Plot and examine chains: 6 regions (no wrap-up; no matrix verb)

JD

March 4, 2021

1 Simple model and plots

This file collects draws and generates plots and info about parameters.

```
burnin <- 500

library(DBI)
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union

library(rstan)

## Loading required package: StanHeaders
## Loading required package: ggplot2
## rstan (Version 2.21.1, GitRev: 2e1f913d3ca3)
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

library(loo)

## This is loo version 2.3.1
## - Online documentation and vignettes at mc-stan.org/loo
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the
'cores' argument or set options(mc.cores = NUM_CORES) for an entire session.
##
## Attaching package: 'loo'
## The following object is masked from 'package:rstan':
##
##    loo

c1 <- read.csv("chain1/chain-0.csv")

dataf <- select(c1, starts_with("subj_mu_rt"))
```

```r
dataf$std <- c1$std

dataf <- dataf[burnin:length(dataf[, 1]), ]

str(dataf)
```

```
## 'data.frame': 5009 obs. of  7 variables:
##  $ subj_mu_rt__0: num  350 350 350 350 350 ...
##  $ subj_mu_rt__1: num  348 348 345 345 347 ...
##  $ subj_mu_rt__2: num  311 311 311 311 311 ...
##  $ subj_mu_rt__3: num  385 385 379 379 384 ...
##  $ subj_mu_rt__4: num  337 337 337 337 337 ...
##  $ subj_mu_rt__5: num  311 311 311 311 311 ...
##  $ std          : num  34.9 34.9 34.9 29.3 26.9 ...
```

```r
dataf2 <- select(c1, starts_with("obj_mu_rt"))

dataf2$std <- c1$std

dataf2 <- dataf2[burnin:length(dataf2[, 1]), ]

str(dataf2)
```

```
## 'data.frame': 5009 obs. of  7 variables:
##  $ obj_mu_rt__0: num  350 350 350 350 350 ...
##  $ obj_mu_rt__1: num  345 345 345 345 345 ...
##  $ obj_mu_rt__2: num  385 385 379 379 384 ...
##  $ obj_mu_rt__3: num  378 382 379 376 380 ...
##  $ obj_mu_rt__4: num  348 349 349 348 349 ...
##  $ obj_mu_rt__5: num  311 311 311 311 311 ...
##  $ std         : num  34.9 34.9 34.9 29.3 26.9 ...
```

```r
c2 <- read.csv("chain2/chain-0.csv")

dataf.c2 <- select(c2, starts_with("subj_mu_rt"))

dataf.c2$std <- c2$std

dataf.c2 <- dataf.c2[burnin:length(dataf.c2[, 1]), ]

dataf <- rbind(dataf, dataf.c2)

str(dataf)
```

