Report on Frequentist and Bayesian Inference of Dissolved Solids

Jiazhou Liang

Background

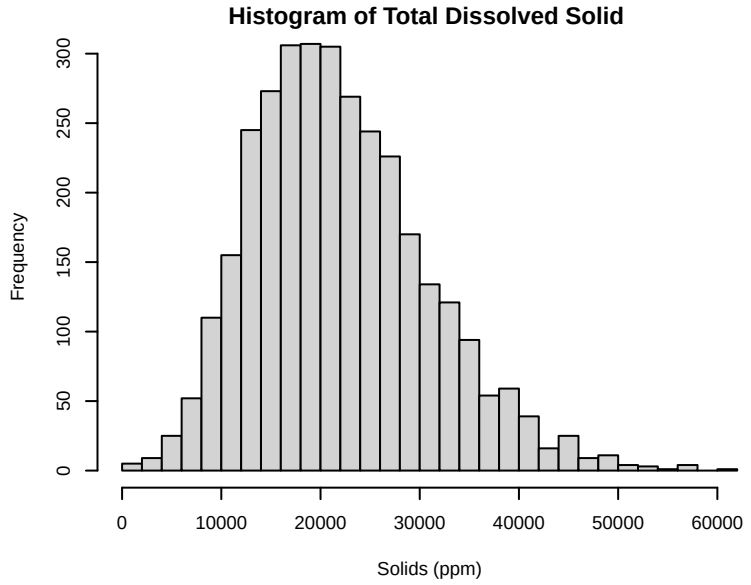
Water is essential for all human life, it is important to have safe drinking water for health protection. This dataset from Kaggle describes the water quality metrics of different water bodies around the world. The sample size,n, for this dataset is 3276. There are 10 variables in this dataset in total:“Ph”,“Hardness”,“Solids”,“Chloramines”,“Sulfate”, “conductivity”, “Organic carbon”,“Trihalomethanes”, “Turbidity”, and “Potability”.

Within these variables, solids(ppm), or the total dissolved solids, are the major evaluation of the inorganic and organic minerals in the public water system. Higher solid level results in unwanted tastes in the water. This analysis will focus on the data on the level of solids (ppm) in the samples. More information about this dataset and the descriptions for each variable can be found here

Using the computing power of modern computers, this project applied unsupervised learning techniques, iteration algorithm, and data generation methods (Bootstrap and MCMC) to two major displaces of statistical inferences, Frequentist And Bayesian. the goal of this project is to estimate the parameters for possible statistical models for the samples.

Data Analysis and Parametric model

Plotting the histogram of the value of soilds in samples from the dataset(Code: Appendix 1.1)



From the plot above, the distribution of the samples has a right-skewed shape. Also, all samples are continuous and positive. Therefore, choosing Gamma Distribution as the parametric model to approximate the dataset.

Frequentist inference

Log-Likelihood function & plots

The distribution function of gamma distribution with shape σ and scale α is given by

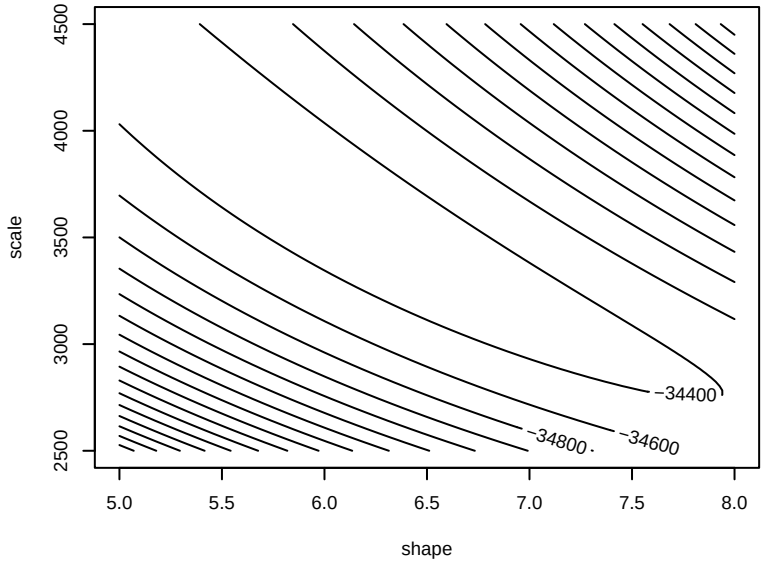
$$f(x \mid \alpha, \sigma) = \frac{1}{\sigma^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\sigma}$$

