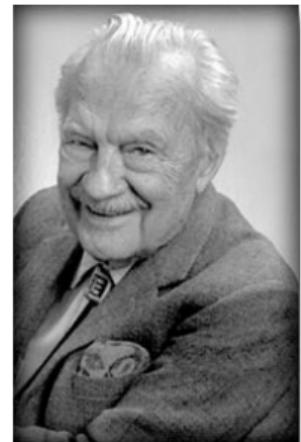
# Metropolis Algorithm

The first broadly used MCMC algorithm to generate samples from a Markov chain was originated in the physics literature in the 1950s and is called Metropolis (Metropolis et al., 1953), in honor of the first author [Nicholas Metropolis](#).

In sum, the Metropolis algorithm is an adaptation of a random walk coupled with an acceptance/rejection rule to converge to the target distribution.

Metropolis algorithm uses a **proposal distribution**  $J_t(\theta^{(*)})$  to define the next values of the distribution  $P^*(\theta^{(*)} | \text{data})$ . This distribution must be symmetric:

$$J_t(\theta^{(*)} | \theta^{(t-1)}) = J_t(\theta^{(t-1)} | \theta^{(*)})$$



# Metropolis Algorithm

Metropolis is a random walk through the parameter sample space, where the probability of the Markov chain changing its state is defined as:

$$P_{\text{change}} = \min \left( \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right).$$

This means that the Markov chain will only change to a new state based in one of two conditions:

- when the probability of the random walk proposed parameters  $P(\theta_{\text{proposed}})$  is **higher** than the probability of the current state parameters  $P(\theta_{\text{current}})$ , we change with 100% probability.
- when the probability of the random walk proposed parameters  $P(\theta_{\text{proposed}})$  is **lower** than the probability of the current state parameters  $P(\theta_{\text{current}})$ , we change with probability equal to the proportion of this probability difference.

# Metropolis Algorithm

---

## Algorithm 1: Metropolis

---

Define an initial set  $\theta^{(0)} \in \mathbb{R}^p$  that  $P(\theta^{(0)} | \mathbf{y}) > 0$

**for**  $t = 1, 2, \dots$

    Sample a proposal of  $\theta^{(*)}$  from a proposal distribution in time  $t$ ,

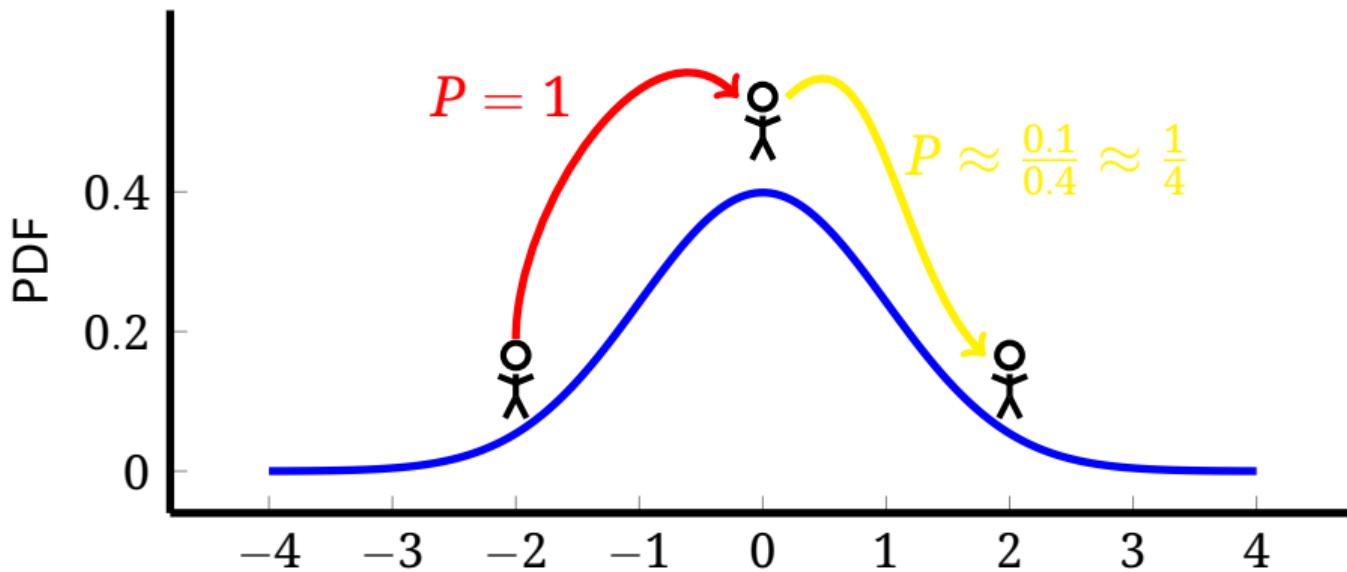
$$J_t(\theta^{(*)} | \theta^{(t-1)})$$

    As an acceptance/rejection rule, compute the proportion of the  
    probabilities:  $r = \frac{P(\theta^{(*)} | \mathbf{y})}{P(\theta^{(t-1)} | \mathbf{y})}$

    Assign:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{with probability } \min(r, 1) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$

---

# Visual Intuition – Metropolis



# Metropolis-Hastings Algorithm

In the 1970s emerged a generalization of the Metropolis algorithm, which **does not need that the proposal distributions be symmetric**:

$$J_t(\theta^{(*)} \mid \theta^{(t-1)}) \neq J_t(\theta^{(t-1)} \mid \theta^{(*)})$$

The generalization was proposed by [Wilfred Keith Hastings](#) ([Hastings, 1970](#)) and is called **Metropolis-Hastings algorithm**.



# Metropolis-Hastings Algorithm

---

## Algorithm 2: Metropolis-Hastings

---

Define an initial set  $\theta^{(0)} \in \mathbb{R}^p$  that  $P(\theta^{(0)} | \mathbf{y}) > 0$

**for**  $t = 1, 2, \dots$

Sample a proposal  $\theta^{(*)}$  from a proposal distribution in time  $t$ ,  $J_t(\theta^{(*)} | \theta^{(t-1)})$

As an acceptance/rejection rule, compute the proportion of the probabilities:

$$r = \frac{\frac{P(\theta^{(*)} | \mathbf{y})}{J_t(\theta^{(*)} | \theta^{(t-1)})}}{\frac{P(\theta^{(t-1)} | \mathbf{y})}{J_t(\theta^{(t-1)} | \theta^{(*)})}}$$

Assign:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{with probability } \min(r, 1) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$

---

# Metropolis-Hastings Animation

See Metropolis-Hastings in action at [chi-feng/mcmc-demo](https://chi-feng/mcmc-demo).

# Limitations of the Metropolis Algorithms

The limitations of the Metropolis-Hastings algorithms are mainly **computational**:

- with the proposals randomly generated, it can take a large number of iterations for the Markov chain to enter higher posterior densities spaces.
- even highly-efficient MH algorithms sometimes accept **less than** 25% of the proposals in the Gaussian case (Beskos et al., 2013; Roberts et al., 1997), often much less in complicated models.
- in lower-dimensional contexts, higher computational power can compensate the low efficiency up to a point. But in higher-dimensional (and higher-complexity) modeling situations, higher computational power alone are rarely sufficient to overcome the low efficiency.

# Hamiltonian Monte Carlo (HMC)

The current most efficient MCMC algorithms for continuous parameters. They try to **avoid the entirely random walk behavior by introducing an auxiliary vector of random momenta using Hamiltonian dynamics**. The proposals (albeit still random) are then “guided” to higher density regions of the sample space. This makes **HMC more efficient by multiple orders of magnitude when compared to MH**.

# Hamiltonian Monte Carlo (HMC)

Metropolis' low acceptance rate in multidimensional problems (where the posterior geometry is highly complex) made a new class of MCMC algorithms emerge. These are called Hamiltonian Monte Carlo (HMC), because they incorporate Hamiltonian dynamics (in honor of Irish physicist [William Rowan Hamilton](#)).



# HMC Algorithm

HMC algorithm is an adaptation of the MH algorithm, and employs a guidance scheme to the generation of new proposals. It boosts the acceptance rate, and, consequently, has a better efficiency.

More specifically, HMC uses the gradient of the posterior's log density to guide the Markov chain to higher density regions of the sample space, where most of the samples are sampled:

$$\frac{d \log P(\theta | \mathbf{y})}{d\theta}$$

As a result, a Markov chain that uses a well-adjusted HMC algorithm will accept proposals with a much higher rate than if using the MH algorithm (Beskos et al., 2013; Roberts et al., 1997).

# History of HMC Algorithm

HMC was originally described in the physics literature<sup>xxv</sup> (Duane et al., 1987).

Soon after, HMC was applied to statistical problems by Neal (1994) who named it as Hamiltonian Monte Carlo (HMC).

For a much more detailed and in-depth discussion (not our focus here) of HMC, I recommend Neal (2011) and Betancourt (2017).

---

<sup>xxv</sup>where is called “Hybrid” Monte Carlo (HMC)

# What Changes With HMC?

HMC uses Hamiltonian dynamics applied to particles efficiently exploring a posterior probability geometry, while also being robust to complex posterior's geometries.

Besides that, HMC is much more efficient than Metropolis

# Intuition Behind the HMC Algorithm

For every parameter  $\theta_j$ , HMC adds a momentum variable  $\phi_j$ . The posterior density  $P(\theta | y)$  is incremented by an independent momenta distribution  $P(\phi)$ , hence defining the following joint probability:

$$P(\theta, \phi | y) = P(\phi) \cdot P(\theta | y)$$

HMC uses a proposal distribution that changes depending on the Markov chain current state. HMC finds the direction where the posterior density increases, the **gradient**, and alters the proposal distribution towards the gradient direction.

The probability of the Markov chain to change its state in HMC is defined as:

$$P_{\text{change}} = \min \left( \frac{P(\theta_{\text{proposed}}) \cdot P(\phi_{\text{proposed}})}{P(\theta_{\text{current}}) \cdot P(\phi_{\text{current}})}, 1 \right)$$

# Momenta Distribution – $P(\phi)$

Generally we give  $\phi$  a multivariate normal distribution with mean 0 and covariance  $\mathbf{M}$ , a “mass matrix”.

To keep things computationally simple, we used a **diagonal** mass matrix  $\mathbf{M}$ . This makes that the diagonal elements (components)  $\phi$  are independent, each one having a normal distribution:

$$\phi_j \sim \text{Normal}(0, M_{jj})$$

# HMC Algorithm

---

## Algorithm 3: Hamiltonian Monte Carlo (HMC)

---

Define an initial set  $\theta^{(0)} \in \mathbb{R}^p$  that  $P(\theta^{(0)} | \mathbf{y}) > 0$

Sample  $\phi$  from a Multivariate Normal( $\mathbf{0}, \mathbf{M}$ )

Simultaneously sample  $\theta^{(*)}$  and  $\phi$  with  $L$  steps and step-size  $\epsilon$ .

Define the current value of  $\theta$  as the proposed value  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta$

**for**  $1, 2, \dots, L$

Use the log of the posterior's gradient  $\theta^{(*)}$  to produce a half-step of  $\phi$ :  $\phi \leftarrow \phi + \frac{1}{2}\epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$

Use  $\phi$  to update  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta^{(*)} + \epsilon \mathbf{M}^{-1} \phi$

Use again  $\theta^{(*)}$  log gradient to produce a half-step of  $\phi$ :  $\phi \leftarrow \phi + \frac{1}{2}\epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$

As an acceptance/rejection rule, compute:  $r = \frac{P(\theta^{(*)} | \mathbf{y}) P(\phi^{(*)})}{P(\theta^{(t-1)} | \mathbf{y}) P(\phi^{(t-1)})}$

Assign:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{with probability } \min(r, 1) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$

---

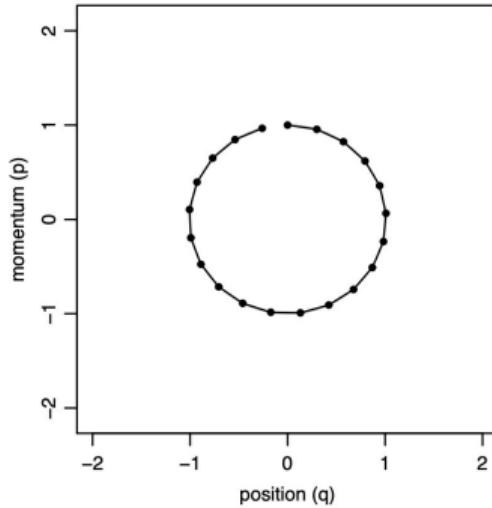
# HMC Animation

See HMC in action at [chi-feng/mcmc-demo](https://chi-feng/mcmc-demo).

# HMC Stepping Algorithm

The main stepping algorithm used in HMC is the **Störmer-Verlet integrator**, also known as **leapfrog integrator**.

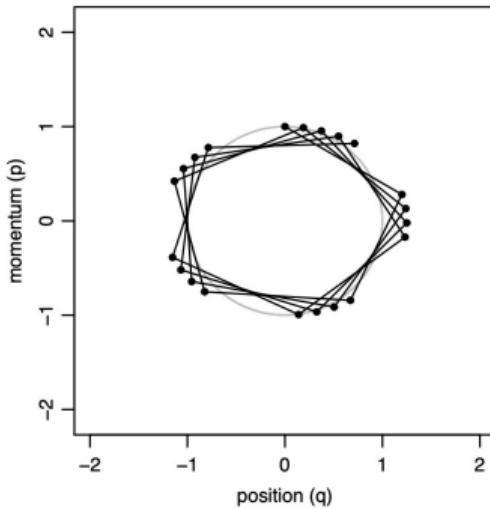
The stepping of the proposal is mathematically equivalent to solving a numerical integration problem, hence the use of integrators.



HMC numerically integrated using leapfrog with  $\epsilon = 0.3$  and  $L = 20$

# Limitations of the HMC Algorithm

As you can see, HMC algorithm is highly sensitive to the choice of leapfrog steps  $L$  and step-size  $\epsilon$ . More specific, the leapfrog integrator allows only a constant  $\epsilon$ . There is a delicate balance between  $L$  and  $\epsilon$ , that are hyperparameters and need to be carefully adjusted.



HMC numerically integrated using leapfrog with  $\epsilon = 1.2$  and  $L = 20$

## No-U-Turn-Sampler (NUTS)

In HMC, we can adjust  $\epsilon$  during the algorithm runtime. But, for  $L$ , we need to “dry run” the HMC sampler to find a good candidate value for  $L$ .

Here is where the idea for **No-U-Turn-Sampler (NUTS)** (Hoffman & Gelman, 2011) enters: you don’t need to **adjust anything**, just “press the button”. It will automatically find  $\epsilon$  and  $L$ .

# No-U-Turn-Sampler (NUTS)

More specifically, we need a criterion that informs that we performed enough Hamiltonian dynamics simulation. In other words, to simulate past beyond would not increase the distance between the proposal  $\theta^{(*)}$  and the current value  $\theta$ .

NUTS uses a criterion based on the dot product between the current momenta vector  $\phi$  and the difference between the proposal vector  $\theta^{(*)}$  and the current vector  $\theta$ , which turns into the derivative with respect to time  $t$  of half of the distance squared between  $\theta$  e  $\theta^{(*)}$ :

$$(\theta^{(*)} - \theta) \cdot \phi = (\theta^{(*)} - \theta) \cdot \frac{d}{dt}(\theta^{(*)} - \theta) = \frac{d}{dt} \frac{(\theta^{(*)} - \theta) \cdot (\theta^{(*)} - \theta)}{2}$$

## No-U-Turn-Sampler (NUTS)

This suggests an algorithm that does not allow proposals to be guided infinitely until the distance between the proposal  $\theta^{(*)}$  and the current  $\theta$  is less than zero.

This means that such algorithm will **not allow u-turns**.

# No-U-Turn-Sampler (NUTS)

NUTS uses the leapfrog integrator to create a binary tree where each leaf node is a proposal of the momenta vector  $\phi$  tracing both a forward ( $t + 1$ ) as well as a backward ( $t - 1$ ) path in a determined fictitious time  $t$ . The growing of the leaf nodes are **interrupted** when an u-turn is detected, both forward or backward.

## No-U-Turn-Sampler (NUTS)

NUTS also uses a procedure called Dual Averaging (Nesterov, 2009) to simultaneously adjust  $\epsilon$  and  $L$  by considering the product  $\epsilon \cdot L$ .

Such adjustment is done during the warmup phase and the defined values of  $\epsilon$  and  $L$  are kept fixed during the sampling phase.

# NUTS Algorithm

---

## Algorithm 4: No-U-Turn-Sampler (NUTS)

---

Define an initial set  $\theta^{(0)} \in \mathbb{R}^p$  that  $P(\theta^{(0)} | y) > 0$

Instantiate an empty binary tree with  $2^L$  leaf nodes

Sample  $\phi$  from a Multivariate Normal( $0, M$ )

Simultaneously sample  $\theta$  and  $\phi$  with  $L$  leapfrog steps and step-size  $\epsilon$ .

Define the current value  $\theta$  as the proposed value  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta$

for  $1, 2, \dots, 2L$

    Choose a direction  $v \sim \text{Uniform } \{-1, 1\}$

    Use the gradient of the log posterior  $\theta^{(*)}$  for a half-step of  $\phi$  in the direction  $v$ :  $\phi \leftarrow \phi + v \frac{1}{2} \epsilon \frac{d \log P(\theta^{(*)} | y)}{d\theta}$

    Use  $\phi$  to update  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta^{(*)} + \epsilon M^{-1} \phi$

    Again use the gradient of the log posterior  $\theta^{(*)}$  for a half-step of  $\phi$  in the direction  $v$ :  $\phi \leftarrow \phi + v \frac{1}{2} \epsilon \frac{d \log P(\theta^{(*)} | y)}{d\theta}$

    Define the node  $L_t^y$  as the proposal  $\theta$

    if The difference between proposal vector  $\theta^{(*)}$  and current vector  $\theta$  in the direction  $v$  is lower than zero:  $v \frac{d}{dt} \frac{(\theta^{(*)} - \theta^{(*)}) \cdot (\theta^{(*)} - \theta^{(*)})}{2} < 0$

        then

            | Stop sampling  $\theta^{(*)}$  in the direction  $v$  and continue sampling only in the direction  $-v$

        if The difference between proposal vector  $\theta^{(*)}$  and current vector  $\theta$  in the direction  $-v$  is lower than zero:  $-v \frac{d}{dt} \frac{(\theta^{(*)} - \theta^{(*)}) \cdot (\theta^{(*)} - \theta^{(*)})}{2} < 0$

            then

                | Stop sampling  $\theta^{(*)}$

As an acceptance/rejection rule, compute:  $r = \frac{P(\theta^{(*)} | y) P(\phi^{(*)})}{P(\theta^{(t-1)} | y) P(\phi^{(t-1)})}$

Assign:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{with probability } \min(r, 1) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$

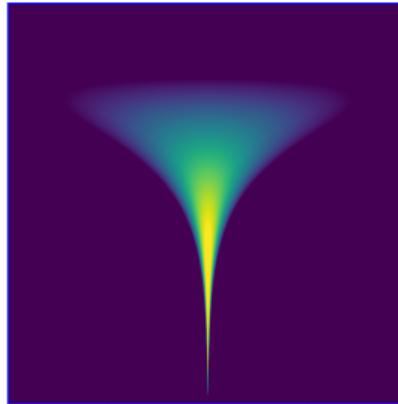
---

# NUTS Animation

See NUTS in action at [chi-feng/mcmc-demo](https://chi-feng/mcmc-demo).

