

HW1

ShanZhong

January 23, 2019

```
library("ggplot2")

cover<-function(pnull,n,alpha){
  phat<-rbinom(1000,n,pnull)/n
  sd<-sqrt(phat*(1-phat)/n)
  CI<-sd*qnorm(1-alpha/2,0,1)
  rate<-sum((pnull>phat-CI)&(pnull<phat+CI))/1000
  return(rate)
}

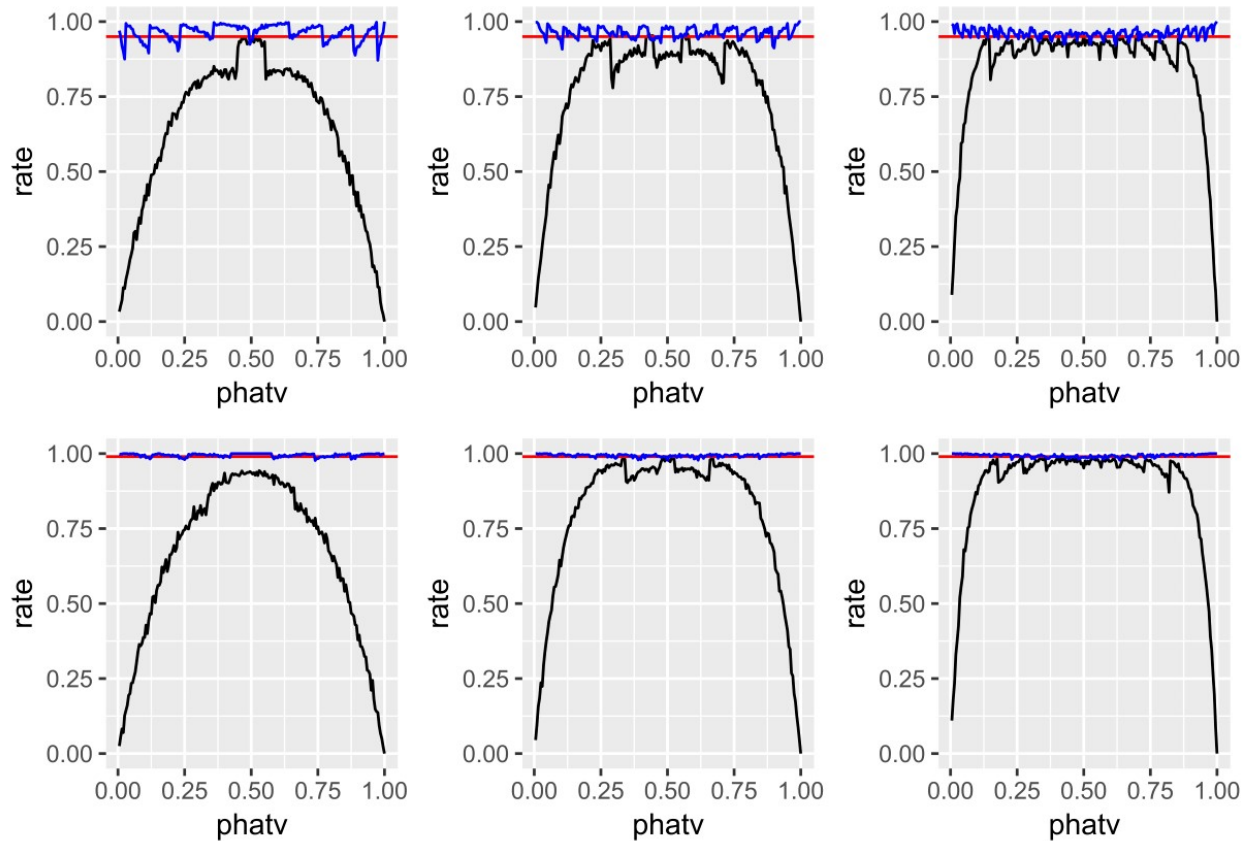
adjcover<-function(pnull,n,alpha){
  phat<-(rbinom(1000,n,pnull)+2)/(n+4)
  sd<-sqrt(phat*(1-phat)/(n+4))
  CI<-sd*qnorm(1-alpha/2,0,1)
  rate<-sum((pnull>phat-CI)&(pnull<phat+CI))/1000
  return(rate)
}

shanplot<-function(n,alpha){
  phatv<-seq(0.005,1, by = 0.005)
  rate<-rep(NA,length(phatv))
  adjrate<-rep(NA,length(phatv))
