

STAC51A1

Yulun Wu

2023-01-21

Q2

(a)

```
(5/30)-1.645*sqrt((5*25/900)/30)
```

```
## [1] 0.05473859
```

```
(5/30)+1.645*sqrt((5*25/900)/30)
```

```
## [1] 0.2785947
```

```
(5+(1.645^2)/2-1.645*sqrt((5*25/30)+(1.645*1.645/4)))/(30+1.645^2)
```

```
## [1] 0.08355728
```

```
(factorial(30)/(factorial(5)*factorial(25)))*(0.1^5)*(0.9^25)
```

```
## [1] 0.1023048
```

```
(factorial(30)/(factorial(5)*factorial(25)))*((5/30)^5)*((25/30)^25)
```

```
## [1] 0.1921081
```

```
-2*log(0.1023048/0.1921081)
```

```
## [1] 1.260203
```

```
# Score test p-value
```

```
2*pnorm(1.217161,lower.tail = F)
```

```
## [1] 0.223543
```

```
# Wald test p-value
```

```
2*pnorm(0.9797959,lower.tail = F)
```

```
## [1] 0.3271869
```

```
# Likelihood ratio test p-value
```

```
pchisq(1.257829,df=1,lower.tail = F)
```

```
## [1] 0.2620624
```

(e)

```
# Score test and Score CI
```

```
prop.test(5,30,p=0.1, conf.level = 0.9, correct=F)
```

```
## Warning in prop.test(5, 30, p = 0.1, conf.level = 0.9, correct = F): Chi-squared
## approximation may be incorrect
```

```
##
## 1-sample proportions test without continuity correction
##
## data: 5 out of 30, null probability 0.1
## X-squared = 1.4815, df = 1, p-value = 0.2235
## alternative hypothesis: true p is not equal to 0.1
## 90 percent confidence interval:
## 0.08356237 0.30492050
## sample estimates:
## p
## 0.1666667
```

Score CI is (0.08356237,0.30492050), and p-value is 0.2235.p;

(f)

```
library(binom)
# Wald CI and Agresti-Coull CI
binom.confint(x=5,n=30,conf.level=0.9,methods=c("asymptotic","agresti-coull"))
```

```
##          method x  n      mean      lower      upper
## 1 agresti-coull 5 30 0.1666667 0.08045514 0.3080277
## 2 asymptotic 5 30 0.1666667 0.05474855 0.2785848
```

Agresti-Coull CI is (0.08045514,0.3080277). Wald CI is (0.05474855,0.2785848)

Q3

(d)

```
# 0.9 quantile of chi-square distribution with 1 df
qchisq(0.9,df=1)
```

```
## [1] 2.705543
```

Likelihood ratio test statistic must be at least 2.705543 to be significant at 0.1 level.

(e)

```
library(rootSolve)
n = 30
y = 5
phat_q3 = 5/30
alpha = 0.1
# function to be solved
f1 = function(pi0){
  -2*(y*log(pi0)+(n-y)*log(1-pi0)-y*log(phat_q3)-(n-y)*log(1-phat_q3))-qchisq(1-alpha,df=1)
}
# solve to find CI
uniroot.all(f=f1,interval=c(0.000001,0.999999))
```

```
## [1] 0.07569361 0.29637349
```

90% Likelihood ratio CI is (0.07569361,0.29637349).

Q4

(a)

```
set.seed(1004912785)
# Generate 100000 samples with trail = 25 and pi=0.06
n_q4a = 25
x_q4=rbinom(100000,n_q4a,0.06)
alpha_q4a = 0.05
z_q4a = qnorm(1-(alpha_q4a/2),0,1)
# vector of pi_hat
phat_q4a=x_q4/n_q4a
# Wald CI lower bound
CI_lowq4a = phat_q4a-z_q4a*sqrt(phat_q4a*(1-phat_q4a)/n_q4a)
# Wald CI upper bound
CI_upq4a = phat_q4a+z_q4a*sqrt(phat_q4a*(1-phat_q4a)/n_q4a)
# column bind CI_low and CI_up to a matrix, each row is corresponding to 1 sample
CI_q4a = cbind(CI_lowq4a,CI_upq4a)
# find the proportion of Wald intervals such that lower bound <= 0.06 and upper bound >= 0.06
mean((CI_q4a[,1]<=0.06)&(CI_q4a[,2]>=0.06))

## [1] 0.78358
```

The proportion of Wald intervals that contains 0.06 is 0.78358, but we are doing 95% confidence interval, that means we expect the proportion to be 0.95. This happens because our pi is 0.06 and this number is too close to 0, so we have many (0,0) CIs.

