Homework 3

Patrick Addona

February 7, 2025

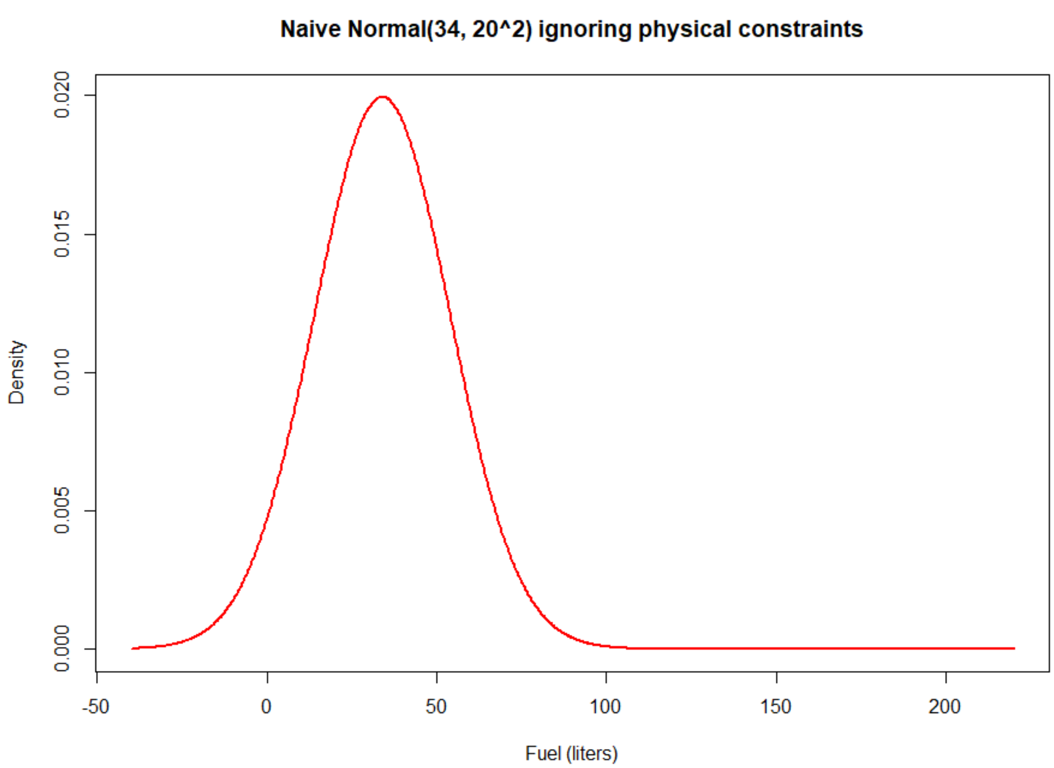
Question 1

Draw the probability density function for the usable fuel in the tank without any other information besides the fuel gauge reading. Determine the expected value of the available fuel, the most likely value of available fuel, and probability of negative fuel in the tank. Do these estimates make sense to you?

By the information we are given from the fuel gauge reading, we understand the PDF for the usable fuel in the tank to be a Gaussian distribution centered at 34 liters, with a standard deviation of 20 liters. In other words, if X is the usable fuel in the tank,

$$X \sim \mathcal{N}(34, 20^2)$$

Visually, this can be represented with the well-known function for the PDF of a Gaussian:



As the expected value of a Gaussian is simply μ,

$$E[X] = 34$$

Likewise, the most likely value of available fuel is the mode of the PDF at 34 liters.

To determine the probability of negative fuel in the tank, we can calculate

$$\int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}20} e^{-\frac{(x-34)^2}{2\cdot 20^2}}\, dx$$

$$= 0.0446 = 4.46\%$$

As far as if this makes sense, the expected value and most likely value both do and do not. If the fuel sensor reads 34, we should hope for the true value to be near this, especially since the value is normally distributed. On the other hand, the standard deviation is so large that the truth could be anywhere from -26 to 94 (a range of three standard deviations). The negative values, on the other hand, do not make any physical sense as the tank cannot have less than 0 liters of fuel and indicates potential problems with our prior.

Question 2

How can you use a proper prior to address the issue of unrealistic fuel estimates in the tank? Define this physically based prior for you (meaning this is your subjective prior).

