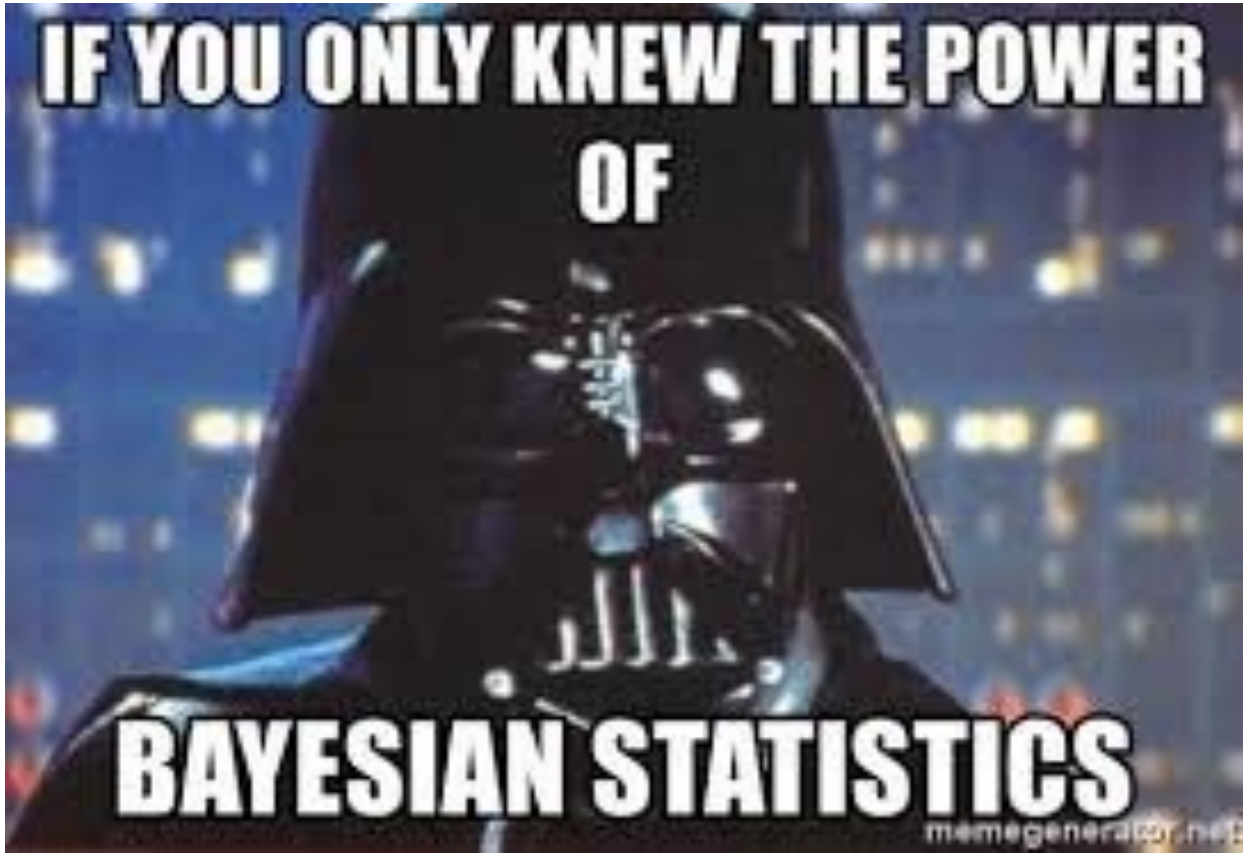
Getting set up for today: installations

```
install.packages('rstanarm')
install.packages('tidybayes')
install.packages('bayesplot’)
```

reach out if you have issues!

Regression: Bayesian Style

You may have heard a lot of things about Bayesian stats….



Myth 1: Bayesian modeling is harder than other kinds of statistics

- No more difficult than other kinds of stats, and even more intuitive for certain applications.
- Bayesian statistics have a reputation for being ‘hard’ or ‘obscure’ because most Bayesian software has been developed within the last 15-20 years
- Many faculty view Bayesian statistics as more difficult or obscure because they never learned early in their careers.

