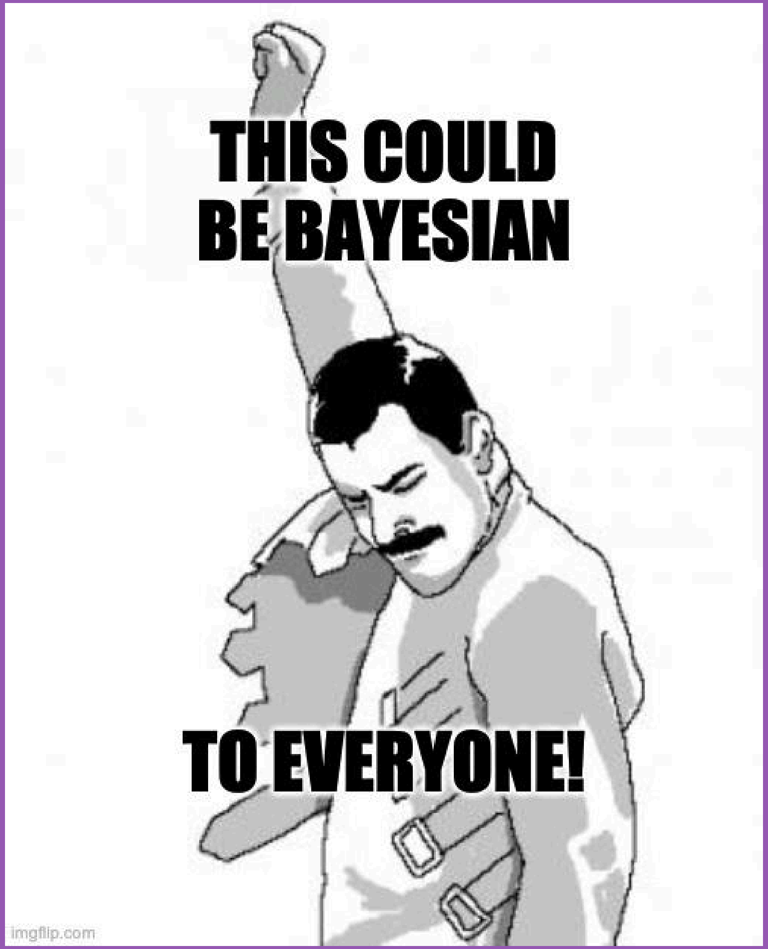
Bayesian Statistics

Jose Storopoli[i]

[i] Universidade Nove de Julho, Pumas-AI


THIS COULD
BE BAYESIAN
TO EVERYONE!
imgflip.com

License

The text and images from these slides have a

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

All links are in blue. Feel free to click on them.

Outline

1. Tools
2. Bayesian Statistics
3. Statistical Distributions
4. Priors
5. Bayesian Workflow
6. Linear Regression
7. Logistic Regression
8. Ordinal Regression
9. Poisson Regression
10. Robust Regression
11. Sparse Regression
12. Hierarchical Models
13. Markov Chain Monte Carlo (MCMC) and Model Metrics
14. Model Comparison
15. Backup Slides

Tools

Recommended References

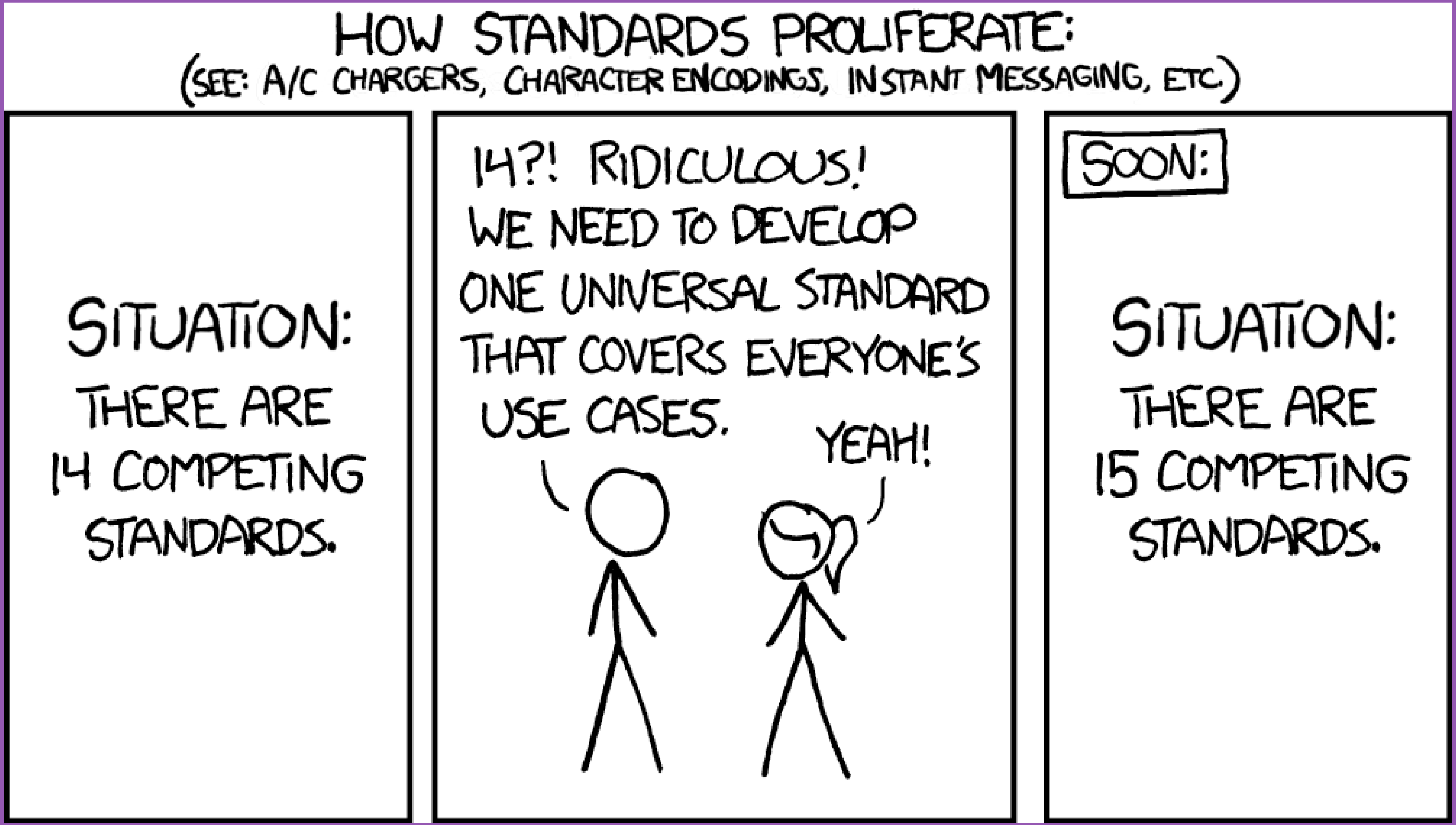
- Ge, Xu and Ghahramani (2018) - Turing paper
- Carpenter *et al.* (2017) - Stan paper
- Salvatier, Wiecki and Fonnesbeck (2016) - PyMC paper
- Bayesian Statistics with Julia and Turing - Why Julia?

A man and his tools make a man and his trade

— Vita Sackville-West

We shape our tools and then the tools shape us

— Winston Churchill


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Tools

- Stan (BSD-3 License)
- Turing (MIT License)
- PyMC (Apache License)
- JAGS (GPL License)
- BUGS (GPL License)

Stan[ii]

- High-performance platform for statistical modeling and statistical computation
- Financial support from NUMFocus:
    - AWS Amazon
    - Bloomberg
    - Microsoft
    - IBM
    - RStudio
    - Facebook
    - NVIDIA
    - Netflix
- Open-source language, similar to C++
- Markov Chain Monte Carlo (MCMC) parallel sampler

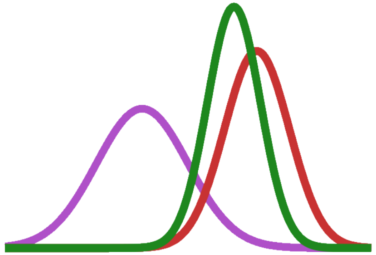
[ii] Carpenter *et al.* (2017)

Stan Code Example

```
data {
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  alpha ~ normal(0, 20);
  beta ~ normal(0, 2);
  sigma ~ cauchy(0, 2.5);
  y ~ normal(alpha + beta * x, sigma);
}
```

Turing[iii]

- Ecosystem of Julia packages for Bayesian Inference using probabilistic programming
- Julia is a fast dynamic-typed language that just-in-time (JIT) compiles into native code using LLVM: “runs like C but reads like Python” ; meaning that is *blazing* fast, easy to prototype and read/write code
- Julia has Financial support from NUMFocus
- Composability with other Julia packages
- Several other options of Markov Chain Monte Carlo (MCMC) samplers



[iii] Ge, Xu and Ghahramani (2018)

Turing Ecosystem

We have several Julia packages under Turing's GitHub organization TuringLang, but I will focus on 6 of those:

- Turing: main package that we use to **interface with all the Turing ecosystem** of packages and the backbone of everything
- MCMCChains: interface to **summarizing MCMC simulations** and has several utility functions for **diagnostics** and **visualizations**
- DynamicPPL: specifies a domain-specific language for Turing, entirely written in Julia, and it is modular
- AdvancedHMC: modular and efficient implementation of advanced Hamiltonian Monte Carlo (HMC) algorithms
- DistributionsAD: defines the necessary functions to enable automatic differentiation (AD) of the log PDF functions from Distributions
- Bijectors: implements a set of functions for transforming constrained random variables (e.g. simplexes, intervals) to Euclidean space

Turing[iv] Code Example

```
@model function linreg(x,  y)
    α ~ Normal(0, 20)
    β ~ Normal(0, 2)
    σ ~ truncated(Cauchy(0, 2.5); lower=0)

    y .~ Normal(α .+ β * x, σ)
end
```

[iv] I believe in Julia’s potential and wrote a whole set of Bayesian Statistics tutorials using Julia and Turing (Storopoli, 2021)

PyMC[v]

- Python package for Bayesian statistics with a Markov Chain Monte Carlo sampler
- Financial support from NUMFocus
- Backend was based on Theano
- Theano **died**, but PyMC developers create a fork named Aesara
- We have no idea what will be the backend in the future. PyMC developers are still experimenting with other backends: TensorFlow Probability, NumPyro, BlackJAX, and so on …

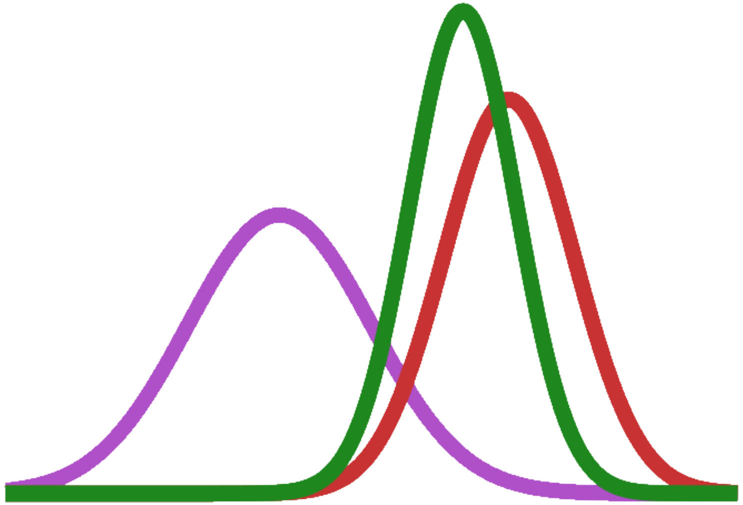
[v] Salvatier, Wiecki and Fonnesbeck (2016)

PyMC Code Example

```python
with pm.Model() as model:
    alpha = pm.Normal("Intercept", mu=0, sigma=20)
    beta = pm.Normal("beta", mu=0, sigma=2)
    sigma = pm.HalfCauchy("sigma", beta=2.5)

    likelihood = pm.Normal("y",
                 mu=alpha + beta * x1,
                 sigma=sigma, observed=y)
```

Which Tool Should You Use?



Turing

Stan

Why Turing

- **Julia** all the way down...
- Can **interface/compose** *any* Julia package
- Decoupling of **modeling DSL, inference algorithms and data**
- Not only HMC-NUTS, but a whole **plethora of MCMC algorithms**, e.g. Metropolis-Hastings, Gibbs, SMC, IS etc.
- Easy to **create/prototype/modify inference algorithms**
- **Transparent MCMC workflow**, e.g. iterative sampling API allows step-wise execution and debugging of the inference algorithm
- Very easy to **do stuff in the GPU**, e.g. NVIDIA's CUDA.jl, AMD's AMDGPU.jl, Intel's oneAPI.jl, and Apple's Metal.jl
- Very easy to do **distributed model inference and prediction**.

Why *Not* Turing

- **Not as fast**, but pretty close behind, as Stan.
- **Not enough learning materials**, example models, tutorials. Also documentation is somewhat lacking in certain areas, e.g. Bijectors.jl.
- **Not as many citations as Stan**, although not very far behind in GitHub stars.
- **Not well-known in the academic community**.

Why Stan

- API for R, Python and Julia.
- Faster than Turing.jl in 95% of models.
- **Well-known in the academic community**.
- **High citation count**.
- **More tutorials, example models, and learning materials available**.

Why *Not* Stan

- If you want to try **something new**, you'll have to do in **C++**.
- Constrained **only to HMC-NUTS** as MCMC algorithm.
- **Cannot decouple model DSL from data** (and also from inference algorithm).
- **Does not compose well with other packages**. For anything you want to do, it has to "exist" in the Stan world, e.g. bayesplot.
- A **not so easy and intuitive ODE interface**.
- **GPU interface depends on OpenCL**. Also not easy to interoperate.

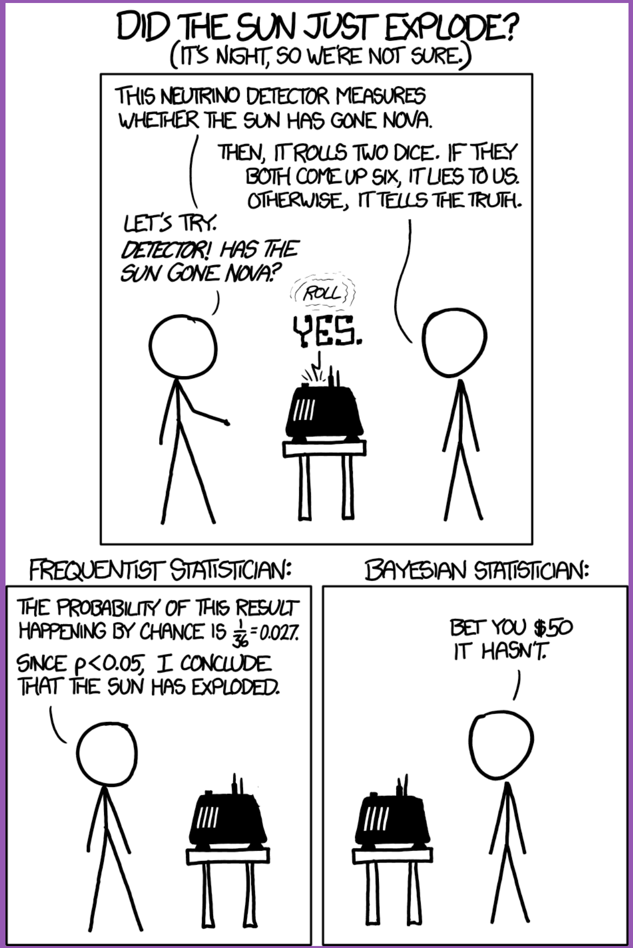
Bayesian Statistics

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - 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)

Inside every nonBayesian there is a Bayesian struggling to get out

— Denis Lindley


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36}=0.027$.
SINCE $p<0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

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. (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

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?

- **Flexibility** - probabilistic building blocks to construct a model[vi]:
  - Probabilistic conjectures about parameters:
    - Prior
    - Likelihood
- Better **uncertainty** treatment:
  - Coherence
  - Propagation
  - We don’t use *“if we sampled infinite times from a population that we do not observe…”*
- No p**-values**:
  - All statistical intuitions makes **sense**
  - 95% certainty that θ’s parameter value is between x and y
  - Almost **impossible** to perform p-hacking

[vi] like LEGO

A little bit more formal

- Bayesian Statistics uses probabilistic statements:
  - one or more parameters θ
  - 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, condition on the observed data from any covariate x

Definition of Bayesian Statistics

The use of Bayes theorem as the procedure to **estimate parameters of interest θ or unobserved data $\tilde{y}$**. (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013)

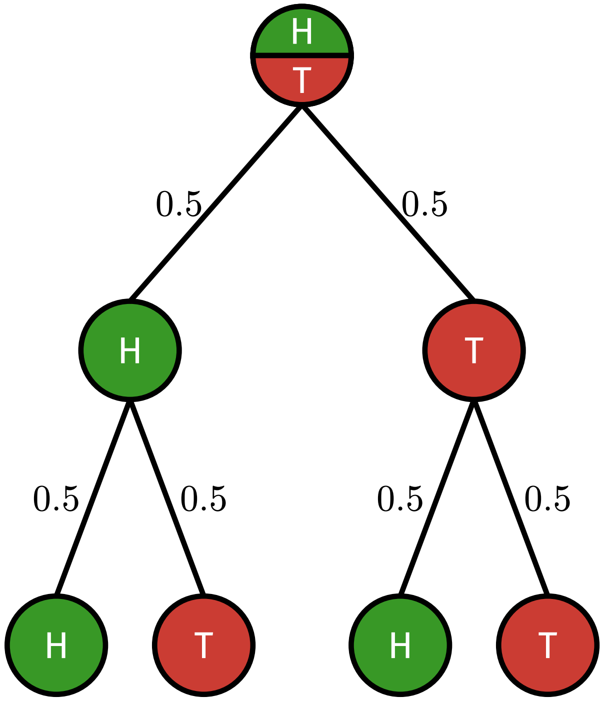
PROBABILITY DOES NOT EXIST![vii]

- Yes, probability does not exist ...
- Or even better, probability as a physical quantity, objective chance, **does NOT exist**
- if we disregard objetive chance *nothing is lost*
- The math of inductive rationality remains **exactly the same**

[vii] Finetti (1974)

PROBABILITY DOES NOT EXIST![viii]

- Consider flipping a biased coin
- The trials are considered independent and, as a result, have an important property: **the order does not matter**
- The frequency is considered a **sufficient statistic**
- Saying that order does not matter or saying that the only thing that matters is frequency are two ways of saying the same thing
- We say that the probability is **invariant under permutations**



[viii] Finetti (1974)

Probability Interpretations

- **Objective** - frequency in the long run for an event:
  - $P(\text{rain}) = \frac{\text{days that rained}}{\text{total days}}$
  - $P(\text{me being elected president}) = 0$ (never occurred)

- **Subjective** - degrees of belief in an event:
  - $P(\text{rain}) = \text{degree of belief that will rain}$
  - $P(\text{me being elected president}) = 10^{-10}$ (highly unlikely)

What is Probability?

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.

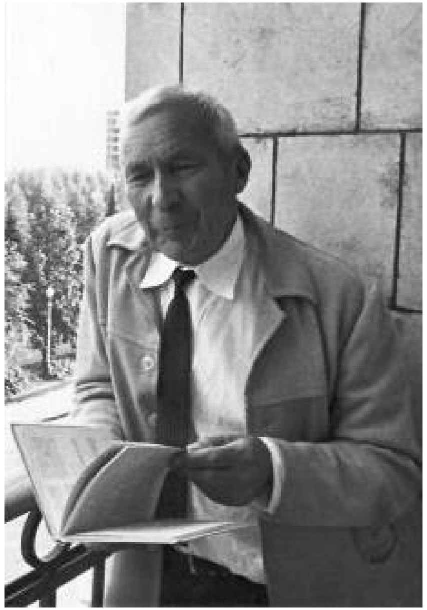
$$
\begin{aligned}
P(A) &\in \mathbb{R} \\
P(A) &\in [0,1] \\
0 \leq P(A) &\leq 1
\end{aligned}
$$

Probability Axioms[ix]

- **Non-negativity**: For every A: $P(A) \geq 0$

- **Additivity**: For every two *mutually exclusive* A and B: $P(A) = 1 - P(B)$ and $P(B) = 1 - P(A)$

- **Normalization**: The probability of all possible events $A_1, A_2, \ldots$ must sum up to 1: $\sum_{n \in \mathbb{N}} A_n = 1$



[ix] Kolmogorov (1933)

Sample Space

- Discrete: $\Theta = \{1, 2, ...\}$

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

Discrete Sample Space

8 planets in our solar system:

- Mercury: ☿
- Venus: ♀
- Earth: ♁
- Mars: ♂
- Jupiter: ♃
- Saturn: ♄
- Uranus: ⛢
- Neptune: ♆

Discrete Sample Space

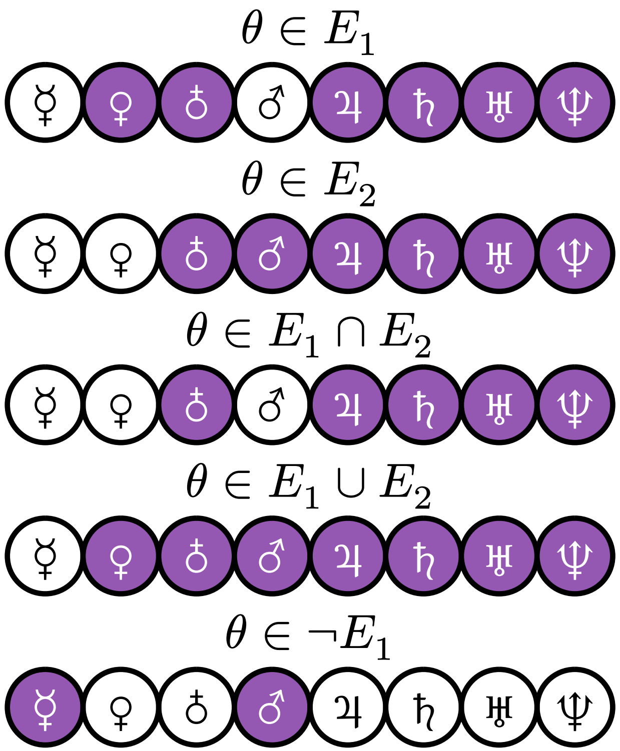
The planet has a magnetic field

The planet has moon(s)

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

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

The planet does *not* have a magnetic field



$\theta \in E_1$

$\theta \in E_2$

$\theta \in E_1 \cap E_2$

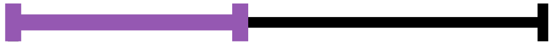
$\theta \in E_1 \cup E_2$

$\theta \in \neg E_1$

Continuous Sample Space

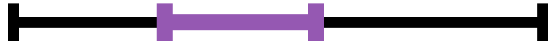
The distance is less than five centimeters

$\theta \in E_1$



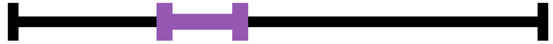
The distance is between three and seven centimeters

$\theta \in E_2$



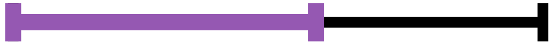
The distance is less than five centimeters
and between three and seven centimeters

$\theta \in E_1 \cap E_2$



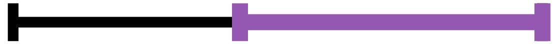
The distance is less than five centimeters
or between three and seven centimeters

$\theta \in E_1 \cup E_2$



The distance is *not* less than five centimeters

$\theta \in \neg E_1$



Discrete versus Continuous Parameters

Everything that has been exposed here was under the assumption that the parameters are discrete.

This was done with the intent to provide an intuition what is probability.

Not always we work with discrete parameters.

Parameters can be continuous, such as: age, height, weight etc. But don’t despair! All probability rules and axioms are valid also for continuous parameters.

The only thing we have to do is to change all s $\sum$ for integrals $\int$. For example, the third axiom of **Normalization** for *continuous* random variables becomes:

$$\int_{x \in X} p(x) \,\mathrm{d}x = 1$$

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”.

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

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

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

Example of Conditional Probability – Poker Texas Hold’em

- **Sample Space**: 52 cards in a deck, 13 types of cards and 4 types of suits.
- $P(A)$: Probability of being dealt an Ace $\left(\frac{4}{52} = \frac{1}{13}\right)$
- $P(K)$: Probability of being dealt a King $\left(\frac{4}{52} = \frac{1}{13}\right)$
- $P(A \mid K)$: Probability of being dealt an Ace, given that you have already a King $\left(\frac{4}{51} \approx 0.078\right)$
- $P(K \mid A)$: Probability of being dealt a King, given that you have already an Ace $\left(\frac{4}{51} \approx 0.078\right)$

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**[x]

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**: $P(\text{catholic} \mid \text{pope}) \neq P(\text{pope} \mid \text{catholic})$

[x] 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

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”.

$$
\begin{aligned}
P(A, B) &= \text{number of elements in A or B} \\
P(A, B) &= P(A \cup B) \\
P(A, B) &= P(B, A)
\end{aligned}
$$

Example of Joint Probability – Revisiting Poker Texas Hold’em

- **Sample Space**: 52 cards in a deck, 13 types of cards and 4 types of suits.
- $P(A)$: Probability of being dealt an Ace $\left(\frac{4}{52} = \frac{1}{13}\right)$
- $P(K)$: Probability of being dealt a King $\left(\frac{4}{52} = \frac{1}{13}\right)$
- $P(A \mid K)$: Probability of being dealt an Ace, given that you have already a King $\left(\frac{4}{51} \approx 0.078\right)$
- $P(K \mid A)$: Probability of being dealt a King, given that you have already an Ace $\left(\frac{4}{51} \approx 0.078\right)$
- $P(A, K)$: Probability of being dealt an Ace *and* being dealt a King

$$
\begin{gathered}
P(A, K) = P(K, A) \\
P(A) \cdot P(K \mid A) = P(K) \cdot P(A \mid K) \\
\frac{1}{13} \cdot \frac{4}{51} = \frac{1}{13} \cdot \frac{4}{51} \\
\approx 0.006
\end{gathered}
$$

Visualization of Joint Probability versus Conditional Probability

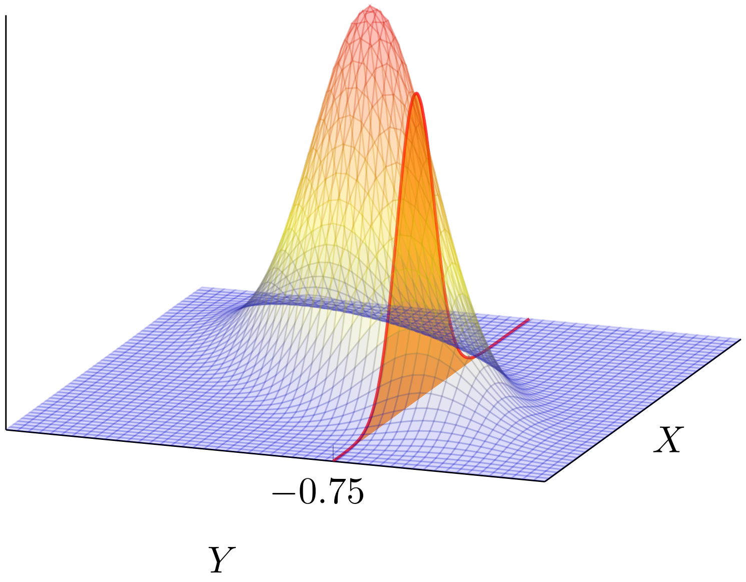


Figure 1: $P(X, Y)$ versus $P(X \mid Y = -0.75)$

Product Rule of Probability[xi]

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

$$
\begin{aligned}
P(A, B) &= P(B, A) \\
P(A) \cdot P(B \mid A) &= P(B) \cdot P(A \mid B)
\end{aligned}
$$

[xi] also called the Product Rule of 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

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:

$$
\begin{aligned}
P(A, B) &= P(B, A) \\
P(A) \cdot P(B \mid A) &= P(B) \cdot P(A \mid B)
\end{aligned}
$$

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

$$
\begin{aligned}
\frac{P(A) \cdot P(B \mid A)}{P(B)} &= \frac{P(B) \cdot P(A \mid B)}{P(B)} \\
\frac{P(A) \cdot P(B \mid A)}{P(B)} &= P(A \mid B) \\
P(A \mid B) &= \frac{P(A) \cdot P(B \mid A)}{P(B)}
\end{aligned}
$$

A Probability Textbook Classic[xii]

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)

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

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

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

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

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

Why Bayes' Theorem is Important?

We can invert the conditional probability:

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

But isn't this the p-value?

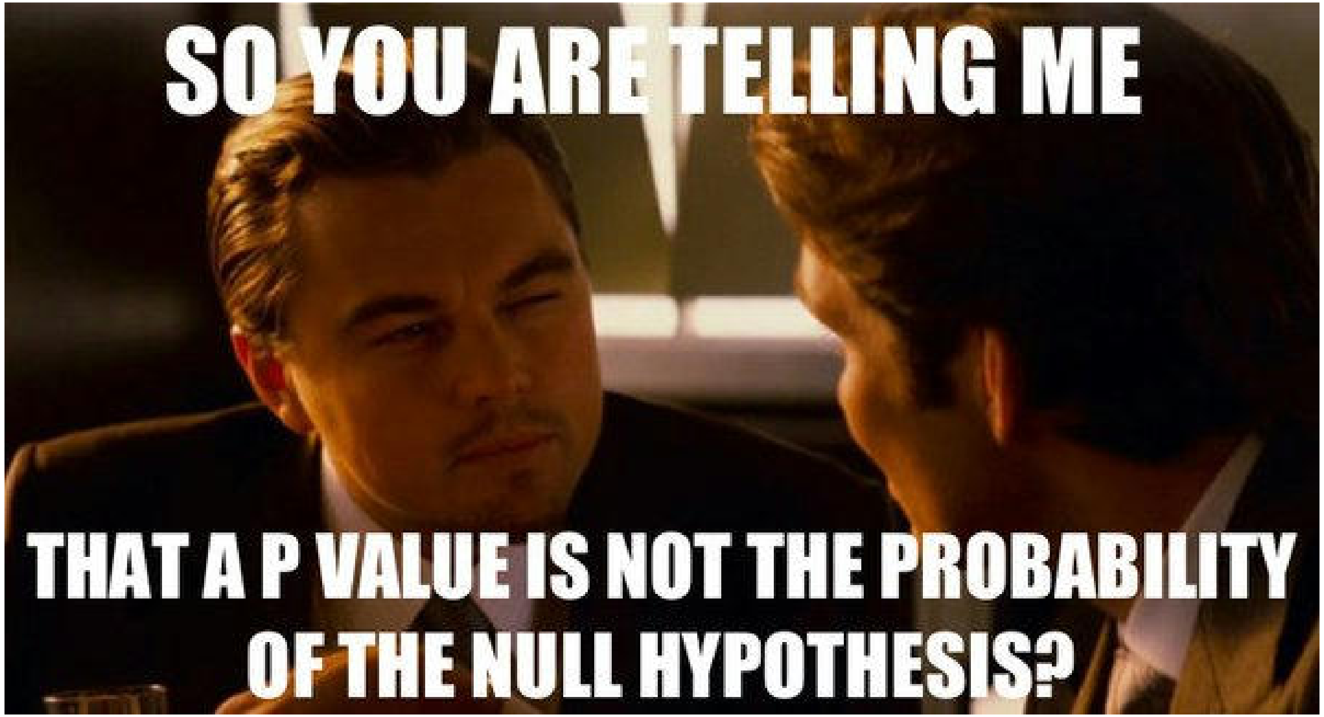
NO!

What are p-values?

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!


SO YOU ARE TELLING ME
THAT A P VALUE IS NOT THE PROBABILITY OF THE NULL HYPOTHESIS?

What p-value is **not**!

- **p-value is not the probability of the null hypothesis**
  - No!
  - 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 again!
  - 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).

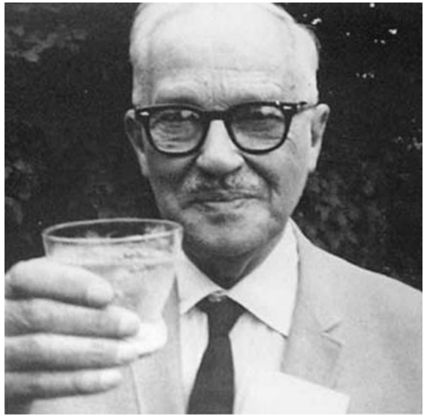
The relationship between p-value and H_0

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

What are 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?

