STAC53 Assignment 1.

1)

a) MGF – Let $(Y_1, Y_2, \ldots Y_k) \sim$ Multinomial $(\pi, \pi_1, \ldots \pi_k)$

$$\Rightarrow P(Y_1 = y_1, \ldots Y_k = y_k) = \frac{n!}{\prod_{i=1}^{k} y_i!} \prod_{i=1}^{k} \pi_i^{y_i}$$

where $y_i$ are nonnegative integers and $\pi_i$ are constants with $\pi_i > 0$. s.t.

$$\sum_{i=1}^{k} \pi_i = 1 \quad \& \quad \sum_{i=1}^{k} y_i = n$$

$$\Rightarrow y_k = n - y_1 - y_2 - y_3 \ldots - y_{k-1}$$

Therefore the MGF is, $M(t_1, t_2, \ldots, t_{k-1})$

$$M(t_1, t_2, \ldots, t_{k-1}) = E(e^{t_1 y_1 + t_2 y_2 + \ldots + t_{k-1} y_{k-1}})$$

$\therefore$ we get

$$P(Y_1 = y_1, \ldots, Y_k = y_k) = \frac{n! \prod_{i=1}^{k} \pi_i^{y_i}}{y_1! y_2! \ldots (n - y_1 - y_2 - \ldots - y_{k-1})!}$$

$$\rightarrow = \sum_{y_1 \ldots y_{k-1}} P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k) e^{t_1 y_1 + t_2 y_2 + \ldots + t_{k-1} y_{k-1}}$$

where $\sum_{y_1 y_2 \ldots y_{k-1}} = \sum_{y_1=0}^{n} \sum_{y_2=0}^{n-y_1} \ldots \sum_{y_{k-1}=0}^{n-y_1-y_2-\ldots-y_{k-2}}$

$\therefore$

$$M(t_1, t_2, \ldots, t_{k-1}) = \sum_{y_1 \ldots y_{k-2}} \frac{n!}{y_1! \ldots y_{k-2}!} \prod_{i=1}^{k-2} (e^{t_i})^{y_i} \pi_i^{y_i} \times$$

$$\sum_{y_{k-1}=0}^{n-y_1-\ldots-y_{k-2}} \prod_{k-1}^{y_{k-1}} \frac{(1 - \pi_1 - \pi_2 \ldots \pi_{k-1})^{y_k}}{y_{k-1}! \, y_k!}$$

$$M(t_1 \ldots t_{k-1}) = \sum_{y_1 \ldots y_{k-2}} \frac{n!}{y_1! \ldots y_{k-2}!} \prod_{i=1}^{k-2} (e^{t_i} \pi_i)^{y_i} \times \frac{1}{(n - y_1 - y_2 \ldots y_{k-2})!}$$

$$\times \sum_{y_{k-1}=0}^{n-y_1-\ldots-y_{k-2}} e^{t_{k-1}} \frac{(n - y_1 - y_2 \ldots y_{k-2})! \, \pi_{k-1}^{y_{k-1}} (1 - \pi_1 - \pi_2 - \pi_3 \ldots \pi_{k-1})^{y_k}}{y_{k-1}! \, (n - y_1 - y_2 \ldots y_{k-1})!}$$

(2)

$$\sum_{y_{k-1}=0}^{n-y_1-y_{k-1}} \frac{(n-y_1-y_2-y_3-\dots-y_{k-2})!\,(\pi_{k-1}e^{t_{k-1}})^{y_{k-1}}(1-\pi_1-\pi_2-\dots-\pi_{k-1})^{n-y_1-y_{k-1}}}{y_{k-1}!\,(n-y_1-y_2-y_3-\dots-y_{k-1})!}$$

let $n^* = n - y_1 - y_2 - \dots - y_{k-2}$

we then get

$$\sum_{y_{k-1}=0}^{n^*} \frac{n^*!\,(\pi_{k-1}e^{t_{k-1}})^{y_{k-1}}(1-\pi_1-\pi_2-\dots-\pi_{k-1})^{n^*-y_{k-1}}}{y_{k-1}!\,(n^*-y_{k-1})!}$$

$$\therefore \quad (e^{t_{k-1}}\pi_{k-1} + 1 - \pi_1 - \pi_2 - \dots - \pi_{k-1})^{n^*}$$

$$= (\pi_k + e^{t_{k-1}}\pi_{k-1})^{n-y_1-y_2-\dots-y_{k-2}}$$

Therefore we reduce it to,

$$M(t_1 \dots t_{k-1}) = \sum_{y_1 \dots y_{k-2}} \frac{n!}{y_1! \dots y_{k-2}!} \prod_{i=1}^{k-2} (e^{t_i}\pi)^{y_i} \times \frac{1}{(n-y_1-\dots-y_{k-2})!} \times$$

$$(\pi_k + e^{t_{k-1}}\pi_{k-1})^{n-y_1-\dots-y_{k-2}}$$

$$\therefore M(t_1 \dots t_k) = (e^{t_1}\pi_1 + \dots + \pi_{k-1}e^{t_{k-1}} + \pi_k e^{t_k})^n$$

$$= \left(\sum_{i=1}^{k} \pi e^{t_k}\right)^n$$

b) $E(Y_j) = n\pi_j$ → Prove.

