Maggie O'Shea
Bayesian Statistical Modeling and Computation
Professor Klaus Keller
Problem Set 3
January 24, 2025

1. **Review the key sources already assigned as reading with a special focus on:**

a) Qian, S. S., Stow, C. A., & Borsuk, M. E. (2003). On Monte Carlo methods for Bayesian inference. Ecological Modelling, 159(2-3), 269–277.
b) D'Agostini, G. (2003). Bayesian reasoning in data analysis: A critical introduction. Singapore: World Scientific Publishing. (Chapter 6 only).
c) Ruckert, K. L., Guan, Y., Bakker, A. M. R., Forest, C. E., & Keller, K. (2017). The effects of time-varying observation errors on semi-empirical sea-level projections. Climatic Change, 140(3-4), 349–360. https://doi.org/10.1007/s10584-016-1858-z

Also reviewed:

d) Kim, Y., Bang, H., Kim, Y. & Bang, H. Introduction to Kalman Filter and Its Applications. in Introduction and Implementations of the Kalman Filter (IntechOpen, 2018). doi:10.5772/intechopen.80600.

**2. 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 available fuel, the most likely value of available fuel, and probability of negative fuel in the tank. Do these estimates make sense to you? (1 point)**

The expected value of available fuel is 33.99 . According to the PDF (Figure 1), there is a 0.045 probability of negative fuel in the tank. The most probable value is 32.288 which is the value with the highest probability. Given the initial gauge reading of 34, the expected value and most likely value of available fuel are possible and make sense. However, it does not make sense that there is a non-zero possibility that there could be negative fuel.

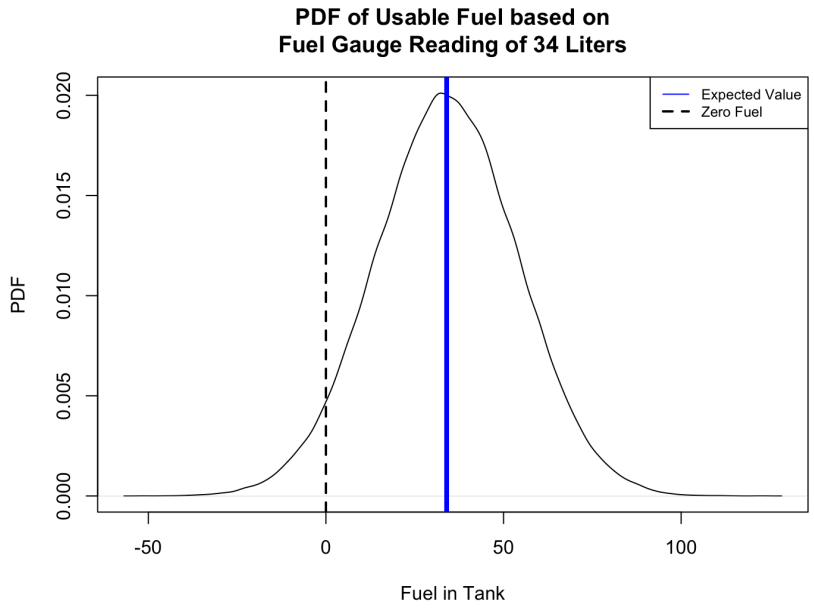


Figure 1: Probability Density Function (PDF) of Usable Fuel based on Fuel Gauge Reading of 34 Liters. The expected value is indicated in blue and the dashed line is at the point of zero fuel.

In terms of uncertainties, in this case the PDF was made by randomly sampling a normal distribution and then calculating the PDF - because I used just one seed there is uncertainty around the seed and if results would differ with differing seeds. Without performing the analysis it isn't clear how this would impact the conclusions, but it might result in a slightly different PDF, as well as help quantify uncertainty around the most probable value and expected value. The choices made here included estimating that the fuel in the tank was gaussian around the mean (observed) 34 liters with a standard deviation of 20, as well as a selected approach to making the PDF through sampling and then estimating the PDF. This is a defensible choice as it is based on observed data (the reading) as well as knowledge about the uncertainty in the sensor reading. This analysis was made reproducible by setting a seed as well as by including comments explaining the steps.

