Week 6 Exercises

Exercise 1 Bayes' Readings

Readings

- Read Western and Jackman (1994). I recommend Jackman (2004) as well.
- Read ch. 2 of Gill (2014) on the Bayesian engine, especially section 2.3. I recommend ch. 1 for helpful context.
- Read pp. 97-128 of Gill (2014) on prior distributions. Gill provides some helpful history on thinking about the prior distribution, including conjugate priors, uniform priors, invariant priors, improper priors, and elicited priors.[1]

Question

In this course, we have now studied two general engines: maximum likelihood and Bayesian inference. How should we think about the relationship between the two? Are they best understood as competing, incompatible alternatives? Or simply as interchangeable tools? If they compete, which approach is right, and why? If they are interchangeable, when might one be more useful than the other?

Exercise 2 Bernoulli

A pollster conducts a sample survey using a simple random sample of 500 adults. Of the 500 respondents, 220 say they approve of the job Donald Trump is doing as president.

1. Using a Bernoulli model with a beta prior, compute the posterior mean and 90% credible interval for the percent of the population that approve. Choose reasonable values for the parameters of the prior distribution. You may use a flat, "weakly informative," or informative prior. Compute the posterior mean and a 90% credible interval.

[1] Most modern priors are "weakly informative," meaning that they rule out only *absurd* values of the parameters, like variances *very* close to zero or logistic regression probabilities *very* close to zero or one. However,For modern thinking on prior distributions, see the Stan wiki.

2. Using the posterior from above, what's the chance that the population percent is larger than 50%? Larger than 45%?
3. Using a Bernoulli model, compute the ML estimate of the population percent. Use the parametric bootstrap to construct a 90% confidence interval.
4. Compare the ML and Bayesian point estimates and interval estimates. Are they similar? How does the *interpretation* differ?
5. From part 1, how absurd can you make the beta prior without meaningful changes in the 90% credible interval?

Part (1)

```
# bayesian credible interval
# -------------------------

# data
N <- 500
k <- 220

# prior; uniform
alpha_star <- 1
beta_star <- 1

# posterior
alpha_prime <- alpha_star + k
beta_prime <- beta_star + (N - k)

# find posterior mean
print(alpha_prime/(alpha_prime + beta_prime), digits = 2)
```

```
[1] 0.44
```

```
# 90% percentile credible interval
print(qbeta(c(0.05, 0.95), alpha_prime, beta_prime), digits = 3)
```

```
[1] 0.404 0.477
```

Part (2)

```
# posterior probabilities
# -----------------------
```

```r
# larger than 50%
1 - pbeta(0.5, alpha_prime, beta_prime)
```

```
[1] 0.00364417
```

```r
# larger than 45%
1 - pbeta(0.45, alpha_prime, beta_prime)
```

```
[1] 0.328787
```

Part (3)

```r
# ml estimates w/ parametric bs
# -----------------------------

# ml est
pi_hat <- k/N
print(pi_hat, digits = 2)
```

```
[1] 0.44
```

```r
# 95% ci
n_bs <- 2000
bs_est <- numeric(n_bs)  # a container for the estimates
for (i in 1:n_bs) {
  bs_y <- rbinom(N, size = 1, prob = pi_hat)
  bs_est[i] <- mean(bs_y)
}
print(quantile(bs_est, probs = c(0.05, 0.95)), digits = 3)  # 95% ci
```

```
   5%   95%
0.404 0.476
```

Part (4)

The Bayesian credible intervals and the parametric bootstrap confidence interval are the same to about the 1/1000-th decimal place. There's certainly no substantively meaningful difference between the two.

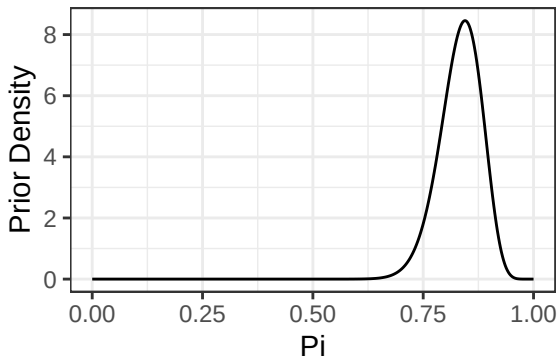
We interpret frequentist 90% CI by claiming that the parameter falls within the interval (realizing that we'll be wrong at most 10% of the time in the long run). We interpret the Bayesian 90% CI by saying that there's a 90% chance that the parameter falls within the interval.

