Bayesian Data Analysis, Part 1

Clay Ford

Spring 2024

Agenda

- Conceptual overview of basic Bayesian data analysis
- How to run a Bayesian analysis using the `rstanarm` package
- How to assess convergence and model fit

Motivating example

Estimate the mean weight of students' backpacks at UVa.

- Take a random sample of 100 students
- Weigh their backpacks in lbs.
- I calculate the mean to be 16.1 lbs.

It's just an estimate. If I take another sample, I'll get a different estimate. Say 19.1. Or 16.8. I want to understand the uncertainty in my estimate.

The traditional approach

Assume there is some fixed but unknown mean backpack weight. If we could see all and know all, we could estimate the "true" mean. We want to estimate that true mean.

- Use 16.1 as our estimated mean[1]
- calculate a 95% confidence interval: $(15.2, 16.9)$
- We might say we're 95% confident the true mean is between 15.2 and 16.9

If we were to repeat this process many, many times, 95% of the confidence intervals would contain the true value. (NOTE: A confidence interval is NOT a probability interval.)

[1] We also have to estimate the standard deviation in order to calculate the traditional confidence interval.

The Bayesian approach

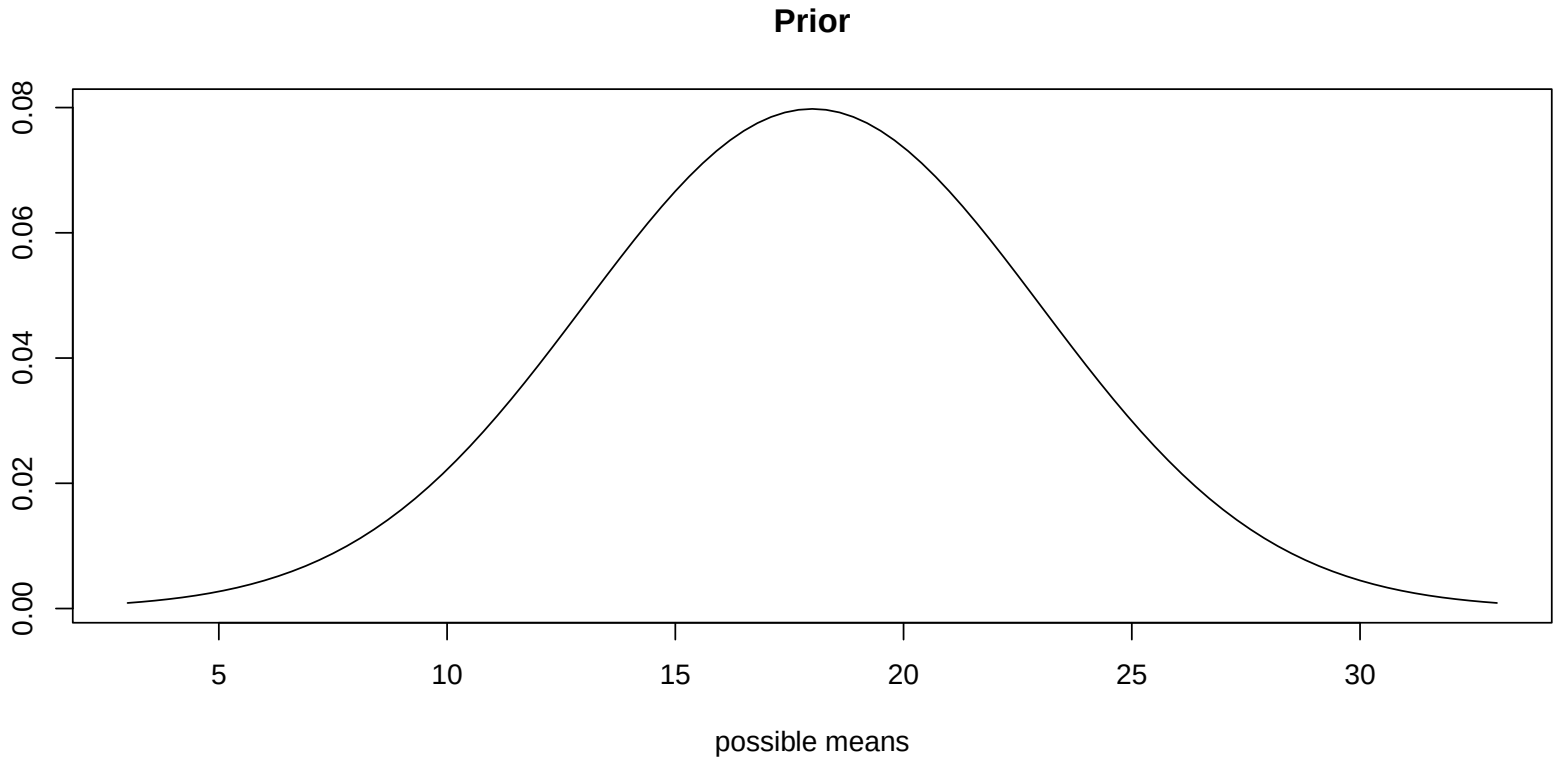
Instead of estimating some "true" mean, we instead estimate the *probability distribution* of possible means.

- begin with a *prior* distribution of how we believe the means are distributed
- update the prior distribution using the data we collected to obtain a *posterior* distribution using *Bayes' theorem*
- use the posterior distribution to describe the mean of students' backpacks

The "estimate" is the *entire* posterior distribution.

