Quiz 5

Lingjun Fu

05 October, 2015

Question 1:

Getting a true simple random sample is expensive. Suppose the pollster could only affordto poll 17 people. Let X be the number of Clinton supporters in the poll and assume X ~ Bin(17, p). Suppose x = 9. Using Stata's cii command, create three 95% confidence intervals for p: a Wald asymptotic Normal interval, a Wilson score interval, and an Exact interval. Note which interval Stata uses by default.

The codes in stata are:
cii 17 9, wald level (95)
cii 17 9, wilson level (95)
cii 17 9, exact level (95)

cii 17 9, level (95) ##default

We note that Stata uses exact interval by default. The results for wald, wilson, exact intervals are: [.2921428, .7666807], [.3096324, .7383489], [.2781183, .7701673],

---

Question 2:

Suppose the pollster got more money and polled 40 people finding x = 20. Using the binconf() command in the Hmisc package, create 95% CI's for this data using the same three methods. Note which interval Hmisc uses by default.

```
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
binconf(20, 40, alpha=0.05, method="asymptotic", include.x=FALSE, include.n=FALSE, return.df=FALSE)
```

```
##  PointEst     Lower     Upper
##       0.5 0.3450512 0.6549488
```

```r
binconf(20, 40, alpha=0.05, method="wilson", include.x=FALSE, include.n=FALSE, return.df=FALSE)
```

```
##  PointEst     Lower     Upper
##       0.5 0.3519953 0.6480047
```

```r
binconf(20, 40, alpha=0.05, method="exact", include.x=FALSE, include.n=FALSE, return.df=FALSE)
```

```
##  PointEst     Lower     Upper
##       0.5 0.3380178 0.6619822
```

```r
binconf(20, 40, alpha=0.05, include.x=FALSE, include.n=FALSE, return.df=FALSE) ##default
```

```
##  PointEst     Lower     Upper
##       0.5 0.3519953 0.6480047
```

We note that Hmisc uses Wilson score interval by default.

Question 3:

We call all three of these intervals 95% confidence intervals. How accurate is that nomenclature? Using the rbinom() command in R, construct and perform a simulation study of the true coverage rates of these three intervals when X ~ Bin(n, 0.50) for n = 17 and n = 40. Discuss.

```r
Q3 <- function(n, methods, sample_size=1000){
    count <- 0
    temp <- rbinom(sample_size, n, 0.5)
    interval <- binconf(temp, n, alpha=0.05, method=methods)
    for(j in 1:length(temp)){
        if(interval[j,2] <= 0.5 & interval[j,3] >= 0.5){
            count = count + 1
        }
    }
    res <- count/sample_size
    return (res)
}
Q3(17, "wilson", 100000)
```

```
## [1] 0.95095
```

```r
Q3(17, "exact", 100000)
```

```
## [1] 0.95126
```

```r
Q3(17, "asymptotic", 100000)
```

```
## [1] 0.95078
```

```r
Q3(40, "wilson", 100000)
```

```
## [1] 0.96172
```

```r
Q3(40, "exact", 100000)
```

```
## [1] 0.96191
```

```r
Q3(40, "asymptotic", 100000)
```

```
## [1] 0.91925
```

```r
# Q3(80, "wilson", 100000)
# Q3(80, "exact", 100000)
# Q3(80, "asymptotic", 100000)
```

We choose a sample size of 100,000 so that only n is the dominant factor affecting accuracy. We see that all three intervals are close to 0.95 when $n = 17$. However, the asymptotic interval behaves poorly when n increases to 40 while wilson and exact intervals are still close to 0.95. Overall, the exact and wilson intervals have better accuracy. The reason is that the central limit theorem applies poorly to this distribution with small n, hence normal approximation is not good for binomial distribution. In other words, the range of n between [17, 40] is not statistically large. So, the accuracy of approximation does not increase monotonically as n increases.

