
# 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. Hierarchical Models
6. Markov Chain Monte Carlo (MCMC) and Model Metrics
7. Model Comparison

# What is Pumas?

Pumas (**P**harmace**U**tical **M**odeling **A**nd **S**imulation) (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 - 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 $y$:
  - $P(\theta \mid y)$
  - $P(\tilde{y} \mid y)$
- We also, implicitly, conditioned on the observed data from any covariate $x$
- Generally, we are interested in:
  - expected response of a new subject to a drug, e.g. $\mathrm{E}[\hat{y} \mid y]$
  - the probability of drug effect is higher than zero, e.g. $P(\theta > 0 \mid y) \geq 0.95$

# 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:
  - $P(\text{rain}) =$ 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[i]

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

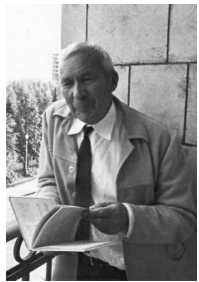
$$P(A) \geq 0$$

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

$$P(A) = 1 - P(B) \text{ and } P(B) = 1 - P(A)$$

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

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



[i] Kolmogorov (1933)

# Sample Space[ii]

- Discrete

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

- Continuous

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

[ii] $\theta$ 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[iii]

The planet has a magnetic field

$\theta \in E_1$

The planet has moon(s)

$\theta \in E_2$

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

$\theta \in E_1 \cap E_2$

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

$\theta \in E_1 \cup E_2$

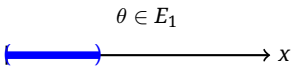
The planet does *not* have a magnetic field

$\theta \in \neg E_1$

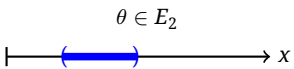
[iii] figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# Continuous Sample Space[iv]

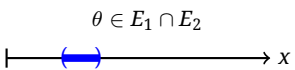
The distance is less than five centimeters



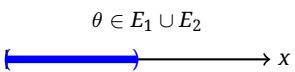

The distance is between three and seven centimeters



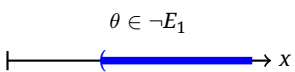

The distance is less than five centimeters
*and* e between three and seven centimeters




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




The distance is *not* less than five centimeters




[iv] 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 $\int$.

# Conditional Probability

## Definition (Conditional Probability)

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

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

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

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

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

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

In the previous example we have the symmetry $P(A \mid K) = P(K \mid A)$, **but not always this is true**[v]

## Example (The Pope is catholic)

- $P(\text{pope})$: Probability of some random person being the Pope, something really small, 1 in 8 billion $\left(\frac{1}{8 \cdot 10^9}\right)$
- $P(\text{catholic})$: Probability of some random person being catholic, 1.34 billion in 8 billion $\left(\frac{1.34}{8} \approx 0.17\right)$
- $P(\text{catholic} \mid \text{pope})$: Probability of the Pope being catholic $\left(\frac{999}{1000} = 0.999\right)$
- $P(\text{pope} \mid \text{catholic})$: Probability of a catholic person being the Pope $\left(\frac{1}{1.34 \cdot 10^9} \cdot 0.999 \approx 7.46 \cdot 10^{-10}\right)$
- **Hence**: $P(\text{catholic} \mid \text{pope}) \neq P(\text{pope} \mid \text{catholic})$

[v]More specific, if the basal rates $P(A)$ and $P(B)$ aren't equal, the symmetry is broken $P(A \mid B) \neq P(B \mid A)$

# 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 $B$".*

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

# Product Rule of Probability[vi]

## Definition (Product Rule)

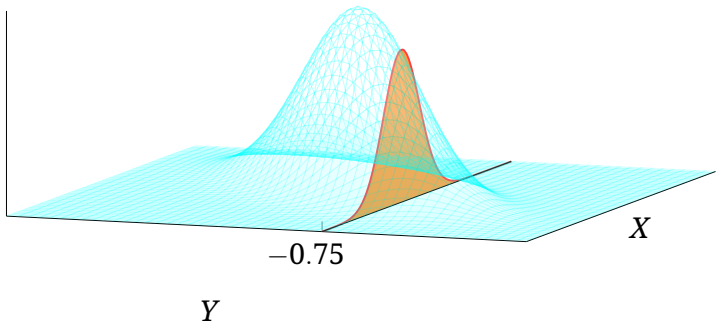
*We can decompose a joint probability $P(A, B)$ into the product of two probabilities:*

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

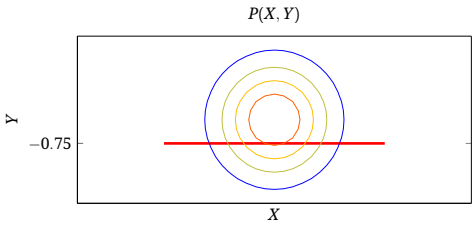
[vi] also called the Product Rule of Probability.

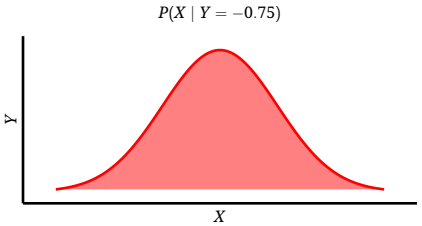
# Visualization of Joint Probability versus Conditional Probability

$P(X, Y)$ 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:*

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

# Bayes' Theorem Proof

Remember the following probability identity:

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

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

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

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

# Another Probability Textbook Classic[vii]

## 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)

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

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

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

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

[vii] Adapted from: Yudkowski - *An Intuitive Explanation of Bayes' Theorem*.

# Why Bayes' Theorem is Important?

## Idea (We can Invert the Conditional Probability)

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

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

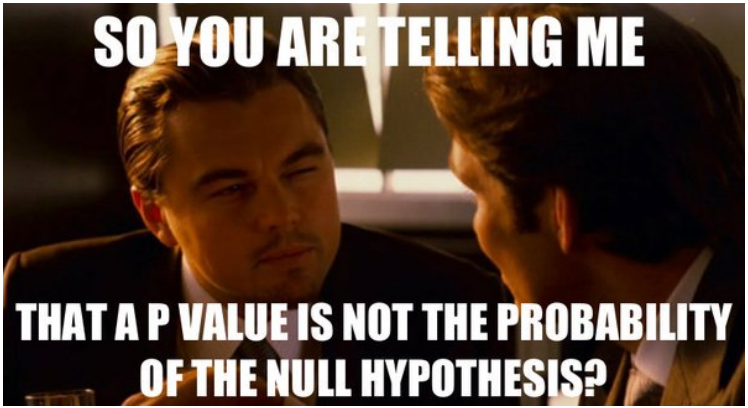
# What are $p$-values?

## 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 $P(D \mid H_0)$ and $P(H_0 \mid D)$. To get $P(H_0 \mid D)$ you need Bayesian statistics.
- **$p$-value is not the probability of data being generated at random** - No! We haven't stated nothing 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 of 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 $H_0$

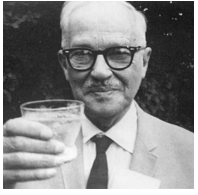
To find out about any $p$-value, **find out what $H_0$ is behind it**. It's definition will never change, since it is always $P(D \mid H_0)$:

- **$t$-test**: $P(D \mid$ the difference between the groups is zero$)$
- **ANOVA**: $P(D \mid$ there is no difference between groups$)$
- **Regression**: $P(D \mid$ coefficient has a null value$)$
- **Shapiro-Wilk**:
  $P(D \mid$ 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)*

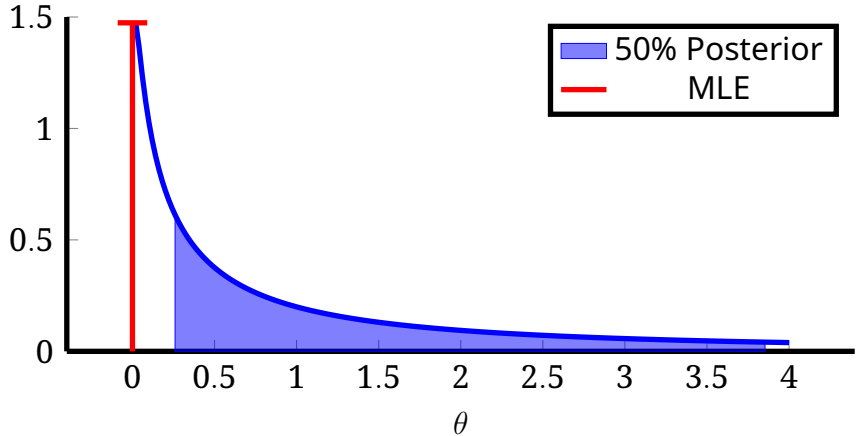
# What are Confidence Intervals?