To address the issue of unrealistic fuel estimates in the tank, we should construct our prior to reflect what we know is physically possible. I believe, before we look at the fuel gauge, we simply know that the fuel can be anywhere from 0 to 182 liters. Thus, we can adopt a uniform distribution for X:

$$X \sim \mathcal{U}[0, 182]$$

and therefore

$$p(X) = \frac{1}{182}, \quad 0 \leq X \leq 182$$

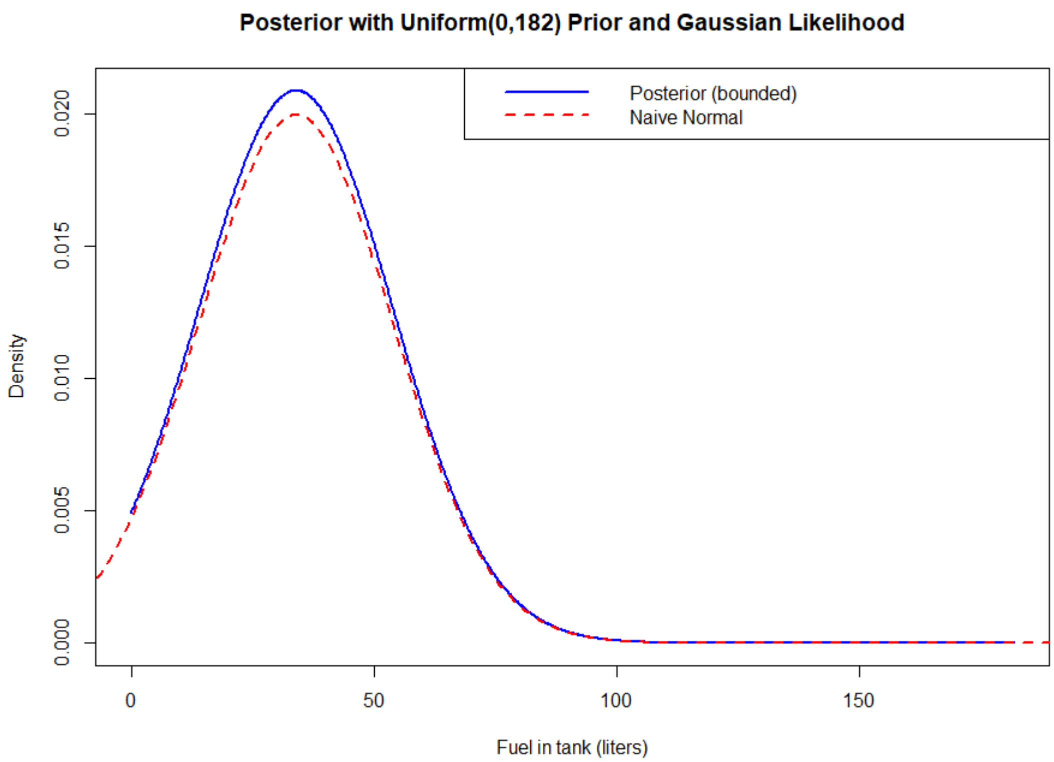
while being 0 otherwise. My reasons for making this choice were three-fold. One, we do not have oracle knowledge of the fuel in the tank before we make an observation of the gauge. Two, adopting a uniform prior simplifies calculations. Three, other distributions would require much justification and assumptions that are less grounded in the scenario we are considering.

However, it should be noted that this does neglect the uncertainties of other priors. Depending on the prior, our Bayesian updates may produce significantly different posteriors. This prior choice is a source of deep uncertainty, and should not be held as binding to the only way of solving the problem.

Question 3

Use a grid-based method to determine your Bayesian update from your prior and the likelihood function. Add this posterior to the plot produced above. Determine now the probability of negative fuel. Has this fixed the issue? If so, how?

To implement this method, I defined a uniformly-spaced grid of 1001 points from 0 to 182. Our prior is a vector of the same probability $\frac{1}{182}$ for each grid point, while the likelihood is a Gaussian with standard deviation of 20 and mean equal to each grid point, compared to the observed value of 34 liters. After multiplying these two distributions, I normalized by dividing the posterior by the sum of the posterior values times the grid spacing. The graph now shows as below:

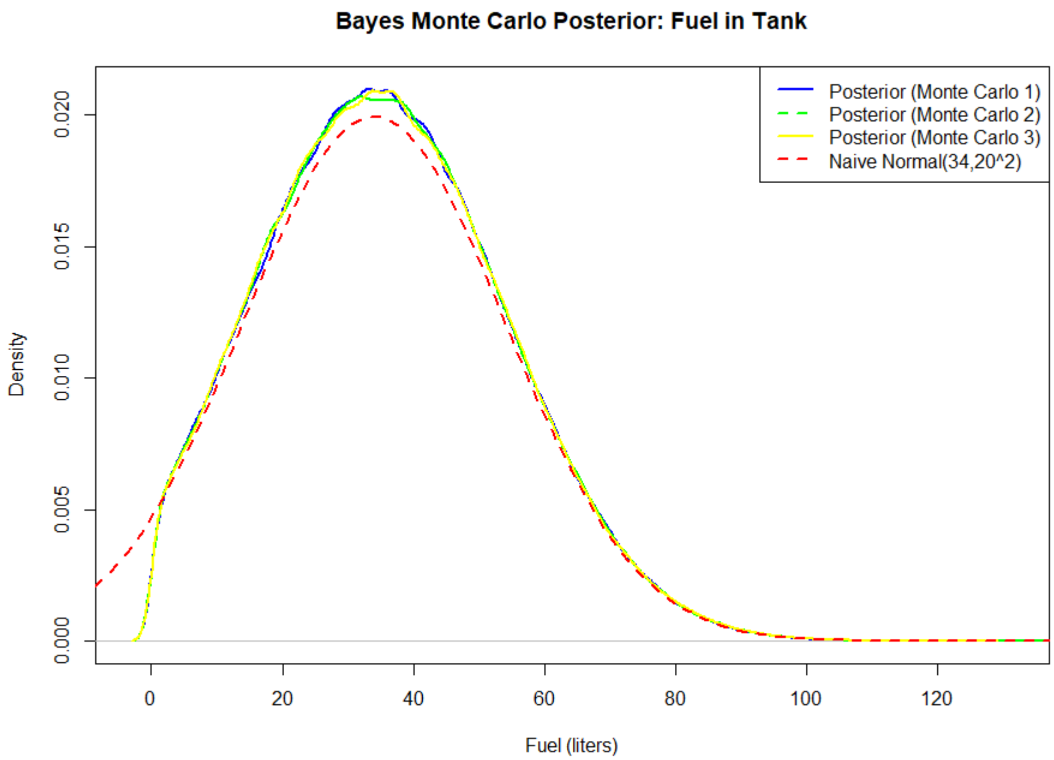


Notice now that by incorporating the uniform prior from 0 to 182 with 0 probability outside of this range, the posterior no longer has support outside $[0, 182]$. As such, the probability of negative fuel is now 0.

Question 4

Repeat the step above using a Bayes Monte Carlo method

To implement a Bayes Monte Carlo method to approximate the posterior distribution, I used a resource as inspiration: the Github documentation of data scientist/engineer David Salazar. This approach of importance sampling requires that we generate a number of random samples (here, 1,000,000) and compute this many samples of the prior, which in this case is normal from 0 to 182. Then, after also create a vector of the likelihood values, the weights of each Monte Carlo sample are calculated as the likelihood values (scaled by the probability of each prior value, which ends up being normalized away). After normalizing, we generate 1,000,000 samples from the possible prior values with replacement, each weighted according to the normalized weights. This has the effect of approximating the posterior, as is evident in the graph.



I chose to generate 1,000,000 samples to ensure proper representation of the posterior distribution. Too few may not resemble to the posterior, and too many would be computationally inefficient. This analysis is reproducible by using the specific seeds (1, 2, and 3) as shown in the code.

David Salazar's Github: https://david-salazar.github.io/2020/06/27/bayesian-data-analysis-week-4-importance-sampling/

Question 5

What assumptions would you need to make for a simple analytical Kalman filter solution to this problem? Are these assumptions realistic?

For a simple analytical Kalman filter solution, we would have to first assume that the state dynamics are linear, as we predict fuel at time $t + 1$ is equal to the fuel at time t plus some Gaussian amount of noise. Similarly, we assume that the readings are modeled as the true fuel plus some Gaussian error. Finally, we would run into the same problem as the naive prior since we could potentially generate negative fuel estimates.

The first two assumptions may not be safe to make if the fuel usage does not change linearly due to conditions like weather, altitude, engine state, etc. As discussed before, the final assumption is also not realistic since we can never have negative fuel; in this case, a simple analytical Kalman filter approach may not be the best one.

Question 6

Produce a plot of the estimated available flight time. Use this plot to address these questions:
a. What is the probability that you make an airport that is 100 minutes flight time away with at least 30 min reserve fuel required by regulations?
b. What is the probability that you run out of fuel trying to make it to this airport?

