Assignment 2

Kerem Karagöz Immanuel Klein

General Information

- **Points**: Assignment 2 comprises of 6 tasks, 2 points each (12 in total). 2 points are obtained for complete and correct answers. 1 point is obtained for a proper approach or if only part of the task is solved.
- **Submission**: Hand in the assignment as a `Markdown` report (RMarkdown or Quarto) rendered as PDF. The PDF report should show the result(s), the code that produced the result(s), and possibly additional text or comment. Also indicate your name. The report should be uploaded on Moodle until Wednesday, June 5, 9:45 am.
- **Working in teams**: Everyone needs to hand in a report on Moodle. However, the report can be handed in as a team work (max. 2 people). When working in teams, state at the beginning of the document, who you worked with. It Ideally, teams use GitHub and add a link to the GitHub repository to which both contributed.
- **Code**: To automate code wrapping (such that long code lines are not cut off), install the formatR package and add the following code chunk at the beginning of the document:

```
knitr::opts_chunk$set(tidy = TRUE, tidy.opts=list(width.cutoff=50))
```

```r
# load packages here
library(dplyr)
library(ggplot2)
library(tinytex)
```

Task Set 1

For tasks 1.1-1.2, suppose there are 3 companies, Company A to C. Company A has a customer satisfaction rate of .70, Company B of .50, and Company C of .80. Further suppose that you receive 10 customer reviews (6 positive, 4 negative) for the same company, but you don't know for which company. Assume that Company B is twice as likely to obtain reviews than Company A and C.

Task 1.1

Show that the posterior probability that Company A was rated is ≈ 0.29.

```r
# Searching for P(A|data)
# P(A|data)=(P(data|A)*P(A))/P(data) (Bayes)

# Prior probabilities e. g. P(A)
prior.A <- 0.25
prior.B <- 0.5
prior.C <- 0.25

# Satisfaction rates
satisfaction.A <- 0.7
satisfaction.B <- 0.5
satisfaction.C <- 0.8

# Likelihoods e. g. P(data|A)
likelihood.A <- dbinom(6, 10, satisfaction.A)
likelihood.B <- dbinom(6, 10, satisfaction.B)
likelihood.C <- dbinom(6, 10, satisfaction.C)

# Calculate P(data)
p.data <- (likelihood.A * prior.A) + (likelihood.B *
    prior.B) + (likelihood.C * prior.C)

# Calculate posterior P(A|data)
```

```r
posterior.A <- (likelihood.A * prior.A)/p.data
posterior.B <- (likelihood.B * prior.B)/p.data
posterior.C <- (likelihood.C * prior.C)/p.data

print(paste("P(A|data):", posterior.A))
```

```
[1] "P(A|data): 0.28655942724975"
```

Task 1.2

Suppose you receive 10 more reviews (9 positive and 1 negative). Show that the posterior probability that Company C received the reviews increases by ≈ 33 percentage points, when considering all 20 rather than only the first 10 reviews. To obtain the updated posterior, compute the likelihood of the 10 most recent reviews only.

```r
# Old posteriors become new priors
prior.A <- posterior.A
prior.B <- posterior.B
prior.C <- posterior.C

# Change likelihoods
likelihood.A <- dbinom(9, 10, satisfaction.A)
likelihood.B <- dbinom(9, 10, satisfaction.B)
likelihood.C <- dbinom(9, 10, satisfaction.C)

# Recalculate (new) P(data)
p.data <- (likelihood.A * prior.A) + (likelihood.B *
    prior.B) + (likelihood.C * prior.C)

# Update posterior P(C|data)
posterior.A <- (likelihood.A * prior.A)/p.data
posterior.B <- (likelihood.B * prior.B)/p.data
posterior.C <- (likelihood.C * prior.C)/p.data

# Calculate increase
increase <- posterior.C - prior.C

print(paste("P(C|data):", posterior.C))
```

```
[1] "P(C|data): 0.455776000537989"
```

```r
print(paste("Increase:", increase))
```

```
[1] "Increase: 0.329650952075531"
```

Task Set 2

For tasks 2.1 and 2.2, suppose there are Factory A and Factory B, producing the same product. The company C receives equally many shipments from both factories. Even though the machines, processes, and standards are virtually identical, the factories differ in their defect rates. Shipments from Factory A entail defective products 10% of the time, shipments from Factory B entail defective products 20% of the time.

