

amputation

2024-05-08

Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(brms)
```

```
## Loading required package: Rcpp
## Loading 'brms' package (version 2.21.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
##
## Attaching package: 'brms'
##
## The following object is masked from 'package:stats':
##
##      ar
```

```
library(ggdist)
```

```
##
## Attaching package: 'ggdist'
##
## The following objects are masked from 'package:brms':
##
##      dstudent_t, pstudent_t, qstudent_t, rstudent_t
```

```
library(tidybayes)
```

```
##
## Attaching package: 'tidybayes'
##
## The following objects are masked from 'package:brms':
##
##      dstudent_t, pstudent_t, qstudent_t, rstudent_t
```

```
library(marginaleffects)
```

Data

```
amputations <- read_csv("../amputation.csv") %>%
  select(Wound.location = 2, Amputation, Woundtype, Colony)

## Rows: 74 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (3): Wound location, Woundtype, Colony
## dbl (2): Sample size, Amputation
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Priors

```
priors_01 <- c(
  prior(normal(0, 1.5),
    class = "Intercept"),
  prior(normal(0, 1.5),
    class = "b"),
  prior(normal(0, 1),
    class = "sd")
)
```

Prior predictive check

```
brm_01.pp <- brm(Amputation ~ Wound.location*Woundtype + (1 | Colony),
  family = "bernoulli",
  prior = priors_01,
  sample_prior = "only",
  data = amputations)
```

```
## Compiling Stan program...
## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 15.0.0 (clang-1500.3.9.4)'
## using SDK: 'MacOSX14.4.sdk'
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Resources/include"
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/StanHeaders/include/src/StanHeaders/mathfun.h:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/include/Eigen/Core:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/include/Eigen/Geometry:1:
## /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/include/Eigen/src/Core/Matrix.h:1:
## #include <cmath>
## ~~~~~
## 1 error generated.
```

```

## make: *** [foo.o] Error 1

## Start sampling

##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.66 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 1: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 1: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 1: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 1: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 1: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 1: Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 1: Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 1: Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 1: Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 1: Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 1: Iteration:  2000 / 2000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 0.019 seconds (Warm-up)
## Chain 1:                0.015 seconds (Sampling)
## Chain 1:                0.034 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 2: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 2: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 2: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 2: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 2: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 2: Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 2: Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 2: Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 2: Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 2: Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 2: Iteration:  2000 / 2000 [100%] (Sampling)
## Chain 2:
## Chain 2: Elapsed Time: 0.019 seconds (Warm-up)
## Chain 2:                0.015 seconds (Sampling)
## Chain 2:                0.034 seconds (Total)
## Chain 2:
##

