Computer Lab 3

Tim Yuki Washio

11/19/2020

Question 1: Stable Distribution

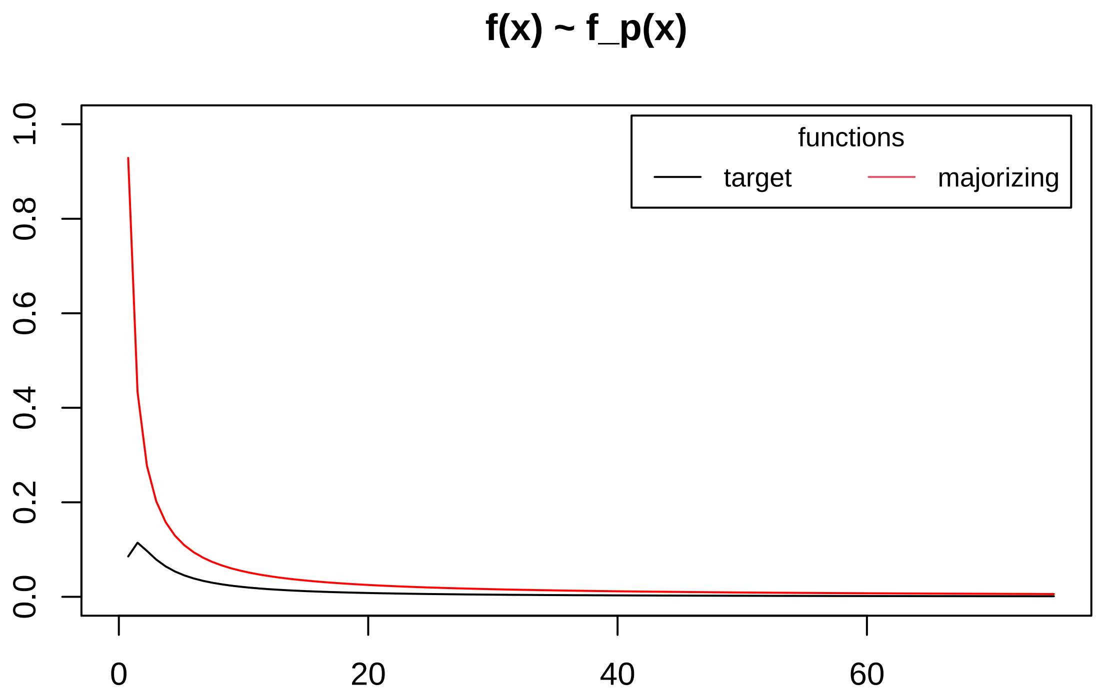
1.

Plotting $f(x)$ with $c = 2$ and $f_p(x)$ with $\alpha = 2$ and $T_{min} = 2$

```
c = 2
t_min = 1.5
alpha = 1.1

plot(1, xlim = c(0,75), ylim = c(0, 1), type = "n", xlab = "", ylab = "", main = "f(x) ~ f_p(x)")
curve(eval(c) * (sqrt(2 * pi)^(-1)) * exp(-eval(c)^(2) / (2 * x)) * x^(-3/2), from=0, to=75, add=TRUE, c
curve((eval(alpha) - 1 / eval(t_min)) * (x / eval(t_min))^(-eval(alpha)), from=0, to=75, add=TRUE, col=

legend("topright", inset=.02, title="functions",
       c("target","majorizing"), horiz=TRUE, cex=0.8, col = 1:2, lty = 1)
```



The power-law distribution should not be used by itself because for small values of x $f_p(x) = \infty$. If we sample from our majorizing distribution in this area, we will get a huge number of rejections. It might be better to combine the power-law distribution with another distribution that majorizes our target function in that area. We could use a uniform distribution with support $(0, T_{min}$ in addition to our given distribution with support (T_{min}, ∞).

