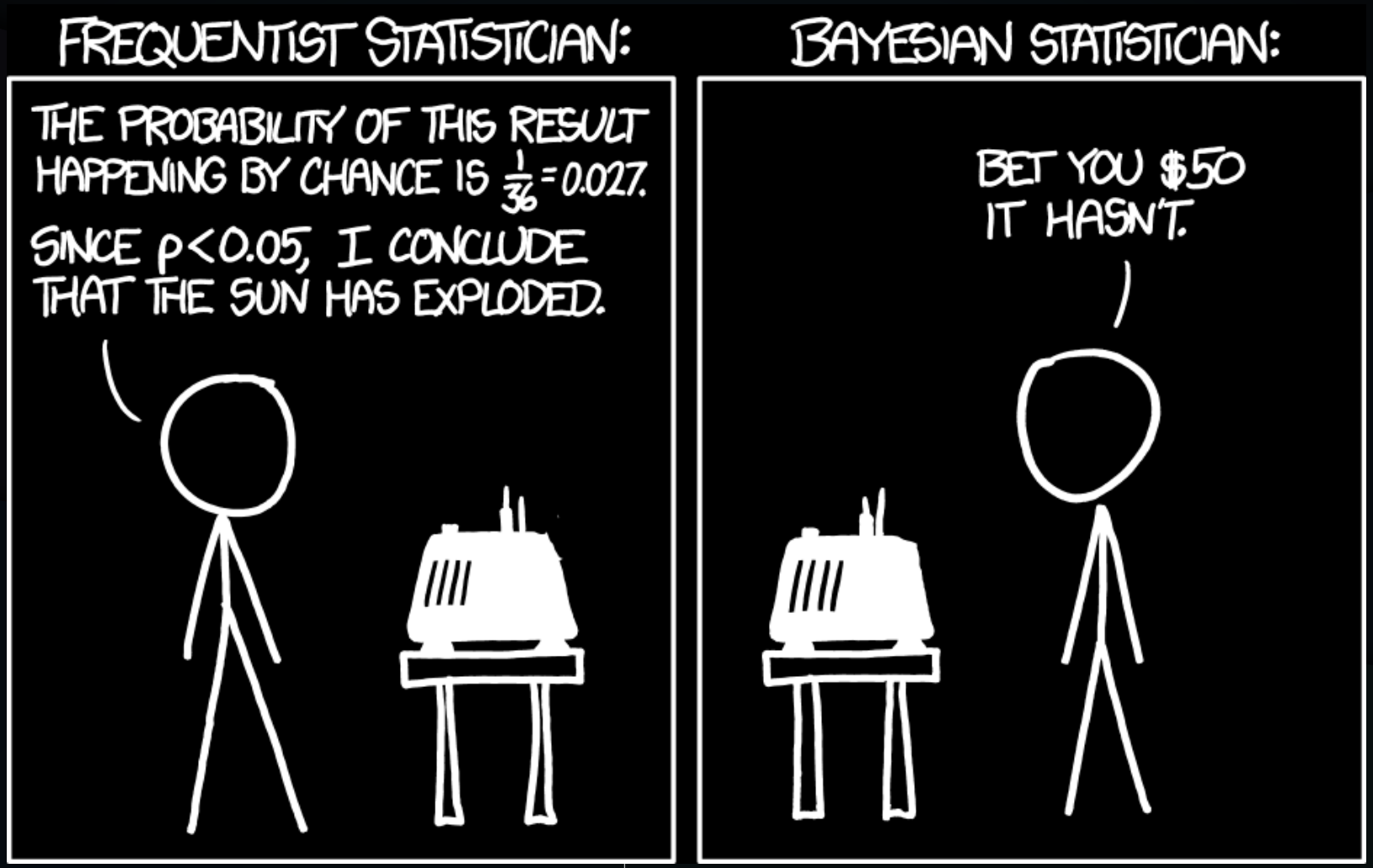
R-Ladies at Domino Data Lab
17th July 2019

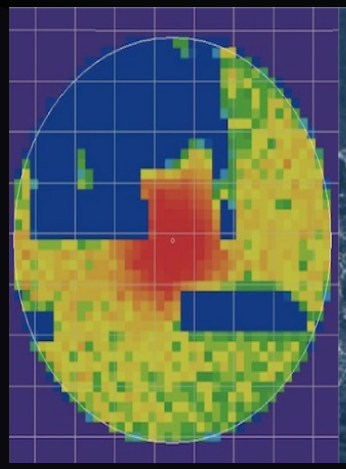
Bayesian Analysis in R via Stan

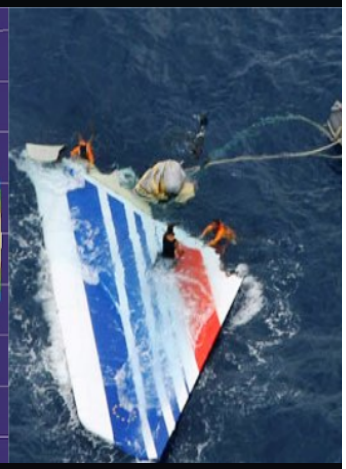
Alice Milivinti a.lice.milivinti@gmail.com

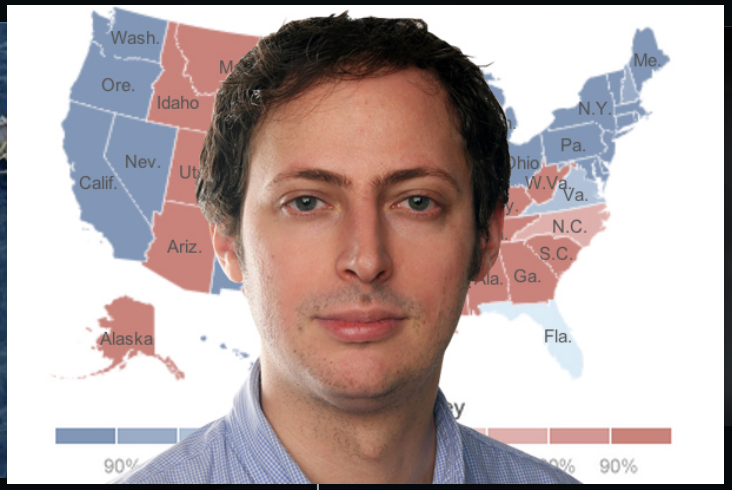
We have a problem…

How to solve it?









- Large uncertainty to quantify
- Integrate variegate information
- They all use Bayesian Methods :)

A New Trend?

Probabilistic Programming

Graphical Models

Bayes Theorem

$$P(\text{blonde} \mid \text{eyes}) = \frac{P(\text{eyes} \mid \text{blonde})\,P(\text{blonde})}{P(\text{eyes})} = \frac{0.5 * 0.5}{0.4} = 0.625$$

	blonde	brunette	Marginal P(Eyes)
brown eyes	P(blonde, brown eyes)=0.25	P(brunette, brown eyes)=0.35	P(brown eyes)=0.6
blue eyes	P(blonde, blue eyes)=0.25	P(brunette, blue eyes)=0.15	P(blue eyes)=0.4
Marginal P(Hair)	P(blonde) = 0.50	P(brunette) = 0.50	1

Let's look only at blonde hairs!

	blue eyes	brown eyes	
P(Eyes | blonde)	0.25/0.5 = 0.5	0.25/0.5=0.5	0.5/0.5=1

Since…

$$P(\text{👁} \mid \text{👱‍♀️}) * P(\text{👱‍♀️}) = P(\text{👱‍♀️}, \text{👁})$$

Re-express Bayes Theorem as:

$$P(\text{👱‍♀️} \mid \text{👁}) = \frac{P(\text{👱‍♀️}, \text{👁})}{P(\text{👁})} = \frac{P(\text{👁} \mid \text{👱‍♀️})\, P(\text{👱‍♀️})}{P(\text{👁})}$$