```

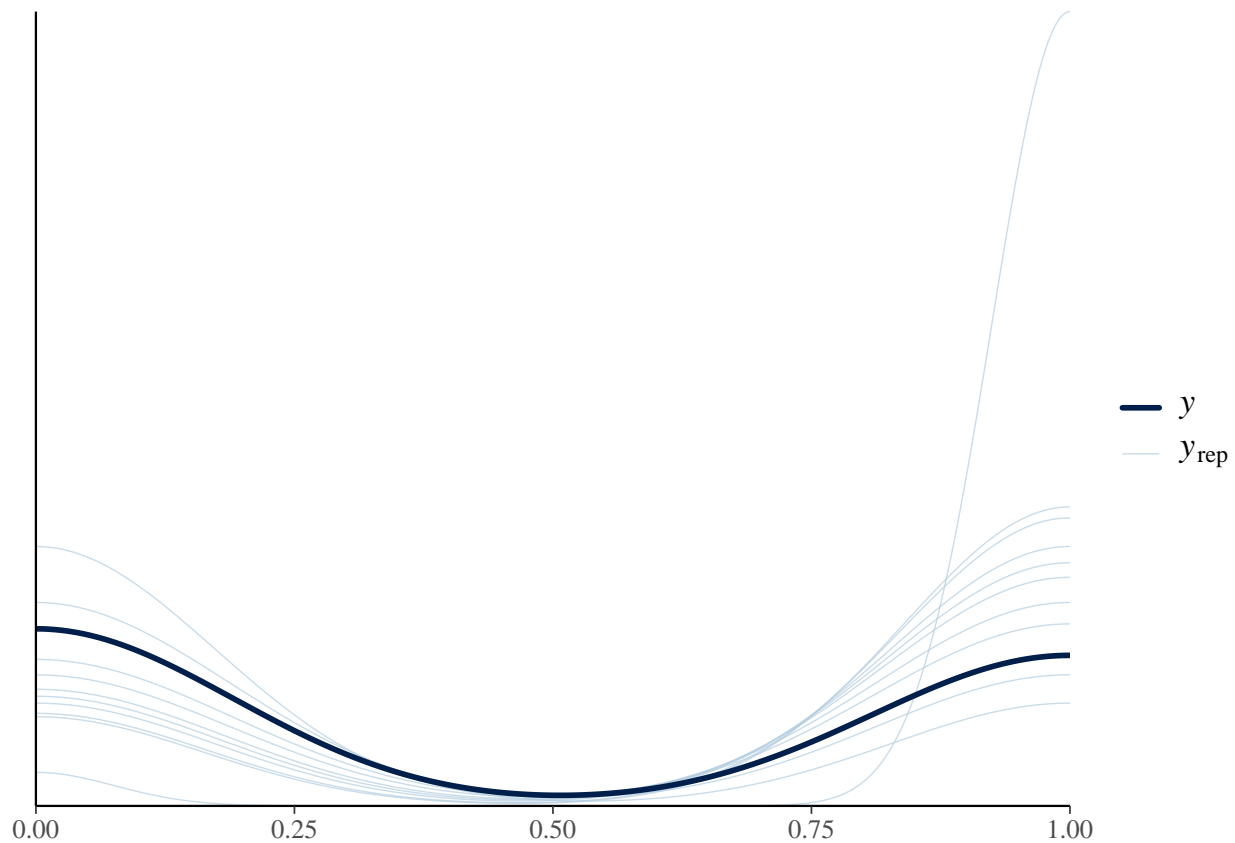
```

## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 5e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 3: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 3: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 3: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 3: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 3: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 3:
## Chain 3: Elapsed Time: 0.02 seconds (Warm-up)
## Chain 3:                0.015 seconds (Sampling)
## Chain 3:                0.035 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 5e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 4: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 4: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 4: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 4: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 4: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 4:
## Chain 4: Elapsed Time: 0.02 seconds (Warm-up)
## Chain 4:                0.014 seconds (Sampling)
## Chain 4:                0.034 seconds (Total)
## Chain 4:

```

```
pp_check(brm_01.pp)
```

```
## Using 10 posterior draws for ppc type 'dens_overlay' by default.
```



Update model with data

```
brm_01 <- update(brm_01.pp,
  file = "./brm_01",
  sample_prior = TRUE,
  iter = 2000)
```