**3. 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). (3 points)**

A new physically based prior is a uniform distribution from 0 to 182. This prior can be used to address the unrealistic fuel estimates in the tank because it sets bounds around the posterior such that they won't surpass the minimum/maximum fuel in the tank. Uncertainty would be relevant here because it's my subjective prior, which can introduce deep uncertainty, and perhaps also if there is any uncertainty around the exact size of the tank.

**4. 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?**

The probability of negative fuel is zero (Figure 2, Figure 3). This has fixed the issue because the prior was a uniform distribution from 0 to 182 such that when combined with the observed fuel, the resulting posterior was a truncated normal distribution limited by the limits of the uniform distribution, ie cut off at 0. The resulting CDF and PDF of the posterior distribution is shown in Figure 2 and 3.

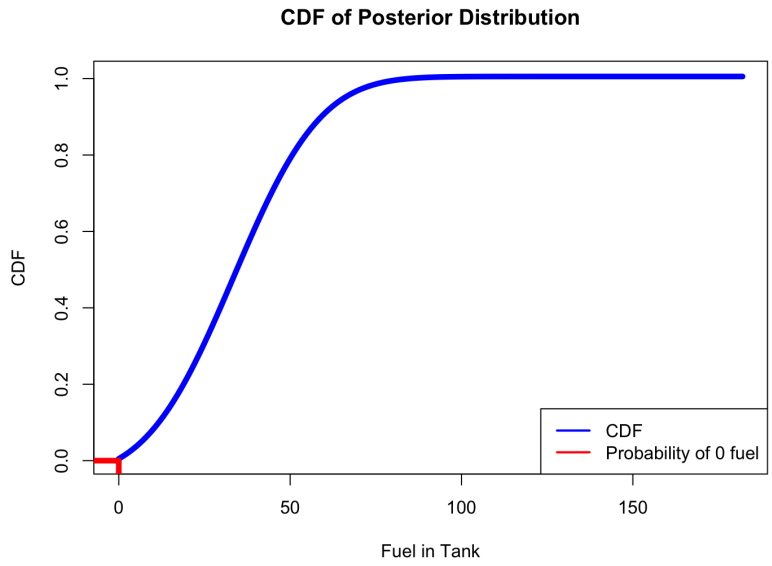


Figure 2: Cumulative Distribution Function (CDF) of Posterior Distribution estimated with physically informed prior based on fuel capacity of tank. Red lines point to the probability of 0 fuel (0).