Question 4:

You can get decent precision taking lots of samples with rbinom(), but you can actually get machine level precision fairly easily using the probabilities of the binomial distribution, i.e. the density function dbinom(). The trick is to think of all n+1 possible outcomes for X, calculate the CI for each outcome, create a variable C (coverage) which = 1 if the CI contains the true p and = 0 otherwise, andcalculate the expectation of C using the known density function of the binomial, e.g. E[C|n=17,p=0.5].Revisit question 03 using this precise approach

```r
Q4 <- function(n, methods, p){
    cov <- 0
    for(i in (0:n)){
        tempCI <- binconf(i, n, alpha=0.05, method=methods)
        if(tempCI[2] <= p & tempCI[3] >= p){
            cov = cov + 1*dbinom(i, n, p)
        }
    }
    return (cov)
}

Q4(17, "exact", 0.5)
```

```
## [1] 0.9509583
```

```
Q4(17, "wilson", 0.5)
```

```
## [1] 0.9509583
```

```
Q4(17, "asymptotic", 0.5)
```

```
## [1] 0.9509583
```

```
Q4(40, "exact", 0.5)
```

```
## [1] 0.9615227
```

```
Q4(40, "wilson", 0.5)
```

```
## [1] 0.9615227
```

```
Q4(40, "asymptotic", 0.5)
```

```
## [1] 0.9193095
```

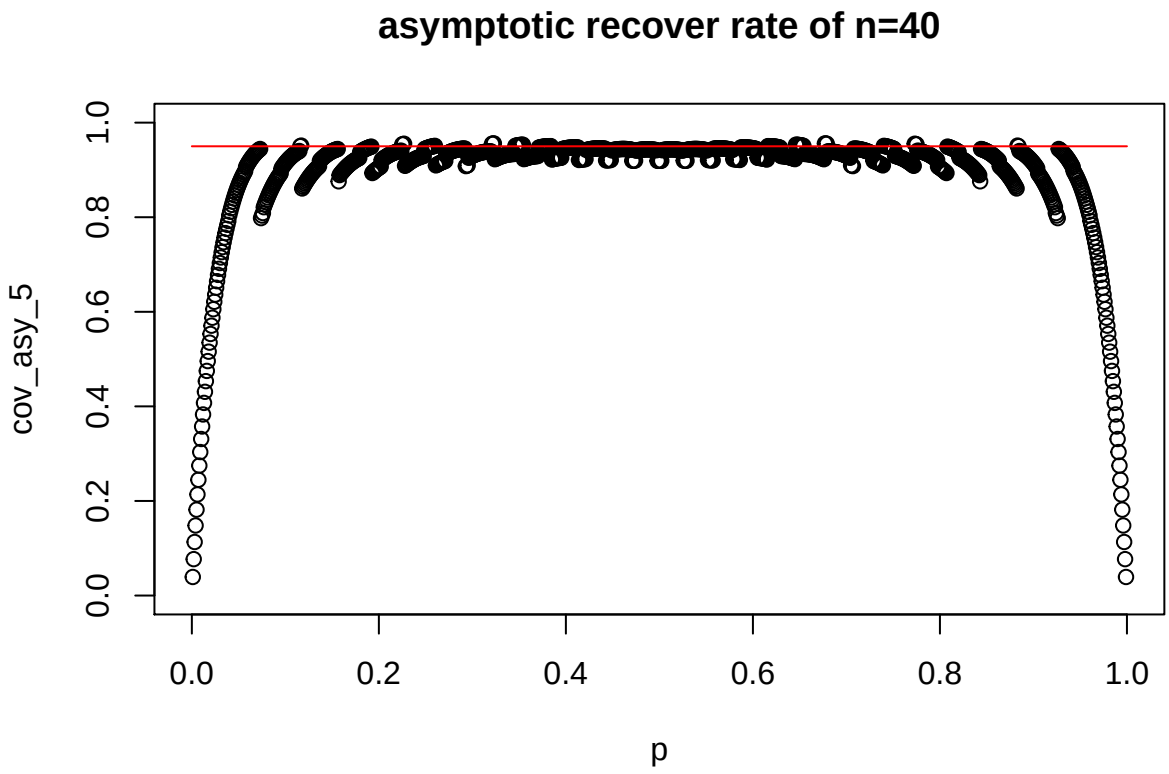
In this precise approach, we have exactly the same accuracy for three intervals when $n = 17$. However, the asymptotic interval behaves badly again when n increases to 40. We note that wilson and exact intervals have the same recover rate regardless of the value of n. The reason roots in the discontinuous nature of the binomial distribution. Note that we calculate the recover rate averaged over $n + 1$ possible discrete outcomes. For each possible outcome, wilson and exact intervals may have different upper and lower bounds, but they have the same probability to cover p.

