Homework_A02

Dan Swart, CPA (ret)

January 19, 2026

Table of contents

Workflow: Bayesian Globe Tossing Analysis (Tidy Approach)

STATISTICAL RETHINKING 2026

HOMEWORK A2

A2. Suppose the globe tossing data (Chapter 2) had turned out to be 3 water and 11 land. Construct the posterior distribution. Then using the poste- rior distribution, compute the posterior predictive distribution for the next 5 tosses of the same globe

Problem Summary

Data: 3 water (W), 11 land (L), total n = 14

Goal 1: Posterior distribution for p (probability of water)

Goal 2: Posterior predictive distribution for next 5 tosses

Step 1: Define the Prior

Choose a prior for p. Standard choice is uniform: Beta(1, 1).

```
# get data
data <- rep(c("W", "L"), times = c(3, 11))

# get counts
W <- sum(data == "W")  # 3
L <- sum(data == "L")  # 11

# Prior parameters (uniform)
a <- 1
b <- 1

# 'Naive' probability for p for W as point estimate
naivep <- 3/14   # 0.2142857
```

The 'naive' proportion of water is 3/14 = 0.2142857

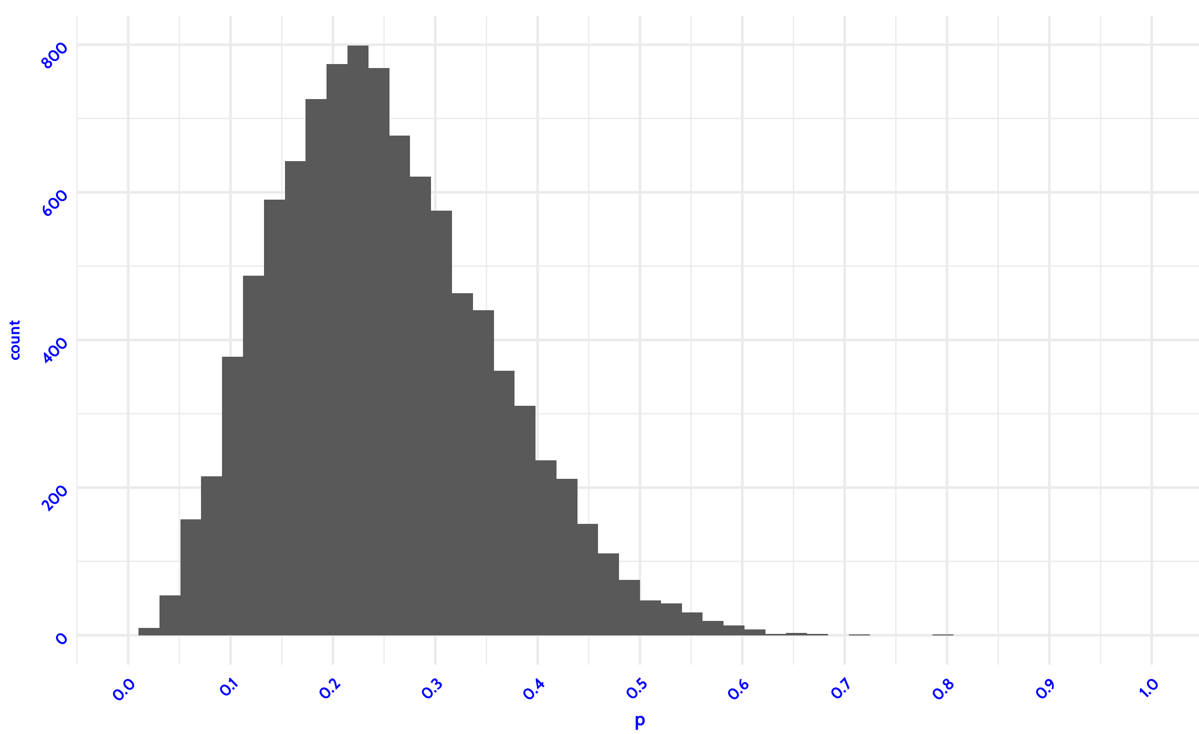
Step 2: Compute the Posterior and Visualize It

