Statistics 2A Coursework

Conor Maguire

19/11/2021

Part 1

Question 1

We want to show that R_1 is a consistent estimator of $cor(X,Y) = \frac{cov(X,Y)}{\sqrt{\sigma_X \sigma_Y}}$ (ie that it converges in probability to this)

By theorem 2.6 of the lecture notes, it suffices to show that $\sqrt{S_x^2 S_y^2}$ converges in probability to $\sigma_X \sigma_Y$ and S_{xy} conerges in probability to $cov(X,Y)$

1.

By Proposition 2.2 of the lecture notes T_1 = $\frac{S_X^2}{n}$ converges in probably to σ_X^2, so by this ($E[X^2 Y^2]$ is finite) and theorem 2.6, S_X^2 converges in probability to $n\sigma_X^2$. A similar argument can be used for S_Y^2. So, by theorem 2.6, $\sqrt{S_x^2 S_y^2}$ converges in probability to $\sqrt{n^2 \sigma_X^2 \sigma_Y^2} = n\sigma_X \sigma_Y$

2.

We have:

$$S_{xy} = \sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})$$
$$= \sum_{i=1}^{n} (X_i Y_i - \overline{X}\overline{Y})$$
$$= \sum_{i=1}^{n} (X_i Y_i - E[X]E[Y])\$$$
$$= \sum_{i=1}^{n} (X_i - E[X])(Y_i - E[Y]) = ncov(X,Y)$$

So R_1 converges in probability to $\frac{cov(X,Y)}{\sqrt{\sigma_X \sigma_Y}}$ and so is a consistent estimator of ρ as required.

For R_2 we use our existing arguments and theorem 2.6 along with out assumption that $\sigma_X^2 = \sigma_Y^2$ to find that it converges in probability to $\frac{2ncov(X,Y)}{2n\sigma_X^2}$ = $cor(X,Y)$ = ρ.

Since R_3 is just the average of R_1 and R_2, both of which are consistent (when $\sigma_X^2 = \sigma_Y^2$) it is also a consistent estimator by theorem 2.6.

Question 2

We write a function to calculate the estimators given n pairs of a bi-variate normal distribution. We first use a for-loop to calcuate the sample quantities, then we use these to obtain the three estimators.

```
estimateCalc=function(data_matrix){
  xVals=data_matrix[,1] #first column of the matrix is x values
  yVals=data_matrix[,2] #second column is y values
  xMean=mean(xVals)
  yMean=mean(yVals)
  sx2=0 #set up sample quantities
  sy2=0
  sxy=0
  n=length(data_matrix[,1]) #set number of repeats to n, the number of rows
  for (i in 1:n){
    #calculate the sample quantities:
    sx2=sx2+(data_matrix[i,1]-xMean)^2
    sy2=sy2+(data_matrix[i,2]-yMean)^2
    sxy=sxy+((data_matrix[i,1]-xMean)*(data_matrix[i,2]-yMean))
  }
  #calculate estimators
  R1=sxy/(sqrt(sx2*sy2))
  R2=(2*sxy)/(sx2+sy2)
  R3=(R1+R2)/2
  estimators=c(R1,R2,R3) #puts all the estimators into a vector
  return(estimators)
}
```

This function produces the following output when the matrix used is the example in the coursework:

```
## [1] 0.5919672 0.5578380 0.5749026
```

Question 3

We want to find the mean squared error of our estimators. The MSE of an estimator can be calculated in the following way:

$$MSE(\hat{\theta}, \theta) = Var(\hat{\theta}) + Bias(\hat{\theta}, \theta)^2$$

We perform a simulation:

```
sim_mse=function(n, mu, var, rho){
  nsim=1000 #numbver of simulations, must stay constant for each n
  est_array=matrix(nrow = nsim, ncol = 3) # create an array to store estimator values
  for (i in 1:nsim){
    sim_data=rbivnorm(n = n, mu = mu, var = var, rho = rho) #creates a random matrix of pairs
from the bi-variate normal dist.
    est_array[i,]=estimateCalc(sim_data) #add the values of the estimators to out array
  }
  #calculate MSE for each estimator using our formula:
  mse_R1=var(est_array[,1])+(mean(est_array[,1])-rho)^2
  mse_R2=var(est_array[,2])+(mean(est_array[,2])-rho)^2
  mse_R3=var(est_array[,3])+(mean(est_array[,3])-rho)^2
  return(c(mse_R1, mse_R2, mse_R3))
}
```

