lec3q12

Jiansong Xu

January 24, 2019

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(binom)
```

3.

```
alpha <- 0.05
n.L <- 1000 #large sample
n.S <- 10 #small sample
w.L <- 0:n.L
w.S <- 0:n.S
```

Large sample

```
CI.L <- binom.confint(x = w.L, n = n.L, conf.level = 1-alpha, methods = c("asymptotic", "agresti-coull", "wilson", "exact"))
print(head(CI.L.wald <- filter(CI.L, method == "asymptotic")))#Wald CIs
```

```
##       method x    n  mean         lower       upper
## 1 asymptotic 0 1000 0.000  0.000000e+00 0.000000000
## 2 asymptotic 1 1000 0.001 -9.589838e-04 0.002958984
## 3 asymptotic 2 1000 0.002 -7.690345e-04 0.004769034
## 4 asymptotic 3 1000 0.003 -3.896612e-04 0.006389661
## 5 asymptotic 4 1000 0.004  8.791974e-05 0.007912080
## 6 asymptotic 5 1000 0.005  6.283576e-04 0.009371642
```

```
print(head(CI.L.ac <- filter(CI.L, method =="agresti-coull")))#agresti-coill CIs
```

```
##          method x    n  mean         lower       upper
## 1 agresti-coull 0 1000 0.000 -7.899577e-04 0.004616716
## 2 agresti-coull 1 1000 0.001 -4.223778e-04 0.006241483
## 3 agresti-coull 2 1000 0.002  4.724247e-05 0.007764209
## 4 agresti-coull 3 1000 0.003  5.814324e-04 0.009222366
## 5 agresti-coull 4 1000 0.004  1.161262e-03 0.010634883
## 6 agresti-coull 5 1000 0.005  1.775581e-03 0.012012910
```

```
print(head(CI.L.s <- filter(CI.L, method == "wilson")))#score CIs
```

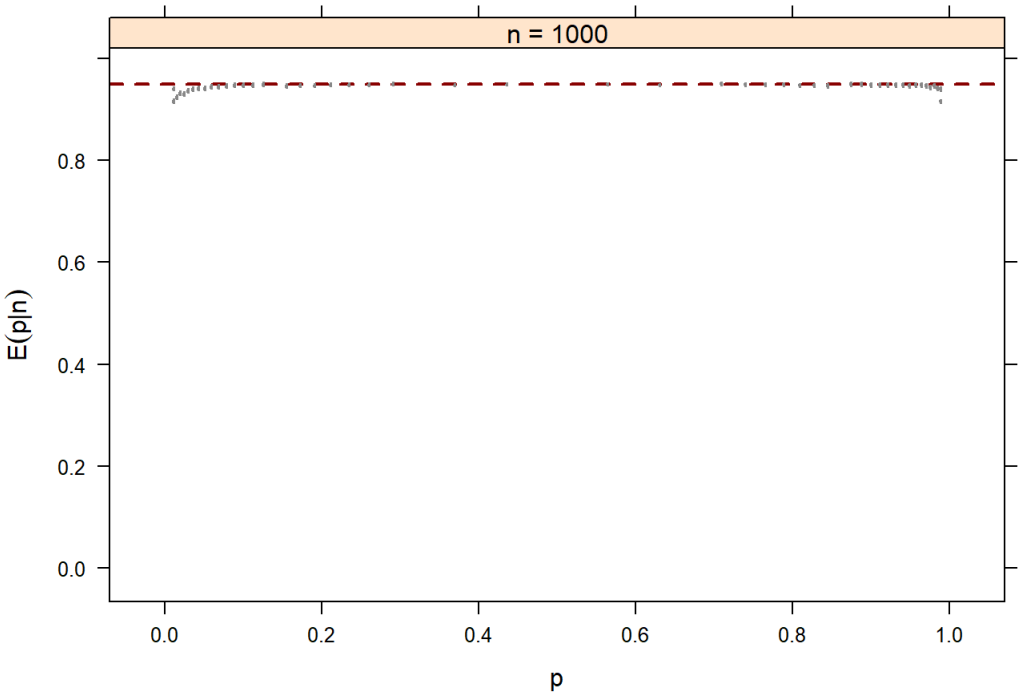
```
##   method x    n  mean        lower       upper
## 1 wilson 0 1000 0.000 2.160106e-19 0.003826758
## 2 wilson 1 1000 0.001 1.765464e-04 0.005642559
## 3 wilson 2 1000 0.002 5.486436e-04 0.007262808
## 4 wilson 3 1000 0.003 1.020784e-03 0.008783014
## 5 wilson 4 1000 0.004 1.556588e-03 0.010239556
## 6 wilson 5 1000 0.005 2.137536e-03 0.011650955
```

