ENGS 107 Problem Set #3: Analytical Methods / Bayes Monte Carlo / Grid Methods

THE PROBLEM:

You are part of a team that is refining a safety system for an airplane that can land an airplane automatically at the next available and safe airport in case of an emergency (see https://www.youtube.com/watch?v=PiGkzgfR_c0 for an example). The computer system makes a decision on which airport to choose to land based on factors such as wind, runway length, legal requirements, fuel use, and available fuel. The team asks you to design, test, and document a draft computer program that takes as input a reading from a fuel sensor (together with some other information) and produces an estimate of the usable fuel in the tank with an estimate of the associated uncertainties. This information is then handed to a system that chooses a location to land. You also need to understand the interface of your tasks to the overall problem.

INPUTS:

- Total fuel tank capacity 182 liters
- Digital fuel sensor has an error *approximated as* following a Gaussian distribution with a standard deviation of 20 liters.
- Your fuel sensor reading is 34 liters
- Your airplane requires 18 liters per hour (*approximated* as uncertain with a Gaussian standard deviation of 2 liters per hour). Adopt estimates of fuel level and fuel consumption that are uncorrelated.

TASK:

In an effort to refine an airplane safety system, the **overarching task of this problem set** was focused on designing, testing, and documenting a draft computer program in R that takes in inputs (primarily a fuel sensor reading among other inputs) and uses Bayesian statistical methods to produce accurate estimates of the usable fuel in the airplane tank with an estimate of the associated uncertainties to ensure that the airplane's computer system makes safe landing decisions based upon available fuel. This overarching task was divided into a series of smaller tasks. After reviewing key literature materials relevant to the problem in the **first task**, the **second task** was focused on producing a probability density function (PDF) for the usable fuel in the airplane tank (i.e., the likelihood function) without any other information besides the fuel gauge reading. This task required plotting the likelihood function and calculating 1) the expected value of available fuel in the airplane tank, 2) the most likely value of available fuel, 3) and the probability of negative fuel in the tank. It also required the student to address if these calculated estimates made sense. The **third task** required the student to conceptually address how a proper prior can be implemented and used to resolve the issue of unrealistic fuel estimates in the airplane fuel tank. It also required the student to define this physically based prior (i.e., subjective prior) both textually and in the code script in R. To visualize the physically based proper prior, the prior should be plotted with the likelihood function from Task 2 to observe how the proper prior constrains the airplane fuel tank capacity to a realistic range.

The **fourth and fifth tasks** required the student to implement two different Bayesian statistical approaches to determine a Bayesian update from the defined prior and the likelihood function. The **fourth task** required writing a code script that used a grid-based method to determine the Bayesian update from the prior and likelihood function. To observe the impact of the prior, the posterior from the grid-based method should have been plotted with the likelihood function and the prior. The **fifth task** required the student to write a code script that used a Bayes Monte Carlo method to determine the Bayesian update from the prior and likelihood function. A histogram plot of the posterior distribution from the Bayes Monte Carlo simulation should have been produced. In addition, the posterior from the Bayesian update using the Bayes Monte Carlo method should have been plotted along with the posterior from the grid-based method, the likelihood function, and the prior to observe the impact of the prior on both posteriors relative to the likelihood function. For both tasks, the 1) expected value of available fuel, the 2) most likely value of available fuel, and 3) the probability of negative fuel in the tank should have been calculated. Based upon the probability of negative fuel in the tank, both tasks required addressing if the issue of unrealistic fuel estimates issue had been fixed, and if so, how.

To extend the analysis, the **sixth task** required the student to outline the set of assumptions that would have to be made for a simple analytical Kalman filter solution to the problem, and to subsequently address if the stated assumptions were realistic considering the problem. The **seventh and final task** required the student to produce a plot of the estimated available flight time and to use the plot to calculate the answers to two questions: 1) the probability that the airplane makes it to an airport that is 100 minutes away with at least 30 minutes of reserve fuel as required by regulations and 2) the probability of running out of fuel while trying to make it to the airport. This task also required the student to produce a plot of the cumulative distribution function (CDF) over the available flight time and a plot of the survival function over the available flight time to further visualize the estimated available flight time.

APPROACH:

Note: ***This section describes my approach for writing the code script to accomplish each task. All answers to the questions and prompts posed in each task are in the results section. All figures for each task are in the figures section. Please refer to the assumptions and choices sections in this document for further insights into what assumptions were made and for an explanation of key choices (with plausible alternatives) that were also made, respectively.***

To accomplish the tasks described above, my **approach** was broadly aimed at producing a cohesive code script in R that sequentially solved each of the problems and produced the required results and plots for each of the tasks. I decided to visually divide the code script into distinct sections for each task to improve readability and I structured each task into a distinct series of logical steps and sub-steps. I decided to articulate my approach for each task 1) conceptually in a paragraph format and 2) in a list of steps and sub-steps detailed below:

The initial phase of my approach was focused on establishing the environment for the code script by defining the problem, storing the inputs as variables, and setting a fixed seed:

1. Stating the overarching problem from PS#3 in the header notes of the code script
2. Storing each of the inputs provided in the problem statement as a separate variable
3. Setting a single fixed seed to facilitate reproducibility of the results

---

<u>My approach for accomplishing Task 2</u> is detailed below 1) conceptually in a paragraph format and 2) in an ordered sequence of steps and sub-steps. The objective of task 2 was to draw the probability density function (PDF) for the usable fuel in the tank (i.e., the likelihood function) without any other info besides the fuel gauge reading. It was also to determine the expected value of available fuel, the most likely value of available fuel, and the probability of negative fuel in the tank and to address if these estimates make sense.

<u>Paragraph conceptually describing my approach for Task 2</u>:

To accomplish Task 2, the **<u>first phase of my approach</u>** was focused on defining the probability density function (PDF) for fuel estimation. To do this, I first defined a broad range of usable fuel levels using the seq function to ensure that the left tail of the distribution was not truncated despite extending into negative fuel levels (which are physically impossible). I subsequently used the dnorm function to calculate the probability density function (i.e., the likelihood function), which models the probability distribution of usable fuel in the tank. After generating the likelihood function, the **<u>second phase of my approach</u>** was to calculate the 1) expected value of usable fuel in the tank (the mean), 2) the most likely value of available fuel (the mode) and 3) the probability of negative fuel in the tank using the pnorm function. The **<u>third phase of my approach</u>** was to plot the likelihood function along with 1) the expected value of available fuel, 2) the most likely value of available fuel, 3) and the standard deviations of the fuel sensor reading for reference. I also printed the calculated estimates onto the second page of PDF file.

<u>Numbered sequence of steps and sub-steps outlining my approach for Task 2</u>:

1. Use the seq function to define a range of possible fuel levels from -60 liters (an arbitrary lower bound) to 182 liters (the maximum fuel capacity) to allow for visualization of the entire left tail of the PDF (likelihood function) without truncation
2. Use the dnorm function to calculate the probability density function (PDF) based on the fuel sensor reading of 34 liters (the mean) and the fuel sensor error (SD = 20 liters)
3. Calculate the following estimates:
    a. The expected value of available fuel (the mean)
    b. The most likely value of available fuel (the mode)
    c. The probability of negative fuel in the tank (using the pnorm function)
4. Specify a PDF file name and open a PDF file to plot the PDF of the usable fuel (i.e., the likelihood function)
5. Plot the PDF for usable fuel in the tank (PDF [y-axis] over Fuel (Liters) [x-axis])
    a. Specify the axes labels and plot title
    b. Expand the x-axis of the plot to ensure visibility of the likelihood function
    c. Add vertical dashed lines on the plot to denote the expected value of available fuel (the mean) and the most likely value of available fuel (the mode)

   d. Add vertical lines to denote ± 1 standard deviation from the mean
   e. Add a legend to label the key elements of the plot
6. Paste the results for each of the estimates on the subsequent PDF file page (page 2)

<u>My approach for accomplishing Task 3</u> is detailed below 1) conceptually in a paragraph format and 2) in an ordered sequence of steps and sub-steps. The objective of task 3 was to 1) answer how a proper prior can be used to address the issue of unrealistic fuel estimates in the tank and 2) define this physically based prior in the code script.

<u>Paragraph conceptually describing my approach for Task 3</u>:

To address the issue of unrealistic fuel estimates in the airplane tank, my approach was to define a proper prior that eliminated negative fuel estimates based upon my previous, real-world knowledge that fuel levels cannot be negative. To accomplish this, the **<u>first phase of my approach</u>** was to define a uniform prior distribution. To do this, I first set the lower bound of the airplane fuel tank capacity to 0 liters (the realistic minimum capacity of the fuel tank). Since probability density functions must integrate to 1, I next determined the prior height over my specified range from 0 to 182 liters. This ensured that the total probability summed to 1, making it a valid probability distribution. I next used an ifelse statement to define the prior probability distribution, such that fuel levels inside of the range of 0 to 182 liters had a constant probability density and fuel levels outside of the range (i.e., negative fuel levels) were a assigned a probability value of zero. The **<u>second phase of my approach</u>** was to use the seq function to create a sequence of fuel values for plotting the prior and to subsequently use the rep function calculate the prior probability values corresponding to each fuel level. To conclude Task 3, the **<u>third phase of my approach</u>** was to plot the likelihood function with my proper prior along with 1) the expected value of available fuel, 2) the most likely value of available fuel, 3) and the standard deviations of the fuel sensor reading for reference.

<u>Numbered sequence of steps and sub-steps outlining my approach for Task 3</u>:

1. Create a variable to specify the minimum realistic fuel capacity in the tank (0 liters)
2. Determine the height of the uniform prior distribution that constrains the airplane fuel tank capacity between 0 liters and 182 liters
3. Use an ifelse function to define the prior probability distribution over the entire range of usable fuel in the airplane tank (0 liters to 182 liters)
4. Use the seq function to create a sequence of fuel values for plotting the proper prior
5. Use the rep function to calculate the prior probability values for each fuel level
6. Plot my physically based proper prior with the likelihood function
   a. Specify the axes labels and plot title
   b. Expand the x-axis of the plot to ensure visibility of the likelihood function
   c. Add vertical dashed lines to denote the expected value of available fuel (the mean) and the most likely value of available fuel (the mode)
   d. Add vertical lines and labels to denote ± 1 standard deviation from the mean
   e. Add vertical and horizontal dashed lines to denote the proper prior on the plot

    f. Add a legend to label all of the key elements of the plot

---

<u>My approach for accomplishing Task 4</u> is detailed below 1) conceptually in a paragraph format and 2) in an ordered sequence of steps and sub-steps. The objective of Task 4 was to use a grid-based method to determine the Bayesian update from the prior and likelihood function and to add this posterior to the plot produced in Task 3. It was also to re-calculate the probability of negative fuel in the tank and to determine if this fixed the issue.

<u>Paragraph conceptually describing my approach for Task 4</u>:

To determine the Bayesian update from my proper prior (defined in Task 3) and the likelihood function (defined in Task 2) using a grid-based method, the **<u>first phase of my approach</u>** was to calculate the unnormalized posterior using Bayes' Theorem (posterior is proportional to likelihood x prior). This posterior was the updated belief about the fuel level based upon the previously defined likelihood and prior. The **<u>second phase of my approach</u>** was to ensure that any negative fuel levels were assigned a probability value of zero. The **<u>third phase of my approach</u>** was to normalize the posterior distribution such that it summed to a value of 1. The purpose of this was to make sure that it was a valid probability distribution. After normalizing the posterior probability distribution to 1, the **<u>fourth phase of my approach</u>** was to re-calculate the 1) expected value of available fuel, 2) the most likely value of available fuel, and 3) the probability of negative fuel in the tank from the posterior. With these updates estimates, the **<u>fifth phase of my approach</u>** was to plot the posterior from the grid-based method with the likelihood function (from Task 2) and my proper prior (from Task 3). The purpose of this was to observe the effect of the prior on the posterior distribution in comparison to the likelihood function. The sixth and final phase of my approach for accomplishing Task 4 was to print the calculated estimates from the posterior from the grid-based method onto a blank page on the PDF file.

<u>Numbered sequence of steps and sub-steps outlining my approach for Task 4</u>:

1. Rename the pdf_usable_fuel variable as 'likelihood' for to improve clarity
2. Calculate the posterior probability using Bayes' Theorem (without normalization)
3. Ensure that no probability mass is assigned to negative fuel values
4. Normalize the posterior so it sums to a value of 1. The purpose of this step is to turn it into a proper probability distribution
5. Calculate the following estimates after the Bayesian update (grid-based method)
    a. The expected value of available fuel (the mean)
    b. The most likely value of available fuel (the mode)
    c. The probability of negative fuel in the tank
6. Plot the posterior of the Bayesian update from the grid-based method with the proper prior and the likelihood function from the Task 3 plot
    a. Specify the axes labels and plot title
    b. Expand the x-axis of the plot to ensure visibility of the likelihood function
    c. Add vertical dashed lines to denote the expected value of available fuel (the mean) and the most likely value of available fuel (the mode)

d. Add vertical lines and labels to denote ± 1 standard deviation from the mean
e. Overlay the posterior distribution from the grid-based method on the plot
f. Add vertical and horizontal dashed lines to denote the proper prior on the plot
g. Add a legend to label all of the key elements of the plot

7. Paste the results for each of the calculated estimates from the Bayesian update using a grid-based method onto the next page of the PDF file

---

<u>My approach for accomplishing Task 5</u> is detailed below 1) conceptually in a paragraph format and 2) in an ordered sequence of steps and sub-steps. The objective of Task 5 was to use a Bayes Monte Carlo method to determine the Bayesian update from the prior and likelihood function and to add this posterior to the plot (from Task 4). It was also to re-calculate the probability of negative fuel in the tank and to determine if this fixed the issue.

<u>Paragraph conceptually describing my approach for Task 5</u>:

To determine the Bayesian update from my proper prior (defined in Task 3) and the likelihood function (defined in Task 2) using a Bayes Monte Carlo method, the **<u>first phase of my approach</u>** was to generate Monte Carlo samples from the prior distribution. To do this, I first had to specify the maximum number of Monte Carlo samples as 10 million. I next used the runif function to generate 10 million samples between 0 liters and 182 liters to ensure a relatively high-resolution approximation of the fuel level distribution in the airplane tank. The **<u>second phase of my approach</u>** was to use the dnorm function to calculate the likelihood for each prior sample. The **<u>third phase of my approach</u>** was to calculate the posterior weights using Bayes' Theorem (i.e., the posterior is proportional to likelihood x prior). Since my prior was a uniform prior, the posterior was just the likelihood scaled to sum to 1. The **<u>fourth phase of my approach</u>** was to implement a stopping criterion to ensure that the maximum number of Monte Carlo samples did not exceed 10 million. The purpose of this was to ensure that my code script did not take a significant amount of computational time to run. I also felt as if 10 million samples was a large enough number of samples to achieve sufficiently accurate enough estimates for the expected value of available fuel, the most likely value of available fuel, and the probability of negative fuel in the tank. The **<u>fifth phase of my approach</u>** was to resample the posterior using importance sampling. Importance sampling selects fuel levels from the prior proportionally to their posterior weight, which produces a new set of samples following the posterior distribution. Following resampling, the **<u>sixth phase of my approach</u>** was to calculate the estimates for the 1) expected value of available fuel, 2) the most likely value of available fuel, and 3) the probability of negative fuel in the tank from the posterior from the Bayes Monte Carlo method. To transition to plotting my results, **<u>the seventh phase of my approach</u>** was to produce a histogram of posterior samples and normalize it to sum to 1 to make it a valid probability distribution. The next step of this phase was to plot the histogram and include the expected value of available fuel and the most likely value of available fuel on the histogram plot for reference. The **<u>eighth phase of my approach</u>** was to produce a plot of the posterior from the Bayes Monte Carlo method overlaid on the plot of the posterior from the grid-based method (Task 4), the proper prior (Task 3), and the likelihood function (Task 2). Finally, the **<u>ninth phase of my approach</u>** was to print the calculated estimates from the posterior onto the PDF file.