Example of a prior distribution

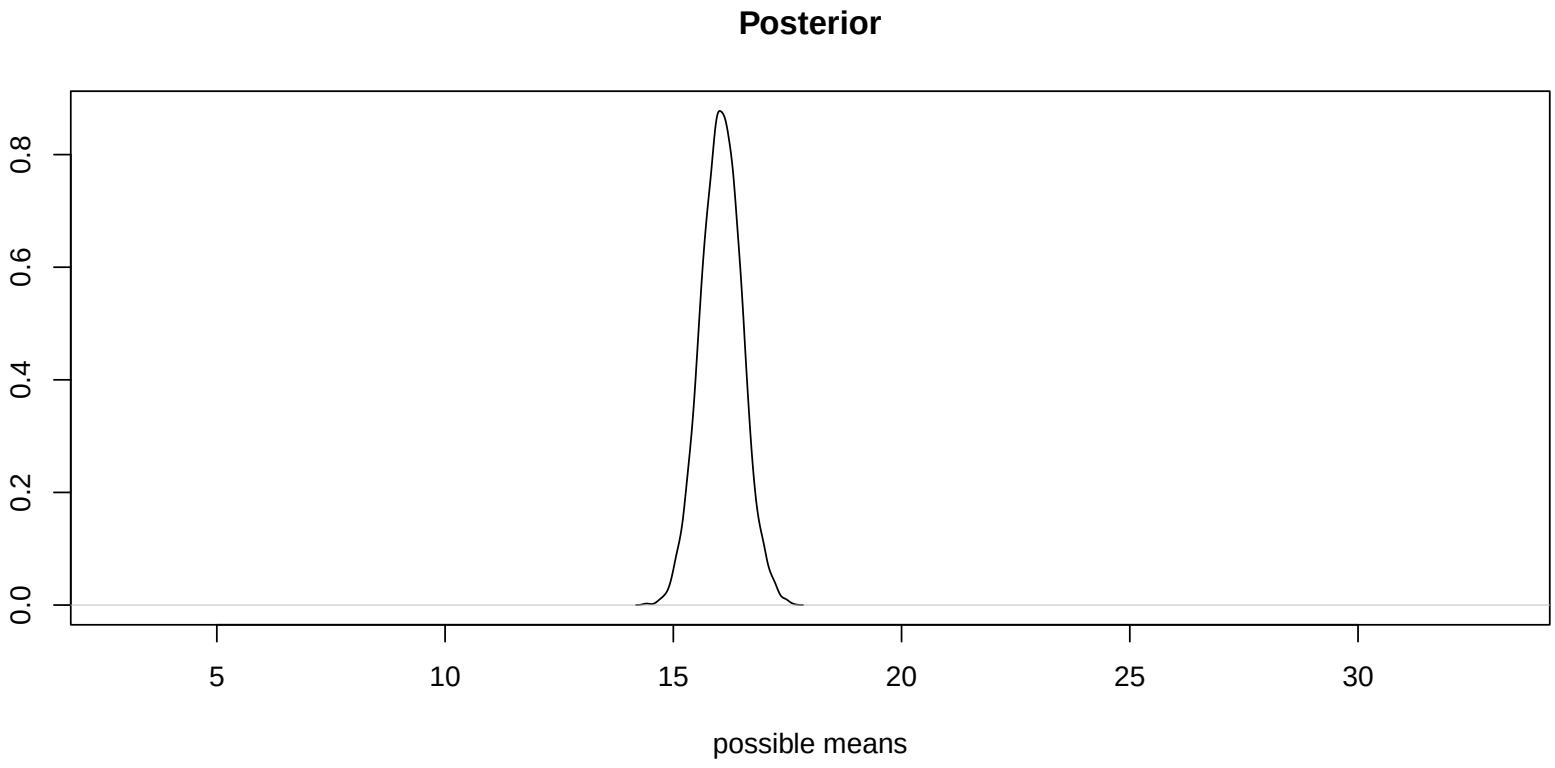
Prior to our analysis, perhaps we think the average backpack weighs around 18 lbs[2]. A possible prior distribution might be a $N(18, 5)$.



[2] https://well.blogs.nytimes.com/2009/07/21/weighing-school-backpacks/

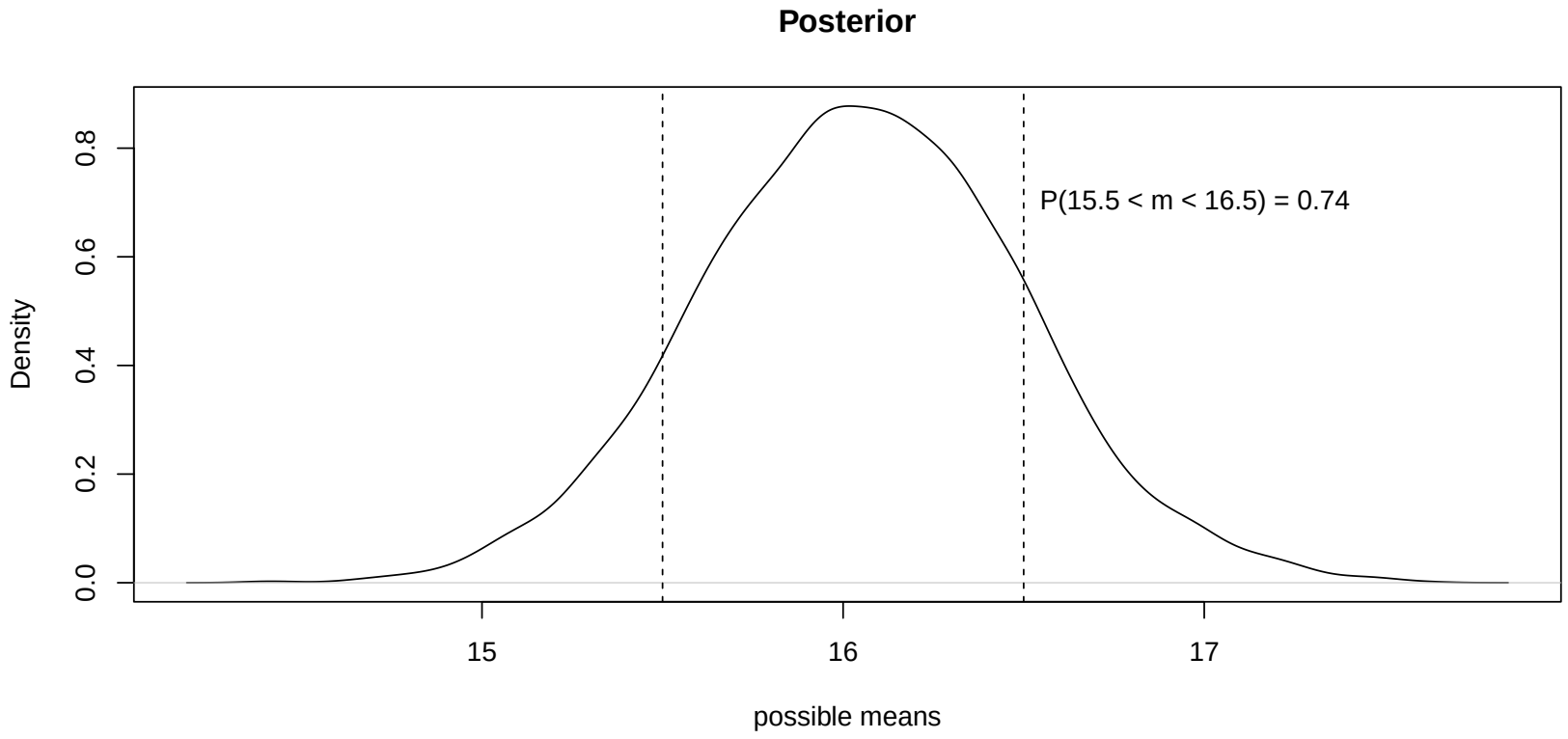
Example of a posterior distribution

After observing our data we use *Bayes' theorem* to update our prior distribution to get a posterior distribution.



