STAB57A2.R

yulunwu

2021-12-09

```r
#1
X=c(11, 13, 15, 17, 19, 21, 23)
d=expand.grid(X,X,X)
X_bar=apply(d,1,mean)
# 1(a)
sample_mean = mean(c(19,21,23))
test_statistic = pnorm(abs((sample_mean-17)/(sqrt(16/3))))
p_value = 2*(1-test_statistic)
sprintf("The p-value under CLT assumption is %f",p_value)
```

```
## [1] "The p-value under CLT assumption is 0.083265"
```

```r
# 1(b)
# Code from A1
freq=table(X_bar)
freq
```

```
## X_bar
##               11 11.6666666666667 12.3333333333333               13
##                1                3                6               10
## 13.6666666666667 14.3333333333333               15 15.6666666666667
##               15               21               28               33
## 16.3333333333333               17 17.6666666666667 18.3333333333333
##               36               37               36               33
##               19 19.6666666666667 20.3333333333333               21
##               28               21               15               10
## 21.6666666666667 22.3333333333333               23
##                6                3                1
```

```r
size_X_bar=length(X_bar)
freq_val=as.vector(freq)
proportion=freq_val/size_X_bar
proportion
```

```
##  [1] 0.002915452 0.008746356 0.017492711 0.029154519 0.043731778 0.061224490
##  [7] 0.081632653 0.096209913 0.104956268 0.107871720 0.104956268 0.096209913
## [13] 0.081632653 0.061224490 0.043731778 0.029154519 0.017492711 0.008746356
## [19] 0.002915452
```

```r
p_value = sum(proportion[16:19])
sprintf("The p-value without CLT assumption is %f",p_value)
```

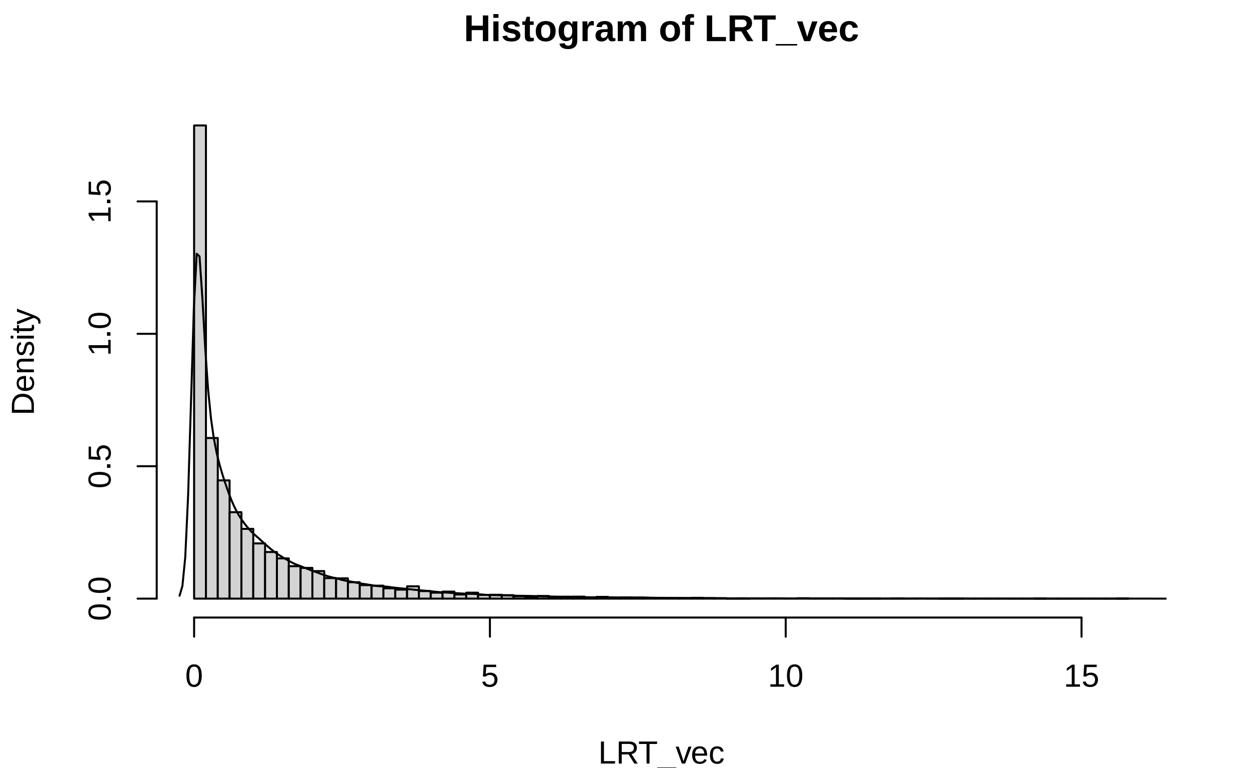
```
## [1] "The p-value without CLT assumption is 0.058309"
```

