STA 3032 — Lab 2: Simulating Bayes' Rule

Ansel Gillman

September 14, 2025

Lab Question 1: Simulating a Medical Test with Prevalence = 1%

Part A: Simulation and confusion matrix

```r
# Parameters
n <- 10000
prev <- 0.01            # prevalence (1%)
sensitivity <- 0.99     # P(+ | disease)
specificity <- 0.95     # P(- | no disease)

# Simulate who has the disease
has_disease <- rbinom(n, 1, prev)

# Simulate test results
test_positive <- ifelse(
  has_disease == 1,
  rbinom(n, 1, sensitivity),
  rbinom(n, 1, 1 - specificity)
)

# Confusion matrix
conf_mat <- table(has_disease, test_positive)
conf_mat
```

```
##            test_positive
## has_disease    0    1
##           0 9360  539
##           1    1  100
```

```r
# Posterior (empirical)
posterior_empirical <- sum(has_disease == 1 & test_positive == 1) / sum(test_positive == 1)
posterior_empirical
```

```
## [1] 0.1564945
```

Answer (Q1A): The confusion matrix is printed above. From it, the probability that a person who tests positive actually has the disease (empirical posterior) is also calculated.

Part B: Theoretical posterior using Bayes' Rule

```
P_D <- prev
P_pos_given_D <- sensitivity
P_pos_given_notD <- 1 - specificity

posterior_theoretical <- (P_pos_given_D * P_D) / (
  P_pos_given_D * P_D + P_pos_given_notD * (1 - P_D)
)
posterior_theoretical
```

```
## [1] 0.1666667
```

Answer (Q1B): The theoretical posterior probability is shown above. It closely matches the empirical simulation, with small differences due to randomness.

Lab Question 2: Changing Prevalence to 10%

Part A: Simulation and confusion matrix

```
prev2 <- 0.10
has_disease2 <- rbinom(n, 1, prev2)

test_positive2 <- ifelse(
  has_disease2 == 1,
  rbinom(n, 1, sensitivity),
  rbinom(n, 1, 1 - specificity)
)

conf_mat2 <- table(has_disease2, test_positive2)
conf_mat2
```

```
##             test_positive2
## has_disease2    0    1
##            0 8419  498
##            1   16 1067
```

```
posterior_empirical2 <- sum(has_disease2 == 1 & test_positive2 == 1) / sum(test_positive2 == 1)
posterior_empirical2
```

```
## [1] 0.6817891
```

Answer (Q2A): The confusion matrix and empirical posterior are displayed above for the 10% prevalence case.

Part B: Theoretical posterior

```r
P_D2 <- prev2
posterior_theoretical2 <- (P_pos_given_D * P_D2) / (
  P_pos_given_D * P_D2 + P_pos_given_notD * (1 - P_D2)
)
posterior_theoretical2
```

```
## [1] 0.6875
```

Answer (Q2B): The theoretical posterior is shown above. Again, it matches the simulation result closely.

Comment: The posterior probability that a person actually has the disease given a positive test result is much higher when prevalence is 10% compared to 1%.

Lab Question 3: Interpretation

Q3A: How does the posterior probability change when prevalence increases?

Answer: It increases. With higher prevalence, a positive test is more likely to be a true positive instead of a false alarm.

Q3B: What does this mean in real-world screening for rare diseases?

Answer: When a disease is very rare, even good tests can produce mostly false positives. This is why confirmatory testing is important — otherwise, screening could cause unnecessary stress and follow-up procedures.

Summary Table Comparing Both Prevalences

```r
summary_df <- tibble(
  prevalence = c(prev, prev2),
  empirical_posterior = c(posterior_empirical, posterior_empirical2),
  theoretical_posterior = c(posterior_theoretical, posterior_theoretical2)
)
summary_df %>% knitr::kable(digits = 4)
```

prevalence	empirical_posterior	theoretical_posterior
0.01	0.1565	0.1667
0.10	0.6818	0.6875

```
## R version 4.5.1 (2025-06-13 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
```

```
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.4 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.1.0     readr_2.1.5     tidyr_1.3.1     tibble_3.3.0
##  [9] ggplot2_3.5.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.6       compiler_4.5.1     tidyselect_1.2.1   scales_1.4.0
##  [5] yaml_2.3.10        fastmap_1.2.0      R6_2.6.1           generics_0.1.4
##  [9] knitr_1.50         pillar_1.11.0      RColorBrewer_1.1-3 tzdb_0.5.0
## [13] rlang_1.1.6        stringi_1.8.7      xfun_0.52          timechange_0.3.0
## [17] cli_3.6.5          withr_3.0.2        magrittr_2.0.3     digest_0.6.37
## [21] grid_4.5.1         rstudioapi_0.17.1  hms_1.1.3          lifecycle_1.0.4
## [25] vctrs_0.6.5        evaluate_1.0.4     glue_1.8.0         farver_2.1.2
## [29] rmarkdown_2.29     tools_4.5.1        pkgconfig_2.0.3    htmltools_0.5.8.1
```