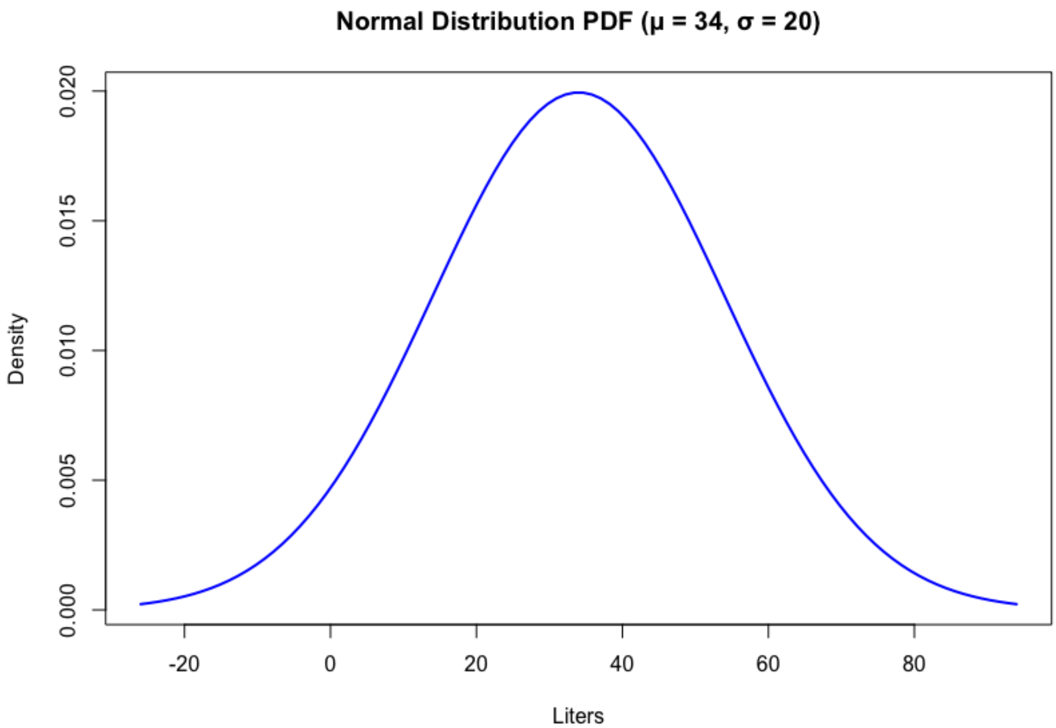
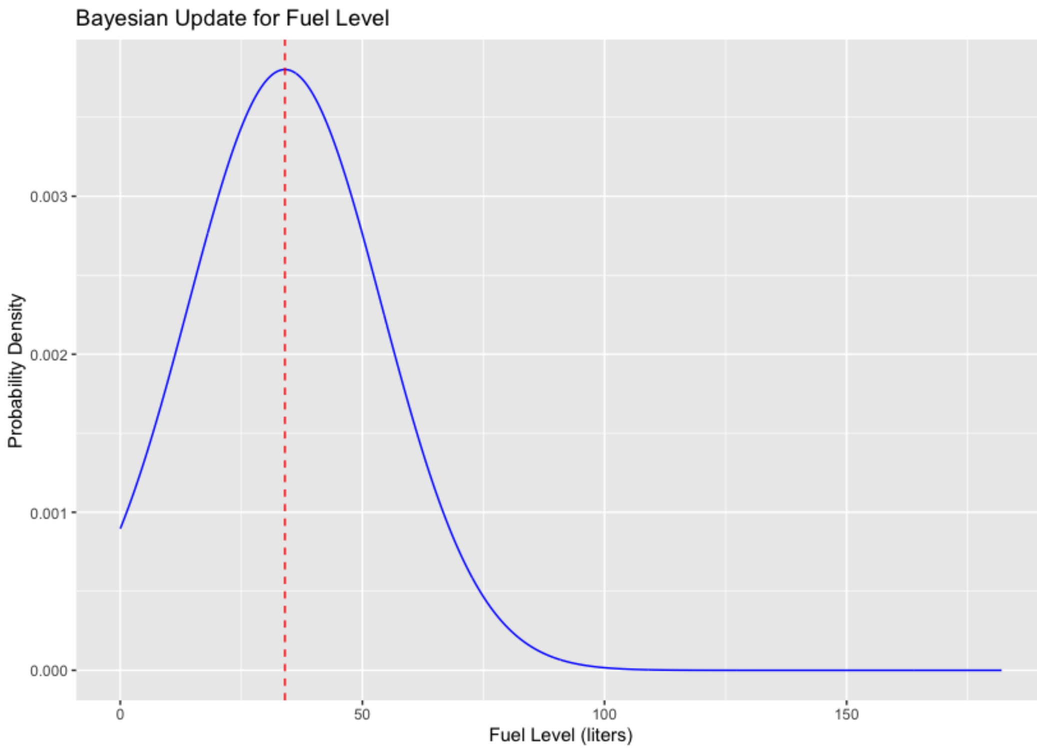
Questions