(b)

```
n_q4b = 25
y_q4b = c(0:n_q4b)
alpha_q4b = 0.05
z_q4b = qnorm(1-(alpha_q4b/2),0,1)
# vector of pi_hat
phat_q4b=y_q4b/n_q4b
# Wald CI lower bound
CI_lowq4b = phat_q4b-z_q4b*sqrt(phat_q4b*(1-phat_q4b)/n_q4b)
# Wald CI upper bound
CI_upq4b = phat_q4b+z_q4b*sqrt(phat_q4b*(1-phat_q4b)/n_q4b)
# column bind CI_low and CI_up to a matrix, each row is corresponding to 1 y value
CI_q4b = cbind(CI_lowq4b,CI_upq4b)
# find the proportion of Wald intervals such that lower bound <= 0.06 and upper bound >= 0.06
includeBool_q4b = (CI_q4b[,1]<=0.06)&(CI_q4b[,2]>=0.06)
density_q4b = dbinom(y_q4b, n_q4b, 0.06)
confidence_level_q4b = sum(includeBool_q4b*density_q4b)
confidence_level_q4b*100

## [1] 78.4026
```

The true confidence level is 78.4026

Q5

```

WaldCI = function(n,pihat,z) {
  # Wald CI lower bound
  CI_low = pihat-z*sqrt(pihat*(1-pihat)/n)
  # Wald CI upper bound
  CI_up = pihat+z*sqrt(pihat*(1-pihat)/n)
  # column bind CI_low and CI_up to a matrix, each row is corresponding to 1 y value
  return(cbind(CI_low,CI_up))
}

ScoreCI = function(n,pihat,z) {
  nstar = n + z^2
  a = (n*pihat+((z^2)/2))/nstar
  b = sqrt(n*pihat*(1-pihat)+((z^2)/4))/nstar
  # Score CI lower bound
  CI_low = a-z*b
  # Score CI upper bound
  CI_up = a+z*b
  # column bind CI_low and CI_up to a matrix, each row is corresponding to 1 y value
  return(cbind(CI_low,CI_up))
}

ACoullCI = function(n,z,y) {
  nstar = n+4
  a = (y+2)/nstar
  b = sqrt(a*(1-a)/nstar)
  # Agresti-Coull CI lower bound
  CI_low = a-z*b
  # Agresti-Coull CI upper bound
  CI_up = a+z*b
  # column bind CI_low and CI_up to a matrix, each row is corresponding to 1 y value
  return(cbind(CI_low,CI_up))
}

CPCI = function(n,f1,f2,y) {
  # Clopper-Pearson CI lower bound
  CI_low = (1+((n-y+1)/(t(y)*f1)))^-1
  # Clopper-Pearson CI upper bound
  CI_up = (1+((n-y)/((t(y)+1)*f2)))^-1
  # column bind CI_low and CI_up to a matrix, each row is corresponding to 1 y value
  return(cbind(t(CI_low),t(CI_up)))
}

probCoverage = function(p,CItype,n,pihat,cv1,cv2=0,y) {
  # Function that return probability coverage
  if (CItype == 0){
    # Calculate Wald CI
    CI = WaldCI(n,pihat,cv1)
  } else if (CItype == 1) {
    # Calculate Score CI
    CI = ScoreCI(n,pihat,cv1)
  } else if (CItype == 2) {
    # Calculate Agresti-Coull CI
    CI = ACoullCI(n,cv1,y)
  } else if (CItype == 3) {
    # Calculate Clopper Pearson CI

```

```

    CI = CPCI(n,cv1,cv2,y)
  }
  # find the proportion of Wald intervals such that lower bound <= p and      upper bound >= p
  includeBool = (CI[,1]<=p)&(CI[,2]>=p)
  # find density of binomial with y=y, n=n, pi=p
  density = dbinom(y, n, p)
  return(sum(includeBool*density))
}