Numbered sequence of steps and sub-steps outlining my approach for Task 5:

1. Specify the maximum number of Monte Carlo samples as 10 million
2. Use the runif function to generate samples (up to the maximum number of samples) from the proper prior assuming that there is a uniform distribution within the specified airplane tank capacity of 0 liters to 182 liters and that all samples are within the specified range
3. Use the dnorm function to calculate the likelihood for each prior sample (assuming a normal distribution centered at the fuel sensor reading)
4. Calculate the posterior weights using Bayes' Theorem (the weights are normalized)
5. Implement a stopping criterion to ensure that the Monte Carlo simulation does not exceed the maximum number of samples (10 million samples)
6. Introduce a variable called sampling_size to ensure that resampling from the posterior does not exceed the maximum number of samples (10 million)
7. Re-sample from the posterior using importance sampling with replacement
8. Calculate the following estimates after the Bayesian update (Monte Carlo method)
    a. The expected value of available fuel (the mean)
    b. The most likely value of available fuel (the mode)
    c. The probability of negative fuel in the tank
9. Create histogram data for the posterior distribution from the Monte Carlo simulation
10. Normalize the histogram density so that the total area sums to a value of 1
11. Plot the normalized histogram from the Monte Carlo simulation
    a. Specify the axes labels and plot title
    b. Extend the x-axis and adding custom numerical labels to improve readability
    c. Use a for loop to add histogram bars to the plot with normalized density
    d. Add vertical lines to denote the expected value of available fuel (the mean) and the most likely value of available fuel (the mode) on the histogram plot
    e. Add a legend to the histogram plot to label the key features
12. Plot the posterior of the Bayesian update from the Monte Carlo method with the posterior from the grid-based method, the proper prior and the likelihood function
    a. Specify the axes labels and plot title
    b. Extend the x-axis and adding custom numerical labels to improve readability
    c. Perform the Monte Carlo posterior density estimation
    d. Normalize the Monte Carlo density function
    e. Overlay the posterior from the Monte Carlo simulation onto the plot
    f. Add vertical and horizontal dashed lines to denote the proper prior
    g. Add vertical lines to denote the expected value of available fuel (the mean) and the most likely value of available fuel (the mode)
    h. Add a legend to label all of the key features on the plot
13. Paste the results for each of the calculated estimates from the Bayesian update using a Monte Carlo method onto the next page of the PDF file

---

***Note: My answers for Task 6 are addressed in the results section of this document. No code was written to accomplish Task 6 as it posed conceptual questions regarding what assumptions should be made for a simple analytical Kalman filter solution to the problem.**

<u>My approach for accomplishing Task 7</u> is detailed below 1) conceptually in a paragraph format and 2) in an ordered sequence of steps and sub-steps. The objective of Task 7 was to produce a plot of the estimated available flight time to address two questions: 1) the probability that you make it to an airport that is 100 minutes flight time away with at least 30 minutes of reserve fuel (as required by regulations), and 2) the probability of running out of fuel trying to make it to the specified airport.

<u>Paragraph conceptually describing my approach for Task 7</u>:

To produce a plot of the estimated available flight time and calculate the requested probabilities, the **<u>first phase of my approach</u>** was to determine the total fuel required to reach the destination. To do this, I first created variables defining the flight time (100 minutes) and the reserve fuel time (30 minutes). I next converted the required flight time into the required fuel consumption using the provided fuel consumption rate (in liters per hour). I next calculated the total fuel required to reach the destination by adding the required fuel plus the reserve fuel. The **<u>second phase of my approach</u>** was to calculate the probability density function (PDF) for the fuel levels based on the fuel sensor reading. To do this, I first used the seq function to create a sequence of fuel levels over the specified range from 0 liters to 182 liters (from the proper prior) with a step size (or increment) of 0.01 liters. I then used the dnorm function to calculate the PDF of the available fuel and subsequently normalized the PDF to sum to 1 to ensure that it was a valid probability distribution. The **<u>third phase of my approach</u>** was to convert the fuel levels to available flight time (for each fuel level). The **<u>fourth phase of my approach</u>** was to calculate the cumulative distribution function (CDF) using the cumsum function. The CDF accumulates the probability mass, which gives the probability that available flight time is less than or equivalent to the given value. With this groundwork laid in my code script, the **<u>fifth phase of my approach</u>** was to calculate the probability of reaching the airport with the reserve fuel. To do this, I wrote an equation to 1) filter the fuel PDF values where the available flight time is greater than or equivalent to 130 minutes and 2) to sum those probabilities to determine the probability of making it safely to the airport. With this calculation complete, the **<u>sixth phase of my approach</u>** was to calculate the probability of running out of fuel. To do this, I wrote an equation to 1) filter the fuel PDF values where the available flight time was less than 100 minutes and 2) to sum those probabilities to determine the probability of running out of fuel before reaching the airport. To visualize the estimated available flight time, the **<u>seventh phase of my approach</u>** was to produce three plots 1) a plot of the PDF over the estimated available flight time, 2) a plot of the CDF on a logarithmic scale over the estimated available flight time, and 3) a plot of the survival function (determined by taking 1 – CDF). It is important to note that the PDF represents the probability distribution of available flight time, the CDF provides the probability of reaching or exceeding a given flight time, and finally the survival function provides the probability of having more than a given flight time. The purpose of the logarithmic scale for the y-axis of the survival function plot was to improve the visualization of the tail probabilities. The **<u>eighth and final phase</u>** of the approach for accomplishing Task 7 was to print the calculated probabilities on the last page of the PDF document and to close the PDF document.

<u>Numbered sequence of steps and sub-steps outlining my approach for Task 7</u>:

1. Define the specified flight time (100 minutes)
2. Define the reserve fuel time (30 minutes)
3. Convert the flight time requirement into a fuel requirement
4. Calculate the total fuel required to reach the destination (accounting for reserve fuel)
5. Define a range of fuel levels using the seq function
6. Use dnorm to calculate the PDF for the fuel levels based on the fuel sensor readings
7. Normalize the PDF so that it sums to a value of 1
8. Calculate the available flight times corresponding to each fuel level
9. Calculate the cumulative distribution function (CDF)
10. Calculate the cumulative probability of meeting the flight requirement with reserve fuel
11. Calculate the cumulative probability of running out of fuel before landing at airport
12. Plot the estimated available flight time
    a. Specify the axes labels and the plot title
    b. Add vertical lines to denote the required flight time and the total required flight time including the reserve
    c. Add a legend to label the key features of the plot
13. Plot the CDF over available flight time
    a. Specify the axes labels and plot title
    b. Add vertical lines to denote the required flight time and the total required flight time including the reserve
    c. Add a legend to label the key features of the plot
14. Plot the survival function over available flight time
    a. Specify the axes labels and plot title
    b. Set the y-axis to be in a logarithmic (log) scale
    c. Add vertical lines to denote the required flight time and the total required flight time including the reserve
    d. Add a legend to label the key features of the plot
15. Paste the results for the two probability questions on the last page of the PDF file
16. Close the entire PDF file using the dev.off() function

<u>ASSUMPTIONS</u>:

In the process writing the code script in R to solve the described tasks, several key assumptions were made. In terms of the provided inputs, **<u>one assumption</u>** is that the fuel sensor error follows a Gaussian (normal) distribution (with a standard deviation of 20 liters). This <u>assumes that the measurement errors are symmetrically distributed</u>, meaning that under- and overestimations are equally probable (including extreme underestimations or overestimations), which may not be entirely realistic under all possible real-world scenarios. Also from the provided inputs, a **<u>second assumption</u>** arises from the statement that the airplane requires 18 liters of fuel per hour, approximated as uncertain with a Gaussian standard deviation of 2 liters per hour. This <u>assumes that variations in fuel consumption follow a normal distribution</u>, which also may not be entirely realistic under all possible real-world scenarios.

Extending beyond the inputs, a **third assumption** was made in setting a single fixed seed for the code script. It was assumed that setting a seed would enable reproducibility of the results. It was also assumed that setting a single fixed seed (compared to multiple fixed seeds), was sufficient for this problem set, as it did not explicitly state that multiple seeds must be used. In terms of plotting the likelihood function in Task 2, a **fourth assumption** made was that setting the lower bound of the x-axis to -60 liters of fuel (a physically impossible value) was sufficient to prevent any truncation of the left tail of the distribution that would interfere with the calculations for the expected value of available fuel and the most likely value of available fuel. Also for Task 2, in terms of calculating the probability of negative fuel in the tank, a **fifth assumption** was that the Gaussian distribution remained valid beyond realistic fuel limits (negative fuel is impossible).

In terms of Task 3, a **sixth assumption** made was that using a uniform prior assumed that all fuel levels were equally probable within the realistic fuel tank capacity ranging from 0 to 182 liters. This assumes no prior knowledge pertaining to airplane fuel tank levels aside from the fact that it is physically impossible to have 1) negative fuel and 2) fuel levels greater than the tank capacity. It is worth noting that previously published data describing airplane fuel consumption patterns could have been used to develop a more informative and detailed prior. Although a requirement for Task 4, a **seventh assumption** arises from the use of a grid-based method to determine the Bayesian update. Using a grid-based method assumes that the calculated posterior is a sufficiently accurate numerical approximation of the true Bayesian update, although other methods may yield different and more accurate results (i.e., exact Bayesian inference).

In terms of Task 5 and determining the Bayesian update using a Bayes Monte Carlo method, an **eighth assumption** made was that setting the maximum number of samples to 10 million samples was a sufficiently large enough number of samples to enable an accurate approximation of the posterior from the prior and likelihood function. Although it would be ideal to use a larger maximum number of samples, selecting a larger value may have led to long computation times when running the code script. Also in terms of Task 5, a **ninth assumption** made was through the use of importance sampling, which assumes that the likelihood function provides an appropriate and proper weighting of prior samples to acquire an accurate posterior distribution.

In terms of Task 7, a **tenth assumption** made in the writing of question A was that a reserve fuel requirement of 30 minutes of additional fuel was a sufficient amount of additional fuel to have in the airplane (based on regulatory requirements). This assumes that this is a sufficient amount of reserve fuel for all airplanes and under all possible flight conditions, which may not be true. Although briefly addressed earlier, an **eleventh assumption** arising from the conversion of fuel levels to flight time was that the fuel consumption rate remains constant over time (with a specified standard deviation), which may not be entirely realistic due to the possibility of changing conditions during a flight (i.e., changes in turbulence, wind direction, altitude etc.)

In terms of additional, general assumptions made in the writing of the code script, a **twelfth assumption** made was that the fuel sensor error was independent from the fuel consumption rate. If an airplane burns more fuel than expected, the fuel sensor reading may also be inaccurate due to variations in fuel distribution or fuel pressure. More broadly, a **thirteenth assumption** made was that running out of fuel was the worst possible outcome or scenario. In reality, an airplane engine failure is another worst possible outcome that was not considered or integrated into this analysis. A **fourteenth assumption** made was that the airplane followed a fixed, straight flight path directly to its airport destination. In reality, airplanes rarely fly in a perfectly straight flight path and often have to make small changes in their route due to directions from air traffic control (as it may have to wait for other planes to land). A **fifteenth assumption** made was that no fuel loss occurs from the fuel tank. It is possible that unexpected fuel loss may occur from the fuel tank during a flight due to leaks or mechanical failures, which would impact the amount of available fuel in the tank. A **sixteenth assumption** made was that all colors, fonts, and character sizes selected in the creation of the plots facilitated optimal readability for all viewers.

RESULTS:

TASK 1:

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

1. Qian, S. S., Stow, C. A., & Borsuk, M. E. (2003). On Monte Carlo methods for Bayesian inference. Ecological Modelling, 159(2-3), 269–277.
2. D'Agostini, G. (2003). Bayesian reasoning in data analysis: A critical introduction. Singapore: World Scientific Publishing. (**Chapter 6 only**).
3. 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

Each of the three literature sources above were carefully read and reviewed

TASK 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 the probability of negative fuel in the tank. Do these estimates make sense to you?

Determine the expected value of available fuel (mean):

- Expected value of available fuel (mean) = 34 liters of fuel

Determine the most likely value of available fuel (mode):

- Most likely value of available fuel (mode) = 34 liters of fuel

Determine the probability of negative fuel in the tank:

- Probability of negative fuel in the tank = 4.5%

<u>Do these estimates make sense to you</u>?

In terms of determining the **<u>expected value of available fuel</u>**, the inputs in the problem statement specified that the airplane fuel sensor had a reading of 34 liters, with an error approximated as following a Gaussian distribution with a standard deviation of 20 liters. Assuming the airplane fuel level follows a normal Gaussian distribution centered at 34 liters with a standard deviation of 20 liters, the **<u>expected value of available fuel (i.e., the mean) was calculated in the code script in R as 34 liters</u>**. Theoretically, this estimation for the expected value of available fuel makes sense because the distribution is symmetric, and thus the mean is located at the peak of the curve (i.e., the fuel sensor reading). Although this estimate theoretically makes sense, **<u>it does not completely make sense from a real-world perspective</u>**. The basis for this arises from the left tail of the probability density function (PDF) extending past 0 liters and into negative fuel levels. The issue with this is that it is not physically possible to have negative liters of fuel in the airplane tank. Thus, a **<u>more accurate estimate of the expected value of available fuel</u>** would be calculated by truncating the probability density function at 0 liters of fuel, as this truncation would shift the expected available fuel value upwards to a value slightly larger than 34 liters of fuel. In conclusion, the estimated expected fuel value of 34 liters may make sense theoretically in this case, but realistically this is not the most accurate estimation for the expected value of available fuel because negative fuel levels are not physically possible.

<u>In terms of determining the most likely value of available fuel, my code script in R calculated a value of 34 liters</u>. The most likely value of available fuel is another term for the mode. Assuming a symmetric Gaussian distribution, the most likely value (mode) is the peak (or maximum) of the distribution, which in this case is 34 liters of fuel (which coincides with the mean). Thus, this estimation makes sense because the fuel sensor was specified as having a reading of 34 liters, and this value was at peak of the normal, symmetric Gaussian distribution.

<u>In terms of determining the probability of negative fuel in the tank, my code script in R calculated value of 4.46%</u>. The probability of negative fuel in the tank was calculated using the pnorm function in R, which determines the cumulative probability of a normal distribution. Since the distribution was centered at 34 liters (the fuel sensor reading), with a standard deviation of 20 liters, the probability that the fuel is below 0 liters corresponds to the left tail of the PDF that extends past 0 liters and into negative fuel levels. The calculated Z-score for a value of 0 liters was: $Z = (0\text{-}34) / (20) = -1.7$. $P(Z < -1.7)$ on a standard normal table corresponds to a value of 0.0446, or 4.46%. After rounding, this estimated value of 4.5% for the probability of negative fuel in the tank makes sense because the normal distribution extended from 34 liters in both directions, (including the left tail of the distribution extending into negative fuel levels), and thus it makes sense that there is a very small probability that the actual fuel level in the airplane tank is a negative value even though the fuel sensor reading was a positive value of 34 liters. To better understand the math mentioned above, the Z-score informs how far a particular value (i.e., 0 liters in this case) is from the mean (i.e., 34 liters of fuel) in terms of standard deviations. My calculated Z score of -1.7 informs that a fuel level of 0 liters is 1.7 standard deviations below the mean (34 liters). Upon consulting the standard normal table, I learned that $Z = -1.7$ gives a probability of 4.46%. This means that about 4.46% of the time, the true level of fuel in the tank is below 0 liters of fuel, even though the fuel sensor reading specifies that the fuel level is 34 liters. Thus, the rounded estimated value of 4.5% for the probability of negative fuel in the airplane tank makes sense because 1) the airplane fuel sensor reading is uncertain, 2) the normal distribution extends past 0 liters into negative fuel levels 3) 0 liters is 1.7 standard deviations below the mean of 34 liters, and 4) a Z-score of -1.7 corresponds to a probability of ~ 4.5%.

TASK 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).

Conceptually, a **proper prior** can be used to address the **issue of unrealistic fuel estimates in the airplane tank** by incorporating a few key elements of real-world knowledge about airplane fuel tank capacity and fuel levels. **First and most importantly, a proper prior should account for the fuel capacity constraints in the airplane tank**. In this case, the fuel level cannot exceed 182 liters (the maximum capacity), and it also cannot go below 0 liters (the minimum capacity) because there physically cannot be a negative level of fuel in the airplane tank. By defining this strict range for the airplane fuel tank capacity from 0 liters to 182 liters, this prior functions to truncate the probability density function for usable fuel in the tank at 0 liters, thus preventing the left tail of the PDF from extending past 0 liters and into impossible negative fuel levels. As a result, this proper prior will ensure that 1) any posterior estimates of airplane fuel levels will remain consistent with the real-world constraints of the airplane fuel tank capacity and 2) will reduce the possibility of obtaining unrealistic fuel estimates that either exceed the fuel capacity of the tank or suggest negative fuel levels, which are physically impossible. By defining a strict range for the airplane fuel tank capacity from 0 liters to 182 liters, this prior will produce more reliable and accurate fuel estimates in light of the large uncertainty present in the digital fuel sensor, thus addressing the issue of unrealistic fuel estimates in the airplane tank.