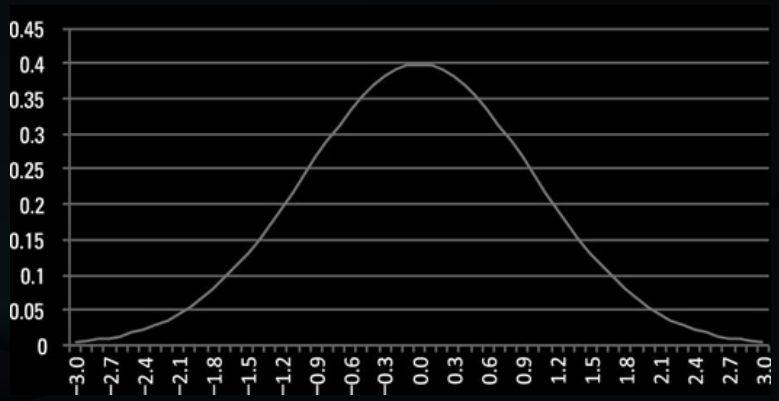
Bayes Theorem

$$P(\text{blonde} \mid \text{blue eye}) = \frac{P(\text{blonde}, \text{blue eye})}{P(\text{blue eye})} = \frac{0.25}{0.4} = 0.625$$

			Marginal P(Eyes)
	P( , )=0.25	P( , )=0.35	P()=0.6
	P( , )=0.25	P( , )=0.15	P()=0.4
Marginal P(Hair)	P() = 0.50	P( ) = 0.50	1

Bayes Theorem with continuous variables

$$f(A|B) = \frac{f(B|A)f(A)}{f(B)}$$



$$f(A|B) \propto f(B|A)f(A)$$

$$f(A|B) \propto f(B,A)$$

What Bayesian is not…..

The term “Bayesian” DOES NOT define a specific set of models, but rather the estimation technique.

Frequentism VS Bayesianism

Frequentist Inference

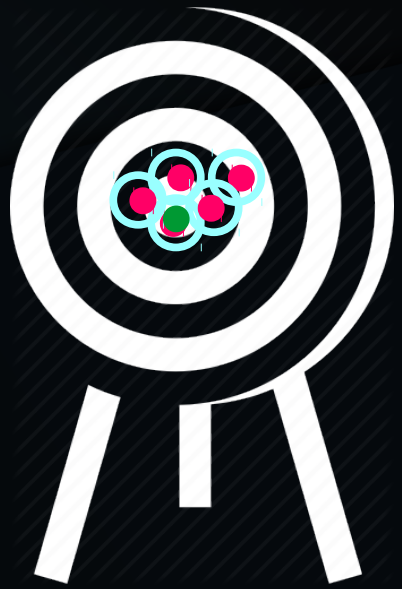
An archer hits within 5cm of the bullseye 95% of time.

- Observed data
- Confidence Interval
- Inferred Bullseye

If I draw 5cm radius circles around the arrows they would contain the bullseye 95% of times.

Reject whenever the 5cm circles does not overlap our location 95% of the time

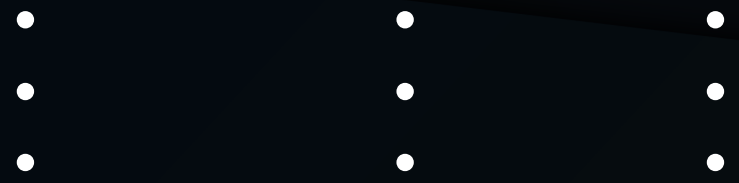
Target Practice



Frequentist Inference = the frequencies of repeated events.

Day 1

Day 2



The interval traps the truth in 95% of experiments.

To define anything frequentist, you have to imagine REPEATED experiments.

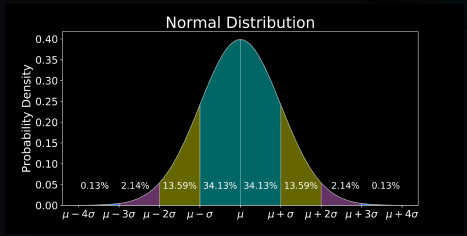
Frequentist require you to think about MANY OTHER DATASETS, not just the one you have to analyze.

Recipe to get θ ?

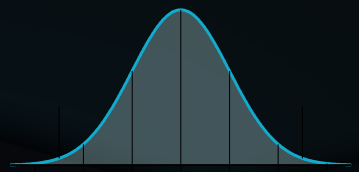
The Maximum Likelihood Function

1. Collect Data

True Population N → ∞



Sample Population N = n



Y=[1,0,0,1,...]

x=[2,3,4,…]

2. Write a model for the Data Generating Process

$$Y = \alpha + \beta x \longrightarrow Y \sim D(\underbrace{\alpha + \beta X}_{\mu}, \sigma_y^2)$$

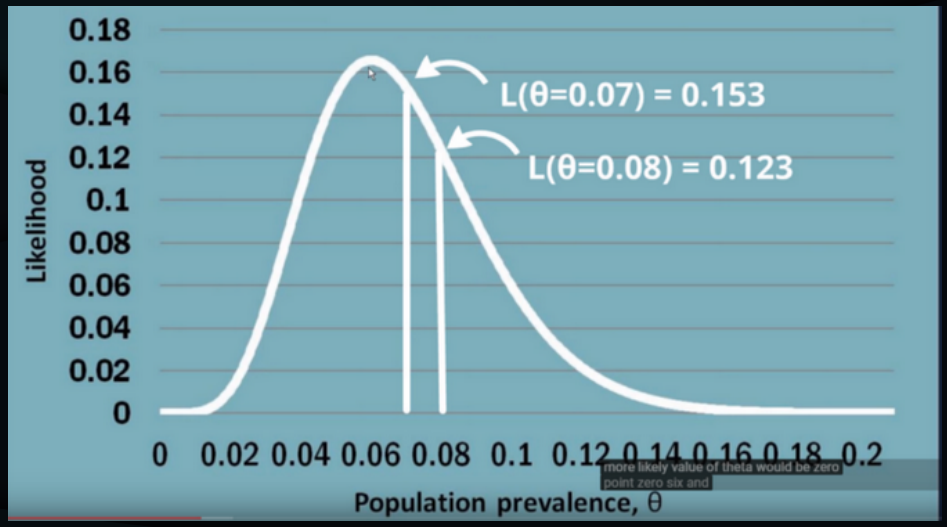
3. Find the values of $\theta = [\alpha, \beta, \sigma_y^2]$ which are THE MOST LIKELY to describe Y according to our model $Y = \alpha + \beta x$

The Maximum Likelihood Function

$$L(\theta, Y) = P(Data|Parameters(\theta))$$

The likelihood describes the extent to which the sample (Y) supports any parameter value θ.

Higher support = higher value for the likelihood of any θ.



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

Frequentist Inference $L(\theta, Y) = P(Data|Parameters(\theta))$

$$L = \underbrace{P(Y, X|\overbrace{\alpha, \beta, \sigma_y^2}^{\theta})}_{P(Data|\theta)}$$

$$\log(L) = \underset{\alpha, \beta, \sigma_y^2}{\operatorname{argmax}} \sum \ln(P(Y, X|\alpha, \beta, \sigma_y^2))$$

We want the values of $\hat{\alpha}, \hat{\beta}, \hat{\sigma}_y$ which MAXIMIZE THE LIKELIHOOD of our model

Frequentist Inference

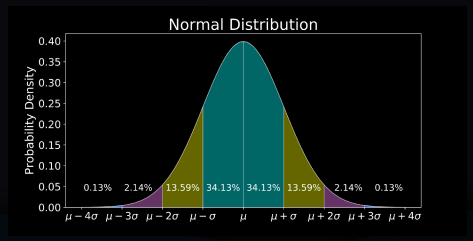
$$L = P(Data | Parameters(\theta))$$

$$Y \sim D(\hat{\alpha} + \hat{\beta} X, \hat{\sigma}_y^2)$$

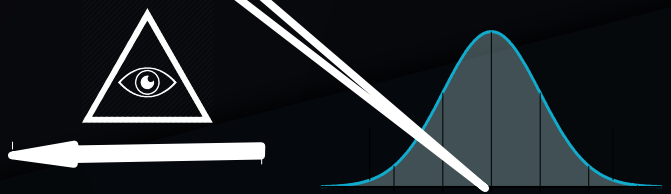
$\hat{\mu}$

UNIQUE "true" values for the unobserved population!

