GR5065 HW3 Answer Key

```
vetoes <- readr::read_csv("vetoes.csv", show_col_types = FALSE)
```

These data were assembled at

https://www.presidency.ucsb.edu/statistics/data/presidential-vetoes

which also has a brief explanation of the process and the variables.

1 Frequentism

One way to avoid being confused by the Frequentist interpretation of probability is to say "the proportion of possible datasets that could have been sampled from a population where ...", instead of "the probability that ...". Those are equivalent statements, but Fisher insisted on using the latter because he denied that there could be any other interpretation of probability.

So, we can say that a p-value is (an estimate of) the proportion of possible datasets that could have been sampled from a population where the estimate is more extreme than the estimate obtained in the actual dataset, given that the null hypothesis is true. Or a 90% confidence interval estimator is an estimator that will contain the true parameter with 90% of the possible datasets that could be sampled from this population.

When phrased that way, it also should be clear that Frequentist interpretation of probability is not applicable when you have the only feasible dataset. Fisher's applied research was mostly in agricultural experiments, so it would be absurd to think that you could ever have a dataset with all possible orange trees, for example. However, for social science data where the unit of analysis is some institution, the data are never a random sample and often comprise all the historical data, or at least, all the historical data since the time someone started keeping track of it. Also, websites collect whatever information they collect for every visiting to that website, rather than a random sample of visitors.

However, even if we have the only feasible dataset, we are still uncertain about the values of the parameters in the model, which model is best, etc. But we would also continue to have that uncertainty even if the data were a random sample from a population that we could apply Frequentist statistics to. Thus, Frequentist statistics does not yield everything a scientist might want to know.

Social scientists apply Frequentist methods even when they are inapplicable. In many cases, those social scientists do not know how to do anything else and are only dimly aware that there is anything else that could be done. Ironically, a big part of the problem is that social scientists will accept Frequentist estimates — which were created by Fisher to prevent Bayesian approaches from taking hold in science — as if they were Bayesian, at which point it seems like unnecessary effort to obtain genuine Bayesian estimates.

2 Generative Model

If the President's party controls either or both of the House of Representatives or the Senate, then a bill that the President opposes is unlikely to even come to a vote and if it does, then it usually will not pass. Thus, the operative consideration is which party is in control of the House and Senate, and almost all of the regular vetoes will occur in situations in which the President's party controls neither the House nor the Senate. So, we can manipulate the data accordingly and include an interaction term in the model.

```
library(dplyr)
vetoes <- mutate(vetoes,
                 House_minority = House < 50,
                 Senate_minority = Senate < 50,
                 House = House - 50,
                 Senate = Senate - 50) %>%
  filter(!is.na(House)) # excludes Andrew Johnson's terms
```

If a bill is vetoed, at least half the House and Senate must have supported it originally, and I think an override vote is mandatory. Thus, when considering whether a veto will be overridden, the number of seats held by members of the President's party is more important because it takes two-thirds in favor to override.

