Computational Statistics and Probability

Problem Set 1 - Bayesian Inference
Due: 23:59:59 12.nov.2025

Fall 2025

Instructions

Assignments must be submitted through Canvas. See the course Canvas page for policies covering collaboration, acceptable file formats (`.Rmd` & `.pdf`), and late submissions. Completed assignments must include executable code (`.Rmd`) for every answer and a corresponding knitted markdown file (`pdf`). The knitted `pdf` is the graded assignment; the `.Rmd` file is used for verification only. A R Markdown cheat sheet is available.

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.)

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

```
# ANSWER 1a)
# The first step is to define your variables to setup Bayes' theorem.
```

$$P(C \mid T) = \frac{P(T \mid C) \times P(C)}{P(T \mid C) \times P(C) + P(T \mid \bar{C}) \times P(\bar{C})} = \frac{0.755 \times 0.10}{0.755 \times 0.100 + 0.005 \times 0.90}$$

```
# Since the variable of interest is binary (i.e., you either have COVID-19 or you
# don't),  the problem is encoded above (for brevity) using the following
# convention: P(C) is the prior probability of contracting COVID-19, P(T|C) is
# the likelihood of a positive home test given that you have COVID-19, and
# C-overbar is the event of not having COVID-19, read "not-C", such that
# P(not-C) = 1- P(C). The denominator is the marginal likelihood.
#
# But to be more explicit, treat T and C as binary random variables. Then,
# we stay that:
#
#  C = 1 encodes the event "has COVID-19"
#  C = 0 encodes the event "does not have COVID-19"
#  T = 1 encodes the event "tested positive for COVID-19 antigens"
#  T = 0 encodes the event "tested negative for COVID-19 antigens"
#
# CALCULATING 1a:
# The probability of testing positive, P( T = 1), is the marginal likelihood.
```

```r
# marginal likelihood func
marginal_likelihood <- function(prior, sensitivity, specificity) {
  # P(T=1) = P(T=1|C=1)*P(C=1) + P(T=1|C=0)*P(C=0)
  # where P(T=1|C=0) = 1 - specificity (false positive rate)
  p_positive <- (sensitivity * prior) + ((1 - specificity) * (1 - prior))
  return(p_positive)
}

# posterior func
ans_1a <- marginal_likelihood(prior = 0.10, sensitivity = 0.755, specificity = 0.995)
print(paste("For you, P(T = 1) =", ans_1a))
```

```
## [1] "For you, P(T = 1) = 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?

```r
# ANSWER 1b):

# To answer the question, change the prior from to 1.5%:
ans_1b <- marginal_likelihood(prior = 0.015, sensitivity = 0.755, specificity = 0.995)
print(paste("For a New Zealander, P(T = 1) =", ans_1b))
```

```
## [1] "For a New Zealander, P(T = 1) = 0.01625"
```

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

```r
# BinaxNOW
sensitivity_binax <- 0.755
specificity_binax <- 0.995

# Competitor
sensitivity_comp <- 0.90
specificity_comp <- 0.99

# Posterior probability of disease given positive test
posterior_given_positive <- function(prior, sensitivity, specificity) {
  p_positive <- marginal_likelihood(prior, sensitivity, specificity)
  (sensitivity * prior) / p_positive
}

# Posterior probability of disease given negative test
posterior_given_negative <- function(prior, sensitivity, specificity) {
  p_negative <- 1 - marginal_likelihood(prior, sensitivity, specificity)
  ((1 - sensitivity) * prior) / p_negative
}
```

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

```r
binax_post <- posterior_given_positive(prior = 0.10,
                        sensitivity = sensitivity_binax,
                        specificity = specificity_binax)
print(paste("P(C=1|T=1) for binax: ", binax_post))
```