Part (5)

```
# prior robustness checks
# -----------------------

# prior; uniform
alpha_star <- 50
beta_star <- 10

# posterior
alpha_prime <- alpha_star + k
beta_prime <- beta_star + (N - k)

# 90% percentile credible interval
print(qbeta(c(0.05, 0.95), alpha_prime, beta_prime), digits = 2)
```

```
[1] 0.45 0.52
```

```
# plot "absurd" prior that *barely* changes the ci
ggplot() +
  xlim(0, 1) +
  stat_function(fun = dbeta, n = 1001,
                args = list(shape1 = alpha_star, shape2 = beta_star)) +
  labs(x = "Pi",
       y = "Prior Density") +
  theme_bw()
```



At least to me, it's really surprising how little impact this extremely informative prior has on the inferences. We get basically the same confidence interval, but using an absurdly strong and implausible prior.

Exercise 3 Poisson

Suppose you model a data set $y = \{y_1, y_2, ..., y_N\}$ as iid draws from a Poisson distribution with parameter (mean) λ, so that $f(y_i; \lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$. The Gamma distribution is the conjugate prior, so that $f(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$.

1. Find a sensible prior for this problem.
2. Use this Poisson model to make inferences about the λ the expected number of operations in Santiago for Holland's (2015) data.
3. Compare the posterior mean and 90% credible interval to the ML estimate and 90% confidence interval.
4. Use simulations to find the posterior mean and SD of the quantity of interest $\tau = \text{SD}(\lambda) = \sqrt{\lambda}$.

Part (1)

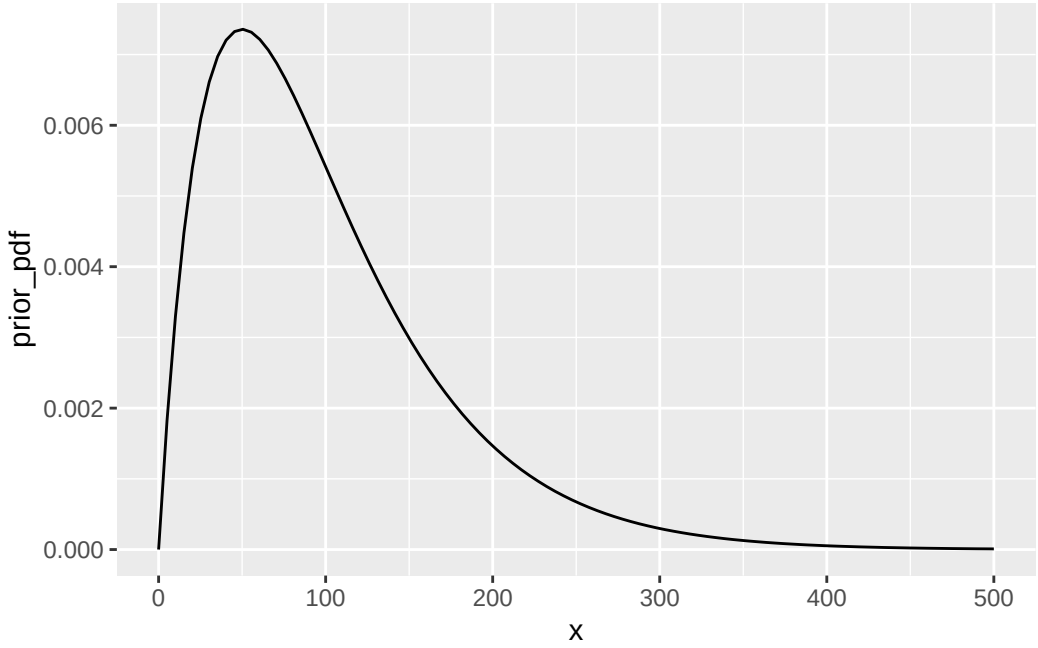
The posterior distribution is Gamma$(\alpha^* + \sum y, \beta^* + n)$, where α^* and β^* are the prior parameters for the gamma distribution. **Be careful, these are *opposite* the gamma posterior for the exponential model below.**

Part (2)

The mean of the gamma distibution is $\frac{\alpha}{\beta}$. I don't have a lot of information about what the average number of casualties per month would be, but maybe something like 100, with a large dispersion around that.

