In [ ]:
```
STAT 420 HW 3
Donghan Liu Netid: Donghan2
```

In [48]:
```
x = c(1, 1.2, 1.4, 1.6, 1.8, 2)
y = c(3.5, 7, 7, 9, 8.6, 11.2)
```

In [49]:
```
# Question 1  a)
x_mean = mean(x)
y_mean = mean(y)
beta1 = sum((y - y_mean)*(x-x_mean))/sum((x-x_mean)^2)
beta0 = y_mean-beta1*x_mean
beta0
beta1
# According to the calculation, beta0 = -1.99047619047619, beta1 = 6.47142857142857
```

-1.99047619047619

6.47142857142857

In [50]:
```
#b)
y_i = beta0+beta1 * x
# when subject (i) = 1
y_1 = beta0+beta1 * 1
# when subject (i) = 2
y_2 = beta0+beta1*1.2
y_1
y_2
# Thus, based on the result that generated, when i =1, the regression of Y
# is equal to 4.48095238095238, and when i = 2, y_i  = 5.77523809523809
```

4.48095238095238

5.77523809523809

In [51]:
```
#c)
SSE = sum((y-y_i)^2)
SSR = sum((y_i - y_mean)^2)
SST = sum((y-y_mean)^2)
SSE+SSR
SST
SSE
# In accordance with the data calculation and verfication, SSE = 4.05276190476191
# and has been double checked by SST and SSR
```

33.3683333333333

33.3683333333333

4.05276190476191

In [52]:
```
#d)
R2 = SSR / SST
R2_1 = 1-(SSE/SST)
R2
R2_1
# Based on the formula that presented in the lecture, R^2 = SST/ SST
# or 1-(SSE/SST), so R^2 = 0.878544670952642
```

0.878544670952642

0.878544670952643

In [53]:
```
#e)
SE = sum(y - y_i)
SE
# In this calculation, the formula is quoted from the lecture note, and
# it is also the sqrt of SSE. In the result, even though it is not a "directly
# zero, but 3.5527136788005e-15 is already extremely close to zero, and
# it might be ignore, so we would consider it as zero in this case.
e = y - beta0-beta1*x
PV = sum (t(x)*e)

PV
# Likewise, the formula is also come from lecture. The result is a very
# small number, which is 3.19744231092045e-14, just like the number in SE,
# so we would think that it is zero as well. Thus, the statement is true.
```

3.5527136788005e-15

5.55111512312578e-15

In [7]:
```
# Question 2 a)
install.packages("faraway",repos = "http://cran.us.r-project.org")
library(faraway)
data(cheddar)
head(cheddar)
# This step is for installing the package, loading the data set, and
# display the partical data from the package
```

```
package 'faraway' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Hans\AppData\Local\Temp\RtmpsNAVZZ\downloaded_packages
```

taste	Acetic	H2S	Lactic
12.3	4.543	3.135	0.86
20.9	5.159	5.043	1.53
39.0	5.366	5.438	1.57
47.9	5.759	7.496	1.81
5.6	4.663	3.807	0.99
25.9	5.697	7.601	1.09

In [54]:
```
x = cheddar$Acetic
y = cheddar$taste
x_mean = mean(x)
x_mean
y_mean = mean(y)
y_mean
s_x = sqrt(var(x))
s_x
s_y = sqrt((var(y)))
s_y
beta1 = sum((y-y_mean)*(x-x_mean))/sum((x-x_mean)^2)
beta0 = y_mean-beta1*x_mean
r_xy = beta1 *(s_x/s_y)
r_xy
xiyi =sum(x*y)
xiyi
# All of the related formula are abstracted from lecture. x mean = 5.49803333333333
# y mean = 24.5333333333333, s_x = 0.570878360335063, s_y = 16.255382839674
# r_xy = 0.549539298804285, and sum of xi*yi = 4194.4421
```

5.49803333333333

24.5333333333333

0.570878360335063

16.255382839674

0.549539298804285

4194.4421

In [55]:
```
#b)
beta1 = sum((y-y_mean)*(x-x_mean))/sum((x-x_mean)^2)
beta0 = y_mean-beta1*x_mean
beta0
beta1
# As presented, beta0 is y-intercept and beta1 is slope of the line
# Therefore, y (taste) = 15.6477672095797*x(acetic acid) --61.4986123771764
```