Prior: Beta(a, b) = Beta(1, 1)

Posterior: Beta(a + W, b + L) = Beta(1 + 3, 1 + 11) = Beta(4, 12)

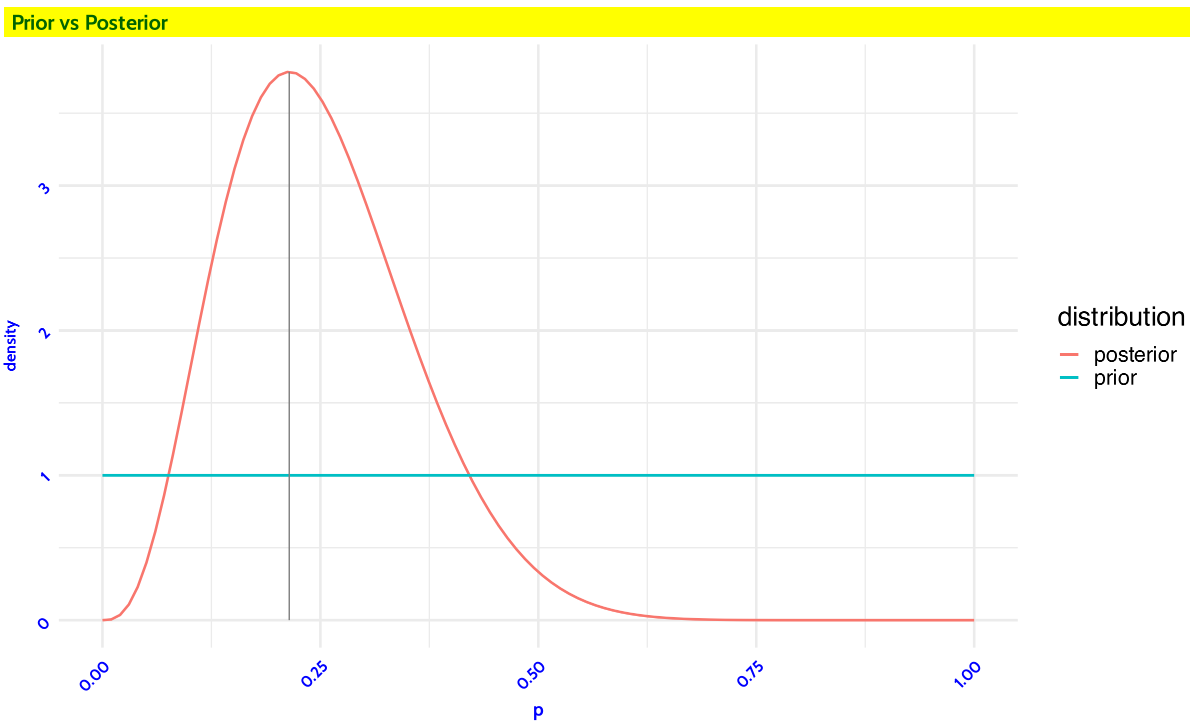
```
# Update uniform distribution to include new data
```

```
# Posterior: Beta(a + W, b + L) = Beta(4, 12)
samples <- tibble::tibble(
  p = stats::rbeta(n = 10000, shape1 = a + W, shape2 = b +
L)
)   # take random 'x' values from beta with specified
parameters and return the height of the curve at each
value


# Visualize posterior with histogram
samples |>
  ggplot2::ggplot(ggplot2::aes(x = p)) +
  ggplot2::geom_histogram(bins = 50) +
  ggplot2::scale_x_continuous(
    breaks = seq(0, 1, by = 0.1),
    limits = c(0, 1)
  )
```



```
# Get mode of Beta(4, 12) to visualize peak point of
updated distrib
mode_p <- (4 - 1) / (4 + 12 - 2)  # gives the peak of the
density function
peak_density <- stats::dbeta(mode_p, shape1 = 4, shape2 =
12)  # returns the height of the curve (density) at the
single point = mode_p


# Now visualize with density curves
densities <- tibble::tibble(
  p = seq(0, 1, length.out = 100),  # 100 evenly spaced x
values between 0 - 1
  prior = stats::dbeta(p, shape1 = 1, shape2 = 1),  #
returns the height of the curve (density) at all the x
values of the uniform shape (=1)
  posterior = stats::dbeta(p, shape1 = 4, shape2 = 12) #
returns the height of the curve (density) at all the x
values of the posterior shape from 0 - 1
) |>
  tidyr::pivot_longer(
    cols = c(prior, posterior),
    names_to = "distribution",
    values_to = "density"
  )

# # view it
# densities     # testing

# plot it
densities|>
  ggplot2::ggplot(ggplot2::aes(x = p, y = density, color =
distribution)) +
  ggplot2::geom_line() +
  ggplot2::annotate(
```