Power-law distribution values:

$$\frac{df(x)}{dx} = \frac{ce^{(-c^2)/2x}(c^2 - 3x)}{2\sqrt(2\pi)x^{7/2}}$$

This has one root for $c \neq 0, x = \frac{c^2}{3}$.

$$max(f(x)) = f(\frac{c^2}{3})$$

$$f_p(x) = f(\frac{c^2}{3})$$

$$x = T_{min}$$

Question 2: Laplace Distribution

Write a code generating random numbers from the double exponential distribution

The double exponential distribution is given by the followinf formula:

$$DE(\mu, \alpha) = \frac{\alpha}{2} exp(-\alpha \mid x - \mu \mid)$$

To get to the inverse CDF, $F^{-1}(p)$, of this function we first need to get the CDF $F(x)$

$$F(x) = \int_{-\infty}^{x} f(u)du = \begin{cases} \frac{1}{2}\mathrm{e}^{(\frac{\mathrm{x}-\mu}{\alpha})} & \text{if } \mathrm{x} < \mu \\ 1 - \frac{1}{2}\mathrm{e}^{(\frac{\mathrm{x}-\mu}{\alpha})} & \text{if } \mathrm{x} \geq \mu \end{cases}$$

After transformation, we get

$$F(x) = \frac{1}{2} + \frac{1}{2} sgn(x - \mu)\left(1 - exp\left(-\frac{|x - \mu|}{\alpha}\right)\right)$$

Now we calculate the inverse CDF $F^{-1}(p)$ by using the CDF $F(x)$. We set:

$$y = \frac{1}{2} + \frac{1}{2} sgn(x - \mu)\big(1 - \exp(-\alpha \mid \mathrm{x} - \mu \mid)\big)$$

And we now solve for x to obtain the inverse CDF. We have two cases. When x $\geqslant \mu$ where we then have:

$$x = \mu - \frac{1}{\alpha} \ln(2 - 2\mathrm{y})$$

and the case when x $< \mu$ when we have:

$$x = \mu + \frac{1}{\alpha} \ln(2\mathrm{y})$$

We combine both expression and express them with respect to the sign of de difference between x and μ. We obtain the following:

$$F^{-1}(y) = \mu.sgn(x - \mu)\frac{1}{\alpha} ln(1 + sgn(x - \mu) - sgn(x - \mu)2y)$$

However, we would like an expression that does not depend on the sign of $x - \mu$. We try to find a quantity related to it but with respect to y. We investigate the critical value of y when the sign of $x - \mu$ changes (i.e. we look for $sgn(x - \mu)\frac{1}{\alpha} ln(1 + sgn(x - \mu) - sgn(x - \mu)2y) = x - \mu$ when x is greater and smaller than μ.)

We find that $sgn(x - \mu) = sgn(y - \frac{1}{2})$. We can simplify the expression of $F^{-1}(u)$ and finally obtain:

$$F^{-1}(u) = \mu - \alpha.sgn(u - \frac{1}{2}) ln(1 - 2|u - \frac{1}{2}|)$$

```
# Given density function
laplace_density <- function(x, mu, alpha) {
  result = (alpha/2)*exp(-alpha*abs(x-mu))
  return(result)
}

# Calculated inverse CDF for Laplace function
```

