Problem Set 1

Michael Fryer

Collaborators: Florian

For this assignment, I collaborated with Florian. We discussed our independent solutions to the assignment and discussed our thoughts on part 4 (reflecting on this assignment).

1 COVID Home Test

In a clinical trial, the BinaxNOW home COVID-19 antigen test correctly gave a positive result 75.5% of the time and correctly gave a negative result 99.5% of the time. For the next set of questions, assume that presence of the antigen suffices for a person to have COVID-19 and absence of the antigen suffices for that person to not have COVID-19. Finally, assume that 10% of the people in your community are currently infected with COVID-19. (This is your base rate for exposure.)

```
# Data from above
p_infected <- 0.1
sensitivity_binax <- 0.755
specificity_binax <- 0.995
# Bayes helper for later
bayes <- function(prior, sensitivity, specificity) {
  p_evidence <- (sensitivity * prior) + ((1 - specificity) * (1 - prior))
  return((sensitivity * prior) / p_evidence)
}
```

a) Suppose you take a BimaxNOW COVID-19 antigen test. What is the probability of an administered BinaxNOW test returning to you a positive result?

The probability of an administered BinaxNOW test returning to you a positive result is equal to $P(T = 1) = sentitivity * prior + (1 - specificity) * (1 - prior)$.

```
p_pos <- (
  sensitivity_binax * p_infected + (1 - specificity_binax) * (1 - p_infected)
)
```

```
## Probability of getting a positive result:  0.08
```

b) Suppose the infection rate in New Zealand is 1.5% and a New Zealander takes a BinaxNOW test. What is the probability that this test will return a positive result?

This is the same as above, however we use different values for our base rate.

```
p_infected_nz <- 0.015
p_pos <- (
  sensitivity_binax * p_infected_nz + (1 - specificity_binax) * (1 - p_infected_nz)
)
```

```
## Probability of getting a positive result (in New Zealand):  0.01625
```

c) A competitor offers a test with a sensitivity 90% but specificity 99.0% (vs BinaxNOW's 99.5%).

- Calculate $P(C = 1|T = 1)$ for both tests.

```
sensitivity_competitor <- .9
specificity_competitor <- .99
# BinaxNOW
p_infected_g_pos_binax <- bayes(
  p_infected, sensitivity_binax, specificity_binax
)
# Competitor
p_infected_g_pos_c <- bayes(
  p_infected, sensitivity_competitor, specificity_competitor
)
```

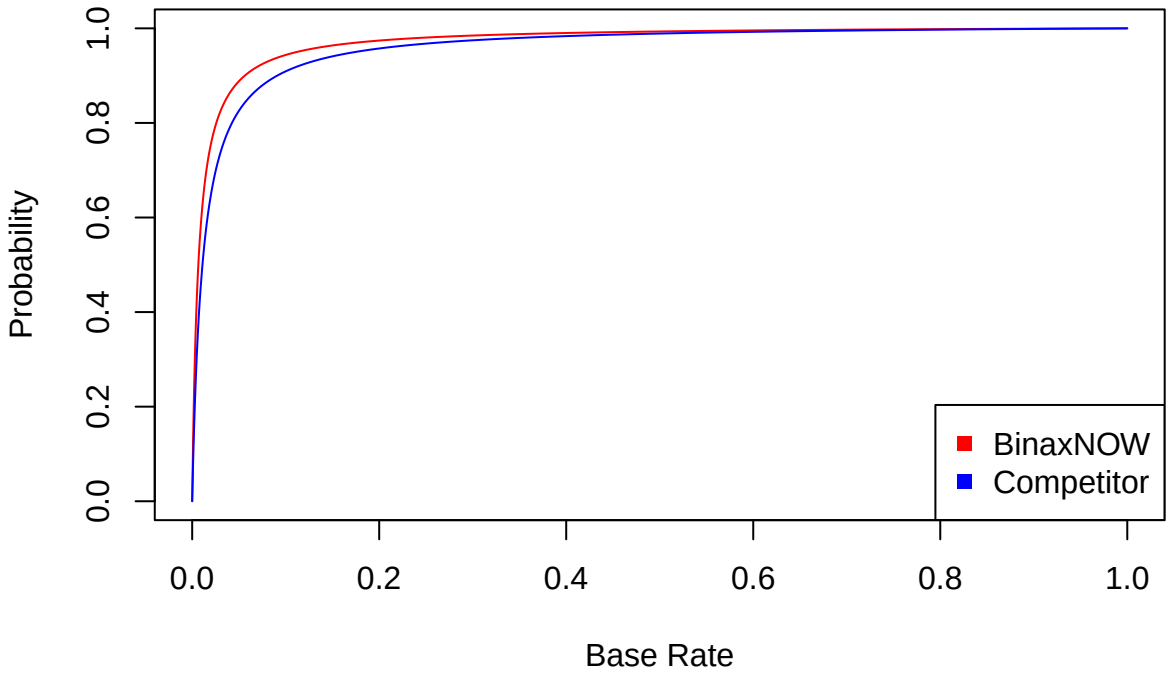
```
## Probability of Having COVID Given Positive Result:

## BinaxNow:    0.94375

## Competitor:  0.9090909
```