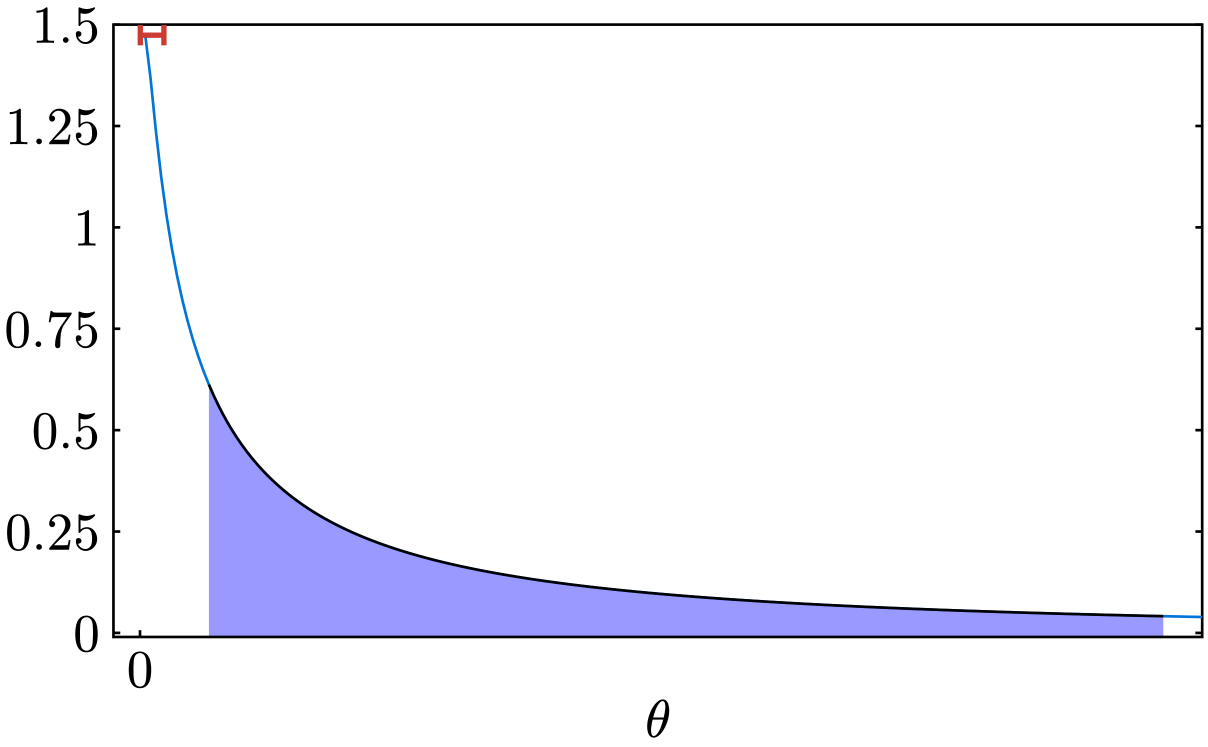
Say you performed a statistical analysis to compare the efficacy of a public policy between two groups and you obtain a difference between the mean of these groups. You can express this difference as a confidence interval. Often we choose 95% confidence.

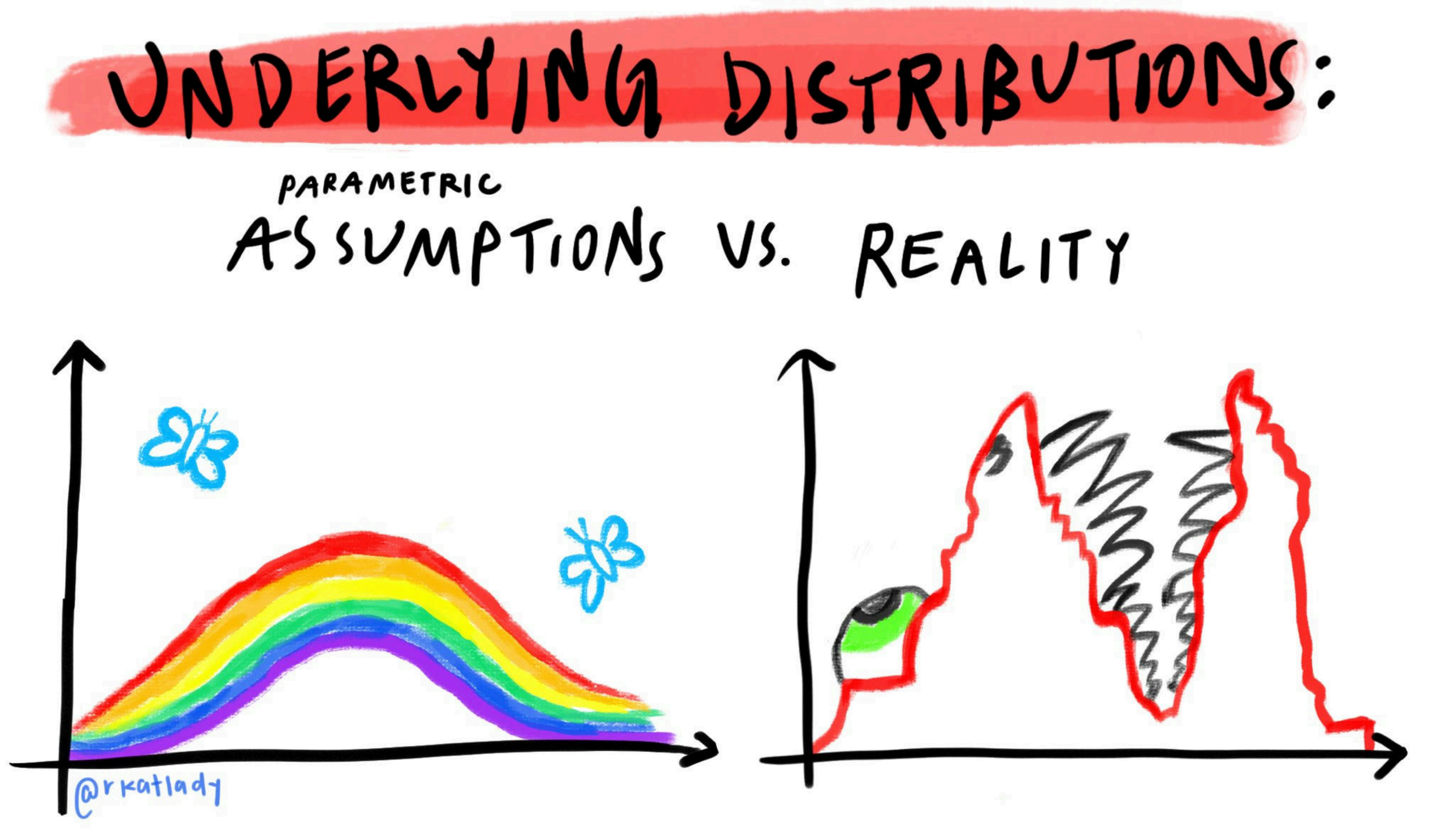
In other words, 95% is *not* the probability of obtaining data such that the estimate of the true parameter is contained in the interval that we obtained, it is the **probability of obtaining data such that, if we compute another confidence interval in the same way, it contains the true parameter**.

The interval that we got in this particular instance is irrelevant and might as well be thrown away.

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

Confidence Intervals versus Posterior Intervals




UNDERLYING DISTRIBUTIONS:
PARAMETRIC
ASSUMPTIONS VS. REALITY
@rkatlady

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:

> p-value is a measure of evidence against the null hypothesis
>
> — Fisher (1925)

- 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 and Rosenthal, 1989).

But why I never heard about Bayesian statistics?[xiii]

> … 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)

[xiii] *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[xiv]

- 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.

[xiv] 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:

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

- θ – 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**[xv]: $P(y)$ 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 and 1 – a valid probability.

[xv] 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	Objective (but with several model assumptions)
Uncertainty	Posterior Interval – $P(\theta \mid y)$	Confidence Interval – $P(y \mid \theta)$

Advantages of Bayesian Statistics

- Natural approach to express uncertainty
- Ability to incorporate previous information
- Higher model flexibility
- Full posterior distribution of the parameters
- Natural propagation of uncertainty

Main disadvantage: Slow model fitting procedure

The beginning of the end of Frequentist Statistics

- Know that you are in a very special moment in history of great changes in statistics
- I believe that frequentist statistics, specially the way we qualify evidence and hypotheses with p -values will transform in a “significant”[xvi] way.
- 8 years ago, the *American Statistical Association* (ASA) published a declaration about p-values (Wasserstein and Lazar, 2016). It states exactly what we exposed here: The main concepts of the null hypothesis significant testing and, in particular p-values, cannot provide what researchers demand of them. Despite what says several textbooks, learning materials and published content, p-values below 0.05 doesn’t “prove” anything. Not, on the other way around, p-values higher than 0.05 refute anything.
- ASA statement has more than 4.700 citations with relevant impact.

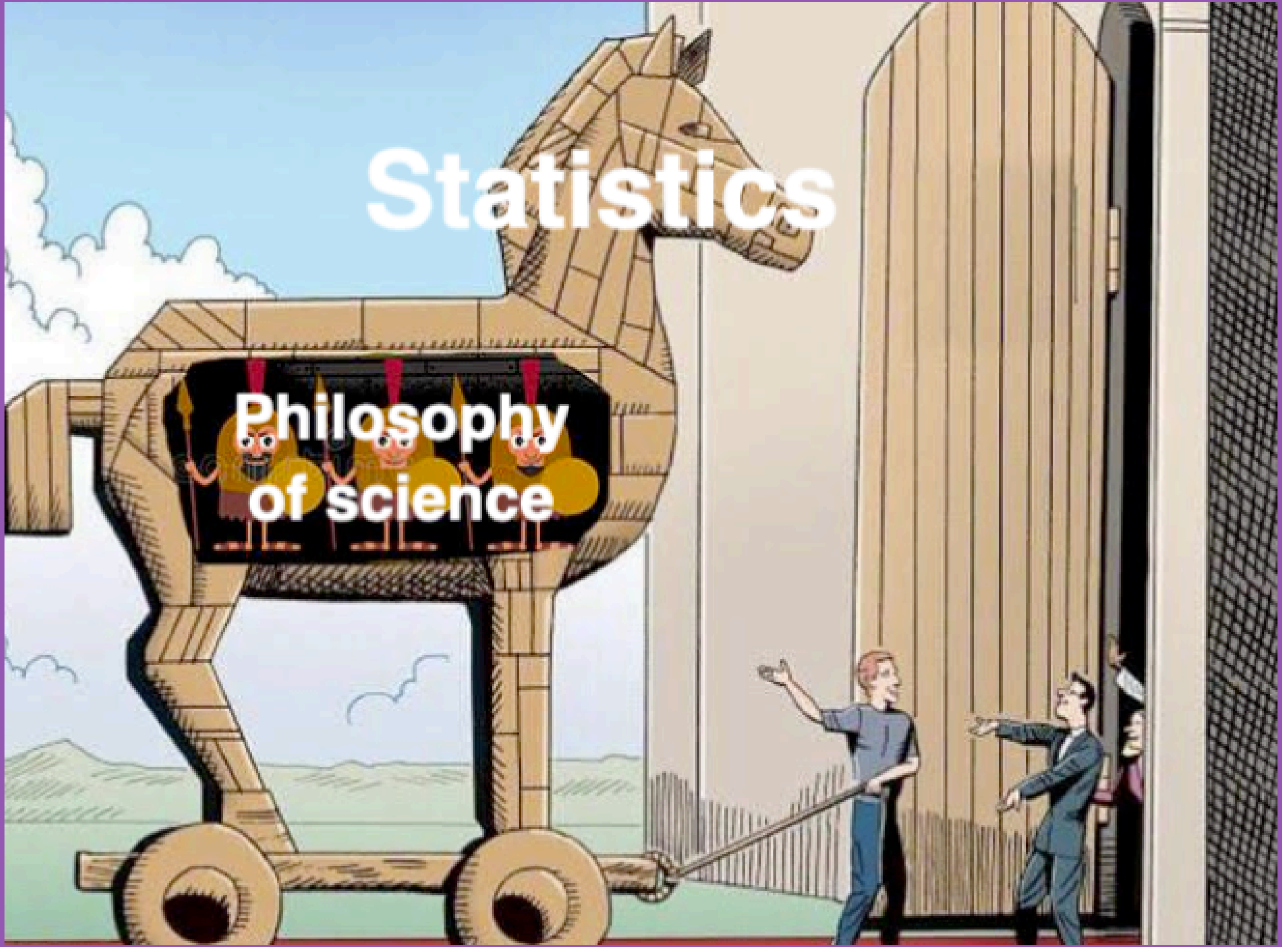
[xvi] pun intended …

The beginning of the end of Frequentist Statistics

- An international symposium was promoted in 2017 which originated an open-access special edition of *The American Statistician* dedicated to practical ways to abandon $p < 0.05$ (Wasserstein, Schirm and Lazar, 2019).

- Soon there were more attempts and claims. In September 2017, *Nature Human Behaviour* published an editorial proposing that the p-value's significance level be decreased from 0.05 to 0.005 (Benjamin *et al.*, 2018). Several authors, including highly important and influential statisticians argued that this simple step would help to tackle the replication crisis problem in science, that many believe be the main consequence of the abusive use of p-values (Ioannidis, 2019).

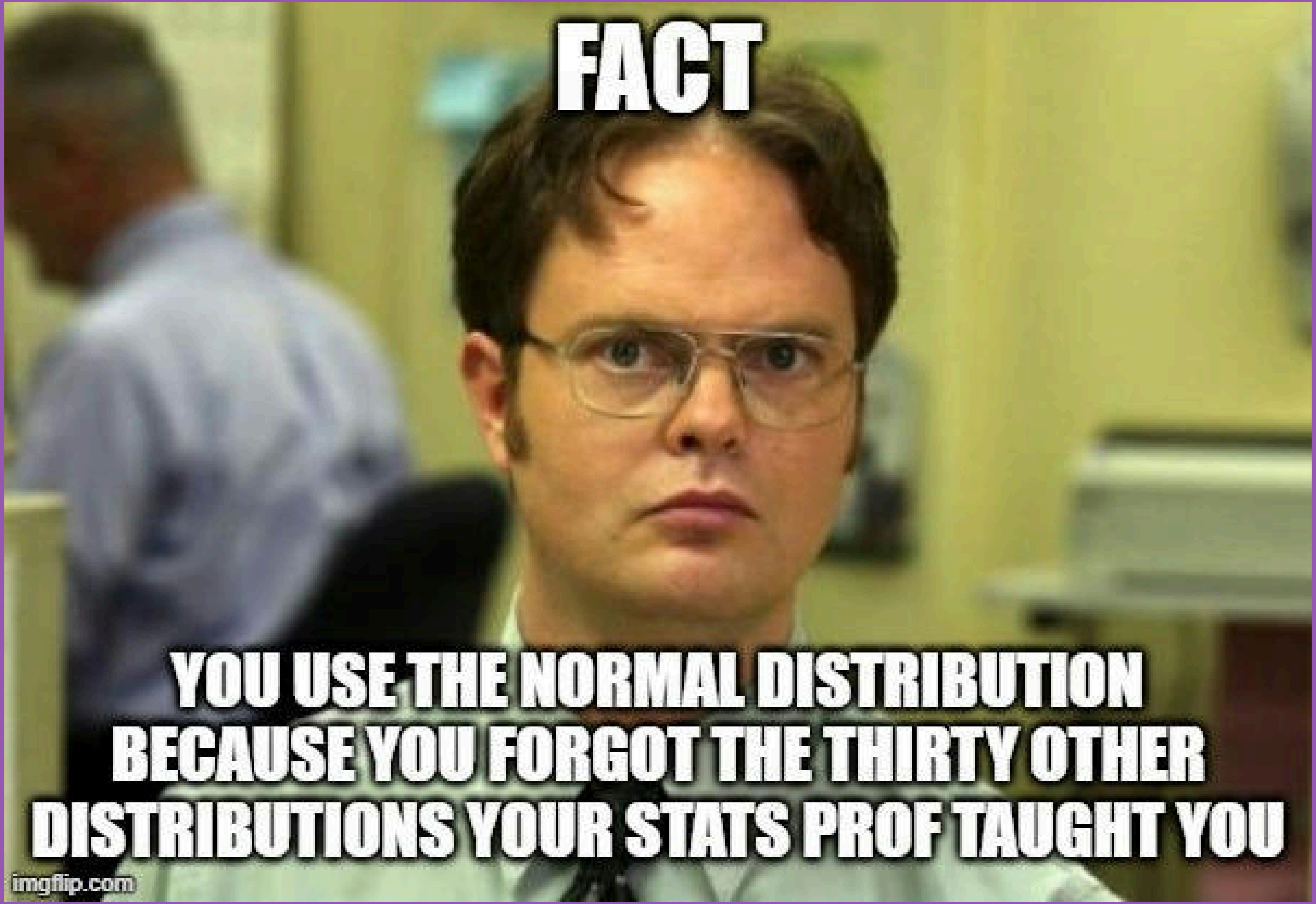
- Furthermore, many went a step ahead and suggested that science banish once for all p-values (Lakens *et al.*, 2018; "It's Time to Talk about Ditching Statistical Significance," 2019). Many suggest (including myself) that the main tool of statistical inference be Bayesian statistics (Goodman, 2016; Amrhein, Greenland and McShane, 2019; Schoot *et al.*, 2021).


Statistics
Philosophy of science

Statistical 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)


FACT
YOU USE THE NORMAL DISTRIBUTION BECAUSE YOU FORGOT THE THIRTY OTHER DISTRIBUTIONS YOUR STATS PROF TAUGHT YOU
imgflip.com

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.

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

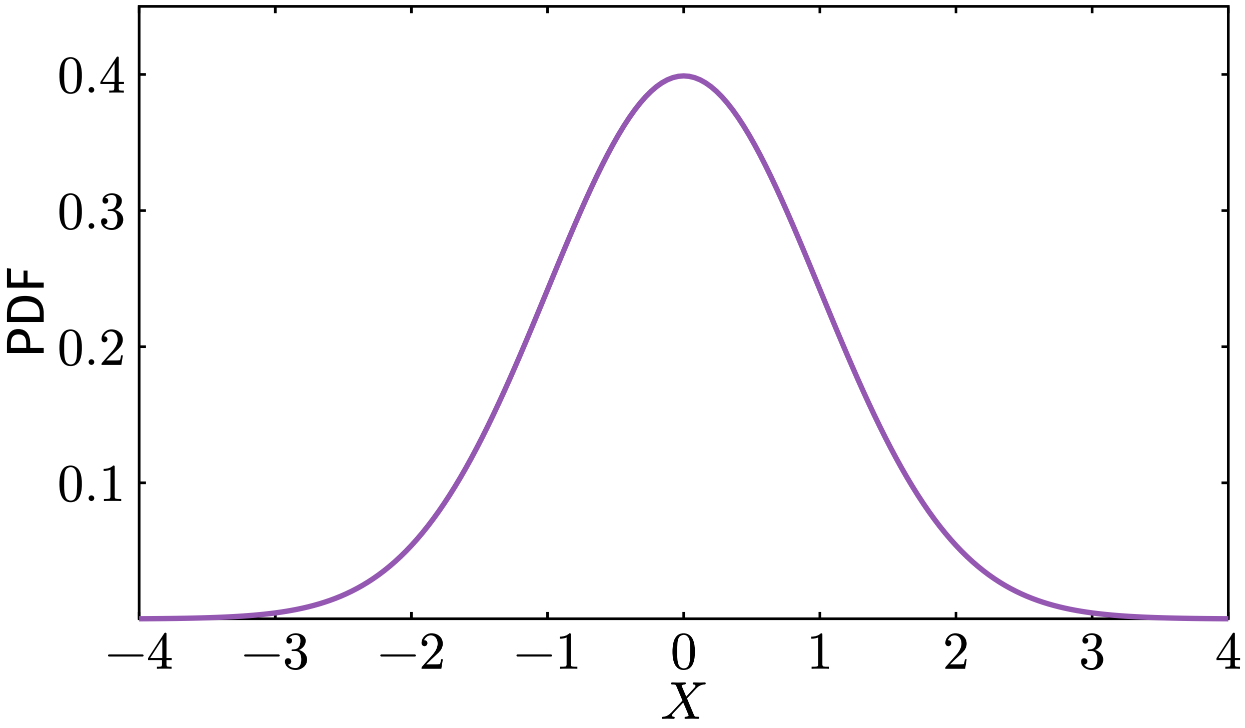
$$X \sim \mathrm{Dist}(\theta_1, \theta_2, ...)$$

where:

- X: random variable
- Dist: distribution name
- $\theta_1, \theta_2, ...$: 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

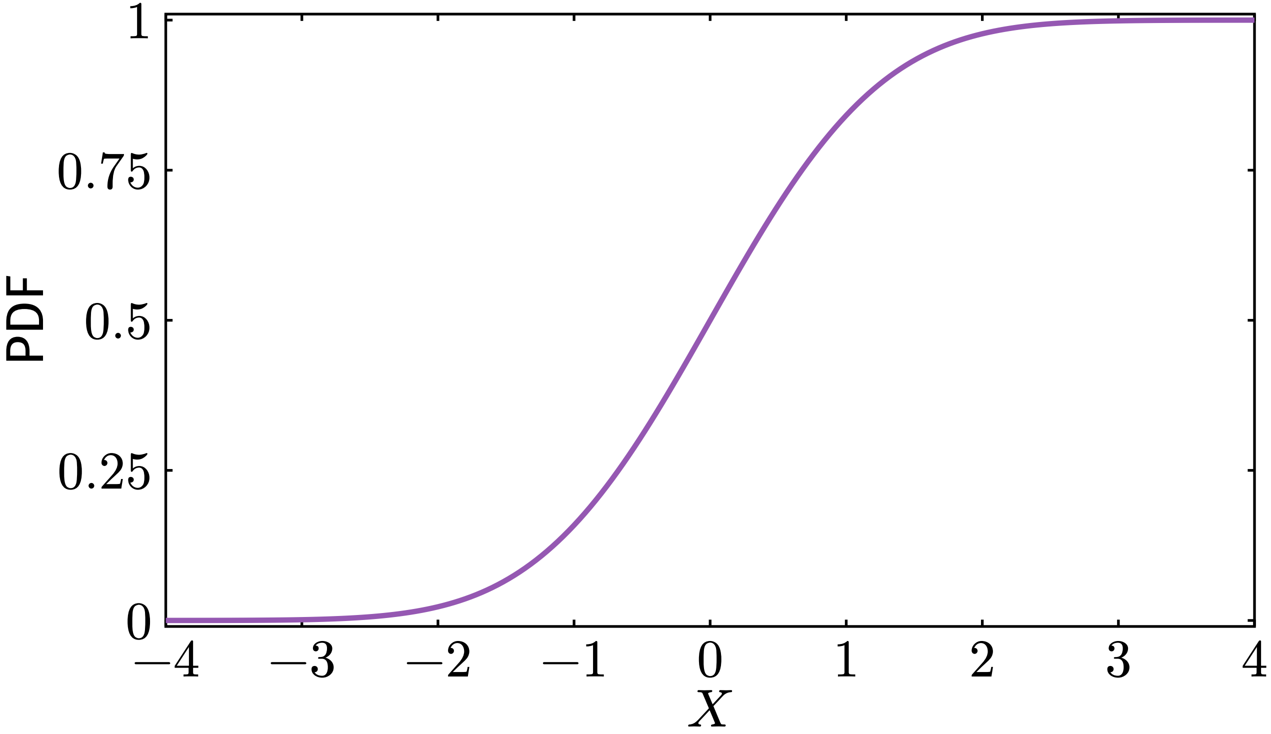


Cumulative Distribution Function

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:

$$\mathrm{CDF} = P(X \leq x)$$

Cumulative Distribution Function



Discrete Distributions

Discrete probability distributions are distributions which the results are a discrete number: $-N, ..., -2, 1, 0, 1, 2, ..., N$ and $N \in \mathbb{Z}$.

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:

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

Discrete Uniform

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

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

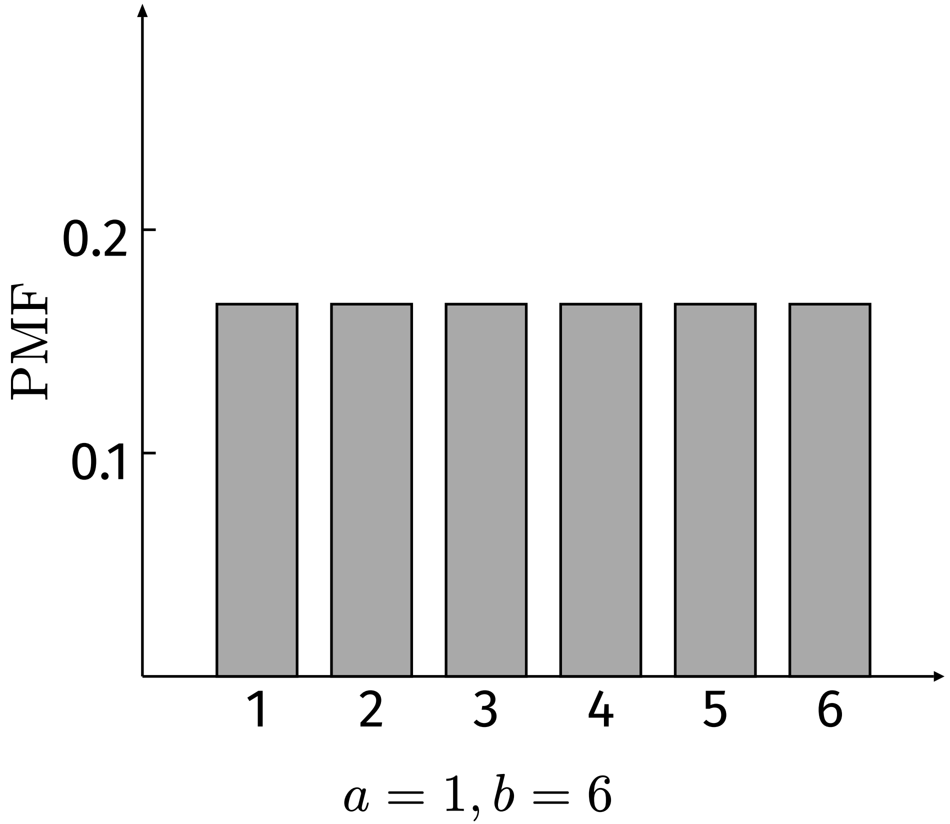
- a – lower bound
- b – upper bound

Example: dice.

Discrete Uniform

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

Discrete Uniform



Bernoulli

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

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

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

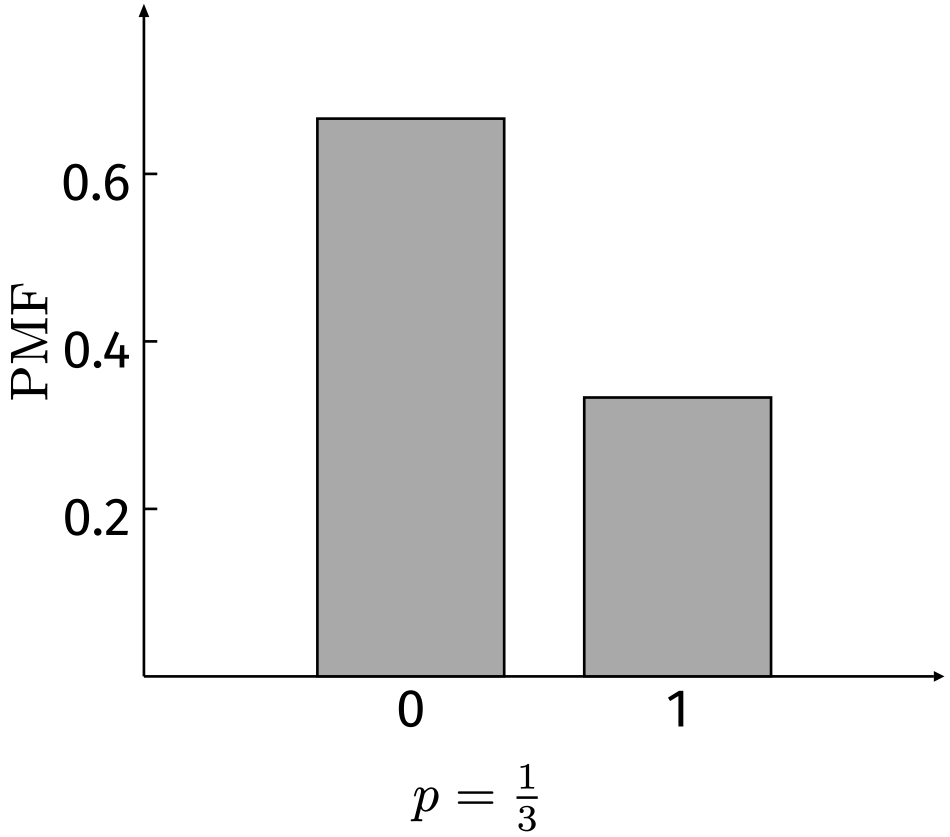
- p – probability of success

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

Bernoulli

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

Bernoulli



Binomial

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

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

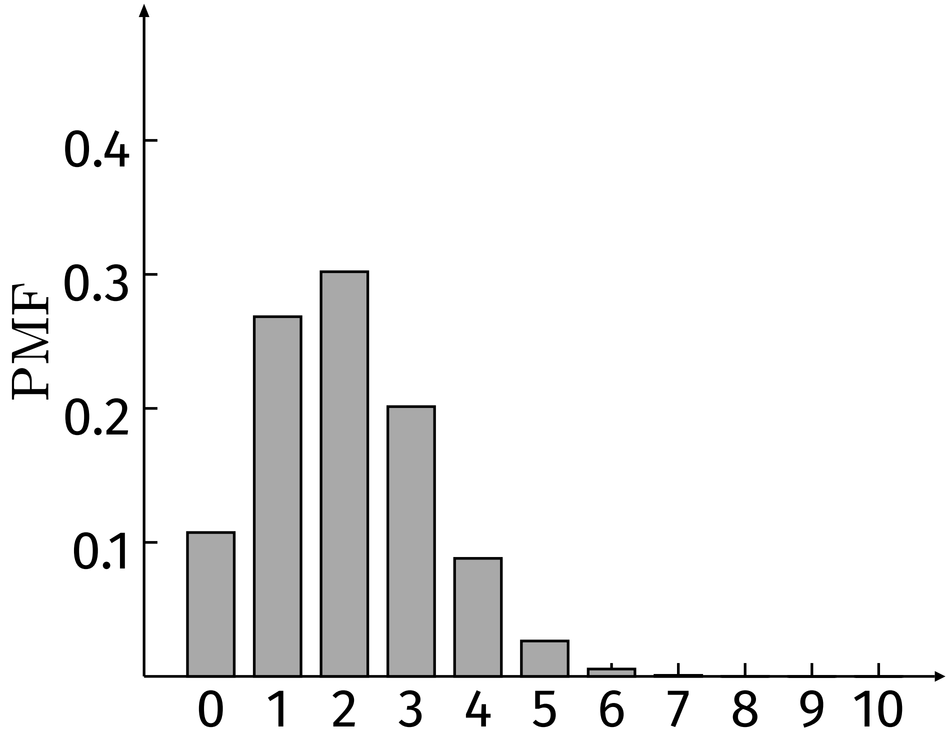
- n – number of experiments
- p – probability of success

Example: number of heads in five coin throws.