```

(a)(b)

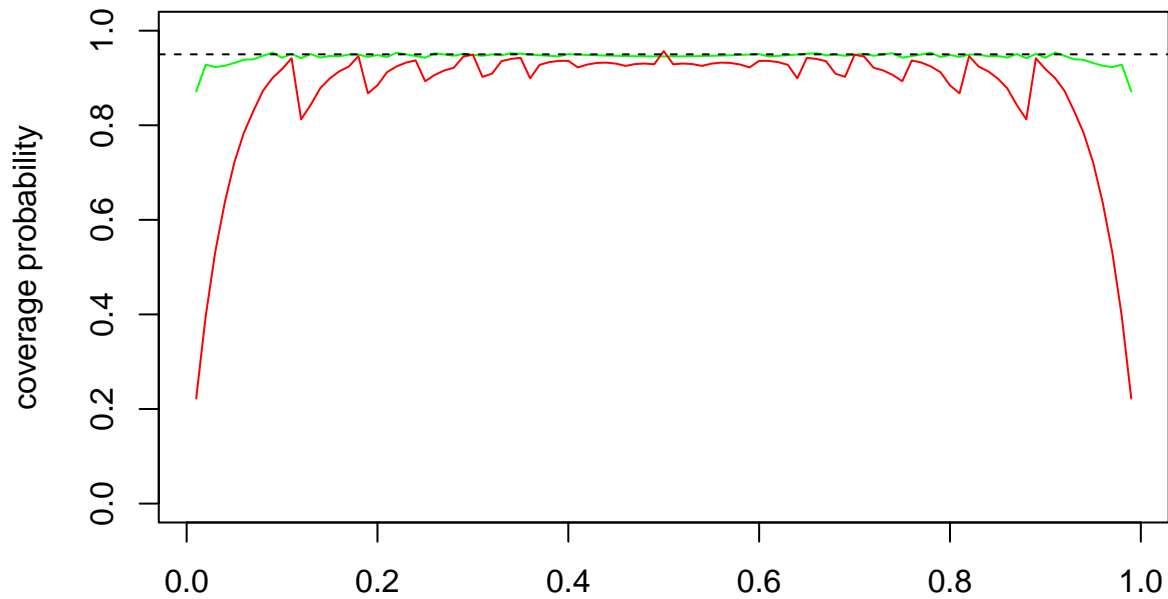
```

#(a)
pi_q5 = as.matrix(seq(0.01, 0.99, by = .01))
n_q5a = 25
alpha_q5 = 0.05
y_q5a = 0:n_q5a
pihat_q5a = y_q5a/n_q5a
z_q5 = qnorm(1-(alpha_q5/2),0,1)

#(b)
n_q5b = 500
y_q5b = 0:n_q5b
pihat_q5b = y_q5b/n_q5b

# Plot graph
# probability coverage of n=500
plot(pi_q5, sapply(pi_q5, probCoverage, CItype=0, n=n_q5b, pihat=pihat_q5b, cv1=z_q5, y=y_q5b), type = "l", ylab="Coverage", col="black")
# probability coverage of n=25
lines(pi_q5, sapply(pi_q5, probCoverage, CItype=0, n=n_q5a, pihat=pihat_q5a, cv1=z_q5, y=y_q5a), type = "l", col="red")
# line of 0.95
abline(h = 1-alpha_q5, lty = 2)

