Assignment1

Hanxiao Du(1004439529), Jeffery Wei Xuan Su(1004139684)

Q1.

(a)

$$Let\ \vec{Y} = [Y_1\ Y_2\ \cdots\ Y_k]^T,\ \vec{t} = [t_1\ t_2\ \cdots t_k]$$

.

$$M_{\vec{Y}}(\vec{t}) = E[exp(\vec{t}^T\vec{Y})] = E[\prod_{i=1}^{k} exp(t_i Y_i)]$$

$$= \sum_{\sum_{i=1}^{k} Y_i = n} (\frac{n!}{Y_1!Y_2!\ \cdots\ Y_k!}) \prod_{i=1}^{k} \pi_i^{Y_i} \prod_{j=1}^{k} exp(t_j))^{Y_j}$$

$$= \sum_{\sum_{i=1}^{k} Y_i = n} (\frac{n!}{Y_1!Y_2!\ \cdots\ Y_k!}) \prod_{i=1}^{k} (\pi_i \cdot exp(t_i))^{Y_i}$$

$$= (\sum_{i=1}^{k} \pi_i \cdot exp(t_i))^n,\ \textit{by multinomial theorem.}$$

(b)

$$M_{Y_j}(t_j) = M_{\vec{Y}}(0, \cdots, 0, t_j, 0, \cdots, 0)$$

$$= (\sum_{i=1}^{k} \pi_i \cdot exp(t_i))^n$$

$$= (\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^n$$

$$E[Y_j] = \frac{\partial}{\partial t_j} M_{Y_j}(t_j)|_{t_j=0}$$

$$= \frac{\partial}{\partial t_j} (\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^n|_{t_j=0}$$

$$= n(\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^{n-1} \pi_j \cdot \exp(t_j)|_{t_j=0}$$

$$= n(\sum_{i=1}^{k} \pi_i)^{n-1} \cdot \pi_j exp(0)$$

$$= n \cdot (1) \cdot \pi_j \cdot (1)$$

$$= n\pi_j$$

(c)

$$E[Y_j^2] = \frac{\partial^2}{\partial t_j^2} M_{Y_j}(t_j)|_{t_j=0}$$

$$= \frac{\partial^2}{\partial t_j^2}(\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^n|_{t_j=0}$$

$$= \frac{\partial}{\partial t_j} n(\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^{n-1} \pi_j \cdot \exp(t_j)|_{t_j=0}$$

$$= n(n-1)(\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^{n-2} \cdot \pi_j \cdot \exp(t_j) \cdot \pi_j \cdot \exp(t_j) + n(\sum_{i=1, i\neq j}^{k} \pi_i + \pi_j \cdot exp(t_j))^{n-1} \cdot \pi_j \cdot \exp(t_j)|_{t_j=0}$$

$$= n(n-1) \cdot \pi_j^2 + n\pi_j$$

$$Var(Y_j) = E[Y_j^2] - E[Y_j]^2$$

$$= n(n-1) \cdot \pi_j^2 + n\pi_j - (n\pi_j)^2$$

$$= n\pi_j - n\pi_j^2$$

$$= n\pi_j(1 - \pi_j)$$

(d)

$$M_{Y_i,\ Y_j}(t_i,\ t_j) = M_{\vec{Y}}(0, \cdots,\ 0,\ t_i,\ 0, \cdots,\ 0,\ t_j,\ 0,\ \cdots,\ 0)$$

$$= (\sum_{l=1,\ l\neq i,\ j}^{k} \pi_l + \pi_i exp(t_i) + \pi_j exp(t_j))^n$$

$$E[Y_i Y_j] = \frac{\partial^2}{\partial t_i \partial t_j} M_{Y_i,\ Y_j}(t_i,\ t_j)|_{t_i=0,\ t_j=0}$$

$$= \frac{\partial^2}{\partial t_i \partial t_j}(\sum_{l=1,\ l\neq i,\ j}^{k} (\pi_l + \pi_i exp(t_i) + \pi_j exp(t_j)))^n|_{t_i=0,\ t_j=0}$$

$$= \frac{\partial}{\partial t_j} n(\sum_{l=1,\ l\neq i,\ j}^{k} (\pi_l + \pi_i exp(t_i) + \pi_j exp(t_j)))^{n-1} \pi_i exp(t_i)|_{t_i=0,\ t_j=0}$$