True Population N → ∞



Predicted Population N = n



Inference

$\hat{\mu}$

Central Limit Theorem

Significance?

$Group\ \sigma$

$H_0 : \hat{\theta} = 0$

F - Statistic

Bayesian Inference $f(\theta|Data) \propto f(Data|\theta) f(\theta)$

An archer hits within 5cm of the bullseye 95% of time.

- Prior bullseye location
- Observed shoots
- Posterior bullseye location

How likely that data point is, under ALL the TRUE possible bullseye locations?

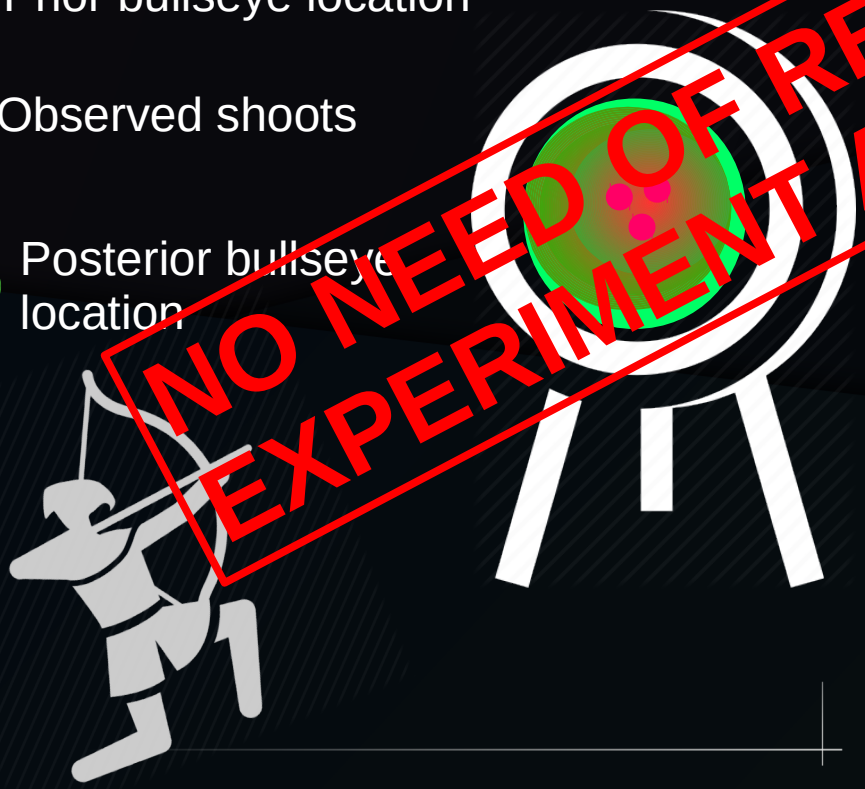
Bayes theorem tells us to update our prior beliefs about the bullseye location which are now proportional to the prior times the likelihood

$$\underbrace{P(\bullet|\bullet)}_{Posterior} \propto \underbrace{P(\bullet|\bullet)}_{Likelihood} \underbrace{P(\bullet)}_{Prior}$$

Bayesian Inference $f(\theta|Data) \propto f(Data|\theta) f(\theta)$

An archer hits within 5cm of the bullseye 95% of time.

- Prior bullseye location
- Observed shoots
- Posterior bullseye location



NO NEED OF REPEATED EXPERIMENT ASSUMPTIONS!

How likely that data point is, under ALL the TRUE possible bullseye locations?

Bayes theorem tells us to update our prior beliefs about the bullseye location which are now proportional to the prior times the likelihood

$$\underbrace{P(\bullet|\bullet)}_{Posterior} \propto \underbrace{P(\bullet|\bullet)}_{Likelihood} \underbrace{P(\bullet)}_{Prior}$$

The Prior Distribution $P(\theta)$

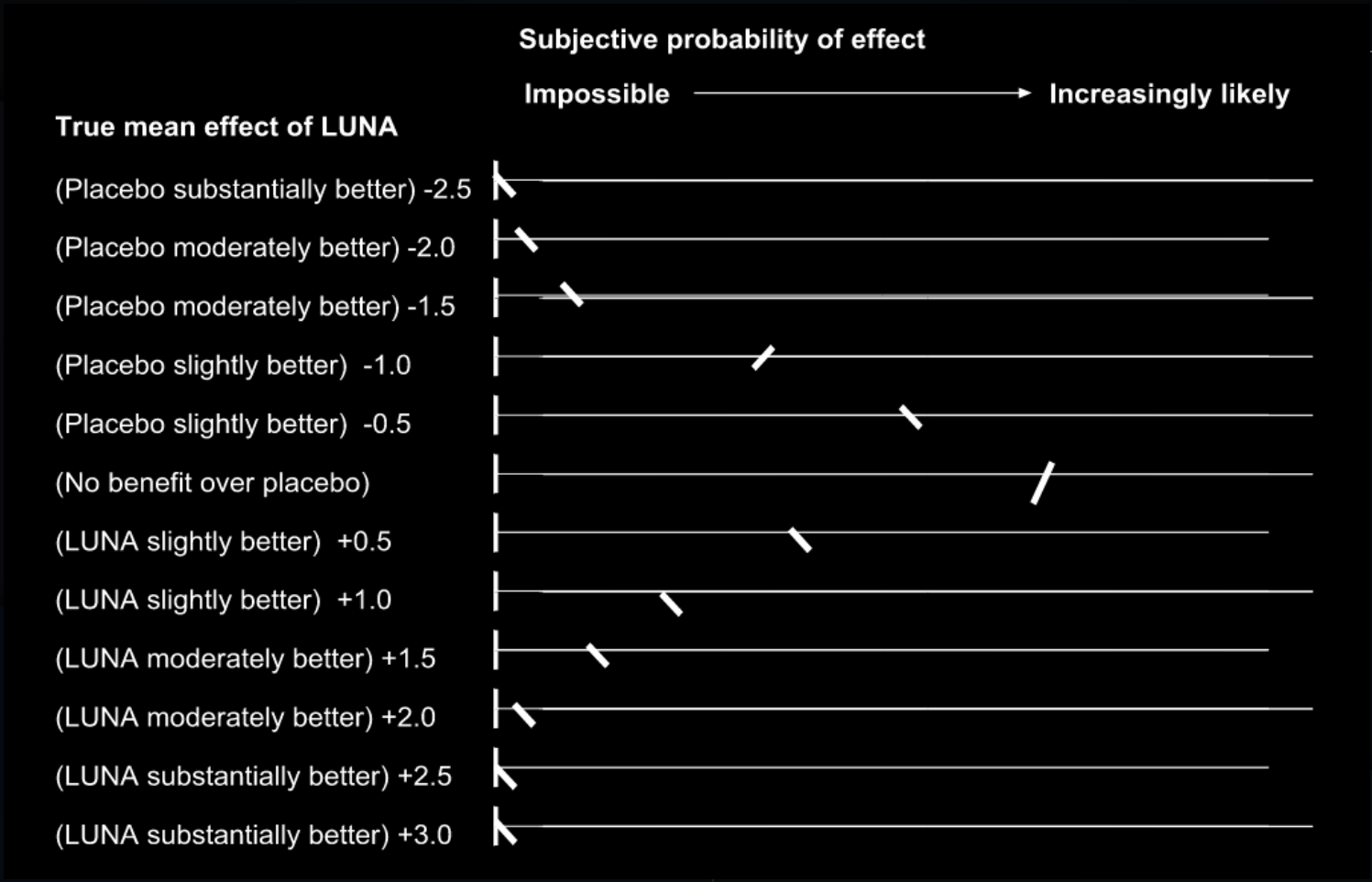
What you know about θ, excluding the information in the data → P(θ).

But where do priors come from?

External Data — Experts Opinions

Conversion?

