# STAT 206 Lab 6

**Due Monday, November 13, 5:00 PM**

***General instructions for labs***: You are encouraged to work in pairs to complete the lab. Labs must be completed as an R Markdown file. Be sure to include your lab partner (if you have one) and your own name in the file. Give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used.

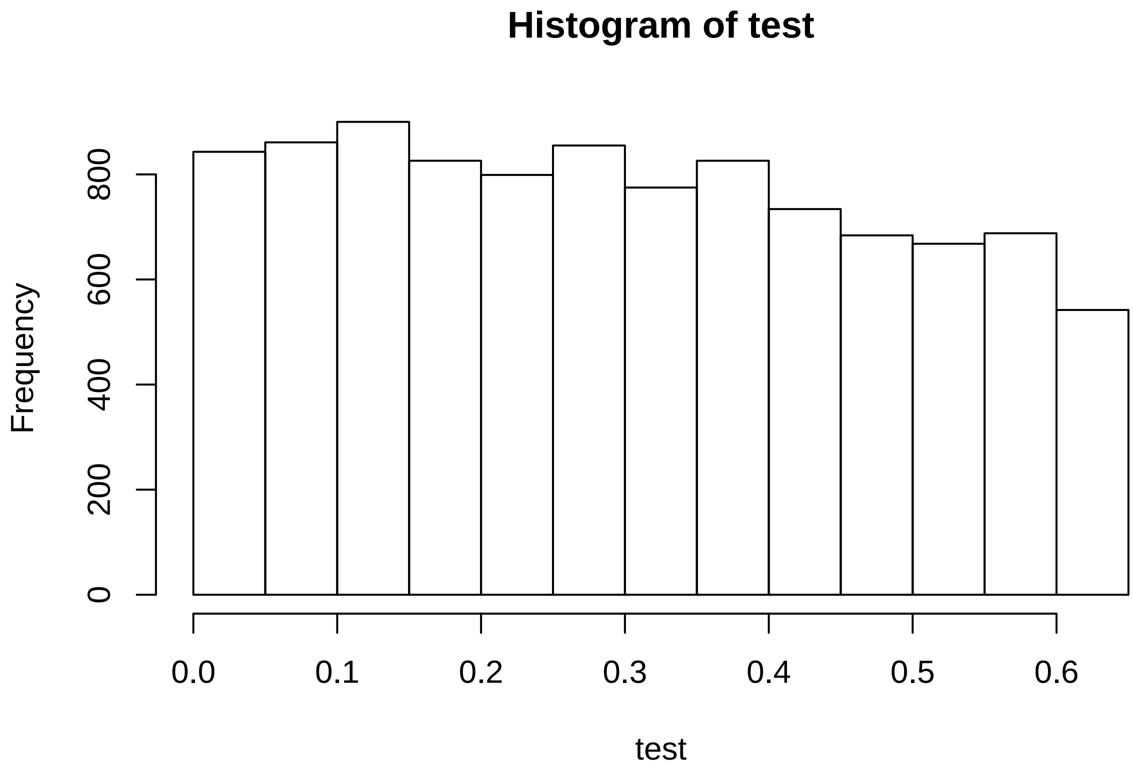
***Agenda***: Accept-Reject algorithm

1. Write a function to simulate $n$ $N(0, 1)$ random variates using the Accept-Reject algorithm with a Cauchy candidate.

```
#First to solve for M we have the normal desnity function in the numerator divided by the Cauchy
#density function in the denominator
# 2pi^-12*exp((-x^2)/2) / pi(1 + x^2)
#This is maximized when the denominator is small as possible
#this means we solve for when x = 0

M = 1.55760

runs = 100
ar_lab = function(n=runs) {
  i = 1
  count = 0
  x = double(n)
  const_u = 1 / (sqrt(2*pi))
  while ( i <= n+1) {
    count = count + 1
    y = rcauchy(1)
    u = runif(1)
    temp = (1/M) * (u/y)
    if(u < temp) {
      x[i] = y
      i = i + 1
    }
  }
  return(x)
}

test = ar_lab(10000)
mean(test)
```

```
## [1] 0.3036664
```

```
hist(test)
```