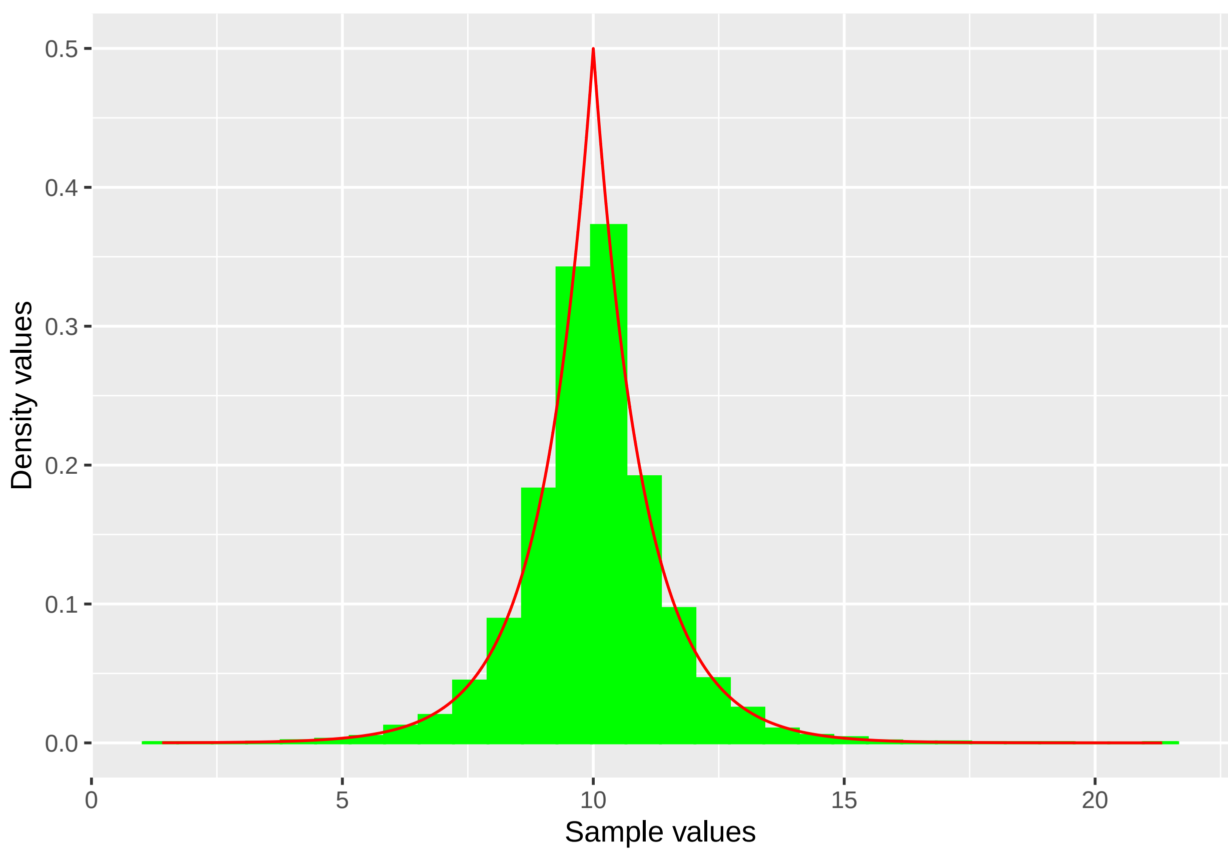
```r
laplace_inv_cdf = function(p, mu, alpha){
  result = mu-alpha*sign(p-0.5)*log(1-2*abs(p-0.5))
  return(result)
}

# Random number generation with n=10000, mu=10, alpha=1
n = 10000
unif_sample = runif(n)
sample = laplace_inv_cdf(unif_sample, 10, 1)
sample_density = laplace_density(sample, 10, 1)
```

```r
# Plotting the histogram
hist_plot <- ggplot() +
geom_histogram(aes(x = sample, y = ..density..), col = "green", fill="green") +
geom_line(aes(x = sample, y = sample_density), col = "red") +
xlab("Sample values") + ylab("Density values")

hist_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From this graph, we can see that our sampled numbers follow the distribution.

Acceptance rejection method using DE(0,1) as a majorizing density for N(0,1).

The main task to solve this exercise is to find the majorizing constant c such that

$$c \cdot f_M(x) \geq f_T(x)$$

Where $f_M(x)$ is the majorizing density and $f_T(x)$ is the density we wish to sample from (target density). This inequality must hold true for all x that are on the support of the target function. We must choose c to be large enough for the inequality to hold true, but not so large that the rejection rate for the acceptance/rejection algorithm becomes too great. Setting the parameters in both densities to be (1,0) and setting our inequality, we have:

$$c \cdot \frac{1}{2} \exp -|x| \geq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

We solve for c and obtain:

$$c \geq \sqrt{\frac{2}{\pi}} \exp\left(|x| - x^2/2\right)$$

We can find a solution for c for $x > 0$ because we have an even function on the right-hand side and the other maximum will be attained at $x = -x_{positive}$ with the same value for c. The expression maximizes for $x = 1$ and yields:

$$c_{major} = \sqrt{\frac{2e}{\pi}}$$

