1 Introduction about Credibility, Models and Parameters

Diego

Table of contents

1 Bayesian inference is reallocation of credibility across probabilities

```
pacman::p_load(tidyverse, here, patchwork,brms, tidybayes)
```

First, we create a data object.

```
d <- crossing(iteration = 1:3,
              stage = factor(c("Prior","Posterior"),
                             levels = c("Prior","Posterior"))) %>%
  expand(nesting(iteration, stage),
         Possibilities = LETTERS[1:4]) %>%
  mutate(Credibility = c(rep(.25, times = 4),
                         0, rep(1/3, times=3),
                         0, rep(1/3, times=3),
                         rep(c(0,.5), each=2),
                         rep(c(0,.5), each=2),
                         rep(0,times = 3),1))
head(d)
## # A tibble: 6 x 4
##   iteration stage     Possibilities Credibility
##       <int> <fct>     <chr>                <dbl>
## 1         1 Prior     A                    0.25
## 2         1 Prior     B                    0.25
## 3         1 Prior     C                    0.25
## 4         1 Prior     D                    0.25
## 5         1 Posterior A                    0
## 6         1 Posterior B                    0.333
```

Second, we create two supplemental data frames. One will supply the coordinates for the annotation in the plot. Another will supply the coordinates for the arrows.

```
text <- tibble(
  Possibilities = "B",
  Credibility = 0.75,
  label = str_c(LETTERS[1:3], " is\nimpossible"),
  iteration = 1:3,
  stage = factor("Posterior", levels = c("Prior","Posterior"))
)

arrow <- tibble(
  Possibilities = LETTERS[1:3],
```

```
  iteration = 1:3
) %>%
  expand(nesting(Possibilities, iteration),
         Credibility = c(0.6,0.01)) %>%
  mutate(stage = factor("Posterior",
                        levels = c("Prior","Posterior")))
```

Now, show the transformation from prior to posterior.

```
d %>%
  ggplot(aes(x = Possibilities, y = Credibility)) +
  geom_col(color = "grey30", fill = "grey30") +
  # annotation in the bottom row
  geom_text(data = text,
            aes(label = label)) +
  # arrows in the bottom row
  geom_line(data = arrow,
            arrow = arrow(length = unit(0.30, "cm"),
                          ends = "first", type = "closed")) +
  facet_grid(stage ~ iteration) +
  theme(axis.ticks.x = element_blank(),
        panel.grid = element_blank(),
        strip.text.x = element_blank())
```