```r
# 1(c)
# In (a), I assume that X_bar follows normal distribution under CLT and
# calculated two-sided p-value.
# But in (b), I don't make this assumption, and sum up the proportion of
# X_bar as X_bar >= 21, this is one-sided p-value.
# Since the size of n is small, there is a difference between p-value/2(this
# convert two-sided p-value to right sided p-value) in (a) and p-value in(b)
# because CLT requires n ->inf, so a small n doesn't work good in CLT.
# If the size of sample(as known as n) is big enough(such as n -> inf) and
# both (a) and (b) calculate the same type of p-value, I expect that the
# two p-values from (a) and (b) are similar.

# 2(a) i
sample_2a=function(){
  s=rnorm(5, mean = 10, sd = 10)
  X_bar = mean(s)
  L_theta0 = exp((-1/200)*sum((s-10)^2))*(200*pi)^(-5/2)
  L_theta1 = exp((-1/200)*sum((s-X_bar)^2))*(200*pi)^(-5/2)
  return(-2*log(L_theta0/L_theta1))
}

# 2(a) ii
LRT_vec = replicate(10000,sample_2a())
# 2(a) iii
LRT_vec_hist = hist(LRT_vec,freq=FALSE, breaks=100)
LRT_vec_hist
```

```
## $breaks
##  [1]  0.0  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8  2.0  2.2  2.4  2.6  2.8
## [16]  3.0  3.2  3.4  3.6  3.8  4.0  4.2  4.4  4.6  4.8  5.0  5.2  5.4  5.6  5.8
## [31]  6.0  6.2  6.4  6.6  6.8  7.0  7.2  7.4  7.6  7.8  8.0  8.2  8.4  8.6  8.8
## [46]  9.0  9.2  9.4  9.6  9.8 10.0 10.2 10.4 10.6 10.8 11.0 11.2 11.4 11.6 11.8
## [61] 12.0 12.2 12.4 12.6 12.8 13.0 13.2 13.4 13.6 13.8 14.0 14.2 14.4 14.6 14.8
## [76] 15.0 15.2 15.4 15.6 15.8
##
## $counts
##  [1] 3574 1213  893  653  527  417  352  304  245  232  208  155  153  124  102
## [16]   98   78   68   93   57   45   53   31   45   28   29   26   18   15   20
## [31]   13   13   15   10   13    5    9    8    5    5    5    4    6    4    3
## [46]    1    1    2    2    2    0    3    2    2    2    1    1    1    0    2
## [61]    0    0    0    1    1    0    0    0    0    0    0    1    0    0    0
## [76]    0    0    0    1
##
## $density
##  [1] 1.7870 0.6065 0.4465 0.3265 0.2635 0.2085 0.1760 0.1520 0.1225 0.1160
## [11] 0.1040 0.0775 0.0765 0.0620 0.0510 0.0490 0.0390 0.0340 0.0465 0.0285
## [21] 0.0225 0.0265 0.0155 0.0225 0.0140 0.0145 0.0130 