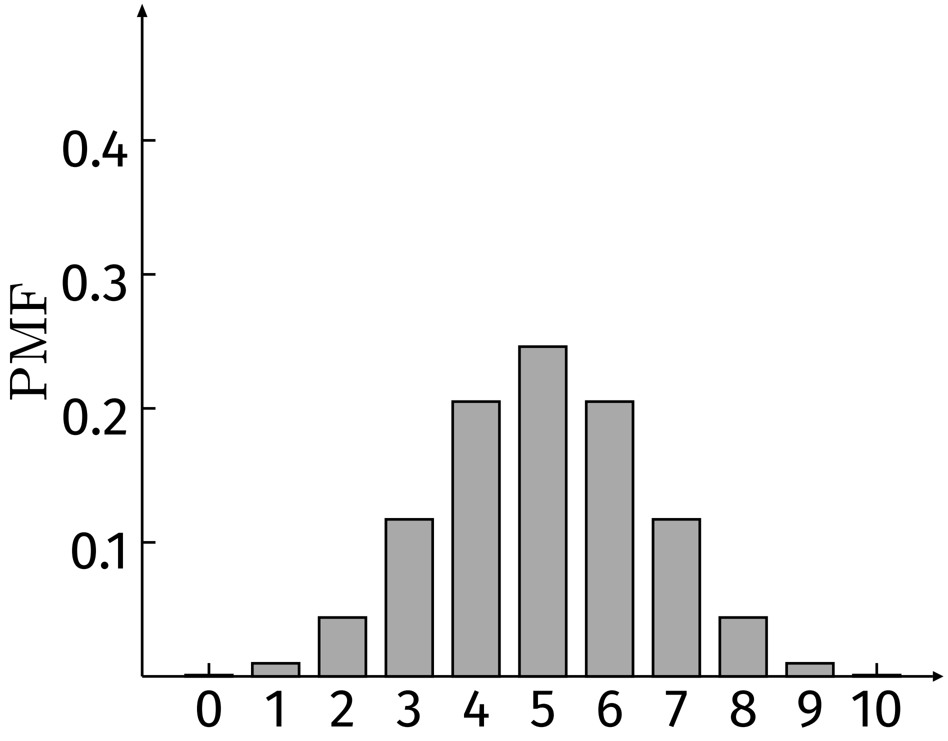
Binomial

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

Binomial



$n = 10, p = \frac{1}{5}$



$n = 10, p = \frac{1}{2}$

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 $\operatorname{Poisson}(\lambda)$:

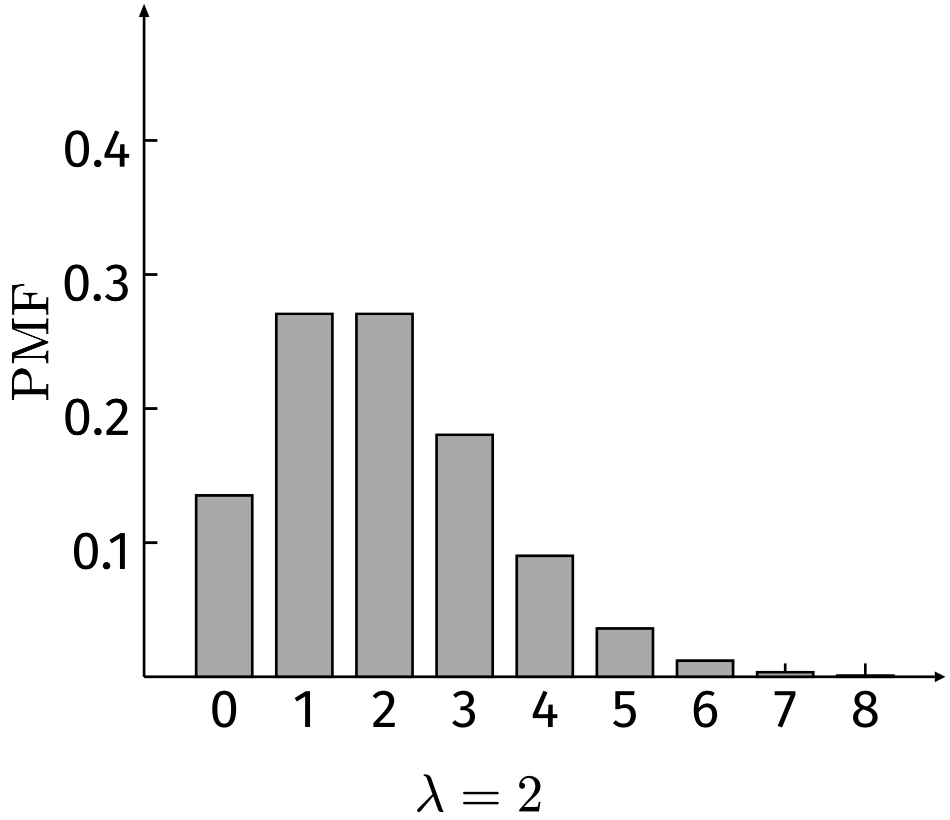
- λ – rate

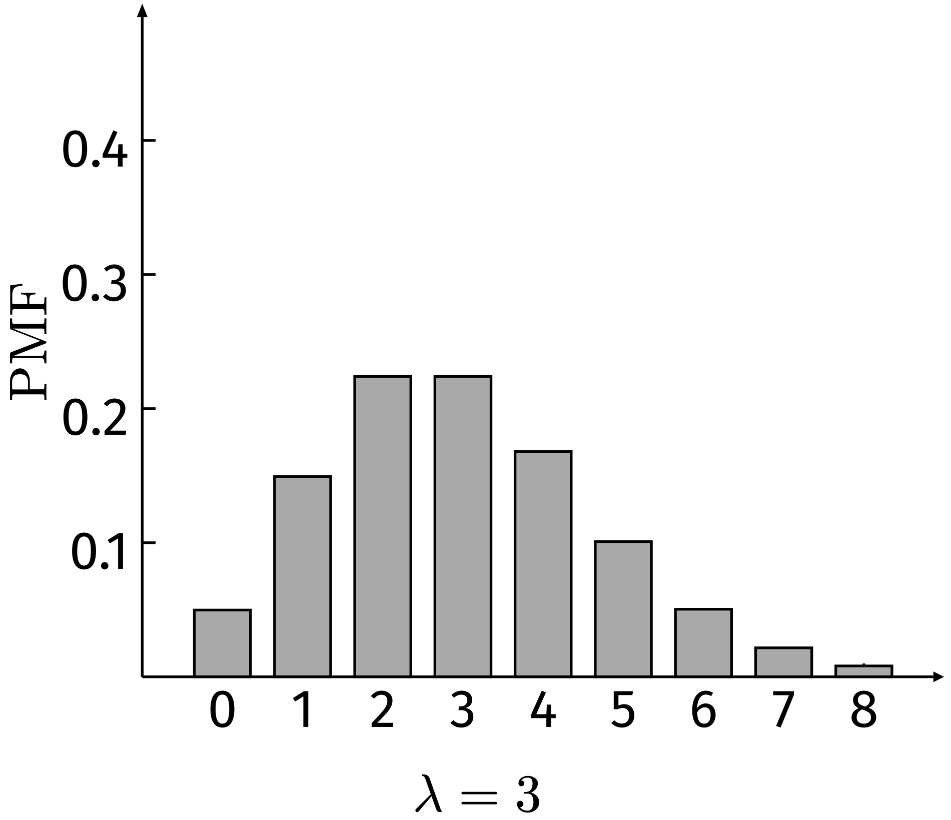
Example: number of e-mails that you receive daily or the number of the potholes you'll find in your commute.

Poisson

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

Poisson


PMF
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
$\lambda = 2$


PMF
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
$\lambda = 3$

Negative Binomial[xvii]

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 $\text{Negative Binomial}(k, p)$:

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

Example: annual occurrence of tropical cyclones.

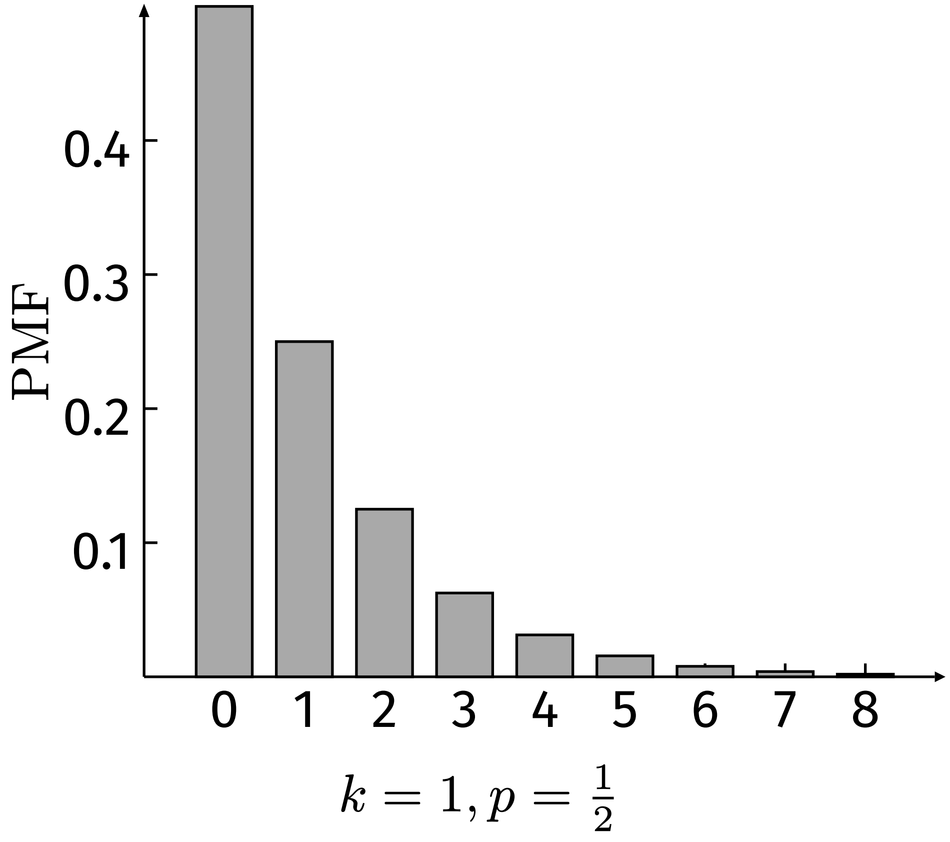
[xvii] any phenomena that can be modeles as a Poisson distribution can be modeled also as negative binomial distribution (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013), (Gelman, Hill and Vehtari, 2020).

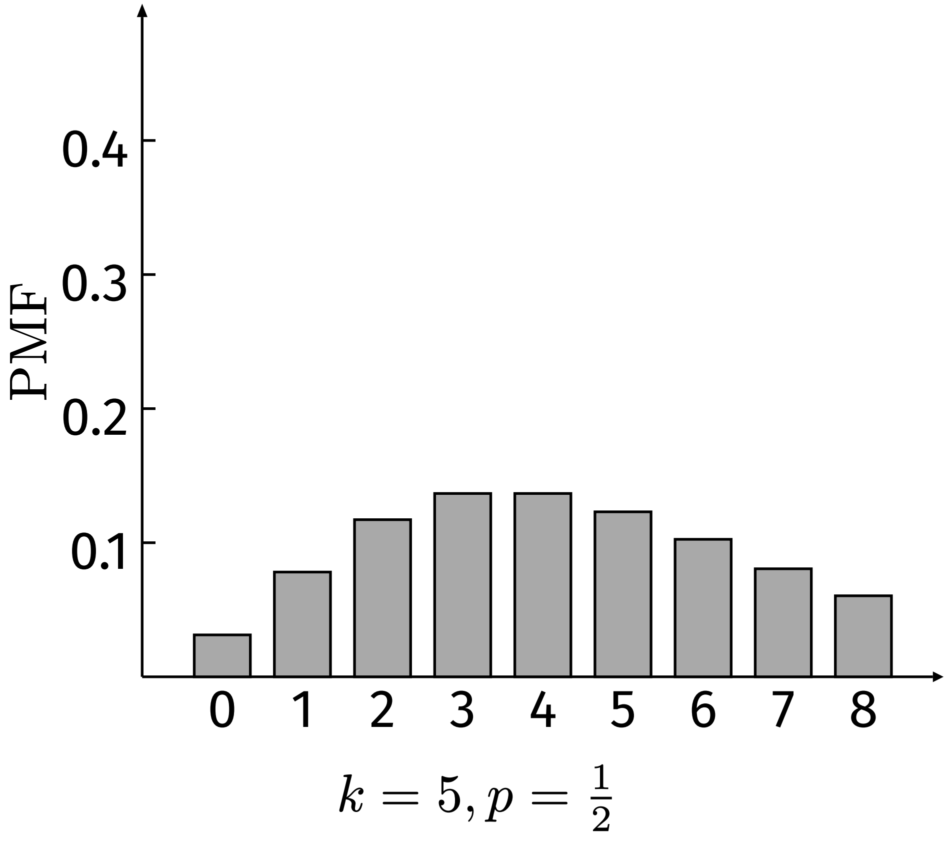
Negative Binomial

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

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

Negative Binomial





Continuous Distributions

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:

$$\operatorname{PDF}(x) = P(a \leq X \leq b) = \int_a^b f(x) \, \mathrm{d}x$$

Continuous Uniform

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

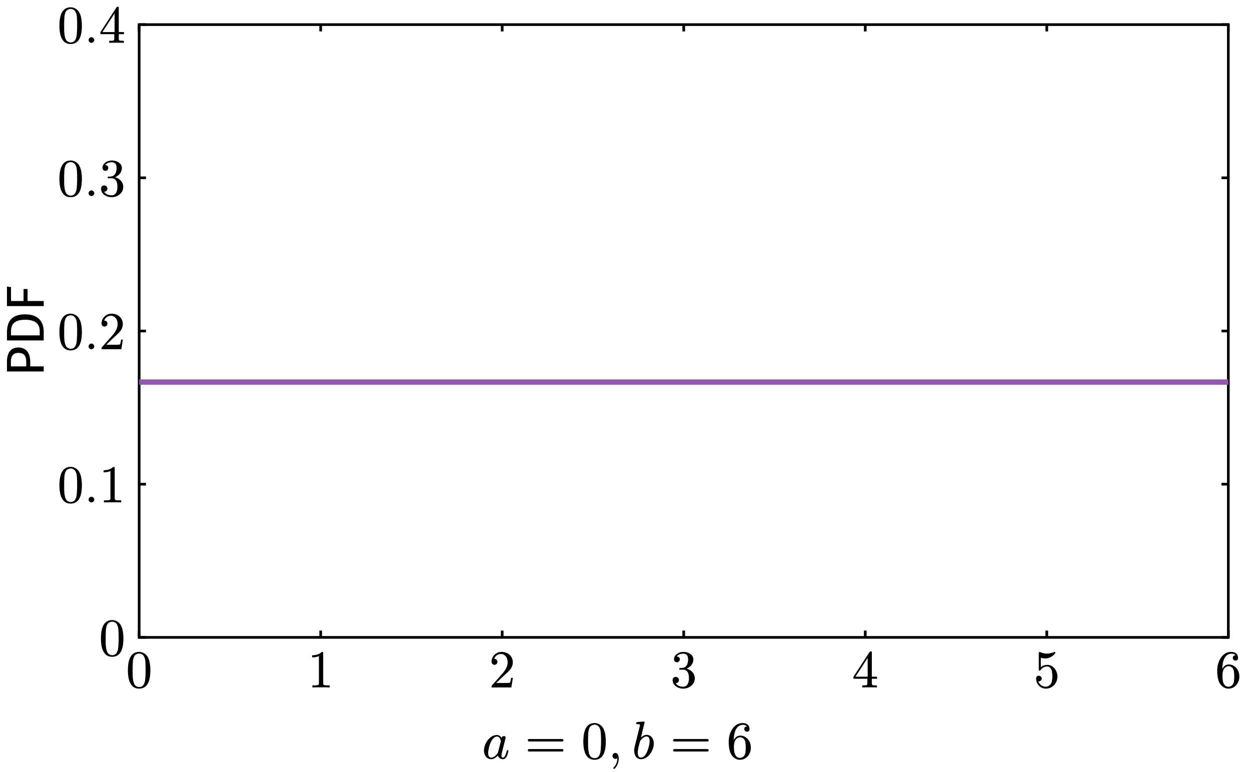
The continuous uniform distribution has two parameters and its notation is $\text{Uniform}(a, b)$:

- a – lower bound
- b – upper bound

Continuous Uniform

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

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 $\text{Normal}(\mu, \sigma)$ or $N(\mu, \sigma)$:

- μ – mean of the distribution, and also median and mode
- σ – standard deviation[xviii], a dispersion measure of how observations occur in relation from the mean

Example: height, weight etc.

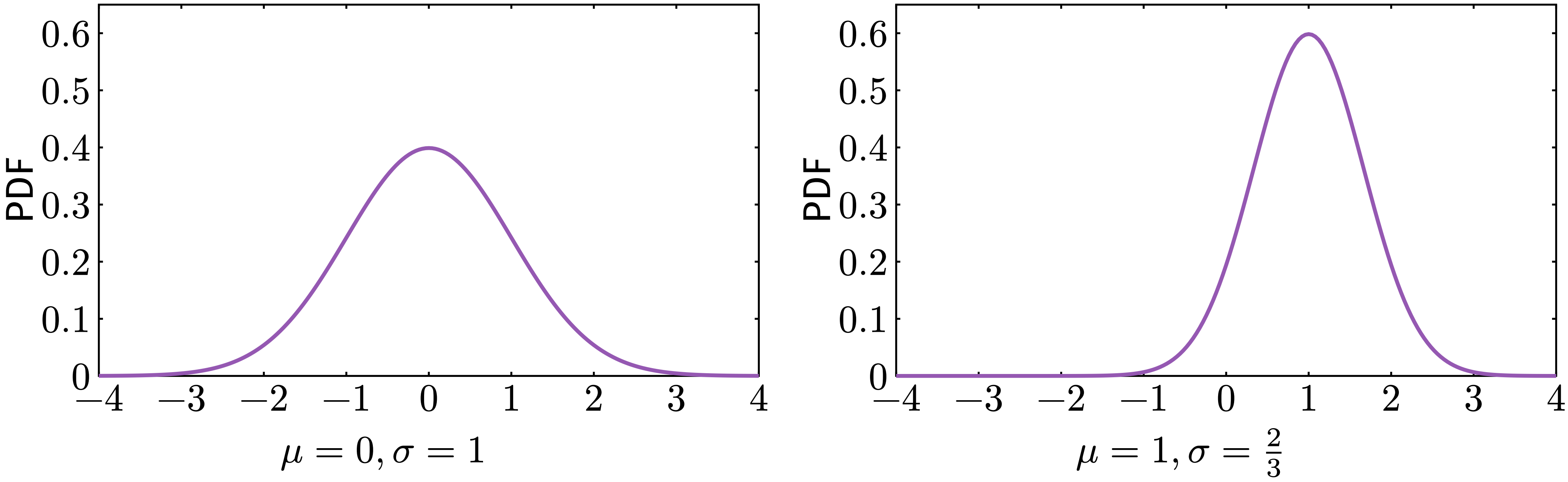
[xviii] sometimes is also parameterized as variance σ^2.

Normal[xix]

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

[xix] see how the normal distribution was derived from the binomial distribution in the backup slides.

Normal


PDF
0.6
0.5
0.4
0.3
0.2
0.1
0
−4
−3
−2
−1
0
1
2
3
4
$\mu = 0, \sigma = 1$
PDF
0.6
0.5
0.4
0.3
0.2
0.1
0
−4
−3
−2
−1
0
1
2
3
4
$\mu = 1, \sigma = \frac{2}{3}$

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 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

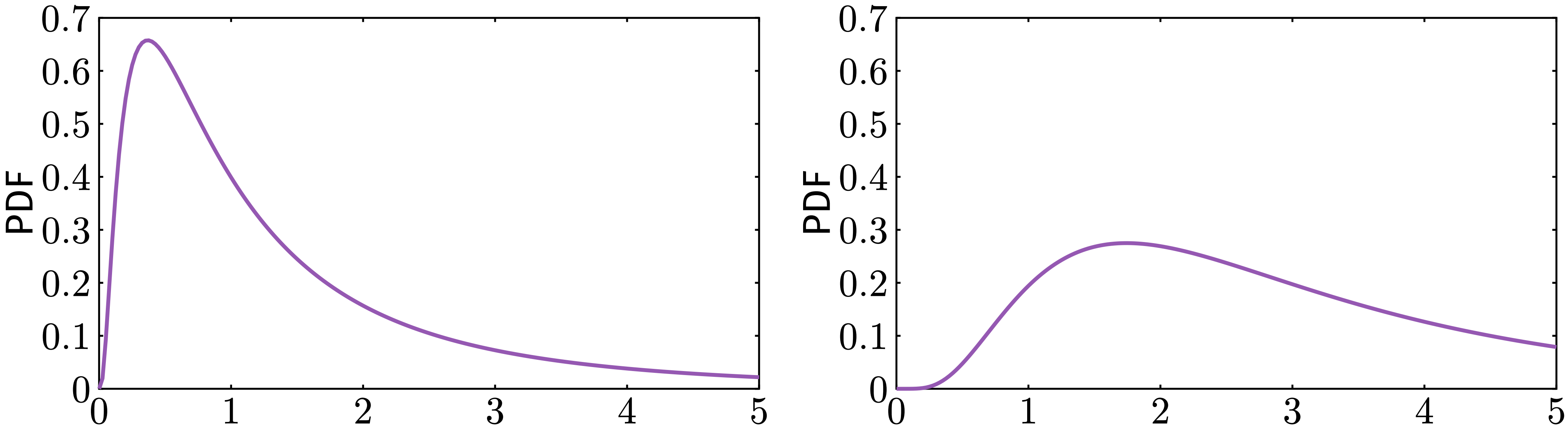
The log-normal distribution has two parameters and its notation is $\operatorname{Log-Normal}(\mu, \sigma^2)$:

- μ – mean of the distribution’s natural logarithm
- σ – square root of the variance of the distribution’s natural logarithm

Log-Normal

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

Log-Normal



$\mu = 0, \sigma = 1$

$\mu = 1, \sigma = \frac{2}{3}$

Exponential

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

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

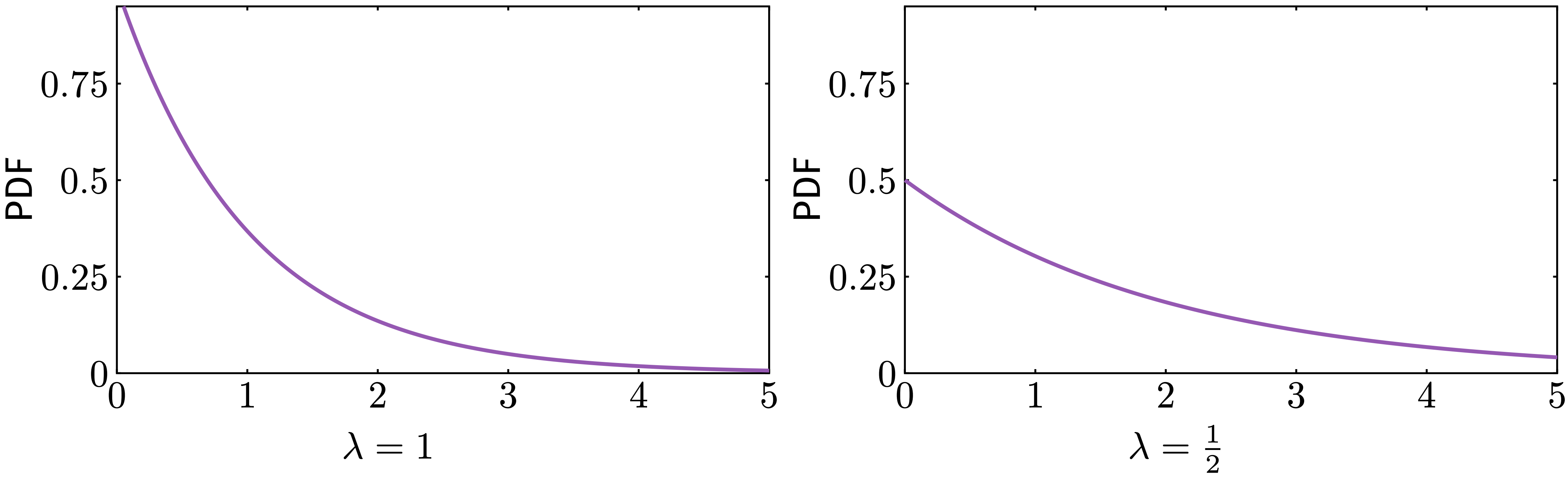
- λ – rate

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

Exponential

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

Exponential



Gamma

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

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

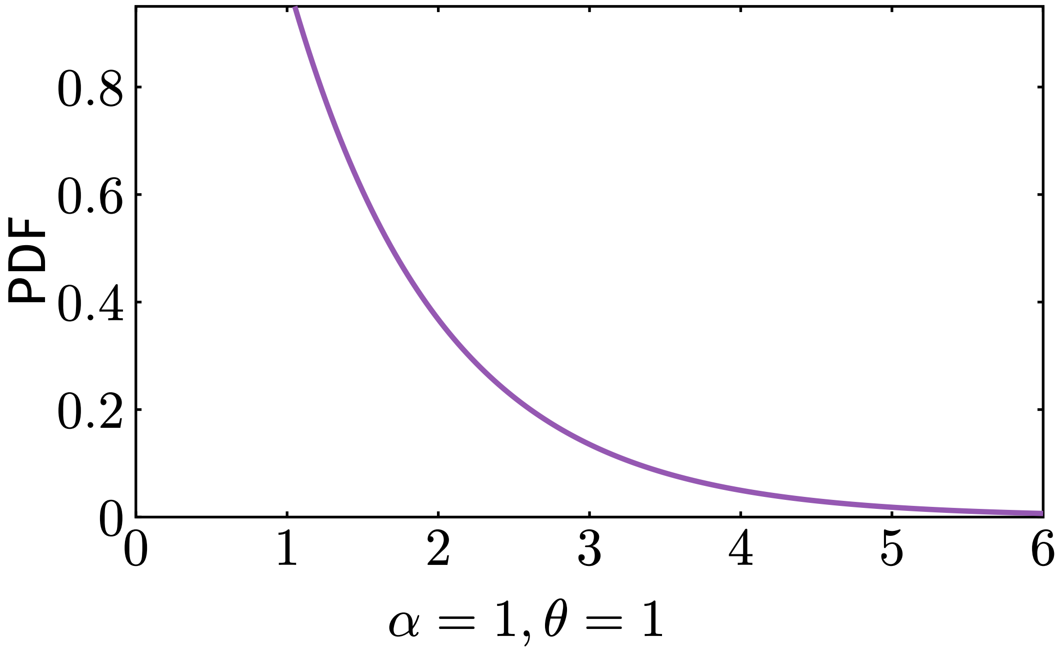
- α – shape parameter
- θ – rate parameter

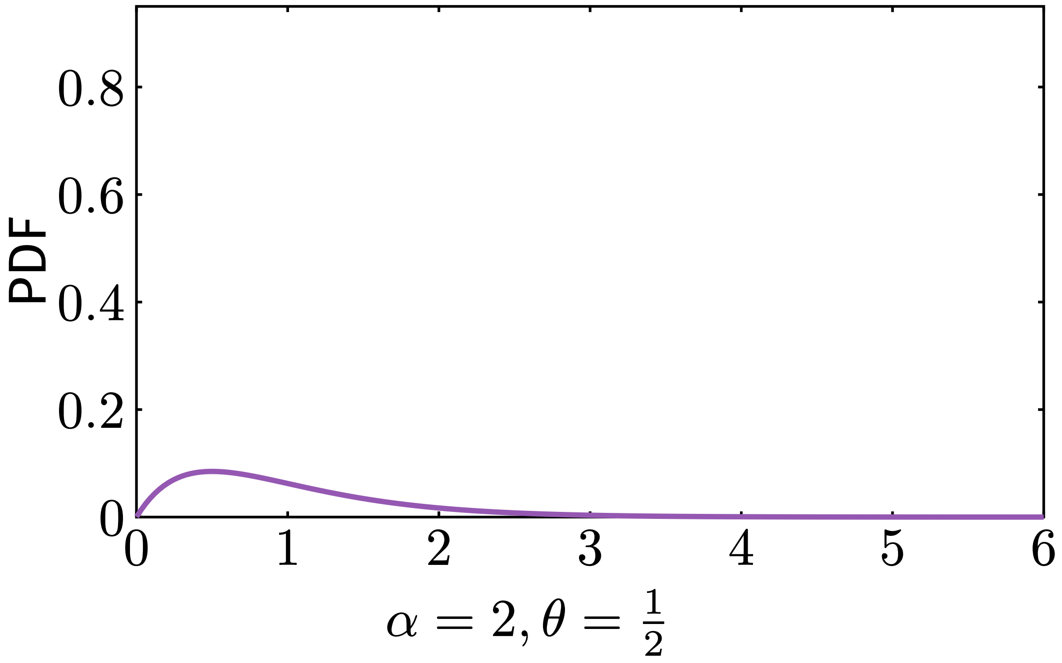
Example: Any waiting time can be modelled with a gamma distribution.

Gamma

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

Gamma


PDF
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
$\alpha = 1, \theta = 1$


PDF
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
$\alpha = 2, \theta = \frac{1}{2}$

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[xx].

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.

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

Student's t

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

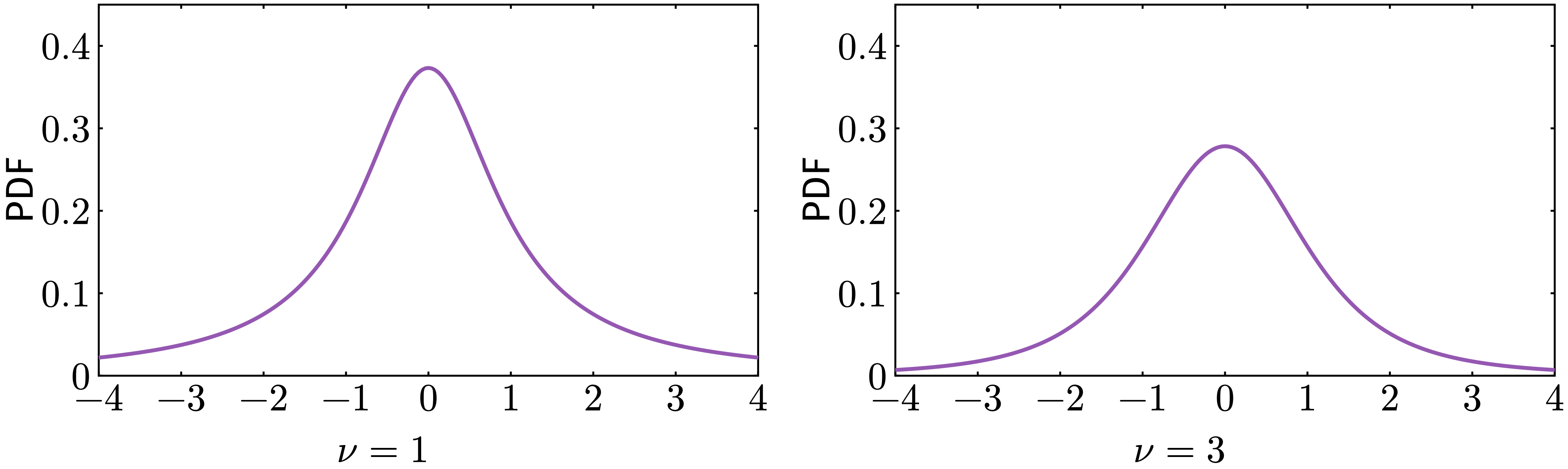
- ν – degrees of freedom, controls how much it resembles a normal distribution

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 $\text{Cauchy}(\mu, \sigma)$:

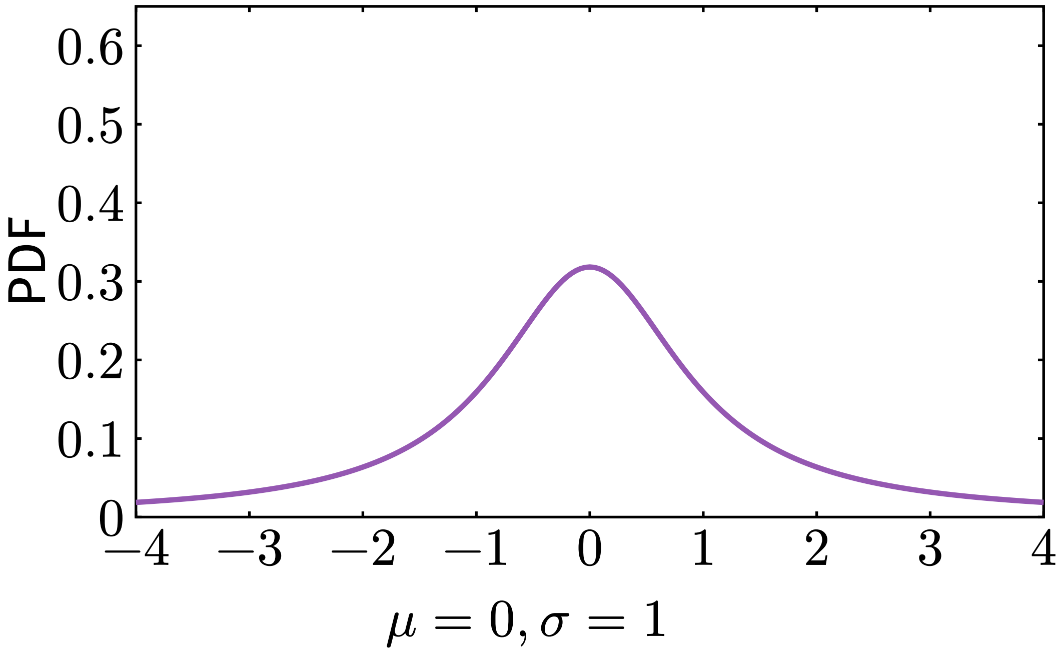
- μ – location parameter
- σ – scale parameter

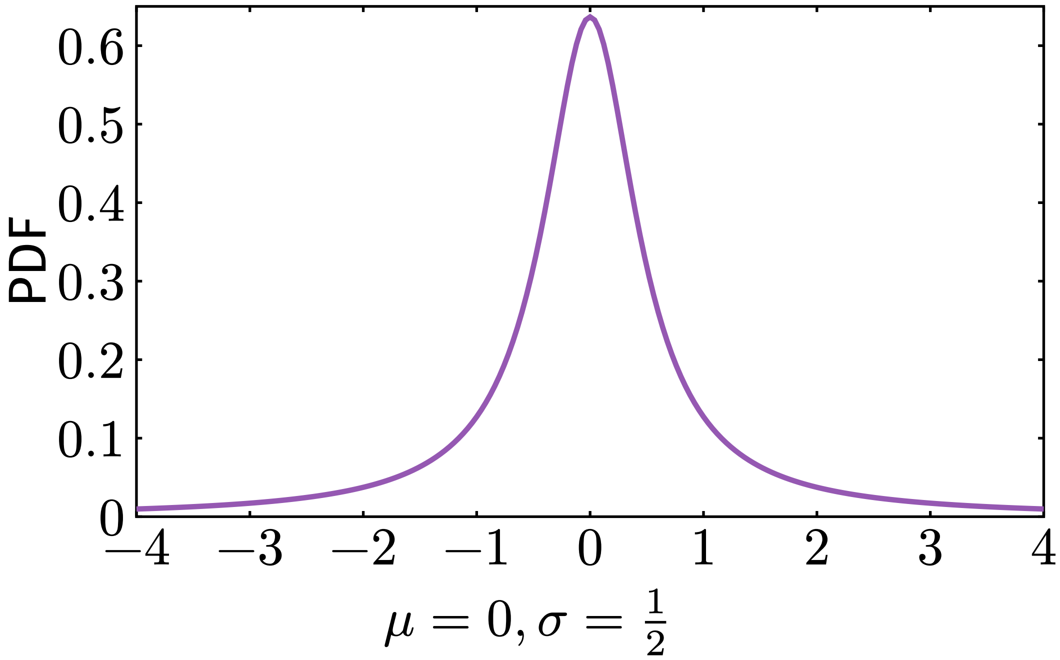
Example: a dataset full of outliers.