I settled on $\alpha^* = 2$ and $\beta^* = 0.02$. The important point is that this prior reflect your prior beliefs about the **average number of civilian casualties per month**.

```
alpha_star <- 2 # shape
beta_star  <- .02  # rate

# plot prior
x <- seq(0, 500, length.out = 100)
prior_pdf <- dgamma(x, shape = alpha_star, rate = beta_star)
gg_data <- tibble(x, prior_pdf)
ggplot(gg_data, aes(x = x, y = prior_pdf)) +
  geom_line()
```



Exercise 4 Exponential

Suppose you model a data set $y = \{y_1, y_2, ..., y_n\}$ as iid draws from an exponential distribution with parameter (rate) λ. The gamma distribution is the conjugate prior, so that $f(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$. In the steps below, use this Bayesian model to make inferences about the expected survival `time` in the `survival::cancer` data set.

1. Find the posterior distribution. Hint: Because the gamma distribution is the conjugate prior, the posterior will also be gamma. This will closely follow the Poisson example from the notes.
2. Find a sensible gamma prior for λ, where the exponential distribution is used to model survival time. Recall that λ is a rate, and it might be easier to think in terms of the mean. You can simulate rates from your gamma prior, convert each simulated rate to mean, and evaluate the sensibility of the distribution of *means* instead.
3. Use the exponential model to model the `time` variable in the `cancer` data set.
4. Find posterior mean of λ; use the identity $E(X) = \frac{\alpha}{\beta}$ for $X \sim \text{gamma}(\alpha, \beta)$. Find the 90% credible interval. Use `qgamma()`.
5. Compare the posterior mean above to the ML estimate and 90% confidence interval.

Exercise 5 Rejection

For the Bernoulli model, the beta distribution is the conjugate prior. And the beta distribution is especially nice because it's flexible. The flexibility allows it to represent a wide range of prior beliefs. However, the beta distribution can be awkward to work with. We are most familiar with a normal distribution. This exercise highlights how posterior simulation can greatly simplify Bayesian inference.

For example, let's use the toothpaste cap problem and data. Before collecting data, we might think that the chance of a top is about 15%, give or take 10% or so. This suggests a normal distribution with $\mu = 0.15$ and $\sigma = 0.1$. However, the normal distribution has support outside the [0, 1] support of π. But this can still work! We can simply set the normal pdf to zero outside the [0, 1] interval and renormalize the distribution. This gives us the *truncated normal* distribution.

Let's use the usual Bernoulli likelihood and the truncated normal prior, find the unnormalized posterior distribution, and use the rejection algorithm to obtain a sample from the posterior distribution.

Bernoulli likelihood

To begin, recall that the probability mass function (pmf) of a single Bernoulli trial with success probability π is

$$f(y \mid \pi) = \pi^y (1-\pi)^{1-y}, \quad y \in \{0, 1\}.$$

Because the trials are independent, the joint likelihood for n observations $y_1, \ldots, y_n$ is the product

$$f(y \mid \pi) = \prod_{i=1}^{n} \pi^{y_i} (1-\pi)^{1-y_i}.$$

Collecting terms, let $k = \sum_{i=1}^{n} y_i$ be the total number of successes. Then the likelihood simplifies to

$$f(y \mid \pi) = \pi^k (1-\pi)^{n-k}.$$

Notice that this expression is the likelihood function $L(\pi) = f(y \mid \pi)$. We just denote them slightly differently to emphasize different pieces depending on whether we are using maximum likelihood or finding the posterior.