As further conceptual considerations, a **proper prior** could also incorporate prior knowledge of the following elements to address the issue of unrealistic fuel estimates. A proper prior could also incorporate prior knowledge of the airplane fuel consumption rate or operational fuel usage based upon prior historical flight data. For instance, if the airplane fuel level was measured before the airplane took off, a reasonable estimate of the fuel level should follow an expected fuel consumption rate (with a specified standard deviation) from that initially measured fuel level. A proper prior could also incorporate prior knowledge pertaining to airplane refueling patterns and practices. For instance, airplane refueling often occurs in set increments following a standardized procedure. Key considerations here include how much reserve fuel to add to the airplane tank, or if it is necessary or not to fill the airplane tank to full capacity depending on the expected duration of the flight, or to specify a minimum refueling threshold. These are important considerations to potentially integrate into a proper prior to address the issue of unrealistic fuel estimates. More broadly, the knowledge integrated in a proper prior should help to avoid the issue of underestimating the fuel level (i.e., potentially leading to unnecessary emergency landings) and also incorporate prior knowledge to help avoid the issue of overestimating the fuel estimate (i.e., leading to the selection of unsafe landing locations). These are all important elements to consider including in a proper prior to produce more accurate fuel estimates and to address the issue of unrealistic fuel estimates to increase the safety of flying in an airplane.

<u>To define my physically based proper prior</u>, based upon the considerations stated above and the inputs provided in the problem statement, the proper prior written into my code script in R constrains the airplane fuel tank capacity to a range from 0 liters to 182 liters to reflect the physical, realistic capacity of the airplane fuel tank. The constraint introduced through this proper prior will produce a resulting probability density function (PDF) that exhibits a <u>truncated normal distribution</u> from 0 liters to 182 liters and will produce a more accurate fuel estimate.

*Please refer to Task 3 within the <u>code script in the Appendix section</u> or to the <u>approach section</u> to observe how this proper prior was specifically defined within the code script in R.

<u>TASK 4</u>:

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?

After writing the code script in R implementing a grid-based method to determine the Bayesian update from my physically based prior and the likelihood function, I produced a plot with the posterior probability density function (PDF) overlaid on the likelihood function. I also recalculated the <u>expected value of available fuel</u>, the <u>most likely value of available fuel</u>, and the <u>probability of negative fuel in the tank</u> of the posterior PDF. These results are stated below:

The expected value of available fuel (mean) of the posterior:

- Expected value of available fuel (mean) = 36 liters of fuel

The most likely value of available fuel (mode) of the posterior:

- Most likely value of available fuel (mode) = 34 liters of fuel

The probability of negative fuel in the tank of the posterior:

- Probability of negative fuel in the tank = 0.0%

As stated above, the probability of negative fuel in the airplane tank for the posterior probability density function (PDF) (i.e., my Bayesian update from my physically based prior and likelihood function) was calculated in my code script in R as 0.0%. **<u>Yes, this has fixed the issue of acquiring unrealistic fuel estimates in the airplane tank as a result of incorporating a physically based prior that constrained the airplane fuel tank capacity from 0 liters to 182 liters that prevents the estimated fuel level from being a negative value</u>**.

For background, the original PDF (i.e., the likelihood function) was solely based upon the fuel sensor reading of 34 liters and its Gaussian distribution with a standard deviation of 20 liters, which resulted in the left tail of the PDF extending into negative fuel levels. The probability of negative fuel was calculated from this PDF, which resulted in calculating the original probability of negative fuel in the tank as 4.5% (i.e., the cumulative probability of fuel being less than 0 liters). In contrast, in the Bayesian update, the physically based prior was included, which constrained the airplane fuel tank levels to a realistic capacity of 0 liters to 182 liters. This omitted unrealistic, negative fuel levels in the tank. The prior was defined as uniform for fuel tank values between 0 liters and 182 liters, but as a value of 0 for any negative fuel values. As a result of the prior assigning a probability of 0 to negative fuel in the tank, the posterior probability of the fuel level being a negative value was also 0. Thus, the resulting probability of negative fuel in the airplane tank was calculated as 0.0%, which resolved the issue of unrealistic fuel estimates in the airplane tank and reflects the reality that fuel levels cannot be negative.

TASK 5:

Repeat the step above using a Bayes Monte Carlo method.

After writing the code script in R implementing a Bayes Monte Carlo method to determine the Bayesian update from my physically based prior and the likelihood function, I produced a plot with the posterior probability density function (PDF) overlaid on the posterior PDF from the grid-based method and the original likelihood function. I also calculated the expected value of available fuel, the most likely value of available fuel, and the probability of negative fuel in the tank of the posterior from the Bayes Monte Carlo method. These results are stated below:

The expected value of available fuel (mean) of the posterior:

- Expected value of available fuel (mean) = 36 liters of fuel

The most likely value of available fuel (mode) of the posterior:

- Most likely value of available fuel (mode) = 35.2 liters of fuel

The probability of negative fuel in the tank of the posterior:

- Probability of negative fuel in the tank = 0.0%

In a similar manner to task 4, the probability of negative fuel in the airplane tank for the posterior probability density function (PDF) (i.e., my Bayesian update from my physically based prior and likelihood function) using a Bayes Monte Carlo method was calculated in my code script in R as 0.0%. **Yes, this has fixed the issue of acquiring unrealistic fuel estimates in the tank as a result of incorporating a physically based prior that constrained the airplane fuel tank capacity from 0 to 182 liters that prevents the estimated fuel level from being negative**.

The likelihood function was based solely upon the fuel sensor reading of 34 liters and its Gaussian distribution with a standard deviation of 20 liters. This resulted in the left tail of the likelihood function extending into negative fuel levels. The probability of negative fuel was calculated from this distribution, which resulted in calculating the original probability of negative fuel in the tank as 4.5% (i.e., the cumulative probability of fuel being less than 0 liters). In contrast, in this Bayesian update using a Bayes Monte Carlo method, the physically based prior was included, which constrained the airplane fuel tank levels to a realistic capacity from 0 to 182 liters. This omitted unrealistic, negative fuel levels in the tank. The prior was defined as uniform for fuel tank values between 0 liters and 182 liters, but as a value of 0 for any negative fuel values. As a result of the prior assigning a probability of 0 to negative fuel in the tank, the posterior probability of the fuel level being a negative value was also 0. Thus, the resulting probability of negative fuel in the airplane tank was calculated as 0.0% (in a similar manner to task 4), which resolved the issue of unrealistic fuel estimates in the airplane tank.

It is also worth noting that the expected available fuel value (i.e., the mean) was calculated as 36 liters of fuel in both the grid-based method and the Bayes Monte Carlo method. However, the most likely value of available fuel was calculated as 34 liters in the grid-based method and as 35.2 liters in the Bayes Monte Carlo method. In the grid-based Bayesian update, the possible fuel levels (within the constraints of the prior) were discretized into a finite number of points. The posterior probability was subsequently calculated at these discrete points and the mode was determined as the fuel level with the highest posterior probability. In contrast in the Bayes Monte Carlo method, 10 million random samples were generated based upon the posterior distribution, which produced a smoother and more continuous estimation of the distribution. The density estimate was subsequently calculated from the randomly generated samples, and the mode was identified as the value where the estimated density function was at its maximum. Thus, the difference in the most likely value of available fuel estimate between the grid-based and Monte Carlo methods arises primarily from the kernel density estimation used in the Monte Carlo method, which can shift the mode slightly compared to the discrete grid-based method.

TASK 6:

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

In the context of this problem, **a simple analytical Kalman filter solution** refers to using a linear Kalman filter to estimate the true level of fuel in the airplane tank based upon 1) noisy sensor measurements and 2) a predictable model of airplane fuel consumption. More broadly, a Kalman filter is comprised of a prediction step and an update step. The prediction step uses the model of fuel consumption to estimate the state of the system at the next time step and the update step incorporates new noisy sensor data to correct and refine the estimated fuel level.

For a **simple analytical Kalman filter solution** to this problem, several key assumptions must be made. Broadly, **one assumption of the Kalman filter** is that the system evolves linearly over time, where the state of the system at the next time step is a linear function of the current state and the control inputs. In the context of this problem, this assumption is stating that the relationship between the fuel sensor reading and the actual usable level of fuel in the airplane tank follows a linear equation over time in which the airplane fuel level at the next time step depends on the current fuel level minus the fuel consumed (with a small uncertainty term). This assumption is important to consider because the Kalman filter works well in linear systems because it can efficiently calculate the next state using matrix multiplications. This assumption is not entirely realistic in the context of this problem because real-world fuel usage in an airplane is not linear at all times. Although it may be roughly linear over a short period of time, it is not linear at all times due to external factors such as: 1) the presence of turbulence or wind resistance while flying (which would increase fuel consumption), 2) changes in thrust (i.e., descending compared to ascending), and 3) changes in engine efficiency due to wear or long-term usage.

Building upon the concepts addressed in the first assumption, a **second assumption of the Kalman filter** is that the next fuel level is dependent only upon the current fuel level and that past fuel levels have no impact on the next fuel level when the current fuel level is known. At every time step in the Kalman filter, the fuel level is predicted based upon the previous state and the estimate of the fuel level is updated using the most recent sensor reading. This assumption is important because without this assumption, the Kalman filter would have to track all prior fuel level measurements, which is inefficient for the real-time processing of fuel levels. This second assumption is mostly realistic because airplane fuel consumption often follows a relatively basic equation in which past fuel estimates do not have to be stored or remembered. However, in some real-world situations this assumption may be violated. For instance, if there is delayed fuel consumption in the airplane (i.e., perhaps due to engine dynamics periodically or inconsistently causing a small amount of lag in fuel consumption), past fuel estimates may influence the current fuel level. In addition, if the airplane fuel gauge overestimates or underestimates the fuel level due to sensor drift, past fuel readings may be relevant to determining the next fuel level.

As a **third assumption**, the Kalman filter assumes that there is an accurate initial estimate of the airplane fuel state. This means that the Kalman filter requires an initial estimate of the fuel level (i.e., fuel sensor reading of 34 liters) and an initial estimate of the uncertainty in the fuel level (i.e., a standard deviation of 20 liters). Without an accurate initial estimate of the airplane fuel state, the Kalman filter may produce unrealistic fuel estimates, converge slowly, and/or make large, incorrect early predictions of the fuel state. This third assumption is important because the Kalman filter is reliant upon prior knowledge to make an accurate initial fuel estimate, as a poor initial estimate may lead to high error propagation in early steps. This third assumption is not entirely realistic because several issues can occur. For instance, the fuel gauge may not truly reflect the actual initial amount of fuel in the tank if the fuel sensor is faulty, old, or damaged.

A **fourth assumption of the Kalman filter** is that all uncertainties (including uncertainty in the fuel consumption rate and measurement noise from the fuel sensor) follow a normal, Gaussian distribution. This assumption is important because the Kalman filter derives its equations assuming a Gaussian distribution, as this facilitates easier mathematical manipulation and optimal estimations due to reducing variance in the error (uncertainty). This assumption is primarily realistic, as most sources of error fall within a normal Gaussian distribution based upon the Central Limit Theorem. However, there are also some cases in which real-world noise and uncertainty do not follow a normal, Gaussian distribution. For instance, the fuel sensor may completely fail (i.e., get stuck), or there may be a sudden, significant fuel leakage in the airplane, or engine shutoff, that may not always follow a normal Gaussian distribution.

TASK 7:

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?**

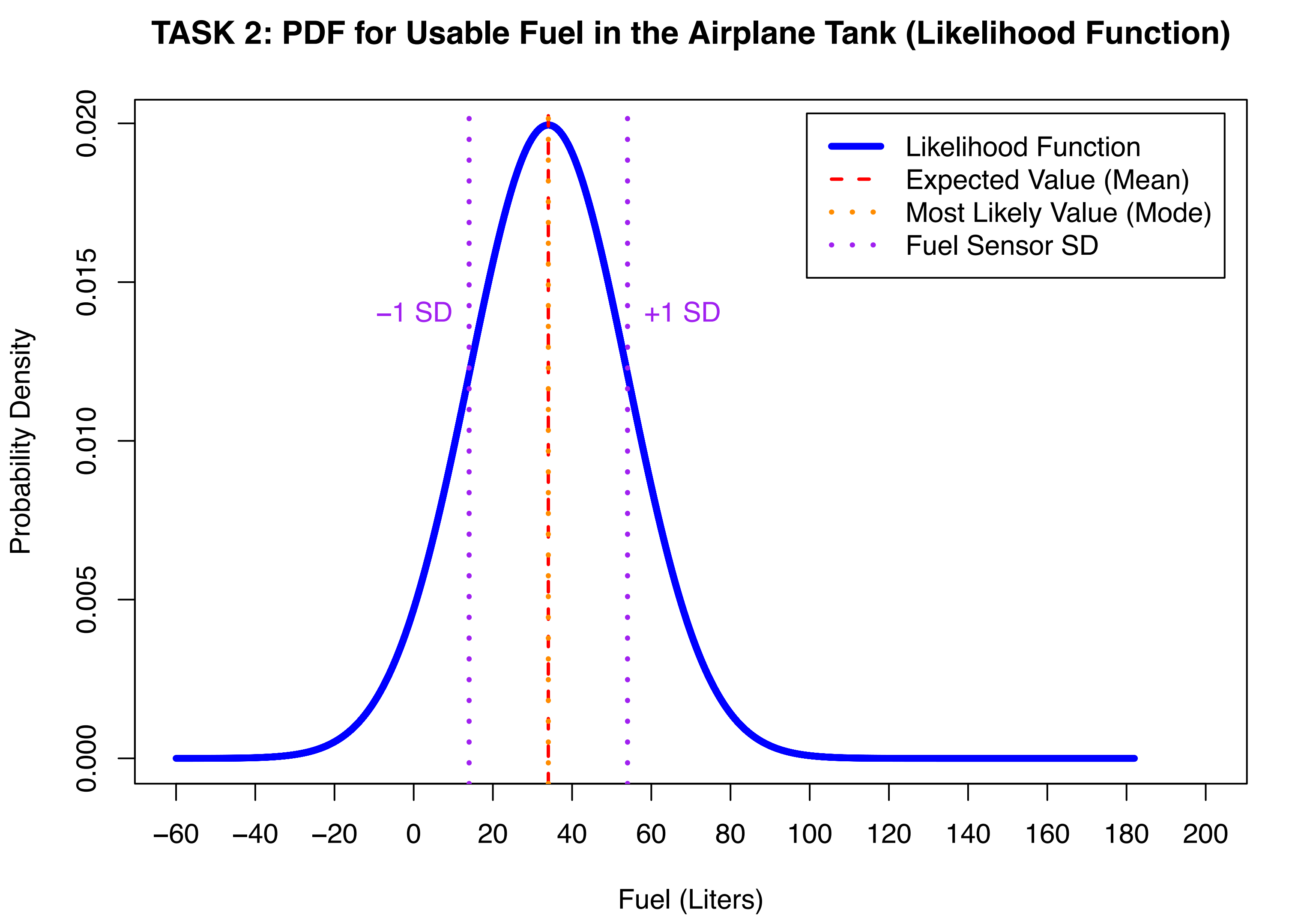
- The probability of making it to the airport that is 100 minutes flight time away with at least 30 minutes of reserve fuel = 42%

B. **What is the probability that you run out of fuel trying to make it to this airport?**

- The probability of running out of fuel before reaching the airport = 39.4%

FIGURES FOR EACH TASK:

Please note that this section of the PDF document was produced from the code script in R



