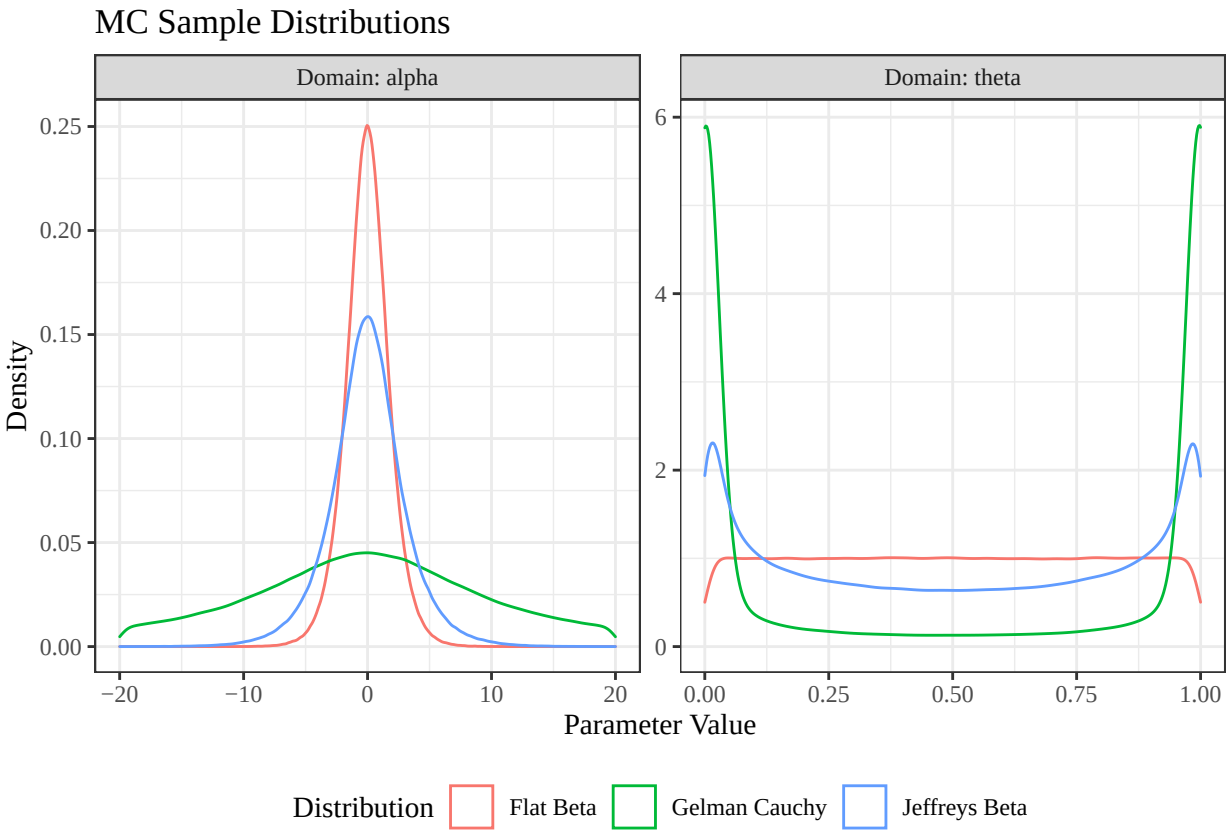
Problem Set #1

Spencer Pease

April 29th, 2021

Problem 1

Question 1.1



Question 1.2

The **flat beta default prior** is best suited for "fair-coin" hypotheses, since it spreads prior belief evenly across the domain of θ, leaving no single peak across the domain. This allows new data to have a stronger influence in establishing a mode, which is useful when trying to determine the "fairness" of an event.

The **Gelman default Cauchy prior**, conversely, is better suited for "extremely rare/common event" hypotheses. This prior concentrates belief around the two extremes of the θ domain, which ensures that

prior knowledge of an event being extremely rare/common manifests in the posterior, even if the observed data does not strongly indicate so in the sample.

A symmetric prior centered around $\theta = 0.5$ ($\alpha = 0$) effectively tells the posterior that we do not believe our estimates are more likely to be on one side of the parameter space than the other. For this case, asymmetry in the posterior is reflective of skew in the likelihood. Therefore, an asymmetrical prior is preferable in situations where we do believe there is some true skew in our parameter of interest, and it would be negligent to ignore it (for example, estimating deaths due to a disease that is known to be more fatal to older people would imply a larger tail on the older side of the distribution). This essentially "primes" the posterior to follow the specified asymmetry, and if the likelihood pulls the posterior away from that skew, there is reason to believe the underlying assumptions of the model are incorrect.