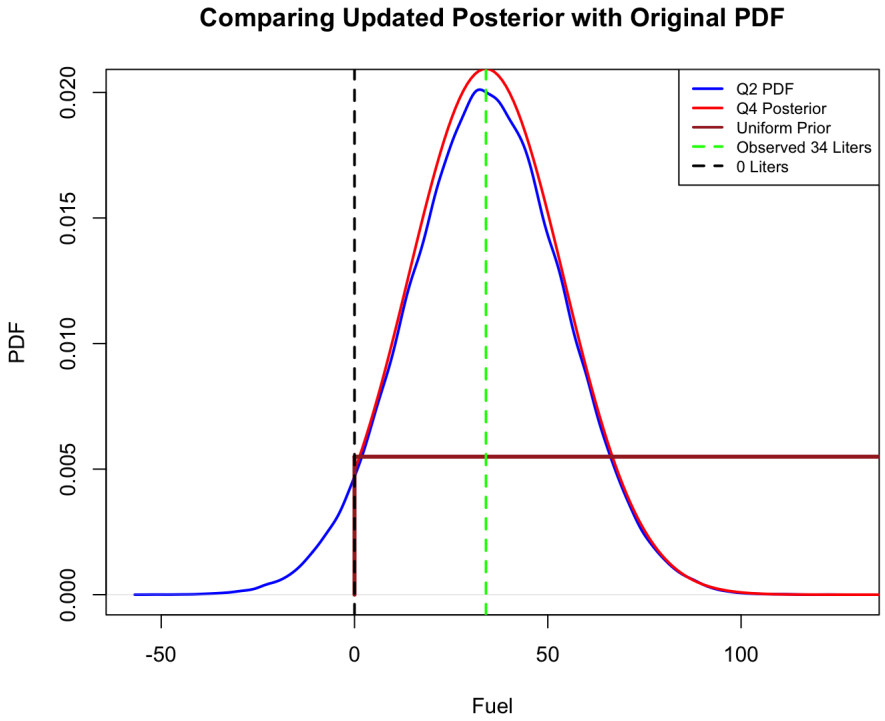


Figure 3: Comparing Updated Posterior with Original PDF from Q2. Brown line indicates uniform prior. Green dashed lines indicates the observed 34 liters. Black Dashed line indicates 0 liters.

In this case there are a few sources of uncertainty and choices that I made. First, the grid values used in the estimation of the posterior were 182 values from 0 to 182 - it's possible that if I used a different number of values ie more or less grid values. This could introduce a type of uncertainty from the data used. In this case, I think the choice that I made using182 values is defensible as 182 allows for all whole number of liters to be represented in the dataset without using so many grid values that the computational processing time is slowed. There is also prior uncertainty and model structure uncertainty because I did not try multiple models or priors to show the range of possible outcomes. IThe prior selected is my subjective prior, however, it is defensible in that it reflects the true capacity of the tank from 0 to 182 liters. Finally, this analysis was made reproducible by commenting the code to explain each piece of it such that one could follow along.

**5. Repeat the step above using a Bayes Monte Carlo method**

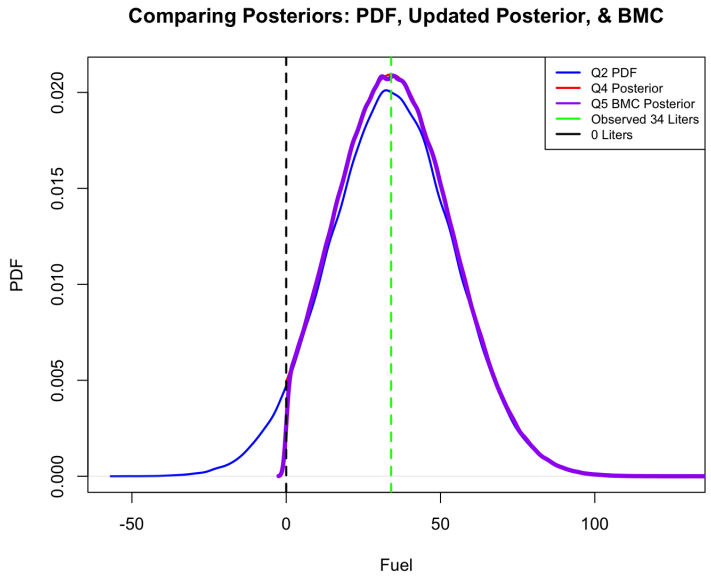


Figure 4: Comparing Results from Q4 Bayesian Posterior, and Bayes Monte Carlo Posterior with original PDF for fuel in the tank. Green dashed line indicates the observed 34 liters and black dashed line at 0 liters.

The steps above were repeated using Bayes Monte Carlo (BMC) method (Figure 4). Again the probability of negative fuel is 0. In this analysis, compared to that of Q4, there are additional sources of uncertainty including but beyond those expressed in Q4. The new sources of uncertainty primarily come from the random sampling required to perform the BMC approach particularly because we randomly sample the prior to start and then randomly sample the posterior as well. This introduces a source of uncertainty around the chosen seed (in this case 930) to perform the analysis. While it's not possible to know completely how this would impact the results if I vary the seeds, it likely would show variability around the resulting expected values or most probable values. Another choice that was made that could introduce uncertainty is the number of trials or samples taken for the random sampling. 3,000,000 samples were used to address the concern with having too few samples leading to biased results. This sample size was selected based on the passing of a convergence test.  I began with 10,000 samples and tested if the result passed the convergence test based on the credible interval width, and increased the sample size until this test was passed. The results around the expected value did not vary widely even between the 10,000 sample BMC posterior and 3,000,000 expected value, however it took that many samples for convergence. Finally, this was made reproducible, again, by commenting the code as well as setting a seed to reproduce the random sampling results.

