PSTAT 194CS: Independant Midterm Project

Lucas Webster

May 14, 2024

Choice of Random Variables

I selected the random variables log-normal(0, 0.32), Weibull(3, 3), and the inverse gamma(2, 3). I selected the log-normal as we have had much practice with a normal distribution, so I figured taking a manipulation of the log-normal could be interesting to work with. Furthermore, because it is simply the log of a Normal Random Variable, its support becomes strictly positive, which would seem to make it a prime distribution to work with using the accept-reject method. Upon further-research, I learned that the gamma distribution takes a similar shape to that of the log-normal, which furthered my push to use it for the accept reject method. I selected the weibull distribution, as it had appeared in my PSTAT 120A and PSTAT 120B courses without much context, and Isaiah mentioned it as well. It has a closed-form CDF, which made it seem like a good choice for the inverse CDF method. Lastly, I chose the Inverse Gamma distribution as its transformation is as the name implies, an inverse of the Gamma, which made it sound like a good distribution to perform the transformation method with. Furthermore, a gamma distribution can be generated from exponential distributions, which makes it seem even more intriguing to perform the transformation method on, as it turns the process into a double transformation of sorts. Therefore, the only random number generation I will rely on R for in this project is 'runif', 'rexp', and 'rgamma' (after demonstrating taking random samples from a gamma using an rexp). This was discussed with TA Isaiah Katz and approved during Office Hours on Thursday May 9th 2024.

Random Number Generation Methods

Transformation:

The transformation method takes a distribution we know how to sample from already, and performs manipulations on the random variable to create a new random variable upon which it is proven that the manipulation of the random variable creates. In the case of a Gamma Distribution ~ X, the Inverse Gamma Distribution is distributed as 1/X. So therefore, the inversions of samples from the Gamma Distribution are samples from the Inverse Gamma Distribution

Accept-Reject:

The accept-reject method takes two distributions, one we currently know to sample from, and another of which we are unable to readily sample from, to estimate samples of the distribution for which we are unable to readily sample from. We define the pdf of the random variable that is difficult to sample from as q(y) and the pdf of the random variable we can readily sample from as p(x). Values from the support of the difficult to sample random variable are y, and values from the support of the easy to sample random variable are x. We then find the maximum of the the ratio $\frac{q(x)}{p(x)}$. Lets call this value m. We take the reciprocal of m, and multiply it by a value of $\frac{q(y_0)}{p(y_0)}$ (y_0 is a sample point generated from the easy to sample random variable). If this value is greater than a realization of a uniform random variable with parameters (0,1). We keep the value y as a random value from the difficult to sample random variable. If it's not greater than this realization of the uniform, we discard it and don't consider it a random value from the difficult to sample from random variable. We repeat this process many times until we collect our desired number of random sample values for the difficult to sample from distribution.

Inverse CDF:

The inverse CDF method takes the CDF of a probability distribution, and sets it equal to a realization of a sample from a uniform(0, 1) random variable. Then, the equation is inverted, so the inverse of the CDF of the chosen probability distribution has a realization of a uniform(0, 1) as its variable. This realization of a uniform(0, 1) evaluated on the inverse CDF of the chosen probability distribution creates a random sample from said probability distribution.

Application of Random Number Generation:

Transformation - Inverse Gamma(2, 3)

W ~ Gamma(2, 1/3) W has a pdf of $\frac{1}{\Gamma(2)(\frac{1}{3})^2}w^{2-1}e^{-3w}$. H ~ InvGamma(2, 3). H $= \frac{1}{W}$. Therefore, to generate samples for H, we need to draw random samples from a W, and take their reciprocal. As discussed in section, for $V_1, V_2, ..., V_\alpha$ ~iid exp(1) $gamma(\alpha, \beta) = \beta \sum_{i=1}^{\alpha} V_i$. Therefore, to draw random samples from W, we need to sum 3 random samples from V_i and multiply by β. Therefore, H $= \frac{1}{\frac{1}{3}\sum_{i=1}^{2} V_i}$.

Accept Reject - Log-Normal(0, 0.32)

Y ~ log-normal(0, 0.32). Y has a pdf of $\frac{1}{y\sqrt{2\pi * 0.32}}e^{-\frac{(ln(y-0))^2}{2*0.32}}$. Lets define this pdf as q(y). A distribution which is similar to Y in shape and identical in support is a Gamma(4, 0.25) X ~ Gamma(4, 0.25) X has a pdf of $\frac{1}{\Gamma(4)0.25^4}x^{4-1}e^{\frac{-x}{0.25}}$. Lets define this pdf as p(x). We now need to find $m = max(\frac{q(x_1)}{p(x_1)})$. As discussed in office hours with Isaiah, we can approximate this value visually. Doing so we take $x_1 = 2.2$. This means that $m = 1.772125507$ and $\frac{1}{m} = 0.5642941182$. We now create a realization of a uniform(0, 1) random variable and set up the inequality $u < \frac{1}{m}\frac{q(y)}{p(y)}$ with a value from the support of Y. If this inequality is is held true by the value y, we keep y as a random sample from Y, otherwise we discard the value. We repeat this process to collect a bunch of random samples from Y.

Inverse CDF - Weibull(3, 1)

CDF of Weibull(3, 3) $= 1 - e^{(\frac{-x}{3})^1}$ u is a realization of uniform(0, 1). $1 - e^{(\frac{-x}{3})^1} = u$ Solving for u results in... $\frac{-ln(-u+1)^{\frac{1}{1}}}{3} = x$

We now plug in random samples from a uniform(0, 1) into the expression on the left (the inverse CDF of a Weibull(3, 1)), which will create random samples from a Weibull(3, 1) random variable.

