

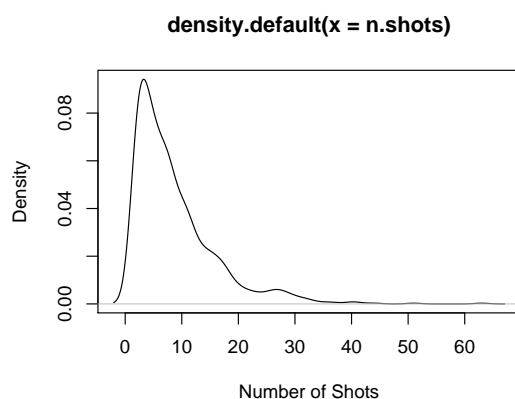
# Stat 505 Assignment 11

Leslie Gains-Germain

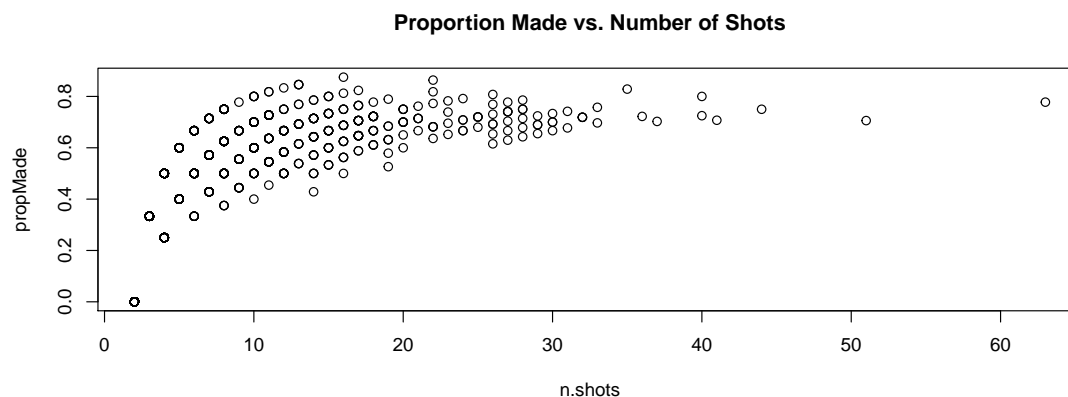
November 17, 2014

- (a) I built a while loop to do this. The loop keeps track of the number of shots taken and the proportion of shots made. See my code in the appendix.  
(b) I wrapped the while loop inside a for loop. For each “game”, the number of shots taken and the proportion of shots made is recorded. I showed the results for the first six games. The mean and standard deviation of the number of shots taken in the 1000 games is 8.713 and 7.253 respectively.

```
##      n.shots propMade
## [1,]      8    0.500
## [2,]      8    0.625
## [3,]     10    0.400
## [4,]      3    0.333
## [5,]      3    0.333
## [6,]      7    0.571
## [1] 8.71
## [1] 7.25
```



- (c) Below is a scatterplot of number of shots and proportion made.



- (a) In general, we sample from a known population distribution, then using this sample we build a confidence interval for a known population parameter. Then, we note

whether the interval contains the known population parameter. We do this many times (1000 or 10000 times), and this gives us an idea of the proportion of times that our confidence interval contains the true parameter.

