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(
}
```

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

```
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
  }
}
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    dist3.2
## 1          10  -2385.372   3235.396 -1260.477  1374.079 -4980.78074 461.129423
## 2         100  -4177.250 252518.940 -1838.584  1024.689   -30.27572   2.020648
## 3        1000  -9590.448  10506.006 -2616.712 91142.986   -26.74634   1.680487
## 4        5000 -59649.799  99839.764 -1350.426 19730.654   -22.34310   1.626133
##      dist4.1      dist4.2     dist5.1  dist5.2       dist6.1     dist6.2
## 1 -791.95250 3480.5689832   -3105.523 86486.96 -66044.452144 48904.43204
## 2  -34.22956    0.5981343  -42496.874 39097.97  -2091.136338   331.61820
## 3  -17.83711    0.5315052  -52680.869 64675.81      2.903958    31.81685
## 4  -20.58158    0.5026713 -192146.637 97899.79      2.991550    26.65018
##       dist7.1      dist7.2    dist8.1     dist8.2    dist9.1    dist9.2
## 1 -1186.93343 1161.3728197  -4981.359    943.6147  -1327.248   3166.241
## 2   -19.62041    0.1843997 -37542.472 100195.9177 -11348.913  53094.740
## 3   -14.05999    0.1496239 -11397.854 316128.7946 -37393.182   5770.495
## 4   -13.32226    0.1473737 -29422.936  17598.2859 -36662.627 390953.368
```

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

```
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 0.5815849 0.9380359 0.48687182 0.8974847 0.5409454 0.9066274
## 2         100 0.7023616 0.8617572 0.17210605 0.6619077 0.3494411 0.7999285
## 3        1000 0.7728000 0.8202457 0.06001619 0.3590393 0.3113774 0.6896659
## 4        5000 0.7875683 0.8078909 0.02463003 0.2377993 0.3795404 0.6490933
##     dist4.1   dist4.2   dist5.1   dist5.2   dist6.1   dist6.2   dist7.1
## 1 0.5478607 0.9191967 0.5987904 0.9290239 0.5854471 0.9155295 0.5568261
## 2 0.4782323 0.8636560 0.7635084 0.8737319 0.7458167 0.8724665 0.6711109
## 3 0.6774581 0.7729831 0.8042235 0.8467731 0.7936889 0.8383448 0.7410292
## 4 0.7124787 0.7579053 0.8185974 0.8345745 0.8070518 0.8268661 0.7579687
##     dist7.2   dist8.1   dist8.2   dist9.1   dist9.2
## 1 0.9333868 0.5535477 0.9104509 0.5607459 0.9139227
## 2 0.8507628 0.5309310 0.8365171 0.7110904 0.8437977
## 3 0.8009158 0.5805285 0.7750741 0.7584286 0.8047708
## 4 0.7854867 0.6812949 0.7554412 0.7752010 0.7964594
```

(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 0.02902172 2.2484074  0.0497183    1174.442 0.03219825 22.210182
## 2         100 0.35855148 1.1161474  1.5482953   21091.341 0.43126469  5.188449
## 3        1000 0.56325700 0.7896943  8.2080569 2858325.559 1.01198837  3.515421
## 4        5000 0.62633537 0.7246010 14.8713182   18516.742 1.27366147  2.545687
##       dist4.1   dist4.2     dist5.1    dist5.2     dist6.1    dist6.2
## 1 0.006538689 1.0798837 0.002106836 0.08755664 0.001672582 0.11484747
## 2 0.052625234 0.2710546 0.015839954 0.03980542 0.013884537 0.05172857
## 3 0.107435582 0.1750219 0.021978540 0.03218715 0.022160075 0.03207551
```

```
## 4 0.122544163 0.1539028 0.024515873 0.02846028 0.025052779 0.02905401
##        dist7.1    dist7.2    dist8.1  dist8.2    dist9.1   dist9.2
## 1 0.0001609526 0.07072990 0.05393821 5.475987 0.04641612 3.0187895
## 2 0.0061472585 0.03196685 0.46711083 1.615937 0.35973515 1.2607216
## 3 0.0128489621 0.02068306 0.75460127 1.221756 0.60188121 0.8385299
## 4 0.0142481733 0.01772583 0.84741197 1.044987 0.66407517 0.7800005
```

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

```r
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      9.9     10.0      9.9       10
## 3        1000      100      100     99.0     99.9     99.1    100.0     99.6      100
## 4        5000      500      500    496.9    499.9    497.3    499.4    498.7      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    99.7     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 $\bar{x} \pm 7.5 \times MD_{\bar{x}}$:

```
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     1.0
## 2         100      10      10     9.5    10.0     9.7    10.0     9.8    10.0
## 3        1000     100     100    97.2    99.9    98.4    99.8    99.5   100.0
## 4        5000     500     500   489.2   499.9   494.7   497.6   498.3   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       1     1.0       1     1.0
## 2      10      10      10      10      10      10     9.8      10     9.9
## 3     100     100     100     100     100     100    99.4     100    99.9
## 4     500     500     500     500     500     500   498.8     500   499.8
##   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)
  {
    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     9.9     9.3    10.0     9.5    10.0
## 3        1000   100.0     100       0    95.4    96.8    99.2    98.9   100.0
## 4        5000   500.0     500       0   469.7   488.4   494.1   497.5   499.8
##   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     0.6       1     0.7       1     0.8
## 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    99.3     100    99.9
## 4   500.0     500   500.0     500   499.9     500   498.5     500   499.9
##   dist9.2
## 1       1
## 2      10
## 3     100
## 4     500
```