STAT3701Hw2

Mingming Xu

2019/7/8

Problem 1

```
est.p=function(n,p,reps){
  u.mat=matrix(runif(reps*n),reps,n)
  x.mat=ifelse(u.mat>p,0,1)
  x1.mat=apply(x.mat,1,sum)
  x2.mat=ifelse(x1.mat<4,0,1)
  return(mean(x2.mat))
}

rep=1000
pro1.est=numeric(rep)
for(r in 1:1000){
  pro1.est[r]=est.p(7,0.3,1e4)
}
pro2.est=numeric(rep)
for(r in 1:1000){
  pro2.est[r]=est.p(7,0.5,1e4)
}
pro3.est=numeric(rep)
for(r in 1:1000){
  pro3.est[r]=est.p(7,0.7,1e4)
}

##
mean(pro1.est)
```

```
## [1] 0.1258652
```

```
mean(pro2.est)
```

```
## [1] 0.49974
```

```
mean(pro3.est)
```

```
## [1] 0.8741497
```

```
## standard error
sd(pro1.est)
```

```
## [1] 0.003315862
```

```
sd(pro2.est)
```

```
## [1] 0.005108067
```

```
sd(pro3.est)
```

```
## [1] 0.003370282
```

Problem 2

i.

```
## the function to generate xi~N(0,1)
sdrnorm=function(n){
  odd=(n%%2)!=0
  if(odd){
    n=n+1
  }
  u1=runif(n/2)
  u2=runif(n/2)
  rad=sqrt(-2*log(u1))
  z.list=c(rad*cos(2*pi*u2), rad*sin(2*pi*u2) )
  if(odd){
    z.list=z.list[-n]
  }
return(z.list)
}
##because sum of xi^2 is chi square distributuion with df n.The function chi.gen(number of sample, df)
chi.gen=function(m,n){
  chi.list=numeric(m)
  for(r in 1:m){
    z.list=sdrnorm(n)
    chi.list[r]=sum(z.list^2)
    }
return(chi.list)
}
## estimate the mean of the distribution and  the standard error
z.mat=matrix(chi.gen(100*10,10),100,10)
z1.mat=apply(z.mat,1,mean)
mean(z1.mat)
```

```
## [1] 9.963441
```

```
sd(z1.mat)
```

```
## [1] 1.350743
```

ii.

```
## the function to generate exp(lambda)
exp.gen=function(n,lambda){
  u=runif(n)
  x=-lambda*log(u)
  return(x)
}

## the function to generate the gamma distribution
gam.gen=function(m,n,lambda){   ## m is the number of generations
 g.list=numeric(m)
  for(r in 1:m){
    x.list=exp.gen(n,lambda)
    g.list[r]=sum(x.list)
    }
return(g.list)
}
##generate iid observations from gamma(n,2) : choose n is 10
```

```r
g.list=gam.gen(10000,10,2)
chi.list=chi.gen(10000,20)
## the mean and standard errors in both case:
mean(g.list)
```

```
## [1] 19.96465
```

```r
mean(chi.list)
```

```
## [1] 19.95695
```

```r
sd(g.list)
```

```
## [1] 6.253388
```

```r
sd(chi.list) ## the mean and standard error of gamma(n,2) are so closed to the chi square(2n)
```

```
## [1] 6.319525
```

Problem 3

i.$\int_a^b \frac{\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}}{\Phi(b)-\Phi(a)}\,dy$

$=\frac{1}{\Phi(b)-\Phi(a)}\int_a^b \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}\,dx$