Question 1.3

Table 1: Summary of distribution quantiles on the Theta domain

Distribution	97.5-2.5% Span	IQR
Flat Beta	0.950	0.501
Gelman Cauchy	1.000	1.000
Jeffreys Beta	0.997	0.706

Examining both the interquartile range and span of the 2.5% to 97.5% quantiles over the domain of Θ, we see that the **Gelman default Cauchy prior** is the most diffuse, and the **flat beta prior** is the most concentrated.

Problem 2

Question 2.1

Fitting the question of medical test results to Bayes' Rule, we get:

- the epidemiological prevalence, $P(C)$, representing the **prior**
- the PPV and NPV, $P(C \mid T)$, representing the **posterior**
- the test sensitivity and specificity, $P(T \mid C)$, representing the **likelihood**

Question 2.2

The PPV, $P(C = 1 \mid T = 1)$, can be represented using known quantities and Bayes' Rule:

$$P(C = 1 \mid T = 1) = \frac{P(T = 1 \mid C = 1)\, P(C = 1)}{P(T = 1)}$$

where we know

- $P(T = 1 \mid C = 1) = 0.99$
- $P(T = 0 \mid C = 0) = 0.99$
- $P(C = 1) = 0.0001$

and $P(T = 1)$ can be calculated by marginalizing T over the domain of $C = \{0, 1\}$

$$\begin{aligned} P(T = 1) &= P(T = 1 \mid C = 1)\ P(C = 1) + P(T = 1 \mid C = 0)\ P(C = 0) \\ &= P(T = 1 \mid C = 1)\ P(C = 1) + (1 - P(T = 0 \mid C = 0))\ (1 - P(C = 1)) \\ &= (0.99)(0.0001) + (1 - 0.99)(1 - 0.0001) \\ &\approx 0.0101 \end{aligned}$$

With these values, we can then solve

$$\begin{aligned} P(C = 1 \mid T = 1) &= \frac{(0.99)(0.0001)}{0.0101} \\ &\approx 0.0098 \end{aligned}$$

to get a calculated PPV of 0.98%.

Question 2.3

Compared to the sensitivity of the test, the PPV shows the patient that they are much less likely to have the condition of interest. This **is not** an example of an "updated belief" however, since the test is equivalent to the likelihood and does not represent the question the patient is trying to answer.

Question 2.4

Compared to the prevalence of the condition, the PPV shows the patient that they are relatively much more likely to have the condition of interest. This **is** and example of an "updated belief", since prior to the test the patient would assume they are as likely to have the condition as the general population, but incorporating new data from the test changes what they know about their chances are of having the condition, answering their question of interest.

