Problem Set 2

CS 169 Problem Set 2 Due Friday, October 13th, 2023

1. Bayesian posterior probability for the binomial distribution For this problem, you will complete exercise 2.6 from the textbook, which repeats the Bayesian analysis in Section 2.9. Recall that we are finding the most likely model, theta=(n,p), for the binomial distribution, given data showing 40 successes in 300 trials. But n, the number of trials, is fixed. So we are really just trying to find the highest likelihood for the success probability, incorporating prior beliefs about the distribution of theta.

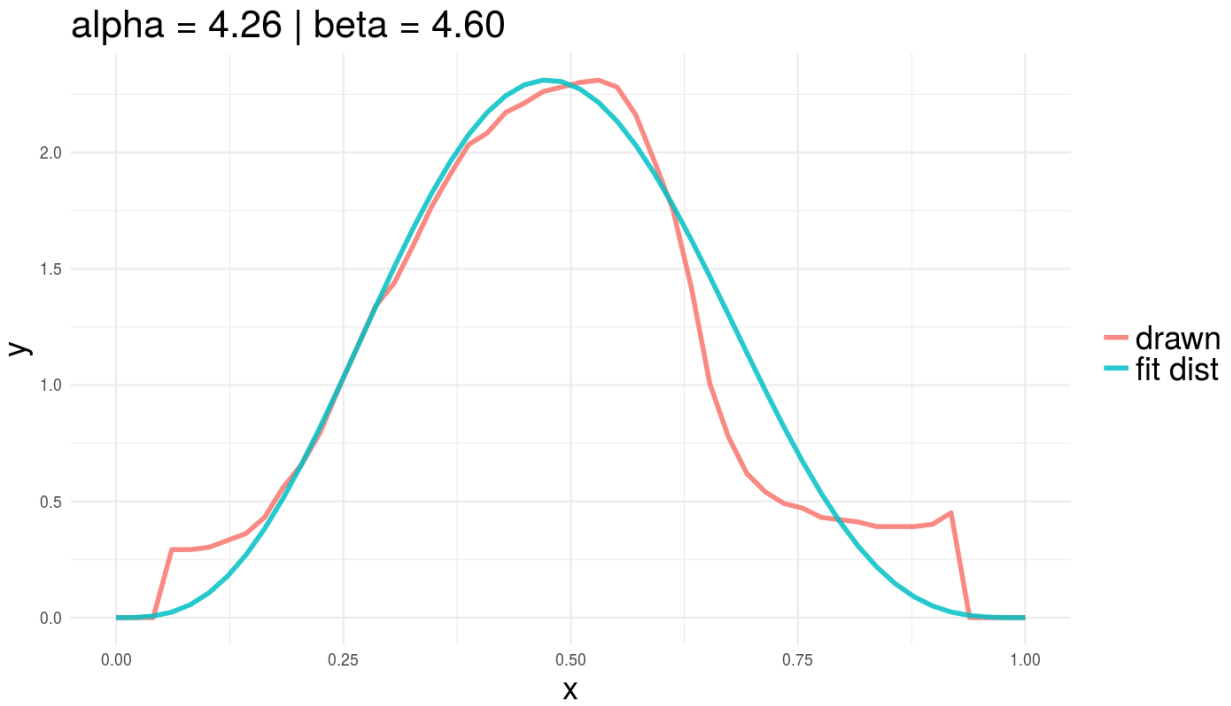
First, use the Shiny app at https://jhubiostatistics.shinyapps.io/drawyourprior/ to draw a prior distribution and find the alpha and beta parameters to the beta distribution that best fit it.

