Computer Lab 3

group 6

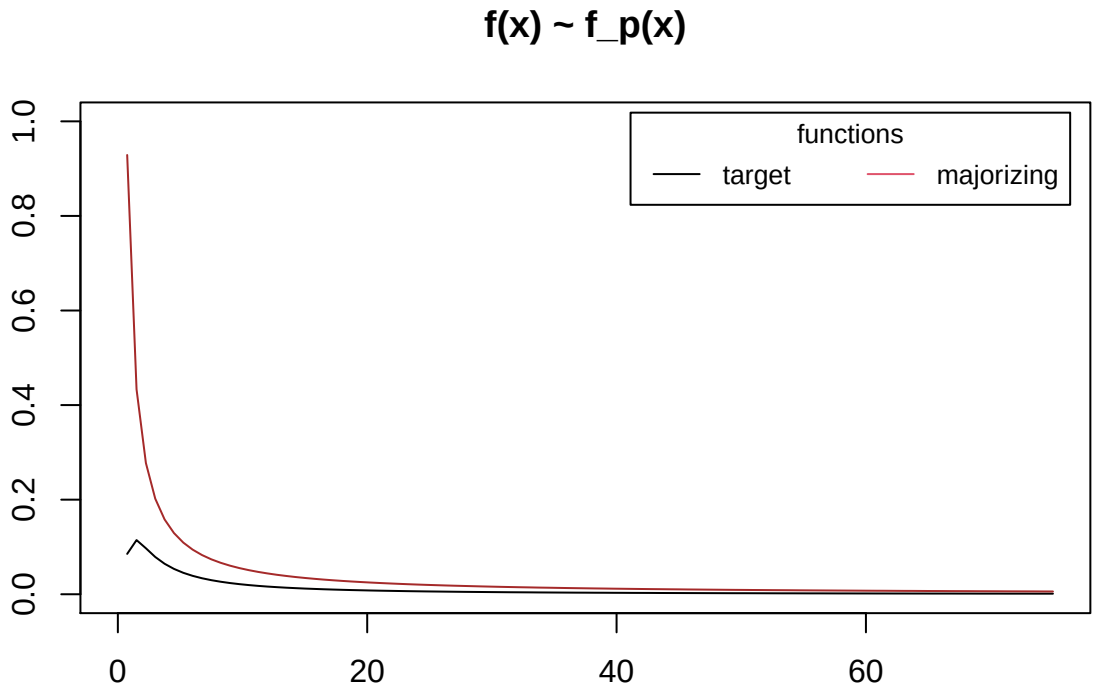
11/23/2020

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, col="black")
curve((eval(alpha) - 1 / eval(t_min))
      * (x / eval(t_min))^(-eval(alpha)),
      from=0, to=75, add=TRUE, col="brown")

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}, ∞).

In order to find a good value for T_{min}, we should try to understand how the target function behaves. We try to identify the location of the maximum and relate this value to c.