Example of using posterior distribution

There's about 74% probability that the mean backpack weight is between 15.5 and 16.5 lbs.



Another motivating example

Examine two brands of rechargeable batteries. How long do they run (in hours) before exhausted?

- Take 25 samples of each brand and calculate mean time
- Calculate difference between means: 0.86 hours
- Appears one brand is superior

If I do this again, will I get the same result? Or is this due to chance? How certain are we about the 0.86 estimate?

The traditional approach

Assume there is some fixed but unknown difference in means. If we had access to all batteries and unlimited time, we could estimate the true difference in means.

- Use 0.86 as our estimate
- calculate a 95% CI of difference: $(0.39, 1.34)$
- We might say we're 95% confident the true difference is between 0.39 and 1.34

If we were to repeat this process many, many times, 95% of the confidence intervals would contain the true value. (Reminder: A CI is not a probability interval.)

We might also do a t-test to estimate the probability of getting a difference as large (or larger) if the true difference was 0 (ie, a *p-value*).

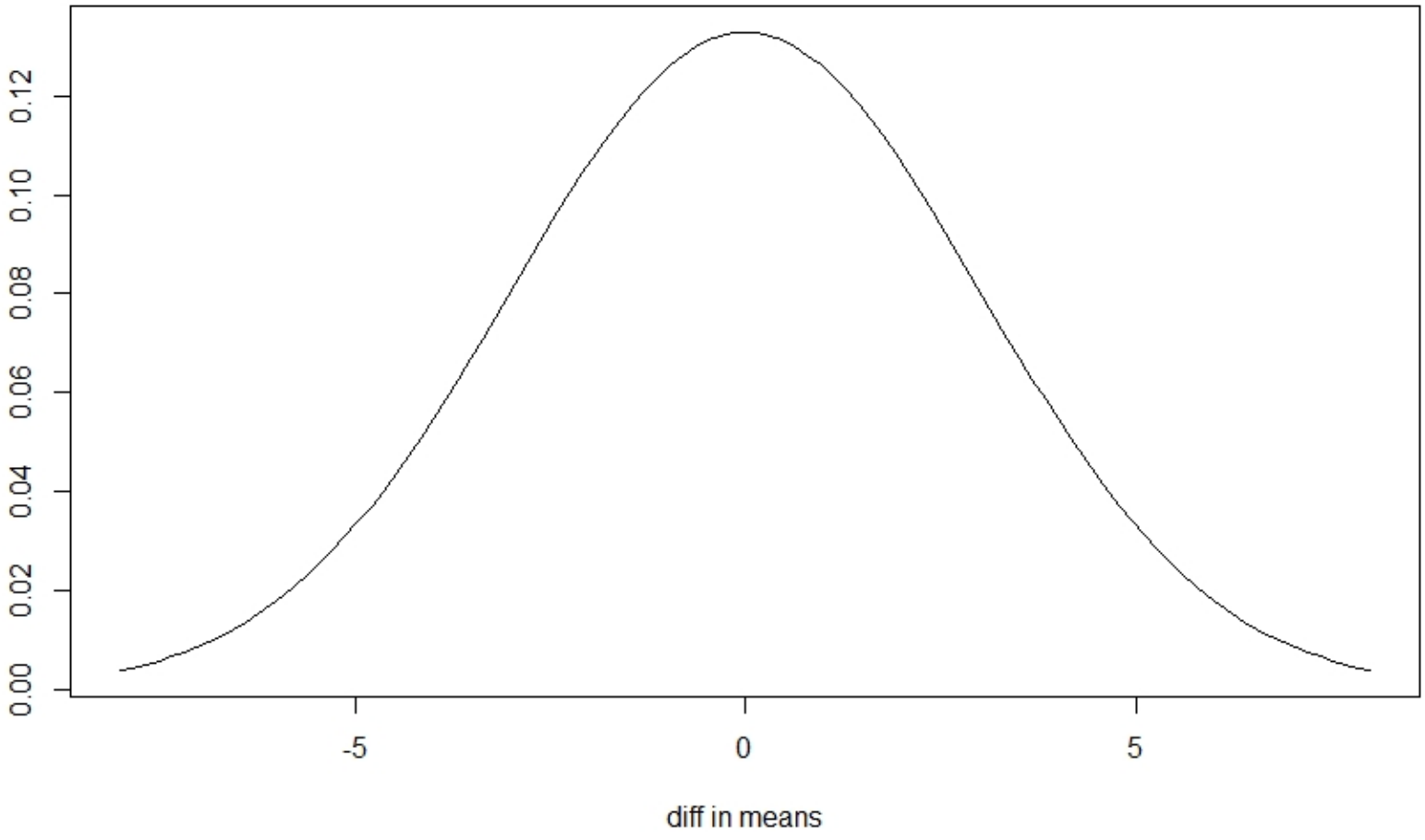
The Bayesian approach

Instead of estimating some "true" value, we instead estimate the *probability distribution* of possible differences.