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

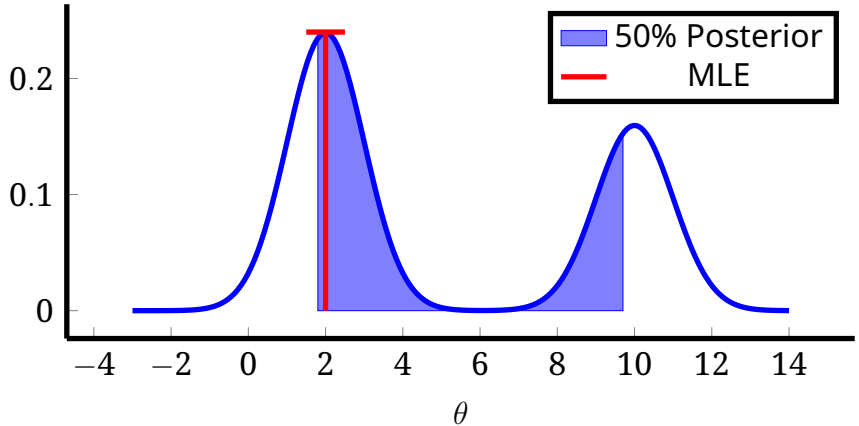
Say you performed a statistical analysis to compare the efficacy of a public policy between two groups and you obtain a difference between the mean of these groups. You can express this difference as a confidence interval. Often we choose 95% confidence. This means that **95 studies out of 100**, that uses the **same sample size and target population**, performing the **same statistical test**, will expect to find a result of the mean difference between groups inside the confidence interval.

Doesn't say anything about you **target population**, but about you **sample** in an insane process of **infinite sampling** ...

# 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 no 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?[viii]

> *… 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)

[viii] *inverse probability* was how Bayes' theorem was called in the beginning of the 20th century

# Inside every nonBayesian, there is a Bayesian struggling to get out[ix]

- 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.

[ix] 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:

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

- $\theta$ – parameter(s) of interest
- $y$ – observed data
- **Priori**: 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**[x]: $P(y)$ does not make any intuitive sense. This probability is transformed and can be interpreted as something that only exists so that the result $P(y \mid \theta)P(\theta)$ be constrained between 0 e 1 – a valid probability.

[x] 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 $P(\theta \mid y)$ 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.

# Bayesian vs Frequentist Stats

| | Bayesian Statistics | Frequentist Statistics |
|---|---|---|
| **Data** | Fixed — Non-random | Uncertain — Random |
| **Parameters** | Uncertain — Random | Fixed — Non-random |
| **Inference** | Uncertainty regarding the parameter value | Uncertainty regarding the sampling process from an infinite population |
| **Probability** | Subjective[xi] | Objective (but with several model assumptions) |
| **Uncertainty** | Posterior Interval — $P(\theta \mid y)$ | Confidence Interval — $P(y \mid \theta)$ |

[xi] with highly informative 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:

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

# The subjectivity of the Prior

- Many critics 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.
- Anything that involves human action will never be free from subjectivity. We have subjectivity in everything and science is no exception.
- The creative and deductive process of theory and hypotheses formulations is **not** objective.
- Frequentist statistics, which bans the use of prior probabilities is also subjective, since there is **A LOT** of subjectivity in choosing which model and likelihood function (Jaynes, 2003; van de Schoot et al., 2021)

# How to Incorporate Subjectivity

- Bayesian statistics **embraces** subjectivity while frequentist statistics **bans** it.
- For Bayesian statistics, **subjectivity guides our inferences** and leads to more robust and reliable models that can assist in decision making.
- Whereas, for frequentist statistics, **subjectivity is a taboo** and all inferences should be objective, even if it resorts to **hiding and omitting model assumptions**.
- Bayesian statistics also has assumptions and subjectivity, but these are **declared and formalized**

# Types of Priors

In general, we can have 3 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**: small amounts of real-world information along with common sense and low specific domain knowledge added.
- **informative**: introduction of medium to high domain knowledge.

# Uniform Prior (Flat)

Starts from the premise that “everything is possible”. There is no limits in the degree of beliefs that the distribution of certain values must be or any sort of restrictions.

Flat and super-vague priors are not usually recommended and some thought should included to have at least weakly informative priors.

Formally, an uniform prior is an uniform distribution over all the possible support of the possible values:

- **model parameters**: $\{\theta \in \mathbb{R} : -\infty < \theta < \infty\}$
- **model error or residuals**: $\{\sigma \in \mathbb{R}^+ : 0 \leq \theta < \infty\}$

# Weekly Informative Prior

Here we start to have “educated” guess about our parameter values. Hence, we don’t start from the premise that “anything is possible”.

I recommend always to transform the priors of the problem at hand into something centered in 0 with standard deviation of 1[xii]:

- $\theta \sim \text{Normal}(0, 1)$ (Andrew Gelman’s preferred choice[xiii])
- $\theta \sim \text{Student}(\nu = 3, 0, 1)$ (Aki Vehtari’s preferred choice)

[xii] this is called standardization, transforming all variables into $\mu = 0$ and $\sigma = 1$.

[xiii] see more about prior choices in the Stan’s GitHub wiki.

# Setting-up Pumas

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

# 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[xiv]:

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

[xiv] for the whole full list check here.

# What are hierarchical models?

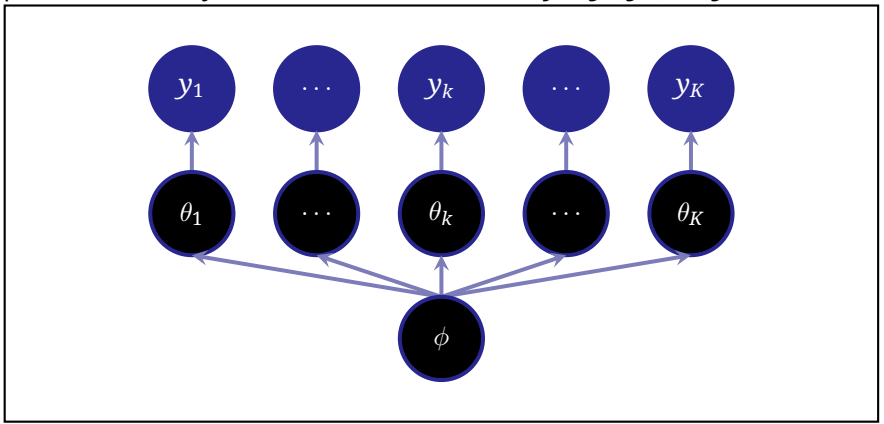
## Definition (Hierarchical Model)

*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?[xv]

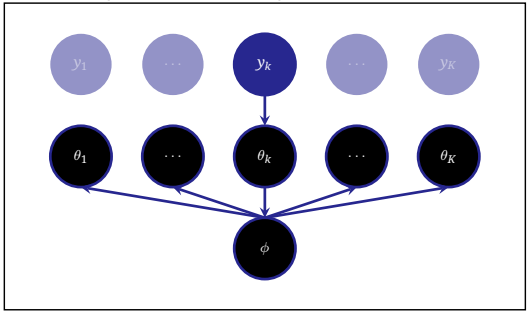
Hyperparameter $\phi$ that parameterizes $\theta_1, \theta_2, \ldots, \theta_K$, that are used to infer the posterior density of some random variable $\mathbf{y} = y_1, y_2, \ldots, y_K$

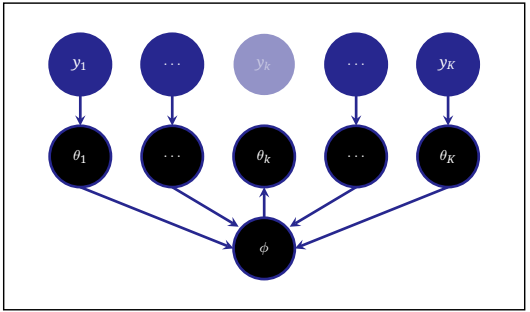



[xv] figure adapted from Michael Betancourt (CC-BY-SA-4.0)

# What are Hierarchical Models?[xvi]

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, $\theta_k$. These parameters, however, inform directly the population-level parameters, $\phi$, 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...

[xvi] 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[xvii]. 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

[xvii] 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?

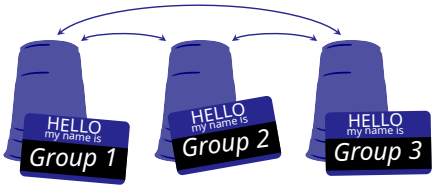
Most important is **not to violate** the **exchangeability assumption** (de Finetti, 1974).

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

# Exchangeability (de Finetti, 1974)[xviii]


[xviii] figures adapted from Michael Betancourt (CC-BY-SA-4.0).

# Exchangeability (de Finetti, 1974)[xix]





[xix] figures adapted from Michael Betancourt (CC-BY-SA-4.0).

# Hyperprior

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

$$\begin{aligned}\mathbf{y} &\sim \text{Normal}(10, \boldsymbol{\theta}) \\ \boldsymbol{\theta} &\sim \text{Normal}(0, \phi) \\ \phi &\sim \text{Exponential}(1)\end{aligned}$$

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

# Frequentist versus Bayesian Approaches

There are also hierarchical models in frequentist statistics. They are mainly available in the `lme4` package (Bates et al., 2015), and also in `MixedModels.jl` (Bates et al., 2022).

- **optimization of the likelihood function** versus **posterior approximation via MCMC**. Almost always lead to convergence failure for models that are not extremely simple.
- **frequentist hierarchical models do not compute $p$-values for the group-level effects**[xx]. This is due to the underlying assumptions of the approximations that frequentist statistics has to to do in order to calculate the group-level effects $p$-values. The main one being that the groups must be balanced. In other words, the groups must be homogeneous in size. Hence, any unbalance in group compositions results in pathological $p$-values that should not be trusted.

[xx] Doug Bates, Reto ... of the lme4 package ...

# Frequentist versus Bayesian Approaches

To sum up, **frequentist approach for hierarchical models is not robust** in both the **inference process** (**convergence flaws** during the maximum likelihood estimation), and also in the **results** from the inference process (do not provide $p$-values due to **strong assumptions that are almost always violated**).