# Limitations of HMC and NUTS Algorithms – Neal (2003)'s Funnel

The famous “Devil’s Funnel”<sup>xxvi</sup>. Here we see that HMC and NUTS, during the exploration of the posterior, have to change often  $L$  and  $\epsilon$  values<sup>xxvii</sup>.



---

<sup>xxvi</sup>very common in hierarchical models.

## Neal (2003)'s Funnel and Non-Centered Parameterization (NCP)

Sometimes the group-level effects do not constrain the hierarchical distribution tightly.

Examples arise when there are not many groups, or when the inter-group variation is high.

In such cases, hierarchical models can be made much more efficient by shifting the data's correlation with the parameters to the hyperparameters.

## Neal (2003)'s Funnel and Non-Centered Parameterization (NCP)

The funnel occurs when we have a variable that its variance depends on another variable variance in an exponential scale. A canonical example of a centered parameterization (CP) is:

$$P(y, x) = \text{Normal}(y | 0, 3) \cdot \text{Normal}\left(x | 0, e^{\left(\frac{y}{2}\right)}\right)$$

This occurs often in hierarchical models, in the relationship between group-level priors and population-level hyperpriors. Hence, we reparameterize in a non-centered way, changing the posterior geometry to make life easier for our MCMC sampler:

$$\begin{aligned} P(\tilde{y}, \tilde{x}) &= \text{Normal}(\tilde{y} | 0, 1) \cdot \text{Normal}(\tilde{x} | 0, 1) \\ y &= \tilde{y} \cdot 3 + 0 \\ x &= \tilde{x} \cdot e^{\left(\frac{y}{2}\right)} + 0 \end{aligned}$$

## More Intuition - Step Size and Acceptance Ratio

- The NUTS algorithm adapts its stepping algorithm to encourage a certain fraction of the proposals to get accepted on average.
- A value of 0.99 means that we want to accept 99% of the proposals the sampler makes.
- This will generally lead to small step sizes between the proposal and the current sample since this increases the chance of accepting such a proposal.

## More Intuition - Step Size and Acceptance Ratio

- On the other hand, a target acceptance fraction of 0.2 means that we want to only accept 20% of the proposals made on average.
- The NUTS algorithm will therefore attempt larger step sizes to ensure it rejects 80% of the proposals.
- In general, a target acceptance ratio value of 0.6-0.8 is recommended to use.

## More Intuition - Exploration vs Exploitation

- In sampling, there is usually a tradeoff between exploration and exploitation.
- If the sampler is too “adventurous”, trying aggressive proposals that are far from the previous sample in each step, the sampler would be more likely to explore the full posterior and not get stuck sampling near a local mode of the posterior.
- However on the flip side, too much exploration will often lead to many sample rejections due to low joint probability of the data and the adventurous proposals. This can decrease the ratio of the effective sample size (ESS) to the total number of samples (aka relative ESS) since a number of samples will be mere copies of each other due to rejections.

## More Intuition - Exploration vs Exploitation

On the other hand if we do less exploration, there are 2 possible scenarios:

- The first scenario is if we initialize the sampler from a mode of the posterior.
  - Making proposals only near the previous sample will ensure that we accept most of the samples.
  - Proposals near a mode of the posterior are likely to be good parameter values.
  - This local sampling behavior around known good parameter values is what we call exploitation.
  - While the samples generated via high exploitation around a mode may not be representative of the whole posterior distribution, they might still give a satisfactory approximation of the posterior predictive distributions.

## More Intuition - Exploration vs Exploitation

- The second scenario is if we initialize the sampler from bad parameter values.
  - Bad parameter values and low exploration often lead to optimization-like behavior where the sampler spends a considerable number of iterations moving towards a mode in a noisy fashion.
  - This optimization-like, mode-seeking behavior causes a high auto-correlation in the samples since the sampler is mostly moving in the same direction (towards the mode).
  - A high auto-correlation means a low ESS because the samples would be less independent from each other.
  - Also until the sampler reaches parameter values that actually fit the data well, it's unlikely these samples will lead to a good posterior predictive distribution.

# More Intuition - Exploration vs Exploitation

- The second scenario is if we initialize the sampler from bad parameter values.
  - This is a fairly common failure mode of MCMC algorithms when the adaptation algorithm fails to find a good stepping algorithm that properly explores the posterior distribution due to bad initial parameters and the model being too complicated and difficult to optimize, let alone sample from its posterior.
  - In this case, all the samples may look auto-correlated and the step sizes between samples will likely be very small (low exploration).
  - It's often helpful to detect such a failure mode early in the sampling and kill the sampling early.

## More Intuition - Mass Matrix, Adaptation and U-Turn

- The NUTS algorithm adapts the level of exploration in its proposal mechanism to achieve the target acceptance ratio.
- However, the definition of an “exploratory” proposal may be different depending on the parameter values of the current sample.
- For relatively flat regions of the posterior where a lot of parameter values are almost equally likely (i.e. they all fit the data well and are almost equally probable according to the prior), proposals far away from the current sample may still be accepted most of the time.
- This is especially likely in the parts of the posterior where the model is non-identifiable or there are high parameter correlations, and the prior is indiscriminate (e.g. due to being a weak prior).

## More Intuition - Mass Matrix, Adaptation and U-Turn

- On the other hand, regions of the posterior that are heavily concentrated around a mode with a high curvature often require a smaller step size to achieve reasonable acceptance ratios.
- Proposals that are even slightly far from the current sample may be extremely improbable according to the prior or may lead to very bad predictions.
- This is especially likely in regions of the posterior where the model is highly sensitive to the parameter values or if the prior is too strongly concentrated around specific parameter values.

## More Intuition - Mass Matrix, Adaptation and U-Turn

- To account for such variation in curvature (wrt the same parameter in different regions), the NUTS algorithm uses a multi-step proposal mechanism with a fixed step size (determined during adaptation and then fixed) and a dynamic number of steps (dynamic in both adaptation and regular sampling).
- In other words, the sampler follows a trajectory of length  $L \geq 1$  steps before proposing a new sample to move to, where  $L$  is different in each proposal.
- This proposal then gets tested and is either accepted or rejected by comparing it to the previous sample.

## More Intuition - Mass Matrix, Adaptation and U-Turn

- The exact trajectory length taken by the NUTS sampler to propose a new sample is obtained by a binary tree search procedure with the maximum depth of the tree being a hyper-parameter of the algorithm.
- The goal of the tree search is to look for a point in the trajectory where a U-Turn happens, i.e. the sampler is beginning to trace back its steps.
- Once a U-turn is found, a point (randomly chosen) before the U-Turn becomes the next proposal and the search is terminated.
- This is typically considered a sign of successful exploration.

## More Intuition - Mass Matrix, Adaptation and U-Turn

- For some complicated models, the adaptation can often result in a step size that is too small.
- In such a case, no U-Turn may be found early and the tree search may have to run to completion before making a single proposal.
- If this happens in almost every iteration of the MCMC sampling, this will be bad for performance because the number of model evaluations required by a complete search tree of depth  $d$  is  $2^d - 1$ .

## More Intuition - Mass Matrix, Adaptation and U-Turn

- The step size and trajectory length are not the only ways which the NUTS algorithm uses to adapt the level of exploration in the sampling.
- The so-called mass matrix (determined during adaptation and then fixed afterwards) allows the sampler to adapt the amount of exploration differently for different directions.
- This is especially important for models where parameters are on different scales so variables smaller in magnitude should generally be changing less between samples than variables larger in magnitude.

## More Intuition - Mass Matrix, Adaptation and U-Turn

- In Pumas, we use a diagonal mass matrix in BayesMCMC (joint MCMC) where different exploration levels are used for different model parameters.
- If the mass matrix is dense (e.g. in MarginalMCMC) (marginal MCMC), the exploration level can be different along arbitrary directions and not just along the parameters' axes.
- This tends to help when the parameters are heavily correlated in the posterior.
- The reason why it's called the mass matrix is an analogy to Hamiltonian dynamics that the HMC and NUTS algorithms were inspired from.

# Stan and NUTS

Stan was the first MCMC sampler to implement NUTS. Besides that, it has an automatic optimized adjustment routine for values of  $L$  and  $\epsilon$  during warmup. It has the following default NUTS hyperparameters' values<sup>xxviii</sup>:

- **target acceptance rate of Metropolis proposals:** 0.8
- **max tree depth:** 10

---

<sup>xxviii</sup>for more information about how to change those values, see [Section 15.2 of the Stan Reference Manual](#).

## Pumas and NUTS

Pumas also uses a NUTS implementation from the package `AdvancedHMC.jl`. It also has an automatic optimized adjustment routine for values of  $L$  and  $\epsilon$  during warmup that's identical to Stan's. It has the same default NUTS hyperparameters' values:

- **target acceptance rate of Metropolis proposals:** 0.8
- **max tree depth:** 10

# Marginal MCMC for Hierarchical Models

12.1 Background

12.2 Hierarchical Models

12.3 Comparison to Joint MCMC

# Joint MCMC

- In regular MCMC, we sample from the joint posterior of all the population and subject-specific parameters.
- Assume there are 3 subjects with subject parameters  $\eta_1$ ,  $\eta_2$  and  $\eta_3$ .
- Let the population parameters be  $\theta$ .
- Joint MCMC samples from the joint posterior:

$$P(\theta, \eta_1, \eta_2, \eta_3 \mid \text{data})$$

- This means that the number of parameters to sample from increases linearly with the number of subjects.

# Marginal Posterior

- When there are many subjects and high correlation in the parameters in the posterior, the NUTS algorithm can often struggle to find good adaptation parameters.
- This is common if there are model identifiability issues and/or the priors are too weak compared to the likelihood.
- This is not uncommon in hierarchical models.
- After sampling from the joint posterior, we are often only interested in population parameters, e.g. to estimate a drug effect.
- The following is the marginal posterior of  $\theta$  given the data.

$$P(\theta \mid \text{data})$$

# Sample-Then-Marginalize vs Marginalize-Then-Sample

- There are 2 ways to marginalize out the subject-specific parameters:
  - Sample from the joint posterior and then ignore the subject-specific parameters. You can easily do this using BayesMCMC.
  - Integrate the subject-specific parameters out and then sample from the marginal posterior only. This is what MarginalMCMC does in Pumas.
- The marginal posterior probability is proportional to:

$$P(\theta \mid \text{data}) \propto P(\theta, \text{data}) = \int_{\eta_3} \int_{\eta_2} \int_{\eta_1} P(\theta, \eta_1, \eta_2, \eta_3, \text{data}) d\eta_1 d\eta_2 d\eta_3$$

- MCMC can sample from  $P(\theta \mid \text{data})$  given  $P(\theta, \text{data})$  only without requiring the normalization constant.

# Hierarchical Models

For hierarchical models, we can further break down the joint probability into the product of independent conditionals and  $P(\theta)$ :

$$P(\theta, \eta_1, \eta_2, \eta_3, \text{data}) = P(\theta) \cdot \prod_{i=1}^3 P(\eta_i, \text{data}_i | \theta)$$

where  $\text{data}_i$  is the data of subject  $i$ .

# Hierarchical Models

The integral can therefore be simplified to the product of  $P(\theta)$  and 3 smaller integrals:

$$\begin{aligned} P(\theta, \text{data}) &= \int_{\eta_3} \int_{\eta_2} \int_{\eta_1} P(\theta) \cdot \prod_{i=1}^3 P(\eta_i, \text{data}_i | \theta) d\eta_1 d\eta_2 d\eta_3 \\ &= P(\theta) \cdot \prod_{i=1}^3 \int_{\eta_i} P(\eta_i, \text{data}_i | \theta) d\eta_i \end{aligned}$$

Each of the above smaller integrals has a low dimension.

# Approximate Marginalization

- The MarginalMCMC algorithm in Pumas uses approximate integration methods, e.g. the Laplace method or the first order conditional estimation (FOCE) approximation of the Laplace method to compute the subject-specific integrals.
- This approximation is accurate if the conditional posterior  $P(\eta_i|\theta, \text{data}_i)$  is approximately Gaussian.
- This is often the case if the model is conditionally identifiable wrt  $\eta_i$  after fixing  $\theta$  and there is enough data per subject to identify the true parameters  $\eta_i$ .

# Efficiency and Cost

- The remaining parameters  $\theta$  are then sampled from their marginal posterior using the NUTS algorithm and a dense mass matrix.
- This tends to leads to better adaptation and more efficient sampling.
- However, each evaluation of  $P(\theta, \text{data})$  requires solving as many optimization problems as the number of subjects which is significantly more expensive than evaluating the total joint probability.

# Efficiency and Cost

- So there is a tradeoff between the cost per HMC step (marginal MCMC is more expensive) and the number of steps per proposal (marginal MCMC requires less steps).
- However, the computational cost of marginal MCMC can be more effectively parallelized by parallelizing the subject integral computations.
- The computational effort per subject per HMC step in marginal MCMC is higher than the joint MCMC method.
- This makes up for the parallelism overhead of having to manage and communicate with multiple threads/processes.

# Marginal vs Joint MCMC

|                             | Marginal MCMC  | Joint MCMC                          |
|-----------------------------|--|-------------------------------------|
| <b>Number of parameters</b> | Low  | High                                |
| <b>Accuracy</b>             | Approximate even for infinite samples unless the conditional posterior $P(\eta_i   \theta, \text{data}_i)$ is Gaussian | Exact with infinite samples         |
| <b>Cost per HMC step</b>    | High   | Low                                 |
| <b>Mass matrix</b>          | Dense  | Diagonal (by default)               |
| <b>Max tree depth</b>       | Often low  | Often high                          |
| <b>Parallelism</b>          | Efficient for all models   | Efficient only for difficult models |

# Example: Rare Disease Modelling

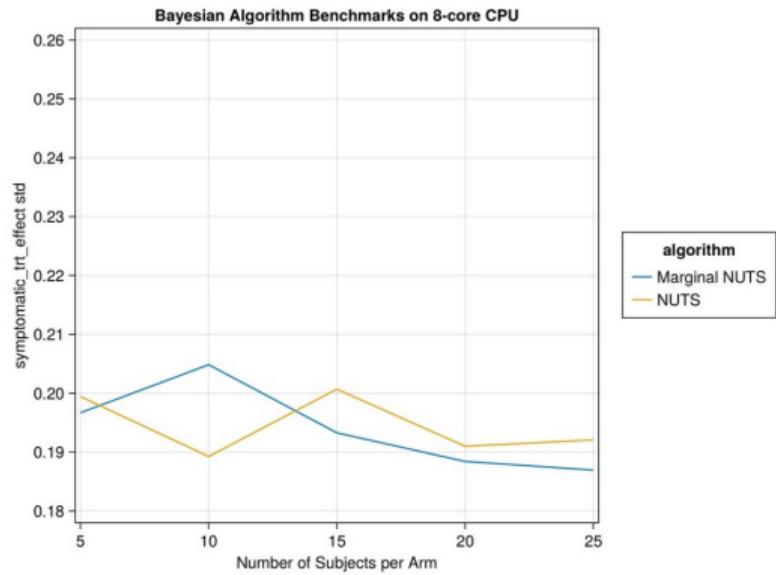
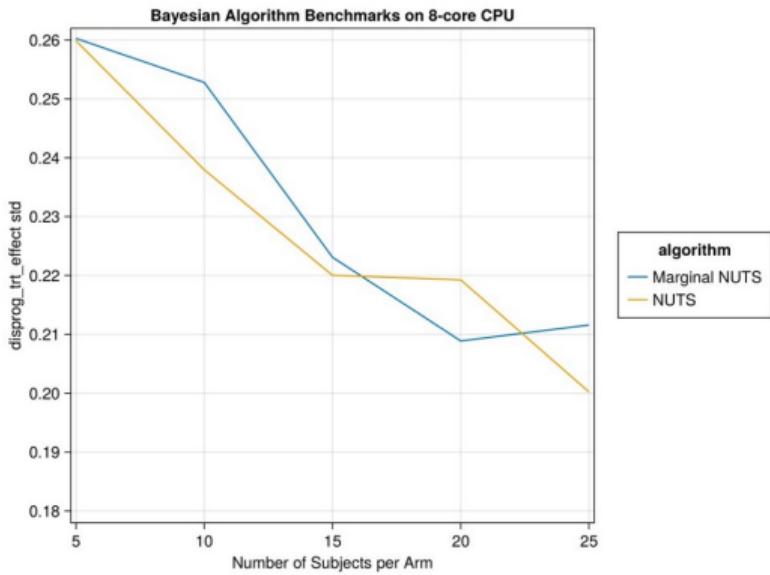
- **Question:** Can we use prior information to reduce the number of subjects needed in a rare diseases study?
- **Experiment:**
  - Given a drug model for a rare disease with long-term disease progression effects and short-term symptomatic effects.
  - Simulate 2 population arms in a study – placebo and treatment ( $\text{trt} = 0/1$ ).
  - Simulate clinical outcomes for each arm under the assumption that the drug has both long- and short-term effects.
  - Run frequentist and Bayesian analysis to infer the drug effects given the synthetic data.
  - Identify confidence and credible intervals for parameters.

