S-043/Stat-151
Analysis for Clustered and Longitudinal Data
(Multilevel & Longitudinal Models)

Lectures ST.2
Our possible friend Stan
or
Fitting models with a very flexible Bayesian modeling framework

Some slides taken from a deck authored by Ken Rice

Roadmap

★ Introduction to Bayesian logic
- The power of Bayesian thinking

★ Introduction to STAN
- We will walk through fitting a MLM in STAN
- Lots of code—but we won't dive into it. Just here to get a feel.
- Code available on website; run it yourself!

Motivation:
Imagine living in an ideal world

You specify your model

You write it down in computer code

The computer fits your model

You then have:

- ★ Point estimates of your parameters
- ★ Uncertainty estimates of your parameters

Stan wants to be your ideal world

Stan (and RStan) is a model-fitting software suite.

Some features:

- ★ It takes strong computers lots of time to fit complex models.
- ★ You need a wide variety of tricky techniques to get your model to “converge.”
- ★ It is unbelievably flexible.
- ★ It is a **Bayesian modeling framework**... which is nominally a different statistical philosophy.

Bayesian Thinking

Bayesian thinking is to be cuddled up in a nice framework/ model that solves all your problems, if the modeling is mostly right.

Bayesian estimation

Bayesian Modeling all boils down to Bayes Rule

Rev. Thomas Bayes

1702 - 1761

$$P(A \text{ if } B) = \frac{P(B \text{ if } A)P(A)}{P(B)}$$

Bayesian Inference

The "Truth" is usually our parameters that we are trying to estimate

This part is the Likelihood, the same one as for maximum likelihood. If this is low, we say the data are unlikely given our assumed truth.

POSTERIOR Probability

PRIOR Probability

$$P(truth \text{ if } data) = \frac{P(data \text{ if } truth)P(truth)}{P(data)}$$

- Prior probability: probability of a statement being true, before looking at the data
- Posterior probability: probability of the statement being true, after updating the prior probability based on the data

Bayesian Inference

- ★ **Bayesian inference** does not think about repeated sampling or repeating the experiment, but only what you can tell from your single observed data set.
- ★ **Probability** is considered to be the subjective degree of belief in some statement.
- ★ In Bayesian inference we **condition on the data**, and find the probability of some unknown parameter, *given the data*.

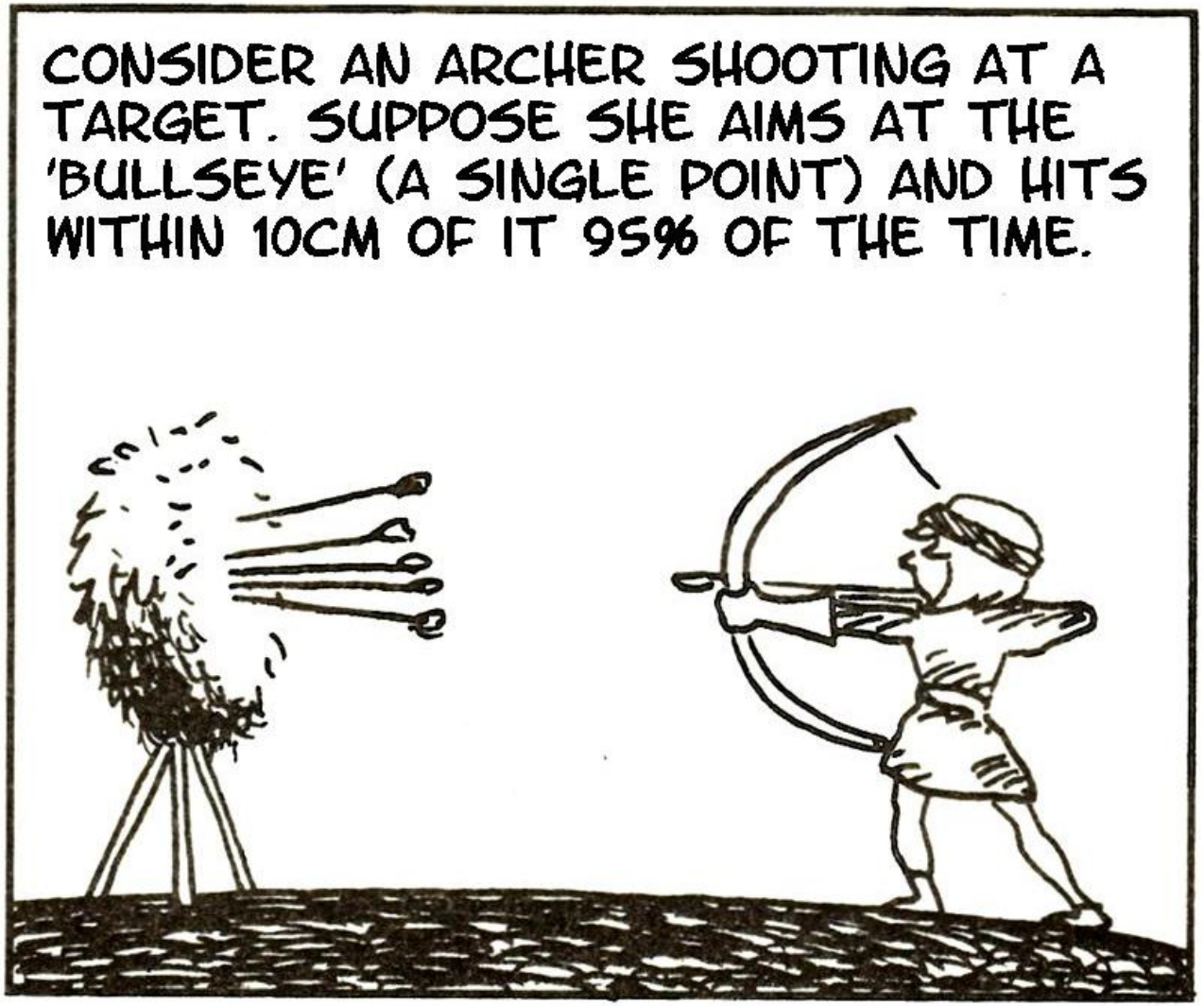
The Posterior

You have some belief (the prior).

You get some further data.

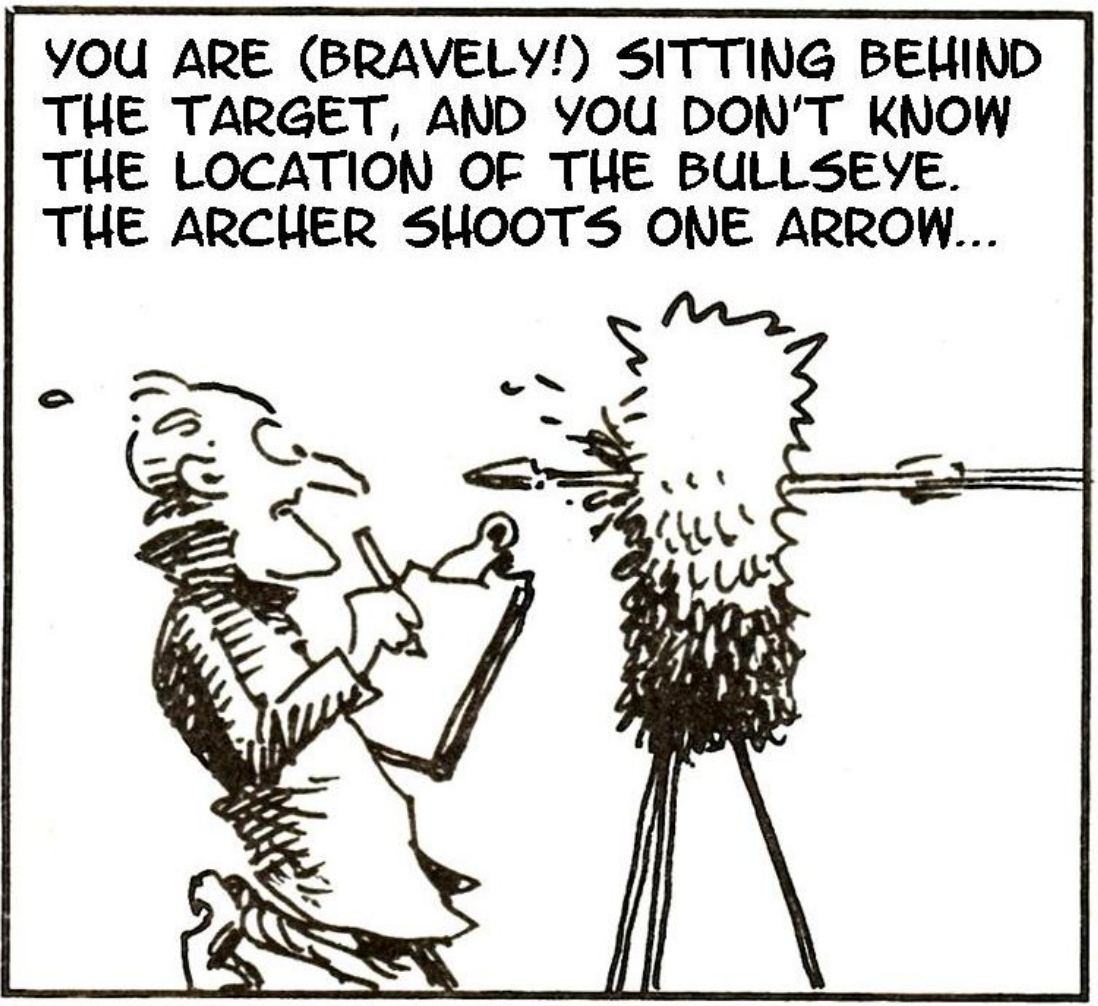
You incorporate that data and **update** to get a new belief (the posterior).