- The competitor's test costs 2x more. For which base rates (if any) would you prefer the competitor's test?

The base rates you would prefer are the ones wherein testing with a single competitor's test out performs testing twice with BinaxNOW. That is, the base rates where $P(C = 1|T = 1)$ is higher for the competitor when testing with at most 2 BinaxNOW tests. In this case, that is none.

- Plot $P(C = 1|T = 1)$ vs base rate for both tests on the same graph.

```
num_rates <- 1000
base_rates <- seq(from=0, to=1, length.out=num_rates)

# BinaxNOW
binax_rates <- sapply(base_rates,
  function(p) bayes(p, sensitivity_binax, specificity_binax)
)
# Competitor
competitor_rates <- sapply(base_rates,
  function(p) bayes(p, sensitivity_competitor, specificity_competitor)
)
```



- Write 2-3 sentences explaining at which base rates each test is preferable and why the preference changes (or doesn't change).

Within the context of covid, we would prefer the BinaxNOW test for all base rates. Compared to the competitor, BinaxNOW is more accurate at determining an infected individual (red line is "above" the blue line).

d) Suppose the competitor offers a second generation test to you with a sensitivity 82% and specificity 99.7%.

- Calculate $P(C = 1|T = 1)$ for BinaxNOW vs the competitor's generation 2 test.

```
sensitivity_competitor <- .82
specificity_competitor <- .997
# BinaxNOW
p_infected_g_pos_binax <- bayes(
  p_infected, sensitivity_binax, specificity_binax
)
p_infected_g_pos_binax_second <- bayes(
  p_infected_g_pos_binax, sensitivity_binax, specificity_binax
)
# Competitor
p_infected_g_pos_c <- bayes(
  p_infected, sensitivity_competitor, specificity_competitor
)
```