Please don't forget to upload the image file you used with your homework solutions! Please also be sure to submit *both* the html and the .Rmd files.

a) (4 points) Show an image of the prior distribution you chose, and indicate the alpha and beta parameters (either make sure they are visible in the image, or include them separately):

Example image you can delete or replace with your solution:

```
knitr::include_graphics("/Users/jiataizhang/Desktop/beta.png")
```



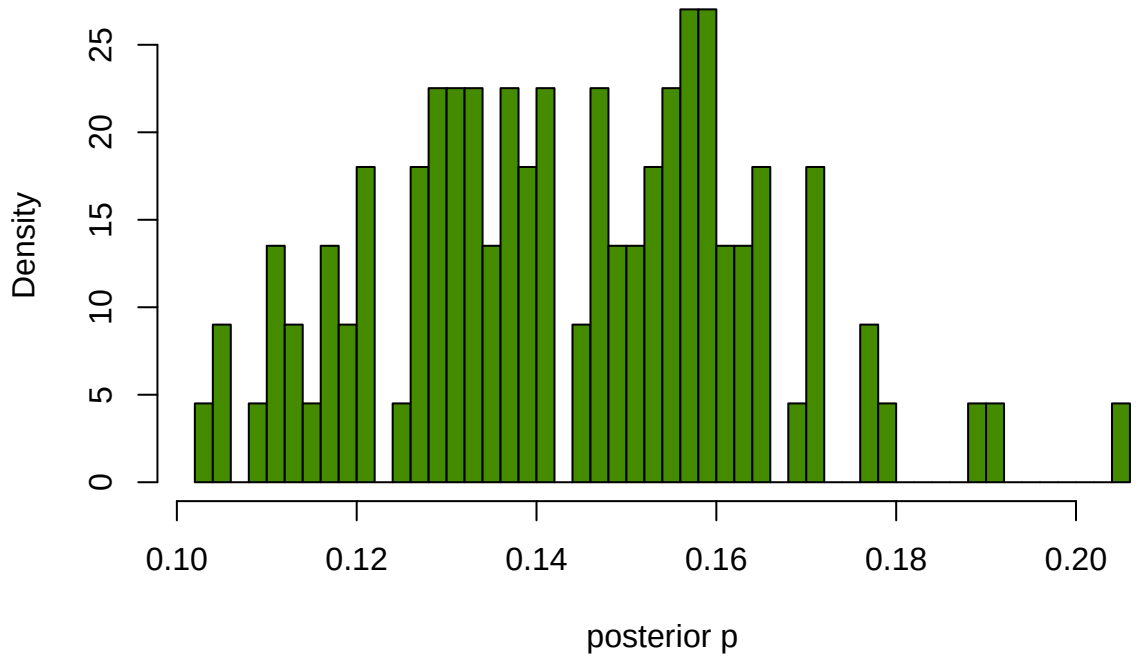
b) (6 points) Using this as the prior probability distribution on p, re-do the analysis in section 2.9.3-4, which uses simulation to estimate the posterior probability distribution. You should change the

analysis from the book to incorporate the new prior on theta that you chose. You should still assume that the data show Y = 40 successes out of n = 300 trials. Plot the histogram of the empirical posterior distribution (thetaPostEmp) given that Y is 40.

Recall that you can find all the code used in the textbook by following the links at the bottom of the course text web site. Your solution will be a minor modification of the code in the text – that is okay for this problem.