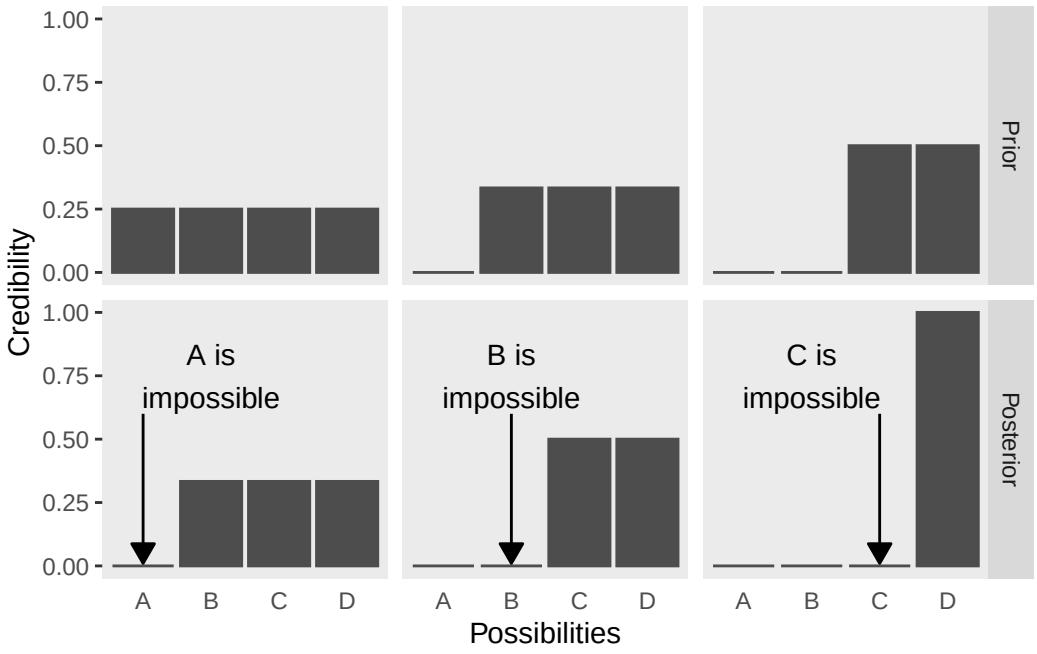

Figure 1: Reallocation from Prior to Posterior-I

Overall, when we have a prior, which shows that the probability of four events is identical, and then we receive more information indicating that A, B and C is impossible, so we will get the ultimate posterior.

```
# primary data
crossing(stage = factor(c("Prior", "Posterior"),
                        levels = c("Prior", "Posterior")),
         Possibilities = LETTERS[1:4]) %>%
  mutate(Credibility = c(rep(0.25, times = 4),
                         rep(0,times = 3),
                         1)) %>%

  # plot!
  ggplot(aes(x = Possibilities, y = Credibility)) +
  geom_col(color = "grey30", fill = "grey30") +
  # annotation in the bottom panel
  geom_text(data = tibble(
    Possibilities = "B",
    Credibility   = .8,
    label         = "D is\nresponsible",
    stage         = factor("Posterior", levels = c("Prior", "Posterior"))
```

```
), aes(label = label)
) +
# the arrow
geom_line(data = tibble(
  Possibilities = LETTERS[c(4, 4)],
  Credibility   = c(.25, .99),
  stage         = factor("Posterior", levels = c("Prior", "Posterior"))
),
arrow = arrow(length = unit(0.30, "cm"), ends = "last", type = "closed"),
color = "grey92") +
facet_wrap(~ stage, ncol = 1) +
theme(axis.ticks.x = element_blank(),
      panel.grid = element_blank())
```

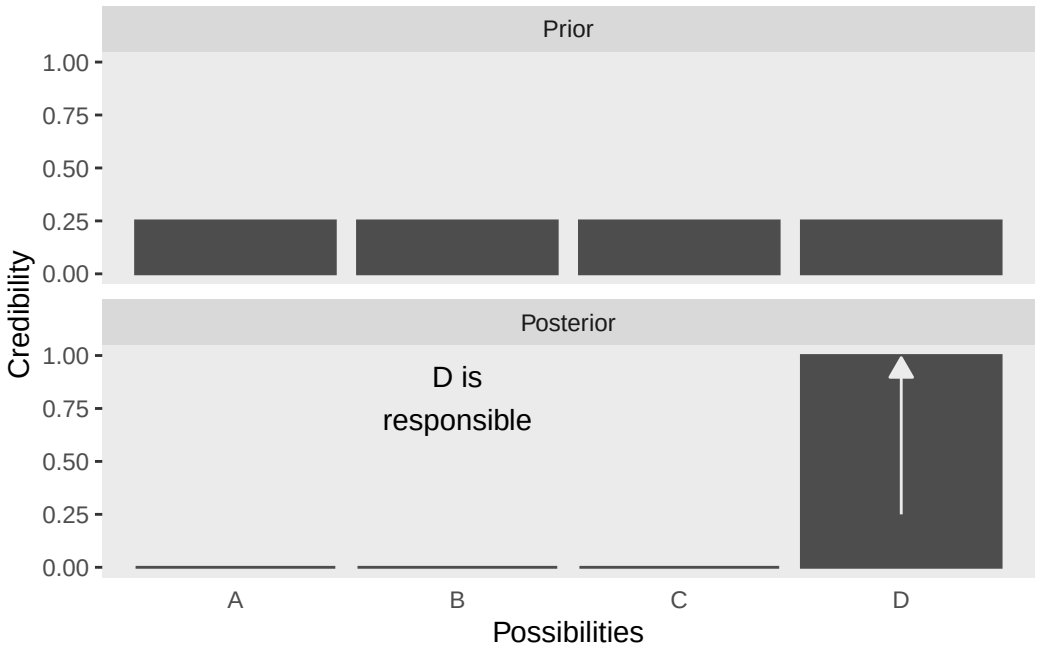

Figure 2: Reallocation from Prior to Posterior-II

1.1 Data are noisy and inferences are probabilistic

The probability of each event is not a single value, instead it would be with a specific probability function or distribution. Here, we assume that each possibilities follows the normal distribution.

```
# data
tibble(mu = 1:4,
       p  = .25) %>%
  expand(nesting(mu, p),
         x = seq(from = -2, to = 6, by = .1)) %>%
  mutate(density = dnorm(x, mean = mu, sd = 1.2)) %>%
  mutate(d_max = max(density)) %>%
  mutate(rescale = p / d_max) %>%
  mutate(density = density * rescale) %>%

  # plot!
  ggplot(aes(x = x)) +
  geom_col(data = . %>% distinct(mu, p),
           aes(x = mu, y = p),
           fill = "grey67", width = 1/3) +
  geom_line(aes(y = density, group = mu)) +
  scale_x_continuous(breaks = 1:4) +
  scale_y_continuous(breaks = 0:5 / 5) +
  coord_cartesian(xlim = c(0, 5),
                  ylim = c(0, 1)) +
  labs(title = "Prior",
       x = "Possibilities",
       y = "Credibility") +
  theme(axis.ticks.x = element_blank(),
        panel.grid = element_blank())
```