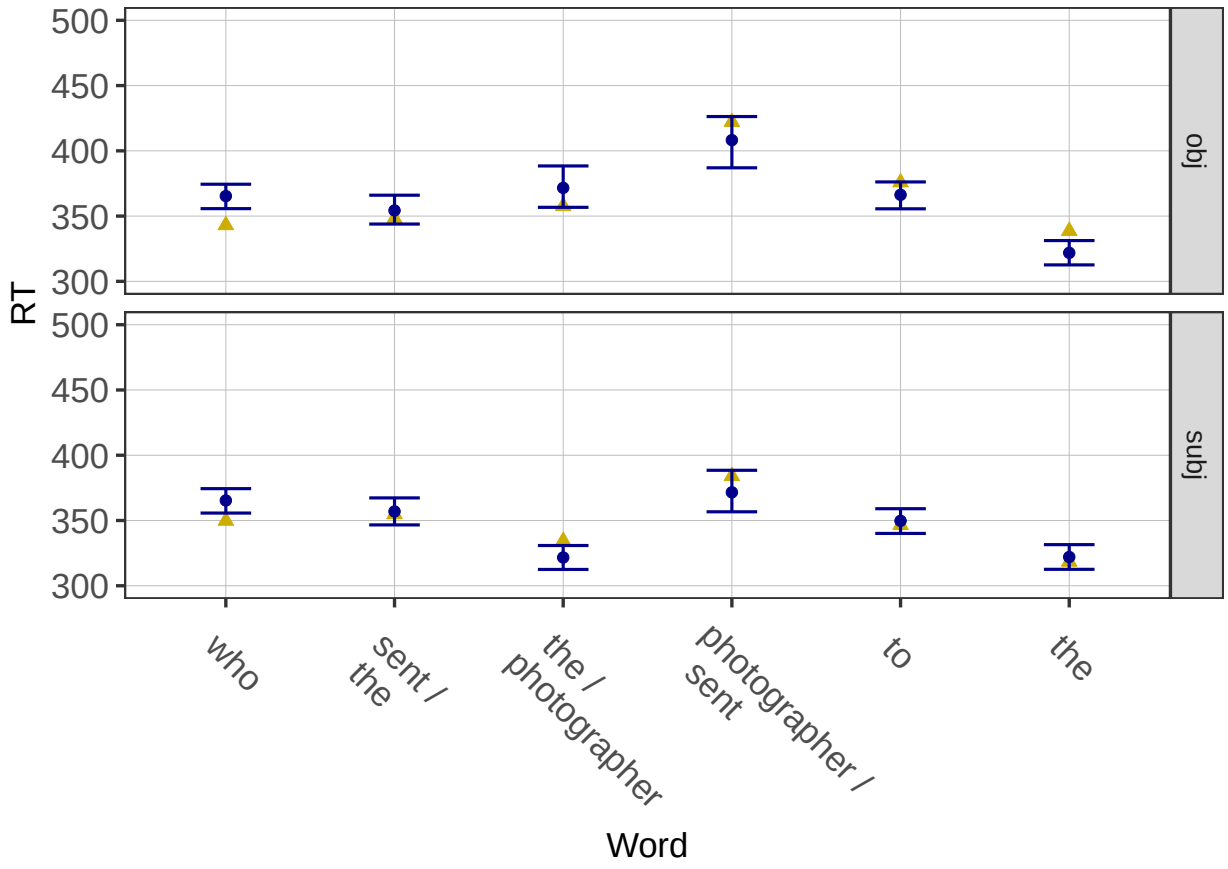
```
## 'data.frame': 10018 obs. of  7 variables:
##  $ subj_mu_rt__0: num  350 350 350 350 350 ...
##  $ subj_mu_rt__1: num  348 348 345 345 347 ...
##  $ subj_mu_rt__2: num  311 311 311 311 311 ...
##  $ subj_mu_rt__3: num  385 385 379 379 384 ...
##  $ subj_mu_rt__4: num  337 337 337 337 337 ...
##  $ subj_mu_rt__5: num  311 311 311 311 311 ...
##  $ std          : num  34.9 34.9 34.9 29.3 26.9 ...
```

```r
dataf2.c2 <- select(c2, starts_with("obj_mu_rt"))
```

```
dataf2.c2$std <- c2$std

dataf2.c2 <- dataf2.c2[burnin:length(dataf2.c2[, 1]), ]

dataf2 <- rbind(dataf2, dataf2.c2)

str(dataf2)

## 'data.frame': 10018 obs. of  7 variables:
##  $ obj_mu_rt__0: num  350 350 350 350 350 ...
##  $ obj_mu_rt__1: num  345 345 345 345 345 ...
##  $ obj_mu_rt__2: num  385 385 379 379 384 ...
##  $ obj_mu_rt__3: num  378 382 379 376 380 ...
##  $ obj_mu_rt__4: num  348 349 349 348 349 ...
##  $ obj_mu_rt__5: num  311 311 311 311 311 ...
##  $ std         : num  34.9 34.9 34.9 29.3 26.9 ...

ndraws <- length(dataf2[, 1])

data.all <- data.frame(word_no = rep.int(rep(paste("No", 3:8, sep = ""), each = ndraws),
    2), word = factor(rep(rep(c("who", "sent /\nthe", "the /\nphotographer",
    "photographer /\nsent", "to", "the"), each = ndraws), 2), levels = c("who",
    "sent /\nthe", "the /\nphotographer", "photographer /\nsent", "to", "the")),
    extraction = c(rep("subj", ndraws * 6), rep("obj", ndraws * 6)), RT = c(dataf[,
        1], dataf[, 2], dataf[, 3], dataf[, 4], dataf[, 5], dataf[, 6], dataf2[,
        1], dataf2[, 2], dataf2[, 3], dataf2[, 4], dataf2[, 5], dataf2[, 6]),
    x = rep(c(349.8, 354.8, 334.3, 384, 346.5, 318.4, 343, 348.1, 357.6, 422.1,
        375.8, 338.6), each = ndraws), std = c(rep(dataf$std, 6), rep(dataf2$std,
        6)))

# data.all <- data.frame(word_no=rep.int(rep(paste('No', 2:8, sep=''),
# each=ndraws), 2), word=factor(rep(c('reporter', 'who', 'sent', 'the',
# 'photographer', 'to', 'the ', 'reporter ', 'who ', 'the ', 'photographer
# ', 'sent ', 'to ', 'the '), each=ndraws), levels=c('reporter', 'who',
# 'sent', 'the', 'photographer', 'to', 'the ', 'reporter ', 'who ', 'the ',
# 'photographer ', 'sent ', 'to ', 'the ')), extraction=c(rep('subj',
# ndraws*7), rep('obj', ndraws*7)), RT=c(dataf[,4], dataf[,5], dataf[,6],
# dataf[,7], dataf[,8], dataf[,9], dataf[,10], dataf2[,4], dataf2[,5],
# dataf2[,6], dataf2[,7], dataf2[,8], dataf2[,9], dataf2[,10]),
# x=rep(c(360.2, 349.8, 354.8, 334.3, 384, 346.5, 318.4, 373.1, 343, 348.1,
# 357.6, 422.1, 375.8, 338.6), each=ndraws))

str(data.all)

## 'data.frame': 120216 obs. of  6 variables:
##  $ word_no   : Factor w/ 6 levels "No3","No4","No5",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ word      : Factor w/ 6 levels "who","sent /\nthe",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ extraction: Factor w/ 2 levels "obj","subj": 2 2 2 2 2 2 2 2 2 2 ...
##  $ RT        : num  350 350 350 350 350 ...
##  $ x         : num  350 350 350 350 350 ...
##  $ std       : num  34.9 34.9 34.9 29.3 26.9 ...
```

Prepare data for plots.

```r
library(ggplot2)

library(dplyr)

data.to.plot <- data.all %>% group_by(extraction, word_no) %>% summarise(Extraction = first(extraction)
    Word.no = as.factor(first(word_no)), Word = as.factor(first(word)), CF1 = quantile(RT,
        probs = c(0.05, 0.95))[1], CF2 = quantile(RT, probs = c(0.05, 0.95))[2],
    RT = mean(RT), Observed = first(x))

## `summarise()` has grouped output by 'extraction'. You can override using the `.groups` argument.

data.to.plot

## # A tibble: 12 x 9
## # Groups:   extraction [2]
##    extraction word_no Extraction Word.no Word     CF1   CF2    RT Observed
##    <fct>      <fct>   <fct>      <fct>   <fct>  <dbl> <dbl> <dbl>    <dbl>
##  1 obj        No3     obj        No3     "who"   356.  374.  365.     343
##  2 obj        No4     obj        No4     "sent~  344.  366.  354.     348.
##  3 obj        No5     obj        No5     "the ~  357.  388.  372.     358.
##  4 obj        No6     obj        No6     "phot~  387.  426.  408.     422.
##  5 obj        No7     obj        No7     "to"    356.  376.  366.     376.
##  6 obj        No8     obj        No8     "the"   313.  331.  322.     339.
##  7 subj       No3     subj       No3     "who"   356.  374.  365.     350.
##  8 subj       No4     subj       No4     "sent~  347.  367.  357.     355.
##  9 subj       No5     subj       No5     "the ~  313.  331.  322.     334.
## 10 subj       No6     subj       No6     "phot~  357.  388.  372.     384
## 11 subj       No7     subj       No7     "to"    340.  359.  350.     346.
## 12 subj       No8     subj       No8     "the"   313.  332.  322.     318.

g1 <- ggplot(data.to.plot, aes(Word, RT))
g1 <- g1 + geom_point(aes(x = Word, y = Observed), fill = "gold3", color = "gold3",
    pch = 24, size = 4) + geom_point(color = "blue4", size = I(4)) + geom_errorbar(aes(ymin = CF1,
    ymax = CF2), color = "blue4", width = 0.3, size = I(1.2)) + theme_bw(28)
g1 <- g1 + theme(axis.text.x = element_text(angle = -45, hjust = 0.1, size = 28),
    axis.text.y = element_text(size = 28), axis.title = element_text(size = 28),
    legend.position = "none", panel.grid.major = element_line(colour = "grey",
        size = (0.25)), panel.grid.minor = element_blank())
g1 <- g1 + coord_cartesian(ylim = c(300, 500)) + facet_grid(Extraction ~ .)
```




```
ggsave("predictions-observed-RT-grodner-gibson-exp1-6regions.pdf", width = 20,
    height = 12)
```