# Approaches to Hierarchical Modeling

- **Varying-intercept** model: One group-level intercept besides the population-level intercept and coefficients.
- **Varying-slope** model: One or more group-level coefficient(s) besides the population-level intercept and coefficients.
- **Varying-intercept-slope** model: One group-level intercept and one or more group-level coefficient(s) besides the population-level intercept and coefficients.
- **Correlated varying-intercept-slope** model: One group-level intercept and one or more group-level coefficient(s) besides the population-level intercept and coefficients. Here the group-level intercept and coefficients priors are **sampled from the same multivariate distribution**.

# Mathematical Specification of Hierarchical Models

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

Here we insert a column filled with 1s in the data matrix $\mathbf{X}$.

Mathematically, this makes the column behave like an "identity" variable (because the number 1 in the multiplication operation $1 \cdot \beta$ is the identity element. It maps $x \to x$ keeping the value of $x$ intact) and, consequently, we can interpret the column's coefficient as the model's intercept.

# Mathematical Specification of Hierarchical Models

Hence, we have as a data matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1K} \\ 1 & x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{NK} \end{bmatrix}$$

# Mathematical Specification – Varying-Intercept Model

This example is for linear regression:

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}\left(\alpha + \alpha_j + \mathbf{X} \cdot \boldsymbol{\beta}, \sigma\right) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\alpha_j &\sim \text{Normal}(0, \tau) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\tau &\sim \text{Cauchy}^{+}(0, \psi_\alpha) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

# Mathematical Specification – Varying-Intercept Model

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

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}(\alpha + \alpha_{j1} + \alpha_{j2} + \mathbf{X}\boldsymbol{\beta}, \sigma) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\alpha_{j1} &\sim \text{Normal}(0, \tau_{\alpha j1}) \\
\alpha_{j2} &\sim \text{Normal}(0, \tau_{\alpha j2}) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\tau_{\alpha j1} &\sim \text{Cauchy}^{+}(0, \psi_{\alpha j1}) \\
\tau_{\alpha j2} &\sim \text{Cauchy}^{+}(0, \psi_{\alpha j2}) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

# Mathematical Specification – Varying-Slope Model

This example is for linear regression:

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}\left(\alpha + \mathbf{X} \cdot \boldsymbol{\beta}_j \cdot \boldsymbol{\tau}, \sigma\right) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta}_j &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\boldsymbol{\tau} &\sim \text{Cauchy}^{+}(\mathbf{0}, \psi_\beta) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

$\boldsymbol{\tau}$ is a vector of priors for the group-level coefficients' standard deviation.

# Mathematical Specification – Varying-Slope Model

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

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\beta}_{j1} \cdot \boldsymbol{\tau}_{j1} + \mathbf{X}\boldsymbol{\beta}_{j2} \cdot \boldsymbol{\tau}_{j2}, \sigma) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta}_{j1} &\sim \text{Normal}(\mu_{\boldsymbol{\beta j1}}, \sigma_{\boldsymbol{\beta j1}}) \\
\boldsymbol{\beta}_{j2} &\sim \text{Normal}(\mu_{\boldsymbol{\beta j2}}, \sigma_{\boldsymbol{\beta j2}}) \\
\boldsymbol{\tau}_{\beta j1} &\sim \text{Cauchy}^{+}(0, \psi_{\beta j1}) \\
\boldsymbol{\tau}_{\beta j2} &\sim \text{Cauchy}^{+}(0, \psi_{\beta j2}) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

# Mathematical Specification – Varying-Intercept-Slope Model

This example is for linear regression:

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}\left(\alpha + \alpha_j + \mathbf{X} \cdot \boldsymbol{\beta}_j \cdot \boldsymbol{\tau}_\beta, \sigma\right) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\alpha_j &\sim \text{Normal}(0, \tau_\alpha) \\
\boldsymbol{\beta}_j &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\tau_\alpha &\sim \text{Cauchy}^+(0, \psi_\alpha) \\
\boldsymbol{\tau}_\beta &\sim \text{Cauchy}^+(0, \psi_\beta) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

$\boldsymbol{\tau}_\beta$ is a vector of priors for the group-level coefficients' standard deviation.

# Mathematical Specification – Varying-Intercept-Slope Model

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

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}(\alpha + \alpha_j + \mathbf{X}\boldsymbol{\beta}_{j1} \cdot \boldsymbol{\tau}_{j1} + \mathbf{X}\boldsymbol{\beta}_{j2} \cdot \boldsymbol{\tau}_{j2}, \sigma) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\alpha_j &\sim \text{Normal}(0, \tau_\alpha) \\
\boldsymbol{\beta}_{j1} &\sim \text{Normal}(\mu_{\boldsymbol{\beta j1}}, \sigma_{\boldsymbol{\beta j1}}) \\
\boldsymbol{\beta}_{j2} &\sim \text{Normal}(\mu_{\boldsymbol{\beta j2}}, \sigma_{\boldsymbol{\beta j2}}) \\
\tau_\alpha &\sim \text{Cauchy}^+(0, \psi_\alpha) \\
\boldsymbol{\tau}_{\beta j1} &\sim \text{Cauchy}^+(0, \psi_{\beta j1}) \\
\boldsymbol{\tau}_{\beta j2} &\sim \text{Cauchy}^+(0, \psi_{\beta j2}) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

# Mathematical Specification – Correlated Varying-Slope Model

This example is for linear regression:

$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}(\mathbf{X}\boldsymbol{\beta}_j, \sigma) \\
\boldsymbol{\beta}_j &\sim \text{Multivariate Normal}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}) \quad \text{for} \quad j \in \{1, \ldots, J\} \\
\boldsymbol{\Sigma} &\sim \text{LKJ}(\eta) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

Each coefficient vector $\boldsymbol{\beta}_j$ represents the model columns $\mathbf{X}$ coefficients for every group $j \in J$. Also the first column of $\mathbf{X}$ is a column filled with 1s (intercept).

# Mathematical Specification – Correlated Varying-Slope Model

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

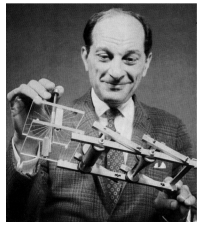
$$
\begin{aligned}
\mathbf{y} &\sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\beta}_{j1} + \mathbf{X}\boldsymbol{\beta}_{j2}, \sigma) \\
\boldsymbol{\beta}_{j1} &\sim \text{Multivariate Normal}(\boldsymbol{\mu}_{j1}, \boldsymbol{\Sigma}_1) \quad \text{for} \quad j_1 \in \{1, \ldots, J_1\} \\
\boldsymbol{\beta}_{j2} &\sim \text{Multivariate Normal}(\boldsymbol{\mu}_{j2}, \boldsymbol{\Sigma}_2) \quad \text{for} \quad j_2 \in \{1, \ldots, J_2\} \\
\boldsymbol{\Sigma}_1 &\sim \text{LKJ}(\eta_1) \\
\boldsymbol{\Sigma}_2 &\sim \text{LKJ}(\eta_2) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

# 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)

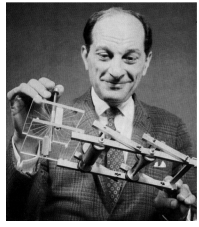
# Monte Carlo Methods

- `Stan` is named after the mathematician Stanislaw Ulam, who was involved in the Manhattan project, and while trying to calculate the neutron diffusion process for the hydrogen bomb ended up creating a whole class of methods called **Monte Carlo** (Eckhardt, 1987).
- Monte Carlo methods employ randomness to solve problems in principle are deterministic in nature. They are frequently used in physics and mathematical problems, and very useful when it is difficult or impossible to use other approaches.



# History Behind the Monte Carlo Methods[xxi]

- 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 where 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.
- 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.



# 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 discrete cases, we can turn the denominator into a sum over all parameters using the **chain rule** of probability:

$$P(A, B \mid C) = P(A \mid B, C) \times P(B \mid C)$$

This is also known as **marginalization**:

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

# Why Do We Need MCMC?

However, in the case of continuous values, the denominator $P(\text{data})$ turns into a very big and nasty integral:

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

In many cases the integral is intractable (not possible of being deterministic evaluated) and, thus, we must find other ways to compute the posterior $P(\theta \mid \text{data})$ without using the denominator $P(\text{data})$.

**This is where Monte Carlo methods comes 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:

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

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

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

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

# 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).

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

All of this is to **eliminate the evaluation** (often impossible) of the **denominator** $P(\text{data})$.

# Markov Chains

- We proceed by defining an **ergodic Markov chain**[a] in which the set of possible states is the sample size and the stationary distribution is the distribution to be *approximated* (or *sampled*).
- Let $X_0, X_1, \ldots, X_n$ 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.

[a] meaning that there is an **unique stationary distribution**.

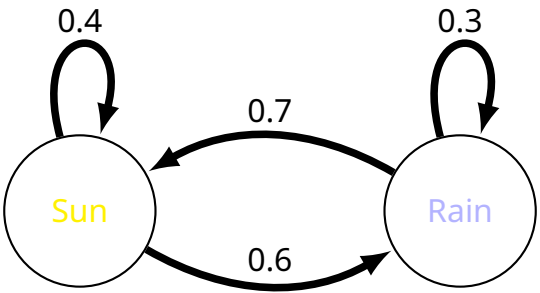
# Markov Chains

- 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**:

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

  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 $X_r, X_{2r}, X_{3r}, \ldots$ 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"[xxii] 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).