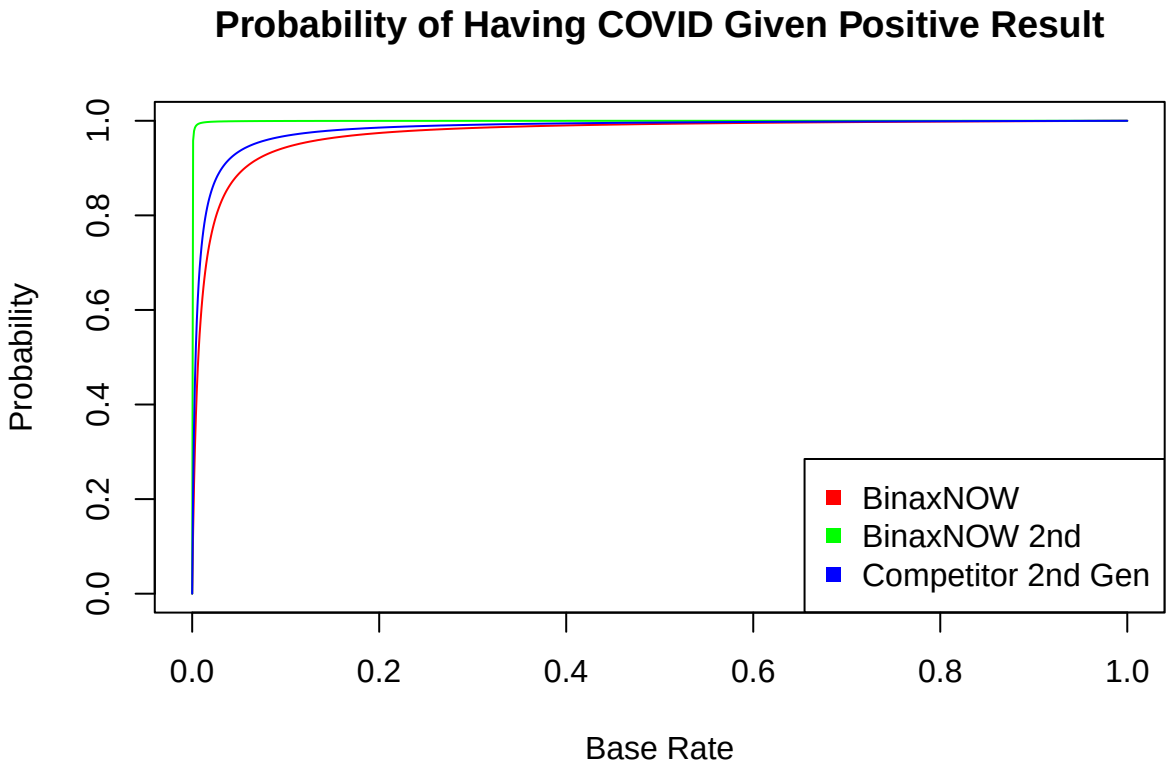
```
## P(C=1|T=1):

## BinaxNow:         0.94375

## BinaxNow (2nd):   0.9996054

## Competitor:       0.9681228
```

- The competitor's new test also costs 2x more than BinaxNOW. For which base rates (if any) would you prefer the competitor's new test?

The same argument applies here as in ***c)*** *from above. That is, when two or less BinaxNOW tests out perform a single competitor's test or, if monetary cost is not a concern.*

- Plot $P(C = 1|T = 1)$ vs base rate for both tests on the same graph.

```
num_rates <- 1000
base_rates <- seq(from=0, to=1, length.out=num_rates)
# BinaxNOW
binax_rates <- sapply(base_rates,
  function(p) bayes(p, sensitivity_binax, specificity_binax)
)
binax_rates_second <- sapply(binax_rates,
  function(p) bayes(p, sensitivity_binax, specificity_binax)
)
# Competitor
competitor_rates <- sapply(base_rates,
  function(p) bayes(p, sensitivity_competitor, specificity_competitor)
)
```