2 WAIC

```
calculate_log_likelihood <- function(predicted, observed, std) {

    log(dnorm(observed, mean = predicted, sd = std))

}

ll <- matrix(calculate_log_likelihood(data.all$RT, data.all$x, data.all$std),
    nrow = ndraws)

str(ll)

##  num [1:10018, 1:12] -4.47 -4.47 -4.47 -4.3 -4.21 ...

waic(ll)

## Warning:
## 2 (16.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
##
## Computed from 10018 by 12 log-likelihood matrix
##
##           Estimate  SE
## elpd_waic    -51.0 1.5
## p_waic         2.6 0.6
## waic         102.0 3.0
##
## 2 (16.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

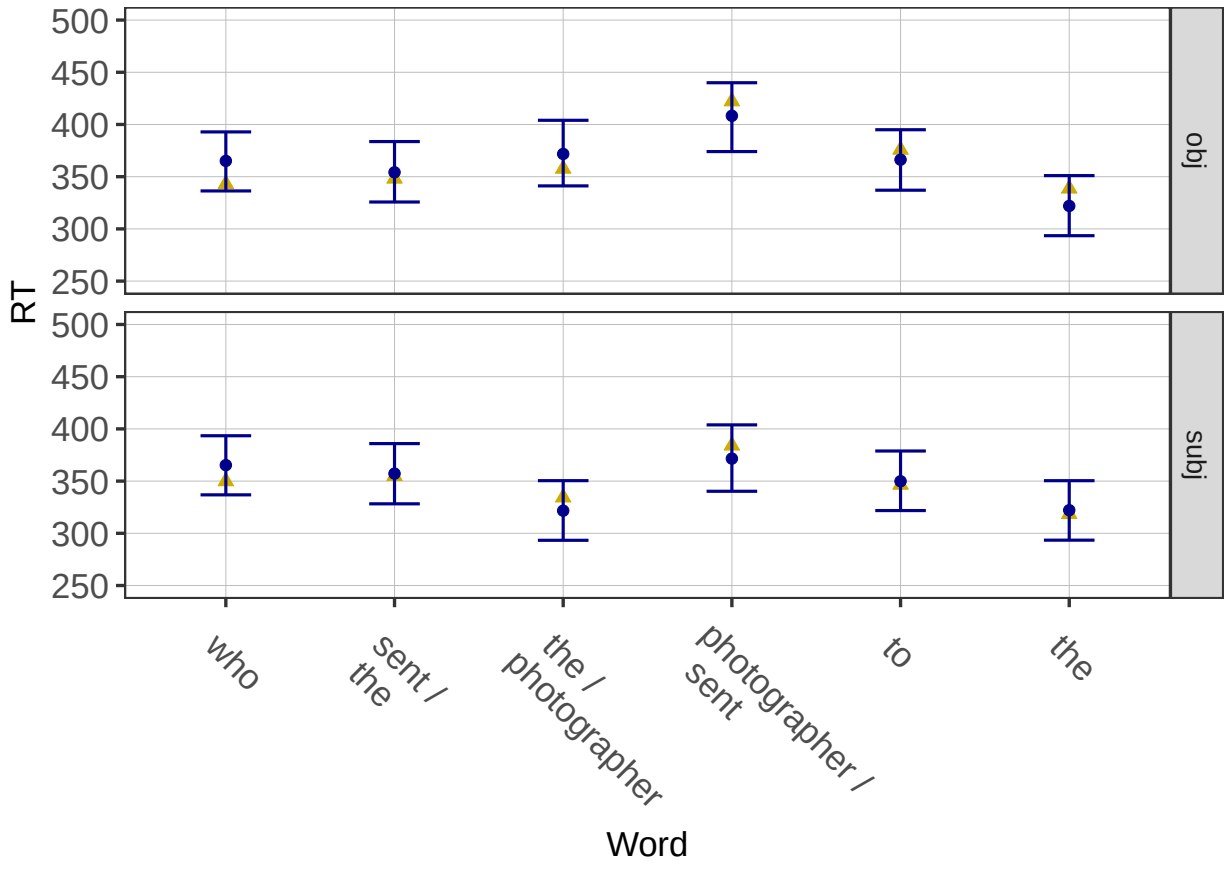
3 Posterior predictive checks

```
dataf <- read.csv("chain1/posterior_predictive_checks_subj.csv", row.names = 1)

str(dataf)

## 'data.frame': 5507 obs. of  6 variables:
##  $ X0: num  422 383 349 386 374 ...
##  $ X1: num  354 358 388 380 317 ...
##  $ X2: num  363 315 362 317 298 ...
##  $ X3: num  373 338 412 388 319 ...
##  $ X4: num  400 403 381 400 392 ...
##  $ X5: num  346 365 360 295 346 ...

dataf2 <- read.csv("chain1/posterior_predictive_checks_obj.csv", row.names = 1)

str(dataf2)