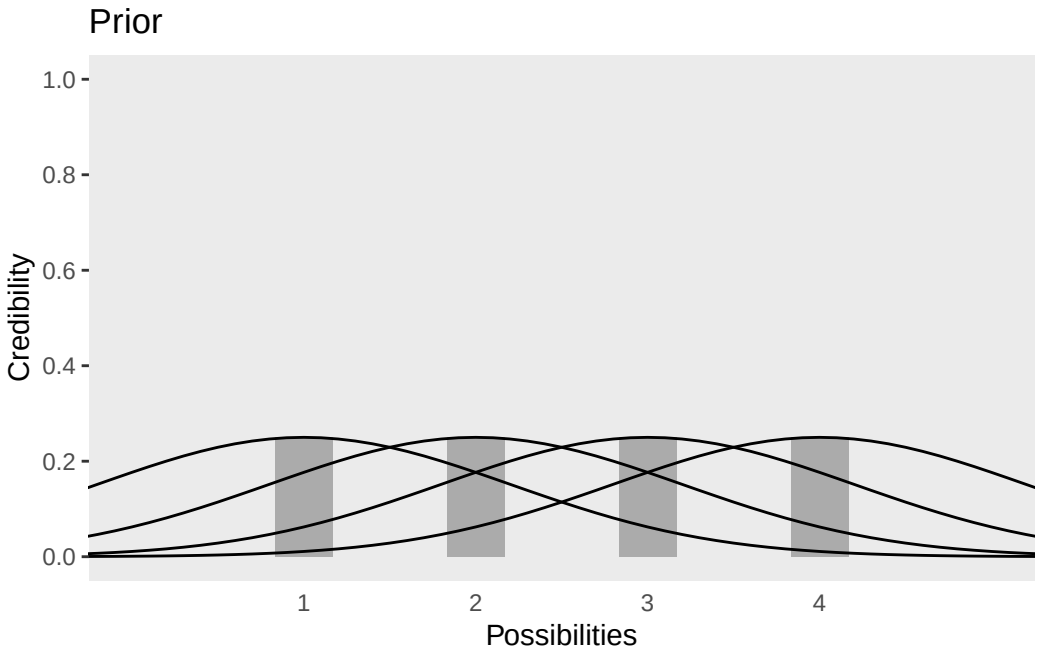

Figure 3: Reallocation from Prior to Posterior-III

After we have more information, we will get the posterior.

```
tibble(mu = 1:4,
       p  = c(.11, .56, .31, .02)) %>%
  expand(nesting(mu, p),
         x = seq(from = -2, to = 6, by = .1)) %>%
  mutate(density = dnorm(x, mean = mu, sd = 1.2)) %>%
  mutate(d_max = max(density)) %>%
  mutate(rescale = p / d_max) %>%
  mutate(density = density * rescale) %>%

  # plot!
  ggplot() +
  geom_col(data = . %>% distinct(mu, p),
           aes(x = mu, y = p),
           fill = "grey67", width = 1/3) +
  geom_line(aes(x = x, y = density, group = mu)) +
  geom_point(data = tibble(x = c(1.75, 2.25, 2.75), y = 0),
             aes(x = x, y = y),
             size = 3, color = "grey33", alpha = 3/4) +
  scale_x_continuous(breaks = 1:4) +
```

```
  scale_y_continuous(breaks = 0:5 / 5) +
  coord_cartesian(xlim = c(0, 5),
                  ylim = c(0, 1)) +
  labs(title = "Posterior",
       x = "Possibilities",
       y = "Credibility") +
  theme(axis.ticks.x = element_blank(),
        panel.grid = element_blank())
```