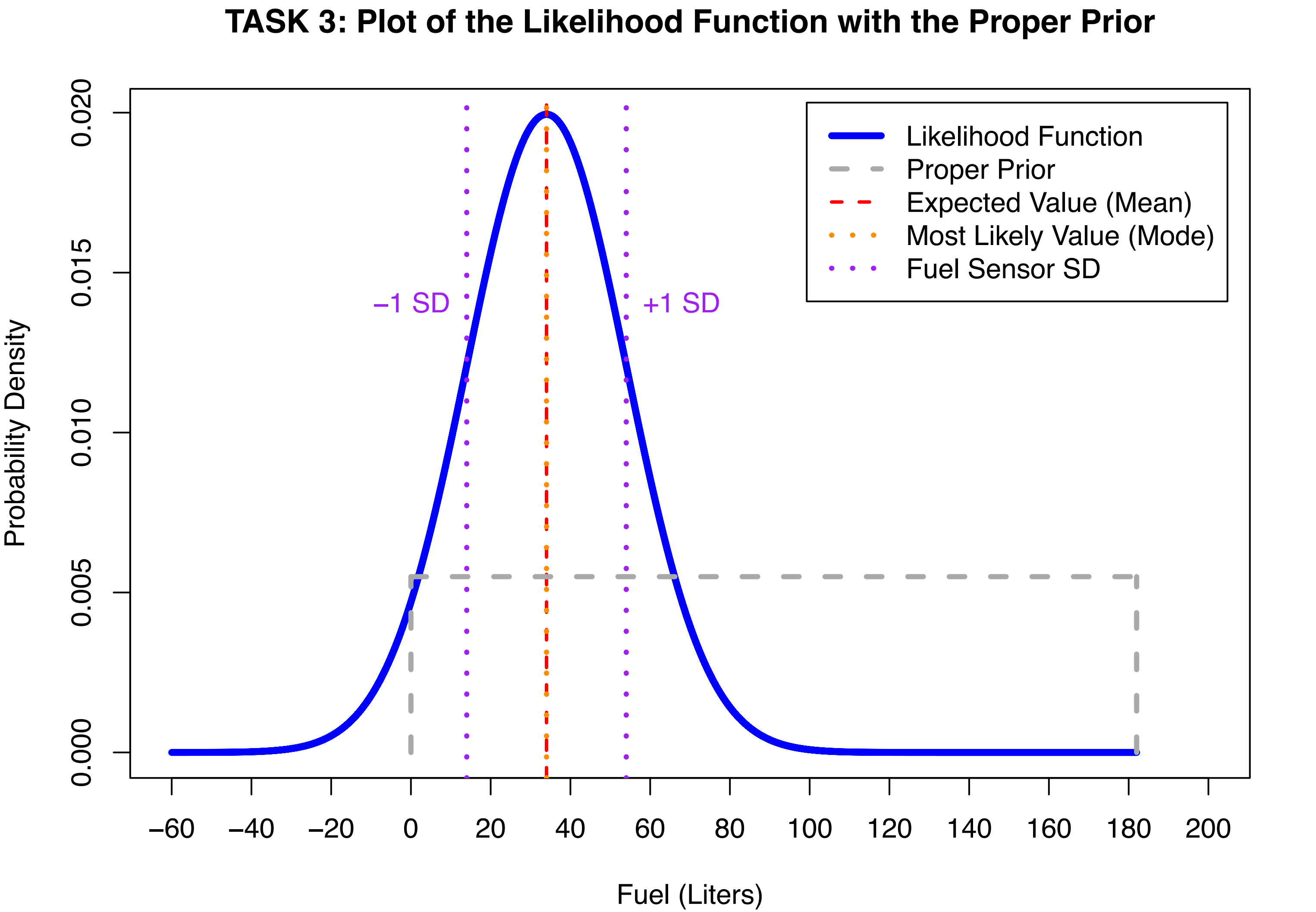
TASK 2: PDF for Usable Fuel in the Airplane Tank (Likelihood Function)
Likelihood Function
Expected Value (Mean)
Most Likely Value (Mode)
Fuel Sensor SD
−1 SD
+1 SD
Probability Density
0.000
0.005
0.010
0.015
0.020
−60
−40
−20
0
20
40
60
80
100
120
140
160
180
200
Fuel (Liters)

TASK 2: Estimates from the PDF of Usable Fuel in the Tank

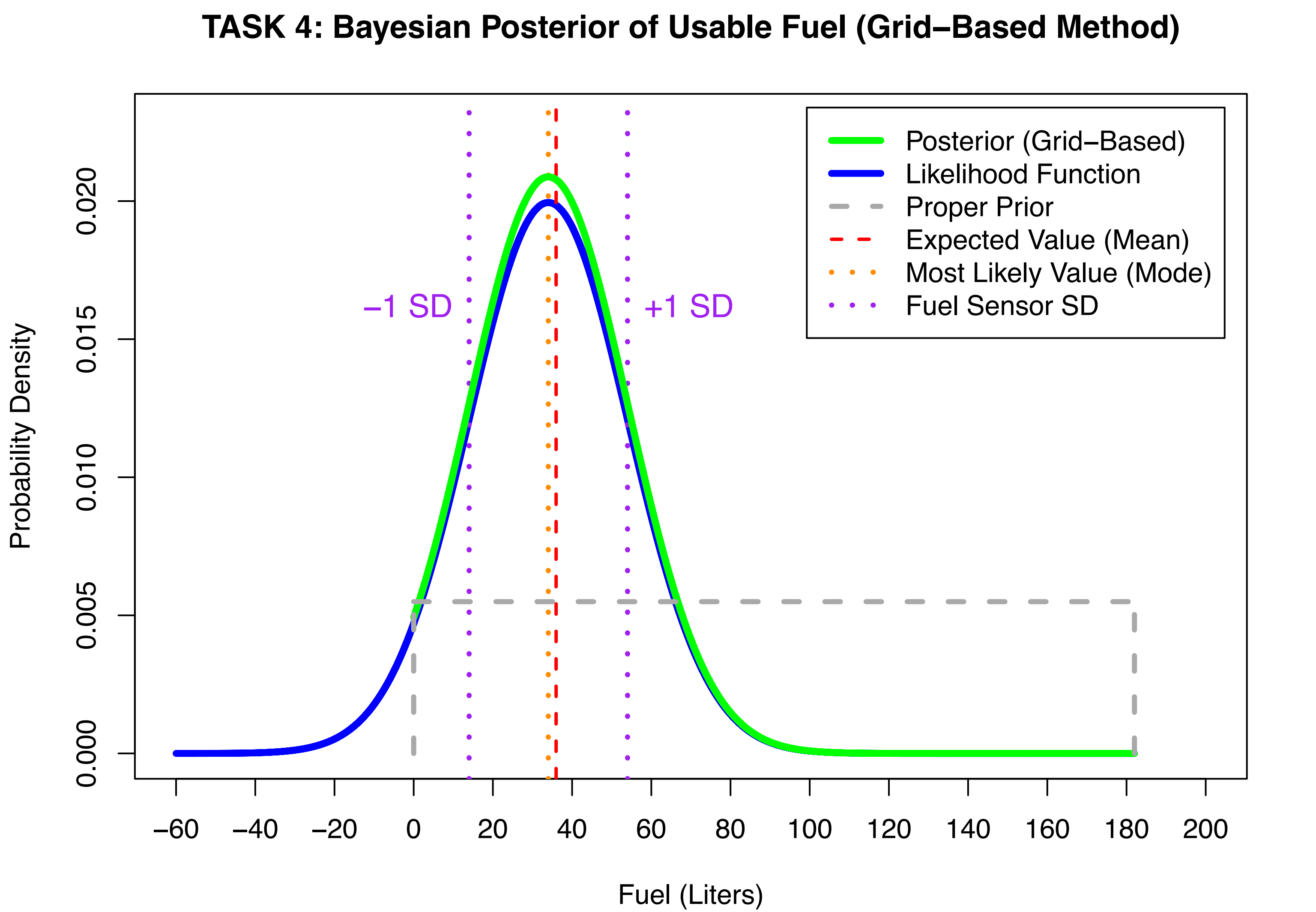
Expected Value of Available Fuel (Mean): 34 liters

Most Likely Value of Available Fuel (Mode): 34 liters

Probability of Negative Fuel in the Tank: 4.5 %

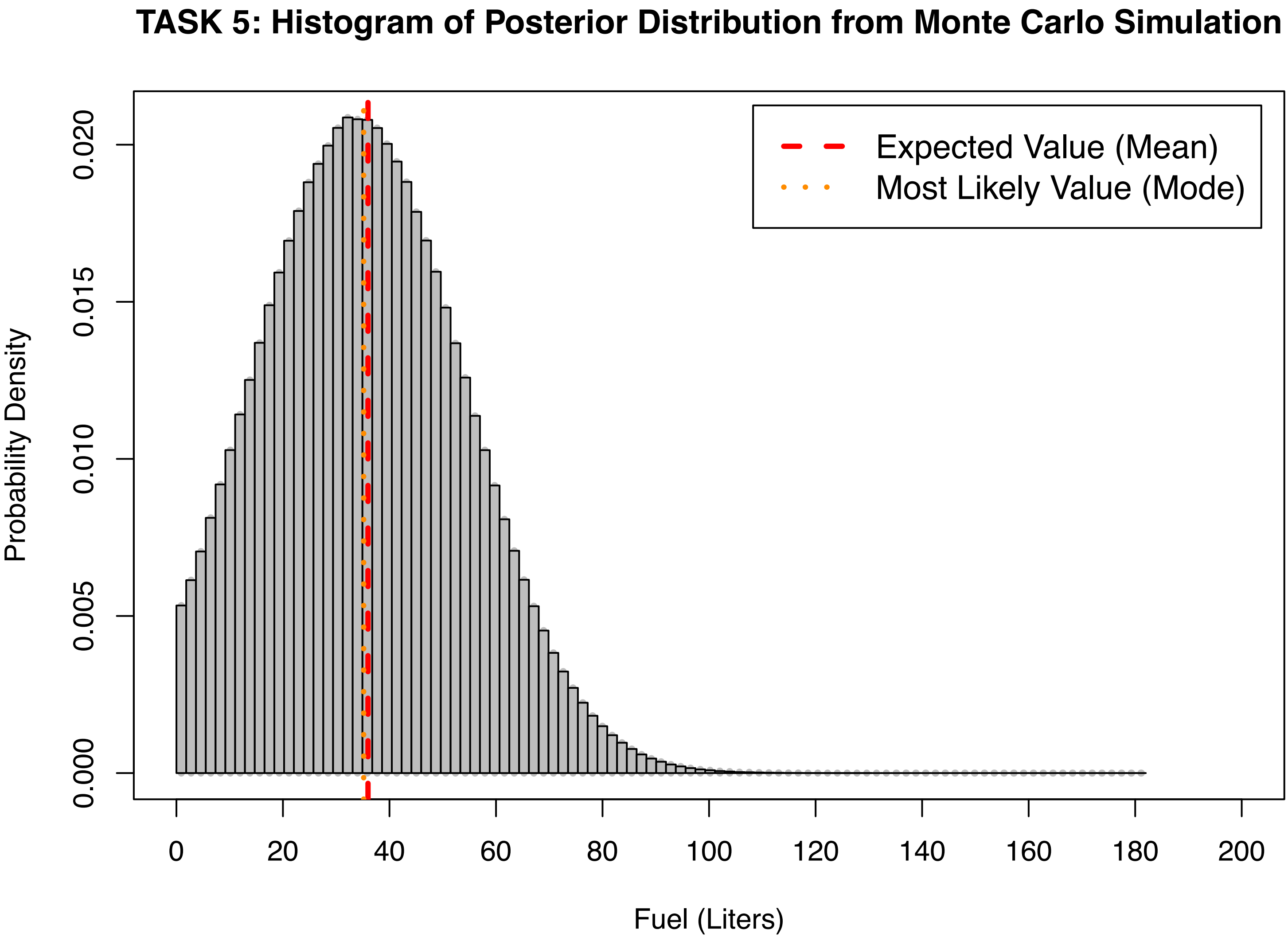


TASK 3: Plot of the Likelihood Function with the Proper Prior
Likelihood Function
Proper Prior
Expected Value (Mean)
Most Likely Value (Mode)
Fuel Sensor SD
−1 SD
+1 SD
Probability Density
0.000
0.005
0.010
0.015
0.020
−60
−40
−20
0
20
40
60
80
100
120
140
160
180
200
Fuel (Liters)

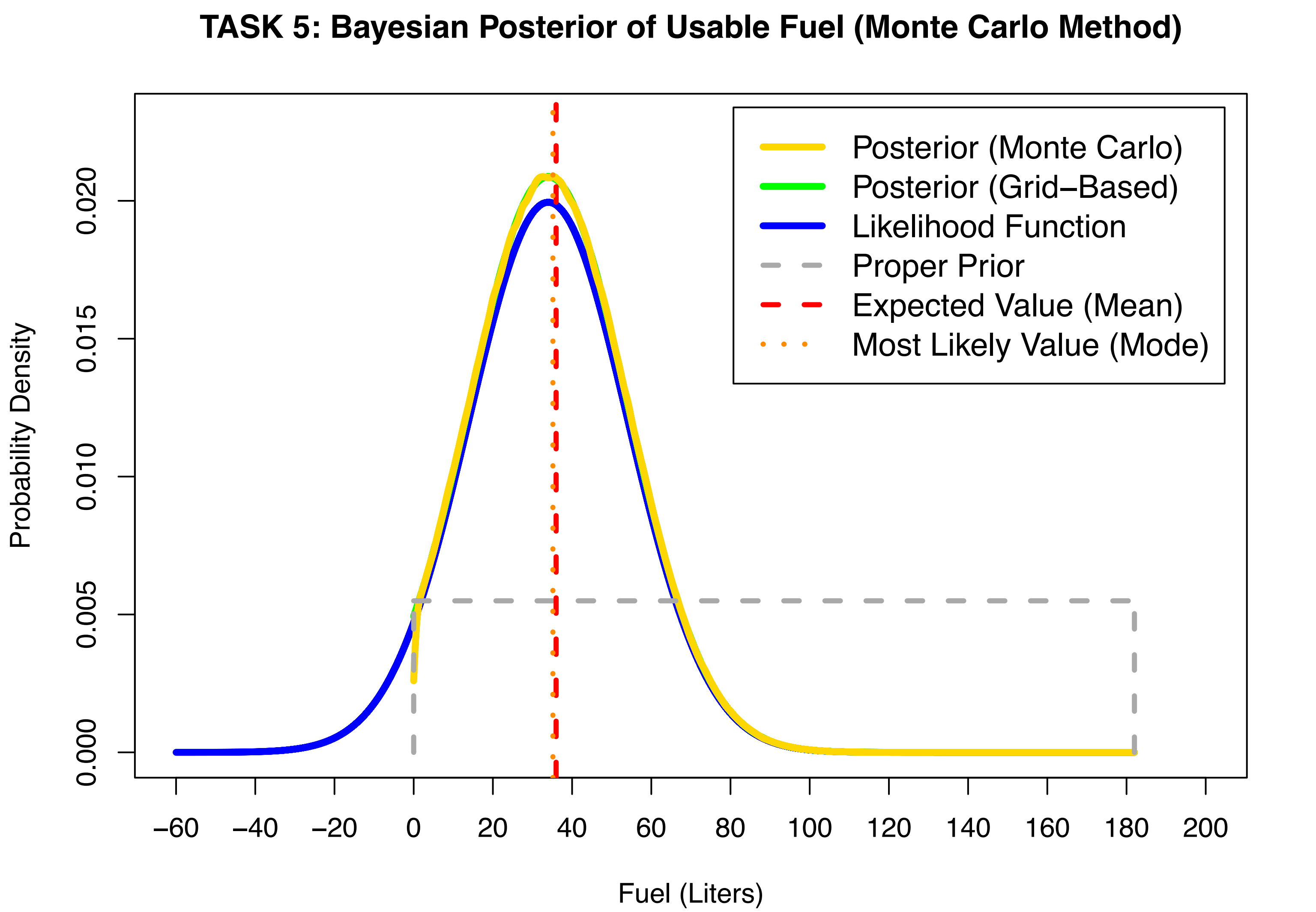

TASK 4: Bayesian Posterior of Usable Fuel (Grid–Based Method)
Posterior (Grid–Based)
Likelihood Function
Proper Prior
Expected Value (Mean)
Most Likely Value (Mode)
Fuel Sensor SD
–1 SD
+1 SD
Probability Density
0.000
0.005
0.010
0.015
0.020
Fuel (Liters)
–60
–40
–20
0
20
40
60
80
100
120
140
160
180
200

TASK 4: Estimates after Bayesian Update using Grid–Based Method

Expected Value of Available Fuel (Mean): 36 liters
Most Likely Value of Available Fuel (Mode): 34 liters
Probability of Negative Fuel in the Tank: 0 %