Question 2.5

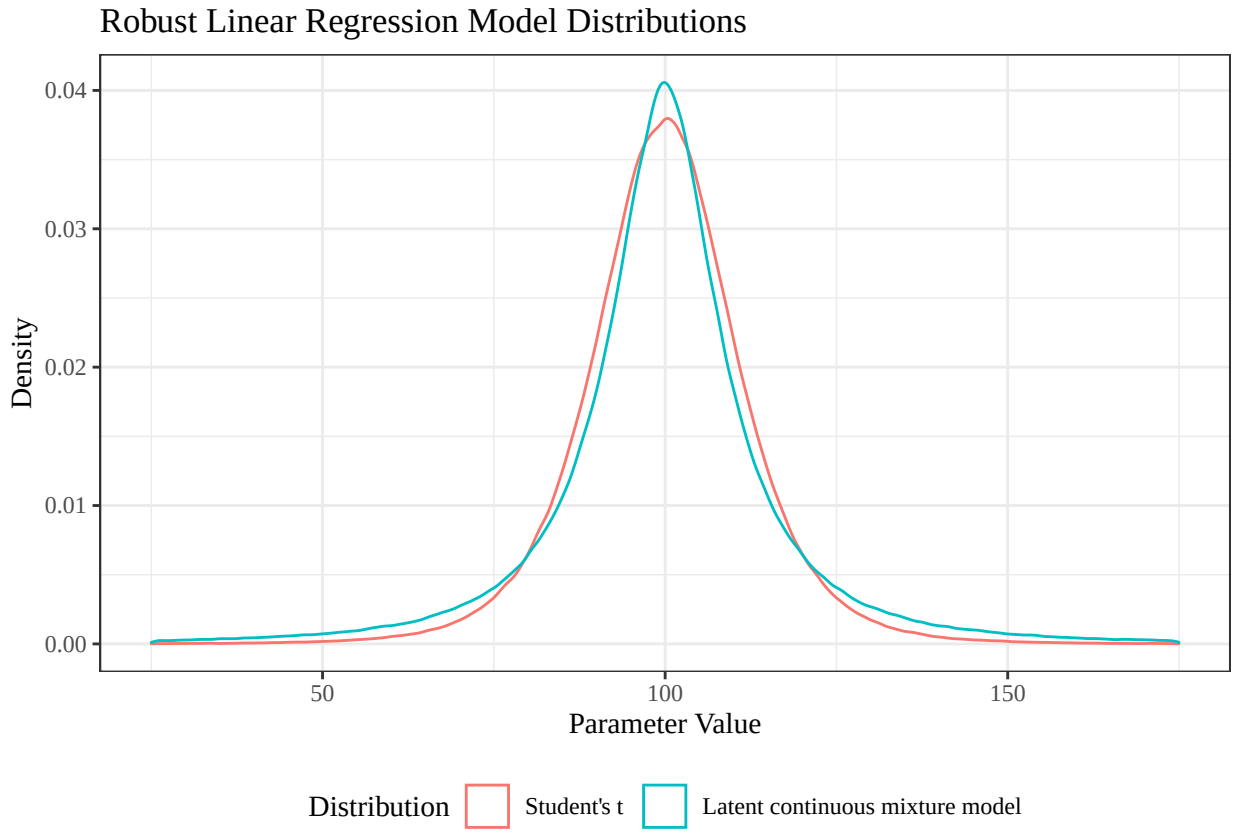
(Note, this question tries to find the probability of the patient not having the condition, given two positive tests.)

Assuming the tests are independent, if a patient receives a second positive test, they can use their updated belief about having the condition from the first test as prior information for determining the probability they have the condition after two positive tests. In other words, the prior for the calculation is now the predicted PPV after the first test.

Following a similar set of steps to calculate $P(C = 1 \mid T = 1)$, substituting the PPV of the first test in for the population prevalence, we get a probability of not having the condition of $1 - PPV = 50.50\%$.

Problem 3

Question 3.1



Comparing the simulated Student's t distribution to the latent continuous mixture model distribution, we see that while they are not an exact match, the differences are small, suggesting that Student's t distribution is interpretable as a latent continuous mixture model.

Problem 4

Question 4.1

Table 2: Entropy for the negative exponential distribution

Cambridge	Libby
10.02	9.9913

Question 4.2

Table 3: Average surprisal for the simulated sample

Cambridge	Libby
10.0345	10.0354

From the calculated average surprisals, we see that the *Libby* rate makes the sample most surprising.

Question 4.3

Table 4: Difference between average surprisal and entropy

Cambridge	Libby
0.0146	0.0441

For each rate, the average sample surprisal is greater than the calculated entropy, but the difference of the *Cambridge* rate is less extreme.

Question 4.4

Table 5: Negative exponential distribution MLE estimates

Rate	Entropy	Avg. Surprisal
1.19e-04	10.0344	10.0344

The sample is slightly less surprising under the maximum likelihood estimate of the model rate than either the *Cambridge* or *Libby* rate, though it is very close to the *Cambridge* rate.

The average surprisal implied by the MLE estimate exactly matches the calculated entropy, coming closer than either of the other rates. This is expected however, because the MLE rate is calculated from the sample, and the entropy is simply the expectation of surprisal, so the analytical calculation of entropy should be equivalent.