## 'data.frame': 5507 obs. of  6 variables:
##  $ X0: num  340 357 363 363 386 ...
##  $ X1: num  382 409 389 369 372 ...
##  $ X2: num  407 361 485 363 385 ...
##  $ X3: num  372 358 356 424 402 ...
##  $ X4: num  405 402 360 319 342 ...
##  $ X5: num  308 344 404 328 392 ...

dataf.c2 <- read.csv("chain2/posterior_predictive_checks_subj.csv", row.names = 1)

dataf <- rbind(dataf, dataf.c2)

str(dataf)

## 'data.frame': 11014 obs. of  6 variables:
##  $ X0: num  422 383 349 386 374 ...
##  $ X1: num  354 358 388 380 317 ...
##  $ X2: num  363 315 362 317 298 ...
##  $ X3: num  373 338 412 388 319 ...
##  $ X4: num  400 403 381 400 392 ...
##  $ X5: num  346 365 360 295 346 ...

dataf2.c2 <- read.csv("chain2/posterior_predictive_checks_obj.csv", row.names = 1)
```

```r
dataf2 <- rbind(dataf2, dataf2.c2)

str(dataf)

## 'data.frame': 11014 obs. of  6 variables:
##  $ X0: num  422 383 349 386 374 ...
##  $ X1: num  354 358 388 380 317 ...
##  $ X2: num  363 315 362 317 298 ...
##  $ X3: num  373 338 412 388 319 ...
##  $ X4: num  400 403 381 400 392 ...
##  $ X5: num  346 365 360 295 346 ...

ndraws <- length(dataf2[, 1])

data.all <- data.frame(word_no = rep.int(rep(paste("No", 3:8, sep = ""), each = ndraws),
    2), word = factor(rep(rep(c("who", "sent /\nthe", "the /\nphotographer",
    "photographer /\nsent", "to", "the"), each = ndraws), 2), levels = c("who",
    "sent /\nthe", "the /\nphotographer", "photographer /\nsent", "to", "the")),
    extraction = c(rep("subj", ndraws * 6), rep("obj", ndraws * 6)), RT = c(dataf[,
        1], dataf[, 2], dataf[, 3], dataf[, 4], dataf[, 5], dataf[, 6], dataf2[,
        1], dataf2[, 2], dataf2[, 3], dataf2[, 4], dataf2[, 5], dataf2[, 6]),
    x = rep(c(349.8, 354.8, 334.3, 384, 346.5, 318.4, 343, 348.1, 357.6, 422.1,
        375.8, 338.6), each = ndraws))

str(data.all)

## 'data.frame': 132168 obs. of  5 variables:
##  $ word_no   : Factor w/ 6 levels "No3","No4","No5",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ word      : Factor w/ 6 levels "who","sent /\nthe",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ extraction: Factor w/ 2 levels "obj","subj": 2 2 2 2 2 2 2 2 2 2 ...
##  $ RT        : num  422 383 349 386 374 ...
##  $ x         : num  350 350 350 350 350 ...

# data.all <- subset(data.all, RT > 50 & RT < 3000)

library(ggplot2)

library(dplyr)

data.to.plot <- data.all %>% group_by(extraction, word_no) %>% summarise(Extraction = first(extraction),
    Word.no = as.factor(first(word_no)), Word = as.factor(first(word)), CF1 = quantile(RT,
        probs = c(0.05, 0.95))[1], CF2 = quantile(RT, probs = c(0.05, 0.95))[2],
    RT = mean(RT), Observed = first(x))

## `summarise()` has grouped output by 'extraction'. You can override using the `.groups` argument.

data.to.plot

## # A tibble: 12 x 9
## # Groups:   extraction [2]
##    extraction word_no Extraction Word.no Word     CF1   CF2    RT Observed
##    <fct>      <fct>   <fct>      <fct>   <fct>  <dbl> <dbl> <dbl>    <dbl>
##  1 obj        No3     obj        No3     "who"   336.  393.  365.     343
##  2 obj        No4     obj        No4     "sent~  326.  384.  354.     348.
##  3 obj        No5     obj        No5     "the ~  341.  404.  372.     358.
```

```
##  4 obj        No6     obj        No6     "phot~  374.  440.  408.     422.
##  5 obj        No7     obj        No7     "to"    337.  395.  366.     376.
##  6 obj        No8     obj        No8     "the"   293.  351.  322.     339.
##  7 subj       No3     subj       No3     "who"   337.  393.  365.     350.
##  8 subj       No4     subj       No4     "sent~  328.  386.  357.     355.
##  9 subj       No5     subj       No5     "the ~  293.  350.  322.     334.
## 10 subj       No6     subj       No6     "phot~  340.  404.  372.     384
## 11 subj       No7     subj       No7     "to"    322.  379.  350.     346.
## 12 subj       No8     subj       No8     "the"   293.  350.  322.     318.
```

```
g1 <- ggplot(data.to.plot, aes(Word, RT))
g1 <- g1 + geom_point(aes(x = Word, y = Observed), fill = "gold3", color = "gold3",
    pch = 24, size = 4) + geom_point(color = "blue4", size = I(4)) + geom_errorbar(aes(ymin = CF1,
    ymax = CF2), color = "blue4", width = 0.3, size = I(1.2)) + theme_bw(28)
g1 <- g1 + theme(axis.text.x = element_text(angle = -45, hjust = 0.1, size = 28),
    axis.text.y = element_text(size = 28), axis.title = element_text(size = 28),
    legend.position = "none", panel.grid.major = element_line(colour = "grey",
        size = (0.25)), panel.grid.minor = element_blank())
g1 <- g1 + coord_cartesian(ylim = c(250, 500)) + facet_grid(Extraction ~ .)
```