Challenges

1. I didn't realize that the inversion of the Gamma to Inverse Gamma meant I had to take the reciprocal of my beta parameter. This meant my values were scaled far lower than they should have been until I incorporated 1/beta into my inverse gamma generation function. After computing the transformation using the cdf method, I derived the relationship between the Gamma and the Inverse Gamma.

2. My q(x)/p(x) was not easily differentiable, which made finding an m difficult. After speaking with Isaiah, I found my m visually using desmos. Additionally, it took some trial and error to find a gamma and log-normal distribution which looked similar to each other.

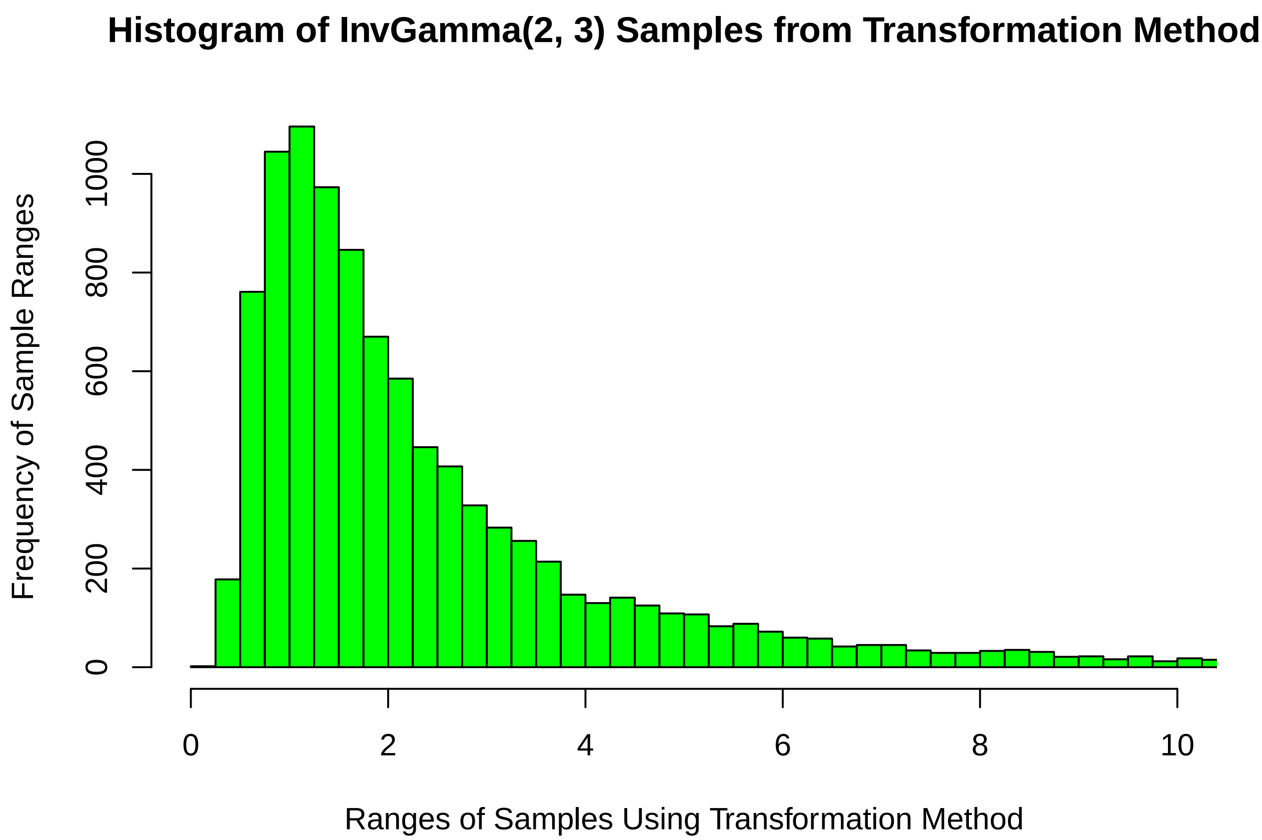
3. When using the Inverse CDF Method for the Weibull Distribution, I originally started with 3 as my parameter for k. However, this resulted in me raising negative values to a fractional power. After talking with Isaiah, to avoid the use of complex numbers, I changed k to be 1 to avoid this problem.