Myth 2: Bayesian statistics are only for 'advanced' or 'computational' analyses

- Most syntax for Bayesian regression modeling is *exactly the same* as the lm() syntax many R users already know
- Bayesian models vary in their complexity, just like other kinds of statistics
- R packages like the ones we'll learn today make Bayesian inference accessible for users of all levels of stats & programming experience

Myth 3: Bayesian statistics aren't for me

- Bayesian stats **are** for you, as long as you want to use them
- Especially within R, there is a big support community of Bayesian statistics users across many levels of expertise and backgrounds
- None of your instructors knew any Bayesian stats until other grad students showed them
- This summer, you'll get support from us in applying Bayesian stats to your work if you want to

Why use Bayesian models?

Sooner or later, frequentist models *will* fail you:

- By refusing to run at all
- By giving you an answer that is qualitatively wrong (a significant effect when it shouldn't be)

lme4/lme4

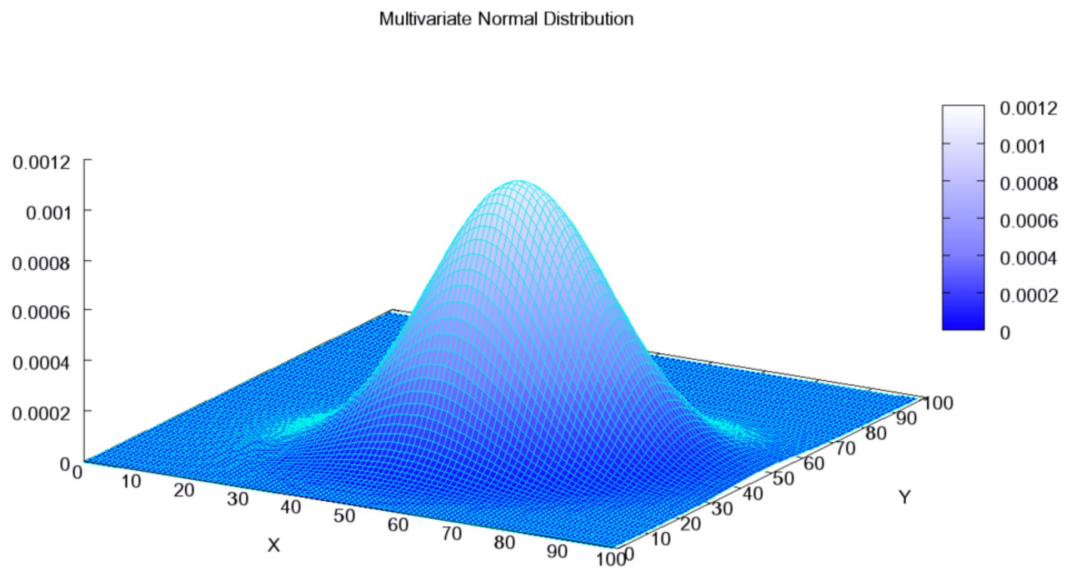
#489 **Model failed to converge with max|grad|**