```
ggsave("posterior-predictive-checks-grodner-gibson-exp1-6regions.pdf", width = 20,
    height = 12)
```

4 Parameters

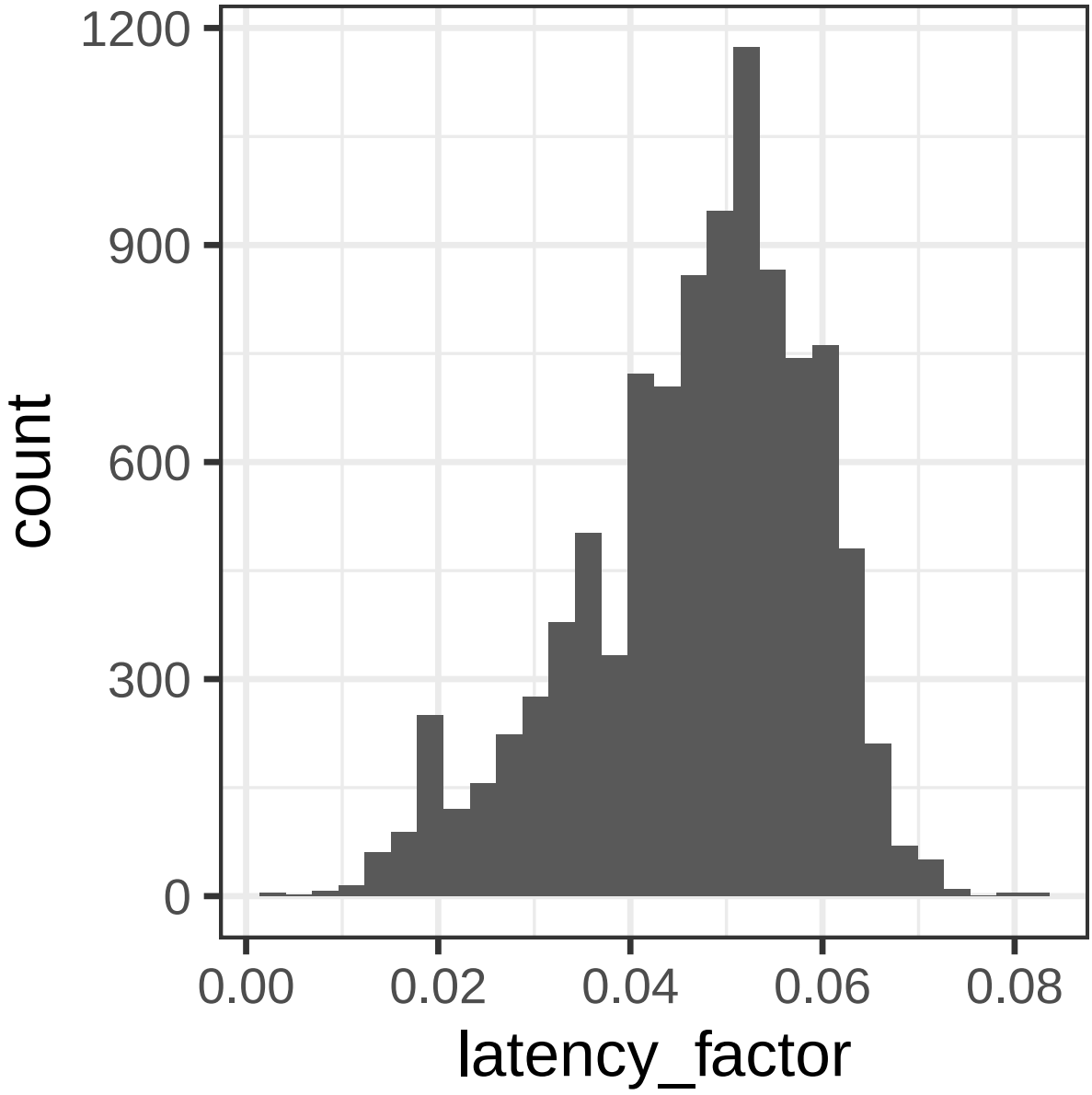
4.1 LF

Rhat:

```
draws <- createdraws("lf")

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(param)` instead of `param` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.

str(draws)

##  num [1:5009, 1:2] 0.0239 0.0239 0.022 0.022 0.0234 ...

Rhat(draws)

## [1] 1.047399
```

Mean etc.

```
tail(draws)

##               [,1]       [,2]
## [5004,] 0.06038476 0.04922450
## [5005,] 0.06038476 0.04922450
## [5006,] 0.06038476 0.04922450
## [5007,] 0.06038476 0.03871595
## [5008,] 0.06038476 0.03871595
## [5009,] 0.06038476 0.03871595

mean(c(draws[, 1:2]))

## [1] 0.04691501

median(c(draws[, 1:2]))

## [1] 0.04912666

sd(c(draws[, 1:2]))

## [1] 0.0121508

g1 <- ggplot(data.frame(latency_factor = c(draws[, 1:2])), aes(latency_factor))
g1 <- g1 + geom_histogram() + theme_bw(28)
g1

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("gg1-lf.pdf", width = 20, height = 12)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

4.2 LE

Rhat:

```
draws <- createdraws("le")

Rhat(draws)

## [1] 1.017002
```

Mean etc.

```
tail(draws)
```

```
##              [,1]      [,2]
## [5004,] 0.1273843 0.1619161
## [5005,] 0.1685181 0.1619161
## [5006,] 0.1685181 0.1619161
## [5007,] 0.1685181 0.1619161
## [5008,] 0.1685181 0.1619161
## [5009,] 0.1816423 0.1097034
```

```
mean(c(draws[, 1:2]))
```

```
## [1] 0.1703266
```

```
median(c(draws[, 1:2]))
```