```
print(head(CI.L.cp <- filter(CI.L, method == "exact")))#clopper-pearson CIs
```

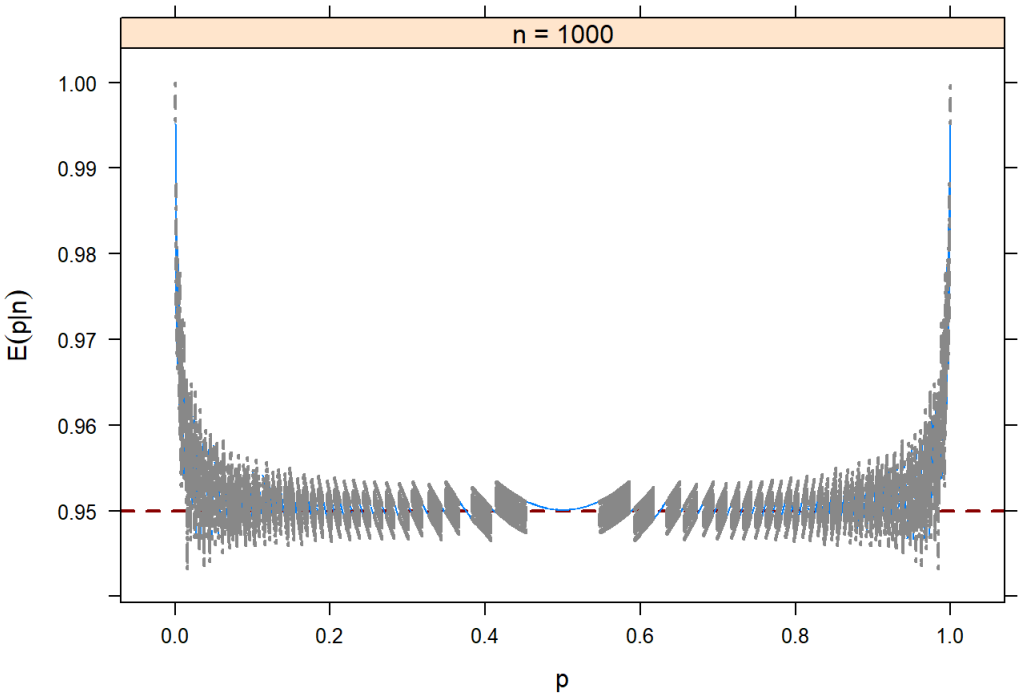
```
##   method x    n  mean        lower       upper
## 1  exact 0 1000 0.000 0.000000e+00 0.003682084
## 2  exact 1 1000 0.001 2.531749e-05 0.005558924
## 3  exact 2 1000 0.002 2.423011e-04 0.007205839
## 4  exact 3 1000 0.003 6.190999e-04 0.008742023
## 5  exact 4 1000 0.004 1.090908e-03 0.010209665
## 6  exact 5 1000 0.005 1.625420e-03 0.011629471
```