[xxii] some references call this "burnin".

# MCMC Algorithms

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

- Metropolis-Hastings (Hastings, 1970; Metropolis et al., 1953).
- Hamiltonian Monte Carlo[xxiv] (Betancourt, 2017; Neal, 2011).

[xxiii] see the Wikipedia page for a full list.
[xxiv] sometimes called Hybrid Monte Carlo, specially in the physics literature.

# MCMC Algorithms – Metropolis-Hastings

These are the first MCMC algorithms. They use an **acceptance/rejection rule for the proposals**. They are characterized by proposals originated from a random walk in the parameter space. The **Gibbs algorithm** can be seen as a **special case** of MH because all proposals are automatically accepted (Gelman, 1992)

Asymptotically, they have an acceptance rate of 23.4%, and the computational cost of every iteration is $\mathcal{O}(d)$, where $d$ is the number of dimension in the parameter space (Beskos et al., 2013).

# MCMC Algorithms – Hamiltonian Monte Carlo

The current most efficient MCMC algorithms. They try to **avoid the random walk behavior by introducing an auxiliary vector of momenta using Hamiltonian dynamics**. The proposals are "guided" to higher density regions of the sample space. This makes **HMC more efficient in orders of magnitude when compared to MH and Gibbs**.

Asymptotically, they have an acceptance rate of 65.1%, and the computational cost of every iteration is $\mathcal{O}(d^{\frac{1}{4}})$, where $d$ is the number of dimension in the parameter space (Beskos et al., 2013).

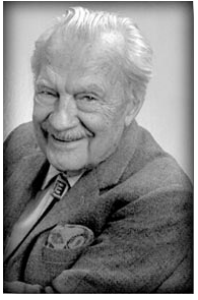
# Metropolis Algorithm

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** $J_t(\boldsymbol{\theta}^{(*)})$ to define the next values of the distribution $P^*(\boldsymbol{\theta}^{(*)} \mid \text{data})$. This distribution must be symmetric:

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



# 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:

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

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

# Metropolis Algorithm

**Algorithm 1:** Metropolis

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

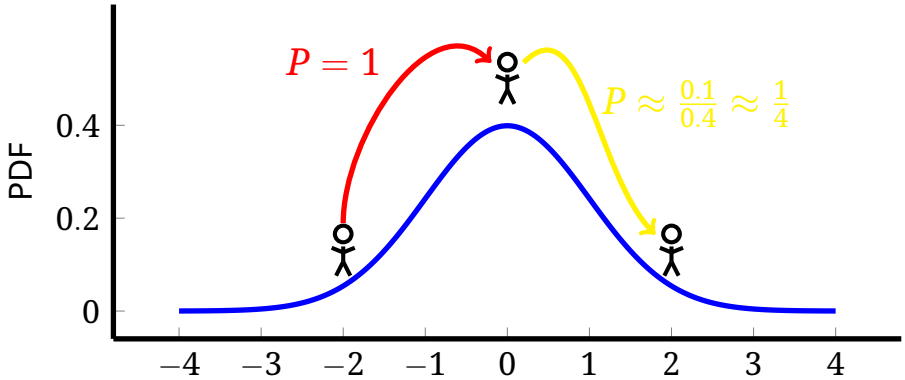
**for** $t = 1, 2, \ldots$

Sample a proposal of $\boldsymbol{\theta}^{(*)}$ from a proposal distribution in time $t$, $J_t\left(\boldsymbol{\theta}^{(*)} \mid \boldsymbol{\theta}^{(t-1)}\right)$

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

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

# 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**:

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

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 $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^p$ that $P\left(\boldsymbol{\theta}^{(0)} \mid \mathbf{y}\right) > 0$

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

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

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

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

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

# Metropolis-Hastings Animation[xxv]

Metropolis Animation

[xxv] see Metropolis-Hastings in action at `chi-feng/mcmc-demo`.

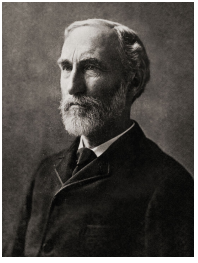
# Limitations of the Metropolis Algorithms

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 (Beskos et al., 2013; Roberts et al., 1997).
- 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.

# Gibbs Algorithm

To circumvent Metropolis' low acceptance rate, the Gibbs algorithm was conceived. Gibbs **do not have an acceptance/rejection rule** for the Markov chain state change: **all proposals are accepted!**
Gibbs algorithm was originally conceived by the physicist Josiah Willard Gibbs while referencing an analogy between a sampling algorithm and statistical physics (a physics field that originates from statistical mechanics). The algorithm was described by the Geman brothers in 1984 (Geman & Geman, 1984), about 8 decades after Gibbs death.



# Gibbs Algorithm

The Gibbs algorithm is very useful in multidimensional sample spaces. It is also known as **alternating conditional sampling**, because we always sample a parameter **conditioned** on the probability of the other model's parameters.

The Gibbs algorithm can be seen as a **special case** of the Metropolis-Hastings algorithm, because all proposals are accepted (Gelman, 1992).

The essence of the Gibbs algorithm is the sampling of parameters conditioned in other parameters:

$$P(\theta_1 \mid \theta_2, \ldots \theta_p)$$

# Gibbs Algorithm

**Algorithm 3:** Gibbs

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

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

Assign: $\boldsymbol{\theta}^{(t)} = \begin{cases} \theta_1^{(t)} & \sim P\left(\theta_1 \mid \theta_2^{(0)}, \ldots, \theta_p^{(0)}\right) \\ \theta_2^{(t)} & \sim P\left(\theta_2 \mid \theta_1^{(t-1)}, \ldots, \theta_p^{(0)}\right) \\ & \vdots \\ \theta_p^{(t)} & \sim P\left(\theta_p \mid \theta_1^{(t-1)}, \ldots, \theta_{p-1}^{(t-1)}\right) \end{cases}$

# Gibbs Animation[xxvi]

Gibbs Animation

[xxvi] see Gibbs in action at `chi-feng/mcmc-demo`.

# Limitations of the Gibbs Algorithm

The main limitation of Gibbs algorithm is with relation to **alternating conditional sampling**:

- In Metropolis, the parameters' random proposals are sampled **unconditionally**, **jointly**, and **simultaneous**. The Markov chain state changes are executed in a **multidimensional** manner. This makes **multidimensional diagonal movements**.
- In the case of the Gibbs algorithm, this movement only happens one parameter at a time, because we sample parameters in a **conditional** and **sequential** manner with respect to other parameters. This makes **unidimensional horizontal/vertical movements**, and never multidimensional diagonal movements.

# Hamiltonian Monte Carlo (HMC)

Metropolis' low acceptance rate and Gibbs' low performance in multidimensional problems (where the posterior geometry is highly complex) made a new class of MCMC algorithms to 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:

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

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[xxvii] (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).

[xxvii] 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 efficiently than Metropolis and does *not* suffer Gibbs' parameters correlation issues

# Intuition Behind the HMC Algorithm

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

$$P(\boldsymbol{\theta}, \phi \mid y) = P(\phi) \cdot P(\boldsymbol{\theta} \mid y)$$

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:

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

# 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 4:** Hamiltonian Monte Carlo (HMC)

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

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

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

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

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

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

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

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

# HMC Animation[xxviii]

HMC Animation

[xxviii] see HMC in action at `chi-feng/mcmc-demo`.

# An Interlude into Numerical Integration

In the field of ordinary differential equations (ODE), we have the idea of "discretizing" a system of ODEs by applying a small step-size $\epsilon$[xxix]. Such approaches are called **numerical integrators** and are composed by an ample class of tools.

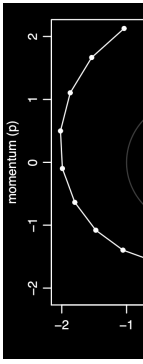
The most famous and simple of these numerical integrators is the Euler method, where we use a step-size $\epsilon$ to compute a numerical solution of system in a future time $t$ from specific initial conditions.

[xxix] sometimes also called $h$

# An Interlude into Numerical Integration

The problem is that Euler method, when applied to Hamiltonian dynamics, **does not preserve volume**. One of the fundamental properties of Hamiltonian dynamics if **volume preservation**[a]. This makes the Euler method a bad choice as a HMC's numerical integrator.

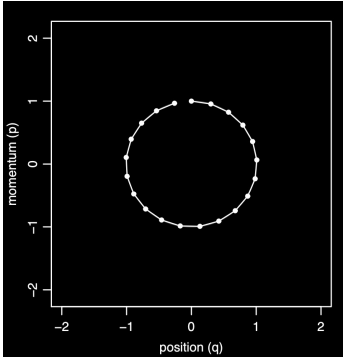
[a] a result called Liouville theorem.




HMC numerica
Euler with $\epsilon =$

# An Interlude into Numerical Integration[xxx]

To preserve volume, we need a numerical **symplectic integrator**. Symplectic integrators are at most second-order and demands a constant step-size $\epsilon$. One of the main numerical symplectic integrator used in Hamiltonian dynamics is the **Störmer–Verlet integrator**, also known as **leapfrog integrator**.

