HW5_20201127_PeiShanYen

Question 01

5.3. Suppose the data $(x_1, \ldots, x_7) = (6.52, 8.32, 0.31, 2.82, 9.96, 0.14, 9.64)$ are observed. Consider Bayesian estimation of μ based on a $N(\mu, 3^2/7)$ likelihood for the minimally sufficient $\bar{x} \mid \mu$, and a Cauchy(5,2) prior.

a. Using a numerical integration method of your choice, show that the proportionality constant is roughly 7.84654. (In other words, find k such that $\int k \times (\text{prior}) \times (\text{likelihood})\, d\mu = 1$.)

Using Riemann Sum, the approximately value of the normalizing constant k is 7.84654.

```r
# Data
x = c(6.52, 8.32, 0.31, 2.82, 9.96, 0.14, 9.64)
x_bar = mean(x)

# Bayesian Model
# X_bar | mu ~ N(mu, 3^2/7)
f_likelihood = function(mu) {1/sqrt(2*pi*9/7) * exp(-(x_bar - mu)^2/(2*9/7))}

# mu ~ Cauchy(5, 2) (prior)
f_prior = function(mu) {1/(pi*2*(1 + ((mu - 5)/2)^2))}

# Find the constant k (normalizing constant)
# Perform variable transformation.
# mu = log(y/(1-y)), -inf < mu < inf  -->  0 < y < 1.
# Jacobian matrix = (1/(y*(1-y))

f_y = function(y) {f_likelihood(log(y/(1-y))) * f_prior(log(y/(1-y))) * (1/(y*(1-y)))}

# Riemann Sum
int_Riemann = function(f, a, b, n = 100000) {
  h = (b - a)/n
  x = seq(a, b, by = h)
  y = f(x)
  result = h * sum(y[1:n])
  return(result)
}

k = round(1/int_Riemann(f_y, 1e-10, 1 - 1e-10),5)
print(k)

## [1] 7.84654
```

b. Using the value 7.84654 from (a), determine the posterior probability that $2 \leq \mu \leq 8$ using the Riemann, trapezoidal, and Simpson's rules over the range of integration [implementing Simpson's rule as in (5.20) by pairing adjacent subintervals]. Compute the estimates until relative convergence within 0.0001 is achieved for the slowest method. Table the results. How close are your estimates to the correct answer of 0.99605?

All the results are close to 0.99605.

```
f_posterior = function(mu) { k * f_likelihood(mu) * f_prior(mu)}

# Riemann Sum
int_Riemann = function(f, a, b, n = 100000) {
  h = (b - a)/n
  x = seq(a, b, by = h)
  y = f(x)
  result = h * sum(y[1:n])
  return(result)
}

# Trapezoidal Rule
int_trapzoidal = function(f, a, b, n = 100000) {
  h = (b - a)/n
  x = seq(a, b, by = h)
  y = f(x)
  result = h * (y[1] + 2*sum(y[2:n]) + y[n+1]) / 2
  return(result)
}

# Simpson's Rule
int_Simpson = function(f, a, b, n = 100000) {
  h = (b - a)/n
  x = seq(a, b, by = h)
  y = f(x)
  if (n == 2) {
    result = (h/3) * (y[1] + 4*y[2] + y[3])
  } else {
    result = (h/3) * (y[1] + sum(2*y[seq(2, n, by = 2)]) + sum(4*y[seq(3, n-1, by = 2)])
+ y[n+1])
  }
  return(result)
}

# 2 <= mu <= 8
data.frame(method = c("Riemann Sum","Trapezoidal Rule","Simpson's Rule"),
           result = c(int_Riemann(f_posterior, 2, 8),int_trapzoidal(f_posterior, 2, 8), i
nt_Simpson(f_posterior, 2, 8)))

##             method    result
## 1      Riemann Sum 0.9960544
## 2 Trapezoidal Rule 0.9960547
## 3   Simpson's Rule 0.9960544
```

c. Find the posterior probability that $\mu \geq 3$ in the following two ways. Since the range of integration is infinite, use the transformation $u = \exp\{\mu\}/(1+\exp\{\mu\})$. First, ignore the singularity at 1 and find the value of the integral using one or more quadrature methods. Second, fix the singularity at 1 using one or more appropriate strategies, and find the value of the integral. Compare your results. How close are the estimates to the correct answer of 0.99086?

d. Use the transformation $u = 1/\mu$, and obtain a good estimate for the integral in part (c).