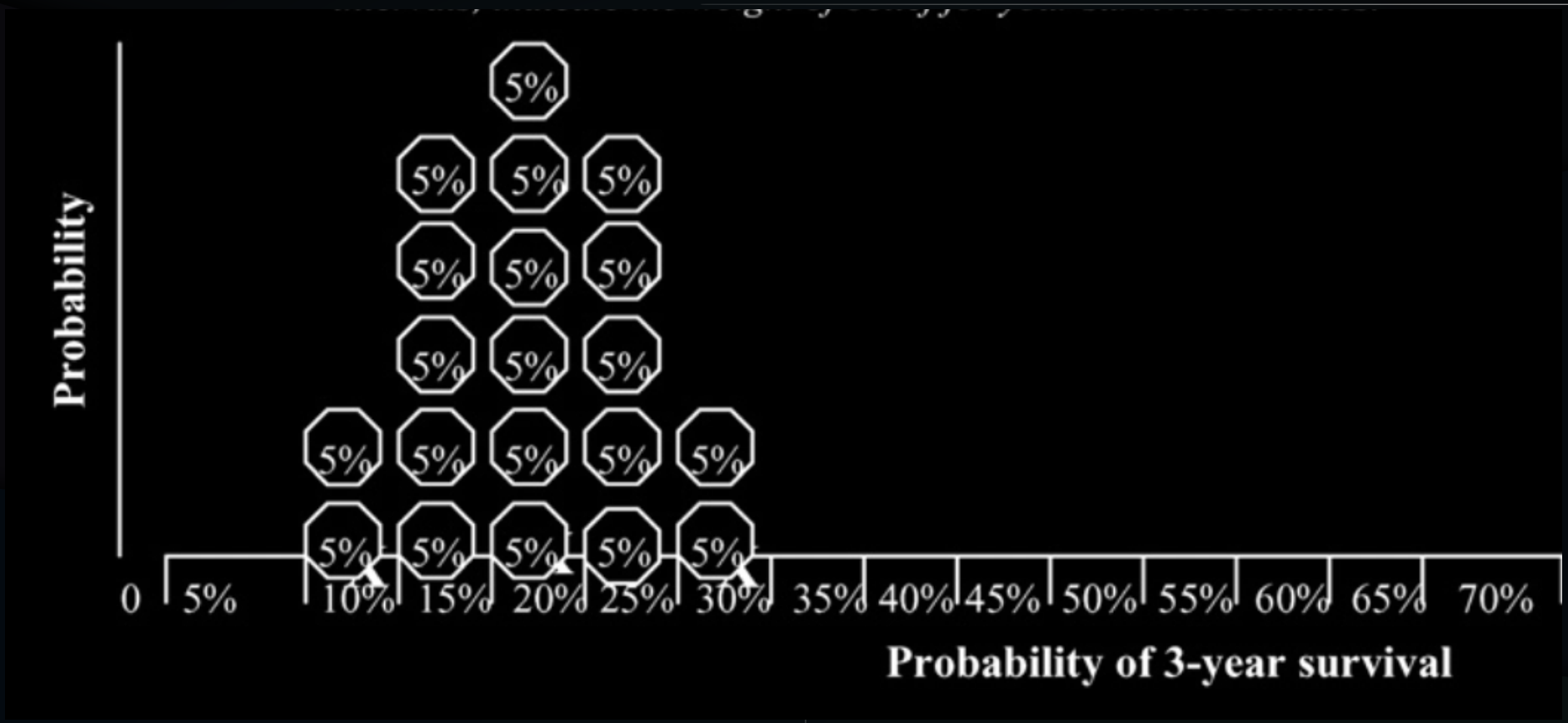
The Prior Distribution Elicitation



Normalize marks for prior on pain effect of LUNA vs placebo
(Latthe et al 2005, J Obs Gync)

http://faculty.washington.edu/kenrice/BayesIntroClassEpi2018.pdf

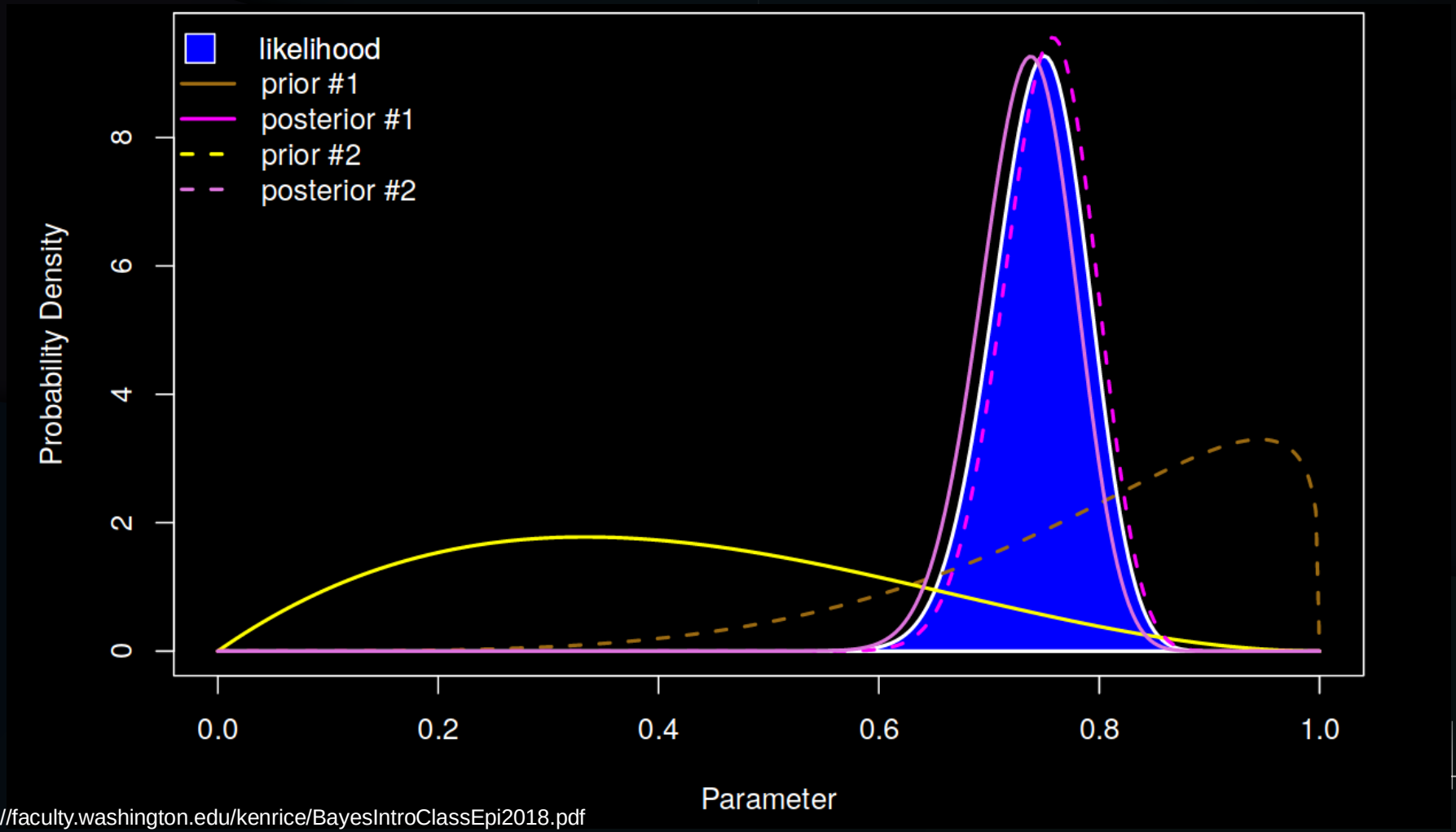
The Prior Distribution Elicitation



Use 20×5% stickers for prior on survival when taking warfarin
(Johnson et al 2010, J Clin Epi)

How much do priors matter?