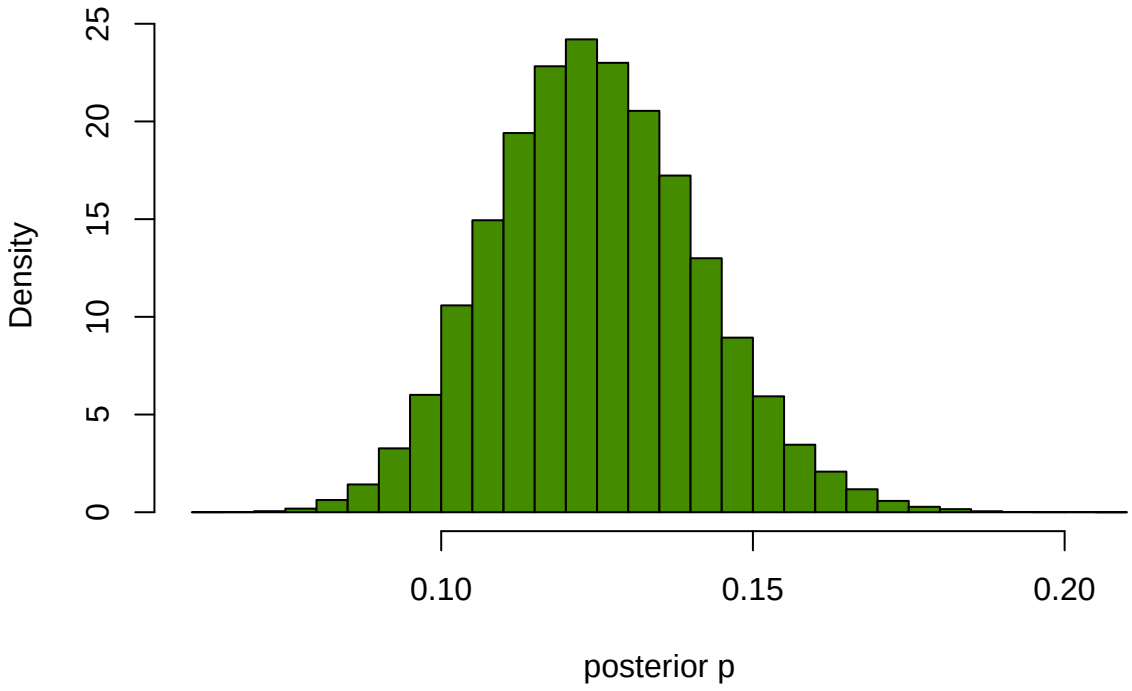
Show your code below.

```
rp <- rbeta(100000, 4.26, 4.60)

y <- vapply(rp, function(x) rbinom(1, prob=x, size=300),
        integer(1))
pPostEmp = rp[ y==40 ]

hist(pPostEmp, breaks = 40, col = "chartreuse4", main = "",
  probability = TRUE, xlab = "posterior p")
```