-61.4986123771764

15.6477672095797

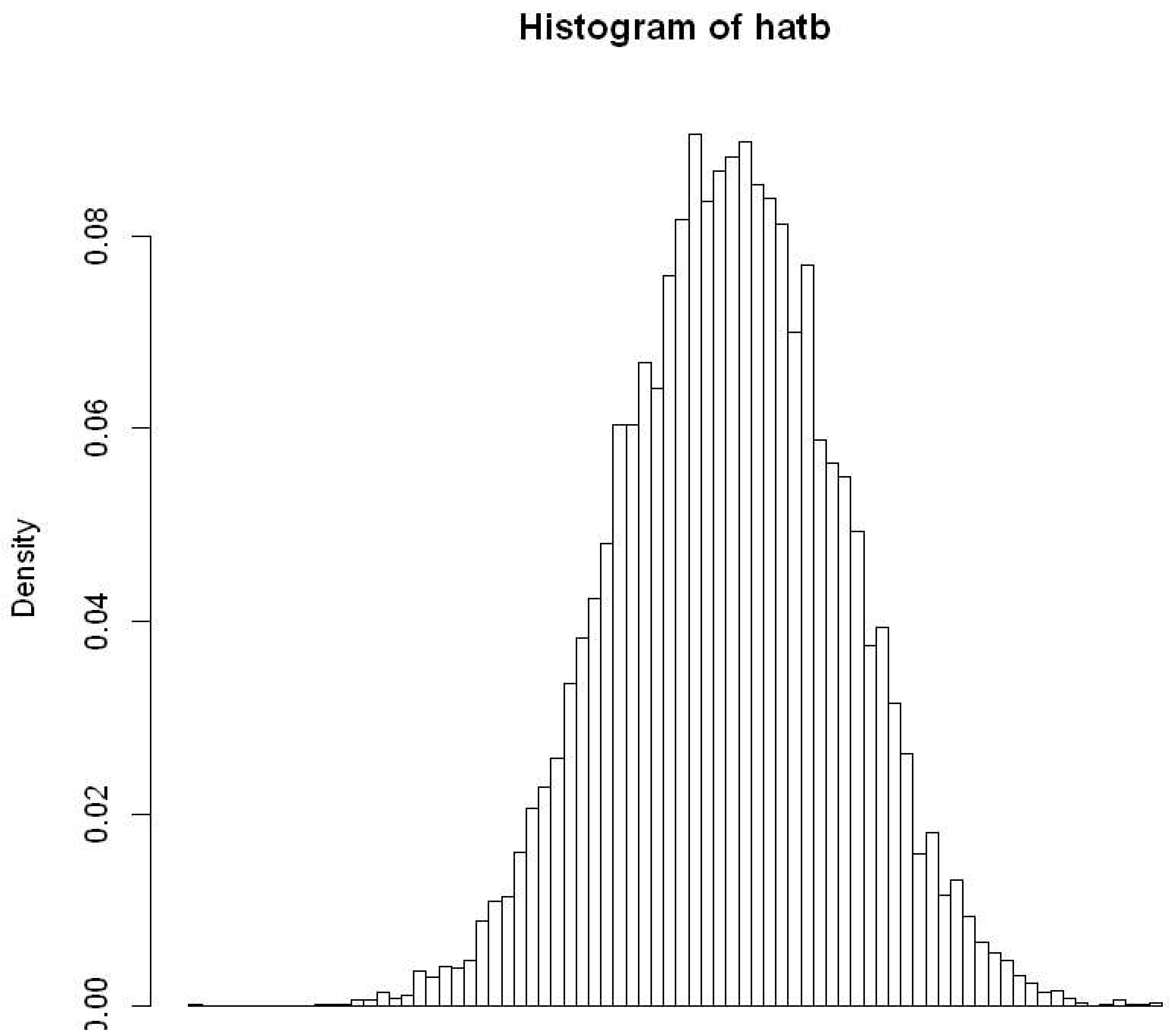
In [56]:
```
#c)
lm(taste~Acetic, data = cheddar)
# For the linear model formula, when x is Acetic, y is taste, the
# intercept is -61.50 and the slope is 15.65, the results that calculated
# in #C and #B are extremely close, thus, the answer from #B is believed
# as same
```