```
## [1] "P(C=1|T=1) for binax:  0.94375"
```

```r
comp_post <- posterior_given_positive(prior = 0.10,
                        sensitivity = sensitivity_comp,
                        specificity = specificity_comp)
print(paste("P(C=1|T=1) for competitor: ", comp_post))
```

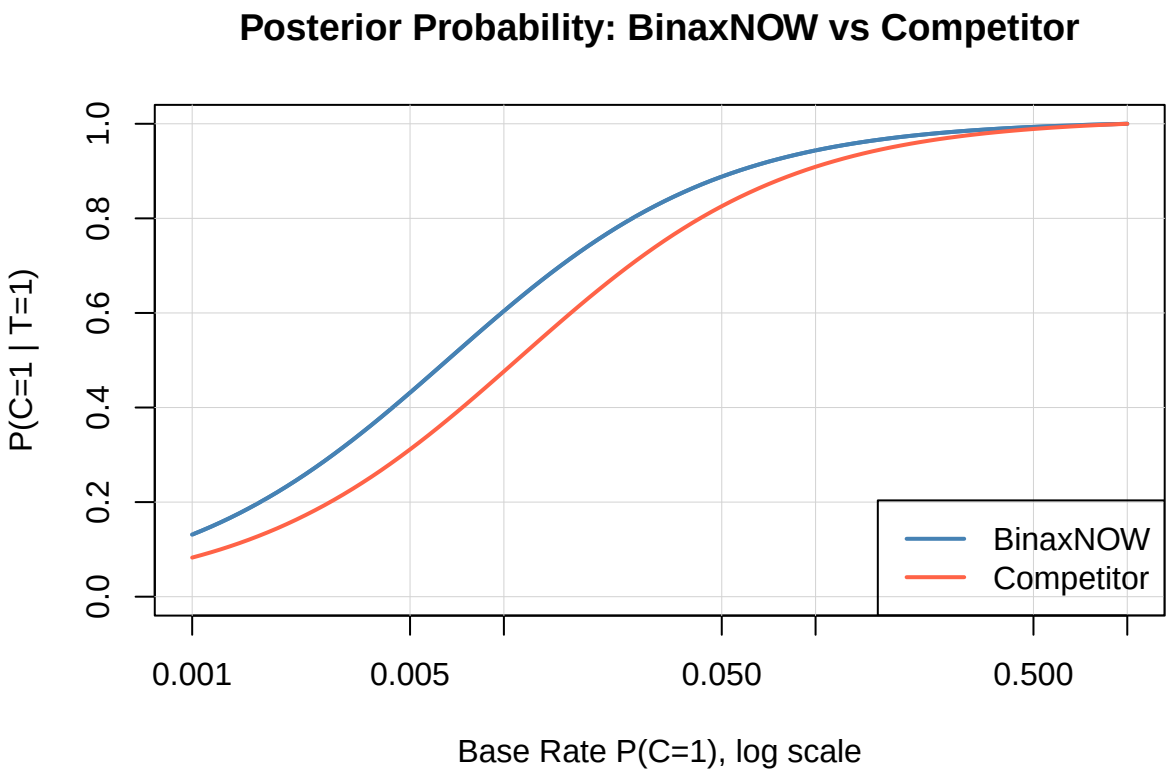
```
## [1] "P(C=1|T=1) for competitor:  0.909090909090909"
```

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

```r
# A table of sample priors, below, points to there being no prior
# at which the competitor dominates BinaxNOW in True Positive Rate.
```

```
##   base_rate  BinaxNOW Competitor  difference better_test
## 1      0.10 0.9437500  0.9090909 -0.03465909    BinaxNOW
## 2      0.05 0.8882353  0.8256881 -0.06254722    BinaxNOW
## 3      0.10 0.9437500  0.9090909 -0.03465909    BinaxNOW
## 4      0.15 0.9638298  0.9407666 -0.02306324    BinaxNOW
## 5      0.20 0.9741935  0.9574468 -0.01674674    BinaxNOW
## 6      0.30 0.9847826  0.9747292 -0.01005337    BinaxNOW
```

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



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

```r
# ANSWER 1c):
# Key point to observe: Binax is unambiguously better for any prior.
# The slight difference in specificity (0.995 vs 0.990) is more
# important in the true positive rate of BinaxNOW dominating the
# Competitor than the larger improvement in sensitivity.
```

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 Binax vs the competitor's generation 2 test.