  for(i in 1:length(phatv)){
    phat<-phatv[i]
    rate[i]<-cover(phat,n,alpha)
    adjrate[i]<-adjcover(phat,n,alpha)
  }
  data<-data.frame(cbind(phatv,rate,adjrate))
  output<-ggplot(data,aes(phatv,rate))+geom_line()+ylim(0,1)+
    geom_hline(yintercept=1-alpha,color="red")+
    geom_line(aes(phatv,adjrate),color="blue")
  return(output)
}

p1<-shanplot(5,0.05)
p2<-shanplot(10,0.05)
p3<-shanplot(20,0.05)
p4<-shanplot(5,0.01)
p5<-shanplot(10,0.01)
p6<-shanplot(20,0.01)

library("gridExtra")

grid.arrange(p1, p2, p3,p4,p5,p6 ,nrow = 2)
```



```

n1<-sample(10:30,10000,replace=TRUE)
n2<-sample(10:30,10000,replace=TRUE)
p1<-runif(10000)
p2<-runif(10000)

twocover<-function(p1,p2,n1,n2,pseudo){
  truediff<-p1-p2
  psim1<-(rbinom(1000,n1,p1)+pseudo/4)/(n1+pseudo/2)
  psim2<-(rbinom(1000,n2,p2)+pseudo/4)/(n2+pseudo/2)
  simdiff<-psim1-psim2
  sd<-sqrt( psim1*(1-psim1)/n1 + psim2*(1-psim2)/n2 )
  CI<-sd*qnrm(1-0.05/2,0,1)
  rate<-sum((truediff>simdiff-CI)&(truediff<simdiff+CI))/1000
  return(rate)
}

prob<-matrix(NA,10000,ncol=5)
pseudo<-c(0,2,4,6,8)

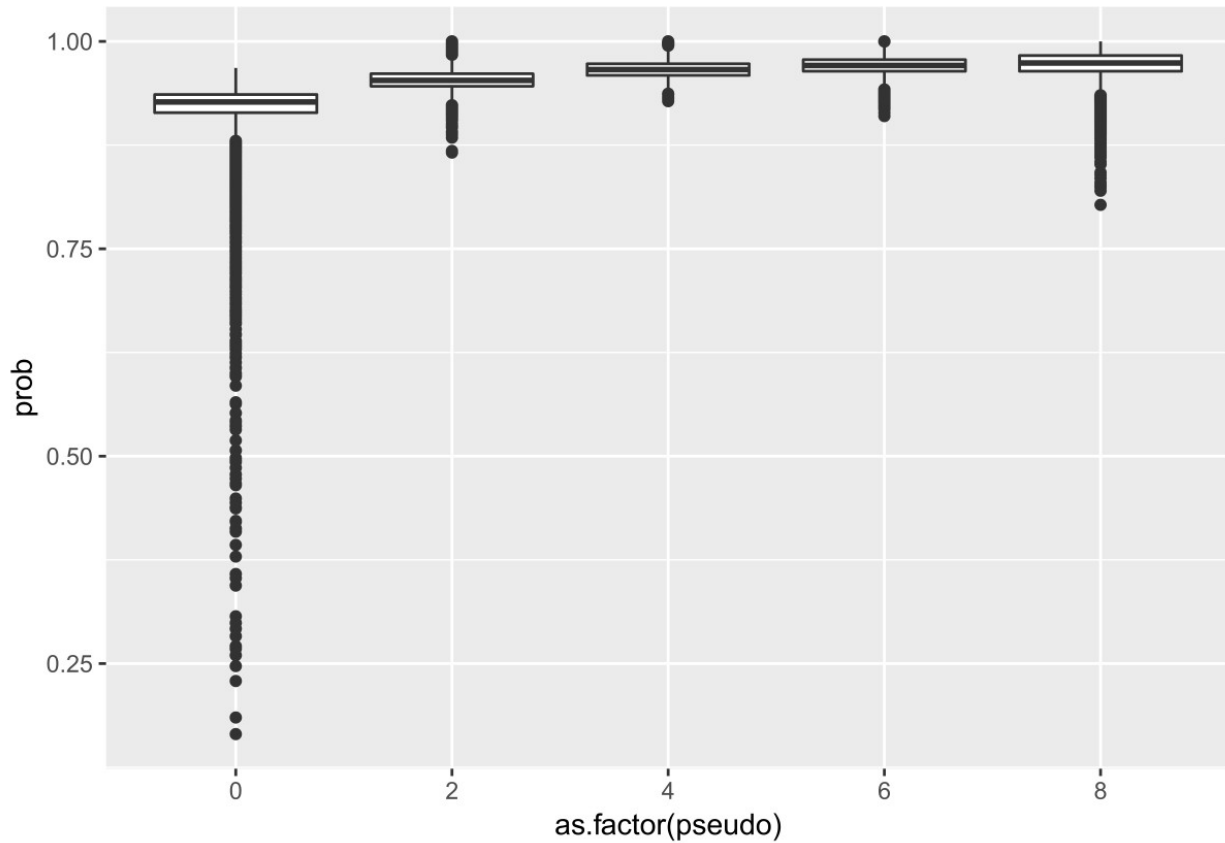
for(j in 1:5){
  for(i in 1:10000){
    prob[i,j]<-twocover(p1[i],p2[i],n1[i],n2[i],pseudo[j])
  }
}

prob<-c(prob)