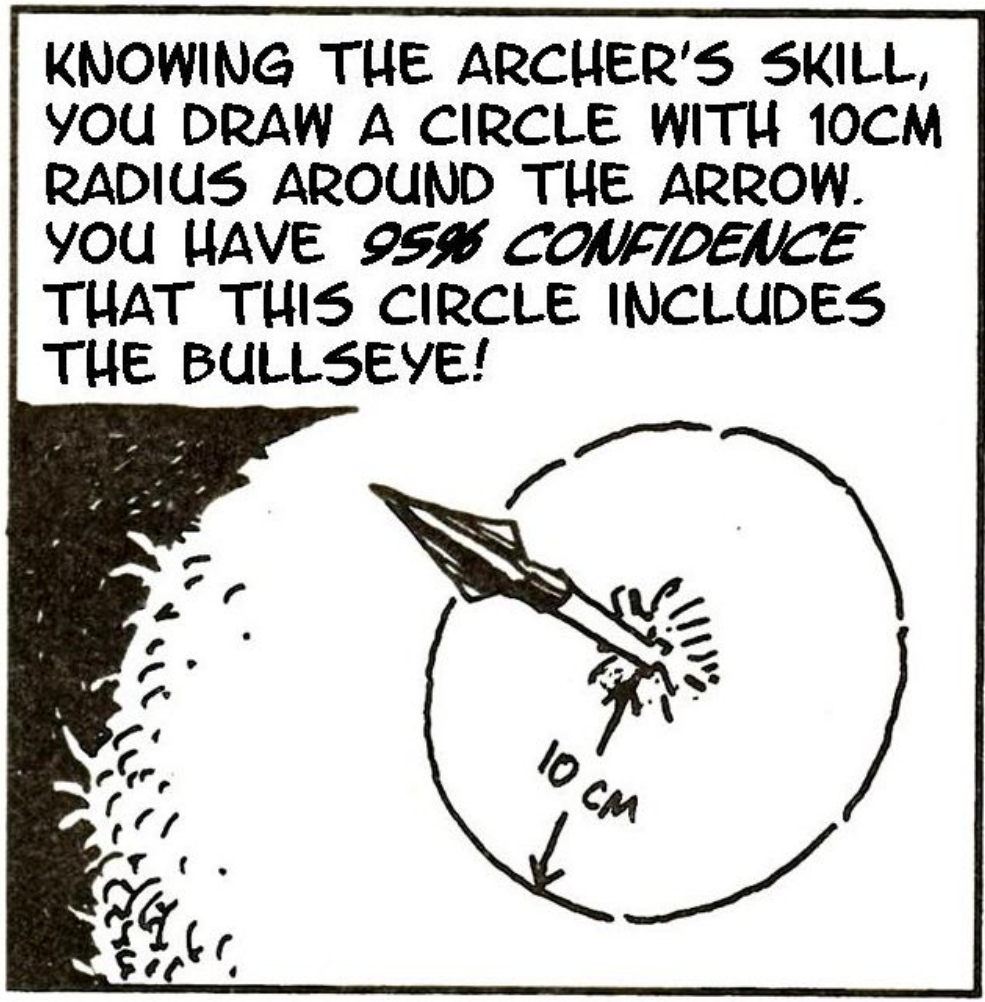
Posterior Distributions: An Archery Example



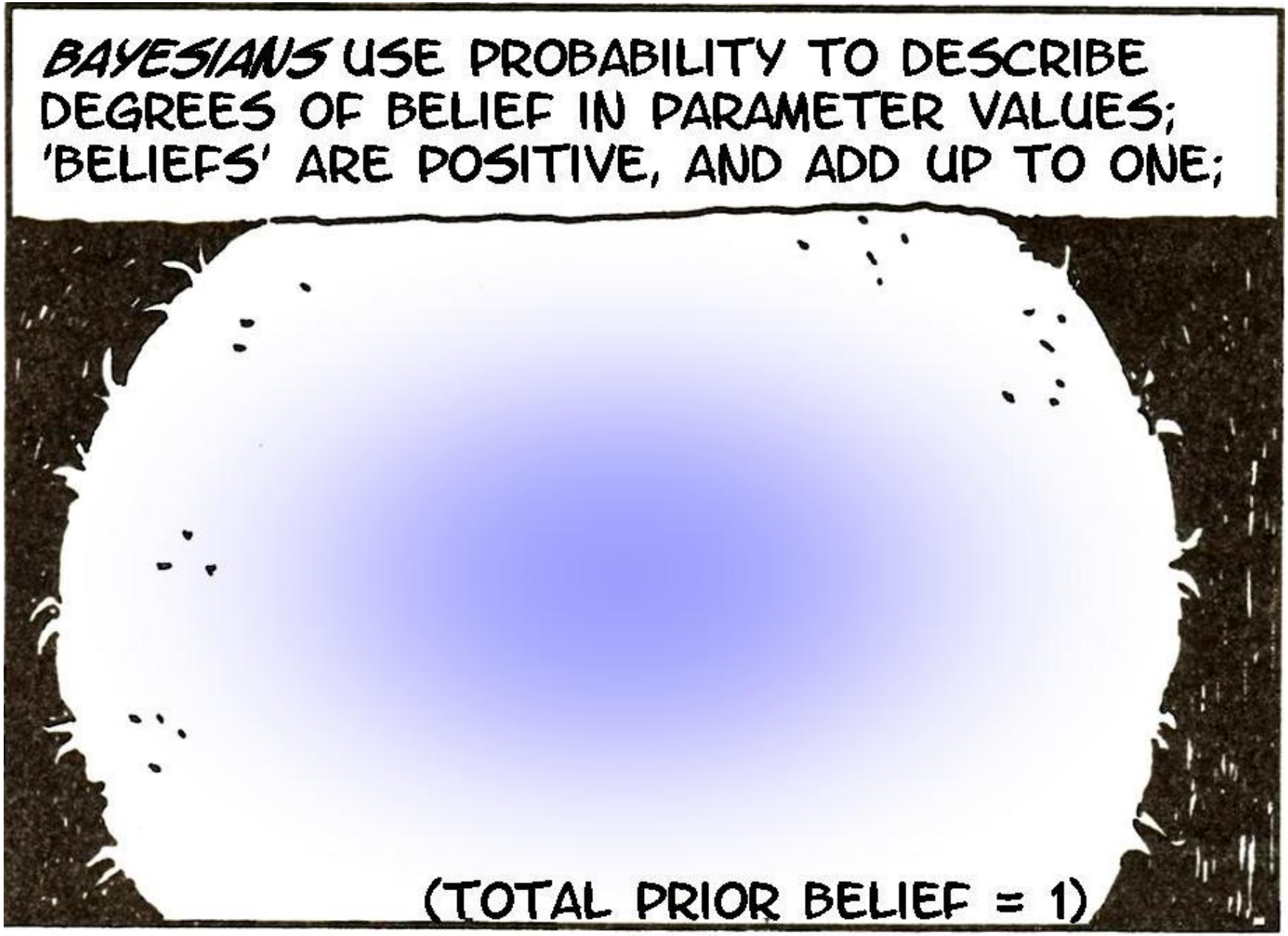
Adapted from Gonick & Smith, *The Cartoon Guide to Statistics*