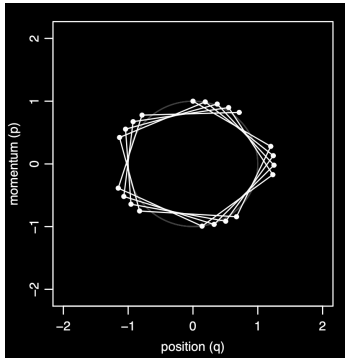



HMC numerically integrated using leapfrog with $\epsilon = 0.3$ and $L = 20$

[xxx] An excellent textbook for numerical and symplectic integrator is Iserles (2008).

# Limitations of the HMC Algorithm

As you can see, HMC algorithm is highly sensible 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 $L = 20$

# **N**o-**U**-**T**urn-**S**ampler (NUTS)

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

Here is where the idea for **N**o-**U**-**T**urn-**S**ampler (NUTS) (Hoffman & Gelman, 2011) enters: you don't need to **adjust anything**, just "press the button". It will automatically find $\epsilon$ and $L$.

# **N**o-**U**-**T**urn-**S**ampler (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 $\theta^{(*)}$:

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

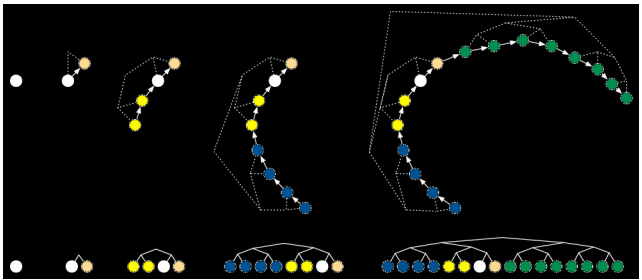
# **N**o-**U**-**T**urn-**S**ampler (NUTS)

This suggests an algorithms that does not allow proposals 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**.

# **N**o-**U**-**T**urn-**S**ampler (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.



NUTS growing leaf nodes forward

# **N**o-**U**-**T**urn-**S**ampler (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 5:** No-U-Turn-Sampler (NUTS)

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

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

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

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

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

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

- Choose a direction $v \sim$ Uniform $(\{-1, 1\})$
- Use the gradient of the log posterior $\boldsymbol{\theta}^{(*)}$ for a half-step of $\boldsymbol{\phi}$ in the direction $v$: $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + v\frac{1}{2}\epsilon\frac{d \log P(\boldsymbol{\theta}^{(*)}|\mathbf{y})}{d\boldsymbol{\theta}}$
- Use $\boldsymbol{\phi}$ to update $\boldsymbol{\theta}^{(*)}$: $\boldsymbol{\theta}^{(*)} \leftarrow \boldsymbol{\theta}^{(*)} + \epsilon\mathbf{M}^{-1}\boldsymbol{\phi}$
- Again use the gradient of the log posterior $\boldsymbol{\theta}^{(*)}$ for a half-step of $\boldsymbol{\phi}$ in the direction $v$: $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + v\frac{1}{2}\epsilon\frac{d \log P(\boldsymbol{\theta}^{(*)}|\mathbf{y})}{d\boldsymbol{\theta}}$
- Define the node $L_i^v$ as the proposal $\boldsymbol{\theta}$
- **if** *The difference between proposal vector* $\boldsymbol{\theta}^{(*)}$ *and current vector* $\boldsymbol{\theta}$ *in the direction* $v$ *is lower than zero:* $v\frac{d}{dt}\frac{(\boldsymbol{\theta}^{(*)}-\boldsymbol{\theta}^{(*)})\cdot(\boldsymbol{\theta}^{(*)}-\boldsymbol{\theta}^{(*)})}{2} < 0$ **then**
  - Stop sampling $\boldsymbol{\theta}^{(*)}$ in the direction $v$ and continue sampling only in the direction $-v$
- **if** *The difference between proposal vector* $\boldsymbol{\theta}^{(*)}$ *and current vector* $\boldsymbol{\theta}$ *in the direction* $-v$ *is lower than zero:* $-v\frac{d}{dt}\frac{(\boldsymbol{\theta}^{(*)}-\boldsymbol{\theta}^{(*)})\cdot(\boldsymbol{\theta}^{(*)}-\boldsymbol{\theta}^{(*)})}{2} < 0$ **then**
  - Stop sampling $\boldsymbol{\theta}^{(*)}$

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

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

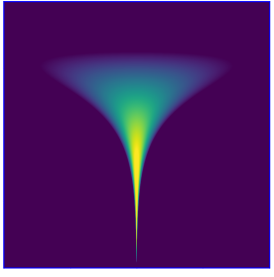
# NUTS Animation[xxxi]

Animação NUTS

[xxxi] see NUTS in action at `chi-feng/mcmc-demo`.

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

The famous "Devil's Funnel"[xxxii]. Here we see that HMC and NUTS, during the exploration of the posterior, have to change often $L$ and $\epsilon$ values[xxxiii].



[xxxii] very common em 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:

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

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:

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

# 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[xxxiv]:

- **target acceptance rate of Metropolis proposals**: 0.8
- **max tree depth** (in powers of 2): 10 (which means $2^{10} = 1024$)

[xxxiv] for more information about how to change those values, see Section 15.2 of the Stan Reference Manual.

# `Turing.jl` and NUTS

`Turing.jl` also implements NUTS which lives, along with other MCMC samplers, inside the package `AdvancedHMC.jl`. It also has an automatic optimized adjustment routine for values of $L$ and $\epsilon$ during warmup. It has the same default NUTS hyperparameters' values[xxxv]:

- **target acceptance rate of Metropolis proposals**: 0.65
- **max tree depth** (in powers of 2): 10 (which means $2^{10} = 1024$)

[xxxv] for more information about how to change those values, see `Turing.jl` Documentation.

# Markov Chain Convergence

MCMC has an interesting property that it will **asymptotically converge to the target distribution**[xxxvi].

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.

[xxxvi] this property is not present on neural networks.

# Convergence Metrics

We have some options on how to measure if the Markov chains converged to the target distribution, i.e. if they are “reliable”:

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

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

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

Where:

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

# Convergence Metrics – Rhat (Gelman et al., 2013b)

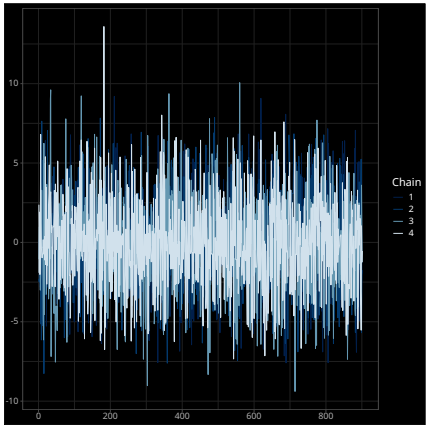
$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}(\psi \mid y)}{W}}$$

where $\widehat{\text{var}}^{+}(\psi \mid y)$ 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:

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

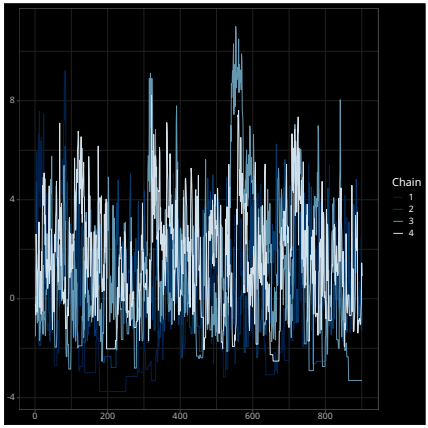
Intuitively, the value is 1.0 if all chains are totally convergent. As a heuristic, if $\widehat{R} > 1.1$, you need to worry because probably the chains have not converged adequate.

# Traceplot – Convergent Markov Chains




A convergent Markov chains traceplot

# Traceplot – Divergent Markov Chains




A divergent Markov chains traceplot

# Stan's Warning Messages[xxxvii]

```
Warning messages:
1: There were 275 divergent transitions after warmup. See
http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.
2: Examine the pairs() plot to diagnose sampling problems

3: The largest R-hat is 1.12, indicating chains have not mixed.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#r-hat
4: Bulk Effective Samples Size (ESS) is too low, indicating posterior
means and medians may be unreliable.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#bulk-ess
5: Tail Effective Samples Size (ESS) is too low, indicating posterior
variances and tail quantiles may be unreliable.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#tail-ess
```

[xxxvii] also see Stan's warnings guide.

# Turing.jl's Warning Messages

**Turing.jl does not give warning messages!** But you can check divergent transitions with `summarize(chn; sections=[:internals])`:

Summary Statistics

| parameters | **mean** | std | naive_**se** | mcse | ess | rhat | ess_per_sec |
|---|---|---|---|---|---|---|---|
| Symbol | Float64 | Float64 | Float64 | Float64 | Float64 | Float64 | Float64 |
| lp | −3.9649 | 1.7887 | 0.0200 | 0.1062 | 179.1235 | 1.0224 | 6.4133 |
| n_steps | 9.1275 | 11.1065 | 0.1242 | 0.7899 | 38.3507 | 1.3012 | 1.3731 |
| acceptance_rate | 0.5944 | 0.4219 | 0.0047 | 0.0322 | 40.5016 | 1.2173 | 1.4501 |
| tree_depth | 2.2444 | 1.3428 | 0.0150 | 0.1049 | 32.8514 | 1.3544 | 1.1762 |
| numerical_error | 0.1975 | 0.3981 | 0.0045 | 0.0273 | 59.8853 | 1.1117 | 2.1441 |

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

**First**: before making any fine adjustments in the number of Markov chains or the number of iterations per chain, etc. Acknowledge that both `Stan`'s and `Turing.jl`'s NUTS sampler is **very efficient and effective in exploring the most crazy and diverse target posterior densities**.