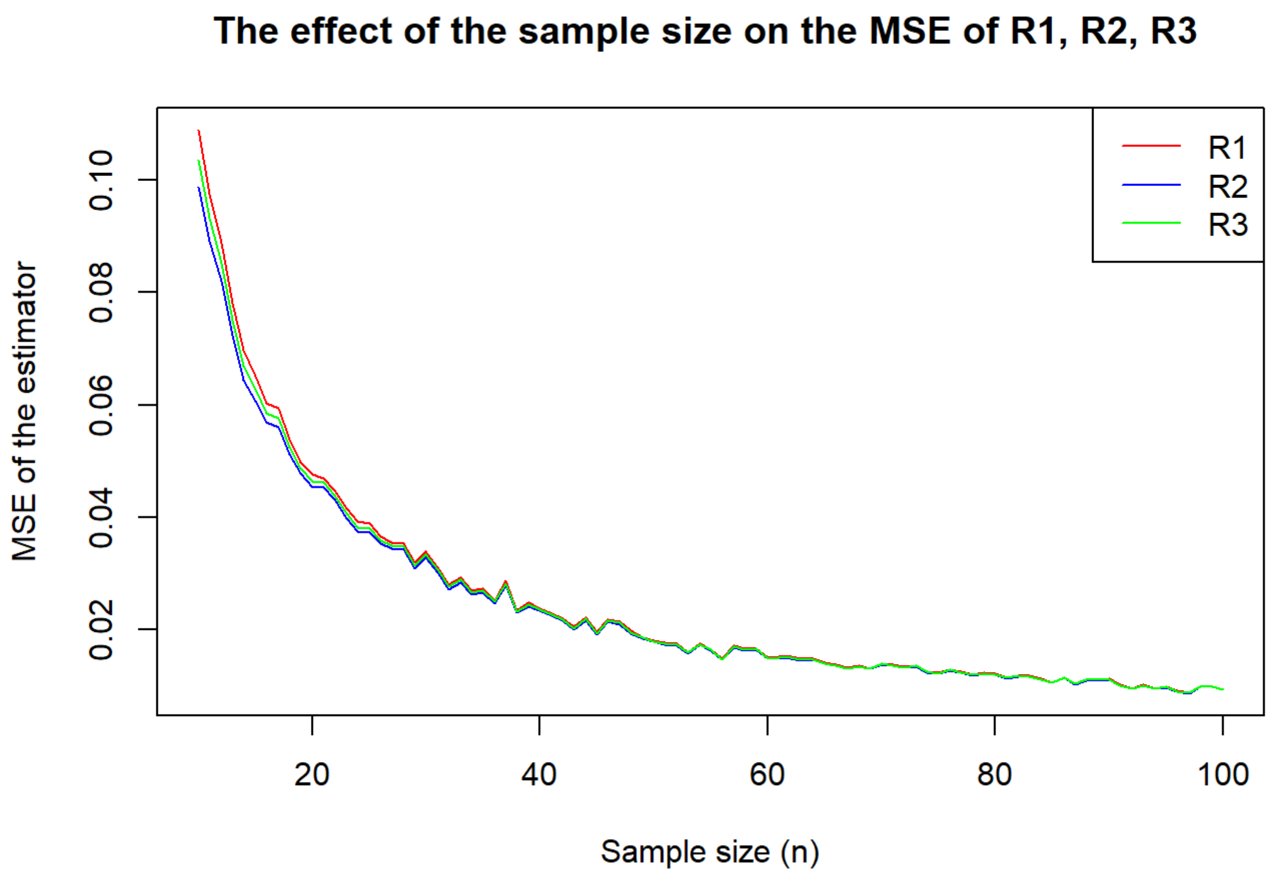
Let’s do a comparison for different values of n: When n=10 the simulation produces the following MSEs:

```
## [1] 0.10321555 0.09335018 0.09803147
```