# Example: Rare Disease Modelling

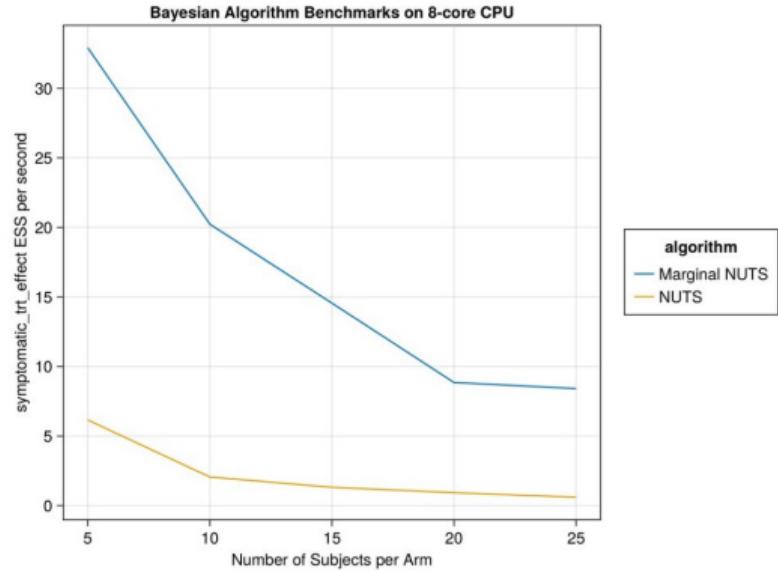
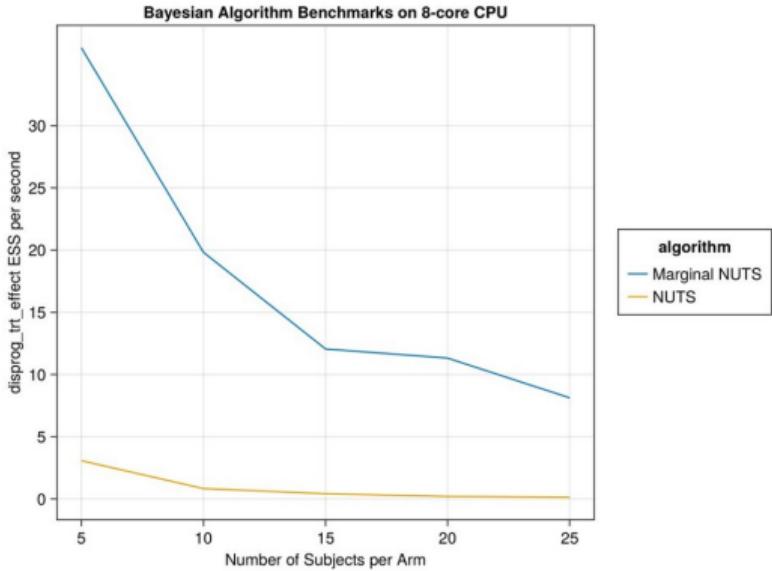
```
● ● ●

model = @model begin
    @param begin
        dispProg_placebo ~ Normal(0.0, 1.0)
        dispProg_trt_effect ~ Normal(-0.2, 0.3)
        symptomatic_placebo ~ Normal(0.0, 1.5)
        symptomatic_trt_effect ~ Normal(0.2, 0.1)
        # mean(d::Gamma) = d.α * d.θ
        # var(d::Gamma) = mean(d) * d.θ
        ω²_1 ~ Gamma(1, 0.1) # mean = 0.1, var = 0.01
        ω²_2 ~ Gamma(1, 0.3) # mean = 0.3, var = 0.09
        kdelay ~ Gamma(20, 0.1) # mean = 2.0, var = 0.2
        σ² ~ Gamma(4, 0.25) # mean = 1.0, var = 0.25
    end
    @random begin
        bsv_DisPro ~ Normal(0.0, sqrt(ω²_1))
        bsv_Symp ~ Normal(0.0, sqrt(ω²_2))
    end
    @covariates trt
    @pre begin
        slope = dispProg_placebo +
            dispProg_trt_effect * trt +
            bsv_DisPro
        symEff = symptomatic_placebo +
            symptomatic_trt_effect * trt +
            bsv_Symp
        μ = slope * t -
            symEff * (1 - exp(-kdelay * t))
    end
    @derived begin
        ClinicalEndpoint ~ @. Normal(μ, sqrt(σ²))
    end
end
```

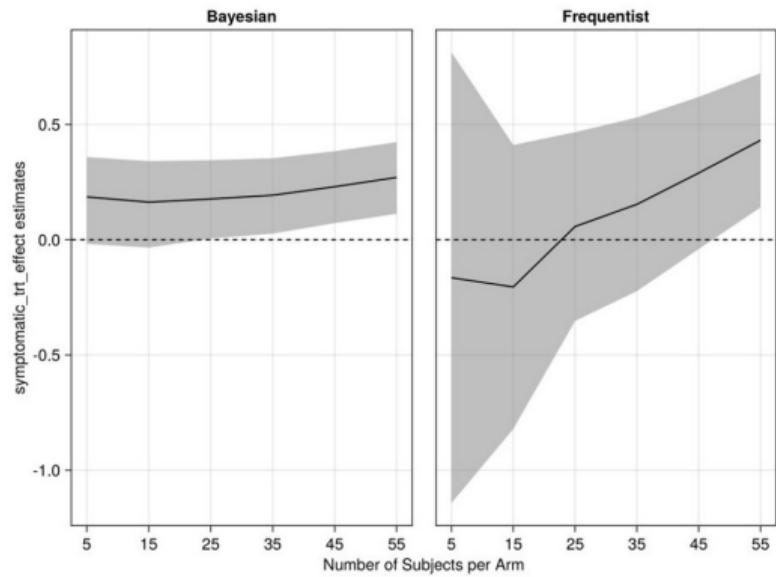
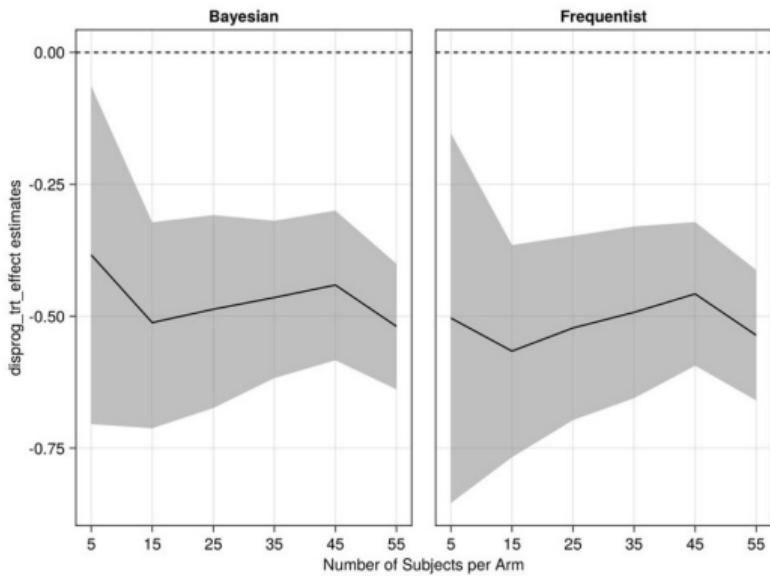
# Example: Rare Disease Modelling



# Example: Rare Disease Modelling



# Example: Rare Disease Modelling



# Convergence Diagnostics

13.1 MCMC Convergence

13.2 Diagnostic Plots

13.3 Other Diagnostics

13.4 What To Do If the Markov Chains Do Not Converge?

13.5 Is Convergence Important?

# Markov Chain Convergence

MCMC has an interesting property that it will **asymptotically converge to the target distribution**<sup>xxix</sup>.

That means, if we have all the time in the world, it is guaranteed, irrelevant of the target distribution posterior geometry, **MCMC will give you the right answer.**

However, we don't have all the time in the world. Different MCMC algorithms, like HMC and NUTS, can reduce the sampling (and warmup) time necessary for convergence to the target distribution.

---

<sup>xxix</sup>this property is not present on neural networks.

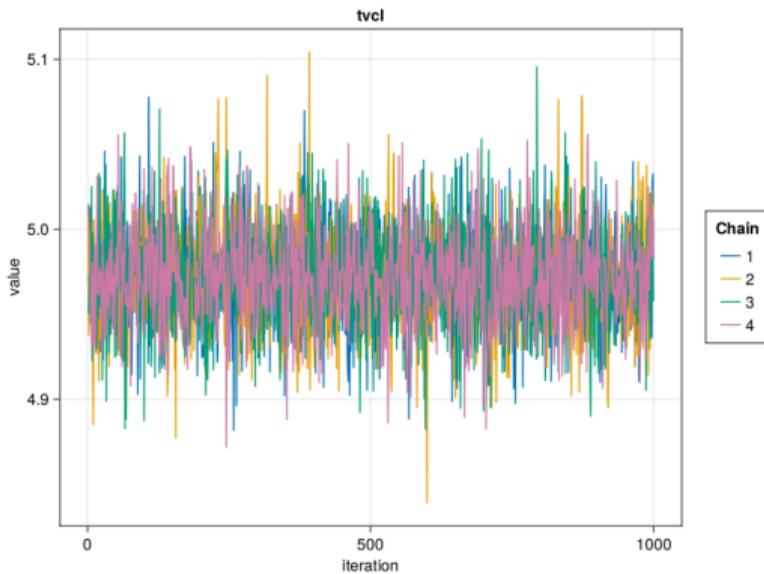
# Can We Prove Convergence?

- In the ideal scenario, the NUTS sampler converges to the true posterior and doesn't miss on any mode.
- Unfortunately, this is not easy to prove in general.
- All the convergence diagnostics are only tests for symptoms of lack of convergence.
- In other words if all the diagnostics look normal, then we can't prove that the sampler didn't converge.
- But we also can't prove that the sampler actually converged.

# Signs of Lack of Convergence

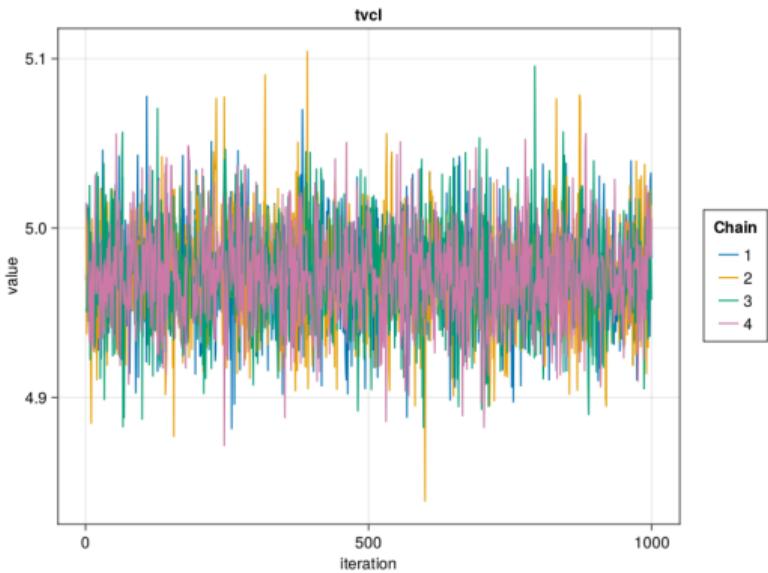
- Some signs of lack of convergence are:
  - Any of the moments (e.g. the mean or standard deviation) is changing with time. This is diagnosed using stationarity tests by comparing different parts of a single chain to each other.
  - Any of the moments is sensitive to the initial parameter values. This is diagnosed using multiple chains by comparing their summary statistics to each other.
- While high auto-correlation is not strictly a sign of lack of convergence, samplers with high auto-correlation will require many more samples to get to the same ESS as another sampler with low auto-correlation. So a low auto-correlation is usually more desirable.

# Trace Plot



The trace plot of a parameter shows the value of the parameter in each iteration of the MCMC algorithm.

# Trace Plot

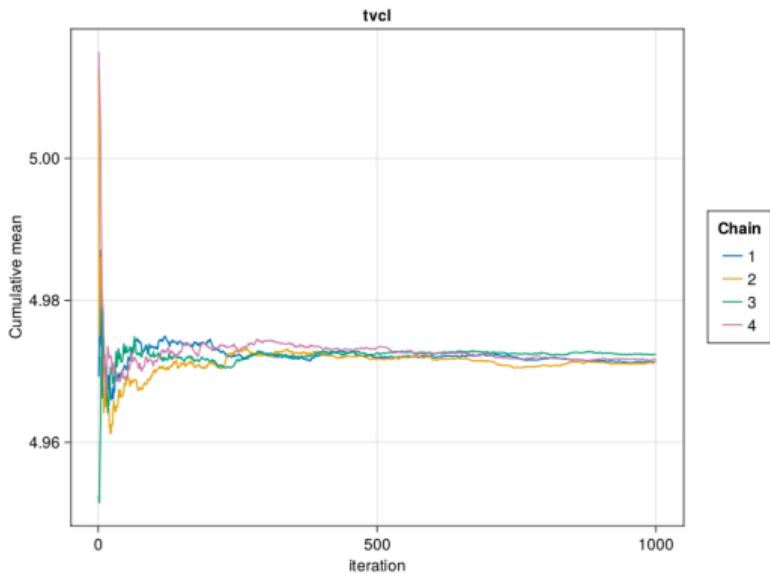


A good trace plot is one that:

- Is noisy, not an increasing or decreasing line for example.
- Has a fixed mean.
- Has a fixed variance.
- Shows all chains overlapping with each other, aka chain mixing.

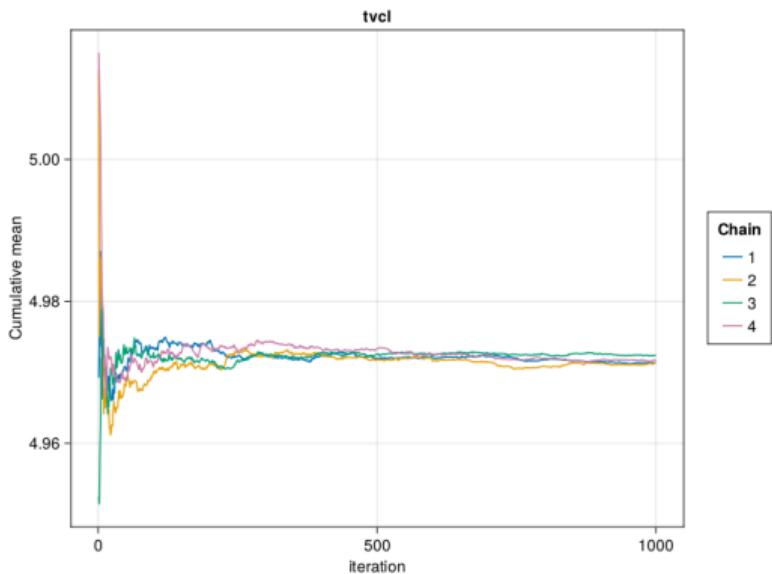
This is an example of somewhat well mixed chains that don't indicate non-convergence.

# Cumulative Mean Plot



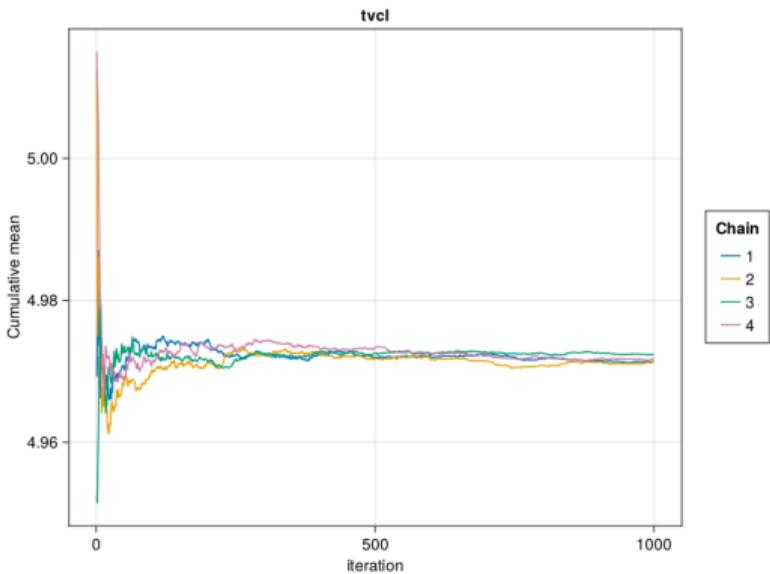
The cumulative mean plot of a parameter shows the mean of the parameter value in each MCMC chain up to a certain iteration.

# Cumulative Mean Plot



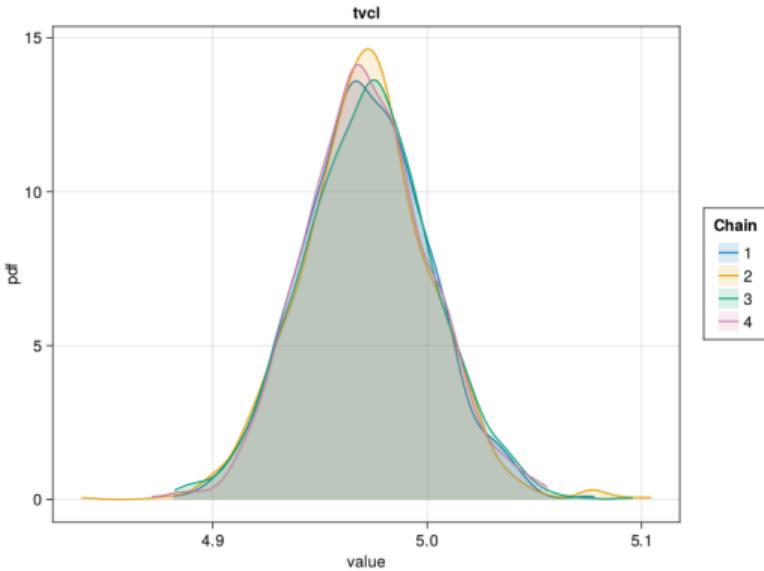
An MCMC chain converging to a stationary posterior distribution should have the cumulative mean of each parameter converge to a fixed value.