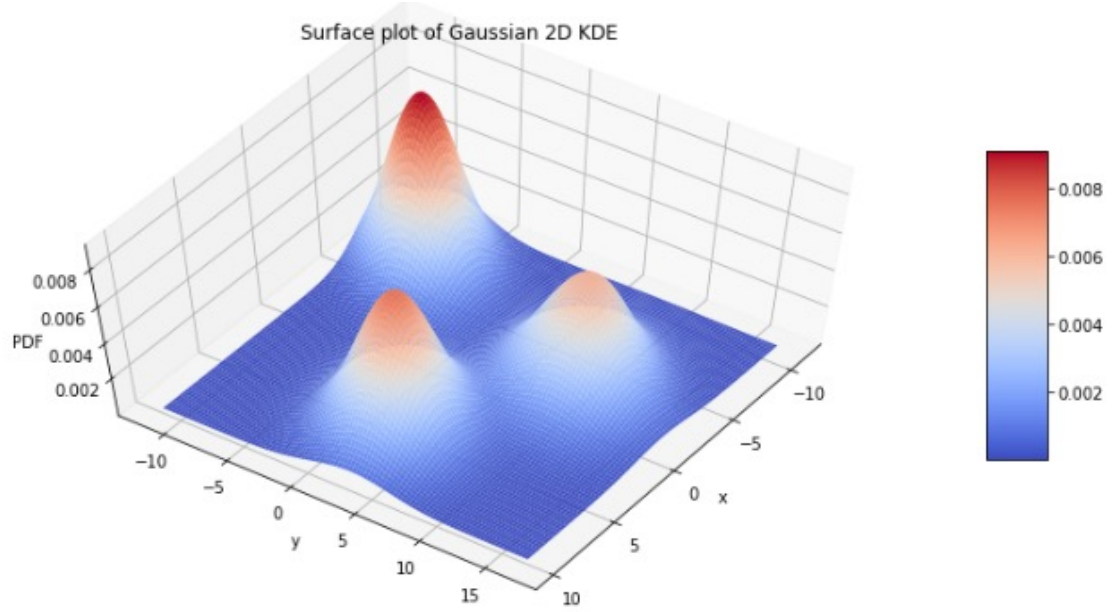
5 comments

Valabe opened on October 29, 2018

The mountain problem

Assuming a normal distribution will be fine for the mountain on the left, but NOT for the one on the right