The Frequentist Model

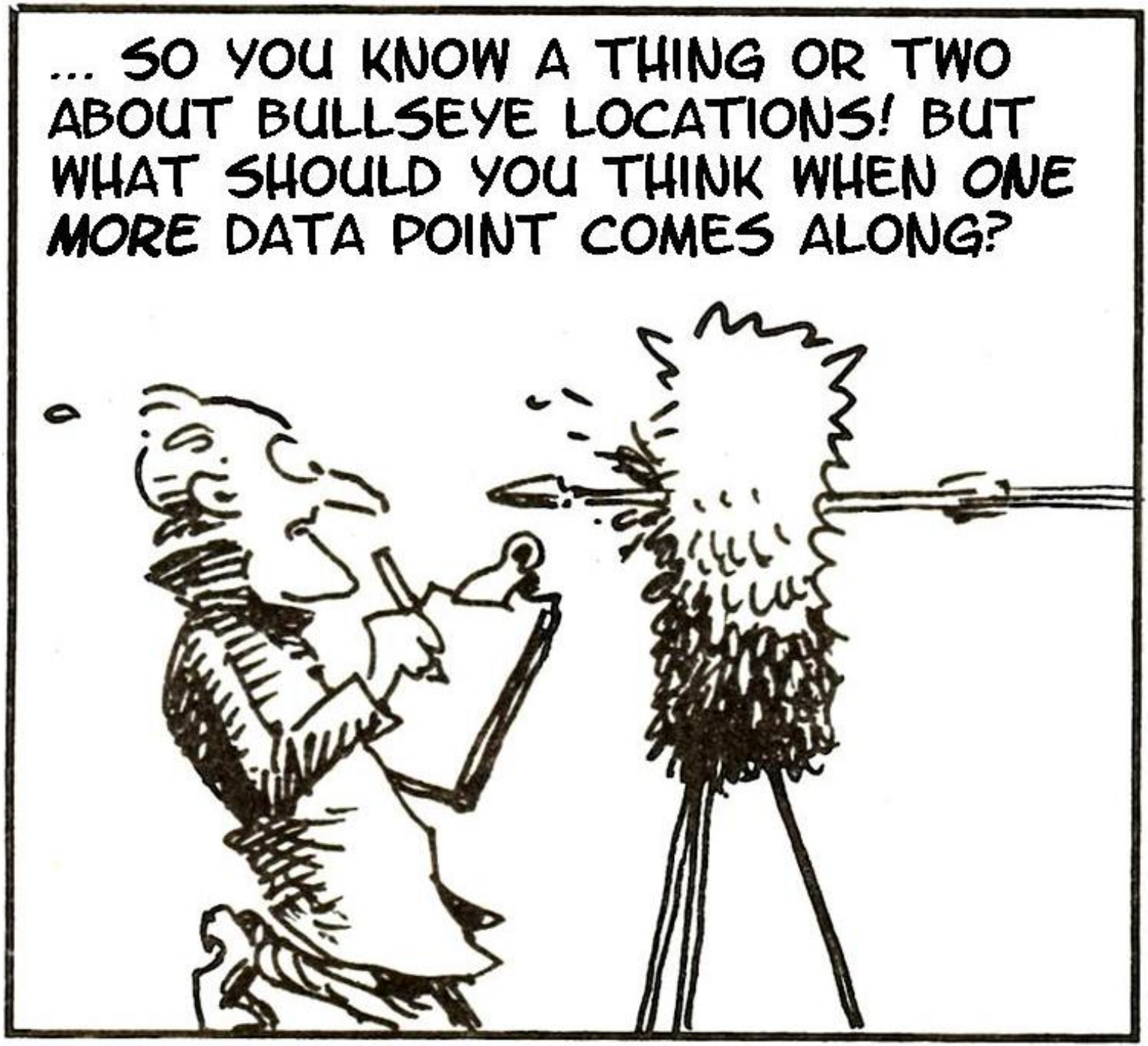


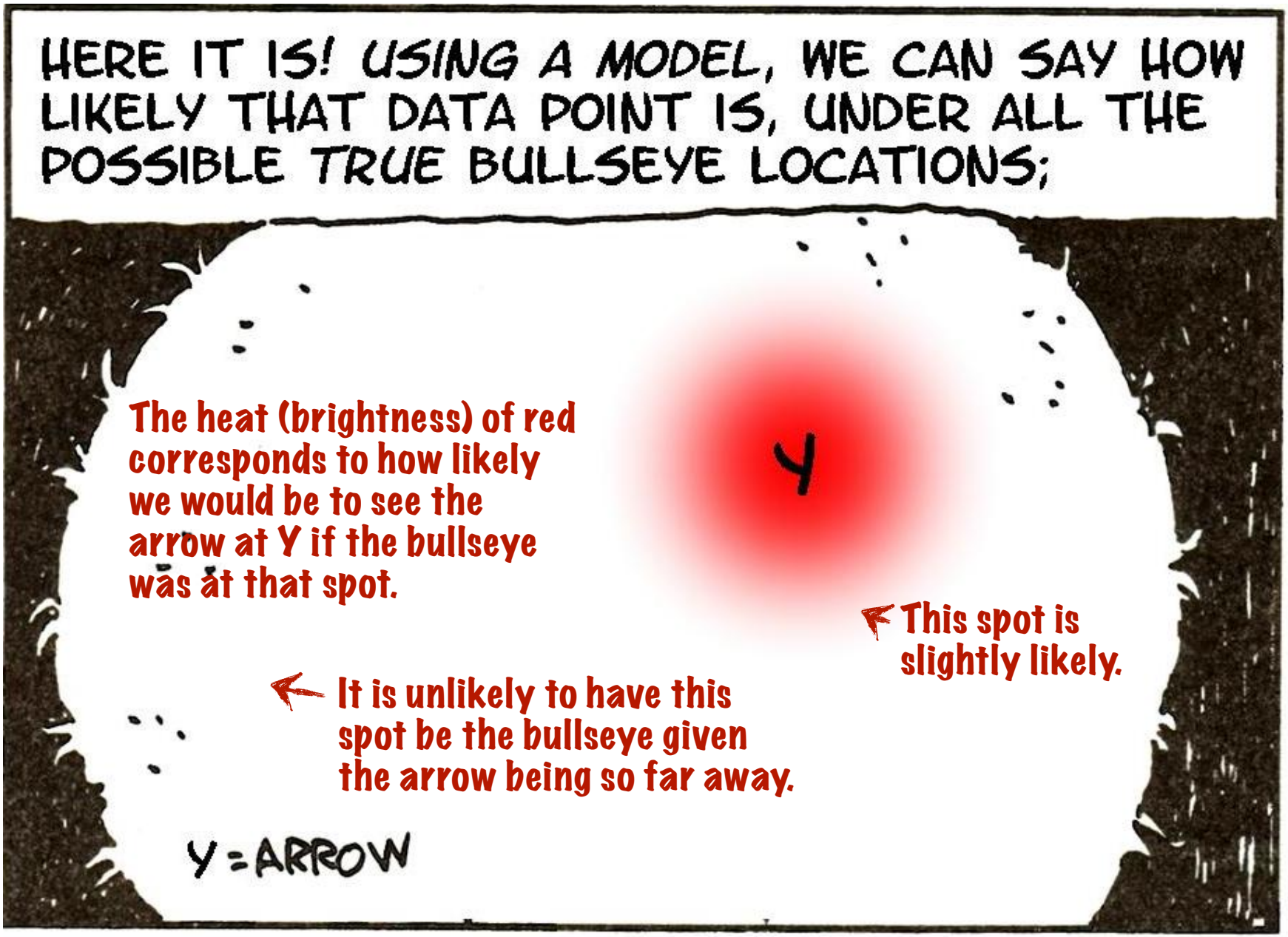


We 'trap' the truth with 95% confidence. Q. 95% of what?

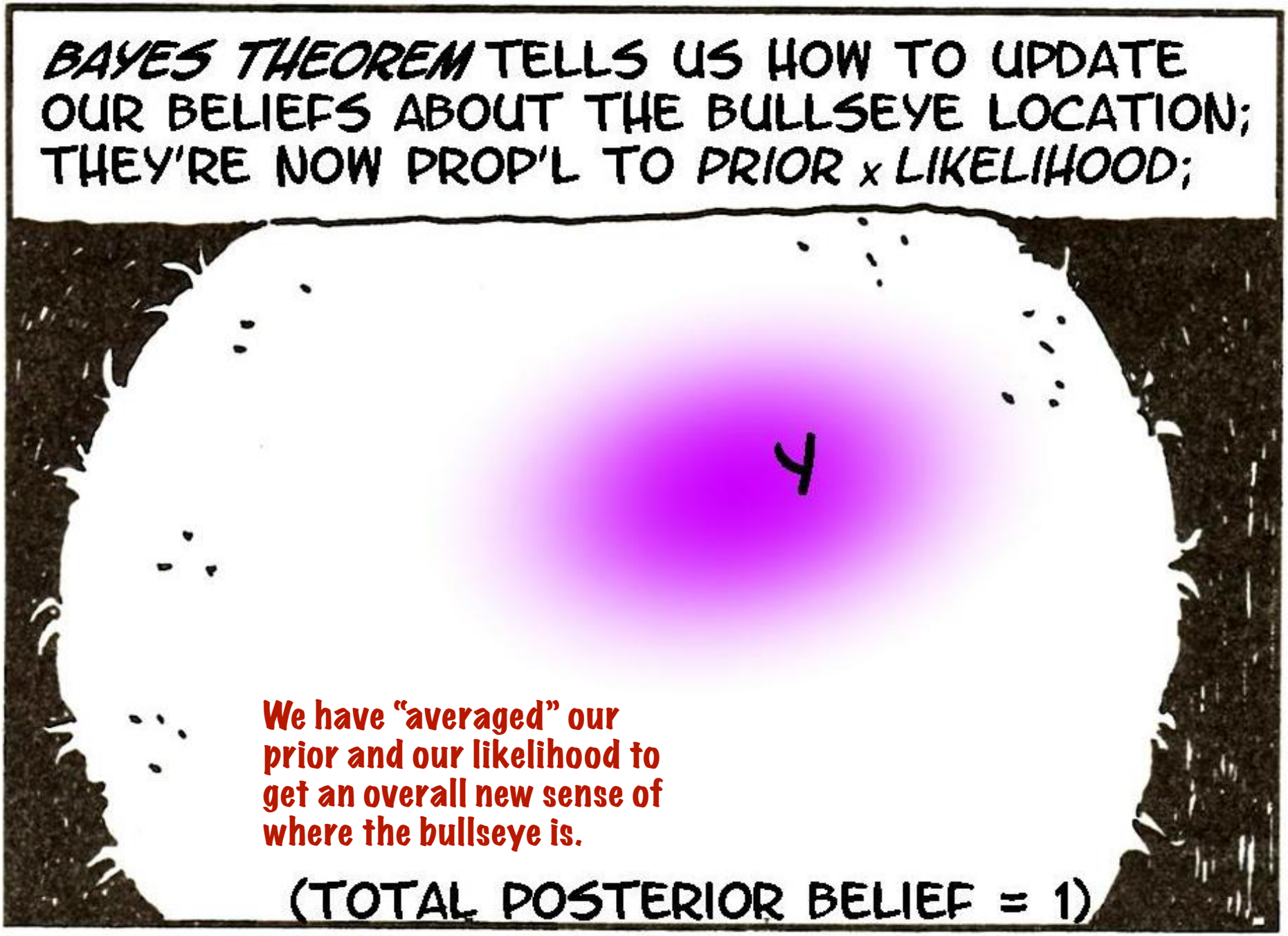


Here we think the bullseye is probably painted in the center, but the painter could have been off by a bit.


... SO YOU KNOW A THING OR TWO ABOUT BULLSEYE LOCATIONS! BUT WHAT SHOULD YOU THINK WHEN ONE MORE DATA POINT COMES ALONG?

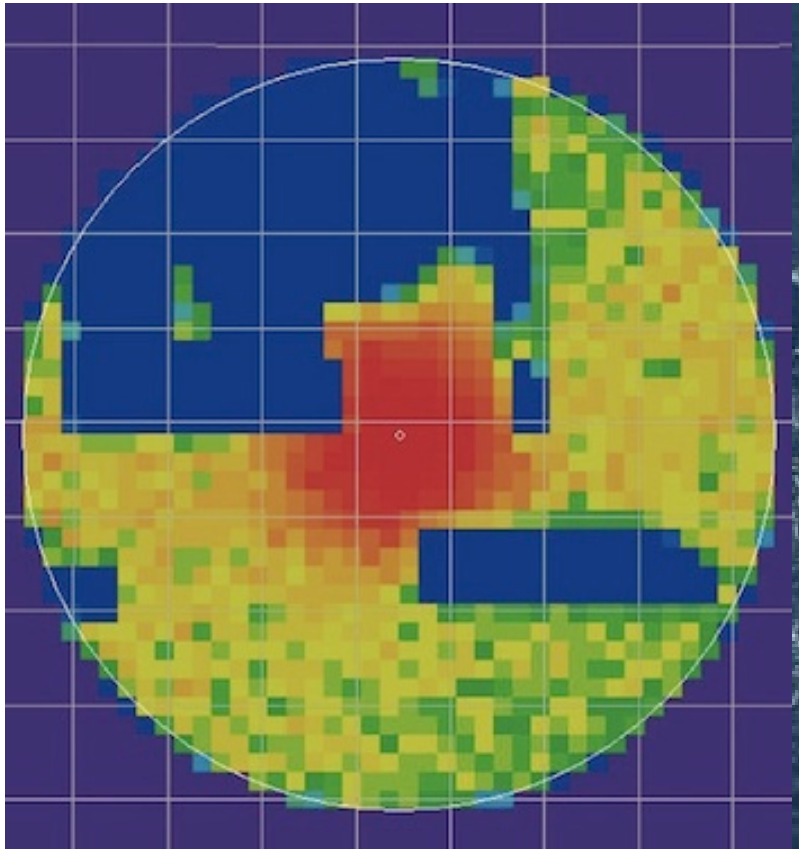


This is the *Likelihood Surface*. We would maximize it to get the MLE.


BAYES THEOREM TELLS US HOW TO UPDATE OUR BELIEFS ABOUT THE BULLSEYE LOCATION; THEY'RE NOW PROP'L TO PRIOR x LIKELIHOOD;
Y
We have "averaged" our prior and our likelihood to get an overall new sense of where the bullseye is.
(TOTAL POSTERIOR BELIEF = 1)

Bayesian inference

Here's *exactly* the same idea, in practice;



- During the search for Air France 447, from 2009-2011, knowledge about the black box location was described via probability – i.e. using Bayesian inference
- Eventually, the black box was found in the red area

Bayes rule for model fitting

If we think of our parameters as ***random*** we have a **posterior probability** of

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)}$$

Read as “The probability of our parameter given our data”

The components:

- ★ the **Prior distribution**: what you know about parameter ß, excluding the information in the data.
- ★ the **Likelihood**: based on modeling assumptions, how [relatively] likely the data Y are if the truth is ß.
- ★ The **denominator:** this is a nuisance, and is a “normalizing constant” which is fixed, given the data. We avoid/ignore it.