Appendix

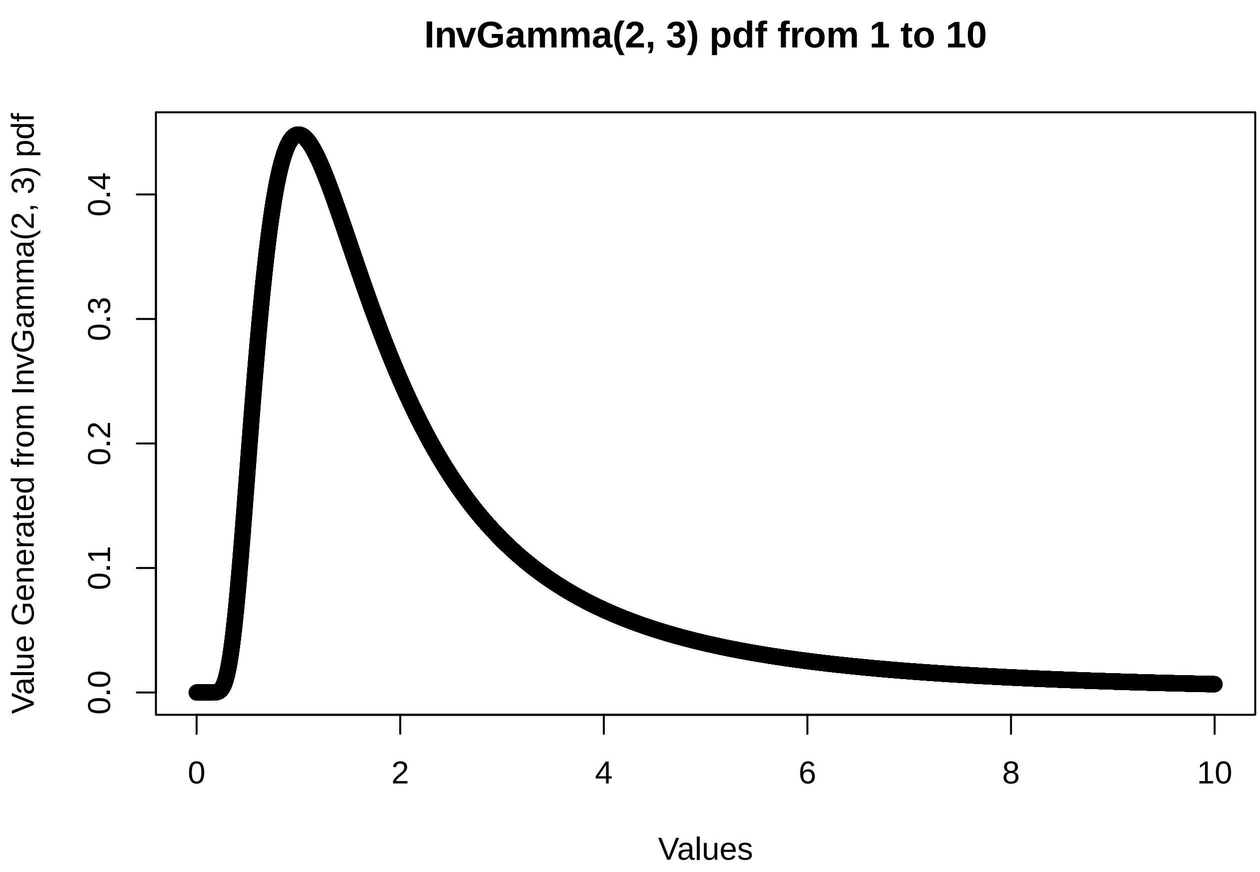
This Appendix will start with my code showing my work for the above random number generation methods, and links to websites I used as reference for this project, and finally be followed by screenshots of my scratch work, a Desmos graph of the Log-Normal(0, 0.32) pdf and the Gamma(4, 0.25) pdf.

Code for InvGamma(2, 3) with Transformation Method

```r
set.seed(100)

# Generating n samples for a Gamma(alpha, beta)
gamma_generation <- function(alpha, beta, n){
  gamma_vec <- rep(0, times = n)
  for(i in seq(from= 1, to = n, by = 1)){
    exp_vec <- rexp(alpha, rate=1)
    exp_sum <- sum(exp_vec)
    gamma_vec[i] <- exp_sum*beta
  }
  return(gamma_vec)
}

# Generating n InvGamma(alpha, beta) Samples through inverting the samples from a Gamma(alpha, beta)
inv_gamma_generation <- function(alpha, beta, n){
  inv_gamma_vec <- 1/gamma_generation(alpha, 1/beta, n)
  return(inv_gamma_vec)
}

#Graphing the InvGamma pdf
seq_var <- seq(0, 9.999, by=0.001)
inv_gamma <- function(alpha, beta, x){
  return(((beta^alpha)/factorial(alpha-1))*((1/x)^(alpha+1))*exp(-beta/x))
}
inv_gamma23_pdf <- inv_gamma(2, 3, seq_var)
inv_gamma23_samples <- inv_gamma_generation(2, 3, 10000)

hist(inv_gamma23_samples, xlab='Ranges of Samples Using Transformation Method', ylab='Frequency of Sampl
```