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

The Kalman filter requires that the process its modeling is linear, as well as that the data's noise is gaussian. In this case, I think it is realistic that this is a linear problem, as well as the fact that the data is normally distributed given the one observation of the fuel in the tank. Without that observation, however, the data is uniformly distributed where it is equally likely to have 34 liters as it is to have 59 liters. In this case, this compromises the ability to use the Kalman filter solution here.

**7. Produce a plot of the estimated available flight time. Use this plot to address these**
**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?**

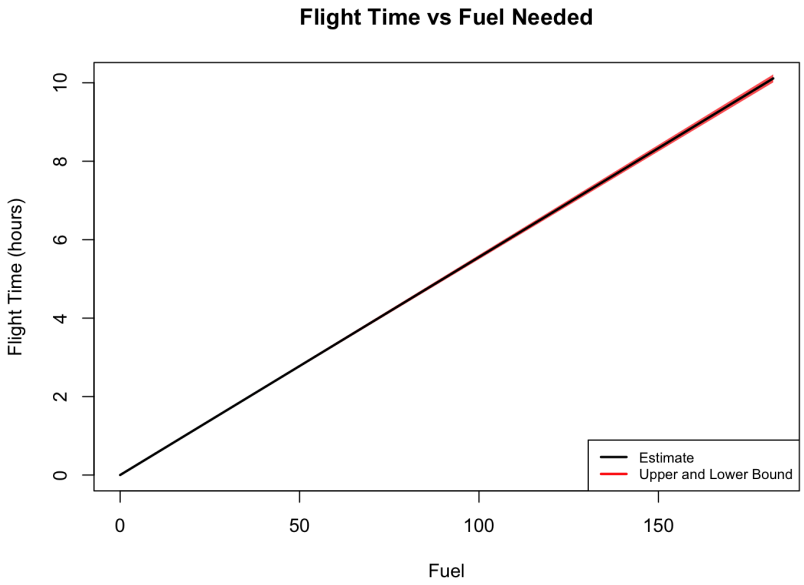


Figure 5: Flight time and necessary fuel with uncertainty bounds. Flight time based on 18 liters per hour estimate of needed fuel, and uncertainty based on 2 liters per hour standard deviation.