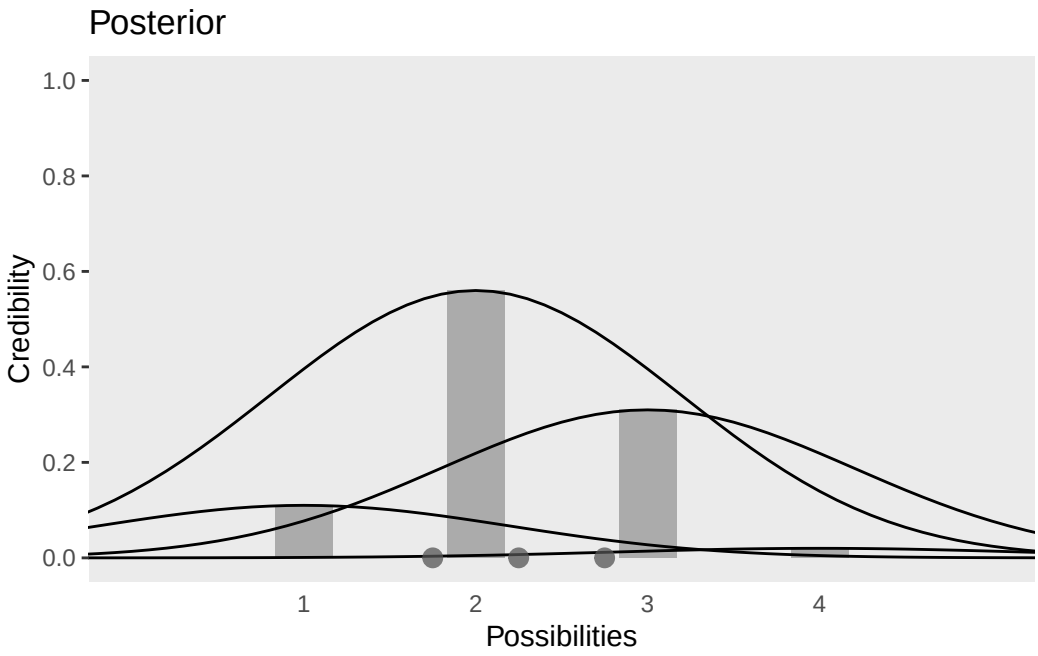

Figure 4: Reallocation from Prior to Posterior-IV

> **!** Important
>
> The essence of Bayesian inference is reallocation of credibility across possibilities. The distribution of credibility initially reflects prior knowledge about the possibilities. Then new data are observed, and the credibility is reallocated. Possibilities that are consistent with the data garner more credibility, while possibilities that are not consistent with the data lose credibility.

1.2 Possibilities are parameter values in descriptive models

Suppose we have a data shown below Figure 5 .

```r
# set the seed to make the simulation reproducible
set.seed(2)
# simulate the data with `rnorm()`
d <- tibble(x = rnorm(2000, mean = 10, sd = 5))

# plot!
ggplot(data = d, aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 1, fill = "grey67",
                 color = "grey92", size = 1/10) +
  geom_line(data = tibble(x = seq(from = -6, to = 26, by = .01)),
            aes(x = x, y = dnorm(x, mean = 10, sd = 5)),
            color = "grey33") +
  coord_cartesian(xlim = c(-5, 25)) +
  labs(subtitle = "The candidate normal distribution\nhas a mean of 10 and SD of 5.",
       x = "Data Values",
       y = "Data Probability") +
  theme(panel.grid = element_blank())
```