# Cumulative Mean Plot



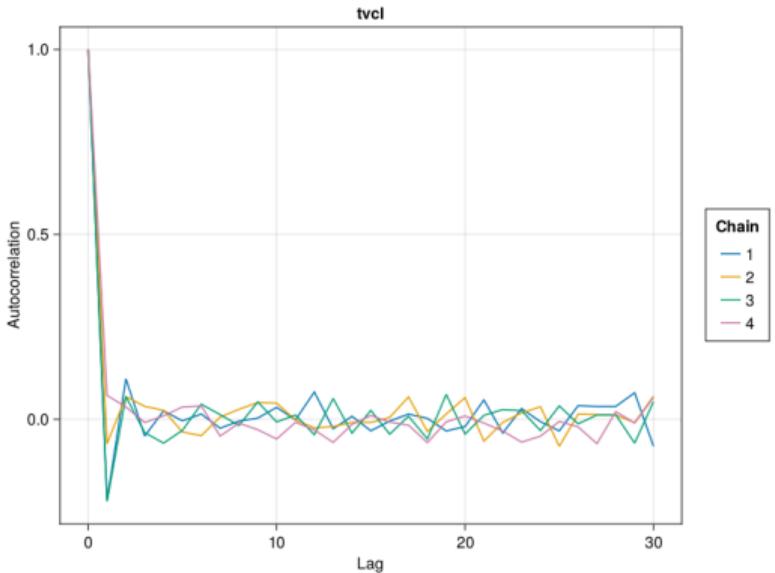
All the chains should be converging to the same mean for a given parameter, the posterior mean.

# Density Plot



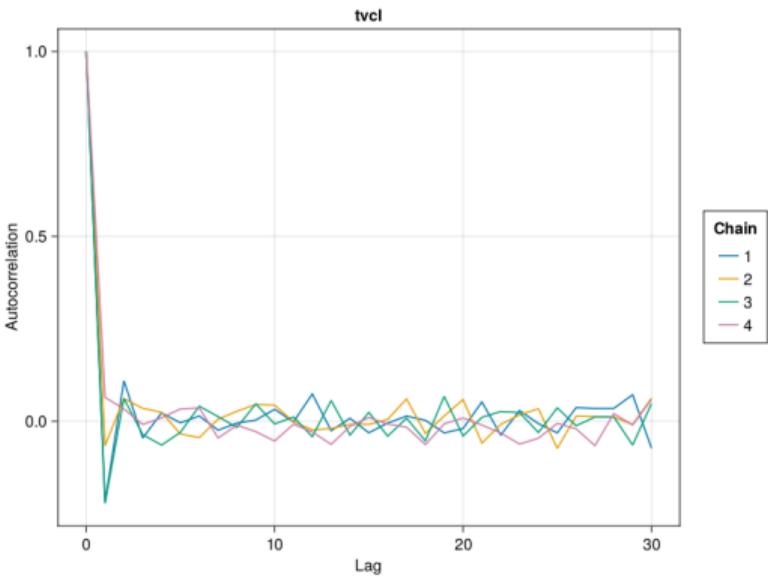
- The density plot of a parameter shows a smoothed version of the histogram of the parameter values, giving an approximate probability density function for the marginal posterior of the parameter considered.
- This helps us visualize the shape of the marginal posterior of each parameter.

# Auto-correlation Plot



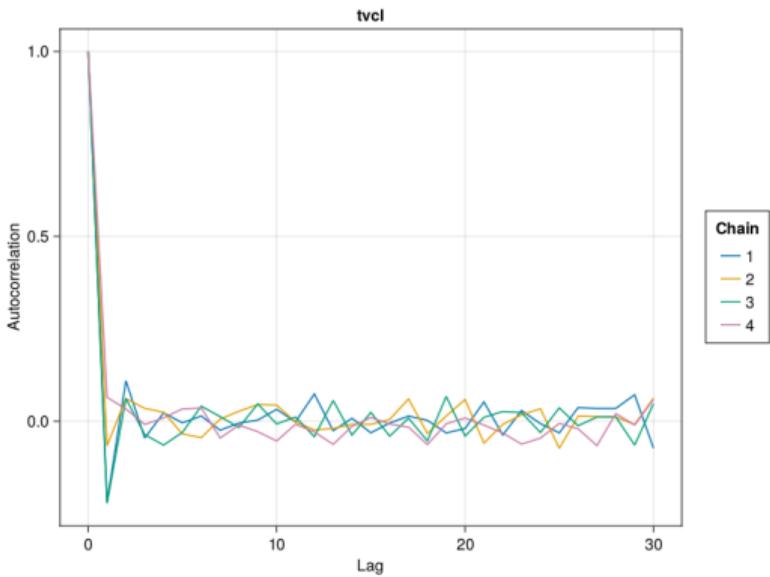
- MCMC chains are prone to auto-correlation between the samples because each sample in the chain is a function of the previous sample.
- The auto-correlation plot shows the correlation between every sample with index  $s$  and the corresponding sample with index  $s + \text{lag}$  for all  $s \in 1 : N - \text{lag}$  where  $N$  is the total number of samples.

# Auto-correlation Plot



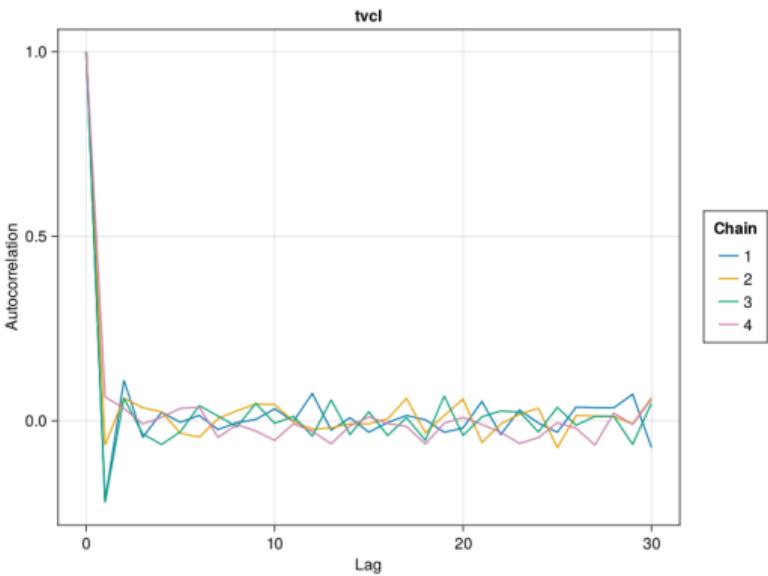
- For each value of lag, we can compute a correlation measure between the samples and their lag-steps-ahead counterparts.
- The correlation is usually a value between 0 and 1 but can sometimes be between -1 and 0 as well.

# Auto-correlation Plot



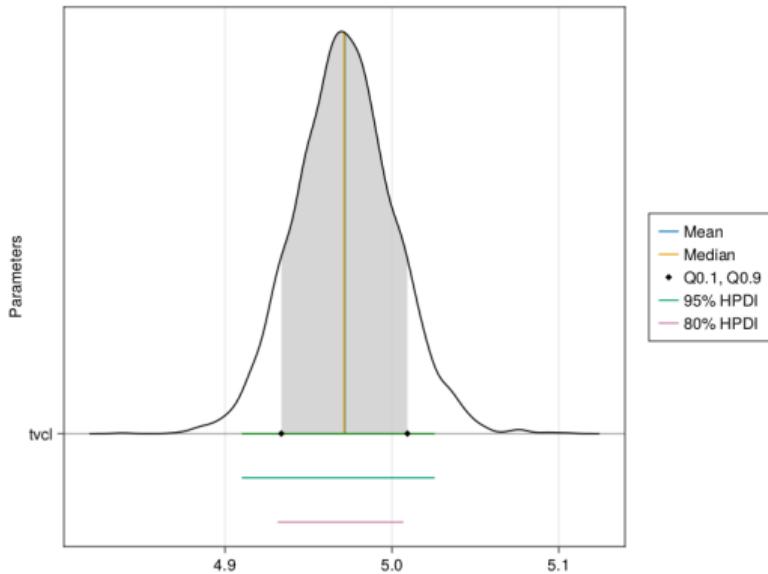
- The auto-correlation plot shows the lag on the x-axis and the correlation value on the y-axis.
- For well behaving MCMC chains when lag increases, the corresponding correlation gets closer to 0.

# Auto-correlation Plot



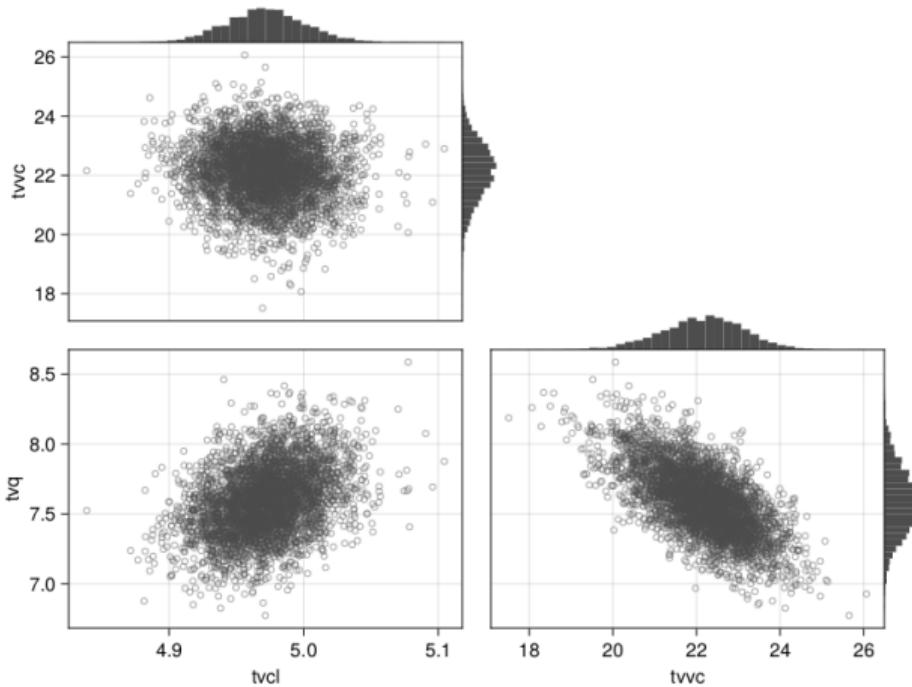
- This means that there is less and less correlation between any 2 samples further away from each other.
- The value of lag where the correlation becomes close to 0 can be used to guide the thinning of the MCMC samples to extract mostly independent samples from the auto-correlated samples.

# Ridge Line Plot



The ridge line plot shows similar information as the density plot in addition to the credible interval and quantile information.

# Corner Plot



# Convergence Metrics

There are a few metrics and diagnostics usually used to assess and diagnose the Markov chains:

- **Effective Sample Size (ESS)**: an approximation of the “number of independent samples” generated by a Markov chain.
- $\hat{R}$  (**Rhat**): potential scale reduction factor, a metric to measure if the Markov chain have mixed, and, potentially, converged.

## Effective Sample Size (Gelman et al., 2013b)

$$\hat{n}_{\text{eff}} = \frac{mn}{1 + \sum_{t=1}^T \hat{\rho}_t}$$

Where:

- $m$ : number of Markov chains.
- $n$ : total samples per Markov chain (discarding warmup).
- $\hat{\rho}_t$ : an autocorrelation estimate.

## Rhat (Gelman et al., 2013b)

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\psi | y)}{W}}$$

where  $\widehat{\text{var}}^+(\psi | y)$  is the Markov chains' sample variance for a certain parameter  $\psi$ . We calculate it by using a weighted sum of the within-chain  $W$  and between-chain  $B$  variances:

$$\widehat{\text{var}}^+(\psi | y) = \frac{n-1}{n}W + \frac{1}{n}B$$

Intuitively, the value is 1.0 if all chains are totally convergent. As a heuristic, if  $\widehat{R} > 1.1$ , you need to worry because probably the chains have not converged adequately.

# Geweke Diagnostic

- The Geweke diagnostic compares the sample means of two disjoint sub-chains  $X_1$  and  $X_2$  of the entire chain.
- It uses a difference of means hypothesis test where the null and alternative hypotheses are:
  - $H_0 : \mu_1 = \mu_2$
  - $H_1 : \mu_1 \neq \mu_2$

where  $\mu_1$  and  $\mu_2$  are the means of  $X_1$  and  $X_2$  respectively.

# Geweke Diagnostic

- The first sub-chain  $X_1$  is taken as the first  $(\text{first} \times 100)\%$  of the samples in the chain. The second sub-chain  $X_2$  is taken as the last  $(\text{last} \times 100)\%$  of the samples in the chain.
- The test statistic used is:  $z_0 = (\bar{x}_1 - \bar{x}_2) / \sqrt{s_1^2 + s_2^2}$  where  $\bar{x}_1$  and  $\bar{x}_2$  are the sample means of  $X_1$  and  $X_2$  respectively, and  $s_1$  and  $s_2$  are the Markov Chain standard error (MCSE) estimates of  $X_1$  and  $X_2$  respectively.
- Auto-correlation is assumed within the samples of each individual sub-chain, but the samples in  $X_1$  are assumed to be independent of the samples in  $X_2$ .

# Geweke Diagnostic

- The p-value output is an estimate of  $P(|z| > |z_0|)$ , where  $z$  is a standard normally distributed random variable.
- Low p-values indicate one of the following:
  - The first and last parts of the chain are sampled from distributions with different means, i.e. non-convergence,
  - The need to discard some initial samples as burn-in, or
  - The need to run the sampling for longer due to lack of samples or high auto-correlation.
- High p-values (desirable) indicate the inability to conclude that the means of the first and last parts of the chain are different with statistical significance.

# Geweke Diagnostic

- However, this alone does not guarantee convergence to a fixed posterior distribution because:
  - Either the standard deviations or higher moments of  $X_1$  and  $X_2$  may be different, or
  - The independence assumption between  $X_1$  and  $X_2$  may not be satisfied when high auto-correlation exists.

# Heidelberger and Welch Diagnostic

The Heidelberger diagnostic attempts to:

1. Identify a cutoff point for the initial transient phase for each parameter, after which the samples can be assumed to come from a steady-state posterior distribution. Can be treated as burn-in.
2. Estimate the relative confidence interval for the mean of the steady-state posterior distribution of each parameter, assuming such steady-state distribution exists in the samples. A large confidence interval implies either the lack of convergence to a stationary distribution or lack of samples.
3. Quantify the extent to which the distribution of the samples is stationary using a hypothesis test. The returned p-value can be considered a measure of mean stationarity. A p-value lower than  $\alpha$  (e.g.  $\alpha = 0.05$ ) implies lack of stationarity of the mean.

# Heidelberger and Welch Diagnostic

The Heidelberger diagnostic only tests for the mean of the distribution. Therefore, it can only be used to detect lack of convergence and not to prove convergence. In other words, even if all the numbers seem normal, one cannot conclude that the chain converged to a stationary distribution or that it converged to the true posterior.

# What To Do If the Markov Chains Do Not Converge?

*When you have computational problems, often there's a problem with your model.*

Gelman (2008) (Folk Theorem)

# What To Do If the Markov Chains Do Not Converge?

- Dynamics-based models with complicated stiff differential equations often suffer from sensitivity to parameter values in 2 ways:
  - First, small changes in the parameter values may lead to extremely different dynamics and wrong predictions thus leading to rejections.
  - And second, changes in the parameter values may make the differential equation highly stiff thus slowing down convergence or even causing divergence of the solver.
- In other words, MCMC for some complicated models can often run slow and fail to give good ESS values at the end.
- In such cases, NUTS may not always be computationally feasible.
- But you can try any of the following remedies and workarounds to poke at the model.

# What To Do If the Markov Chains Do Not Converge?

- Lower the target acceptance ratio. This may alleviate the need for a small step size and a full tree exploration.
- Re-parameterize your model to have less parameter dependence.
- Fix some parameter values to known good values, e.g. values obtained by maximum-a-posteriori (MAP) optimization.
- Initialize the sampling from good parameter values.
- Use a stronger prior around suspected good parameter values.
- Simplify your model, e.g. using simpler dynamics.
- Try the marginal MCMC algorithm MarginalMCMC instead of the full joint MCMC algorithm BayesMCMC.

# What To Do If the Markov Chains Do Not Converge?

- If you find the sampler regularly hitting the maximum tree depth of 10 in the initial exploration phase, it might make sense to decrease that initially to have quicker iterations when in the exploration phase of the study.
- This is effectively limiting the level of exploration in the sampling so it might make sense to use good initial values when doing this.
- However in the final phase of the study, it is best to make sure that the maximum tree depth is not reached by the sampler (increasing it if necessary).
- This might also slow down your sampling significantly so there can be a tradeoff here.
- It's also best to ensure that the sampler converges to the posterior when starting from multiple different random initial points using different chains.

# Is Convergence Important?

- Since we can't prove that the sampler explored the full posterior in general, is exploring the full posterior always absolutely necessary?
- That depends on what you want to do. If you are trying to answer questions about the parameters, e.g. estimating the probability that an effect is greater than or less than 0 for a go/no-go decision, then you need your sampler to sample from the true posterior.
- Of course, we cannot prove this in general anyways but you should generally follow all the best practices and you should not ignore signs of lack of convergence.

# Is Convergence Important?

- Some bad signs to watch out for if you want to sample from the true posterior are:
  - Non-stationarity of the samples' distribution
  - Dependence of the samples' distribution on the initial parameters after the adaptation steps
  - High auto-correlation in the samples after the adaptation steps
  - Too many rejections and ODE solver divergences
  - Low ESS values relative to the number of samples
  - Extremely small step sizes and hitting the maximum tree depth often

# Is Convergence Important?

- On the other hand, if your goal is not to answer questions about the parameters but only to make predictions using the posterior predictive distribution as an ensemble of predictions, then sampling from the true posterior may not be strictly necessary in this case.
- If the posterior predictive distribution gives enough accuracy and uncertainty in the predictions to reflect the uncertainty in the unseen data, then that may suffice and we can live with some imperfections in the sampling.

# Is Convergence Important?

- Some imperfections in the sampling include:
  - Having to initialize the sampler from a mode to get the sampler to work
  - Using a low maximum tree depth and allowing the maximum to be reached
  - Using a high target acceptance ratio to decrease exploration and sample around a mode
  - High auto-correlation in the samples even after the adaptation steps and low relative ESS
  - Using a few adaptation steps 'nадапts'

# Is Convergence Important?

- If doing any or all of the above resulted in fast sampling that gives a good enough posterior predictive distribution but potentially bad posterior exploration and if predictions are what you care the most about then perhaps you don't need to sample from the true posterior in your use case.
- Such an imperfect solution is often satisfactory in the context of Bayesian neural networks for example where parameters are generally meaningless and we only care about predictions.
- Of course, we don't advocate for doing this in general since this goes against the best practices of MCMC but it's an option you have.