```
sensitivity_comp2 <- 0.82
specificity_comp2 <- 0.997
print(paste("P(C=1|T=1) for binax: ", binax_post))
```

```
## [1] "P(C=1|T=1) for binax:  0.94375"
```

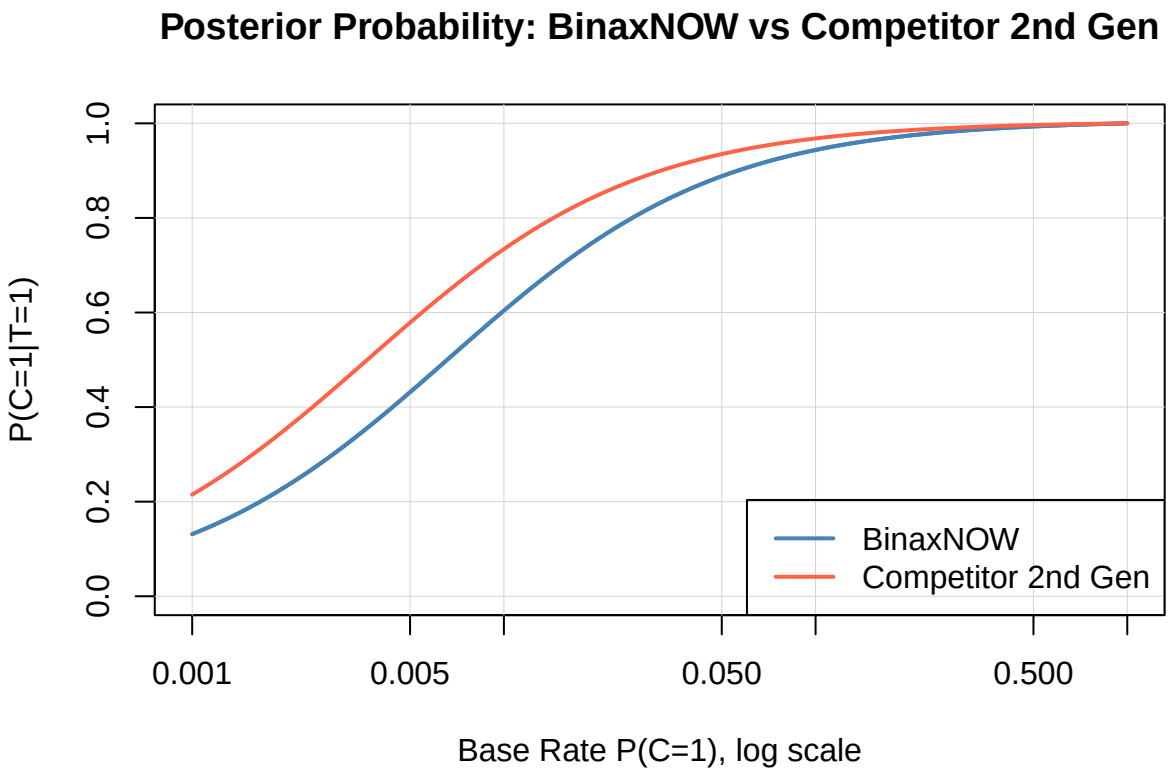
```
comp_post2 <- posterior_given_positive(prior = 0.10,
                        sensitivity = sensitivity_comp2,
                        specificity = specificity_comp2)
print(paste("P(C=1|T=1) for competitor 2nd gen: ", comp_post2))
```

```
## [1] "P(C=1|T=1) for competitor 2nd gen:  0.968122786304604"
```

- 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?

```
##   base_rate  BinaxNOW Competitor2  difference better_test
## 1      0.10 0.9437500   0.9681228 0.024372786 Competitor2
## 2      0.05 0.8882353   0.9350057 0.046770407 Competitor2
## 3      0.10 0.9437500   0.9681228 0.024372786 Competitor2
## 4      0.15 0.9638298   0.9796894 0.015859580 Competitor2
## 5      0.20 0.9741935   0.9855769 0.011383375 Competitor2
## 6      0.30 0.9847826   0.9915357 0.006753062 Competitor2
```

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



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

```r
# NOTE: The Competitor 2nd generation test is superior at ALL base rates shown.

# HOWEVER, since the competitor's generation 2 is twice the cost of a single
# BinaxNOW test, let's compare TWO BinaxNOW tests to ONE competitor test.