$$\Rightarrow E(Y_j) = \frac{\partial}{\partial t_j} M(t_1 \dots t_{k-1}) = n\pi_j (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^n e^{t_j}$$

$$= \left[\frac{\partial^k y(t_1 \dots t_k)}{\partial t_j}\right] = \left[n\pi_j (\pi_1 + \dots + \pi_{k-1} + \pi_k) e^0\right]$$

$$= n\pi_j * 1 = n\pi_j$$

(hence proved)

c) $\text{Var}(Y_j) = n\pi_j(1-\pi_j)$ → Prove

$$E(Y_j)^2 = \frac{\partial^2}{\partial t_j^2} M(t_1 \dots t_{k-1}) = \frac{\partial}{\partial t}\left\{\frac{\partial}{\partial t} M(t_1 \dots t_{k-1})\right\}$$

$$\therefore \frac{\partial}{\partial t}\left\{n\pi_j (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-1} e^{t_j}\right\}$$

$$= n\pi_j (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-1} e^{t_j} + n\pi_j (n-1)\pi_j (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-1} e^{t_j}$$

$$= \frac{\partial^2}{\partial t^2} M(t_1, t_2 \dots t_{k-1}) = n\pi_j + n(n-1)\pi_j^2$$

$$\text{Var}(Y_j) = E(Y_j^2) - [E(Y_j)]^2$$

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

$$= n\pi_j - n\pi_j^2 = n\pi_j(1-\pi_j)$$ (hence proved)

d) $\text{Cov}(Y_i, Y_j) = -n\pi_i\pi_j$ → prove

$$\Rightarrow \text{Cov}(Y_i, Y_j) = E(Y_i Y_j) - E(Y_i)E(Y_j)$$

$$E(Y_j) = n\pi_j \quad \& \quad E(Y_i) = n\pi_i$$

$$E(Y_i Y_j) = \frac{\partial^2}{\partial t_i \partial t_j} M(t_1, t_2 \dots t_{k-1})$$

$$\hookrightarrow = \frac{\partial}{\partial t_i}\left\{\frac{\partial}{\partial t_j} M(t_1 \dots t_{k-1})\right\}$$

$$= \frac{\partial}{\partial t_i}\left\{n\pi_j (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-1} e^{t_j}\right\}$$

$$= \frac{\partial}{\partial t_i}\left\{(e^{t_1}\pi_1 + e^{t_2}\pi_2 + \dots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-1}\right\} n\pi_j e^{t_j}$$

→

(4)

$= (n-1)\pi_j * n\pi_j e^{t_i} e^{t_j} (e^{t_1}\pi_1 + e^{t_2}\pi_2 + \cdots + e^{t_{k-1}}\pi_{k-1} + \pi_k)^{n-2}$

$$\frac{\partial^2}{\partial t_i t_j} = n(n-1)\pi_i \pi_j$$

$E(Y_i Y_j) = n(n-1)\pi_i \pi_j$

$\text{Cov}(Y_i Y_j) = n(n-1)\pi_i \pi_j - n\pi_j \, n\pi_i$

$= n(n-1)\pi_i \pi_j - n^2 \pi_j \pi_j$

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

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

$= -n\pi_i \pi_j$

b. $Cor(Y_1, Y_2) = \frac{Cov(Y_1, Y_2)}{\sqrt{Var(Y_1)Var(Y_2)}}$

$= \frac{-n\pi_1\pi_2}{\sqrt{n\pi_1(1-\pi_1) \cdot n\pi_2(1-\pi_2)}}$

$= \frac{-n\pi_1\pi_2}{n\sqrt{\pi_1(1-\pi_1)\pi_2(1-\pi_2)}}$ cancell out the n

$= \frac{-\pi_1\pi_2}{\sqrt{\pi_1\pi_2\pi_2\pi_1}}$ Since there only 2 population by complement $(1-\pi_1) = \pi_2$

$= \frac{-\pi_1\pi_2}{\sqrt{\pi_1^2\pi_2^2}}$

$= \frac{-\pi_1\pi_2}{\pi_1\pi_2}$

$= -1$

Q.2

- a) - Score test -

$$z_s = \left| \frac{\hat{\pi} - \pi^*}{\sqrt{\pi^*(1-\pi^*)/n}} \right| < z_{\alpha/2}$$