Task 2.1

You receive a shipment from one of the factories, and upon inspection, you find that the shipment contains defective products. Compute the probability that the next shipment from this company will also contain defective products.

```
# Defect rates for Factory A and Factory B
defect.A <- 0.1
defect.B <- 0.2

# Prior Probabilities (They are equally likely at
# the first state)
prior.A <- 0.5
prior.B <- 0.5

# Calculate the likelihood
likelihood.defect.A <- defect.A
likelihood.defect.B <- defect.B

# Calculate the posterior probabilities with the
# help of normalization constant
posterior.A.unnormalized <- prior.A * likelihood.defect.A
posterior.B.unnormalized <- prior.B * likelihood.defect.B

constant <- posterior.A.unnormalized + posterior.B.unnormalized

posterior.A <- posterior.A.unnormalized/constant
posterior.B <- posterior.B.unnormalized/constant

# Probability that the next shipment contains
# defective products
p.next.defect <- posterior.A * likelihood.defect.A +
    posterior.B * likelihood.defect.B
```

```r
print(paste("P Next Defect:", p.next.defect))
```

```
[1] "P Next Defect: 0.166666666666667"
```

Task 2.2

Suppose the R&D department came up with a Machine Learning algorithm that (imperfectly) identifies the factory based on the shipped products. But the classification algorithm is imperfect. This is the information you have about the algorithm:

- The probability it correctly identifies a Factory A product is 93%.
- The probability it correctly identifies a Factory B product is 87%.

When applying the the algorithm to the shipped products, the test is positive for Factory A. Including the defect data from 2.1, compute the posterior probability that your shipment is from Company A.

```r
# Accuracies of the ML algorithms
accuracy.A <- 0.93
accuracy.B <- 0.87

# Probability of a positive test result for
# Factory A and Factory B
p.positive.A <- accuracy.A
p.positive.B <- 1 - accuracy.B

# Calculate the posterior probabilities given the
# test result using prior posteriors from Task 1
posterior.A.test.unnormalized <- posterior.A * p.positive.A
posterior.B.test.unnormalized <- posterior.B * p.positive.B

constant.test <- posterior.A.test.unnormalized + posterior.B.test.unnormalized

posterior.A.test <- posterior.A.test.unnormalized/constant.test

# Print the updated posterior probability for
# Factory A
print(paste("Posterior Prob That Shipment is Coming from Company A:",
    posterior.A.test))
```

```
[1] "Posterior Prob That Shipment is Coming from Company A: 0.781512605042017"
```

Task Set 3

For Task 3.1 and 3.2, suppose, one last time, you want to estimate the proportions of land on the earth's.

