STAT400 - Homework 8

Your Name

Due 11/12/2020 by 4pm

Be sure to `set.seed(400)` at the beginning of your homework.

```
#reproducibility
set.seed(400)
```

```
# useful libraries
library(tidyverse)
```

1. Sign up for a GitHub account (https://github.com) and request an educational account (https://education.github.com/benefits). What is your GitHub user name?

2. *Coverage for a two-sided CI for a proportion π.*

   If you have a sample of data that consists of 0's and 1's, you may want to estimate the proportion of 1' based on the sample. In this problem we will compare the properties of two different estimators of the proportion π. The goal of the problem is to compare the coverage for confidence intervals computed using the two different estimators.

   Let $p = $ the proportion of 1's in a sample of 0's and 1's. So, p estimates π. Compute the coverage for a 95% CI for π using the approaches below.

   95% Confidence Intervals for π:

   Method 1: Standard approach - use $p \pm z_{0.975}\sqrt{p(1-p)/n}$, where $z_{0.975}$ = the 0.975 quantile from a $N(0,1)$.

   Method 2: This method uses a different estimator for π. First, add 2 successes and 2 failures to your data and then use the interval from Method 1. Note that you need to adjust both p and n from Method 1.

   For the following, let $m = 1000$ in your Monte Carlo estimations.

   a. Simulate $n = 20$ observations from a Bernoulli distribution with $\pi = 0.05$. Compare the empirical coverage for methods 1 and 2. Use the same data to compare methods 1 and 2.

```
# number of MC iterations
m <- 1000
n<-20
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(20,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}

# n = 20, pi = 0.05
```

```
# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.645
```

```
# method 2
m <- 1000
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(20,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}
```

```
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.989
```

b. Simulate $n = 100$ observations from a Bernoulli distribution with $\pi = 0.05$. Compare the empirical coverage for methods 1 and 2. Use the same data to compare methods 1 and 2.

```
m <- 1000
n<-100
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}

# n = 100, pi = 0.05
# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.874
```

```
# method 2
m <- 1000
samples<-rep(NA,1000)
```

```
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}
```

```
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.971
```

c. Repeat problems a. and b. but set $\pi = 0.1$ when you simulate the data.

```
#n=20 and pi=0.1
m <- 1000
n<-20
samples<-rep(NA,1000)
i=1
pi=0.1
while(i<=1000){
  x<-rbinom(n,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}


# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.653
```

```
# method 2
m <- 1000
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}
```

```
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.981
```

```
# n = 100, pi = 0.1
  m <- 1000
n<-100
samples<-rep(NA,1000)
i=1
pi=0.1
while(i<=1000){
  x<-rbinom(n,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}

# n = 20, pi = 0.5
# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.39
```

```
# method 2
m <- 1000
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}
```

```
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.974
```

d. Repeat problems a. and b. but set $\pi = 0.5$ when you simulate the data.

```
# n = 20, pi = 0.5
 m <- 1000
n<-20
samples<-rep(NA,1000)
```

```r
i=1
pi=0.5
while(i<=1000){
  x<-rbinom(n,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}

# n = 20, pi = 0.05
# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0
```

```r
# method 2
m <- 1000
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}



 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.987
```

```r
# n = 100, pi = 0.5
  m <- 1000
n<-100
samples<-rep(NA,1000)
i=1
pi=0.5
while(i<=1000){
  x<-rbinom(n,1,0.05)
  samples[i]<-mean(x)
  i=i+1
}

# n = 20, pi = 0.05
# method 1
#calculate the ci and take the means of the ci
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
```

```
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0
```

```
# method 2
m <- 1000
samples<-rep(NA,1000)
i=1
pi=0.05
while(i<=1000){
  x<-rbinom(n,1,0.05)
  add <-c(1,1,0,0)
  x<-append(x,add)
  samples[i]<-mean(x)
  i=i+1
}
```

```
 ci <- cbind(samples-qnorm(0.975)*sqrt(samples*(1-samples)/n), samples+qnorm(0.975)*sqrt(samples*(1
y <- ci[,1] <= pi & ci[,2] >= pi
 print(mean(y))
```

```
## [1] 0.966
```

e. Summarize your findings in a table. Which method do you recommend based on these results?

```
# make a table of results
table <- matrix(c(0.633,0.981,0.877,0.976,0.648,0.989,0.366,0.971,0,0.98,0,0.972),ncol=2,byrow = TR
colnames(table)<-c("Method 1","Method 2")
rownames(table)<-c("n=20 , pi=0.05","n=100 , pi=0.05","n=20 , pi=0.1","n=100 , pi=0.1","n=20 , pi=0
table <- as.table(table)
table
```

```
##                  Method 1 Method 2
## n=20 , pi=0.05      0.633    0.981
## n=100 , pi=0.05     0.877    0.976
## n=20 , pi=0.1       0.648    0.989
## n=100 , pi=0.1      0.366    0.971
## n=20 , pi=0.5       0.000    0.980
## n=100 , pi=0.5      0.000    0.972
```

Based on the results on the table I would choose method 2, as it has higher coverage across the board.

Note: Method 2 is from the article “Approximate Is Better than”Exact" for Interval Estimation of Binomial Proportions," by Alan Agresti and Brent A. Coull, *The American Statistician*, Vol. 52, No. 2 (May, 1998), pp. 119-126