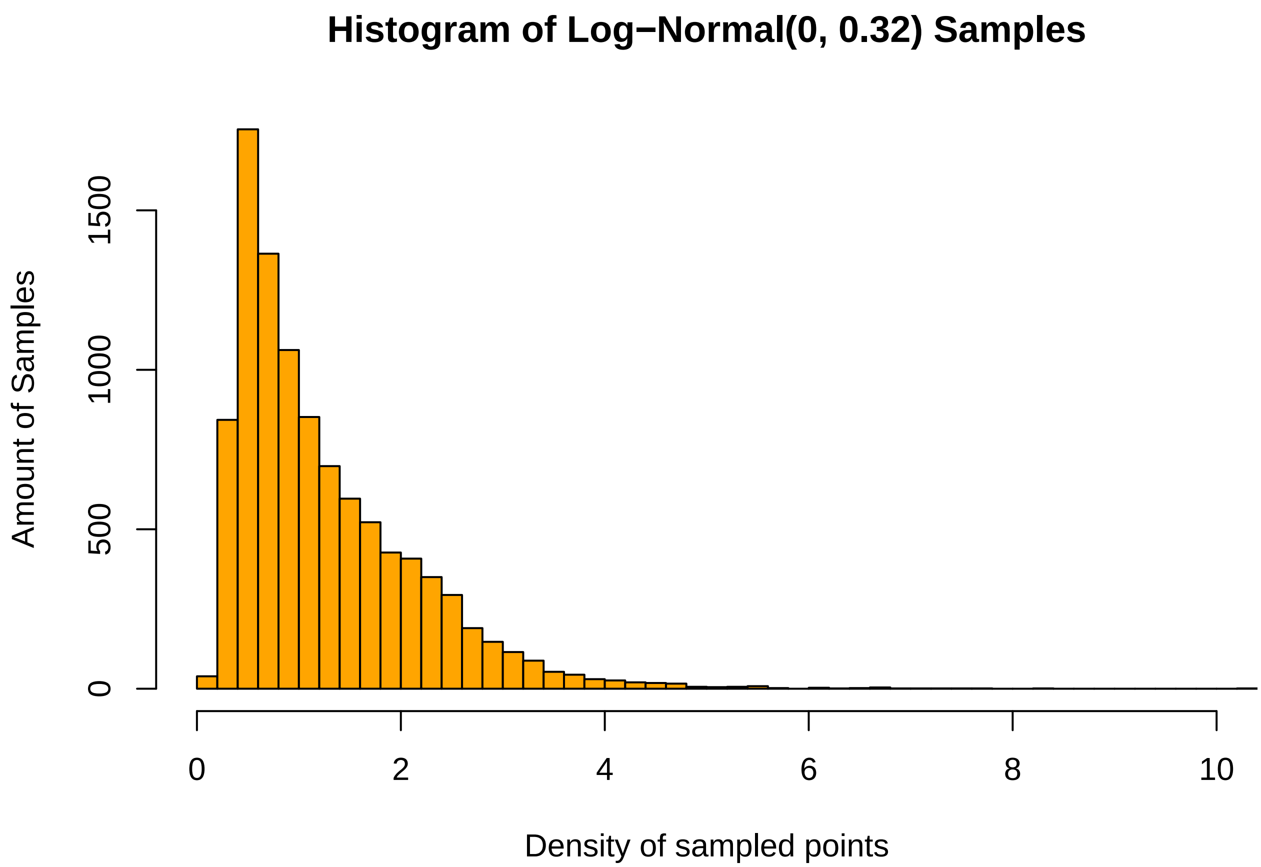

```
plot(x = seq_var, y = inv_gamma23_pdf, xlab='Values', ylab='Value Generated from InvGamma(2, 3) pdf', ma
```



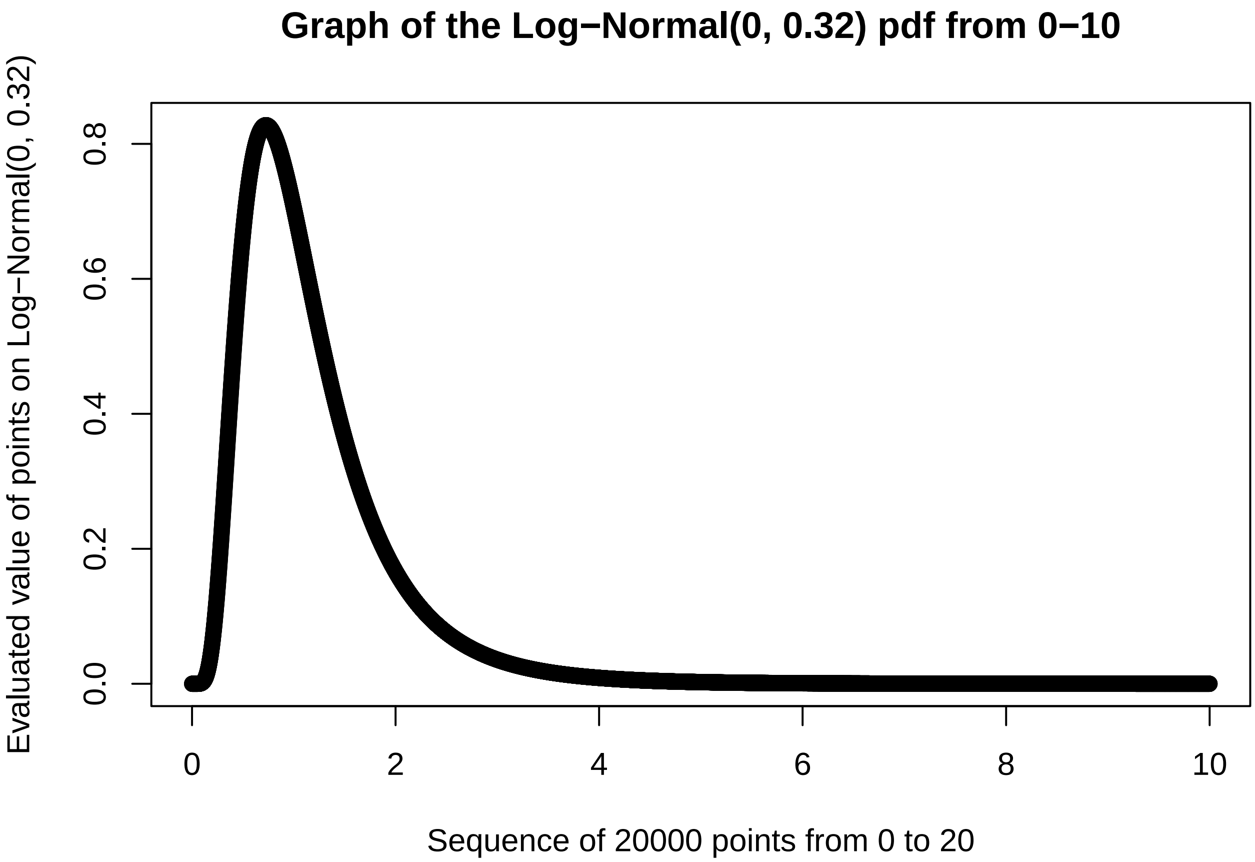
Code for Log-Normal(0, 0.32) with Accept-Reject Method

```
#Function for the Hard to Sample Function, the Log-Normal(0, 0.32)
q <- function(y){
   return((1/(y*sqrt(2*pi*0.32)))*exp((log(y)^2)/-0.64))
 }

#Value of m
m <- 1.772125507

#Accept Reject Function
accept_reject <- function(n){
  accepted_samples <- c()
  #In order to generate n samples
  while(length(accepted_samples) < n){
    unif_point_compare <- runif(1, 0, 1)

    y <- rgamma(1, shape = 0.25, rate = 0.25)

    #If point accepted, add it to vector of samples (also R's shape parameter is inverse my pdf)
    if(unif_point_compare < (1/m)*(q(y)/dgamma(y, shape=0.25, rate=0.25))){
      accepted_samples <- append(accepted_samples, values = y)
    }
  }
  return(accepted_samples)
```

```r
}

log_normal_samples <- accept_reject(10000)

hist(log_normal_samples, xlab='Density of sampled points', ylab='Amount of Samples', main='Histogram of
```