$$= \pi_i exp(t_i)[n(n-1)(\sum_{l=1,\ l\neq i,\ j}^{k} (\pi_l + \pi_i exp(t_i) + \pi_j exp(t_j)))^{n-2} \pi_j exp(t_j)]|_{t_i=0,\ t_j=0}$$

$$= \pi_i[n(n-1)\pi_j]$$

$$Cov(Y_i,\ Y_2) = E[Y_i Y_j] - E[Y_i]E[Y_j]$$

$$= \pi_i[n(n-1)\pi_j] - \pi_i n \pi_j n$$

$$= -n\pi_i\pi_j$$

(e)

$$Corr(Y_i, Y_j) = \frac{Cov(Y_i, Y_j)}{Sd(Y_i)Sd(Y_j)}$$

$$= \frac{Cov(Y_i, Y_j)}{\sqrt{Var(Y_i) \cdot Var(Y_j)}}$$

$$= \frac{-n\pi_i\pi_j}{\sqrt{n\pi_i(1-\pi_i) \cdot n\pi_j(1-\pi_j)}}$$

$$= \frac{-n\pi_i\pi_j}{\sqrt{n^2\pi_i\pi_j\pi_i\pi_j}}$$
$$= \frac{-n\pi_i\pi_j}{n\pi_i\pi_j}$$
$$= -1$$

Explanation: Since C=2 and $Y_1 + Y_2 = n$, so $Y_1 = n - Y_2$ and $Y_2 = n - Y_1$, which means Y_1 is negatively related to Y_2 and they must sum to some constant n. Therefore, $Corr(Y_1, Y_2) = -1$.

Q2.

(a)

$$Y \sim Bin(n, \pi),\ where\ n = 40, \pi = 0.2$$

We observed the value of Y, $y = 10$

95% Wald confidence interval:

$$\hat{\pi} = \frac{y}{n} = \frac{10}{40} = 0.25$$
$$(\hat{\pi} - Z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}, \hat{\pi} + Z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}})$$
$$= (0.25 - 1.96\sqrt{\frac{0.25(1-0.25)}{40}}, 0.25 + 1.96\sqrt{\frac{0.25(1-0.25)}{40}})$$
$$= (0.115808, 0.384192)$$

95% score confidence interval:

$$\hat{\pi} = \frac{y}{n} = \frac{10}{40} = 0.25$$
$$(\frac{(n\hat{\pi} + z^2_{\frac{\alpha}{2}}/2) - z_{\frac{\alpha}{2}}\sqrt{n\hat{\pi}(1-\hat{\pi}) + z^2_{\frac{\alpha}{2}}/4}}{n + z^2_{\frac{\alpha}{2}}}, \frac{(n\hat{\pi} += z^2_{\frac{\alpha}{2}}/2) + z_{\frac{\alpha}{2}}\sqrt{n\hat{\pi}(1-\hat{\pi}) + z^2_{\frac{\alpha}{2}}/4}}{n + z^2_{\frac{\alpha}{2}}})$$
$$= (\frac{(40(0.25) + 1.96^2/2) - 1.96\sqrt{40(0.25)(1-0.25) + 1.96^2/4}}{40 + 1.96^2}, \frac{(40(0.25) + 1.96^2/2) + 1.96\sqrt{40(0.25)(1-0.25) + 1.96^2/4}}{40 + 1.96^2})$$
$$= (0.141870, 0.401943)$$

95% Agresti-coull confidence interval:

$$\hat{\pi}^* = \frac{y+2}{n+4} = \frac{10+2}{40+4} = \frac{3}{11} = 0.272727$$
$$(\hat{\pi}^* - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}^*(1-\hat{\pi}^*)}{n+4}}, \hat{\pi}^* + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}^*(1-\hat{\pi}^*)}{n+4}})$$
$$= (\frac{3}{11} - 1.96\sqrt{\frac{\frac{3}{11}(1-\frac{3}{11})}{40+4}}, \frac{3}{11} + 1.96\sqrt{\frac{\frac{3}{11}(1-\frac{3}{11})}{40+4}})$$
$$= (0.141131, 0.404323)$$

(b)

```r
library(rootSolve)
n=40
y=10
pi_hat=y/n  # MLE of pi
alpha=0.05
f1 <- function(pi){
  -2*(y*log(pi) + (n-y)*log(1-pi)-y*log(pi_hat) - (n-y)*log(1-pi_hat)) -
    qchisq(1-alpha, df=1)
}
uniroot.all(f=f1, interval=c(0,1)) # Find the roots of the function to determine C.I.
```

```
## [1] 0.1342088 0.3969699
```

95% confidence interval π based on likelihood ratio test:

$$(0.1342088, 0.3969699)$$

Q3.