Question 5:

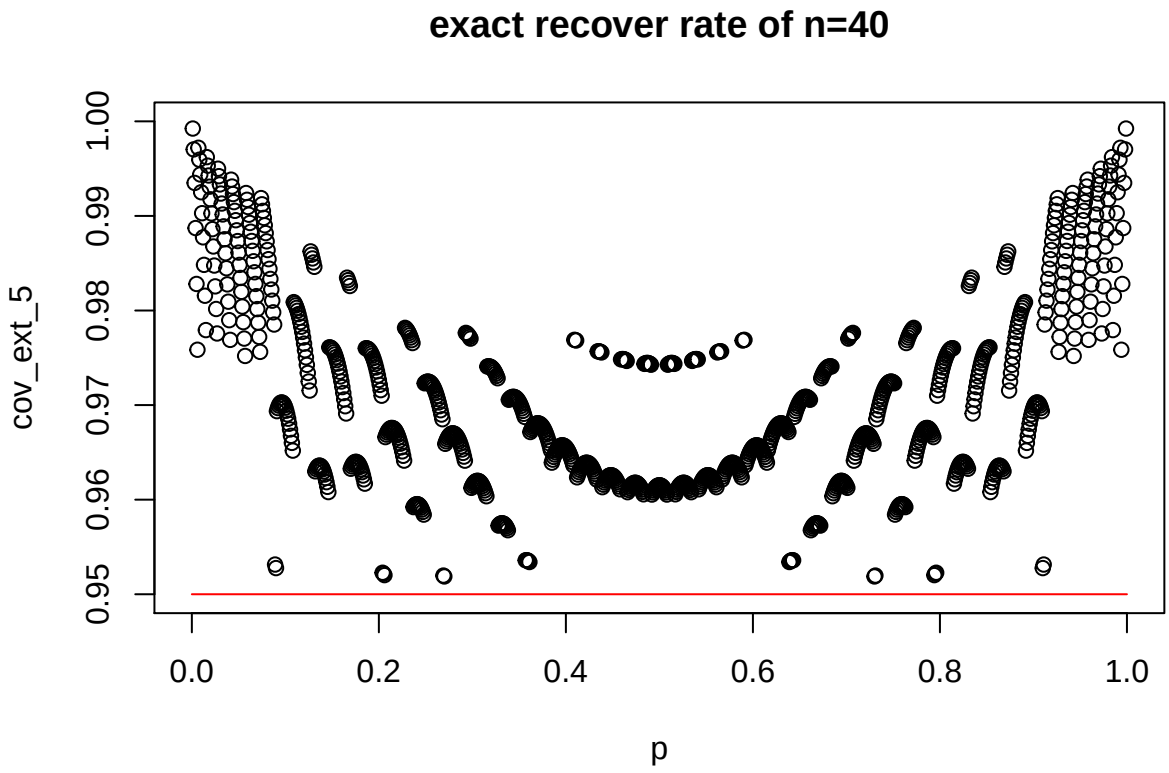
Now rather than varying n, fix n = 40 and vary p, i.e. let X ~ Bin(40, p). Create a plot of coverage rate by p for p in (0, 1) for the three methods. Describe any unusual behavior and the relative performance of the methods. Take a fine gradation of p, e.g. delta <- 0.001; p <- seq(delta,1-delta,delta);. Which interval method would you pick for this setting of a smallish n and unknown p

```
delta <- 0.001
p <- seq(delta, 1-delta, delta)
Q5 <- function(p, methods, n=40){
    cov_rate <- rep(0, length(p))
    for(i in 1:length(p)){
        cov_rate[i] <- Q4(n, methods, p[i])
    }
    return (cov_rate)
}
cov_asy_5 <- Q5(p, "asymptotic")
```