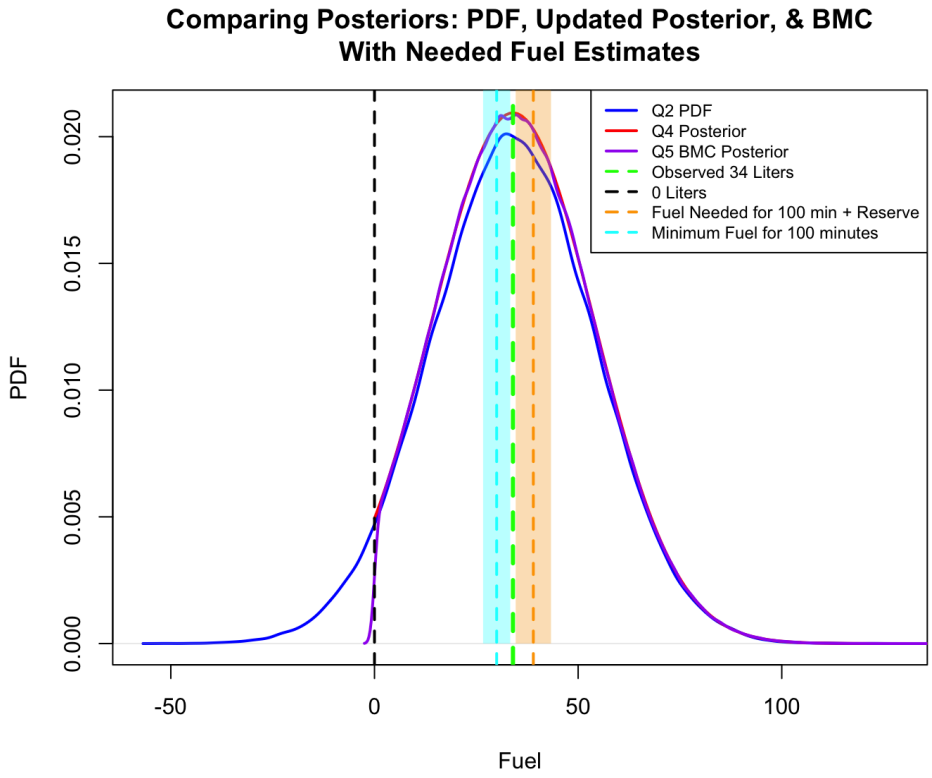


Figure 6: Examining the posteriors relative to the needed fuel for 100 minute flight time with 30 minute reserve (orange) and minimum fuel needed for just 100 minute flight time. Shaded orange and yellow represent uncertainty around the estimated probability of enough fuel.

a) The probability of having enough fuel to make it to the airport with a 100 minute flight time and 30 minutes of fuel reserve is 0.59. However there are uncertainties in this estimate given that there are uncertainties around the fuel needed per hour. Given this, the range of probability of having enough fuel to do this with the 30 minute reserve is 0.49 to 0.66. This result was made reproducible by, again, commenting the code thoroughly.
b) The probability of running out of fuel to make it to this airport 100 minutes away is 0.40, however, again there is uncertainty in this estimate due to uncertainties around the amount of fuel needed per hour. The range of probabilities of running out of fuel range from 0.33 to 0.48. Again, in this case this result was made reproducible by commenting the code thoroughly.

**Works Cited:**


Qian, S. S., Stow, C. A., & Borsuk, M. E. (2003). On Monte Carlo methods for Bayesian inference. Ecological Modelling, 159(2-3), 269–277.

D'Agostini, G. (2003). Bayesian reasoning in data analysis: A critical introduction. Singapore: World Scientific Publishing. (Chapter 6 only).

Ruckert, K. L., Guan, Y., Bakker, A. M. R., Forest, C. E., & Keller, K. (2017). The effects of time-varying observation errors on semi-empirical sea-level projections. Climatic Change, 140(3-4), 349–360. https://doi.org/10.1007/s10584-016-1858-z

Kim, Y., Bang, H., Kim, Y. & Bang, H. Introduction to Kalman Filter and Its Applications. in Introduction and Implementations of the Kalman Filter (IntechOpen, 2018). doi:10.5772/intechopen.80600.

***Appendix I:***

Code written in software R (Version 2024.12.0+467).

- Downloaded from: https://www.r-project.org/

Saved to Github Repository:
https://github.com/maggieoshea/BayesianStatisticalModelingandComputation/tree/main/problemset3

Full R Script below.

```
##########################################
##  file: margaret.oshea.gr@dartmouth.edu_PS3.R
## Written on R Version 2024.12.0+467
#########################################################
##  Maggie O'Shea
##  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)
#########################################################
##   Course: Bayesian Statistical Modeling & Computation
##   Professor Klaus Keller
##   February 14, 2025
##   Problem Set #3
#################################################
# contact: margaret.oshea.gr@dartmouth.edu
#################################################
# sources:
# Density plots in R: https://www.geeksforgeeks.org/histograms-and-
density-plots-in-r/
# Integration in R: https://stackoverflow.com/questions/40851328/
compute-area-under-density-estimation-curve-i-e-probability
# R 'norm' funtions: https://seankross.com/notes/dpqr/
# Qian, S. S., Stow, C. A., & Borsuk, M. E. (2003). On Monte Carlo
methods for Bayesian inference. Ecological Modelling, 159(2-3), 269–
277.
# D'Agostini, G. (2003). Bayesian reasoning in data analysis: A
critical introduction. Singapore: World Scientific Publishing.
(Chapter 6 only).
# Ruckert, K. L., Guan, Y., Bakker, A. M. R., Forest, C. E., & Keller,
K. (2017). The effects of time-varying observation errors on semi-
empirical sea-level projections. Climatic Change, 140(3–4), 349–360.
https://doi.org/10.1007/s10584-016-1858-z
# Kim, Y., Bang, H., Kim, Y. & Bang, H. Introduction to Kalman Filter
and Its Applications. in Introduction and Implementations of the
Kalman Filter (IntechOpen, 2018). doi:10.5772/intechopen.80600.

#####################################################
# Clear any existing variables and plots.
rm(list = ls())
graphics.off()

## Packages ##
# If packages ggplot2 and dplyr not already downloaded, un-tag (remove
the #) to download packages before running the rest of the script.
#install.packages("ggplot2")
#install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