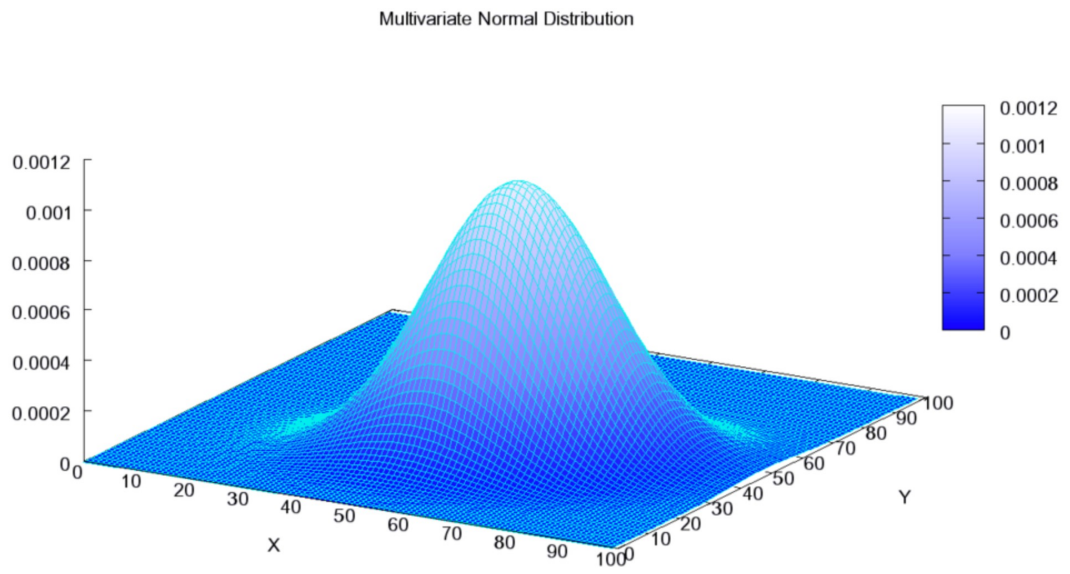


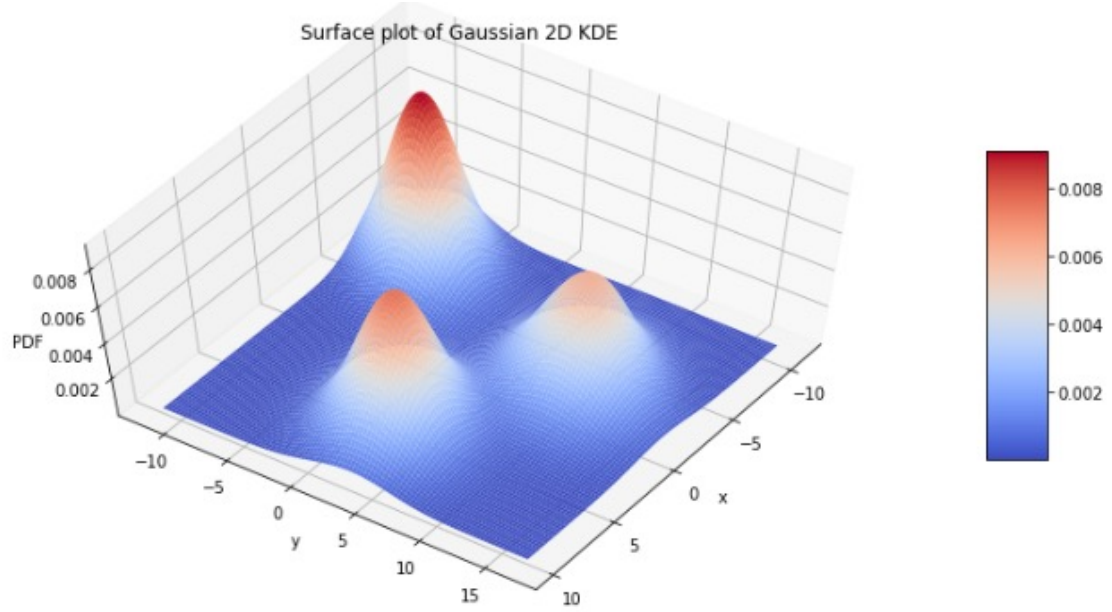


Bayesian models solve the mountain problem by using *sampling*

No assumed mountain shape, draws many sample ‘iterations’ to characterize the full shape of the mountain

You can think of the models we try today as taking 4000 ‘samples’ of the terrain to figure out the mountain shape