$=\frac{1}{\Phi(b)-\Phi(a)}(\Phi(b)-\Phi(a))$

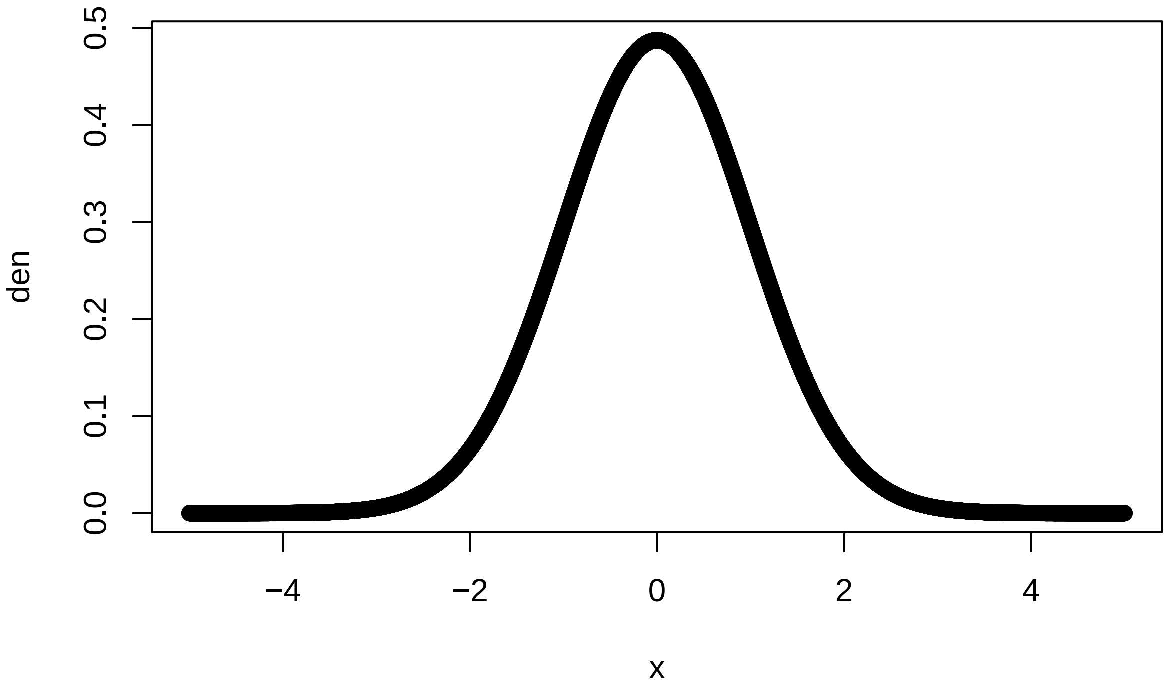
=1

So, this is a proper pdf.

ii. If b>0 and a=-b, the expectation of X following this distriebution is 0 because of symmetric.

iii.

```r
a=-1
b=2
x=seq(-5,5,length.out = 1e4)
dom=pnorm(b)-pnorm(a)
tp=dnorm(x)
den=tp/dom
plot(x,den)
```



iv.

For the truncated normal distribution, F(X)=$\int_a^x \frac{\frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}}}{\Phi(b)-\Phi(a)}\,dx$

$=\frac{\Phi(x)-\Phi(a)}{\Phi(b)-\Phi(a)}$. Let F(x)=u, where u is uniform(0,1).

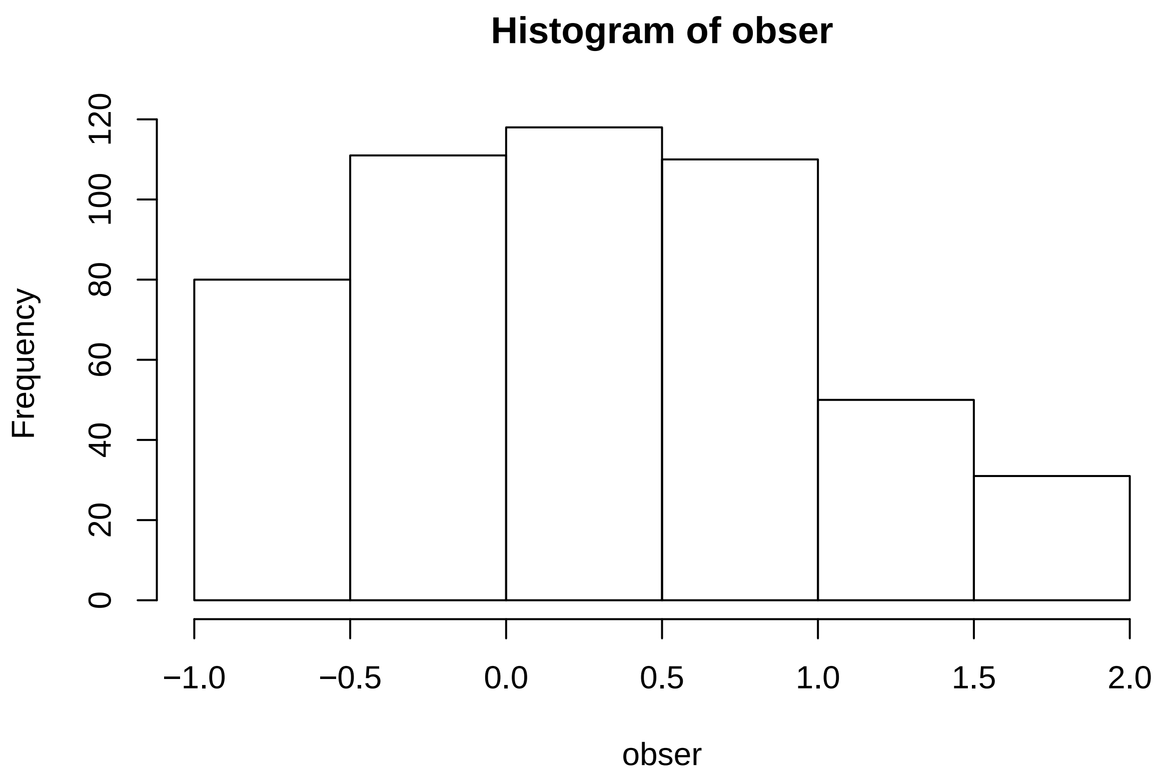
$\Phi(x) = \Phi(a) + u(\Phi(b) - \Phi(a))$