```
# Takes as arguments the following:
#       · The number n of observations to generate;
#       · The target density f (standar normal distribution);
#       · The majorizing density g (standard laplace distribution);
#       · The function rg to generate random numbers according to the density g;
#       · The majorizing constant k.


# random_norm_AR <- acc_rej_norm(2000, dnorm, dlaplace, rlaplace, k)
# random_norm_st <- rnorm(2000)

set.seed(12345)
c_major <- sqrt( (2*exp(1)) / pi )


accept_reject <- function(n){
  results <- rep(0,times = n)
  counter <- 0

  for(i in 1:n){
    reject <- TRUE
    while(reject == TRUE){
      counter <- counter + 1
      random_major <- laplace_inv_cdf(runif(1), 0, 1)
      random_uniform <- runif(1)
      if(random_uniform <= dnorm(random_major)/(c_major*laplace_density(random_major, 0, 1))){
        results[i] <- random_major
```

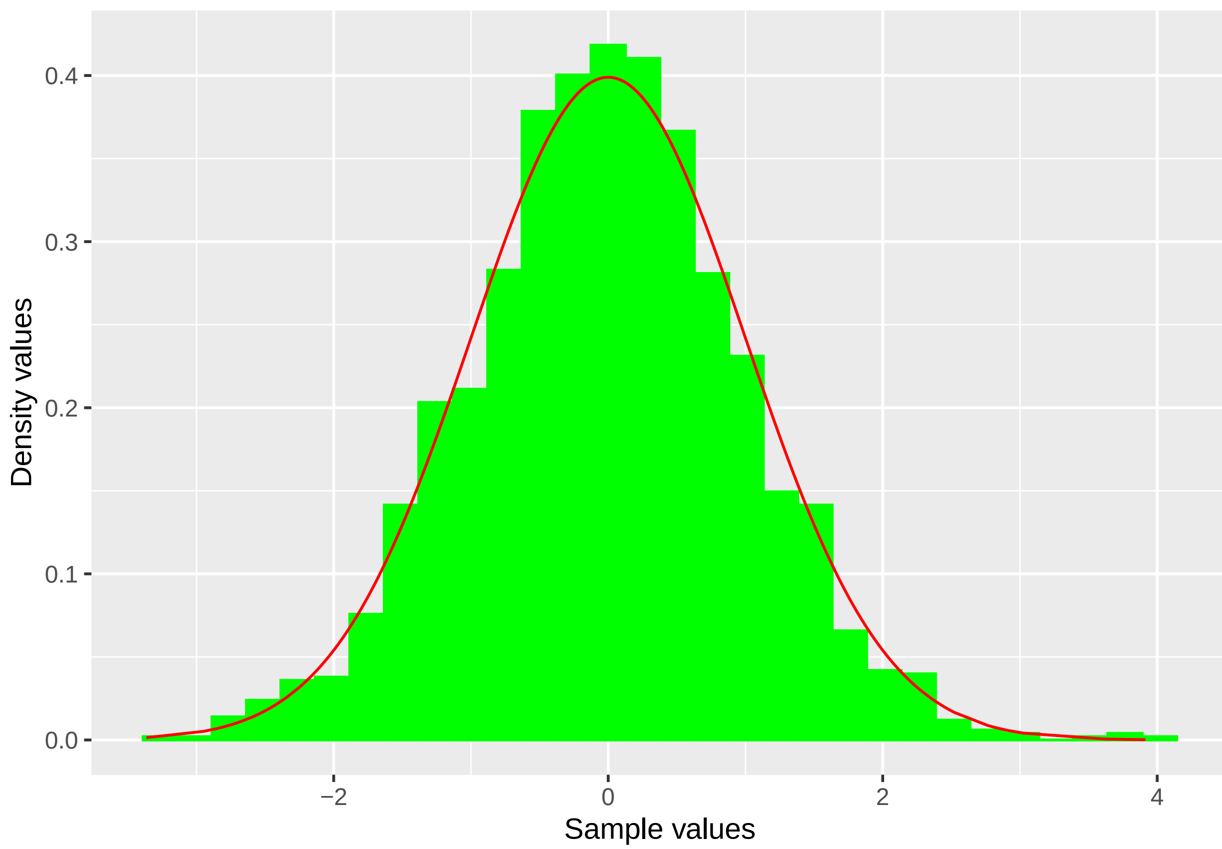
```r
        reject <- FALSE
      }
    }
  }
  return(list(rn = results, draws = counter))
}
```