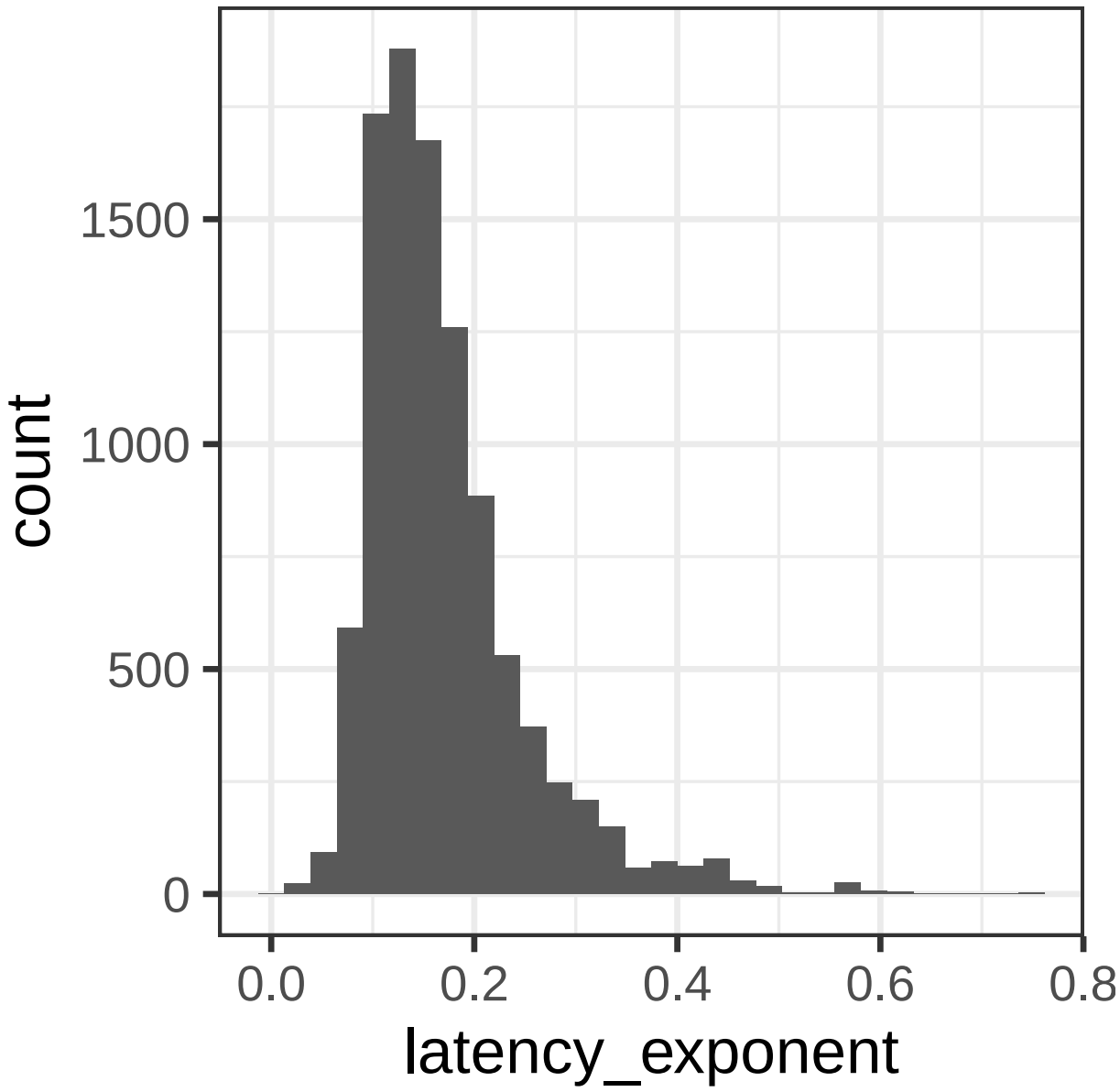
```
## [1] 0.1511095
```

```
sd(c(draws[, 1:2]))
```

```
## [1] 0.07967335
```

```
g1 <- ggplot(data.frame(latency_exponent = c(draws[, 1:2])), aes(latency_exponent))
g1 <- g1 + geom_histogram() + theme_bw(28)
g1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("gg1-le.pdf", width = 20, height = 12)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

4.3 RF

Rhat:

```
draws <- createdraws("rf")

Rhat(draws)

## [1] 1.049294
```

Mean etc.

```
tail(draws)
```

```
##               [,1]       [,2]
## [5004,] 0.03094929 0.03350533
## [5005,] 0.03094929 0.03354615
## [5006,] 0.03094929 0.03354615
## [5007,] 0.03094929 0.03354615
## [5008,] 0.03094929 0.03354615
## [5009,] 0.03094929 0.03354615
```