But Priors aren't "scientific." Avoiding our prior

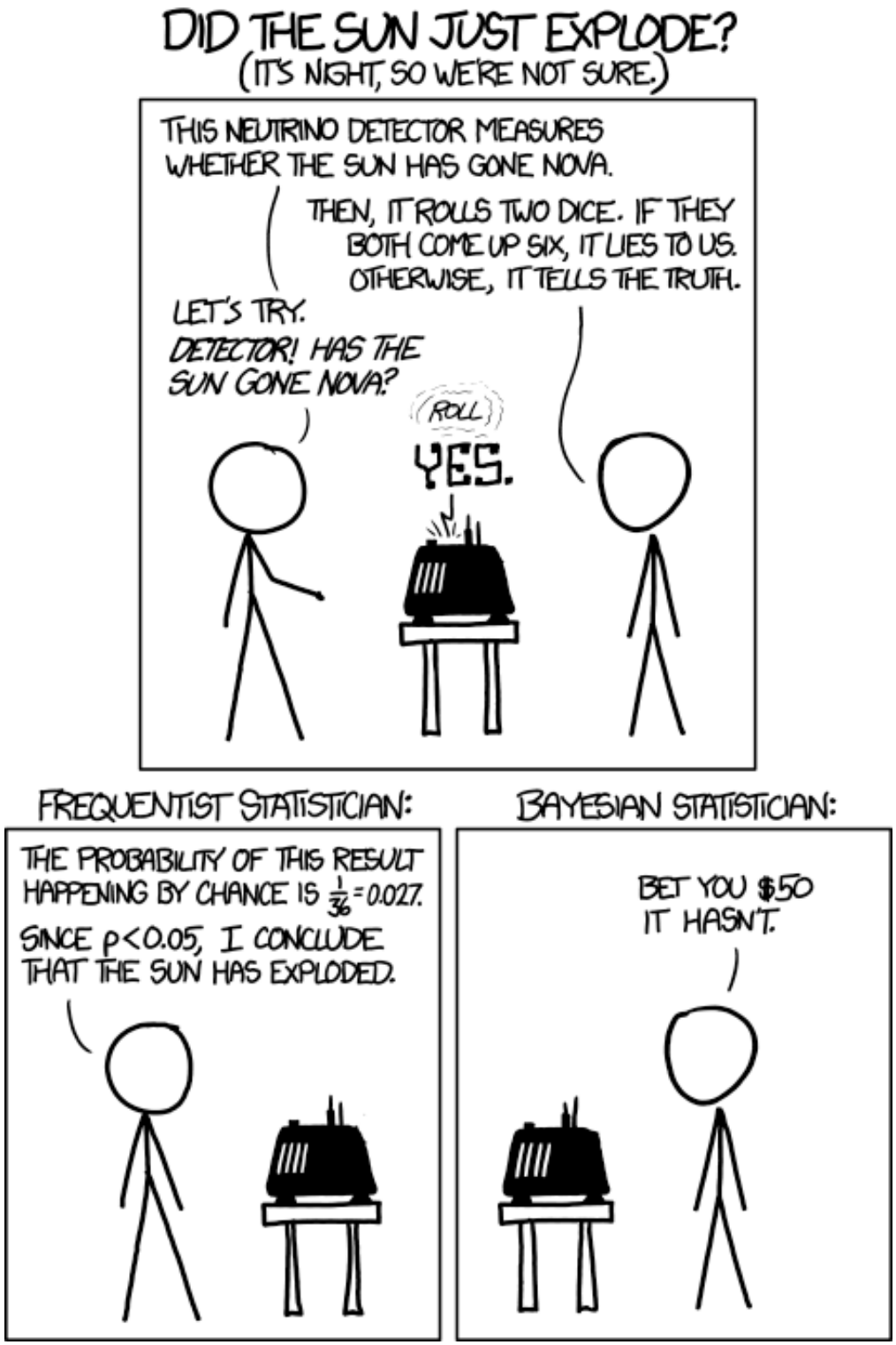
We can extend Bayesian thinking by trying to be as noncommittal as possible regarding the prior.

Some options

- ★ Uninformative priors: you want to be a *frequentist*.
- ★ Minimally informative priors: you believe you know a little.
- ★ Informative priors: you believe you know something.

Regardless: both kinds of inference have the same goal, and it is a goal fundamental to statistics:

to use information from the data to gain information about the unknown truth


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 1/36 = 0.027.
SINCE p<0.05, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

XKCD

Fitting Bayesian Models

Posterior Distributions

Given the prior and the likelihood (the model) you can write down the posterior

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)}$$

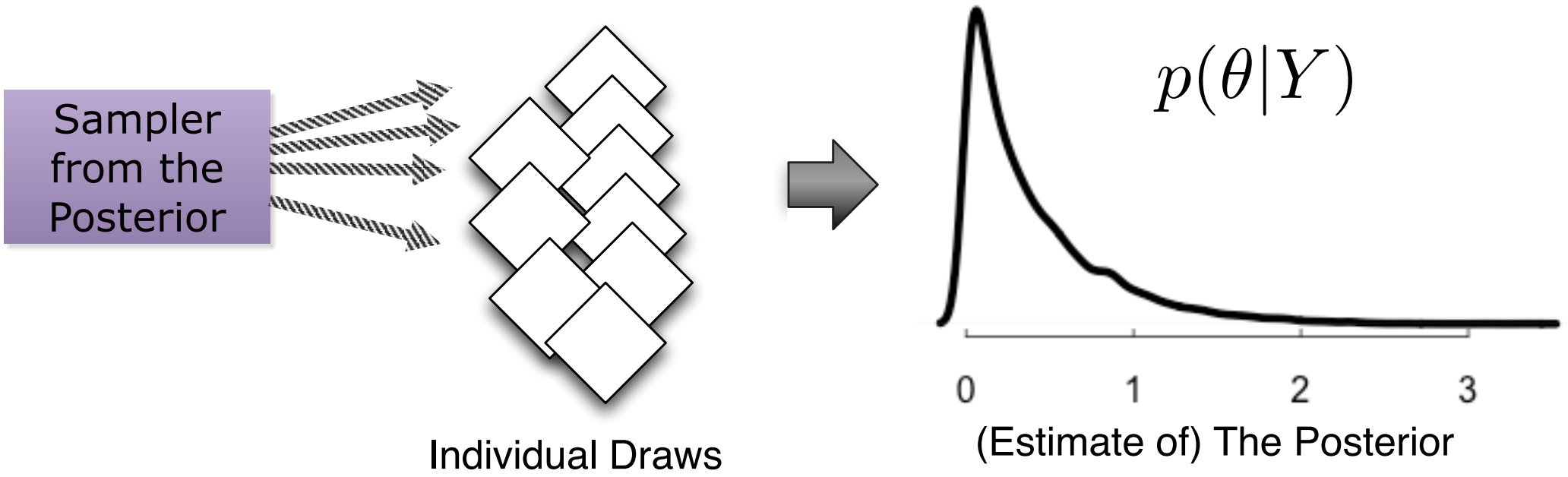
The posterior is a *distribution on where you believe θ is*

For interesting models, it is very hard to evaluate analytically.

Instead we simulate or numerically estimate.

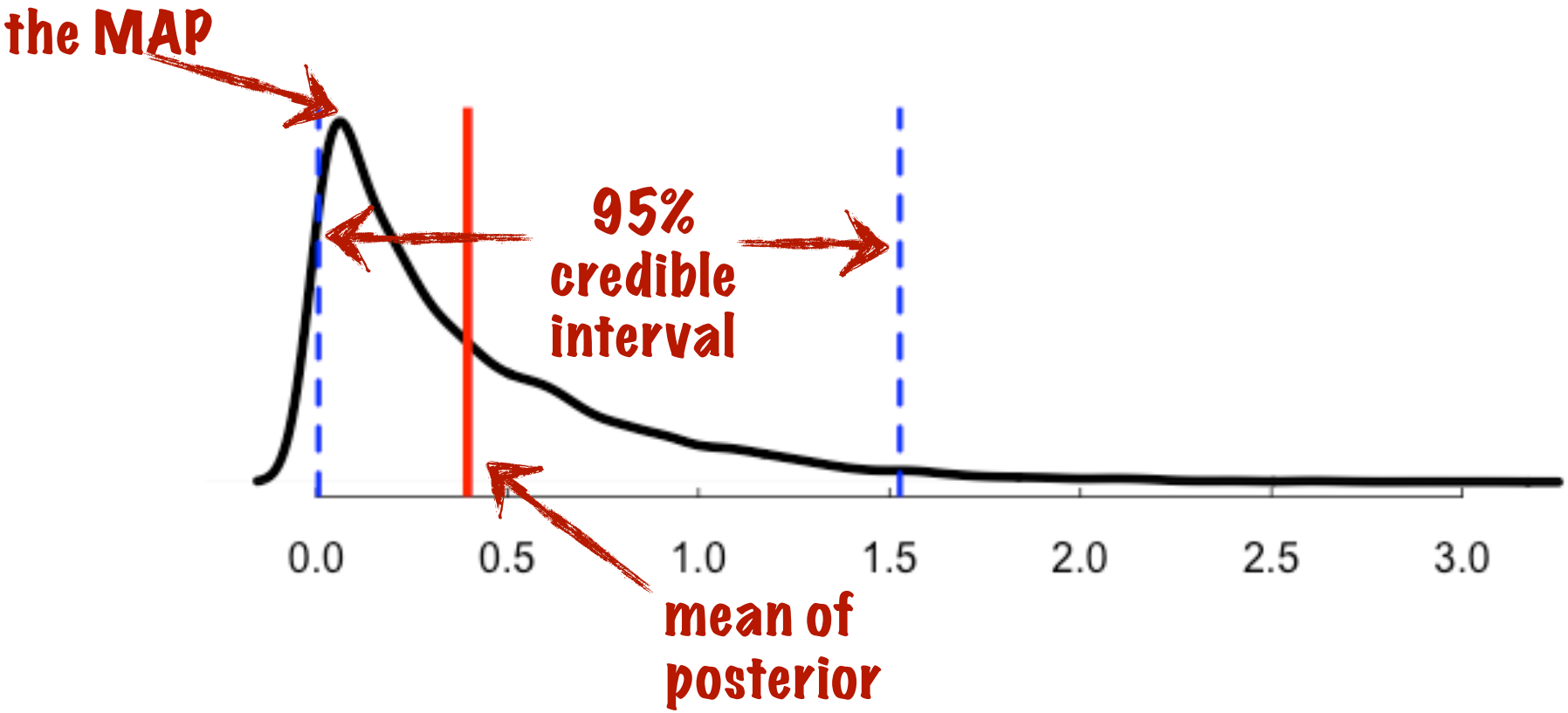
Modern Bayesian Inference: Big Computing

Instead of analyzing the posterior, people generally have computers simulate draws from the posterior.



One classic approach is "MCMC" which stands for Markov Chain Monte Carlo

Using your posterior



Two useful summaries of the posterior:

- ★ The MAP (Maximum a posteriori estimate) is kind of like the maximum likelihood estimate pushed over by the prior.
- ★ The mean (or median) of the posterior. These are easier to obtain. **use these for point estimates, usually.**

Stan, a Big Simulator

Stan simulates from the posterior of a model you specify.