- begin with a *prior* distribution of how we believe the differences are distributed
- update the prior distribution using the data we collected to obtain a *posterior* distribution using *Bayes' theorem*
- use the posterior distribution to describe the difference in brands

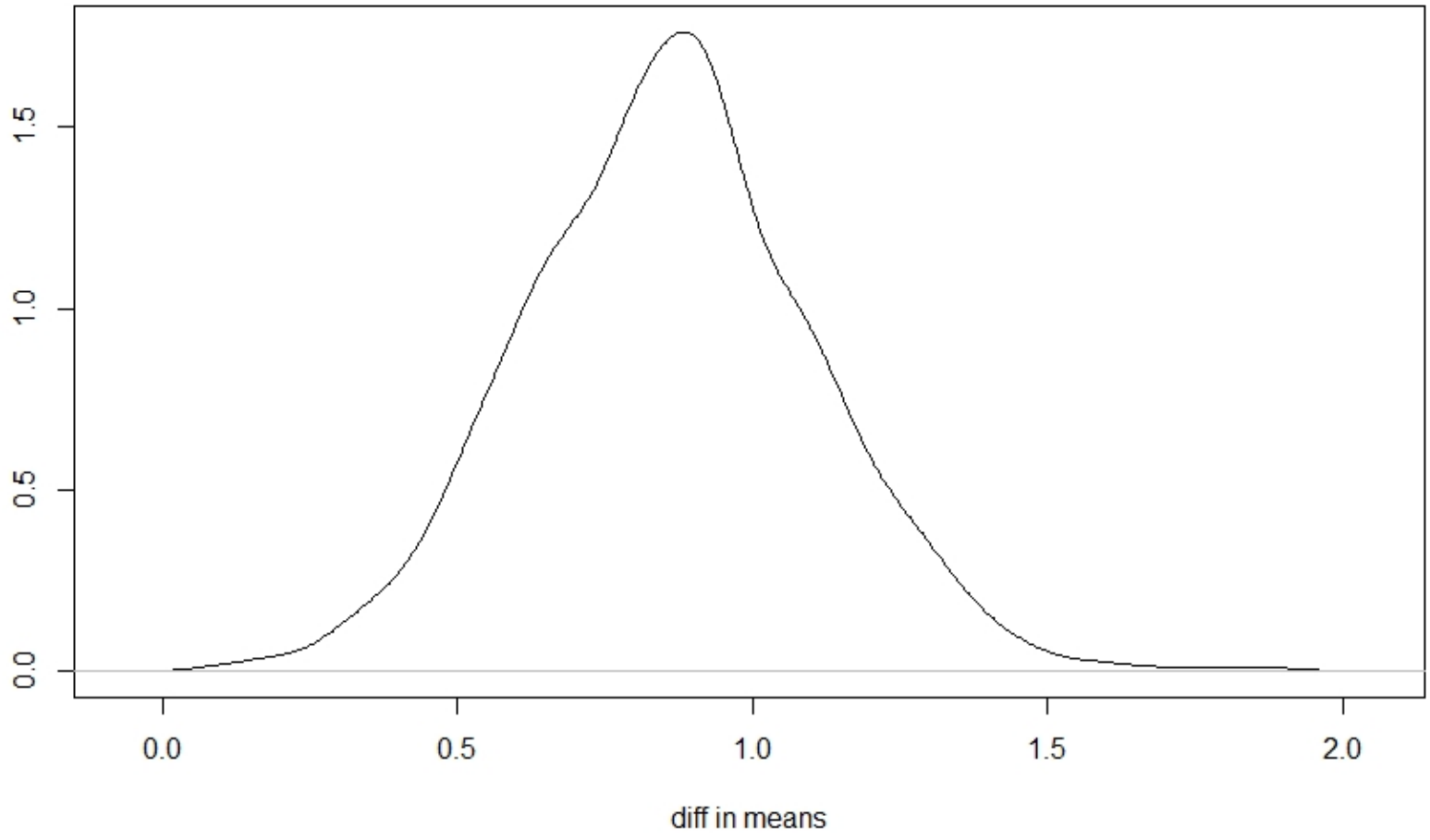
Example of a prior distribution

Perhaps we're not sure which brand is better. A possible prior distribution might look like this, a $N(0, 3)$ distribution.



