Assignment_1(4)

ankit.tewari@estudiant.upc.edu

Assignment 1(4)

This simulation is a part of the assignment exercise provided under the subject Advanced Statistical Inference. The simulations will be carried out on predefined formula for random random variables T_n, V_n and U_n and wherever required, comparisons will be drawn. Further, the simulation will also draw some attention on proving the distributions to which these random variables belong to.

THe sampling is carried out 50 times using a fixed sample size of 40 and results cn be improved by examining the conditions that arise as we increase the sample size. However, due to lack of time and computational resources, the author has tried to restrict himself in a specific set of circumstances.

```
#Defining Populations 1 and Populations 2 from where we will conduct sampling
set.seed(124)
population_1<-rnorm(100000, mean = 0.00, sd=1)
set.seed(154)
population_2<-rnorm(100000, mean = 0.5, sd=sqrt(3)*sd(population_1))

#Means of the two populations
mean(population_1)
```

```
## [1] 0.0006122519
```

```
mean(population_1)
```

```
## [1] 0.0006122519
```

```
#Variances of the two populations
var(population_1)
```

```
## [1] 1.002974
```

```
var(population_2)
```

```
## [1] 3.023199
```

```
# Standard deviation of the two ppopulations
sd(population_1)
```

```
## [1] 1.001486
```

```
sd(population_2)
```

```
## [1] 1.738735
```

```
##IDEA 1: We will bring many samples of same sizes from the population first and conduct all experiments
##IDEA 2: Later, we will define the samples of bigger sizes for two populations and conduct some measure



sample_size<-40 ####For each sample
number_of_samples<-50  ####Total number of samples

#Creating 50 Samples X1, X2, . . . . ,X50 where each is of size 40
sample_x<-matrix(rep(0, 200), nrow = 50, ncol = 40)
for(i in 1:dim(sample_x)[1]) {
```

```
  sample_x[i,]<-sample(population_1, 40)
}

#Creating 50 Samples Y1, Y2, . . . . ,Y50 where each is of size 40
sample_y<-matrix(rep(0, 200), nrow = 50, ncol = 40)
for(i in 1:dim(sample_y)[1]) {

  sample_y[i,]<-sample(population_2, 40)
}



#-----------------We are about to obtain the distribution of sample means now!--------------------

#Computing the Means for 50 X's
sample_means_x<-rep(0,50)
sample_means_x<-apply(sample_x,1,mean )
#Computing the Means for 50 Y's
sample_means_y<-rep(0,50)
sample_means_y<-apply(sample_y,1,mean )

##Verifying if the Expectation of distribution of sample means estimates true population means
mean(sample_means_x) #.....................For population X
```

```
## [1] 0.003411877
```

```
mean(sample_means_y) #.....................For population Y
```

```
## [1] 0.5046903
```

```
##-----------We are about to define the Sample Variances for the 50 samples from 2 Populations


##Matrix of squared mean deviations divided by (1/n-1) for X's
sample_x_mean_deviation<- matrix(rep(0,200), nrow = 50, ncol = 40)
for(i in 1:dim(sample_x)[1]) {
  m<-mean(sample_x[i,])
  sample_x_mean_deviation[i,]<-(1/(sample_size-1))*(sample_x[i,] - m)**2
}
sample_variances_x<-rep(0,50)
sample_variances_x<-apply(sample_x_mean_deviation, 1, sum)

##Matrix of squared mean deviations divided by (1/n-1) for Y's
sample_y_mean_deviation<- matrix(rep(0,200), nrow = 50, ncol = 40)
for(i in 1:dim(sample_y)[1]) {
  m<-mean(sample_y[i,])
  sample_y_mean_deviation[i,]<-(1/(sample_size-1))*(sample_y[i,] - m)**2
}
sample_variances_y<-rep(0,50)
sample_variances_y<-apply(sample_y_mean_deviation, 1, sum)



#
```

```r
mean(sample_variances_x)
```

```
## [1] 0.9433618
```

```r
var(population_1)
```

```
## [1] 1.002974
```

```r
mean(sample_variances_y)
```

```
## [1] 2.869958
```