(a)

```r
set.seed(1004439529)
N=100000 # sample size

# Parameters for Binomial distribution.
n = 30
pi = 0.2
b=rbinom(N, n, pi) # Generate N samples from Bin(n, pi).

# Function calculates 95% Wald C.I. for single observation.
ci <- function(y) {
  pi_hat = y/n
  return(c(pi_hat-1.96*(pi_hat*(1-pi_hat)/n)^(1/2), pi_hat+
             1.96*(pi_hat*(1-pi_hat)/n)^(1/2)))
}

count = 0 # Counts how many times out of N that pi is in the 95% Wald C.I.

# Loop through all samples generated.
for (s in b) {
  int = ci(s) # 95% Wald C.I. for current sample.
  # Check if the true pi is in the C.I.
  if (pi >= int[1] && pi <= int[2]) {
    count <- (count + 1) # Increases count by 1.
  }
}

count/N # Observed (or true) coverage probability of confidence intervals
```

```
## [1] 0.94606
```

Comments: From the result, by Monte Carlo simulation, the Observed (or true) coverage probability of confidence intervals is 94.606%. However, the targeted coverage probability of confidence intervals is 95%, so that they are not necessary equal. Furthermore, this is also one of the drawbacks of Wald C.I. that Observed (or true) coverage probability of confidence intervals (94.606% in this case) could be less than targeted coverage probability of confidence intervals (95%).

(b)

```
n = 30
pi = 0.2
# Function calculates 95% Wald C.I. for single observation.
ci <- function(y) {
  pi_hat = y/n
  return(c(pi_hat-1.96*(pi_hat*(1-pi_hat)/n)^(1/2),
           pi_hat+1.96*(pi_hat*(1-pi_hat)/n)^(1/2)))
}

true_ci_level = 0 # The true C.I. level

# Loop through
for (y in 0:n) {
  int = ci(y)
  if (pi >= int[1] && pi <= int[2]) { # Equivalent to I(y) in the question.
    true_ci_level = (true_ci_level + dbinom(y, n, pi)) # Summation
  }
}
true_ci_level
```

```
## [1] 0.9463279
```