```
binom.plot(n = 1000, method = binom.asymp, np = 1000, conf.level = 0.95)
```

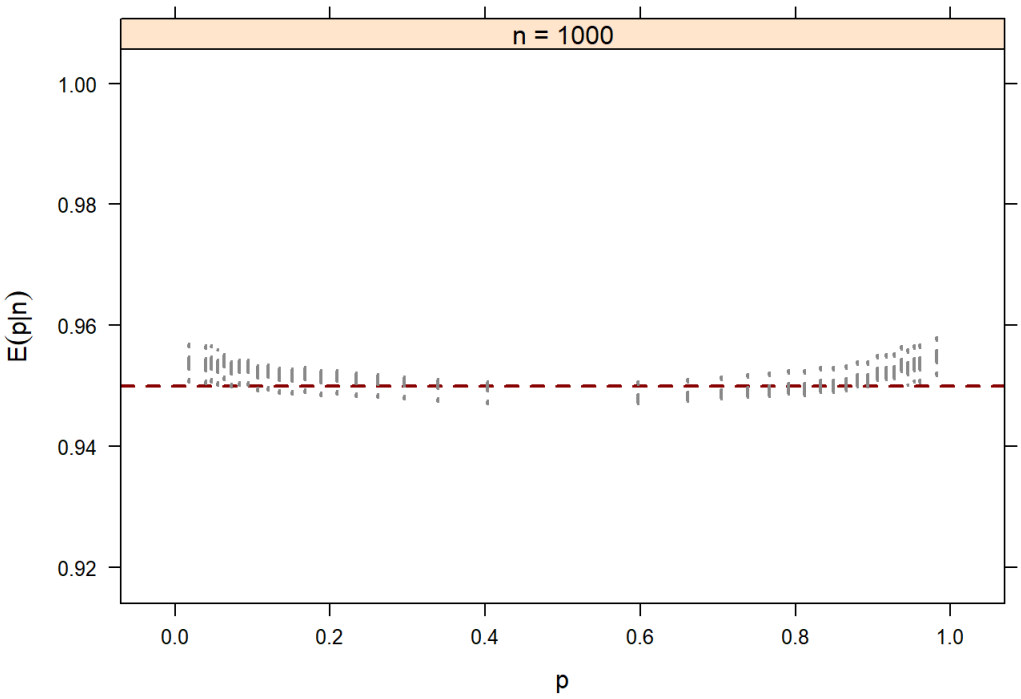
```
## Loading required package: lattice
```



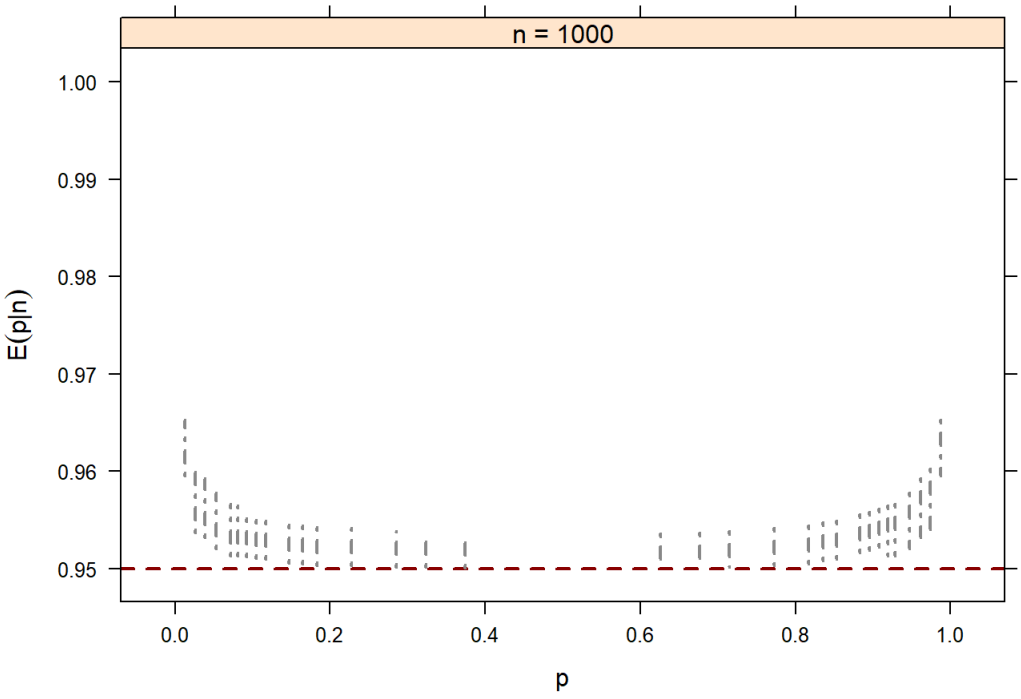

```
binom.plot(n = 1000, method = binom.agresti.coull, np = 3000, conf.level = 0.95)
```



```
binom.plot(n = 1000, method = binom.wilson, np = 1000, conf.level = 0.95)
```