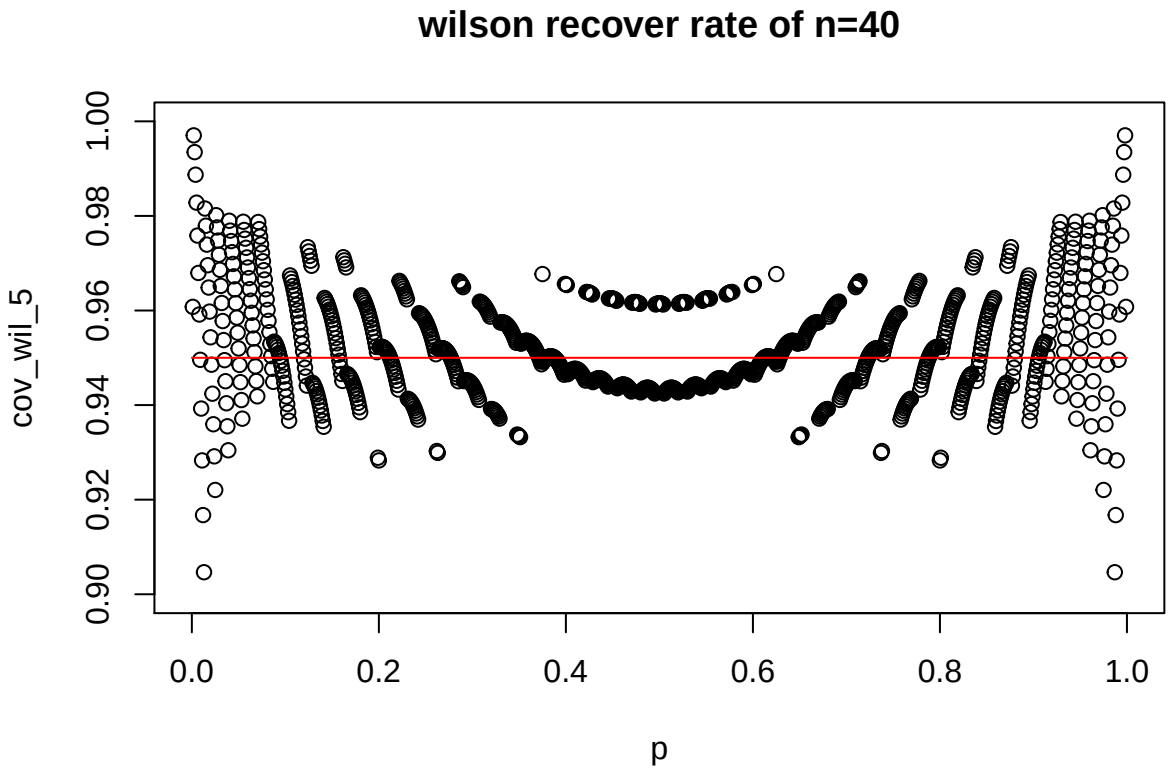
```r
cov_ext_5 <- Q5(p, "exact")
cov_wil_5 <- Q5(p, "wilson")

plot(p, cov_asy_5, type='p', xlim=c(0,1), ylim=c(0,1), main="asymptotic recover rate of n=40")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")
```




```r
plot(p, cov_ext_5, type='p', xlim=c(0,1), ylim=c(0.95,1), main="exact recover rate of n=40")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")
```




```
plot(p, cov_wil_5, type='p', xlim=c(0,1), ylim=c(0.9,1), main="wilson recover rate of n=40")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")
```