And the standard settings, **2,000 iterations and 4 chains**, works perfectly for 99% of the time.

# 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?

If you experiencing convergence issues, **and you've discarded that something is wrong with you model**, here is a few steps to try[xxxviii]. Here listed in increasing complexity:

1. **Increase the number of iterations and chains**: try first increasing the number of iterations, then try increasing the number of chains. (remember the default is 2,000 iterations and 4 chains).

[xxxviii] besides that, maybe should be worth to do a QR decomposition in the data matrix $\mathbf{X}$, thus having an orthogonal basis (non-correlated) for the sampler to explore. This makes the target distribution's geometry much more friendlier, in the topological/geometrical sense, for the MCMC sampler explore. Check the backup slides.

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

2. **Change the HMC's warmup adaptation routine**: make the HMC sampler to be more conservative in the proposals. This can be changed by increasing the hyperparameter **target acceptance rate of Metropolis proposals**[xxxix]. The maximum value is 1.0 (not recommended). Então qualquer valor entre 0.8 e 1.0 o torna mais conservador.

3. **Model reparameterization**: there are two approaches. Centered parameterization (CP) and non-centered parameterization (NCP).

[xxxix] `Stan`'s default is `0.8` and `Turing.jl`'s default is `0.65`.

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

4. **Collect more data**: sometimes the model is too complex and we need a higher sample size for stable estimates.

5. **Rethink the model**: convergence issues with an adequate sample size might be due to incompatibility between priors and likelihood function(s). In this case you need to rethink the whole data generating process underlying the model, in which the model assumptions stems from.

# 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 Posterior Predictive Checks?

To analyze and compare models using predictive accuracy 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:

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

We compute the divergence by multiplying entropy by $-2$[xl], so lower values are preferable:

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

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

[xl] historical reasons.

# Historial Interlude – AIC (Akaike, 1973)

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

where $k$ is the number of the model's free parameters and $\text{lppd}_{\text{mle}}$ is the **m**aximum **l**ikelihood **e**stimate of the **l**og **p**ointwise **p**redictive **d**ensity.

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:

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

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[xli].

We can discuss **predictive accuracy objective metrics**

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

[xli] 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:*

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

*Where $p(\theta \mid y)$ 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 $p(\tilde{y} \mid y)$, 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:

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

where elpd is the **e**xpected **l**og **p**ointwise **p**redictive **d**ensity, 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):

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

where

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

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

[xlii] upcoming...

# 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:

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

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

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

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, which almost involves a **high computational cost**.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

PSIS uses **importance sampling**[xliii], which means a importance weighting scheme approach.

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

[xliii] another class of MCMC algorithm that we did not cover yet.

# Importance Sampling

If the $N$ samples are conditionally independent[xliv] (Gelfand et al., 1992), we can compute LOO with $\theta^s$ posterior' samples $P(\theta \mid y)$ 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):

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

[xliv] 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 $P(\theta \mid y)$ often has low variance and shorter tails than the LOO distributions $P(\theta \mid y_{-1})$. 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 $\widehat{k}$ 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 $\widehat{k} > 0.5$ is a warning sign, but empirically there is still a good performance up to $\widehat{k} < 0.7$.

# 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).

Bates, D., Alday, P., Kleinschmidt, D., José Bayoán Santiago Calderón, P., Zhan, L., Noack, A., Bouchet-Valat, M., Arslan, A., Kelman, T., Baldassari, A., Ehinger, B., Karrasch, D., Saba, E., Quinn, J., Hatherly, M., Piibeleht, M., Mogensen, P. K., Babayan, S., & Gagnon, Y. L. (2022). Juliastats/mixedmodels.jl. https://doi.org/10.5281/ZENODO.6925652

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

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. 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

Brooks, S., Gelman, A., Jones, G., & Meng, X.-L. (2011, May 10). *Handbook of Markov Chain Monte Carlo*. CRC Press.

# References II


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

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

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.

# 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. (1992). Iterative and Non-Iterative Simulation Algorithms. *Computing Science and Statistics (Interface Proceedings)*, *24*, 457–511.

Gelman, A. (2008). *The folk theorem of statistical computing*. 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]. Retrieved February 4, 2021, from http://arxiv.org/abs/2011.01808

Geman, S., & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*(6), 721–741. https://doi.org/10.1109/TPAMI.1984.4767596

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

Iserles, A. (2008). *A first course in the numerical analysis of differential equations* (2nd). Cambridge University Press.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press.

# References V


Khan, M. E., & Rue, H. (2021, July 9). *The Bayesian Learning Rule*. arXiv: 2107.04562 [cs, stat]. Retrieved July 13, 2021, from http://arxiv.org/abs/2107.04562

Kolmogorov, A. N. (1933). *Foundations of the Theory of Probability*. Julius Springer.

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.

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.

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.

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.

# References VII

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).*

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

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

# Mathematical Notation

We use the notation

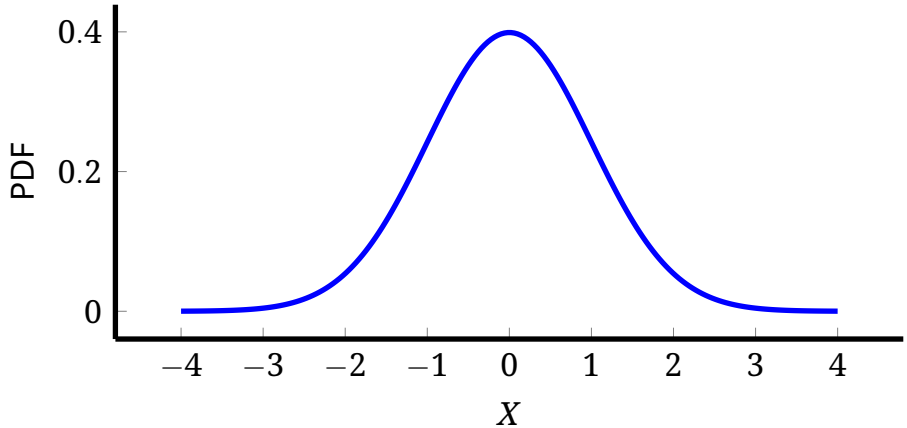
$$X \sim \text{Dist}(\theta_1, \theta_2, \dots)$$

where:

- $X$: random variable
- 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

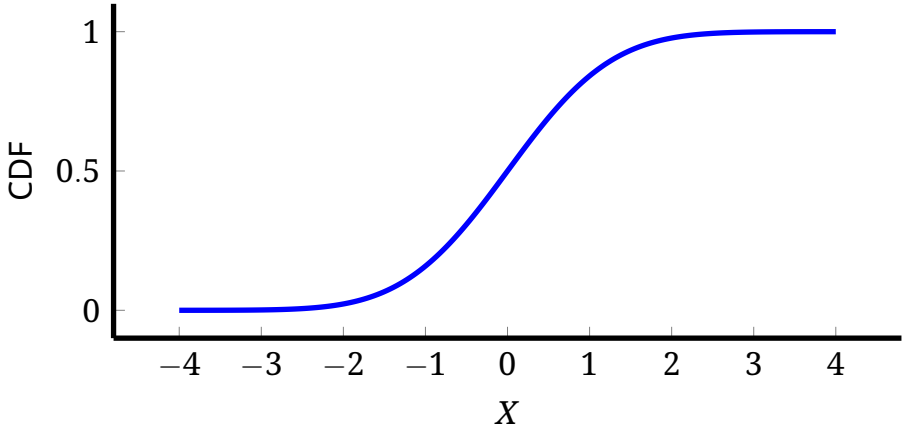


## Definition (Cumulative Distribution Function)

*The cumulative distribution function (CDF) of a random variable $X$ evaluated at $x$ is the probability that $X$ will take values less or qual 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:* $-N, \ldots, -2, 1, 0, 1, 2, \ldots, N$ *e* $N \in \mathbb{Z}$*. 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* $x$*:*

$$PMF(x) = P(X = x)$$

# 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 Uniform$(a, b)$:

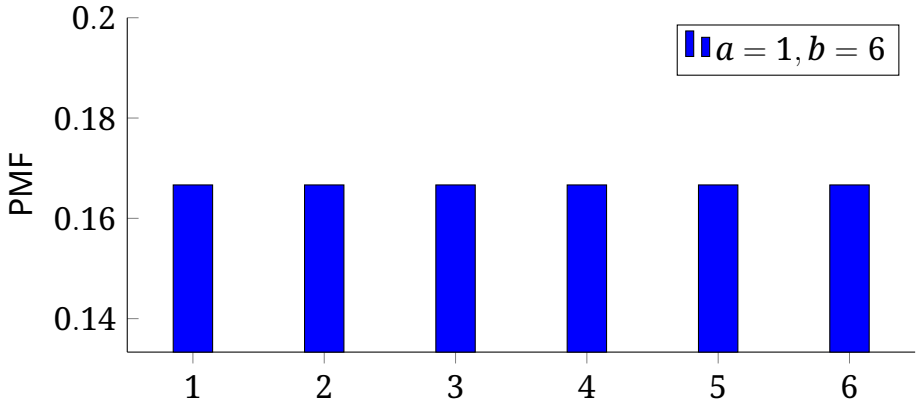
- $a$ – lower bound
- $b$ – upper bound

Example: dice.