Using Riemann sum method, both results are extremely close to 0.99086.

```
# Perform variable transformation.
# mu = log(u/(1-u)), 3 <= mu < inf -->  exp(3)/(1 + exp(3)) < u < 1 .
# Jacobian matrix = (1/(u*(1-u))

fu_posterior = function(u) { k * f_likelihood(log(u/(1-u))) * f_prior(log(u/(1-u))) * (1/
(u*(1-u)))}
int_Riemann(fu_posterior, exp(3)/(1 + exp(3)), 1 - 1e-10)

## [1] 0.9908596

# Perform variable transformation.
# mu = 1/u, , 3 <= mu < inf  -->  0 < u < 1/3.
# Jacobian matrix = 1/u^2
fu_posterior = function(u) { k * f_likelihood(1/u) * f_prior(1/u) * (1/u^2)}
int_Riemann(fu_posterior, 1e-10, 1/3)

## [1] 0.9908591
```

Question 02

5.4. Let $X \sim \text{Unif}[1, a]$ and $Y = (a-1)/X$, for $a > 1$. Compute $E\{Y\} = \log a$ using Romberg's algorithm for $m = 6$. Table the resulting triangular array. Comment on your results.

When a is a small value, the result obtained from Romberg's algorithm for m = 6 is close to the theoretical solution. When a = e, the Romberg solution for m = 6 is 1. However, as the value of a increases, such as 100, the dimension of the triangular array needs to enlarge to obtain an acceptable result. (The theoretical solution is 4.60517. The Romberg solution for m = 6, 8, 10, 12, are 4.768311, 4.608066, 4.605174, 4.60517, respectively)

```
# Reference: https://en.wikipedia.org/wiki/Romberg%27s_method
# Romberg Integration
int_Romberg = function(f, a, b, m) {
  R = matrix(NA, m, m)
  h = b - a
  R[1,1] = (f(a) + f(b)) * h/2
  for (i in 2:m) { R[i,1] = 1/2 * (R[i-1,1] + h * sum(f(a + (1:2^(i-2) - 0.5) * h)))
    for (j in 2:i) { R[i,j] = R[i,j-1] + (R[i,j-1] - R[i-1,j-1]) / (4^(j-1) - 1) }
    h = h/2 }

  result = R[m,m]
  return(list(R, result))
}

EY = function(a, m){

  # Romberg's Integration
  int_Romberg(function(x) 1/x, 1, a, m)

  # Simulated Solution (MC integration)
   set.seed(20200824)
   x = runif(n=100000, min = 1, max = a)
   y = (a -1 )/x
   mean(y)

  # Theoretical Solution
  log(a)

   return(list(Romberg_Triangular_Array = int_Romberg(function(x) 1/x, 1, a, m )[[1]],
               Romberg_Solution = int_Romberg(function(x) 1/x, 1, a, m)[[2]],
               MC_integration_Solution= mean(y),
               Theoretical_Solution = log(a)))
}

EY(a=exp(1),m=6)

## $Romberg_Triangular_Array
##          [,1]     [,2]     [,3]     [,4] [,5] [,6]
## [1,] 1.175201       NA       NA       NA   NA   NA
## [2,] 1.049718 1.007890       NA       NA   NA   NA
## [3,] 1.013039 1.000813 1.000341       NA   NA   NA
## [4,] 1.003307 1.000063 1.000013 1.000008   NA   NA
## [5,] 1.000830 1.000004 1.000000 1.000000    1   NA
```

```
## [6,] 1.000208 1.000000 1.000000 1.000000     1     1
##
## $Romberg_Solution
## [1] 1
##
## $MC_integration_Solution
## [1] 0.999145
##
## $Theoretical_Solution
## [1] 1
```

```
EY(a=exp(2),m=6)
```

```
## $Romberg_Triangular_Array
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 3.626860       NA       NA       NA       NA       NA
## [2,] 2.575024 2.224412       NA       NA       NA       NA
## [3,] 2.178272 2.046022 2.034129       NA       NA       NA
## [4,] 2.049460 2.006522 2.003889 2.003409       NA       NA
## [5,] 2.012847 2.000642 2.000250 2.000192 2.000180       NA
## [6,] 2.003248 2.000049 2.000009 2.000005 2.000004 2.000004
##
## $Romberg_Solution
## [1] 2.000004
##
## $MC_integration_Solution
## [1] 1.997491
##
## $Theoretical_Solution
## [1] 2
```