Truncated normal prior

Suppose we want to model prior information about π using a truncated normal distribution. Begin with the usual normal distribution, then restrict the support to the interval $[0, 1]$ since

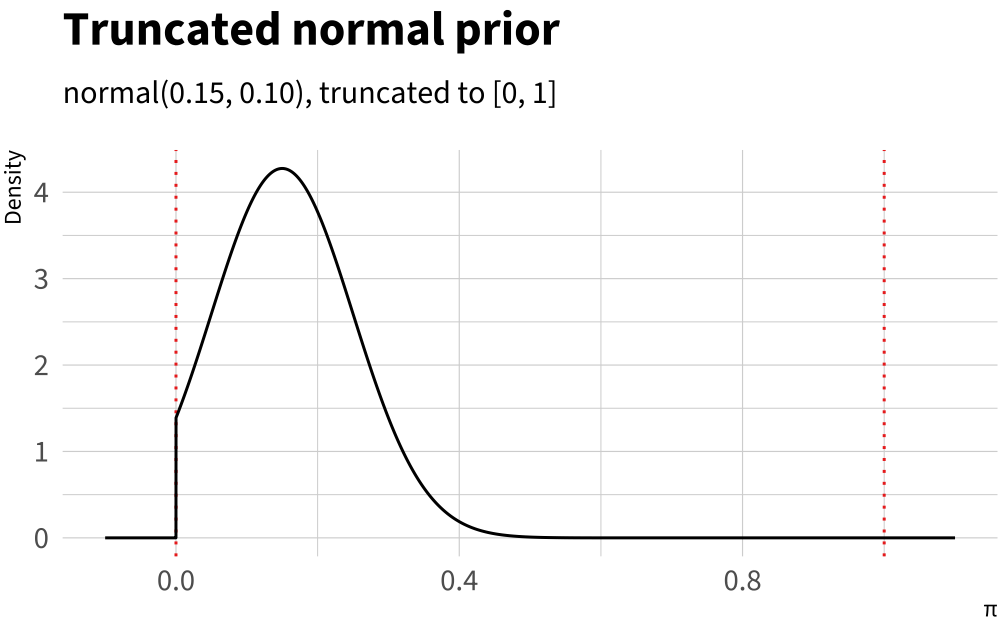
π must be a valid probability. Formally, if $Z \sim \mathcal{N}(\mu, \sigma^2)$, then the truncated normal prior is defined as

$$f(\pi) = \frac{\phi\left(\frac{\pi-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\frac{1-\mu}{\sigma}\right) - \Phi\left(\frac{0-\mu}{\sigma}\right)\right]}, \quad 0 \le \pi \le 1,$$

where $\phi(\cdot)$ is the standard normal density and $\Phi(\cdot)$ is the standard normal distribution function. The denominator is important–$\sigma\left[\Phi\left(\frac{1-\mu}{\sigma}\right) - \Phi\left(\frac{0-\mu}{\sigma}\right)\right]$ serves as a normalizing constant so that the density integrates to one over the admissible interval.

This prior can be especially useful when we have an approximate guess for π and a give-or-take (i.e., an SD) around that guess. It's quite intuitive to model this guess using a normal distribution.

For example, in the toothpaste cap problem, if we believe the probability is about 0.15 give-or-take 0.10, then the truncated normal prior places most of its mass near 0.15 but avoids assigning probability outside the valid range of $[0, 1]$.



Unnormalized Posterior

To find the posterior distribution, we multiply the likelihood times the prior. Bayes' rule tells us

$$f(\pi \mid y) \propto f(y \mid \pi)\, f(\pi),$$

where $f(y \mid \pi)$ is the likelihood and $f(\pi)$ is the prior.[2] Substituting in the expressions we derived above, we obtain

$$f(\pi \mid y) \propto \underbrace{[\pi^k(1-\pi)^{n-k}]}_{\text{Bernoulli likelihood}} \cdot \underbrace{\left[\frac{\phi\left(\frac{\pi-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\frac{1-\mu}{\sigma}\right) - \Phi\left(\frac{0-\mu}{\sigma}\right)\right]}\right]}_{\text{truncated normal prior}}, \quad 0 \leq \pi \leq 1.$$

This is the *unnormalized* posterior (i.e., it does NOT intergate to one, as required of pdfs). To make this a *proper* posterior, we would need to divide find normalizing constant for the right-hand side. However, the rejection algorithm does not require a proper posterior—the unnormalized posterior is sufficient.

