DS\_Assignment\_1

Srijan Kundu

Nov 13, 2022

library("ie2misc")

## Warning: package 'ie2misc' was built under R version 4.2.2

library("stats")

## Question:

Carry out for the following distributions:

1. ,
2. ,
3. ,
4. ,
5. ,
6. ,
7. ,
8. ,
9. .

### Steps:

1. Draw a random sample of size from .
2. Compute the following quantities:
   1. the ratio ,
   2. the ratio ,
   3. the ratio ,
   4. of observations in the interval ,
   5. of observations in the interval ,
   6. of observations in the interval .
3. Repeat for times. This will generate values for each sets of quantities computed in .
4. For each of the quantities obtained in , repeat the minimum and maximum value in the form of an interval.
5. Carry out for .

## Answer:

n = c(10, 100, 1000, 5000)  
R = 1000

### Data collection and preparation for manipulation:

data1 = array(NA, c(9, 10, 1000))  
data2 = array(NA, c(9, 100, 1000))  
data3 = array(NA, c(9, 1000, 1000))  
data4 = array(NA, c(9, 5000, 1000))  
  
data1[1, , ] = replicate(R, rnorm(n[1], 0, 1))  
data1[2, , ] = replicate(R, rcauchy(n[1], 0, 1))  
data1[3, , ] = replicate(R, rlnorm(n[1], 0, 1))  
data1[4, , ] = replicate(R, rgamma(n[1], shape = 1, scale = 0.5))  
data1[5, , ] = replicate(R, rbeta(n[1], 3, 3))  
data1[6, , ] = replicate(R, rbeta(n[1], 3, 1))  
data1[7, , ] = replicate(R, rbeta(n[1], 0.5, 3))  
data1[8, , ] = replicate(R, rt(n[1], 5))  
data1[9, , ] = replicate(R, rt(n[1], 20))  
  
data2[1, , ] = replicate(R, rnorm(n[2], 0, 1))  
data2[2, , ] = replicate(R, rcauchy(n[2], 0, 1))  
data2[3, , ] = replicate(R, rlnorm(n[2], 0, 1))  
data2[4, , ] = replicate(R, rgamma(n[2], shape = 1, scale = 0.5))  
data2[5, , ] = replicate(R, rbeta(n[2], 3, 3))  
data2[6, , ] = replicate(R, rbeta(n[2], 3, 1))  
data2[7, , ] = replicate(R, rbeta(n[2], 0.5, 3))  
data2[8, , ] = replicate(R, rt(n[2], 5))  
data2[9, , ] = replicate(R, rt(n[2], 20))  
  
data3[1, , ] = replicate(R, rnorm(n[3], 0, 1))  
data3[2, , ] = replicate(R, rcauchy(n[3], 0, 1))  
data3[3, , ] = replicate(R, rlnorm(n[3], 0, 1))  
data3[4, , ] = replicate(R, rgamma(n[3], shape = 1, scale = 0.5))  
data3[5, , ] = replicate(R, rbeta(n[3], 3, 3))  
data3[6, , ] = replicate(R, rbeta(n[3], 3, 1))  
data3[7, , ] = replicate(R, rbeta(n[3], 0.5, 3))  
data3[8, , ] = replicate(R, rt(n[3], 5))  
data3[9, , ] = replicate(R, rt(n[3], 20))  
  
data4[1, , ] = replicate(R, rnorm(n[4], 0, 1))  
data4[2, , ] = replicate(R, rcauchy(n[4], 0, 1))  
data4[3, , ] = replicate(R, rlnorm(n[4], 0, 1))  
data4[4, , ] = replicate(R, rgamma(n[4], shape = 1, scale = 0.5))  
data4[5, , ] = replicate(R, rbeta(n[4], 3, 3))  
data4[6, , ] = replicate(R, rbeta(n[4], 3, 1))  
data4[7, , ] = replicate(R, rbeta(n[4], 0.5, 3))  
data4[8, , ] = replicate(R, rt(n[4], 5))  
data4[9, , ] = replicate(R, rt(n[4], 20))