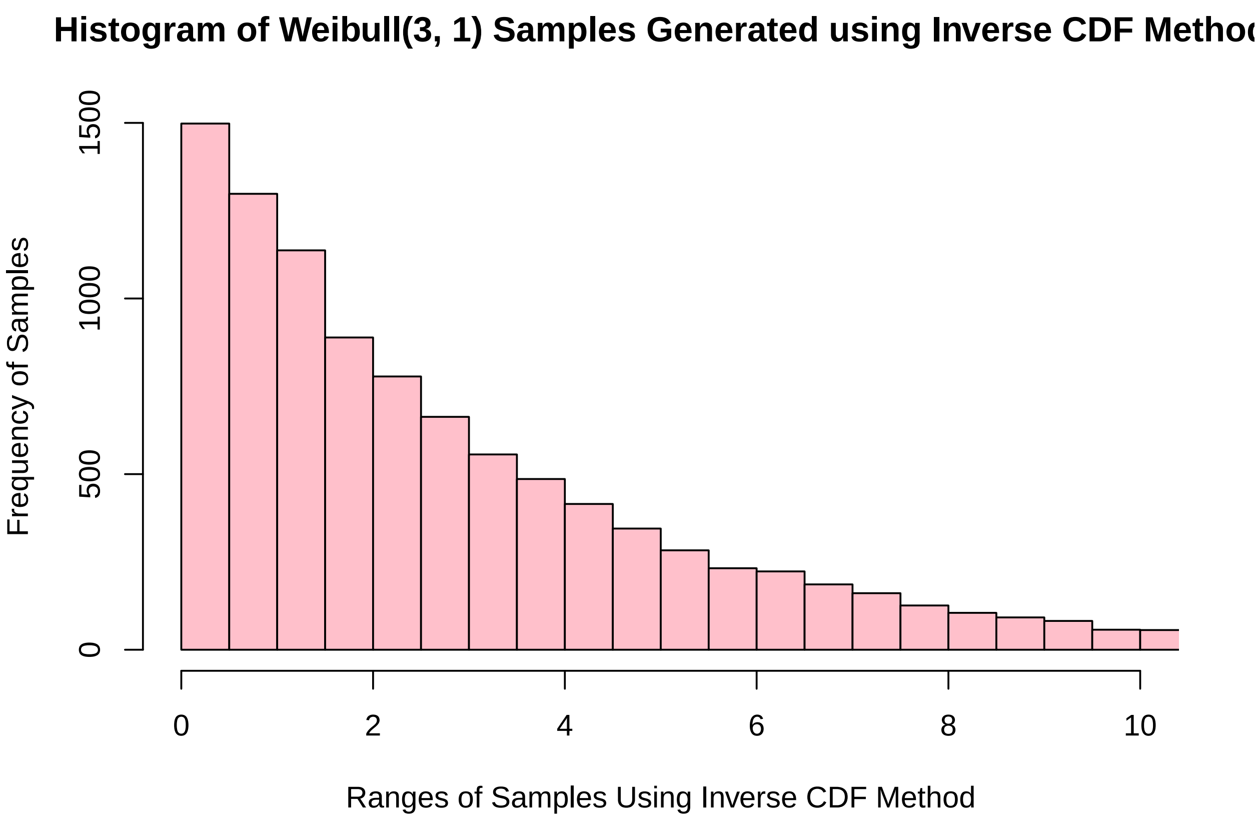



```r
plot(x = seq_var, y = q(seq_var), xlab='Sequence of 20000 points from 0 to 20', ylab='Evaluated value of
```