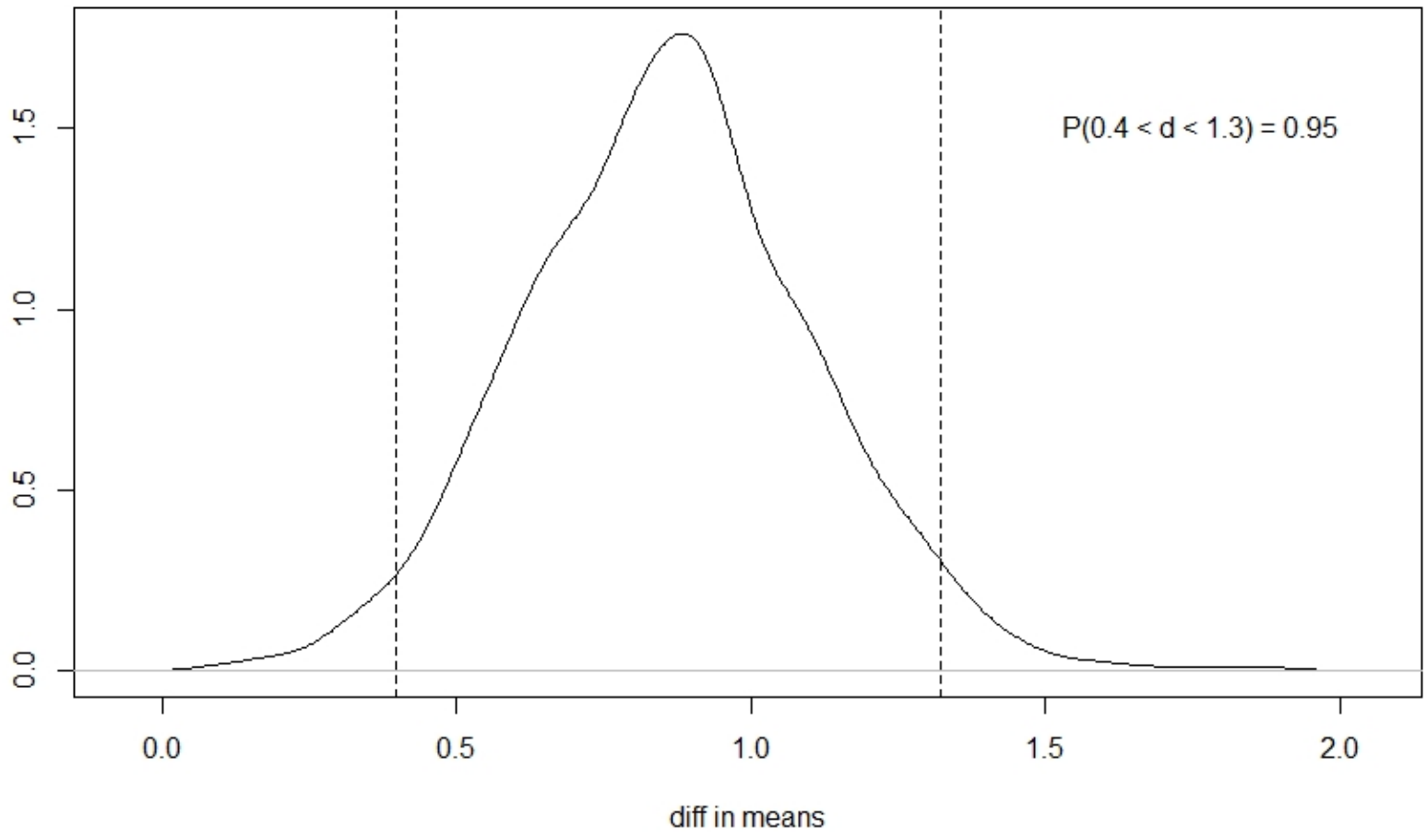
Example of a posterior distribution

After observing our data we use *Bayes' theorem* to update our prior distribution to get a posterior distribution.



Example of using posterior distribution

There's about a 95% probability the mean difference is between 0.4 and 1.3.



Bayes' theorem in words

We use Bayes' theorem to update our prior distribution to get a posterior distribution.

$$\textit{posterior} = \frac{\textit{likelihood} \times \textit{prior}}{\textit{average likelihood}}$$

- prior is how we believe a *parameter* of interest is distributed
- likelihood is the probability of observed data given a certain parameter, eg. $P(\text{data}|\mu = 17)$
- posterior is the updated prior
- average likelihood ensures the posterior is a distribution with area that sums to 1 (ie, a probability distribution)

Using Bayes' theorem in real life

- Using Bayes' theorem mathematically is impractical for all but the simplest problems
- Instead we use something called *Markov chain Monte Carlo (MCMC)*
- MCMC draws samples from the posterior distribution
- This requires computing time, sometimes lots of time
- The goal today is show you how to do this in R using the `rstanarm` package

The `rstanarm` package

- R + Stan + arm
  - Stan: platform for statistical modeling
  - arm: applied regression modeling
- *"The primary target audience is people who would be open to Bayesian inference if using Bayesian software were easier but would use frequentist software otherwise."*
- Allows us to specify Bayesian models using traditional R modeling syntax
- `rstanarm` generates Stan code for us, and then uses the `rstan` package to run the model

Traditional statistical approach for backpack data

Estimate the mean and calculate a confidence interval using `t.test`

```
t.test(dat$backpacks)
```

```
##
##  One Sample t-test
##
## data:  dat$backpacks
## t = 37.061, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  15.1907 16.9093
## sample estimates:
## mean of x
##     16.05
```

Traditional modeling approach for backpack data

We can also analyze the backpack data using an intercept-only linear model:

```
mod1 <- lm(backpacks ~ 1, data = dat)
confint(mod1)
```

```
##                 2.5 %  97.5 %
## (Intercept) 15.1907 16.9093
```

This is an important concept because doing Bayesian statistics requires us to frame our analysis as a model.

Traditional modeling approach for backpack data

In addition to `lm`, we can use `glm` with `family = gaussian`. GLM = Generalized Linear Model.

```
mod1 <- glm(backpacks ~ 1, data = dat,
            family = gaussian)
confint(mod1)
```

```
##    2.5 %  97.5 %
## 15.2012 16.8988
```

Bayesian modeling approach for backpack data

```
library(rstanarm)
bm1 <- stan_glm(backpacks ~ 1, data = dat,
                family = gaussian,
                prior_intercept = normal(18, 5))
```

Instead of a confidence interval we obtain a Bayesian uncertainty interval:

```
posterior_interval(bm1, prob = 0.95,
                   pars = "(Intercept)")
```

```
##                   2.5%    97.5%
## (Intercept) 15.17081 16.96359
```

Traditional statistical approach for battery data

Calculate a confidence interval using `t.test`

```
bat <- read.csv("data/batteries.csv")
t.test(y ~ grp, data = bat)
```

```
##
##  Welch Two Sample t-test
##
## data:  y by grp
## t = -3.6472, df = 44.32, p-value = 0.000694
## alternative hypothesis: true difference in means betwee
## 95 percent confidence interval:
##  -1.3406537 -0.3864785
## sample estimates:
## mean in group 0 mean in group 1
##        10.16867        11.03223
```

Traditional modeling approach for battery data

Can also analyze the data using a linear model

```
lm.out <- lm(y ~ grp, data = bat)
confint(lm.out, parm = "grp")
```

```
##          2.5 %  97.5 %
## grp 0.3875019 1.33963
```

Or using glm

```
glm.out <- glm(y ~ grp, data = bat, family = gaussian)
confint(glm.out, parm = "grp")
```

```
##     2.5 %    97.5 %
## 0.3994994 1.3276329
```

Bayesian modeling approach for battery data

```
bm.out <- stan_glm(y ~ grp, data = bat,
                   family = gaussian,
                   prior = normal(0,3))
```

Instead of a confidence interval we obtain a Bayesian uncertainty interval:

```
posterior_interval(bm.out, prob = 0.95,
                   pars = "grp")
```

```
##             2.5%    97.5%
## grp 0.3868503 1.331482
```

Prior distributions

- Bayesian Data Analysis requires we provide prior distributions.
- Seems burdensome but it allows us to incorporate prior knowledge into our model.
- Is there a "correct" prior? No. But some are better than others.
- Different but reasonable priors will often lead to the same basic conclusion.
- We can vary priors and repeat the analysis to to see how different states of initial information influence the outcome.
- The smaller the sample size, the larger the effect a prior will have on the inference.

Prior distributions in `rstanarm`

- `rstanarm` provides *weakly informative* priors by default. (ie, rules out extreme positive or negative values)
- *Uninformative priors* place equal weight on all possible values (not recommended)
- The types of priors and how you set them vary from model to model; for example `stan_lm` vs `stan_glm`
- Using the default priors usually leads to similar results obtained by traditional modeling
- To see the priors, call the `prior_summary` function on the model object

Let's go to the R script and work with our examples.

