HW3_Alex_Parra

Alex Parra

15/2/2022

Exercise

Write a function **VarianceCI** that does the following.

(1) takes the number of simulations **N**, the sample size **n**, the parameters **mean** and **sdev**, level of confidence **alpha** as inputs,

(2) generates random samples of size **n** from the normal distribution with parameters **mean** and **sdev**,

(3) compute $(100 - alpha)\%$ CI for the variance **sdev**^2 (you may want to use **qchisq**, the quantile function of the chi-squared distribution),

(4) repeat (2)-(3) **N** times,

(5) compute the successful coverage proportion, that is the number of times the CI actually contains **sdev**^2, divided by **N**

(6) draw a plot for coverage using matplot as we did in the file TDistrCofidenceIntervals.R last week.

Finally, run this function with **mean** = 0 and **sdev** = 1 and with both **N** =10, **N** =100, for both **n** = 5, **n** = 100 (that is, 4 cases in total). Discuss your finding.

Submit your code as well as the resulting plots and discussions in an R Markdown file.

```
set.seed(1)
```

Funcion:

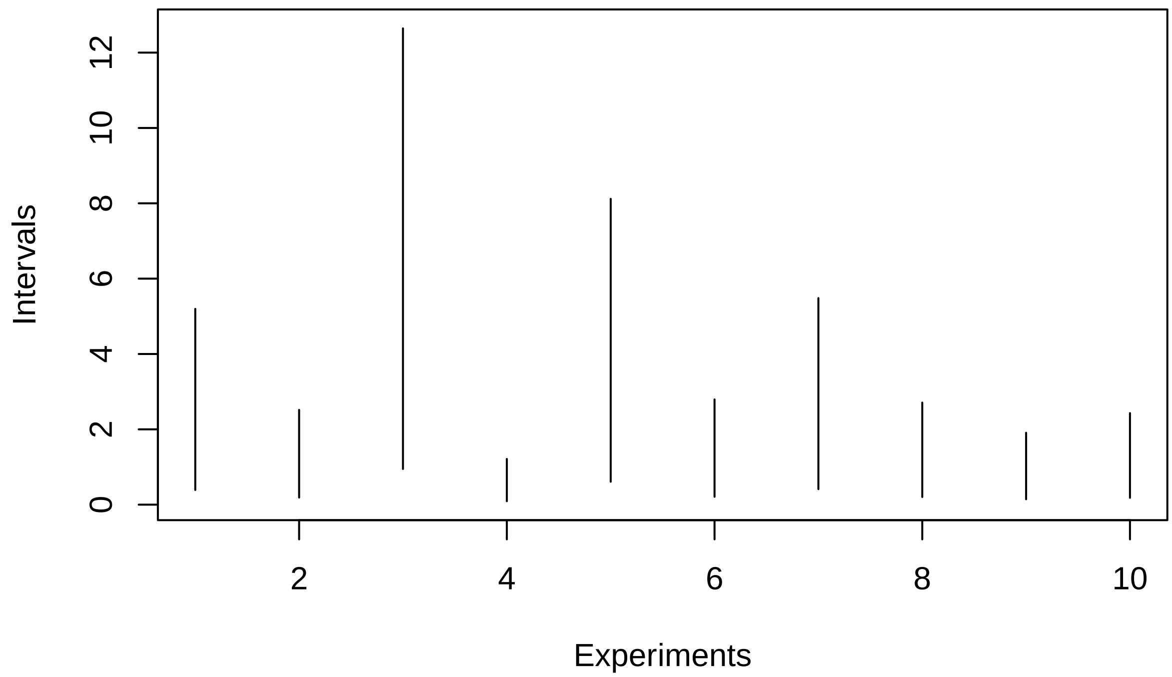
```
VarianceCI = function(N, n, mean, sdev, alpha=0.05){

  upperBound <- numeric(N)
  lowerBound <- numeric(N)

  for(i in 1:N){

    ## generating random variables
    sample = rnorm(n=n,mean=mean,sd=sdev)

    #compute confidence interval:
    upper = qchisq(alpha, n-1)
    lower = qchisq(alpha, n-1,lower.tail = FALSE)
```

```
    upperBound[i] = ((n-1)*(sd(sample))^2)/upper
    lowerBound[i] = ((n-1)*(sd(sample))^2)/lower

  }


  # Let us check which confidence interval
  # contains sdev^2 and which does not
  success <- numeric(N)
  for(i in 1:N){
      if(lowerBound[i] <= sdev^2
         & upperBound[i] >= sdev^2){
          success[i] <- 1
      }
  }
  # proportion of good confidence intervals:
  print("Proportion of good CI:")
  print(mean(success))


  # Now plot the confidence intervals,
  # but mark the bad ones:
  intervals <- rbind(lowerBound,upperBound)
  indices <- rbind(1:N,1:N)
  successColor <- ifelse(success==1,1,2)
  matplot(y=intervals,x=indices,
          type="l",lty=1,
          xlab="Experiments",ylab="Intervals",
          col=successColor)
}
```