```

```
pseudo<-c(rep(0,10000),rep(2,10000),rep(4,10000),rep(6,10000),rep(8,10000))
prob<-data.frame(cbind(prob,pseudo))
```

```
ggplot(prob,aes(y=prob,x=as.factor(pseudo)))+geom_boxplot()
```



```
twocover<-function(p1,p2,n1,n2,pseudo){
  truediff<-p1-p2
  psim1<-(rbinom(1000,n1,p1)+pseudo/4)/(n1+pseudo/2)
  psim2<-(rbinom(1000,n2,p2)+pseudo/4)/(n2+pseudo/2)
  simdiff<-psim1-psim2
  sd<-sqrt( psim1*(1-psim1)/n1 + psim2*(1-psim2)/n2 )
  CI<-sd*qnorm(1-0.05/2,0,1)
  rate<-sum((truediff>simdiff-CI)&(truediff<simdiff+CI))/1000
  return(rate)
}
```

```
shanplot<-function(p1,p2,n,alpha){
  rate<-rep(NA,length(p1))
  adjrate<-rep(NA,length(p1))
  for(i in 1:length(p1)){
    p1hat<-p1[i]
    p2hat<-p2[i]
    rate[i]<-twocover(p1hat,p2hat,n,n,0)
    adjrate[i]<-twocover(p1hat,p2hat,n,n,4)
  }
  data<-data.frame(cbind(p1,rate,adjrate))
}
```

```

    output<-ggplot(data,aes(p1,rate))+geom_line()+ylim(0,1)+
      geom_hline(yintercept=1-alpha,color="red")+
      geom_line(aes(p1,adjrate),color="blue")
    return(output)
  }

library("ggplot2")
p1<-seq(0.005,1, by = 0.005)
p2<-p1
g1<-shanplot(p1,p2,10,0.05)

p1<-seq(0.2,1, by = 0.005)
p2<-p1-0.2
g2<-shanplot(p1,p2,10,0.05)

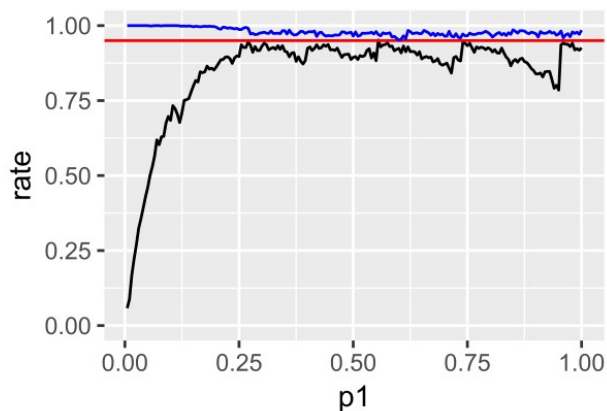
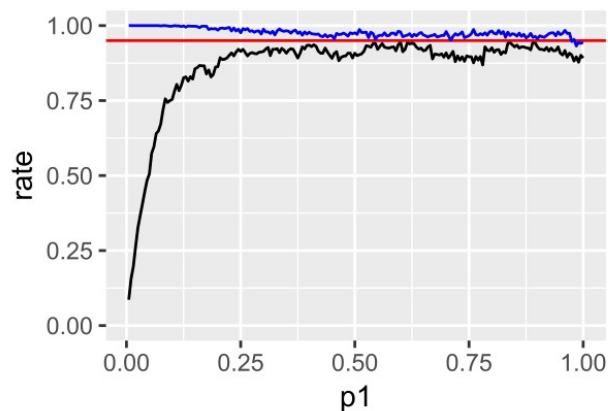
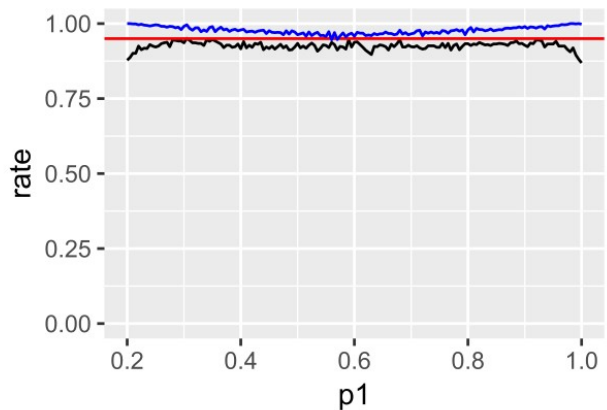
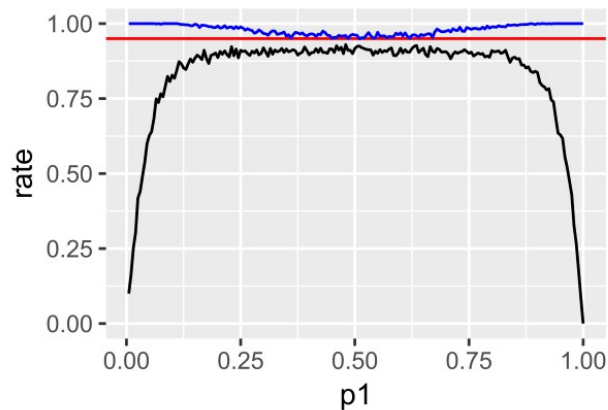
p1<-seq(0.005,1, by = 0.005)
p2<-p1/2
g3<-shanplot(p1,p2,10,0.05)

p1<-seq(0.005,1, by = 0.005)
p2<-p1/4
g4<-shanplot(p1,p2,10,0.05)

library("gridExtra")

grid.arrange(g1, g2, g3,g4,nrow = 2)