```
EY(a=2,m=6)
```

```
## $Romberg_Triangular_Array
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.7500000        NA        NA        NA        NA        NA
## [2,] 0.7083333 0.6944444        NA        NA        NA        NA
## [3,] 0.6970238 0.6932540 0.6931746        NA        NA        NA
## [4,] 0.6941219 0.6931545 0.6931479 0.6931475        NA        NA
## [5,] 0.6933912 0.6931477 0.6931472 0.6931472 0.6931472        NA
## [6,] 0.6932082 0.6931472 0.6931472 0.6931472 0.6931472 0.6931472
##
## $Romberg_Solution
## [1] 0.6931472
##
## $MC_integration_Solution
## [1] 0.6927141
##
## $Theoretical_Solution
## [1] 0.6931472
```

```
EY(a=100,m=6)
```

```
## $Romberg_Triangular_Array
##           [,1]      [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 49.995000        NA       NA       NA       NA       NA
## [2,] 25.977698 17.971931       NA       NA       NA       NA
```

```
## [3,] 14.278918 10.379324 9.873151       NA       NA       NA
## [4,]  8.727329  6.876799 6.643297 6.592030       NA       NA
## [5,]  6.214877  5.377393 5.277433 5.255753 5.250512       NA
## [6,]  5.165740  4.816027 4.778603 4.770685 4.768782 4.768311
##
## $Romberg_Solution
## [1] 4.768311
##
## $MC_integration_Solution
## [1] 4.606614
##
## $Theoretical_Solution
## [1] 4.60517
```

```
EY(a=100,m=12)
```

```
## $Romberg_Triangular_Array
##            [,1]      [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 49.995000        NA       NA       NA       NA       NA       NA       NA
##  [2,] 25.977698 17.971931       NA       NA       NA       NA       NA       NA
##  [3,] 14.278918 10.379324 9.873151       NA       NA       NA       NA       NA
##  [4,]  8.727329  6.876799 6.643297 6.592030       NA       NA       NA       NA
##  [5,]  6.214877  5.377393 5.277433 5.255753 5.250512       NA       NA       NA
##  [6,]  5.165740  4.816027 4.778603 4.770685 4.768782 4.768311       NA       NA
##  [7,]  4.777158  4.647631 4.636405 4.634148 4.633612 4.633480 4.633447       NA
##  [8,]  4.652598  4.611078 4.608641 4.608200 4.608098 4.608073 4.608067 4.608066
##  [9,]  4.617457  4.605743 4.605387 4.605336 4.605325 4.605322 4.605321 4.605321
## [10,]  4.608274  4.605213 4.605178 4.605175 4.605174 4.605174 4.605174 4.605174
## [11,]  4.605948  4.605173 4.605170 4.605170 4.605170 4.605170 4.605170 4.605170
## [12,]  4.605365  4.605170 4.605170 4.605170 4.605170 4.605170 4.605170 4.605170
##           [,9]    [,10]   [,11]   [,12]
##  [1,]       NA       NA      NA      NA
##  [2,]       NA       NA      NA      NA
##  [3,]       NA       NA      NA      NA
##  [4,]       NA       NA      NA      NA
##  [5,]       NA       NA      NA      NA
##  [6,]       NA       NA      NA      NA
##  [7,]       NA       NA      NA      NA
##  [8,]       NA       NA      NA      NA
##  [9,] 4.605321       NA      NA      NA
## [10,] 4.605174 4.605174      NA      NA
## [11,] 4.605170 4.605170 4.60517      NA
## [12,] 4.605170 4.605170 4.60517 4.60517
##
## $Romberg_Solution
## [1] 4.60517
##
## $MC_integration_Solution
## [1] 4.606614
##
## $Theoretical_Solution
## [1] 4.60517
```