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

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

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

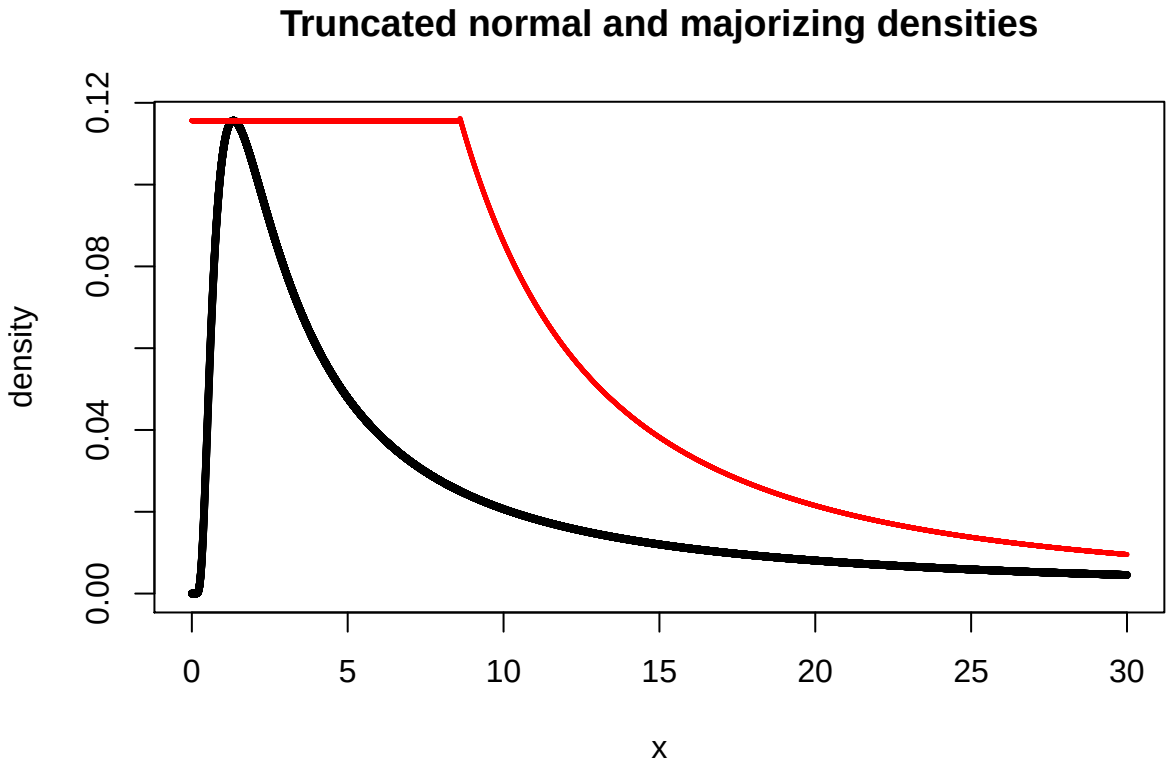
We will set T_{min} to be the value of x for which $f_p(x)$ is the following:

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

with $x_{cutoff} = t_{min}$ from here on out.

We must now set a value for α. With no additional information, other than $\alpha > 1$, we must choose an appropriate alpha for which the majorizing function is greater than our target function for every x on the support (T_{min}, ∞). We choose alpha as small as possible in order to reduce rejections in the acceptance-rejection algorithm. The chosen value of α after visual inspection is 1.3.

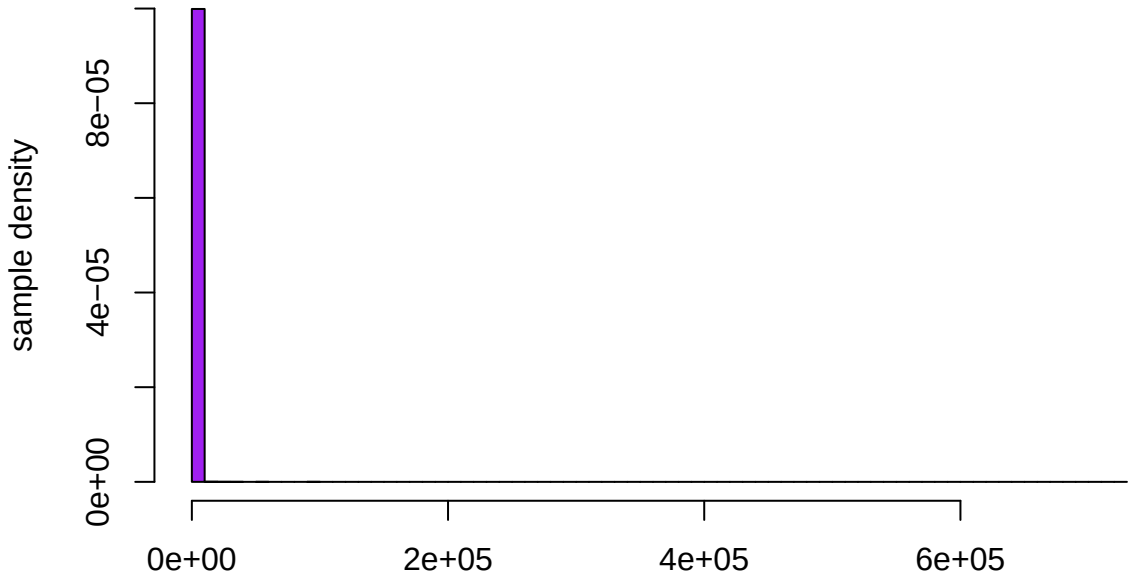
2.



