Problem Set 1

Prateek Kumar

6 September 2018

# 1. Reading and commenting code

Creates a vector x of numeric class

x <- c(1,2,3,4,5,6,7)

This function calculates the arithmetic mean of the elements stored in vector x and assigns to variable y

y <- mean(x)

This subtracts the mean of elements of vector x from each element of the vector x and assigns it to x2, so x2 is also a vector

x2 <- x - y

Creates a vector with 100 uniform random numbers between 0.0 and 1.0 and assigns it to a1, we can change the limits by assigning values to mix= and max= inside runif() function

a1 <- runif(100)

Creates a vector with 100 normal random numbers and assigns it to a2, we can add arguments like mean and standard deviation inside rnorm() function

a2 <- rnorm(100)

The asterisk(\*) does element-wise multiplication i.e. the element of same index of vector a1 is multiplied to that of vector a2

a3 <- a1 \* a2

Here the cbind() function takes the vectors a1,a2 & a3 and combines them by rows and stores it to the variable a.tab, here a.tab is of matrix class

a.tab <- cbind(a1,a2,a3)

pairs() function creates pair-wise scatter plots of all the variables, here we have 3 variables a1,a2 and a3. This is used to view plots when we want to see for a bunch of variables at once.

pairs(a.tab)

cor() function on a matrix returns the correlation between the variables which here are a1,a2 & a3. Eg. cor(a.tab)[1,2] and cor(a.tab)[2,1] gives cor(a1,a2) as output

cor(a.tab)

# 2. Assignment of values

a <- 100 b <- a a <- 200

Answer: <- assigns the value 100 to the variable a, then b <- a means the value stored in the variable a i.e. 100 is assigned to variable b and finally a <- 200 overwrites value 100 and now 200 is assigned to variable a so finally a=200 and b=100.

a <- c(1,2,3,4,5) b <- a b[3] <- 10

Answer: The c() function here creates a vector of 5 elements of numeric class and assigns to the variable a next we assign the values stored in a to the variable b so b is also a vector and at last we modify the vector b, we replace its 3rd element with 10 so finally a is 1 2 3 4 5 and b is 1 2 10 4 5.

# 3. Computing Means of data

x1 <- runif(1000) x1.mean <- sum(x1)/length(x1) x1.gmean <- exp(mean(log(x1))) x1.hmean <- 1 / mean(1/x1) hist(x1,breaks=15, density=50,angle=135,col=“grey”,main=“x1 <- runif(1000)”,sub=paste(“AM=”, round(x1.mean,3),‘|’,“GM=”,round(x1.gmean,3),‘|’,“HM=”,round(x1.hmean,3))) abline(v=x1.mean,lwd=3, col=“blue”) text(x1.mean,30,“Arithmetic mean”,cex=0.75,font=2) abline(v=x1.hmean,lwd=3,col=“green”) text(x1.hmean,50,“Harmonic mean”,cex=0.75,font=2) abline(v=x1.gmean,lwd=3,col=“red”) text(x1.gmean,40,“Geometric mean”,cex=0.75,font=2)

x2 <- runif(1000) \* 2 x2.mean <- sum(x2)/length(x2) x2.gmean <- exp(mean(log(x2))) x2.hmean <- 1 / mean(1/x2) hist(x2,breaks=15, density=50,angle=135,col=“grey”,main=“x2 <- runif(1000) \* 2”,sub=paste(“AM=”, round(x2.mean,3),‘|’,“GM=”,round(x2.gmean,3),‘|’,“HM=”,round(x2.hmean,3))) abline(v=x2.mean,lwd=3, col=“blue”) text(x2.mean,40,“Arithmetic mean”,cex=0.75,font=2) abline(v=x2.hmean,lwd=3,col=“green”) text(x2.hmean,60,“Harmonic mean”,cex=0.75,font=2) abline(v=x2.gmean,lwd=3,col=“red”) text(x2.gmean,50,“Geometric mean”,cex=0.75,font=2)