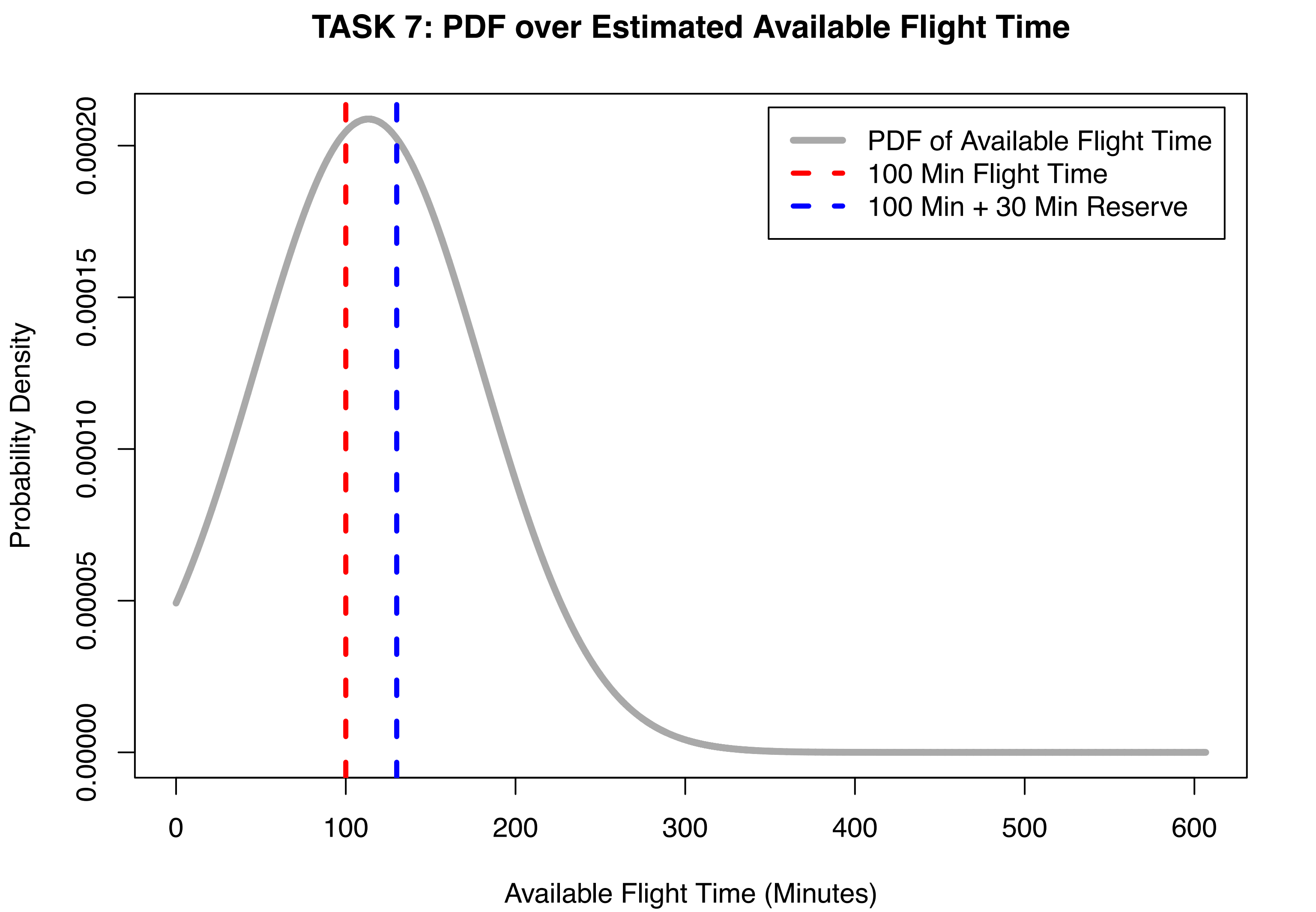
TASK 5: Histogram of Posterior Distribution from Monte Carlo Simulation
Expected Value (Mean)
Most Likely Value (Mode)
Probability Density
0.000
0.005
0.010
0.015
0.020
0
20
40
60
80
100
120
140
160
180
200
Fuel (Liters)



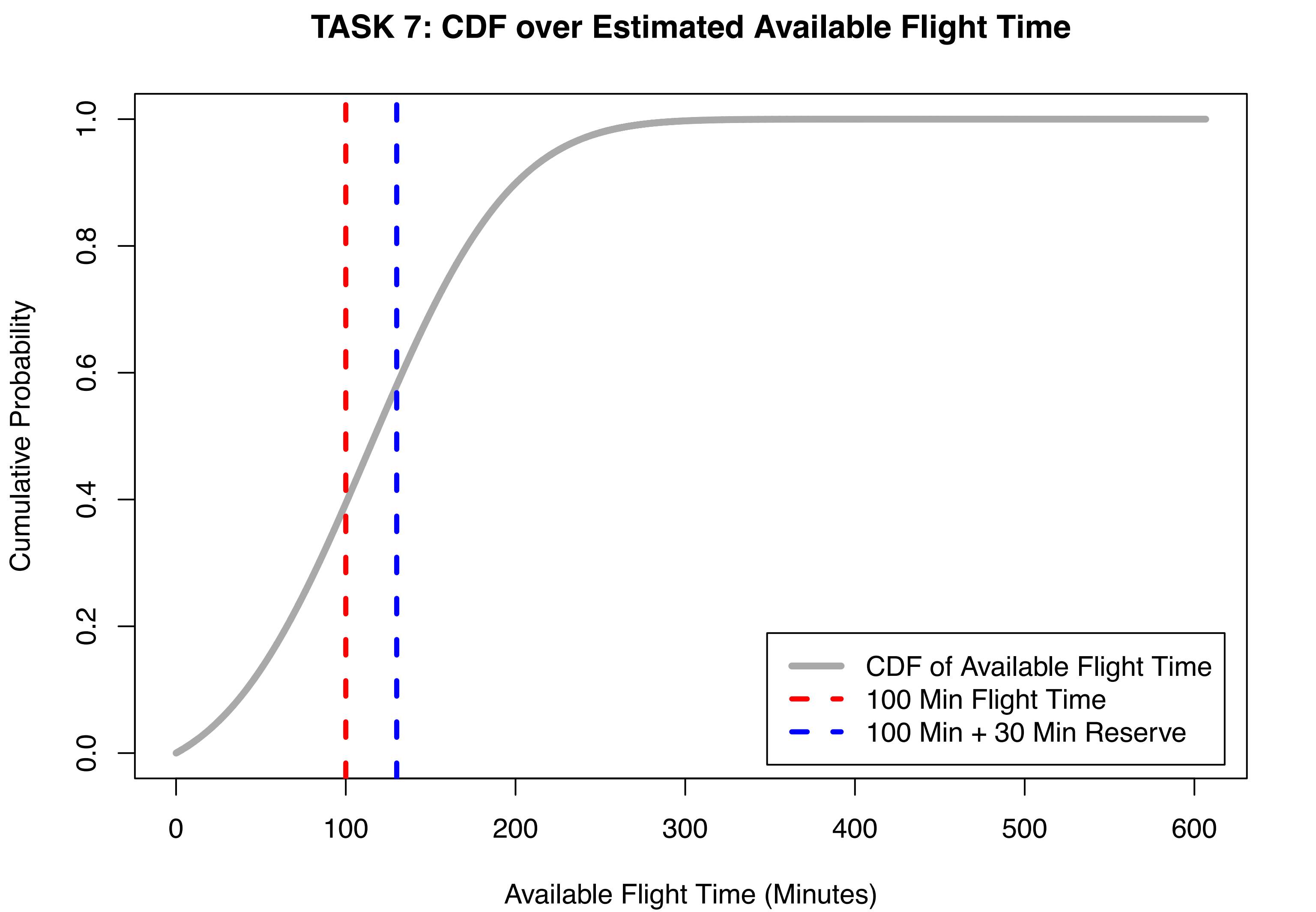
TASK 5: Bayesian Posterior of Usable Fuel (Monte Carlo Method)
Posterior (Monte Carlo)
Posterior (Grid–Based)
Likelihood Function
Proper Prior
Expected Value (Mean)
Most Likely Value (Mode)
Probability Density
0.000
0.005
0.010
0.015
0.020
−60
−40
−20
0
20
40
60
80
100
120
140
160
180
200
Fuel (Liters)

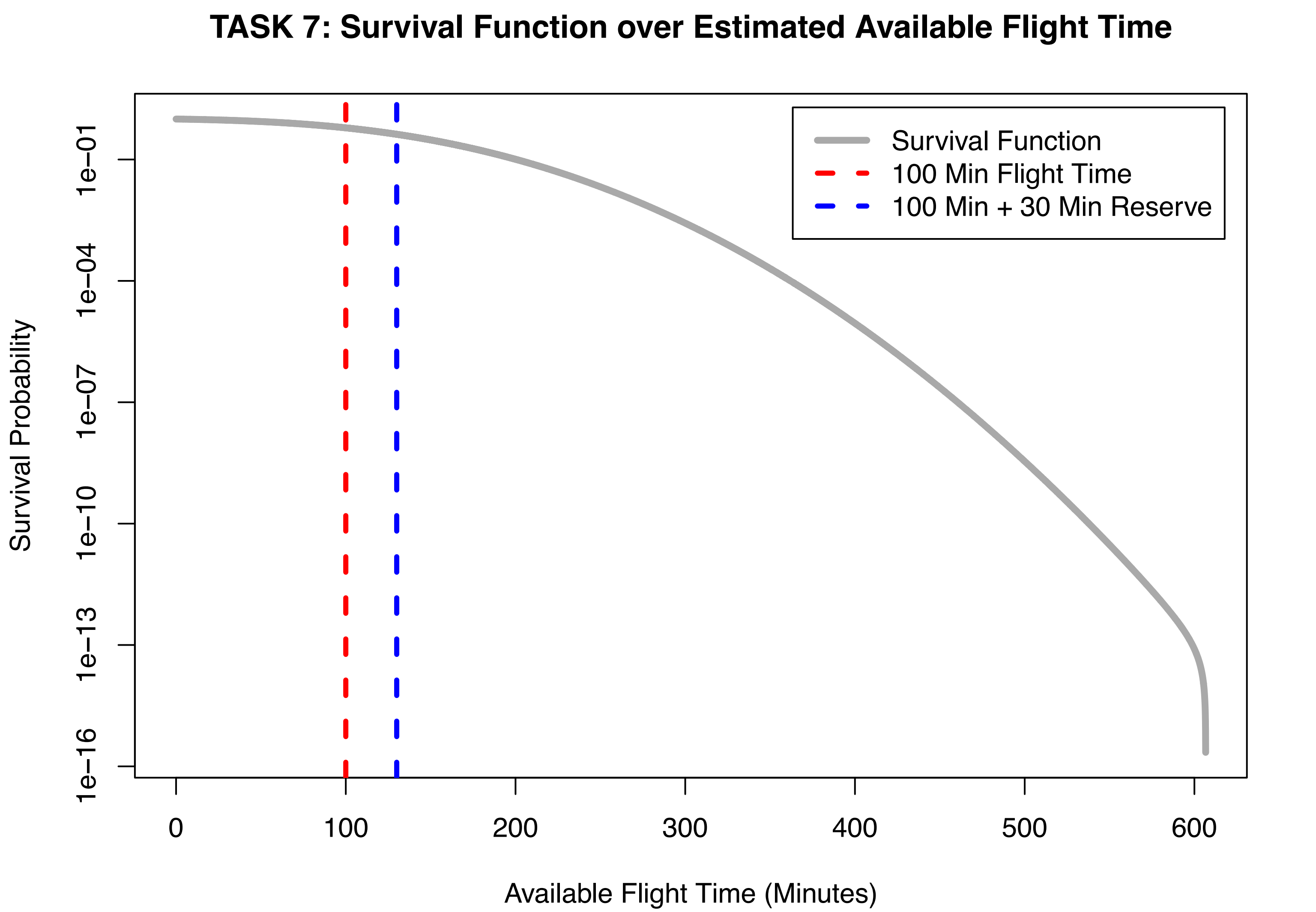
TASK 5: Estimates after Bayesian Update using Monte Carlo Method

Expected Value of Available Fuel (Mean): 36 liters
Most Likely Value of Available Fuel (Mode): 35.2 liters
Probability of Negative Fuel in the Tank: 0 %



TASK 7: PDF over Estimated Available Flight Time
PDF of Available Flight Time
100 Min Flight Time
100 Min + 30 Min Reserve
Probability Density
0.00000
0.00005
0.00010
0.00015
0.00020
0
100
200
300
400
500
600
Available Flight Time (Minutes)


TASK 7: CDF over Estimated Available Flight Time
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
600
Available Flight Time (Minutes)
CDF of Available Flight Time
100 Min Flight Time
100 Min + 30 Min Reserve



TASK 7: Survival Function over Estimated Available Flight Time
Survival Probability
1e−01
1e−04
1e−07
1e−10
1e−13
1e−16
0
100
200
300
400
500
600
Available Flight Time (Minutes)
Survival Function
100 Min Flight Time
100 Min + 30 Min Reserve

TASK 7: Probabilities from the Plot of Estimated Available Flight Time

Probability of making it to the airport with reserve fuel: 42 %
Probability of running out of fuel before reaching the airport: 39.4 %

CHOICES:

To address each of the tasks, several key choices were made in the process of writing the code script in R. **One choice** was the decision to initially set the lower bound for the range of fuel levels in the tank to -60 liters when plotting the likelihood function in Task 2. A plausible alternative choice could have been to set the lower bound to a larger negative value such as -100 liters or to a smaller negative value such as -20 liters. Setting the lower bound to -100 liters may have been an unnecessarily large negative number to ensure that the left tail of the distribution was not truncated. In contrast, setting the lower bound to -20 liters would have truncated the left tail of the likelihood function, which may have led to an inaccurate initial estimate for the expected value of available fuel in the tank. Thus, I decided to select -60 liters as my lower bound for plotting of the likelihood function because I thought that this was a sufficiently large enough negative number to ensure that the left tail of the distribution would not be truncated.

In the process of defining a broad range of fuel levels in the airplane tank, a **second choice** that I made was to set the step (or increment) size to 0.001 (i.e., by = 0.001). A plausible alternative could have been to set the step size to a larger value such as 0.01 or a smaller value such as 0.0001, but I opted to set the step size to 0.001 because I thought that this step size would not significantly sacrifice accuracy, while still maintaining good computational efficiency.

In the plotting of the likelihood function, a **third choice** that I made was to remove the default x-axis labels, specify my own upper and lower x-axis limits, and add custom x-axis labels to improve the readability of the plot. An alternative choice could have been to use the default x-axis labels in base R plotting, but this would have undesirably omitted the left tail of the likelihood function from view on the plot. Thus, to ensure that the full likelihood function was visible on the plot, I opted to remove the default x-axis labels, specify my own custom x-axis range of values from -60 to 200, and set custom labels on the x-axis in increments of 20.

A **fourth choice** that I made was to set a single random seed for reproducibility of the results. A plausible alternative could have been to set multiple random seeds to introduce further controlled randomness and reproducibility of the estimated results. A less plausible alternative could have been to set no seed at all; however, this would have eliminated the reproducibility of the results. Thus, for the sake of simplicity, I decided to set only a single fixed seed in my code script.

In the process of defining a physically based proper prior, a **fifth choice** that I made was to constrain the range of possible fuel levels to a physically realistic range of 0 liters (the minimum) to 182 liters (the maximum capacity), because negative fuel levels are not physically possible. An alternative choice could have been to set the range of possible fuel levels to a negative value such as -20 liters, but this would have been a poor decision because negative fuel levels are not physically possible, and thus this would have been a poor prior. Another alternative choice could have been to maintain the lower bound of the range of possible fuel levels at -60 liters (as was the case for the likelihood function). However, the problem with this is that the posteriors would have not been distinct from the likelihood function on the plot, as the prior would not have introduced any new beneficial and realistic prior knowledge or information to achieve more realistic fuel estimates. This also would have been a poor choice because negative fuel levels are not possible. Thus, I decided to set the range of possible fuel levels for the proper prior from 0 to 182 liters of fuel, as this introduced a realistic range of possible fuel levels for this problem.

Also in terms of setting a proper prior, a **sixth choice** that I made was to use a truncated uniform prior within the range of 0 liters to 182 liters, which assumed that all values in this plausible range were equally likely before incorporating the fuel sensor reading. One plausible alternative could have been to use a beta distribution prior, which could have been used to favor likely values and suppress extreme or less likely values (i.e., a completely full or empty tank). Another plausible alternative could have been to use a normal prior centered around a typical refueling level, such as 70% or 75% of the full capacity of the tank. However, I decided to use a truncated uniform prior for simplicity and because I felt that it was realistic to assume that all values within the range of possible fuel levels were equally probable of occurring.

Also in the process of setting a proper prior, a **seventh choice** that I made was to assign a probability of zero to fuel levels outside of the specified range in my proper prior. An alternative choice could have been to assign a non-zero probability to negative fuel levels outside of my specified range, however, this may have led to Bayesian updates that may have suggested a non-negligible chance of negative fuel in the tank, which is physically impossible. Thus, I decided to set the probability to zero for fuel levels outside of my specified range in the prior to ensure that the posterior estimates remained within the physically realistic bounds of my proper prior.

An **eighth choice** that I made was to normalize the posteriors to sum to a value of 1 to ensure that they formed a valid probability distribution. A less plausible alternative choice could have been to not normalize the posteriors; however, this would have resulted in the posteriors retaining their shape but not being directly interpretable as probability density functions, (as probabilities can only range between 0 and 1). Thus, I decided to normalize the posteriors to sum to 1 to ensure that the posteriors could be interpreted as probability density functions (PDFs) that accurately reflect probability proportions.

In terms of determining the Bayesian update from the proper prior and likelihood function using a Bayes Monte Carlo method, a **ninth choice** that I made was to set the maximum number of Monte Carlo samples to 10 million. A plausible alternative choice could have been to set the maximum number of samples to a lower number such as 1 million, but this would have reduced the accuracy of the estimates from the posterior despite being more computationally efficient. Conversely, another plausible alternative choice could have been to increase the maximum number of samples to 100 million. Although this would have improved the accuracy of the estimates, this would have been very computationally expensive and may have increased the amount of time to run the code script. Despite these alternatives, I decided to set the maximum number of samples for the Monte Carlo simulation to 10 million, as I thought that this was a sufficiently large enough number of samples to acquire relatively accurate posterior estimates, while not being too computationally expensive.

Along these same lines, a **tenth choice** that I made was to introduce a stopping criterion in Task 5 as a post-processing check to ensure that the Monte Carlo simulation did not exceed the maximum number of samples (10 million). An alternative choice could have been to establish no stopping criterion, but this could result in the code not truncating extra samples in the instance of the Monte Carlo simulation exceeding the maximum number of samples. Thus, I decided that integrating a stopping criterion into this code script was a wise decision to ensure computational efficiency while still allowing for the calculations of relatively accurate posterior estimates.