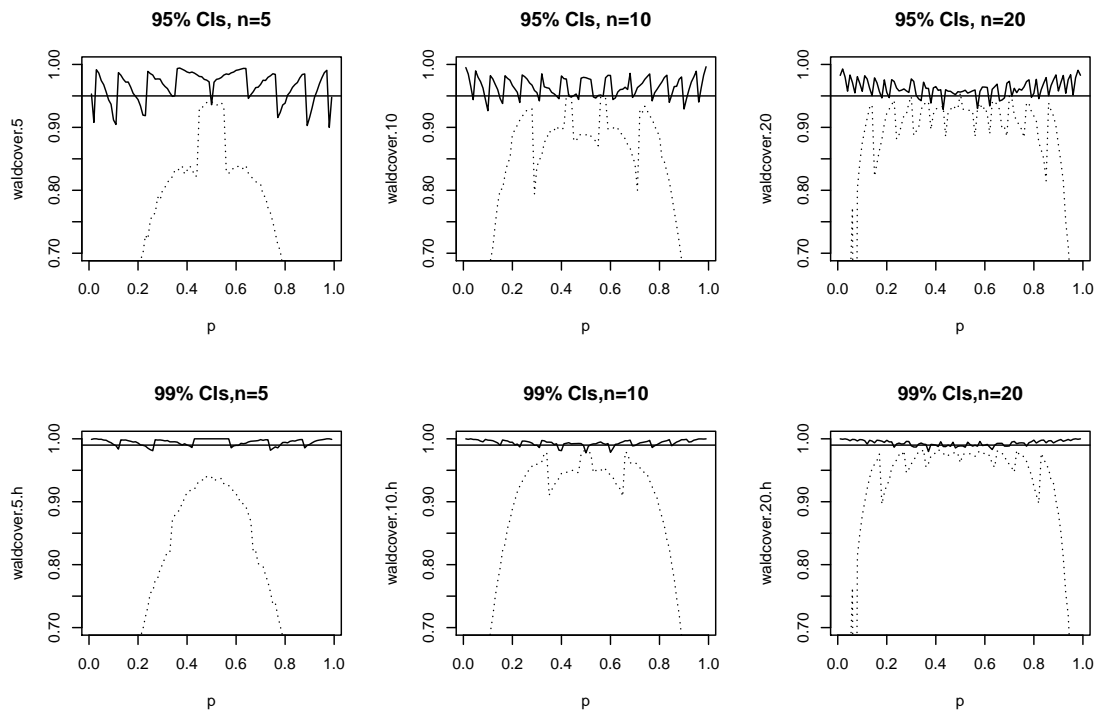
- (b) My `cover` function outputs the wald confidence intervals and the proportion of intervals that covered the true parameter. Here is an example:

```
## $waldcis
##      [,1] [,2]
## [1,] 0.332 0.768
## [2,] 0.232 0.668
## [3,] 0.185 0.615
## [4,] 0.332 0.768
## [5,] 0.385 0.815
## [6,] 0.385 0.815
## [7,] 0.185 0.615
## [8,] 0.441 0.859
## [9,] 0.281 0.719
## [10,] 0.281 0.719
##
## $prop.cover
## [1] 1
```

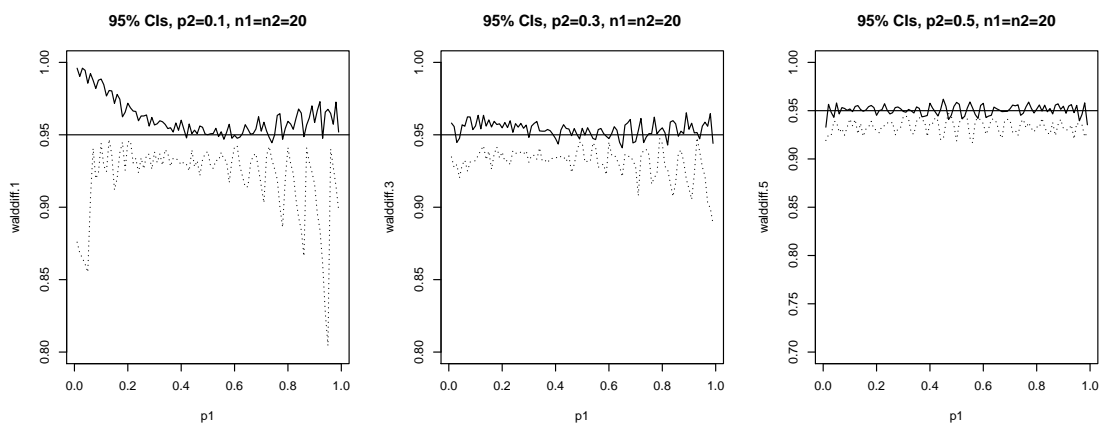
- (c) My `cover.wilson` function outputs the wilson confidence intervals and the proportion of intervals that contain the true parameter. Here is an example:

```
## $wilsoncis
##      [,1] [,2]
## [1,] 0.342 0.741
## [2,] 0.478 0.855
## [3,] 0.110 0.474
## [4,] 0.300 0.700
## [5,] 0.342 0.741
## [6,] 0.219 0.614
## [7,] 0.219 0.614
## [8,] 0.431 0.819
## [9,] 0.386 0.781
## [10,] 0.386 0.781
##
## $prop.cover
## [1] 0.9
```