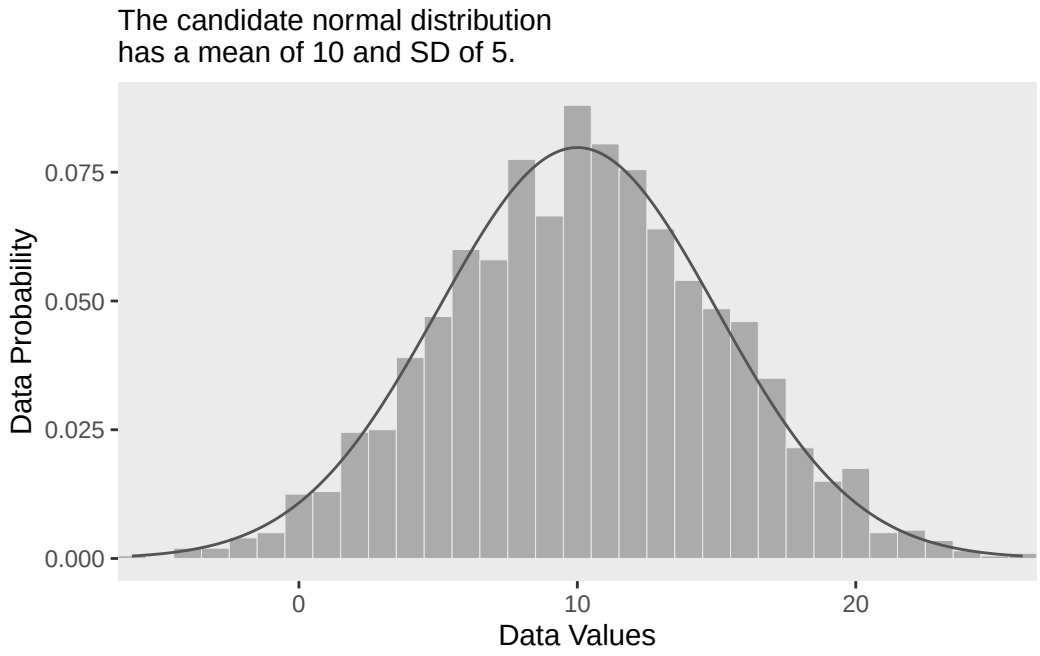

Figure 5: Distribution of prior

1.3 Steps of Bayesian data analysis

> **!** Steps of Bayesian analysis
>
> 1. Identify the data relevant to the research questions.
>   1. the measurement scales of the data
>   2. type of data variables to be predicted and supposed to act as predictors
> 2. Define a descriptive model for the relevant data. The mathematical form and its parameters should be meaningful and appropriate to the theoretical purposes.
> 3. Specify a prior distribution on the parameters.
> 4. Use Bayesian inference to re-allocate credibility across parameter values. Interpret the posterior distribution with respect to theoretically meaningful issues.
> 5. Check the posterior predictions mimic the data with reasonable accuracy. If not, consider a different descriptive model.

Suppose we have a dataset:

```r
HtWtDataGenerator <- function(nSubj, rndsd = NULL, maleProb = 0.50) {
  # Random height, weight generator for males and females. Uses parameters from
  # Brainard, J. & Burmaster, D. E. (1992). Bivariate distributions for height and
  # weight of men and women in the United States. Risk Analysis, 12(2), 267-275.
  # Kruschke, J. K. (2011). Doing Bayesian data analysis:
  # A Tutorial with R and BUGS. Academic Press / Elsevier.
  # Kruschke, J. K. (2014). Doing Bayesian data analysis, 2nd Edition:
  # A Tutorial with R, JAGS and Stan. Academic Press / Elsevier.

  # require(MASS)

  # Specify parameters of multivariate normal (MVN) distributions.
  # Men:
  HtMmu   <- 69.18
  HtMsd   <- 2.87
  lnWtMmu <- 5.14
  lnWtMsd <- 0.17
  Mrho    <- 0.42
  Mmean   <- c(HtMmu, lnWtMmu)
  Msigma  <- matrix(c(HtMsd^2, Mrho * HtMsd * lnWtMsd,
                      Mrho * HtMsd * lnWtMsd, lnWtMsd^2), nrow = 2)
  # Women cluster 1:
  HtFmu1   <- 63.11
```

```
  HtFsd1   <- 2.76
  lnWtFmu1 <- 5.06
  lnWtFsd1 <- 0.24
  Frho1    <- 0.41
  prop1    <- 0.46
  Fmean1   <- c(HtFmu1, lnWtFmu1)
  Fsigma1  <- matrix(c(HtFsd1^2, Frho1 * HtFsd1 * lnWtFsd1,
                       Frho1 * HtFsd1 * lnWtFsd1, lnWtFsd1^2), nrow = 2)
  # Women cluster 2:
  HtFmu2   <- 64.36
  HtFsd2   <- 2.49
  lnWtFmu2 <- 4.86
  lnWtFsd2 <- 0.14
  Frho2    <- 0.44
  prop2    <- 1 - prop1
  Fmean2   <- c(HtFmu2, lnWtFmu2)
  Fsigma2  <- matrix(c(HtFsd2^2, Frho2 * HtFsd2 * lnWtFsd2,
                       Frho2 * HtFsd2 * lnWtFsd2, lnWtFsd2^2), nrow = 2)

  # Randomly generate data values from those MVN distributions.
  if (!is.null(rndsd)) {set.seed(rndsd)}
  datamatrix <- matrix(0, nrow = nSubj, ncol = 3)
  colnames(datamatrix) <- c("male", "height", "weight")
  maleval <- 1; femaleval <- 0 # arbitrary coding values
  for (i in 1:nSubj)  {
    # Flip coin to decide sex
    sex <- sample(c(maleval, femaleval), size = 1, replace = TRUE,
                  prob = c(maleProb, 1 - maleProb))
    if (sex == maleval) {datum = MASS::mvrnorm(n = 1, mu = Mmean, Sigma = Msigma)}
    if (sex == femaleval) {
      Fclust = sample(c(1, 2), size = 1, replace = TRUE, prob = c(prop1, prop2))
      if (Fclust == 1) {datum = MASS::mvrnorm(n = 1, mu = Fmean1, Sigma = Fsigma1)}
      if (Fclust == 2) {datum = MASS::mvrnorm(n = 1, mu = Fmean2, Sigma = Fsigma2)}
    }
    datamatrix[i, ] = c(sex, round(c(datum[1], exp(datum[2])), 1))
  }

  return(datamatrix)
} # end function
```