```r
n = 2000
unif_sample = rnorm(n)
sample = accept_reject(n)


# Plotting the histogram
p1 <- ggplot() +
geom_histogram(aes(x = sample$rn, y = ..density..), col = "green", fill="green") +
geom_line(aes(x = sample$rn, y = dnorm(sample$rn)), col = "red") +
xlab("Sample values") + ylab("Density values")

p1
```

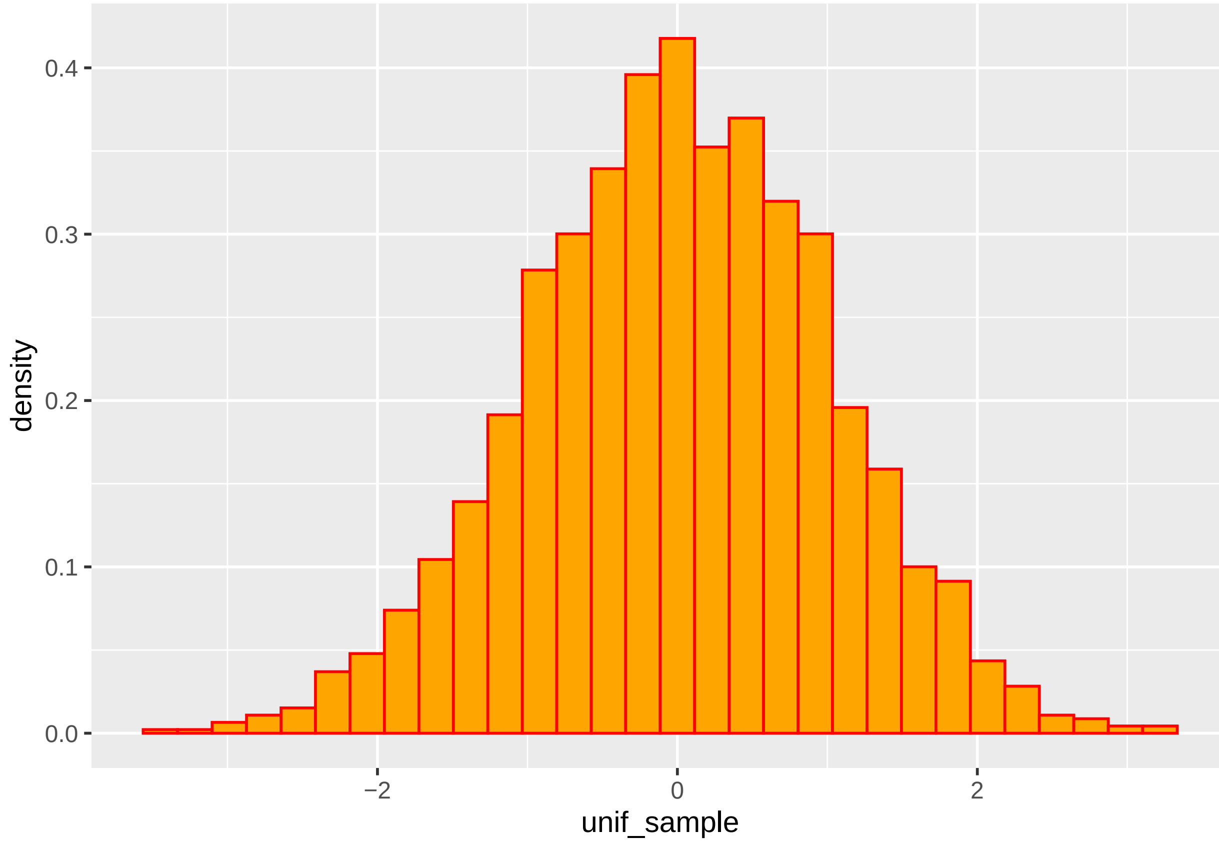
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
# historgram using rnorm
p2 <- ggplot() +
  geom_histogram(aes(x= unif_sample, y= ..density..), col = "red", fill = "orange")

p2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
c = 2
t_min = 1.5
alpha = 1.1

plot(1, xlim = c(0,75), ylim = c(0, 1), type = "n", xlab = "", ylab = "", main = "f(x) ~ f_p(x)")
curve(eval(c) * (sqrt(2 * pi)^(-1)) * exp(-eval(c)^(2) / (2 * x)) * x^(-3/2), from=0, to=75, add=TRUE, c
curve((eval(alpha) - 1 / eval(t_min)) * (x / eval(t_min))^(-eval(alpha)), from=0, to=75, add=TRUE, col=

legend("topright", inset=.02, title="functions",
       c("target","majorizing"), horiz=TRUE, cex=0.8, col = 1:2, lty = 1)
# Given density function
laplace_density <- function(x, mu, alpha) {
  result = (alpha/2)*exp(-alpha*abs(x-mu))
  return(result)
}

# Calculated inverse CDF for Laplace function
laplace_inv_cdf = function(p, mu, alpha){
  result = mu-alpha*sign(p-0.5)*log(1-2*abs(p-0.5))
  return(result)
}

# Random number generation with n=10000, mu=10, alpha=1
n = 10000
unif_sample = runif(n)
sample = laplace_inv_cdf(unif_sample, 10, 1)
sample_density = laplace_density(sample, 10, 1)
# Plotting the histogram
hist_plot <- ggplot() +