# Specifically, use both BinaxNOW tests in series and only call it positive if
# BOTH are positive.  This maximizes specifificty at the cost of sensitivity:

sensitivity_2binax <- sensitivity_binax^2  # Both must detect: 0.755^2 = 0.570
specificity_2binax <- 1 - (1 - specificity_binax)^2  # At least one must
                                                     # correctly reject: 0.9999975
```

```
## Two BinaxNOW (AND rule):
```

```
##   Sensitivity: 0.570025
```

```
##   Specificity: 0.999975
```

```
## Competitor 2nd Gen:
```

```
##   Sensitivity: 0.82
```

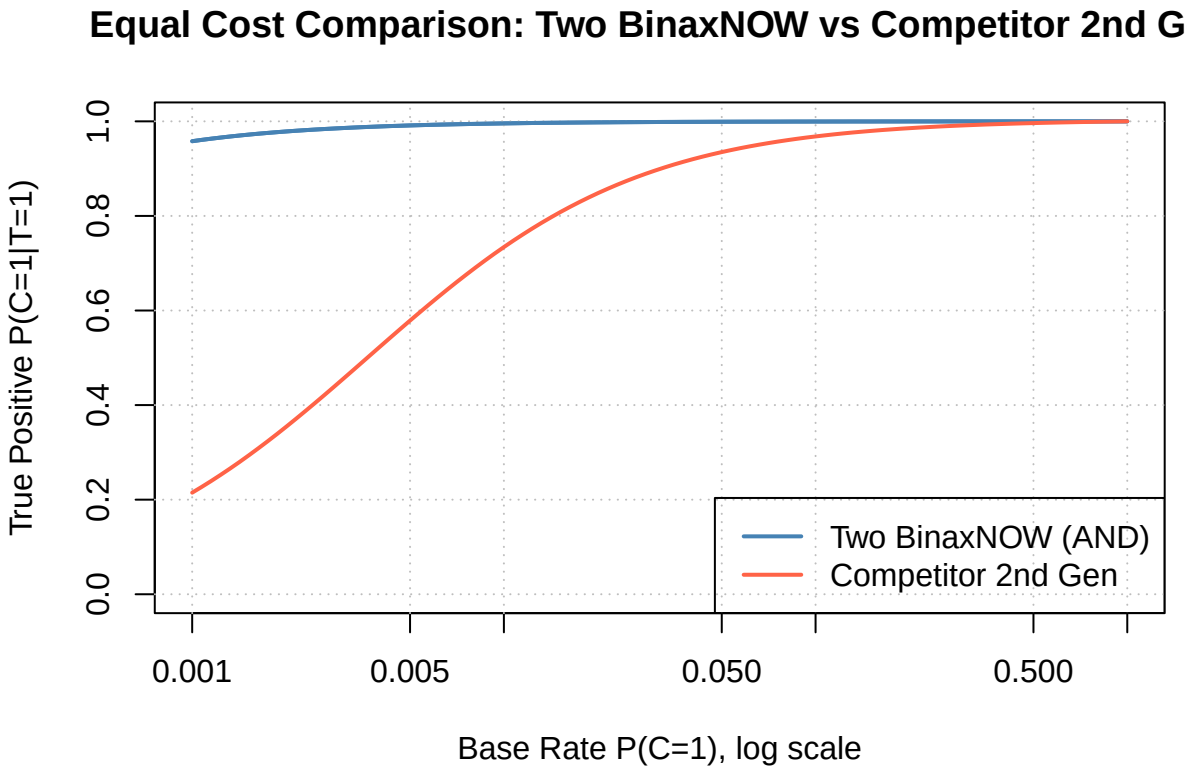
```
##   Specificity: 0.997
```

```r
# Calculate posterior w/ prior = 0.10
binax2_post <- posterior_given_positive(prior = 0.10,
                                        sensitivity = sensitivity_2binax,
                                        specificity = specificity_2binax)
print(paste("P(C=1|T=1) for TWO BinaxNOW: ", binax2_post))
```

```
## [1] "P(C=1|T=1) for TWO BinaxNOW:  0.999605436212187"
```

```r
# Comparison across base rates
comp_post2 <- posterior_given_positive(prior = 0.10,
                                       sensitivity = sensitivity_comp2,
                                       specificity = specificity_comp2)
print(paste("P(C=1|T=1) for competitor 2nd gen: ", comp_post2))
```