```
binom.plot(n = 1000, method = binom.exact, np = 1000, conf.level = 0.95)
```




```
#Small sample
CI.S <- binom.confint(x=w.S, n=n.S, conf.level = 1-alpha, methods = c("asymptotic", "agresti-coull", "wilson
", "exact"))
print(head(filter(CI.S, method == "asymptotic")))#wald CIs
```

```
##       method x  n mean       lower     upper
## 1 asymptotic 0 10  0.0  0.00000000 0.0000000
## 2 asymptotic 1 10  0.1 -0.08593851 0.2859385
## 3 asymptotic 2 10  0.2 -0.04791801 0.4479180
## 4 asymptotic 3 10  0.3  0.01597423 0.5840258
## 5 asymptotic 4 10  0.4  0.09636369 0.7036363
## 6 asymptotic 5 10  0.5  0.19010248 0.8098975
```

```
print(head(filter(CI.S, method == "agresti-coull")))#agresti-coull CIS
```

```
##          method x  n mean        lower     upper
## 1 agresti-coull 0 10  0.0 -0.043354506 0.3208873
## 2 agresti-coull 1 10  0.1 -0.003941498 0.4259677
## 3 agresti-coull 2 10  0.2  0.045887270 0.5206324
## 4 agresti-coull 3 10  0.3  0.103338418 0.6076747
## 5 agresti-coull 4 10  0.4  0.167110626 0.6883959
## 6 agresti-coull 5 10  0.5  0.236593091 0.7634069
```

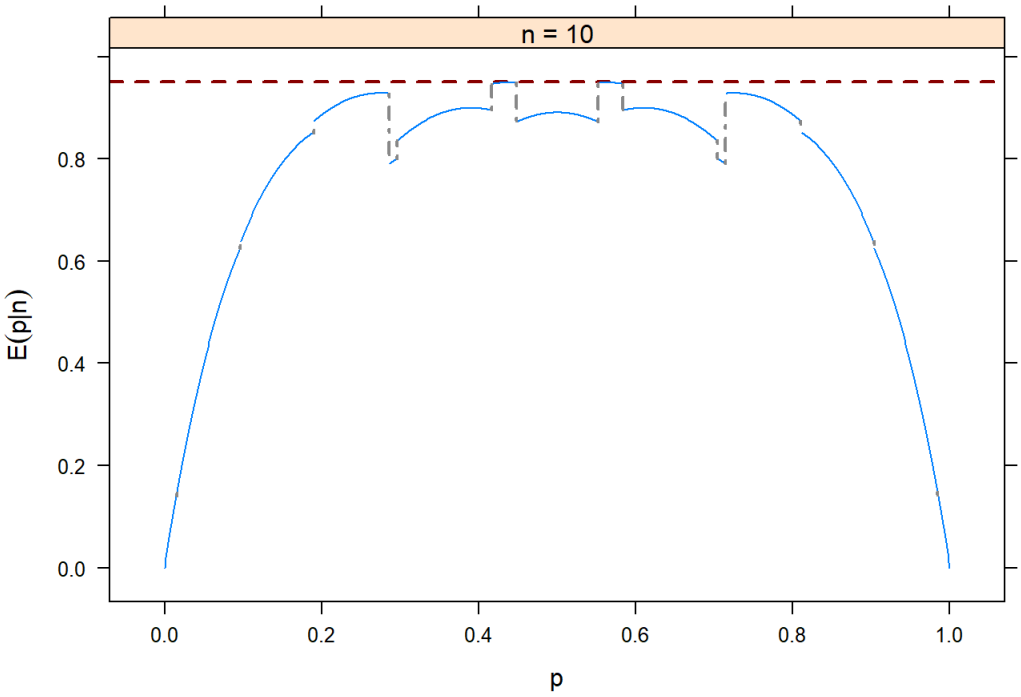
```
print(head(filter(CI.S, method == "wilson")))#score CIs
```