Assume we want to have 57 samples and the probable is set to be 0.5 based on those from men.

```r
set.seed(2)

d <- HtWtDataGenerator(nSubj = 57, maleProb = 0.5) %>%
  as_tibble()
d %>% head()
## # A tibble: 6 x 3
##    male height weight
##   <dbl>  <dbl>  <dbl>
## 1     0   62.6   109.
## 2     0   63.3   154.
## 3     1   71.8   155
## 4     0   67.9   146.
## 5     0   64.4   135.
## 6     0   66.8   119
```

We will use the **Hamiltonian Monte Carlo (HMC)** method. Traditionally, the use of diffuse and non-informative priors is discouraged with HMC, as is the uniform distribution of sigma. Here, we will use weakly-regularizing priors for the intercept and slope and a half Cauchy with a fairly large scale parameter for σ.

```r
fit1.1 <- brm(data = d,
              family = gaussian,
              weight ~ 1+height,
              prior = c(prior(normal(0,100), class = Intercept),
                        prior(normal(0,100), class = b),
                        prior(cauchy(0,10), class = sigma)),
              chains = 4, cores = 4, iter = 2000, warmup = 1000,
              seed = 2,
              file = here("fits","fit01.01"))
```

To visualize the model,

```r
# extract the posterior draws
draws <- as_draws_df(fit1.1)

# this will subset the output
n_lines <- 150

# plot!
draws %>%
  slice(1:n_lines) %>%
```

```
ggplot() +
geom_abline(aes(intercept = b_Intercept, slope = b_height, group = .draw),
            color = "grey50", size = 1/4, alpha = .3) +
geom_point(data = d,
           aes(x = height, y = weight),
           shape = 1) +
labs(
  subtitle = eval(substitute(paste("Data with",
                                   n_lines,
                                   "credible regression lines"))),
     x = "Height in inches",
     y = "Weight in pounds") +
coord_cartesian(xlim = c(55, 80),
                ylim = c(50, 250)) +
theme(panel.grid = element_blank())
```