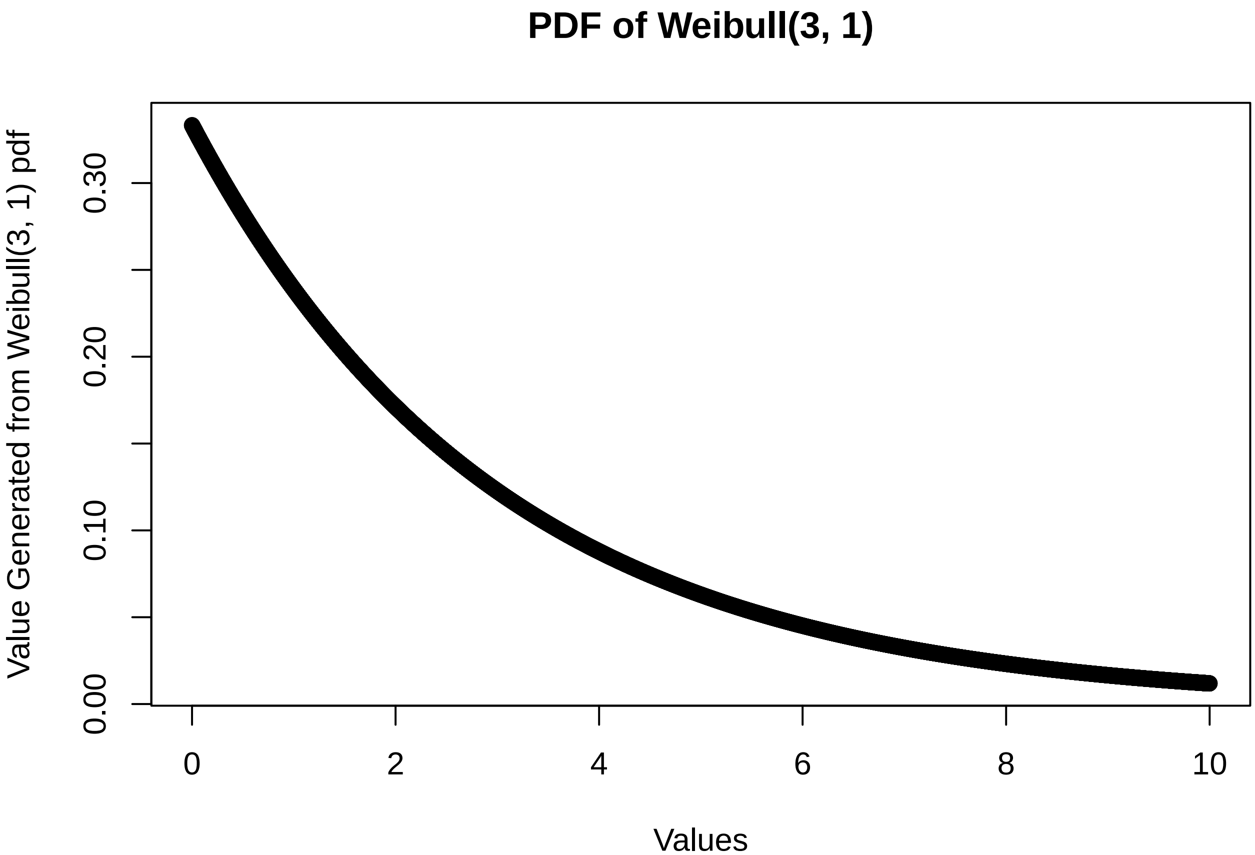


Code for Weibull(3, 3) with Inverse CDF Method

```
#Function for the inverse cdf of a Weibull(l, k)
inv_weibull_cdf <- function(l, k, x){
  return(-l*(log(-x+1)^(1/k)))
}

#Function to generate samples from Weibull(l, k) using inverse cdf
weibull_sample <- function(l, k, n){
  unifs <- runif(n, 0, 1)
  return(inv_weibull_cdf(l, k, unifs))
}

#pdf of a Weibull(l,k)
weibull_pdf <- function(l, k, x){
  return(((k/l)*(x/l)^(k-1))*exp(-(x/l)^k))
}

weibull33_pdf <- weibull_pdf(3, 1, seq_var)
weibull33_samples <- weibull_sample(3, 1, 10000)

hist(weibull33_samples, xlab='Ranges of Samples Using Inverse CDF Method', ylab='Frequency of Samples',
```




```
plot(x = seq_var, y = weibull33_pdf, xlab='Values', ylab='Value Generated from Weibull(3, 1) pdf', main=
```



References

Reference 1 Stack Exchange I used this as a reference to help me find a distriubtion similar to the Log-Normal.

Reference 2 Wikipedia I used this to learn more about the Log-Normal distribution.

Reference 3 Leemis Distribution Chart I used this site to serve as a general guide for finding distributions and information about them.

Screenshots (Scratchwork and Desmos)

