Individual HW04

Your Name Here

```
library(bayesrules) # R package for our textbook
library(tidyverse) # Collection of packages for tidying and plotting data
library(janitor) # Helper functions like tidy and tabyl
library(rstan) # for MCMC
library(bayesplot) # for plotting
```

1 Q1

Explain the difference between N_{eff} and the actual number of MCMC samples

2 Q2: `mcmc_rank_hist` and `mcmc_rank_overlay`

Trace plots are a natural way to view a chain, but they become hard to read when we have many samples and/or many chains. An alternative way to view the chains is called a **rank histogram** or **trace rank plot**. What this means is to (a) take all of the samples for a parameter and rank them, (b) draw a histogram of the ranks for each individual chain

The code chunk below loads two pre-fit stan models.

```
load(url("https://aluby.github.io/stat340-f25/data/hw04-q2.rda"))
```

(a) Create traceplots, rank histograms (`mcmc_rank_hist`), and trace rank plots (`mcmc_rank_overlay`) for both MCMC samples.
(b) Which sample is "healthy"? Which is "unhealthy"? Explain how you can tell from the new plots
(c) Which type of plot do **you** find more useful?

3 Q3: Revisiting BR Exercise 5.11

Prof. Abebe and Prof. Morales both recently finished their PhDs and are teaching their first statistics classes at Bayesian University. Their colleagues told them that the average final exam score across all students, μ, varies Normally from year to year with a mean of 80 points and a standard deviation of 4. Further, individual students' scores Y vary Normally around μ with an *unknown* standard deviation σ.

(a) Suggest three possible prior distributions for σ (at least two should be different named distributions). Include density plots for each prior. (You can find available probability distributions in the Stan documentation: https://mc-stan.org/docs/functions-reference/positive_continuous_distributions.html)

(b) Below is starter code for running a Stan model in this scenario. Delete the `#| eval: false` line and fill in the blanks

```
fn_model2 <- "
  data {
    int<lower=0> N;
    vector[N] y;
  }

  parameters {
    real mu;
    real<lower=0> sigma;
  }

  model {
    y ~ normal(_____, ____);
    mu ~ normal(_____, _____);
    sigma ~ __________;
  }
"
```

(c) Run your model on the data below from Prof. Abebe and Prof. Morales' combined scores.

```
scores<- c(90.3, 81.3,  85.2, 85.6, 86.4, 83.8, 86.7, 90.1, 82.3, 88.4, 89.2, 90.6,81.2, 85.9, 8
```

(d) Run thorough diagnostics on your MCMC sample. This should include: traceplots of all relevant parameters, density plots for each chain for all relevant parameters, R-hat values, and N-eff. What do you conclude about the validity of your sample?

(e) What do you conclude about the posterior mean $\mu|\vec{y}$? How does your answer differ from HW3's conjugate analysis?

(f) Rewrite the stan model code to take a flexible normal prior for μ. That is, we should be able to specify the prior mean θ and prior sd τ (or prior variance τ^2, whichever you prefer), instead of hard-coding them as 80 and 4.

4 BR 7.4: Tuning the Metropolis-Hastings

In this exercise you will consider how to tune a Uniform proposal model with half-width w for a Metropolis-Hastings algorithm.

(a) Draw a trace plot for a tour where the Uniform proposal model uses a very small w.
(b) Why is it problematic if w is too small, and hence defines the neighborhood around the current chain value too narrowly?
(c) Draw a trace plot for a tour where the Uniform proposal model uses a very large w
(d) Why is it problematic if w is too large, and hence defines the neighborhood too widely?
(e) Draw a trace plot for a tour where the Uniform proposal model uses a w that is neither too small or too large.
(f) Describe how you would go about finding an appropriate half-width w for a Uniform proposal model.

5 BR 7.5: One iteration with uniform proposal model

The function `one_mh_iteration()` from the text utilizes a Uniform proposal model, $\mu'|\mu \sim \text{Unif}(\mu - w, \mu + w)$, with half-width $w = 1$. Starting from a current value of $\mu = 3$ and using `set.seed(1)`, run the code below and comment on the returned `proposal`, `alpha`, and `next_stop` values.

> **Note**
>
> You can use `one_mh_iteration` directly from the text

(a) `one_mh_iteration(w = 0.01, current = 3)`
(b) `one_mh_iteration(w = 0.5, current = 3)`
(c) `one_mh_iteration(w = 1, current = 3)`
(d) `one_mh_iteration(w = 3, current = 3)`

6 BR 7.11: Changing the proposal model

For this exercise, modify `one_mh_iteration()` to create a new function, `one_mh_iteration_normal()`, which utilizes a symmetric Normal proposal model, centered at the current chain value μ with standard deviation s:

$$\mu' | \mu \sim N(\mu, s^2)$$

Subsequently, starting from a current value of $\mu = 3$ and `set.seed(1)`, run this function under each setting below. Comment on the returned `proposal`, `alpha`, and `next_stop` values.

(a) `one_mh_iteration_normal(s = 0.01, current = 3)`
(b) `one_mh_iteration_normal(s = 0.5, current = 3)`
(c) `one_mh_iteration_normal(s = 1, current = 3)`
(d) `one_mh_iteration_normal(s = 3, current = 3)`

7 BR 7.12 (Metropolis-Hastings tour with normal proposals)

Upon completing the previous exercise, modify `mh_tour()` to create a new function, `mh_tour_normal()`, which constructs a chain of μ values using a Normal proposal model with standard deviation s. Subsequently, using `set.seed(84735)`, run this function under each setting below and construct a trace plot of the chain.

(a) 20 iterations, s=0.01
(b) 20 iterations, s=10
(c) 1000 iterations, s=0.01
(d) 1000 iterations, s=10
(e) Contrast the trace plots in a and b. Explain in simple terms why changing the standard deviation of the Normal proposal model causes these differences.
(f) Reflecting on the above results, tune your Metropolis-Hastings algorithm. That is, identify a reasonable value for standard deviation s and provide a trace plot as proof.