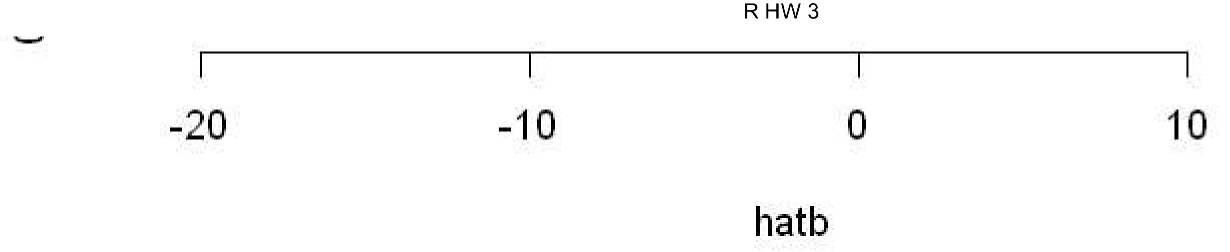
```
Call:
lm(formula = taste ~ Acetic, data = cheddar)

Coefficients:
(Intercept)       Acetic
     -61.50        15.65
```

In [57]:
```
#d)
y1= c(3.5, 7, 7, 9, 8.6, 11.2)
beta00 = mean( y1 )
n = nrow ( cheddar )
sigma2 = sum( lm ( taste~Acetic, data = cheddar ) $ residuals^2 ) / ( n-2)
hatb = rep (NA, 10000)
for ( i in 1:10000 )
{
y = beta00 + rnorm ( n , mean = 0 , sd = sqrt( sigma2 ) )
hatb [ i ] = lm ( y ~ x ) $coef [ 2 ]
}
hist(hatb, prob = TRUE, breaks = 100)
mean( hatb  > 15.65)
# First, sigma^2 = SSE / (n-2), which is the formula that I applied above
# Second, we want our sample as large as possible in order to have a
# relatively high accurate result, so the 10000 position is set for hatb
# Then, since the y follow the normal distribution, we apply rnorm in here
# In the following code, I use lm ( y ~ x ) $coef [ 2 ] since it represents
# the slope of the line, which is the element that we are doing the test
# Finally, use the mean and variance of hatb to calculate the probability
# of the slope is greater to one by utilizing pnorm(), and the final result
# is 0.0006, obviously, it is much less than 0.05, which is
# the cut-off for siginifance level. Therefore, we would accept the alternative,
# and reject the null hypothesis, namely, the slope of line is unlikely
# equal to 0
```