```



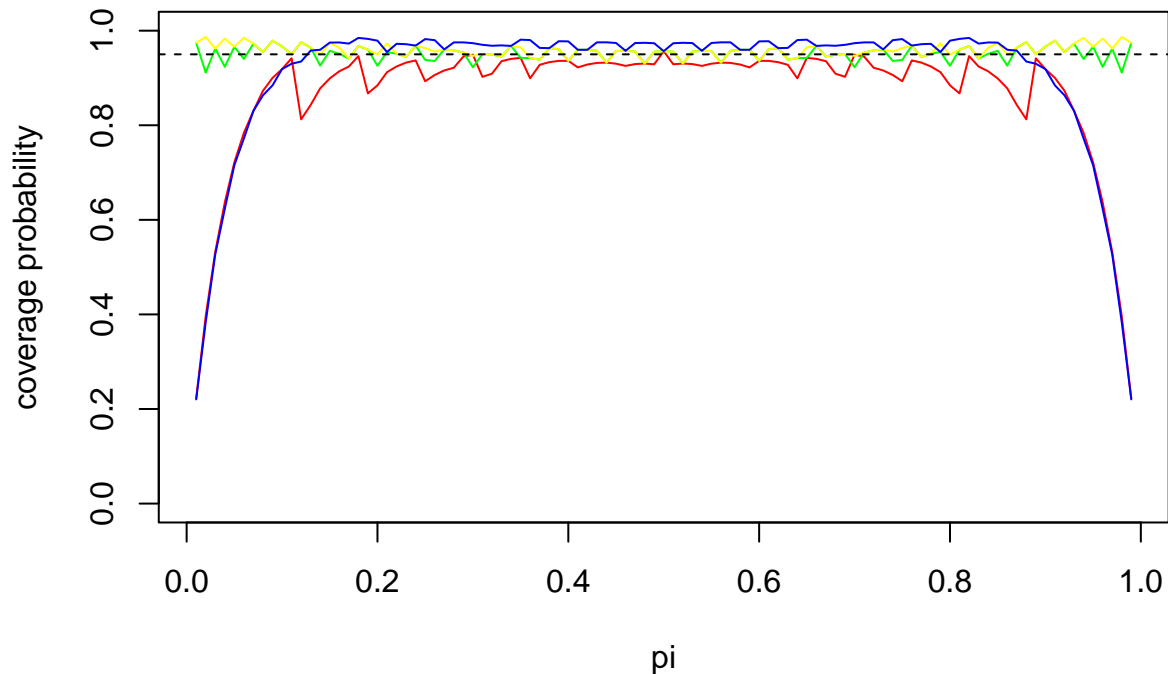
pi_q5

(a):

The coverage probability of a 95% Wald confidence interval is mostly below 95% when $n = 25$. (b): The coverage probability of a 95% Wald confidence interval is mostly 95% or slightly lower than 95% when $n = 500$.

(c)

```
# y and pi_hat for Clopper Pearson CI, because Clopper Pearson CI doesn't work when y = 0 or n
y_q5cp = 1:(n_q5a-1)
pihat_q5cp = y_q5cp/n_q5a
# Used in Clopper Pearson CI
f_q5c1 = qf(alpha_q5/2,df1=2*y_q5cp,df2=2*(n_q5a-y_q5cp+1))
f_q5c2 = qf(1-(alpha_q5/2),df1=2*(y_q5cp+1),df2=2*(n_q5a-y_q5cp))
# coverage probability of Wald CI
plot(pi_q5, sapply(pi_q5, probCoverage,CItype=0,n=n_q5a,pihat=pihat_q5a,cv1=z_q5,y=y_q5a), type = "l",
     col="red",ylab = "coverage probability",xlab="pi",ylim=c(0,1))
# coverage probability of Score CI
lines(pi_q5, sapply(pi_q5, probCoverage,CItype=1,n=n_q5a,pihat=pihat_q5a,cv1=z_q5,y=y_q5a), type = "l",
     col="green")
# coverage probability of Agresti-Coull CI
lines(pi_q5, sapply(pi_q5, probCoverage,CItype=2,n=n_q5a,pihat=pihat_q5a,cv1=z_q5,y=y_q5a), type = "l",
     col="yellow")
# coverage probability of Clopper-Pearson CI
lines(pi_q5, sapply(pi_q5, probCoverage,CItype=3,n=n_q5a,pihat=pihat_q5cp,cv1=f_q5c1,cv2=f_q5c2,y=y_q5cp), type = "l",
     col="blue")
# line of 0.95
abline(h = 1-alpha_q5, lty = 2)
```