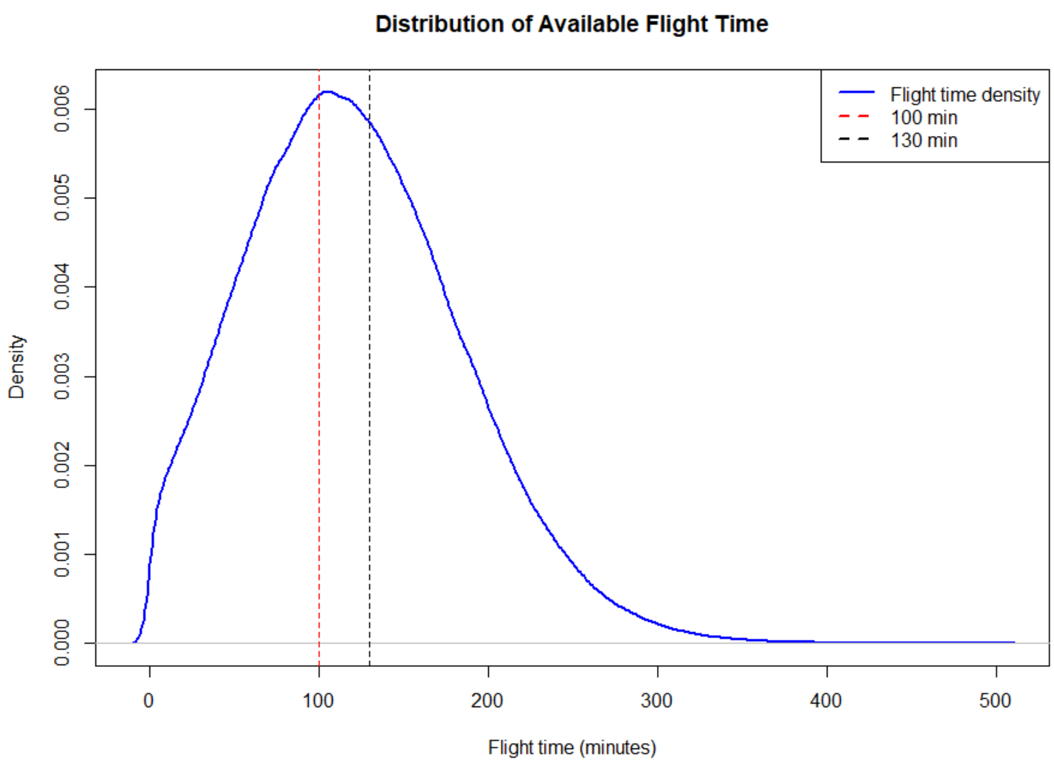
To solve the problem of available flight time, I continued the previous BMC work by constructing the PDF of the fuel rate:

$$F \sim \mathcal{N}(18, 2^2)$$

and constructing the flight time in minutes as

$$T = 60 \cdot \frac{X_{post}}{F}$$

Then, all that was needed was to plot the density of T:



The lines represent the regions of the PDF corresponding to 100 minutes (if we run out of fuel) and 130 minutes (if we make it to the airport with 30 minute fuel reserves).

a. The probability that we make it to the airport with 30 minutes of fuel to spare is given by $P(T \geq 130)$, and to find this, I discovered that R's "mean()" function calculates the probability of a PDF defined by a density vector by creating a logical array and taking the average of it. This is an interesting and intuitive technique, and I learned about it from a commenter on a Stack Overflow post: https://stackoverflow.com/questions/69544032/can-mean-function-show-probability-of-cumulative-distribution-function. Suffice to say, this probability is around 42.11%.

b. In a similar approach, the probability that we run out of fuel before the airport is around 39.59%.

Appendix

Problem 1:

```r
##  author: Patrick Addona
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)

x_naive  <- seq(-40, 220, length.out = 1001)
naive_pdf <- dnorm(x_naive, mean = 34, sd = 20)

plot(x_naive, naive_pdf,
     type = "l", lwd = 2, col = "red",
     xlab = "Fuel (liters)",
     ylab = "Density",
     main = "Naive Normal(34, 20^2) ignoring physical constraints")
```

Problem 3:

```r
##  author: Patrick Addona
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)

x_naive  <- seq(-40, 220, length.out = 1001)
X_grid <- seq(0, 182, length.out = 1001) # 1001 points from 0 to 182
dx <- X_grid[2] - X_grid[1] # spacing between grid points

prior  <- rep(1/182, length(X_grid)) # uniform prior

likelihood <- dnorm(34, mean = X_grid, sd = 20) # likelihood function

posterior_unnorm <- prior * likelihood # before normalization
normalizing_const <- sum(posterior_unnorm) * dx
posterior <- posterior_unnorm / normalizing_const

# plot posterior vs. naive normal
plot(X_grid, posterior, type = "l", lwd = 2, col = "blue",
     xlab = "Fuel in tank (liters)", ylab = "Density",
     main = "Posterior with Uniform(0,182) Prior and Gaussian Likelihood")

# naive normal from before
lines(x_naive, naive_pdf, col = "red", lty = 2, lwd = 2)

legend("topright", legend = c("Posterior (bounded)", "Naive Normal"),
       col = c("blue", "red"), lty = c(1, 2), lwd = 2)
```