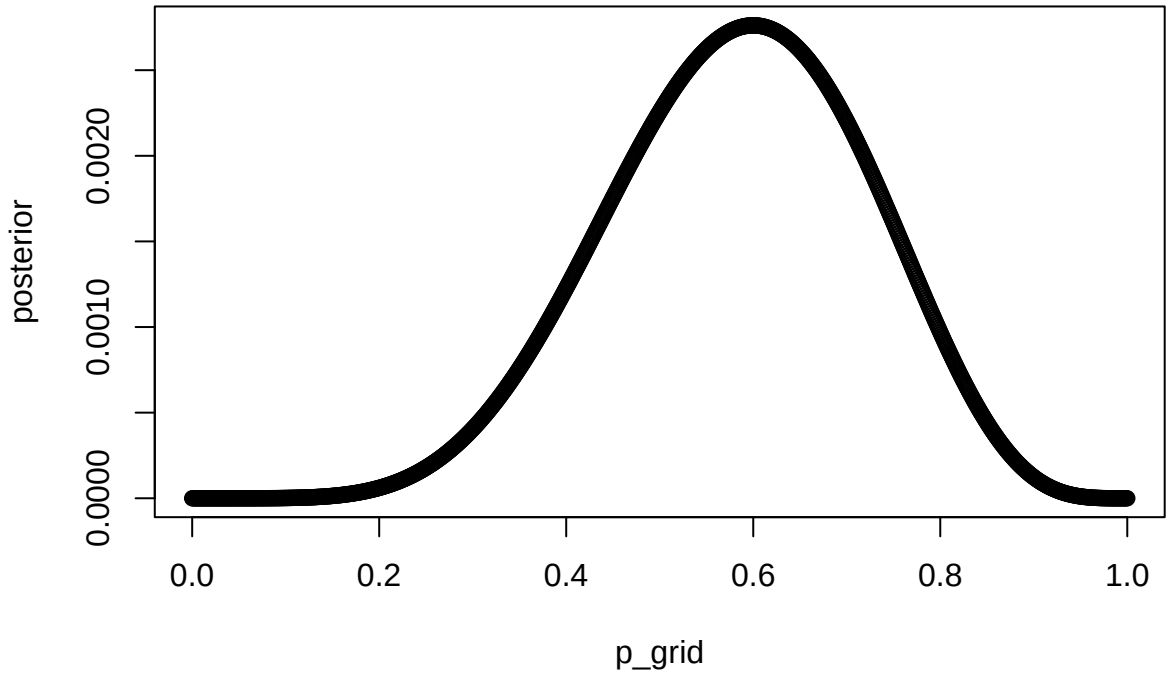
- Write 2-3 sentences explaining at which base rates each test is preferable and why the preference changes (or doesn't change).

Unlike ***c)*** *from above, and barring monetary concern, the competitor is preferred at all base rate values. The same argument applies here as above, that being blue above red.*

2 Computing Probabilities

Implement and run the following chunk of code to create a distribution, samples. Create a plot of that distribution. Then, where called for, write a short line of R code to compute an answer to each question. Analytical solutions will not be accepted.

```r
set.seed(712)
p_grid <- seq(from=0, to=1, length.out=1000)
prior <- rep(1, 1000)
likelihood <- dbinom(6, size=10, prob=p_grid)
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
samples <- sample(p_grid, prob=posterior, size=1e4, replace=TRUE)
plot(p_grid, posterior, type="b")
```



a) How much posterior probability lies below $p = 0.5$?

```r
sum(samples < 0.5) / length(samples)
```

```
## [1] 0.2757
```

b) How much posterior probability lies above $p = 0.8$?

```r
sum(samples > 0.8) / length(samples)
```

```
## [1] 0.0486
```

c) How much posterior probability likes between $p = 0.2$ and $p = 0.8$?

```r
sum(samples > 0.2 & samples < 0.8) / length(samples)
```

```
## [1] 0.9491
```

d) 20% of the posterior probability likes below which value of p?

```r
quantile(samples, 0.2)
```

```
##       20%
## 0.4624625
```

3 Swing Voters

Write your own R code chunks to answer the following questions. Analytical solutions will not suffice.

a) Imagine a country where there are only two political parties, Red and Blue, which divide the electorate equally. One difference between registered Blue voters and registered Red voters is their willingness to vote for the opposing party's candidate. Blue voters vote Red 20% of the time, otherwise they vote Blue. Red voters vote Blue 10% of the time, otherwise they vote Red. Voters who switch are called swing voters.

Smith was a swing voter in the last election but you do not know whether he is Red or Blue. (Nobody changes parties.) What is the probability that Smith will be a swing voter in the next election? Explain your reasoning.