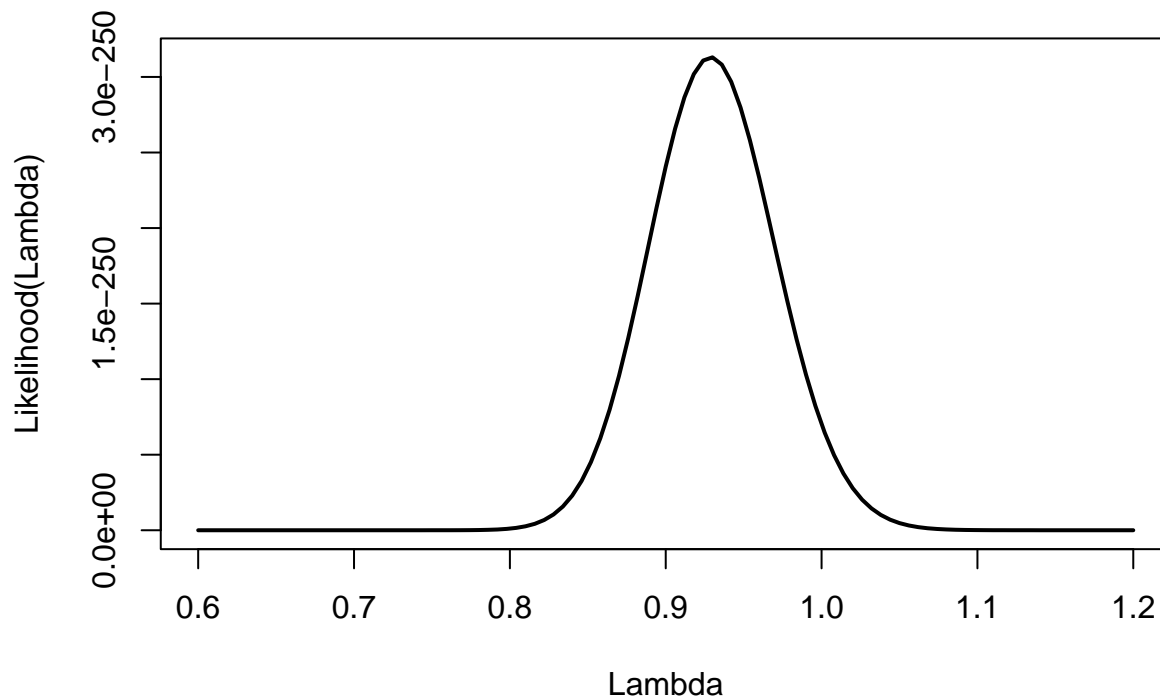
```
#library(binom)
#binom.coverage(pi_q5, 25, conf.level = 0.95, method = "exact")
```

Coverage probability of 95% Wald confidence interval is the poorest among 4 types of confidence intervals, it mostly below 95% and as π gets closer to 0 or 1, the coverage probability can get as low as around 20%. Clopper-Pearson confidence interval always greater than 95% when π is not close to 0 or 1. Agresti-Coull confidence interval perform better when π is close to 0 or 1, and as π gets closer to 0.5, the coverage probability tends to below 95%. Score confidence interval is more stable, its coverage probability just waving around about 95% all the time.

Q6

(b)

```
# Likelihood function obtained in (a)
Likelihood = function(lam){
  return(exp(-576*lam)*(lam^535))
}
curve(Likelihood,from=0.6,to=1.2,xlab="Lambda",ylab="Likelihood(Lambda)",lwd=2)
```



```
optimize(Likelihood,interval=c(0.6,1.2),maximum = TRUE)
```

```
## $maximum
## [1] 0.9288118
##
## $objective
## [1] 3.1311e-250
```

(c)

```
lam_hat = 0.9288118
exp(-576*lam_hat)*(lam_hat^535)
```

```
## [1] 3.1311e-250
```

The maximum likelihood of lambda is 3.1311e.

(d)

```
lam0 = 1
alphaq6 = 0.05
l0 = exp(-576*lam0)*(lam0^535)
l0
```

```
## [1] 7.020668e-251
```

```
l1 = exp(-576*lam_hat)*(lam_hat^535)
l1
```

```
## [1] 3.1311e-250
```

```
test_stat = -2*log(l0/l1)
test_stat
```



```
## [1] 2.990222
```

```
cv = qchisq(1-alphaq6,df=1)  
cv
```

```
## [1] 3.841459
```

Test statistics = 2.990222, critical value = 3.841459, $2.990222 < 3.841459$, therefore we fail to reject the null hypothesis that $\lambda=1$ at level=0.05.