$= \hat{\pi} = 5/30 = 0.167 \qquad \pi^* = 0.1$

$$z_s = \frac{0.167 - 0.1}{\sqrt{0.1(1-0.1)/30}} \approx 1.217$$

Wald test -

$$z_w = \left| \frac{\hat{\pi} - \pi^*}{\sqrt{\hat{\pi}(1-\hat{\pi})/n}} \right| < z_{\alpha/2}$$

$$\Rightarrow \frac{0.167 - 0.1}{\sqrt{0.167(1-0.167)/30}} \approx 0.9797$$

LRT

$l_1 = l(n, \hat{\pi}_0)$

$l_0 = \binom{n}{y} \pi_0^y (1-\pi_0)^{n-y}$

$l_0 \Rightarrow \binom{30}{5} (0.1)^5 (1-0.1)^{25} \approx 0.1023$

$\ln(l_0) = -2.27$

$l_1 \Rightarrow \binom{30}{5} (0.167)^5 (1-0.167)^{25} \approx 0.1921$

$\ln(l_1) = -1.65$

$\rightarrow$ LRT $\Lambda = \frac{0.1023}{0.1921} =$

$l_0 < l_1$ sign to reject $H_0$

$\chi^2 = -2\log(l_0/l_1) =$ 1.26

Pvalue → 0.26165

The test is 2 sided

P-value → for score $2 * P(z > z_s) =$ 0.2236

P-value for wald → $2 * P(z > z_w) =$ 0.3272

(6)

b) 90% wald CI

$(1-\alpha) = 0.9 \qquad \alpha = 0.1$

Std. Normal table $\quad z_{\alpha/2} = 1.645$

$$\left( \hat{\pi}_{\bullet} - 1.645 \sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}} \;,\; \hat{\pi} + 1.645 \sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}} \right)$$

$\Rightarrow$ (0.0547, 0.279)

c) 90% Score CI.

$$\left( \hat{\pi}\left(\frac{n}{n+z^2_{\alpha/2}}\right) + \frac{1}{2}\left(\frac{z^2_{\alpha/2}}{n+z^2_{\alpha/2}}\right) \pm z_{\alpha/2} \sqrt{\frac{1}{n+z^2_{\alpha/2}} \left[ \hat{\pi}(1-\hat{\pi}) \frac{n}{n+z^2_{\alpha/2}} + \frac{1}{4}\left(\frac{z^2_{\alpha/2}}{n+z^2_{\alpha/2}}\right) \right]} \right.$$

$\Rightarrow \left( .167\left(\frac{30}{30+1.645}\right) + \ldots \right.$ Just fill in $n=30$ & $z_{\alpha/2} = 1.645$ at 90% CI

$\Rightarrow$ (0.08356, 0.30493)

d) 90% Agresti Coull CI

$\rightarrow \quad \tilde{\pi} = \frac{5+2}{30+4} = \frac{7}{34} = 0.206$

$$= 0.206 \pm 1.645 \sqrt{\frac{0.206(1-0.206)}{34}} = (0.089, .320)$$

(0.089, 0.322)

2e and 2f is in the R code below

Q.3

a) $l_0 = \left( \binom{n}{y} * (\pi_0)^y (1-\pi_0)^{n-y} \right)$

$P(Y=y) = \binom{30}{5} (0.1)^5 (1-0.1)^{25}$

$l_0 = 0.1023$, $\ln(l_0) = -2.28$

b) over all possible values of $\pi$

MLE

$\hat{\pi}_{MLE} = \frac{5}{30} = 0.167$

$l_1 = P(Y=y_1) = \binom{30}{5} (0.167)^5 (1-0.167)^{25} = 0.1921$

$\ln(l_1) = -1.6496$

c) LRT $\quad H_0 : \pi = \pi_0$

$x^2 = -2 (\log(l_0) - \log(l_1))$

$= -2 ( -2.28 - (-1.6496)$

$= 1.260$

d) $x^2 \sim x^2_1$ under $H_0$ using

q chisq (0.9, 1) R command

$x^2_{0.9,1} = 2.71$

3e is in the Rcode Below

Question 4 and 5 is also below in the R code

Q6

a) 6a $H_0: \pi = \pi_0$ vs $H_1: \pi > \pi_0$

$\alpha$ - level of significance

$\hat{\pi}$ - sample proportion

Neyman Pearson's Lemma

Reject $H_0$ when $\hat{\pi} > k$

$$P(\hat{\pi} > k) = \alpha$$

large sample size, we can use normal distribution approximation.