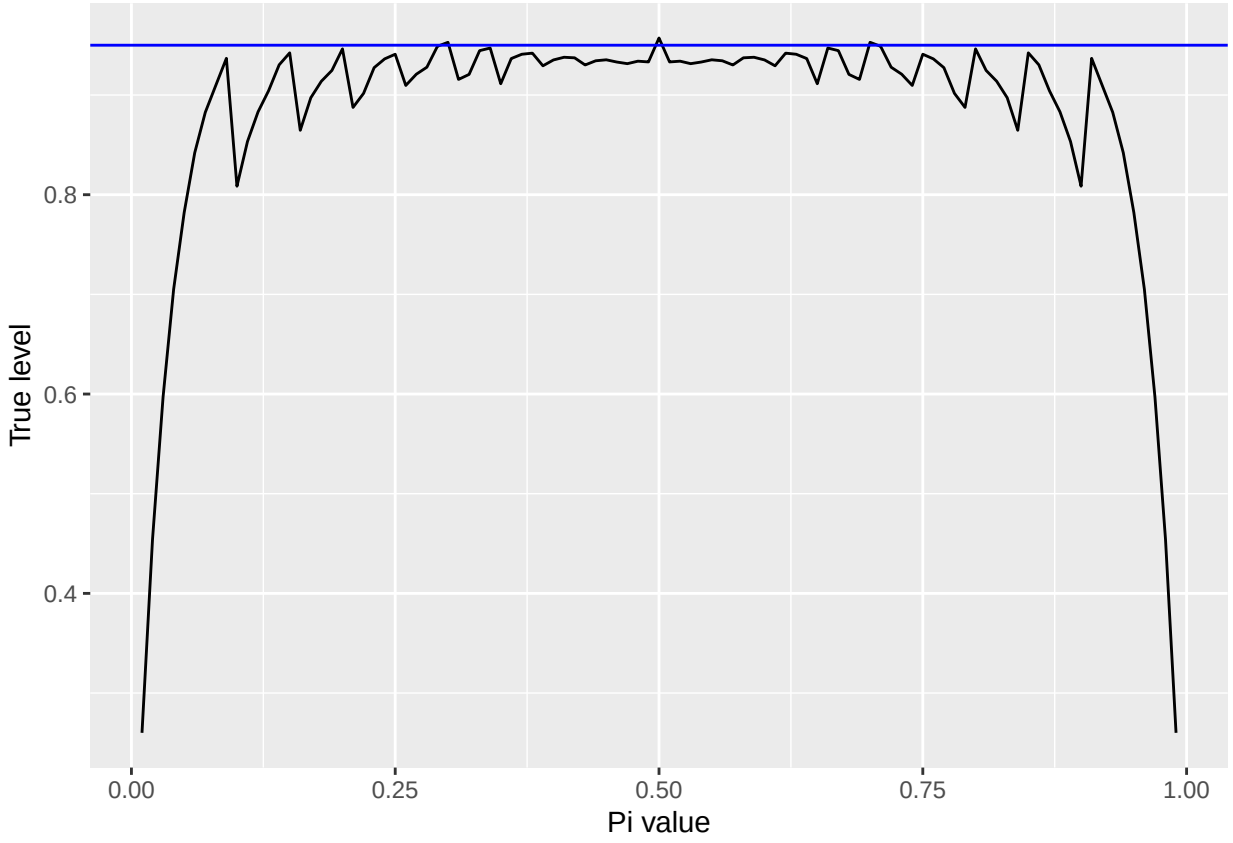
Comments: By direct calculation, the Observed (or true) coverage probability of confidence intervals is 94.63279%. However, the targeted coverage probability of confidence intervals is 95%, so that they are not necessary equal. Also, these two are different because we are using the pmf of Binomial distribution directly to compute instead of using the normal distribution.

Q4

(a)

```
n = 30

# Function calculates 95% Wald C.I. for single observation.
ci <- function(y) {
  pi_hat = y/n
  return(c(pi_hat-1.96*(pi_hat*(1-pi_hat)/n)^(1/2),
           pi_hat+1.96*(pi_hat*(1-pi_hat)/n)^(1/2)))
}


true_level<- function(pi) {
  true_ci_level = 0 # The true C.I. level
  # Loop through
  for (y in 0:n) {
    int = ci(y)
    if (pi >= int[1] && pi <= int[2]) { # Equivalent to I(y) in the question.
      true_ci_level = (true_ci_level + dbinom(y, n, pi)) # Summation
    }
  }
  return(true_ci_level)
}

true_levels = c()
```

```
pi_values = seq(0.01, 0.99, by = 0.01)
for (pi in pi_values) {
  true_levels <- append(true_levels, true_level(pi))
}

points <- data.frame("Pi"=pi_values, "True level"=true_levels)

library(ggplot2)
ggplot(points, aes(x=pi_values, y = true_levels)) + geom_line()+ geom_hline(yintercept=0.95, color="blu
```