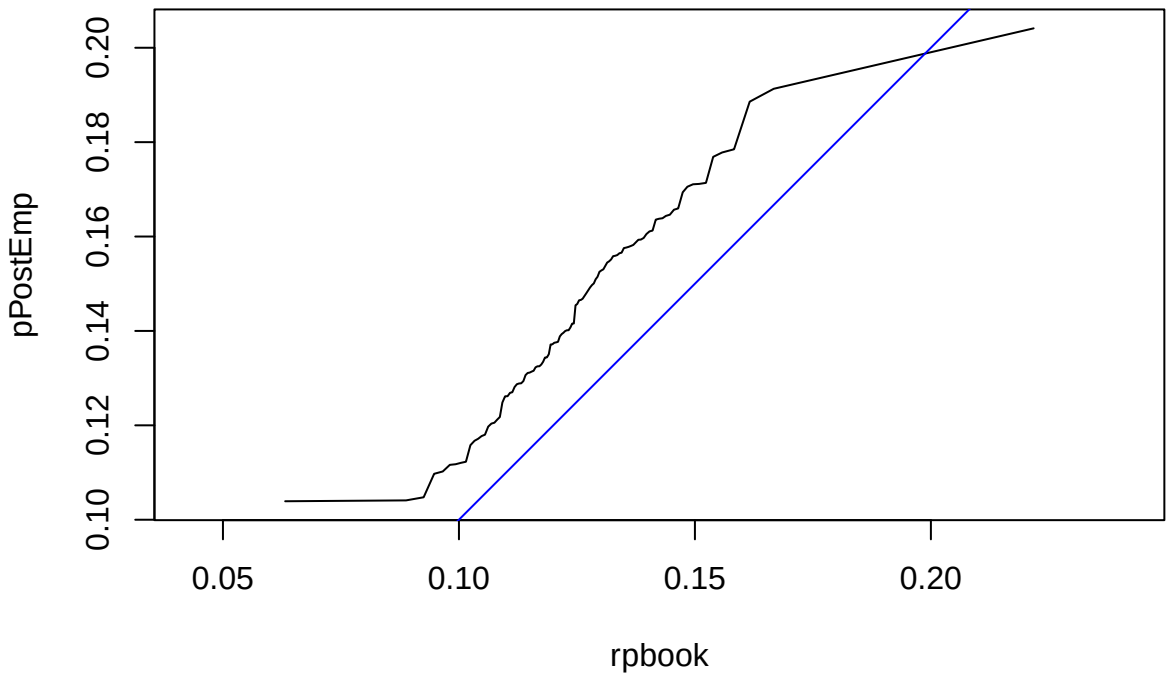


c) (6 points) Show a QQ plot, as in section 2.9.4, comparing your posterior distribution to the empirical posterior distribution given the prior of beta(50,350) used in the textbook.

```
rp = rbeta(100000, 50, 350)
p=rp[y==40]
hist(rp, breaks = 40, col = "chartreuse4", main = "",
  probability = TRUE, xlab = "posterior p")
```




```
rpbook = rbeta(100000, 50, 350)
qqplot(rpbook,pPostEmp,type = "l", asp = 1)
abline(a = 0, b = 1, col = "blue")
```



e) (5 points) To what degree did your changing the prior affect the posterior probability?

Write your answer here Changing the prior from the textbook prior (beta(50, 350)) to the new prior (beta(4.26, 4.60)) caused a significant change. The new prior causes the mean of the posterior distribution to decrease, the variance to decrease, and the distribution to be skewed toward 0, providing more definitive parameter estimates. A QQ plot comparing the new posterior distribution with the empirical posterior distribution from the textbook highlights the difference between the two different priors. This emphasizes the critical nature of prior selection in Bayesian analysis and the impact of different priors on posterior probabilities, as well as the flexibility of the Bayesian approach.

2. Bayes' theorem and COVID-19 testing Imagine that we have two COVID-19 tests we could deploy at a large university. Test 1 has 75% sensitivity (i.e., 3 of every 4 infected people will test positive) and 98% specificity (98 of every 100 uninfected people will test negative). Test 2 has 98% sensitivity but 75% specificity.

a) (5 points) Write a function to compute the "positive predictive value," or PPV, which is the probability that someone who tests positive has the disease. The function should take, as arguments, the sensitivity, specificity, and the overall (prior) probability of infection in the population, and should return the PPV.

```
  ppv <- function(sensitivity, specificity, prior_probability) {
  numerator <- sensitivity * prior_probability
  denominator <- sensitivity * prior_probability + (1 - specificity) * (1 - prior_probability)

  ppv <- numerator / denominator
  return(ppv)
}
```

b) (5 points) Suppose that on "arrival day" the entire student population of 50,000 is tested, and assume that the infection rate in the overall population is 32 per 100,000 (so you'd expect 16 of the 50,000 to be infected).

Given this information, show how you would use your function to compute, for each of the two tests, the probability that someone who tests positive has the disease.

test1:

```
sensitivity_1 <- 0.75
specificity_1 <-0.98
prior_probability <- 16/50000
ppv_1 <- ppv(sensitivity_1, specificity_1, prior_probability)
ppv_1
```

```
## [1] 0.01186146
```

test2

```
sensitivity_2 <- 0.98
specificity_2 <- 0.75
prior_probability <- 16/50000
ppv_2 <- ppv(sensitivity_2, specificity_2, prior_probability)
ppv_2
```

```
## [1] 0.001253229
```

c) (5 points) For each of the two tests, under these population assumptions, how many false positives (uninfected people who test positive) would you expect to see? How many infected people will be expected to have (false) negative test results? Show your work. You don't really need R for this necessarily, but we want to see your thought process.