The likelihood function and log-likelihood function will be

$$L(x|\alpha,\sigma) = \prod_{i=1}^{n} \frac{1}{\sigma^{\alpha}\Gamma(\alpha)} x_i^{\alpha-1} e^{-x_i/\sigma} = (\frac{1}{\sigma^{\alpha}\Gamma(\alpha)})^n \prod_{i=1}^{n} x_i^{\alpha-1} e^{\sum_{i=1}^{n} -x_i/\sigma}$$

$$l(x|\alpha,\sigma) = log(\frac{1}{\sigma^{\alpha}\Gamma(\alpha)} \prod_{i=1}^{n} x_i^{\alpha-1} e^{\sum_{i=1}^{n} -x_i/\sigma}) = (\alpha-1)\sum_{i=1}^{n} log\ x_i - n\ log\ \Gamma(\alpha) - n\alpha\ log\ \sigma - \frac{1}{\sigma}\sum_{i=1}^{n} x_i$$

From the histogram of dataset distribution, the value of solid(ppm) is mostly in the range of 10000 to 50000. Since there are two parameters in the gamma distribution, the plot for the log-likelihood function will be a two-dimension plot. In this case, the contour plot will be the fittest choice to see the peak of the log likelihood.(Code: Appendix 1.2)



Maximum likelihood Estimation

Since the Newton-Raphson method to maximum likelihood estimation (MLE) is not parametric, it can be used in more general cases. Also, the second derivative of Gamma Distribution is relatively easy to calculate. Therefore, the Newton-Raphson method will have a much faster calculation to approximate the MLE than the Gradient descent method. The disadvantage of the Newton-Raphson method is requiring a good starting position. The log-likelihood plots above or methods of moments provide an efficient way to solve this problem. Therefore, this data analysis will use the Newton-Raphson method to find MLE.

The score function can be calculated by the first derivative of the log-likelihood function respects to α and σ

$$S(x|\alpha) = \sum_{i=1}^{n} log\ x_i - n\ \Gamma'(\alpha) - n\ log\ \sigma \quad S(x|\sigma) = -n\alpha\frac{1}{\sigma} + \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i$$

The Observed information matrix can be calculated by the negative second derivative of log-likelihood function,

$$-\begin{bmatrix} \frac{\partial S(x|\alpha)}{\partial \alpha} & \frac{\partial S(x|\alpha)}{\partial \sigma} \\ \frac{\partial S(x|\sigma)}{\partial \alpha} & \frac{\partial S(x|\sigma)}{\partial \sigma} \end{bmatrix} = -\begin{bmatrix} -n\ \Gamma''(\alpha) & -\frac{n}{\sigma} \\ -\frac{n}{\sigma} & n\alpha\frac{1}{\sigma^2} - \frac{2}{\sigma^3}\sum_{i=1}^{n} x_i \end{bmatrix}$$

From the log-likelihood plot, the peak point of the contour is range from $\sigma = 5$ to 8, $\alpha = 2500$ to 4000. Therefore, the starting point will be (5,4000) to approach the MLE from the top left of the plot. Using the Newton-Raphson Method for Observed Matrix, we obtain the MLE (code: Appendix 1.3)

```
##       [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## shape    5    5.509    5.850    5.872    5.872    5.872    5.872
## scale 4000 3996.675 3748.808 3749.156 3748.822 3748.822 3748.822
```

The MLE($\alpha = 3748.822$,$\sigma = 5.872$) is obtained after 5 iterations with starting point ($\alpha = 4000$,$\sigma = 5$)

Confidence Interval

There are several ways to calculate the Confidence Interval. Wald Statistic and the log-likelihood ratio is efficient to calculate but provides less accuracy than other methods. Profile Likelihood method has good accuracy. However, it has large complexity which takes more computational time than others. In bootstrap sampling, since there is no information's about the dataset's normality, the standard bootstrap confidence interval cannot apply to this analysis.