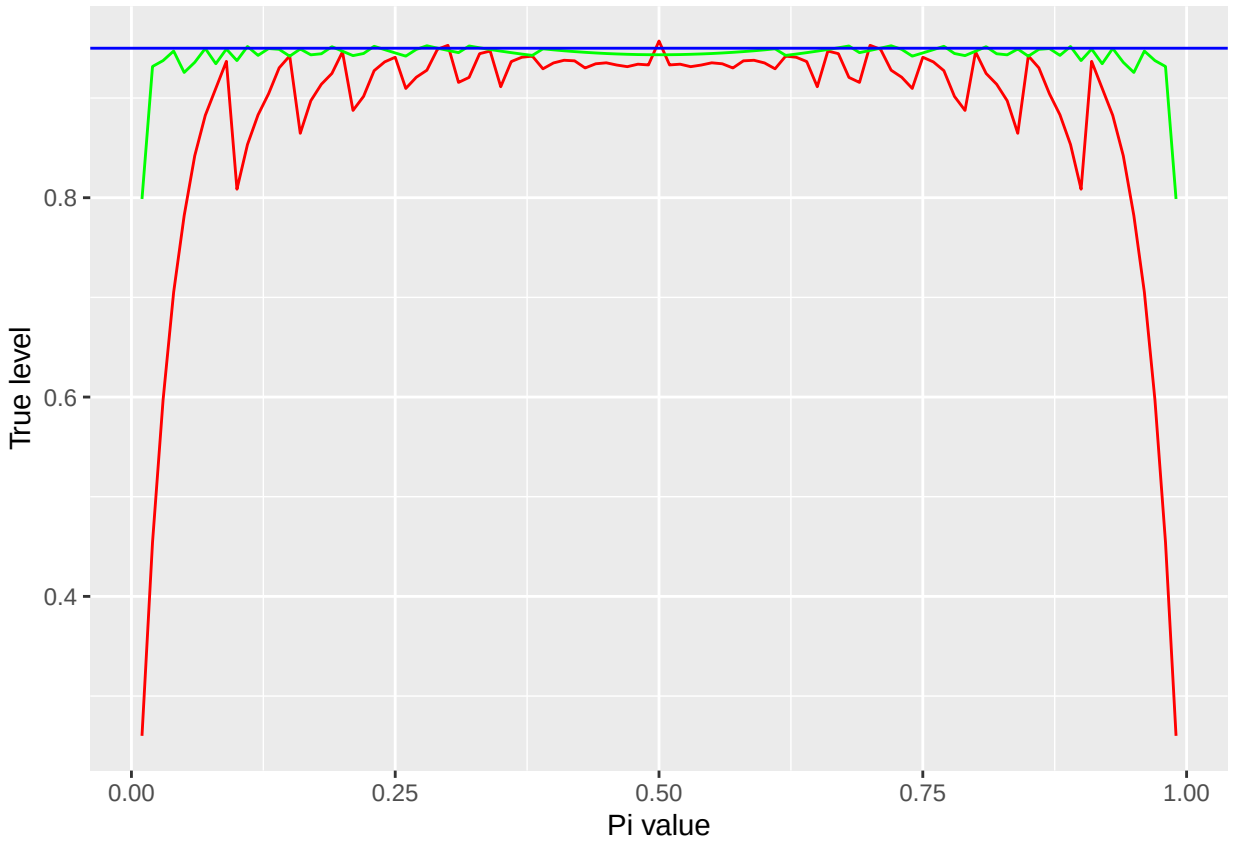


Comments: From the plot, Wald C.I. usually have a true coverage probability of confidence intervals less than the targeted coverage probability of confidence intervals (95%, blue line in the graph). In addtion, Wald C.I. is collapsing as $\pi \to 0$ and $\pi \to 1$.

(b)

```
n1 = 30
n2 = 300

# Function calculates 95% Wald C.I. for single observation.
ci <- function(y, n) {
  pi_hat = y/n
  return(c(pi_hat-1.96*(pi_hat*(1-pi_hat)/n)^(1/2),
           pi_hat+1.96*(pi_hat*(1-pi_hat)/n)^(1/2)))
}
```

```r
true_level<- function(pi, n) {
  true_ci_level = 0 # The true C.I. level
  # Loop through
  for (y in 0:n) {
    int = ci(y, n)
    if (pi >= int[1] && pi <= int[2]) { # Equivalent to I(y) in the question.
      true_ci_level = (true_ci_level + dbinom(y, n, pi)) # Summation
    }
  }
  return(true_ci_level)
}

true_levels_n1 = c()
true_levels_n2 = c()

pi_values = seq(0.01, 0.99, by = 0.01)
for (pi in pi_values) {
  true_levels_n1 <- append(true_levels_n1, true_level(pi, n1))
  true_levels_n2 <- append(true_levels_n2, true_level(pi, n2))
}

points_n1 <- data.frame("Pi"=pi_values, "True level"=true_levels_n1)
points_n2 <- data.frame("Pi"=pi_values, "True level"=true_levels_n2)

library(ggplot2)

ggplot() + geom_line(data=points_n1, aes(x=pi_values, y=true_levels_n1), color="red") + geom_line(data=p
```