test1

```
expect_infect <- 16
FP_1 <- (1 - specificity_1) *(50000-16)
FN_1 <- (1 - sensitivity_1) *expect_infect



FP_1
```

```
## [1] 999.68
```

```
FN_1
```

```
## [1] 4
```

test2

```
FP_2 <- (1 - specificity_2) * (50000-16)
FN_2 <- (1 - sensitivity_2) * expect_infect

FP_2
```

```
## [1] 12496
```

```
FN_2
```

```
## [1] 0.32
```

d) (7 points) Now suppose that there is an outbreak: 2000 of the 50,000 people on campus truly have the disease. Under these circumstances, for each of the two tests, what is the probability that someone who tests positive has the disease? How many false positives (uninfected people who test positive) would you expect to see? How many infected people will be expected to have (false) negative test results? Show your work.

test1

```
prevalence <- 2000/50000
ppv_1_1 <- ppv(sensitivity_1, specificity_1,prevalence)


ppv_1_1
```

```
## [1] 0.6097561
```

```
FP_1_1 <- (1 - specificity_1) * (50000-2000)
FN_1_1 <- (1 - sensitivity_1) * 2000

FP_1_1
```

```
## [1] 960
```

```
FN_1_1
```

```
## [1] 500
```

test2

```
ppv_2_2 <- ppv(sensitivity_2, specificity_2,prevalence)
ppv_2_2
```

```
## [1] 0.1404011
```

```
FP_2_2 <- (1 - specificity_2) * (50000-2000)
FN_2_2 <- (1 - sensitivity_2) * 2000


FP_2_2
```

```
## [1] 12000
```

```
FN_2_2
```

```
## [1] 40
```

3. COVID-19 vaccine accuracy Next, we'll consider the following table from an August, 2021 Israeli data set that was used to estimate the first Pfizer vaccine's effectiveness:

```
vaxtable=matrix(c(1302912,5634634,214,301,1116834,3501118,43,11,186078,2133516,171,290),byrow=T,nrow=3)
dimnames(vaxtable)[[1]]=c("All_ages","Under_50","Over_50")
dimnames(vaxtable)[[2]]=c("Pop_Not_Vaxxed","Pop_Vaxxed","Hosp_Not_Vaxxed","Hosp_Vaxxed")
vaxtable
```

```
##          Pop_Not_Vaxxed Pop_Vaxxed Hosp_Not_Vaxxed Hosp_Vaxxed
## All_ages        1302912    5634634             214         301
## Under_50        1116834    3501118              43          11
## Over_50          186078    2133516             171         290
```

The left two columns show the total numbers in the population in each group, vaccinated or not, all together and then broken down by age. The right two columns show the numbers of patients hospitalized with severe disease in each category.

We compute a vaccine's efficacy against hospitalization as 1 - (V/N), where V is the rate of hospitalization *per 100,000* fully-vaccinated people, and N is the rate of hospitalization per 100,000 non-vaccinated people. So, for example, in the all ages group, V = 301 / 5634634 * 100000, or 5.34 per 100K people; other rates per 100,000 can be computed for each age group and vaccination status.

This formula, 1 - (V/N), measures the percent reduction in the hospitalization rate in the vaccinated versus the unvaccinated groups. This value is between 0 and 1, but we can multiply it by 100 to read the value as percent accuracy.

a) (5 points) From the table data for the all-ages group, what is Pr(Vaccinated|Hospitalized)?

```
hospitalized_vaxxed <- vaxtable["All_ages", "Hosp_Vaxxed"]
hospitalized_not_vaxxed <- vaxtable["All_ages", "Hosp_Not_Vaxxed"]

pr <- hospitalized_vaxxed / (hospitalized_vaxxed + hospitalized_not_vaxxed)

pr
```

```
## [1] 0.584466
```

b) (5 points) What is wrong with the argument that, since this value is relatively large, the vaccine isn't very effective?