Furthermore, the parametric model Gamma Distribution allows the dataset to use parametric bootstrap for smoother samples. Therefore, Bootstrap Confidence Interval with parametric bootstrap sampling is used in this case. (code: Appendix 1.4)

```
##         lower   upper
## Shape    5.53    6.22
## Scale 3546.22 3969.61
```

The 95% marginal confidence intervals using Bootstrap method with 1000 bootstrap sample is given by Shape parameter (5.53,6.22), Scale parameter (3546.22,3969.61).

Bayesian inference

Prior and Posterior Distribution

Since both value of α and σ is unknown in this study, a bivariate prior will be used for Bayesian inferences. Also, there is only little or no prior knowledge available about the parameters. Therefore, it is hard to find informative priors to this dataset. We know that as the sample sizes increase, the difference between posterior distributions from different priors will be decreased. Since the samples size of this dataset is 3276, using improper priors with $p(\alpha, \sigma) = 1$ is a good representation of the parameters' belief.

Using the improper prior with $p(\alpha, \sigma) = 1$, Since the log-likelihood function provide an efficient way to implement, the log scale of the posterior would be

$$\log(\pi(\gamma|x)) = \log(\frac{L(x|\alpha,\sigma)p(\alpha,\sigma)}{p(x)}) = \log L(x|\alpha,\sigma)1 - \log(1) = \log L(x|\alpha,\theta) - 0$$
$$= (\alpha - 1)\sum_{i=1}^{n} log\ x_i - n\ log\ \Gamma(\alpha) - n\alpha\ log\ \sigma - \frac{1}{\sigma}\sum_{i=1}^{n} x_i$$

Since the posterior distribution is in an unknown shape, using bivariate MCMC random walk to generate the estimated posterior.

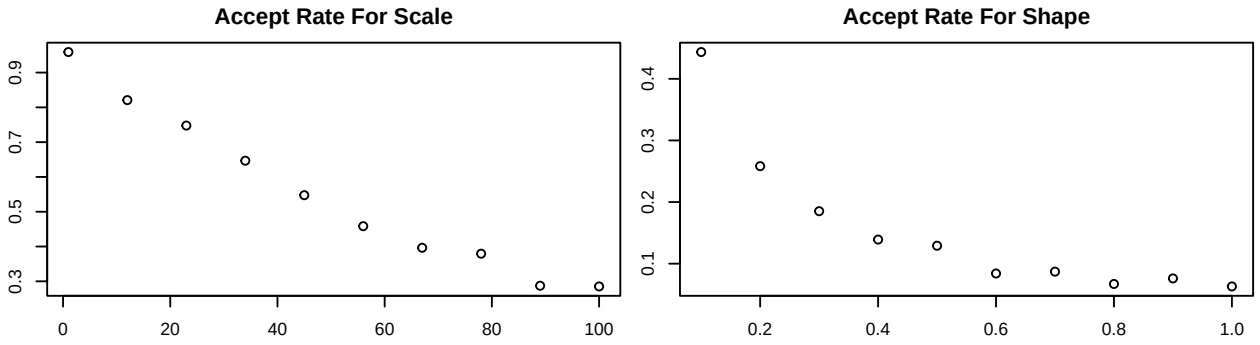
Markov chain Monte Carlo (MCMC)

Implementing the function to return the density of log scale posterior distribution