### Functions initialization:

getmode <- function(v) #for mode calculation  
{  
 uniqv <- unique(v)  
 uniqv[which.max(tabulate(match(v, uniqv)))]  
}  
  
f1 = function(data) #1\_to\_calculate\_2(i)  
{  
 return(mean(data) - getmode(data)/(mean(data) - median(data)))  
}  
f2 = function(data) #2\_to\_calculate\_2(ii)  
{  
 return(madstat(data)/sd(data))  
}  
f3 = function(data) #3\_to\_calculate\_2(iii)  
{  
 return((quantile(data, 0.75)-quantile(data, 0.25))/2\*sd(data))  
}  
f4 = function(data) #4\_to\_calculate\_2(iv)  
{  
 return(100\*sum(data > (mean(data) - 6\*sd(data)) & data < (mean(data) + 6\*sd(data)) )/1000)  
}  
f5 = function(data) #5\_to\_calculate\_2(v)  
{  
 return(100\*sum(data > (mean(data) - 7.5\*madstat(data)) & data < (mean(data) + 7.5\*madstat(data)) )/1000)  
}  
f6 = function(data) #6\_to\_calculate\_2(vi)  
{  
 return(100\*sum(data > (mean(data) - 9\*(quantile(data, 0.75)-quantile(data, 0.25))/2) & data < (mean(data) + 9\*(quantile(data, 0.75)-quantile(data, 0.25))/2))/1000)  
}

*Note*: In the following tables,

* dist(i).1 represents the *minimum* value of the metric corresponding to distribution in the question.
* dist(i).2 represents the *maximum* value of the metric corresponding to distribution in the question.

### (i) For the ratio :

vect1 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect1[1, i, j] = f1(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect1[2, i, j] = f1(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect1[3, i, j] = f1(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect1[4, i, j] = f1(data4[i, , j]) #4  
 }  
}  
  
m1 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m1[i, j, ] = c(min(vect1[i, j, ]), max(vect1[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m1[, 1, ], "dist2"= m1[, 2, ], "dist3" = m1[, 3, ], "dist4" = m1[, 4, ], "dist5" = m1[, 5, ], "dist6" = m1[, 6, ], "dist7" = m1[, 7, ], "dist8" = m1[, 8, ], "dist9" = m1[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2  
## 1 10 -1540.705 3535.725 -5347.200 727.339 -1475.99330 394.751829  
## 2 100 -13723.739 4694.503 -6393.277 6691.010 -26.41766 2.413123  
## 3 1000 -420988.910 11137.539 -2993.422 3430.586 -19.11159 1.631065  
## 4 5000 -29339.224 135556.999 -35169.741 1993.775 -64.29963 1.598910  
## dist4.1 dist4.2 dist5.1 dist5.2 dist6.1 dist6.2  
## 1 -829.87404 3129.7533994 -5107.577 6634.362 -4020.649583 5327.07796  
## 2 -24.49270 0.5530995 -105258.534 120999.400 -234.325958 1777.56243  
## 3 -21.97092 0.5055645 -31430.004 244345.221 2.717495 32.81278  
## 4 -24.99898 0.5008713 -264467.997 611161.606 2.695045 27.40643  
## dist7.1 dist7.2 dist8.1 dist8.2 dist9.1 dist9.2  
## 1 -9774.78113 1651.2876365 -795.3425 419.9058 -700.4757 518.8672  
## 2 -16.79420 0.1745233 -1145.5503 1921.7526 -10554.6594 5801.4790  
## 3 -13.22835 0.1488011 -47149.2887 129594.1052 -128243.4219 2908.7442  
## 4 -14.89356 0.1467838 -130841.8198 30345.5283 -22503.1041 644769.0584

### (ii) For the ratio :

vect2 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect2[1, i, j] = f2(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect2[2, i, j] = f2(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect2[3, i, j] = f2(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect2[4, i, j] = f2(data4[i, , j]) #4  
 }  
}  
  