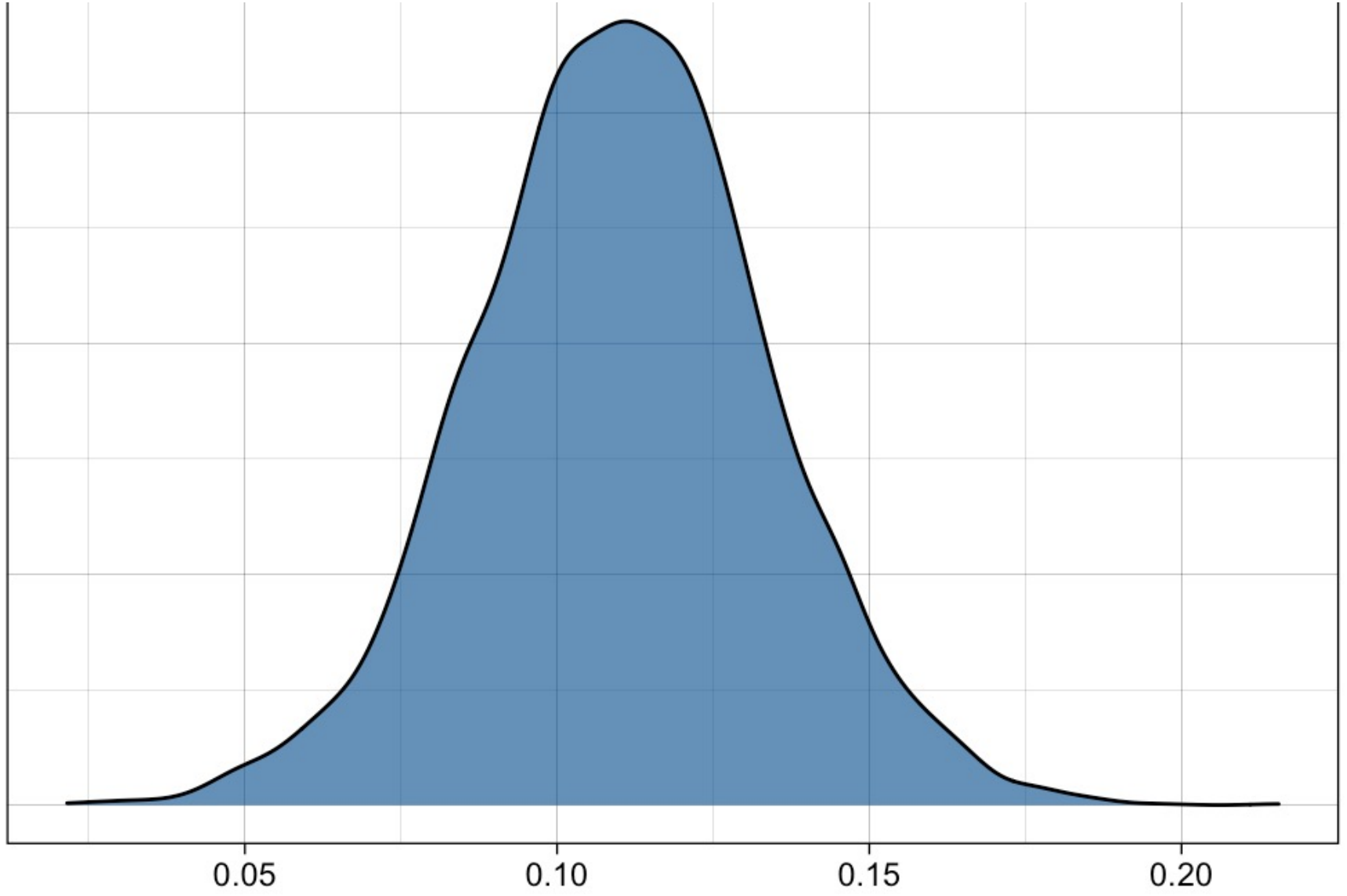




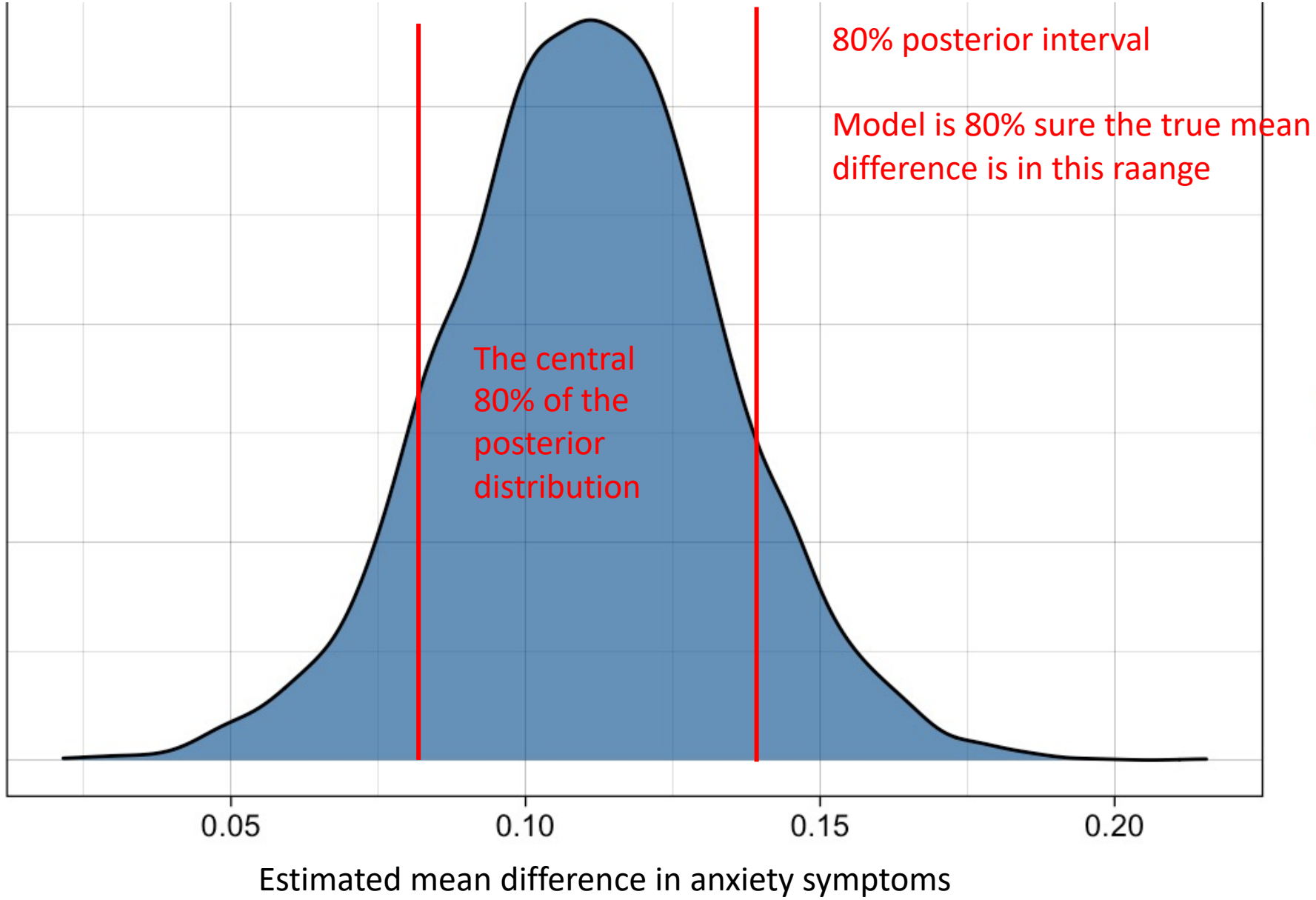
Why use Bayesian models?