```
## [1] "P(C=1|T=1) for competitor 2nd gen:  0.968122786304604"
```



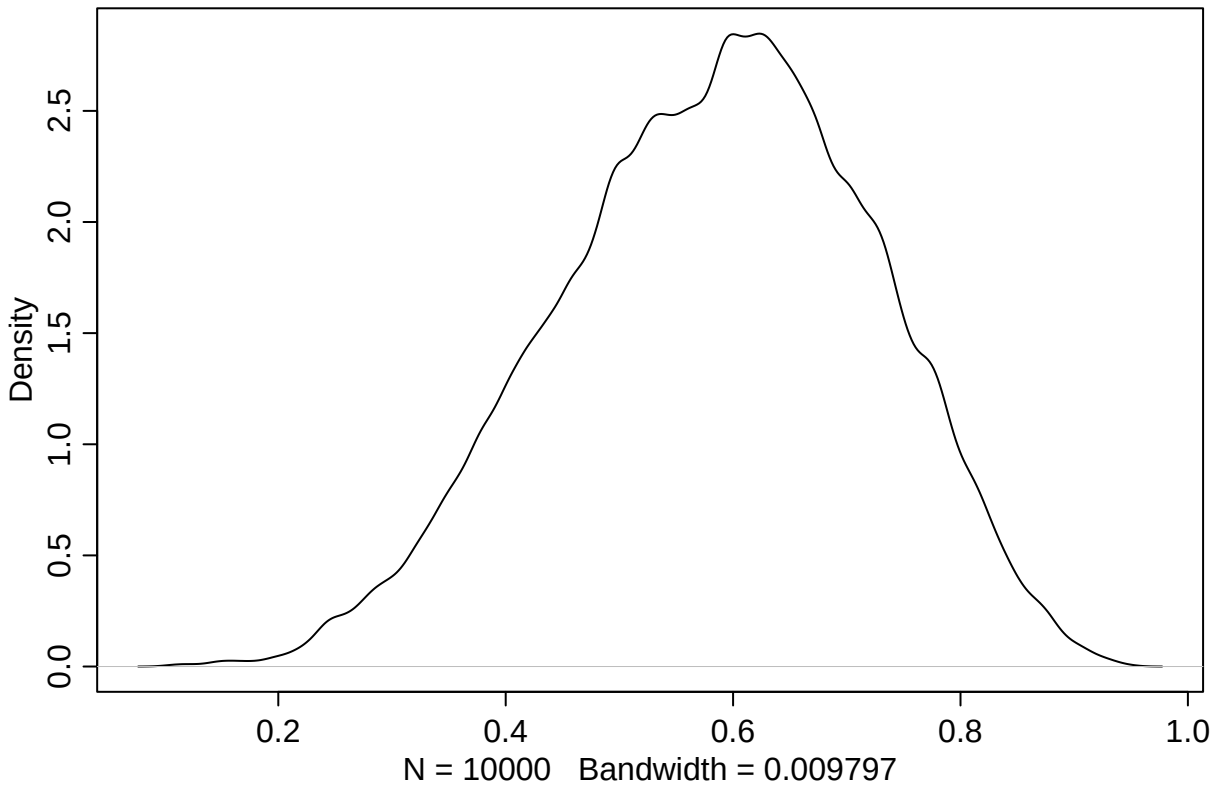

```
# ANSWER 1d)
# At equal cost (2x BinaxNOW vs 1x Competitor), using two BinaxNOW tests
# in series (both must be positive) DOMINATES the competitor at ALL base rates.
# The exceptionally high specificity (99.9975% vs 99.7%) from requiring two
# independent confirmations more than compensates for the reduced sensitivity
# (57% vs 82%). This strategy essentially eliminates false positives, making
# the true positive rate superior across the entire range of disease prevalence.
```

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.