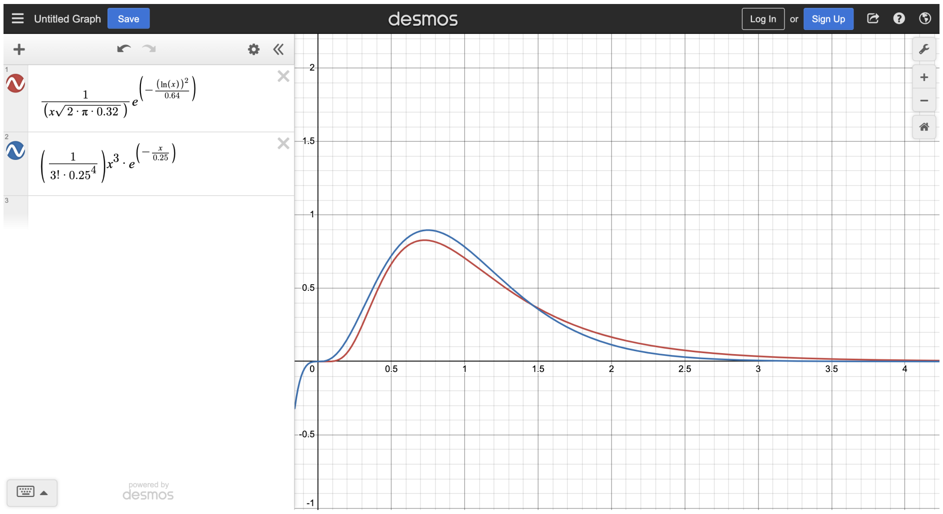


Figure 1: Desmos - Screenshot

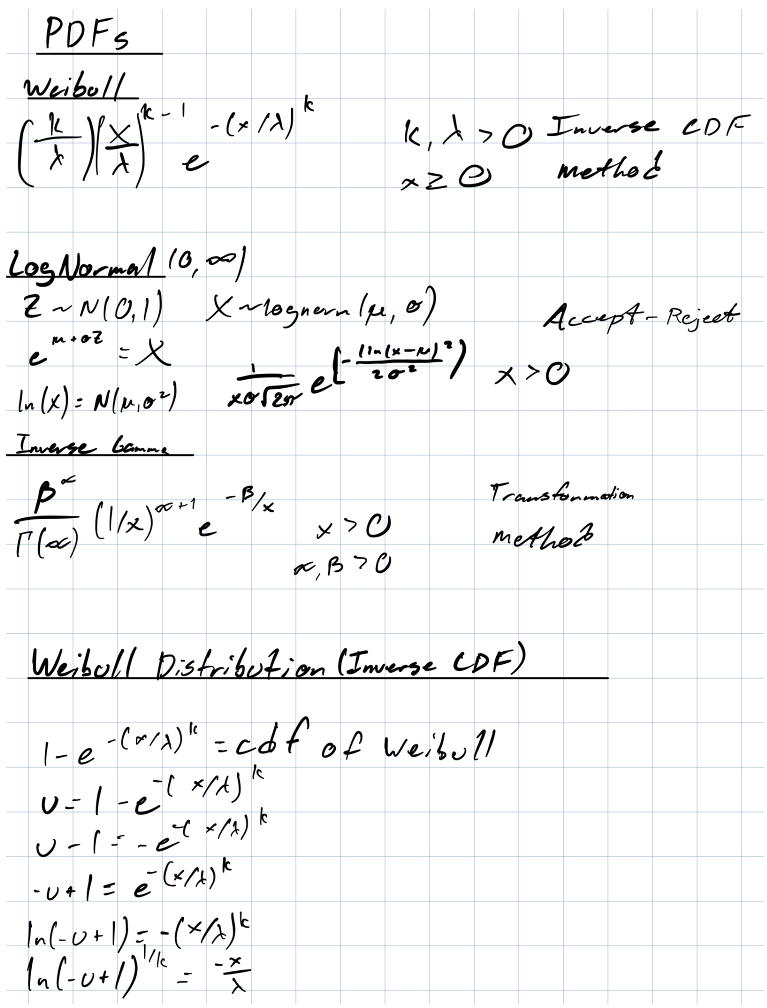
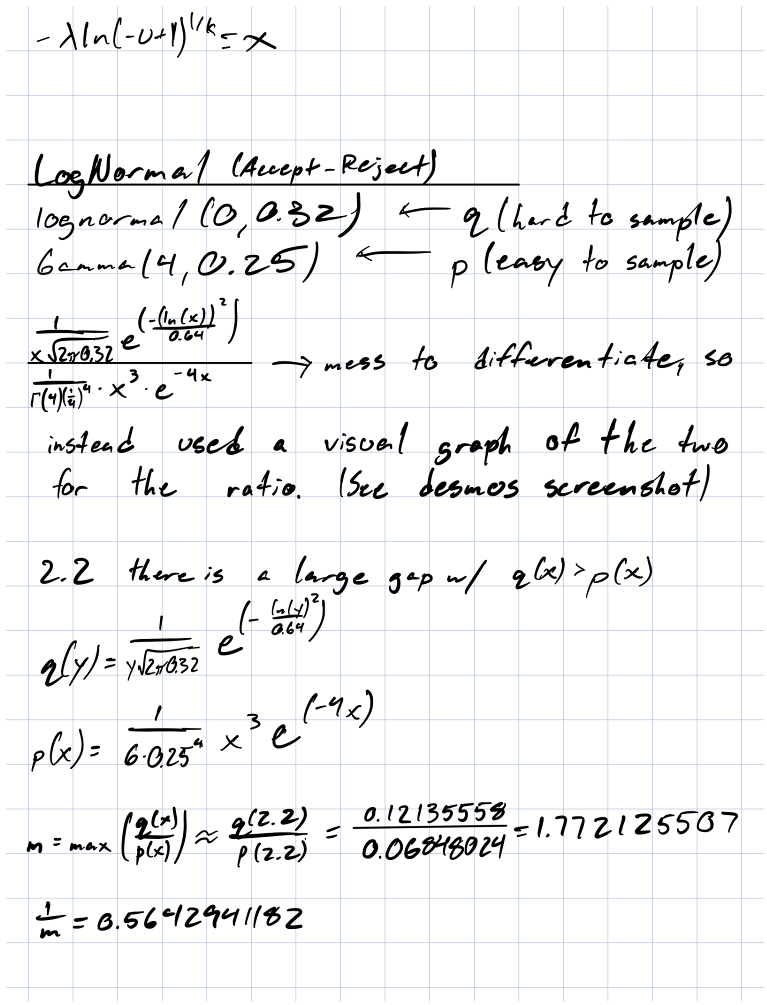


Figure 2: Scratchwork - Screenshot



$-\lambda \ln(-U+1)^{1/k} = x$

LogNormal (Accept-Reject)

lognormal$(0, 0.32)$ ← q (hard to sample)

Gamma$(4, 0.25)$ ← p (easy to sample)

$$\frac{\frac{1}{x\sqrt{2\pi 0.32}} e^{\left(-\frac{(\ln(x))^2}{0.64}\right)}}{\frac{1}{\Gamma(4)(\frac{1}{4})^4}\cdot x^3 \cdot e^{-4x}} \rightarrow$$ mess to differentiate, so instead used a visual graph of the two for the ratio. (See desmos screenshot)