# Model Comparison

14.1 Model Comparison - Recommended References

14.2 Why Compare Models?

14.3 Model Comparison Techniques

14.4 Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

# Model Comparison - Recommended References

- Gelman et al. (2013b) - Chapter 7: Evaluating, comparing, and expanding models
- Gelman, Hill, and Vehtari (2020) - Chapter 11, Section 11.8: Cross validation
- McElreath (2020) - Chapter 7, Section 7.5: Model comparison
- Vehtari et al. (2015)
- Spiegelhalter et al. (2002)
- Van Der Linde (2005)
- Watanabe and Opper (2010)
- Gelfand (1996)
- Watanabe and Opper (2010)
- Geisser and Eddy (1979)

# Why Compare Models?

After model parameters estimation, many times we want to measure its **predictive accuracy** by itself, or for **model comparison**, **model selection**, or computing a **model performance metric** (Geisser & Eddy, 1979).

## But What About VPCs?

To analyze and compare models VPCs is a **subjective and arbitrary approach**.

There is an **objective approach to compare Bayesian models** which uses a robust metric that helps us select the best model in a set of candidate models.

Having an objective way of comparing and choosing the best model is very important. In the **Bayesian workflow**, we generally have several iterations between priors and likelihood functions resulting in several different models (Gelman, Vehtari, et al., [2020](#)).

# Model Comparison Techniques

We have several model comparison techniques that use **predictive accuracy**, but the main ones are:

- Leave-one-out cross-validation (LOO) (Vehtari et al., 2015).
- Deviance Information Criterion (DIC) (Spiegelhalter et al., 2002), but it is known to have some issues, due to not being full-Bayesian, because it is only based on point estimates (Van Der Linde, 2005),
- Widely Applicable Information Criteria (WAIC) (Watanabe & Opper, 2010), full-Bayesian, in the sense that uses the full posterior distribution density, and it is asymptotically equal to LOO (Vehtari et al., 2015).

# Historical Interlude

In the past, we did not have computational power and data abundance. Model comparison was done based on a theoretical divergence metric originated from information theory's entropy:

$$H(p) = -E \log(p_i) = -\sum_{i=1}^N p_i \log(p_i)$$

We compute the divergence by multiplying entropy by  $-2^{xxx}$ , so lower values are preferable:

$$D(y, \theta) = -2 \cdot \underbrace{\sum_{i=1}^N \log \frac{1}{S} \sum_{s=1}^S P(y_i | \theta^s)}_{\text{log pointwise predictive density - lppd}}$$

where  $n$  is the sample size and  $S$  is the number of posterior draws.

---

<sup>xxx</sup>historical reasons.

## Historial Interlude – AIC (Akaike, 1973)

$$AIC = D(y, \theta) + 2k = -2\text{Ippd}_{\text{mle}} + 2k$$

where  $k$  is the number of the model's free parameters and  $\text{Ippd}_{\text{mle}}$  is the **maximum likelihood estimate of the log pointwise predictive density**.

AIC is an approximation and can only be reliable when:

- The priors are uniform (flat priors) or totally dominated by the likelihood function.
- The posterior is approximate a multivariate normal distribution.
- The sample size  $N$  is much larger than the number of the model's free parameters  $k$ :  $N \gg k$

## Historical Interlude – DIC (Spiegelhalter et al., 2002)

A generalization of the AIC, where we replace the maximum likelihood estimate for the posterior mean and  $k$  by a data-based bias correction:

$$\text{DIC} = D(y, \theta) + k_{\text{DIC}} = -2\text{lppd}_{\text{Bayes}} + 2 \underbrace{\left( \text{lppd}_{\text{Bayes}} - \frac{1}{S} \sum_{s=1}^S \log P(y | \theta^s) \right)}_{\text{bias-corrected } k}$$

DIC removes the restriction on uniform AIC priors, but still keeps the assumptions of the posterior being a multivariate Gaussian/normal distribution and that  $N \gg k$ .

# Predictive Accuracy

With current computational power, we do not need approximations<sup>xxxii</sup>.

We can discuss **predictive accuracy objective metrics**

But, first, let's define what is predictive accuracy.

---

<sup>xxxii</sup>AIC, DIC etc.

# Predictive Accuracy

## Definition (Predictive Accuracy)

*Bayesian approaches measure predictive accuracy using posterior draws  $\tilde{y}$  from the model. For that we have the predictive posterior distribution:*

$$p(\tilde{y} | y) = \int p(\tilde{y}_i | \theta)p(\theta | y)d\theta$$

*Where  $p(\theta | y)$  is the model's posterior distribution. The above equation means that we evaluate the integral with respect to the whole joint probability of the model's predictive posterior distribution and posterior distribution.*

*The **higher** the predictive posterior distribution  $p(\tilde{y} | y)$ , the **better** will be the model's predictive accuracy.*

# Predictive Accuracy

To make samples comparable, we calculate the expectation of this measure for each one of the  $N$  sample observations:

$$\text{elpd} = \sum_{i=1}^N \int p_t(\tilde{y}_i) \log p(\tilde{y}_i | y) d\tilde{y}$$

where elpd is the **expected log pointwise predictive density**, and  $p_t(\tilde{y}_i)$  is the distribution that represents the  $\tilde{y}_i$ 's true underlying data generating process. The  $p_t(\tilde{y}_i)$  are unknown and we generally use cross-validation or approximations to estimate elpd.

# Leave-One-Out Cross-Validation (LOO)

We can compute the elpd using LOO (Vehtari et al., 2015):

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^N \log p(y_i | y_{-i})$$

where

$$p(y_i | y_{-i}) = \int p(y_i | \theta) p(\theta | y_{-i}) d\theta$$

which is the predictive density conditioned on the data without a single observation  $i$  ( $y_{-i}$ ). Almost always we use the PSIS-LOO approximation due to its robustness and low computational cost.

# Widely Applicable Information Criteria (WAIC)

WAIC (Watanabe & Opper, 2010), like LOO, is also an alternative approach to compute the elpd, and is defined as:

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lppd}} - \widehat{p}_{\text{waic}}$$

where  $\widehat{p}_{\text{waic}}$  is the number of effective parameters based on:

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^N \text{var}_{\text{post}}(\log p(y_i | \theta))$$

which we can compute using the posterior variance of the log predictive density for each observation  $y_i$ :

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^N V_{s=1}^S (\log p(y_i | \theta^s))$$

where  $V_{s=1}^S$  is the sample's variance:

$$V_{s=1}^S a_s = \frac{1}{S-1} \sum_{s=1}^S (a_s - \bar{a})^2$$

## *K*-fold Cross-Validation (*K*-fold CV)

In the same manner that we can compute the elpd using LOO with  $N - 1$  sample partitions, we can also compute it with any desired partition number.

Such approach is called ***K*-fold cross-validation** (*K*-fold CV).

Contrary to LOO, we cannot approximate the actual elpd using *K*-fold CV, and we need to compute the actual elpd over *K* partitions.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

PSIS uses **importance sampling**<sup>xxxii</sup>, which means a importance weighting scheme approach.

The **Pareto smoothing** is a technique to increase the importance weights' reliability.

---

<sup>xxxii</sup>another class of MCMC algorithm that we did not cover yet.

# Importance Sampling

If the  $N$  samples are conditionally independent<sup>xxxiii</sup> (Gelfand et al., 1992), we can compute LOO with  $\theta^s$  posterior' samples  $P(\theta | y)$  using **importance weights**:

$$r_i^s = \frac{1}{P(y_i | \theta^s)} \propto \frac{P(\theta^s | y_{-i})}{P(\theta^s | y)}$$

Hence, to get Importance Sampling Leave-One-Out (IS-LOO):

$$P(\tilde{y}_i | y_{-i}) \approx \frac{\sum_{s=1}^S r_i^s P(\tilde{y}_i | \theta^s)}{\sum_{s=1}^S r_i^s}$$

---

<sup>xxxiii</sup>that is, they are independent if conditioned on the model's parameters, which is a basic assumption in any Bayesian (and frequentist) model

# Importance Sampling

However, the posterior  $P(\theta | y)$  often has low variance and shorter tails than the LOO distributions  $P(\theta | y_{-1})$ . Hence, if we use:

$$P(\tilde{y}_i | y_{-i}) \approx \frac{\sum_{s=1}^S r_i^s P(\tilde{y}_i | \theta^s)}{\sum_{s=1}^S r_i^s}$$

we will have **instabilities** because the  $r_i$  can have **high, or even infinite, variance**.

# Pareto Smoothed Importance Sampling

We can enhance the IS-LOO estimate using a **Pareto Smoothed Importance Sampling** (Vehtari et al., 2015).

When the tails of the importance weights' distribution are long, a direct usage of the importance is sensible to one or more large value. By **fitting a generalized Pareto distribution to the importance weights' upper-tail**, we smooth out these values.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

Finally, we have PSIS-LOO:

$$\widehat{\text{elpd}}_{\text{psis-loo}} = \sum_{i=1}^n \log \left( \frac{\sum_{s=1}^S w_i^s P(y_i | \theta^s)}{\sum_{s=1}^S w_i^s} \right)$$

where  $w$  is the truncated weights.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

We use the importance weights Pareto distribution's estimated shape parameter  $\hat{k}$  to assess its reliability:

- $k < \frac{1}{2}$ : the importance weights variance is finite, the central limit theorem holds, and the estimate rapidly converges.
- $\frac{1}{2} < k < 1$  the importance weights variance is infinite, but the mean exists (is finite), the generalized central limit theorem for stable distributions holds, and the estimate converges, but slower. The PSIS variance estimate is finite, but could be large.
- $k > 1$  both the importance weights variance and mean do not exist (they are infinite). The PSIS variance estimate is finite, but could be large.

Any  $\hat{k} > 0.5$  is a warning sign, but empirically there is still a good performance up to  $\hat{k} < 0.7$ .

# References I

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *Second international symposium on information theory* (pp. 266–281).
- Bertsekas, D. P., & Tsitsiklis, J. N. (2008, July 15). *Introduction to probability, 2nd edition* (2nd edition). Athena Scientific.
- Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., & Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19, 1501–1534. <https://doi.org/10.3150/12-BEJ414>
- Betancourt, M. (2017, January 9). *A Conceptual Introduction to Hamiltonian Monte Carlo*. arXiv: [1701.02434](https://arxiv.org/abs/1701.02434). Retrieved November 6, 2019, from <http://arxiv.org/abs/1701.02434>
- Betancourt, M. (2019, June). *Probabilistic building blocks* [Beta & alpha]. Retrieved May 27, 2021, from [https://betanalpha.github.io/assets/case\\_studies/probability\\_densities.html](https://betanalpha.github.io/assets/case_studies/probability_densities.html)
- Bonate, P. L. (2011). *Pharmacokinetic-pharmacodynamic modeling and simulation*. Springer US. <https://doi.org/10.1007/978-1-4419-9485-1>
- Brooks, S., Gelman, A., Jones, G., & Meng, X.-L. (2011, May 10). *Handbook of Markov Chain Monte Carlo*. CRC Press.
- Casella, G., & George, E. (1992). Explaining the gibbs sampler [Publisher: Taylor & Francis \_eprint: <https://www.tandfonline.com/doi/pdf/10.1080/00031305.1992.10475878>]. *The American Statistician*, 46(3), 167–174. <https://doi.org/10.1080/00031305.1992.10475878>

## References II

- Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4), 327–335. <https://doi.org/10.1080/00031305.1995.10476177>
- de Finetti, B. (1974). *Theory of Probability* (Volume 1). John Wiley & Sons.
- Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., & Meester, L. E. (2010, October 19). *A modern introduction to probability and statistics: Understanding why and how*. Springer.
- Diaconis, P., & Skyrms, B. (2019, October 8). *Ten great ideas about chance* [Google-Books-ID: 68iXDwAAQBAJ]. Princeton University Press.
- Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2), 216–222. [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
- Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo Method. *Los Alamos Science*, 15(30), 131–136.
- Fisher, R. A. (1925). *Statistical methods for research workers*. Oliver; Boyd.
- Fisher, R. A. (1962). Some Examples of Bayes' Method of the Experimental Determination of Probabilities A Priori. *Journal of the Royal Statistical Society. Series B (Methodological)*, 24(1), 118–124.
- Gabrielsson, J., Weiner, D., & Weiner, D. (2006). *Pharmacokinetic & pharmacodynamic data analysis: Concepts and applications* (4. ed., rev.expanded). Apotekarsocieteten, Swedish Academy of Pharmaceutical Sciences.

# References III

- Geisser, S., & Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365), 153–160.
- Gelfand, A. E., Dey, D. K., & Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), *Bayesian statistics* (pp. 147–167). Oxford University Press.
- Gelfand, A. E. (1996). Model determination using sampling-based methods. *Markov chain Monte Carlo in practice*, 145–161.
- Gelman, A. (2008). *The folk theorem of statistical computing*.  
[https://statmodeling.stat.columbia.edu/2008/05/13/the\\_folk\\_theore/](https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/)
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013a). Basics of Markov Chain Simulation. *Bayesian Data Analysis*. Chapman and Hall/CRC.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013b). *Bayesian Data Analysis*. Chapman and Hall/CRC.
- Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
- Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press.

# References IV

- Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020, November 3). *Bayesian Workflow*. arXiv: [2011.01808 \[stat\]](https://arxiv.org/abs/2011.01808). Retrieved February 4, 2021, from <http://arxiv.org/abs/2011.01808>
- Grimmett, G., & Stirzaker, D. (2020). *Probability and Random Processes: Fourth Edition* (Fourth Edition, New to this Edition:). Oxford University Press.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97–109. <https://doi.org/10.1093/biomet/57.1.97>
- Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., & Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLoS Biol*, 13(3), e1002106.
- Hoffman, M. D., & Gelman, A. (2011). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1), 1593–1623. <http://arxiv.org/abs/1111.4246>
- Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press.
- Khan, M. E., & Rue, H. (2021, July 9). *The Bayesian Learning Rule*. arXiv: [2107.04562 \[cs, stat\]](https://arxiv.org/abs/2107.04562). Retrieved July 13, 2021, from <http://arxiv.org/abs/2107.04562>
- Kolmogorov, A. N. (1933). *Foundations of the Theory of Probability*. Julius Springer.

# References V

- Kruschke, J. K., & Vanpaemel, W. (2015). Bayesian estimation in hierarchical models. In J. R. Busemeyer, Z. Wang, J. T. Townsend, & A. Eidels (Eds.), *The Oxford handbook of computational and mathematical psychology* (pp. 279–299). Oxford University Press Oxford, UK.
- Kurt, W. (2019, July 9). *Bayesian statistics the fun way: Understanding statistics and probability with star wars, LEGO, and rubber ducks* (Illustrated edition). No Starch Press.
- Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100(9), 1989–2001.
- Margossian, C. C., Zhang, Y., & Gillespie, W. R. (2022). Flexible and efficient bayesian pharmacometrics modeling using stan and torsten, part i. *CPT: Pharmacometrics & Systems Pharmacology*, psp4.12812. <https://doi.org/10.1002/psp4.12812>
- McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC press.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6), 1087–1092. <https://doi.org/10.1063/1.1699114>
- Neal, R. M. (1994). An Improved Acceptance Procedure for the Hybrid Monte Carlo Algorithm. *Journal of Computational Physics*, 111(1), 194–203. <https://doi.org/10.1006/jcph.1994.1054>
- Neal, R. M. (2003). Slice Sampling. *The Annals of Statistics*, 31(3), 705–741.

# References VI

- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of markov chain monte carlo*.
- Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1), 221–259.
- Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767), 333–380.
- Owen, J. S., & Fiedler-Kelly, J. (2014). *Introduction to population pharmacokinetic/pharmacodynamic analysis with nonlinear mixed effects models*. Wiley.
- Rackauckas, C., Ma, Y., Noack, A., Dixit, V., Mogensen, P. K., Byrne, S., Maddhashiya, S., Santiago Calderón, J. B., Nyberg, J., Gobburu, J. V., et al. (2020). Accelerated predictive healthcare analytics with pumas, a high performance pharmaceutical modeling and simulation platform.
- Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7(1), 110–120.  
<https://doi.org/10.1214/aoap/1034625254>
- Rosnow, R. L., & Rosenthal, R. (1989). Statistical procedures and the justification of knowledge in psychological science. *American Psychologist*, 44, 1276–1284.

# References VII

- Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the royal statistical society: Series b (statistical methodology)*, 64(4), 583–639.
- van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J., & Yau, C. (2021). Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1), 1–26. <https://doi.org/10.1038/s43586-020-00001-2>
- Van Der Linde, A. (2005). Dic in variable selection. *Statistica Neerlandica*, 59(1), 45–56.
- Vehtari, A., Gelman, A., & Gabry, J. (2015, July 16). *Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC*. arXiv: 1507.04544. <https://doi.org/10.1007/s11222-016-9696-4>  
1221 citations (Semantic Scholar/DOI) [2021-02-13] 1221 citations (Semantic Scholar/arXiv) [2021-02-13]
- Watanabe, S., & Opper, M. (2010). Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory.. *Journal of machine learning research*, 11(12).

# Backup Slides

# Probability Distributions - Recommended References

- Grimmett and Stirzaker (2020)
  - Chapter 3: Discrete random variables
  - Chapter 4: Continuous random variables
- Dekking et al. (2010)
  - Chapter 4: Discrete random variables
  - Chapter 5: Continuous random variables
- Betancourt (2019)

# Probability Distributions

Bayesian statistics uses probability distributions as the inference engine of the parameter and uncertainty estimates.

Imagine that probability distributions are small “Lego” pieces. We can construct anything we want with these little pieces. We can make a castle, a house, a city; literally anything. The same is valid for Bayesian statistical models. We can construct models from the simplest ones to the most complex using probability distributions and their relationships.