Question 03

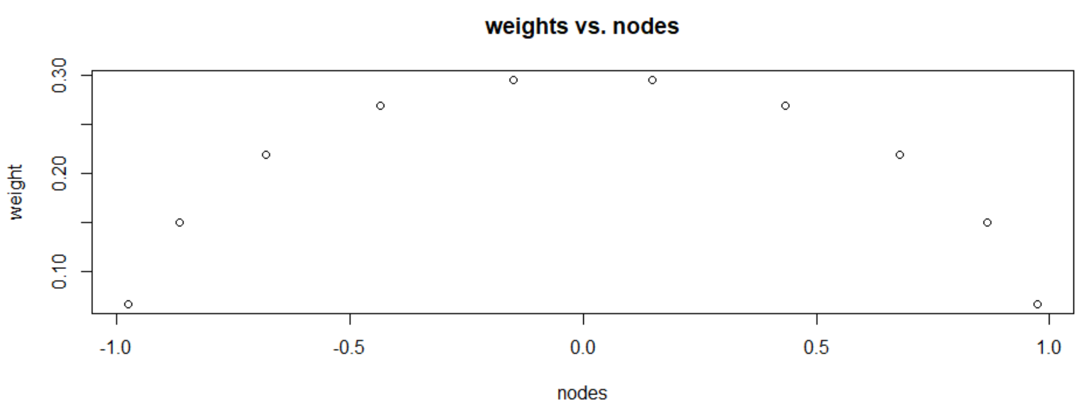
5.5. The Gaussian quadrature rule having $w(x) = 1$ for integrals on $[-1, 1]$ (cf. Table 5.6) is called *Gauss–Legendre quadrature* because it relies on the Legendre polynomials. The nodes and weights for the 10-point Gauss–Legendre rule are given in Table 5.8.

a. Plot the weights versus the nodes.

b. Find the area under the curve $y = x^2$ between -1 and 1. Compare this with the exact answer and comment on the precision of this quadrature technique.

The area obtained by the 10-point Gaussian quadrature method is identical to the theoretical solution.

```
# 10-point Gaussian quadrature rule
nodes = c(-0.148874338981631, 0.148874338981631, -0.433395394129247, 0.433395394129247,
          -0.679409568299024, 0.679409568299024, -0.865063366688985, 0.865063366688985,
          -0.973906528517172, 0.973906528517172)
weights = c(0.295524224714753, 0.295524224714753, 0.269266719309996, 0.269266719309996,
            0.219086362515982, 0.219086362515982, 0.149451394150581, 0.149451394150581,
            0.066671344308688, 0.066671344308688)

# Plot the weights versus the nodes.
plot(x = nodes, y = weights, type = 'p',
     main = 'weights vs. nodes', xlab = 'nodes', ylab = 'weight')
```




```
Area = function(nodes, weights, fun, lower, upper){
  return(list( Theoretical_Solution = integrate(fun, lower , upper),
               number_of_Gaussian_quadrature = length(nodes),
               Area_Gaussian_quadrature = sum(weights * f(nodes)) )) }
f = function(x) {x^2}

Area(nodes= nodes, weights= weights, fun = f, lower=-1, upper=1)

## $Theoretical_Solution
## 0.6666667 with absolute error < 7.4e-15
##
## $number_of_Gaussian_quadrature
## [1] 10
##
## $Area_Gaussian_quadrature
## [1] 0.6666667
```

Question 04

5.6. Suppose 10 i.i.d. observations result in $\bar{x} = 47$. Let the likelihood for μ correspond to the model $\bar{X} \mid \mu \sim N(\mu, 50/10)$, and the prior for $(\mu - 50)/8$ be Student's t with 1 degree of freedom.

a. Show that the five-point Gauss–Hermite quadrature rule relies on the Hermite polynomial $H_5(x) = c(x^5 - 10x^3 + 15x)$.

<u>Hermite</u> $\alpha_K = 0, \quad B_K = 1, \quad \gamma_K = K-1, \quad W(x) = e^{-\frac{x^2}{2}}$

orthogonal polynomial:

$$P_K(x) = (\alpha_K + x B_K) P_{K-1}(x) - \gamma_K P_{K-2}(x)$$

Assume $P_0(x) = 1$

$\cancel{P_{-1}(x) = 0 = (0 + x)}$

$$P_1(x) = (0 + x) P_0(x) - (0) P_{-1}(x) = x P_0(x) = x$$

$$P_2(x) = (0 + x) P_1(x) - P_0(x) = x^2 - 1$$

$$P_3(x) = (0 + x) P_2(x) - 2P_1(x) = x(x^2 - 1) - 2x = x^3 - 3x$$

$$P_4(x) = (0 + x) P_3(x) - 3P_2(x) = x(x^3 - 3x) - 3(x^2 - 1)$$
$$= x^4 - 6x^2 + 3$$