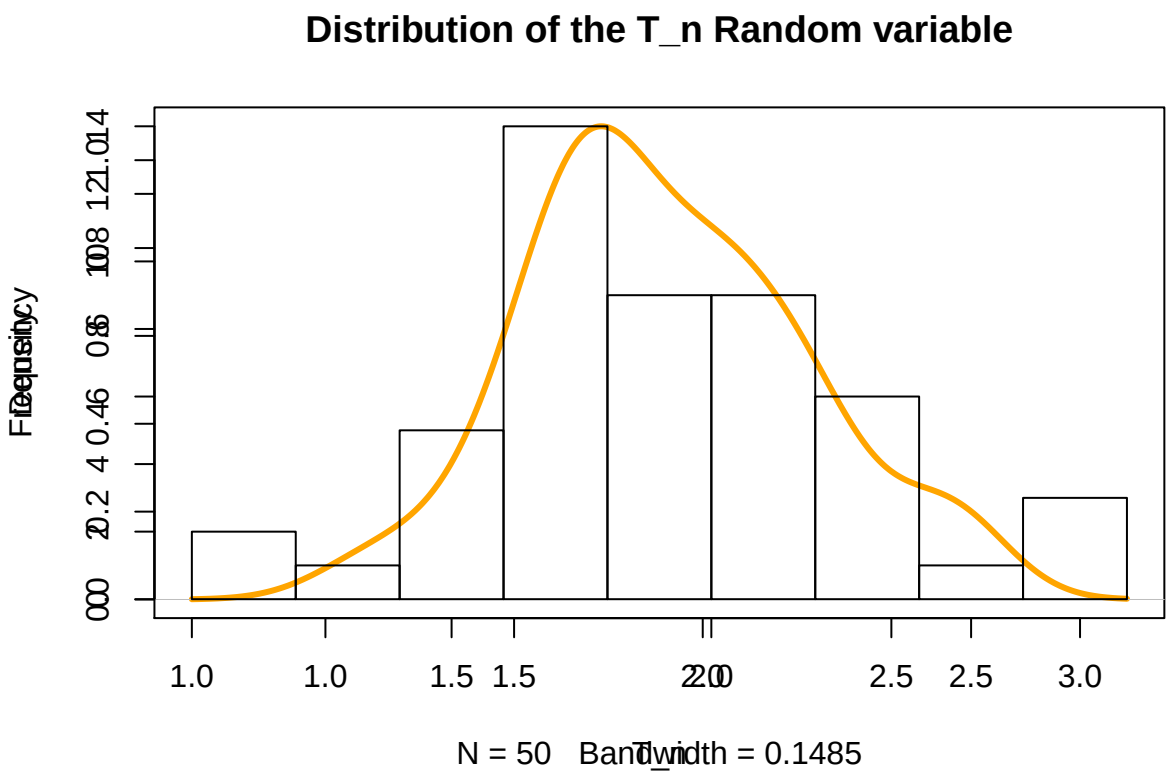
```r
var(population_2)
```

```
## [1] 3.023199
```