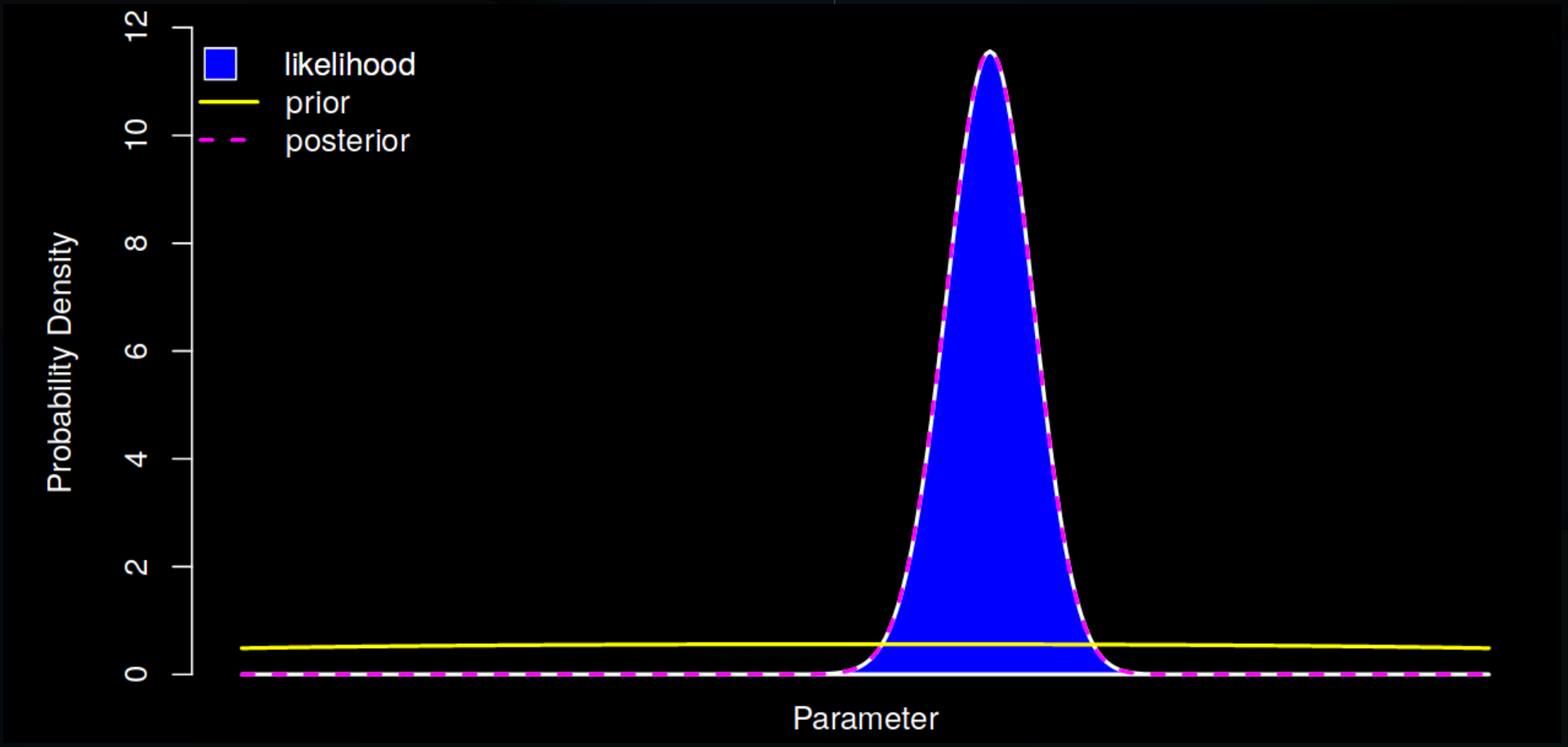
LITTLE when the data provide a lot more information than the prior



http://faculty.washington.edu/kenrice/BayesIntroClassEpi2018.pdf

How much do priors matter?

LITTLE when using flat priors



http://faculty.washington.edu/kenrice/BayesIntroClassEpi2018.pdf

The Prior Distribution 

What you know about θ, excluding the information in the data → P(θ).

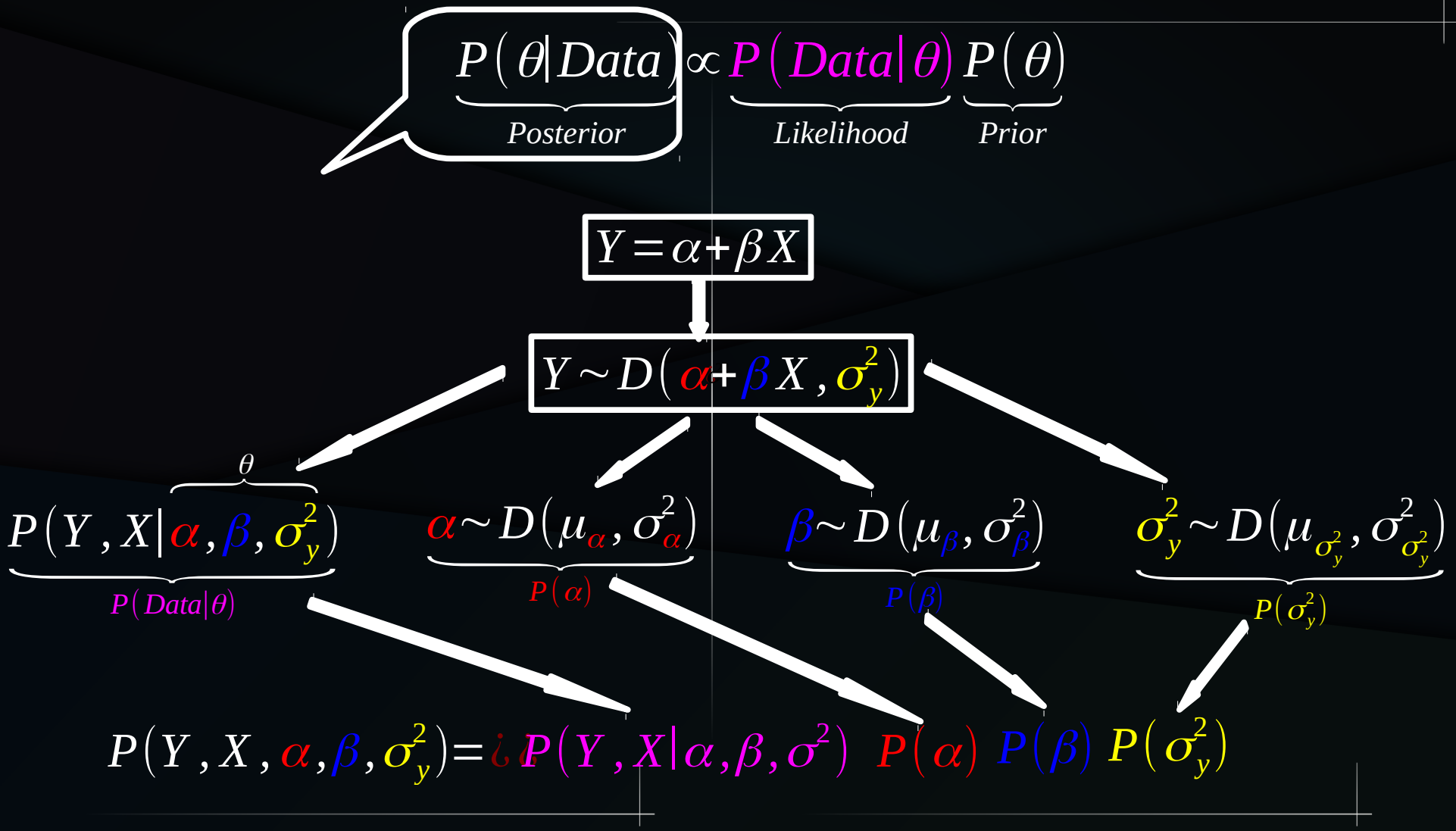
The Likelihood 

Based on modeling assumptions, how (relatively) likely the data Y are IF the truth is θ → P(Y | θ)

The Posterior 

$$\underbrace{P(\theta|Data)}_{Posterior} \propto \underbrace{P(Data|\theta)}_{Likelihood} \underbrace{P(\theta)}_{Prior}$$

From Theorem to Inference



Credit: Marco Wirthlin https://www.slideshare.net/MarcoWirthlin/gentle-introduction-bayesian-modelling-and-probabilistic-programming-in-r

From Theorem to Inference

We cannot obtain the conditional posterior probability directly with Continuous Values because averaging over the complete parameter space via integration is impractical!

$$\underbrace{P(\theta|Data)}_{Posterior} \propto \underbrace{P(Data|\theta)}_{Likelihood}\ \underbrace{P(\theta)}_{Prior}$$

$$\underbrace{P(Data, \theta)}_{Joint}$$

We sample from this joint probability distribution with smart MCMC algorithms!