Using this posterior you **estimate** and **assess uncertainty**.

To use Stan

1. Write down a model
2. Define the data your model uses
3. Define the parameters your model has
4. Define the priors for these parameters
5. Encode the relationship of the data to the parameters (i.e., specify the model)
6. Simulate!
7. Analyze results.

Speedy Code Warning

We are about to go through a bunch of R code, but not in detail.

Focus on the big picture.

Step 1: Write down a model

Writing models in the multilevel form is best.

We will use our old friend, High School and Beyond.

$$Math_i = \alpha_{j[i]} + \beta_{j[i]} SES_i + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma_y^2)$$

$$\alpha_j \sim N(\mu, \sigma_\alpha^2)$$

$$\beta_j = \beta_0 + \beta_1 Sector_j$$

No random slope.

We also need priors on: $\sigma_y^2, \sigma_\alpha^2, \beta, \mu$

Major components of the ".stan" file

Your model is specified in a separate file.

```
data {
}

transformed data {
}

parameters {
}

transformed parameter {
}

model {
}
```

Step 2: Define the Data

```
data {
    int<lower=0> nstudents; //Total number of students
    int<lower=0> nschools; //Number of schools

    int<lower=1> schoolIndexes[nstudents];

    // student covariates
    vector[nstudents] ses;

    // school covariates
    vector[nschools] sector;

    // outcome
    vector[nstudents] mathach;

}
```

Which school each student is in. Note: school IDs must run from 1 to nschools.

(So in practice you might have to recode your school IDs to be in sequential order.)

Step 3: Define the Parameters

```
parameters {
    // Population intercept (a real number)
    real mu;

    // Fixed effect of school sector
    real beta_sector;

    // Population slope by SES
    real beta_ses;

    // Interaction of SES and school type (sector)
    real beta_int;

    // Level-1 error
    real<lower=0.001> sigma_y;

    // Level-2 random effect
    real<lower=0.001> sigma_alpha;

    vector[nschools] alpha;
}
```

The random intercepts are parameters too!

Step 4: Define the Priors

These are in your model block

```
// Priors
beta_ses ~ cauchy(0, 10);
beta_sector ~ cauchy(0, 10);
```

They are difficult to decide on. Much controversy. You can also do the following:

- Do nothing (a “flat prior”)
- Set the support (possible values) of your parameter and then do nothing:

```
real<lower=0.001, upper=100> sigma_alpha;
```

Doing nothing is doing something. It is saying you have a flat prior.

Step 5: Define the Model

```
  vector[nstudents] y_hat;

  for (i in 1:nstudents) {
    y_hat[i] <- mu + alpha[ sid[i] ] + ses[i] * beta_ses +
                sector[ sid[i] ]*beta_sector;
  }

  // Priors
  beta_ses ~ cauchy(0, 10);  // Could have let Stan use a
default flat prior here
  beta_sector ~ cauchy(0, 10);  // Could have let Stan use a
default flat prior here
  // The rest are assumed flat.


  // Random effects distribution
  alpha ~ normal(0, sigma_alpha);

  mathach ~ normal( y_hat, sigma_y);
```

We have our reduced form model here to predict the y for each student

This is our distribution on our random intercepts.

This is our outcome as a residual bump around the predicted value

Step 6: Compile and fit your model (Simulate!)

```
> # compile model
> hsb_model <- stan_model(file="lec21_stan_model.stan")

> # fit the model
> model1mcmc <- sampling(hsb_model, data = hsb_dat,
                  chains=4, iter=1000)

SAMPLING FOR MODEL 'lec21_stan_model' NOW (CHAIN 1).

Chain 1, Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 1, Iteration: 100 / 1000 [ 10%]  (Warmup)
...
Chain 1, Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1, Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1, Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1, Iteration: 1000 / 1000 [100%]  (Sampling)
#  Elapsed Time: 26.7856 seconds (Warm-up)
#                4.40143 seconds (Sampling)
#                31.187 seconds (Total)
SAMPLING FOR MODEL 'lec21_stan_model' NOW (CHAIN 2).
```

The raw, fitted model

```
> model1mcmc
Inference for Stan model: lec21_stan_model.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=
```

	mean	se_mean	sd	2.5%	25%
mu	11.73	0.01	0.22	11.28	11.58
beta_ses	2.37	0.00	0.11	2.16	2.30
beta_sector	2.10	0.02	0.34	1.46	1.86
sigma_y	6.09	0.00	0.05	5.99	6.05
alpha[1]	-0.83	0.02	0.83	-2.51	-1.39
alpha[2]	1.07	0.02	1.09	-1.10	0.30
alpha[3]	-2.54	0.02	0.80	-4.12	-3.06
alpha[4]	0.80	0.02	1.11	-1.42	0.04
alpha[5]	-1.21	0.02	0.82	-2.79	-1.77

Number of different start points

How many samples we want from our posterior.

Posterior means (point estimates of parameters)

How variable our parameters are (aka Standard Errors)

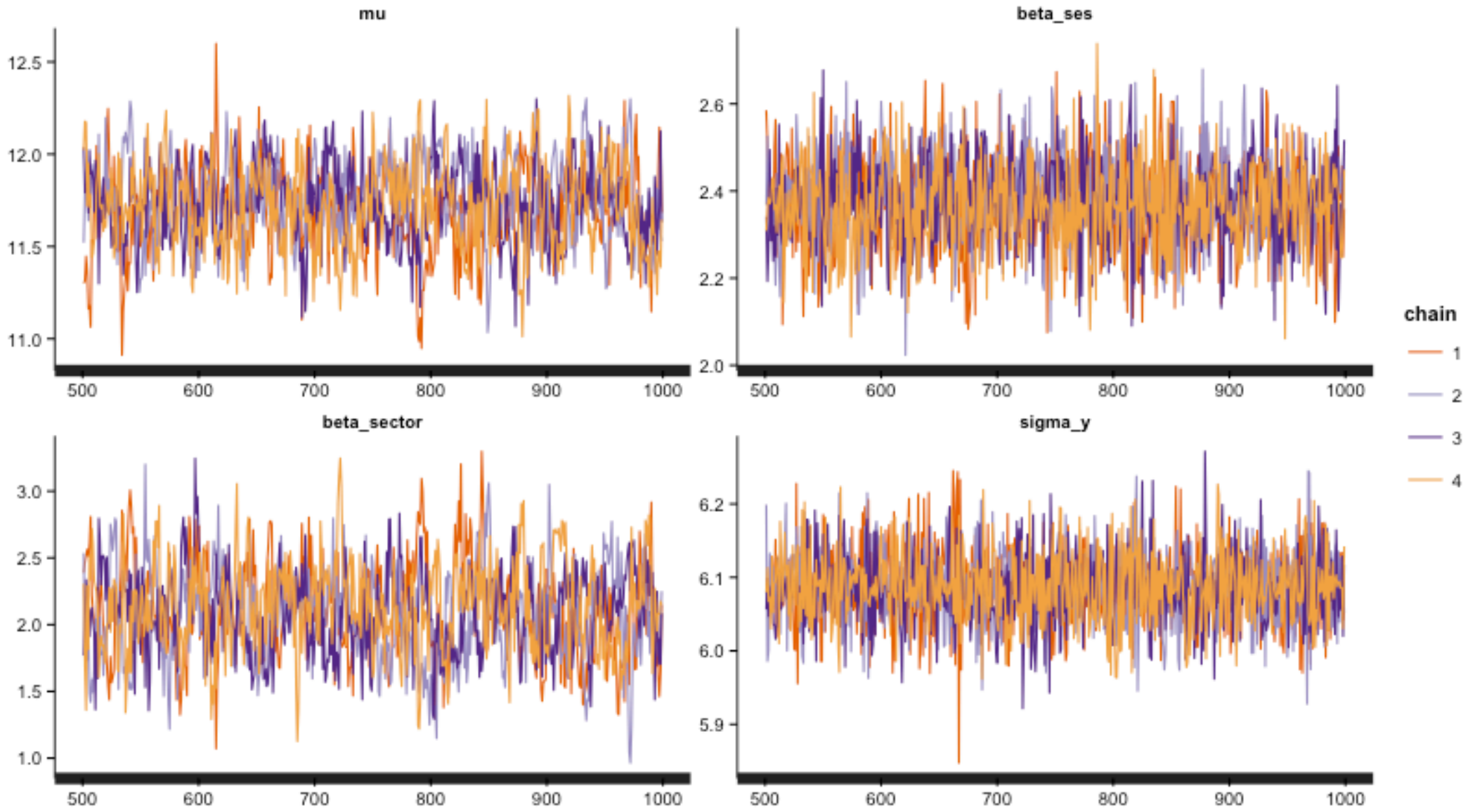
What we get

We get draws from our posterior. Each draw is a plausible value of our parameters:

```
> all_samps <- as.data.frame(model1mcmc)
> dim( all_samps )
[1] 2000  166
> all_samps[1,]
     mu beta_ses beta_sector sigma_y alpha[1] alp
1 11.8    2.338        2.175   6.135  -0.4022   0
```

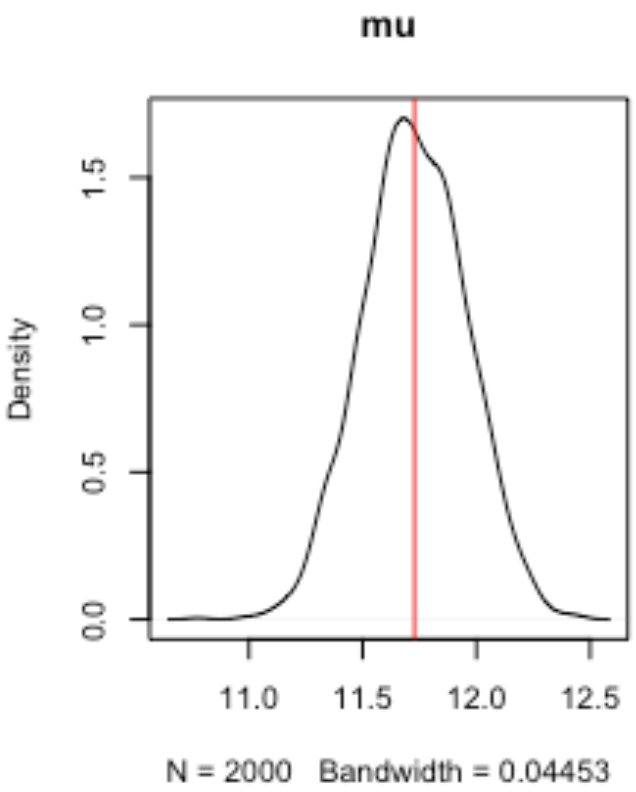
We then work with these plausible values to assess uncertainty and look at how the parameters are connected to each other.