- Bayesian models give us *more information* in the form of a full *posterior distribution* for the thing we want to know
- Instead of just a mean and standard error like lm()
- Bayesian stats don’t use p-values!

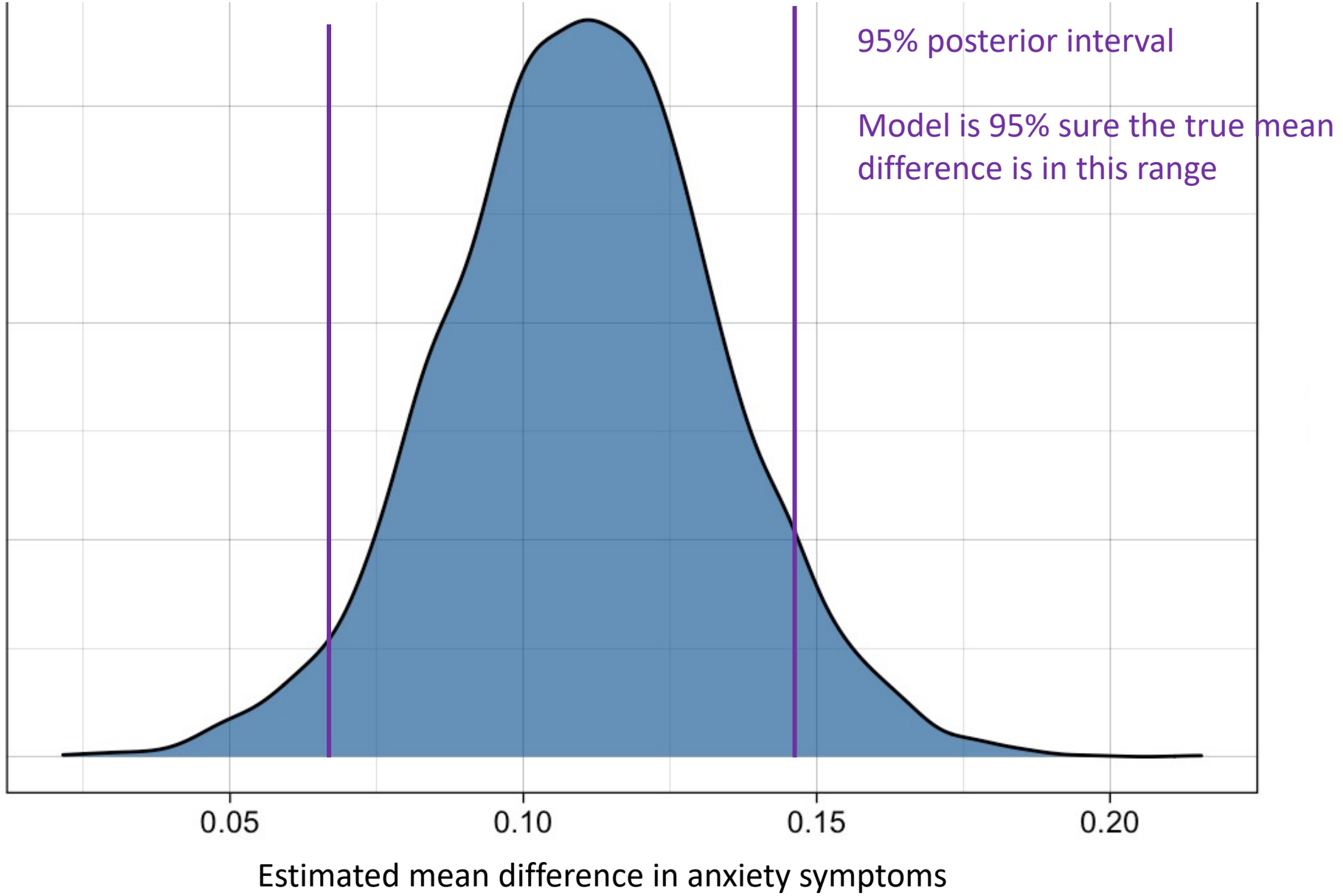
Posterior distribution for the average difference in anxiety symptoms between the group A > group B