$$P\left( \frac{\hat{\pi} - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} > \frac{k - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \right) = \alpha$$

$$Z = \frac{\hat{\pi} - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \sim N(0,1) \text{ under } H_0$$

$$= P\left( Z > \frac{k - \pi}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \right) = \alpha \quad \rightarrow (1)$$

Using standard Normal table

$$P(Z > Z_\alpha) = \alpha \quad \rightarrow (2)$$

where $Z_\alpha$ is the critical value for one sided test for $\alpha$.

now we compare (1) & (2)

$$\frac{(k - \pi_0)}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} = Z_\alpha \Rightarrow k = \pi_0 + Z_\alpha \sqrt{\frac{\pi_0(1-\pi_0)}{n}}$$

→

(10)

Now the power of the test is

$P(\hat{\pi} > k \mid \text{under } H_1) = Q(\pi)$

under $H_1 : \pi = \pi > \pi_0$

$\Rightarrow P(\hat{\pi} > k, \text{ when } \pi = \pi > \pi_0)$

$$\Rightarrow P\left( \frac{(\hat{\pi} - \pi)}{\sqrt{\frac{\pi(1-\pi)}{\sqrt{n}}}} > \frac{k - \pi}{\sqrt{\frac{\pi(1-\pi)}{n}}} \right)$$

$\Rightarrow$ Now $z = \dfrac{\hat{\pi} - \pi}{\sqrt{\frac{\pi(1-\pi)}{n}}} \sim N(0,1)$ under $H_1$

$$\Rightarrow P\left( z > \frac{k - \pi}{\sqrt{\frac{\pi(1-\pi)}{n}}} \right)$$

$$\Rightarrow P\left( z > \frac{(\pi_0 - \pi)}{\sqrt{\frac{\pi(1-\pi)}{n}}} + z_\alpha \frac{\sqrt{\pi_0(1-\pi_0)/n}}{\sqrt{\pi(1-\pi)/n}} \right)$$

$$\Rightarrow Q(\pi) = P\left( z > \frac{(\pi - \pi_0)}{\sqrt{\frac{\pi(1-\pi)}{n}}} + z_\alpha \frac{\sqrt{\pi_0(1-\pi_0)}}{\sqrt{\pi(1-\pi)}} \right)$$

Please refer below for Question 6b,6c,6d

# STAC51

```
wald <- function(prediction, n , alpha){
    z = qnorm(1-alpha/2)
    lower_b = prediction - z*sqrt((prediction*(1-prediction))/n)
    upper_b = prediction + z*sqrt((prediction*(1-prediction))/n)
    return(c(lower_b,upper_b))
}

score <- function(prediction, n , alpha){
    z = qnorm(1-alpha/2)
    lower_b = ((n*prediction+z^2/2)-z*sqrt(n*prediction*(1-prediction)
+z^2/4))/(n+z^2)
    upper_b = ((n*prediction+z^2/2)+z*sqrt(n*prediction*(1-prediction)
+z^2/4))/(n+z^2)
    return(c(lower_b,upper_b))
}

agresti <- function(y, n , alpha){
    z = qnorm(1-alpha/2)
    prediction = (y+2)/(n+4)
    lower_b =  prediction - z*sqrt((prediction*(1-prediction))/(n+4))
    upper_b =  prediction + z*sqrt((prediction*(1-prediction))/(n+4))
    return(c(lower_b,upper_b))
}

clopper <- function(y, n , alpha){
    alpha = alpha/2
    lower_b =  (1+(n-y+1)/(y*qf(alpha,2*y,2*(n-y+1))))^-1
    upper_b =  (1+(n-y)/((y+1)*qf(1-alpha,2*(y+1),2*(n-y))))^-1
    return(c(lower_b,upper_b))
}

#Q2e
x=5; n=30
#z for score test
z_score= sqrt(n)*(x/n-0.1)/sqrt(0.1*0.9)
CI = score(x/n,n,0.09)
cat(sprintf("The score test value for the given test is %g \n",z_score))

## The score test value for the given test is 1.21716

cat(sprintf("The Score's connfidence interval is (%g,%g) \n",CI[1],CI[2]))

## The Score's connfidence interval is (0.081824,0.309799)

#Q2f
#Confidence intervals for different methods
```

```r
x=5; n=30
#z for score test
z_score= sqrt(n)*(x/n-0.1)/sqrt(0.1*0.9)

CI1= wald(x/n,n,0.05)

CI3= agresti(x,n,0.05)
cat(sprintf("The Wald's connfidence interval is (%g,%g) \n",CI1[1],CI1[2]))
```

```
## The Wald's connfidence interval is (0.033308,0.300025)
```

```r
cat(sprintf("The Agresti-Coull's connfidence interval is (%g,%g)
\n",CI3[1],CI3[2]))
```

```
## The Agresti-Coull's connfidence interval is (0.0699695,0.341795)
```

```r
#3(d)
qchisq(0.9,1)
```