Checking sampling quality

When doing Bayesian data analysis with MCMC sampling, we want to confirm that the sample we obtained describes the posterior.

We should do this before interpreting results and making inferences.

There are several diagnostics that can help with this.

Thankfully `rstanarm` provides warnings when sampling is suspect.

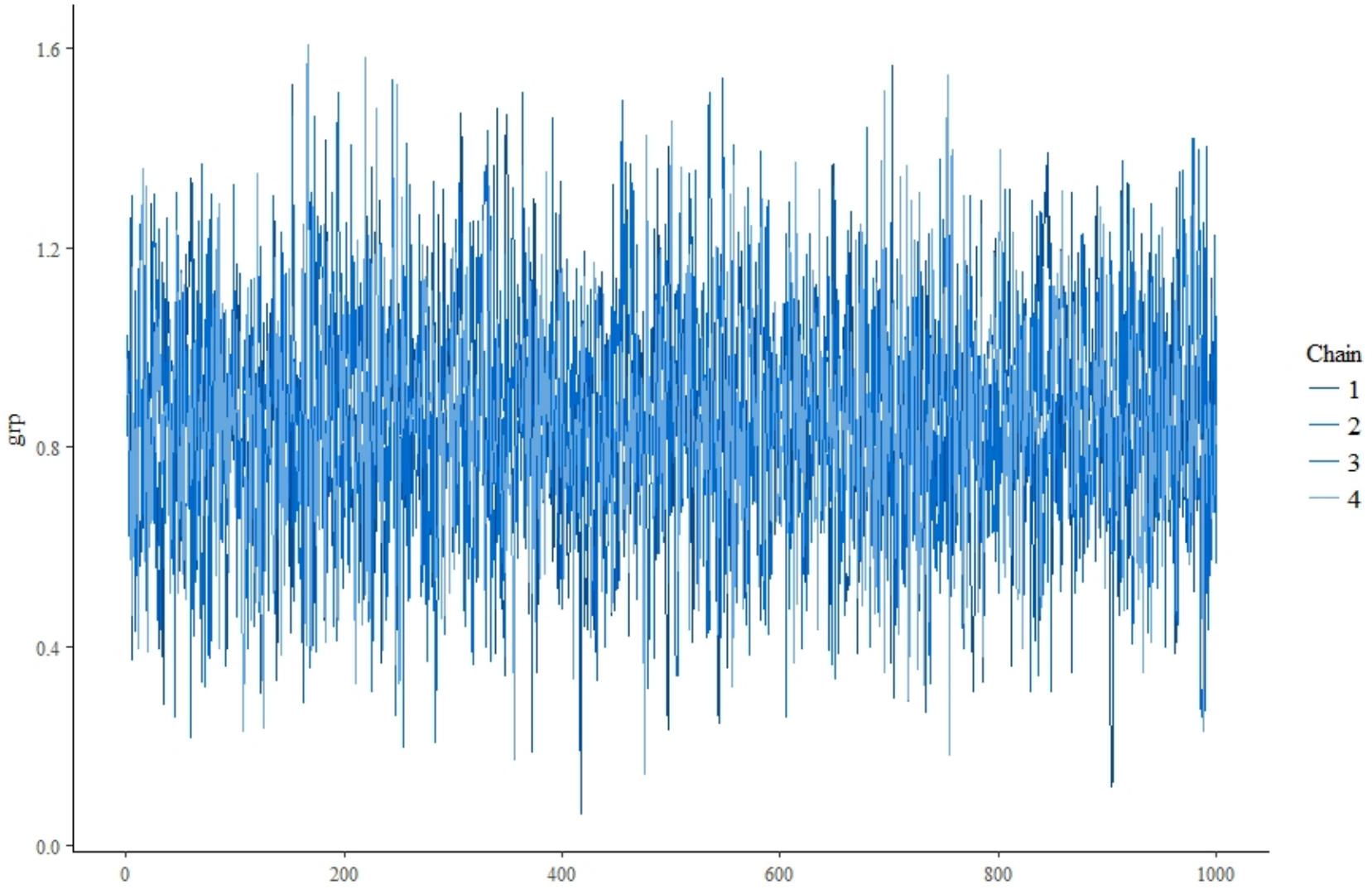
Visual assessment of convergence

- MCMC uses *Markov chains* to explore the shape of the posterior distribution
- `rstanarm` runs 4 chains by default. Can think of these as 4 expeditions to explore the posterior.
- We want these chains to converge to the same distribution
- We can visually assess with a *trace plot*