$$P_5(x) = (0 + x) P_4(x) - 4P_3(x) = x(x^4 - 6x^2 + 3) - 4(x^3 - 3x)$$
$$= x^5 - 10x^3 + 15x$$

$$P_6(x) = (0 + x) P_5(x) - 5P_4(x) = x(x^5 - 10x^3 + 15x) - 5(x^4 - 6x^2 + 3)$$
$$= x^6 - 15x^4 + 45x^2 - 15$$

$$\Rightarrow H_5(x) = c(x^5 - 10x^3 + 15x)$$

b. Show that the normalization of $H_5(x)$ [namely, $\langle H_5(x), H_5(x)\rangle = 1$] requires $c = 1/\sqrt{120\sqrt{2\pi}}$. You may wish to recall that a standard normal distribution has odd moments equal to zero and rth moments equal to $r!/[(r/2)!2^{r/2}]$ when r is even.

$\boxed{2}$ To show normalization of $H_5(x)$

$\cancel{\int_{-\infty}^{\infty} W(x) = e}$ $\quad W(x) = e^{-\frac{x^2}{2}}$

If $\int_a^b f(x)^2 w(x) < \infty$, f is square-integrable with respect to w on $[a,b]$

$\langle f,g\rangle_{w,[a,b]} = \int_a^b f(x)g(x)w(x)\,dx$

If f and g are scaled, $\langle f,f\rangle_{w,[a,b]} = \langle g,g\rangle_{w,[a,b]} = 1$

$\Rightarrow$ f and g are orthonormal with respect to w on $[a,b]$.

To find c in $H_5(x)$

$$\int_{-\infty}^{\infty} [H_5(x)][H_5(x)]\,W(x) = 1$$

$$= \int_{-\infty}^{\infty} c^2(x^5-10x^3+15x)^2 \cdot e^{-\frac{x^2}{2}}\,dx = 1$$

$$= \int_{-\infty}^{\infty} c^2(x^{10}-20x^8+130x^6-300x^4+225x^2)\cdot e^{-\frac{x^2}{2}}\,dx = 1$$

Define $Y \sim \text{Gamma}(\alpha,\beta)$ $\quad f(y) = \int_0^\infty \frac{\beta^\alpha y^{\alpha-1}e^{-\beta y}}{\Gamma(\alpha)}\,dy \Rightarrow \int f(y)\,dy = 1$

Let $\beta = \frac{1}{2}$, $y = x^2$

$$\frac{\Gamma(\alpha)}{\frac{1}{2}^{\alpha}} = \int_0^\infty y^{\alpha-1}e^{-\frac{1}{2}y}\,dy = \int_{-\infty}^{\infty} x^{2\alpha-1}e^{-\frac{1}{2}x^2}\,dx$$

Let $2\alpha-1 = 2n$

$$\Rightarrow \int_{-\infty}^{\infty} x^{2n}e^{-\frac{1}{2}x^2}\,dx = \frac{\Gamma(n+\frac{1}{2})}{\frac{1}{2}^{n+\frac{1}{2}}}$$

$$n=1,\ \int_{-\infty}^{\infty} x^2 e^{-\frac{1}{2}x^2}\,dx = \frac{\Gamma(1+\frac{1}{2})}{\frac{1}{2}^{1+\frac{1}{2}}} = \frac{\frac{1}{2}\sqrt{\pi}}{\frac{1}{2}^{\frac{3}{2}}}$$

$\Gamma(z+1) = z\Gamma(z)$, $\Gamma(\frac{1}{2}) = \sqrt{\pi}$

$$n=2,\ \int_{-\infty}^{\infty} x^4 e^{-\frac{1}{2}x^2}\,dx = \frac{\Gamma(2+\frac{1}{2})}{\frac{1}{2}^{2+\frac{1}{2}}} = \frac{\frac{3}{2}\cdot\frac{1}{2}\sqrt{\pi}}{\frac{1}{2}^{\frac{5}{2}}}$$