2. Yes the estimates makes sense. Our best possible estimate with the available information is the sensor reading. Hence the expected value and most likely value of available fuel is 34 liters. The probability of negative fuel is 4.5%



3. A proper prior typically should avoid negative fuel values. In this case we can use a uniform prior in the form U(0,180) developed from the total available fuel. With the Bayes update the probability of negative fuel improves to 0%. One major factor for this result is that the uniform prior U(0,180) defined assigns zero probability to fuel level outside of the range.

4.

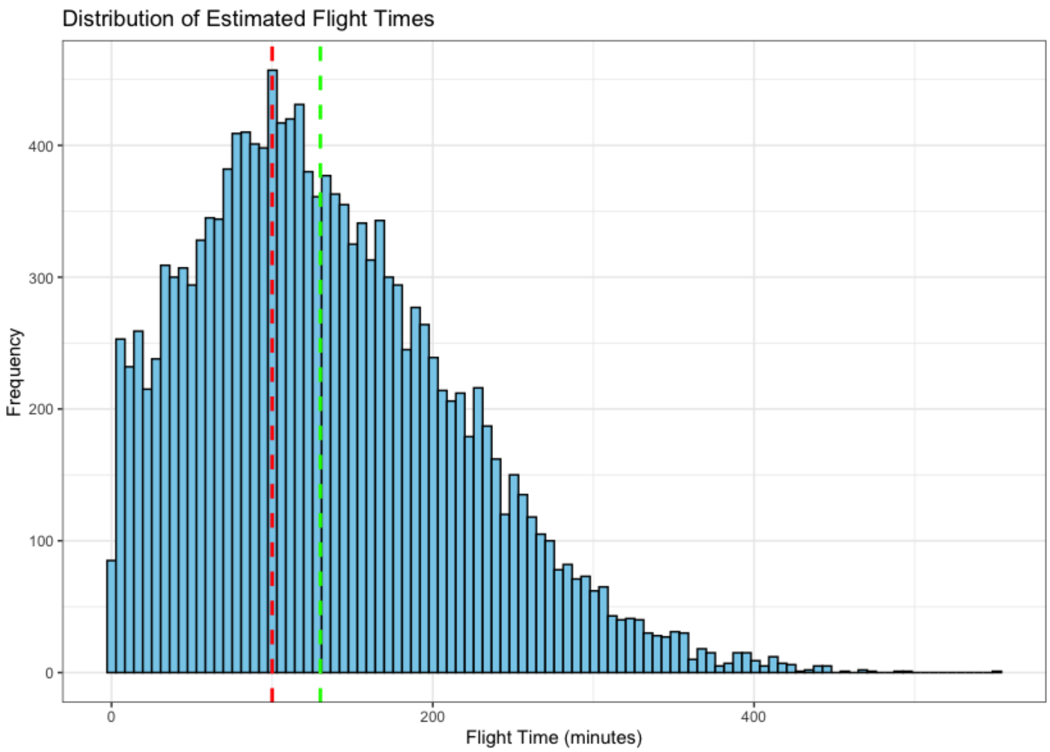


6. To implement a simple Kalman Filter to the problem, we can make the following assumptions:

- The error in measurement follows a known Gaussian distribution.
- The system follows a Markov process.
- The noise in the sensor and fuel consumption are Gaussian.

The assumptions are generally realistic as the Kalman filter requires specification of mean and covariance of the measurement noise which might not be perfectly known and may have to be estimated.