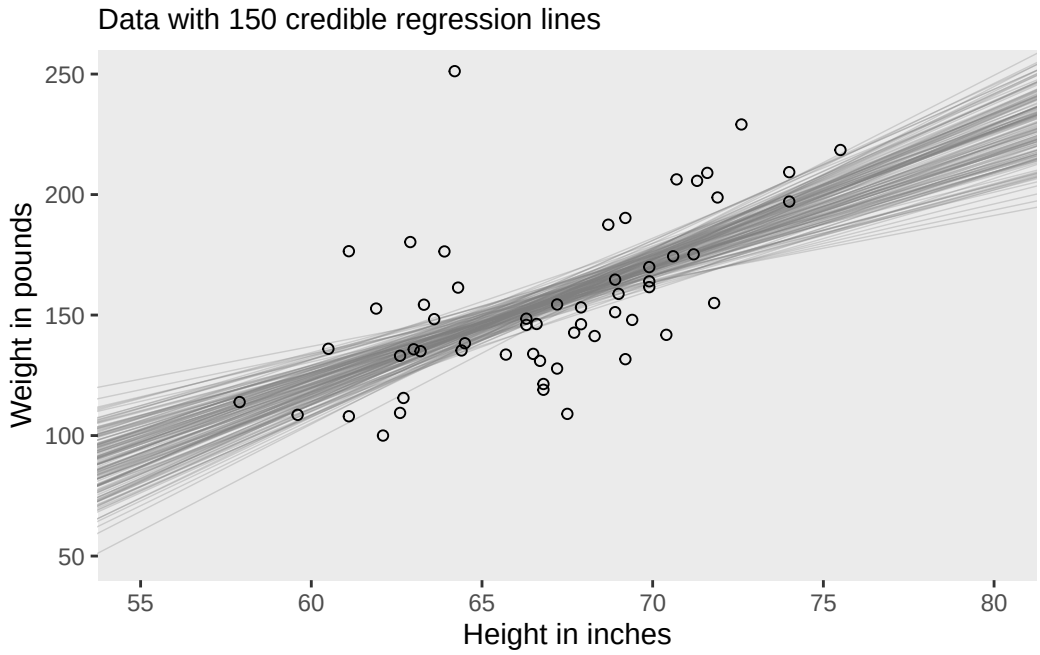

Figure 6: Model 1.1

To show the posterior distribution, we will mark off the mode and 95% highest density interval (HDI)

```
draws %>%
  ggplot(aes(x = b_height, y = 0)) +
  stat_histinterval(point_interval = mode_hdi, .width = .95,
                    fill = "grey67", slab_color = "grey92",
                    breaks = 40, slab_size = .2, outline_bars = T) +
  scale_y_continuous(NULL, breaks = NULL) +
  coord_cartesian(xlim = c(0, 8)) +
  labs(title = "The posterior distribution",
       subtitle = "The mode and 95% HPD intervals are\nthe dot and horizontal line at the
       x = expression(beta[1]~(slope))) +
  theme(panel.grid = element_blank())
```

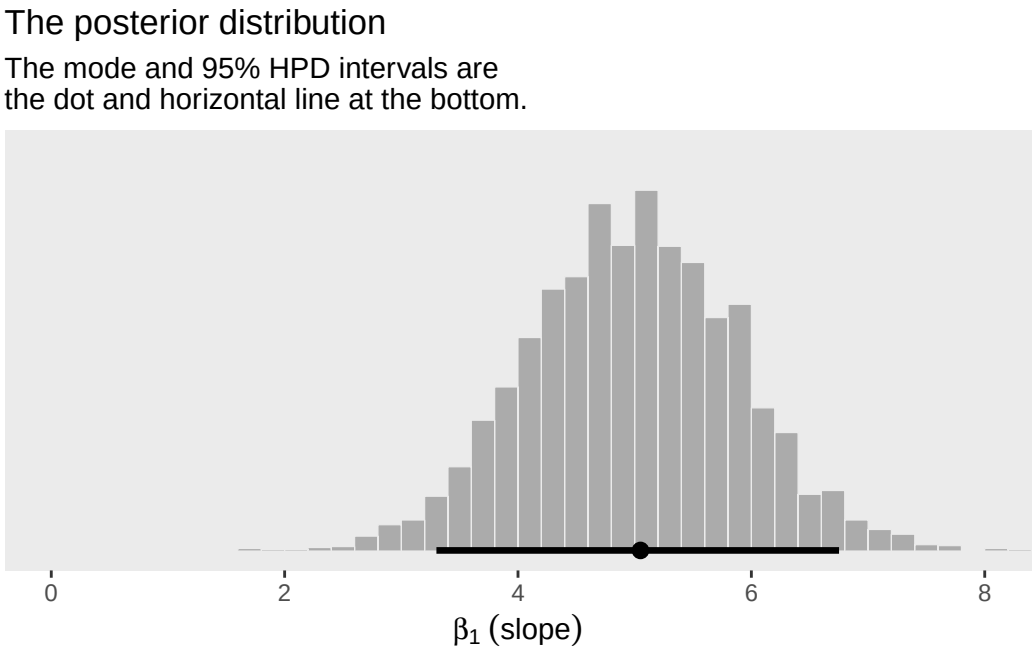

Figure 7: Slope of Model 1.1

To show the percentile-based 95% intervals of the model,

```
nd <- tibble(height = seq(from = 53, to = 81, length.out = 20))

predict(fit1.1, newdata = nd) %>%
  data.frame() %>%
  bind_cols(nd) %>%
```

```
  ggplot(aes(x = height)) +
  geom_pointrange(aes(y = Estimate, ymin = Q2.5, ymax = Q97.5),
                  color = "grey67", shape = 20) +
  geom_point(data = d,
             aes(y = weight),
             alpha = 2/3) +
  labs(subtitle = "Data with the percentile-based 95% intervals and\nthe means of the post
       x = "Height in inches",
       y = "Weight in inches") +
  theme(panel.grid = element_blank())

nd <- tibble(height = seq(from = 53, to = 81, length.out = 30))

predict(fit1.1, newdata = nd) %>%
  data.frame() %>%
  bind_cols(nd) %>%

  ggplot(aes(x = height)) +
  geom_ribbon(aes(ymin = Q2.5, ymax = Q97.5), fill = "grey75") +
  geom_line(aes(y = Estimate),
            color = "grey92") +
  geom_point(data =  d,
             aes(y = weight),
             alpha = 2/3) +
  labs(subtitle = "Data with the percentile-based 95% intervals and\nthe means of the post
       x = "Height in inches",
       y = "Weight in inches") +
  theme(panel.grid = element_blank())
```