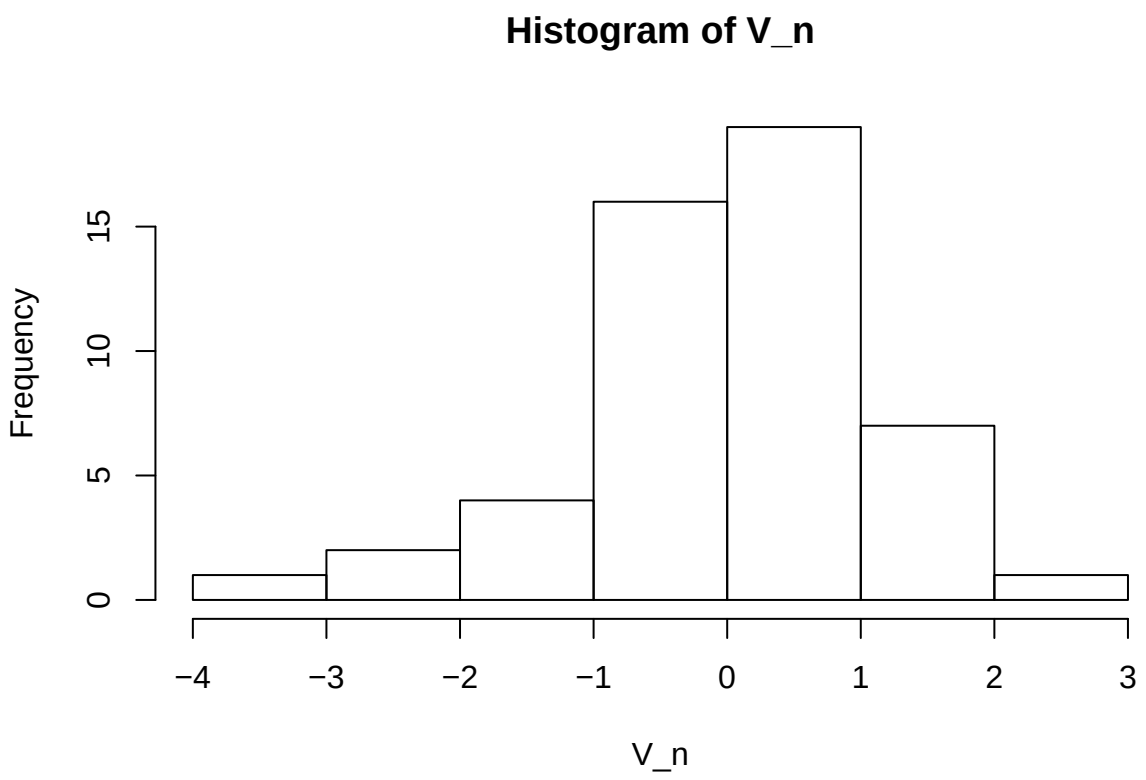
```r
##Investigations begin from here-

##Defining T_n
T_n<- sample_variances_x + (1/3)*sample_variances_y
summary(20*T_n/var(population_1))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.77   33.02   36.82   37.89   42.67   53.42
```

```r
plot(density(T_n), main = "Distribution of the T_n Random variable", col="Orange", lwd=3 )
par(new=T)
hist(T_n, main="")
```



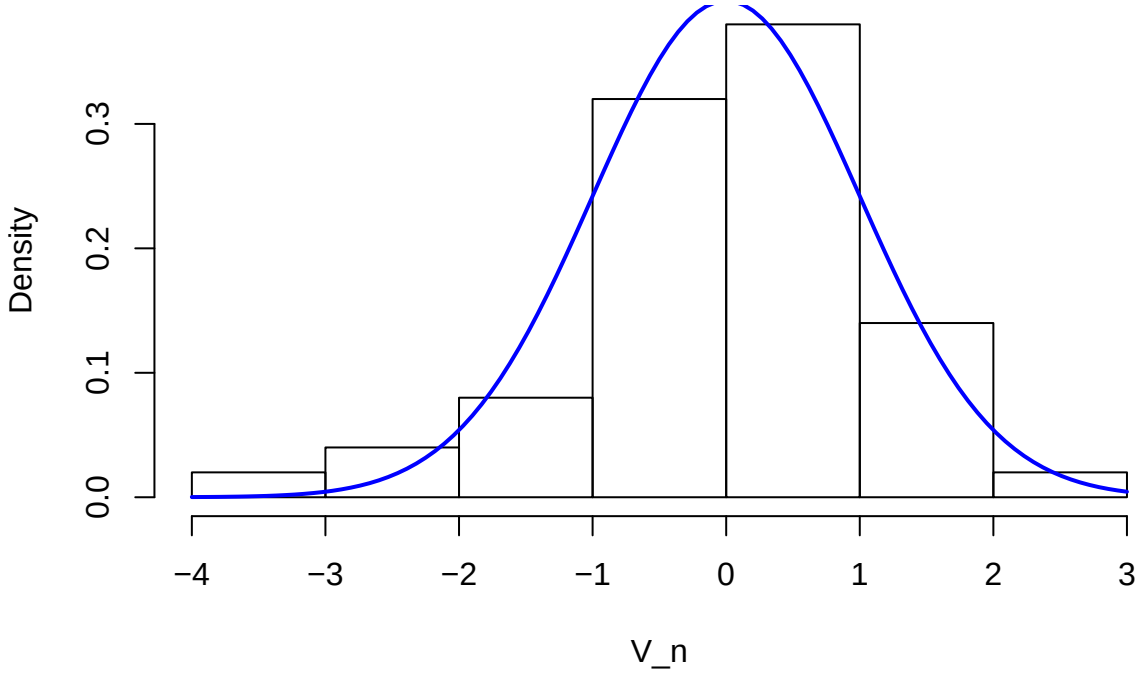
```r
## 4(1)) Defining V_n
sigma<-sqrt(var(population_1))
V_n<- (1/(2*sigma))*sqrt(sample_size)*(sample_means_x - sample_means_y - mean(population_1) +mean(popula
hist(V_n)
```