# Discrete Uniform

$$\text{Uniform}(a, b) = f(x, a, b) = \frac{1}{b - a + 1} \text{ for } a \leq x \leq b \text{ and } x \in \{a, a + 1, \ldots, b - 1$$

# 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 Bernoulli($p$):

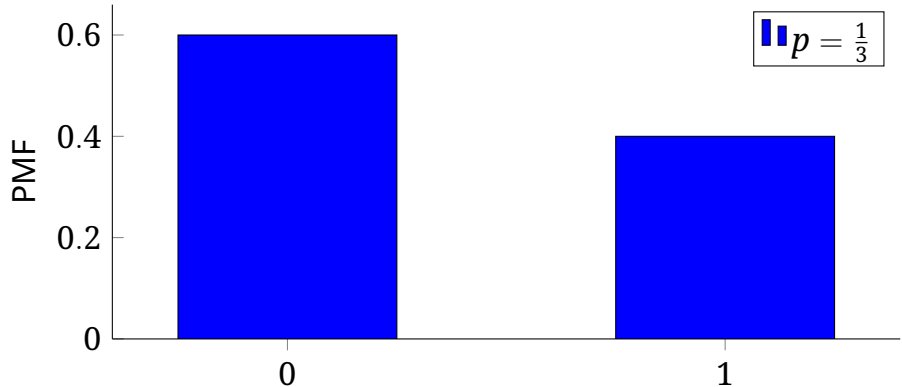
- $p$ – probability of success

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 Binomial$(n, p)$ :

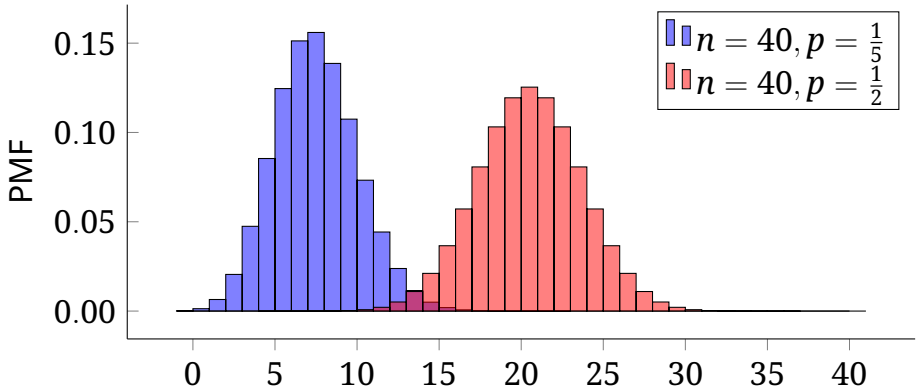
- $n$ – number of experiments
- $p$ – probability of success

Example: number of heads in five coin throws.

# Binomial

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

# Binomial


PMF
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
35
40
$n = 40, p = \frac{1}{5}$
$n = 40, p = \frac{1}{2}$

# 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 Poisson($\lambda$):

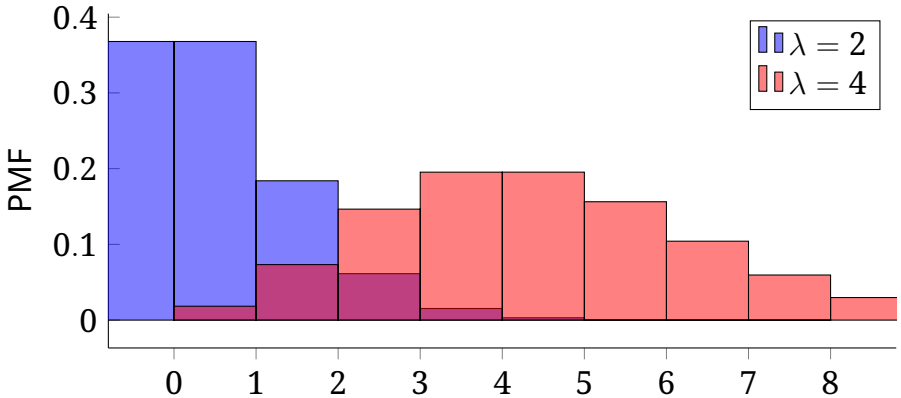
- $\lambda$ – rate

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[xlv]

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 $p$ until $k$ successes. Notice that it becomes the Poisson distribution in the limit as $k \to \infty$. This makes it a robust option to replace a Poisson distribution to model phenomena with overdispersion (presence of greater variability in data than would be expected).

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

- $k$ – number of successes
- $p$ – probability of success

Example: annual occurrence of tropical cyclones.

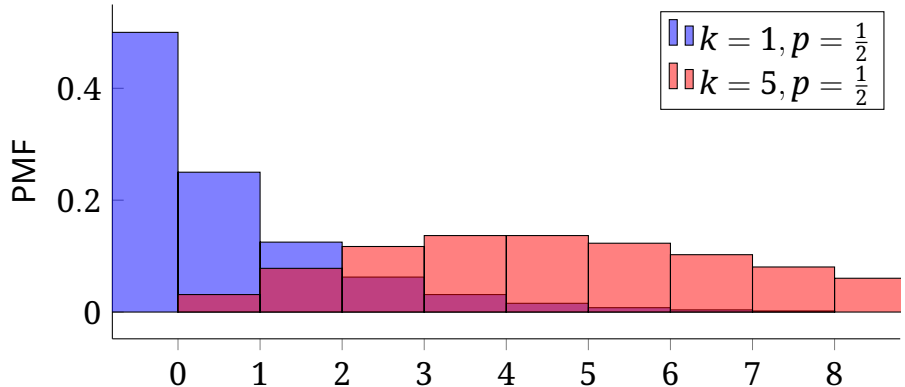
[xlv] any phenomena that can be modeles as a Poisson distribution can be modeled also as negative binomial distribution (Gelman et al., 2013b; Gelman, Hill, & Vehtari, 2020).

# Negative Binomial

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

$$\text{for } x \in \{0, 1, \ldots, n\}$$

# 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:*

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

# 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 $\frac{1}{n}$.

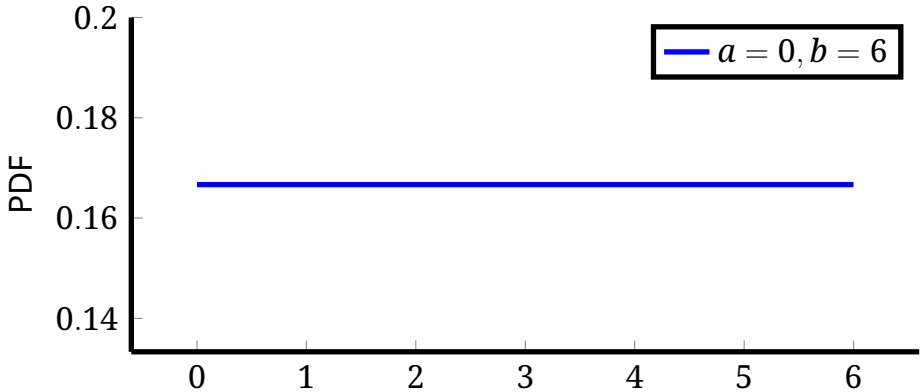
The continuous uniform distribution has two parameters and its notation is Uniform$(a, b)$:

- $a$ – 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 $n \to \infty$).

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 Normal($\mu, \sigma^2$) or N($\mu, \sigma^2$):

- $\mu$ – mean of the distribution, and also median and mode
- $\sigma$ – standard deviation[xlvi], a dispersion measure of how observations occur in relation from the mean

Example: height, weight etc.

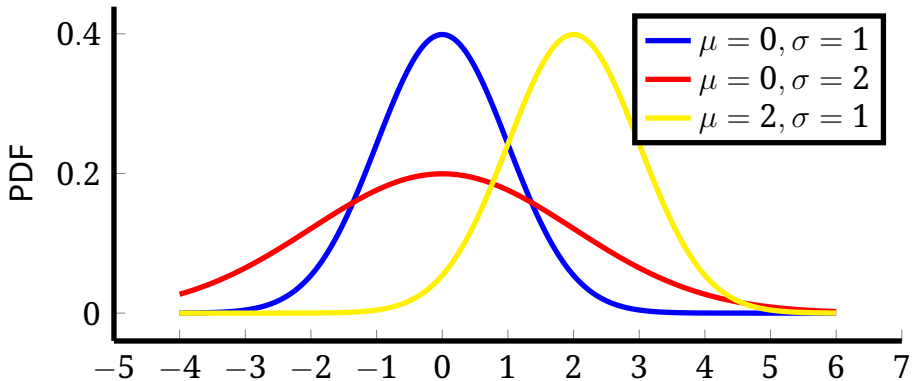
[xlvi] sometimes is also parameterized as variance $\sigma^2$.