```
## Q2 ##
# 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 available fuel, the most likely value of available fuel, and
probability
#of negative fuel in the tank.

# define a seed for reproducibility
set.seed(930)

# Number of trials
n_trials <- 10^5

# sample from normal distribution with observed fuel as mean, and sd
as error in fuel sensor readings
samples <- rnorm(n_trials, 34, 20)


expected_value <- mean(samples)
density_estimate <- density(samples)
df <- approxfun(density(samples))

# Plot PDF
plot(density(samples), main="",
     xlab="Fuel in Tank", ylab="PDF")
abline(v=expected_value,col="blue",lty=1,lwd=4)
abline(v=0,col="black",lty=2,lwd=2)
legend("topright", c("Expected Value","Zero Fuel"),
       lwd=c(1,2,2), lty=c(1,2,2), col=c("blue","black"), cex=0.75)
title(main="PDF of Usable Fuel based on\nFuel Gauge Reading of 34
Liters")

# Find the probability of negative fuel

# density function (interpolating across points in density() values to
get function to integrate)
density_function <- approxfun(density_estimate$x, density_estimate$y,
rule=2)
# integrate from negative infinity to 0
prob_less_than_zero <- integrate(density_function, lower=-Inf,
upper=0)$value


# Find the most probable value
max_probability = max(density_estimate$y)
index <- which(density_estimate$y == max_probability)
mostprobablevalue <- density_estimate$x[index]
print(mostprobablevalue)
```

```
## Q4 ##
#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?

# New Prior is based on uniform distribution representing the capacity
of the tank
# max is based on total fuel capacity of tank
max=182
min=0
observed = 34
tanksd = 20
# grid-based method
gridvalues = seq(min, max, length.out=182)

# prior is uniform distribution from 0 to 182
uniform_prior <- dunif(gridvalues, min=min, max=max)

# likelihood
likelihood_values <- dnorm(gridvalues, mean=observed, sd=tanksd)

posterior <- likelihood_values*uniform_prior
posterior_normed <- posterior / sum(posterior)

# Plot CDF
dx = (gridvalues[2] - gridvalues[1])
cdf_posterior <- cumsum(posterior_normed) * dx
plot(gridvalues, cdf_posterior, type = "l", lwd = 5, col = "blue",
     main = "CDF of Posterior Distribution", xlab = "Fuel in Tank",
ylab = "CDF")

# Add vertical line at x=0, and horizontal at y=0 to 'point' at prob
of 0
segments(0, par("usr")[3], 0, 0, col="red", lwd=5)
segments(par("usr")[1], 0, 0, 0, col="red", lwd=5)

legend("bottomright", legend = c("CDF","Probability of 0 fuel"),
       col = c("blue", "red"), lwd = 2, cex = 1)


# Find probability of negative fuel
zero_grid_values_index <- gridvalues<0
probnegfuel <- sum(posterior_normed[zero_grid_values_index])
print(probnegfuel)

# Plotting all together
```

```
plot(density(samples), col="blue",
     lwd=2,
     ylab= "PDF",
     xlab='Fuel',
     main="Comparing Updated Posterior with Original PDF")
lines(gridvalues, posterior_normed, col = "red", lwd = 2)

# Add uniform distribution
lines(gridvalues, uniform_prior, col = "brown", lwd = 3)
segments(0, 0, 0, max(uniform_prior), col = "brown", lwd = 3)
segments(182, 0, 182, max(uniform_prior), col = "brown", lwd = 3)