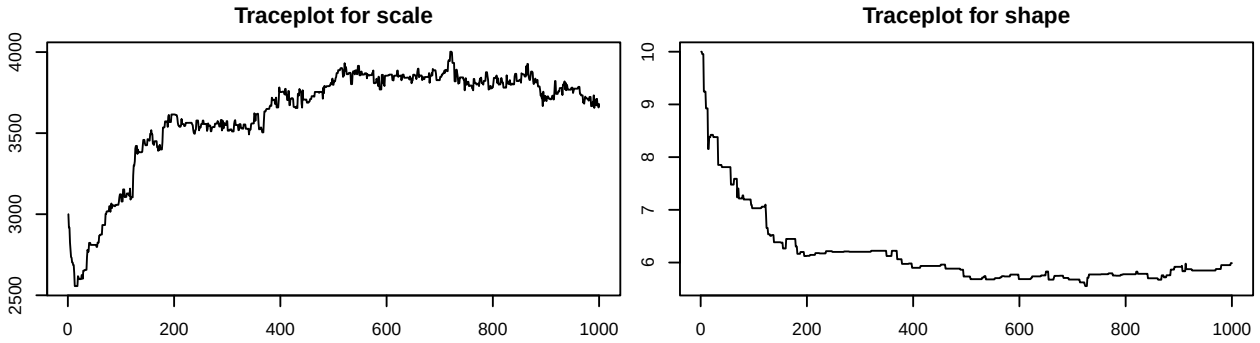
```
n = length(Solids)
posterior <- function (a,sigma){
 densities = sum(dgamma(Solids, shape = sigma, scale = a, log = TRUE))
  return(densities)
}
```

Since Normal distribution provides some degrees of flexibility for tunning a random walk, the proposal distribution will be $N(X^t, k)$, where k is the variance. Since it is symmetric, we only need to consider the ratio of posterior distribution. Also, using single parameter updates to adjust the value of k independently for better performance. The code for THE random walk function with a single parameter update with proposal distribution $N(X^t, k)$ can be found in Appendix 1.6.

The only evidence about the location of parameters is the MLE from frequentist inferences ($\alpha = 3748.822$,$\sigma = 5.872$) , using the initial value for $(\alpha^0, \sigma^0) = (3000, 10)$ and locating the mass density area with relative small acceptance probabilities. (Code for calculating acceptance probability for a range of variance is in Appendix 1.7)



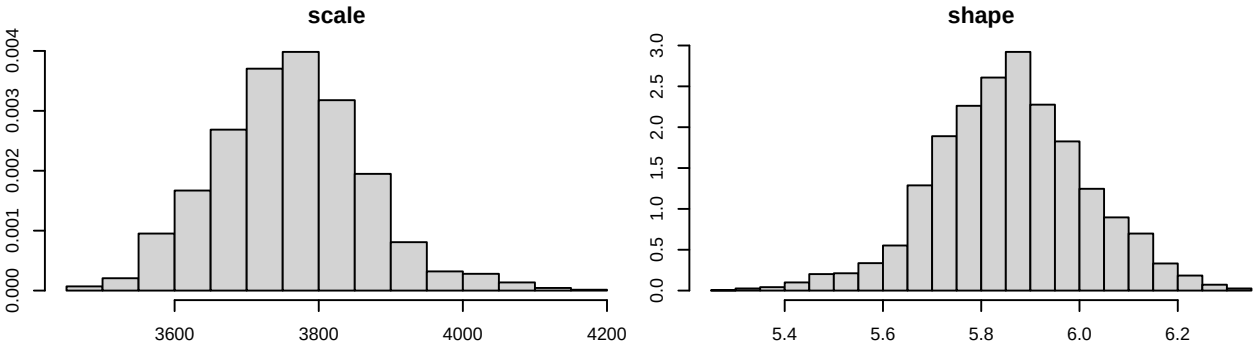
From above plots, the variance (60,0.6) has desired acceptance probabilities to locate mass density area. (Code for below traceplots can be found in appendix 1.8)