0.0090 0.0075 0.0100
## [31] 0.0065 0.0065 0.0075 0.0050 0.0065 0.0025 0.0045 0.0040 0.0025 0.0025
## [41] 0.0025 0.0020 0.0030 0.0020 0.0015 0.0005 0.0005 0.0010 0.0010 0.0010
## [51] 0.0000 0.0015 0.0010 0.0010 0.0010 0.0005 0.0005 0.0005 0.0000 0.0010
## [61] 0.0000 0.0000 0.0000 0.0005 0.0005 0.0000 0.0000 0.0000 0.0000 0.0000
## [71] 0.0000 0.0005 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0005
##
## $mids
```

```
##  [1]  0.1  0.3  0.5  0.7  0.9  1.1  1.3  1.5  1.7  1.9  2.1  2.3  2.5  2.7  2.9
## [16]  3.1  3.3  3.5  3.7  3.9  4.1  4.3  4.5  4.7  4.9  5.1  5.3  5.5  5.7  5.9
## [31]  6.1  6.3  6.5  6.7  6.9  7.1  7.3  7.5  7.7  7.9  8.1  8.3  8.5  8.7  8.9
## [46]  9.1  9.3  9.5  9.7  9.9 10.1 10.3 10.5 10.7 10.9 11.1 11.3 11.5 11.7 11.9
## [61] 12.1 12.3 12.5 12.7 12.9 13.1 13.3 13.5 13.7 13.9 14.1 14.3 14.5 14.7 14.9
## [76] 15.1 15.3 15.5 15.7
##
## $xname
## [1] "LRT_vec"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```r
# 2(a) iv
chisq_sample = rchisq(100000, df = 1)
lines(density(chisq_sample))
```



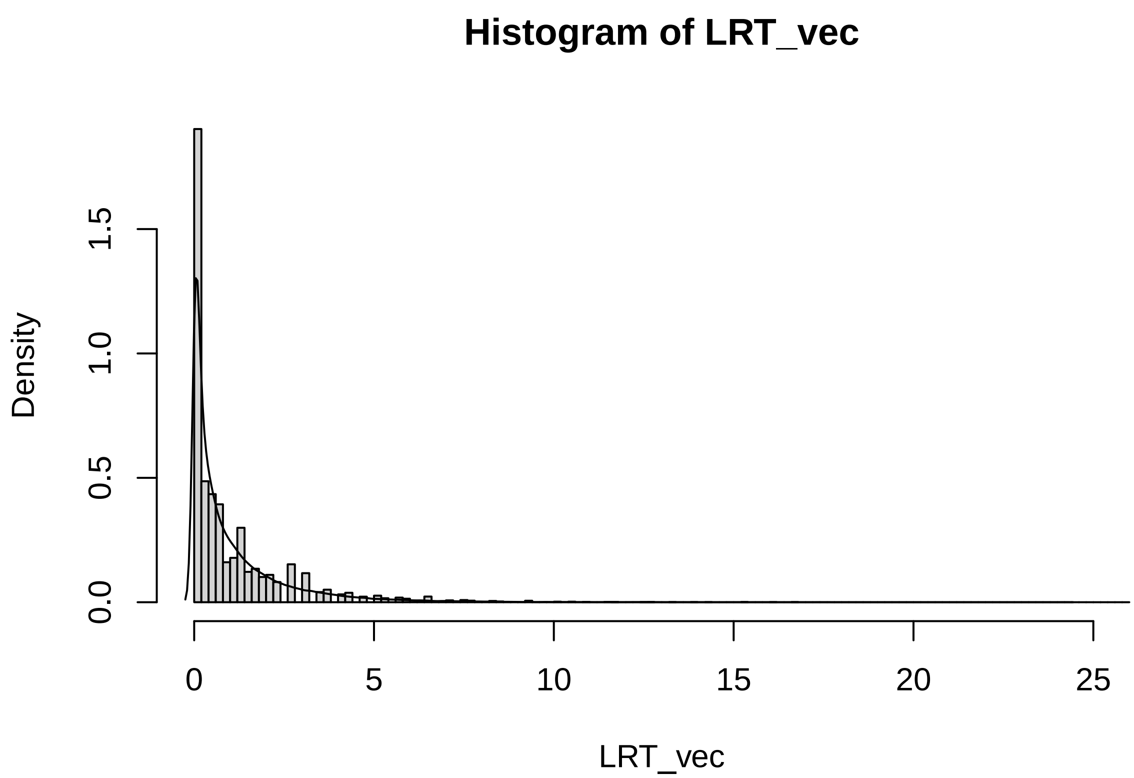

```r
# 2(b) i
sample_2b=function(){
  s=rpois(5, lambda = 10)
  X_bar = mean(s)
  L_theta0 = prod(exp(-10)*10^s/factorial(s))
  L_theta1 = prod(exp(-X_bar)*X_bar^s/factorial(s))
  return(-2*log(L_theta0/L_theta1))
}

# 2(b) ii
```

```r
LRT_vec = replicate(100000,sample_2b())
# 2(b) iii
LRT_vec_hist = hist(LRT_vec,freq=FALSE, breaks=100)
LRT_vec_hist
```

```
## $breaks
##   [1]  0.0  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8  2.0  2.2  2.4  2.6  2.8
##  [16]  3.0  3.2  3.4  3.6  3.8  4.0  4.2  4.4  4.6  4.8  5.0  5.2  5.4  5.6  5.8
##  [31]  6.0  6.2  6.4  6.6  6.8  7.0  7.2  7.4  7.6  7.8  8.0  8.2  8.4  8.6  8.8
##  [46]  9.0  9.2  9.4  9.6  9.8 10.0 10.2 10.4 10.6 10.8 11.0 11.2 11.4 11.6 11.8
##  [61] 12.0 12.2 12.4 12.6 12.8 13.0 13.2 13.4 13.6 13.8 14.0 14.2 14.4 14.6 14.8
##  [76] 15.0 15.2 15.4 15.6 15.8 16.0 16.2 16.4 16.6 16.8 17.0 17.2 17.4 17.6 17.8
##  [91] 18.0 18.2 18.4 18.6 18.8 19.0 19.2 19.4 19.6 19.8 20.0 20.2 20.4 20.6 20.8
## [106] 21.0 21.2 21.4 21.6 21.8 22.0 22.2 22.4 22.6 22.8 23.0 23.2 23.4 23.6 23.8
## [121] 24.0 24.2 24.4 24.6 24.8 25.0 25.2 