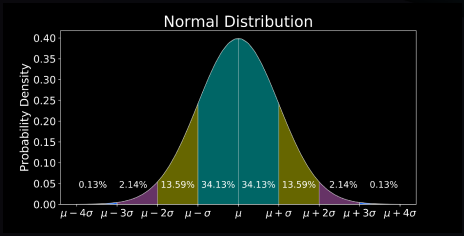
$$P(Y, X, \alpha, \beta, \sigma_y^2) = \propto P(Y, X|\alpha, \beta, \sigma^2)\ P(\alpha)\ P(\beta)\ P(\sigma_y^2)$$

Credit: Marco Wirthlin https://www.slideshare.net/MarcoWirthlin/gentle-introduction-bayesian-modelling-and-probabilistic-programming-in-r

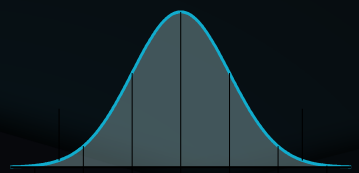
Bayesian Inference Recap

1. Collect Data

True Population N → ∞



Sample Population N = n



Y=[1,0,0,1,...]

x=[2,3,4,…]

2. Write a model for the Data Generating Process

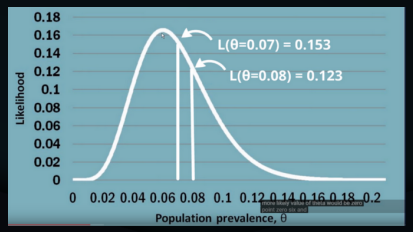
$$Y = \alpha + \beta x \longrightarrow Y \sim D(\underbrace{\alpha + \beta X}_{\mu}, \sigma_y^2)$$

3. We choose the prior distributions for $P(\theta) = P(\alpha, \beta, \sigma_y^2)$

Bayesian Inference Recap

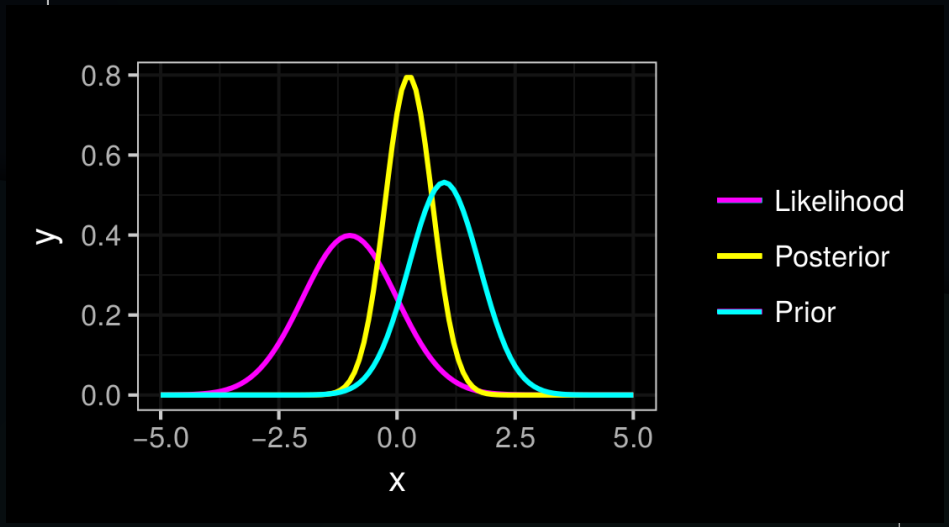
4. We construct the Likelihood function for the data Y, X

$$P(Data|\theta)$$



5. We update our priors by multiplying the priors by the likelihood

$$P(\theta, Data) = P(Data|\theta)\,P(\theta)$$

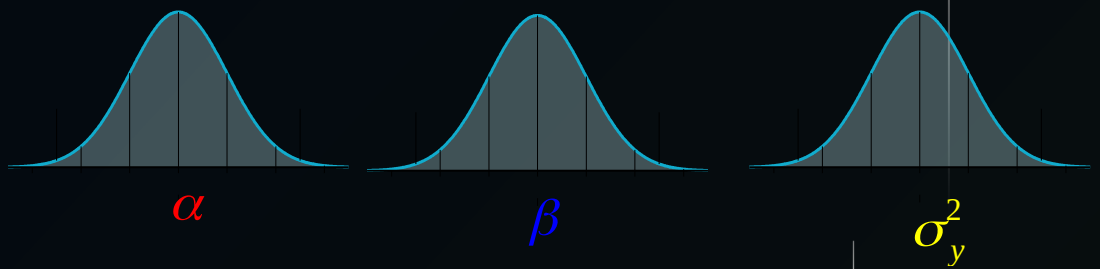


Bayesian Inference Recap

6. Since $\underbrace{P(\theta|Data)}_{Posterior} \propto \underbrace{\underbrace{P(Data|\theta)}_{Likelihood}\underbrace{P(\theta)}_{Prior}}_{\underbrace{P(Data,\theta)}_{Joint}}$

We use MCMC algorithms to sample from the Joint

7. We obtain one parameter for each iteration of the MCMC!

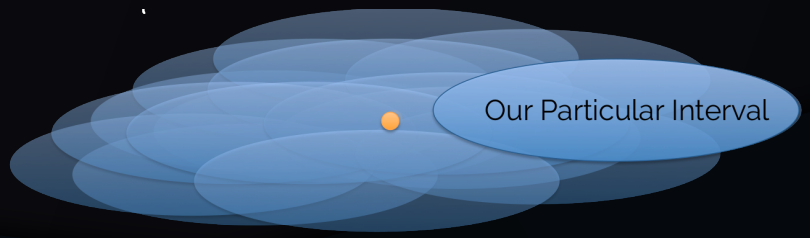


Unlike the frequentist approach where we have a UNIQUE “true” parameter

The statistical diatribe in (very) brief

Frequentism

is a probabilistic recipe for generating confidence intervals given a fixed model parameter

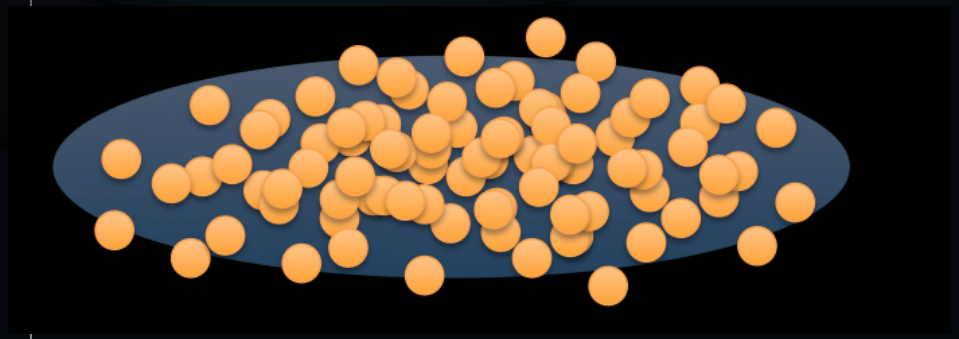


A 95% of such Confidence Intervals in repeated experiments will contain the true value!

$$P(Data|\theta)$$

Bayesianism

is a probabilistic statement about model parameters given a fixed credible region



A 95% Credible Region is 95% likely to contain the true value!

$$\underbrace{P(\theta|Data)}_{Posterior} \propto \underbrace{P(Data|\theta)}_{Likelihood} \underbrace{P(\theta)}_{Prior}$$

Credits: VanderPlas, Jake (2014). url:https://speakerdeck.com/jakevdp/frequentism-and-bayesianism-whats-the-big-deal-scipy-2014.

WHAT IS Bayesian Analysis?