```



```
#H0:p1=p2
#H1:p1<p2

dat<-matrix(c(2,13,4,15),2)

fisher.test(dat,alternative = "two.sided")

##
## Fisher's Exact Test for Count Data
##
## data: dat
## p-value = 0.6722
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.04590206 4.89390008
## sample estimates:
## odds ratio
## 0.586089

x<-2
y<-4
n1<-15
n2<-19

p1<-x/n1
p2<-y/n2
```

```

se<-sqrt((p1*(1-p1)/n1)+(p2*(1-p2)/n2))

z<-(p2-p1)/se

pnorm(z)

## [1] 0.7263579

x<-2
y<-4
n1<-15
n2<-19
alpha1<-1
beta1<-1
alpha2<-1
beta2<-1

p1<-rbeta(1000, x + alpha1, n1 - x + beta1)
p2<-rbeta(1000, y + alpha2, n2 - y + beta2)

rd <- p2 - p1

quantile(rd, c(.025, .975))

##          2.5%          97.5%
## -0.1948100  0.3172322

plot(density(rd))

```

$\overline{p_2} \mid H_0: X < Y$

$$P(X=x, Y=y) = \binom{n_1+n_2}{x+y} p^{x+y} (1-p)^{n_1+n_2-(x+y)}$$

$$P(X=x) = \binom{n_1}{x} p^x (1-p)^{n_1-x}$$

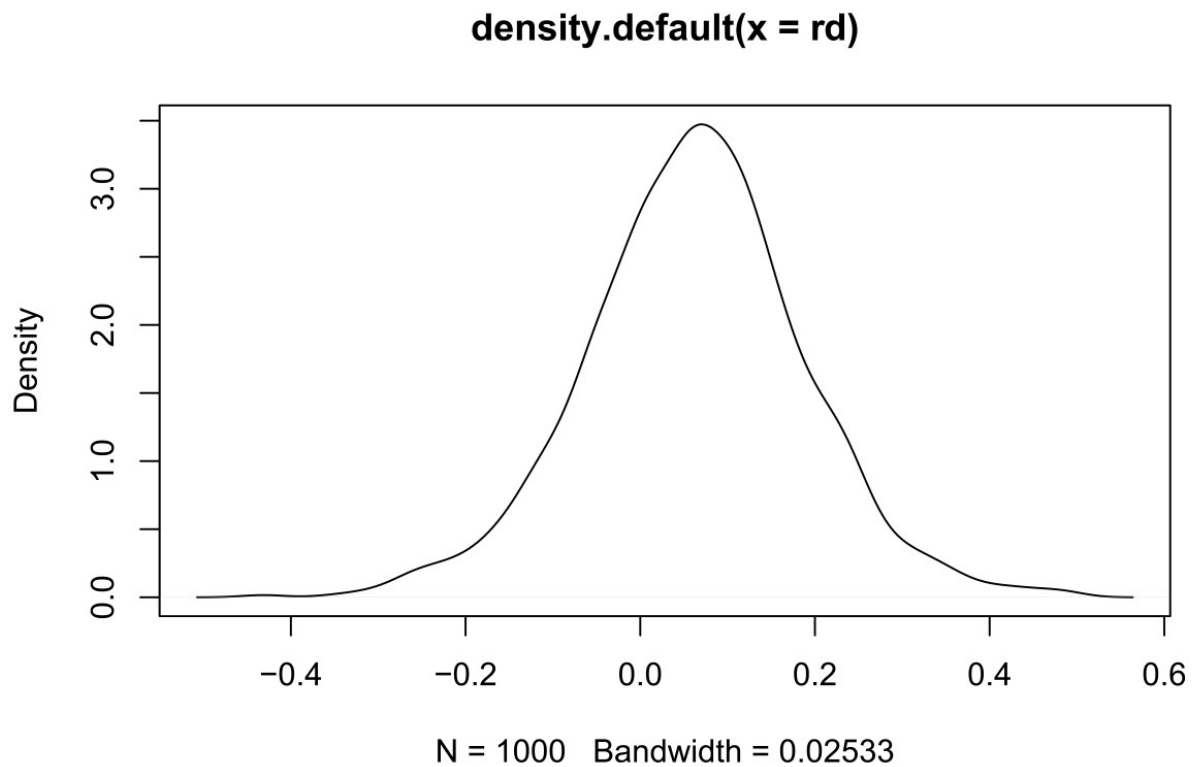
$$P(Y=y) = \binom{n_2}{y} p^y (1-p)^{n_2-y}$$

$$P(X=2, Y=4 \mid X+Y=6) = \frac{\binom{n_1}{2} \binom{n_2}{4}}{\binom{n_1+n_2}{6}} = \frac{\binom{15}{2} \binom{19}{4}}{\binom{34}{6}} = 0.3026$$

$$P(X=1, Y=5 \mid X+Y=6) = \frac{\binom{15}{1} \binom{19}{5}}{\binom{34}{6}} = 0.12969$$