m2 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m2[i, j, ] = c(min(vect2[i, j, ]), max(vect2[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m2[, 1, ], "dist2"= m2[, 2, ], "dist3" = m2[, 3, ], "dist4" = m2[, 4, ], "dist5" = m2[, 5, ], "dist6" = m2[, 6, ], "dist7" = m2[, 7, ], "dist8" = m2[, 8, ], "dist9" = m2[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2  
## 1 10 0.5772385 0.9218626 0.46123608 0.9263717 0.5462530 0.9320463  
## 2 100 0.7257836 0.8581465 0.17274228 0.6482465 0.2856455 0.8018505  
## 3 1000 0.7779558 0.8184771 0.05907344 0.3598001 0.2211136 0.6986292  
## 4 5000 0.7872979 0.8065567 0.02735416 0.2372742 0.3568099 0.6492790  
## dist4.1 dist4.2 dist5.1 dist5.2 dist6.1 dist6.2 dist7.1  
## 1 0.5496848 0.9361450 0.5840007 0.9233938 0.5679915 0.9201680 0.5550194  
## 2 0.5355697 0.8376981 0.7626300 0.8786842 0.7307391 0.8821765 0.6737813  
## 3 0.6857091 0.7787450 0.8047555 0.8426339 0.7956165 0.8397586 0.7393445  
## 4 0.7135257 0.7546062 0.8168744 0.8334223 0.8073871 0.8268573 0.7606396  
## dist7.2 dist8.1 dist8.2 dist9.1 dist9.2  
## 1 0.9277248 0.5462938 0.9176030 0.5642206 0.9091491  
## 2 0.8590082 0.5322577 0.8349360 0.7149167 0.8494114  
## 3 0.8031689 0.5625626 0.7742301 0.7574316 0.8073311  
## 4 0.7862726 0.6768609 0.7546537 0.7755984 0.7965822

### (iii) For the ratio :

vect3 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect3[1, i, j] = f3(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect3[2, i, j] = f3(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect3[3, i, j] = f3(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect3[4, i, j] = f3(data4[i, , j]) #4  
 }  
}  
  
m3 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m3[i, j, ] = c(min(vect3[i, j, ]), max(vect3[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m3[, 1, ], "dist2"= m3[, 2, ], "dist3" = m3[, 3, ], "dist4" = m3[, 4, ], "dist5" = m3[, 5, ], "dist6" = m3[, 6, ], "dist7" = m3[, 7, ], "dist8" = m3[, 8, ], "dist9" = m3[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2  
## 1 10 0.0548736 2.7171819 0.07623843 16224.71 0.0274034 106.264090  
## 2 100 0.3466538 1.1883821 1.62831783 12104.57 0.4560463 5.323702  
## 3 1000 0.5778828 0.7928144 6.85543056 47250.76 1.0693944 5.242154  
## 4 5000 0.6254718 0.7173617 15.73197232 166247.61 1.2839281 2.794691  
## dist4.1 dist4.2 dist5.1 dist5.2 dist6.1 dist6.2  
## 1 0.002510714 0.8628781 0.0009165346 0.08698983 0.001128547 0.09525659  
## 2 0.068812084 0.2948958 0.0132689866 0.04184784 0.014084984 0.04664327  
## 3 0.107495236 0.1797680 0.0218708662 0.03164104 0.022472208 0.03284555  
## 4 0.123623894 0.1542089 0.0244113940 0.02917174 0.024665856 0.02925580  
## dist7.1 dist7.2 dist8.1 dist8.2 dist9.1 dist9.2  
## 1 0.0005727793 0.07844678 0.06985666 4.209505 0.05824684 3.1084903  
## 2 0.0071158838 0.03138693 0.47616359 1.943318 0.36265204 1.2142656  
## 3 0.0118464076 0.02020573 0.75340666 1.281810 0.61116260 0.8749784  
## 4 0.0143016644 0.01791907 0.85145297 1.036672 0.66254780 0.7875890

### (iv) For of observations in the interval :

vect4 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect4[1, i, j] = f4(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect4[2, i, j] = f4(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect4[3, i, j] = f4(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect4[4, i, j] = f4(data4[i, , j]) #4  
 }  
}  
  
m4 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m4[i, j, ] = c(min(vect4[i, j, ]), max(vect4[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m4[, 1, ], "dist2"= m4[, 2, ], "dist3" = m4[, 3, ], "dist4" = m4[, 4, ], "dist5" = m4[, 5, ], "dist6" = m4[, 6, ], "dist7" = m4[, 7, ], "dist8" = m4[, 8, ], "dist9" = m4[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2 dist4.1 dist4.2  
## 1 10 1 1 1.0 1.0 1.0 1.0 1.0 1  
## 2 100 10 10 9.8 10.0 9.8 10.0 9.9 10  
## 3 1000 100 100 99.0 99.9 99.1 100.0 99.7 100  
## 4 5000 500 500 497.4 499.9 497.0 499.3 498.9 500  
## dist5.1 dist5.2 dist6.1 dist6.2 dist7.1 dist7.2 dist8.1 dist8.2 dist9.1  
## 1 1 1 1 1 1 1 1.0 1 1.0  
## 2 10 10 10 10 10 10 9.9 10 10.0  
## 3 100 100 100 100 100 100 99.6 100 99.9  
## 4 500 500 500 500 500 500 499.2 500 499.9  
## dist9.2  
## 1 1  
## 2 10  
## 3 100  
## 4 500