However when n=100 we instead get:

```
## [1] 0.009762505 0.009692456 0.009726829
```

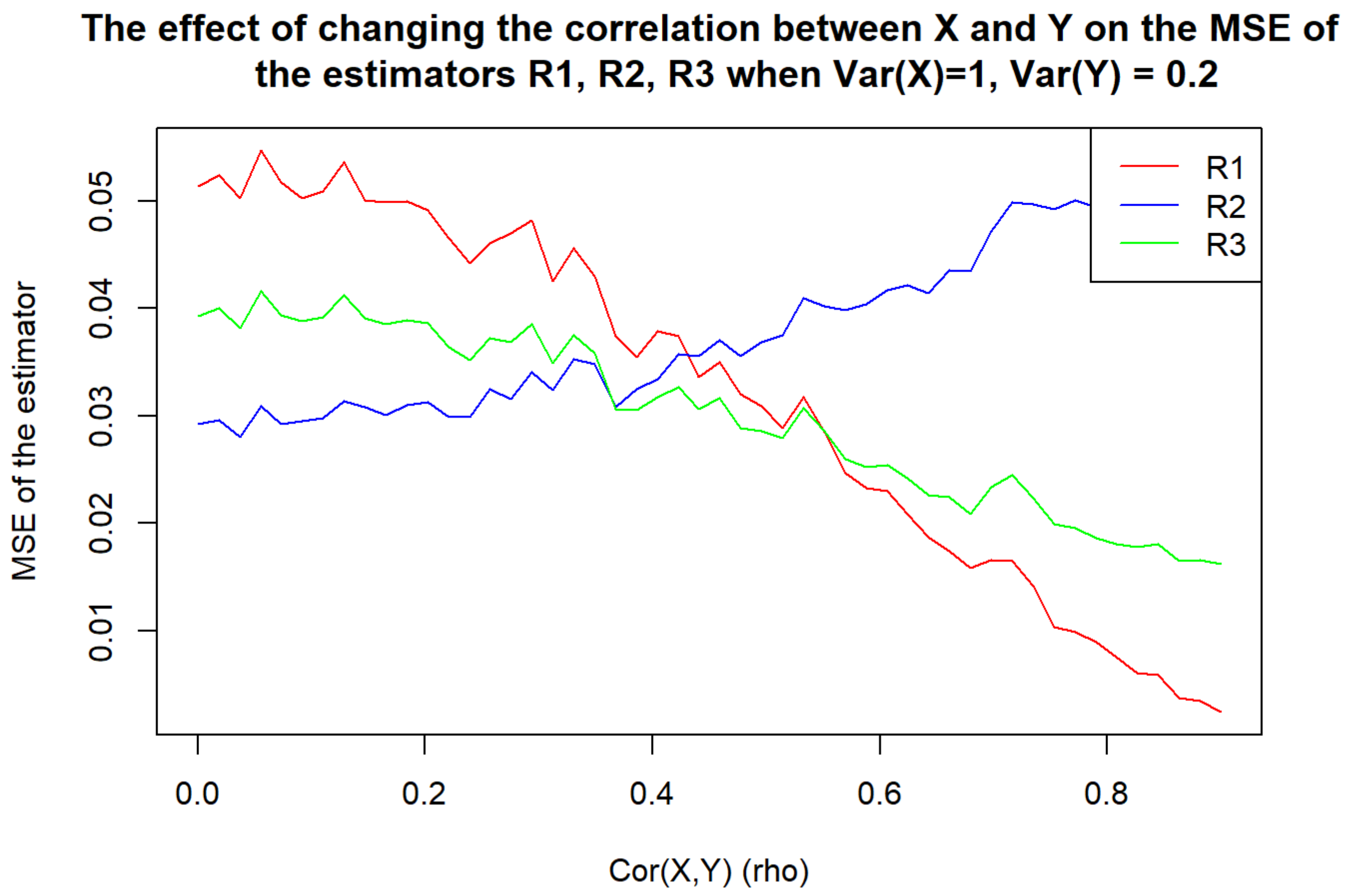
These values are much lower than when n equaled 10. Plotting the results for each estimator gives us the following graph:



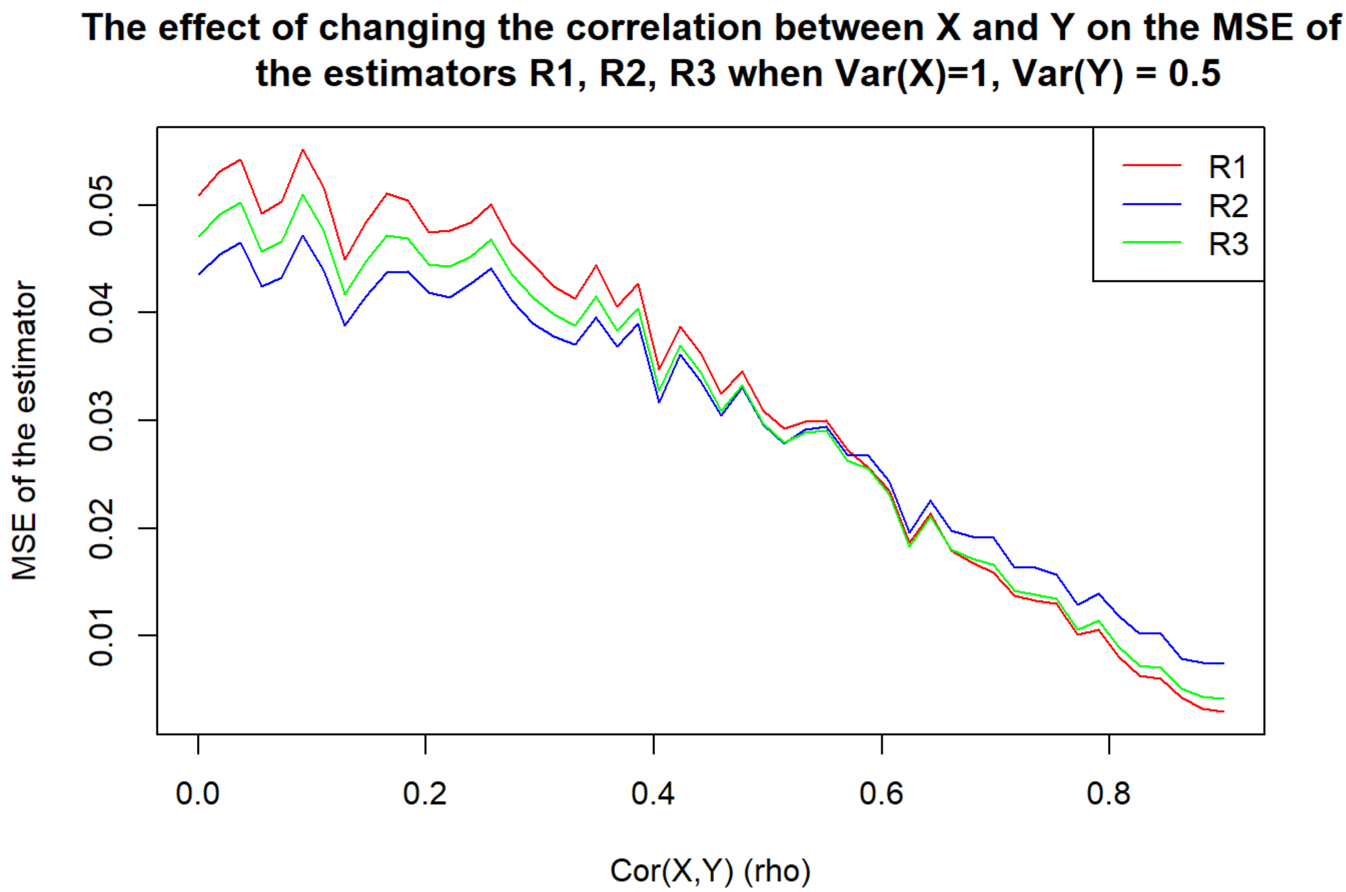
From the plot we can conclude that, although each estimator tends towards 0 as n grows, R2 has a lower MSE for smaller values of n so is the estimator of choice followed by R3 then finally R1. R2 (as well as R1 and R3) is also consistent as Var(X) = Var(Y) in this case.