The desired updates require recompiling the model

Validate model

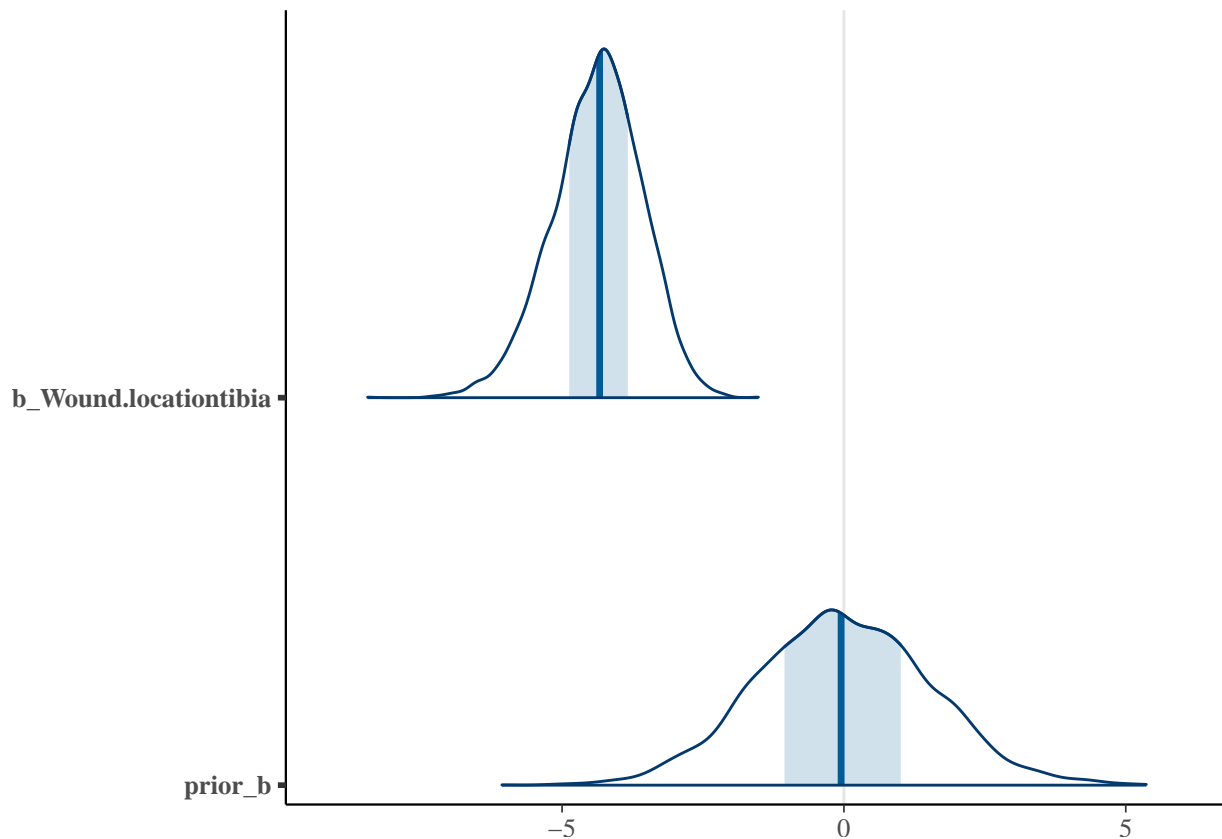
Check influence of priors

```
get_variables(brm_01)
```

```
## [1] "b_Intercept"
## [2] "b_Wound.locationtibia"
## [3] "b_WoundtypeSterile"
## [4] "b_Wound.locationtibia:WoundtypeSterile"
## [5] "sd_Colony__Intercept"
## [6] "Intercept"
## [7] "r_Colony[C0,Intercept]"
## [8] "r_Colony[C1,Intercept]"
## [9] "r_Colony[C10,Intercept]"
## [10] "r_Colony[C27,Intercept]"
```

```
## [11] "r_Colony[C8,Intercept]"
## [12] "r_Colony[C9,Intercept]"
## [13] "prior_Intercept"
## [14] "prior_b"
## [15] "prior_sd_Colony"
## [16] "lprior"
## [17] "lp_-"
## [18] "accept_stat_-"
## [19] "stepsize_-"
## [20] "treedepth_-"
## [21] "n_leapfrog_-"
## [22] "divergent_-"
## [23] "energy_-"
```

```
mcmc_plot(brm_01,
  variable = c("b_Wound.locationtibia", "prior_b"),
  regex = FALSE,
  type = "areas")
```