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

```r
cat(sprintf("The likelihood ratio test statistics should be atleast %g to be
significant"
            , qchisq(0.9,1)))
```

```
## The likelihood ratio test statistics should be atleast 2.70554 to be
significant
```

```r
#3e.
library(rootSolve)
n= 30
y = 5
phat = y/n
alpha=.1
func <- function(pi0){
  -2*(y*log(pi0)+(n-y)*log(1-pi0)-y*log(phat)-(n-y)*log(1-phat))-qchisq(1-
alpha,df=1)
}
uniroot.all(f=func, interval=c(0,1))
```

```
## [1] 0.0756936 0.2963735
```

```r
set.seed(1003599732)
#4a.
sum = 0
for (i in rbinom(100000,25,0.06))
  {if(0.06>=wald(i/25,25,.05)[1] & 0.06<=wald(i/25,25,0.05)[2]){
    sum = sum +1}}
sum/100000
```

```
## [1] 0.7834
```

I would expect for this number to be closer to .95 since we are taking the 95 percent interval, but this might be due to the fact that pi is really small so its skewed.

```
#4bi
for (i in 0:25){
  print(wald(i/25,25,0.05))}
```

```
## [1] 0 0
## [1] -0.03681459  0.11681459
## [1] -0.02634498  0.18634498
## [1] -0.007382581  0.247382581
## [1] 0.01629307 0.30370693
## [1] 0.04320288 0.35679712
## [1] 0.07258649 0.40741351
## [1] 0.1039957 0.4560043
## [1] 0.1371447 0.5028553
## [1] 0.1718435 0.5481565
## [1] 0.2079635 0.5920365
## [1] 0.2454199 0.6345801
## [1] 0.2841605 0.6758395
## [1] 0.3241605 0.7158395
## [1] 0.3654199 0.7545801
## [1] 0.4079635 0.7920365
## [1] 0.4518435 0.8281565
## [1] 0.4971447 0.8628553
## [1] 0.5439957 0.8960043
## [1] 0.5925865 0.9274135
## [1] 0.6432029 0.9567971
## [1] 0.6962931 0.9837069
## [1] 0.7526174 1.0073826
## [1] 0.813655 1.026345
## [1] 0.8831854 1.0368146
## [1] 1 1
```

```
#4bii
for (i in 0:25)
  {if(0.06>=wald(i/25,25,.05)[1] & 0.06<=wald(i/25,25,0.05)[2])
    {print(i)}}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

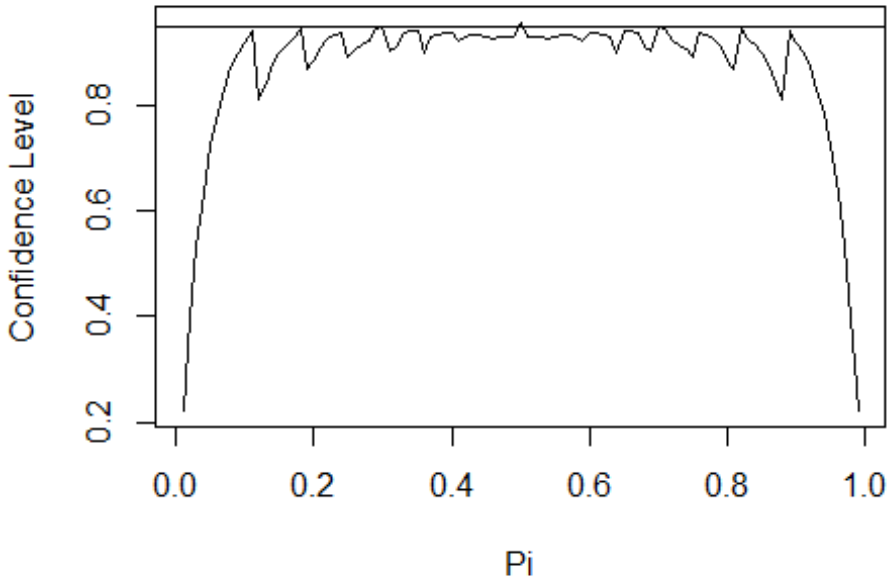
If y =1,2,3,4,5 then I(y) = 1 , and 0 other wise