Write your answer here In simple terms, the argument that a relatively large value of Pr(Vaccinated|Hospitalized) implies poor vaccine effectiveness is incorrect. This conditional probability only reflects the proportion of vaccinated individuals among those hospitalized and does not directly measure the overall vaccine efficacy. To assess vaccine effectiveness correctly, one should compare the hospitalization rates between vaccinated and unvaccinated individuals while considering other relevant factors.

c) (8 points) Write a function that takes as input a vector containing the four values in a row of the table (pop_unvaxxed, pop_vaxxed, hosp_unvaxxed, hosp_vaxxed) and returns the percent vaccine efficacy. Show how to use this function to compute the percent vaccine efficacy for all three age groups in the table.

```
 vaccine_efficacy <- function(row_values) {
  pop_unvaxxed <- row_values[1]
  pop_vaxxed <- row_values[2]
  hosp_unvaxxed <- row_values[3]
  hosp_vaxxed <- row_values[4]


  rate_hosp_vaxxed <- (hosp_vaxxed / pop_vaxxed) * 100000


  rate_hosp_unvaxxed <- (hosp_unvaxxed / pop_unvaxxed) * 100000


 percent_vaccine_efficacy <- 1 - (rate_hosp_vaxxed / rate_hosp_unvaxxed)

  return(percent_vaccine_efficacy)
}


all_ages_values <- c(1302912, 5634634, 214, 301)
under_50_values <- c(1116834, 3501118, 43, 11)
over_50_values <- c(186078, 2133516, 171, 290)

efficacy_all_ages <- vaccine_efficacy(all_ages_values)
efficacy_under_50 <- vaccine_efficacy(under_50_values)
efficacy_over_50 <-  vaccine_efficacy(over_50_values)


efficacy_all_ages
```

```
## [1] 0.6747614
```

```
efficacy_under_50
```

```
## [1] 0.918397
```

```
efficacy_over_50
```

```
## [1] 0.8520888
```

e) (5 points) Can you explain why the overall efficacy might be so much worse in the full data set than in the groups broken down by age? Specifically, what might be confounded in this data set? (Researching "Simpson's Paradox" online might be helpful here, but please try to explain in your own words.

The overall lower vaccine efficacy in the full dataset compared to the age-group breakdown can typically be attributed to two main factors: the distribution of age groups and baseline risk. These two factors may lead to the occurrence of Simpson's Paradox.Those below 50 years of age make up the majority of the data, and their baseline risk is usually lower. This situation would result in a lower overall baseline risk in the dataset.

4. Counting mutant mice Suppose that you have mouse strains with two different mutations of the neuroligin 1 (NLGN1) gene, whose role in psychiatric and neurological disease is suspected but under-characterized. The mutations are called NLGN1mutA and NLGN1mutB. Homozygotes for either mutation (e.g., mice with two NLGNmutA or two NLGNmutB alleles) are unable to reproduce. To create a mouse with two different mutant copies of the gene, you therefore breed NLGN1mutA/+ and NLGN1mutB/+ mice, hoping for NLGN1mutA/NLGN1mutB offspring. The problem is that only 1/4 of the pups have this genotype.

Suppose you need 5 mice with this genotype to conduct your experiments. You would expect to see 5 such pups out of the first 20. However, there is a lot of variability in this. In this problem, you will estimate the probability that you would need to examine more than 40 pups to get the desired 5.

Specifically, you will simulate the probability of this occurring, based on the assumption that the desired genotype occurs independently in each animal with probability 1/4. We will model this as a Bernoulli process where the probability of a "success" is 0.25 and the probability of a "failure" - that is, of seeing any of the other three possible genotypes - is 0.75.