Visualize model

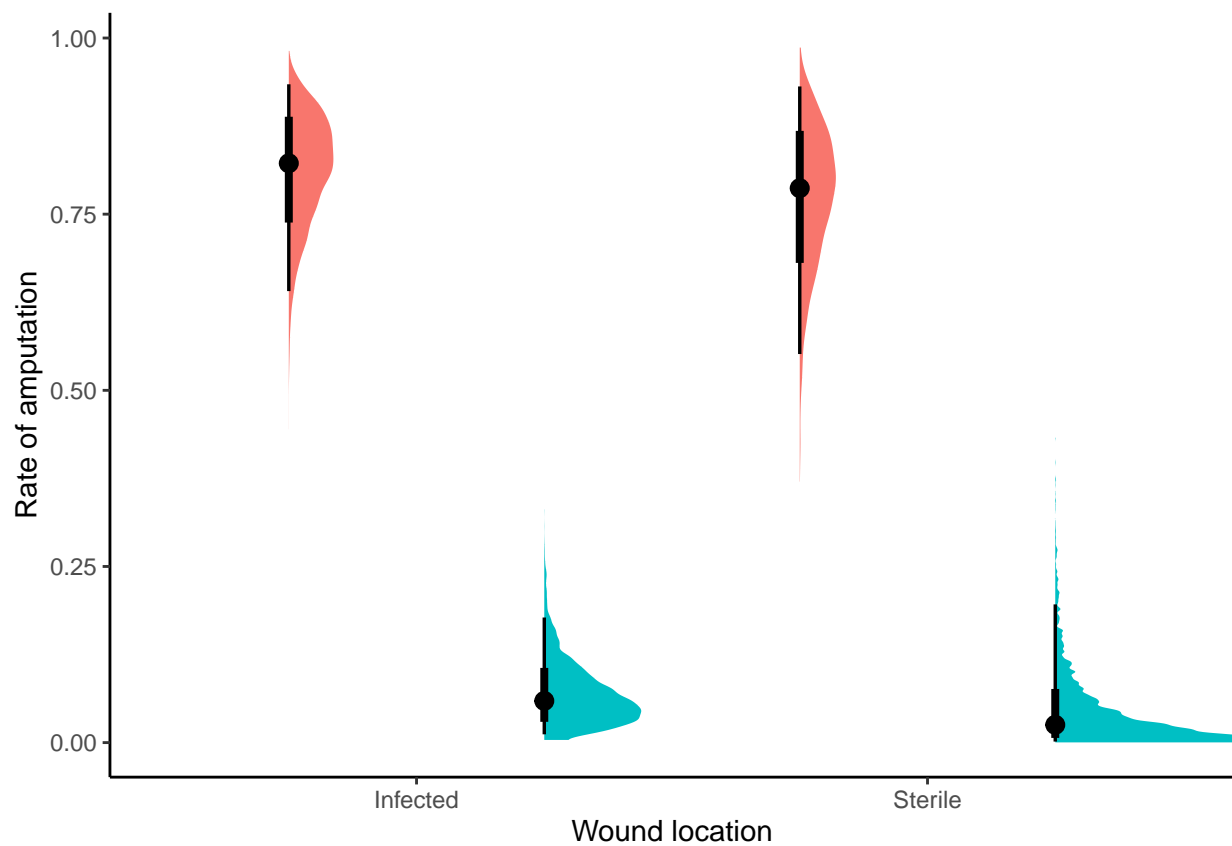
```
# New data frame to generate predictions for the full design matrix of the model
nd <- amputations %>%
  select(Wound.location, Woundtype) %>%
  distinct()
```

```

# Predictions from full design matrix
brm_01_epred <- brm_01 %>%
  add_epred_draws(
    newdata = nd,
    re_formula = NA,
    allow_new_levels = FALSE)