```
##   method x  n mean        lower     upper
## 1 wilson 0 10  0.0 2.005249e-17 0.2775328
## 2 wilson 1 10  0.1 1.787621e-02 0.4041500
## 3 wilson 2 10  0.2 5.668215e-02 0.5098375
## 4 wilson 3 10  0.3 1.077913e-01 0.6032219
## 5 wilson 4 10  0.4 1.681803e-01 0.6873262
## 6 wilson 5 10  0.5 2.365931e-01 0.7634069
```

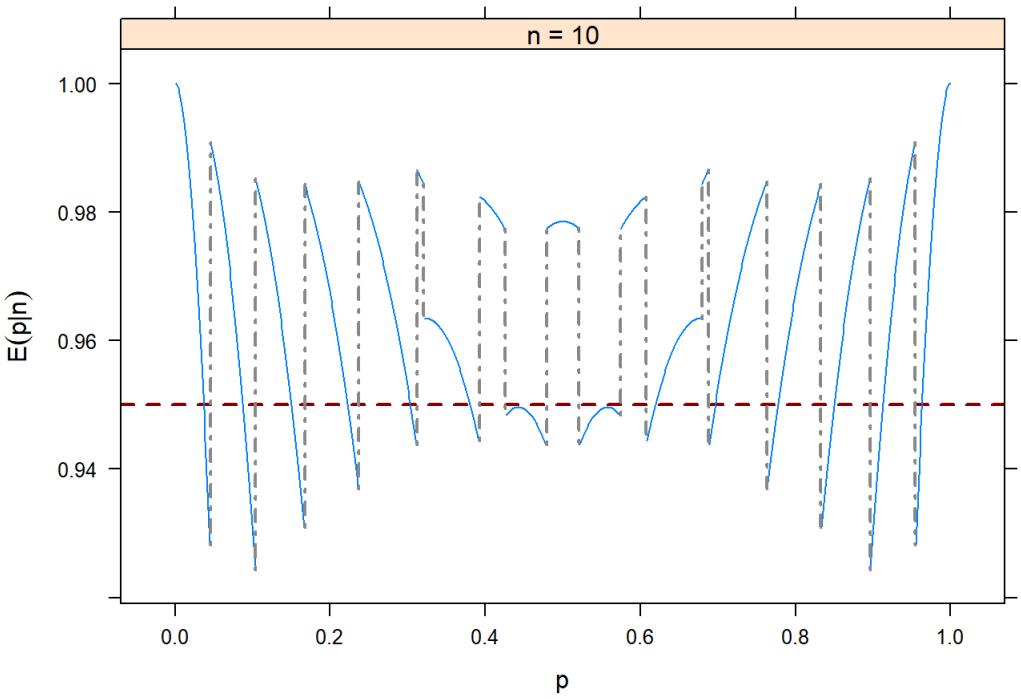
```
print(head(filter(CI.S, method == "exact")))#clopper-pearson CIS
```