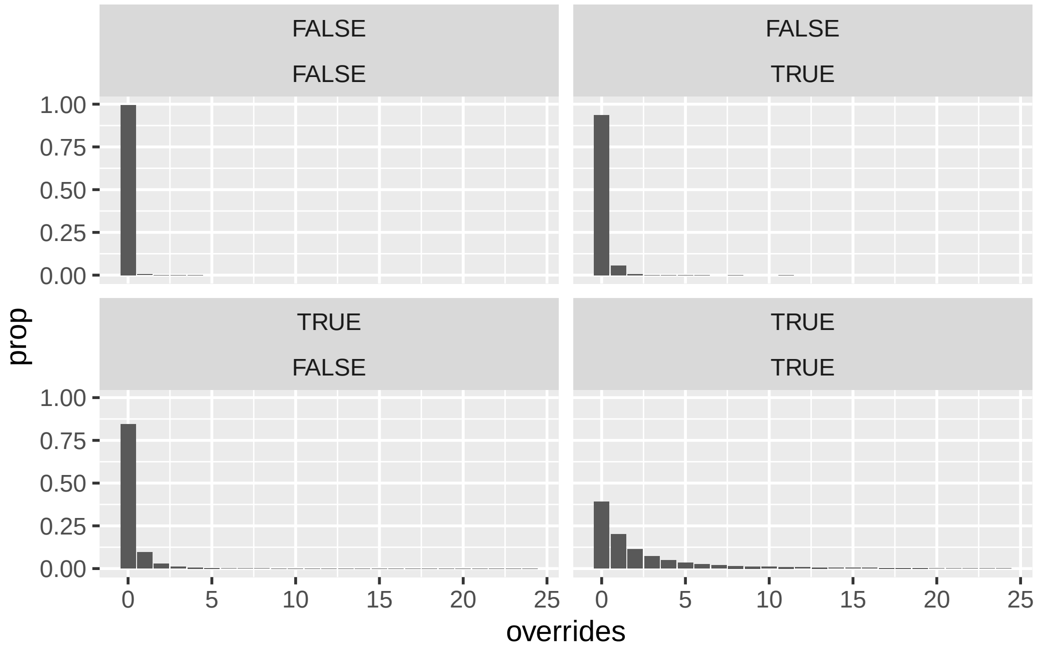
In a count model, it is difficult and sometimes impossible to keep the maximum realization in a reasonable range, particularly when your priors are independent. Shifting the intercept left can help. In linear models, it was usually acceptable to think of the intercept — relative to centered predictors — as the expected outcome irrespective of the predictors. In count models with log link functions, that is not quite right. We need to think of the intercept as the log expected count given that all the predictors are average. When the House and Senate are "average", there are not going to be many vetoes and almost none of those are going to get overriden. Almost all of the vetoes stem from situations where the House and Senate are far from average.

```
R <- 10000
avg_House_minority <- mean(vetoes$House_minority)
avg_Senate_minority <- mean(vetoes$Senate_minority)
avg_both_minority <- mean(vetoes$House_minority * vetoes$Senate_minority)
avg_House = mean(vetoes$House)
avg_Senate = mean(vetoes$Senate)

draws <- # start with parameters for the regular vetoes (gamma -- phi)
  tibble(gamma = rnorm(R, mean = log(1), sd = 0.75),
         beta_House_minority = rnorm(R, mean = 0.25, sd = 0.1),
         beta_Senate_minority = rnorm(R, mean = 0.25, sd = 0.1),
         beta_both_minority = rnorm(R, mean = 1, sd = 0.5),
         beta_House = rnorm(R, mean = 0, sd = 0.05),
         beta_Senate = rnorm(R, mean = 0, sd = 0.05),
         alpha = gamma - beta_House_minority * avg_House_minority -
           beta_Senate_minority * avg_Senate_minority -
           beta_both_minority * avg_both_minority -
           beta_House * avg_House - beta_Senate * avg_Senate,
         phi = rexp(R, rate = 1 / 20),
         lambda = rnorm(R, mean = qlogis(0.05), sd = 0.2),
         theta_House = rnorm(R, mean = -0.25, sd = 0.1),
         theta_Senate = rnorm(R, mean = -0.25, sd = 0.1)) %>%
  rowwise %>% # then draw the outcomes
  summarize(eta = alpha +
              beta_House_minority * vetoes$House_minority +
              beta_Senate_minority * vetoes$Senate_minority +
              beta_both_minority *
              vetoes$House_minority * vetoes$Senate_minority +
              beta_House * vetoes$House +
              beta_Senate * vetoes$Senate,
            mu = exp(eta),
            epsilon = rgamma(nrow(vetoes), shape = phi, rate = phi),
            regular = rpois(nrow(vetoes), mu * epsilon),
            overrides = rbinom(nrow(vetoes), size = regular,
                               prob = plogis(lambda + theta_House *
                                               vetoes$House +
                                               theta_Senate *
                                               vetoes$Senate)),
            House_minority = vetoes$House_minority,
            Senate_minority = vetoes$Senate_minority) %>%
  ungroup
```