p=ggplot(brm_01_epred, aes(Woundtype, .epred, fill = Wound.location)) +
  stat_halfeye(position=position_dodge()) +
  labs(x = "Wound location", y = "Rate of amputation") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank())

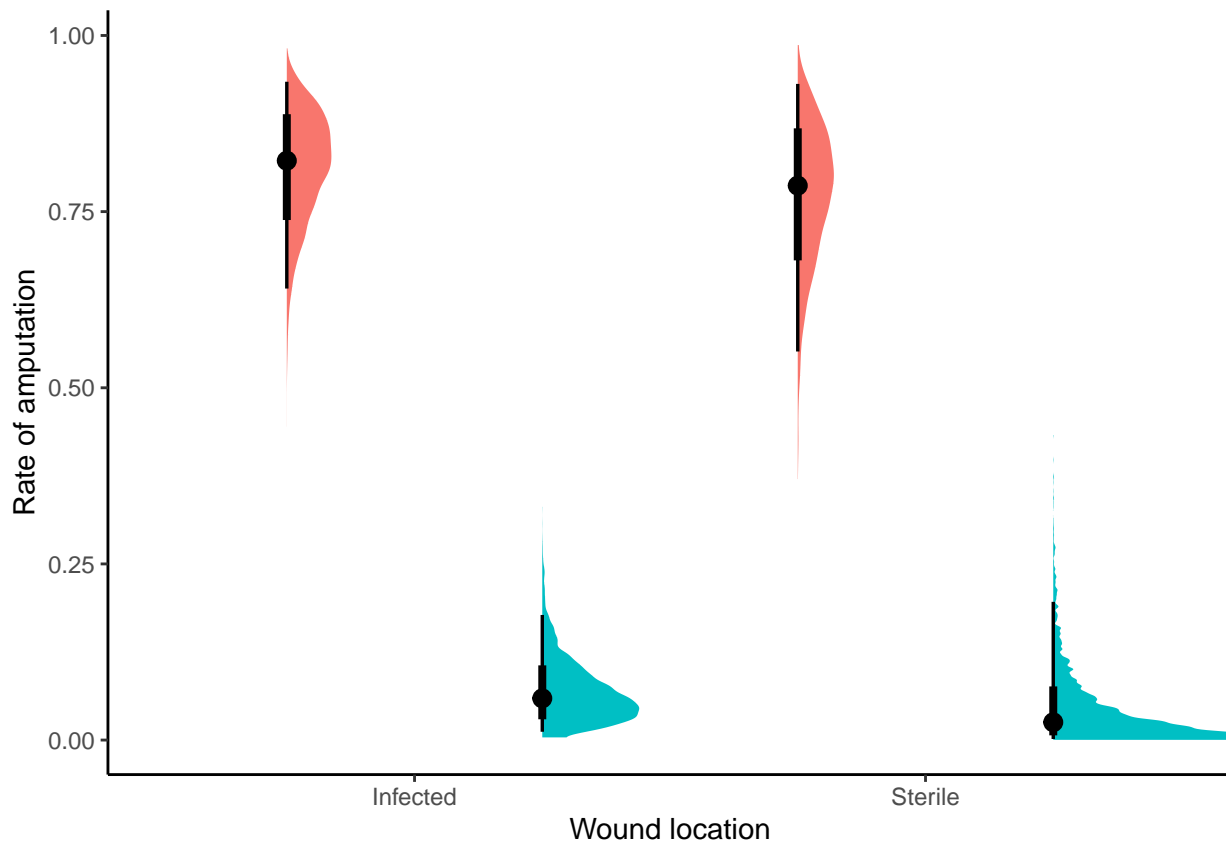
```



```

ggsave(filename="BayesianAmputationrate.pdf", plot=print(p),device="pdf",
  width=8.9,height=8.9,units="cm",dpi=300)