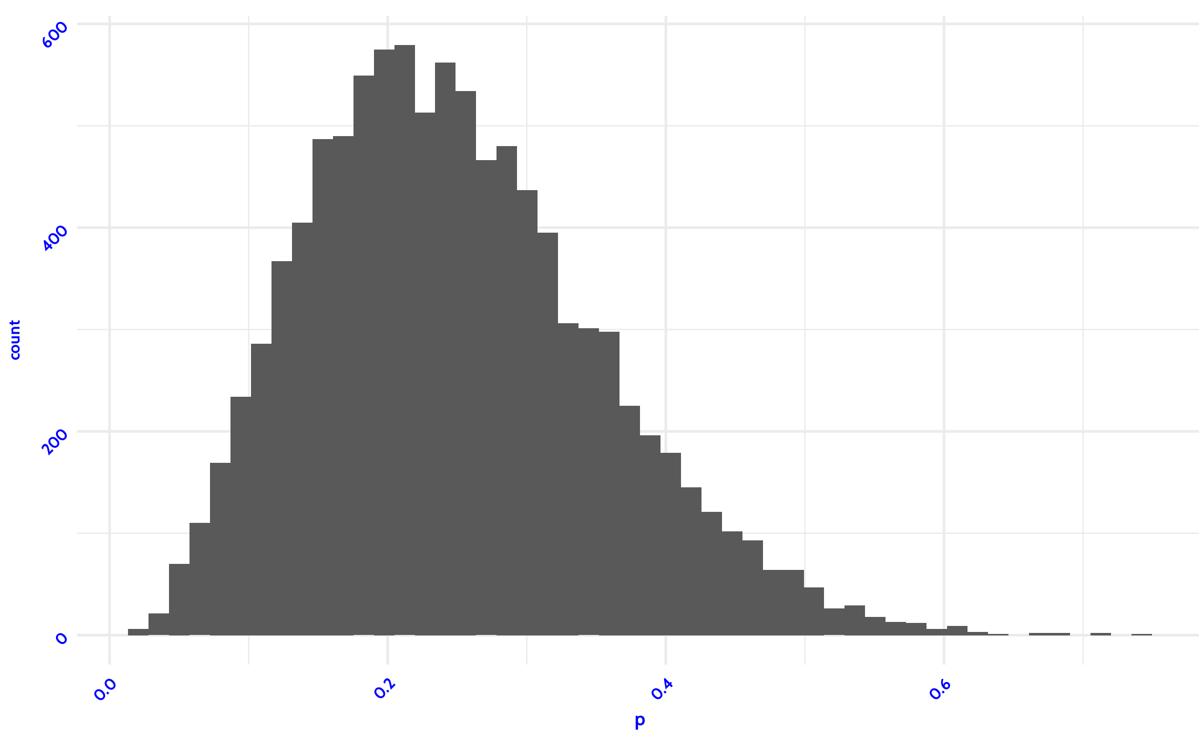
```r
    "segment",
    x = mode_p, xend = mode_p,
    y = 0, yend = peak_density,
    color = "gray50",
    linewidth = 0.5
  ) +
  ggplot2::labs(title = "Prior vs Posterior") +
  ggplot2::theme(legend.position = "right")
```