## Definition (Probability Distribution Function)

*A probability distribution function is a mathematical function that outputs the probabilities for different results of an experiment. It is a mathematical description of a random phenomena in terms of its sample space and the event probabilities (subsets of the sample space).*

$$P(X) : X \rightarrow \mathbb{R} \in [0, 1]$$

*For discrete random variables, we define as “mass”, and for continuous random variables, we define as “density”.*

# Mathematical Notation

We use the notation

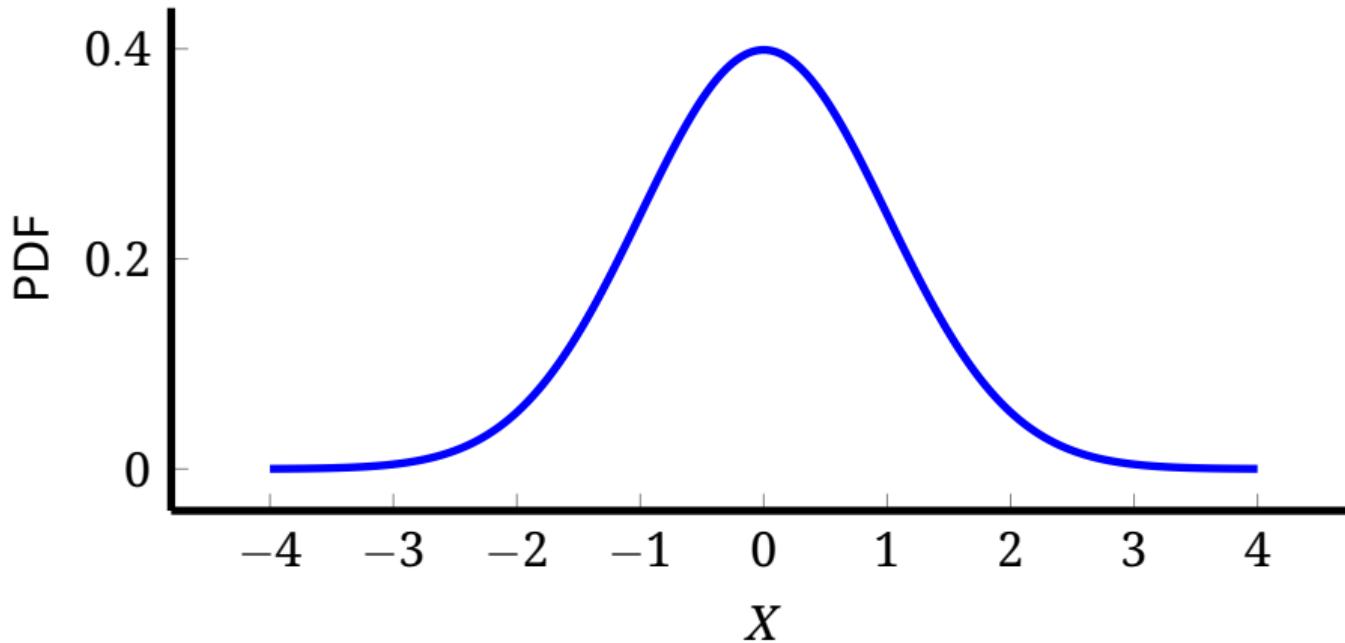
$$X \sim \text{Dist}(\theta_1, \theta_2, \dots)$$

where:

- $X$ : random variable
- $\text{Dist}$ : distribution name
- $\theta_1, \theta_2, \dots$ : parameters that define how the distribution behaves

Every probability distribution can be “parameterized” by specifying parameters that allow to control certain distribution aspects for a specific goal.

# Probability Distribution Function

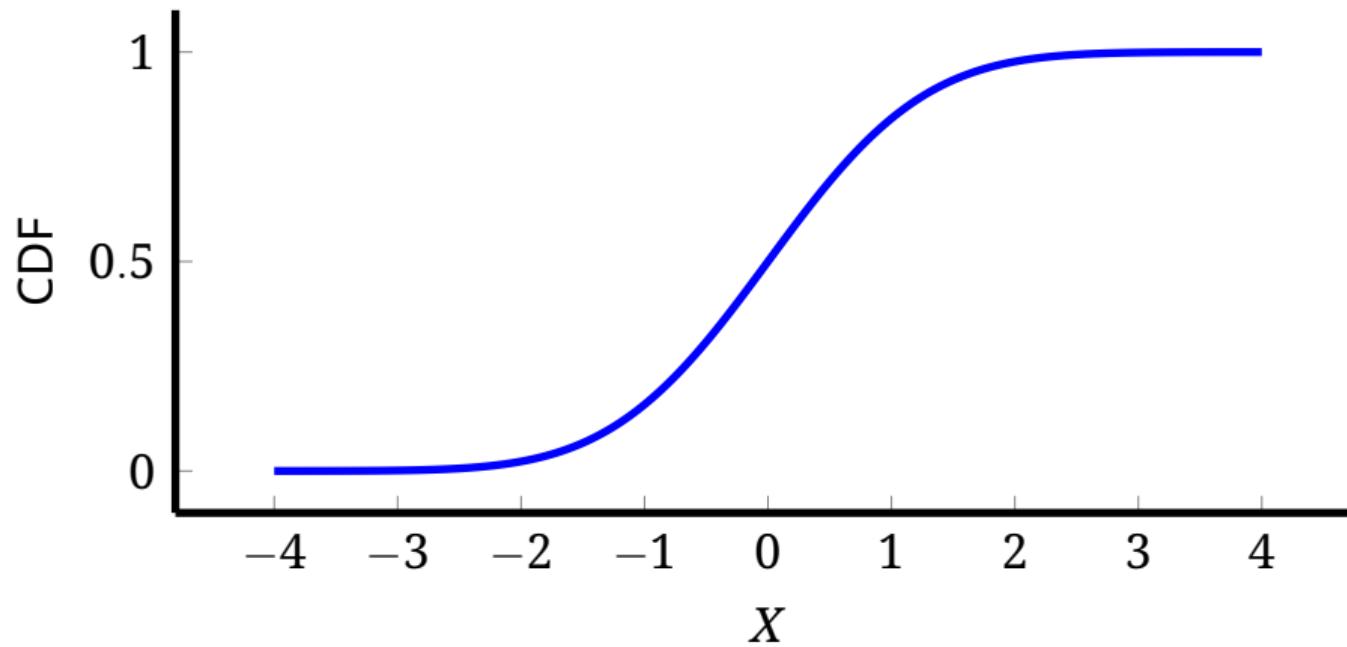


## Definition (Cumulative Distribution Function)

*The cumulative distribution function (CDF) of a random variable  $X$  evaluated at  $x$  is the probability that  $X$  will take values less or equal than  $x$ :*

$$CDF = P(X \leq x)$$

# Cumulative Distribution Function



## Definition (Discrete Distributions)

*Discrete probability distributions are distributions which the results are a discrete number:  $-N, \dots, -2, 1, 0, 1, 2, \dots, N$  where  $N \in \mathbb{Z}$ . In discrete probability distributions we call the probability of a distribution taking certain values as “mass”. The probability mass function (PMF) is the function that specifies the probability of a random variable  $X$  taking value  $x$ :*

$$\text{PMF}(x) = P(X = x)$$

# Discrete Uniform

The discrete uniform is a symmetric probability distribution in which a finite number of values are equally likely of being observable. Each one of the  $n$  values have probability  $\frac{1}{n}$ .

The uniform discrete distribution has two parameters and its notation is  $\text{Uniform}(a, b)$ :

- $a$  - lower bound
- $b$  - upper bound

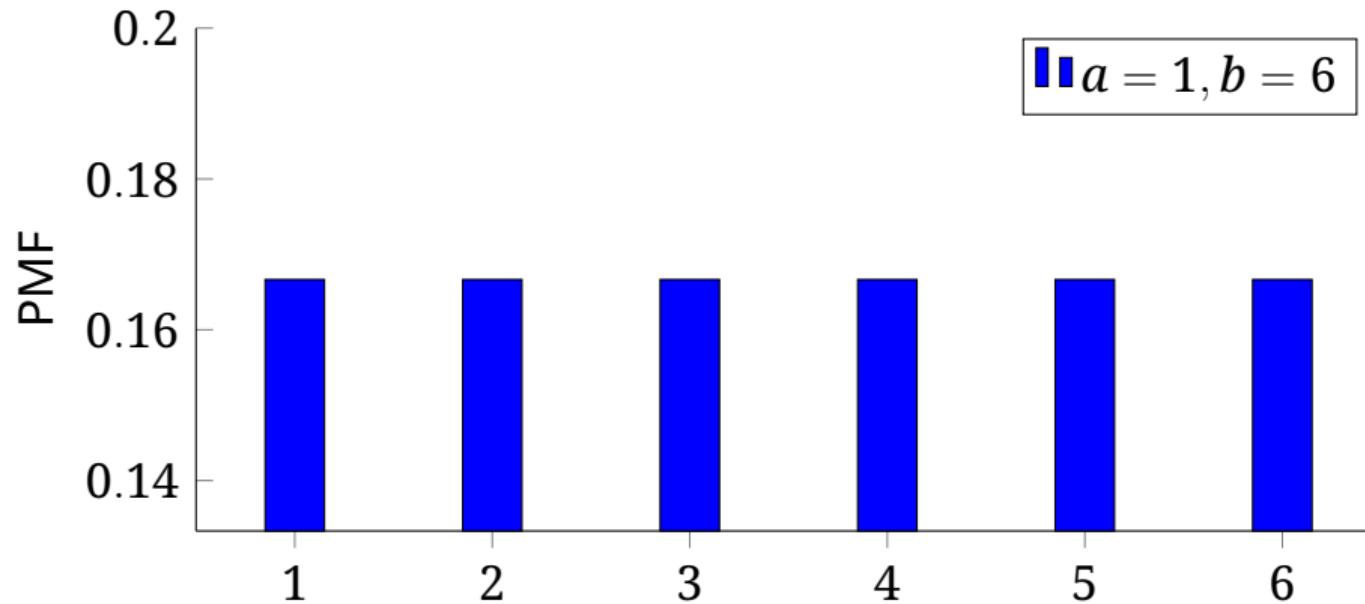
Example: dice.

# Discrete Uniform

$$\text{Uniform}(a, b) = f(x, a, b)$$

$$= \frac{1}{b - a + 1} \text{ for } a \leq x \leq b \text{ and } x \in \{a, a + 1, \dots, b - 1, b\}$$

# Discrete Uniform



# Bernoulli

Bernoulli distribution describes a binary event of the success of an experiment. We represent 0 as failure and 1 as success, hence the result of a Bernoulli distribution is a binary variable  $Y \in \{0, 1\}$ .

Bernoulli distribution is often used to model binary discrete results where there is only two possible results.

Bernoulli distribution has only a single parameter and its notation is  $\text{Bernoulli}(p)$ :

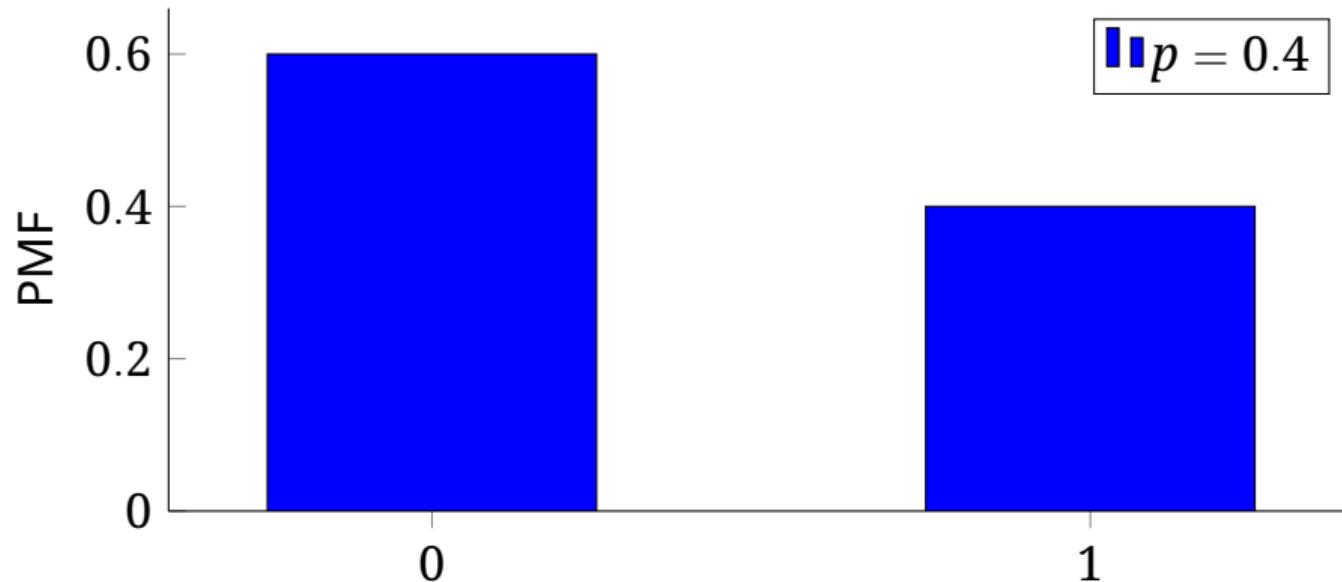
- $p$  – probability of success

Example: If the patient survived or died or if the client purchased or not.

# Bernoulli

$$\text{Bernoulli}(p) = f(x, p) = p^x(1 - p)^{1-x} \text{ for } x \in \{0, 1\}$$

# Bernoulli



# Binomial

The binomial distribution describes an event in which the number of successes in a sequence  $n$  independent experiments, each one making a yes-no question with probability of success  $p$ . Notice that Bernoulli distribution is a special case of the binomial distribution where  $n = 1$ .

The binomial distribution has two parameters and its notation is  $\text{Binomial}(n, p)$  :

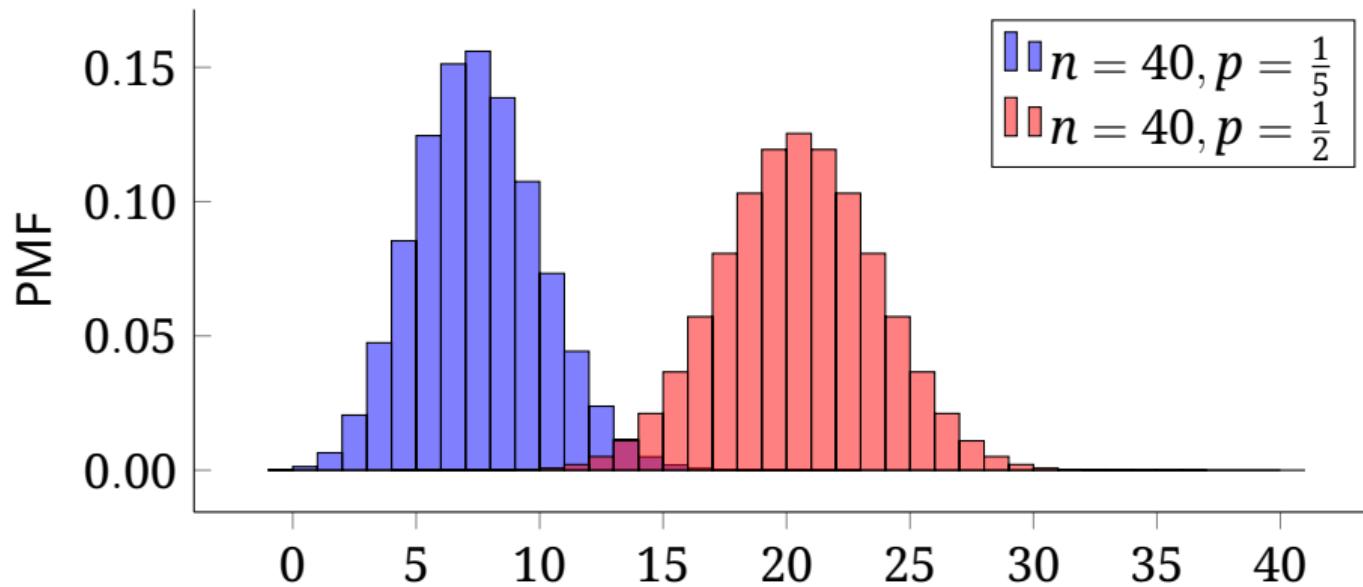
- $n$  – number of experiments
- $p$  – probability of success

Example: number of heads in five coin throws.

# Binomial

$$\text{Binomial}(n, p) = f(x, n, p) = \binom{n}{x} p^x (1 - p)^{n-x} \text{ for } x \in \{0, 1, \dots, n\}$$

# Binomial



# Poisson

Poisson distribution describes the probability of a certain number of events occurring in a fixed time interval if these events occur with a constant mean rate which is known and independent since the time of last occurrence. Poisson distribution can also be used for number of events in other type of intervals, such as distance, area or volume.

Poisson distribution has one parameter and its notation is  $\text{Poisson}(\lambda)$ :

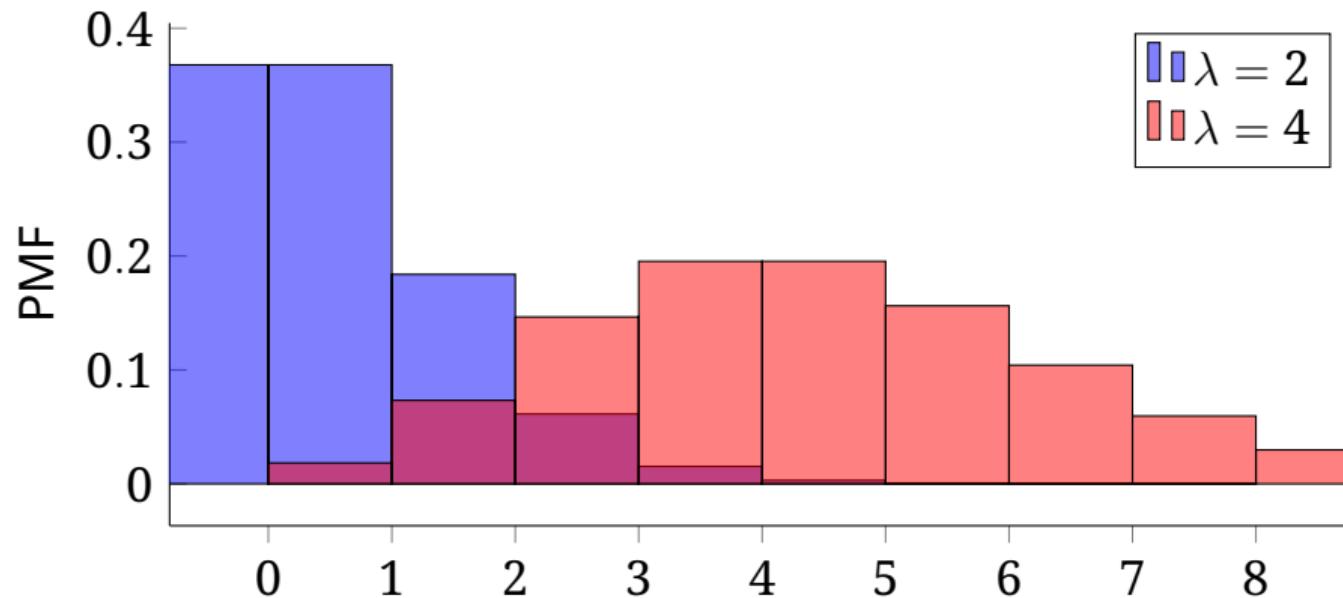
- $\lambda$  – rate

Example: number of e-mails that you receive daily or the number of the potholes you'll find in your commute.