Appendix

Analysis

```r
# Prep work ---------------------------------------------------------------

library(dplyr)
library(LaplacesDemon)
library(stringr)
library(ggplot2)

# From: https://ro-che.info/articles/2018-08-11-logit-logistic-r
logit <- qlogis
logistic <- plogis


# Question 1.1 ------------------------------------------------------------

set.seed(587)

n_guesses <- 1e6

tbl_priors <-
  tibble(
    theta_flat_beta = rbeta(n_guesses, 1, 1),
    theta_jeffreys_beta = rbeta(n_guesses, 0.5, 0.5),
    alpha_gelman_cauchy = rcauchy(n_guesses, location = 0, scale = 10)
  ) %>%
  mutate(
    alpha_flat_beta = logit(theta_flat_beta),
    alpha_jeffreys_beta = logit(theta_jeffreys_beta),
    theta_gelman_cauchy = logistic(alpha_gelman_cauchy)
  ) %>%
  tidyr::pivot_longer(
    everything(),
    names_to = c("Domain", "Distribution"),
    names_pattern = "^(theta|alpha)_(.*)$"
  ) %>%
  mutate(Distribution = str_to_title(str_replace_all(Distribution, "_", " ")))

plot_priors <- tbl_priors %>%
  filter(between(value, -20, 20)) %>%
  ggplot(aes(x = value, color = Distribution)) +
  geom_density() +
  facet_wrap(vars(Domain), scales = "free", labeller = "label_both") +
  labs(
    title = "MC Sample Distributions",
    x = "Parameter Value",
    y = "Density"
  ) +
  theme_bw(base_family = "serif") +
  theme(legend.position = "bottom")
```

```r
# Question 1.3 ------------------------------------------------------------

tbl_theta_summary <- tbl_priors %>%
  filter(Domain == "theta") %>%
  group_by(Distribution) %>%
  summarise(
     span95 = quantile(value, .975) - quantile(value, .025),
     IQR = IQR(value)
  )


# Question 2.2 ------------------------------------------------------------

calc_ppv_npv <- function(sens, spec, prev) {

  ppv_likelihood <- sens
  ppv_prior <- prev
  ppv_data <- (sens * prev) + ((1 - spec) * (1 - prev))

  npv_likelihood <- spec
  npv_prior <- 1 - prev
  npv_data <- (spec * (1 - prev)) + ((1 - sens) * prev)

  list(
    ppv = (ppv_likelihood * ppv_prior) / ppv_data,
    npv = (npv_likelihood * npv_prior) / npv_data
  )

}

med_test <- calc_ppv_npv(sens = 0.99, spec = 0.99, prev = 1e-4)


# Question 2.5 ------------------------------------------------------------

med_test2 <- calc_ppv_npv(sens = 0.99, spec = 0.99, prev = med_test$ppv)


# Question 3.1 ------------------------------------------------------------

n_samples <- 1e6

tbl_lrm_mc <-
  tibble(
    st = LaplacesDemon::rst(n = n_samples, nu = 5, mu = 100, sigma = 10),
    lcmm = rnorm(
      n = n_samples,
      mean = 100,
      sd = LaplacesDemon::rinvchisq(n = n_samples, df = 5, scale = 10)
    )
  ) %>%
  tidyr::pivot_longer(everything(), names_to = "Distribution") %>%
  mutate(
```

```
    Distribution = factor(
      Distribution,
      levels = c("st", "lcmm"),
      labels = c("Student's t", "Latent continuous mixture model")
    )
  )

plot_rlrm_dist <-
  ggplot(tbl_lrm_mc, aes(x = value, color = Distribution)) +
  geom_density() +
  xlim(25, 175) +
  labs(
    title = "Robust Linear Regression Model Distributions",
    x = "Parameter Value",
    y = "Density"
  ) +
  theme_bw(base_family = "serif") +
  theme(legend.position = "bottom")


# Question 4.1 ------------------------------------------------------------

calc_exp_rate <- function(half_life) -log(0.5) / half_life
calc_exp_entropy <- function(rate) 1 - log(rate)

half_life <- list(Cambridge = 5730, Libby = 5568)
exp_rate <- lapply(half_life, calc_exp_rate)
exp_entropy <- lapply(exp_rate, calc_exp_entropy)


# Question 4.2 ------------------------------------------------------------

set.seed(1949)
neg_exp_MC <- rexp(n = 1000, rate = exp_rate$Cambridge)

calc_avg_surprisal <- function(rate) mean(-dexp(neg_exp_MC, rate = rate, log = TRUE))
exp_surprisal <- lapply(exp_rate, calc_avg_surprisal)


# Question 4.3 ------------------------------------------------------------

exp_diff <- mapply(`-`, exp_surprisal, exp_entropy)


# Question 4.4 ------------------------------------------------------------

exp_rate_mle <- 1 / mean(neg_exp_MC)
exp_entropy_mle <- calc_exp_entropy(exp_rate_mle)
exp_surprisal_mle <- calc_avg_surprisal(exp_rate_mle)
```