|  |
| --- |
| > #IST 687  > #Daniel Fernandes  > # HW 4  >  > # This sets a seed for the random distribution generator. Using this seed function  > # the distribution is reproducible because the same seed produces the same result  > # in the random distribution generation algorithm ( or in general for any random  > # number generation algorithm)  > set.seed(2)  >  > # Setting a variable to store sample size  > sampleSize <- 30  >  > # Create a random normal distribution of 20000 numbers whose mean is 20 with a  > # standard deviation of 3  > studentPop <- rnorm(20000,mean=20,sd=3)  >  > # sample the above data set ; sample size of 30 and replace the selected value  > # in the data  > undergrads <- sample(studentPop,size=sampleSize,replace=TRUE)  >  > # Create another set of data of size 30 with mean of 25 and standard deviation of  > # 3  > grads <- rnorm(sampleSize,mean=25,sd=3)  >  > # This generates a uniform distribution ( by default from 0 to 1); if greater than  > # 0.5 then testsample is grad sample else undergrad sample  > if (runif(1)>0.5) { testSample <- grads } else { testSample <- undergrads }  >  > # calculate mean of this testsample  > meanValue <- mean(testSample)  >  > # create a list of 100 sample means from student population  > sampleMean <- replicate(100, mean(sample(studentPop, size= sampleSize, replace= TRUE)))  >  > # check the quantiles of this list of means  > quantileValues <- quantile(sampleMean, probs = c(0.025,0.975), na.rm = FALSE)  >  > quantileValues  2.5% 97.5%  19.14567 21.10799  >  > # If mean of testsample is beyond limits then its an extreme value.  > if(meanValue< quantileValues[1] || meanValue > quantileValues[2]){  + print("Sample mean is extreme")  + }else{  + print("Sample mean is not extreme")  + }  [1] "Sample mean is extreme" |
|  |
| |  | | --- | | > | |