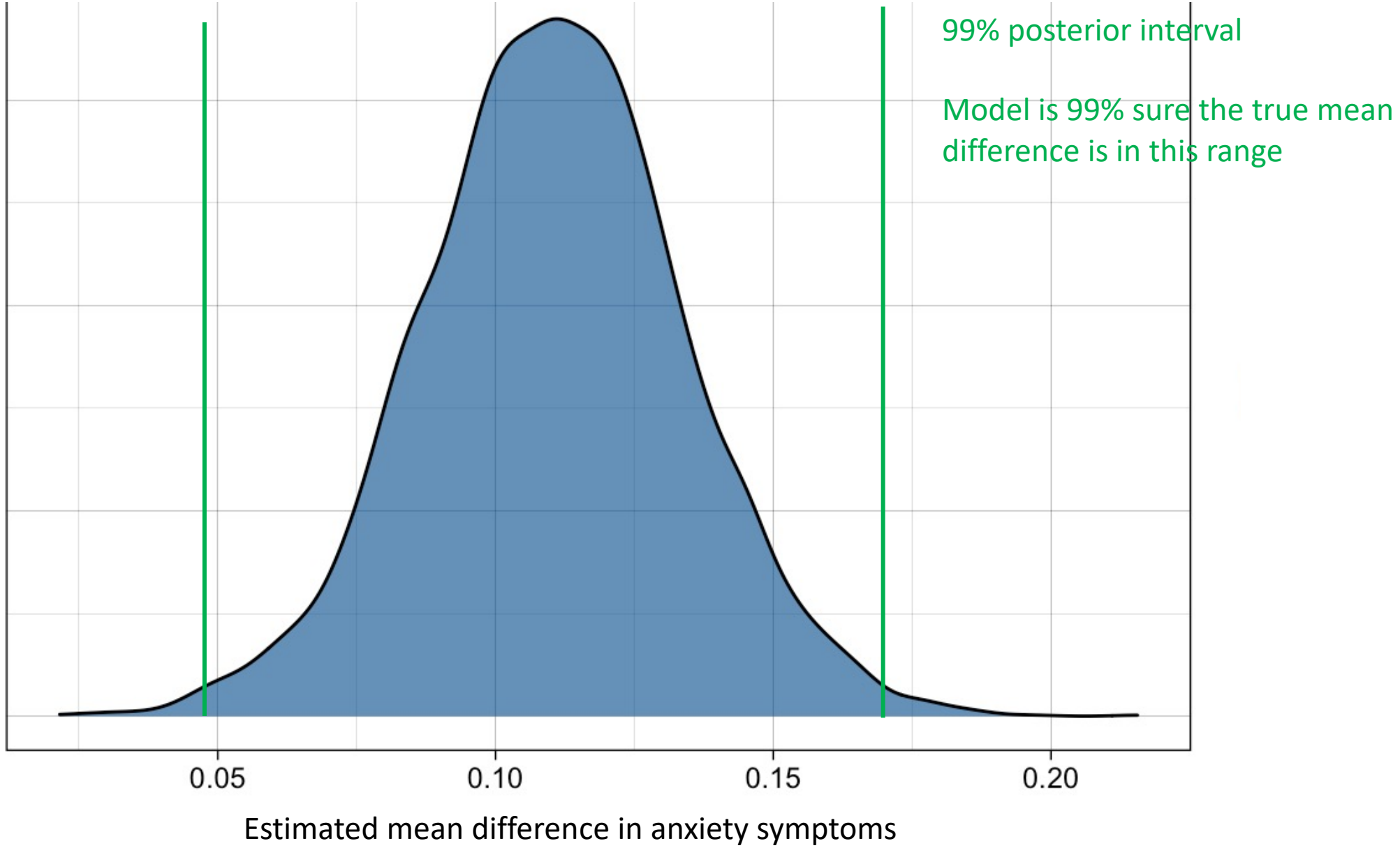
Posterior distribution for the average difference in anxiety symptoms between the group A > group B



Posterior distribution for the average difference in anxiety symptoms between the group A > group B



Posterior distribution for the average difference in anxiety symptoms between the group A > group B



No p-values?! How will we make decisions?

- With posterior intervals!
- With Bayesian models, one way we can set decision thresholds is by asking if *posterior intervals include 0*
- If a posterior interval contains only values of one sign (i.e. all positive, or all negative), we could be more confident that the parameter is in that direction (i.e. the 'effect' is positive)

The estimated mean difference was 0.12 (95% PI [0.05, 0.18])

What are your questions?

Let’s code!