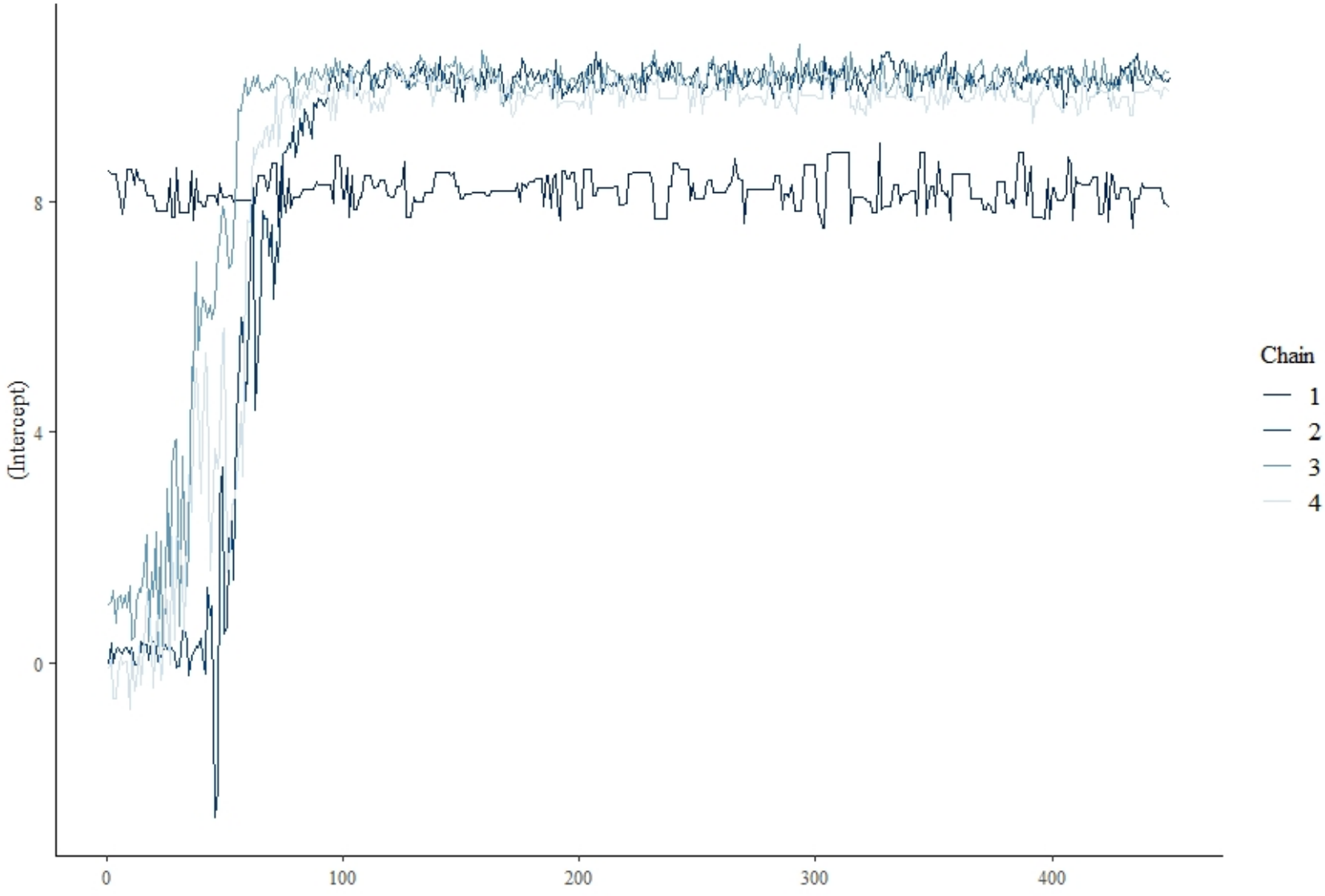
```
plot(stan_mod, plotfun = "trace", pars = "var_name")
```

We hope to see the chains indistinguishable from one another

Example of trace plot showing convergence



Example of trace plot showing non-convergence



Numerical assessment of convergence (`Rhat`)

- The $\hat{R}$ statistic compares variation between the chains to the variation within chains
- If all chains converge to the same region, then $\hat{R}$ should be close to 1
- If $\hat{R} < 1.1$ we have good, though not infallible, evidence of convergence
- Each parameter that has a posterior distribution will have a separate $\hat{R}$ statistic
- We would like them all less than 1.1
- Included in the `summary` output of the model in the "MCMC diagnostics" section

Dealing with convergence issues

- If the Markov chains do not converge, you'll get a warning that says "`Markov chains did not converge! Do not analyze results!`"
- This can sometimes be resolved by increasing the number of iterations using the `iter` argument.
- Example, increase from 2000 to 4000:

```
stan_glm(y ~ grp, data = bat, family = gaussian,
         iter = 4000)
```

Effective sample size (`n_eff`)

- MCMC sampling is not independent and can exhibit autocorrelation
- The Effective Sample Size (`n_eff`) corrects for any autocorrelation and approximates the number of independent samples we have
- We'd like to see `n_eff` > 1000
- Included in the `summary` output of the model in the "MCMC diagnostics" section
- Each of the 4 chains is run for 2000 iterations, with the first 1000 as a "warm-up"
- "warm-up" iterations are discarded
- This means we have $4 \times 1000 = 4000$ total samples
- `n_eff` is the approximate number of independent samples

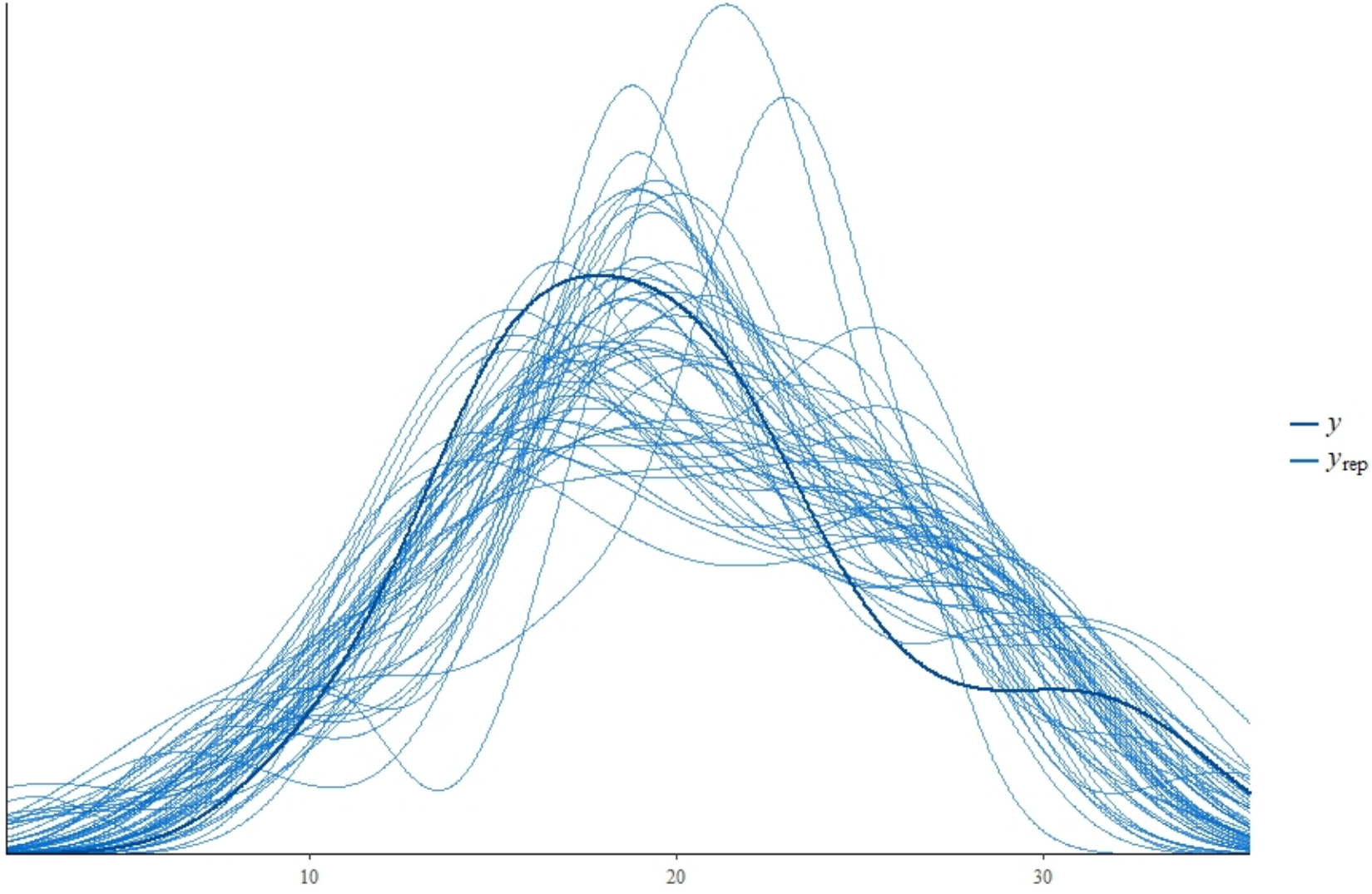
Monte carlo standard error (`mcse`)