$\vdots$

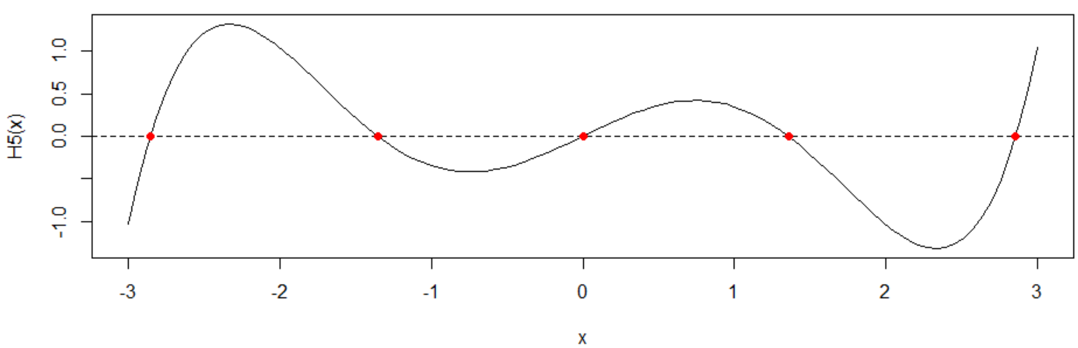
$$n=5,\ \int_{-\infty}^{\infty} x^{10} e^{-\frac{1}{2}x^2}\,dx = \frac{\Gamma(5+\frac{1}{2})}{\frac{1}{2}^{5+\frac{1}{2}}} = \frac{\frac{9}{2}\frac{7}{2}\frac{5}{2}\cdot\frac{3}{2}\cdot\frac{1}{2}\sqrt{\pi}}{\frac{1}{2}^{\frac{11}{2}}}$$

Hence

$$c = \frac{1}{\sqrt{\int(x^{10}-20x^8+130x^6-300x^4+225x^2)\,e^{-\frac{x^2}{2}}\,dx}} = \frac{1}{\sqrt{120\sqrt{2\pi}}}.$$

c. Using your favorite root finder, estimate the nodes of the five-point Gauss–Hermite quadrature rule. (Recall that finding a root of f is equivalent to finding a local minimum of $|f|$.) Plot $H_5(x)$ from -3 to 3 and indicate the roots.

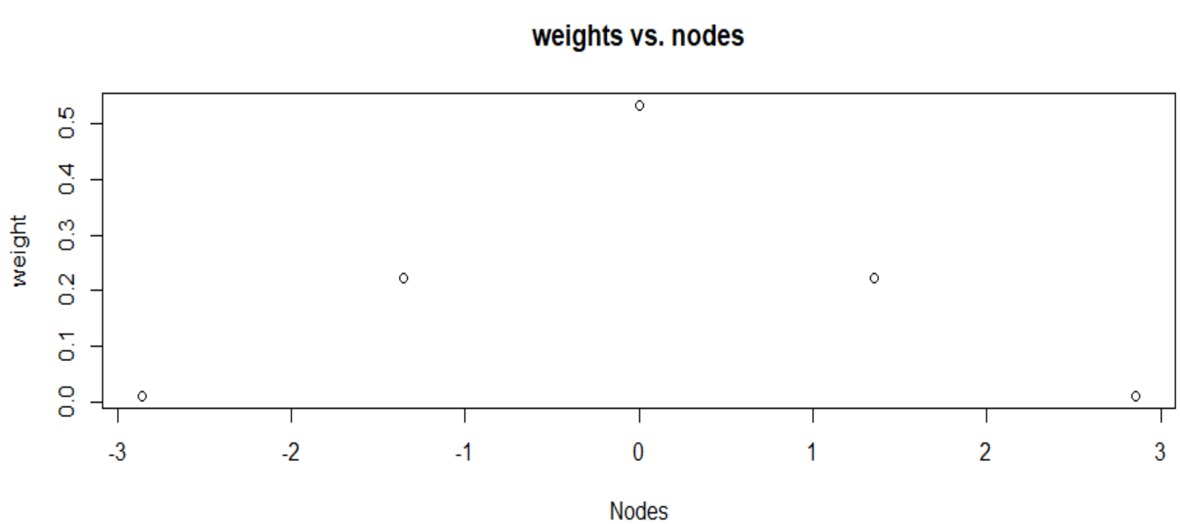
```
# Find the root of H5(x)
# Bisection method
bisection = function(f, a, b, n = 1000, tol = 1e-7) {
  if (f(a) * f(b) > 0) {
    stop('signs of f(a) and f(b) differ')}
  for (i in 1:n) { c = (a + b) / 2
    if (f(c) == 0 | ((b - a) / 2) < tol) { return(c)}
    if (sign(f(c)) == sign(f(a))) { a = c } else { b = c }}
  print('Too many iterations')}

# Plot H5(x) from -3 to 3.
x = seq(-3, 3, length = 100)
H5 = function(x) {(1/sqrt(120*sqrt(2*pi))) * (x^5 - 10*x^3 + 15*x)}

plot(x = x, y = H5(x), type = 'l')
abline(h = 0, lty = 2)
points(x = bisection(f=H5, a=-3, b=-2), y = 0, pch = 16, col = 'red')
points(x = bisection(f=H5, a=-2, b=-1), y = 0, pch = 16, col = 'red')
points(x = bisection(f=H5, a=-1, b=1),  y = 0, pch = 16, col = 'red')
points(x = bisection(f=H5, a=1,  b=2),  y = 0, pch = 16, col = 'red')
points(x = bisection(f=H5, a=2,  b=3),  y = 0, pch = 16, col = 'red')
```




```
data.frame(point=c(1,2,3,4,5),
           root=c(bisection(f=H5, a=-3, b=-2),
                  bisection(f=H5, a=-2, b=-1),
                  bisection(f=H5, a=-1, b=1),
                  bisection(f=H5, a=1,  b=2),
                  bisection(f=H5, a=2,  b=3)))

##   point      root
## 1     1 -2.856970
## 2     2 -1.355626
## 3     3  0.000000
## 4     4  1.355626
## 5     5  2.856970
```