$$P(X=0, Y=6 \mid X+Y=6) = 0.02017$$

$$\Rightarrow P_{\text{value}} = 0.45246$$



```
rr<-p1/p2
quantile(rr, c(.025, .975))

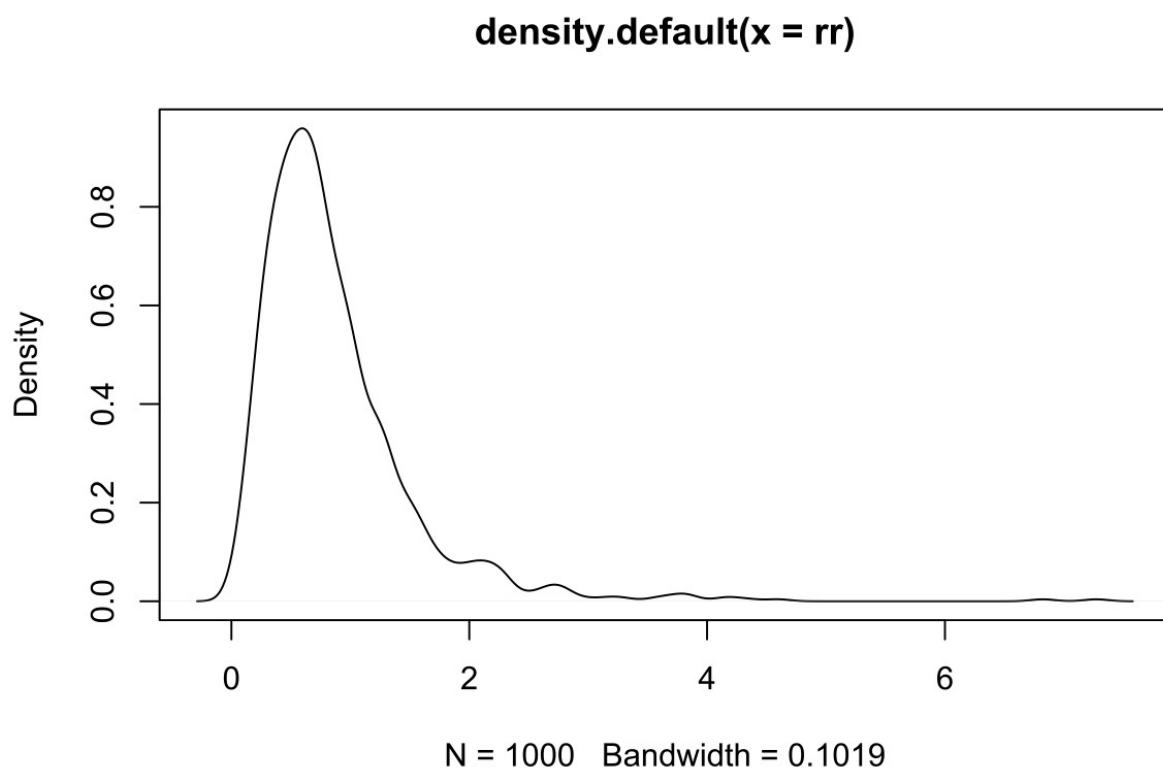
##      2.5%      97.5%
## 0.1568747 2.6028116
plot(density(rr))
```

P3 Qa

$$se \text{ of } \hat{p} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

$$g(x) = (\hat{p})^{-\frac{1}{2}} \Rightarrow g'(x) = -\frac{1}{2}(\hat{p})^{-\frac{3}{2}}$$

$$\Rightarrow se \text{ of } \sqrt{\hat{p}} = \frac{1}{2} \cdot \frac{1}{\sqrt{\hat{p}}} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$



```
odd<-(p1/(1-p1))/(p2/(1-p2))
```