$x = \Phi^{-1}(\Phi(a) + u(\Phi(b) - \Phi(a)))$

```
trun.gen=function(n,a,b){
  u=runif(n)
  phia=pnorm(a,0,1)
  phib=pnorm(b,0,1)
  phix=phia+u*(phib-phia)
  x=qnorm(phix,0,1)
  return(x)
}
```

v.

```
obser=trun.gen(500,-1,2)
hist(obser)
```



vi.

```
## confidence interval with a=-1 ,b=2
n=500
t.mat=matrix(trun.gen(100*n,-1,2),100,n)
tbar.mat=apply(t.mat,1,mean)
s.mat=apply(t.mat,1,sd)
tper=qt(0.975,n-1)
upper.ci=tbar.mat-tper*s.mat/sqrt(n)
lower.ci=tbar.mat+tper*s.mat/sqrt(n)
mean(upper.ci)
```

```
## [1] 0.1727204
```

```
mean(lower.ci)
```

```
## [1] 0.2995867
```

```
ci=c(mean(upper.ci),mean(lower.ci))
ci
```

```
## [1] 0.1727204 0.2995867
```

Problem 4

i. When n is 10,type I error: P(we reject H_0 | H_0 is ture)=P($\frac{X}{n} \leq \frac{3}{10}|\theta = .5$)=P($X \leq 3|\theta = 0.5$)

```
typeI=pbinom(3,10,0.5)
typeI
```

```
## [1] 0.171875
```

Power: P(we reject $H_0|H_a$ is ture)=P($\frac{X}{n} \leq \frac{3}{10}|\theta = 0.25$)=P($X \leq 3|\theta = 0.25$)

```
pow=pbinom(3,10,0.25)
pow
```

```
## [1] 0.7758751
```

ii.

```
n=10
reps=10^4
theta=0.25
## the function to generate observations from binomial distribution
rbinom.gen=function(k,n,theta){
  b=c()
  for(i in 1:k){
    u=runif(n)
    b1=c()
    for(j in 1:n){
      b1[j]=ifelse(u[j]>theta,0,1)
    }
  b[i]=sum(b1)
    }
return(b)
}
##a simulation study to corroborate results
x.list=rbinom.gen(reps,n,theta)
phat=ifelse(x.list>3,0,1)
mean(phat)
```

```
## [1] 0.7816
```

```
p=mean(phat)
## confidence interval
ci=c(p-1/sqrt(10^4),p+1/sqrt(10^4))
ci
```

```
## [1] 0.7716 0.7916
```

By a simulation study, the result is so closed to my result about the power of this decision rule from part i.

iii.

```
reps=10^4
theta=0.25
x1.list=rbinom.gen(reps,10,theta)
p1hat=ifelse(x1.list>3,0,1)
p1=mean(p1hat)
x2.list=rbinom.gen(reps,20,theta)
p2hat=ifelse(x2.list>3,0,1)
p2=mean(p2hat)
x3.list=rbinom.gen(reps,30,theta)
p3hat=ifelse(x3.list>3,0,1)
p3=mean(p3hat)
x4.list=rbinom.gen(reps,40,theta)
p4hat=ifelse(x4.list>3,0,1)
p4=mean(p4hat)
x5.list=rbinom.gen(reps,50,theta)
p5hat=ifelse(x5.list>3,0,1)
```

```
p5=mean(p5hat)
p1;p2;p3;p4;p5
```

```
## [1] 0.7781
## [1] 0.2238
## [1] 0.0389
## [1] 0.0042
## [1] 5e-04
```

When n is larger and the rest are fixed, the power is samller.

Problem 5

a. Since Y and V are independent, $E(X) = E(YV) = E(Y)E(V) = \theta\lambda$.