Step 3: Sample from the Posterior

Draw many samples (e.g., 10,000) from the posterior distribution and return height of curve at each draw of x-axis value.

```
# p is the probability of water on a single globe toss.
shape1 = (α)Prior + 'successes'     shape2 = (β)Prior +
'failures'

samples_from_post <- tibble::tibble(
  p = stats::rbeta(n = 10000, shape1 = 4, shape2 = 12)
)   # rbeta() = take n random samples as you walk along
the curve created from a Beta distrib with empirically
specified parameters (the posterior).  shape1 =
'successes'   shape2 = 'failures' and return the height of
the curve at each sample point on the curve

# visualize it with histogram (don't use density curves;
they are for continuous data)
samples_from_post |>
  ggplot2::ggplot(ggplot2::aes(x = p)) +
  ggplot2::geom_histogram(bins = 50)
```



Step 5: Generate Posterior Predictive Distribution

For each of the 10k posterior samples of p, simulate 5 globe tosses and count water.
This is vectorized: one call handles all samples.

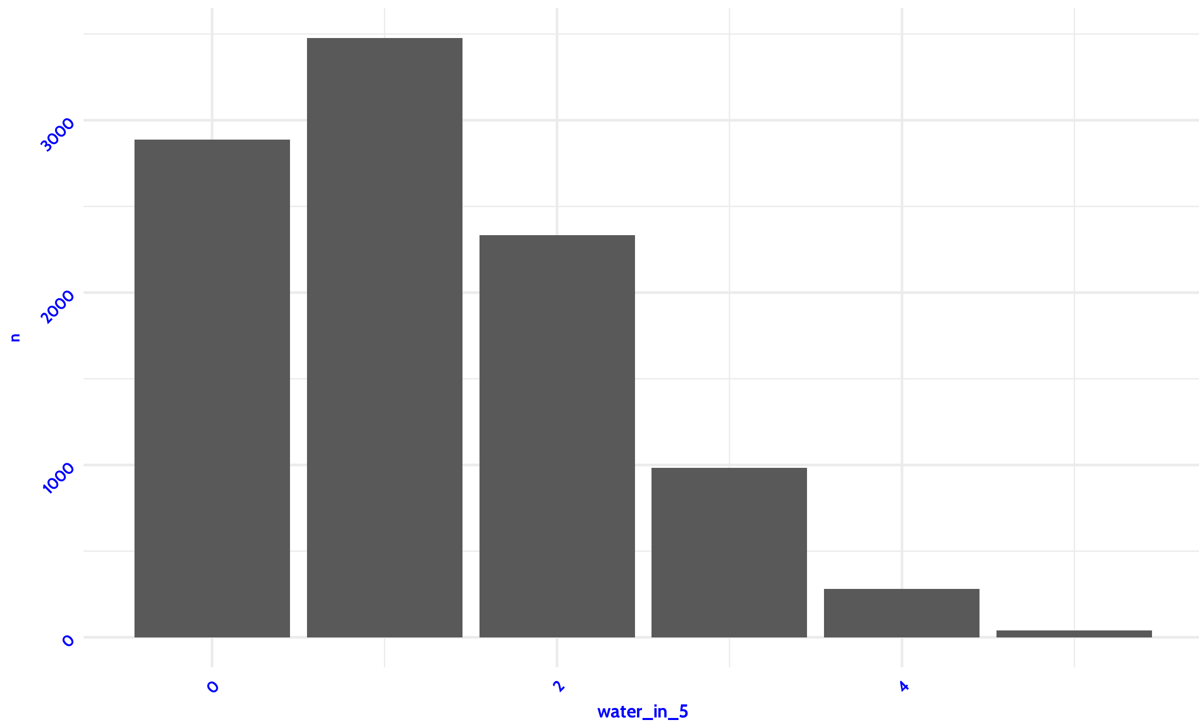
```
# Simulate 5 tosses for each of the 10,000 posterior
sample
predictions <- samples_from_post |>
  dplyr::mutate(
    water_in_5 = stats::rbinom(n = dplyr::n(), size = 5,
prob = p)
  )  # rbinom() — simulate binomial draws (n = 5, prob =
posterior samples)

# Save the summary
pred_summary <- predictions |>
  dplyr::count(water_in_5) |>
  dplyr::mutate(prob = n / sum(n))

# Display results in table
pred_summary |>
  flextable::flextable() |>
  flextable::add_header_lines("Posterior Predictive
Distribution") |>
  flextable::color(i = 1, part = "header", color = "blue")
|>
  flextable::italic(i = 1, part = "header") |>
  flextable::align(i = 1, part = "header", align = "left")
|>
  flextable::fontsize(i = 1, part = "header", size = 12) |
```

```
>
  flextable::bg(i = 1, part = "header", bg = "white") |>
  flextable::bg(i = 2, part = "header", bg = "palegreen")
|>
  flextable::autofit()
```