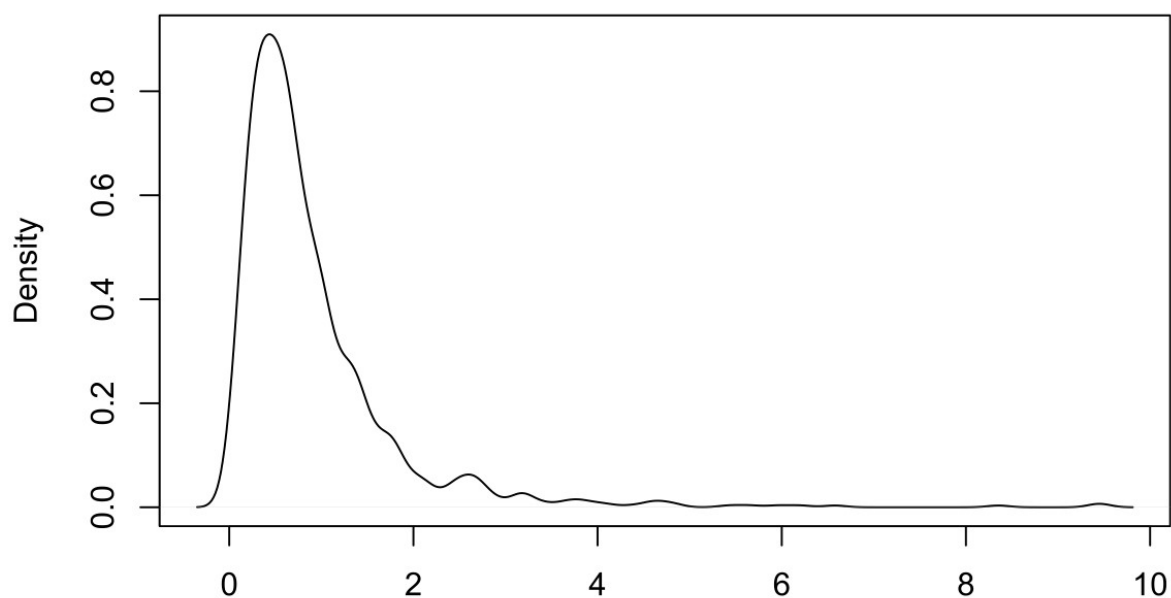
```
quantile(odd, c(.025, .975))
```

```
##      2.5%      97.5%
```

```
## 0.1047694 3.2031358
```

```
plot(density(odd))
```


density.default(x = odd)



N = 1000 Bandwidth = 0.1203

```
truelp<-0.5
n<-200

# simulate a 1000 observation
#rbinom(1000,n,truelp)/n

truelp*(1/sqrt(truelp))*sqrt(truelp*(1-truelp)/n)
```

```
## [1] 0.025
```

```
sd(sqrt(rbinom(1000,n,truelp)/n))
```

```
## [1] 0.02575247
```

```
p1<-c(0.1,0.5,0.9)
```

```
p2<-c(0.1,0.5,0.9)
```

```
n<-100
```

```
rate<-matrix(NA,ncol=3,nrow=3)
```

```
for(i in 1:3){
```

```
  for(j in 1:3){
```

```
    n11<-rbinom(1000,n,p1[i])
```

```
    n12<-n-n11
```

```
    n21<-rbinom(1000,n,p2[j])
```

```
    n22<-n-n21
```

```
    CI<-1.96*sqrt(1/n11+1/n12+1/n21+1/n22)
```

```

    orhat<-log((n11*n22)/(n12*n21))
    trueor<-log((p1[i]/(1-p1[i]))/(p2[j]/(1-p2[j])))

    rate[i,j]<-sum((trueor>orhat-CI)&(trueor<orhat+CI))/1000
  }
}

rate

```

```

##      [,1] [,2] [,3]
## [1,] 0.967 0.952 0.965
## [2,] 0.948 0.947 0.945
## [3,] 0.956 0.948 0.957

```

```

p1<-c(0.1,0.5,0.9)
p2<-c(0.1,0.5,0.9)
n<-100

rate<-matrix(NA,ncol=3,nrow=3)
for(i in 1:3){
  for(j in 1:3){
    p1hat<-rbinom(1000,n,p1[i])/n

    p2hat<-rbinom(1000,n,p2[j])/n

    CI<-1.96*sqrt((1-p1hat)/(p1hat*n)+(1-p2hat)/(p2hat*n))

    rrhat<-log(p1hat/p2hat)
    truerr<-log(p1[i]/p2[j])

    rate[i,j]<-sum((truerr>rrhat-CI)&(truerr<rrhat+CI))/1000
  }
}

rate

```

```

##      [,1] [,2] [,3]
## [1,] 0.975 0.960 0.950
## [2,] 0.967 0.961 0.952
## [3,] 0.963 0.948 0.963

```

```

library(knitr)
purl("HW1.Rmd")

```