R Computing Midterm Exam:Take Home

葉早彬,陳威宇,劉瑞祥

Q1

```
heart_up=function(x){
        sqrt(1-(abs(x)-1)^2)
}

heart_lo=function(x){
        acos(1-abs(x))-pi
}
x=seq(-2,2,0.05)
# plot(x,heart_lo(x),ylim=c(heart_lo(0),1),type='l')
# lines(x,heart_up(x))
n=100000
machine_gun=function(n){
        up_bound=max(heart_up(x))
        lo_bound=min(heart_lo(x))
        bullet.x=runif(n,-2,2)
        bullet.y=runif(n,lo_bound,up_bound)
        hit=sum(bullet.y<=heart_up(bullet.x)&bullet.y>=heart_lo(bullet.x))
        box=4*(up_bound-lo_bound)
        box*hit/n
}
machine_gun(n)
```

Q2

```
bessel.element=function(a,v,z,m){
        denominator=(gamma(m+a+1)*factorial(m))^v
        numerator=(z/2)^(2*m+a)
        numerator/denominator
}

fn=function(a,v,z,max,tolerance,m){
        if(m>max){
                return (0)
        }

        i=bessel.element(a,v,z,m)
        if(i>tolerance)
        {
                return (i+fn(a,v,z,max,tolerance,m+1))
        }
        else{
                return (0)
        }
}

basselI_Gen=function(a,v,z,max,tolerance){
        fn(a,v,z,max,tolerance,0)
```

```
}
```

Q3

```
wald.interval=function(size,theta.hat){
        upper_bound=theta.hat+1.96*sqrt(theta.hat*(1-theta.hat)/size)
        lower_bound=theta.hat-1.96*sqrt(theta.hat*(1-theta.hat)/size)
        result=list(upper_bound=upper_bound,lower_bound=lower_bound)
        return (result)
}

adjustwald.interval=function(size,theta.hat){
        theta.tilde=(size*theta.hat+2)/(size+4)
        upper_bound=theta.tilde+1.96*sqrt(theta.tilde*(1-theta.tilde)/size)
        lower_bound=theta.tilde-1.96*sqrt(theta.tilde*(1-theta.tilde)/size)
        result=list(upper_bound=upper_bound,lower_bound=lower_bound)
        return (result)
}

coverage.sim=function(size,theta){
        n=5000
        y=rbinom(n,size,theta)
        thetas=rep(theta,n)
        theta.hat=y/size
        w=wald.interval(size,theta.hat)
        a=adjustwald.interval(size,theta.hat)
        w.overlap=thetas[thetas>=w$lower_bound&thetas<=w$upper_bound]
        a.overlap=thetas[thetas>=a$lower_bound&thetas<=a$upper_bound]
        wald.coverage=length(w.overlap)/n
        adjust.coverage=length(a.overlap)/n
        return (list(wald=wald.coverage,adjust=adjust.coverage))
}

coverage.graph=function(){
        size=20
        thetas=seq(0.01,0.49,0.02)
        wald=c()
        adjust=c()
        for(i in 1:length(thetas)){
                theta=thetas[i]
                coverage=coverage.sim(size,theta)
                wald[i]=coverage$wald
                adjust[i]=coverage$adjust
        }
        plot(cbind(c(0,0.5),seq(0.65,1,0.05)),type="n",ylab='Coverage',xlab=expression(the
        abline(h=0.95)
        lines(thetas,wald,col='blue')
        lines(thetas,adjust,col='red')
}
coverage.graph()
```