Also in terms of setting up the Bayes Monte Carlo method, an **eleventh choice** that I made was to perform importance sampling with replacement. An alternative choice could have been to sample without replacement, however, if this choice was made, the posterior would have been limited to the exact values present in the prior sample set. In addition, by setting a large number of samples (i.e., 10 million), the method could have run out of available prior samples if sampling was performed without replacement. Therefore, I decided to perform importance sampling with replacement, because 1) each sample can be drawn multiple times (ensuring a stable posterior estimate), 2) the sampling process reconstructs the posterior density through emphasizing values with higher likelihood, and 3) frequently drawing high-likelihood values produces a smoother approximation of the continuous posterior.

In terms of plotting the results for Task 7, a **twelfth choice** that I made was to set the y-axis of the survival function plot to a logarithmic (log) scale. An alternative choice could have been to simply use a linear scale (like the other plots), however, I decided to use a logarithmic scale because the logarithmic scale better represents small probability values.

Broadly in terms of plotting the results for each of the tasks in this problem set, I made many small stylistic choices in terms of color selection, fonts, character size, axes labels, plot titles, legend formatting, and labeling key features on the plots (i.e., the expected value of available fuel, the most likely value of available fuel, the standard deviation from the fuel sensor reading, etc.). Although a nearly infinite number of alternative choices exist in the preparation of these plots, my primary objective was to maximize readability and clarity to ensure that the key takeaways from each plot could be easily identified.

NEGLECTED UNCERTAINTIES:

Provided below is a discussion of several sources of uncertainty that were neglected in this analysis, and their impact on the conclusions:

One primary source of uncertainty in this analysis arises from the choice of the seed. In this case, I decided to set a single, fixed seed value of 210 to facilitate reproducibility of the results. Despite enabling reproducibility of the results each time the code script is run using a seed value of 210, different results can be obtained when a different integer value is set as the seed. In terms of impact on the conclusions, choosing a different integer as the seed value may result in changes to the estimates of the expected value of available fuel in the tank and the most likely value of available fuel for the two Bayesian updates (grid-based method and Monte Carlo method), and it may also change the calculated probability estimates in task 7 from the plot of the estimated available flight time. One approach for partially resolving this source of uncertainty is to set multiple random seed values (i.e., 3-5 different fixed seeds) and to perform the analyses for each seed value and compare the results to examine the impact of different seeds on the calculations.

A **<u>second source of uncertainty</u>** in this analysis arises from the <u>proper prior</u>, which was set to function as a truncated uniform prior with a realistic range from 0 liters to 182 liters. As a truncated uniform prior, this prior assumed that all fuel levels within this specified range were treated as equally likely. This assumption is a source of uncertainty in this analysis, because fuel levels are rarely evenly distributed. Instead, operational fuel levels in airplanes are often grouped within specific ranges due to standardized refueling practices, fuel reserve policies, and flight planning strategies. In terms of impact on the conclusions, if the prior does not accurately reflect the true distribution of fuel levels, the resulting posterior distributions for the grid-based and Bayes Monte Carlo methods may be skewed and the estimates may not be as accurate as possible if the prior is not fully integrating all previous knowledge of airplane fuel levels.

A **<u>third source of uncertainty</u>** in this analysis arises from the number of random samples (i.e., the sample size) taken from the posterior distribution in the Bayes Monte Carlo simulation (Task 5). For this task, the Bayes Monte Carlo simulation drew a large but finite number of random samples up to the value of 10 million. This random sampling process is a source of uncertainty because it is subject to sampling bias, discretization errors, and a lack of convergence diagnostics. It would be ideal to set an infinitely large maximum number of random samples for the Monte Carlo simulation to enable convergence and achieve highly accurate estimates, however this is not computationally feasible. In terms of impacting the conclusions, this source of uncertainty can result in the estimates for the expected value of available fuel and the most likely value of available fuel not being as accurate as possible. It can also result in underestimating the probability of fuel shortages, which is a source of concern.

In terms of the provided inputs, **<u>an additional source of uncertainty</u>** arises from the assumption that the airplane fuel consumption rate is approximated as uncertain with a Gaussian standard deviation. This assumption may not fully account for the uncertainty arising from <u>large variations in the airplane fuel consumption rate</u> arising from factors including: 1) fuel leaks, 2) strong wind resistance and turbulence, 3) engine malfunctions or engine failure during flight, 4) bad weather, or 5) large changes in altitude to avoid turbulence, etc. In terms of the impact on the conclusions, neglecting the uncertainty arising from potentially large variations in the airplane fuel consumption rate may lead to undesirable over- or underestimations of the fuel consumption rate if the conditions vary considerably from the assumed Gaussian distribution. In this case, the probability of reaching a specific airport or the estimated available flight time may be incorrectly overestimated, which is a potential source for major concern.

Also pertaining to the provided inputs, **<u>another source of uncertainty</u>** arises from the assumption that the fuel sensor has an error approximated as following a Gaussian distribution (with a relatively large standard deviation). Despite a large standard deviation range, this assumption may not explicitly account for <u>the uncertainty of the fuel sensor measurement potentially having a systematic bias</u>, which may cause the fuel sensor to regularly overestimate or underestimate the fuel levels in the airplane tank. In terms of impact on the conclusions, if the fuel sensor is consistently biased towards over- or underestimating the fuel level, the expected value of available fuel from the posterior distribution (i.e., the mean) may be shifted, leading to inaccurate estimates of the usable fuel in the tank. In addition, the probability of negative fuel in the tank may also be incorrectly estimated, depending on if the bias is negative or positive.

REPRODUCIBILITY OF THE ANALYSES:

Due to a few distinct elements integrated into the code script, the analyses for each of the tasks in this problem set are reproducible.

First and most importantly, a single random seed value of 210 was set and integrated at the top of the code script to ensure that any stochastic processes within the code script (i.e., the Monte Carlo simulation etc.) produce the same result each time that the code script is run.

A second element that facilitates reproducibility of the analyses was the clearly defined input values at the top of the code script (i.e. the fuel sensor reading and its standard deviation, the fuel consumption rate and its standard deviation etc.). Explicitly defining these input parameters ensures that these values remain fixed and consistent each time that the code script is run.

Third, using a grid-based method to determine the Bayesian update ensures reproducible results across different runs of the code script because the grid-based method does not rely upon random sampling (in contrast to the Monte Carlo method).

Fourth, utilizing standard, deterministic probability functions such as pnorm (for cumulative probability) or dnorm (for a Gaussian distribution) introduce inherent reproducibility of the results when the same inputs are used.

Fifth, introducing a fixed, proper prior with a constrained range for the airplane fuel tank to a realistic range of 0 liters to 182 liters (the maximum tank capacity) also aided to some extent in achieving reproducibility of the Bayesian updates in both the grid-based and Monte Carlo methods, as the introduction of this prior ensured that the posteriors were offset from the likelihood function due to the truncation of the left tail of the distribution at 0 liters of fuel.

REFERENCES (CITATIONS)

1. Short, T. (2004). *R Reference Card.* Retrieved from https://cran.r-project.org/doc/contrib/Short-refcard.pdf.
2. Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2024). *An Introduction to R.* Retrieved from https://intro2r.com/Rbook.pdf.
3. Sriver, R., & Keller, K. (2024). *coin-example.R.* Retrieved from https://canvas.dartmouth.edu/courses/70487/files/folder/example%20codes?preview=13458702.
4. Applegate, P. J., Sriver, R. L., Garner, G. G., Bakker, A., Alley, R. B., & Keller, K. (2016). *Risk Analysis in the Earth Sciences: A Lab Manual with Exercises in R (Version 1.2).* SCRiM. Retrieved from https://leanpub.com/raes.
5. Grolemund, G. (2014). *Hands-On Programming with R: Write Your Own Functions and Simulations.* O'Reilly Media.
6. Venables, W. N., Smith, D. M., & the R Core Team. (2024). *An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics Version 4.4.2.* Retrieved from https://cran.r-project.org/doc/manuals/R-intro.pdf.
7. Verzani, J. (2002). *simpleR – Using R for Introductory Statistics.* Retrieved from https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf.
8. Rizzo, M. L. (2019). *Statistical Computing with R* (2nd ed.). Chapman & Hall/CRC.
9. Chang, W. (2013). *R Graphics Cookbook: Practical Recipes for Visualizing Data.* O'Reilly Media.
10. Murrell, P. (2005). *R Graphics* (2nd ed.). Chapman & Hall/CRC.
11. Gillespie, C., & Lovelace, R. (2016). *Efficient R Programming: A Practical Guide to Smarter Programming.* O'Reilly Media.
12. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). CRC Press.
13. Bernardo, J. M., & Smith, A. F. M. (2000). *Bayesian Theory.* Wiley.
14. Jaynes, E. T. (2003). *Probability Theory: The Logic of Science.* Cambridge University Press.
15. Robert, C. P., & Casella, G. (2004). *Monte Carlo Statistical Methods* (2nd ed.). Springer.
16. Wickham, H. (2019). *Advanced R* (2nd ed.). CRC Press.
17. Welch, G. & Bishop, G. (2006). *An Introduction to the Kalman Filter*. Retrieved from https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.
18. Kim, Y. & Bang, H. (2018). *Introduction to Kalman Filter and its Applications*. Retrieved from https://www.intechopen.com/chapters/63164.
19. Anderson, J. D. (2010). *Introduction to Flight* (7th ed.). McGraw-Hill.
20. Federal Aviation Administration (FAA). *Pilot's Handbook of Aeronautical Knowledge (FAA-H-8083-25B).* United States Department of Transportation. Retrieved from https://www.faa.gov/aviation/phak/pilots-handbook-aeronautical-knowledge-faa-h-8083-25b.
21. Yan, L. & Cain, J. (2020). *Central Limit Theorem*. Retrieved from https://web.stanford.edu/class/archive/cs/cs109/cs109.1212/lectureNotes/LN18_clt.pdf.

22. Biswal, A. (2024). *An In-Depth Explanation of Cumulative Distribution Function*. Retrieved from https://www.simplilearn.com/tutorials/statistics-tutorial/cumulative-distribution-function.
23. Chen, Y.C. (2018). *Lecture 5: Survival Analysis*. Retrieved from https://faculty.washington.edu/yenchic/18W_425/Lec5_survival.pdf
24. Ross, S.M. (2006). Simulation (4th ed). Academic Press.

Attached below is the link to access the files in my ENGS 107 GitHub Repository:
https://github.com/isaiah-richardson28/Dartmouth-ENGS-107-Winter-2025

APPENDIX:

***Note: When copied and pasted into RStudio the code script below produces the PDF file**

Course: ENGS 107: Bayesian Statistical Modeling and Computation
Problem Set 3: Analytical Methods / Bayes Monte Carlo / Grid Methods
Professor: Dr. Klaus Keller PhD
Due Date: Friday, February 14, 2025 at 11:59 pm
File name: [isaiah.d.richardson.th@dartmouth.edu].ps#3.R
Software: R and RStudio
#____
Author: Isaiah D. Richardson, M.S. (IDR)
Copyright: copyright by the author
License: this code is distributed under the GNU GENERAL PUBLIC LICENSE v3.0
License: for more information regarding GNU GENERAL PUBLIC LICENSE Version 3 visit: https://www.gnu.org/licenses/why-not-lgpl.html
There is no warranty on this R code script for ENGS 107 Question 4A
#____
Version 1: last changes made on: February 7, 2025
#____
Sources:
- Personal correspondence with Dr. Klaus Keller during in-person office hours to conceptually discuss the distinction between the prior, likelihood function, and posterior(s) for this problem set, the purpose of the prior, and how to graph the results (01/31/2025; 02/03/2025; 02/05/2025; 02/07/2025)
- Personal correspondence with TA Siddhi Gothivrekar during office hours in-person and over Zoom to discuss and clarify the expectations for each of the tasks, how to graph the results, and whether or not the posterior PDF should overlap with the likelihood PDf for tasks 4 and 5 (01/31/2025; 02/03/2025)
- R help files accessed through R-studio for syntax
- Short, T. (2004). R Reference Card. For understanding how to use functions and commands in R including: seq, dnorm, pnorm, xaxt, xlim, ylim, abline, bty, lty, lwd, ifelse, max, text, plot, cex, $, font.main, cex.main
- Douglas, A., Roos, D., Mancini, F., Couto, A. & Lusseau, D. (2024). An Introduction to R. For understanding how to produce plots and histograms in R using the base R plot commands.
- The coin-example.R from Canvas for general formatting of the code script and layout of the header notes
- Applegate, P.J., Keller, K. et al.(2016). Risk Analysis in the Earth Sciences: A Lab Manual with Exercises in R. For understanding the RStudio interface and basic functions in R
- Grolemund, G. Hands-On Programming with R. For understanding the RStudio interface and basic functions in R

- Venables, W.N., Smith, D.M. & the R Core Team. (2024). An Introduction to R Notes on R: A Programming Environment for Data Analysis and Graphics Version 4.4.2 for understanding functions, commands, and the basics of how to plot results
- Verzani, J. (2002). simpleR - Using R for Introductory Statistics. Used for understanding notation, functions, and formatting of the code script
- Rizzo, M.L. (2019). Statistical Computing with R (2nd edition). For understanding syntax, functions, and how to perform a Monte Carlo simulation for task 5
- Chang, W. (2013). R Graphics Cookbook: Practical Recipes for Visualizing Data. O'Reilly Media. For understanding the syntax, functions, formatting, and basics for producing plots and visualizing data in R.
- Murrell, P. (2005). R Graphics. Chapman & Hall/CRC. For understanding the basics of how to plot results visually and textually in R.
- Gillespie, C. & Lovelace, R. (2016). Efficient R Programming: A Practical Guide to Smarter Programming. O'Reilly Media. For better understanding the RStudio interface, and optimizing the code structure and performance.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis (3rd ed.). CRC Press. For deepening my conceptual understanding of grid-based methods for Bayesian updating
- Bernardo, J. M., & Smith, A. F. M. (2000). Bayesian Theory. Wiley. To gain a conceptual understanding of grid-based methods.
- Jaynes, E. T. (2003). Probability Theory: The Logic of Science. Cambridge University Press. For understanding uniform and truncated uniform priors while I was workin to establish my proper prior.
- Robert, C. P., & Casella, G. (2004). Monte Carlo Statistical Methods (2nd ed.). Springer. For understanding how to set up a Monte Carlo simulation for task 5
- Wickham, H. (2019). Advanced R (2nd ed.). CRC Press. For understanding functions and how to perform a Monte Carlo simulation for task 5
- Biswal, A. (2024). An In-Depth Explanation of Cumulative Distribution Function. To gain a basic understanding of the cumulative distribution function (CDF)
- Chen, Y.C. (2018). Lecture 5: Survival Analysis. To gain a fundamental understanding of the survival function
- Ross, S.M. (2006). Simulation (4th ed). Academic Press. For understanding the basics of convergence and stopping criteria for a Bayes Monte Carlo simulation
#____
__
Collaborators:
Emma Vejcik - Conceptually discussed the expectations for tasks 2 and 3. Discussed how to calculate the three estimates for task 2. Worked together to try to conceptually understand the meaning of a proper prior for task 3 (01/28/2025)
#____
__
To run this code script file:
1. Open the ascii file in R
2. Use the cursor to highlight the entire code script
3. Press either 'Source', 'Ctrl + Enter', or 'run' to run the entire code script
4. Open the resulting PDF file to analyze the results

```
#===========================================================================
===========
#### THE PROBLEM ####
# You are part of a team that is refining a safety system for an airplane that can land an airplane
automatically
# at the next available and safe airport in case of an emergency.
# The computer system makes a decision on which airport to choose to land based on factors
such as wind, runway length, legal
# requirements, fuel use, and available fuel. The team asks you to design, test, and document a
draft computer program that takes
# as input a reading from a fuel sensor (together with some other information) and produces an
estimate of the usable fuel in the
# tank with an estimate of the associated uncertainties.
#===========================================================================
===========
#### THE INPUTS ####

# Specifying the total fuel tank capacity (182 liters)
total_fuel_tank_capacity <- 182

# Specifying the digital fuel sensor standard deviation (20 liters)
# Note: The digital fuel sensor has an error approximated as following a Gaussian distribution
with a standard deviation of 20 liters
fuel_sensor_standard_deviation <- 20

# Specifying the fuel sensor reading (34 liters)
fuel_sensor_reading <- 34

# Specifying the airplane fuel consumption rate (18 liters per hour)
# Note: This value is approximated as uncertain with a Gaussian standard deviation of 2 liters per
hour
airplane_fuel_consumption_rate <- 18

# Specifying the standard deviation value for the airplane fuel consumption rate
# Note: 18 liters approximated as uncertain with a Gaussian standard deviation of 2 liters per
hour
airplane_fuel_consumption_standard_deviation <- 2
#___________________________________________________________________________
__

# Setting a seed to enable reproducibility of the results
set.seed(210)
#===========================================================================
===========
#### TASK 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 the probability of negative fuel in the tank.
Do these estimates make sense to you?
#===

# Defining a broad range of fuel levels in the airplane tank
# Note: The seq function generates a sequence of numbers; -60 indicates the starting value of the sequence up to the total fuel tank capacity of 182 liters
# Note: It is obviously not possible to have a negative amount of fuel in the tank. -60 was selected as a lower limit for the PDF to prevent truncation of the left tail of the PDF for task 2
# Note: by = 0.001 specifies the step size (or increment) from -60 to 182
usable_fuel_in_tank <- seq(-60, total_fuel_tank_capacity, by = 0.001)

# Using the dnorm function to calculate the probability density function (PDF) of usable fuel in the airplane tank based on a Gaussian distribution
pdf_usable_fuel <- dnorm(usable_fuel_in_tank, mean = fuel_sensor_reading, sd = fuel_sensor_standard_deviation)
#____
CALCULATING THE ESTIMATES FOR TASK 2

# Calculating the expected value of available fuel in the tank (*this is the mean)
expected_value_available_fuel <- sum(usable_fuel_in_tank * pdf_usable_fuel) / sum(pdf_usable_fuel)

# Calculating the most likely value of available fuel in the tank (*this is the mode)
most_likely_value_available_fuel <- usable_fuel_in_tank[which.max(pdf_usable_fuel)]

# Using the pnorm function to calculate the probability of negative fuel in the tank
probability_negative_fuel_in_tank <- pnorm(0, mean = fuel_sensor_reading, sd = fuel_sensor_standard_deviation)
#____
PREPARING THE PDF FILE FOR REPORTING THE RESULTS

# Specifying the PDF file name
pdf_filename <- "ENGS_107_Problem_Set_3_Results.pdf"

# Opening a PDF file for plotting; also specifying the width and height
pdf(pdf_filename, width = 8, height = 6)
#____

PLOTTING THE PROBABILITY DENSITY FUNCTION (PDF) FOR TASK 2 (THE LIKELIHOOD FUNCTION)