From the trace plot, the mass density area for the value of the scale parameter is around 3800. The shape parameter has a mass density area around 6. Therefore, using the initial value (3800,5) with the variance that has the best mixing properties in the density area. (The code for calculating the mixing properties of a range of variances can be found in appendix 1.9)

```
c(sigma[which.max(alpha_mixing)],sigma2[which.max(theta_mixing)])
```

From the above, the variance (56,0.119) has the best mixing properties to generate the posterior samples in mass density area.Increasing the number of generated samples to 10000 for better accuracy. (Code for the Histogram of posterior in Appendix 1.10)



The above histograms are the estimated marginal posterior distribution for both parameters. The generated samples can be used to calculate the point estimate of parameters from expected value and credible interval for uncertainty. Since there are correlation between generated samples, using coda library to fit an autoregressive (AR) model to the data and then estimates its spectral density without losing useful information.

```
library(coda)
c(mean(mcmc[, 1]),mean(mcmc[, 2]))
mean(mcmc[, 1]) + c(-1, 1) * qnorm(0.975) * sd(mcmc[, 1])/sqrt(effectiveSize(mcmc[, 1]))
mean(mcmc[, 2]) + c(-1, 1) * qnorm(0.975) * sd(mcmc[, 2])/sqrt(effectiveSize(mcmc[, 2]))
```

```
## [1] 3760.376830    5.858261
## [1] 3737.399 3783.354
## [1] 5.824532 5.891989
```

Therefore, the point estimate from posterior distribution will be $(\hat{\alpha}, \hat{\sigma}) = (3760.377, 5.875)$. The marginal credible interval for scale parameter $\hat{\alpha}$ will be (3737.399, 3783.354).The marginal credible interval for scale parameter $\hat{\sigma}$ will be (5.824, 5.892).

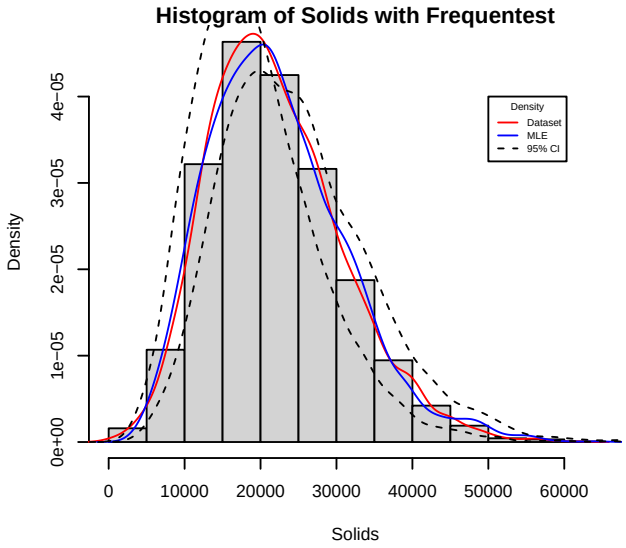
Performance Anaylsis Conclusion

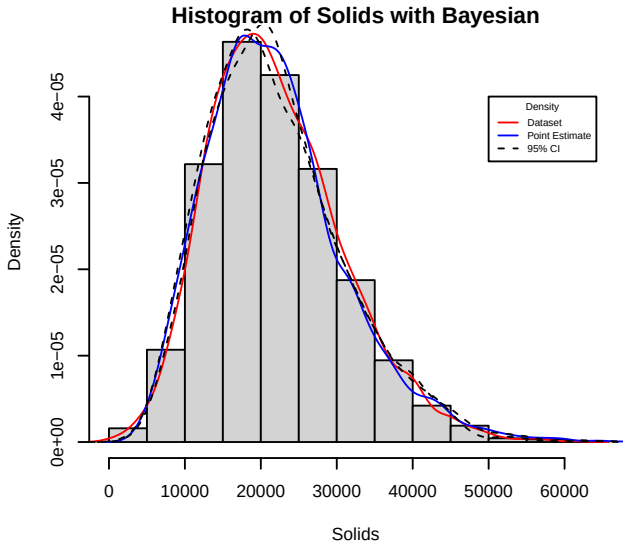
Type	Estimate Scale	Lower	Upper	Estimate Shape	Lower	Upper
Frequentist	3748.822	3546.22	3969.61	5.872	5.53	6.22
Bayesian	3760.377	3737.399	3783.354	5.875	5.824	5.892