7. a. Probability you make to airport that is 100mins away with 30 minutes reserve is 47.2%
   b. Probability you run out of fuel trying to reach the airport is 38.0%



Documentation

b. In determining the appropriate prior from question 3, I decided to use a uniform prior based on the fuel capacity given the uncertainty regarding factors affecting useable fuel for the flight. I believe the fuel capacity represented a more realistic factor that had a key impact on the fuel. However, I could have also used a Gaussian prior(truncated), but with the uncertainty, I though the uniform distribution was a better choice.

c. Some uncertainties I believe I may have missed include mechanical factors like fuel leakage, engine efficiency, and climatic factors (ie jet streams). These uncertainties could have significantly impacted useable fuel for the journey.

d. I believe my analysis is reproducible because I provided the seed for my random samplings which can be used to replicate my results.

e. I would like to reference sampling techniques employed from the Confidence Interval Estimation chapter of Verzani's "Simple R"

f. The code utilizes the GNU license. I assign copyright of this work to myself.

Appendix

```
#PSet 3
library(ggplot2)
library(dplyr)
library(tibble)



#Question 2
#pdf with sensor reading of 34 liters and sensor sd of 20 liters
s_mu = 34
s_sigma = 20

xvals <- seq(s_mu - 3*s_sigma, s_mu + 3*s_sigma, length.out = 100)

pdf_vals <- dnorm(xvals, mean = s_mu, sd = s_sigma)

#Plotting
plot(xvals, pdf_vals, type = "l", col = "blue", lwd = 2,
    xlab = "Liters", ylab = "Density",
    main = "Normal Distribution PDF (μ = 34, σ = 20)")

#prob of neg fuel
pneg <- pnorm(0, mean = s_mu, sd = s_sigma)
cat("probability of negative fuel",pneg)



#Q2.4 - Grid-based Bayesian
#Using a uniform prior U(0,180)
```

```
tfuel = 182
userate = 18
use_sd = 2

#defining grid
fuel = seq(0,tfuel,length.out=1000)

prior = rep(1 / length(fuel), length(fuel)) #uniform prior

likely <- dnorm(fuel, mean = s_mu, sd = s_sigma) #likelihood

#posterior
post <- likely * prior
post <- post / sum(post)

#Plotting
ggplot(data.frame(fuel, post), aes(x = fuel, y = post)) +
  geom_line(color = "blue") +
  geom_vline(xintercept = s_mu, color = "red", linetype = "dashed") +
  labs(title = "Bayesian Update for Fuel Level", x = "Fuel Level (liters)", y = "Probability
Density")

#neg prob
pneg_bayes <- sum(post[fuel < 0])
cat("Probability of negative fuel :", pneg_bayes, "\n")



#2.5 Bayes Monte Carlo

set.seed(42)
```

```
sim = 15000
fuel_samples = runif(sim, 0, tfuel)
sensor_noise = rnorm(sim, mean = 0, sd = s_sigma)
measured_fuel = fuel_samples + sensor_noise

#Likelihood
likely_mc <- dnorm(measured_fuel, mean = s_mu, sd = s_sigma)
weights_mc <- likely_mc / sum(likely_mc)

post_mc <- sample(fuel_samples, size = sim, replace = TRUE, prob = weights_mc)



#2.7 Estimated Flight Time

# Sampling consumption rates
use_samples <- rnorm(sim, mean = userate, sd = use_sd)
use_samples[use_samples < 0] <- 0

# Calculating flight times in minutes
flight_times <- (post_mc / use_samples) * 60

# probability of making it with at least 30 minutes reserve
req_time <- 100 + 30
prob_success <- mean(flight_times >= req_time)

# probability you run out of fuel
prob_failure <- mean(flight_times < 100)

#Plotting of estimated flight time
ggplot(data.frame(flight_times), aes(x = flight_times)) +
  geom_histogram(bins = 100, fill = "skyblue", color = "black") +
```

```
  geom_vline(xintercept = 100, color = "red", linetype = "dashed", size = 1) +  # Target time
  geom_vline(xintercept = req_time, color = "green", linetype = "dashed", size = 1) +  # Target +
reserve
  labs(title = "Distribution of Estimated Flight Times", x = "Flight Time (minutes)", y =
"Frequency") +
  theme_bw()

cat("Probability of success with 30 minutes reserve :", prob_success, "\n")
cat("Probability of failure :", prob_failure, "\n")
```