Cauchy

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

Cauchy


PDF
0.6
0.5
0.4
0.3
0.2
0.1
0
−4
−3
−2
−1
0
1
2
3
4
$\mu = 0, \sigma = 1$


PDF
0.6
0.5
0.4
0.3
0.2
0.1
0
−4
−3
−2
−1
0
1
2
3
4
$\mu = 0, \sigma = \frac{1}{2}$

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 $\operatorname{Beta}(\alpha, \beta)$:

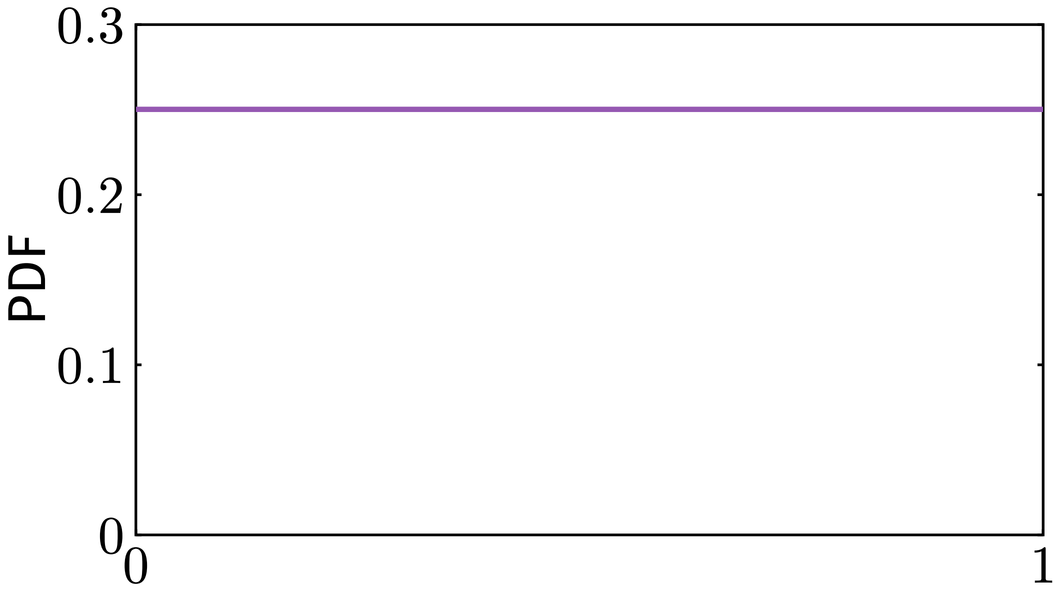
- α (or sometimes a) – shape parameter, controls how much the shape is shifted towards 1
- β (or sometimes b) – shape parameter, controls how much the shape is shifted towards 0

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

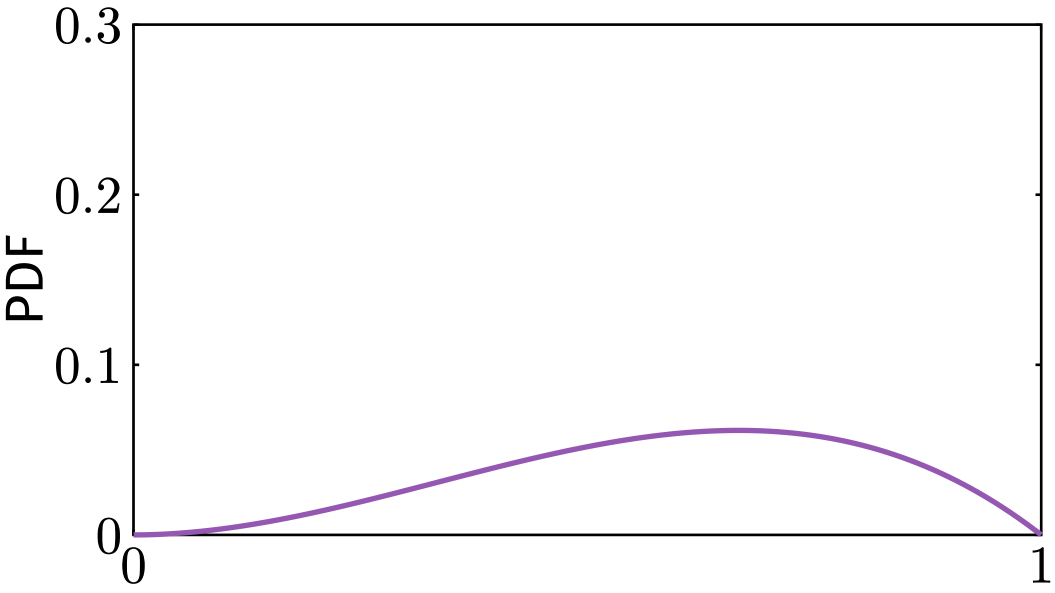
Beta

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

Beta



$\alpha = 1, \beta = 1$

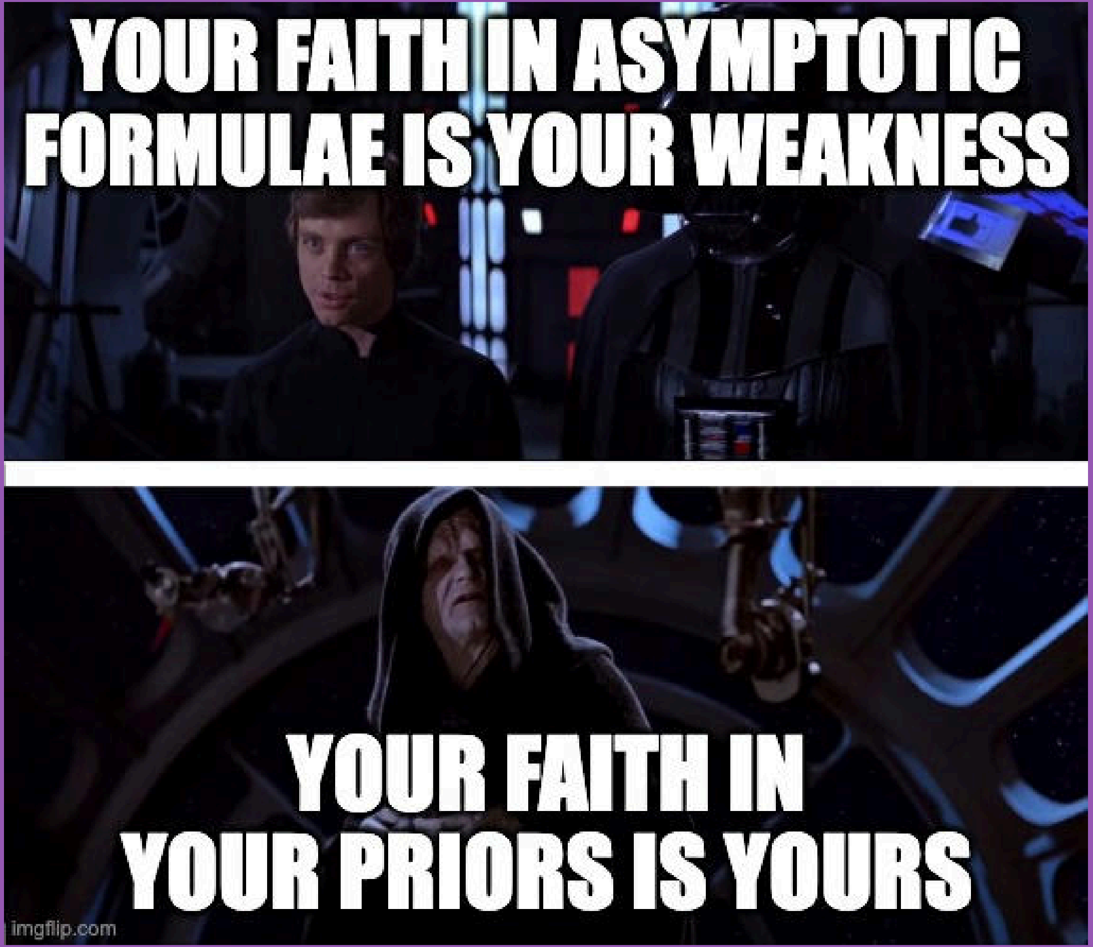


$\alpha = 3, \beta = 2$

Priors

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013):
  - 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
- Schoot *et al.* (2021)


YOUR FAITH IN ASYMPTOTIC FORMULAE IS YOUR WEAKNESS
YOUR FAITH IN YOUR PRIORS IS YOURS
imgflip.com

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; 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 (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013; McElreath, 2020; 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**: $\{\sigma \in \mathbb{R}^+ : 0 < \theta < \infty\}$

Weakly Uninformative 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[xxi]:

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

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

[xxii] see more about prior choices in the Stan's GitHub wiki.

An Example of a Robust Prior

A nice example comes from a Ben Goodrich's lecture[xxiii] (Columbia professor and member of Stan's research group).

He discuss about one of the biggest effect sizes observed in social sciences. In the exit pools for the 2008 USA presidential election (Obama vs McCain), there was, in general, around 40% of support for Obama. If you changed the respondent race from non-black to black, this was associated with an increase of 60% in the probability of the respondent to vote on Obama

In logodds scales, 2.5x increase (from 40% to almost 100%) would be equivalent, on a Bernoulli/logistic/binomial model, to a coefficient value of ≈ 0.92[xxiv]. This effect size would be easily derived from a $\text{Normal}(0, 1)$ prior.

[xxiii] https://youtu.be/p6cyRBWahRA, in case you want to see the full video, the section about priors related to the argument begins at minute 40

[xxiv] $\log(\text{odds ratio}) = \log(2.5) = 0.9163$.

Informative Prior

In some contexts, it is interesting to use an informative prior. Good candidates are when data is scarce or expensive and prior knowledge about the phenomena is available.

Some examples:

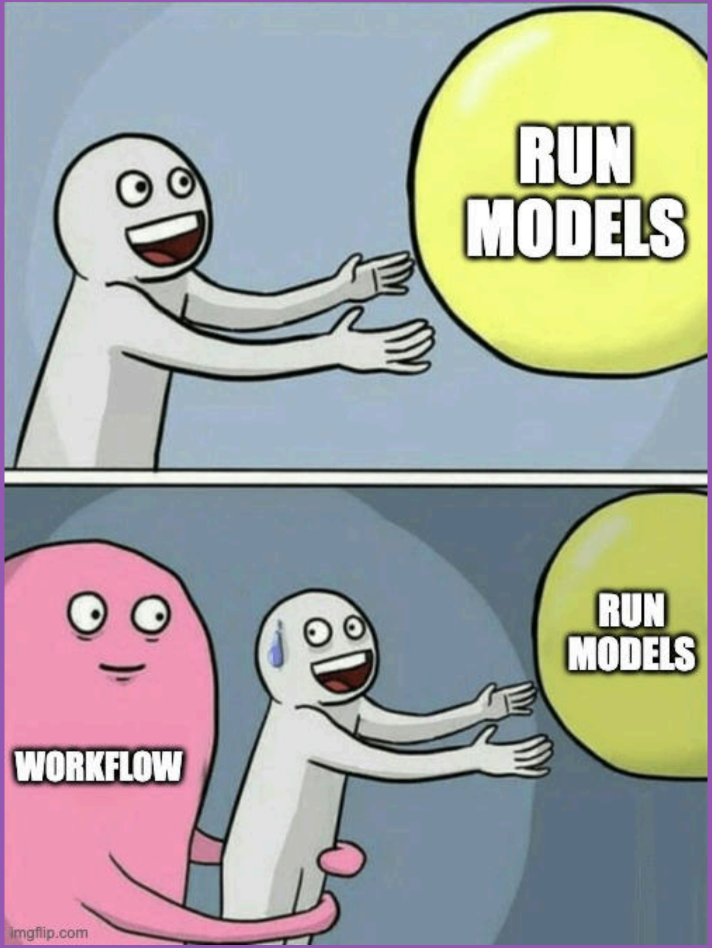
- $\text{Normal}(5, 20)$
- $\text{Log-Normal}(0, 5)$
- $\text{Beta}(100, 9803)$[xxv]

[xxv] this is used in COVID-19 models from the CoDatMo Stan research group.

Bayesian Workflow

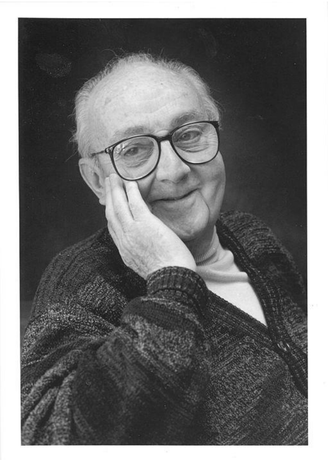
Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - Chapter 6: Model checking
- McElreath (2020) - Chapter 4: Geocentric Models
- Gelman, Hill and Vehtari (2020):
  - Chapter 6: Background on regression modeling
  - Chapter 11: Assumptions, diagnostics, and model evaluation
- Gelman *et al.* (2020) - "Workflow Paper"

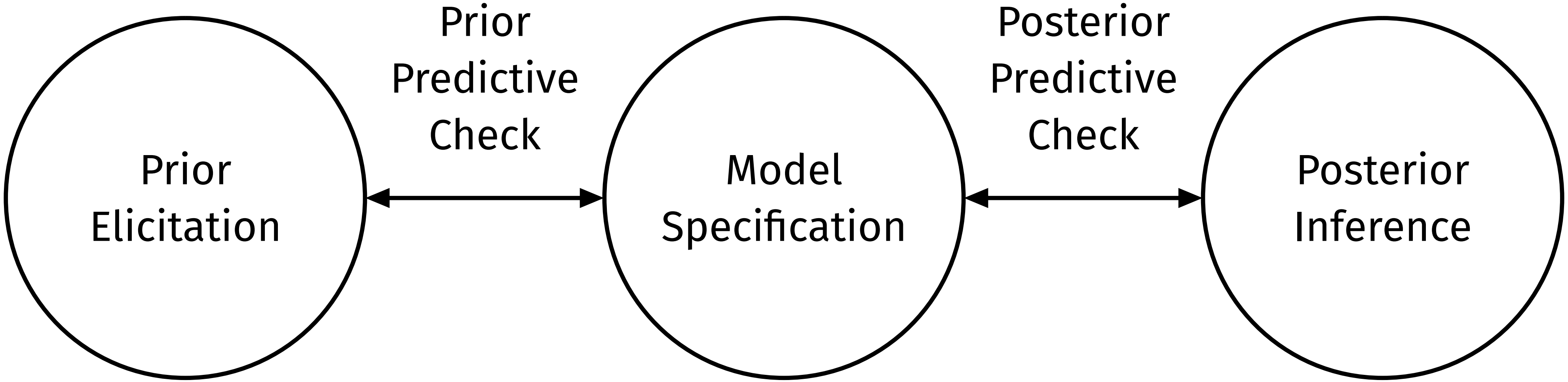

RUN MODELS
WORKFLOW
RUN MODELS
imgflip.com

All Models Are Wrong

> All models are wrong but some are useful
>
> — George Box (Box, 1976)



Bayesian Workflow[xxvi]



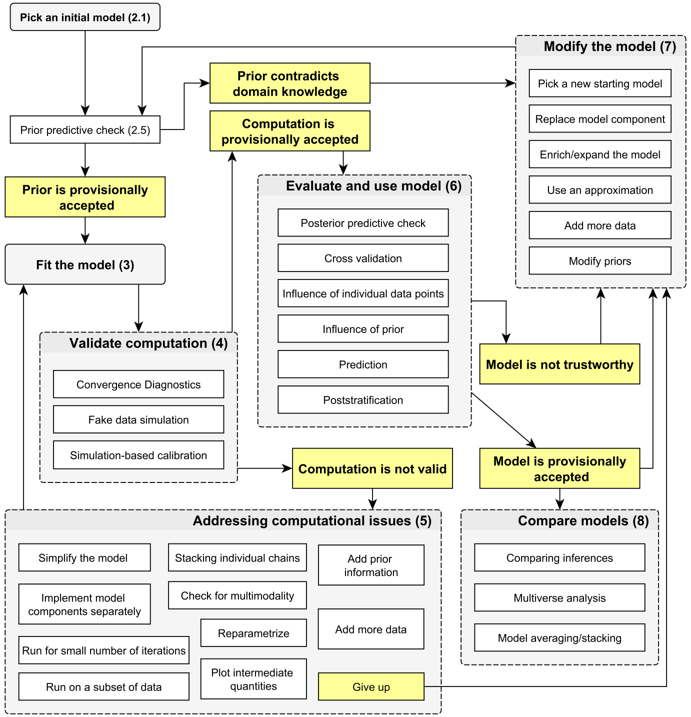
[xxvi] based on Gelman *et al.* (2020)

Bayesian Workflow[xxvii]

- Understand the domain and problem.
- Formulate the model mathematically.
- Implement model, test, and debug.
- Perform prior predictive checks.
- Fit the model.
- Assess convergence diagnostics.
- Perform posterior predictive checks.
- Improve the model iteratively: from baseline to complex and computationally efficient models.

[xxvii] adapted from Elizaveta Semenova.

Actual Bayesian Workflow



Pick an initial model (2.1)

Prior predictive check (2.5)

Prior contradicts domain knowledge

Computation is provisionally accepted

Prior is provisionally accepted

Fit the model (3)

Validate computation (4)
- Convergence Diagnostics
- Fake data simulation
- Simulation-based calibration

Computation is not valid

Addressing computational issues (5)
- Simplify the model
- Implement model components separately
- Run for small number of iterations
- Run on a subset of data
- Stacking individual chains
- Check for multimodality
- Reparametrize
- Plot intermediate quantities
- Add prior information
- Add more data
- Give up

Evaluate and use model (6)
- Posterior predictive check
- Cross validation
- Influence of individual data points
- Influence of prior
- Prediction
- Poststratification

Model is not trustworthy

Model is provisionally accepted

Modify the model (7)
- Pick a new starting model
- Replace model component
- Enrich/expand the model
- Use an approximation
- Add more data
- Modify priors

Compare models (8)
- Comparing inferences
- Multiverse analysis
- Model averaging/stacking

Figure 2: Bayesian workflow by Gelman *et al.* (2020).

Not a “new idea”

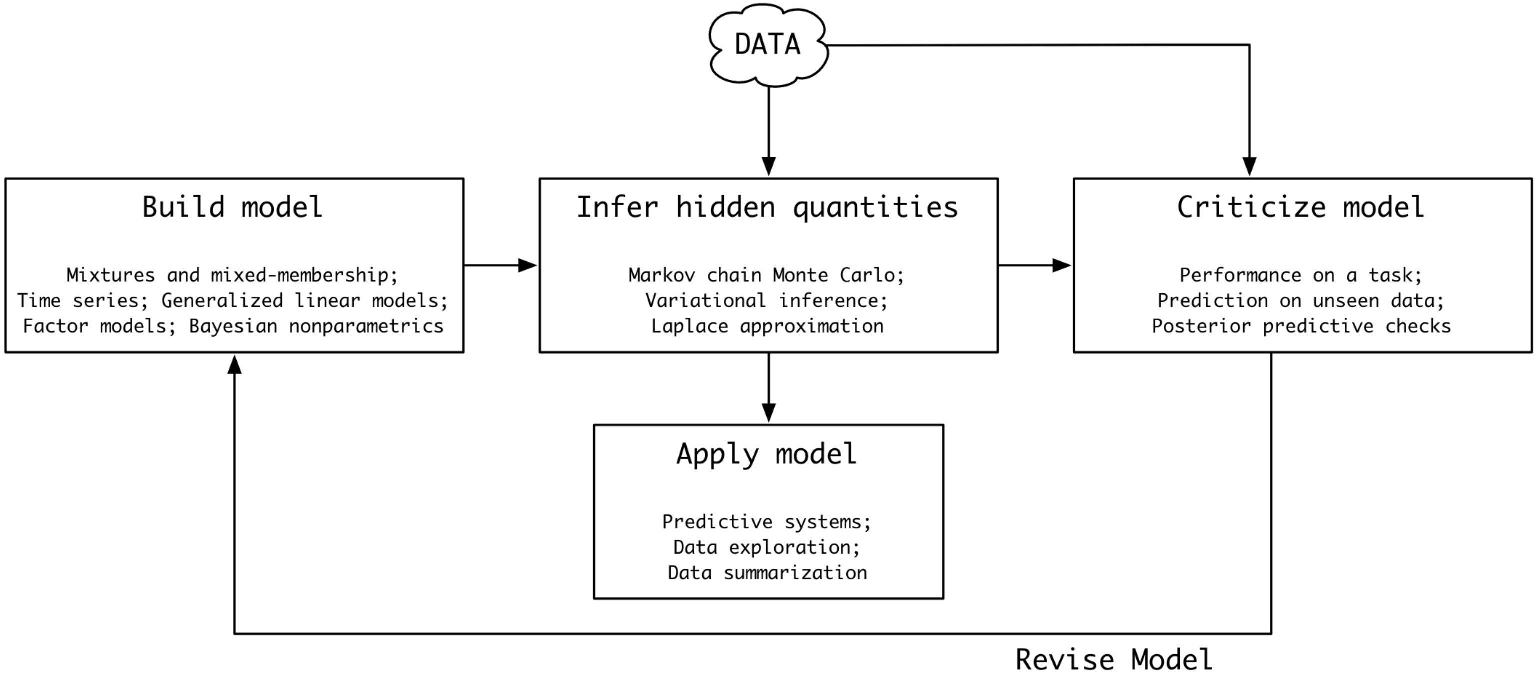


Figure 3: Box’s Loop from Box (1976) but taken from Blei (2014).

Prior Predictive Check

Before we feed data into our model, we need to check all of our priors.

In a very simple way, it consists in simulate parameter values based on prior distribution without conditioning on any data or employing any likelihood function.

Independent of the level of information specified in the priors, it is always important to perform a prior sensitivity analysis in order to have a deep understanding of the prior influence onto the posterior.

Posterior Predictive Check

We need to make sure that the posterior distribution of y, namely $\tilde{y}$, can capture all the nuances of the real distribution density/mass of y.

This procedure is called **posterior predictive check**, and it is generally carried on by a visual inspection[xxviii] of the real density/mass of y against generated samples of y by the Bayesian model.

The purpose is to compare the histogram of the dependent variable y against the histograms of simulated dependent variables y_{rep} by the model after parameter inference.

The idea is that the real and simulated histograms blend together and we do not observer any divergences.

[xxviii] we also perform mathematical/exact inspections, see the section on *Model Comparison*.

Examples of Posterior Predictive Checks

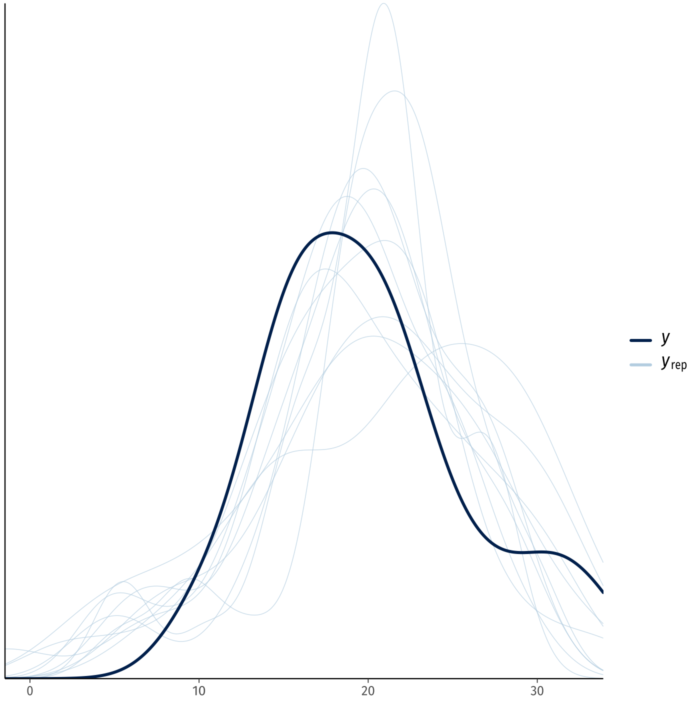


Figure 4: Real versus Simulated Densities

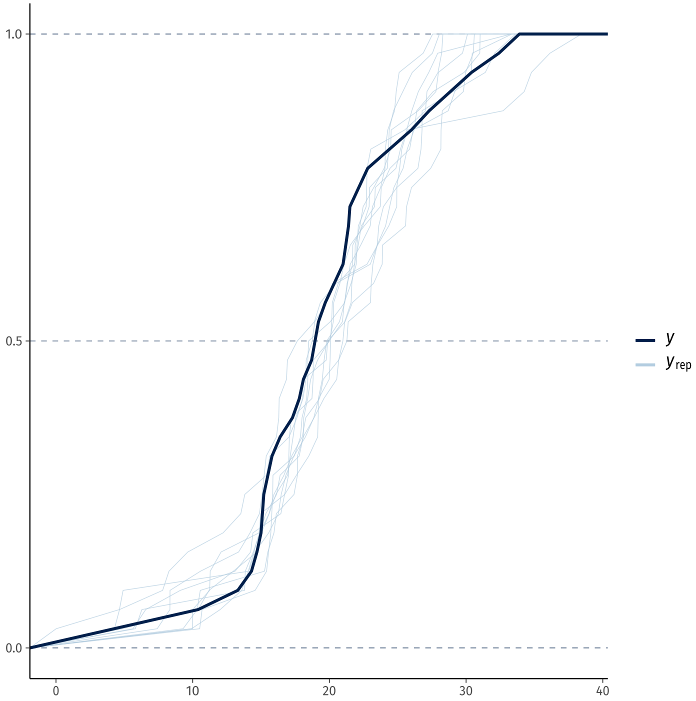


Figure 5: Real versus Simulated Empirical CDFs

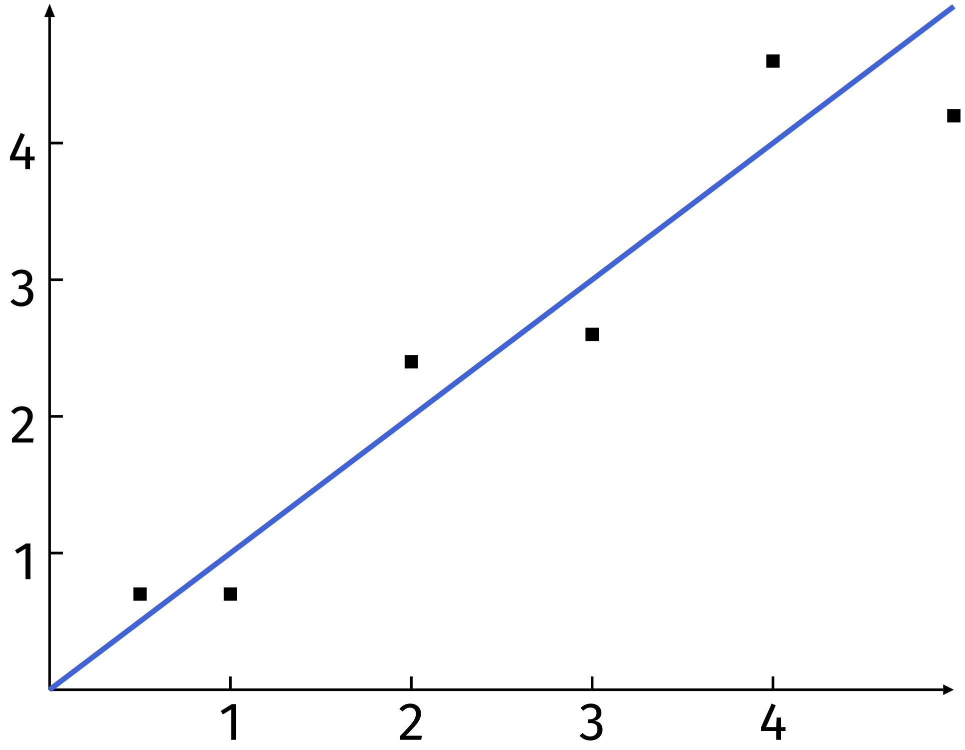
Linear Regression

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013):
  - Chapter 14: Introduction to regression models
  - Chapter 16: Generalized linear models
- McElreath (2020) - Chapter 4: Geocentric Models
- Gelman, Hill and Vehtari (2020):
  - Chapter 7: Linear regression with a single predictor
  - Chapter 8: Fitting regression models
  - Chapter 10: Linear regression with multiple predictors

LINEAR REGRESSION
EVERYWHERE
imgflip.com

What is Linear Regression?



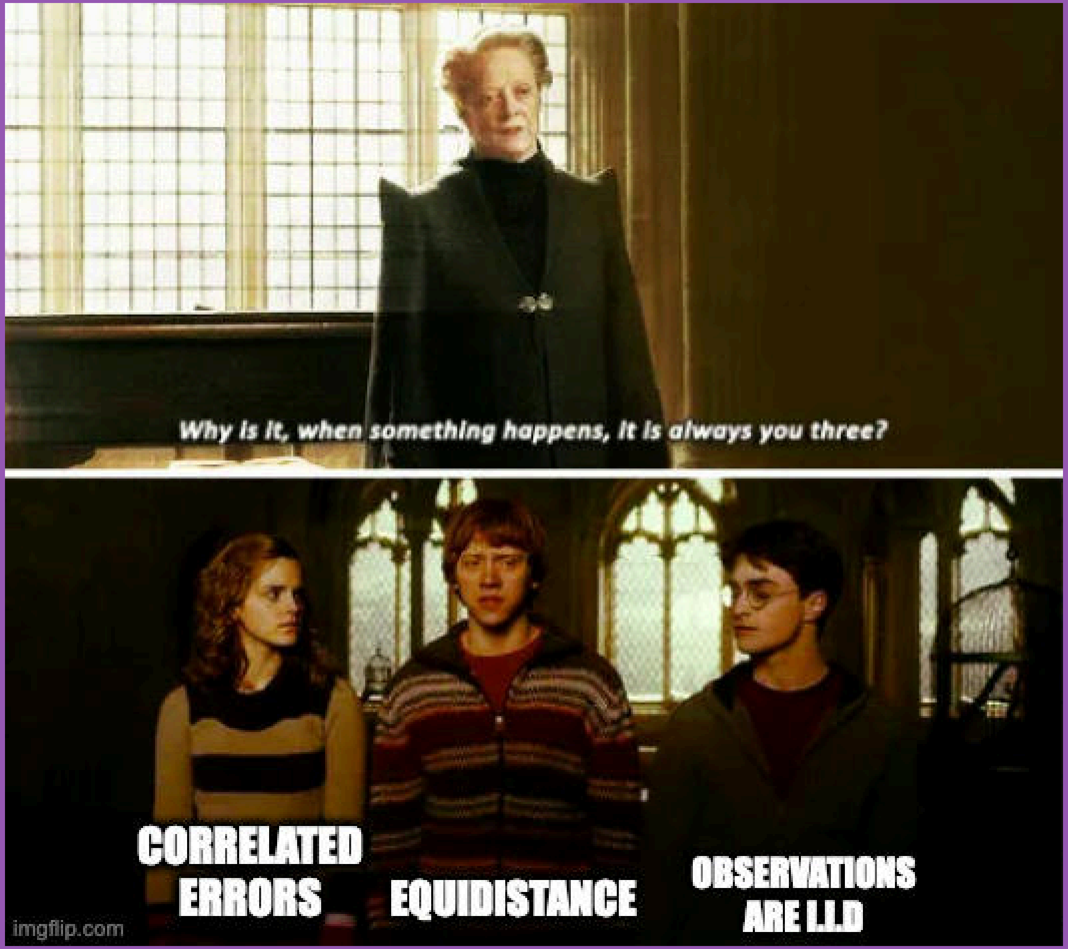
What is Linear Regression?

The ideia here is to model a dependent variable as a linear combination of independent variables.

$$\boldsymbol{y} = \alpha + \boldsymbol{X} \boldsymbol{\beta} + \varepsilon$$

where:

- $\boldsymbol{y}$ – dependent variable
- α – intercept (also called as constant)
- $\boldsymbol{\beta}$ – coefficient vector
- $\boldsymbol{X}$ – data matrix
- ε – model error


Why is it, when something happens, it is always you three?
CORRELATED ERRORS
EQUIDISTANCE
OBSERVATIONS ARE I.I.D
imgflip.com

Linear Regression Assumptions

- model error ε is independent of $\boldsymbol{X}$ and $\boldsymbol{y}$.
- Dependent variable $\boldsymbol{y}$ is continuous, unbounded, and, more importantly, “metric”-scaled, i.e. **equidistant**.
  - e.g. the increase from 1 to 2 is the same from 3 to 4. Generally violated when $\boldsymbol{y}$ is interval-scaled.
- Observations are I.I.D[xxix].