The above table summarizes the result from both Frequentist and Bayesian inferences. From the table, the difference between the point estimate from the two inference methods is relatively small. Therefore, these two methods provide a similar estimating result in both parameters. However, the range of uncertainty from Bayesian inferences (95% credible interval) is much smaller than the range of frequentists (95% Confidence interval).

Applying the density plot to the histogram of the value of solids with four simulations. In the first plot, the red line represents the density of the dataset. The blue line for the density of random generated Gamma Distribution sample using the estimation from Frequentist inference($\alpha = 3748.822$,$\sigma = 5.872$). Instead of using the frequentist inference, the blue line in the second plot shows the result from Bayesian inferences. The shapes of the two lines are closely matched in both plots, which provides a piece of evidence to support that the dataset follows the Gamma Distribution. Notices that the peak of samples that are generated by parameters from Bayesian is slightly lower than the original dataset. But the difference is very small.

Furthermore, the black dash line represents the lower and upper bond of the 95% marginal confidence (credible) interval. In the Bayesian inference, the range of two dash lines is much smaller than Frequentist's range. In both plots, the density of the dataset is within the range of lower and upper bounds. Therefore, from the result of both Frequentist and Bayesian inference, Gamma Distribution should be a fit parametric model for this dataset.(Code: Appendix 1.11)


Histogram of Solids with Frequentest
Density
Dataset
MLE
95% CI
Density
0e+00
1e−05
2e−05
3e−05
4e−05
0
10000
20000
30000
40000
50000
60000
Solids


Histogram of Solids with Bayesian
Density
Dataset
Point Estimate
95% CI
Density
0e+00
1e−05
2e−05
3e−05
4e−05
0
10000
20000
30000
40000
50000
60000
Solids

Appendix

1.1 Histogram of Total Dissolved Solid

```
par(mar=c(4,4,1,0.5),cex = 0.6)
water <- read.csv("water_potability.csv")
Solids <- water$Solids
hist(Solids,main = "Histogram of Total Dissolved Solid ", breaks  = 40,
     xlab = "Solids (ppm)")
```

1.2 Log-liklihood Plot

```
par(mar=c(4,4,1,0.5),cex = 0.6)
loglik.gam <- function(shape = NULL, scale = NULL, x = NULL) {
val = matrix(0, nrow = length(shape), ncol = length(scale))
for (i in 1:length(shape)) {
  for (j in 1:length(scale)) {
    val[i, j] = sum(dgamma(x, shape = shape[i], scale = scale[j], log = TRUE))
  }
  }
return(val)
}
shape.seq = seq(5, 8, length.out = 100)
scale.seq = seq(2500, 4500, length.out = 100)
gam.loglik = loglik.gam(shape = shape.seq, scale = scale.seq, x = Solids)
contour(x = shape.seq, y = scale.seq, z = gam.loglik, nlevels = 20,
        xlab = "shape", ylab = "scale")
```

1.3 MLE Estimation using Newton-Raphson

```
x = Solids
score.fn <- function(x = NULL, shape = NULL, scale = NULL) {
    n = length(x)
    val = numeric(2)
    val[1] = -n * digamma(shape) - n * log(scale) + sum(log(x))
    val[2] = -n * shape/scale + sum(x)/scale^2
    return(val)
}
obs.inf <- function(x = NULL, shape = NULL, scale = NULL) {
    n = length(x)
    val = matrix(0, ncol = 2, nrow = 2)
    val[1, 1] = -n * trigamma(shape)
    val[2, 2] = n * shape/scale^2 - 2 * sum(x)/scale^3
    val[1, 2] = -n/scale
    val[2, 1] = -n/scale
    val = -1 * val
    return(val)
}
nstep = 6
xpar = matrix(0, nrow = nstep + 1, ncol = 2, dimnames = list(NULL, c("shape",
    "scale")))
xpar[1, ] = c(5, 4000)
for (i in 2:(nstep + 1)) {
```

```
    xpar[i, ] = xpar[i - 1, ] + solve(obs.inf(x = x, shape = xpar[i - 1, 1],
        scale = xpar[i - 1, 2])) %*% score.fn(x = x, shape = xpar[i - 1, 1],
        scale = xpar[i - 1, 2])
}
round(t(xpar),3)
```