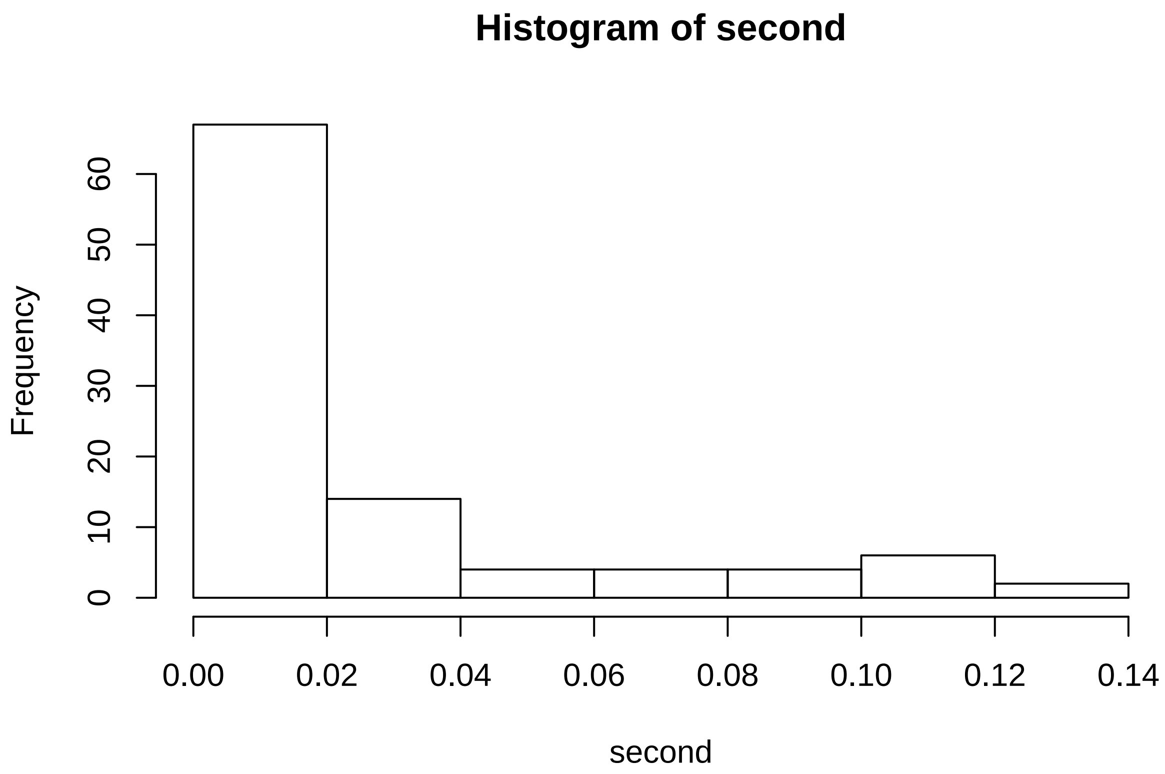


2. Simulate 1000 $N(0,1)$ random variates using your function to estimate $E[Y^3 \log(1+|Y|)]$ and $Pr(Y \in [-1,2])$. Be sure to include a Monte Carlo standard error with your estimates.

```
#We use our last function to estimate the function above for 1000 runs. We also get a sum of how many r

runs = 100
test = ar_lab(runs)

temp = function(x) {
  (x^3) * (log( 1 + abs(x)))
}


second = temp(test)
second
```

```
##   [1] 1.504395e-07 2.983789e-02 8.539870e-02 1.089744e-01 6.285934e-02
##   [6] 5.021715e-04 1.483786e-03 3.020759e-02 2.775746e-08 8.321524e-03
##  [11] 7.874932e-03 3.267079e-05 7.589438e-04 8.610289e-04 6.777909e-02
##  [16] 4.763871e-03 2.135359e-03 2.699294e-05 1.613648e-03 4.649147e-02
##  [21] 3.084268e-03 1.773376e-05 3.088151e-02 6.344987e-05 1.907336e-02
##  [26] 1.368865e-08 8.458065e-03 1.598383e-04 1.175248e-01 8.539395e-02
##  [31] 5.829482e-03 1.475251e-05 3.492883e-03 1.222910e-05 9.505884e-03
##  [36] 6.019548e-02 3.852238e-02 9.621017e-03 3.154843e-02 1.439529e-02
##  [41] 1.338333e-03 8.644980e-02 2.990591e-06 2.388547e-02 2.494753e-02
##  [46] 1.475791e-04 3.680396e-02 1.226408e-02 1.992042e-02 2.402050e-04
##  [51] 9.584995e-03 4.189873e-02 1.703124e-02 1.045675e-03 2.848633e-02
##  [56] 1.179876e-01 2.861292e-06 2.729176e-02 6.453431e-03 6.609346e-02
##  [61] 3.660320e-02 2.817656e-07 2.649217e-02 1.227761e-02 3.554498e-03
##  [66] 7.335528e-05 1.172247e-01 1.277339e-03 7.042127e-03 1.125368e-02
##  [71] 1.479667e-05 4.833003e-02 2.912632e-03 8.286652e-02 3.077660e-02
##  [76] 1.867705e-02 4.373365e-05 1.059816e-06 2.854682e-04 4.233980e-02
##  [81] 1.520352e-03 8.983039e-03 2.610916e-03 1.431801e-02 5.860400e-03
##  [86] 1.311176e-03 1.257884e-01 4.194801e-04 1.287693e-01 8.413049e-03
##  [91] 2.187019e-03 1.014236e-01 1.163181e-01 1.077316e-06 2.312169e-04
##  [96] 2.314380e-02 3.467786e-08 6.400813e-06 7.634302e-06 7.889785e-04
## [101] 3.828268e-03
```

```
hist(second)
```



```
cond1 = sum(second <= 2 & second >=-1)
#cond1

prob = cond1 / runs
prob
```

```
## [1] 1.01
```

```
sdsecond = sd(second)
sdsecond
```