3 Prior Prediction

```
library(ggplot2)
filter(draws, overrides < 25) %>%
ggplot() +
  geom_bar(aes(x = overrides, y = after_stat(prop))) +
  facet_wrap(~ House_minority + Senate_minority)
```



This is somewhat plausible in the sense that if the President's party controls both the House and the Senate, there will be no overrides because there will be essentially no vetoes to override. If the President's party controls neither the House nor the Senate, almost anything could happen but none of those individually have much probability. If the President's party controls either the House or the Senate but not both, then the modal number of overrides is zero, but perhaps there could be some. However, it seems implausible that there would be more than a handful.

4 Posterior | Regular Vetoes

```
library(rstanarm)
options(mc.cores = parallel::detectCores())
```

You need to make sure that your priors are in the same order that R assumes, which is to put the interaction term last.

```
post_regular <- stan_glm.nb(Regular ~ House_minority * Senate_minority +
                              House + Senate,
                            data = vetoes,
                            prior_intercept = normal(log(1), 0.75),
                            prior = normal(c(.25, .25, 0, 0, 1),
                                           c(.1, .1, .05, .05, .5)),
                            prior_aux = exponential(1 / 20))
```

The output can be confusing because it is compatible with that of `MASS::glm.nb`. What is called `reciprocal_dispersion` is $\frac{1}{\phi}$.

```
as_tibble(post_regular) %>%
  summarize(mean_phi = mean(1 / reciprocal_dispersion))
```

```
# A tibble: 1 x 1
  mean_phi
     <dbl>
1     2.74
```

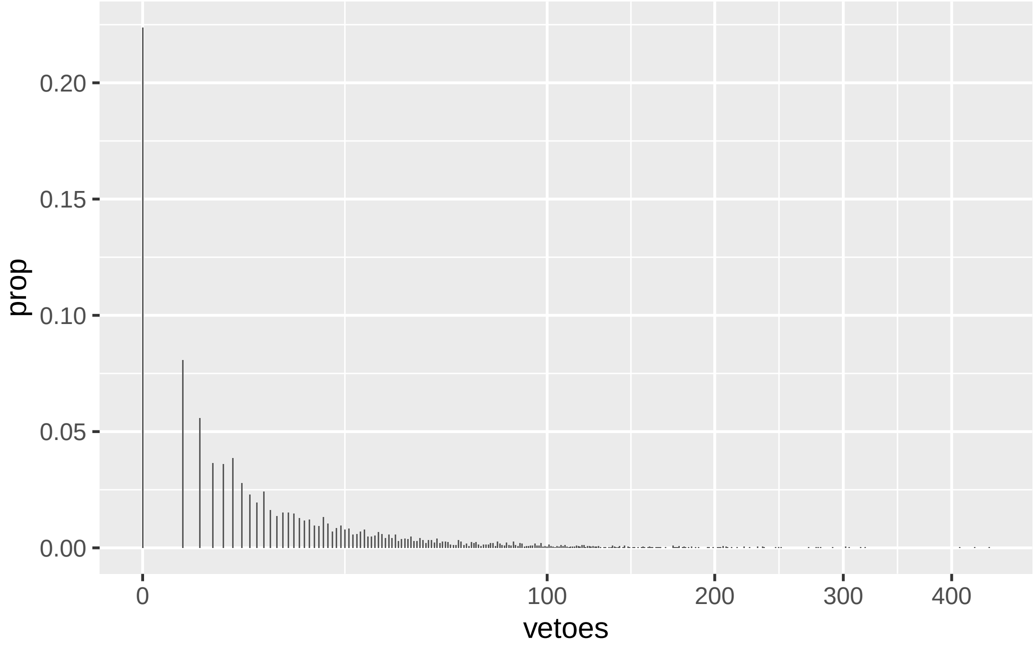
As $\phi \uparrow \infty$, the negative binomial approaches the Poisson (because each $\epsilon_n \rightarrow 1$) and in this case, the average draw of ϕ is much smaller than that (so ϵ_n has some variance).