6e-04

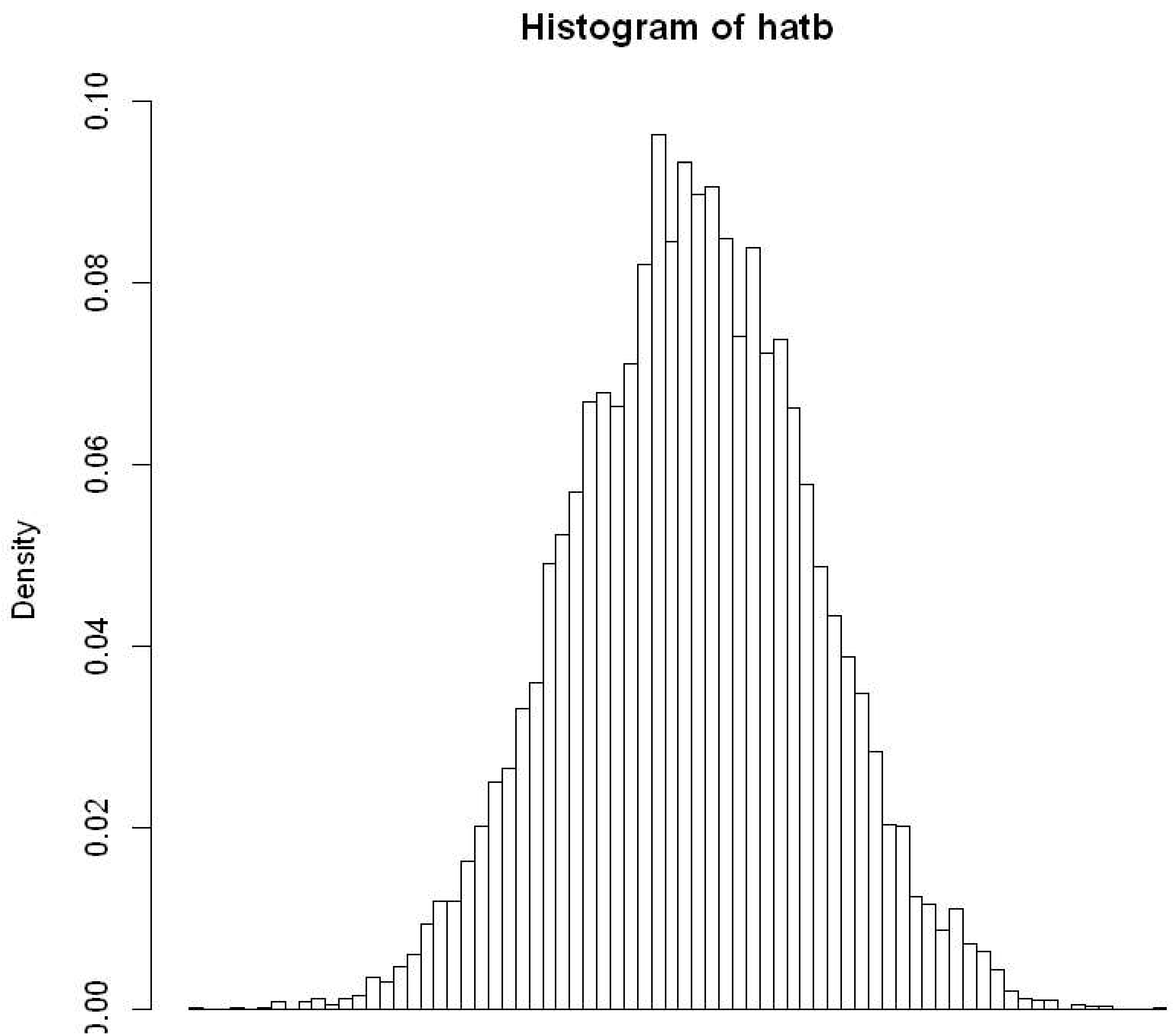

Histogram of hatb
Density
0.08
0.06
0.04
0.02
0.00

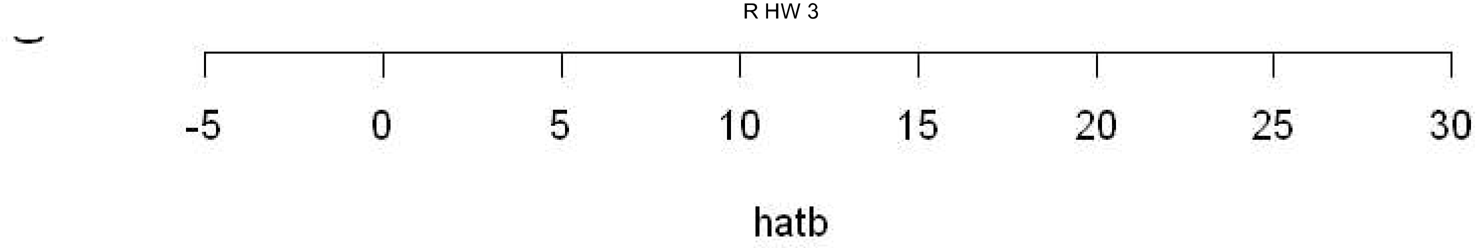

-20
-10
0
10
hatb

In [58]:
```
#e)
y1 = c(3.5, 7, 7, 9, 8.6, 11.2)
beta00 = mean( y1 )
n = nrow ( cheddar )
sigma2 = sum( lm ( taste~Acetic, data = cheddar ) $ residuals^2 ) / ( n-2)
hatb = rep (NA, 10000)
beta1 = 12
for ( i in 1:10000 )
{
y = beta00 + rnorm ( n , mean = 0, sd = sqrt( sigma2 ) ) + beta1*x
hatb [ i ] = (lm ( y ~ x ) $coef [ 2 ])
}
hist(hatb, prob = TRUE, breaks = 100)
mean( hatb > 15.65)

# Similar with the analysis that indicated above, and the only difference
# is that the alternative is 12 instead of 0. So we have to use Y = beta0
# + beta1*x + e and set beta1 = 12, which is the null hyphoesis.
# Thus, as the result shows, the probability is 0.2036 > 0.05, so we would like to \
# conclude that when beta1 = 12, the null hypothesis is accpeted.
```

0.2036


Histogram of hatb
Density
0.00
0.02
0.04
0.06
0.08
0.10


-5
0
5
10
15
20
25
30
hatb