- (d) *Figure 1 recreated.*



(e) *Figure 4 recreated.*



## R Code

```
n.shots <- rep(0,1000)
propMade <- rep(0,1000)
for(i in 1:1000){
  keep.going <- 0
  score <- 0
  shots <- 0
  miss <- 0
  prev.miss <- 0
  while(keep.going<2){
    shots <- shots+1
    prev.miss <- miss
    miss <- ifelse(runif(1,0,1)>.6,1,0)
    score <- score+ifelse(miss==1,0,1)
    keep.going <- prev.miss+miss
    #keep.going <- ifelse(miss+prev.miss==2,FALSE,TRUE)
    ## set keep.going to FALSE when condition is met
  }
  n.shots[i]<-shots
  propMade[i] <- score/shots
  #print(unlist( list( n.shots= shots, propMade = score / shots)))
}
```

```
data <- cbind(n.shots,propMade)
head(data)
mean(n.shots)
sd(n.shots)
par(mfrow=c(1,2))
plot(density(n.shots), xlab="Number of Shots")
```

```
plot(propMade~n.shots, main="Proportion Made vs. Number of Shots")
```

```
cover <- function (n,p,Nreps,confidence){
  phat.vec<-rbinom(Nreps,n,p)/n
  wald.ci.l<-phat.vec+qnorm((1-confidence)/2)*sqrt(phat.vec*(1-phat.vec)/n)
  wald.ci.u<-phat.vec-qnorm((1-confidence)/2)*sqrt(phat.vec*(1-phat.vec)/n)
  cis <- matrix(c(wald.ci.l,wald.ci.u),nrow=Nreps,ncol=2)
  prop.cover <- sum(ifelse(cis[,1]<p & cis[,2]>p,1,0))/Nreps
  finals <-list("waldcis"=cis,"prop.cover"=prop.cover)
  return(finals)
}
cover(20,0.5,10,0.95)
```

```
cover.wilson <- function (n,p,Nreps,confidence){
  ptilde.vec<-(rbinom(Nreps,n,p)+2)/(n+4)
  wald.ci.l<-ptilde.vec+qnorm((1-confidence)/2)*sqrt(ptilde.vec*(1-ptilde.vec)/(n+4))
  wald.ci.u<-ptilde.vec-qnorm((1-confidence)/2)*sqrt(ptilde.vec*(1-ptilde.vec)/(n+4))
  cis <- matrix(c(wald.ci.l,wald.ci.u),nrow=Nreps,ncol=2)
  prop.cover <- sum(ifelse(cis[,1]<p & cis[,2]>p,1,0))/Nreps
  finals <-list("wilsoncis"=cis,"prop.cover"=prop.cover)
  return(finals)
}
cover.wilson(20,0.5,10,0.95)
```

```
Nreps <- 10000
waldcover.5 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
  waldcover.5[100*i]<-cover(5,i,Nreps,0.95)$prop.cover
}
waldcover.10 <- rep(0,99)
```

```

for(i in seq(0.01,0.99,by=0.01)){
waldcover.10[100*i]<-cover(10,i,Nreps,0.95)$prop.cover
}
waldcover.20 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
waldcover.20[100*i]<-cover(20,i,Nreps,0.95)$prop.cover
}

waldcover.5.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
waldcover.5.h[100*i]<-cover(5,i,Nreps,0.99)$prop.cover
}
waldcover.10.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
waldcover.10.h[100*i]<-cover(10,i,Nreps,0.99)$prop.cover
}
waldcover.20.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
waldcover.20.h[100*i]<-cover(20,i,Nreps,0.99)$prop.cover
}
}

Nreps<-10000
wilsoncover.5 <- c(rep(0,99))
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.5[zapsmall(100*i)] <- cover.wilson(5,i,Nreps,0.95)$prop.cover
}

wilsoncover.10 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.10[zapsmall(100*i)]<-cover.wilson(10,i,Nreps,0.95)$prop.cover
}
wilsoncover.20 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.20[zapsmall(100*i)]<-cover.wilson(20,i,Nreps,0.95)$prop.cover
}

wilsoncover.5.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.5.h[zapsmall(100*i)]<-cover.wilson(5,i,Nreps,0.99)$prop.cover
}
wilsoncover.10.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.10.h[zapsmall(100*i)]<-cover.wilson(10,i,Nreps,0.99)$prop.cover
}
wilsoncover.20.h <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
wilsoncover.20.h[zapsmall(100*i)]<-cover.wilson(20,i,Nreps,0.99)$prop.cover
}
}

par(mfrow=c(3,3))
p <- seq(0.01,0.99,by=0.01)
plot(p,waldcover.5, ylim=c(0.7,1), type="l", lty=3, main="95% CIs, n=5")
abline(h=0.95)
lines(p,wilsoncover.5)

plot(p,waldcover.10, ylim=c(0.7,1), type="l", lty=3, main="95% CIs, n=10")
abline(h=0.95)
lines(p,wilsoncover.10)

plot(p,waldcover.20, ylim=c(0.7,1), type="l", lty=3, main="95% CIs, n=20")
abline(h=0.95)
lines(p,wilsoncover.20)

plot(p,waldcover.5.h, ylim=c(0.7,1), type="l", lty=3, main="99% CIs,n=5")