But we can take further advantage of this. The denominator $\sigma\left[\Phi\left(\frac{1-\mu}{\sigma}\right) - \Phi\left(\frac{0-\mu}{\sigma}\right)\right]$ of the prior is *also a constant* with respect to π. For sampling purposes we can treat it as part of the proportionality constant. Thus, the unnormalized posterior density is

$$f(\pi \mid y) \propto \pi^k(1-\pi)^{n-k}\,\phi\left(\frac{\pi-\mu}{\sigma}\right), \quad 0 \leq \pi \leq 1.$$

Conveniently, the difficult truncation part of the prior just drops out, and we are left with a regular normal pdf.

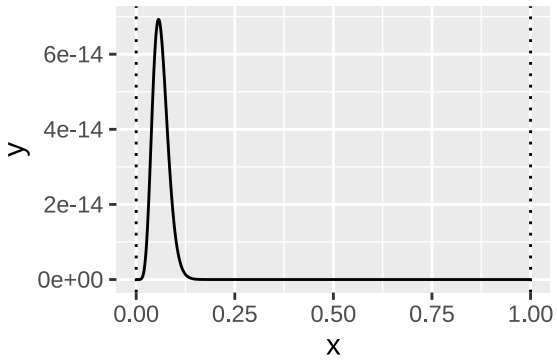
We can write this function easily in R and plot it.

```
# unnormalized posterior for our data and prior
f <- function(pi, k = 8, n = 150, mu = 0.15, sigma = 0.10) {
  ifelse(
    pi < 0 | pi > 1,  # check if outside [0, 1]
    0,                # return 0 if outside [0, 1]
    (pi^k) * ((1 - pi)^(n - k)) * dnorm(pi, mean = mu, sd = sigma)
  )
}
```

Notice that the posterior densities are *very* small. This will be numerically challenging. For this reason, we tend to perform the computation using the unnormalized log-posterior. However, the intuition for the algorthim isn't immediately obvious. Since our goal here is the learn the intuition, we're sticking with the unlogged posterior.

```
ggplot() +
  geom_vline(xintercept = c(0, 1), linetype = "dotted") +
  stat_function(fun = f, n = 1001)
```

[2] Notice that the meaning of f depends on the context; f represents the prior, likelihood, and posterior. This is a common sloppiness in probability theory because the context usually makes the meaning clear.



From the figure, we can see that an envelope constant $M = 8 \times 10^{-14}$ would work well. From there, we can let the algorithm do the work.

Task

For the toothpaste cap problem data ($k = 8$; $n = 150$) and the model and priors discussed above (Bernoulli likelihood; truncated normal prior with $\mu = 0.15$ and $\sigma = 0.10$), use the rejection algorithm to find the posterior mean and a 90% credible interval for the odds *of failure*. That is, generate posterior simulations of π, transform those simulations using odds of failure $= \frac{1-\pi}{\pi}$, and then summarize the simulations.

Solution

First, borrow the rejection algorithm from the notes.

```
rej <- function(f, S, M) {

  # record start time
  start_time <- Sys.time()

  # create containers and initialize counters
  samples <- numeric(S)  # container to store samples
  rejects <- NULL  # container to track rejected values; for teaching; slow!
  s <- 1 # currently trying to take sample 1
  n_prop <- 0  # count proposals (for an acceptance-rate message)

  # so long as the current sample s is less
  #   than the desired samples S.
  #   do the following:
  while (s <= S) {

    # A: propose z ~ uniform(0,1)
    z <- runif(1)
```

```r
    # B: draw u ~ uniform(0,1)
    u <- runif(1)

    # C: Accept or reject
    fz <- f(z) # compute once, for effeciency

      ## scenario 1: u <= f(z)/M  →  Accept
      if (u <= fz / M) {
        samples[s] <- z
        s <- s + 1
      }

      ## scenario 2: f(z) > M  →  shouldn't happen; error
      if (fz > M) stop("Stop: Envelope M is too small.")  # find appropriate M

      ## scenario 3: u > f(z)/M  →  Reject
      ##   tracking these values just for teaching and learning--not needed usually
      if (u > fz / M) {
         rejects <- c(rejects, z)
      }

    # track total proposals so far
    n_prop <- n_prop + 1
  }

# print a summary report
  message(
    paste0(
      "  Successfully generated ", scales::comma(S), " samples!  \n\n",
      "  Accepted samples: ", scales::comma(S), "\n",
      "  Rejected samples: ", scales::comma(length(rejects)), "\n",
      "  Acceptance rate: ", scales::percent(S / n_prop, accuracy = 1), "\n",
      "  Total time: ", prettyunits::pretty_dt(Sys.time() - start_time)
    )
  )

  # return
  list(
    n_prop = n_prop,
    acc_rate = S / n_prop,
    samples = samples,
    rejects = rejects
```

```
  )
}
```