# Add vertical lines for expected values
abline(v = 0, col = "black", lty = 2, lwd = 2)
abline(v = observed, col = "green", lty = 2, lwd = 2)

legend("topright", legend = c("Q2 PDF", "Q4 Posterior", "Uniform
Prior", "Observed 34 Liters", "0 Liters"),
       col = c("blue", "red", "brown", "green", "black"),
       lwd = 2,
       lty = c(1, 1, 1, 2, 2),  # Add line types here, 2 for dashed
       cex = 0.75)

## Q5 ##
# Repeat the step above using a Bayes Monte Carlo method.

# define a seed for reproducibility
set.seed(930)

# randomly drawing n samples from parameter distribution
# min/max from tank capacity
max=182
min=0
n_trials = 30*10^5
prior_samples <- runif(n_trials, min=min, max=max)

# Likelihood --
# likelihood is dnorm, so combining the samples with dnorm is
posterior
posterior_values <- dnorm(prior_samples, mean = observed, sd = tanksd)

# Normalize the posterior
BMC_posterior <- posterior_values / sum(posterior_values)

# Posterior Sample
BMC_posterior_samples <- sample(prior_samples, size = n_trials,
replace = TRUE, prob = BMC_posterior)

plot(density(BMC_posterior_samples), col = "purple", lwd = 2,
     xlab = "Fuel in Tank", ylab = "Density", main = "Posterior from
```

```
Bayes Monte Carlo (BMC)")


# Test convergence of BMC #
convergence_test <- 2 * qt(0.975, length(BMC_posterior_samples) - 1) *
sd(BMC_posterior_samples) / sqrt(length(BMC_posterior_samples))

# Now check if the CI is sufficiently small (<= 0.05)
if (convergence_test <= 0.05) {
  print("Confidence Interval is sufficiently small")
} else {
  print("Confidence Interval is too large")
}

# Plotting all together
plot(density(samples), col="blue",
     lwd=2,
     ylab= "PDF",
     xlab='Fuel',
     main="Comparing Posteriors: PDF, Updated Posterior, & BMC",
     ylim = c(0, 0.021))
lines(gridvalues, posterior_normed, col = "red", lwd = 2)
lines(density(BMC_posterior_samples), col = "purple", lwd = 4)

# Add vertical lines for expected values
abline(v = 0, col = "black", lty = 2, lwd = 2)
abline(v = observed, col = "green", lty = 2, lwd = 2)
legend("topright", legend = c("Q2 PDF","Q4 Posterior", "Q5 BMC
Posterior", "Observed 34 Liters", "0 Liters"),
       col = c("blue", "red", "purple", "green",  "black"), lwd = 2,
cex = 0.75)


# Find probability of negative fuel
zero_grid_values_index_BMC <- BMC_posterior_samples<0
probnegfuel_BMC <- sum(BMC_posterior[zero_grid_values_index_BMC])


## Q7 ##
# Produce a plot of the estimated available flight time.
#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?
#What is the probability that you run out of fuel trying to make it to
this airport?
fuelperminute = 18/60
sd_fuelperminute = 2/60

# 100 minutes plus 30 minutes reserve
fuel_needed_100min = fuelperminute*130
```

```
fuel_needed_100min_upper = (fuelperminute*130)+(sd_fuelperminute*130)
fuel_needed_100min_lower = (fuelperminute*130)-(sd_fuelperminute*130)


# Probability of having enough fuel for 130 minutes
min130_grid_values_index <- gridvalues<=fuel_needed_100min
probenoughfuel <- sum(posterior_normed[min130_grid_values_index])
print(probenoughfuel)

min130_grid_values_index_upper <- gridvalues<=fuel_needed_100min_upper
probenoughfuel_upper <-
sum(posterior_normed[min130_grid_values_index_upper])
print(probenoughfuel_upper)

min130_grid_values_index_lower <- gridvalues<=fuel_needed_100min_lower
probenoughfuel_lower <-
sum(posterior_normed[min130_grid_values_index_lower])
print(probenoughfuel_lower)