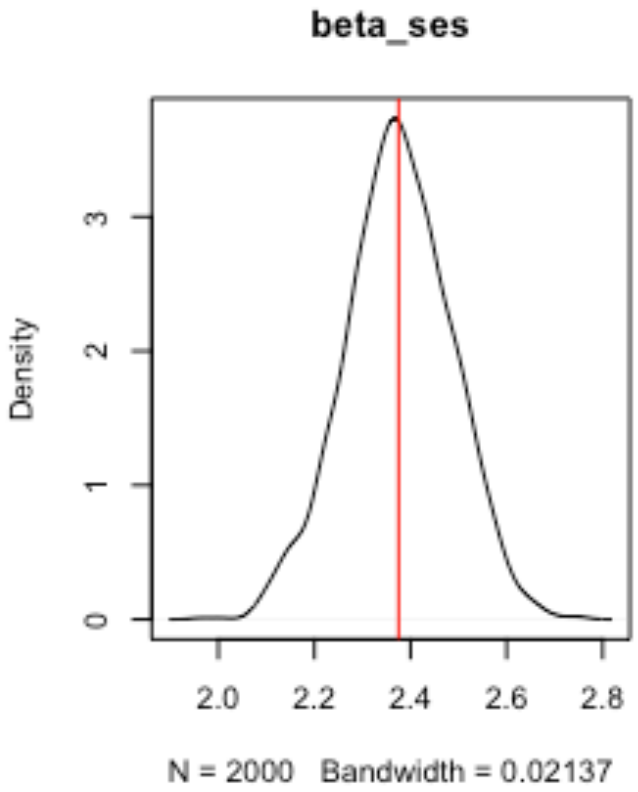
Traceplots to see if chains converged

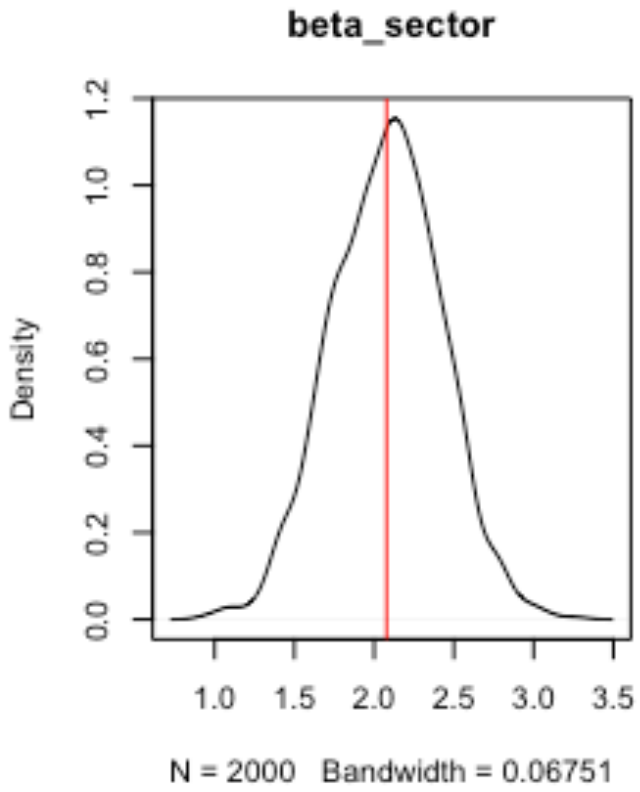


You want to see a mess and lots of overlap of the chains

Marginal Posteriors of your Parameters





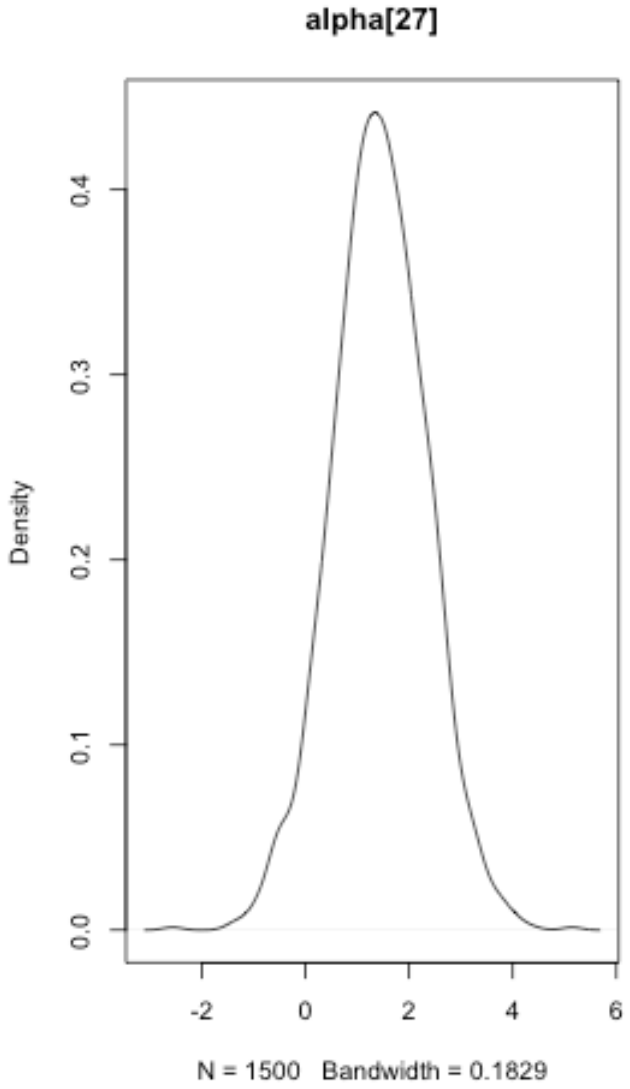


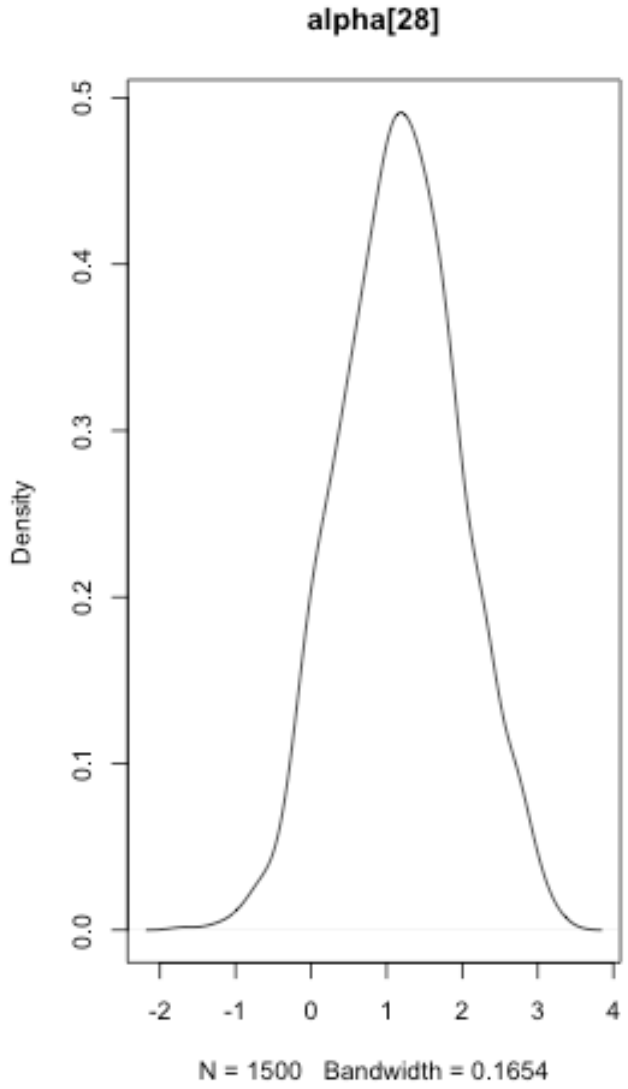
Here we have three of our main parameters.

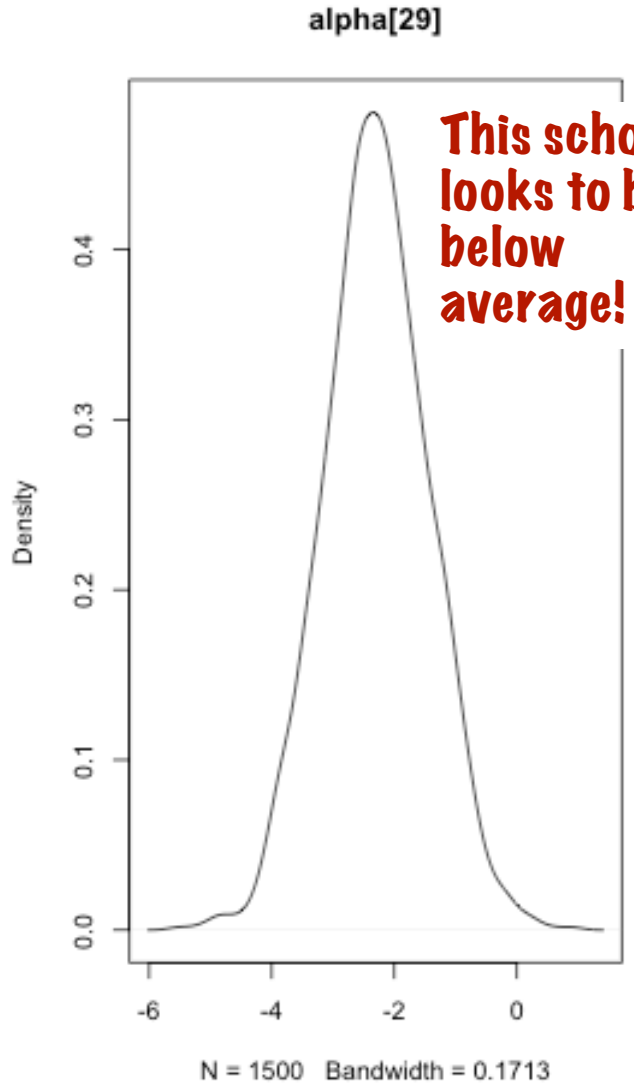
The peak suggests the most likely value (our estimate)

The width is the variability of the estimate.