Next, create the unnormalized log-posterior.

```
# unnormalized posterior for our data and prior
f <- function(pi, k = 8, n = 150, mu = 0.15, sigma = 0.10) {
  ifelse(
    pi < 0 | pi > 1,  # check if outside [0, 1]
    0,                # return 0 if outside [0, 1]
    (pi^k) * ((1 - pi)^(n - k)) * dnorm(pi, mean = mu, sd = sigma)
  )
}
```

Now run the rejection algorithm. $S = 1,000$ samples should be sufficient. The question identifies $M = 8 \times 10^{-14}$ as a good choice.

```
r <- rej(f, S = 1000, M = 8e-14)
```

```
  Successfully generated 1,000 samples!

  Accepted samples: 1,000
  Rejected samples: 22,896
  Acceptance rate: 4%
  Total time: 892ms
```

Finally, transform the posterior samples to obtain simulations of the quantity of interest and then summarize those simulations.

```
pi_tilde <- r$samples  # extract simulations of pi
oof_tilde <- (1 - pi_tilde)/pi_tilde  # transform to odds of failure
mean(oof_tilde)  # posterior mean
```

```
[1] 17.27257
```

```
quantile(oof_tilde, probs = c(0.05, 0.95))  # 90% equal-tailed ci
```

```
       5%       95%
 9.557012 27.908373
```

In this problem, we use a Bernoulli likelihood with $k = 8$ successes out of $n = 150$ trials. We model our prior beliefs with a truncated normal distribution with $\mu = 0.15$ and $\sigma = 0.10$, reflecting a belief that the chance of success is "about 15% give or take 10 percentage points or so" (but constrained to the $[0, 1]$ range). Combining these with Bayes' rule yields the posterior, which we sampled from using a rejection algorithm. The posterior mean of the odds of failure is about 16.7, with a 90% credible interval from 9.4 to 27.8. This means that, after updating our beliefs with the data, we conclude that failure is about 16–17 times more likely than success with 90% probability the odds of failure lie fall between about 9:1 and 28:1.

Exercise 6 King et al. (1990)

Reading. Read ch 1, 2, and 3 of Box-Steffensmeier and Jones (2004).

Data: Load the `coalcold.tab` dataset from this Dropbox folder.

```
# load data
coal <- read_tsv("data/coalcold.tab") |>
  # make sure we interpret the censoring variable correctly
  mutate(censoring = case_when(CIEPTW == 1 ~ "none",
                               CIEPTW == 0 ~ "right"))
```

Fit Models: Use the `survreg()` function in the {survial} package and/or the `flexsurvreg()` function in the {flexsurv} package to fit several[3] duration models with censoring to the `ZeligData::coalition` from King et al. (1990). See the `dist` argument in `?survival::survreg` and `?flexsurv::flexsurvreg` for the distributions supported by these functions.[4] The code below uses `survreg()` and `flexsurvreg()` to reproduce the Weibull model shown in Table 3.3 on p. 43 of Box-Steffensmeier and Jones.

```
# load packages
library(survival)
library(flexsurv)

# corresponds to table 3.3 on p. 43 of BSJ
f <- Surv(DURAT, CIEPTW) ~ INVEST + POLAR + NUMST2 + FORMAT + ELTIME2 + CARETK2

# fit w/ survreg()
fit_weib <- survreg(f, data = coal, dist = "weibull")
```

[3] You can choose which and how many.