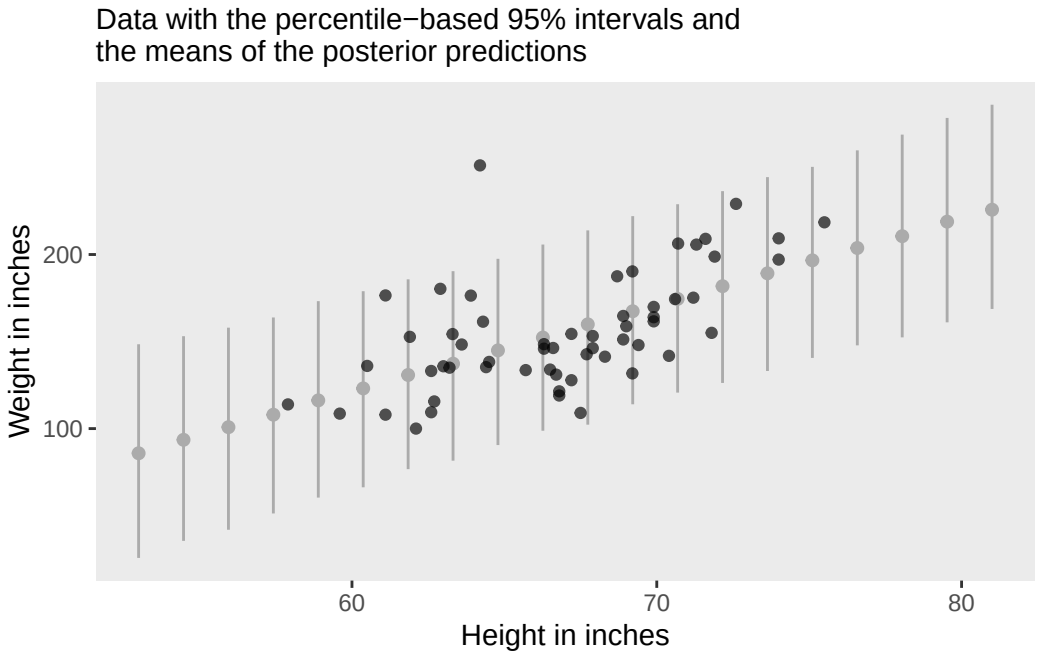

Figure 8: Percentile-based 95% intervals of Model 1.1

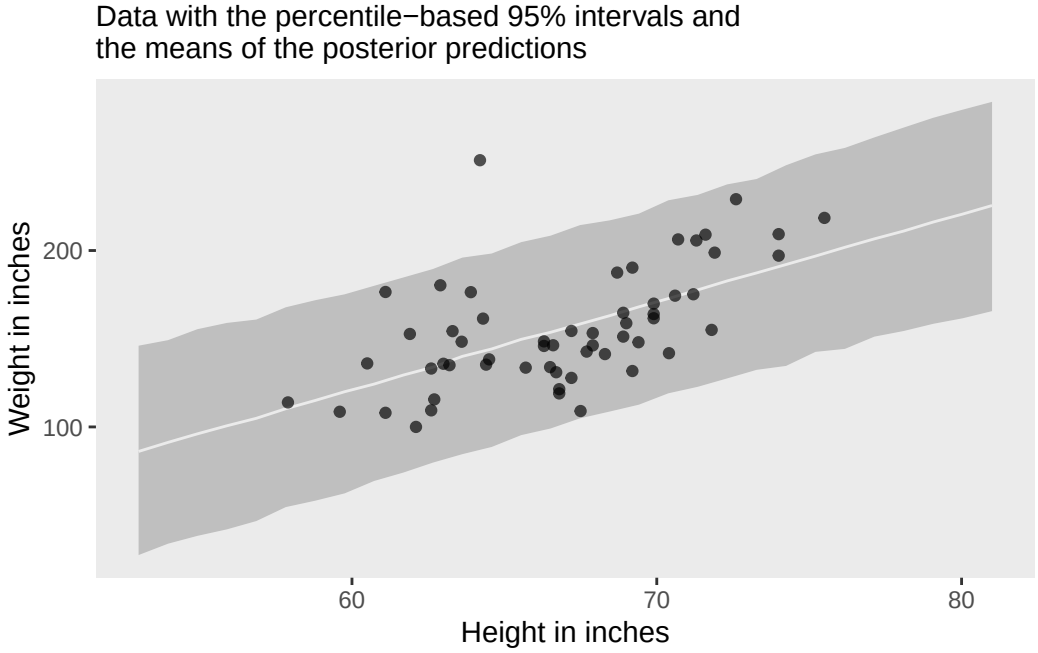

Figure 9: Percentile-based 95% intervals of Model 1.1