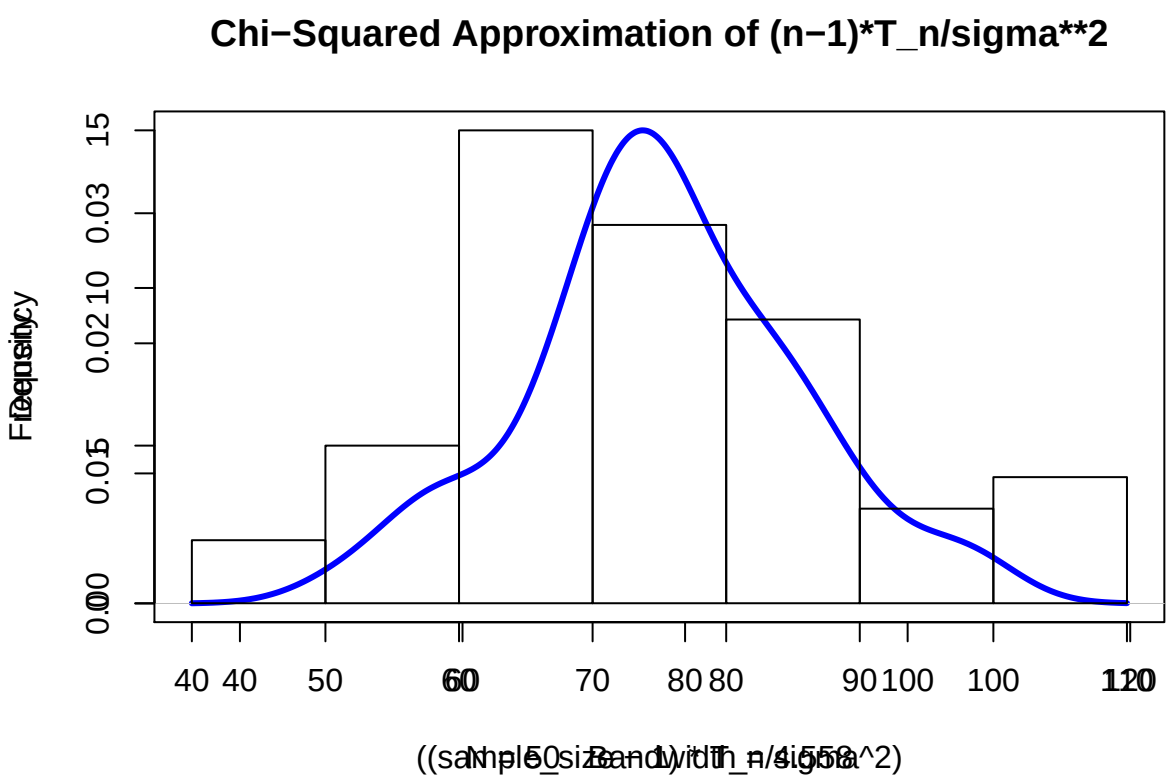

```r
density(V_n)
```

```
##
## Call:
##  density.default(x = V_n)
##
## Data: V_n (50 obs.); Bandwidth 'bw' = 0.3189
##
##        x                 y
##  Min.   :-4.3673   Min.   :0.0002814
##  1st Qu.:-2.5264   1st Qu.:0.0206715
##  Median :-0.6855   Median :0.0880232
##  Mean   :-0.6855   Mean   :0.1356643
##  3rd Qu.: 1.1554   3rd Qu.:0.2008299
##  Max.   : 2.9963   Max.   :0.4674705
```

```r
hist(V_n, freq=FALSE)
curve(dnorm(x, 0, 1), col="blue", add=TRUE, lwd=2)
```




```
## 4(2) Proving that (n-1)*T_n/sigma**2 follows chi-squared distribution
#A sample chi-squared random variable with 2(n-1)=78 degrees of freedom
chi_sq<-rchisq(50, df=78)

##Plotting the curves of the two for examining the relationship


plot(density(chi_sq), col="Blue", lwd=3, main = "Chi-Squared Approximation of (n-1)*T_n/sigma**2")
par(new=TRUE)
hist(((sample_size-1)*T_n/sigma**2), main = "")
```




```
#The relationship is approximately chi-squared. Results can be improved by increasing the sample size.

summary((sample_size-1)*T_n/sigma**2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   42.44   64.40   71.81   73.88   83.20  104.16
```

```
summary(chi_sq)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   49.35   71.22   77.51   77.90   86.06  106.03
```