3.

```
Nsample=100000
my_vec = c()
for (i in 1:Nsample) {
  res=NA
  component=sample(1:2,1,prob=c(2/3,1/3))
  if(component==1){
    res=runif(1)
  }
  if(component==2){
    res=rplcon(1, t_min, alpha)
  }
  my_vec = c(my_vec, res)
}


fgentruncnormal=function(majorizing_constant){
  x=NA
  num_reject=0
  while (is.na(x)) {
    y=rmajorizing(1)
```

```
    u=runif(1)
    if (u<=target_fun(y, 2)/(majorizing_constant*majorizing_fun(y, 2, 8.6))) {
      x=y
    }
    else {
      num_reject=num_reject+1
    }
  }
  c(x,num_reject)
}


vtruncnormal_acceptreject = sapply(rep(x_max_target,Nsample),fgentruncnormal)[1,]
vtruncnormal_direct = rnorm(2*Nsample)
vtruncnormal_direct = vtruncnormal_direct[vtruncnormal_direct>=0]

hist(vtruncnormal_acceptreject, col="purple", breaks=100, xlab="", ylab="sample density", freq=FALSE, ma
```



Question 2: Laplace Distribution

The double exponential distribution is given by :

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

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 $\mathrm{x} \geqslant \mu$ where we then have:

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

and the case when $\mathrm{x} < \mu$ 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:

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

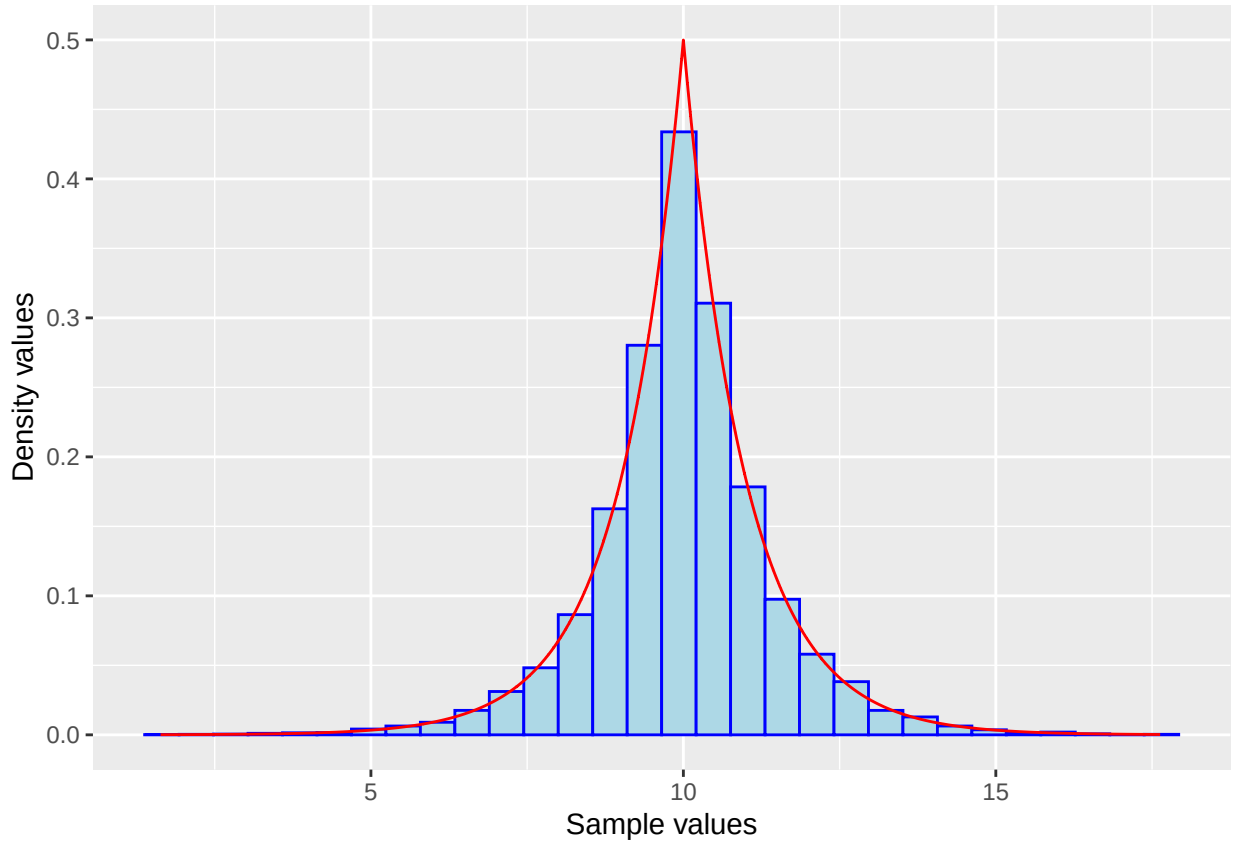
However, we would like an expression that does not depend on the sign of $x - \mu$. We try to find a quantity related to it but with respect to y. We investigate the critical value of y when the sign of $x - \mu$ changes (i.e. we look for $sgn(x - \mu)\frac{1}{\alpha} ln(1 + sgn(x - \mu) - sgn(x - \mu)2y)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