geom_histogram(aes(x = sample, y = ..density..), col = "green", fill="green") +
geom_line(aes(x = sample, y = sample_density), col = "red") +
xlab("Sample values") + ylab("Density values")

hist_plot
# Takes as arguments the following:
#       · The number n of observations to generate;
#       · The target density f (standar normal distribution);
#       · The majorizing density g (standard laplace distribution);
#       · The function rg to generate random numbers according to the density g;
#       · The majorizing constant k.


# random_norm_AR <- acc_rej_norm(2000, dnorm, dlaplace, rlaplace, k)
# random_norm_st <- rnorm(2000)

set.seed(12345)
c_major <- sqrt( (2*exp(1)) / pi )
```

```r
accept_reject <- function(n){
  results <- rep(0,times = n)
  counter <- 0

  for(i in 1:n){
    reject <- TRUE
    while(reject == TRUE){
      counter <- counter + 1
      random_major <- laplace_inv_cdf(runif(1), 0, 1)
      random_uniform <- runif(1)
      if(random_uniform <= dnorm(random_major)/(c_major*laplace_density(random_major, 0, 1))){
        results[i] <- random_major
        reject <- FALSE
      }
    }
  }
  return(list(rn = results, draws = counter))
}

n = 2000
unif_sample = rnorm(n)
sample = accept_reject(n)


# Plotting the histogram
p1 <- ggplot() +
geom_histogram(aes(x = sample$rn, y = ..density..), col = "green", fill="green") +
geom_line(aes(x = sample$rn, y = dnorm(sample$rn)), col = "red") +
xlab("Sample values") + ylab("Density values")

p1

# historgram using rnorm
p2 <- ggplot() +
  geom_histogram(aes(x= unif_sample, y= ..density..), col = "red", fill = "orange")

p2
```