```



```
# Contrast
brm_01_epred_tibia <- brm_01_epred %>%
  filter(Wound.location == "tibia") %>%
  rename(epred_tibia = .epred) %>%
  ungroup() %>%
  select(.draw, epred_tibia)

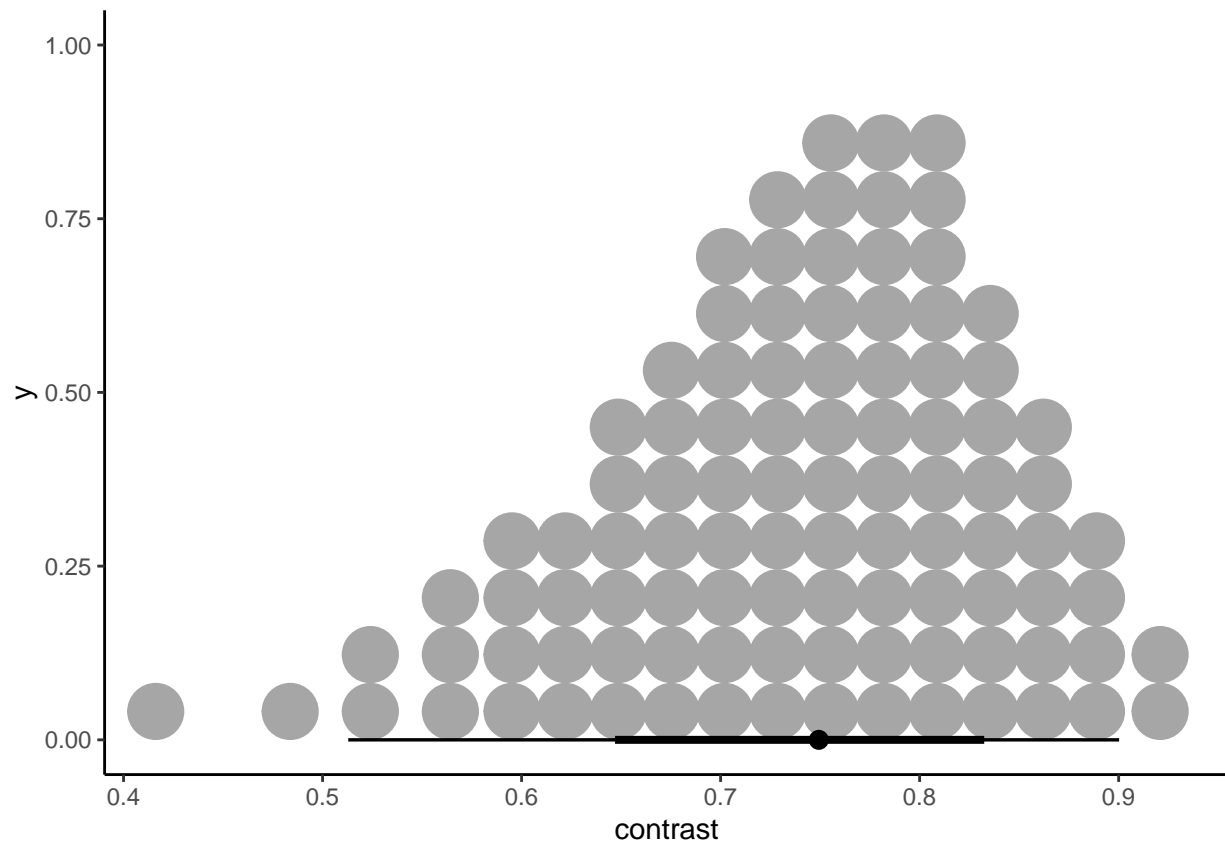
brm_01_epred_femur <- brm_01_epred %>%
  filter(Wound.location == "femur") %>%
  rename(epred_femur = .epred) %>%
  ungroup() %>%
  select(.draw, epred_femur)

brm_01_epred_contrast <- full_join(brm_01_epred_tibia, brm_01_epred_femur) %>%
  mutate(contrast = epred_femur - epred_tibia)

## Joining with `by = join_by(.draw)`

## Warning in full_join(brm_01_epred_tibia, brm_01_epred_femur): Detected an unexpected many-to-many relationship.
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 1 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.

ggplot(brm_01_epred_contrast, aes(contrast)) +
  stat_dotsinterval(quantiles = 100) +
  theme_classic()
```

```
ggplot(brm_01_epred_contrast, aes(contrast)) +
  stat_halfeye() +
  labs(y = "Probability density", x = "femur - tibia") +
  theme_classic()
```