[xxix] independent and identically distributed.

Linear Regression Specification

To estimate the intercept α and coefficients β we use a Gaussian/normal likelihood function. Mathematically speaking, Bayesian linear regression is:

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Normal}(\alpha + \boldsymbol{X}\beta, \sigma) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\beta &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

Linear Regression Specification

What we are missing is the prior probabilities for the model's parameters:

- Prior Distribution for α – Knowledge that we have about the model's intercept.
- Prior Distribution for β – Knowledge that we have about the model's independent variable coefficients.
- Prior Distribution for σ – Knowledge that we have about the model's error.

Good Candidates for Prior Distributions

First, center ($\mu = 0$) and standardize ($\sigma = 1$) the independent variables.

- α – either a normal or student-t ($\nu = 3$), with mean as $\mu_{\boldsymbol{y}}$ and standard deviation as $2.5 \cdot \sigma_{\boldsymbol{y}}$ (also you can use the median and median absolute deviation).
- $\boldsymbol{\beta}$ – either a normal or student-t ($\nu = 3$), with mean 0 and standard deviation 2.5.
- σ – anything that is long-tailed (mass towards lower values) and restrained to positive values only. Exponential is a good candidate.

Posterior Computation

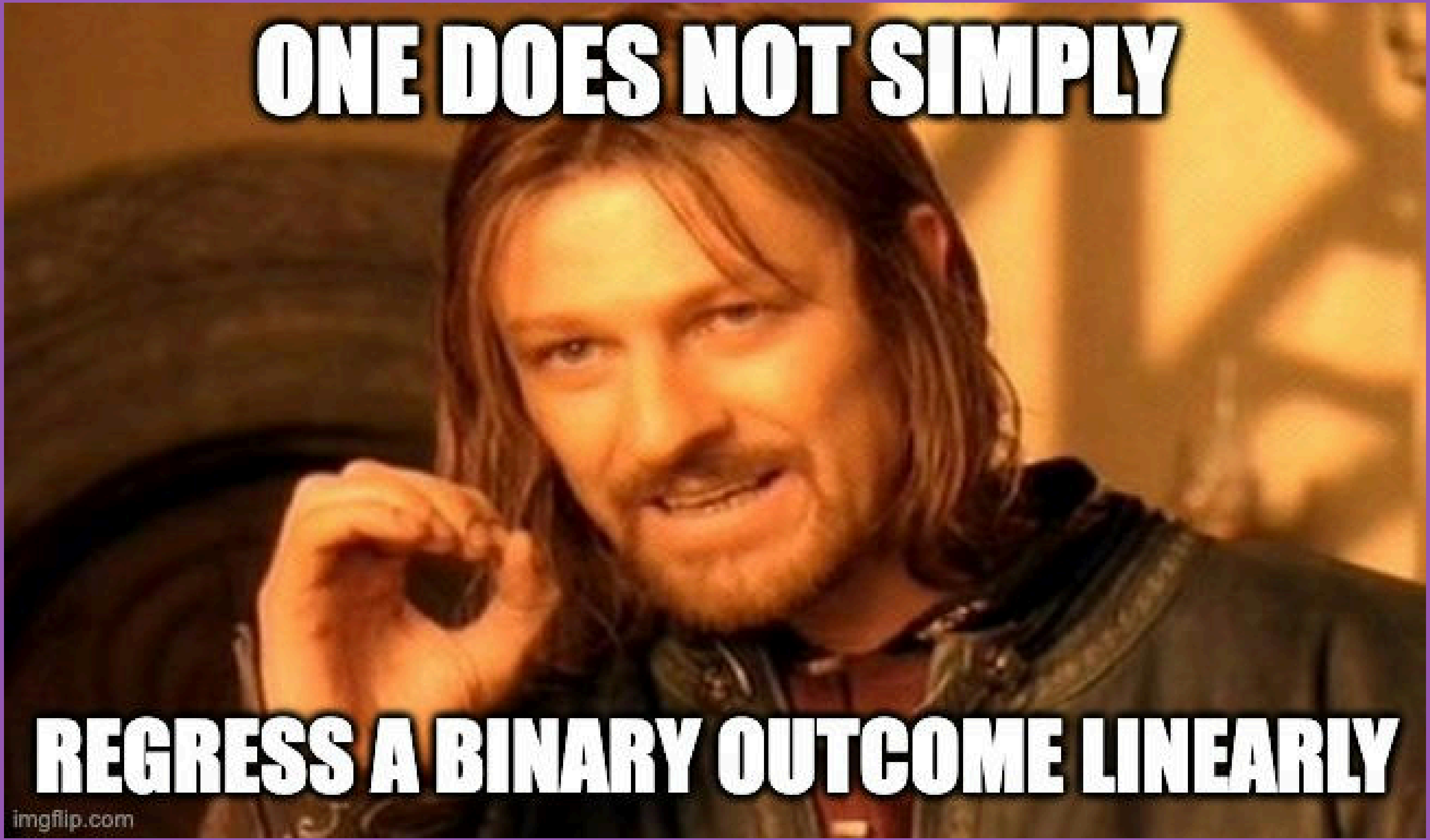
Our aim to is to **find the posterior distribution of the model's parameters of interest** (α and β) by computing the full posterior distribution of:

$$P(\boldsymbol{\theta} \mid \boldsymbol{y}) = P(\alpha, \boldsymbol{\beta}, \sigma \mid \boldsymbol{y})$$

Logistic Regression

Recommended References

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


ONE DOES NOT SIMPLY
REGRESS A BINARY OUTCOME LINEARLY
imgflip.com

Welcome to the Magical World of the Linear Generalized Models

Leaving the realm of the linear models, we start to adventure to the generalized linear models – GLM.

The first one is **logistic regression** (also called Bernoulli regression or binomial regression).

Binary Data[xxx]

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

[xxx] also known as dichotomous, dummy, indicator variable, etc.

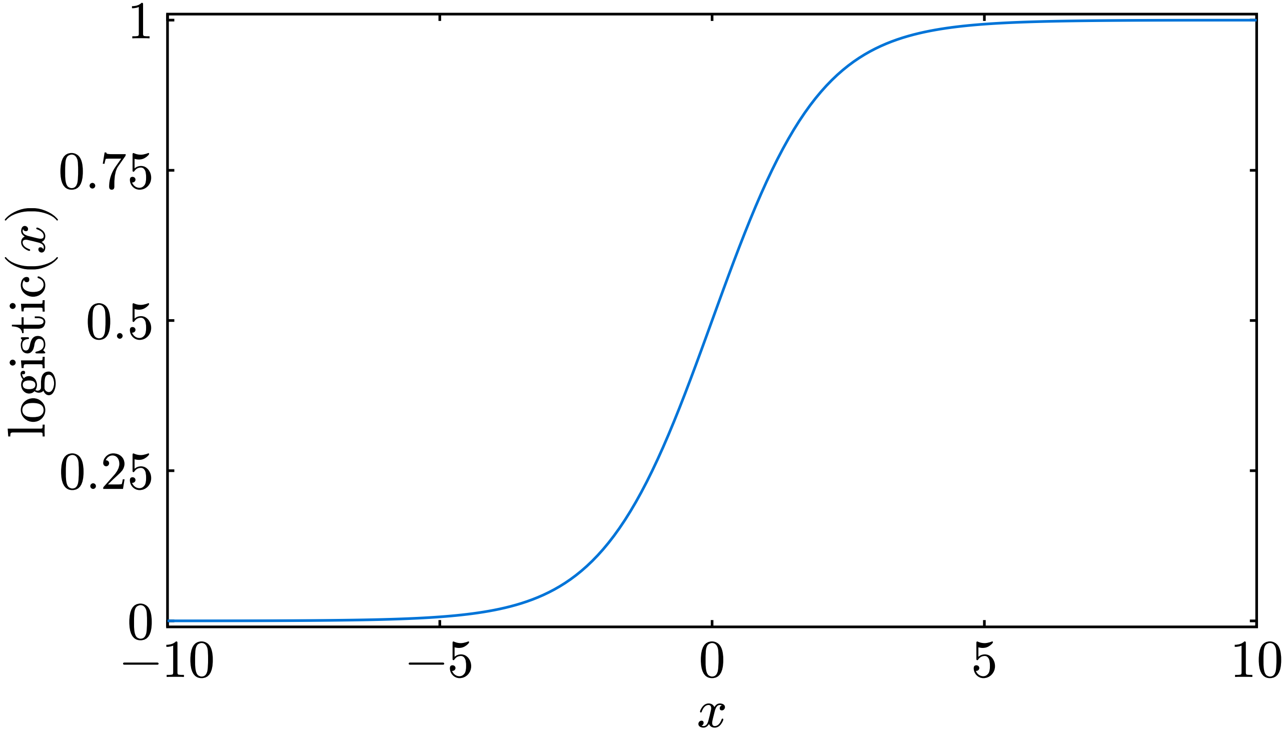
What is Logistic Regression

Logistic regression behaves exactly as a linear model: it makes a prediction by simply computing a weighted sum of the independent variables $\boldsymbol{X}$ using the estimated coefficients $\boldsymbol{\beta}$, along with a constant term α.

However, instead of outputting a continuous value y, it returns the **logistic function** of this value:

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

Logistic Function

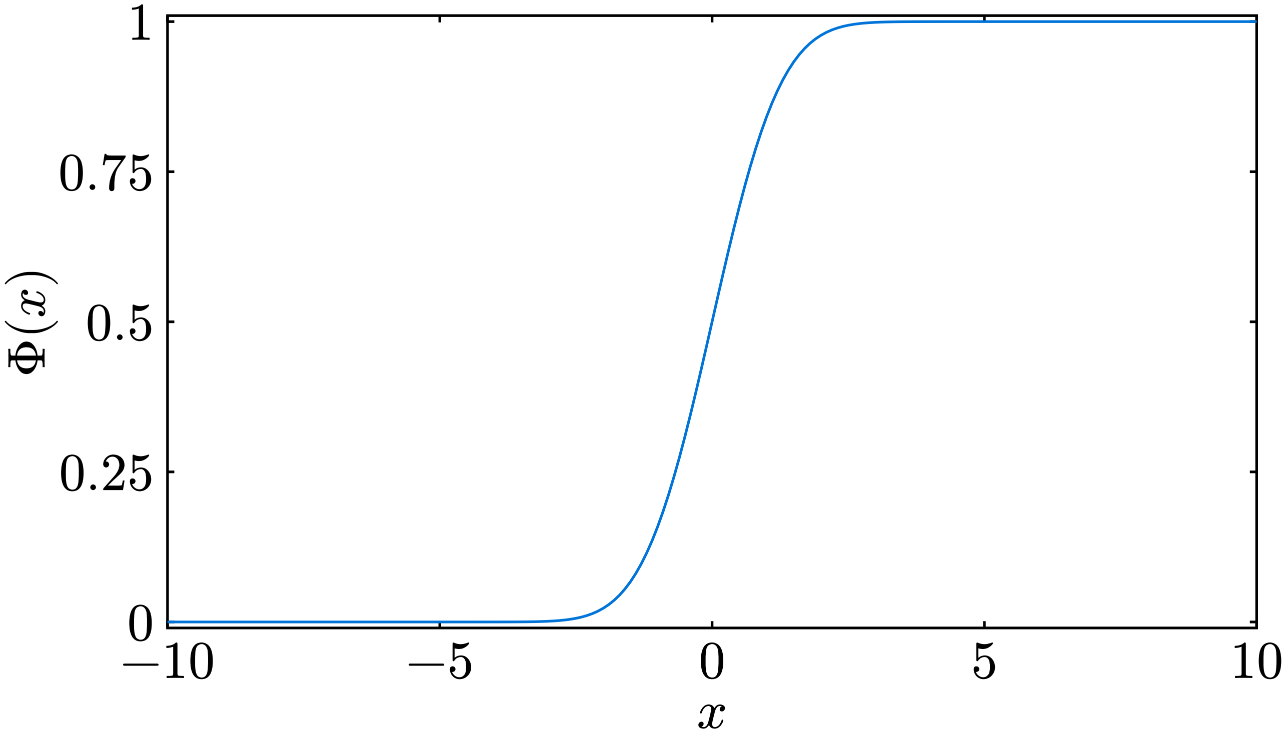


Probit Function

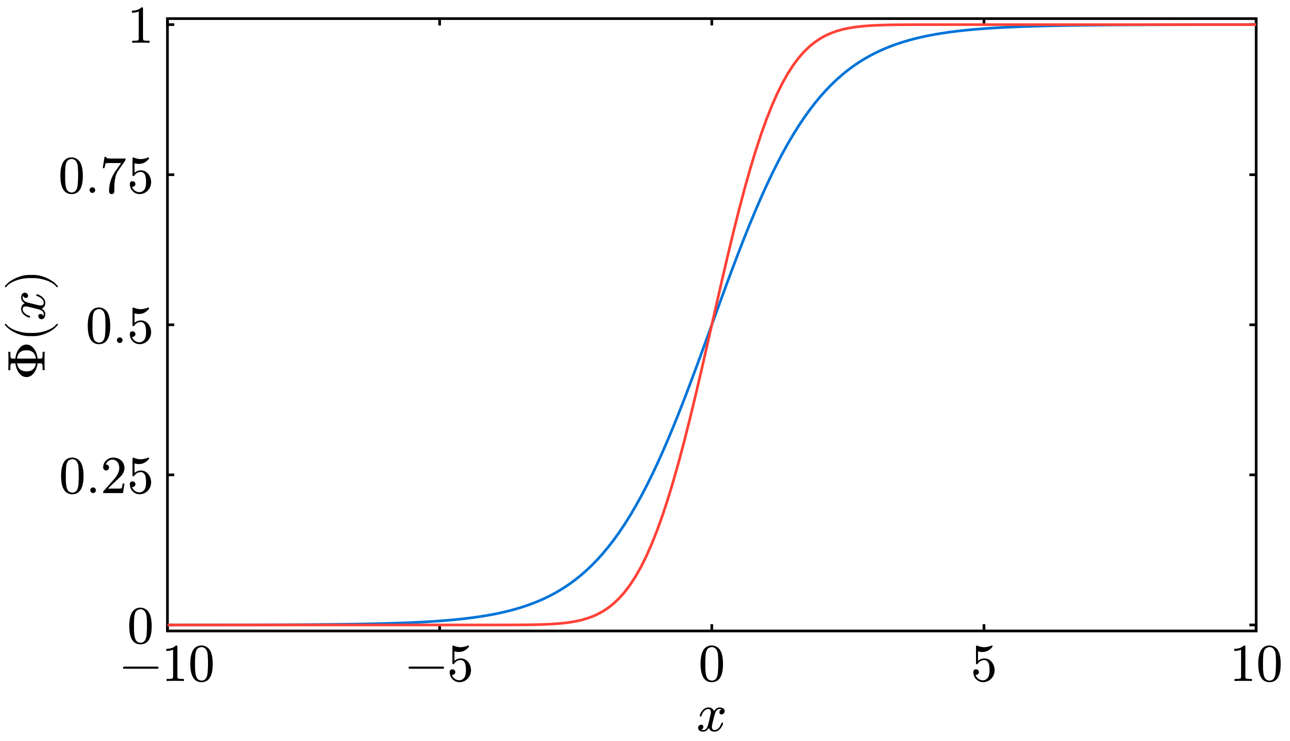
We can also opt to choose to use the **probit function** (usually represented by the Greek letter Φ) which is the CDF of a normal distribution:

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

Probit Function



Logistic Function versus Probit Function



Comparison with Linear Regression

Linear regression follows the following mathematical expression:

$$\operatorname{linear} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$$

- α – intercept.
- $\boldsymbol{\beta} = \beta_1, \beta_2, \ldots, \beta_k$ – independent variables’ $x_1, x_2, \ldots, x_k$ coefficients.
- k – number of independent variables.

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

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

Logistic Regression Specification

We can model logistic regression using two approaches:

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

Bernoulli Likelihood

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Bernoulli}(p) \\
p &= \text{logistic}/\text{probit}(\alpha + \boldsymbol{X}\boldsymbol{\beta}) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}})
\end{aligned}
$$

where:

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

Binomial Likelihood

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Binomial}(n, p) \\
p &= \text{logistic/probit}(\alpha + \boldsymbol{X}\boldsymbol{\beta}) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}})
\end{aligned}
$$

where:

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

Posterior Computation

Our aim to is to **find the posterior distribution of the model's parameters of interest** (α and β) by computing the full posterior distribution of:

$$P(\boldsymbol{\theta} \mid \boldsymbol{y}) = P(\alpha, \boldsymbol{\beta} \mid \boldsymbol{y})$$

How to Interpret Coefficients

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

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

Probability versus Odds

But before that, we need to discern between **probability and odds**[xxxi].

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

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

  where p is the probability.

[xxxi] mathematically speaking.

Probability versus Odds

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

where p is the probability.

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

Logodds

If you revisit the logistic function, you'll se that the intercept α and coefficients β are literally the **log of the odds** (logodds):

$$
\begin{aligned}
p &= \operatorname{logistic}(\alpha + \boldsymbol{X} \boldsymbol{\beta}) \\
p &= \operatorname{logistic}(\alpha) + \operatorname{logistic}(\boldsymbol{X} \boldsymbol{\beta}) \\
p &= \frac{1}{1 + e^{-\boldsymbol{\beta}}} \\
\boldsymbol{\beta} &= \log(\text{odds})
\end{aligned}
$$

Logodds

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

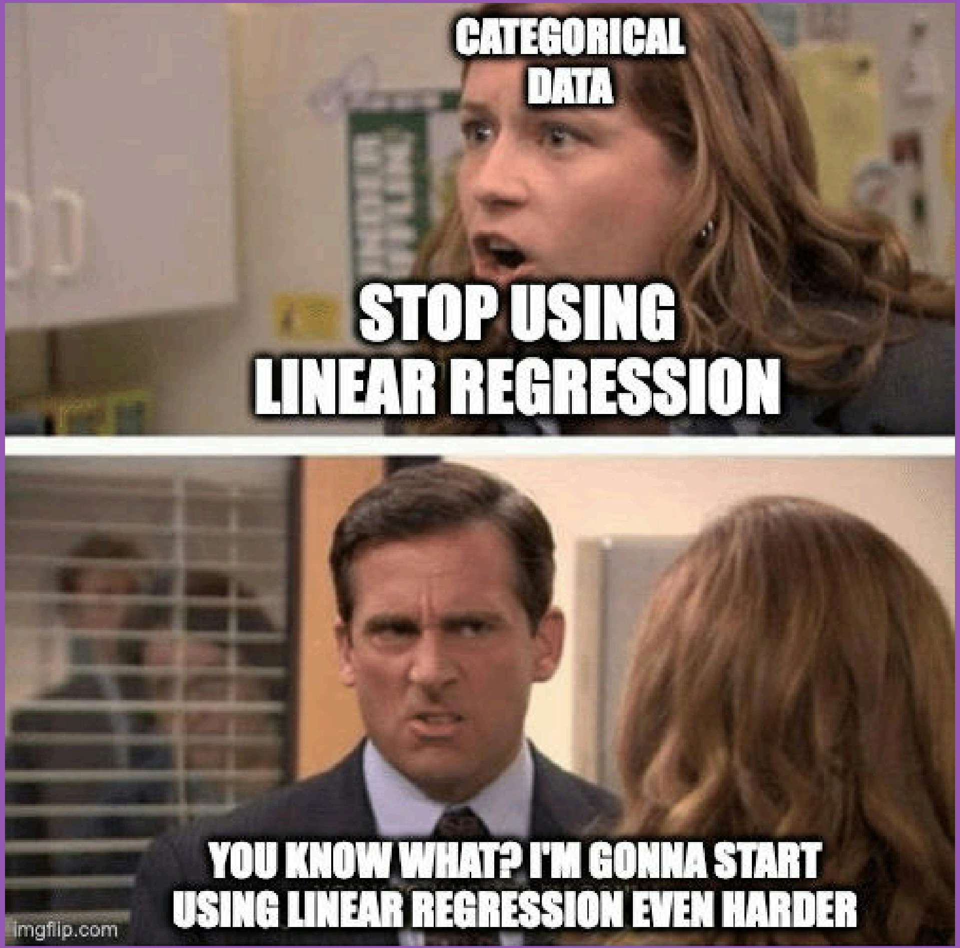
$$\operatorname{odds}(\alpha) = e^{\alpha}$$

$$\operatorname{odds}(\boldsymbol{\beta}) = e^{\boldsymbol{\beta}}$$

Ordinal Regression

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - Chapter 16: Generalized linear models, Section 16.2: Models for multivariate and multinomial responses
- McElreath (2020) - Chapter 12, Section 12.3: Ordered categorical outcomes
- Gelman, Hill and Vehtari (2020) - Chapter 15, Section 15.5: Ordered and unordered categorical regression
- Bürkner and Vuorre (2019)
- Semenova (2019)


CATEGORICAL DATA
STOP USING LINEAR REGRESSION
YOU KNOW WHAT? I'M GONNA START USING LINEAR REGRESSION EVEN HARDER
imgflip.com

What is Ordinal Regression?

Ordinal regression is a regression model for **discrete data** and, more specific, when the **values of the dependent variables have a “natural ordering”**.

For example, opinion polls with its plausible ordered values from agree-disagree, or a patient perception of pain score.

Why not just use Linear Regression?

The main reason to not simply use linear regression with ordinal discrete outcomes is that the categories of the dependent variable could not be **equidistant**.

This is an assumption in linear regression (and in almost all models that use “metric” dependent variables): the distance between, for example, 2 and 3 is not the same distance between 1 and 2.

This assumption can be **violated in an ordinal regression**.

How to deal with an Ordinal Dependent Variable?

Surprise! Plot twist!

Another **non-linear transformation**.

Cumulative Distribution Function – CDF

In the case of ordinal regression, first we need to transform the **dependent variable into a cumulative scale**

For this, we use the cumulative distribution function (CDF):

$$P(Y \leq y) = \sum_{i=y_{\min}}^{y} P(Y = i)$$

CDF is a **monotonically increasing function** that represents the **probability of a random variable Y taking values less than a certain value y**

Log-cumulative-odds

Still, this is not enough. We need to apply the **logit function onto the CDF**:

$$\operatorname{logit}(x) = \operatorname{logistic}^{-1}(x) = \ln\left(\frac{x}{1-x}\right)$$

where $\ln$ is the natural log function.

The logit function is the inverse of the logistic function: it takes as input any value between 0 and 1 (e.g. a probability) and outputs an unconstrained real number which we call **logodds**[xxxii].

As the transformation is performed onto the CDF, we call the result as the CDF logodds or **log-cumulative-odds**.

[xxxii] we already seen it in logistic regression.

$K - 1$ Intercepts

What do we do with this **log-cumulative-odds**?

It allows us to construct **different intercepts for all possible values of the ordinal dependent variable**. We create an **unique intercept for** $k \in K$.

Actually is $k \in K - 1$. Notice that the maximum value of the CDF of Y will always be 1. Which translates to a log-cumulative-odds of ∞, since $p = 1$:

$$\ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{1}{1-1}\right) = \ln(0) = \infty$$

Hence, we need only $K - 1$ **intercepts for all** K **possible values that** Y **can take**.

Violation of the Equidistant Assumption

Since each intercept implies a different CDF value for each $k \in K$, we can safely **violate the equidistant assumption** which is not valid in almost all ordinal variables.

Cut Points

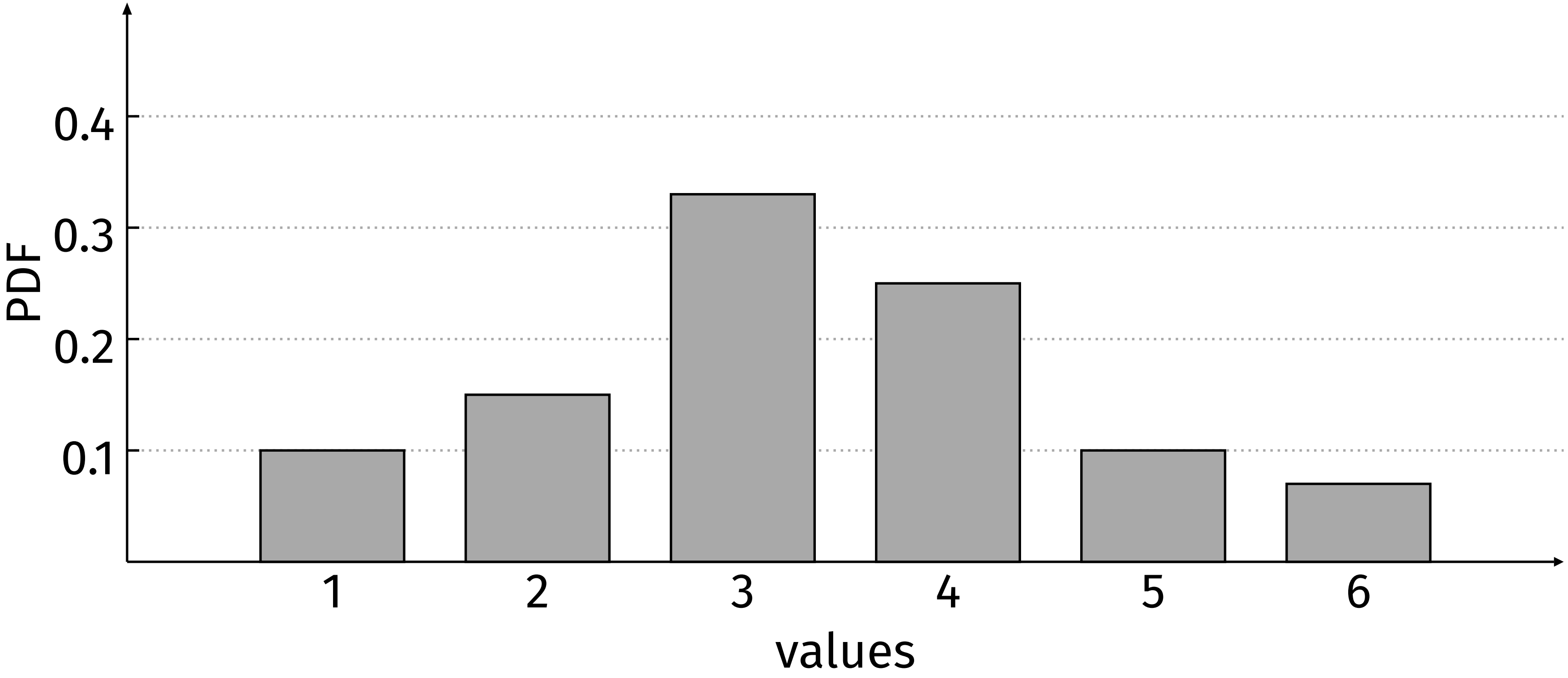
Each intercept implies in a log-cumulative-odds for each $k \in K$; We need also to **undo the cumulative nature of the $K - 1$ intercepts**. Firstly, we **convert the log-cumulative-odds back to a valid probability with the logistic function**:

$$\operatorname{logit}^{-1}(x) = \operatorname{logistic}(x) = \left(\frac{1}{1 + e^{-x}}\right)$$

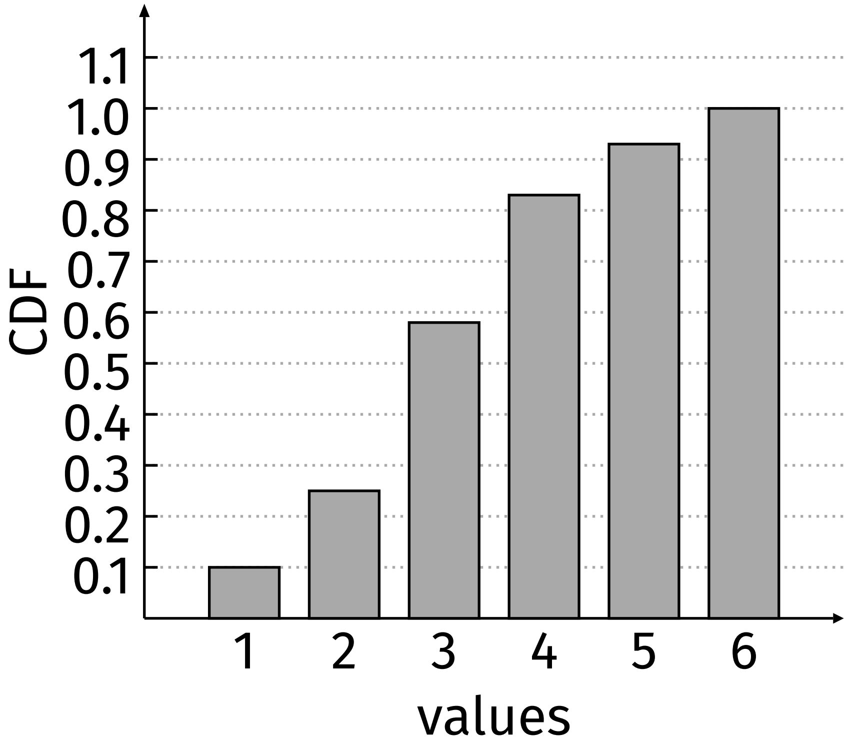
Then, finally, we remove the cumulative nature of the CDF by **subtracting every one of the k cut points by the $k - 1$ cut point**:

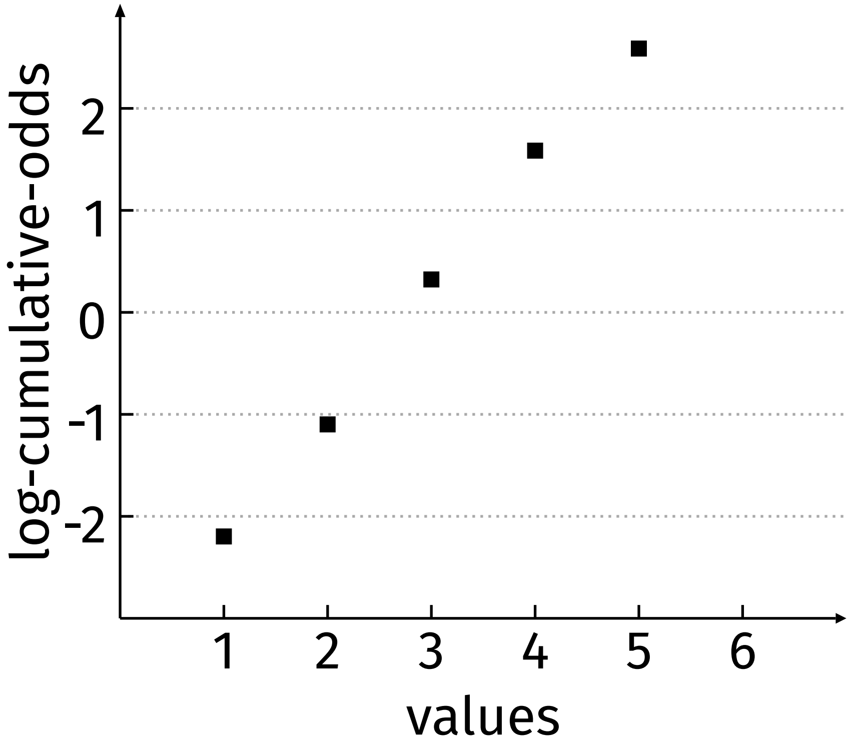
$$P(Y = k) = P(Y \leq k) - P(Y \leq k - 1)$$

Example - Probability Mass Function of an Ordinal Variable



Example - CDF versus log-cumulative-odds





Adding Coefficients β

With the equidistant assumption solved with $K - 1$ intercepts, we can add coefficients to represent the independent variable's effects into our ordinal regression model.

More Log-cumulative-odds

We’ve transformed all intercepts into log-cumulative-odds so that we can add effects as weighted sums of the independent variables to our basal rates (intercepts).

For every $k \in K - 1$, we calculate:

$$\varphi = \alpha_k + \beta_i x_i$$

where α_k is the log-cumulative-odds for the $k \in K - 1$ intercepts, β_i is the coefficient for the ith independent variable x_i.

Lastly, φ_k represents the linear predictor for the kth intercept.

Matrix Notation

This can become more elegant and computationally efficient if we use matrix/vector notation:

$$\boldsymbol{\varphi} = \boldsymbol{\alpha} + \boldsymbol{X}c \cdot \boldsymbol{\beta}$$

where $\boldsymbol{\varphi}$, $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$[xxxiii] are vectors and $\boldsymbol{X}$ is the data matrix, in which every line is an observation and every column an independent variable.

[xxxiii] note that both the coefficients and intercepts will have to be interpret as odds, like we did in logistic regression.