```
##   method x  n mean       lower     upper
## 1  exact 0 10  0.0 0.000000000 0.3084971
## 2  exact 1 10  0.1 0.002528579 0.4450161
## 3  exact 2 10  0.2 0.025210726 0.5560955
## 4  exact 3 10  0.3 0.066739511 0.6524529
## 5  exact 4 10  0.4 0.121552258 0.7376219
## 6  exact 5 10  0.5 0.187086028 0.8129140
```

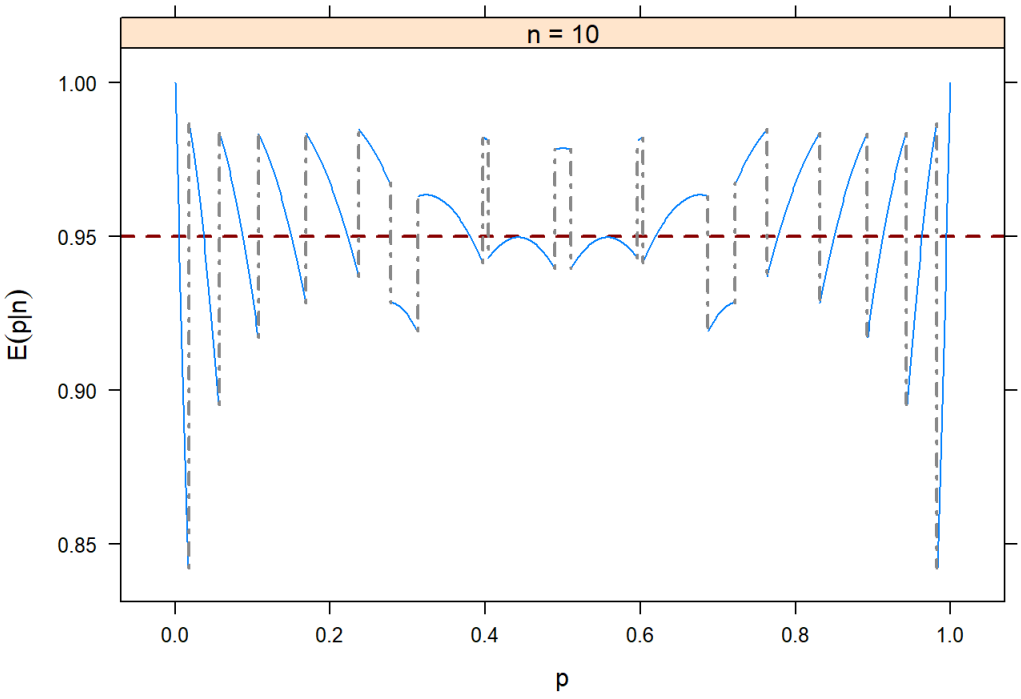
```
binom.plot(n = 10, method = binom.asymp, np = 1000, conf.level = 0.95)
```



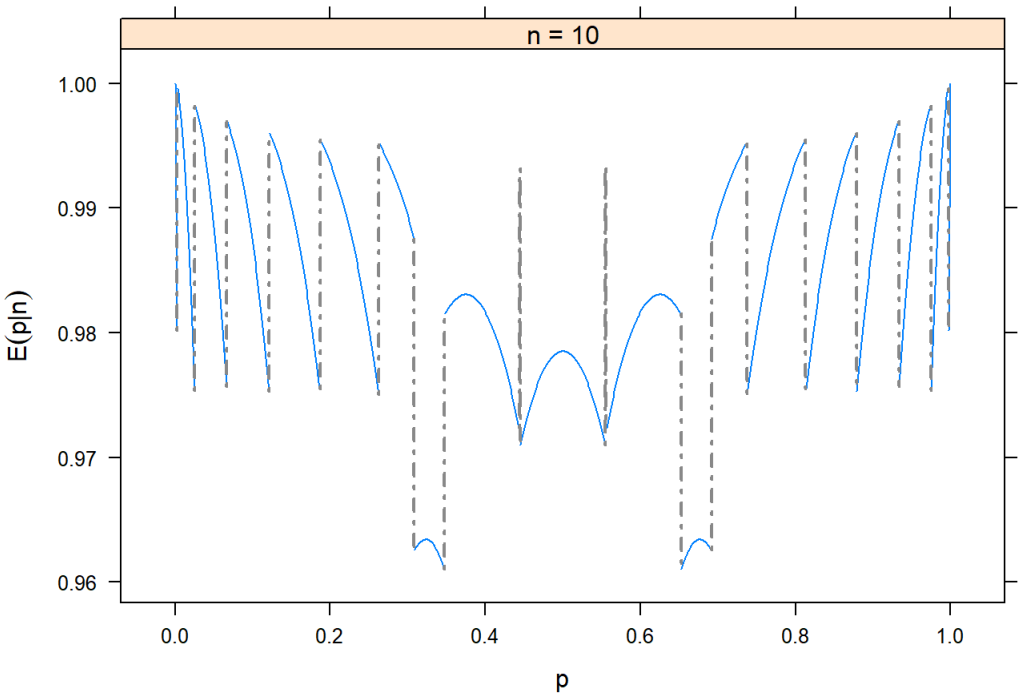
```
binom.plot(n = 10, method = binom.agresti.coull, np = 1000, conf.level = 0.95)
```




```
binom.plot(n = 10, method = binom.wilson, np = 1000, conf.level = 0.95)
```




```
binom.plot(n = 10, method = binom.exact, np = 1000, conf.level = 0.95)
```



a. For large sample, all 4 types of intervals appear to be narrower than with small sample. In both sample sizes, Clopper-Pearson CIs are slightly wider than other 3.

b. For large sample size, the patterns of coverage seem to behave well with minor differences among different methods. However with small sample, comparing to large sample, the curves are less likely to stay along the .95 line(this is very obvious in graphs for wald and clopper-pearson), which means the CIs are less likely to guarantee having 95% confident level when dealing with small sample.