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:

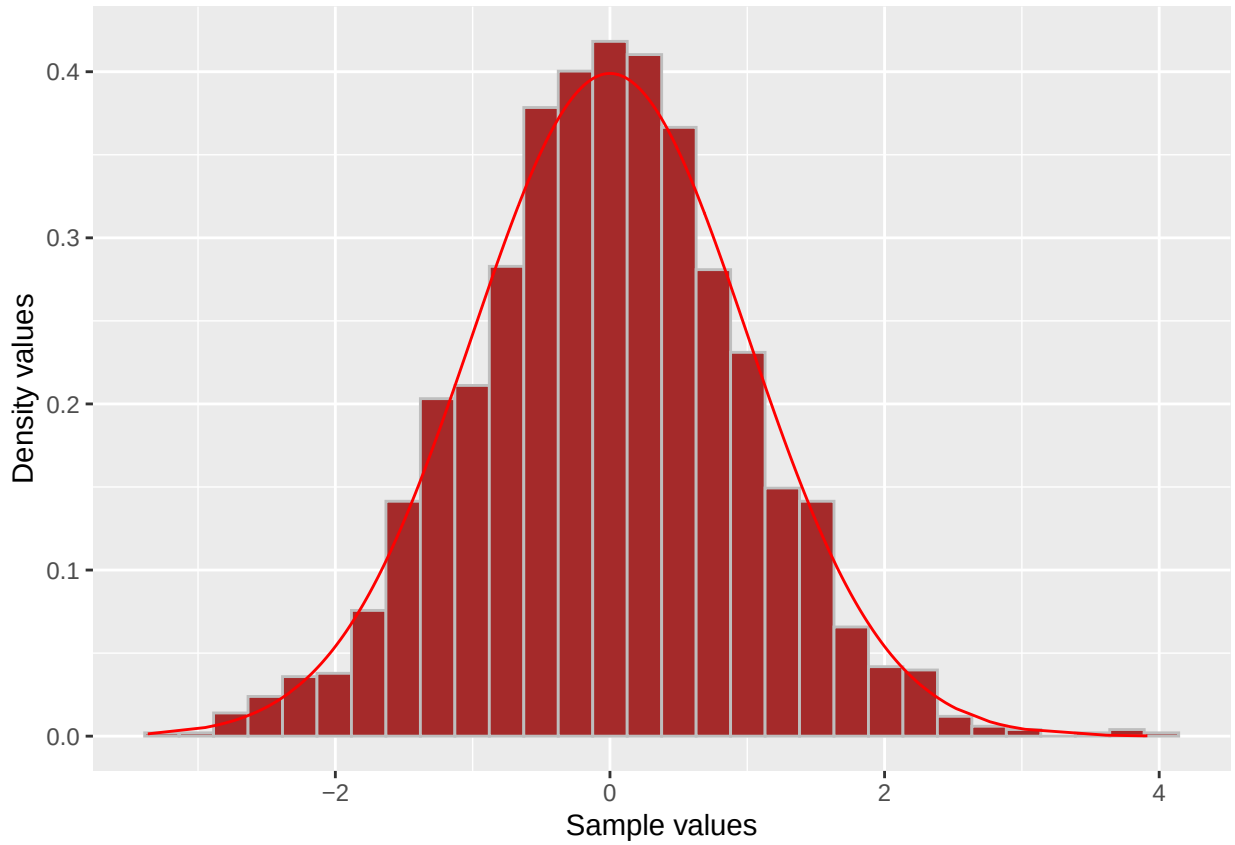
$$c \geq \sqrt{\frac{2}{\pi}} \exp\left(|x| - x^2/2\right)$$

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:

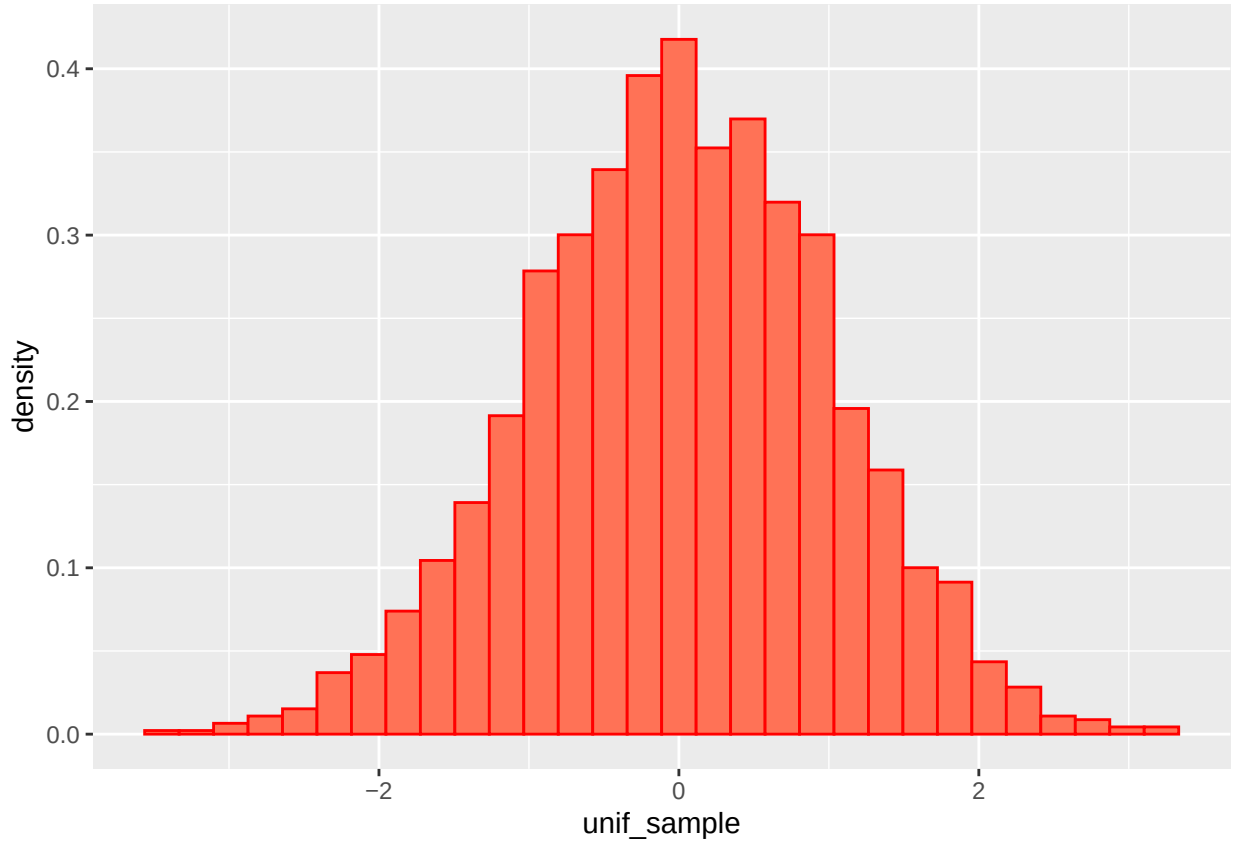
$$c_{major} = \sqrt{\frac{2e}{\pi}}$$

```
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
        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$rn, y = ..density..), col = "grey", fill="brown") +
geom_line(aes(x = sample$rn, y = dnorm(sample$rn)), col = "red") +
xlab("Sample values") + ylab("Density values")

p1
```




```
# Histogram using rnorm
p2 = ggplot() +
  geom_histogram(aes(x= unif_sample, y= ..density..), col = "red", fill = "coral1")

p2
```



Both historgrams have a similar shape.We now compute the rejection rate and the expected rejection rate.

```
reject_rate = 2000/sample$draws
expected_rejection = 1/c_major
difference = abs(reject_rate-expected_rejection)
relative = reject_rate/expected_rejection

cat("The rejection rate is: ",
    round(reject_rate, 4),"\nThe expected rejection rate is: ", round(expected_rejection, 4),
    "\nThe difference in rejection rate is:
    ",difference)
```

```
## The rejection rate is:  0.7637
## The expected rejection rate is:  0.7602
## The difference in rejection rate is:
##      0.003476798
```