x3 <- runif(1000) + 2 x3.mean <- sum(x3)/length(x3) x3.gmean <- exp(mean(log(x3))) x3.hmean <- 1 / mean(1/x3) hist(x3,breaks=15, density=50,angle=135,col=“grey”,main=“x3 <- runif(1000) + 2”,sub=paste(“AM=”, round(x3.mean,3),‘|’,“GM=”,round(x3.gmean,3),‘|’,“HM=”,round(x3.hmean,3))) abline(v=x3.mean,lwd=3, col=“blue”) text(x3.mean,40,“Arithmetic mean”,cex=0.75,font=2) abline(v=x3.hmean,lwd=3,col=“green”) text(x3.hmean,60,“Harmonic mean”,cex=0.75,font=2) abline(v=x3.gmean,lwd=3,col=“red”) text(x3.gmean,50,“Geometric mean”,cex=0.75,font=2)

x4 <- 5/(runif(1000)+.04) x4.mean <- sum(x4)/length(x4) x4.gmean <- exp(mean(log(x4))) x4.hmean <- 1 / mean(1/x4) hist(x4,breaks=15, density=50,angle=135,col=“grey”,main=“x4 <- 5/(runif(1000)+.04)”,sub=paste(“AM=”, round(x4.mean,3),‘|’,“GM=”,round(x4.gmean,3),‘|’,“HM=”,round(x4.hmean,3))) abline(v=x4.mean,lwd=3, col=“blue”) text(x4.mean,200,“Arithmetic mean”,cex=0.75,font=2) abline(v=x4.hmean,lwd=3,col=“green”) text(x4.hmean,400,“Harmonic mean”,cex=0.75,font=2) abline(v=x4.gmean,lwd=3,col=“red”) text(x4.gmean,300,“Geometric mean”,cex=0.75,font=2)

x5 <- exp(rnorm(1000)) x5.mean <- sum(x5)/length(x5) x5.gmean <- exp(mean(log(x5))) x5.hmean <- 1 / mean(1/x5) hist(x5,breaks=15, density=50,angle=135,col=“grey”,main=“x5 <- exp(rnorm(1000))”,sub=paste(“AM=”, round(x5.mean,3),‘|’,“GM=”,round(x5.gmean,3),‘|’,“HM=”,round(x5.hmean,3))) abline(v=x5.mean,lwd=3, col=“blue”) text(x5.mean,200,“Arithmetic mean”,cex=0.75,font=2) abline(v=x5.hmean,lwd=3,col=“green”) text(x5.hmean,400,“Harmonic mean”,cex=0.75,font=2) abline(v=x5.gmean,lwd=3,col=“red”) text(x5.gmean,300,“Geometric mean”,cex=0.75,font=2)

We see in all 5 graphs that the means change with the change in values, we will analyze each dataset one by one:  
  
1. x1 <- runif(1000)  
This function will generate 1000 uniform random numbers between 0 and 1 but none of the numbers will be equal to 0 or 1 so we get the values for both geometric and harmonic mean. Since the numbers are uniformly distributed between 0 and 1 we always get arithmetic mean approximately equal to 0.5, the geometric mean approximately equal to 0.4 and harmonic mean will be very small approx 0.01  
  
2. x2 <- runif(1000) \* 2  
This is a similar case as the above but the only difference is that we multiplied 2 to every number so here the numbers uniformly ranges between 0 and 2 so the arithmetic mean is approximately equal to 1, the geometric mean to 0.7 but since the range is wide the value of harmonic mean is greater than the previous data set approx 0.3  
  
3. x3 <- runif(1000) + 2  
In this case we are adding 2 to every number generated from runif() function. When we observe the numbers we see that the difference between the numbers is very minimal so we can assume that the numbers are approximately same. Henceforth, the arithmetic, geometric and harmonic means are almost same.  
  
4. x4 <- 5/(runif(1000)+.04)  
Here we have the division operator in the equation so the histogram tends to be like a rectangular hyperbola. When we observe the numbers stored in x4 we see that when the characteristic value is small the population is high and as the characteristic value increases the population gets scarse so we get the arithmetic, geometric and harmonic means where the population of numbers is dense.  
  