Question 4

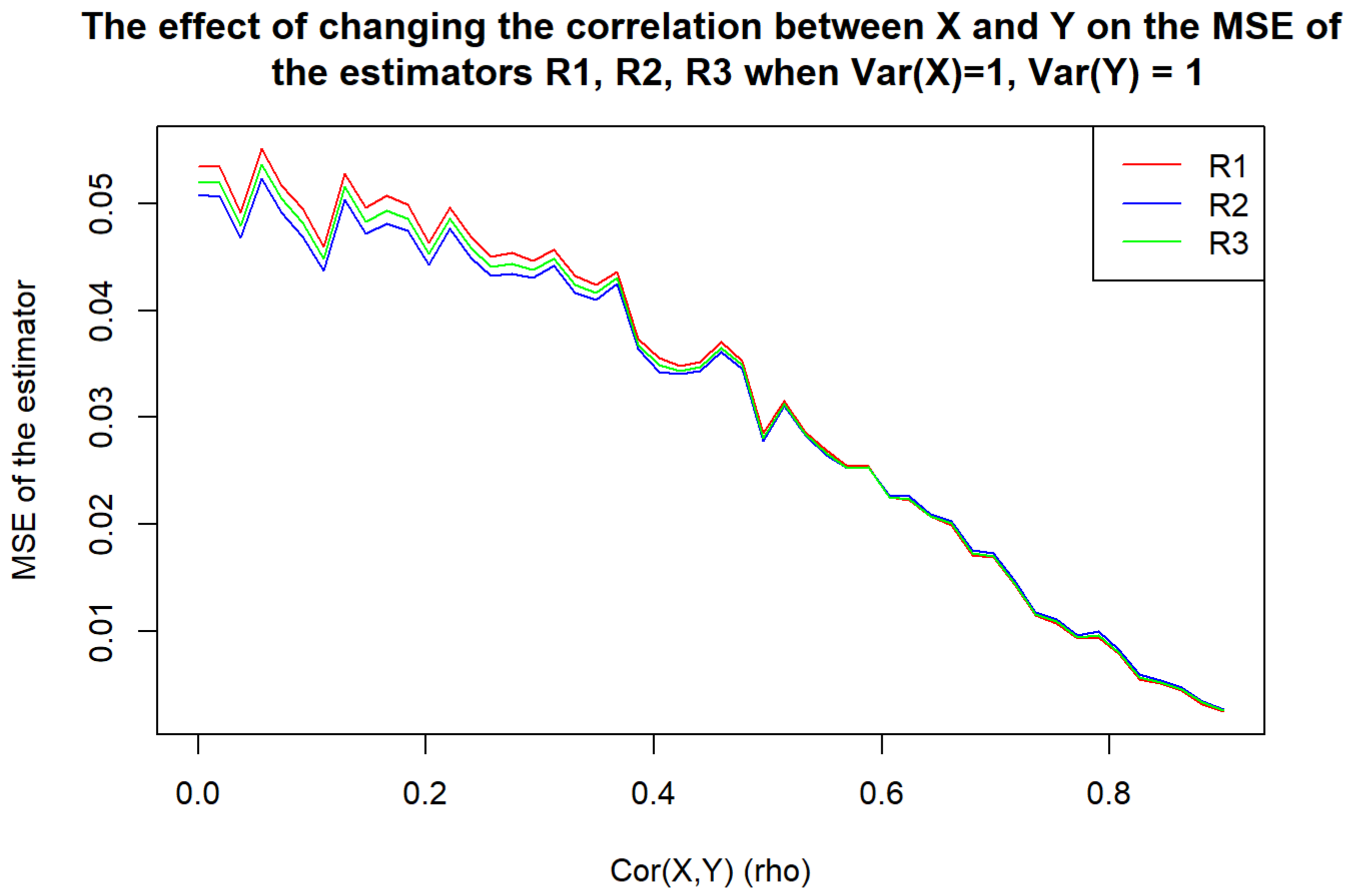
We repeat the simulation, changing the variances to (1, a) for $a \in 0.2, 0.5, 1$ starting with a = 0.2:



We see on this plot that the lines intersect at roughly $\rho = 0.45$ so for $\rho < 0.45$ R2 is the best choice of estimator, but R1 has a smaller MSE when $\rho > 0.45$ is the the best choice in those cases. Also since the variances aren't equal, R2 isn't necessarily a consistent estimator, however R1 always is so may be a better choice. We repeat for a = 0.5:



Here we notice that the lines intersect at around $\rho = 0.55$, with R2 being the best estimator (having the lowest MSE) before then, and R1 having the lowest MSE afterwards. However the variances still aren't equal so R2 is not necessarily consistent. Lastly for a = 1:



Here R2 is our best choice of estimator for ρ until around $\rho = 0.5$, after which the estimators have roughly the same MSE. In conclusion, R2 is the best estimator for ρ up to a certain number dependent on Var(Y) after which R1 is the better choice, unless Var(Y) = Var(X) in which case all estimators are roughly equally as good after this point. If the coordinate of the intersection is not known then R3 can be used as it is the average of the two so will never have the greatest MSE.

Part 2

Question 5

We construct a 95% confidence interval for ρ:

We use the definition of convergence in law along with the central limit theorem to turn our given equation into: $\frac{\sqrt{n}(R_1-\rho)}{1-\rho^2} \to N(0,1)$ (in law)

We can apply Slutsky's theorem to replace ρ on the denominator with R1 since it is a consistent estimator of ρ, then we can rearrange to get the following confidence interval (a= 0.05):

$$(R_1 - 1.96\frac{1-R_1^2}{\sqrt{n}}, R_1 + 1.96\frac{1-R_1^2}{\sqrt{n}})$$

We will perform a simulation to check the coverage of this confidence interval:

```
ciSim=function(nsim, n, mu, var, rho){
  ciVals=matrix(0, nrow=nsim, ncol=3) #cols 1 and 2 for CI values, col 3 for coverage checkin
g
  for (i in 1:nsim){
    sim_data=rbivnorm(n = n, mu = mu, var = var, rho = rho)
    est_values=estimateCalc(sim_data) #calculate estimators
    R1=est_values[1]
    #Calculate confidence interval
    ciVals[i,1] = R1-1.96*((1-R1^2)/sqrt(n))
    ciVals[i,2] = R1+1.96*((1-R1^2)/sqrt(n))
    if ((ciVals[i,1]<0) && (ciVals[i,2]>0)){ #check if rho(=0) is in the confidence interval
      ciVals[i,3]=1 #if rho is in the CI then we put a 1 in column three, otherwise we leave
 it as zero
    }
  }
coverage=mean(ciVals[,3]) #calculating the mean of the third row gives us the coverage
return(coverage)
}
```