Ordinal Regression Specification

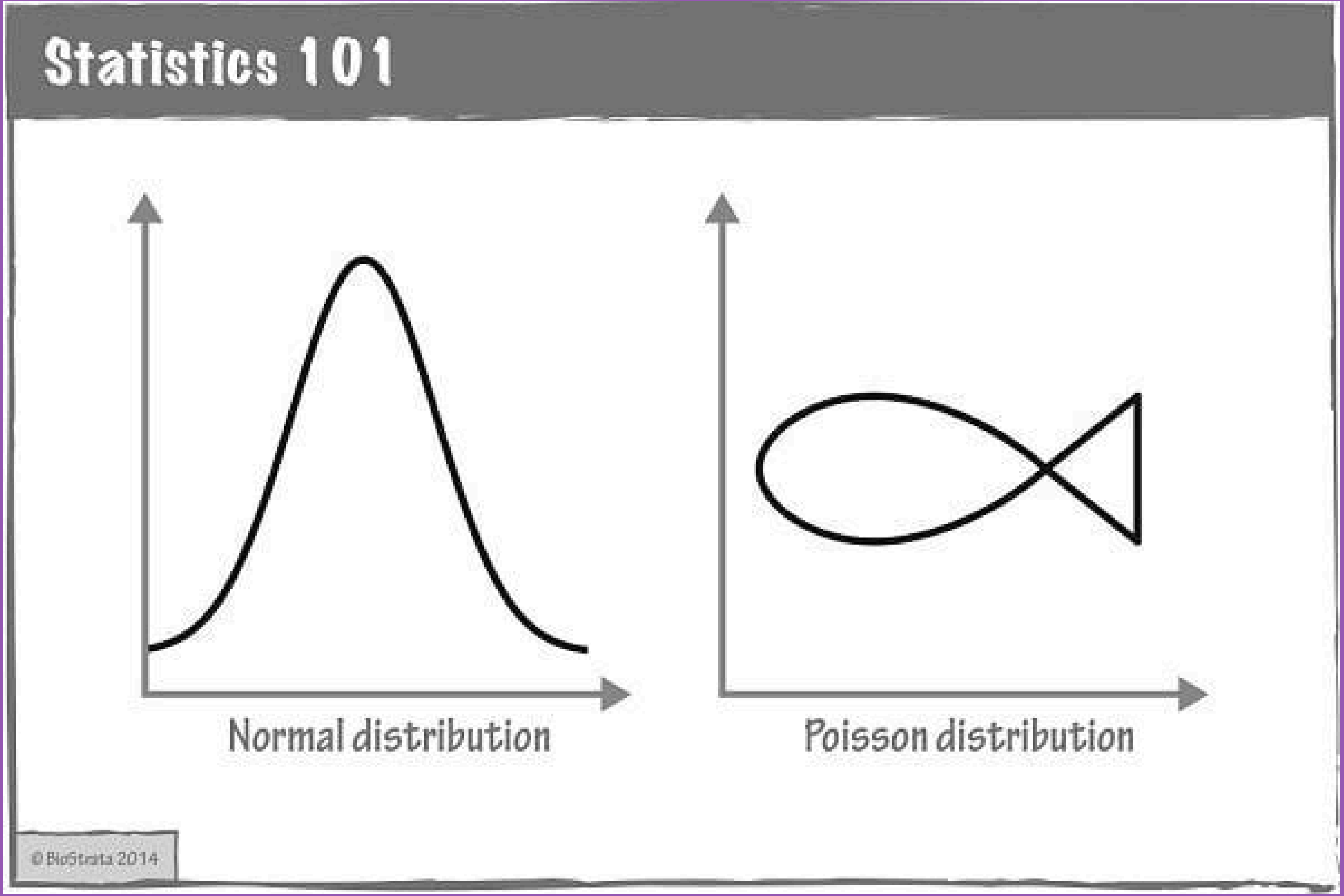
$$
\begin{aligned}
\boldsymbol{y} &\sim \operatorname{Categorical}(\boldsymbol{p}) \\
\boldsymbol{p} &= \operatorname{logistic}(\boldsymbol{\varphi}) \\
\boldsymbol{\varphi} &= \boldsymbol{\alpha} + \boldsymbol{c} + \boldsymbol{X}c \cdot \boldsymbol{\beta} \\
c_1 &= \operatorname{logit}(\operatorname{CDF}(y_1)) \\
c_k &= \operatorname{logit}(\operatorname{CDF}(y_k) - \operatorname{CDF}(y_{k-1})) \text{ for } 2 \leq k \leq K - 1 \\
c_K &= \operatorname{logit}(1 - \operatorname{CDF}(y_{K-1})) \\
\boldsymbol{\alpha} &\sim \operatorname{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \operatorname{Normal}\left(\mu_{\boldsymbol{\beta}}, \sigma_{\{\boldsymbol{\beta}\}}\right)
\end{aligned}
$$

- $\boldsymbol{y}$ – ordinal discrete dependent variable.
- $\boldsymbol{p}$ – probability vector of size K.
- K: number of possible values that $\boldsymbol{y}$ can take, i.e. number of ordered discrete values.
- $\boldsymbol{\varphi}$: log-cumulative-odds, i.e. the cut points considering the intercepts and the weighted sum of the independent variables.
- c_k: cutpoint in log-cumulative-odds for every $k \in K - 1$.
- α_k: intercept in log-cumulative-odds for every $k \in K - 1$.
- $\boldsymbol{X}$: data matrix of the independent variables.
- $\boldsymbol{\beta}$: coefficient vector with size the same as the number of columns of $\boldsymbol{X}$.

Poisson Regression

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - Chapter 16: Generalized linear models
- McElreath (2020):
  - Chapter 10: Big Entropy and the Generalized Linear Model
  - Chapter 11, Section 11.2: Poisson regression
- Gelman, Hill and Vehtari (2020) - Chapter 15, Section 15.2: Poisson and negative binomial regression



Statistics 101

Normal distribution

Poisson distribution

Count Data

Poisson regression is used when our dependent variable can only take **positive values**, usually in the context of **count data**.

What is Poisson Regression?

Poisson regression behaves exactly like a linear model: it makes a prediction by simply computing a weighted sum of the independent variables $\boldsymbol{X}$ with the estimated coefficients $\boldsymbol{\beta}$: $\boldsymbol{y}$.

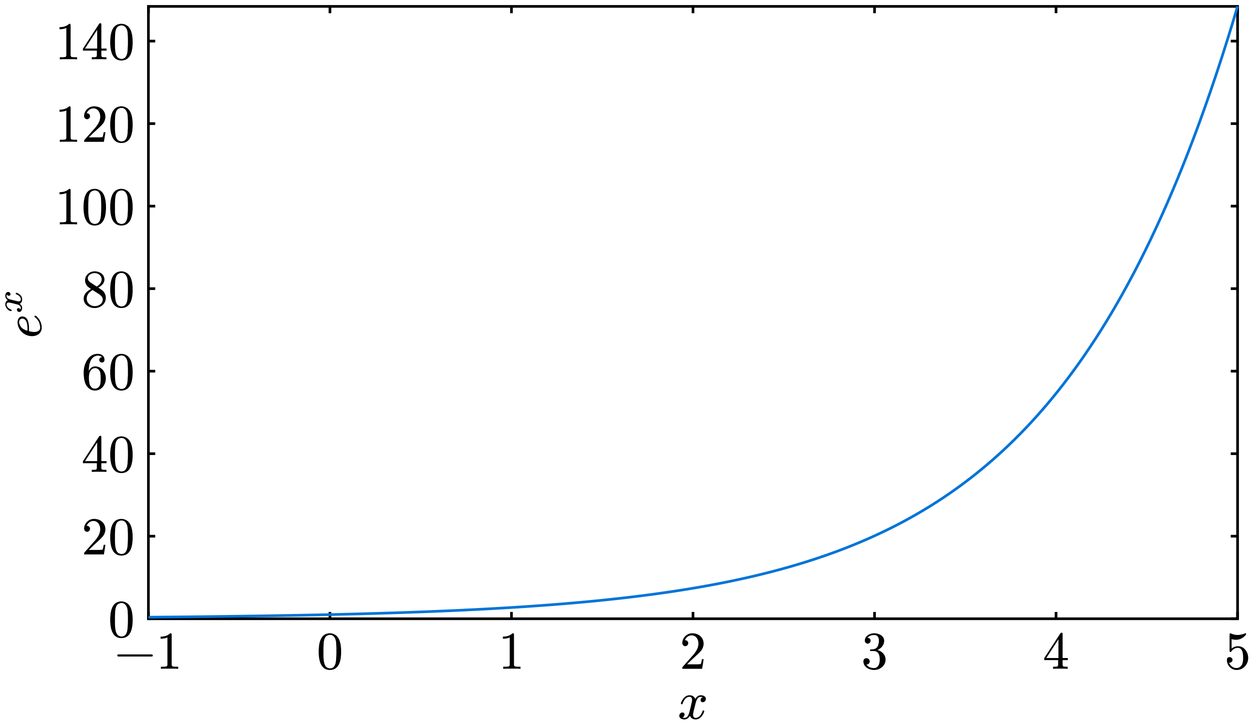
But, different from linear regression, it outputs the **natural log** of y:

$$\log(\boldsymbol{y}) = \alpha \cdot \beta_1 x_1 \cdot \beta_2 x_2 \cdot \ldots \cdot \beta_k x_k$$

which is the same as:

$$\boldsymbol{y} = e^{(\alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k)}$$

Exponential Function



Comparison with Linear Regression

Linear regression has the following mathematical expression:

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

where:

- α – intercept.
- $\boldsymbol{\beta} = \beta_1, \beta_2, \ldots, \beta_k$ – independent variables' $x_1, x_2, \ldots, x_k$ coefficients.
- k – number of independent variables.

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

- $\log(y) = e^{\text{Linear}} = e^{\alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k}$

Poisson Regression Specification

We can use Poisson regression if the dependent variable $\boldsymbol{y}$ has count data, i.e., $\boldsymbol{y}$ only takes positive values.

Poisson likelihood function uses an intercept α and coefficients $\boldsymbol{\beta}$, however these are "exponentiated" (e^x):

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Poisson}\left(e^{(\alpha + \boldsymbol{X}\boldsymbol{\beta})}\right) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}})
\end{aligned}
$$

Interpreting the Coefficients

When we see the Poisson regression specification, we realize that the coefficient interpretation requires a transformation. What we need to do is undo the logarithm transformation:

$$\log^{-1}(x) = e^x$$

So, we need to "exponentiate" the values of α and β:

$$\begin{aligned} \boldsymbol{y} &= e^{(\alpha + \boldsymbol{X} \boldsymbol{\beta})} \\ &= e^{\alpha} \cdot e^{\left(X_{(1)} \cdot \beta_{(1)}\right)} \cdot e^{\left(X_{(2)} \cdot \beta_{(2)}\right)} \cdot \ldots \cdot e^{\left(X_{(k)} \cdot \beta_{(k)}\right)} \end{aligned}$$

Interpreting the Coefficients

Finally, notice that, when transformed, our dependent variables is no more a “weighted sum of an intercept and independent variables”:

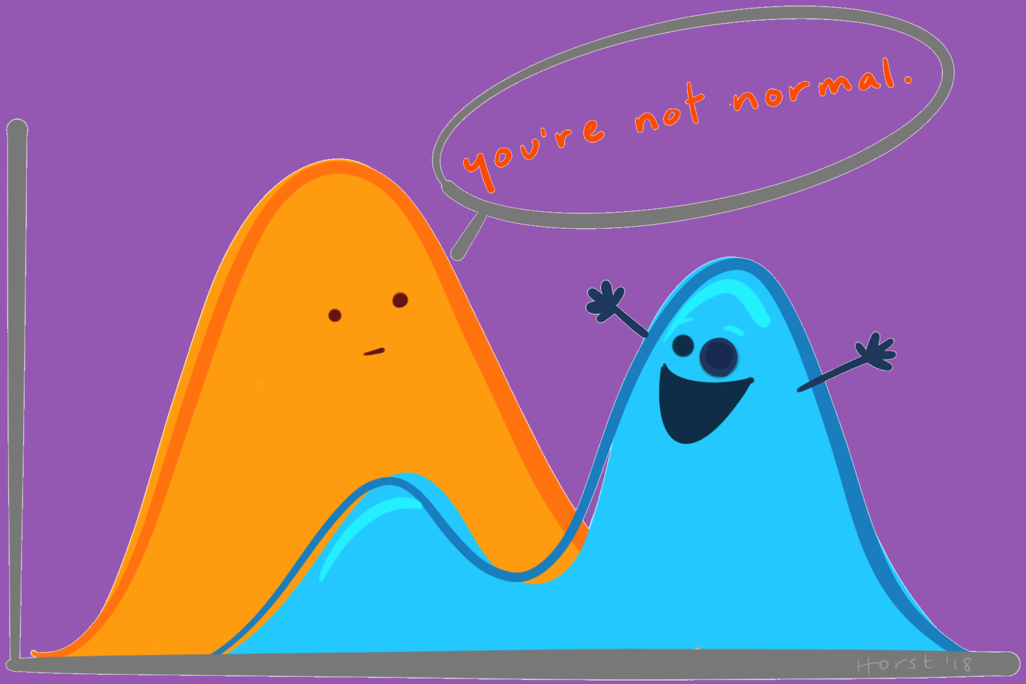
$$
\begin{aligned}
\boldsymbol{y} &= e^{(\alpha + \boldsymbol{X} \boldsymbol{\beta})} \\
&= e^{\alpha} \cdot e^{\left(X_{(1)} \cdot \beta_{(1)}\right)} \cdot e^{\left(X_{(2)} \cdot \beta_{(2)}\right)} \cdot \ldots \cdot e^{\left(X_{(k)} \cdot \beta_{(k)}\right)}
\end{aligned}
$$

It becomes a **“weighted product”**.

Robust Regression

Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - Chapter 17: Models for robust inference
- McElreath (2020) - Chapter 12: Monsters and Mixtures
- Gelman, Hill and Vehtari (2020):
  - Chapter 15, Section 15.6: Robust regression using the t model
  - Chapter 15, Section 15.8: Going beyond generalized linear models



you're not normal.
Horst '18

Robust Models

Almost always data from real world are really strange.

For the sake of convenience, we use simple models. But always ask yourself. How many ways might the posterior inference depends on the following:

- extreme observations (outliers)?
- unrealistic model assumptions?

Outliers

Models based on the **normal distribution are notoriously “non-robust” against outliers**, in the sense that a **single observation can greatly affect the inference of all model’s parameters**, even those that has a shallow relationship with it.

Overdispersion

Superdispersion and underdispersion[xxxiv] refer to data that have more or fewer variation than expected under a probability model (Gelman, Hill and Vehtari, 2020).

For each one of the models we covered, there is a **natural extension** in which **a single parameter** is added to allow for overdispersion (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

[xxxiv] rarer to find in the real world.

Overdispersion Example

Suppose you are analyzing data from car accidents. The model we generally use in this type of phenomena is **Poisson regression**.

Poisson distribution has the same parameter for both the mean and variance: the rate parameter λ.

Hence, if you find a higher variability than expected under the Poisson likelihood function allows, then probably you won't be able to model properly the desired phenomena.

Student's t instead of Normal

Student's t distribution has **wider**[xxxv] **tails** than the Normal distribution.

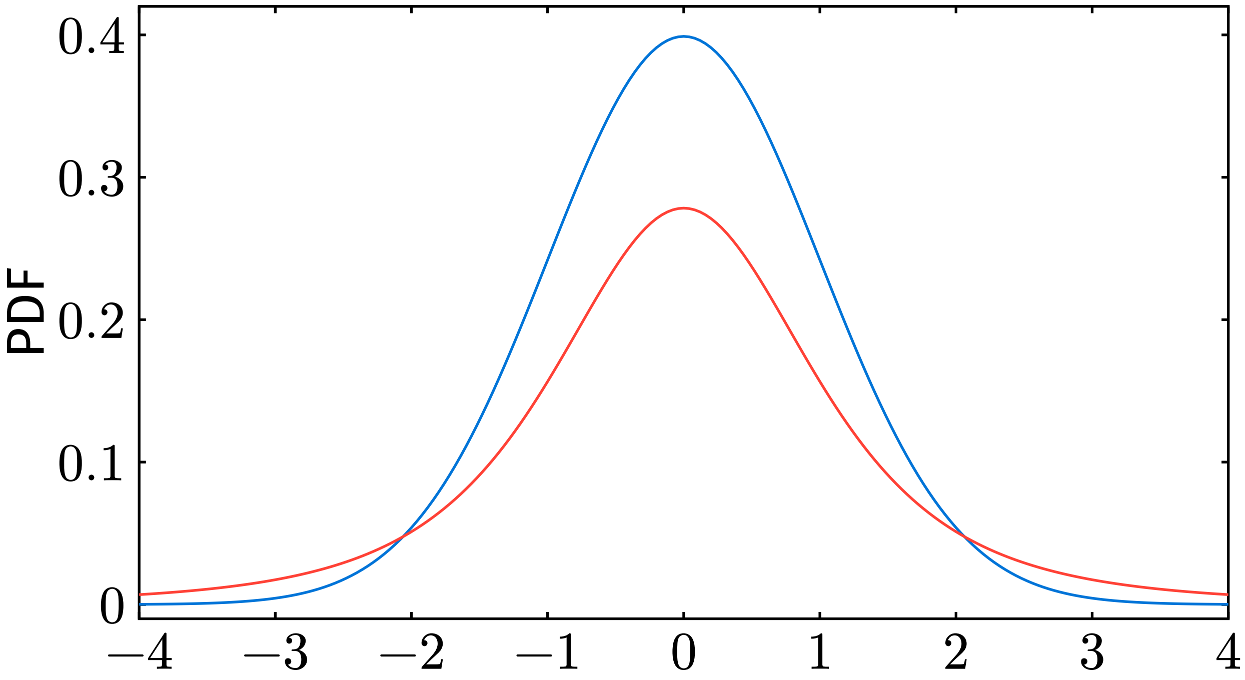
This makes it a good candidate to **fit outliers without instabilities in the parameters' inference**.

From the Bayesian viewpoint, there is nothing special or magical in the Gaussian/Normal likelihood.

It is just another distribution specified in a statistical model. We can make our model robust by using the Student's t distribution as a likelihood function.

[xxxv] or "fatter".

Student's t instead of Normal



Student's t instead of Normal

By using a Student's t distribution instead of the Normal distribution as likelihood functions, the model's error σ does *not* follow a Normal distribution, but a Student's t distribution:

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Student}(\nu, \alpha + \boldsymbol{X}\boldsymbol{\beta}, \sigma) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}\big(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}\big) \\
\nu &\sim \text{Log-Normal}(2, 1) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

Note that we are including an extra parameter ν, which represents the Student's t distribution degrees of freedom, to be estimated by the model (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

This controls how wide or narrow the "tails" of the distribution will be. A heavy-tailed, positive-only prior is advised.

Beta-Binomial instead of the Binomial

The binomial distribution has a practical limitation that we only have one free parameter to estimate[xxxvi] (p). This implies in the **variance to determined by the mean**. Hence, the binomial distribution **cannot** tolerate overdispersion.

A robust alternative is the **beta-binomial distribution**, which, as the name suggests, is a **beta mixture of binomials distributions**. Most important, it **allows that the variance to be independent of the mean**, making it **robust against overdispersion**.

[xxxvi] since n already comes from data.

Beta-Binomial instead of Binomial

The **beta-binomial distribution** is a binomial distribution, where the probability of success p is parameterized as a $\mathrm{Beta}(\alpha, \beta)$.

Generally, we use α as the binomial's probability of the success p, and β[xxxvii] is the additional parameter to control and allow for overdispersion.

Values of $\beta \geq 1$ make the beta-binomial behave the same as a binomial.

[xxxvii] sometimes specified as φ

Beta-Binomial instead of Binomial

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Beta-Binomial}(n, p, \varphi) \\
p &\sim \text{Logistic/Probit}(\alpha + \boldsymbol{X}\boldsymbol{\beta}) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}\left(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}\right) \\
\varphi &\sim \text{Exponential}(1)
\end{aligned}
$$

It is also proper to include the overdispersion β parameter as an additional parameter to be estimated by the model (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013; McElreath, 2020). A heavy-tailed, positive-only prior is advised.

Student's t instead Binomial

Also known as Robit[xxxviii] (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013; Gelman, Hill and Vehtari, 2020). The idea is to make the logistic regression robust by using a **latent variable** z as the linear predictor. z's errors, ε, are distributed as a Student's t distribution:

$$
\begin{aligned}
y_i &= \begin{cases} 0 \text{ if } z_i < 0 \\ 1 \text{ if } z_i > 0 \end{cases} \\
z_i &= X_i \boldsymbol{\beta} + \varepsilon_i \\
\varepsilon_i &\sim \text{Student}\left(\nu, 0, \sqrt{\frac{\nu - 2}{\nu}}\right) \\
\nu &\sim \text{Gamma}(2, 0.1) \in [2, \infty)
\end{aligned}
$$

Here we are using the gamma distribution as a truncated Student's t distribution for the degrees of freedom parameter $\nu \geq 2$. Another option would be to fix $\nu = 4$.

[xxxviii] there is a great discussion between Gelman, Vehtari and Kurz at Stan’s Discourse .

Negative Binomial instead of Poisson

This is the overdispersion example. The Poisson distribution uses a **single parameter for both its mean and variance**.

Hence, if you find overdispersion, probably you'll need a robust alternative to Poisson. This is where the **negative binomial**, with an extra parameter φ, that makes it **robust to overdispersion**.

φ controls the probability of success p, and we generally use a gamma distribution as its prior. φ is also known as a "reciprocal dispersion" parameter.

Negative Binomial instead of Poisson

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Negative Binomial}\left(e^{(\alpha + \boldsymbol{X} \boldsymbol{\beta})}, \varphi\right) \\
\varphi &\sim \text{Gamma}(0.01, 0.01) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}\left(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}\right)
\end{aligned}
$$

Here we also give a heavy-tailed, positive-only prior to φ. Something like the $\text{Gamma}(0.01, 0.01)$ works.

Negative Binomial Mixture instead of Poisson

Even using a negative binomial likelihood, if you encounter acute overdispersion, specially when there is a lot of zeros in your data (zero-inflated), your model can still perform a bad fit to the data.

Another suggestion is to use a mixture of negative binomial (McElreath, 2020).

Negative Binomial Mixture instead of Poisson

Here, S_i is a dummy variable, taking value 1 if the ith observation has a value $\neq 0$. S_i can be modeled using logistic regression:

$$
\begin{aligned}
\boldsymbol{y} &\begin{cases} = 0 \text{ if } S_i = 0 \\ \sim \text{Negative Binomial}\left(e^{(\alpha + \boldsymbol{X} \boldsymbol{\beta})}, \varphi\right) \text{ if } S_i = 1 \end{cases} \\
P(S_i = 1) &= \text{Logistic/Probit}(\boldsymbol{X} \boldsymbol{\gamma}) \\
\gamma &\sim \text{Beta}(1, 1)
\end{aligned}
$$

γ is a new coefficients which we give uniform prior of $\text{Beta}(1, 1)$.

Why Use Non-Robust Models?

The **central limit theorem** tells us that the **normal distribution** is an appropriate model for data that arises as a **sum of independent components**.

Even when they are naturally not implicit in a phenomena structure, **simpler non-robust models are computational efficient**.

Finally, there's **occam's razor**, also known as the **principle of parsimony**, which states the preference for simplicity in the scientific method.

Of course, you must always guide the model choice in a **principled manner**, taking into account the underlying phenomena data generating process. And make sure to make **posterior predictive checks**.

Sparse Regression

Recommended References

- Gelman, Hill and Vehtari (2020) - Chapter 12, Section 12.8: Models for regression coefficients
- Horseshoe Prior: Carvalho, Polson and Scott (2009)
- Horseshoe+ Prior: Bhadra *et al.* (2015)
- Regularized Horseshoe Prior: Piironen and Vehtari (2017)
- R2-D2 Prior: Zhang *et al.* (2022)
- Betancourt's Case study on Sparsity: Betancourt (2021)

What is Sparsity?

Sparsity is a concept frequently encountered in statistics, signal processing, and machine learning, which refers to situations where the vast majority of elements in a dataset or a vector are zero or close to zero.

How to Handle Sparsity?

Almost all techniques deal with some sort of **variable selection**, instead of altering data.

This makes sense from a Bayesian perspective, as data is **information**, and we don't want to throw information away.

Frequentist Approach

The frequentist approach deals with sparse regression by staying in the “optimization” context but adding **Lagrangian constraints**[xxxix]:

$$\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - x_i^T \beta \right)^2 \right\}$$

suject to $\| \ \beta \ \|_p \leq t$.

Here $\| \cdot \|_p$ is the p-norm.

[xxxix] this is called **LASSO** (least absolute shrinkage and selection operator) from Tibshirani (1996); Zou and Hastie (2005).

Variable Selection Techniques

- **discrete mixtures**: *spike-and-slab* prior
- **shrinkage priors**: *Laplace* prior and *horseshoe* prior (Carvalho, Polson and Scott, 2009)

Discrete Mixtures – Spike-and-Slab Prior

Mixture of two distributions—one that is concentrated at zero (the “spike”) and one with a much wider spread (the “slab”). This prior indicates that we believe most coefficients in our model are likely to be zero (or close to zero), but we allow the possibility that some are not.

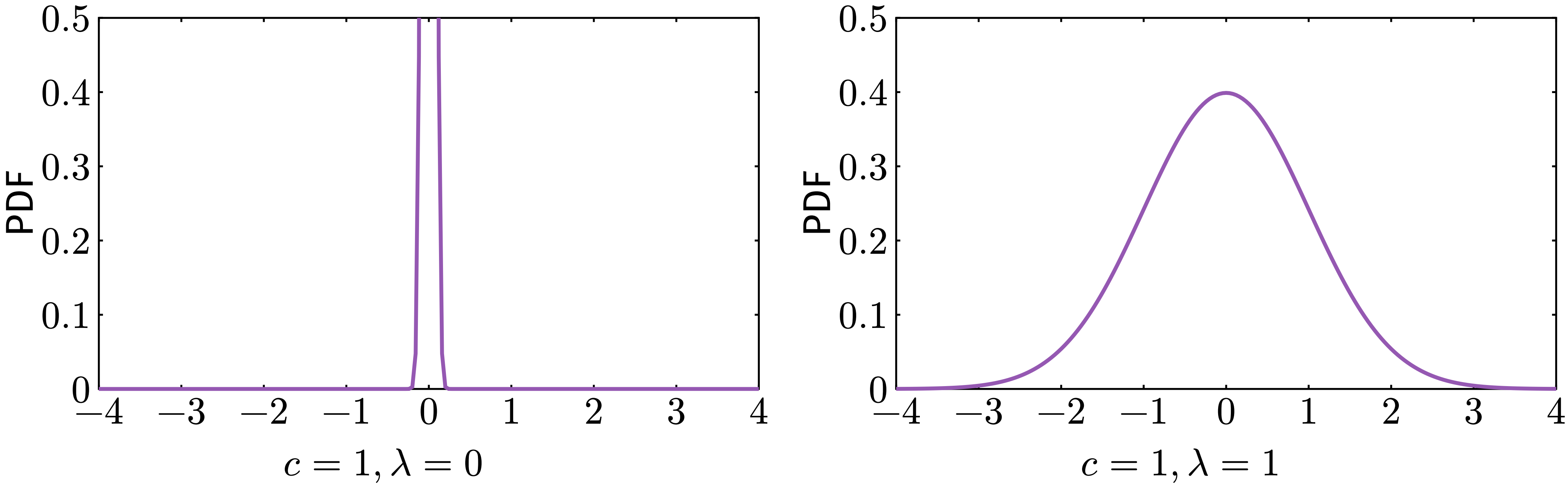
Here is the Gaussian case:

$$\beta_i \mid \lambda_i, c \sim \operatorname{Normal}\left(0, \sqrt{\lambda_i^2 c^2}\right)$$
$$\lambda_i \sim \operatorname{Bernoulli}(p)$$

where:

- c: *slab* width
- p: prior inclusion probability; encodes the prior information about the sparsity of the coefficient vector β
- $\lambda_i \in \{0, 1\}$: whether the coefficient β_i is close to zero (comes from the “spike”, $\lambda_i = 0$) or nonzero (comes from the “slab”, $\lambda_i = 1$)

Discrete Mixtures – Spike-and-Slab Prior



Shinkrage Priors – Laplace Prior

The Laplace distribution is a continuous probability distribution named after Pierre-Simon Laplace. It is also known as the double exponential distribution.

It has parameters:

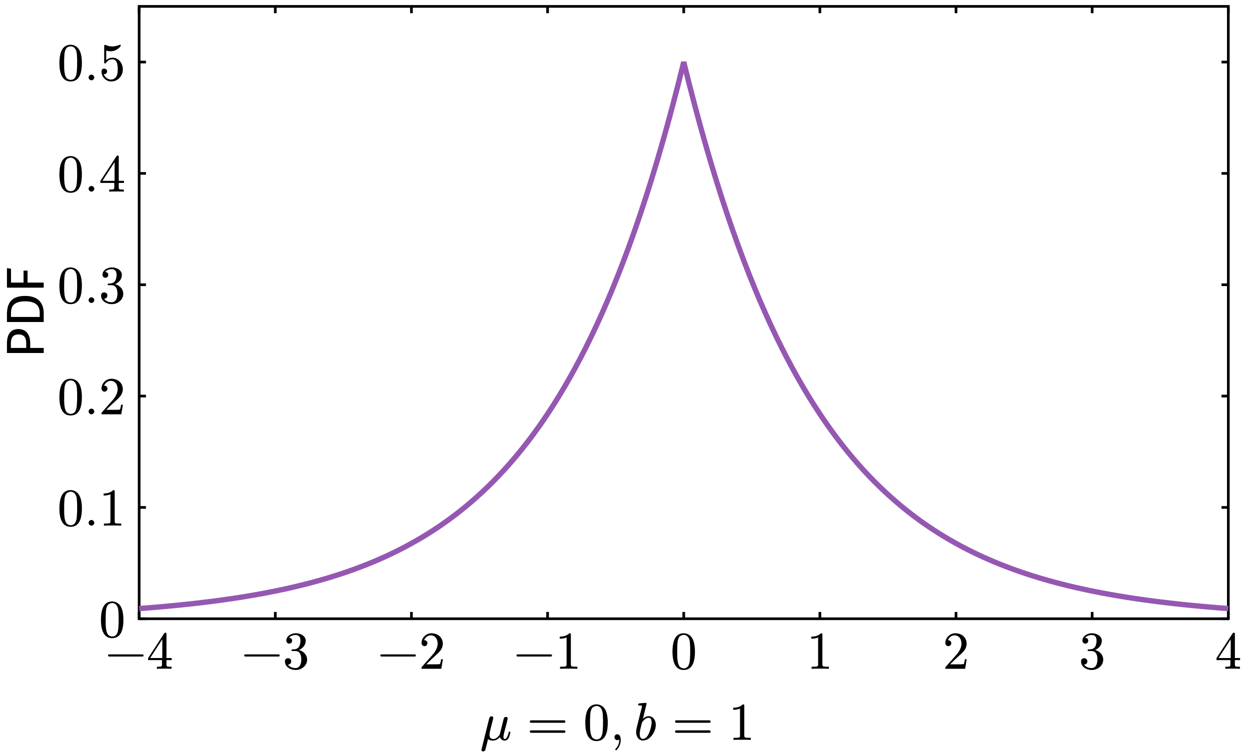
- μ: location parameter
- b: scale parameter

The PDF is:

$$\operatorname{Laplace}(\mu, b) = \frac{1}{2b} e^{-\left(\frac{| x-\mu |}{b}\right)}$$

It is a symmetrical exponential decay around μ with scale governed by b.

Shinkrage Priors – Laplace Prior



Shinkrage Priors – Horseshoe Prior

The horseshoe prior (Carvalho, Polson and Scott, 2009) assumes that each coefficient β_i is conditionally independent with density $P_{\text{HS}}(\beta_i \mid \tau)$, where P_{HS} can be represented as a scale mixture of Gaussians:

$$\beta_i \mid \lambda_i, \tau \sim \operatorname{Normal}\left(0, \sqrt{\lambda_i^2 \tau^2}\right)$$

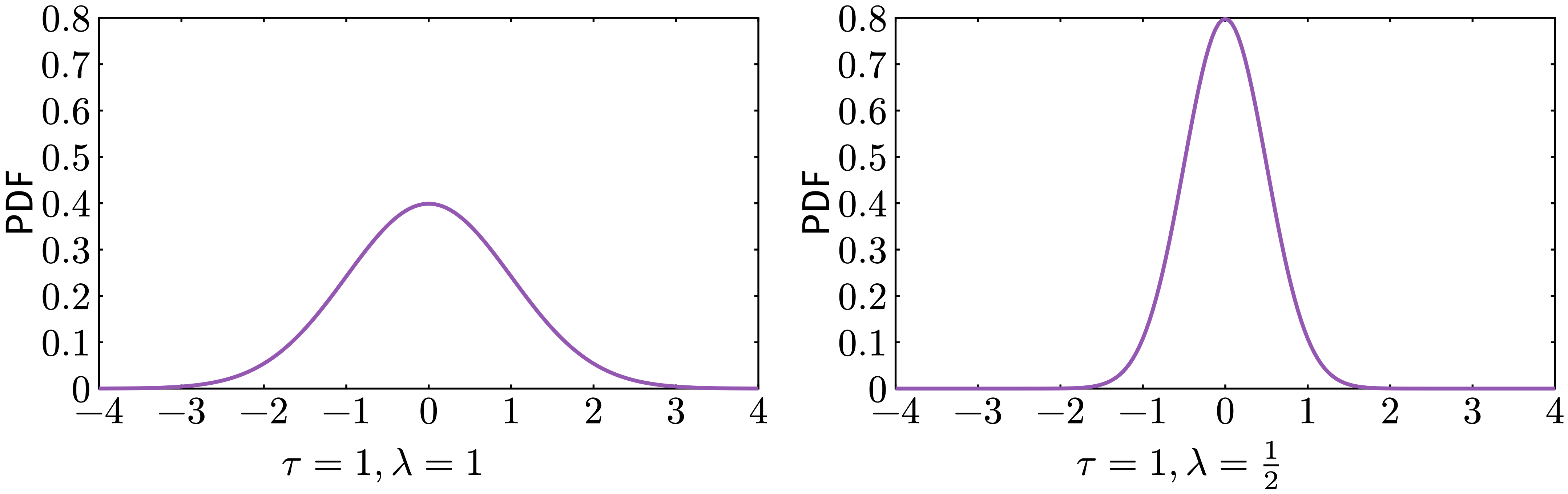
$$\lambda_i \sim \operatorname{Cauchy}^+(0, 1)$$

where:

- τ: *global* shrinkage parameter
- λ_i: *local* shrinkage parameter
- Cauchy^+ is the half-Cauchy distribution for the standard deviation λ_i

Note that it is similar to the spike-and-slab, but the discrete mixture becomes a “continuous” mixture with the Cauchy^+.