- The use of **probability** to represent **uncertainty** in all parts of the model
- No replications needed (unlike frequentist)
- Keep adding data, and updating knowledge, as data becomes available... knowledge will concentrate around true θ
- Information efficient method (Priors!) :)
- Computationally intensive :(

WHAT IS NOT?

- A category of models
- Subjective

What is uncertain?

Are Data uncertain?

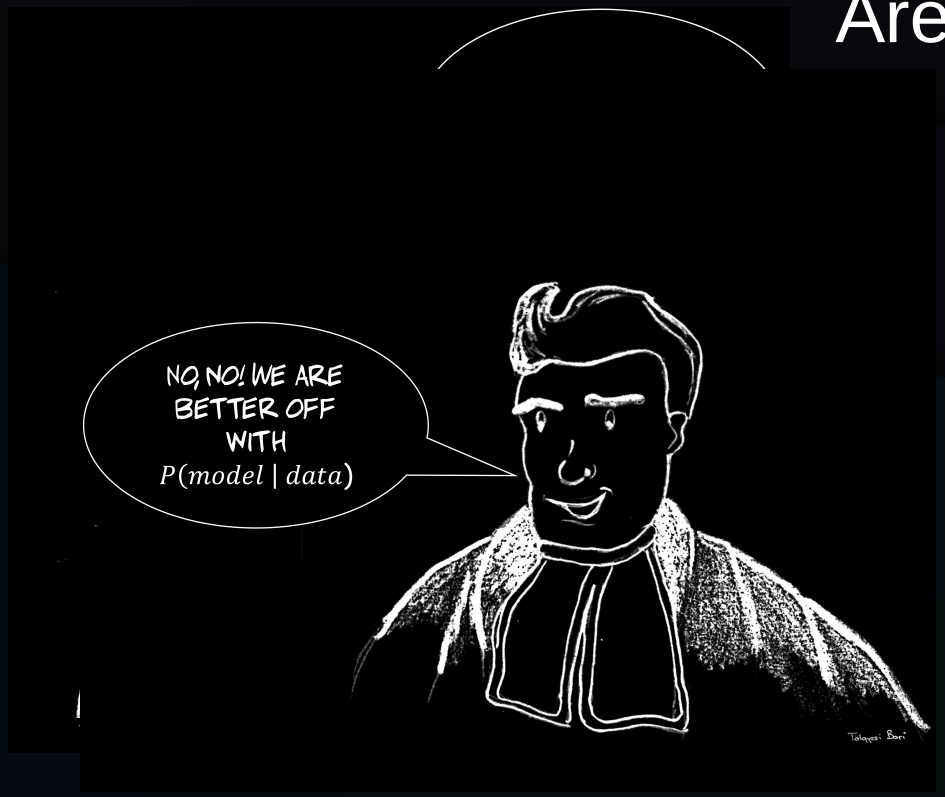
$$P(x)?\,P(Y)?$$

Is the model uncertain?

$$P(\alpha+\beta x)?$$

Are the parameters uncertain?

$$P(\beta)?\,P(\alpha)?\,P(\theta)?$$



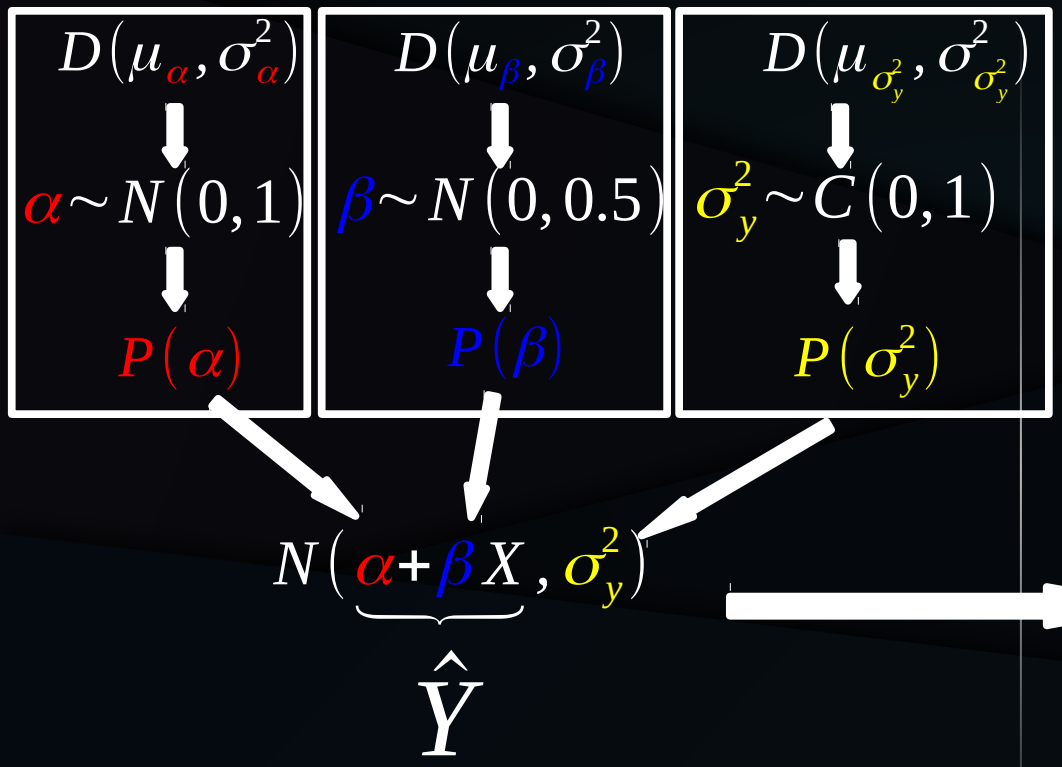
Pierre-Simon, marquis de Laplace (1749–1827)
Inventor of Bayesian inference

STAN

Stan language makes use of the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014):

- Higher effective sample size per iteration for complex posteriors.
- Higher number of effective samples per second.
- Does not require any special behavior for conjugate priors.

Take it to STAN




```
data {
  int<lower=0> N;
  vector[N] X;
  vector[N] Y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
transformed parameters {
  vector[N] y_hat;
  y_hat = beta * X + alpha;
}
model {
  Y ~ normal(y_hat, sigma);
// priors
  alpha ~ normal(0,1);
  beta  ~ normal(0,0.5);
  sigma ~ cauchy(0,1);
// likelihood
  Y ~ normal(y_hat, sigma);
}
```

Data

Parameters

Transformed Parameters

Model

Credit: Marco Wirthlin https://www.slideshare.net/MarcoWirthlin/gentle-introduction-bayesian-modelling-and-probabilistic-programming-in-r

R & STAN

- `rstan`: R Interface to Stan C++ library for Bayesian estimation.

  Upload your STAN code and run it through R.

- `rstanarm & brms:` runs `rstan` in the back-end.

  You specify models via the R syntax with a `formula` and `data.frame` plus some additional arguments for priors.

STAN

```
my_wonderful_model <- data {
  int<lower=0> N;
  vector[N] X;
  vector[N] Y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
transformed parameters {
  real y_hat;
  y_hat = alpha + beta * X;
}
model {
  y ~ normal(y_hat, sigma);
  // priors
  alpha ~ normal(0,1);
  beta  ~ normal(0,0.5);
  sigma ~ gamma(0,0.1);
  // likelihood
  err ~ normal(0,sigma);
}
```

rstan

R

```
library(rstan)
library(rstantools)

my_data <- list(
  'N' = length(data$Y),
  'X' = data$X,
  'Y' = data$Y)

fit_1 <- stan(model_code =  my_wonderful_model,
              data = my_data,
              iter = 4000,
              warmup = 1000,
              chains = 4,
              seed = 1234)

estimates <- rstan::extract(fit_1, permuted = TRUE)
```

My Data

Load Model

Estimates

rstanarm & brms

~ 16 contributors

```
stan_glm(formula = mpg ~ wt + am + cyl,
         data = mtcars,
         prior = NULL,
         family = gaussian(),
         chains = 4,
         iter = 2000,
         warmup = 1000)
```

Random effects:

```
formula = mpg ~ wt + a
```

Smooth Terms:

```
formula = y ~ s(x0) + x1
```

```
my_wonderful_model <- data {
  int<lower=0> N;
  vector[N] X;
  vector[N] Y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
transformed parameters {
  real y_hat;
  y_hat = alpha + beta * X;
}
model {
  y ~ normal(y_hat, sigma);
  // priors
  alpha ~ normal(0,1);
  beta  ~ normal(0,0.5);
  sigma ~ gamma(0,0.1);
  // likelihood
  err ~ normal(0,sigma);
}
```

Sistem.time()

`rstanarm`

system.time():

user	system	elapsed
0.924	0.000	0.918

`brms`

system.time():

Compiling the C++ model

user	system	elapsed
50.728	1.276	52.083

But in terms of sampling they have the same `system.time()` !