# Normal[xlvii]

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

[xlvii] 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 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 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

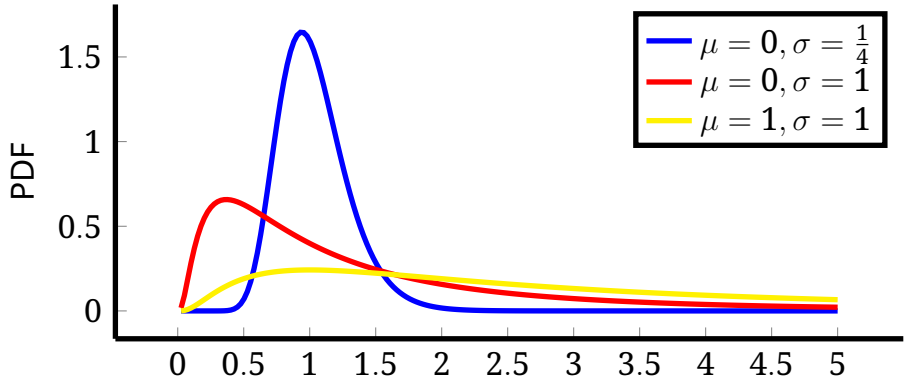
The log-normal distribution has two parameters and its notation is Log-Normal($\mu, \sigma^2$):

- $\mu$ – 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


PDF
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
$\mu = 0, \sigma = \frac{1}{4}$
$\mu = 0, \sigma = 1$
$\mu = 1, \sigma = 1$

# Exponential

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

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

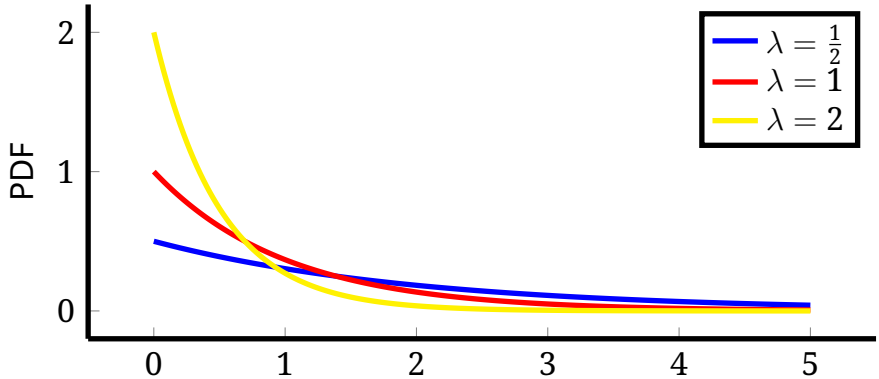
- $\lambda$ – rate

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

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

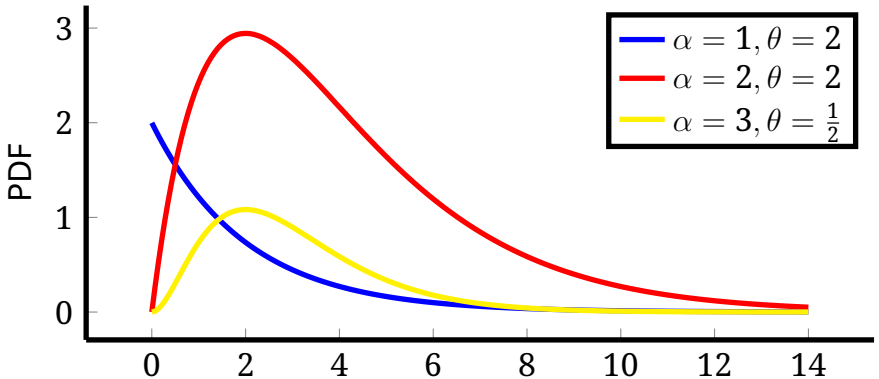
- $\alpha$ – shape parameter
- $\theta$ – rate parameter

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


PDF
3
2
1
0
0
2
4
6
8
10
12
14
$\alpha = 1, \theta = 2$
$\alpha = 2, \theta = 2$
$\alpha = 3, \theta = \frac{1}{2}$

# 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[xlviii].

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.

[xlviii] this is where the ubiquitous Student's *t* test.

# Student's *t*

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

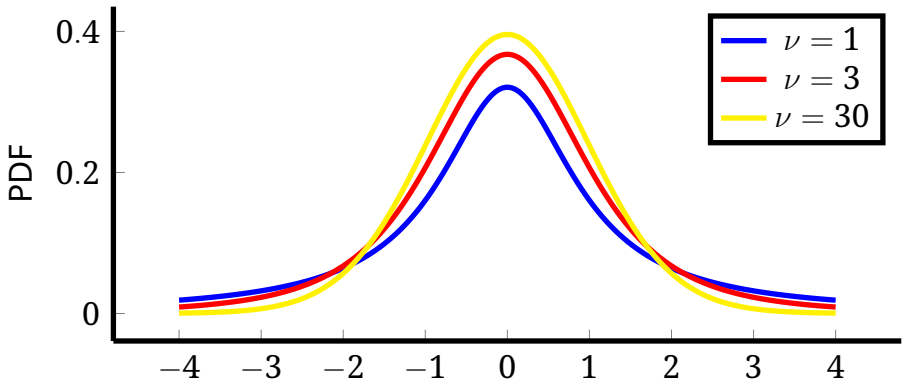
- $\nu$ – degrees of freedom, controls how much it resembles a normal distribution

Example: a dataset full of outliers.

# Student's $t$

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

# 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 Gamma$(\alpha, \theta)$:

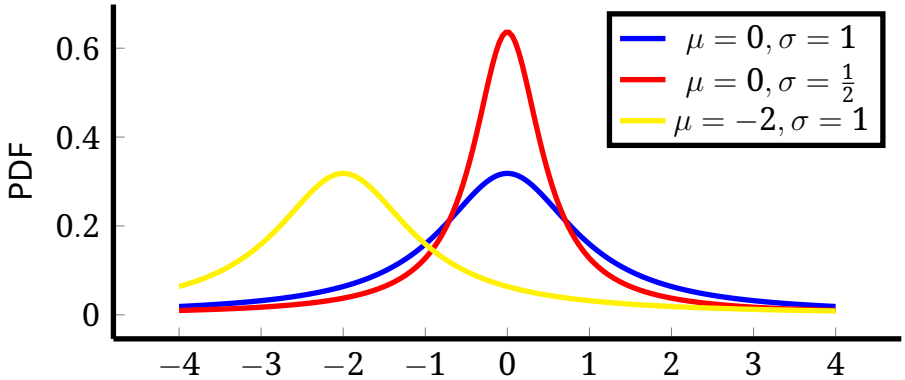
- $\mu$ – location parameter
- $\sigma$ – scale parameter

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


PDF
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
$\mu = 0, \sigma = 1$
$\mu = 0, \sigma = \frac{1}{2}$
$\mu = -2, \sigma = 1$

# Beta

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

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

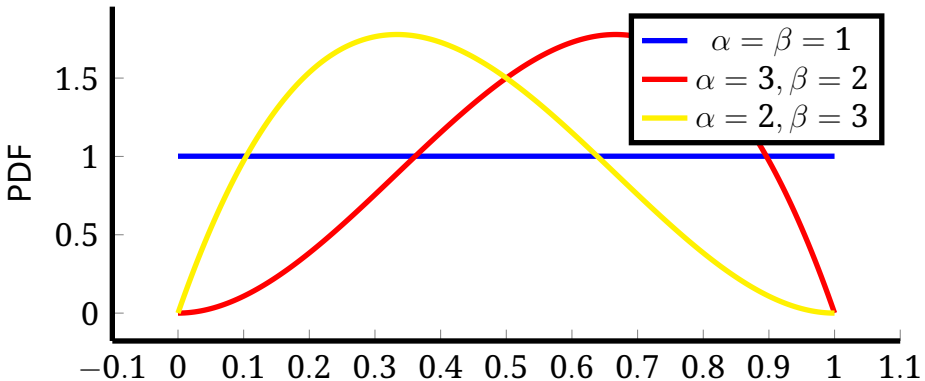
- $\alpha$ or sometimes $a$ – shape parameter, controls how much the shape is shifted towards $1$
- $\beta$ or sometimes $b$ – shape parameter, controls how much the shape is shifted towards $0$

Example: A basketball player that has already scored 5 free throws and missed 3 in a total of 8 attempts – 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


PDF
1.5
1
0.5
0
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
$\alpha = \beta = 1$
$\alpha = 3, \beta = 2$
$\alpha = 2, \beta = 3$

# Matrix-variate Distributions

So far we've seen random distributions that returns 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-semidefinite if it satisfies the following equivalent conditions:*

- $\mathbf{M}$ *is congruent with a diagonal matrix with positive real entries*
- $\mathbf{M}$ *is symmetric or Hermitian, and all its eigenvalues are real and positive*
- $\mathbf{M}$ *is symmetric or Hermitian, and all its leading principal minors are positive*
- *There exists an invertible matrix* $\mathbf{B}$ *with conjugate transpose* $\mathbf{B}^*$ *such that* $\mathbf{M} = \mathbf{B}^*\mathbf{B}$

# Covariance Matrices

## Example (Positive Semi-Definite Matrix)

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

Conversely, **every positive semi-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

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

where:

- $p$ dimensionality of the matrix-variate distribution
- $\mathbf{\Sigma}$ 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
- $\mathbf{\Psi}$ scale matrix $p \times p$

# LKJ[xlix]

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

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

where:

- $p$ dimensionality of the matrix-variate distribution
- $\mathbf{\Sigma}$ 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.

[xlix] Lewandowski et al. (2009) – LKJ are the authors' last name initials – **L**ewandowski, **K**urowicka and **J**oe.

# LKJ

One interesting property of the LKJ distribution is that if we disregard the product over the beta functions[l], then the PDF is proportional to the determinant exponentiated by $\eta - 1$:

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

where:

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

[l] generally the beta function can be expressed as a fraction of two gamma functions.

# Priors for Covariance Matrices

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

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

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

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

$$\mathbf{C} = \mathbf{L_C} \mathbf{L_C}^T$$

where $\mathbf{L_C}$ is a lower-triangular matrix.

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