25.4 25.6 25.8 26.0
##
## $counts
##   [1] 38045  9727  8692  7871  3213  3567  5986  2442  2692  2028  2202  1621
##  [13]     0  3048     0  2337     0   816  1009     0   630   759     0   446
##  [25]     0   528   321     0   372   281     0     0   454     0     0   147
##  [37]     0   174   125     0     0   100    58     0     0     0   118     0
##  [49]     0     0    42     0    43     0    21     0     0    21    13     0
##  [61]     0     0     8    13     0     0     5     0     0     4     0     2
##  [73]     0     0     0     0    10     0     0     0     2     0     0     4
##  [85]     0     1     0     0     0     0     0     1     0     0     0     0
##  [97]     0     0     0     0     0     0     0     0     0     0     0     0
## [109]     0     0     0     0     0     0     0     0     0     0     0     0
## [121]     0     0     0     0     0     0     0     0     0     1
##
## $density
##   [1] 1.90225 0.48635 0.43460 0.39355 0.16065 0.17835 0.29930 0.12210 0.13460
##  [10] 0.10140 0.11010 0.08105 0.00000 0.15240 0.00000 0.11685 0.00000 0.04080
##  [19] 0.05045 0.00000 0.03150 0.03795 0.00000 0.02230 0.00000 0.02640 0.01605
##  [28] 0.00000 0.01860 0.01405 0.00000 0.00000 0.02270 0.00000 0.00000 0.00735
##  [37] 0.00000 0.00870 0.00625 0.00000 0.00000 0.00500 0.00290 0.00000 0.00000
##  [46] 0.00000 0.00590 0.00000 0.00000 0.00000 0.00210 0.00000 0.00215 0.00000
##  [55] 0.00105 0.00000 0.00000 0.00105 0.00065 0.00000 0.00000 0.00000 0.00040
##  [64] 0.00065 0.00000 0.00000 0.00025 0.00000 0.00000 0.00020 0.00000 0.00010
##  [73] 0.00000 0.00000 0.00000 0.00000 0.00050 0.00000 0.00000 0.00000 0.00010
##  [82] 0.00000 0.00000 0.00020 0.00000 0.00005 0.00000 0.00000 0.00000 0.00000
##  [91] 0.00000 0.00005 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
## [100] 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
## [109] 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
## [118] 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
## [127] 0.00000 0.00000 0.00000 0.00005
##
## $mids
##   [1]  0.1  0.3  0.5  0.7  0.9  1.1  1.3  1.5  1.7  1.9  2.1  2.3  2.5  2.7  2.9
##  [16]  3.1  3.3  3.5  3.7  3.9  4.1  4.3  