d. Find the quadrature weights. Plot the weights versus the nodes. You may appreciate knowing that the normalizing constant for $H_6(x)$ is $1/\sqrt{720\sqrt{2\pi}}$.

```
nodes = c(bisection(H5, -3, -2), bisection(H5, -2, -1), bisection(H5, -1, 1), bisection(H
5, 1, 2), bisection(H5, 2, 3))

fun_qw = function(x) {
  c5 = 1/sqrt(120*sqrt(2*pi))
  c6 = 1/sqrt(720*sqrt(2*pi))
  qw = - c6 / (c5 * c6 * (x^6 - 15*x^4 + 45*x^2 - 15) * c5 * (5*x^4 - 30*x^2 + 15)) # for
mula from section 5.3.2 in textbook
  return(qw) }
data.frame(points=c(1,2,3,4,5),nodes=nodes, weights=fun_qw(nodes)/sum(fun_qw(nodes)))

##   points     nodes    weights
## 1      1 -2.856970 0.01125741
## 2      2 -1.355626 0.22207592
## 3      3  0.000000 0.53333333
## 4      4  1.355626 0.22207592
## 5      5  2.856970 0.01125741
```



e. Using the nodes and weights found above for five-point Gauss–Hermite integration, estimate the posterior variance of μ. (Remember to account for the normalizing constant in the posterior before taking posterior expectations.)

The posterior variance of u is 4.53585.

$\bar{x} = 47.$

* $\bar{x} | u \sim N(u, \frac{50}{10}) \rightarrow f(\bar{x}|u) = \frac{1}{\sqrt{2\pi \cdot 5}} \cdot e^{-\frac{(\bar{x}-u)^2}{2 \cdot 5}} = \frac{1}{\sqrt{10\pi}} e^{-\frac{(\bar{x}-u)^2}{10}}$

* prior. $y = \frac{u-50}{8} \sim t_{(1=df)} = \text{Cauchy}(0, 1)$

$$f(y) = \frac{\Gamma(\frac{1+1}{2})}{\sqrt{\pi} \cdot \Gamma(\frac{1}{2})} (1+y^2)^{-\frac{1+1}{2}} = \frac{1}{\pi(1+y^2)}$$

$\rightarrow$ variable transformation)

$y = \frac{u-50}{8} \Rightarrow u = 8y + 50 \rightarrow du = 8dy$

$$f(u) = \frac{1}{\pi[1+(\frac{u-50}{8})^2]} \cdot \frac{1}{8} = \text{Cauchy}(50, 8)$$