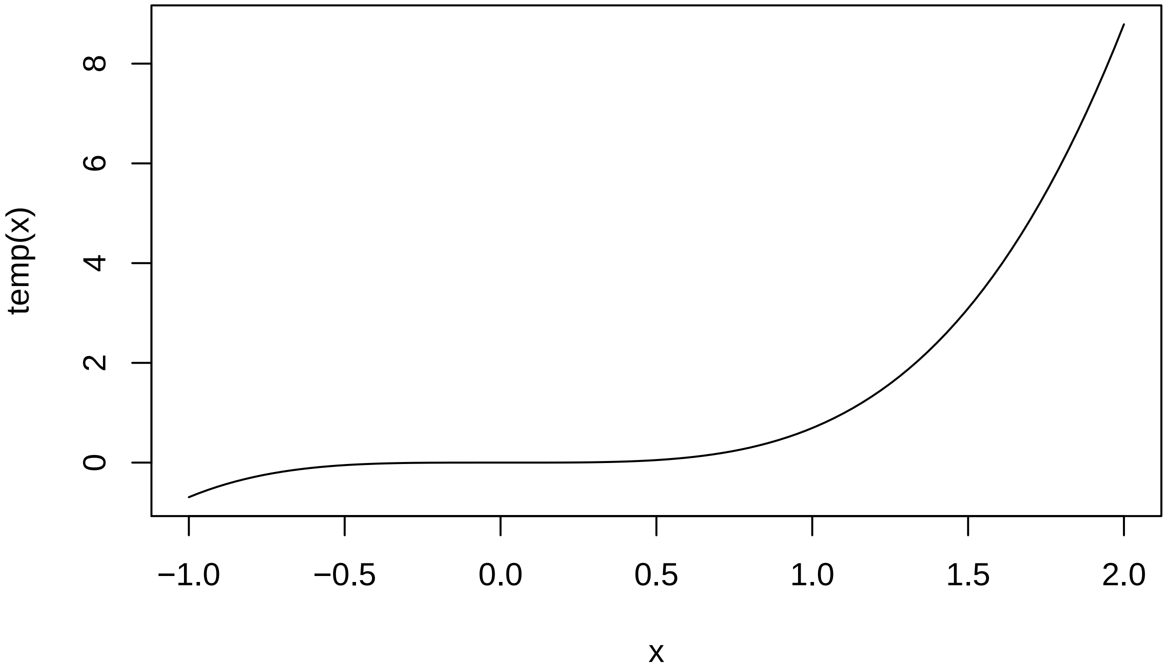
```
## [1] 0.03488459
```

```
se_second = sdsecond / sqrt(runs)
se_second
```

```
## [1] 0.003488459
```

3. What was the acceptance rate of the Accept-Reject algorithm? Is this close to the theoretical acceptance rate?

```
#The area of the window is about 16 and the area under the curve is about 4.3 so
# the theoretical acceptance should be near 25%
#However I have a bug and I seem to get 100% acceptance

curve(temp(x), xlim = c(-1,2))
```



4. Write a function that continues simulation until the sample size is large enough so that your Monte Carlo error is less than $\epsilon = 0.01$ for estimating a general statistic `stat` (which will be a function). You function should also return the observed acceptance rate of the Accept-Reject algorithm and a Monte Carlo standard error.

```
#This function takes two arguments, 1 is the function from problem 2 and the second is the stopping
#condition of epsilon. Here we use a while loop so the loop terminates when the stopping condition of
#epsilon of .01 has been met. We then have a nested for loop that looks at each value in the output
#of the function and stores the sd in a vector, we calculate the se every time we go through the loop
#when the se is below our stopping condition the loop teminates
#we can see that it took 10002 loops. However for some reason I'm getting NA for the standard error

stat = 1
stds = c()
i = 1
count = 0
stopper = function(estimator, stat) {
  while( stat >= .01) {
    for(value in estimator) {
      i = i + 1
      std = sd(value)
      stds = c(stds, std)
      final_std = sd(stds)
      se = stds / sqrt(i)
      stat = se
      done = final_std/i

    }
    return(c(i, done))
  }
}

stopper(ar_lab(10000), 1)
```

```
## [1] 10002    NA
```

5. Use your function to estimate $E[Y^3 \log(1+|Y|)]$ and $Pr(Y \in [-1,2])$ with $\epsilon = 0.01$. Report your estimates along with confidence intervals based on the Monte Carlo standard error. What was the acceptance rate?

```
#this turns out very simliar because i've introduced a bug that I don't understand in my code

stopper(temp(test), 1)
```

```
## [1] 102  NA
```