```
mean(c(draws[, 1:2]))
```

```
## [1] 0.03258386
```

```
median(c(draws[, 1:2]))
```

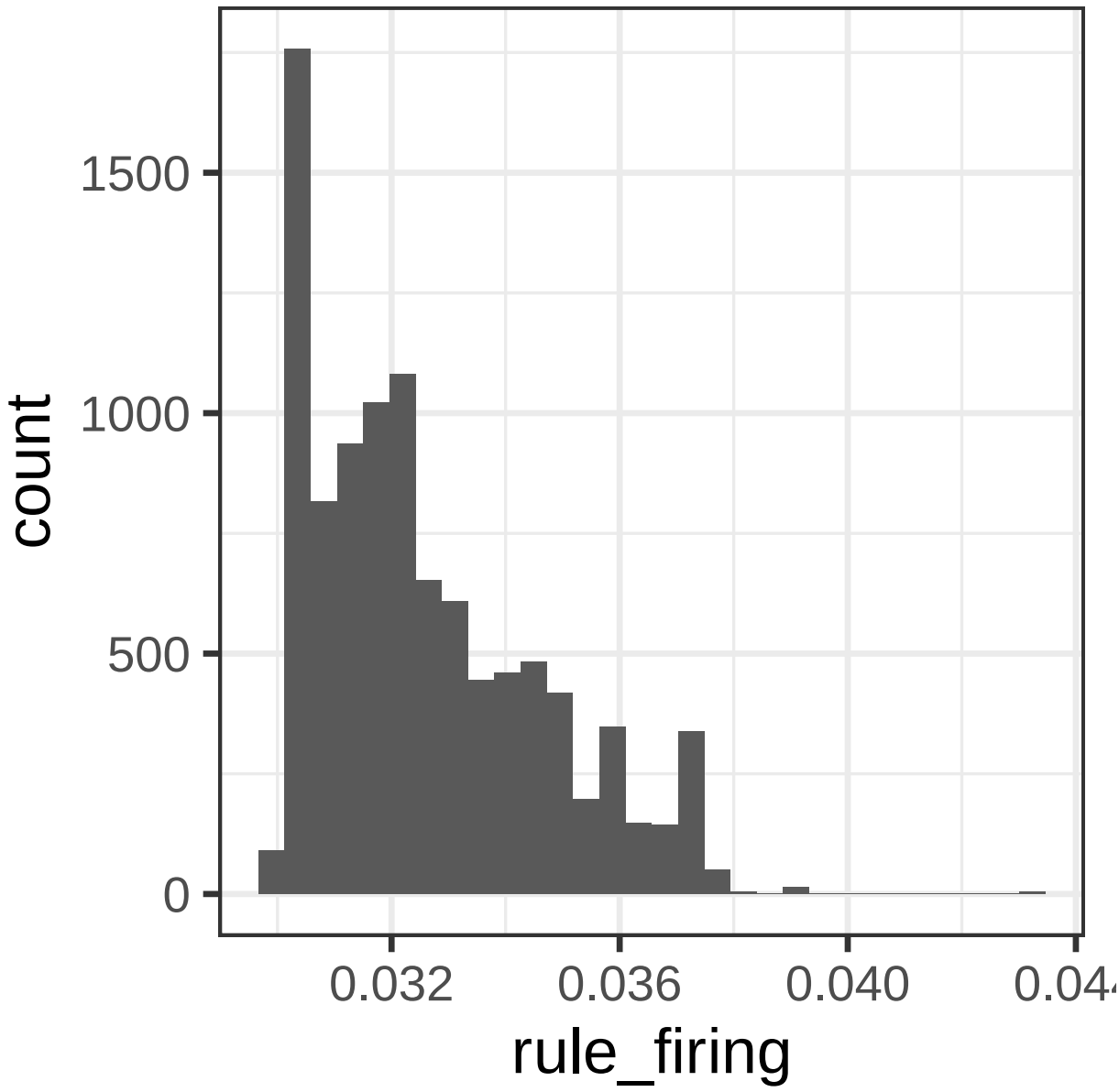
```
## [1] 0.03209849
```

```
sd(c(draws[, 1:2]))
```

```
## [1] 0.002003408
```

```
g1 <- ggplot(data.frame(rule_firing = c(draws[, 1:2])), aes(rule_firing))
g1 <- g1 + geom_histogram() + theme_bw(28)
g1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("gg1-rf.pdf", width = 20, height = 12)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

4.4 Weight

Rhat:

```
draws <- createdraws("weight")

Rhat(draws)

## [1] 1.010444
```

Mean etc.

```
tail(draws)
```

```
##             [,1]     [,2]
## [5004,] 25.07908 11.08713
## [5005,] 12.75977 11.08713
## [5006,] 12.75977 27.17773
## [5007,] 16.89661 11.15825
## [5008,] 17.30988 21.06276
## [5009,] 29.46745 27.63666
```

```
mean(c(draws[, 1:2]))
```

```
## [1] 35.37922
```

```
median(c(draws[, 1:2]))
```

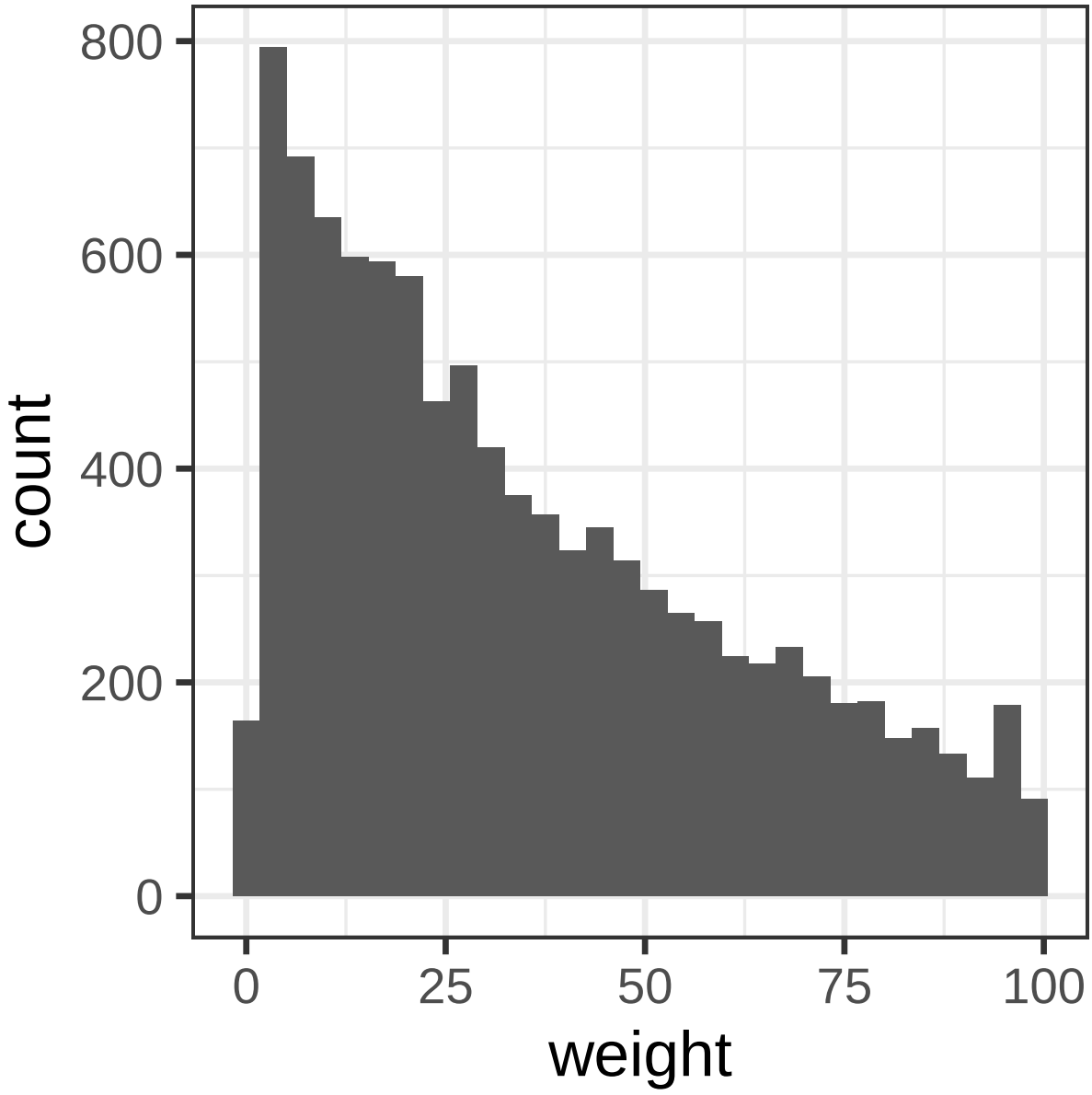
```
## [1] 28.87603
```

```
sd(c(draws[, 1:2]))
```

```
## [1] 26.45121
```

```
g1 <- ggplot(data.frame(weight = c(draws[, 1:2])), aes(weight))
g1 <- g1 + geom_histogram() + theme_bw(28)
g1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("gg1-weight.pdf", width = 20, height = 12)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

4.5 Std

Rhat:

```
draws <- createdraws("std")

Rhat(draws)

## [1] 1.004977
```

Mean etc.

```
tail(draws)
```

```
##             [,1]     [,2]
## [5004,] 16.98274 13.44250
## [5005,] 16.98274 13.03193
## [5006,] 16.98274 22.11548
## [5007,] 16.98274 22.11548
## [5008,] 16.98274 22.11548
## [5009,] 16.98274 17.42381
```

```
mean(c(draws[, 1:2]))
```

```
## [1] 16.07252
```

```
median(c(draws[, 1:2]))
```

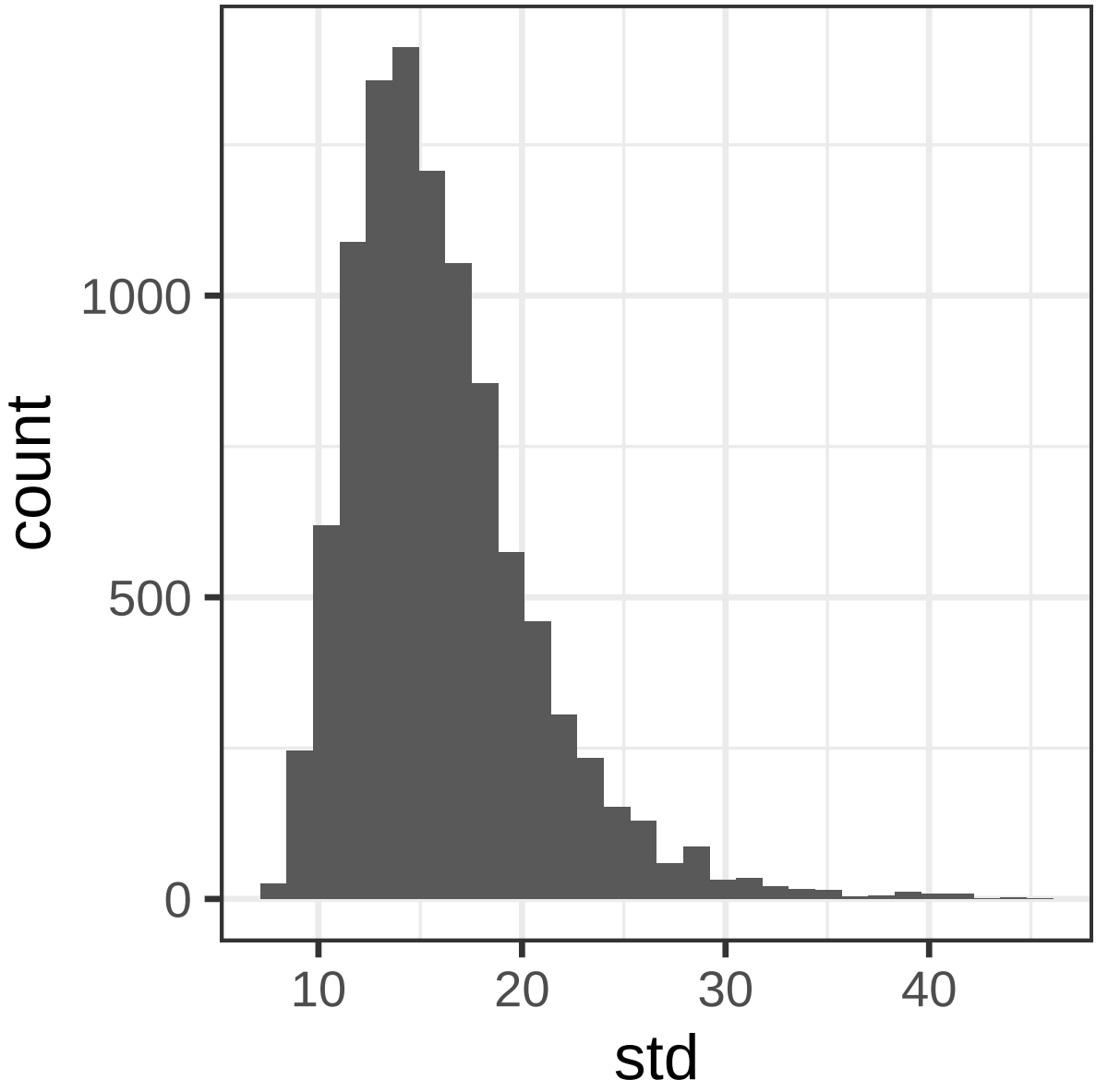
```
## [1] 15.22213
```

```
sd(c(draws[, 1:2]))
```

```
## [1] 4.627666
```

```
g1 <- ggplot(data.frame(std = c(draws[, 1:2])), aes(std))
g1 <- g1 + geom_histogram() + theme_bw(28)
g1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("gg1-std.pdf", width = 20, height = 12)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```