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 *steps* $1-5$ for the following distributions:

1. $N(0,1)$,
2. $Cauchy(0,1)$,
3. $LogNormal(0,1)$,
4. $Gamma(1, \frac{1}{2})$,
5. $Beta(3,3)$,
6. $Beta(3,1)$,
7. $Beta(\frac{1}{2},3)$,
8. t_5,
9. t_{20}.

Steps:

1. Draw a random sample of size n from $f_\theta(x)$.
2. Compute the following quantities:
   i) the ratio $\dfrac{\bar{x} - mode(x)}{\bar{x} - median(x)}$,
   ii) the ratio $\dfrac{MD_{\bar{x}}}{sd(x)}$,
   iii) the ratio $\dfrac{SIQR(x)}{SD(x)}$,
   iv) % of observations in the interval $\bar{x} \pm 6 \times SD$,
   v) % of observations in the interval $\bar{x} \pm 7.5 \times MD_{\bar{x}}$,
   vi) % of observations in the interval $\bar{x} \pm 9 \times QD$.
3. Repeat *steps* $1-2$ for $R = 1000$ times. This will generate R values for each sets of quantities computed in *step* 2.
4. For each of the quantities obtained in *step* 2, repeat the minimum and maximum value in the form of an interval.
5. Carry out *steps* $1-4$ for $n = 10, 100, 1000, 5000$.

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)) )/1
}
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(
}
```

Note: In the following 6 tables,

- `dist(i).1` represents the *minimum* value of the metric corresponding to i^{th} distribution in the question.
- `dist(i).2` represents the *maximum* value of the metric corresponding to i^{th} distribution in the question.

(i) For the ratio $\dfrac{\bar{x} - mode(x)}{\bar{x} - median(x)}$:

```
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
  }
}
```

```r
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
```

```
##   Sample_size     dist1.1    dist1.2    dist2.1    dist2.2    dist3.1
## 1          10  -36867.901   875.9189  -402.2271 10838.8264 -928.69092
## 2         100  -50358.148  2196.9922  -598.0064   652.6609  -36.84708
## 3        1000   -6411.331  3900.5909 -1909.5923  6767.7628  -29.02655
## 4        5000 -548283.081 52431.1495 -1822.8683 73351.7636  -62.27972
##       dist3.2     dist4.1      dist4.2    dist5.1   dist5.2       dist6.1
## 1 1565.772827 -4458.95877 1591.2080163 -18912.956  9284.389 -20963.734700
## 2    2.194479   -22.76844    0.5907569  -8548.369 41427.296   -507.256600
## 3    1.648077   -20.64123    0.4973963 -21246.678 12549.203      3.462329
## 4    1.614179   -18.27181    0.4914966 -50422.563 87852.445      3.102681
##      dist6.2    dist7.1     dist7.2    dist8.1     dist8.2    dist9.1   dist9.2
## 1 6705.61538 -328.65508 442.4947230  -1507.198    1581.062  -811.1358  2167.451
## 2 1318.39038  -19.95873   0.1752239 -58264.422 1104231.300 -9840.4335 10378.396
## 3   32.81746  -13.57395   0.1538974  -2080.548   13782.726 -9225.6776  8644.232
## 4   26.01908  -13.51197   0.1471128 -36245.713   81293.601 -7292.0491 22917.578
```

(ii) For the ratio $\frac{MD_{\bar{x}}}{sd(x)}$:

```r
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
```

```
##   Sample_size   dist1.1   dist1.2    dist2.1   dist2.2   dist3.1   dist3.2
## 1          10 0.5281702 0.9047796 0.44738741 0.9003238 0.5543003 0.9097297
## 2         100 0.7146919 0.8513728 0.18004642 0.6869063 0.2798234 0.8081842
## 3        1000 0.7744126 0.8220669 0.05803324 0.3964885 0.3255829 0.6967145
## 4        5000 0.7869327 0.8067809 0.02685114 0.2255202 0.2698857 0.6452336
##     dist4.1   dist4.2   dist5.1   dist5.2   dist6.1   dist6.2   dist7.1
## 1 0.5522470 0.9173873 0.5877821 0.9257850 0.5798259 0.9251865 0.5527727
## 2 0.5191251 0.8395129 0.7733021 0.8727197 0.7333817 0.8805986 0.6753939
## 3 0.6836250 0.7771628 0.8073219 0.8436115 0.7929875 0.8389846 0.7437175
```