```
# Plotting the probability density function (PDF) for task 2
# Note: type = 'l' specifies a line plot; lwd specifies the line width
plot(usable_fuel_in_tank, pdf_usable_fuel, type = "l", col = "blue", lwd = 4,

    # Specifying the header/title for the plot
    main = "TASK 2: PDF for Usable Fuel in the Airplane Tank (Likelihood Function)",

    # Specifying the x-axis label
    xlab = "Fuel (Liters)",

    # Specifying the y-axis label
    ylab = "Probability Density",

    # Removing the default x-axis labels
    xaxt = "n",

    # Specifying the upper and lower limits of the x-axis to extend the x-axis improve readability
of the likelihood function
    xlim = c(-60, 200))

# Adding custom labels to the x-axis to enhance the readability of the plot
# Note: Specifying the bounds and labeled increments on the x-axis
axis(1, at = seq(-60, 200, by = 20), labels = seq(-60, 200, by = 20))

# Adding a vertical line to the plot denoting the expected value of available fuel (*the mean)
# Note: lwd specifies the line width; lty = 2 specifies the line type as dashed
abline(v = expected_value_available_fuel, col = "red", lwd = 2, lty = 2)

# Adding a vertical line denoting the most likely value of available fuel (*the mode)
# Note: lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = most_likely_value_available_fuel, col = "darkorange", lwd = 3, lty = 3)

# Adding vertical lines to denote the fuel sensor standard deviation
# Note: lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = c(fuel_sensor_reading - 20, fuel_sensor_reading + 20), col = "purple", lwd = 3, lty =
3)

# Adding text to label the + 1 and - 1 standard deviation marks + 20 liters and - 20 liters from 34
liters (the fuel sensor reading) on the plot
text(fuel_sensor_reading - 20, max(pdf_usable_fuel) * 0.7, "-1 SD", pos = 2, col = "purple")
text(fuel_sensor_reading + 20, max(pdf_usable_fuel) * 0.7, "+1 SD", pos = 4, col = "purple")

# Adding a legend to label the key features of the probability density function (PDF) plot
```

```
# Note: lwd specifies the line width, lty specifies the line type (1 = solid, 2 = dashed, 3 = dotted)
# Note: bty = 'o' draws a complete box (this is the default); box.col specifies the box color; inset
offsets the box from the plot border
legend("topright", legend = c("Likelihood Function", "Expected Value (Mean)", "Most Likely
Value (Mode)", "Fuel Sensor SD"),
       col = c("blue", "red", "darkorange", "purple"), lwd = c(4, 2, 3, 3), lty = c(1, 2, 3, 3), bty =
"o", box.col = "black", inset = 0.02)

#___________________________________________________________________________
__

# Opening a new plot to print the estimates calculated above on the second page of the PDF file
plot.new()

# Adding a title at the top of the results page of the PDF file
title(main = "TASK 2: Estimates from the PDF of Usable Fuel in the Tank", font.main = 2,
cex.main = 1.2)

text(0.4, 0.8, paste(

  # Expected Value of Available Fuel (Mean) and rounded to 2 decimal places
  "Expected Value of Available Fuel (Mean): ", round(expected_value_available_fuel, 2),
"liters\n",

  # Most likely value of available fuel (mode) and rounded to 2 decimal places
  "Most Likely Value of Available Fuel (Mode): ", round(most_likely_value_available_fuel, 2),
"liters\n",

  # Probability of negative fuel in the tank, rounded to 1 decimal place and specified as a
percentage
  "Probability of Negative Fuel in the Tank: ", round(probability_negative_fuel_in_tank * 100,
1), "%\n"
), cex = 1.2)

#=================================================================
===========
#### TASK 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)
#=================================================================
===========

# Creating a variable to specify the minimum realistic fuel capacity in the airplane tank as 0 liters
of fuel
minimum_fuel_tank_capacity <- 0
```

```
# Determining the height of the uniform prior distribution that constrains the airplane fuel level
between 0 liters and 182 liters
# Note: The total probability must integrate to 1, so the height is being set accordingly
# Note: The uniform prior assumes that all fuel levels are equally likely within the realistic range
of 0 - 182 liters
prior_height <- 1 / (total_fuel_tank_capacity - minimum_fuel_tank_capacity)

# Defining the prior probability distribution over the entire range of usable fuel in the tank
# Note: This distribution assigns a constant probability density within the realistic fuel range and
a probability of zero to values outside of the range
prior <- ifelse(usable_fuel_in_tank >= minimum_fuel_tank_capacity & usable_fuel_in_tank <=
total_fuel_tank_capacity, prior_height, 0)

# Creating a sequence of fuel values for plotting the prior
fuel_range <- seq(minimum_fuel_tank_capacity, total_fuel_tank_capacity, length.out = 100)

# Calculating the prior probability values corresponding to each fuel level
# Note: There is a constant prior probability for all valid fuel levels
prior_values <- rep(prior_height, length(fuel_range))
#___________________________________________________________________________
__

# Plotting the proper prior with the likelihood function for task 3
# Note: type = 'l' specifies a line plot; lwd specifies the line width; lty = 1 specifies the line type
as solid
plot(usable_fuel_in_tank, pdf_usable_fuel, type = "l", col = "blue", lwd = 4, lty = 1,

    # Specifying the header/title for the plot
    main = "TASK 3: Plot of the Likelihood Function with the Proper Prior",

    # Specifying the x-axis label
    xlab = "Fuel (Liters)",

    # Specifying the y-axis label
    ylab = "Probability Density",

    # Removing the default x-axis labels
    xaxt = "n",

    # Specifying the upper and lower limits of the x-axis to extend the x-axis improve readability
of the likelihood function
    xlim = c(-60, 200))