- Helps assess error introduced by MCMC sampling
- Approximated by dividing posterior standard deviation of a parameter by its ESS
- a low MCSE relative to the posterior standard deviation results in higher ESS, which is what we want
- Included in the `summary` output of the model in the "MCMC diagnostics" section

Posterior predictive checking (PPC)

- Does our model adequately fit the data?
- PPC simulates data according to our fitted model and compares simulated data to observed data
- A good fitting model should replicate features of the observed data
- A visual check: `pp_check(stan_mod)`

Example of a PPC graph



Let's return to the R script.

Summarizing and interpreting results

Once we have assessed the sampling quality and model fit, we're ready to summarize and interpret the posterior distributions.

Common summaries include:

- point estimates
- posterior intervals
- posterior standard deviations

Point estimates

- These are usually the mean or median of a posterior distribution.
- Both mean and median are provided in the `summary` of a model fit with `rstanarm`
- Just one incomplete summary of the posterior distribution
- Point estimates should be provided with additional information, such as posterior intervals

Posterior intervals

- These are the range of possible parameter values stated in terms of probability
- Example: a 95% posterior interval of (0.4, 1.3) is interpreted to mean that there is 95% probability the parameter is between 0.4 and 1.3
- The 95% posterior intervals are provided in the `summary` output in the 2.5% and 97.5% columns

Posterior standard deviations

- These summarize the spread of the posterior distribution
- Probably not useful if a posterior distribution is skewed
- Also provided in the `summary` output

Posterior distributions revisited

Things to keep in mind:

- We have one posterior distribution for each parameter in our model
- The posterior distibutions are the result of MCMC sampling; each posterior distribution we obtain is actually a sample
- The default "sample size" in `rstanarm` is 4000 (4 chains at 2000 iterations each, with first 1000 as a warm-up)
- The `summary` output in `rstanarm` summarizes these samples from the posterior distributions

Working directly with posterior distributions

We can create our own summaries of the posterior distributions.

The `as.matrix` and `as.data.frame` functions extract the posterior samples into a matrix and data frame, respectively.

Let's say we want to estimate the probability that the mean backpack weight is greater than 16 lbs. Notice we specify the `(Intercept)` parameter with the $ operator. That's because there are two parameters to choose from (`Intercept` and `sigma`)

```
post_samp <- as.data.frame(bmod1)
mean(post_samp$`(Intercept)` > 16)
```

```
## [1] 0.544
```

Working directly with posterior distributions

Let's say we want to estimate the probability that the difference in battery life is greater than 1 hour. Notice we specify the `grp` parameter with the `$` operator. That's because there are three parameters to choose from (`Intercept`, `grp`, and `sigma`)

```
post_samp2 <- as.data.frame(bm.out)
mean(post_samp2$grp > 1)
```

```
## [1] 0.28475
```

Working directly with posterior distributions

Let's say we want to create 95% credibility intervals for the means of the two brands of batteries. The `Intercept` is the estimated mean of grp 0. The sum of the `Intercept` and `grp` coefficient is the estimated mean of grp 1.

```
quantile(post_samp2$`(Intercept)`,
         probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
##  9.82601 10.49629
```

```
quantile(post_samp2$`(Intercept)` + post_samp2$grp,
         probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 10.69355 11.35564
```

Let's return to the R script.

Using ShinyStan with `rstanarm` models

- `rstanarm` models can be diagnosed and explored with the ShinyStan web app
- ShinyStan is powered by the Shiny web application framework by RStudio
- To use: `launch_shinystan(mod)` where `mod` is your `rstanarm` model
- The ShinyStan application will open in your default web browser
- ShinyStan has a GUI that allows you to interactively explore and diagnose your model

Some journal articles using `rstanarm`

Espe, M. et al. (2016) Yield gap analysis of US rice production systems shows opportunities for improvement. *Field Crops Research*, 196:276-283.

Kubrak, O. et al. (2017) Adaptation to fluctuating environments in a selection experiment with Drosophila melanogaster. *Ecology and Evolution*, 7:3796-3807.

Herzog, S. et al. (2017) Sun Protection Factor Communication of Sunscreen Effectiveness. *JAMA Dermatology*, 153(3):348-350.

Kovic, M. and Hänsli, N. (2017) The impact of political cleavages, religiosity, and values on attitudes towards nonprofit organizations. *Social Sciences*, 7(1), 2.

References

McElreath, M. (2020). *Statistical Rethinking, 2nd Ed*. CRC Press. Boca Raton.

Muth, C., Oravecz, Z., & Gabry, J. (2018). User-friendly Bayesian regression modeling: A tutorial with rstanarm and shinystan. *The Quantitative Methods for Psychology*, 14(2), 99-119. doi:10.20982/tqmp.14.2.p099

`rstanarm` web site: http://mc-stan.org/rstanarm/

Betancourt, Michael. (2018) *A Conceptual Introduction to Hamiltonian Monte Carlo*. https://arxiv.org/pdf/1701.02434.pdf