Here are some individual schools







The individual school intercepts are estimated with noise, just like the other parameters

Calculating credible intervals

```
> pointAndInterval = function(samps, conf=0.95) {
+     trim.m=mean(samps,trim=0.1)
+     m = mean(samps)
+     med = median(samps)
+     cred = quantile(samps,c((1-conf)/2,1-(1-conf)/2) )
+     names( cred ) = c( "C.low", "C.high" )
+     trim.sd = sqrt(sum((samps-trim.m)^2)/(length(samps)
+     data.frame( mean=m, se=sd(samps), cred, median=med,
+        trim.mean=trim.m, trim.se=trim.sd )
+ }
```

The standard deviations of our simulation draws are kind of like standard errors

```
> map_df( draws, pointAndInterval )
           .id   mean       se  C.low C.high median trim.me
1           mu 11.728 0.22629 11.293 12.161 11.724    11.7
2     beta_ses  2.375 0.10991  2.148  2.583  2.374     2.3
3 beta_sector  2.078 0.34302  1.410  2.758  2.088     2.0
4      sigma_y  6.089 0.05424  5.980  6.197  6.089     6.0
5 sigma_alpha  1.933 0.14192  1.680  2.232  1.925     1.9
```

Comparison case: lmer fit

```
> M0 = lmer( mathach ~ ses + sector + (1|sid),
data=hsstud )
> display( M0 )
lmer(formula = mathach ~ ses + sector + (1 | sid), data
= hsstud)
             coef.est coef.se
(Intercept) 11.72     0.23
ses          2.37     0.11
sector       2.10     0.34

Error terms:
 Groups   Name          Std.Dev.
 sid      (Intercept)   1.92
 Residual               6.09
---
number of obs: 7185, groups: sid, 160
AIC = 46621.2, DIC = 46602
deviance = 46606.4
```

Compare our standard errors to the prior slide results. They are very similar.

But with Bayesian methods (prior slide) we ALSO get uncertainty on our variance parameters!

Why bother with Stan?

First notice we got uncertainty on *everything*, right off the bat.

But really the reasons are because you can write down more complex models such as:

- ★ Different variances for different groups
- ★ Nonnormal distributions for residuals or random effects
- ★ Weird interactions of random effects

Also, Bayesian inference arguably works better than MLM for small numbers of clusters because you can stabilize your estimates of the variance parameters.

Recap

What just happened?

Check In:
http://cs179.org/lecST2

A crash course on Bayesian logic

- ★ Much more to Bayesian machinery. Courses of it, in fact

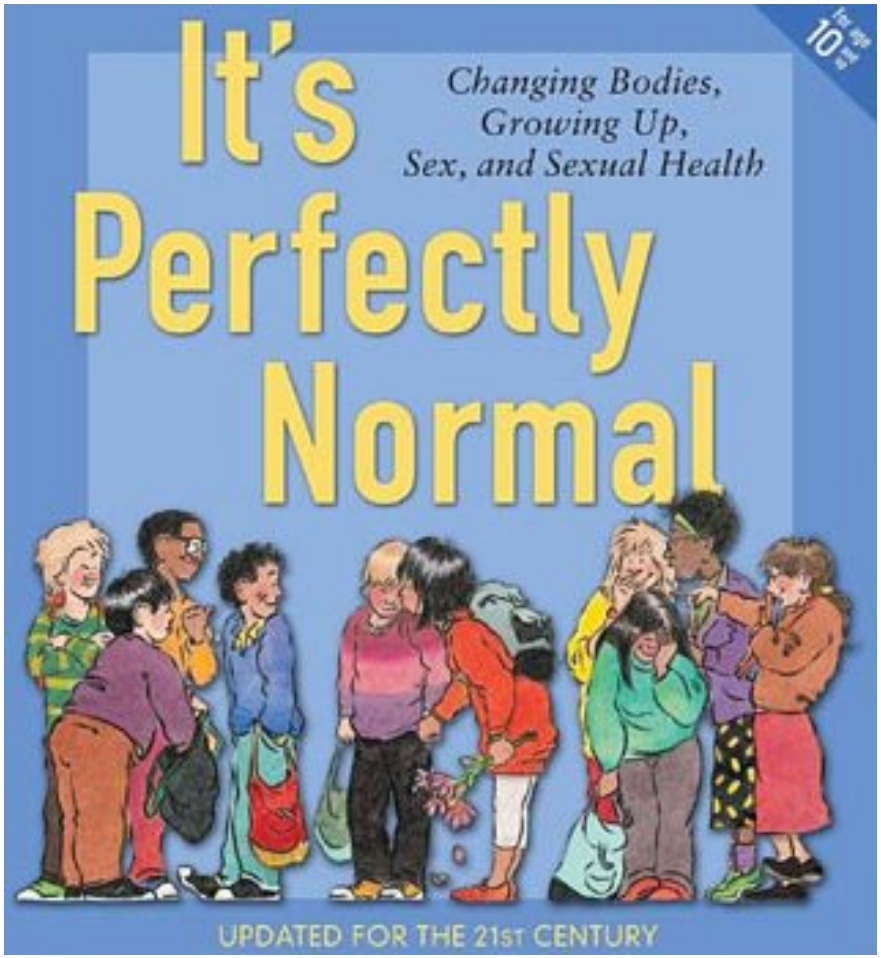
STAN: A Bayesian statistical computing package

- ★ You can write down MLMs as equations just as we are used to
- ★ This allows for easy changes of things, e.g., t-distribution on the residuals
- ★ Gateway to *very* complex models.
- ★ "Full Bayes" handles small clusters better.

Supplementary Slides

But where do priors come from?

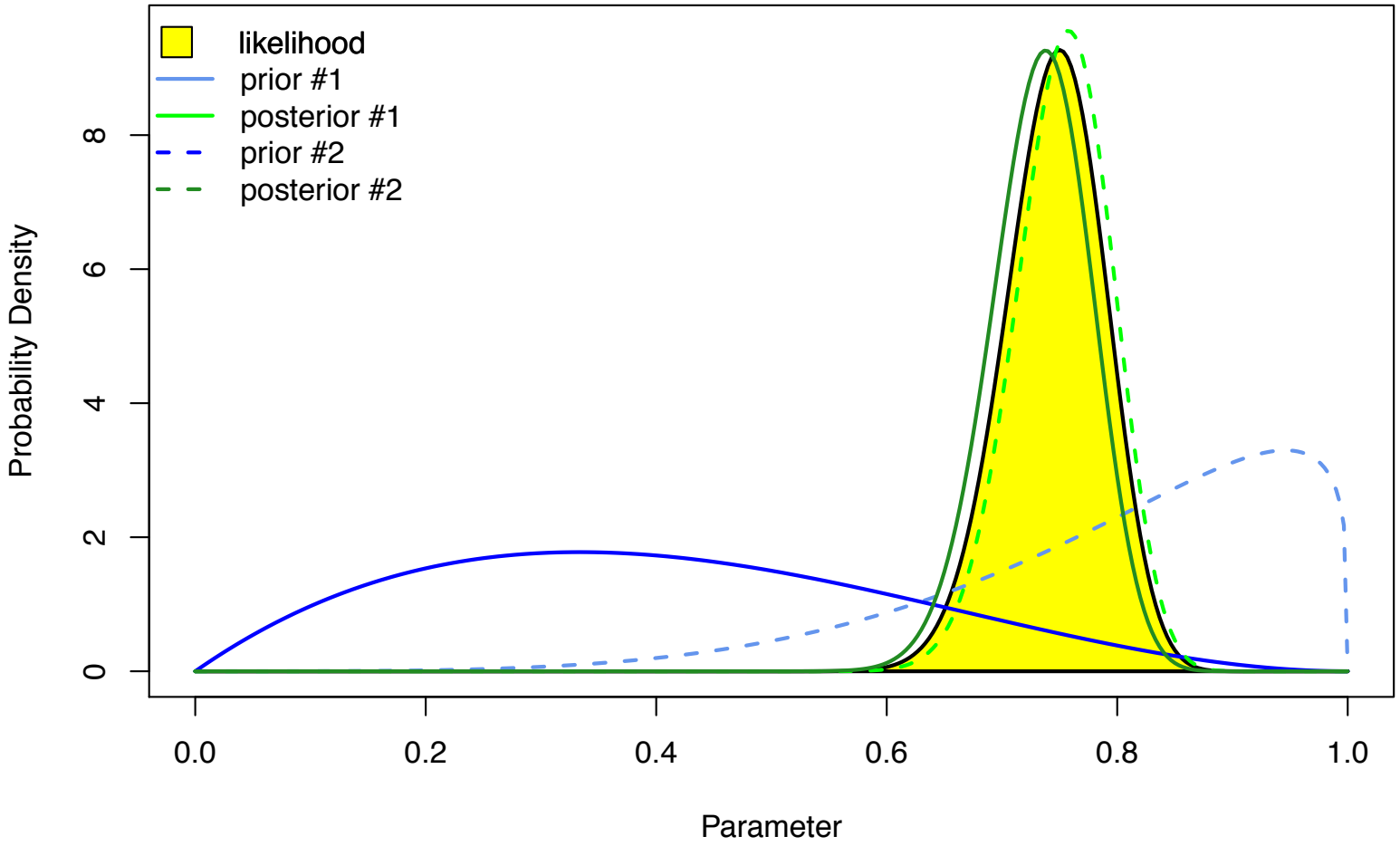
An important day at statistician-school?



There's nothing wrong, dirty, unnatural or even *unusual* about making assumptions – carefully. Scientists & statisticians all make assumptions... even if they don't like to talk about them.

When don't priors matter (much)?

When the data provide a lot more information than the prior, this happens



These priors (& many more) are *dominated* by the likelihood, and they give very similar posteriors – i.e. everyone agrees. (Phew!)

Uninformative Priors

Often these can turn Bayesian procedures into maximum likelihood procedures.

Issues:

- ★ Can be tricky to implement.
- ★ Get model instability.
- ★ Don't scale nicely.