# Poisson

$$\text{Poisson}(\lambda) = f(x, \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \text{ for } \lambda > 0$$

# Poisson



# Negative Binomial

The negative binomial distribution describes the distribution of the number of failures before the  $r^{th}$  success in a sequence of independent Bernoulli (yes/no) trials with the probability of success is  $p$ .

The negative binomial has two parameters  $r$  and  $p$  and its notation is Negative-Binomial( $r, p$ ):

Example: annual occurrence of tropical cyclones.

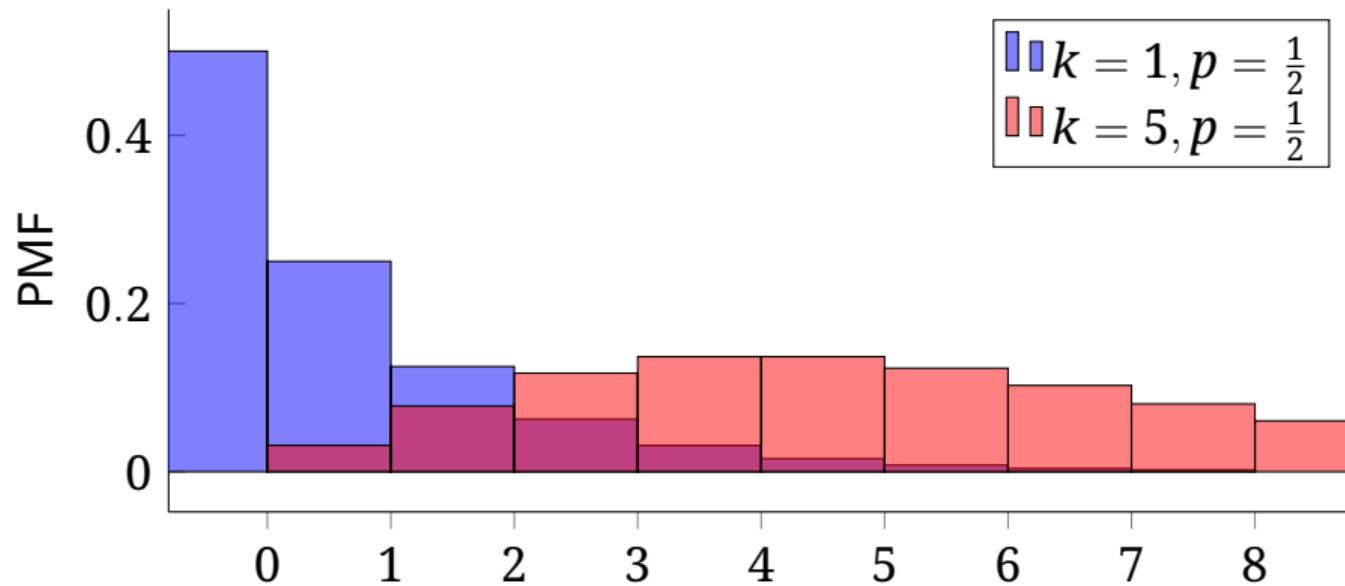
Any phenomena that can be modeled as a Poisson distribution can be modeled also as negative binomial distribution (Gelman et al., 2013b; Gelman, Hill, & Vehtari, 2020).

# Negative Binomial

$$\text{Negative Binomial}(k, p) = f(x, r, p) = \binom{x + r - 1}{r - 1} p^r (1 - p)^x$$

for  $x \in \{0, 1, \dots, n\}$

# Negative Binomial



# Continuous Distributions

## Definition (Continuous Distributions)

*Continuous probability distributions are distributions which the results are values in a continuous real number line:  $(-\infty, +\infty) \in \mathbb{R}$ . In continuous probability distributions we call the probability of a distribution taking values as "density". Since we are referring to real numbers we cannot obtain the probability of a random variable  $X$  taking exactly the value  $x$ . This will always be 0, since we cannot specify the exact value of  $x$ .  $x$  lies in the real number line, hence, we need to specify the probability of  $X$  taking values in an interval  $[a, b]$ . The probability density function (PDF) is defined as:*

$$\text{PDF}(x) = P(a \leq X \leq b) = \int_a^b f(x)dx$$

# Continuous Uniform

The continuous uniform distribution is a symmetric probability distribution in which an infinite number of value intervals are equally likely of being observable. Each one of the infinite  $n$  intervals have probability  $\frac{1}{n}$ .

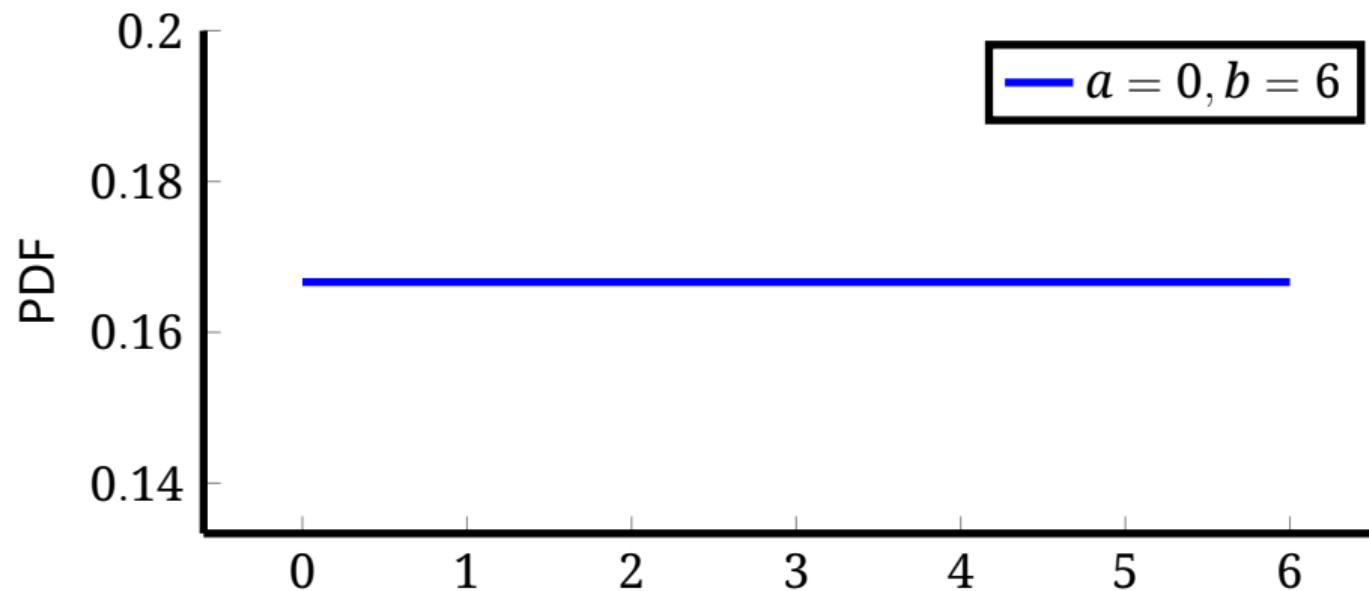
The continuous uniform distribution has two parameters and its notation is  $\text{Uniform}(a, b)$ :

- $a$  – lower bound
- $b$  – upper bound

# Continuous Uniform

$$\text{Uniform}(a, b) = f(x, a, b) = \frac{1}{b - a} \text{ for } a \leq x \leq b \text{ e } x \in [a, b]$$

# Continuous Uniform



# Normal

This distribution is generally used in social and natural sciences to represent continuous variables in which its underlying distribution are unknown. This assumption is due to the central limit theorem (CLT) that, under precise conditions, the mean of many samples (observations) of a random variable with finite mean and variance is itself a random variable which the underlying distribution converges to a normal distribution as the number of samples increases (as  $n \rightarrow \infty$ ).

Hence, physical quantities that we assume that are the sum of many independent processes (with measurement error) often have underlying distributions that are similar to normal distributions.

# Normal

The normal distribution has two parameters and its notation is  $\text{Normal}(\mu, \sigma^2)$  or  $N(\mu, \sigma^2)$ :

- $\mu$  – mean of the distribution, and also median and mode
- $\sigma$  – standard deviation<sup>xxxiv</sup>, a dispersion measure of how observations occur in relation from the mean

Example: height, weight etc.

---

<sup>xxxiv</sup>sometimes is also parameterized as variance  $\sigma^2$ .

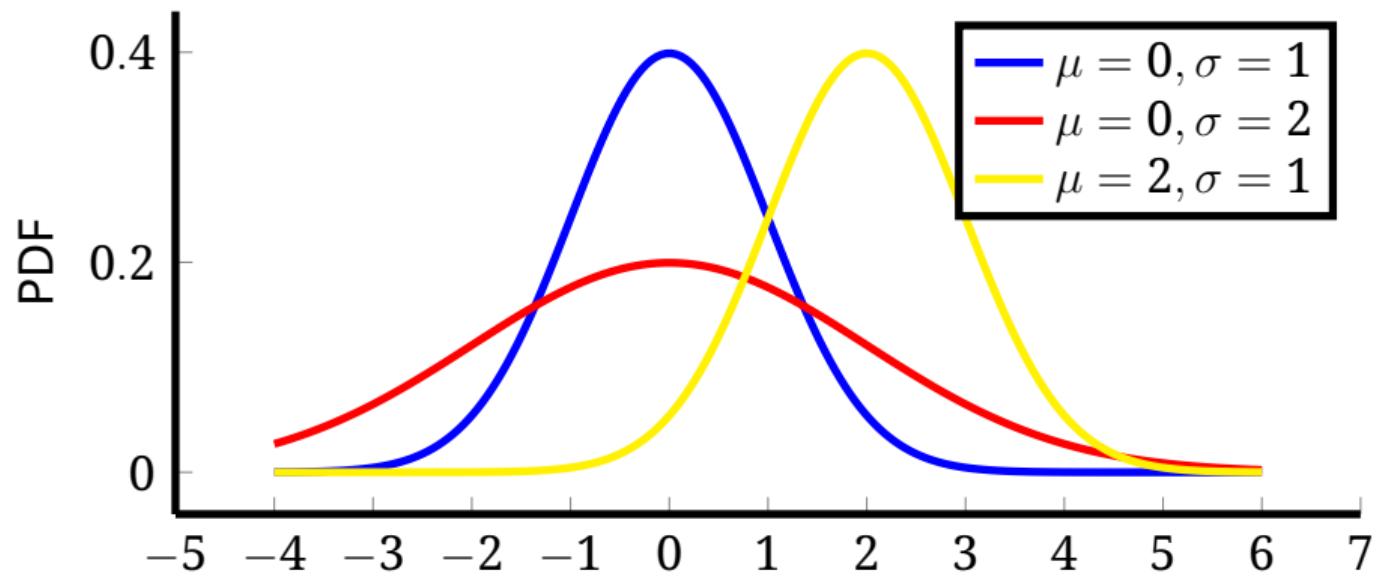
# Normal<sup>xxxv</sup>

$$\text{Normal}(\mu, \sigma) = f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \text{ for } \sigma > 0$$

---

<sup>xxxv</sup>see how the normal distribution was derived from the binomial distribution in the [backup slides](#).

# Normal



# Log-Normal

The log-normal distribution is a continuous probability distribution of a random variable which its natural logarithm is distributed as a normal distribution. Thus, if the natural logarithm of a random variable  $X$ ,  $\ln(X)$ , is distributed as a normal distribution, then  $Y = \ln(X)$  is normally distributed and  $X$  is log-normally distributed.

A log-normal random variable only takes positive real values. It is a convenient and useful model for measurements in exact and engineering sciences, as well as in biomedical, economical and other sciences. For example, energy, concentrations, length, financial returns and other measurements.

A log-normal process is the statistical realization of a multiplicative product of many independent positive random variables.

# Log-Normal

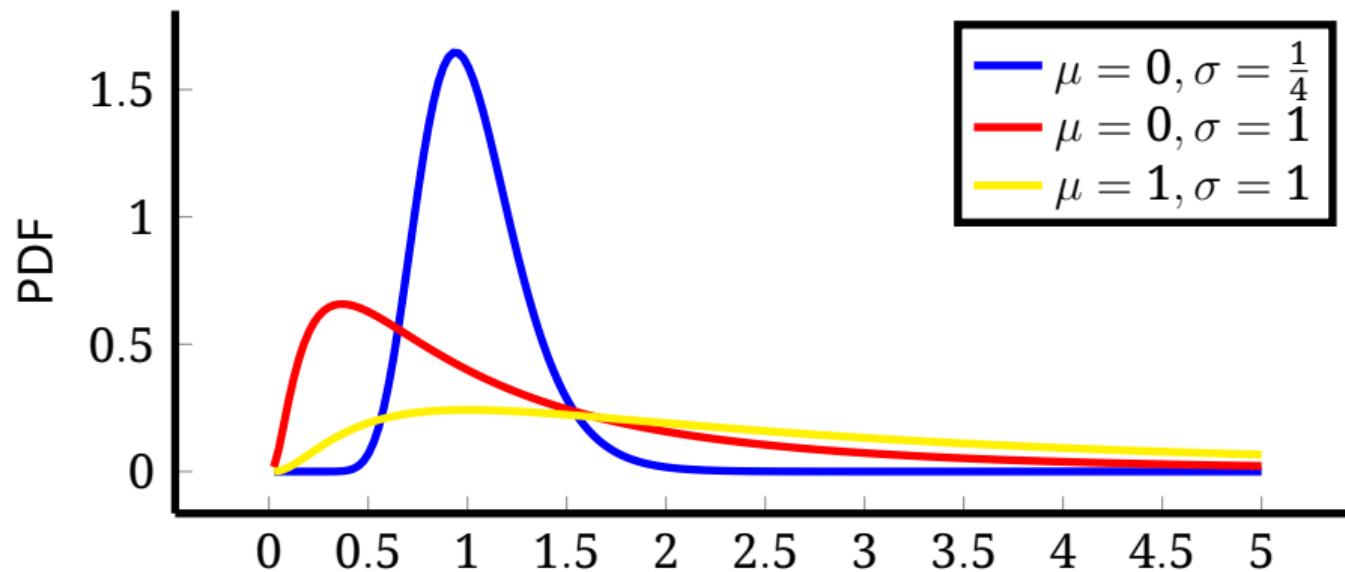
The log-normal distribution has two parameters and its notation is  $\text{Log-Normal}(\mu, \sigma^2)$ :

- $\mu$  – mean of the distribution's natural logarithm
- $\sigma$  – square root of the variance of the distribution's natural logarithm

# Log-Normal

$$\text{Log-Normal}(\mu, \sigma) = f(x, \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\left(\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right)} \text{ for } \sigma > 0$$

# Log-Normal



# Exponential

The exponential distribution is the probability distribution of the time between events that occurs in a continuous manner, are independent, and have constant mean rate of occurrence.

The exponential distribution has one parameter and its notation is  $\text{Exponential}(\lambda)$ :

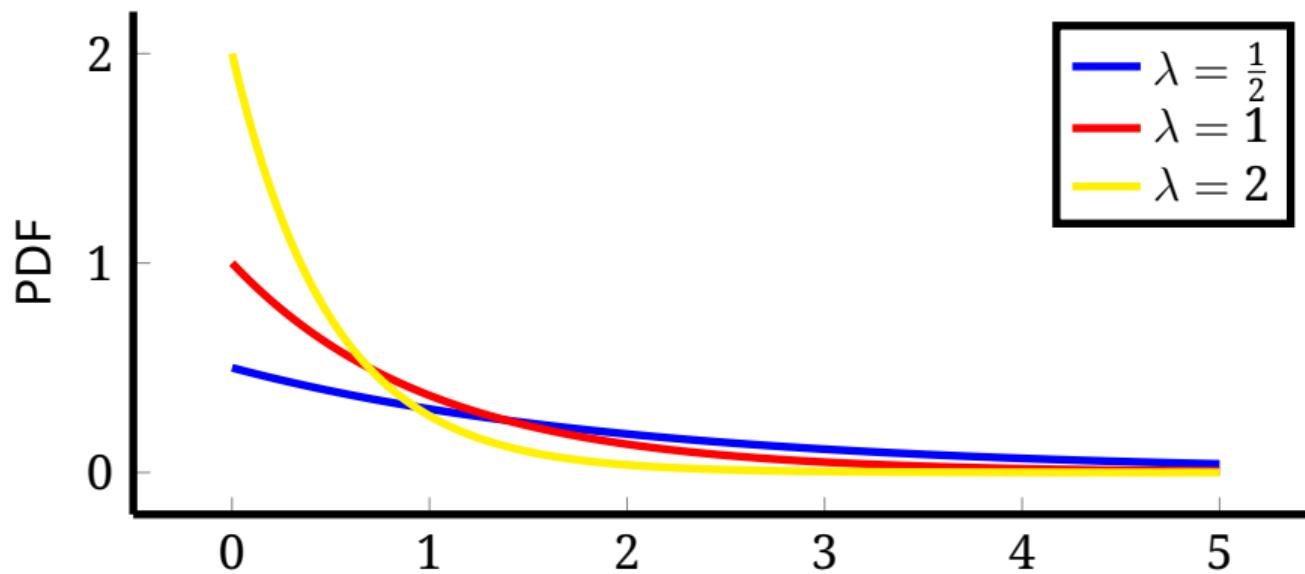
- $\lambda$  – rate

Example: How long until the next earthquake or how long until the next bus arrives.

# Exponential

$$\text{Exp}(\lambda) = f(x, \lambda) = \lambda e^{-\lambda x} \text{ for } \lambda > 0$$

# Exponential



# Gamma

The gamma distribution is a long-tailed distribution with support only for positive real numbers.