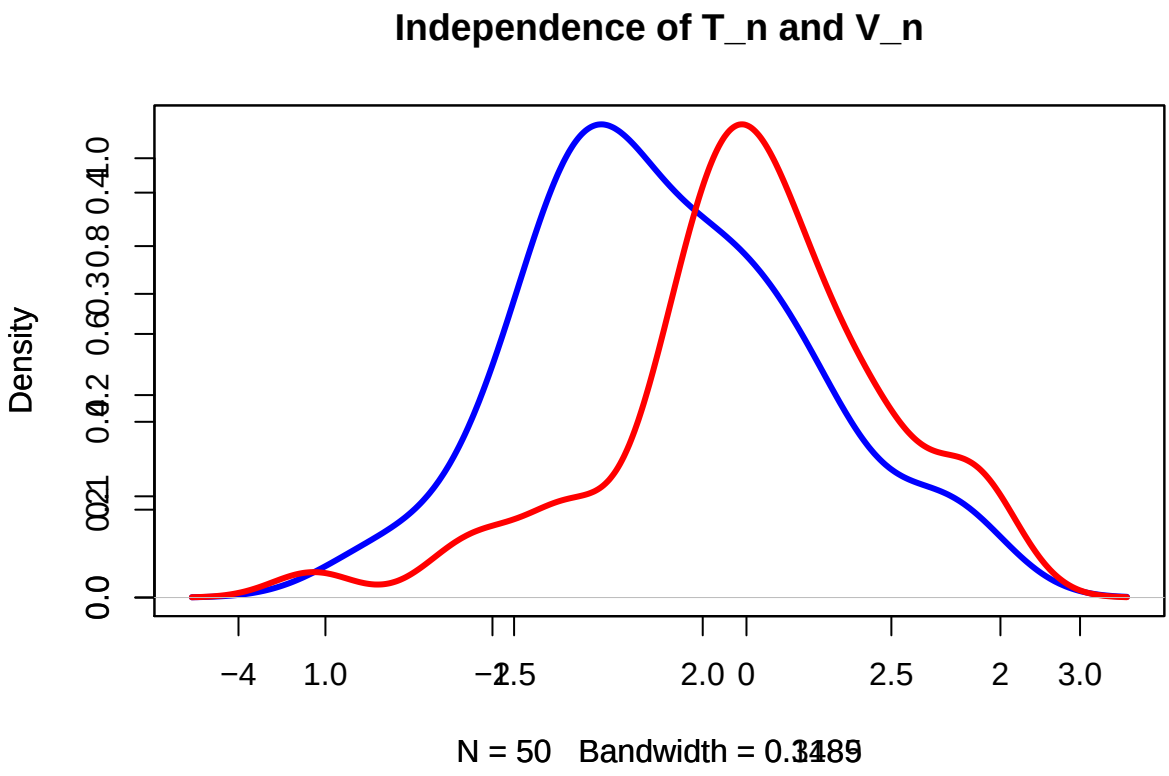
```
## As we can see that they are almost similarly distributed.
#The distributions becomes precise Once the sample size increases to a very high value


##### d)
diff_vector<-T_n- 2*(sigma**2)
mean(diff_vector)
```

```
## [1] -0.105933
```