For N = 10, n=5

```
VarianceCI(10, 5, 0, 1)
```

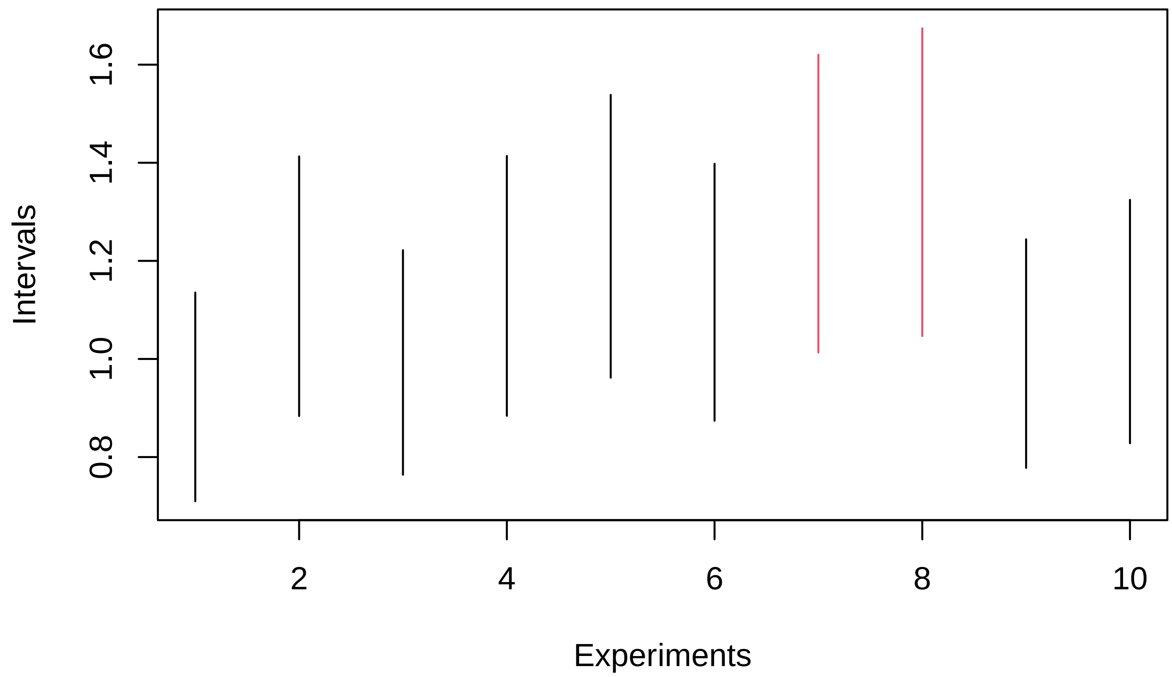
```
## [1] "Proportion of good CI:"
## [1] 1
```