The gamma distribution has two parameters and its notation is  $\text{Gamma}(\alpha, \theta)$ :

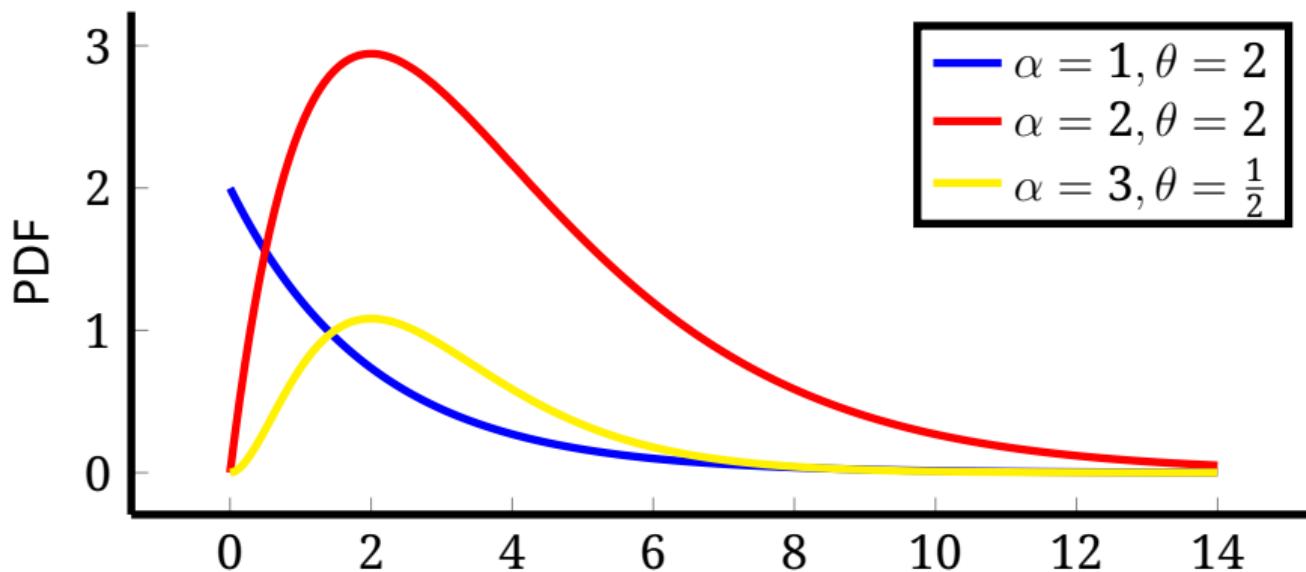
- $\alpha$  – shape parameter
- $\theta$  – rate parameter

Example: Any waiting time can be modelled with a gamma distribution.

# Gamma

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

# Gamma



## Student's $t$

Student's  $t$  distribution arises by estimating the mean of a normally-distributed population in situations where the sample size is small and the standard deviation is known<sup>xxxvi</sup>.

If we take a sample of  $n$  observations from a normal distribution, then Student's  $t$  distribution with  $\nu = n - 1$  degrees of freedom can be defined as the distribution of the location of the sample mean in relation to the true mean, divided by the sample's standard deviation, after multiplying by the scaling term  $\sqrt{n}$ .

Student's  $t$  distribution is symmetric and in a bell-shape, like the normal distribution, but with long tails, which means that has more chance to produce values far away from its mean.

---

<sup>xxxvi</sup>this is where the ubiquitous Student's  $t$  test.

## Student's $t$

Student's  $t$  distribution has one parameter and its notation is  $\text{Student}(\nu)$ :

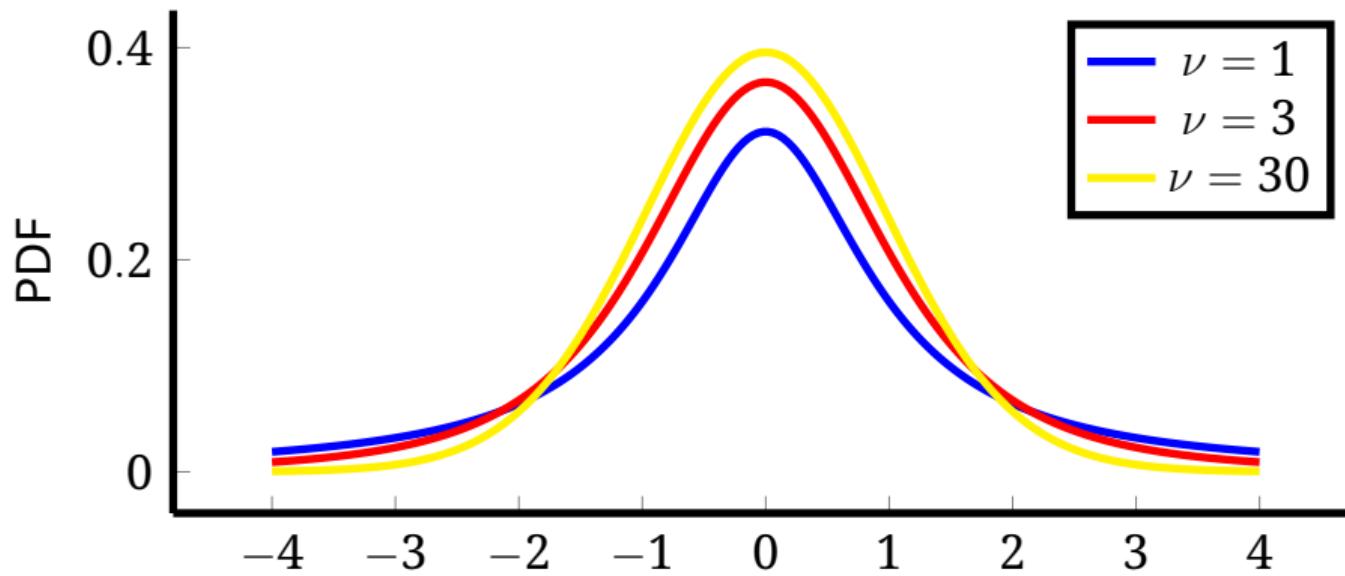
- $\nu$  – degrees of freedom, controls how much it resembles a normal distribution

Example: a dataset full of outliers.

# Student's $t$

$$\text{Student}(\nu) = f(x, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi} \Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \text{ for } \nu \geq 1$$

# Student's $t$



# Cauchy

The Cauchy distribution is bell-shaped distribution and a special case for Student's  $t$  with  $\nu = 1$ .

But, differently than Student's  $t$ , the Cauchy distribution has two parameters and its notation is  $\text{Cauchy}(\mu, \sigma)$ :

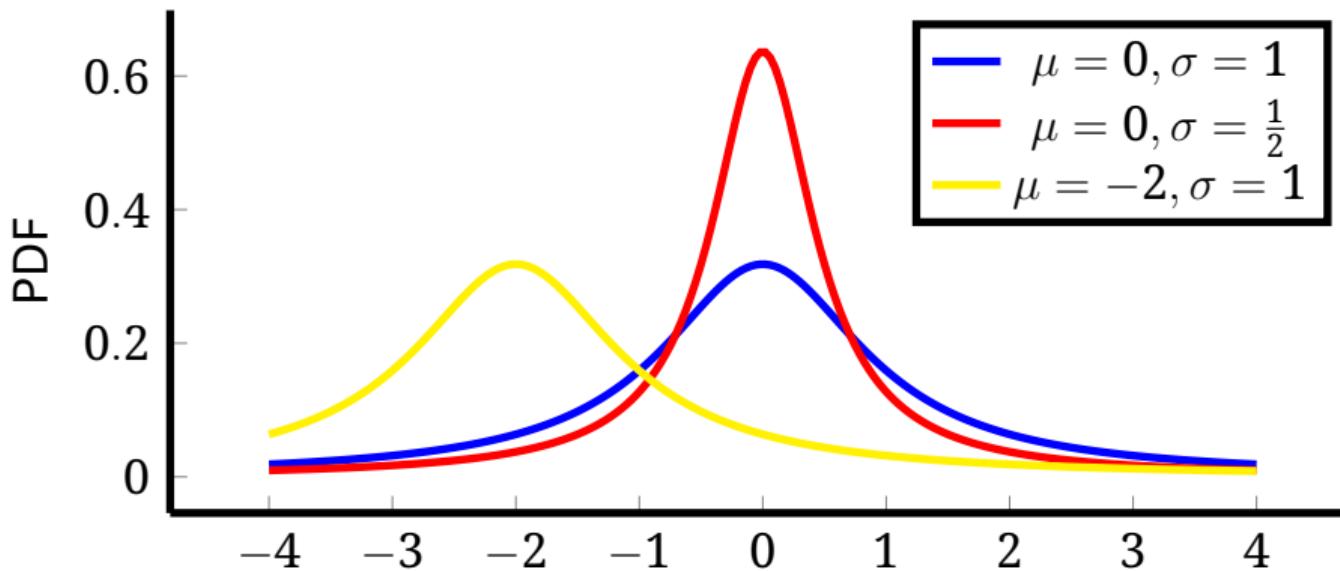
- $\mu$  – location parameter
- $\sigma$  – scale parameter

Example: a dataset full of outliers.

# Cauchy

$$\text{Cauchy}(\mu, \sigma) = \frac{1}{\pi\sigma \left(1 + \left(\frac{x-\mu}{\sigma}\right)^2\right)} \text{ for } \sigma \geq 0$$

# Cauchy



# Beta

The beta distribution is a natural choice to model anything that is restricted to values between 0 and 1. Hence, it is a good candidate to model probabilities and proportions.

The beta distribution has two parameters and its notations is  $\text{Beta}(\alpha, \beta)$ :

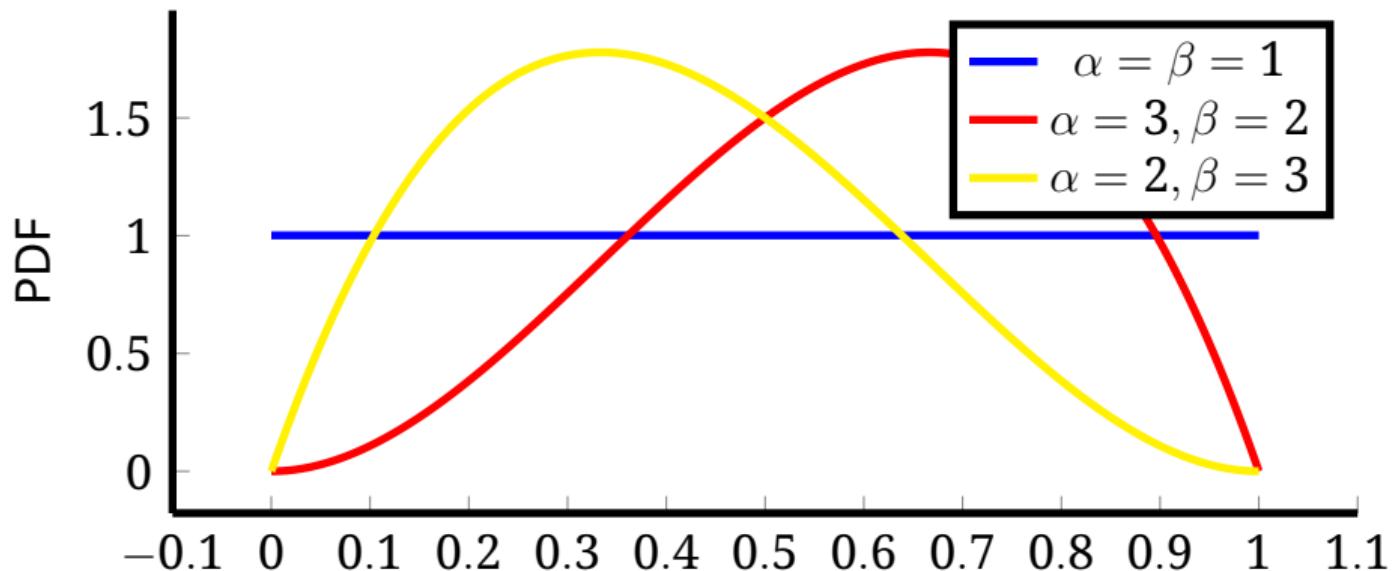
- $\alpha$  or sometimes  $a$  – shape parameter, controls how much the shape is shifted towards 1
- $\beta$  or sometimes  $b$  – shape parameter, controls how much the shape is shifted towards 0

Example: A basketball player that has already scored 5 free throws and missed 3 in a total of 8 attempts –  $\text{Beta}(3, 5)$

# Beta

$$\text{Beta}(\alpha, \beta) = f(x, \alpha, \beta) \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}} \text{ for } \alpha, \beta > 0 \text{ and } x \in [0, 1]$$

# Beta



# Matrix-variate Distributions

So far we've seen random distributions that return a **scalar value**. That means a single number.

But we can also return a **matrix** instead using a **Matrix-variate distribution**.

There are several ways to generate random matrices. But we'll focus on a special case of matrix: **positive (semi-)definite matrices**.

# Positive (Semi-)Definite Matrix

## Definition (Positive (Semi-)Definite Matrix)

*A matrix  $\mathbf{M}$  is positive (semi-)definite if it is symmetric or Hermitian, and all its eigenvalues are real and positive (non-negative).*

# Covariance Matrices

## Example (Positive (Semi-)Definite Matrix)

The **covariance matrix** of a multivariate probability distribution is always **positive definite**.

Conversely, **every positive definite matrix is the covariance matrix** of some multivariate distribution.

## Inverse Wishart

The inverse Wishart was one of the first computationally-efficient distributions to model covariance matrices (Gelman et al., 2013b). It is being supplanted by more modern implementations.

It is a generalization of the inverse gamma distribution to  $p \times p$  positive definite matrices.

# Inverse Wishart

$$\text{Inverse Wishart}(\nu, \Psi) = \frac{|\Psi|^{\nu/2}}{2^{\nu p/2} \Gamma_p(\frac{\nu}{2})} |\Sigma|^{-(\nu+p+1)/2} e^{-\frac{1}{2} \text{tr}(\Psi \Sigma^{-1})}$$

where:

- $p$  dimensionality of the matrix-variate distribution
- $\Sigma$  is a  $p \times p$  positive definite matrix
- $|\cdot|$  is the determinant
- $\text{tr}$  is the trace
- $\Gamma_P(\cdot)$  is the multivariate gamma function

Parameters:

- $\nu > p - 1$  is the degrees of freedom
- $\Psi$  scale matrix  $p \times p$

## LKJ<sup>xxxvii</sup>

LKJ is the go-to distribution for covariance matrices in a Bayesian framework.

$$\text{LKJ}(\eta) = \left[ \prod_{k=1}^{p-1} \pi^{\frac{k}{2}} \frac{\Gamma\left(\eta + \frac{p-1-k}{2}\right)}{\Gamma\left(\eta + \frac{p-1}{2}\right)} \right]^{-1} |\Sigma|^{\eta-1}$$

where:

- $p$  dimensionality of the matrix-variate distribution
- $\Sigma$  is a  $p \times p$  positive definite matrix
- $|\cdot|$  is the determinant

LKJ has a single parameter  $\eta > 0$  which acts as a shape parameter.

---

<sup>xxxvii</sup>Lewandowski et al. (2009) – LKJ are the authors' last name initials – Lewandowski, Kurowicka and Joe.

## LKJ

One interesting property of the LKJ distribution is that if we disregard the product over the beta functions<sup>xxxviii</sup>, then the PDF is proportional to the determinant exponentiated by  $\eta - 1$ :

$$\text{LKJ} \propto |\Sigma|^{\eta-1}$$

where:

- $\Sigma$  is a  $p \times p$  positive definite matrix
- $|\cdot|$  is the determinant

---

<sup>xxxviii</sup>generally the beta function can be expressed as a fraction of two gamma functions.

# How to Interpret Logistic Regression Coefficients

# How to Interpret Logistic Regression Coefficients

If we revisit logistic transformation mathematical expression, we see that, in order to interpret coefficients  $\beta$ , we need to perform a transformation.

Specifically, we need to undo the logistic transformation. We are looking for its inverse function.

# Probability versus Odds

But before that, we need to discern between **probability and odds**<sup>xxxix</sup>.

- **Probability:** a real number between 0 and 1 that represents the certainty that an event will occur, either by long-term frequencies (frequentist approach) or degrees of belief (Bayesian approach).
- **Odds:** a positive real number ( $\mathbb{R}^+$ ) that also measures the certainty of an event happening. However this measure is not expressed as a probability (between 0 and 1), but as the **ratio between the number of results that generate our desired event and the number of results that do not generate our desired event**:

$$\text{odds} = \frac{p}{1 - p}$$

where  $p$  is the probability.

---

<sup>xxxix</sup>mathematically speaking.

# Probability versus Odds

$$\text{odds} = \frac{p}{1 - p}$$

where  $p$  is the probability.

- Odds with a value of 1 is a neutral odds, similar to a fair coin:  $p = \frac{1}{2}$
- Odds below 1 decrease the probability of seeing a certain event.
- Odds over 1 increase the probability of seeing a certain event.

# Logodds

If you revisit the logistic function, you'll see that the intercept  $\alpha$  and coefficients  $\beta$  are literally the **log of the odds** (logodds):

$$p = \text{logistic}(\alpha + \mathbf{X}\beta)$$

$$p = \text{logistic}(\alpha) + \text{logistic}(\mathbf{X}\beta)$$

$$p = \frac{1}{1 + e^{(-\beta)}}$$

$$\beta = \log(\text{odds})$$

# Logodds

Hence, the coefficients of a logistic regression are expressed in logodds, in which 0 is the neutral element, and any number above or below it increases or decreases, respectively, the changes of obtaining a “success” in  $y$ . To have a more intuitive interpretation (similar to the betting houses), we need to **convert the logodds into chances** by undoing the log function. We need to perform an **exponentiation** of  $\alpha$  and  $\beta$  values:

$$\text{odds}(\alpha) = e^\alpha$$

$$\text{odds}(\beta) = e^\beta$$

## How the Normal distribution arose<sup>xl</sup>

$$\text{Binomial}(n, k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \quad (\text{Stirling})$$

$$\lim_{n \rightarrow \infty} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}$$

We know that in the binomial:  $E = np$  and  $\text{Var} = npq$ ; hence replacing  $E$  by  $\mu$  and  $\text{Var}$  by  $\sigma^2$ :

$$\lim_{n \rightarrow \infty} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(k-\mu)^2}{\sigma^2}}$$

---

<sup>xl</sup>Origins can be traced back to Abraham de Moivre in 1738. A better explanation can be found by [clicking here](#).

## **Temporary page!**

$\text{\LaTeX}$  was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will disappear, because  $\text{\LaTeX}$  now knows how many pages to expect for this document.