5 Posterior Prediction

We just need to create a `tibble` that has predictors in the counterfactual scenario where the Democrats only have 49 seats in the Senate.

```
nd <- tibble(House_minority = TRUE, Senate_minority = TRUE,
             House = 49 - 50, Senate = 49 - 50)
PPD <- posterior_predict(post_regular, newdata = nd)
```

```
ggplot(tibble(vetoes = PPD[, 1])) +
  geom_bar(aes(x = vetoes, y = after_stat(prop))) +
  scale_x_sqrt()
```



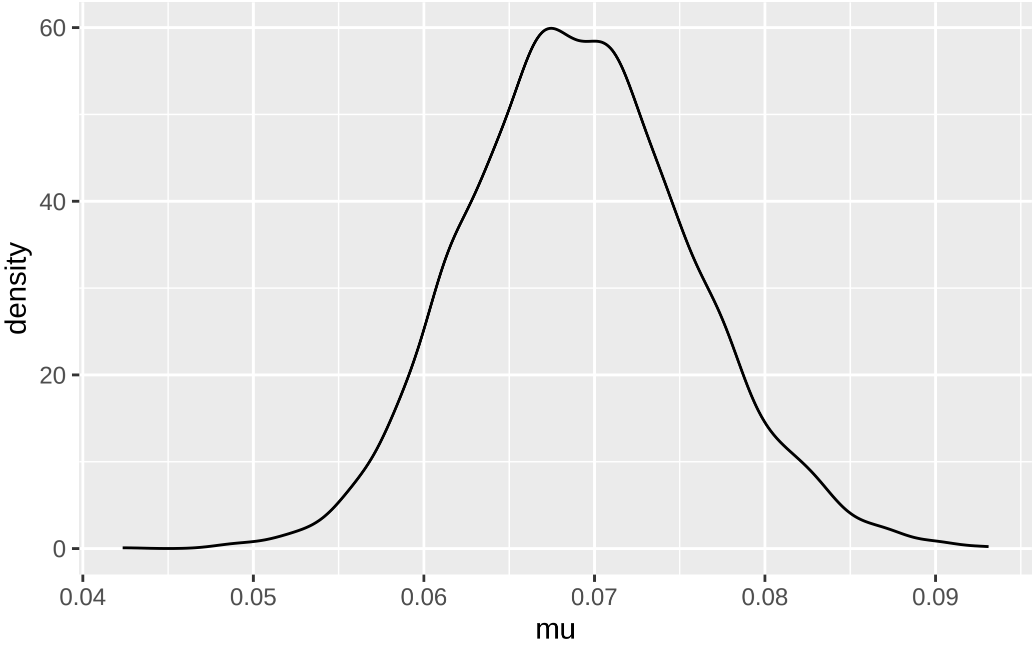
The model still thinks that the most likely number of vetoes is zero, although that only has about a 25% chance. Historically, that might make some sense but it is important to remember that the data-generating process can change over time. If the Republicans controlled the current Senate in addition to the House, there would be a lot of votes on bills that Biden opposes. Maybe the Democrats in the Senate would try to filibuster in order to prevent the vote, but the Republican leadership would probably exclude those bills from being eligible for a filibuster. Thus, Biden would probably veto dozens of bills, although none of them would be overriden.

6 Posterior | Overrides

```
post_overrides <- stan_glm(cbind(Overrides, Regular - Overrides) ~
                             House + Senate,
                           data = vetoes,
                           family = binomial(link = "logit"),
                           prior_intercept = normal(qlogis(0.05), 0.2),
                                    prior = normal(-0.25, 0.1))
```

We can calculate the probability of an override of a vetoed bill as

```
nd <- slice_tail(vetoes, n = 1) # 118th Congress
mu <- plogis(posterior_linpred(post_overrides, newdata = nd))
ggplot(tibble(mu = mu[ , 1])) +
  geom_density(aes(x = mu))
```



Thus, the override probability implied by the model is very small and the above plot probably overstates the reality of it.