For N = 10, n=100

```
VarianceCI(10, 100, 0, 1)
```

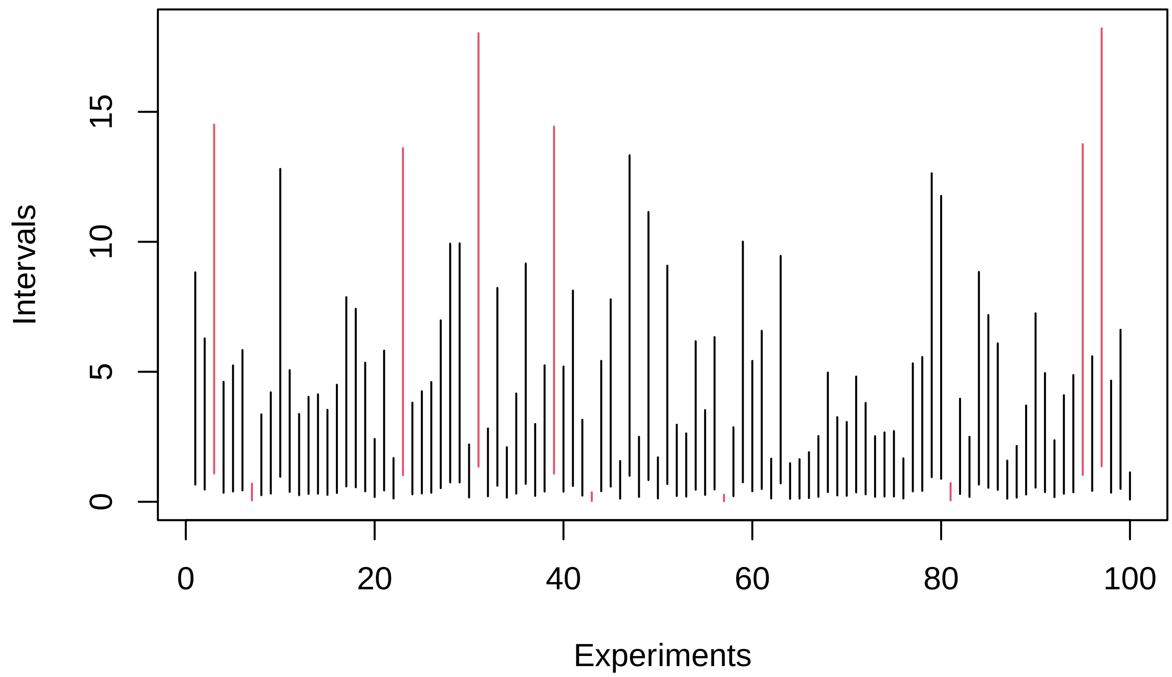
```
## [1] "Proportion of good CI:"
## [1] 0.8
```



For N = 100, n=5

```
VarianceCI(100, 5, 0, 1)
```

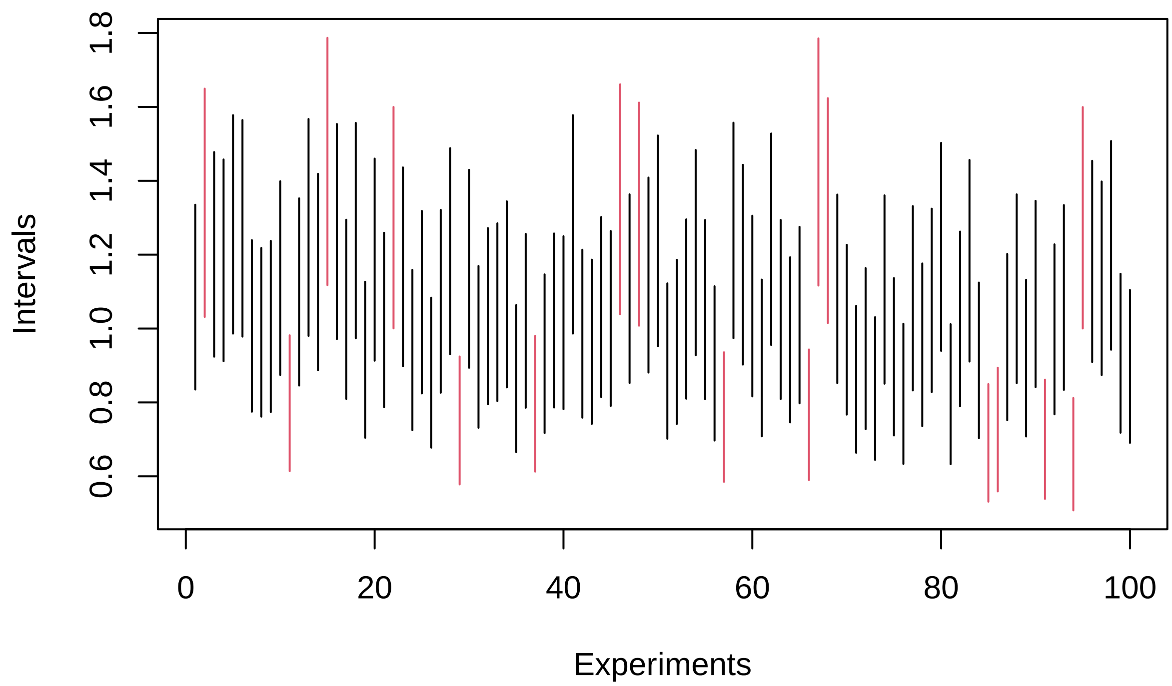
```
## [1] "Proportion of good CI:"
## [1] 0.9
```



For N = 100, n=100

```
VarianceCI(100, 100, 0, 1)
```

```
## [1] "Proportion of good CI:"
## [1] 0.83
```



We can see that when we increase the number N, indicating how many times we run the experiment, the closer the proportion of good CI gets to the real value, vase on the Alpha. The same effect can be seen when we increase the number n, since we are creating more points the sample mean ans sample stdev gets closer to the real values, and in turn the the Proportion of good CI get closer, the same as before.