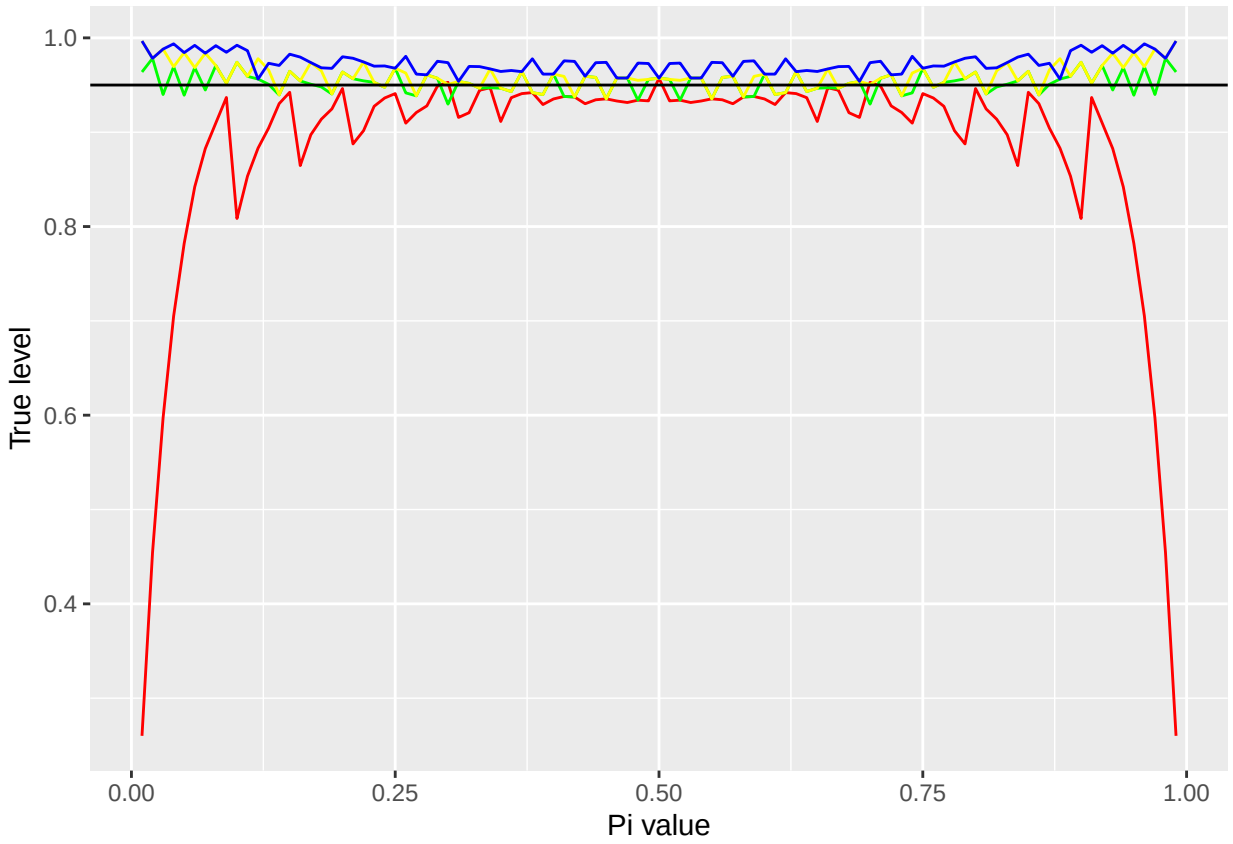


Comment: As n gets larger (i.e. n=300, green line on the graph), the true coverage probability of confidence intervals of Wald C.I. tends to get closer to the targeted coverage probability of confidence intervals (95%, blue line on the graph). Furthermore, it is still collapsing as $\pi \to 0$ and $\pi \to 1$, but the rate of collapsing is smaller than smaller n (i.e. n=30, red line on the graph).