### (v) For of observations in the interval :

vect5 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect5[1, i, j] = f5(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect5[2, i, j] = f5(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect5[3, i, j] = f5(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect5[4, i, j] = f5(data4[i, , j]) #4  
 }  
}  
  
m5 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m5[i, j, ] = c(min(vect5[i, j, ]), max(vect5[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m5[, 1, ], "dist2"= m5[, 2, ], "dist3" = m5[, 3, ], "dist4" = m5[, 4, ], "dist5" = m5[, 5, ], "dist6" = m5[, 6, ], "dist7" = m5[, 7, ], "dist8" = m5[, 8, ], "dist9" = m5[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2 dist4.1 dist4.2  
## 1 10 1 1 1.0 1.0 1.0 1.0 1.0 1.0  
## 2 100 10 10 9.4 10.0 9.7 10.0 9.8 10.0  
## 3 1000 100 100 97.0 99.9 98.5 99.8 99.4 100.0  
## 4 5000 500 500 489.2 499.9 494.4 497.6 498.4 499.9  
## dist5.1 dist5.2 dist6.1 dist6.2 dist7.1 dist7.2 dist8.1 dist8.2 dist9.1  
## 1 1 1 1 1 1.0 1 1.0 1 1.0  
## 2 10 10 10 10 9.9 10 9.9 10 10.0  
## 3 100 100 100 100 100.0 100 99.6 100 99.9  
## 4 500 500 500 500 500.0 500 498.9 500 499.9  
## dist9.2  
## 1 1  
## 2 10  
## 3 100  
## 4 500

### (vi) For of observations in the interval :

vect6 = array(NA, c(4, 9, 1000))  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect6[1, i, j] = f6(data1[i, , j]) #1  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect6[2, i, j] = f6(data2[i, , j]) #2  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect6[3, i, j] = f6(data3[i, , j]) #3  
 }  
}  
for (i in 1:9)  
{  
 for (j in 1:1000)  
 {  
 vect6[4, i, j] = f6(data4[i, , j]) #4  
 }  
}  
  
m6 = array(NA, c(4, 9, 2))  
for (i in 1:4)  
{  
 for (j in 1:9)  
 {  
 m6[i, j, ] = c(min(vect6[i, j, ]), max(vect6[i, j, ]))  
 }  
}  
  
data.frame("Sample\_size" = n, "dist1" = m6[, 1, ], "dist2"= m6[, 2, ], "dist3" = m6[, 3, ], "dist4" = m6[, 4, ], "dist5" = m6[, 5, ], "dist6" = m6[, 6, ], "dist7" = m6[, 7, ], "dist8" = m6[, 8, ], "dist9" = m6[, 9, ])

## Sample\_size dist1.1 dist1.2 dist2.1 dist2.2 dist3.1 dist3.2 dist4.1 dist4.2  
## 1 10 0.7 1 0 1.0 0.0 1.0 0.6 1.0  
## 2 100 10.0 10 0 9.9 9.4 10.0 9.5 10.0  
## 3 1000 100.0 100 0 95.3 96.8 99.3 99.1 100.0  
## 4 5000 500.0 500 0 469.7 488.2 493.8 497.4 499.7  
## dist5.1 dist5.2 dist6.1 dist6.2 dist7.1 dist7.2 dist8.1 dist8.2 dist9.1  
## 1 0.8 1 0.8 1 0.8 1 0.7 1 0.8  
## 2 10.0 10 10.0 10 9.7 10 9.6 10 9.9  
## 3 100.0 100 100.0 100 99.8 100 99.4 100 99.9  
## 4 500.0 500 500.0 500 500.0 500 498.4 500 499.9  
## dist9.2  
## 1 1  
## 2 10  
## 3 100  
## 4 500