We will simulate voting using the conditions outlined above. By running this simulation, we can use it to predict whether smith will end up swinging again.

```
set.seed(42)
# Probabilities of each party
p_red <- 0.5
p_blue <- 0.5
# probabilities of swing voters in each party
p_red_swing <- 0.1
p_blue_swing <- 0.2
# Number of voters to simulate
num_times <- 1e4
# Probability of a swing voter
p_swing <- p_red_swing * p_red + p_blue_swing * p_blue
# should be 1/3
p_red_given_swing <- p_red_swing * p_red / p_swing
# should be 2/3
p_blue_given_swing <- p_blue_swing * p_blue / p_swing
# Simulate Smith's party
smith_party <- sample(
  c("R", "B"), prob=c(p_red_given_swing, p_blue_given_swing),
  size=num_times, replace=TRUE
)
# Simulate what Smith will vote next
smith_vote_next <- ifelse(smith_party == "R",
  sample(
    c("R", "S"), prob=c(1 - p_red_swing, p_red_swing),
    size=num_times, replace=TRUE
  ),
  sample(
    c("B", "S"), prob=c(1 - p_blue_swing, p_blue_swing),
    size=num_times, replace=TRUE
  )
)
# Calculate the mean to get the probability Smith will be a swing voter again
smith_swing <- smith_party != smith_vote_next
p_swing_again <- sum(smith_swing) / length(smith_swing)
```

```
## Probability Smith will swing again:  0.1669
```

b) Now imagine a country where there are three political parties: Red, Blue, and Green. Red voters vote Blue 10% of the time, vote for Green 5% of the time, and vote their own party, Red, 85% of the time. Blue voters vote Red 15% of the time, Green 5% of the time, and their own party the remaining 80% of the time. Finally, Green votes Blue 20% of the time, Red 10% of the time, and Green the remainder. The electorate is

evenly among the three parties.

What is the probability that a swing voter in the last election between Red, Blue, and Green, will be a swing voter in the next election? (Like before, nobody changes parties.) Explain your reasoning.

The same general idea as above holds, we will update our priors based on new information. However, there is additional computation since there are now 3 parties instead of just 2.

```
set.seed(42)
# Probabilities of each party
prior <- c(
  R=1/3,
  B=1/3,
  G=1/3
)
# Probabilities of swing voters in each party
p_party_swing <- c(
  R=0.1 + 0.05,
  B=0.15 + 0.05,
  G=0.1 + 0.2
)
# Number of voters to simulate
num_times <- 1e4
# Probability of a swing voter
p_swing <- sum(prior * p_party_swing)
p_party_given_swing <- prior * p_party_swing / p_swing
# Simulate Smith's party
smith_party <- sample(
  c("R", "B", "G"),
  prob=p_party_given_swing,
  size=num_times, replace=TRUE
)
# Simulate what Smith will vote next
smith_vote_next <- sapply(smith_party,
  function(x) ifelse(runif(1) < p_party_swing[[x]], "S", x)
)
# Calculate the mean to get the probability Smith will be a swing voter again
smith_swing <- smith_party != smith_vote_next
p_swing_again <- sum(smith_swing) / length(smith_swing)
```

```
## Probability Smith will swing again:  0.2376
```

4 Reflection

Look back at problems 1 to 3. In each case, you updated beliefs based on observations:

- In problem 1, test result → disease status.
- In problem 2, data → parameter value.
- In problem 3, past behavior → future behavior.

Write a 3-4 sentence paragraph explaining what these three problems have in common from a Bayesian perspective. What role do priors play in each?

From a Bayesian perspective, we are concerning ourselves with updating our prior beliefs with new information. This is circular reasoning, I admit, but important nonetheless. In each example we take some previous knowledge, and update our small-world view with new information. In problem 1 this is done using Bayes,

in problem 2 we draw conclusions directly from the posterior (our small-world view), and in problem 3 by generating samples and then again drawing conclusions directly from the posterior.

5 AI Declaration

ChatGPT was used to help understand how to write R code. Specifically for problem 3, ChatGPT was used to understand how broadcasting operations works. This was useful for both the analytical solution (omitted from the knitted pdf) and for the numeric solution. Aside from those cases, ChatGPT was not used in this assignment.