Including Plots

You can also embed plots, for example:

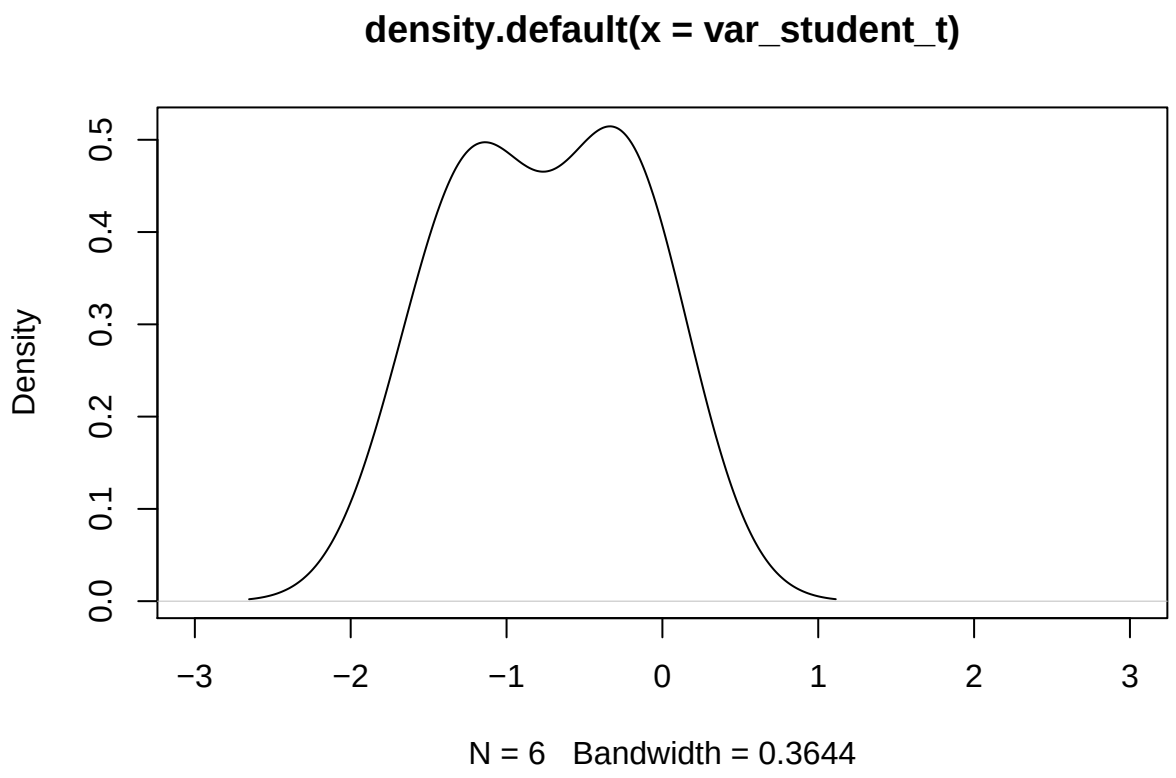
```
## [1] 0.02144282
```




```
## 4(3)
## Creating a t-distributed random variable and comparing it's properties with our U_n
## We are calling U_n as "Cover_2
var_student_t<-rt(6, 98)
density(var_student_t)
```

```
##
## Call:
##  density.default(x = var_student_t)
##
## Data: var_student_t (6 obs.);    Bandwidth 'bw' = 0.3644
##
##        x                y
##  Min.   :-2.6520   Min.   :0.002102
##  1st Qu.:-1.7109   1st Qu.:0.048794
##  Median :-0.7698   Median :0.276420
##  Mean   :-0.7698   Mean   :0.265273
##  3rd Qu.: 0.1713   3rd Qu.:0.473096
##  Max.   : 1.1123   Max.   :0.514551
```

```
plot(density(var_student_t), xlim=range(-3,3))
```




```
summary(var_student_t)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.55873 -1.12838 -0.74049 -0.75013 -0.35198  0.01911
```

```
var(var_student_t)
```

```
## [1] 0.3623217
```