4.5  4.7  4.9  5.1  5.3  5.5  5.7  5.9
##  [31]  6.1  6.3  6.5  6.7  6.9  7.1  7.3  7.5  7.7  7.9  8.1  8.3  8.5  8.7  8.9
##  [46]  9.1  9.3  9.5  9.7  9.9 10.1 10.3 10.5 10.7 10.9 11.1 11.3 11.5 11.7 11.9
##  [61] 12.1 12.3 12.5 12.7 12.9 13.1 13.3 13.5 13.7 13.9 14.1 14.3 14.5 14.7 14.9
##  [76] 15.1 15.3 15.5 15.7 15.9 16.1 16.3 16.5 16.7 16.9 17.1 17.3 17.5 17.7 17.9
##  [91] 18.1 18.3 18.5 18.7 18.9 19.1 19.3 19.5 19.7 19.9 20.1 20.3 20.5 20.7 20.9
```

```
## [106] 21.1 21.3 21.5 21.7 21.9 22.1 22.3 22.5 22.7 22.9 23.1 23.3 23.5 23.7 23.9
## [121] 24.1 24.3 24.5 24.7 24.9 25.1 25.3 25.5 25.7 25.9
##
## $xname
## [1] "LRT_vec"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```r
# 2(b) iv
lines(density(chisq_sample))
```




```r
# 2(c)
# The the histograms and the density are close to each other doesn't
# matter the simple size is big or small.

# 3(a)
sample_q3 = c(11, 12, 8, 5, 11)
# Score
l_p_theta = function(t){-5+sum(sample_q3)/t}
# double differentiation
l_pp_theta = function(t){-sum(sample_q3)/t^2}
t_old = 4.5
iter = 1
dif = 1
while(dif>0.00001){
  t_new = t_old - l_p_theta(t_old)/l_pp_theta(t_old)
```

```r
  print(t_new)
  dif = abs(t_new - t_old)
  t_old = t_new
  iter = iter + 1
}
```

```
## [1] 6.845745
## [1] 8.705934
## [1] 9.348752
## [1] 9.399721
## [1] 9.4
## [1] 9.4
```

```r
sprintf("After %d iteration, the MLE of lambda is %f", iter-1, t_old)
```

```
## [1] "After 6 iteration, the MLE of lambda is 9.400000"
```

```r
# 3(b)
# -E[double differentiation]
e_pp_theta = function(t){5/t}
t_old = 4.5
iter = 1
dif = 1
while(dif>0.00001){
  t_new = t_old + l_p_theta(t_old)/e_pp_theta(t_old)
  print(t_new)
  dif = abs(t_new - t_old)
  t_old = t_new
  iter = iter + 1
}
```

```
## [1] 9.4
## [1] 9.4
```

```r
sprintf("After %d iteration, the MLE of lambda is %f", iter-1, t_old)
```

```
## [1] "After 2 iteration, the MLE of lambda is 9.400000"
```

```r
# 3(c)
# The updating equation in (a) sometimes fail to give me the solution,
# and takes more iteration to converge. For example, (a) fails to converge
# when initial guess is 25, and when initial guess is 4.5, (a) takes 6
# iteration to run while (b) only takes 2 iteration.
```