$\bar{x}|\mu \sim N(\mu, 5)$ $\frac{\sigma^2}{10}=5$ $n=10,\ \bar{x}=47$

$x|\mu \sim N(\mu, 50) \Rightarrow \prod_{i=1}^{n} f(x_i|\mu)$ $-\infty<\mu<\infty$

$$C=\int_{-\infty}^{\infty} \frac{1}{\sqrt{10\pi}} e^{-\frac{(\bar{x}-\mu)^2}{10}} \cdot \frac{1}{8}\frac{1}{\pi} \frac{1}{1+\left(\frac{\mu-50}{8}\right)^2}\, d\mu$$

$$=\int_{-\infty}^{\infty} \frac{1}{\sqrt{10\pi}} e^{-\frac{(47-\mu)^2}{10}} \frac{1}{8\pi} \cdot \frac{1}{1+\left(\frac{\mu-50}{8}\right)^2}\, d\mu$$

define $y=\frac{\mu-47}{\sqrt{5}}$ $dy=\frac{1}{\sqrt{5}}\, d\mu$

$\frac{(\mu-47)^2}{10}=\frac{y^2}{2\cdot\sigma^2}=\frac{y^2}{2\cdot 5}$

$$=\int_{-\infty}^{\infty} \frac{\sqrt{5}}{\sqrt{10\pi}} \cdot \frac{1}{8\pi} e^{-\frac{y^2}{2}} \cdot \frac{1}{1+\left(\frac{\sqrt{5}y-3}{8}\right)^2}\, dy$$

$\mu=\sqrt{5}y+47$

$\frac{\mu-50}{8}=\frac{\sqrt{5}y-3}{8}$

$$=\int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \cdot \frac{1}{\sqrt{128\pi^3}} \cdot \frac{1}{1+\left(\frac{\sqrt{5}y-3}{8}\right)^2}\, dy$$

$$=\frac{1}{\sqrt{128\pi^3}}\left[\sum_{g=1}^{5} \frac{1}{1+\left(\frac{\sqrt{5}\,\text{node}-3}{8}\right)^2} \cdot \text{Weight}(\text{node})\right]$$

$\Rightarrow$ From Hermite orthonormalizing polynomial

$\downarrow$

5-point quadrature $g=5$

$=(\text{normalizing constant})^{-1}$

Find $Var(\mu|\bar{x})$, $= E(\mu^2|\bar{x}) - [E(\mu|\bar{x})]^2$

$E(\mu|\bar{x}) = \frac{1}{c}\int_{-\infty}^{\infty} \mu \cdot f(\bar{x}|\mu) f(\mu) d\mu$

$$= \frac{1}{c}\int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \cdot \frac{1}{\sqrt{128\pi^3}} \cdot \frac{\sqrt{5}y+47}{1+\left(\frac{\sqrt{5}y-3}{8}\right)^2} dy$$

$$= \frac{1}{c} \cdot \frac{1}{\sqrt{128\pi^3}} \left[\sum_{q=1}^{5} \frac{\sqrt{5}\,\text{node}+47}{1+\left(\frac{\sqrt{5}\,\text{node}-3}{8}\right)^2} \cdot \text{weight}(\text{node}) \right]$$

Similarly

$$E(\mu^2|\bar{x}) = \frac{1}{c} \cdot \frac{1}{\sqrt{128\pi^3}} \left[\sum_{q=1}^{5} \frac{(\sqrt{5}\,\text{node}+47)^2}{1+\left(\frac{\sqrt{5}\,\text{node}-3}{8}\right)^2} \text{weight}(\text{node}) \right]$$ #

```
f_constant = function(x) {1/( 1 + ((sqrt(5)*x - 3)/8)^2 )}
weights=fun_qw(nodes)/sum(fun_qw(nodes))
c = 1/sqrt(128*pi^3) * sum(weights * f_constant(nodes))
c

## [1] 0.01342143

f_mu = function(x) {(sqrt(5)*x + 47)/(1 + ((sqrt(5)*x - 3)/8)^2)}
E_mu = (1/c) * (1/sqrt(128*pi^3)) * sum(weights * f_mu(nodes))
E_mu

## [1] 47.32485

f_mu2 = function(x) {(sqrt(5)*x + 47)^2/(1 + ((sqrt(5)*x - 3)/8)^2)}
E_mu2 = (1/c) * (1/sqrt(128*pi^3)) * sum(weights * f_mu2(nodes))
E_mu2

## [1] 2244.177

Var_mu = E_mu2 - (E_mu)^2
print(Var_mu)

## [1] 4.53585
```