Posterior Predictive Distribution

water_in_5	n	prob
0	2 887	0.2887
1	3 477	0.3477
2	2 333	0.2333
3	982	0.0982
4	281	0.0281
5	40	0.0040

```
# Display results in bar chart - Each bar shows "how many
of my 10,000 simulated samples of 5 'futures' had 0, 1,
2... waters, expressed as a proportion"
predictions |>
  dplyr::count(water_in_5) |>
  ggplot2::ggplot(ggplot2::aes(x = water_in_5, y = n)) +
  ggplot2::geom_col()
```



Conclusion: the small group size (5) of the draws from the posterior distribution would support the hypothesis that the proportion of water is most likely to be 20% or 0.00%. This reflects the larger uncertainty in the small samples of 5? It is very different from the huge sample of 10,000 in the posterior.

Function Summary

```
tibble::tibble(
  Step = c("3", "3", "4", "5", "5", "6", "6", "7"),
  Function = c("stats::rbeta()", "tibble::tibble()",
"ggplot2::geom_histogram()",
               "stats::rbinom()", "dplyr::mutate()",
"dplyr::count()",
               "dplyr::mutate()", "ggplot2::geom_col()"),
```

```
  Purpose = c("Sample posterior (no tidy alt)", "Store
samples", "Plot histogram",
              "Simulate predictions (no tidy alt)", "Add
prediction column to df",
              "Get counts of each sample result", "Add
column to data frame", "Plot predictive distribution as
bar chart")
) |>
  flextable::flextable() |>
  flextable::add_header_lines("Key R Functions for
Bayesian Globe Analysis") |>
  flextable::color(i = 1, part = "header", color = "blue")
|>
  flextable::italic(i = 1, part = "header") |>
  flextable::align(i = 1, part = "header", align = "left")
|>
  flextable::fontsize(i = 1, part = "header", size = 12) |
>
  flextable::bg(i = 1, part = "header", bg = "white") |>
  flextable::bg(i = 2, part = "header", bg = "palegreen")
|>
  flextable::autofit()
```

Key R Functions for Bayesian Globe Analysis

Step	Function	Purpose
3	stats::rbeta()	Sample posterior (no tidy alt)
3	tibble::tibble()	Store samples
4	ggplot2::geom_histogram()	Plot histogram
5	stats::rbinom()	Simulate predictions (no tidy alt)
5	dplyr::mutate()	Add prediction column to df
6	dplyr::count()	Get counts of each sample result
6	dplyr::mutate()	Add column to data frame
7	ggplot2::geom_col()	Plot predictive distribution as bar chart