```
#4iii
sum=0
for (i in 1:5){
   k = dbinom(i,25,0.06)
```

```
      sum = k + sum}
sum
```

```
## [1] 0.784026
```

```
#5a
set.seed(1003599732)
n = 25
pi = seq(.01,.99 , by= 0.01)
index = 1
confi_lvl = matrix(,nrow=99, ncol = 2)
for(p in pi){
  temp = c()
  sum = 0
  for (i in 0:25){
    ci = wald(i/25, 25, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      temp <- c(temp,i)
    }
  }

  for (c in temp){
    binomial = dbinom(c,25, p)
    sum = binomial + sum
  }

  confi_lvl[index,] = c(p,sum)
  index = index+1
}
plot(x = confi_lvl[,1], y = confi_lvl[,2], xlab = "Pi" , ylab = "Confidence
Level", type="l") + abline(h = 0.95)
```

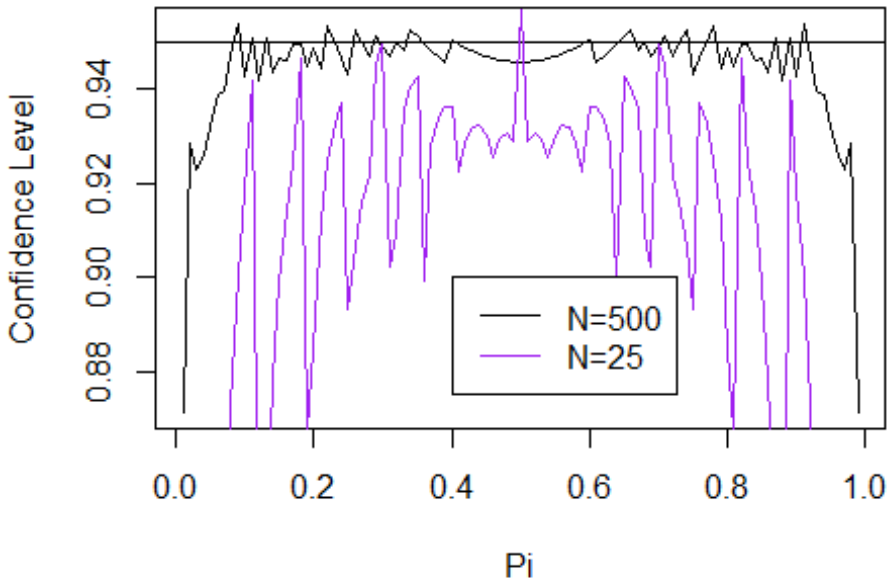



```
## integer(0)
```

I learn from my plot that as pi goes to the extreme(when pi is close to 0 or 1), the confidence level drops very fast. Which mean that as the pi level goes to the extreme we shouldn't use walds CI, since the confidence level is low.

```
#5b
set.seed(1003599732)
n = 500
pi = seq(.01,.99 , by= 0.01)
index = 1
confi_lvl500 = matrix(,nrow=99, ncol = 2)
for(p in pi){
  temp = c()
  sum = 0
  for (i in 0:500){
    ci = wald(i/500, 500, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      temp <- c(temp,i)
    }
  }
  for (c in temp){
    binomial = dbinom(c,500, p)
    sum = binomial + sum
  }
  confi_lvl500[index,] = c(p,sum)
  index= index+1
```

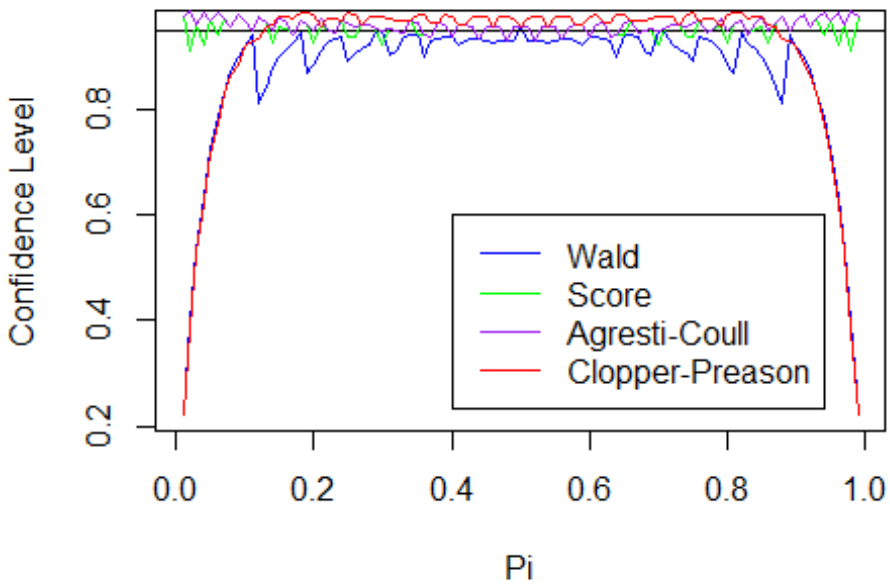
```
}
plot(x = confi_lvl500[,1], y = confi_lvl500[,2], xlab = "Pi" , ylab =
"Confidence Level", type = "l") + abline(h = 0.95)+ lines(x= confi_lvl[,1],
y= confi_lvl[,2], col= "purple")

## integer(0)

legend(.4,.9,legend=c("N=500","N=25"),col=c("black","purple"),lty=1)
```




We can see with this graph that, as N increases, the confidence level manages to stay at the 95 confidence level for much longer then when N is lower. We can see from this graph that When N = 500 it is also much more consistently around the 95 percent confidence level. Where as N=25 it varies a lot.