Discrete Mixtures – Spike-and-Slab Prior



Discrete Mixtures versus Shinkrage Priors

Discrete mixtures offer the correct representation of sparse problems (Carvalho, Polson and Scott, 2009) by placing positive prior probability on $\beta_i = 0$ (regression coefficient), but pose several difficulties: mostly computational due to the **non-continuous nature**.

Shrinkage priors, despite not having the best representation of sparsity, can be very attractive computationally: again due to the **continuous property**.

Horseshoe versus Laplace

The advantages of the Horseshoe prior over the Laplace prior are primarily:

- **shrinkage**: The Horseshoe prior has infinitely heavy tails and an infinite spike at zero. Parameters estimated under the Horseshoe prior can be shrunken towards zero more aggressively than under the Laplace prior, promoting sparsity without sacrificing the ability to detect true non-zero signals.
- **signal** detection: Due to its heavy tails, the Horseshoe prior does not overly penalize large values, which allows significant effects to stand out even in the presence of many small or zero effects.
- **uncertainty** quantification: With its heavy-tailed nature, the Horseshoe prior better captures uncertainty in parameter estimates, especially when the truth is close to zero.
- **regularization**: In high-dimensional settings where the number of predictors can exceed the number of observations, the Horseshoe prior acts as a strong regularizer, automatically adapting to the underlying sparsity level without the need for external tuning parameters.

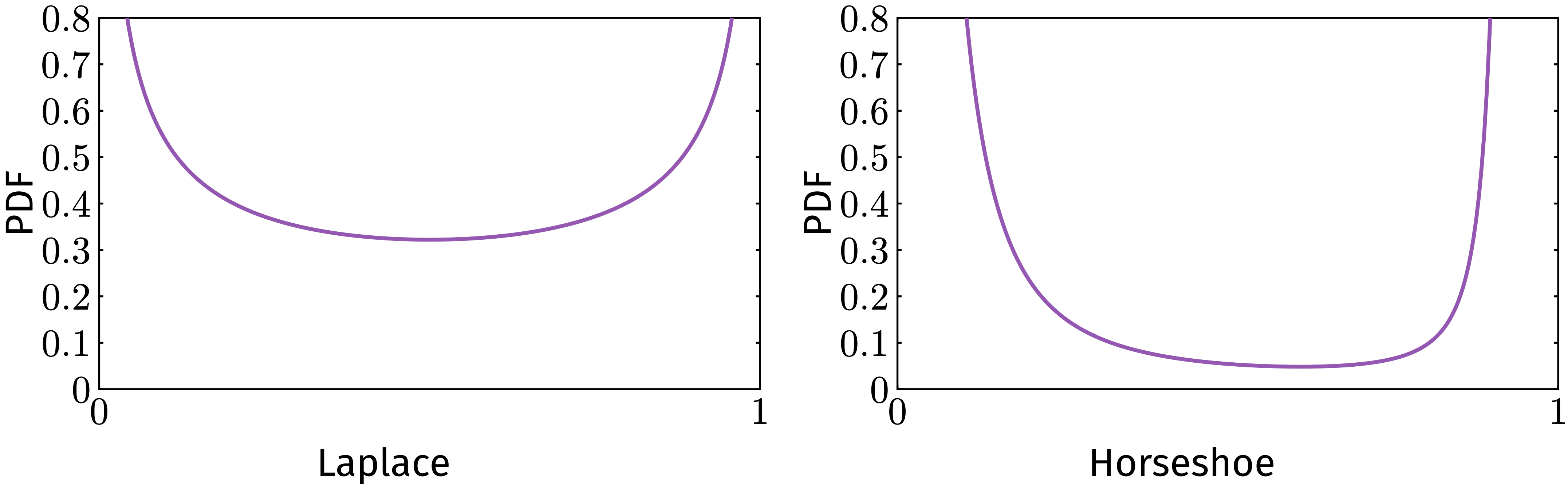
Effective Shinkrage Comparison

Makes more sense to compare the shinkrage effects of the proposed approaches so far. Assume for now that $\sigma^2 = \tau^2 = 1$, and define $\kappa_i = \frac{1}{1+\lambda_i^2}$.

Then κ_i is a random shrinkage coefficient, and can be interpreted as the amount of weight that the posterior mean for β_i places on 0 once the data $\boldsymbol{y}$ have been observed:

$$E(\beta_i \mid y_i, \lambda_i^2) = \frac{\lambda_i^2}{1+\lambda_i^2} y_i + \frac{1}{1+\lambda_i^2} 0 = (1 - \kappa_i) y_i$$

Effective Shinkrage Comparison[xl]



[xl] spike-and-slab with $p = \frac{1}{2}$ would be very similar to Horseshoe but with discontinuities.

Shinkrage Priors – Horseshoe+

Natural extension from the Horseshoe that has improved performance with highly sparse data (Bhadra *et al.*, 2015).

Just introduce a new half-Cauchy mixing variable η_i in the Horseshoe:

$$\begin{aligned}
\beta_i \mid \lambda_i, \eta_i, \tau &\sim \text{Normal}(0, \lambda_i) \\
\lambda_i \mid \eta_i, \tau &\sim \text{Cauchy}^+(0, \tau\eta_i) \\
\eta_i &\sim \text{Cauchy}^+(0, 1)
\end{aligned}$$

where:

- τ: *global* shrinkage parameter
- λ_i: *local* shrinkage parameter
- η_i: additional *local* shrinkage parameter
- Cauchy^+ is the half-Cauchy distribution for the standard deviation λ_i and η_i

Shinkrage Priors – Regularized Horseshoe

The Horseshoe and Horseshoe+ guarantees that the strong signals will not be overshrunk. However, this property can also be harmful, especially when the parameters are weakly identified.

The solution, Regularized Horseshoe (Piironen and Vehtari, 2017) (also known as the “Finnish Horseshoe”), is able to control the amount of shrinkage for the largest coefficient.

Shinkrage Priors – Regularized Horseshoe

$$\beta_i \mid \lambda_i, \tau, c \sim \text{Normal}\left(0, \sqrt{\tau^2 \tilde{\lambda_i}^2}\right)$$

$$\tilde{\lambda_i}^2 = \frac{c^2 \lambda_i^2}{c^2 + \tau^2 \lambda_i^2}$$

$$\lambda_i \sim \text{Cauchy}^+(0, 1)$$

where:

- τ: *global* shrinkage parameter
- λ_i: *local* shrinkage parameter
- $c > 0$: regularization constant
- Cauchy^+ is the half-Cauchy distribution for the standard deviation λ_i

Note that when $\tau^2 \lambda_i^2 \ll c^2$ (coefficient $\beta_i \approx 0$), then $\tilde{\lambda_i}^2 \to \lambda_i^2$; and when $\tau^2 \lambda_i^2 \gg c^2$ (coefficient β_i far from 0), then $\tilde{\lambda_i}^2 \to \frac{c^2}{\tau^2}$ and β_i prior approaches $\text{Normal}(0, c)$.

Shinkrage Priors – R2-D2

Still, we can do better. The **R2-D2**[xli] prior (Zhang *et al.*, 2022) has heavier tails and higher concentration around zero than the previous approaches.

The idea is to, instead of specifying a prior on $\boldsymbol{\beta}$, we construct a prior on the coefficient of determination R^2footnote{ square of the correlation coefficient between the dependent variable and its modeled expectation.}. Then using that prior to "distribute" throughout the $\boldsymbol{\beta}$.

[xli] R^2-induced Dirichlet Decomposition

Shinkrage Priors – R2-D2

$$
\begin{aligned}
R^2 &\sim \operatorname{Beta}(\mu_{R^2}\sigma_{R^2}, (1 - \mu_{R^2})\sigma_{R^2}) \\
\varphi &\sim \operatorname{Dirichlet}(J, 1) \\
\tau^2 &= \frac{R^2}{1 - R^2} \\
\boldsymbol{\beta} &= Z \cdot \sqrt{\varphi\tau^2}
\end{aligned}
$$

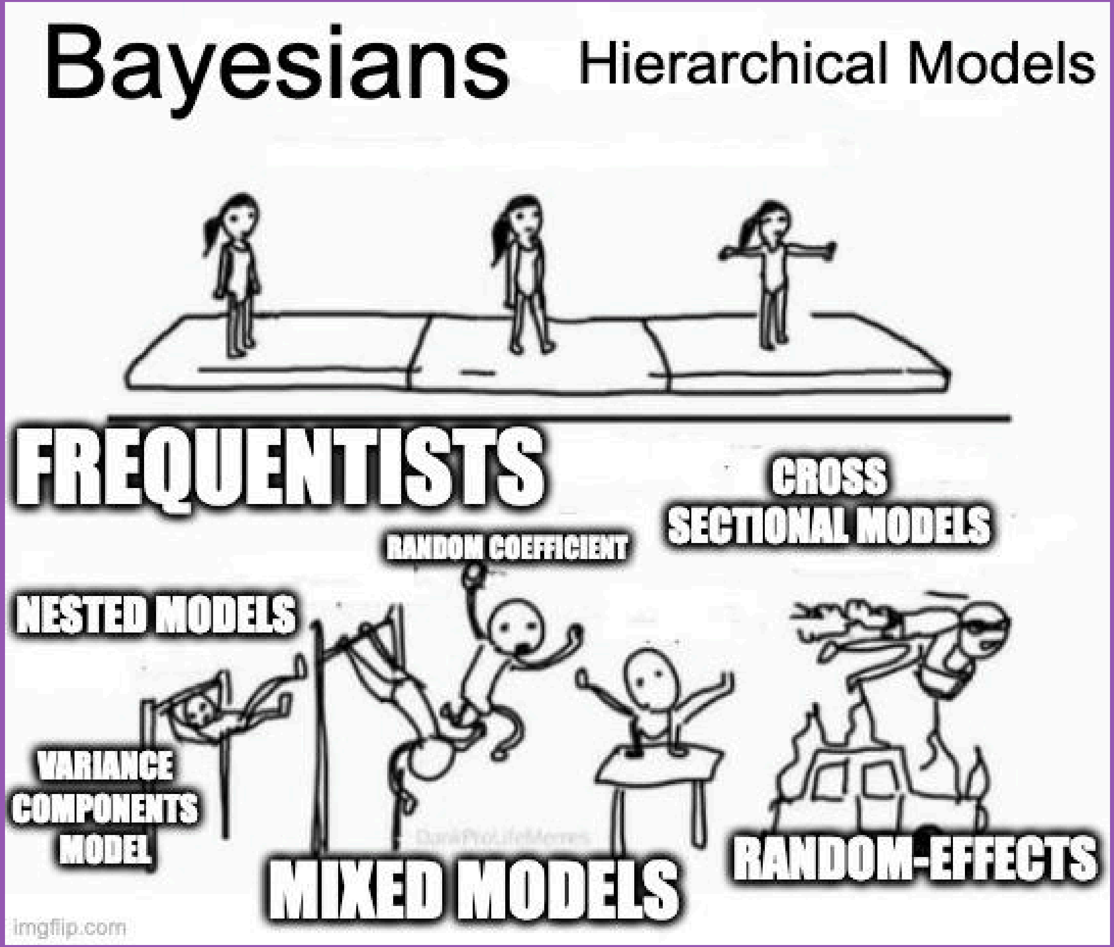
where:

- τ: *global* shrinkage parameter
- φ: proportion of total variance allocated to each covariate, can be interpreted as the *local* shrinkage parameter
- μ_{R^2} is the mean of the R^2 parameter, generally $\frac{1}{2}$
- σ_{R^2} is the precision of the R^2 parameter, generally 2
- Z is the standard Gaussian, i.e. $\operatorname{Normal}(0, 1)$

Hierarchical Models

Recommended References

- Gelman, Hill and Vehtari (2020):
  - 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)


Bayesians
Hierarchical Models
FREQUENTISTS
CROSS SECTIONAL MODELS
RANDOM COEFFICIENT
NESTED MODELS
VARIANCE COMPONENTS MODEL
MIXED MODELS
RANDOM-EFFECTS
imgflip.com

I have many names...

Hierarchical models are also known for several names[xlii]

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

[xlii] for the whole full list check here.

What are hierarchical models?

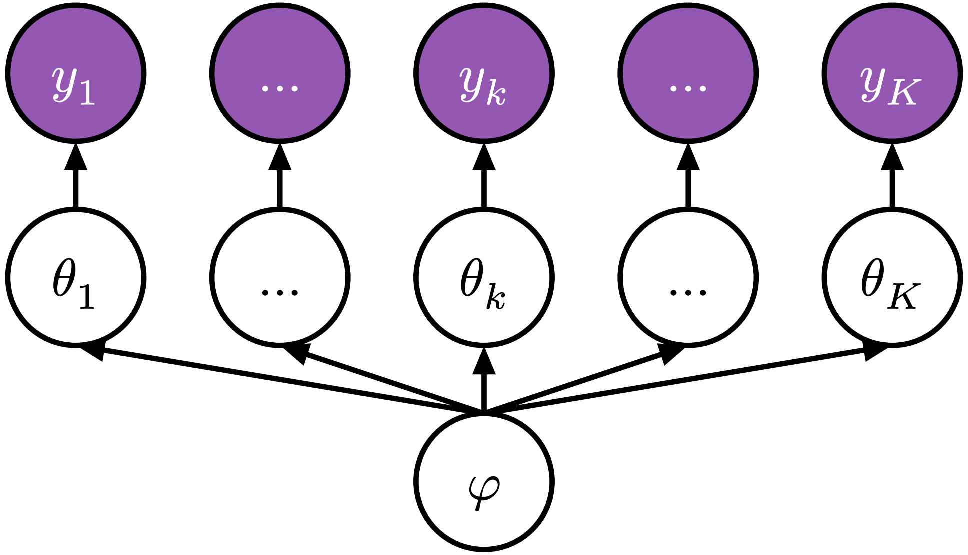
Statistical model specified in multiple levels that estimates parameters from the posterior distribution using a Bayesian approach.

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.

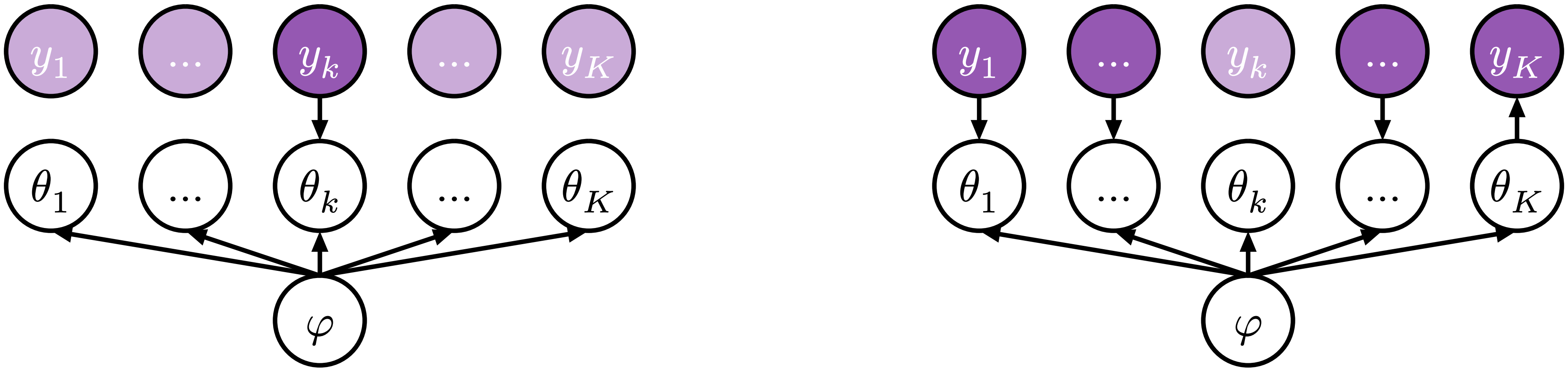
What are hierarchical models?

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



What are hierarchical models?

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 kth group, y_k, informs directly the parameters that quantify the kth group’s behavior, θ_k. These parameters, however, inform directly the population-level parameters, φ, 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…

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[xliii].

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

[xliii] also known as nested data.

When to Use Hierarchical Models?

Another good use case is **big data** (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

- 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** (Finetti, 1974).

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

Hyperprior

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

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

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

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**[xliv]. 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.

[xliv] see https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html [Douglas Bates, creator of the lme4 package explanation].

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 coefficients.

- **Varying-slope** model: One or more group-level coefficient(s) besides the population-level intercept.

- **Varying-intercept-slope** model: One group-level intercept and one or more group-level coefficient(s).

Mathematical Specification of Hierarchical Models

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

Mathematical Specification – Varying-Intercept Model

This example is for linear regression:

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

Mathematical Specification – Varying-Intercept Model

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

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Normal}\left(\alpha_{j_1} + \alpha_{j_2} + \boldsymbol{X}\boldsymbol{\beta}, \sigma\right) \\
\alpha_{j_1} &\sim \text{Normal}\left(\alpha_1, \tau_{\alpha_{j_1}}\right) \\
\alpha_{j_2} &\sim \text{Normal}\left(\alpha_2, \tau_{\alpha_{j_2}}\right) \\
\alpha_1 &\sim \text{Normal}\left(\mu_{\alpha_1}, \sigma_{\alpha_1}\right) \\
\alpha_2 &\sim \text{Normal}\left(\mu_{\alpha_2}, \sigma_{\alpha_2}\right) \\
\boldsymbol{\beta} &\sim \text{Normal}\left(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}\right) \\
\tau_{\alpha_{j_1}} &\sim \text{Cauchy}^+\left(0, \psi_{\alpha_{j_1}}\right) \\
\tau_{\alpha_{j_2}} &\sim \text{Cauchy}^+\left(0, \psi_{\alpha_{j_2}}\right) \\
\sigma &\sim \text{Exponential}\left(\lambda_\sigma\right)
\end{aligned}
$$

Mathematical Specification – Varying-(Intercept-)Slope Model

If we want a varying intercept, we just insert a column filled with 1s in the data matrix $\boldsymbol{X}$.

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 – Varying-(Intercept-)Slope Model

Hence, we have as a data matrix:

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

Mathematical Specification – Varying-(Intercept-)Slope Model

This example is for linear regression:

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

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

Mathematical Specification – Varying-(Intercept-)Slope Model

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

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

Priors for Covariance Matrices

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

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

$$\boldsymbol{\Sigma} = \mathrm{diag}_{\mathrm{matrix}}(\boldsymbol{\tau}) \cdot \boldsymbol{\Omega} \cdot \mathrm{diag}_{\mathrm{matrix}}(\boldsymbol{\tau})$$

where $\boldsymbol{\Omega}$ is a correlation matrix with 1s in the diagonal and the off-diagonal elements between -1 e 1 $\rho \in (-1, 1)$.

$\boldsymbol{\tau}$ is a vector composed of the variables' standard deviation from $\boldsymbol{\Sigma}$ (is is the $\boldsymbol{\Sigma}$'s diagonal).

Priors for Covariance Matrices

Additionally, the correlation matrix $\boldsymbol{\Omega}$ 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):

$$\boldsymbol{\Omega} = \boldsymbol{L}_{\Omega} \boldsymbol{L}_{\Omega}^{T}$$

where $\boldsymbol{L}_{\Omega}$ is a lower-triangular matrix.

What we are missing is to define a prior for the correlation matrix $\boldsymbol{\Omega}$. Not a long time ago, we've used a Wishart distribution as a prior (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

But this has been abandoned after the proposal of the LKJ distribution by Lewandowski, Kurowicka and Joe (2009)[xlv] as a prior for correlation matrices.

[xlv] LKJ are the authors' last name initials – Lewandowski, Kurowicka and Joe.

Markov Chain Monte Carlo (MCMC) and Model Metrics

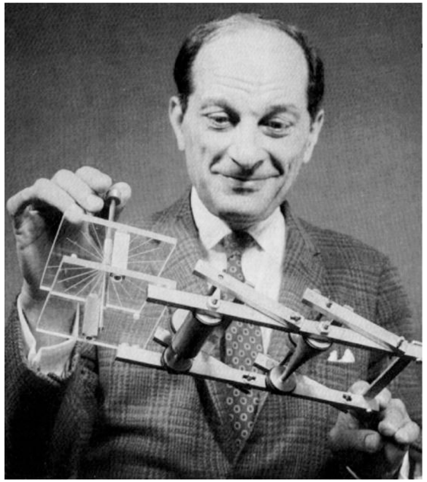
Recommended References

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013):
  - 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)

WHEN YOU ESTIMATE A BAYESIAN MODEL
This little maneuver is gonna cost us 51 years
imgflip.com

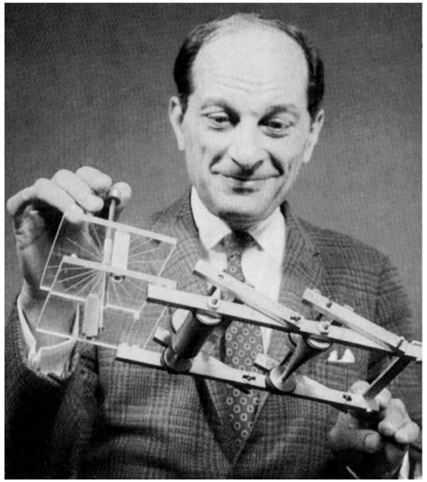
Monte Carlo Methods

- 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[xlvi]

- 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.



[xlvi] those who are interested, should read Eckhardt (1987).

Why Do We Need MCMC?

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

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

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

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

This is also known as **marginalization**:

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

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:

$$P(\text{data}) = \int_\theta P(\text{data} \mid \theta) \cdot P(\theta) \, \mathrm{d}\theta$$

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:

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

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 $\boldsymbol{\theta}$ directly from the posterior probability $P(\boldsymbol{\theta} \mid \text{data})$.

Instead, we collect samples in an iterative manner, where every step of the process we expect that the distribution which we are sampling from $P^*\left(\boldsymbol{\theta}^{(*)} \mid \text{data}\right)$ becomes more similar in every iteration to the posterior $P(\boldsymbol{\theta} \mid \text{data})$.

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**[xlvii] in which the set of possible states is the sample size and the stationary distribution is the distribution to be *approximated* (or *sampled*).
- Let $X_0, X_1, ..., X_n$ 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.

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

Markov Chains

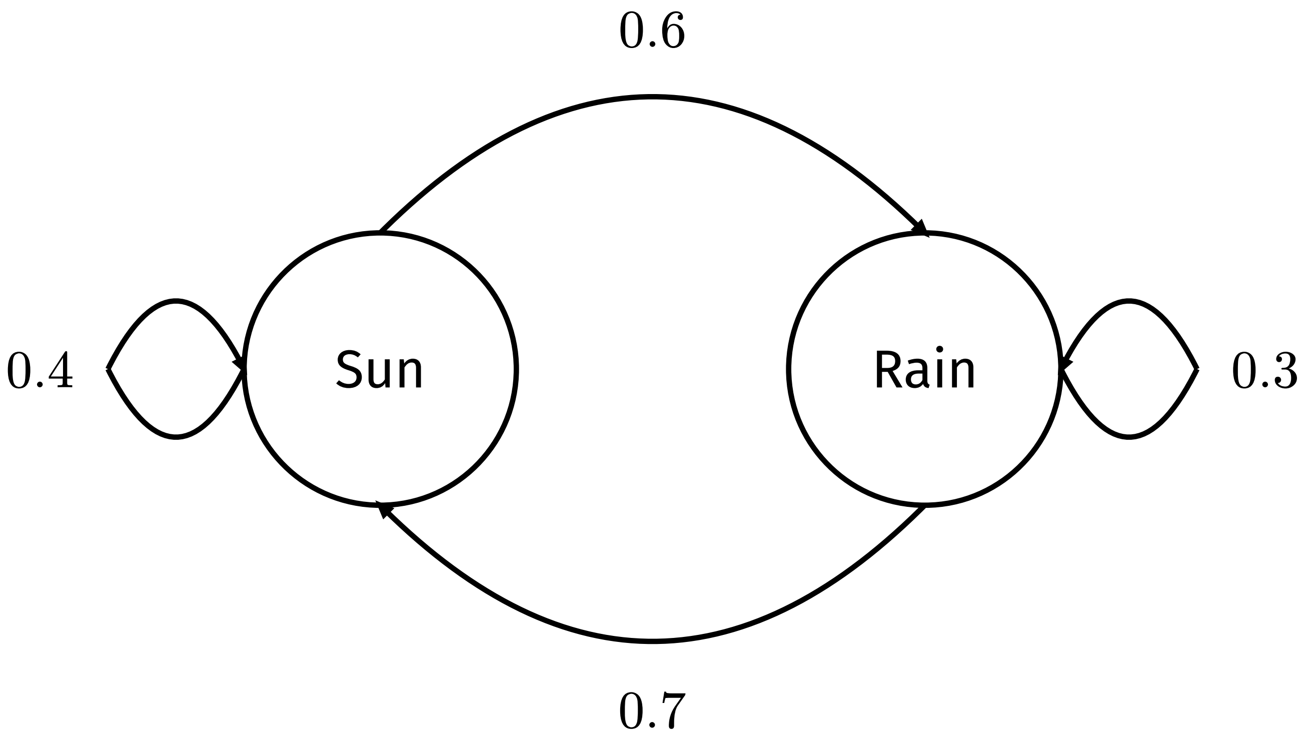
- Markov chains have a property that the probability distribution of the next state **depends only on the current state and not in the sequence of events that preceded**:

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

  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 $X_r, X_{2r}, X_{3r}, ...$ 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"[xlviii] 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** (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013).

[xlviii] some references call this "burnin".

MCMC Algorithms

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

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

[xlix] see the Wikipedia page for a full list.

[l] 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).

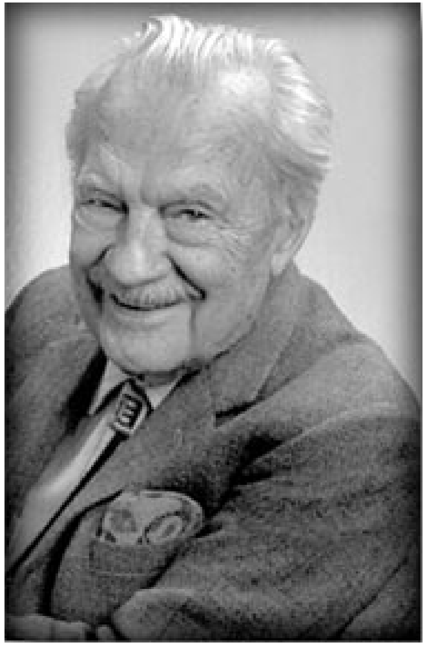
MCMC Algorithms – Hamiltonian Monte Carlo

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 $\mathcal{O}\left(d^{\frac{1}{4}}\right)$, 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:

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

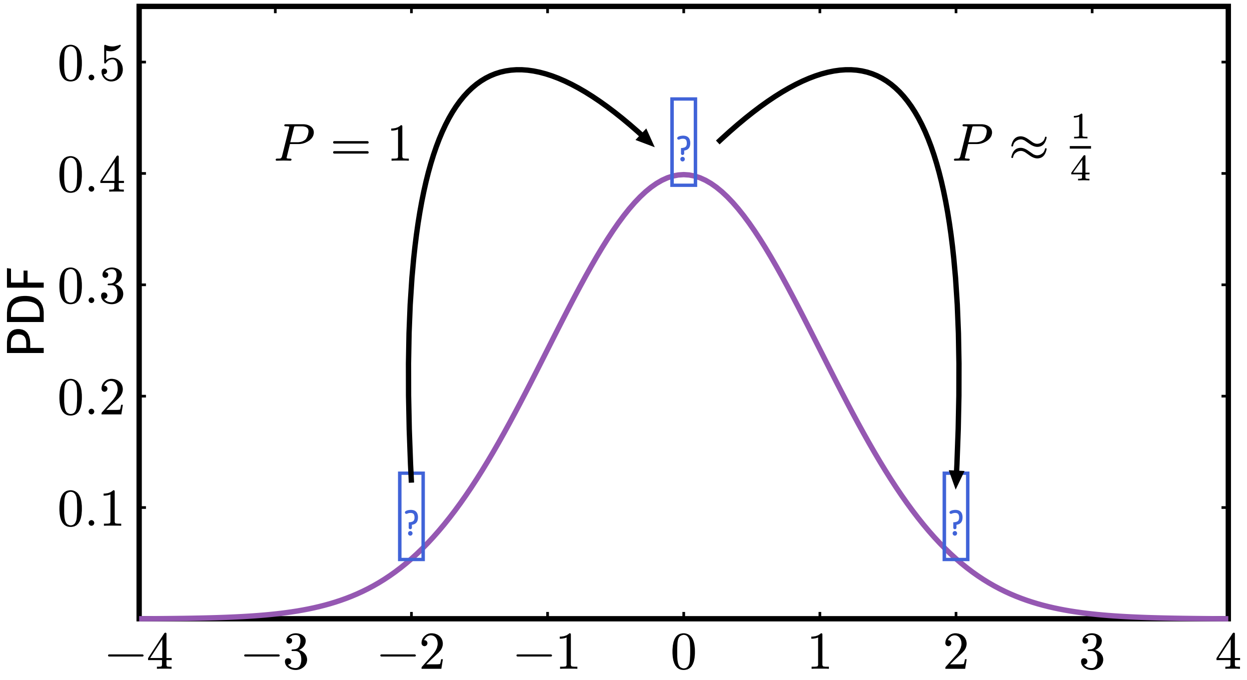
This means that the Markov chain will only change to a new state based in one of two conditions:

- 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

```
Define an initial set $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ that $P(\boldsymbol{\theta}^0 \mid \boldsymbol{y}) > 0$
for $t = 1, 2, \dots$
    Sample a proposal of $\boldsymbol{\theta}^*$ from a proposal distribution in time $t$, $J_t(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{t-1})$
    As an acceptance/rejection rule, compute the proportion of the probabilities:
        $r = \frac{P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{P(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{y})}$
    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

```
Define an initial set $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ that $P(\boldsymbol{\theta}^0 \mid \boldsymbol{y}) > 0$
for $t = 1, 2, \dots$
    Sample a proposal of $\boldsymbol{\theta}^*$ from a proposal distribution in time $t$, $J_t(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{t-1})$
    As an acceptance/rejection rule, compute the proportion of the probabilities:
        $$r = \frac{\frac{P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{J_t(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{t-1})}}{\frac{P(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{y})}{J_t(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{\theta}^*)}}$$
    Assign:
    $$\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^* & \text{with probability} \min(r, 1) \\ \boldsymbol{\theta}^{t-1} & \text{otherwise} \end{cases}$$
```

Metropolis-Hastings Animation

See Metropolis-Hastings in action at `chi-feng/mcmc-demo` .

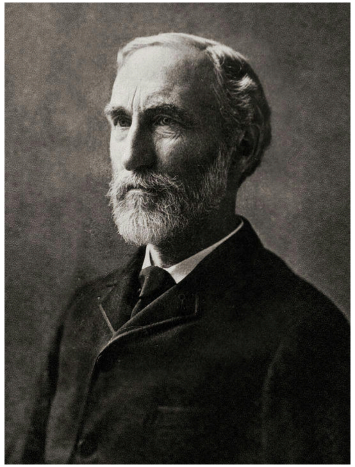
Limitations of the Metropolis Algorithms

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 (Roberts, Gelman and Gilks, 1997; Beskos *et al.*, 2013).
- 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 and 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:

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

Gibbs Algorithm