(c)

```
n = 30
# Function calculates 95% Wald C.I. for single observation.
ci_wald <- function(y) {
  pi_hat = y/n
  return(c(pi_hat-1.96*(pi_hat*(1-pi_hat)/n)^(1/2),
           pi_hat+1.96*(pi_hat*(1-pi_hat)/n)^(1/2)))
}
# Function calculates 95% score C.I. for single observation.
ci_score <- function(y) {
  pi_hat = y/n
  return(c(((n*pi_hat + 1.96^2 / 2) - 1.96*((n*pi_hat*(1 - pi_hat) +1.96^2 /4))^(1/2))/(n + 1.96^2), ((
}

# Function calculates 95% Agresti-Coull C.I. for single observation.
ci_agresti_coull <- function(y) {
  pi_hat_star = (y+2)/(n+4)
  return(c(pi_hat_star - 1.96*((pi_hat_star*(1-pi_hat_star))/(n+4))^(1/2), pi_hat_star + 1.96*((pi_hat_s
}

# Function calculates 95% Clopper-Pearson C.I. for single observation.
ci_clopper_pearson <- function(y) {
```

```
  if (y == 0) {
    return(c(0, (1 +(n-y)/((y+1)*qf(1-(0.05/2), 2*(y+1), 2*(n-y))))^(-1)))
  }
  if (y == n) {
    return(c((1 +(n-y+1)/(y*qf(0.05/2, 2*y, 2*(n-y+1))))^(-1), 1))
  }
  return(c((1 +(n-y+1)/(y*qf(0.05/2, 2*y, 2*(n-y+1))))^(-1), (1 +(n-y)/((y+1)*qf(1-(0.05/2), 2*(y+1), 2
}

# Function calculates the true coverage probablility level
true_level<- function(pi, ci_func) {
  true_ci_level = 0 # The true C.I. level
  # Loop through
  for (y in 0:n) {
    int = ci_func(y)
    if (pi >= int[1] && pi <= int[2]) { # Equivalent to I(y) in the question.
      true_ci_level = (true_ci_level + dbinom(y, n, pi)) # Summation
    }
  }
  return(true_ci_level)
}

# Data points
true_levels_wald = c()
true_levels_score = c()
true_levels_agresti_coull = c()
true_levels_clopper_pearson = c()

pi_values = seq(0.01, 0.99, by = 0.01)
for (pi in pi_values) {
  true_levels_wald <- append(true_levels_wald, true_level(pi, ci_wald))
  true_levels_score <- append(true_levels_score, true_level(pi, ci_score))
  true_levels_agresti_coull <- append(true_levels_agresti_coull, true_level(pi, ci_agresti_coull))
  true_levels_clopper_pearson <- append(true_levels_clopper_pearson, true_level(pi, ci_clopper_pearson))
}

points_wald <- data.frame("Pi"=pi_values, "True level"=true_levels_wald)
points_score <- data.frame("Pi"=pi_values, "True level"=true_levels_score)
points_agresti_coull <- data.frame("Pi"=pi_values, "True level"=true_levels_agresti_coull)
points_clopper_pearson <- data.frame("Pi"=pi_values, "True level"=true_levels_clopper_pearson)


library(ggplot2)
ggplot()+geom_line(data=points_wald, aes(x=pi_values, y=true_levels_wald), color="red")+geom_line(data=p
```



Comment:

Wald: Red

Score: Green

Agresti coull: Yellow

Clopper Peason: Blue

Comparing to the other three confidence intervals the true coverage probability of Wald confidence interval is below the targeted coverage probability most of the time, while the score and Agresti-coull are performing similar to each other. Their true coverage probabilities are both close to the targeted coverage probability throughout all values for π. On the other hand, Clopper-Pearson confidence interval is above the targeted coverage probability. The true coverage probability of Wald confidence interval is collapsing when $\pi \to 0$ and $\pi \to 1$, while the other three confidence intervals do not collapse for all π values. In addition, the true coverage probability of Wald C.I. is shaking with higher amplitude than others.

Q5.