5. x5 <- exp(rnorm(1000))  
Here we have the rnorm() function so the numbers we get are between -4 to 4. Further the rnorm() function is enclosed inside the exp() function. So the numbers stored is x5 have a very less variation when the characteristic value of number generate by rnorm() is less and grows exponentially when the characteristic value increases. Here as well the histogram tends to be like a rectangular hyperbola but the population is much dense when the characteristic value is small and gets very scarse and the characteristic value increases so we get the arithmetic, geometric and harmonic means similar to that of above but more shifted towards the y-axis(frequency).  
  
Now, each of the 3 means are based upon the values and are appropriate in different contexts:  
  
1. Arithmetic mean is appropriate when there is not much fluctations between the different samples but if we have an outlier in the sample, arithmetic mean is not an appropriate measure because the mean moves towards the outlier.  
  
2. Geometric mean is appropriate when the values change exponentially but it cannot be used when any of the value is 0 or negative because in that case the geometric mean will be 0.  
  
3. Harmonic mean is appropriate when the reciprocal of the values are more useful i.e. the sample contains fractions and/or extreme values but likewise in geometric mean it cannot be used when any of the value is 0 because in that case it will calculate 1/0 which is Inf.  
  
So if all the values in a data set is same, then all the 3 means will be identical but as the variablity in the data increases, the difference among them increases but we can observe from the histogram that AM >= GM >= HM for positive samples. So usually the arithmetic mean is used, occasionally the geometric mean, and very rarely the harmonic mean.

# 4. Normalizing data

x1 <- 1:20 par(mfrow=c(1,2)) plot(x1, main=“before”,sub=paste(“mean=”, round(mean(x1),4),‘|’,“sd=”,round(sd(x1),4)),cex=.5) x1.norm <- (x1-mean(x1))/sd(x1) plot(x1.norm, main=“after”,sub=paste(“mean=”,round(mean(x1.norm),4),‘|’,“sd=”,round(sd(x1.norm),4)),cex=.5) round(mean(x1.norm),3) round(sd(x1.norm),3)

x2 <- runif(100) + rnorm(100)\*3 par(mfrow=c(1,2)) plot(x2, main=“before”,sub=paste(“mean=”, round(mean(x2),4),‘|’,“sd=”,round(sd(x2),4)),cex=.5) x2.norm <- (x2-mean(x2))/sd(x2) plot(x2.norm, main=“after”,sub=paste(“mean=”,round(mean(x2.norm),4),‘|’,“sd=”,round(sd(x2.norm),4)),cex=.5) round(mean(x2.norm),3) round(sd(x2.norm),3)

x3 <- seq(1,100,3) par(mfrow=c(1,2)) plot(x3, main=“before”,sub=paste(“mean=”, round(mean(x3),4),‘|’,“sd=”,round(sd(x3),4)),cex=.5) x3.norm <- (x3-mean(x3))/sd(x3) plot(x3.norm, main=“after”,sub=paste(“mean=”,round(mean(x3.norm),4),‘|’,“sd=”,round(sd(x3.norm),4)),cex=.5) round(mean(x3.norm),3) round(sd(x3.norm),3)

x4 <- c(runif(10),300) par(mfrow=c(1,2)) plot(x4, main=“before”,sub=paste(“mean=”, round(mean(x4),4),‘|’,“sd=”,round(sd(x4),4)),cex=.5) x4.norm <- (x4-mean(x4))/sd(x4) plot(x4.norm, main=“after”,sub=paste(“mean=”,round(mean(x4.norm),4),‘|’,“sd=”,round(sd(x4.norm),4)),cex=.5) round(mean(x4.norm),3) round(sd(x4.norm),3)

x5 <- 1/(runif(50)\*3) par(mfrow=c(1,2)) plot(x5, main=“before”,sub=paste(“mean=”, round(mean(x5),4),‘|’,“sd=”,round(sd(x5),4)),cex=.5) x5.norm <- (x5-mean(x5))/sd(x5) plot(x5.norm, main=“after”,sub=paste(“mean=”,round(mean(x5.norm),4),‘|’,“sd=”,round(sd(x5.norm),4)),cex=.5) round(mean(x5.norm),3) round(sd(x5.norm),3)