a) (5 points) First, let's confirm that we will almost definitely see at least five of the mutA/mutB mice if we examine a huge number of pups, say 100.
Create a vector containing the number of successes observed in 10000 simulated Bernoulli trials of size 100 with success probability 0.25. What is the lowest number of successes you ever observe?

```
n <- 10000
size <- 100
prob <- 0.25

results <- rbinom(n, size, prob)

lowest_number <- min(results)

lowest_number
```

```
## [1] 10
```

b) (6 points) Now, simulate 100 individual (size=1) Bernoulli trials with a probability of success of 0.25. This should print a vector of 100 zeros and ones. Use the cumsum() cumulative sum function to create a vector of the number of successes seen so far (it should look like '0 0 0 0 1 1 2 2 2 2 2 3 3 . . . ').

```
num_trials <- 100
prob <- 0.25

individual_trials <- rbinom(num_trials, 1, prob)

cumulative_sum <- cumsum(individual_trials)

cumulative_sum
```

```
##   [1] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7
##  [75] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9
```

c) (6 points) How can you use the result in part b) to find the *position* of the 5th success in 100 individual (size=1) Bernoulli trials with success probability 0.25? (For example, if there are successes in the 2nd, 3rd, 7th, 11th, and 17th positions, you want to print 17.) Show code that finds this position.

```
position_of_5th_success <- which(cumulative_sum == 1)[5]
position_of_5th_success
```

```
## [1] 6
```

d) (7 points) Now, we'll try to answer our initial question using Monte Carlo simulation. Create a vector called `fifthmice` containing 10000 replicates of the calculation of the position of the fifth successes (i.e., the fifth suitable mutant mouse) in part c. What is the 95th percentile of this distribution? If you have to examine 40 mice to find 5 of the right ones, to what quantile does that correspond?

```
fifthmice <- numeric(10000)
for (i in 1:10000) {
individual_trials <- rbinom(100, 1, prob = 0.25)
fifthmice[i] <- which(individual_trials == 1)[5]}

percentile_95th <- quantile(fifthmice, probs = 0.95, na.rm = TRUE)
mice_40 <- sum(fifthmice <= 40) / 10000
percentile_95th
```

```
## 95%
##  35
```

```
mice_40
```

```
## [1] 0.9833
```

e) (5 points) Next, let's try to model the distribution of the number of "failures" (in this case, mice with other genotypes) we see before we see our fifth double-mutant mouse.

Load the "vcd" package in R. This package includes the `goodfit` function described in Chapter 2 of your text.

Then, create a vector called `othermice` that contains the values obtained by subtracting 5 from the values in `fifthmice`. (This corresponds to the number of mice with other genotypes observed before the fifth double mutant mouse in each replicate).

To which, if any, of the three distributions in goodfit (Poisson, binomial, negative binomial) is the `othermice` vector a good fit? What are the parameters of the best fitted model?

```
library(vcd)
```

```
## Loading required package: grid
```

```
othermice <- fifthmice - 5

poisson_fit <- goodfit(othermice, type = "poisson")

binomial_fit <- goodfit(othermice, type = "binomial")
```

```
## Warning in goodfit(othermice, type = "binomial"): size was not given, taken as
## maximum count
```

```r
negbin_fit <- goodfit(othermice, type = "nbinomial")

summary(poisson_fit)
```

```
##
##   Goodness-of-fit test for poisson distribution
##
##                       X^2 df P(> X^2)
## Likelihood Ratio 15704.37 57        0
```

```r
summary(binomial_fit)
```

```
##
##   Goodness-of-fit test for binomial distribution
##
##                       X^2 df P(> X^2)
## Likelihood Ratio 26114.29 57        0
```

```r
summary(negbin_fit)
```

```
##
##   Goodness-of-fit test for nbinomial distribution
##
##                       X^2 df   P(> X^2)
## Likelihood Ratio 77.82356 56 0.02848093
```

f) (5 points) Given what you know or can look up about the distribution you selected, do your conclusions make sense? Why?

The results of the Goodness-of-Fit tests for both the Poisson and Binomial distributions show that these two distributions are not well suited to characterize my data because there are significant differences. The test results for the Negative Binomial distribution show some differences, but these differences may not be very significant. This may indicate that the Negative Binomial distribution fits my data better.