```
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)
set.seed(712)
samples <- sample( p_grid , prob=posterior , size=1e4, replace=TRUE )
```

```
# First, let's visualize the distribution of `samples`:
dens(samples)
```



a)

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

```
# Answer to a):
sum(samples < 0.5) / length(samples)
```

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

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

```
# Answer to b):
sum(samples > 0.8) / length(samples)
```

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

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

```
# Answer to c):
sum( samples > 0.2 & samples < 0.8) / length(samples)
```

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

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

```
# Answer to d):
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.

```
# ANSWER:
# The form of the question is a conditional probability. Given that Smith was a
# swing voter in the previous election (swing1), what is the probability he is
# a swing voter in the next (swing2), that is:
#
```

$$P(Swing_2 \mid Swing_1) = \frac{P(Swing_2, Swing_1)}{P(Swing_1)}$$

```
# which is then answered by calculating the joint probability P(swing1, swing2)
# and the marginal probability P(swing1).

# The marginal probability P(swing1) is simply the probability of a voter in
# this equally-divided country being a swing voter, which is
#
p_twins <- 0.5*0.1 + 0.5*0.2
```

```
## [1] "P(swing1) = 0.15"
```

```
# The probability that a Blue voter is a swing voter in two successive elections is
# 0.2 * 0.2 = 0.04. The probability that a Red voter is a swing voter in two
# successive elections is 0.1 * 0.1 = 0.01. There is an equal chance that a voter
# is Red or Blue, so P(swing1, swing2) =
p_joint <- 0.5 * 0.04 + 0.5*0.01
```

```
## [1] "P(swing2, swing1) = 0.025"
```

```
# Finally, the conditional probability P(swing2 | swing1) is
```

```
## [1] "P(swing2 | swing1) = 0.166666666666667"
```

```
# Observe that P(swing2|swing1) > P(swing1).  Although we do not know which party
# Smith belongs to, learning that he was a swing voter in the last election
# provides some information about which party Smith belongs to, which is then
# factored into the estimate of the probability that he will be a swing voter
# in this election.
```

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.

```
# PRIOR: P(Blue) = P(Red) = P(Green) = 1/3
party_prior = 1/3

# The swing voter probabilities for each party are:
#
```

```r
# BLUE: P(Swing | Blue) = 1 - P(Vote Blue | member_of_Blue)
#                       = 1 - 0.80
#                       = 0.15 + 0.05
#                       = 0.20
swing_blue <- 0.20

# RED: P(Swing | Red)   = 0.10 + 0.05 = 0.15
swing_red <- 0.15

# GRN: P(Swing | Grean) = 0.20 + 0.10 = 0.30
swing_green <- 0.30

# MARGINAL PROB of SWING VOTERS in LAST ELECTION
swing1 <- swing_blue*party_prior + swing_red*party_prior + swing_green*party_prior
```

```
## [1] "P(swing1) = 0.216666666666667"
```

```r
# JOINT PROBABILITY
# The probability that a BLUE voter is a swing voter in two successive elections
# is P(swing2_blue, swing1_blue) = P(swing_blue) * P(swing_blue)
#                                = (0.20 * 0.20)
#                                = swing_blue^2
#
# The joint for a RED voter or a GREEN voter being swing voters in two successive
# elections is calculated in the same way.
#
# CONDITIONAL PROBABILITY
# To answer the question, you need to compute the conditional probability of
# each type of swing voter in the last election being a swing voter in the next.
#
# For instance, for BLUE voters:
# P(swing2_blue | swing1_blue) = P(swing2_blue, swing1_blue) * P(blue)
#                              = (0.20 * 0.20) * 1/3
#
# The conditional probabilities for RED and GREEN swing voters are calcuated
# in the same way.  The conditional probability P(swing2 | swing1) then is
# the sum of the conditional probabilities for BLUE, RED and GREEN:
#
swing2_swing1 <- swing_blue^2 * 1/3 + swing_red^2 * 1/3 + swing_green^2 * 1/3
```

```
## [1] "P(swing2, swing1) = 0.0508333333333333"
```

```r
# Finally, the conditional probability P(swing2 | swing1) is
```

```
## [1] "P(swing2 | swing1) = 0.234615384615385"
```

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?

5. AI Declaration

Please declare your collaborators in the class and how you used AI (if at all) to complete this assignment. If you used AI, include the prompts you used and explain what you learned from its responses that you didn't understand initially.