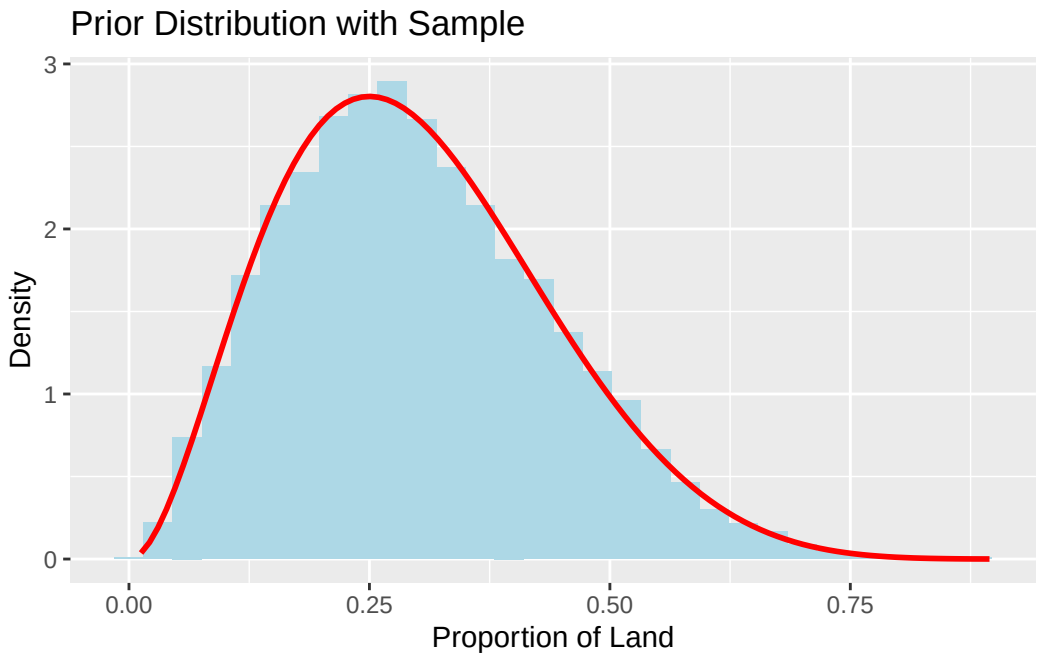
Task 3.1

Specify a prior distribution and store 10,000 random samples from it in a vector `sample`. Plot the prior distribution and briefly explain your choice of the prior.

```
# Generate 10,000 random samples from the Beta
# distribution Beta because it is easy to
# 'customize' We have a prior believe that the
# proportion of land is around 0.3.  Also, the
# distribution should allow for some variability
# when updating -> Right skewed distribution with
# mean of 0.3 -> beta with alpha=3 and beta=7
set.seed(123)  # for reproducibility
alpha <- 3
beta <- 7
sample <- rbeta(10000, alpha, beta)

# Plot the prior and the sample
samples.df <- data.frame(sample)

ggplot(samples.df, aes(x = sample)) + geom_histogram(aes(y = after_stat(density)),
    bins = 30, fill = "lightblue") + stat_function(fun = dbeta,
    args = c(3, 7), color = "red", linewidth = 1) +
    labs(title = "Prior Distribution with Sample",
        x = "Proportion of Land", y = "Density")
```



Task 3.2

Run the following code chunk that uses your object sample to obtain prior probabilities for the possible proportions of land 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 that approximate your prior distribution.

```
prop <- seq(0, 1, length.out = 12)
priors <- vector("numeric", length(prop))
for (i in seq_along(prop)) {
    priors[i] <- round(sum(sample >= prop[i] & sample <
        prop[i + 1])/10000, 2)
}
poss <- tibble(prop_L = seq(0, 1, 0.1), prior = priors[1:11])
```

Use these priors to compute the posterior probability after observing 26 times land in 100 globe tosses. Take 1,000 samples from the posterior distribution and with each sample, predict the outcome of 100 globe tosses. Plot the posterior predictions in a histogram.

```
# Calculate likelihoods for each proportion
likelihoods <- dbinom(26, 100, poss$prop_L)

# Bayes rule
```

```r
posterior.probabilities <- (likelihoods * poss$prior)/sum(likelihoods *
    poss$prior)

set.seed(123)  # for reproducibility
# Take samples from posterior distribution
posterior.sample <- sample(poss$prop_L, size = 1000,
    replace = TRUE, posterior.probabilities)
# Make prediction for each sample
predictions <- rbinom(1000, 100, posterior.sample)

# Plot histogram
predictions.df <- data.frame(predictions)

ggplot(predictions.df, aes(x = predictions)) + geom_histogram(bins = 30) +
    labs(title = "Posterior Predictions for 100 Globe Tosses",
        x = "Number of Land Outcomes in 100 Tosses",
        y = "Frequency") + xlim(0, 100)
```