If we run this function with nsim = 1000, n = 250, mu = (0,2), var = (1,2), rho = 0 we get the following coverage:

```
## [1] 0.934
```

This is close to 95% coverage as expected since a=0.05

Question 6

Using the first statement and the same steps as before we can construct the following:: $(arctanh(R_1) - \frac{1.96}{\sqrt{n-2}} < arctanh(\rho) < arctanh(R_1) + \frac{1.96}{\sqrt{n-2}})$ So our confidence interval is:

$$(tanh(arctanh(R_1) - \frac{1.96}{\sqrt{n-2}})), tanh(arctanh(R_1) + \frac{1.96}{\sqrt{n-2}})))$$

The second statement involves the t-distribution so we can form the following CI:

$$(R_1 - t_{n,0.975}\sqrt{\frac{1-R_1^2}{n-1}}, R_1 + t_{n,0.975}\sqrt{\frac{1-R_1^2}{n-1}}$$

Note that when $\sigma_X^2 = \sigma_Y^2$, $\sqrt{\frac{S_x^2}{S_y^2}} = \frac{Var(X)}{Var(Y)} = 1$

Question 7

We modify our function from question 6 and perform a simulation to calculate the relative merit of each CI:

```
ciCompare=function(nsim, n, mu, var, rho){
  #CI in question 5, we use the existing function
  coverage1=ciSim(nsim, n, mu, var, rho)
  #first CI in question 6, from previously used function
  ciVals=matrix(0, nrow=nsim, ncol=3) #reset data matrix
  for (i in 1:nsim){
    sim_data=rbivnorm(n = n, mu = mu, var = var, rho = rho)
    est_values=estimateCalc(sim_data) #calculate estimators
    R1=est_values[1]
    #Calculate confidence interval
    ciVals[i,1] = tanh(atanh(R1)-1.96/(sqrt(n-2)))
    ciVals[i,2] = tanh(atanh(R1)+1.96/(sqrt(n-2)))
    if ((ciVals[i,1]<0) && (ciVals[i,2]>0)){ #check if rho is in the confidence interval
      ciVals[i,3]=1 #if rho is in the CI then we put a 1 in column three, otherwise we leave
 it as zero
      }
    }
  coverage2=mean(ciVals[,3]) #calculate coverage
  #second CI in question 6
  ciVals=matrix(0, nrow=nsim, ncol=3) #reset data matrix
  for (i in 1:nsim){
    sim_data=rbivnorm(n = n, mu = mu, var = var, rho = rho)
    est_values=estimateCalc(sim_data) #calculate estimators
    R1=est_values[1]
    #Calculate confidence interval
    ciVals[i,1] = R1 - qt(p = 0.975, df = n)*(sqrt(1-R1^2))/sqrt(n-1)
    ciVals[i,2] = R1 + qt(p = 0.975, df = n)*(sqrt(1-R1^2))/sqrt(n-1)
    if ((ciVals[i,1]<0) && (ciVals[i,2]>0)){ #check if rho is in the confidence interval
      ciVals[i,3]=1 #if rho is in the CI then we put a 1 in column three, otherwise we leave
 it as zero
      }
    }
  coverage3=mean(ciVals[,3])
  return(c(coverage1,coverage2,coverage3)) #calculate coverage
}
```

Now we will carry out some simulations. First we shall try the same values we used in question 5 but with n = 20 instead:

```
## [1] 0.889 0.944 0.945
```

(The third value is invalid here since the variances are not the same (1,2)) We can see that the second confidence interval has the best coverage in this case.

Now we try var=(3,3) (all other values the same):

```
## [1] 0.896 0.949 0.934
```

We can see that the third confidence interval has the best coverage and is valid since the variances are the same.

Overall, the first confidence interval has the worst coverage for smaller samples so should not be used when n is small. The third has the best coverage, but is only valid when Var(X) = Var(Y). The second CI should be used when n is small the the variances are not equal.