# Adding custom labels to the x-axis to enhance the readability of the plot
# Note: Specifying the bounds and labeled increments on the x-axis
```

axis(1, at = seq(-60, 200, by = 20), labels = seq(-60, 200, by = 20))

# Adding a vertical line to the plot denoting the expected value of available fuel (*the mean)
# Note: lwd specifies the line width; lty = 2 specifies the line type as dashed
abline(v = expected_value_available_fuel, col = "red", lwd = 2, lty = 2)

# Adding a vertical line denoting the most likely value of available fuel (*the mode)
# Note: lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = most_likely_value_available_fuel, col = "darkorange", lwd = 3, lty = 3)

# Adding vertical lines to denote the fuel sensor standard deviation
# Note: lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = c(fuel_sensor_reading - 20, fuel_sensor_reading + 20), col = "purple", lwd = 3, lty = 3)

# Adding text to label the + 1 and - 1 standard deviation marks + 20 liters and - 20 liters from 34 liters (the fuel sensor reading) on the plot
text(fuel_sensor_reading - 20, max(pdf_usable_fuel) * 0.7, "-1 SD", pos = 2, col = "purple")
text(fuel_sensor_reading + 20, max(pdf_usable_fuel) * 0.7, "+1 SD", pos = 4, col = "purple")

# Adding vertical dashed lines to denote the bounds for the proper prior
segments(x0 = minimum_fuel_tank_capacity, y0 = 0, x1 = minimum_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)
segments(x0 = total_fuel_tank_capacity, y0 = 0, x1 = total_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)

# Adding a horizontal dashed line for the proper prior between 0 liters and 182 liters to create a rectangle
segments(x0 = minimum_fuel_tank_capacity, y0 = prior_height, x1 = total_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)

Adding a legend to label the key features of the probability density function (PDF) plot
Note: lwd specifies the line width, lty specifies the line type (1 = solid, 2 = dashed, 3 = dotted)
Note: bty = 'o' draws a complete box (this is the default); box.col specifies the box color; inset offsets the box from the plot border
legend("topright", legend = c("Likelihood Function", "Proper Prior", "Expected Value (Mean)", "Most Likely Value (Mode)", "Fuel Sensor SD"),
    col = c("blue", "darkgray", "red", "darkorange", "purple"), lwd = c(4, 3, 2, 3, 3), lty = c(1, 2, 2, 3, 3), bty = "o", box.col = "black", inset = 0.02)

#===

TASK 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?
#==========================================================================
===========

# Renaming the pdf_usuable_fuel variable as 'likelihood' for clarity in the remainder of the code
script
# Note: The likelihood function represents the probability of observing the data given a specific
fuel level
likelihood <- pdf_usable_fuel

# Calculating the posterior probability using Bayes' Theorem (without normalization)
# Note: posterior is proportional to the product of the prior * likelihood
posterior <- likelihood * prior

# Ensuring that no probability mass is assigned to negative fuel values as this physically
unrealistic
posterior[usable_fuel_in_tank < 0] <- 0

# Normalizing the posterior so that it sums to 1. The purpose of this is to turn it into a proper
probability distribution
# Note: The normalization factor is the sum of all weighted probabilities (considering grid
spacing)
posterior <- posterior / sum(posterior * diff(usable_fuel_in_tank)[1])
#__________________________________________________________________________
__

# Calculating the expected value of available fuel (the mean) from the Bayesian update using the
grid-based method
posterior_expected_value <- sum(usable_fuel_in_tank * posterior *
diff(usable_fuel_in_tank)[1])

# Calculating the most likely value of available fuel (the mode) from the Bayesian update using
the grid-based method
posterior_most_likely_value <- usable_fuel_in_tank[which.max(posterior)]

# Calculating the probability of negative fuel in the airplane tank from the Bayesian update using
the grid-based method
# Note: This value should be 0 as a result of integrating the proper prior written into the code
script under task 3
posterior_probability_negative_fuel <- sum(posterior[usable_fuel_in_tank < 0] *
diff(usable_fuel_in_tank)[1])
#__________________________________________________________________________
__
#### PLOTTING THE POSTERIOR FROM THE GRID-BASED METHOD WITH THE
LIKELIHOOD FUNCTION ####
```

```r
# Adjusting the y-axis limit to ensure that both the likelihood and posterior distributions are visible on the plot
# Note: Multiplied by 1.1 to add some buffer on the y-axis
max_density <- max(c(pdf_usable_fuel, posterior)) * 1.1

# Plotting the posterior of the Bayesian update from the grid-based method with the likelihood function
# Note: type = 'l' specifies a line plot; lwd specifies the line width; lty = 1 specifies the line type as solid
plot(usable_fuel_in_tank, pdf_usable_fuel, type = "l", col = "blue", lwd = 4, lty = 1,

    # Specifying the header/title of the plot
    main = "TASK 4: Bayesian Posterior of Usable Fuel (Grid-Based Method)",

    # Specifying the x-axis label
    xlab = "Fuel (Liters)",

    # Specifying the y-axis label
    ylab = "Probability Density",

    # Removing the default x-axis labels
    xaxt = "n",

    # Specifying the upper and lower limits of the x-axis to extend the x-axis to ensure that the entire likelihood function is visible on the plot
    xlim = c(-60, 200),

    # Specifying the y-axis limits to ensure that both the likelihood function and posterior are visible on the plot
    ylim = c(0, max_density))

# Adding custom x-axis labels to ensure that the entire likelihood function is visible on the plot
axis(1, at = seq(-60, 200, by = 20), labels = seq(-60, 200, by = 20))

# Including a vertical line to denote the expect value of available fuel calculated from the posterior
# Note: lwd specifies the line width; lty = 2 specifies the line type as dashed
abline(v = posterior_expected_value, col = "red", lwd = 2, lty = 2)

# Including a vertical line to denote the most likely value of available fuel calculated from the posterior
# Note: lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = posterior_most_likely_value, col = "darkorange", lwd = 3, lty = 3)
```

```
# Including a vertical line to denote + 1 and - 1 standard deviations from the fuel sensor reading of 34 liters
# Note: the standard deviation is 20 liters; lwd specifies the line width; lty = 3 specifies the line type as dotted
abline(v = c(fuel_sensor_reading - 20, fuel_sensor_reading + 20), col = "purple", lwd = 3, lty = 3)

# Adding text to label the vertical purple, dotted lines denoting the standard deviation range of +/- 20 liters from the fuel sensor reading of 34 liters
text(fuel_sensor_reading - 20, max_density * 0.7, "-1 SD", pos = 2, col = "purple", cex = 1.2)
text(fuel_sensor_reading + 20, max_density * 0.7, "+1 SD", pos = 4, col = "purple", cex = 1.2)

# Overlaying the posterior distribution from the grid-based method on the plot with the likelihood function and the prior
# Note: This only includes valid (non-negative) fuel values (i.e., fuel values between 0 liters and 182 liters)
# Note: lwd specifies the line width and lty = 1 specifies the line type as solid
valid_indices <- usable_fuel_in_tank >= 0
lines(usable_fuel_in_tank[valid_indices], posterior[valid_indices], col = "green", lwd = 4, lty = 1)

# Adding vertical dashed lines to denote the bounds of the proper prior
segments(x0 = minimum_fuel_tank_capacity, y0 = 0, x1 = minimum_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)
segments(x0 = total_fuel_tank_capacity, y0 = 0, x1 = total_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)

# Adding a horizontal dashed line for the proper prior between 0 liters and 182 liters to create a rectangle
segments(x0 = minimum_fuel_tank_capacity, y0 = prior_height, x1 = total_fuel_tank_capacity, y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)

# Including a legend in the upper right corner of the plot to denote the key aspects on the plot
# Note: bty = 'o' places a box around the legend; box.col specifies the box color as black; cex = 1.0 specifies the character size (font size)
legend("topright", legend = c("Posterior (Grid-Based)", "Likelihood Function", "Proper Prior", "Expected Value (Mean)", "Most Likely Value (Mode)", "Fuel Sensor SD"),
       col = c("green", "blue", "darkgray", "red", "darkorange", "purple"), lwd = c(4, 4, 3, 2, 3, 3), lty = c(1, 1, 2, 2, 3, 3),
       bty = "o", box.col = "black", inset = 0.02, cex = 1.0)
#__________________________________________________________________________________

# Opening a new plot to print the estimates calculated above on the fourth page of the PDF file
plot.new()
```

```
# Adding a title at the top of the results page
title(main = "TASK 4: Estimates after Bayesian Update using Grid-Based Method", font.main =
2, cex.main = 1.2)

text(0.4, 0.8, paste(

  # Expected Value of Available Fuel (mean) and rounded to 1 decimal place
  "Expected Value of Available Fuel (Mean): ", round(posterior_expected_value, 1), "liters\n",

  # Most likely value of Available Fuel (mode) and rounded to 2 decimal places
  "Most Likely Value of Available Fuel (Mode): ", round(posterior_most_likely_value, 2),
"liters\n",

  # Probability of negative fuel in the tank, rounded to 1 decimal place, and specified as a
percentage
  "Probability of Negative Fuel in the Tank: ", round(posterior_probability_negative_fuel * 100,
1), "%\n"
), cex = 1.2)

#=====================================================================
===========
#### TASK 5 ####
# Repeat the step above using a Bayes Monte Carlo method
#=====================================================================
===========

# Specifying the maximum number of Monte Carlo samples (10 million)
maximum_iterations <- 10000000

# Generating samples from the proper prior
# Note: Assuming that there is a uniform distribution within the specified airplane tank capacity
of 0 liters to 182 liters
# Note: Also ensuring that all sampled values are within the specified fuel range
prior_samples <- runif(maximum_iterations, min = 0, max = total_fuel_tank_capacity)

# Calculating the likelihood for each prior sample (assuming a normal distribution centered at the
fuel sensor reading)
likelihood_values <- dnorm(prior_samples, mean = fuel_sensor_reading, sd =
fuel_sensor_standard_deviation)

# Calculating the posterior weights using Bayes' Theorem (i.e., the posterior is proportional to
likelihood * prior)
# Note: The weights are also normalized
posterior_weights <- likelihood_values / sum(likelihood_values)
```

```
# Implementing a stopping condition to ensure that the Monte Carlo simulation does not exceed
the maximum number of samples (10 million)
current_iterations <- length(prior_samples)
if (current_iterations > maximum_iterations) {
  prior_samples <- prior_samples[1:maximum_iterations]
  posterior_weights <- posterior_weights[1:maximum_iterations]
}

# Introducing a variable called sampling size to ensure that resampling from the posterior does
not exceed the maximum number of samples (10 million)
sampling_size <- min(current_iterations, maximum_iterations)

# Re-sampling from the posterior using importance sampling
# Note: replace = TRUE specifies to perform sampling with replacement
posterior_samples <- sample(prior_samples, size = sampling_size, replace = TRUE, prob =
posterior_weights)
#________________________________________________________________________________
__

# Calculating the expected value of available fuel in the airplane tank from the Bayesian update
using the Monte Carlo simulation
expected_value_monte_carlo <- mean(posterior_samples)

# Calculating the most likely value of available fuel in the airplane tank from the Bayesian
update using the Monte Carlo simulation
most_likely_value_monte_carlo <-
density(posterior_samples)$x[which.max(density(posterior_samples)$y)]

# Calculating the probability of negative fuel in the tank from the Bayesian update using the
Monte Carlo simulation
probability_negative_fuel_monte_carlo <- sum(posterior_samples < 0) /
length(posterior_samples)
#________________________________________________________________________________
__
#### PRODUCING A HISTOGRAM PLOT OF THE POSTERIOR DISTRIBUTION FROM
THE MONTE CARLO SIMULATION ####

# Creating histogram data for the posterior distribution from the Monte Carlo simulation
# Note: length.out creates a sequence of 100 evenly spaced breakpoints between 0 and 182 liters;
plot = FALSE prevents the histogram from being plotted before normalization
histogram_data <- hist(posterior_samples, breaks = seq(0, total_fuel_tank_capacity, length.out =
100), plot = FALSE)

# Normalizing the histogram density so that the total area sums to a value of 1
bin_width <- diff(histogram_data$breaks)[1]
```

```
normalized_histogram_density <- histogram_data$counts / (sum(histogram_data$counts) *
bin_width)

# Plotting the normalized histogram from the Monte Carlo simulation
# Note: type = 'h' specifies a histogram; lwd specifies the line width
plot(histogram_data$mids, normalized_histogram_density, type = "h", col = "gray", lwd = 4,

    # Specifying the header/title for the histogram plot
    main = "TASK 5: Histogram of Posterior Distribution from Monte Carlo Simulation",

    # Specifying the x-axis label for the histogram plot
    xlab = "Fuel (Liters)",

    # Specifying the y-axis label for the histogram plot
    ylab = "Probability Density",

    # Removing the default x-axis labels
    xaxt = "n",

    # Specifying the desired x-axis labels to extend from 0 to 200
    xlim = c(0, 200))

# Adding custom x-axis labels to enhance readability of the histogram plot
axis(1, at = seq(0, 200, by = 20), labels = seq(0, 200, by = 20))

# Using a for loop to add histogram bars to the plot (with normalized density)
for (i in 1:length(histogram_data$counts)) {
  rect(histogram_data$breaks[i], 0, histogram_data$breaks[i + 1],
normalized_histogram_density[i],
       border = "black", col = "gray")
}

# Adding a vertical line to denote the expected value of available fuel (mean) on the histogram
plot
abline(v = expected_value_monte_carlo, col = "red", lwd = 3, lty = 2)

# Adding a vertical line to denote the most likely value of available fuel (mode) on the histogram
plot
abline(v = most_likely_value_monte_carlo, col = "darkorange", lwd = 3, lty = 3)

# Adding a legend to label the key components of the histogram plot
# Note: lwd specifies the line width, lty specifies the line type
legend("topright", legend = c("Expected Value (Mean)", "Most Likely Value (Mode)"),
       col = c("red", "darkorange"), lwd = 3, lty = c(2, 3), bty = "o", box.col = "black", inset = 0.02,
cex = 1.2)
```

```
#______________________________________________________________________________
__
#### PLOTTING THE POSTERIOR FROM THE MONTE-CARLO SIMULATION WITH
THE POSTERIOR FROM THE GRID-BASED METHOD AND THE LIKELIHOOD
FUNCTION ####

# Plotting the posterior from the Monte Carlo simulation with the likelihood function and the
posterior from the grid-based method
# Note: Multiplied by 1.1 to enhance the visability of the plot
# Note: lwd specifies the line width, lty = 1 specifies the line type as solid
max_density <- max(c(pdf_usable_fuel, posterior, density(posterior_samples)$y)) * 1.1
plot(usable_fuel_in_tank, pdf_usable_fuel, type = "l", col = "blue", lwd = 4, lty = 1,

    # Specifying the title/header of the plot
    main = "TASK 5: Bayesian Posterior of Usable Fuel (Monte Carlo Method)",

    # Specifying the x-axis label for the plot
    xlab = "Fuel (Liters)",

    # Specifying the y-axis label for the plot
    ylab = "Probability Density",

    # Removing the default x-axis labels
    xaxt = "n",

    # Specifying the x-axis range to ensure that the full likelihood function is visible on the plot
    xlim = c(-60, 200),

    # Specifying the y-axis limits to ensure that both PDFs and the likelihood function are fully
visible on the plot
    ylim = c(0, max_density))

# Adding custom x-axis labels to enhance readability of the plot
axis(1, at = seq(-60, 200, by = 20), labels = seq(-60, 200, by = 20))
lines(usable_fuel_in_tank[valid_indices], posterior[valid_indices], col = "green", lwd = 4, lty =
1)

# Performing the Monte Carlo posterior density estimation
monte_carlo_density <- density(posterior_samples, from = 0, to = total_fuel_tank_capacity, n =
length(usable_fuel_in_tank))

# Normalizing the Monte Carlo density function
bin_width_mc <- diff(monte_carlo_density$x)[1]
normalized_monte_carlo_density <- monte_carlo_density$y / sum(monte_carlo_density$y *
bin_width_mc)
```

```
# Overlaying the Monte Carlo posterior PDF onto the plot
valid_mc_indices <- (monte_carlo_density$x >= 0) & (monte_carlo_density$x <=
total_fuel_tank_capacity)
lines(monte_carlo_density$x[valid_mc_indices],
normalized_monte_carlo_density[valid_mc_indices], col = "gold", lwd = 4)

# Adding vertical dashed lines to denote the bounds of the proper prior from 0 liters to 182 liters
segments(x0 = minimum_fuel_tank_capacity, y0 = 0, x1 = minimum_fuel_tank_capacity, y1 =
prior_height, col = "darkgray", lwd = 3, lty = 2)
segments(x0 = total_fuel_tank_capacity, y0 = 0, x1 = total_fuel_tank_capacity, y1 =
prior_height, col = "darkgray", lwd = 3, lty = 2)

# Adding a horizontal dashed line for the proper prior prior from 0 liters to 182 liters to form a
rectangle
segments(x0 = minimum_fuel_tank_capacity, y0 = prior_height, x1 = total_fuel_tank_capacity,
y1 = prior_height, col = "darkgray", lwd = 3, lty = 2)

# Adding a vertical line to denote the expected value of available fuel (mean) on the histogram
plot
abline(v = expected_value_monte_carlo, col = "red", lwd = 3, lty = 2)

# Adding a vertical line to denote the most likely value of available fuel (mode) on the histogram
plot
abline(v = most_likely_value_monte_carlo, col = "darkorange", lwd = 3, lty = 3)

# Adding a legend to label the key elements of the plot
# Note: lwd specifies the line width, lty specifies the line type
legend("topright", legend = c("Posterior (Monte Carlo)", "Posterior (Grid-Based)", "Likelihood
Function", "Proper Prior", "Expected Value (Mean)", "Most Likely Value (Mode)"),
      col = c("gold", "green", "blue", "darkgray", "red", "darkorange"), lwd = c(4, 4, 4, 3, 3, 3), lty
= c(1, 1, 1, 2, 2, 3),
      bty = "o", box.col = "black", inset = 0.02, cex = 1.2)
#________________________________________________________________________________
__

# Opening a new plot to print the estimates calculated above on sixth page of the PDF file
plot.new()

# Adding a title at the top of the results page
title(main = "TASK 5: Estimates after Bayesian Update using Monte Carlo Method", font.main =
2, cex.main = 1.2)
text(0.4, 0.8, paste(

  # Expected value of available fuel and rounded to 1 decimal place
  "Expected Value of Available Fuel (Mean):", round(expected_value_monte_carlo, 1), "liters\n",
```

```
  # Most likely value of available fuel and rounded to 1 decimal place
  "Most Likely Value of Available Fuel (Mode):", round(most_likely_value_monte_carlo, 1),
"liters\n",

  # Probability of negative fuel in the tank rounded to 1 decimal place and specified as a
percentage
  "Probability of Negative Fuel in the Tank:", round(probability_negative_fuel_monte_carlo *
100, 1), "%\n"
), cex = 1.2)

#==========================================================================
===========
#### TASK 7 ####
# 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 the airport?
#==========================================================================
===========

# Defining the required flight time (100 minutes)
required_flight_time <- 100

# Defining the reserve fuel time (30 minutes)
reserve_fuel_time <- 30

# Converting the flight time requirement into a fuel requirement
required_fuel <- (required_flight_time / 60) * airplane_fuel_consumption_rate
reserve_fuel <- (reserve_fuel_time / 60) * airplane_fuel_consumption_rate

# Calculating the total fuel required to reach the destination (accounting for the reserve fuel)
total_required_fuel <- required_fuel + reserve_fuel
#__________________________________________________________________________
__

# Defining a range of fuel levels
# Note: by = 0.01 specifies the increment or step size
fuel_levels <- seq(0, total_fuel_tank_capacity, by = 0.01)

# Calculating the probability density function (PDF) for the fuel levels based on the fuel sensor
readings
fuel_pdf <- dnorm(fuel_levels, mean = fuel_sensor_reading, sd =
fuel_sensor_standard_deviation)

# Normalizing the PDF to ensure that it sums to a value of 1
fuel_pdf <- fuel_pdf / sum(fuel_pdf)
```

```r
# Calculating the available flight times corresponding to each fuel level
available_flight_times <- (fuel_levels / airplane_fuel_consumption_rate) * 60

# Calculating the cumulative distribution function (CDF) using the cumsum function
fuel_cdf <- cumsum(fuel_pdf)
#____________________________________________________________________________
__

# Calculating the probability of meeting the flight requirement (including the reserve fuel)
probability_meeting_requirement <- sum(fuel_pdf[available_flight_times >=
(required_flight_time + reserve_fuel_time)])

# Calculating the probability of running out of fuel before making it to the airport
probability_running_out_of_fuel <- sum(fuel_pdf[available_flight_times <
required_flight_time])

#____________________________________________________________________________
__
#### PLOTTING THE ESTIMATED AVAILABLE FLIGHT TIME ####

# Plotting the probability density function of available flight time
# Note: lwd specifies the line width; lty = 1 specifies the line type as solid
plot(available_flight_times, fuel_pdf, type = "l", col = "darkgray", lwd = 4, lty = 1,

    # Specifying the title/header of the plot
    main = "TASK 7: PDF over Estimated Available Flight Time",

    # Specifying the x-axis label for the plot
    xlab = "Available Flight Time (Minutes)",

    # Specifying the y-axis label for the plot
    ylab = "Probability Density")

# Adding a vertical line to the plot to denote the required flight time (100 min)
abline(v = required_flight_time, col = "red", lwd = 3, lty = 2)

# Adding a vertical line to the plot to denote the total required flight time including reserve (100
min + 30 min reserve)
abline(v = (required_flight_time + reserve_fuel_time), col = "blue", lwd = 3, lty = 2)

# Adding a legend to the plot to label the key features on the plot
legend("topright", legend = c("PDF of Available Flight Time", "100 Min Flight Time", "100 Min
+ 30 Min Reserve"),
      col = c("darkgray", "red", "blue"), lwd = c(4, 3, 3), lty = c(1, 2, 2),
      bty = "o", box.col = "black", inset = 0.02)
```

```
#_____________________________________________________________________________
__
#### PLOTTING THE CDF OF AVAILABLE FLIGHT TIME ####

# Plotting the cumulative distribution function (CDF) (y-axis) over the available flight time (x-
axis)
# Note: lwd specifies the line width; lty = 1 specifies the line type as solid
plot(available_flight_times, fuel_cdf, type = "l", col = "darkgray", lwd = 4, lty = 1,

    # Specifying the title/header of the plot
    main = "TASK 7: CDF over Estimated Available Flight Time",

    # Specifying the x-axis label
    xlab = "Available Flight Time (Minutes)",

    # Specifying the y-axis label
    ylab = "Cumulative Probability")

# Adding a vertical line to the plot to denote the required flight time (100 min)
abline(v = required_flight_time, col = "red", lwd = 3, lty = 2)

# Adding a vertical line to the plot to denote the total required flight time including reserve (100
min + 30 min reserve)
abline(v = (required_flight_time + reserve_fuel_time), col = "blue", lwd = 3, lty = 2)

# Adding a legend to label the key features of the plot
legend("bottomright", legend = c("CDF of Available Flight Time", "100 Min Flight Time", "100
Min + 30 Min Reserve"),
      col = c("darkgray", "red", "blue"), lwd = c(4, 3, 3), lty = c(1, 2, 2),
      bty = "o", box.col = "black", inset = 0.02)
#_____________________________________________________________________________
__
##### PLOTTING THE SURVIVAL FUNCTION OVER AVAILABLE FLIGHT TIME ####

# Calculating the survival function
# Note: that the survival function = 1 - CDF
survival_function <- 1 - fuel_cdf

# Plotting the survival function with a logarithmic y-axis over the available flight time (x-axis)
# Note: lwd specifies the line width, lty = 1 specifies the line type as solid
plot(available_flight_times, survival_function, type = "l", col = "darkgray", lwd = 4, lty = 1,

    # Specifying the title/header for the plot
    main = "TASK 7: Survival Function over Estimated Available Flight Time",
```

```
     # Specifying the x-axis label
     xlab = "Available Flight Time (Minutes)",

     # Specifying the y-axis label
     ylab = "Survival Probability",

     # Setting the y-axis to a logarithmic (log) scale
     log = "y")

# Adding a vertical line to the plot to denote the required flight time (100 min)
abline(v = required_flight_time, col = "red", lwd = 3, lty = 2)

# Adding a vertical line to the plot to denote the total required flight time including reserve (100
min + 30 min reserve)
abline(v = (required_flight_time + reserve_fuel_time), col = "blue", lwd = 3, lty = 2)

# Adding a legend to label the key features of the plot
legend("topright", legend = c("Survival Function", "100 Min Flight Time", "100 Min + 30 Min
Reserve"),
       col = c("darkgray", "red", "blue"), lwd = c(4, 3, 3), lty = c(1, 2, 2),
       bty = "o", box.col = "black", inset = 0.02)
#__________________________________________________________________________________
__

# Opening a new plot to print the estimates calculated above on the last (eighth) page of the PDF
file
plot.new()

# Adding a title at the top of the last page of the PDF file
title(main = "TASK 7: Probabilities from the Plot of Estimated Available Flight Time",
font.main = 2, cex.main = 1.2)

# Probability of making it to the airport with reserve fuel
text(0.4, 0.8, paste(

  # Probability of making it to the airport with reserve fuel rounded to 1 decimal place and
specified as a percentage
  "Probability of making it to the airport with reserve fuel: ",
round(probability_meeting_requirement * 100, 1), "%\n",

  # Probability of running out of fuel before reaching the airport rounded to 1 decimal place and
specified as a percentage
  "Probability of running out of fuel before reaching the airport: ",
round(probability_running_out_of_fuel * 100, 1), "%\n"
), cex = 1.0)
```

```
#====================================================================
===========
# Closing the PDF file
dev.off()
```