Appendix

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

```r
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, col="black")
curve((eval(alpha) - 1 / eval(t_min))
      * (x / eval(t_min))^(-eval(alpha)),
      from=0, to=75, add=TRUE, col="brown")

legend("topright", inset=.02, title="functions",
       c("target","majorizing"),
       horiz=TRUE, cex=0.8, col = 1:2, lty = 1)
target_fun = function(x, c) {
  res = 0
  res = ((c * (sqrt(2 * pi)^(-1)) * exp((-c^(2)) / (2 * x)) * x^(-3/2)))
  res[x<0] = 0
  return (res)
}

power_law = function(x, alpha, t_min) {
  return(((alpha - 1) / t_min) * (x / t_min)^(-alpha))
}


majorizing_fun = function(x, alpha, t_min) {
  sapply(x, function(y) {
    res = NA
    if (y<0) {
      res = 0
    }
    if ((y>=0) && (y<=t_min)) {
      res = x_max_target
    }
    if (y>t_min) {
      res = power_law(y, alpha, t_min)
    }
    res
  }, simplify = TRUE)
}

alpha = 2
c = 2
t_min = 8.6

# Find out maximum value max(f(x)) for given parameters
x_max_target = c * (sqrt(2 * pi)^(-1)) * exp(-c^(2) / (2 * (c^2/3))) * (c^2/3)^(-3/2)

# Plot both the target function and the combined majorizing function
vx = c(seq(0, t_min, t_min/10000), seq(t_min, 30, 30/10000))
plot(vx, (target_fun(vx, c)), pch=19, cex=0.4, xlab="x", ylab="density", main="Truncated normal and maj
points(vx, (majorizing_fun(vx, alpha, t_min)), pch=19, cex=0.2, col="red")

# Implementation acceptance-rejection algorithm
```

```
rmajorizing=function(n) {
  sapply(1:n,function(i) {
    res=NA
    component=sample(1:2,1,prob=c(2/3,1/3))
    if(component==1){res=x_max_target*runif(1)}
    if(component==2){res=rplcon(1, t_min, alpha)}
    return(res)
  })
}
Nsample=100000
my_vec = c()
for (i in 1:Nsample) {
  res=NA
  component=sample(1:2,1,prob=c(2/3,1/3))
  if(component==1){
    res=runif(1)
  }
  if(component==2){
    res=rplcon(1, t_min, alpha)
  }
  my_vec = c(my_vec, res)
}


fgentruncnormal=function(majorizing_constant){
  x=NA
  num_reject=0
  while (is.na(x)) {
    y=rmajorizing(1)
    u=runif(1)
    if (u<=target_fun(y, 2)/(majorizing_constant*majorizing_fun(y, 2, 8.6))) {
      x=y
    }
    else {
      num_reject=num_reject+1
    }
  }
  c(x,num_reject)
}


vtruncnormal_acceptreject = sapply(rep(x_max_target,Nsample),fgentruncnormal)[1,]
vtruncnormal_direct = rnorm(2*Nsample)
vtruncnormal_direct = vtruncnormal_direct[vtruncnormal_direct>=0]

hist(vtruncnormal_acceptreject, col="purple", breaks=100, xlab="", ylab="sample density", freq=FALSE, ma
# 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)
# Plotting the histogram
hist_plot = ggplot() +
geom_histogram(aes(x = sample, y = ..density..), col = "blue", fill="lightblue") +
geom_line(aes(x = sample, y = sample_density), col = "red") +
xlab("Sample values") + ylab("Density values")

hist_plot
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
        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$rn, y = ..density..), col = "grey", fill="brown") +
geom_line(aes(x = sample$rn, y = dnorm(sample$rn)), col = "red") +
xlab("Sample values") + ylab("Density values")

p1
```

```r
# Histogram using rnorm
p2 = ggplot() +
  geom_histogram(aes(x= unif_sample, y= ..density..), col = "red", fill = "coral1")

p2
reject_rate = 2000/sample$draws
expected_rejection = 1/c_major
difference = abs(reject_rate-expected_rejection)
relative = reject_rate/expected_rejection

cat("The rejection rate is: ",
    round(reject_rate, 4),"\nThe expected rejection rate is: ", round(expected_rejection, 4),
    "\nThe difference in rejection rate is:
    ",difference)
```