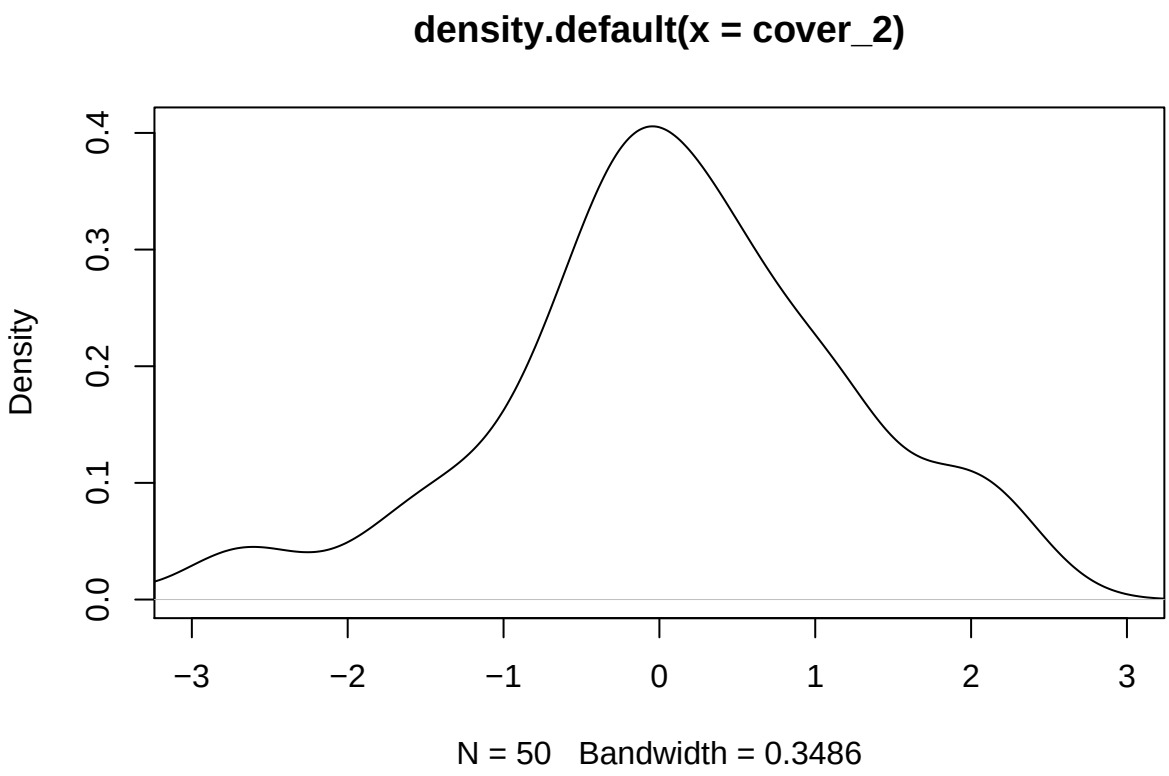
```
cover_2<-sqrt(number_of_samples)*(sample_means_x - sample_means_y - mean(population_1) +mean(population_
summary(cover_2)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.97124 -0.49559  0.03008  0.01209  0.63924  2.26371
```

```
var(cover_2)
```

```
## [1] 1.545938
```

```
plot(density(cover_2), xlim=range(-3,3))
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.