(a)

$$L(\lambda|y_1, \cdots, y_{576}) = \prod_{i=1}^{576} \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}$$

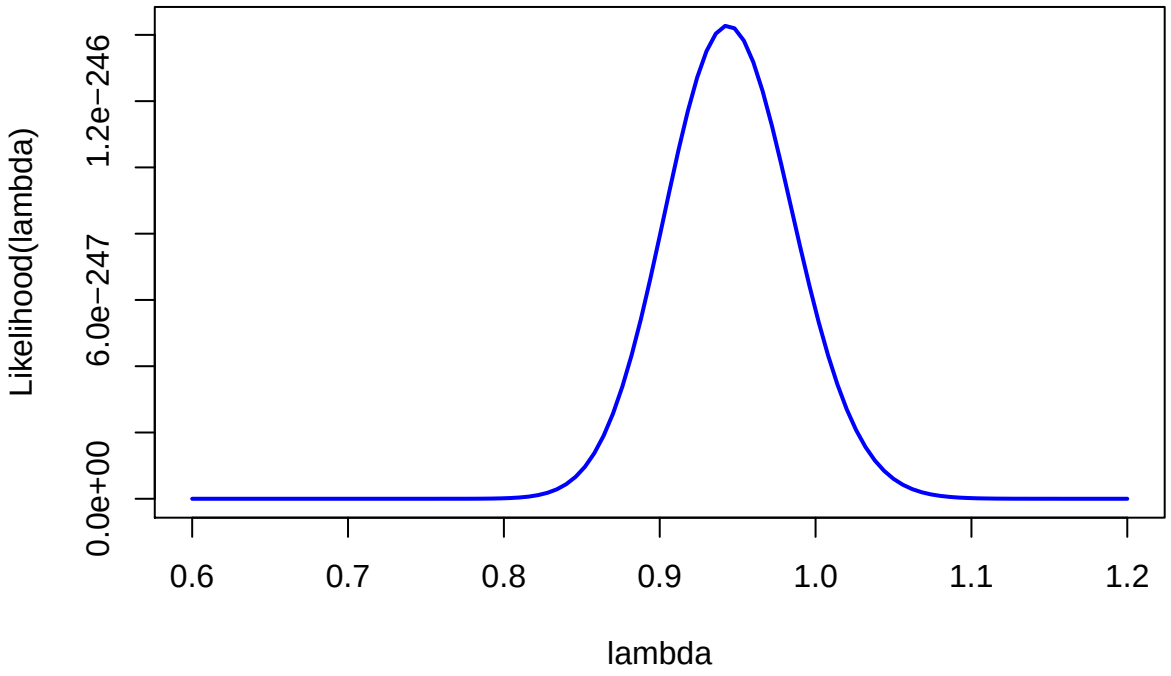
$$= \prod_{i=1}^{576} e^{-\lambda}\lambda^{y_i} \prod_{j=1}^{576} \frac{1}{y_i!}$$

$$= \lambda^{\sum_{i=1}^{576} y_i} e^{-576\lambda} \prod_{j=1}^{576} \frac{1}{y_i!}$$

$$\propto \lambda^{\sum_{i=1}^{576} y_i} e^{-576\lambda}$$

(b)

```
library(geometry)
freq = c(229, 211, 93, 35, 7, 1)
hit = 0:5
sum_yi = dot(freq, hit)

likelihood <- function(lambda) {
  return(lambda^sum_yi * exp(-567*lambda))
}
curve(likelihood, from=0.6, to=1.2, xlab="lambda", ylab="Likelihood(lambda)", lwd=2, col="blue")
```



(c)

```
library(geometry)
freq = c(229, 211, 93, 35, 7, 1)
hit = 0:5
sum_yi = dot(freq, hit)

likelihood <- function(lambda) {
  return(lambda^sum_yi * exp(-567*lambda))
}
optimize(likelihood, interval = c(0.6,1.2), maximum = TRUE)
```

```
## $maximum
## [1] 0.9435488
##
## $objective
## [1] 1.4282e-246
```

The maximum likelihood estimate of λ is 0.9435488.

(d)

```r
library(geometry)
freq = c(229, 211, 93, 35, 7, 1)
hit = 0:5
sum_yi = dot(freq, hit)

likelihood <- function(lambda) {
  return(lambda^sum_yi * exp(-567*lambda))
}

L0 = likelihood(1)
L1 = likelihood(0.9435488) #MLE of lambda from part (c)
ratio= -2 * log(L0/L1)
ratio
```

```
## [1] 1.840964
```

```r
qchisq(0.95, df=1)
```

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

Since qchisq(0.95, df=1) $>$ ratio, we failed to reject H_0.