Priors Specification: the Dirty Job :)

Student t family

Hierarchical shrinkage family

Dirichlet family

Product-normal family

Laplace family

```
prior <- c(set_prior("normal(0,10)", class = "b"),
           set_prior("normal(1,2)", class = "b", coef = "wt"),
           # Sd of group-level ('random') effects
           set_prior("cauchy(0,2)", class = "sd", group = "cyl", coef = "Intercept")),
           set_prior("student_t(3, 0, 10)", class = "sigma")))
```

Only in `brms`

Break vectorization, it may slow down the process.

Some helpful commands

- `brms`: which priors to specify?

```
get_prior(formula = mpg ~ wt + am + (1|cyl), dat
```

- `brms & rstanarm`: how to paralle

```
brm(formula = mpg ~ wt + am + cyl, data = mtcars
```

- `brms`: how to see the STAN code?

```
make_stancode(mpg ~ wt + am + cyl, data = mtcars
```

```
my_wonderful_model <- data {
  int<lower=0> N;
  vector[N] X;
  vector[N] Y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
transformed parameters {
  real y_hat;
  y_hat = alpha + beta * X;
}
model {
  y ~ normal(y_hat, sigma);
  // priors
  alpha ~ normal(0,1);
  beta  ~ normal(0,0.5);
  sigma ~ gamma(0,0.1);
  // likelihood
  err ~ normal(0,sigma);
}
```

Results' Diagnostics: `shinystan`

`shinystan` works both for `stanreg` and `brmsfit` objects:

```
m1 <- stan_glmer(formula = mpg ~ wt + am + (1|cyl),
                 data = mtcars,  prior = NULL,
                 family = gaussian())

launch_shinystan(m1)
```

Residuals Correlations

`brms`:

- **cor_arma**: autoregressive-moving average (ARMA) structure.
- **cor_arr**: response autoregressive (ARR) structure
- **cor_car**: Spatial conditional autoregressive (CAR) structure
- **cor_sar**: Spatial simultaneous autoregressive (SAR) structure
- **cor_bsts**: Bayesian structural time series (BSTS) structure
- **cor_fixed**: fixed user-defined covariance structure

Final Remarks

- `rstanarm` easiest package to start with since pre-compiled, but limiting (intentionally) for more advanced needs.
- `brms` is more flexible and customizable.
- `rstan` fully flexible, but you need to learn STAN programming (maybe with the help of `make_stancode`).

Warnings 

- Sampling can be slow (especially with inaccurate priors/model specification)
- You need to be really careful about diagnostics
- You need to have ideas about priors

I don’t know if I am Bayesian…
When is it convenient to be a Bayesian?

- Prior beliefs/useful information you want to incorporate
- Few data
- A lot of uncertainty to quantify
- Many unknown model parameters
- Unique events

#rstanarm

josie hughes @josie_shoes · 24 Mar 2017

I have good results! On Friday afternoon! The **#rstanarm** R pkg is lovely. Thanks @mcmc_stan. #rstats

1 6

Heidi Lorimor @heidi_lorimor · 28 Oct 2016

running my first bayesian models today! **#rstanarm** and #shinystan are amazing!

2

Roland Schäfer @codeslapper · 11 Apr 2016

But I want to make it clear: **#rstanarm** is REALLY a great tool, esp. for a #frequentist who wants to play around...

> **Roland Schäfer** @codeslapper
> With #rstanarm I can now easily verify that all my #glmer models turn out exactly the same with #Bayesian estimation taking 50x longer ;)

#brms

Frank Harrell
@f2harrell

Follow

Bayesian regression modeling: R brms package a breakthrough, & article by Bürkner is as well written as it is useful: jstatsoft.org/article/view/v…

Shravan Vasishth @vasishthlab · 22 Sep 2017

Replying to @f2harrell

Bürkner deserves a prize.

Stephen Martin @smartin2018 · 23 Sep 2017

And brms is just crazy potent. Want a location-scale-shape crossed random effects mixture model? You can. Goodness.

1

Not enough?

”...in terms of forecasting ability, ...a good Bayesian will beat a non-Bayesian, who, in turn, will do better than a bad Bayesian.”

C.W.J. Granger (1986, p. 16)

Not enough?

“If you are not using informative prior, you are leaving money on the table.”

Robert Weiss, UCLA

Other Sources (Credit: Marco Wirthlin @marcowirthlin)

About Generative vs. Discriminative models:
Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison
of logistic regression and naive bayes. In Advances in neural information processing systems,
pages 841–848.
Rasmus Bååth:
Video Introduction to Bayesian Data Analysis, Part 1: What is Bayes?:
https://www.youtube.com/watch?time_continue=366&v=3OJEae7Qb_o

When to use ML vs. Statistical Modelling:
Frank Harrell's Blog:
http://www.fharrell.com/post/stat-ml/
http://www.fharrell.com/post/stat-ml2/

Frequentist approach: How do sampling distributions work (applet):
http://onlinestatbook.com/stat_sim/sampling_dist/index.html

Bayesian inference and computation:
John Kruschke: Doing Bayesian Data Analysis:
A Tutorial with R, JAGS, and Stan Chapter 5
Rasmus Bååth:
http://www.sumsar.net/blog/2017/02/introduction-to-bayesian-data-analysis-part-two/
Richard McElreath:
Statistical Rethinking book and lectures
https://www.youtube.com/watch?v=4WVelCswXo4

Many model examples in Stan:
https://mc-stan.org/users/documentation/case-studies

About Bayesian Neural Networks:
https://alexgkendall.com/computer_vision/bayesian_deep_learning_for_safe_ai/
https://twiecki.io/blog/2018/08/13/hierarchical_bayesian_neural_network/

Volatility Examples:
Hidden Markov Models:
https://github.com/luisdamiano/rfinance17
Volatility Garch Model and Bayesian Workflow:
https://luisdamiano.github.io/personal/volatility_stan2018.pdf

Dictionary: Stats ↔ ML
https://ubc-mds.github.io/resources_pages/terminology/

The Bayesian Workflow:
https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html

Algorithm explanation applet for MCMC exploration of the parameter space:
http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/

Probabilistic Programming Conference Talks:
https://www.youtube.com/watch?v=crvNIGyqGSU

Bayesian on Twitter (Credit: Marco Wirthlin)

- Chris Fonnesbeck @fonnesbeck (pyMC3)
- Thomas Wiecki @twiecki (pyMC3)
- Blog: https://twiecki.io/ (nice intros)
- Bayes Dose @BayesDose (general info and papers)
- Richard McElreath @rlmcelreath (ecology, Bayesian statistics expert)
- All his lectures: https://www.youtube.com/channel/UCNJK6_DZvcMqNSzQdEkzvzA
- Michael Betancourt @betanalpha (Stan)
- Blog: https://betanalpha.github.io/writing/
- Specifically: https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html
- Rasmus Bååth @rabaath
- Great video series: http://www.sumsar.net/blog/2017/02/introduction-to-bayesian-data-analysis-part-one/
- Frank Harrell @f2harrell (statistics sage)
- Great Blog: http://www.fharrell.com/
- Andrew Gelman @StatModeling (statistics sage)
- https://statmodeling.stat.columbia.edu/
- Judea Pearl @yudapearl
- Book of Why: http://bayes.cs.ucla.edu/WHY/ (more about causality, BN and DAG)
- AND MANY MORE!

Thanks!

Actively looking for job opportunities in the area!

a.lice.milivinti@gmail.com