```

```

abline(h=0.99)
lines(p,wilsoncover.5.h)

plot(p,waldcover.10.h, ylim=c(0.7,1), type="l", lty=3, main="99% CIs,n=10")
abline(h=0.99)
lines(p,wilsoncover.10.h)

plot(p,waldcover.20.h, ylim=c(0.7,1), type="l", lty=3, main="99% CIs,n=20")
abline(h=0.99)
lines(p,wilsoncover.20.h)

```

```

cover.diff <- function (n1,n2,p1,p2,Nreps,confidence){
  phat1.vec<-rbinom(Nreps,n1,p1)/n1
  phat2.vec<-rbinom(Nreps,n2,p2)/n2
  wald.ci.l<-phat1.vec-phat2.vec+qnorm((1-confidence)/2)*sqrt(phat1.vec*(1-phat1.vec)/n1+phat2.vec*(1-phat2.vec)/n2)
  wald.ci.u<-phat1.vec-phat2.vec-qnorm((1-confidence)/2)*sqrt(phat1.vec*(1-phat1.vec)/n1+phat2.vec*(1-phat2.vec)/n2)
  cis <- matrix(c(wald.ci.l,wald.ci.u),nrow=Nreps,ncol=2)
  diff<-p1-p2
  prop.cover <- sum(ifelse(cis[,1]<diff & cis[,2]>diff,1,0))/Nreps
  finals <-list("waldcis"=cis,"prop.cover"=prop.cover)
  return(finals)
}

```

```

coverwilson.diff <- function (n1,n2,p1,p2,Nreps,confidence){
  ptilde1.vec<-(rbinom(Nreps,n1,p1)+1)/(n1+2)
  ptilde2.vec<-(rbinom(Nreps,n2,p2)+1)/(n2+2)
  wald.ci.l<-ptilde1.vec-ptilde2.vec+qnorm((1-confidence)/2)*
    sqrt(ptilde1.vec*(1-ptilde1.vec)/(n1+2)+ptilde2.vec*(1-ptilde2.vec)/(n2+2))
  wald.ci.u<-ptilde1.vec-ptilde2.vec-qnorm((1-confidence)/2)*
    sqrt(ptilde1.vec*(1-ptilde1.vec)/(n1+2)+ptilde2.vec*(1-ptilde2.vec)/(n2+2))
  cis <- matrix(c(wald.ci.l,wald.ci.u),nrow=Nreps,ncol=2)
  diff<-p1-p2
  prop.cover <- sum(ifelse(cis[,1]<diff & cis[,2]>diff,1,0))/Nreps
  finals <-list("waldcis"=cis,"prop.cover"=prop.cover)
  return(finals)
}

```

```

Nreps <- 10000
walddiff.1 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
  walddiff.1[zapsmall(100*i)]<-cover.diff(20,20,i,0.1,Nreps,0.95)$prop.cover
}
walddiff.3 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
  walddiff.3[zapsmall(100*i)]<-cover.diff(20,20,i,0.3,Nreps,0.95)$prop.cover
}
walddiff.5 <- rep(0,99)
for(i in seq(0.01,0.99,by=0.01)){
  walddiff.5[zapsmall(100*i)]<-cover.diff(20,20,i,0.5,Nreps,0.95)$prop.cover
}

```

```

Nreps<-10000
wilsondiff.1 <- c(rep(0,99))
for(i in seq(0.01,0.99,by=0.01)){
  wilsondiff.1[zapsmall(100*i)] <- coverwilson.diff(20,20,i,0.1,Nreps,0.95)$prop.cover
}
wilsondiff.3 <- c(rep(0,99))
for(i in seq(0.01,0.99,by=0.01)){
  wilsondiff.3[zapsmall(100*i)] <- coverwilson.diff(20,20,i,0.3,Nreps,0.95)$prop.cover
}
wilsondiff.5 <- c(rep(0,99))
for(i in seq(0.01,0.99,by=0.01)){
  wilsondiff.5[zapsmall(100*i)] <- coverwilson.diff(20,20,i,0.5,Nreps,0.95)$prop.cover
}

```

```

par(mfrow=c(1,3))
p <- seq(0.01,0.99,by=0.01)
plot(p,walddiff.1, ylim=c(0.8,1), type="l", lty=3, main="95% CIs, p2=0.1, n1=n2=20", xlab="p1")
abline(h=0.95)
lines(p,wilsondiff.1)

plot(p,walddiff.3, ylim=c(0.8,1), type="l", lty=3, main="95% CIs, p2=0.3, n1=n2=20", xlab="p1")
abline(h=0.95)
lines(p,wilsondiff.3)

plot(p,walddiff.5, ylim=c(0.7,1), type="l", lty=3, main="95% CIs, p2=0.5, n1=n2=20", xlab="p1")
abline(h=0.95)
lines(p,wilsondiff.5)

```