```
#helper code for 5c
sum = 0
sumbin <- function(list, n, p){
  for(c in list){
    binomial = dbinom(c,n, p)
    sum = binomial + sum
  }
  return(sum)
}

#5c
n = 25
pi = seq(.01,.99 , by= 0.01)
index = 1
```

```
confi_lvl = matrix(,nrow=99, ncol = 5)
confi_lvl[,1] = pi
for(p in pi){
  waldCI = c()
  for (i in 0:25){
    ci = wald(i/25, 25, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      waldCI <- c(waldCI,i)
    }
  }
  scoreCI = c()
  for (i in 0:25){
    ci = score(i/25, 25, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      scoreCI <- c(scoreCI,i)
    }
  }
  agrestiCI = c()
  for (i in 0:25){
    ci = agresti(i, 25, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      agrestiCI <- c(agrestiCI,i)
    }
  }
  clopperCI = c()
  for (i in 1:24){
    ci = clopper(i, 25, 0.05)
    if(p>=ci[1] & p<=ci[2]){
      clopperCI <- c(clopperCI,i)
    }
  }

  confi_lvl[index,2] = sumbin(waldCI,25,p)
  confi_lvl[index,3] = sumbin(scoreCI,25,p)
  confi_lvl[index,4] = sumbin(agrestiCI,25,p)
  confi_lvl[index,5] = sumbin(clopperCI,25,p)

  index= index+1
}
plot(x = confi_lvl[,1], y = confi_lvl[,2], xlab = "Pi" , ylab = "Confidence
Level", type = "l", col= "blue") + abline(h = 0.95) + lines(x=
confi_lvl[,1],y = confi_lvl[,3], col= "green")+lines(x=
confi_lvl[,1],y=confi_lvl[,4],col= "purple") + lines(x=
confi_lvl[,1],y=confi_lvl[,5], col = "red")

## integer(0)

legend(.4,.6,legend=c("Wald","Score","Agresti-Coull","Clopper-
Preason"),col=c("blue","green","purple","red"),lty=1)
```

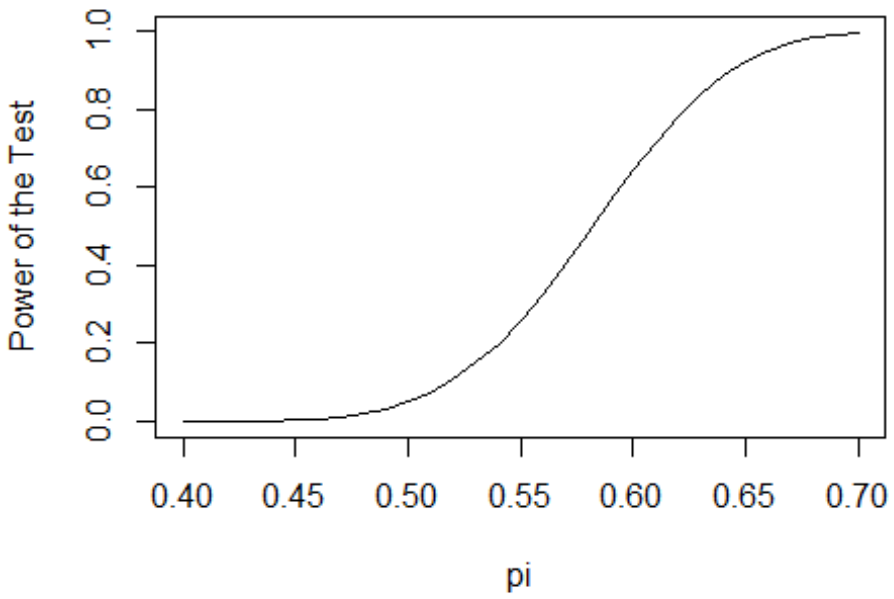



We can see from this graph that Argesti-Coull and Score's confidence level stays much closer to the 95 percent confidence level. Where as wald and clopper-pearson's confidence level drops very fast to near 20 percent when pi is at the extremes (when pi is close to 0 or 1). However when pi is not around the extreme's clopper-pearon's confidence level is the highest among all the rest.