Problem 4:

```r
##  author: Patrick Addona
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)

set.seed(1)  # for reproducibility

N <- 1e6  # 1,000,000 monte carlo samples
X_prior <- runif(N, min = 0, max = 182)
likelihood <- dnorm(34, mean = X_prior, sd = 20)

weights_unnorm <- likelihood * 1/182 # multiply by constant 1/182 since prior is uniform
weights_norm <- weights_unnorm / sum(weights_unnorm) # normalize weights

#importance sampling using normalized weights
idx <- sample(seq_len(N), size = N, replace = TRUE, prob = weights_norm)
X_posterior <- X_prior[idx]

plot(density(X_posterior),
     main = "Bayes Monte Carlo Posterior: Fuel in Tank",
     xlab = "Fuel (liters)",
```

```r
     lwd  = 2,
     col  = "blue")

set.seed(2)  # for reproducibility

N <- 1e6  # 1,000,000 monte carlo samples
X_prior <- runif(N, min = 0, max = 182)
likelihood <- dnorm(34, mean = X_prior, sd = 20)

weights_unnorm <- likelihood * 1/182 # multiply by constant 1/182 since prior is uniform
weights_norm <- weights_unnorm / sum(weights_unnorm) # normalize weights

idx <- sample(seq_len(N), size = N, replace = TRUE, prob = weights_norm)
X_posterior <- X_prior[idx]

lines(density(X_posterior),
      main = "Bayes Monte Carlo Posterior: Fuel in Tank",
      xlab = "Fuel (liters)",
      lwd  = 2,
      col  = "green")

set.seed(3)  # for reproducibility

N <- 1e6  # 1,000,000 monte carlo samples
X_prior <- runif(N, min = 0, max = 182)
likelihood <- dnorm(34, mean = X_prior, sd = 20)

weights_unnorm <- likelihood * 1/182 # multiply by constant 1/182 since prior is uniform
weights_norm <- weights_unnorm / sum(weights_unnorm) # normalize weights

idx <- sample(seq_len(N), size = N, replace = TRUE, prob = weights_norm)
X_posterior <- X_prior[idx]

lines(density(X_posterior),
      main = "Bayes Monte Carlo Posterior: Fuel in Tank",
      xlab = "Fuel (liters)",
      lwd  = 2,
      col  = "yellow")

curve(dnorm(x, mean=34, sd=20), from=-40, to=220,
      add=TRUE, col="red", lwd=2, lty=2)

legend("topright",
       legend=c("Posterior (Monte Carlo 1)", "Posterior (Monte Carlo 2)", "Posterior (Monte Carlo 3)", "Naive Normal(34,20^2)"),
       col=c("blue", "green", "yellow", "red"), lty=c(1,2), lwd=2)
```

Problem 6:

```r
##  author: Patrick Addona
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)

set.seed(123)  # for reproducibility

N <- 1e6  # 1,000,000 monte carlo samples
X_prior <- runif(N, min = 0, max = 182)
likelihood <- dnorm(34, mean = X_prior, sd = 20)

weights_unnorm <- likelihood * 1/182 # multiply by constant 1/182 since prior is uniform
weights_norm <- weights_unnorm / sum(weights_unnorm) # normalize weights

#importance sampling using normalized weights
idx <- sample(seq_len(N), size = N, replace = TRUE, prob = weights_norm)
X_posterior <- X_prior[idx]

fuel_rate <- rnorm(N, mean = 18, sd = 2) # fuel rate pdf

time   <- 60 * X_posterior / fuel_rate # flight time in minutes

plot(density(time),
```

```
     xlab = "Flight time (minutes)",
     main = "Distribution of Available Flight Time",
     lwd  = 2, col  = "blue")

abline(v = 100, col = "red", lty = 2)
abline(v = 130, col = "black", lty = 2)

legend("topright",
       legend=c("Flight time density","100 min","130 min"),
       col=c("blue","red","black"), lty=c(1,2,2), lwd=2)


# a) probability of having enough flight time to go 100 min + 30 min reserve:
# P(time >= 130)
p_enough_reserve <- mean(time >= 130) #calculates logical array and finds probability
#by calculating mean

# b) Probability that you run out of fuel trying to make it to an airport
#    that is 100 min away: P(time < 100)
p_run_out <- mean(time < 100)

cat("Probability( time >= 130 ) =", p_enough_reserve, "\n")
cat("Probability( time < 100 )   =", p_run_out, "\n")
```