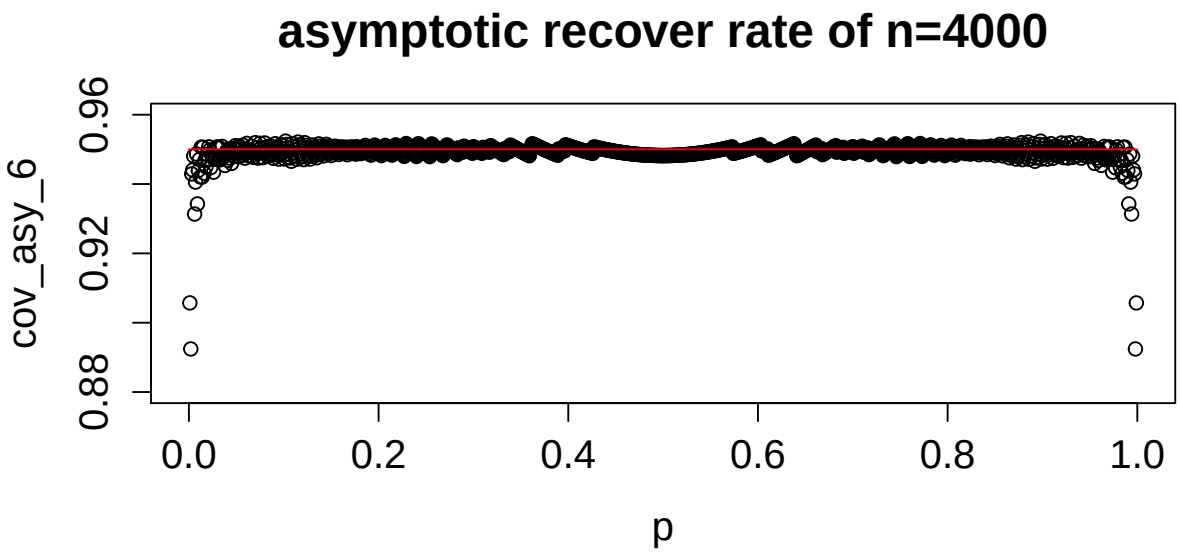
The asymptotic interval totally fails when p is close to extreme values (0 or 1). The true recover rate of exact interval is close to but well above 0.95. The true recover rate of wilson interval is close to 0.95 and it can be either wider or narrower. All three intervals tend to be more stable (less oscillation) when p approaches to 0.5 from both sides (0 or 1). For small n and unknown p, I would pick wilson since it has better precision than asymptotic when p is at extreme values and it has less weird points (cover rate $> 98\%$) than exact.

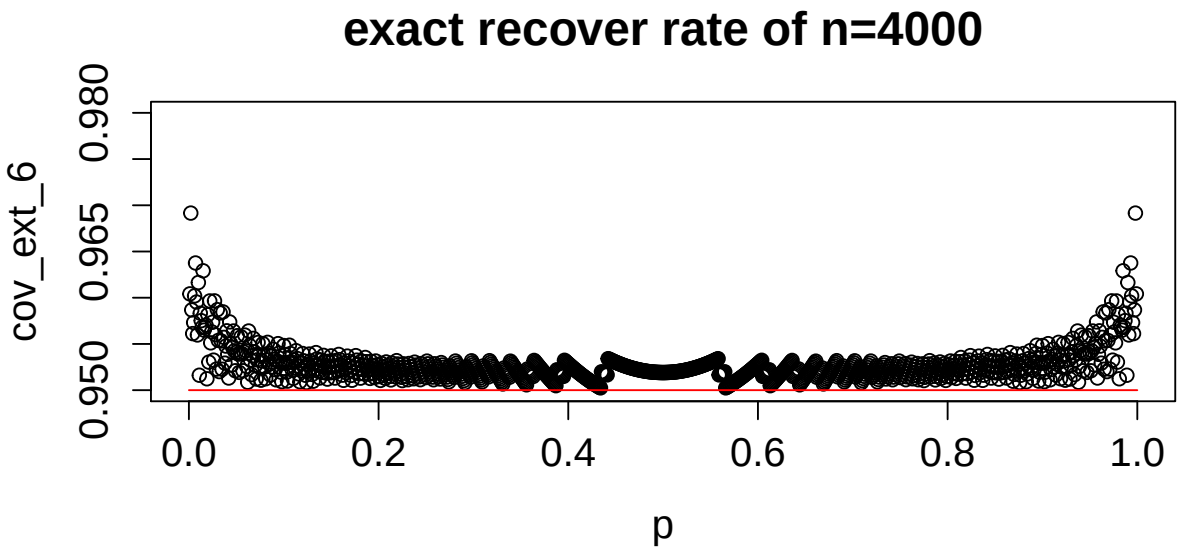
Question 6:

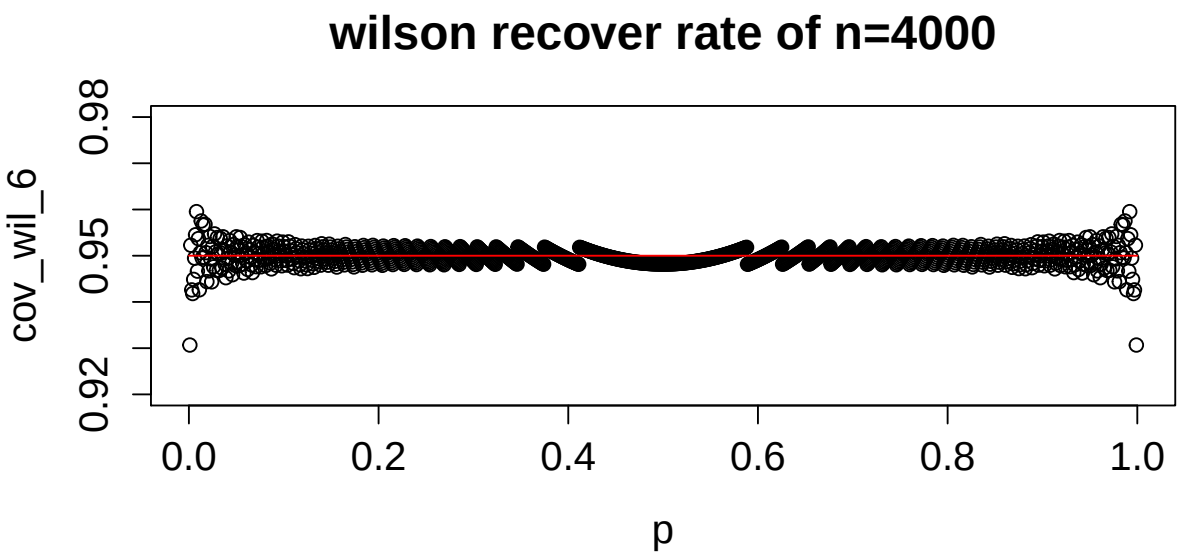
Suppose the pollster's company had money to burn and could take a huge sample. Revisit question 05 for n = 4000. Describe any unusual behavior and the relative performance of the methods. Comment on any differences in your findings from question 05. Which interval method would you pick for this setting of a smallish n and unknown p?

```
cov_asy_6 <- Q5(p, "asymptotic", 4000)
cov_ext_6 <- Q5(p, "exact", 4000)
cov_wil_6 <- Q5(p, "wilson", 4000)

plot(p, cov_asy_6, type='p', xlim=c(0,1), ylim=c(0.88,0.96), main="asymptotic recover rate of n=4000")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")

plot(p, cov_ext_6, type='p', xlim=c(0,1), ylim=c(0.95,0.98), main="exact recover rate of n=4000")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")
```

```r
plot(p, cov_wil_6, type='p', xlim=c(0,1), ylim=c(0.92,0.98), main="wilson recover rate of n=4000")
lines(x=seq(0,1,0.001),y=rep(0.95,length(seq(0,1,0.001))),col="red")
```







We see that all of the three intervals become more stable as n increases from 40 to 4000. In particular, asymptotic has much better performace at extreme values of p although it's still the worst of the three. Exact interval keeps overestimating 0.95 while wilson interval keeps oscillating around 0.95. Both of these two intervals have few weired points (cover rate largely deviated from 0.95). Therefore, I will pick either wilson or exact interval when n is large and p is unknown. If you really wants me to pick one, I may go with the exact one since it can offer a conservative cover rate of at least 0.95.