```
power <- function(p,p_0,n,alpha){
  z = qnorm(1-alpha)
  z1 = (p_0-p)/sqrt(p*(1-p)/n)+z*sqrt((p_0*(1-p_0))/(p*(1-p)))
  pp = pnorm(z1, lower.tail = F)
  return(pp)
}

#Q6 (b)

n = 100
p = 0.55   #given alternative prob
p_0 = 0.5   #given hypothesis prob
alt = "greater"  #H_a: p>p_0
bb = power(p , p_0 , n , 0.05)
bb

## [1] 0.2584594

#for power calculation we are using exact test
```

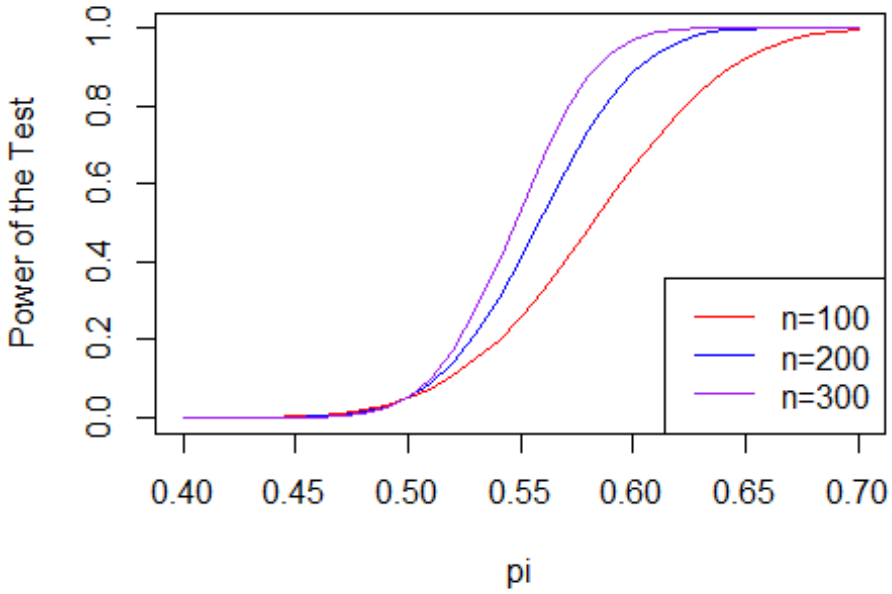
This is the probability that the test reject the null hypothesis given the null hypothesis is false. So the probability of not making a type two error. This is surprisingly low, because we would want this to be as high as possible. As we increase alpha, the power also increases.

```
#Q6 (c)
index = 1
n = 100
p = seq(0.4,0.7,0.01)  #given alternative prob
p_0 = 0.5    #given hypothesis prob

alpha = 0.01
pmatrix = matrix(,nrow=40,ncol=2)
for(i in p){
  powerlvl = power(i , p_0 , n , 0.05)
  pmatrix[index,] = c(i,powerlvl)
  index = index + 1
}

plot(pmatrix[,1],pmatrix[,2],xlab = "pi", ylab = "Power of the Test",
type="l")
```




As we increase pi, the power of the test increases. This reminds me of the logistic curve where in the begining it is nearly flat then there is exponential growth and then near the end it goes back to being flat. Which means that anything after .7 would have a high power of test and any below .4 would have a very low power of test.

```r
#Q6 (d)
p = seq(0.4,0.7,0.01)
p_0 = 0.5
alpha = 0.01

power100 = matrix(,nrow=40,ncol=2)
index = 1
for(i in p){
  powerlvl = power(i , p_0 , 100 , 0.05)
  power100[index,] = c(i,powerlvl)
  index = index + 1
}

power200= matrix(,nrow=40,ncol=2)
index = 1
for(i in p){
  powerlvl = power(i , p_0 , 200 , 0.05)
  power200[index,] = c(i,powerlvl)
  index = index + 1
}

power300= matrix(,nrow=40,ncol=2)
index = 1
for(i in p){
  powerlvl = power(i , p_0 , 300 , 0.05)
  power300[index,] = c(i,powerlvl)
  index = index + 1
}

plot(power100[,1],power100[,2],xlab = "pi", ylab = "Power of the Test",
type="l", col = "red") + lines(power200[,1],power200[,2], col = "blue")
+lines(power300[,1],power300[,2], col = "purple")

## integer(0)

legend("bottomright",legend=c("n=100","n=200","n=300"),col=c("red","blue","pu
rple"),lty=1)
```




From this graph we can see that as we increase n, the power of the test increases much faster. Which means when we have more samples the power of the test is also much higher, so our predictions are much more accurate. Making sure we don't get type two errors.