```
## 4 0.7065123 0.7535939 0.8173214 0.8354601 0.8067448 0.8266055 0.7593877
##     dist7.2   dist8.1   dist8.2   dist9.1   dist9.2
## 1 0.9184594 0.5576075 0.9220307 0.5418802 0.9095923
## 2 0.8718920 0.4182187 0.8362344 0.7097068 0.8637636
## 3 0.8018861 0.6242566 0.7742858 0.7602277 0.8116578
## 4 0.7862195 0.3593133 0.7605110 0.7757231 0.7982137
```

(iii) For the ratio $\frac{SIQR(x)}{SD(x)}$:

```
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
```

```
##   Sample_size    dist1.1   dist1.2    dist2.1    dist2.2    dist3.1   dist3.2
```

```
## 1          10 0.06899302 2.3153988  0.1601788   4010.070 0.03432818 21.047269
## 2         100 0.30865898 1.0223280  1.8387525   7450.869 0.49259180  4.730037
## 3        1000 0.56865339 0.8030383  7.7296160 229233.922 1.04378138  3.151701
## 4        5000 0.62302313 0.7244501 16.9955670 102607.415 1.29236621  3.605994
##       dist4.1   dist4.2     dist5.1    dist5.2    dist6.1    dist6.2
## 1 0.008464797 0.7020165 0.001108549 0.07794572 0.00194304 0.08182013
## 2 0.053753469 0.2649154 0.015631843 0.04459130 0.01439736 0.04378081
## 3 0.109613224 0.1827355 0.022105275 0.03085259 0.02138329 0.03196576
## 4 0.121615395 0.1548629 0.024636104 0.02819315 0.02508100 0.02893002
##        dist7.1    dist7.2    dist8.1  dist8.2    dist9.1   dist9.2
## 1 0.0001721102 0.07389744 0.04121523 4.626171 0.05380949 2.6685709
## 2 0.0061200237 0.03448538 0.49366351 1.714296 0.36445112 1.2227889
## 3 0.0123500070 0.01960311 0.77512530 1.160958 0.59905812 0.8493657
## 4 0.0145543343 0.01765526 0.84950243 1.951231 0.67102974 0.7805678
```

(iv) For % of observations in the interval $\bar{x} \pm 6 \times SD$:

```
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
```

```
##   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    99.2   100.0    99.6     100
## 4        5000     500     500   497.5   499.9   497.1   499.5   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   499.0     500   499.9
##   dist9.2
## 1       1
## 2      10
## 3     100
## 4     500
```

(v) For $\%$ of observations in the interval $\bar{x} \pm 7.5 \times MD_{\bar{x}}$:

```
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
```

```
##   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     9.5    10.0     9.7    10.0     9.8      10
## 3        1000     100     100    97.1    99.9    98.6    99.9    99.4     100
## 4        5000     500     500   488.8   499.9   494.3   497.6   498.5     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    99.5     100    99.9
## 4     500     500     500     500     500     500   498.6     500   499.9
##   dist9.2
## 1       1
## 2      10
## 3     100
## 4     500
```

(vi) For $\%$ of observations in the interval $\bar{x} \pm 9 \times QD$:

```
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)
  {
```

```r
    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
```

```
##   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     0.7     1.0
## 2         100    10.0      10       0    10.0     9.3    10.0     9.6    10.0
## 3        1000   100.0     100       0    95.0    96.6    99.4    99.1   100.0
## 4        5000   500.0     500       0   469.7   487.5   494.2   497.3   499.6
##   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     0.7
## 2    10.0      10    10.0      10     9.7      10     9.8      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   499.9     500   498.1     500   499.9
##   dist9.2
## 1       1
## 2      10
## 3     100
## 4     500
```