[4] These two packages offer different distributions with different parameterizations, so pay careful attention to the documentation, especially if directly interpreting coefficients.

```r
# fit w/ flexsurvreg()
fit2_weib <- flexsurvreg(f, data = coal, dist = "weibull")
```

Evaluate Models: Use the BIC to find the best model. Explain and interpret.

Meaningful Quantities of Interest: For the best **two** models and worst model (three models total), use {marginaleffects} to compute one or more quantities of interest. Explain and interpret the estimates and justify your choice of quantities.

Exercise 7 Exam review

Prepare for the exam. There's not a devoted review week, so please plan ahead. The questions will follow directly from the lectures, notes, and review exercises. The exam will be a mixture very short objective questions (e.g., multiple choice, fill-in-the-blank, matching); short, open-ended questions requiring 1-3 sentences; and longer open-ended questions requiring several sentences and/or longer derivations. In developing the exam questions, I draw from examples in the slides, notes, and exercises.

I hesitate to give a "study guide" because it might lead you to *exclude* material that appears on the exam. Nonetheless, here are some things to definitely *include* in your preparation:

1. The mathematics emphasized during week 1, especially those tools that have come up repeatedly (e.g., logs, gradient, Hessian).
2. ML, Fisher information, invariance property, delta method; and how all these fit together.
3. Explain "consistency."
4. Define the predictive distribution. Explain how we can use it to evaluate a fitted model. Connect this to Poisson versus negative binomial models of counts.
5. Explain the `optim()` function, including its arguments.
6. What is the "sampling distribution"? What are the three features we care about? Connect these three features to other relevant concepts.
7. What is a parametric bootstrap and how can it be used to create a 90% CI?
8. Work out closed form SEs using Fisher information and delta method for simple examples.
9. Explain "coverage" and describe how we might use a Monte Carlo simulation to evaluate coverage.
10. Demonstrate the equivalence of scalar and matrix forms of $X_i\beta$.
11. Explain how to use R formulas to include interactions, polynomials, and qualitative variables in a design matrix. Explain what the design matrix looks like in each case (i.e., what columns does it have)?
12. Be familiar with the special formula operators, like `+`, `*`, `:`, and `^`. Understand when `I()` is needed and not.
13. Explain the logit model. How is a probit model different?

14. What is a first difference and expected value?
15. Explain how to use `glm()` to fit logit and probit models.
16. Explain how to use {marginaleffects}. What is the conceptual framework? What are the main functions? What are the main arguments to those functions? What is a good default?
17. Describe AIC/BIC and how to use them to choose among models.
18. Explain how to add zero-inflation to a likelihood.
19. Explain how to use R to fit our suite of count models (i.e., Poisson, NB, ZINB)
20. Explain the basic logic of Bayesian inference.
21. Explain "conjugate prior." Give an example.
22. Find the posterior for simple, conjugate problems like Bernoulli, Poisson, and exponential.
23. Explain how to obtain point estimates and interval estimates from posterior simulations.
24. Explain the right way to compute the posterior mean for a transformation.
25. Explain a simple rejection algorithm to sample from a distribution.
26. Explain how to add censoring to a likelihood.
27. Explain how to use R to fit our suite of duration models (i.e., `survival::survreg()` and `flexsurv::flexsurvreg()`; especially the `Surv()` part of the formula; the `dist` argument and its numerous choices).
28. Using the diverse normal linear model, zero-inflated negative binomial model, and censored Weibull model as examples, explain how the expected value and first difference unify our interpretations of these complicated models.

References

Box-Steffensmeier, Janet M., and Bradford S. Jones. 2004. *Event History Modeling: A Guide for Social Scientists*. Cambridge: Cambridge University Press.

Jackman, Simon. 2004. "Bayesian Analysis for Political Research." *Annual Review of Political Science* 7 (1): 483–505. https://doi.org/10.1146/annurev.polisci.7.012003.104706.

King, Gary, James E. Alt, Nancy Elizabeth Burns, and Michael Laver. 1990. "A Unified Model of Cabinet Dissolution in Parliamentary Democracies." *American Journal of Political Science* 34 (3): 846. https://doi.org/10.2307/2111401.

Western, Bruce, and Simon Jackman. 1994. "Bayesian Inference for Comparative Research." *The American Political Science Review* 88 (2): 412–23. https://doi.org/10.2307/2944713.