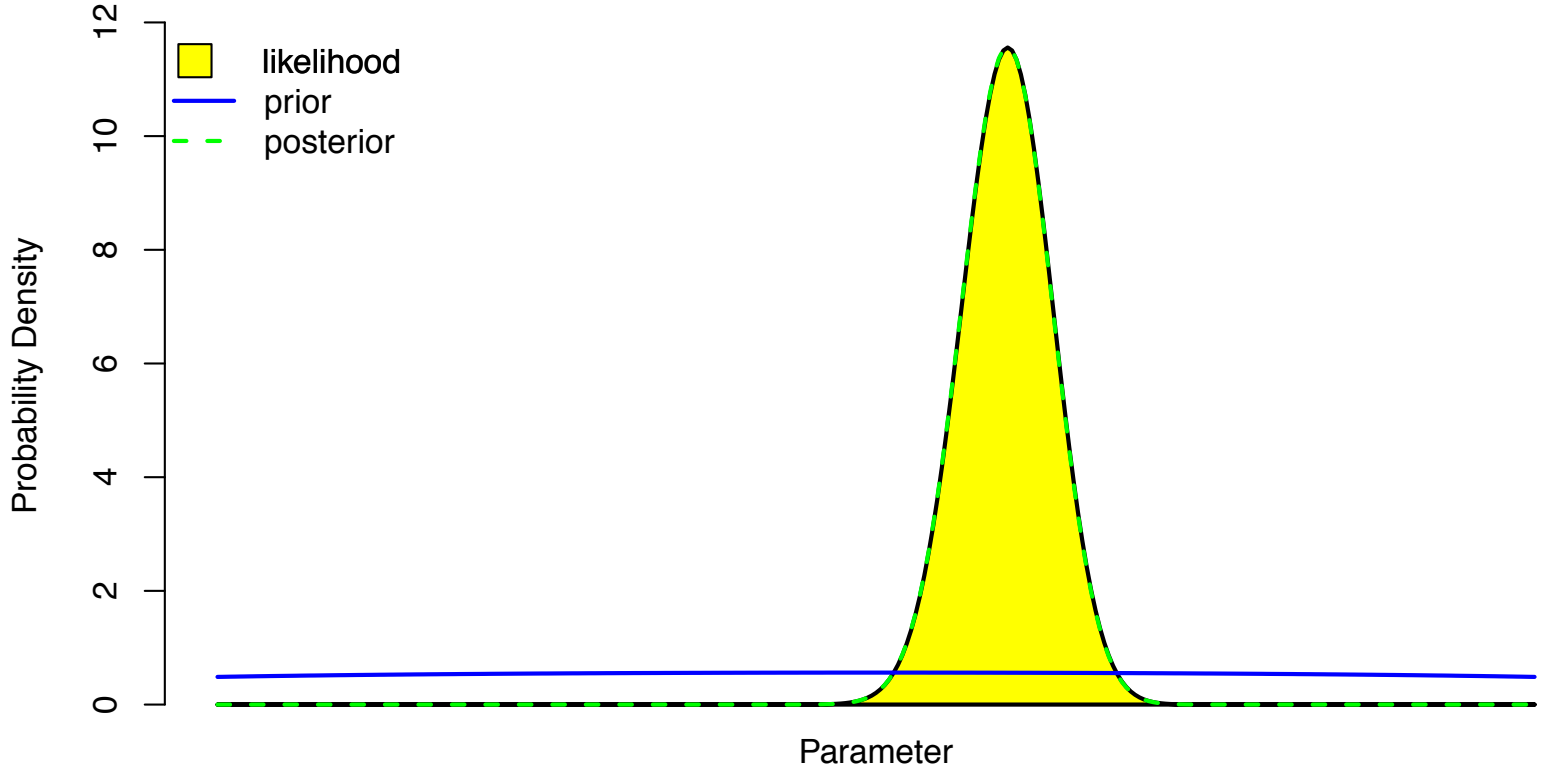
Example:

- ★ The "flat prior" on the mean of a constant (this is an *improper prior)*.

Uncertainty (the Posterior) will typically align with the classic normal approximation asymptotically.

When don't priors matter (much)?

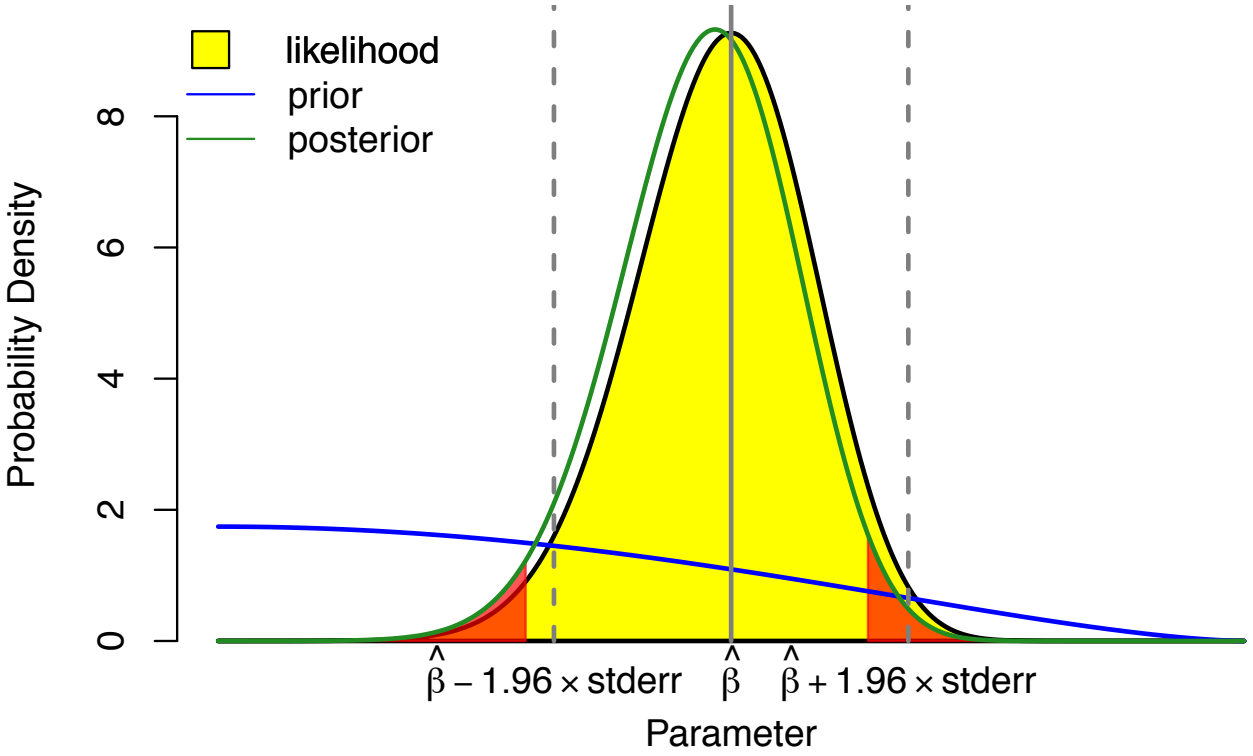
A related idea; try using very flat priors to represent ignorance;



- Flat priors do NOT actually represent ignorance! Most of their support is for *very* extreme parameter values
- For β parameters in '1st year' regression models, this idea works okay – it's more generally known as 'Objective Bayes'
- For many other situations, it doesn't, so use it carefully. (And also recall that prior elicitation is a useful exercise)

When don't priors matter (much)?

Back to having very informative data – now zoomed in;



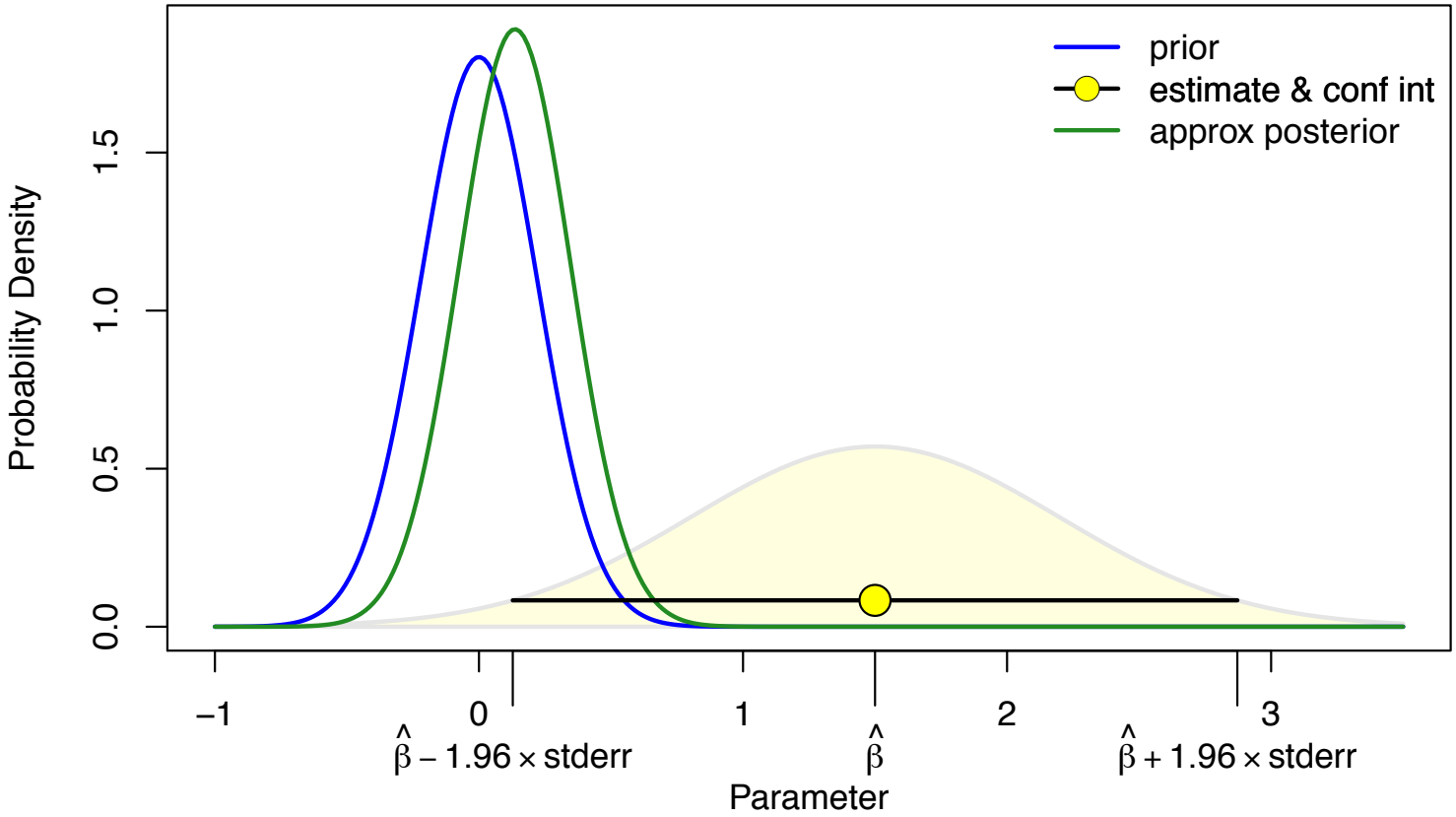
The likelihood *alone* (yellow) gives the classic 95% confidence interval. But, to a good approximation, it goes from 2.5% to 97.5% points of Bayesian posterior (red) – a 95% *credible* interval.

- With large samples*, sane frequentist confidence intervals and sane Bayesian credible intervals are essentially identical
- With large samples, it's actually *okay* to give Bayesian interpretations to 95% CIs, i.e. to say we have ≈95% posterior belief that the true β lies within that range.

* *and some regularity conditions*

When don't priors matter (much)?

Let's try it, for a prior strongly supporting small effects, and with data from an imprecise study;



- 'Textbook' classical analysis says 'reject' ($p < 0.05$, woohoo!)
- Compared to the CI, the posterior is 'shrunk' toward zero; posterior says we're sure true β is very small (& so hard to replicate) & we're unsure of its sign. So, hold the front page

Minimally Informative Priors

These priors are very “diffuse”

Example:

- ★ Saying you believe the the mean is somewhere in N(0, 100^2)
- ★ The high variance says you know it is not gigantic, but other than that you will not commit.

Associated uncertainty will also typically converge to frequentist intervals asymptotically.