1.4 Bootstrap Confidence Interval

```
set.seed(440)
n = length(Solids)
x = Solids

loglik2.gam <- function(x = NULL, datax = NULL) {
    val = sum(dgamma(datax, shape = x[1], scale = x[2], log = TRUE))
    return(val)
}
B = 1000
options(warn = -1)
theta.boot = t(sapply(1:B, FUN = function(b) {
    Q = x[sample(n, n, replace = TRUE)]
    xbar = mean(Q)
    varx = var(Q) * (length(Q) - 1)/length(Q)
    xpar = c(xbar^2/varx, varx/xbar)
    optim(par = xpar, loglik2.gam, datax = Q, control = list(fnscale = -1))$par
}))
theta.boot <- na.omit(theta.boot)
a = 0.05
temp = matrix(NA, nrow = 2, ncol = 2)
dimnames(temp) = list(c("Shape", "Scale"), c("lower", "upper"))
temp[1, ] = quantile(theta.boot[, 1], c(a/2, 1 - a/2))
temp[2, ] = quantile(theta.boot[, 2], c(a/2, 1 - a/2))
round(temp, 2)
```

1.5 Log-scale Posterior

```
n = length(Solids)
posterior <- function (a,sigma){
 densities = sum(dgamma(Solids, shape = sigma, scale = a, log = TRUE))
  return(densities)
}
```

1.6 Random Walk Fucntion

```
random.walk.mcmc <- function(n = NULL, sigma, x0 = c(0, 0)) {
    x = matrix(0, nrow = n, ncol = 2)
    x[1, ] = x0
    for (i in 2:n) {
      ## Update shape
        y2 = rnorm(1, x[i - 1,2], sd = sigma[2])
        u = runif(1)
         if (y2 > 0){
        accept = exp(posterior(x[i-1, 1], y2)-posterior(x[i-1, 1], x[i - 1, 2]))
```

```
        if (u < accept)
           x[i, 2] = y2 else x[i, 2] = x[i - 1, 2]
           }else x[i, 2] = x[i - 1, 2]

      ## Update scale
        y = rnorm(1, x[i - 1,1], sd = sigma[1])
        u = runif(1)
         if (y > 0){
        accept = exp(posterior(y, x[i, 2])-posterior(x[i - 1, 1], x[i, 2]))
        if (u < accept)
           x[i, 1] = y else x[i, 1] = x[i - 1, 1]
           }else x[i, 1] = x[i - 1, 1]

    }
    return(x)
}
```

1.7 Acceptance Proabilities Plots

```
set.seed(440)
alpha_accept = c()
alpha_mixing =c()
n0 = 1000
sigma = seq(1,100,length.out = 10)
for (i in 1:10){
mcmc = random.walk.mcmc(n = n0, sigma = c(sigma[i],1), x0 = c(3000, 10))
diff1 = mcmc[, 1][-1] - mcmc[, 1][-n0]
alpha_accept[i] = mean(diff1 != 0)
}
theta_accept = c()
theta_mixing =c()
sigma2 = seq(0.1,1,length.out = 10)

for (i in 1:10){
mcmc = random.walk.mcmc(n = n0, sigma = c(20,sigma2[i]), x0 = c(3000, 10))
diff2 = mcmc[, 2][-1] - mcmc[, 2][-n0]
theta_accept[i] = mean(diff2 != 0)
}
par(mfrow = c(1, 2),mar=2.5 * c(1, 1, 1, 0.1),mfrow = c(1,2),cex = 0.6)
plot(sigma ,alpha_accept,main = "Accept Rate For Scale")
plot(sigma2 ,theta_accept,main = "Accept Rate For Shape")
```