# Probability of running out of fuel
minimum_fuel_needed_100min = fuelperminute*100
minimum_needed_100min_upper = (fuelperminute*100)+
(sd_fuelperminute*100)
minimum_needed_100min_lower = (fuelperminute*100)-
(sd_fuelperminute*100)

fuelmin_grid_values_index <- gridvalues<=minimum_fuel_needed_100min
probrunningout <- sum(posterior_normed[fuelmin_grid_values_index])
print(probrunningout)

fuelmin_grid_values_index_upper <-
gridvalues<=minimum_needed_100min_upper
probrunningout_upper <-
sum(posterior_normed[fuelmin_grid_values_index_upper])
print(probrunningout_upper)

fuelmin_grid_values_index_lower <-
gridvalues<=minimum_needed_100min_lower
probrunningout_lower <-
sum(posterior_normed[fuelmin_grid_values_index_lower])
print(probrunningout_lower)


# Plot results on PDFs
plot(density(samples), col="blue",
     lwd=2,
     ylab= "PDF",
     xlab='Fuel',
```

```
     main="Comparing Posteriors: PDF, Updated Posterior, & BMC\nWith
Needed Fuel Estimates",
     ylim=c(0, 0.021))
lines(gridvalues, posterior_normed, col = "red", lwd = 2)
lines(density(BMC_posterior_samples), col = "purple", lwd = 2)

# Add the vertical lines as you have already
abline(v = 0, col = "black", lty = 2, lwd = 2)
abline(v = observed, col = "green", lty = 2, lwd = 3)
abline(v = fuel_needed_100min , col = "orange", lty = 2, lwd = 2)
abline(v = minimum_fuel_needed_100min, col = "cyan1", lty = 2, lwd =
2)

# Fill the area between the upper and lower values using polygon()
# Using orange color with alpha transparency for the fuel_needed range
polygon(c(fuel_needed_100min_lower, fuel_needed_100min_lower,
fuel_needed_100min_upper, fuel_needed_100min_upper),
        c(0, 0.055, 0.055, 0), col = rgb(1, 0.647, 0, alpha = 0.3),
border = NA)

# Using yellow color with alpha transparency for the minimum
fuel_needed range
polygon(c(minimum_needed_100min_lower, minimum_needed_100min_lower,
minimum_needed_100min_upper, minimum_needed_100min_upper),
        c(0, 0.055, 0.055, 0), col = rgb(0, 1, 1, alpha = 0.3), border
= NA)

# Legend
legend("topright", legend = c("Q2 PDF", "Q4 Posterior", "Q5 BMC
Posterior", "Observed 34 Liters", "0 Liters",
                              "Fuel Needed for 100 min + Reserve",
"Minimum Fuel for 100 minutes"),
       col = c("blue", "red", "purple", "green", "black", "orange",
"cyan1"),
       lwd = 2,
       lty = c(1, 1, 1, 2, 2, 2, 2),
       cex = 0.75)


minuteperfuel = 1/fuelperminute
sd_minuteperfuel = 1/sd_fuelperminute

available_flighttime = gridvalues*minuteperfuel
available_flighttime_upper = (gridvalues*minuteperfuel)+
(gridvalues*sd_fuelperminute)
available_flighttime_lower = (gridvalues*minuteperfuel)-
(gridvalues*sd_fuelperminute)

# Plot Estimated available flight time relative to fuel
plot(x = gridvalues, y=(available_flighttime/60),
```

```
     type='l',
     lwd=1,
     ylab= "Flight Time (hours)",
     xlab='Fuel',
     main="Flight Time vs Fuel Needed")
polygon(c(gridvalues, rev(gridvalues)),
        c((available_flighttime_upper / 60),
rev(available_flighttime_lower / 60)),
        col = rgb(1, 0, 0, 0.3, alpha=0.7), border = NA)
lines(x = gridvalues, y=(available_flighttime/60), col = "black", lwd
= 2)
legend("bottomright", legend = c("Estimate", "Upper and Lower Bound"),
       col = c("black", "red"), lwd = 2, cex = 0.75)
```