2.2 there is a large gap w/ $q(x) > p(x)$

$$q(y) = \frac{1}{y\sqrt{2\pi 0.32}} e^{\left(-\frac{(\ln(y))^2}{0.64}\right)}$$

$$p(x) = \frac{1}{6 \cdot 0.25^4} x^3 e^{(-4x)}$$

$$m = \max\left(\frac{q(x)}{p(x)}\right) \approx \frac{q(2.2)}{p(2.2)} = \frac{0.12135558}{0.06848024} = 1.772125507$$

$$\frac{1}{m} = 0.5642941182$$

Figure 3: Scratchwork - Screenshot

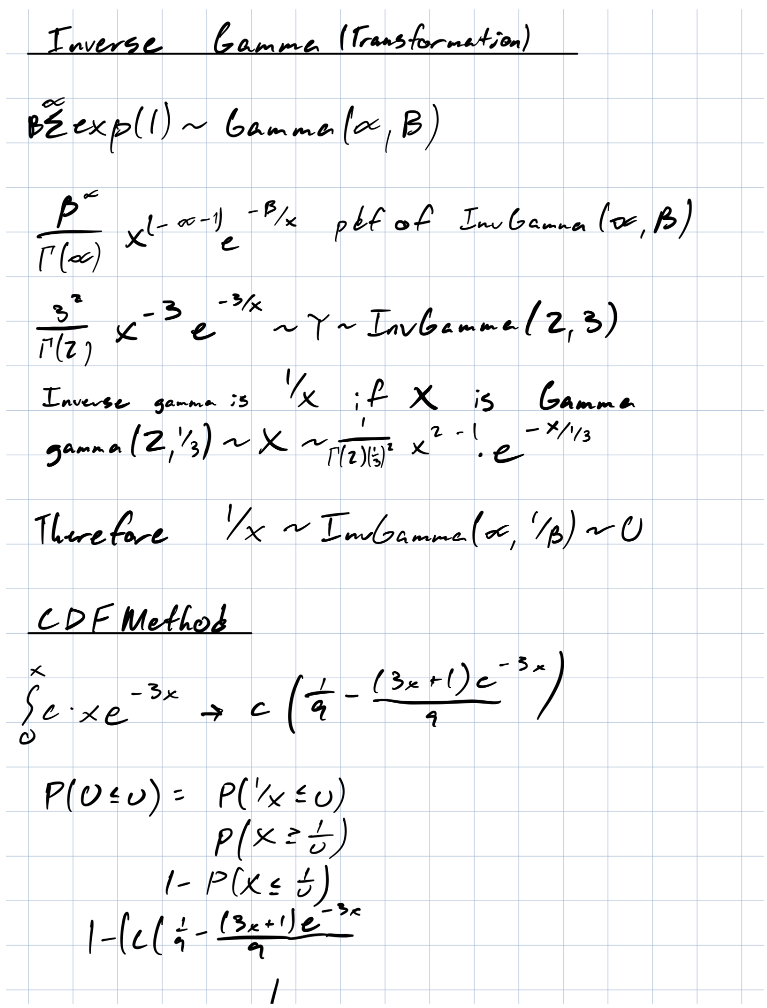


Figure 4: Scratchwork - Screenshot

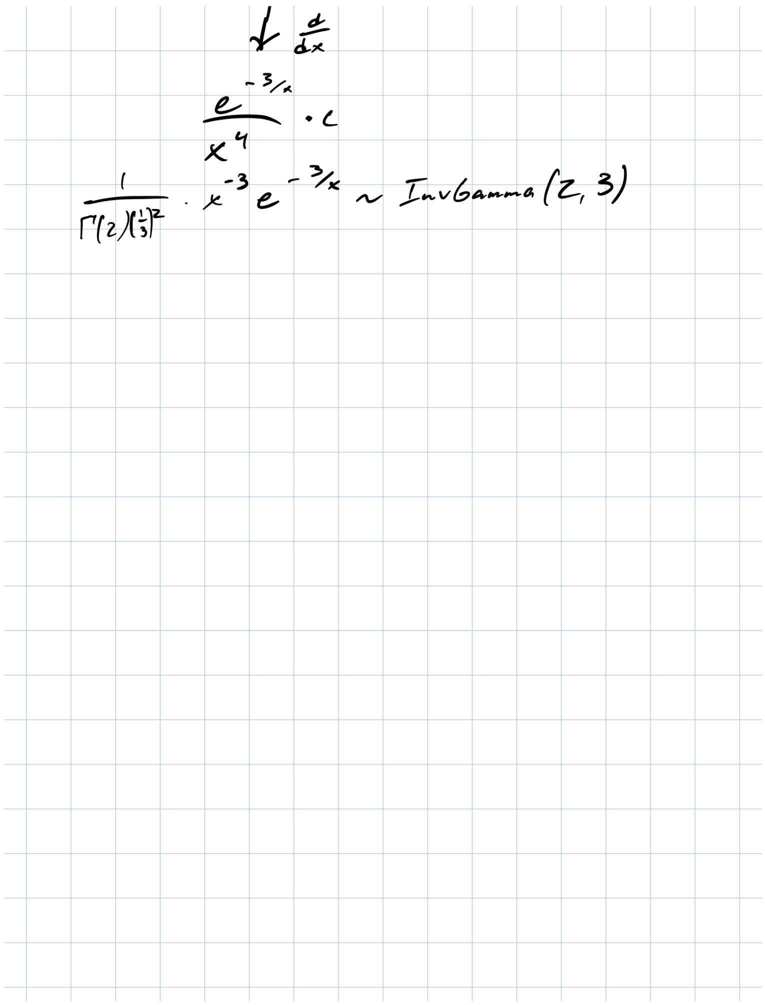


Figure 5: Scratchwork - Screenshot