```
Define an initial set $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ that $P(\boldsymbol{\theta}^0 \mid \boldsymbol{y}) > 0$
for $t = 1, 2, \ldots$
    Assign:
    $\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^t_1 \sim P\left(\theta_1 \mid \theta^0_2, \ldots, \theta^0_p\right) \\ \boldsymbol{\theta}^t_2 \sim P\left(\theta_2 \mid \theta^{t-1}_1, \ldots, \theta^{t-1}_p\right) \\ \vdots \\ \boldsymbol{\theta}^t_p \sim P\left(\theta_p \mid \theta^{t-1}_1, \ldots, \theta^{t-1}_{p-1}\right) \end{cases}$
```

Gibbs Animation

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:

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

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 (Roberts, Gelman and Gilks, 1997; Beskos *et al.*, 2013).

History of HMC Algorithm

HMC was originally described in the physics literature[li] (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).

[li] 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 θ_j, HMC adds a momentum variable φ_j. The posterior density $P(\boldsymbol{\theta} \mid y)$ is incremented by an independent momenta distribution $P(\boldsymbol{\varphi})$, hence defining the following joint probability:

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

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:

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

Momenta Distribution – $P(\varphi)$

Generally we give φ a multivariate normal distribution with mean 0 and covariance M, a “mass matrix”.

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

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

HMC Algorithm

```
Define an initial set $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ that $P(\boldsymbol{\theta}^0 \mid \boldsymbol{y}) > 0$
Sample $\varphi$ from a $\text{Multivariate Normal}(\mathbf{0}, \boldsymbol{M})$
Simultaneously sample $\boldsymbol{\theta}^*$ **and** $\varphi$ with $L$ steps **and** step-size $\varepsilon$
Define the current value of $\boldsymbol{\theta}$ as the proposed value $\boldsymbol{\theta}^*$**:** $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
**for** $1, 2, ..., L$
    Use the $\log$ of the posterior's gradient $\boldsymbol{\theta}^*$ **to** produce a half-step of $\varphi$**:** $\varphi \leftarrow \varphi + \frac{1}{2}\varepsilon \frac{\mathrm{d} \log P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{\mathrm{d}\theta}$
    Use $\varphi$ **to** update $\boldsymbol{\theta}^*$**:** $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* + \varepsilon \boldsymbol{M}^{-1} \varphi$
    Use again $\boldsymbol{\theta}^*$ $\log$ gradient **to** produce a half-step of $\varphi$**:** $\varphi \leftarrow \varphi + \frac{1}{2}\varepsilon \frac{\mathrm{d} \log P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{\mathrm{d}\theta}$
As an acceptance/rejection rule, compute**:**
    $r = \frac{P(\boldsymbol{\theta}^* \mid \boldsymbol{y}) P(\varphi^*)}{P(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{y}) P(\varphi^{t-1})}$
Assign**:**
$\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^* & \text{with probability} \min(r, 1) \\ \boldsymbol{\theta}^{t-1} & \text{otherwise} \end{cases}$
```

HMC Animation

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 ε[lii]. 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 ε to compute a numerical solution of system in a future time t from specific initial conditions.

[lii] 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**.

This makes the Euler method a bad choice as a HMC's numerical integrator.

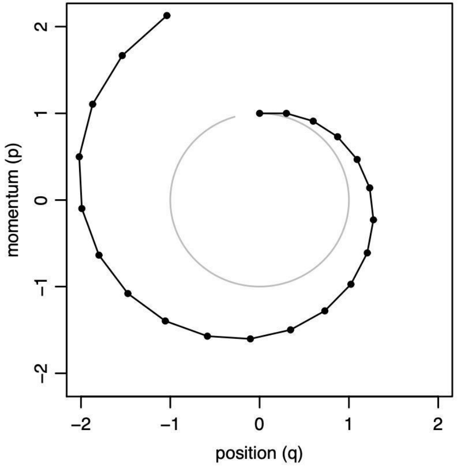


Figure 6: HMC numerically integrated using Euler with $\varepsilon = 0.3$ and $L = 20$

An Interlude into Numerical Integration[liii]

To preserve volume, we need a numerical **symplectic integrator**.

Symplectic integrators are at most second-order and demands a constant step-size ε.

One of the main numerical symplectic integrator used in Hamiltonian dynamics is the **Störmer–Verlet integrator**, also known as **leapfrog integrator**.

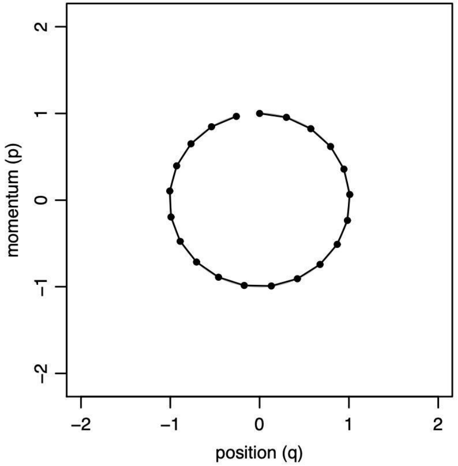


Figure 7: HMC numerically integrated using leapfrog with $\varepsilon = 0.3$ and $L = 20$

[liii] 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 ε,

More specific, the leapfrog integrator allows only a constant ε.

There is a delicate balance between L and ε, that are hyperparameters and need to be carefully adjusted.

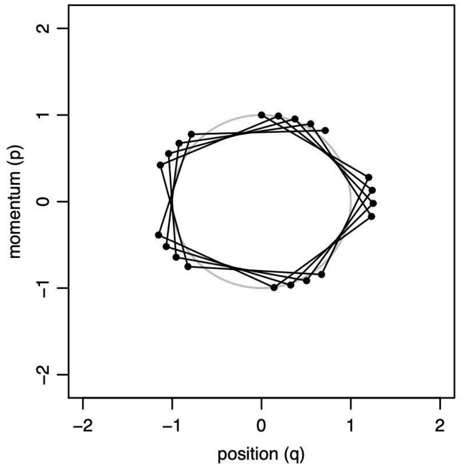


Figure 8: HMC numerically integrated using leapfrog with $\varepsilon = 1.2$ and $L = 20$

No-U-Turn-Sampler (NUTS)

In HMC, we can adjust ε 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 No-U-Turn-Sampler (NUTS) (Hoffman and Gelman, 2011) enters: you don’t need to **adjust anything**, just “press the button”.

It will automatically find ε and L.

No-U-Turn-Sampler (NUTS)

More specifically, we need a criterion that informs that we performed enough Hamiltonian dynamics simulation.

In other words, to simulate past beyond would not increase the distance between the proposal $\boldsymbol{\theta}^*$ and the current value $\boldsymbol{\theta}$.

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

$$(\boldsymbol{\theta}^* - \boldsymbol{\theta}) \cdot \boldsymbol{\varphi} = (\boldsymbol{\theta}^* - \boldsymbol{\theta}) \cdot \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\theta}^* - \boldsymbol{\theta}) = \frac{\mathrm{d}}{\mathrm{d}t} \frac{(\boldsymbol{\theta}^* - \boldsymbol{\theta}) \cdot (\boldsymbol{\theta}^* - \boldsymbol{\theta})}{2}$$

No-U-Turn-Sampler (NUTS)

This suggests an algorithms that does not allow proposals be guided infinitely until the distance between the proposal θ^* and the current θ is less than zero.

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

No-U-Turn-Sampler (NUTS)

NUTS uses the leapfrog integrator to create a binary tree where each leaf node is a proposal of the momenta vector φ 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.

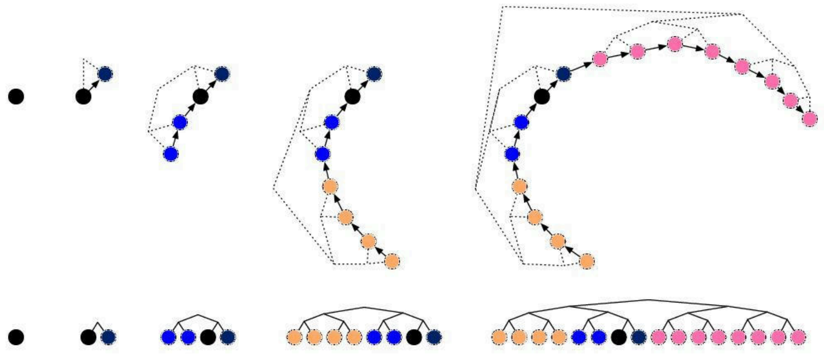


Figure 9: NUTS growing leaf nodes forward

No-U-Turn-Sampler (NUTS)

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

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

NUTS Algorithm

```
Define an initial set $\boldsymbol{\theta}^0 \in \mathbb{R}^p$ that $P(\boldsymbol{\theta}^0 \mid \boldsymbol{y}) > 0$
Instantiate an empty binary tree with $2^L$ leaf nodes
Sample $\boldsymbol{\varphi}$ from a $\text{Multivariate Normal}(\mathbf{0}, \boldsymbol{M})$
Simultaneously sample $\boldsymbol{\theta}^*$ and $\boldsymbol{\varphi}$ with $L$ steps and step-size $\varepsilon$
Define the current value of $\boldsymbol{\theta}$ as the proposed value $\boldsymbol{\theta}^*$: $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
for $1, 2, \ldots, L$
    Choose a direction $v \sim \text{Uniform}(\{-1, 1\})$
    Use the $\log$ of the posterior's gradient $\boldsymbol{\theta}^*$ to produce a half-step of $\boldsymbol{\varphi}$: $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \frac{1}{2}\varepsilon \frac{\mathrm{d} \log P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{\mathrm{d}\theta}$
    Use $\boldsymbol{\varphi}$ to update $\boldsymbol{\theta}^*$: $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* + \varepsilon \boldsymbol{M}^{-1} \boldsymbol{\varphi}$
    Use again $\boldsymbol{\theta}^*$ $\log$ gradient to produce a half-step of $\boldsymbol{\varphi}$: $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \frac{1}{2}\varepsilon \frac{\mathrm{d} \log P(\boldsymbol{\theta}^* \mid \boldsymbol{y})}{\mathrm{d}\theta}$
Define the node $L_t^v$ 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: $v \frac{\mathrm{d}}{\mathrm{d}t} \frac{(\boldsymbol{\theta}^* - \boldsymbol{\theta}^*) \cdot (\boldsymbol{\theta}^* - \boldsymbol{\theta}^*)}{2} < 0$ or $L$ steps have been reached
    Stop sampling $\boldsymbol{\theta}^*$ in the direction $v$ and continue sampling only in the direction $-v$
        The difference between proposal vector $\boldsymbol{\theta}^*$ and current vector $\boldsymbol{\theta}$ in the direction $-v$ is lower than zero: $-v \frac{\mathrm{d}}{\mathrm{d}t} \frac{(\boldsymbol{\theta}^* - \boldsymbol{\theta}^*) \cdot (\boldsymbol{\theta}^* - \boldsymbol{\theta}^*)}{2} < 0$ or $L$ steps have been reached
            Stop sampling $\boldsymbol{\theta}^*$
Choose a random node from the binary tree as the proposal
As an acceptance/rejection rule, compute:
    $r = \frac{P(\boldsymbol{\theta}^* \mid \boldsymbol{y}) P(\boldsymbol{\varphi}^*)}{P(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{y}) P(\boldsymbol{\varphi}^{t-1})}$
Assign:
$\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^* & \text{with probability } \min(r, 1) \\ \boldsymbol{\theta}^{t-1} & \text{otherwise} \end{cases}$
```

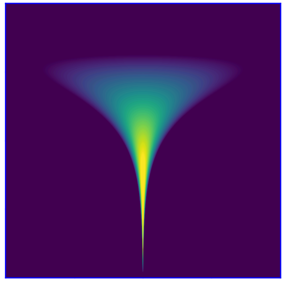
NUTS Animation

See NUTS in action at `chi-feng/mcmc-demo` .

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

The famous "Devil's Funnel"[liv].

Here we see that HMC and NUTS, during the exploration of the posterior, have to change often L and ε values[lv].



[liv] very common em hierarchical models.

[lv] remember that L and ε are defined in the warmup phase and kept fixed during sampling.

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:

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

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:

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

Non-Centered Parameterization – Varying-Intercept Model

This example is for linear regression:

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Normal}\left(\alpha_j + \boldsymbol{X} \cdot \boldsymbol{\beta}, \sigma\right) \\
\alpha_j &= z_j \cdot \tau + \alpha \\
z_j &\sim \text{Normal}(0, 1) \\
\alpha &\sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \\
\boldsymbol{\beta} &\sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}}) \\
\tau &\sim \text{Cauchy}^+(0, \psi_\alpha) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

Non-Centered Parameterization – Varying-(Intercept-)Slope Model

This example is for linear regression:

$$
\begin{aligned}
\boldsymbol{y} &\sim \text{Normal}\left(\boldsymbol{X}\boldsymbol{\beta}_j, \sigma\right) \\
\boldsymbol{\beta}_j &= \boldsymbol{\gamma}_j \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\gamma}_j \\
\boldsymbol{\gamma}_j &\sim \text{Multivariate Normal}(\boldsymbol{0}, \boldsymbol{I}) \;\text{ for } j \in \{1, ..., J\} \\
\boldsymbol{\Sigma} &\sim \text{LKJ}(\eta) \\
\sigma &\sim \text{Exponential}(\lambda_\sigma)
\end{aligned}
$$

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

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 ε during warmup.

It has the following default NUTS hyperparameters' values[lvi]:

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

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

Turing and NUTS

Turing 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 ε during warmup.

It has the same default NUTS hyperparameters' values[lvii]:

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

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

Markov Chain Convergence

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

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.

[lviii] 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.
- $\hat{R}$ (**Rhat**): potential scale reduction factor, a metric to measure if the Markov chain have mixed, and, potentially, converged.

Convergence Metrics – Effective Sample Size (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013)

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

where:

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

Convergence Metrics – Rhat (Andrew Gelman, John B. Carlin, Stern, *et al.*, 2013)

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

where $\widehat{\operatorname{var}}^+(\psi \mid y)$ is the Markov chains' sample variance for a certain parameter ψ.

We calculate it by using a weighted sum of the within-chain W and between-chain B variances:

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

Intuitively, the value is 1.0 if all chains are totally convergent.

As a heuristic, if $\hat{R} > 1.1$, you need to worry because probably the chains have not converged adequate.

Traceplot – Convergent Markov Chains

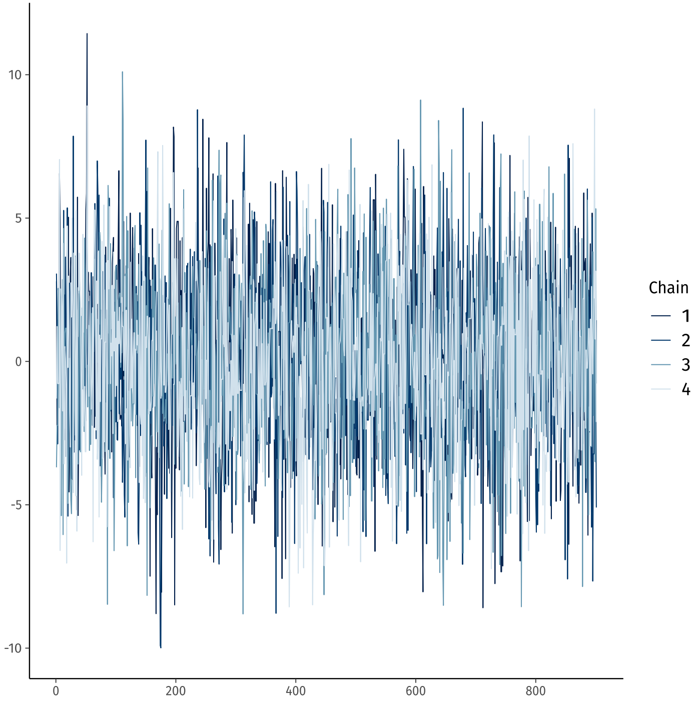


Figure 10: A convergent Markov chains traceplot

Traceplot – Divergent Markov Chains

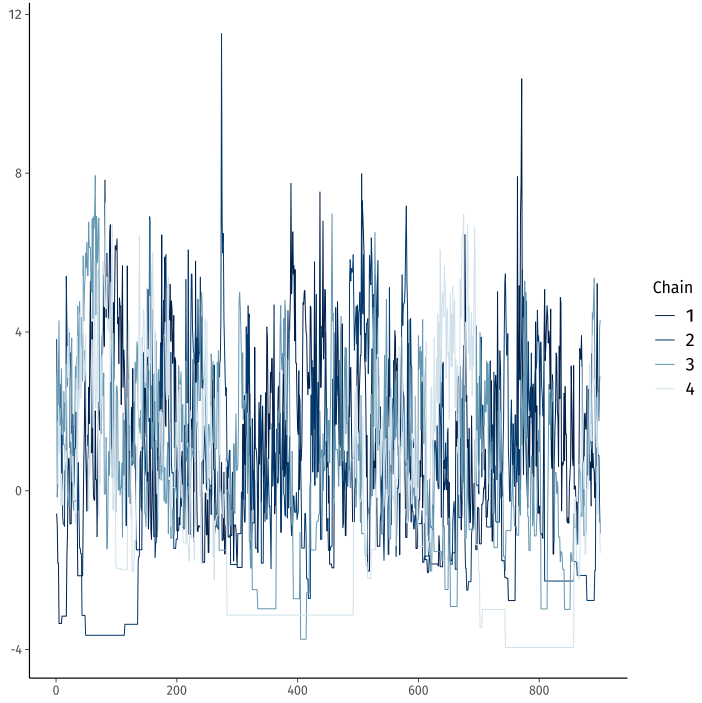


Figure 11: A divergent Markov chains traceplot

Stan’s Warning Messages[lix]

```
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
```

[lix] also see Stan’s warnings guide.

Turing’s Warning Messages

Turing 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'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)

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[lx].

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).

[lx] 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**[lxi]. The maximum value is 1.0 (not recommended). Then, any value between 0.8 and 1.0 is more conservative.
3. **Model reparameterization**: there are two approaches. Centered parameterization (CP) and non-centered parameterization (NCP).

[lxi] Stan's default is 0.8 and Turing'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

- Andrew Gelman, John B. Carlin, Stern, *et al.* (2013) - 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, Gelman and Gabry (2015)
- Spiegelhalter *et al.* (2002)
- Van Der Linde (2005)
- Watanabe and Opper (2010)
- Gelfand (1996)
- Watanabe and Opper (2010)
- Geisser and Eddy (1979)

THEN I JUST PICKED THE
MODEL WITH THE LOWEST AIC
imgflip.com

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 and Eddy, 1979).

But What About Visual Posterior Predictive Checks?

To analyze and compare models using visual posterior predictive checks 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 *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, Gelman and Gabry, 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 and Opper, 2010), full-Bayesian, in the sense that uses the full posterior distribution density, and it is asymptotically equal to LOO (Vehtari, Gelman and Gabry, 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:

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

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

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

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

[lxii] historical reasons.

Historical Interlude – AIC (Akaike, 1973)

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

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

AIC is an approximation and can only be reliable when:

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

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

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

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

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[lxiii].

We can discuss **predictive accuracy objective metrics**

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

[lxiii] AIC, DIC etc.

Predictive Accuracy

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

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

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:

$$\mathrm{elpd} = \sum_{i=1}^N \int p_{t(\tilde{y}_i)} \log p(\tilde{y}_i \mid y) \, \mathrm{d}\tilde{y}$$

where elpd is the expected log pointwise predictive density, and $p_{t(\tilde{y}_i)}$ is the distribution that represents the $\tilde{y}_i$'s true underlying data generating process.

The $p_{t(\tilde{y}_i)}$ are unknown and we generally use cross-validation or approximations to estimate elpd.

Leave-One-Out Cross-Validation (LOO)

We can compute the elpd using LOO (Vehtari, Gelman and Gabry, 2015):

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

where

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

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

[lxiv] upcoming...

Widely Applicable Information Criteria (WAIC)

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

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

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

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

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

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

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

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

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**[lxv], which means a importance weighting scheme approach.

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

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

Importance Sampling

If the N samples are conditionally independent[lxvi] (Gelfand, Dey and Chang, 1992), we can compute LOO with $\boldsymbol{\theta}^s$ 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):

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

[lxvi] 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:

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

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, Gelman and Gabry, 2015).

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

Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

Finally, we have PSIS-LOO:

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

where w is the truncated weights.

Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

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

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

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

Backup Slides

How the Normal distribution arose[lxvii]

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

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

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

We know that in the binomial: $E = np$ and $\operatorname{Var} = npq$; hence replacing E by μ and Var by σ^2:

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

[lxvii] Origins can be traced back to Abraham de Moivre in 1738. A better explanation can be found by clicking here.

QR Decomposition

In Linear Algebra 101, we learn that any matrix (even non-square ones) can be decomposed into a product of two matrices:

- $\boldsymbol{Q}$: an orthogonal matrix (its columns are orthogonal unit vectors, i.e. $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$)
- $\boldsymbol{R}$: an upper-triangular matrix

Now, we incorporate the QR decomposition into the linear regression model. Here, I am going to use the “thin” QR instead of the “fat”, which scales $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices by a factor of $\sqrt{n-1}$ where n is the number of rows in $\boldsymbol{X}$. In practice, it is better to implement the thin QR, than the fat QR decomposition. It is more numerical stable. Mathematically speaking, the thing QR decomposition is:

$$\begin{aligned}
\boldsymbol{X} &= \boldsymbol{Q}^* \boldsymbol{R}^* \\
\boldsymbol{Q}^* &= \boldsymbol{Q} \cdot \sqrt{n-1} \\
\boldsymbol{R}^* &= \frac{1}{\sqrt{n-1}} \cdot \boldsymbol{R} \\
\boldsymbol{\mu} &= \alpha + \boldsymbol{X} \cdot \boldsymbol{\beta} + \sigma \\
&= \alpha + \boldsymbol{Q}^* \cdot \boldsymbol{R}^* \cdot \boldsymbol{\beta} + \sigma \\
&= \alpha + \boldsymbol{Q}^* \cdot (\boldsymbol{R}^* \cdot \boldsymbol{\beta}) + \sigma \\
&= \alpha + \boldsymbol{Q}^* \cdot \tilde{\boldsymbol{\beta}} + \sigma
\end{aligned}$$

Bibliography

Akaike, H. (1973) “Information theory and an extension of the maximum likelihood principle,” *Second International Symposium on Information Theory*. Edited by B. N. Petrov and F. Csaki

Amrhein, V., Greenland, S. and McShane, B. (2019) “Scientists Rise up against Statistical Significance,” *Nature*, 567(7748), pp. 305–307. Available at: https://doi.org/10.1038/d41586-019-00857-9

Bates, D. *et al.* (2022) *JuliaStats/MixedModels.jl*. Zenodo. Available at: https://doi.org/10.5281/ZENODO.6925652

Bates, D. *et al.* (2015) “Fitting Linear Mixed-Effects Models Using lme4,” *Journal of Statistical Software*, 67(1), pp. 1–48. Available at: https://doi.org/10.18637/jss.v067.i01

Benjamin, D. J. *et al.* (2018) “Redefine Statistical Significance,” *Nature Human Behaviour*, 2(1), pp. 6–10. Available at: https://doi.org/10.1038/s41562-017-0189-z

Bertsekas, D. P. and Tsitsiklis, J. N. (2008) *Introduction to Probability, 2nd Edition*. 2nd edition. Belmont, Massachusetts: Athena Scientific

Beskos, A. *et al.* (2013) “Optimal Tuning of the Hybrid Monte Carlo Algorithm,” *Bernoulli*, 19(5A), pp. 1501–1534. Available at: https://doi.org/10.3150/12-BEJ414

Betancourt, M. (2017) *A Conceptual Introduction to Hamiltonian Monte Carlo*. Available at: http://arxiv.org/abs/1701.02434 (Accessed: November 6, 2019)

Betancourt, M. (2019) *Probabilistic Building Blocks*. Available at: https://betanalpha.github.io/assets/case_studies/probability_densities.html (Accessed: May 27, 2021)

Betancourt, M. (2021) *Sparsity Blues*. Available at: https://betanalpha.github.io/assets/case_studies/modeling_sparsity.html (Accessed: December 9, 2023)

Bhadra, A. *et al.* (2015) “The Horseshoe+ Estimator of Ultra-Sparse Signals”

Blei, D. M. (2014) “Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models,” *Annual Review of Statistics and Its Application*, 1(1), pp. 203–232. Available at: https://doi.org/10.1146/annurev-statistics-022513-115657

Box, G. E. P. (1976) “Science and Statistics,” *Journal of the American Statistical Association*, 71(356), pp. 791–799. Available at: https://doi.org/10.2307/2286841

Brooks, S. *et al.* (2011) *Handbook of Markov Chain Monte Carlo*. CRC Press

Bürkner, P.-C. and Vuorre, M. (2019) “Ordinal Regression Models in Psychology: A Tutorial,” *Advances in Methods and Practices in Psychological Science*, 2(1), p. 77–78. Available at: https://doi.org/10.1177/2515245918823199

Carpenter, B. *et al.* (2017) “Stan : A Probabilistic Programming Language,” *Journal of Statistical Software*, 76(1). Available at: https://doi.org/10.18637/jss.v076.i01

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2009) “Handling sparsity via the horseshoe,” in *Artificial intelligence and statistics*, pp. 73–80

Casella, G. and George, E. I. (1992) “Explaining the Gibbs Sampler,” *The American Statistician*, 46(3), pp. 167–174. Available at: https://doi.org/10.1080/00031305.1992.10475878

Chib, S. and Greenberg, E. (1995) “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, 49(4), pp. 327–335. Available at: https://doi.org/10.1080/00031305.1995.10476177

Dekking, F. M. *et al.* (2010) *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer

Diaconis, P. and Skyrms, B. (2019) *Ten Great Ideas about Chance*. Princeton University Press

Duane, S. *et al.* (1987) “Hybrid Monte Carlo,” *Physics Letters B*, 195(2), pp. 216–222. Available at: 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), pp. 131–136

Finetti, B. de (1974) *Theory of Probability*. Volume1 ed. New York: John Wiley & Sons

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), pp. 118–124

Ge, H., Xu, K. and Ghahramani, Z. (2018) “Turing: A Language for Flexible Probabilistic Inference,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1682–1690

Geisser, S. and Eddy, W. F. (1979) “A predictive approach to model selection,” *Journal of the American Statistical Association*, 74(365), pp. 153–160

Gelfand, A. E., Dey, D. K. and Chang, H. (1992) “Model determination using predictive distributions with implementation via sampling-based methods,” *Bayesian Statistics*. Edited by J. M. Bernardo et al. Oxford University Press

Gelfand, A. E. (1996) “Model determination using sampling-based methods,” *Markov chain Monte Carlo in practice*, pp. 145–161

Gelman, A. (1992) “Iterative and Non-Iterative Simulation Algorithms,” in *Computing Science and Statistics (Interface Proceedings)*. PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS, pp. 457–511

Gelman, A. (2008) *The Folk Theorem of Statistical Computing*. Available at: https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/

Gelman, A. and Hill, J. (2007) *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge university press

Gelman, Andrew, Carlin, John B., Stern, *et al.* (2013) “Basics of Markov Chain Simulation,” *Bayesian Data Analysis*. Chapman and Hall/CRC

Gelman, Andrew, Carlin, John B., Stern, *et al.* (2013) *Bayesian Data Analysis*. Chapman and Hall/CRC

Gelman, A., Hill, J. and Vehtari, A. (2020) *Regression and Other Stories*. Cambridge University Press

Gelman, A. *et al.* (2020) *Bayesian Workflow*. Available at: http://arxiv.org/abs/2011.01808 (Accessed: February 4, 2021)

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

Goodman, S. N. (2016) “Aligning Statistical and Scientific Reasoning,” *Science*, 352(6290), pp. 1180–1181. Available at: https://doi.org/10.1126/science.aaf5406

Grimmett, G. and Stirzaker, D. (2020) *Probability and Random Processes: Fourth Edition*. Fourth Edition, New to this Edition:. Oxford, New York: Oxford University Press

Hastings, W. K. (1970) “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, 57(1), pp. 97–109. Available at: https://doi.org/10.1093/biomet/57.1.97

Head, M. L. *et al.* (2015) “The extent and consequences of p-hacking in science,” *PLoS Biol*, 13(3), p. e1002106

Hoffman, M. D. and Gelman, A. (2011) “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo,” *Journal of Machine Learning Research*, 15(1), pp. 1593–1623. Available at: http://arxiv.org/abs/1111.4246

Ioannidis, J. P. A. (2019) "What Have We (Not) Learnt from Millions of Scientific Papers with <i>P</i> Values?," *The American Statistician*, 73(sup1), pp. 20–25. Available at: https://doi.org/10.1080/00031305.2018.1447512

Iserles, A. (2008) *A First Course in the Numerical Analysis of Differential Equations*. 2nd ed. USA: Cambridge University Press

Jaynes, E. T. (2003) *Probability Theory: The Logic of Science*. Cambridge university press

Khan, M. E. and Rue, H. (2021) *The Bayesian Learning Rule*. Available at: http://arxiv.org/abs/2107.04562 (Accessed: July 13, 2021)

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

Kruschke, J. K. and Vanpaemel, W. (2015) “Bayesian Estimation in Hierarchical Models,” *The Oxford Handbook of Computational and Mathematical Psychology*. Edited by J. R. Busemeyer et al. Oxford University Press Oxford, UK

Kurt, W. (2019) *Bayesian Statistics the Fun Way: Understanding Statistics and Probability with Star Wars, LEGO, and Rubber Ducks*. Illustrated edition. San Francisco: No Starch Press

Lakens, D. *et al.* (2018) “Justify Your Alpha,” *Nature Human Behaviour*, 2(3), pp. 168–171. Available at: https://doi.org/10.1038/s41562-018-0311-x

Lewandowski, D., Kurowicka, D. and Joe, H. (2009) “Generating random correlation matrices based on vines and extended onion method,” *Journal of multivariate analysis*, 100(9), pp. 1989–2001

Van Der Linde, A. (2005) “DIC in variable selection,” *Statistica Neerlandica*, 59(1), pp. 45–56

McElreath, R. (2020) *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC press

Metropolis, N. *et al.* (1953) “Equation of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, 21(6), pp. 1087–1092. Available at: 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), pp. 194–203. Available at: https://doi.org/10.1006/jcph.1994.1054

Neal, R. M. (2003) “Slice Sampling,” *The Annals of Statistics*, 31(3), pp. 705–741

Neal, R. M. (2011) “MCMC Using Hamiltonian Dynamics,” *Handbook of Markov Chain Monte Carlo*. Edited by S. Brooks et al.

Nesterov, Y. (2009) “Primal-dual subgradient methods for convex problems,” *Mathematical programming*, 120(1), pp. 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), pp. 333–380

Piironen, J. and Vehtari, A. (2017) “Sparsity information and regularization in the horseshoe and other shrinkage priors,” *Electronic Journal of Statistics*, 11(2), pp. 5018–5051. Available at: https://doi.org/10.1214/17-EJS1337SI

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997) “Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms,” *Annals of Applied Probability*, 7(1), pp. 110–120. Available at: https://doi.org/10.1214/aoap/1034625254

Rosnow, R. L. and Rosenthal, R. (1989) “Statistical procedures and the justification of knowledge in psychological science,” *American Psychologist*, 44, pp. 1276–1284

Salvatier, J., Wiecki, T. V. and Fonnesbeck, C. (2016) “Probabilistic programming in Python using PyMC3,” *PeerJ Computer Science*, 2, p. e55

Schoot, R. van de *et al.* (2021) “Bayesian Statistics and Modelling,” *Nature Reviews Methods Primers*, 1(1), pp. 1–26. Available at: https://doi.org/10.1038/s43586-020-00001-2

Semenova, E. (2019) *Ordered Logistic regression and Probabilistic Programming:with examples in Stan, PyMC3 and Turing*. Available at: https://medium.com/@liza_p_semenova/ordered-logistic-regression-and-probabilistic-programming-502d8235ad3f

Spiegelhalter, D. J. *et al.* (2002) “Bayesian measures of model complexity and fit,” *Journal of the royal statistical society: Series b (statistical methodology)*, 64(4), pp. 583–639

Storopoli, J. (2021) *Bayesian Statistics with Julia and Turing*. Available at: https://storopoli.io/Bayesian-Julia

Tibshirani, R. (1996) “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), pp. 267–288

Vehtari, A., Gelman, A. and Gabry, J. (2015) *Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC*. Available at: https://doi.org/10.1007/s11222-016-9696-4

Wasserstein, R. L. and Lazar, N. A. (2016) “The ASA's Statement on p-Values: Context, Process, and Purpose,” *American Statistician*, 70(2), pp. 129–133. Available at: https://doi.org/10.1080/00031305.2016.1154108

Wasserstein, R. L., Schirm, A. L. and Lazar, N. A. (2019) “Moving to a World Beyond “p $<$ 0.05”, *American Statistician*, 73(sup1), pp. 1–19. Available at: https://doi.org/10.1080/00031305.2019.1583913

Watanabe, S. and 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)

Zhang, Y. D. *et al.* (2022) “Bayesian regression using a prior on the model fit: The r2-d2 shrinkage prior,” *Journal of the American Statistical Association*, 117(538), pp. 862–874

Zou, H. and Hastie, T. (2005) “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2), pp. 301–320

“It’s Time to Talk about Ditching Statistical Significance” (2019) *Nature*, 567(7748), p. 283–284. Available at: https://doi.org/10.1038/d41586-019-00874-8