1.8 Traceplot

```
set.seed(440)
n0 = 1000
mcmc = random.walk.mcmc(n = n0, sigma = c(60,0.6), x0 = c(3000, 10))
par(mfrow = c(1, 2),mar=2.5 * c(1, 1, 1, 0.1),mfrow = c(1,2),cex = 0.6)
plot(mcmc[, 1], type = "l",main = "Traceplot for scale")
plot(mcmc[, 2], type = "l",main = "Traceplot for shape")
```

1.9 Mixing Properites Plot

```
set.seed(440)
alpha_accept = c()
alpha_mixing =c()
n0 = 1000
sigma = seq(1,100,length.out = 10)
for (i in 1:10){
mcmc = random.walk.mcmc(n = n0, sigma = c(sigma[i],1), x0 = c(3800, 6))
diff1 = mcmc[, 1][-1] - mcmc[, 1][-n0]
alpha_mixing[i] =  mean(diff1^2)
}
theta_accept = c()
theta_mixing =c()
sigma2 = seq(0.01,0.5,length.out = 10)

for (i in 1:10){
mcmc = random.walk.mcmc(n = n0, sigma = c(20,sigma2[i]), x0 = c(3800, 6))
diff2 = mcmc[, 2][-1] - mcmc[, 2][-n0]
theta_mixing[i] =  mean(diff2^2)
}
```

1.10 Posterior Histogram

```
set.seed(440)
n0 = 10000
mcmc = random.walk.mcmc(n = n0, sigma = c(56,0.119), x0 = c(3800, 6))
par(mfrow = c(1, 2),mar=2.5 * c(1, 1, 1, 0.1),mfrow = c(1,2),cex = 0.6)
hist(mcmc[, 1], main = "scale",probability = TRUE)
hist(mcmc[, 2], main = "shape",probability = TRUE)
```

1.11 Conclusion

```
par(mar=c(4,4,1,0.5),mfrow = c(1,2),cex = 0.6)
hist(x = Solids, freq = FALSE,
     main = "Histogram of Solids with Frequentest")
gamma_x = rgamma(length(Solids),scale = 3748.82,shape = 5.8)
lines(x = density(x = Solids), col = "red")
lines(x = density(x = rgamma(length(Solids),scale = 3748.822 ,
                             shape = 5.872)), col = "blue")
lines(x = density(x = rgamma(length(Solids),scale =temp[2, 1] ,
                             shape = temp[1, 1])), col = "black",lty = 2)
lines(x = density(x = rgamma(length(Solids),scale = temp[2,2] ,
                             shape = temp[1,2])), col = "black",lty = 2)
legend(50000,0.00004, title = "Density",
       legend=c("Dataset", "MLE","95% CI"),
       col=c("red", "blue","black"), lty=c(1,1,2), cex=0.6)

hist(x = Solids, freq = FALSE,
     main = "Histogram of Solids with Bayesian")
gamma_x = rgamma(length(Solids),scale = 3760.377,shape = 5.875)
lines(x = density(x = Solids), col = "red")
lines(x = density(x = rgamma(length(Solids),scale = 3748.822 ,
```

```
                              shape = 5.872)), col = "blue")
lines(x = density(x = rgamma(length(Solids),scale =3737.399,
                              shape = 5.824)), col = "black",lty = 2)
lines(x = density(x = rgamma(length(Solids),scale = 3783.354 ,
                              shape = 5.892)), col = "black",lty = 2)
legend(50000,0.00004, title = "Density",
       legend=c("Dataset", "Point Estimate","95% CI"),
       col=c("red", "blue","black"), lty=c(1,1,2), cex=0.6)
```