$Var(X) = Var(YV) = E(Y^2V^2) - E(YV)^2 = E(Y^2)E(V^2) - (E(Y)E(V))^2$. Because $E(Y^2) = Var(Y) + E(Y)^2 = \theta(1-\theta) + \theta^2 = \theta$,$E(V^2) = Var(V) + E(V)^2 = \lambda^2 + \lambda^2 = 2\lambda^2$, $Var(YV) = 2\theta\lambda^2 - (\lambda\theta)^2 = \lambda^2\theta(2-\theta)$

Set $\theta = 1/4, \lambda = 3$, $E(X) = 3/4, Var(X) = 63/14$

```
theta=1/4
lambda=3
Ex=theta*lambda
Vx=lambda^2*theta*(2-theta)
Ex;Vx
```

```
## [1] 0.75
## [1] 3.9375
```

b.

```
## the function to generate binomial observations
mybio.gen=function(k,n,p){
b=c();
for (j in 1:k) {
b1=c();
u=runif(n);
for (i in 1:n) {
b1[i]=ifelse(u[i]>p,1,0);
}
b[j]=sum(b1)
}
return (b)
}
## the function to generate exponential
myexp.gen=function(n,lambda){
  u=runif(n)
  x=-lambda*log(u)
  return(x)
}
## the function to generate x
x.gen=function(n,theta,lambda){
y.list=mybio.gen(n,1,theta)
v.list=myexp.gen(n,lambda)
x.list=y.list*v.list
```

```
return(x.list)
}
n=10^4
theta=1/4
lambda=3
x=x.gen(n=n,theta=theta,lambda = lambda)
mean(x)
```

```
## [1] 2.27701
```

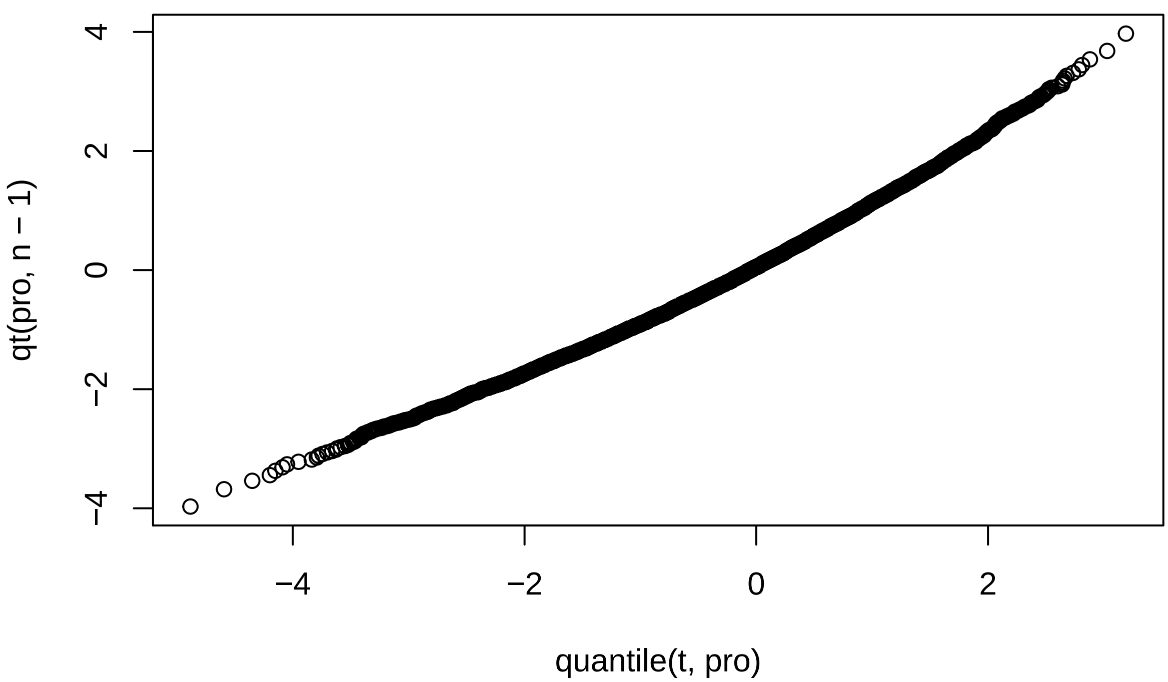
```
var(x)
```

```
## [1] 8.307212
```

My result has diffrence to the part a.

c. Test statistic:$T = \frac{\sqrt{n}(\bar{X}-1)}{S}$

```
reps=10^4
n=200
t=c()
xbar=numeric(reps)
s=numeric(reps)
t=numeric(reps)
for(r in 1:reps){
  x=x.gen(n,1/2,2)
  xbar[r]=mean(x)
  s[r]=sd(x)
  t[r]=sqrt(n)*(xbar[r]-1)/s[r]
}
pro=ppoints(reps)
plot(quantile(t,pro),qt(pro,n-1))
```

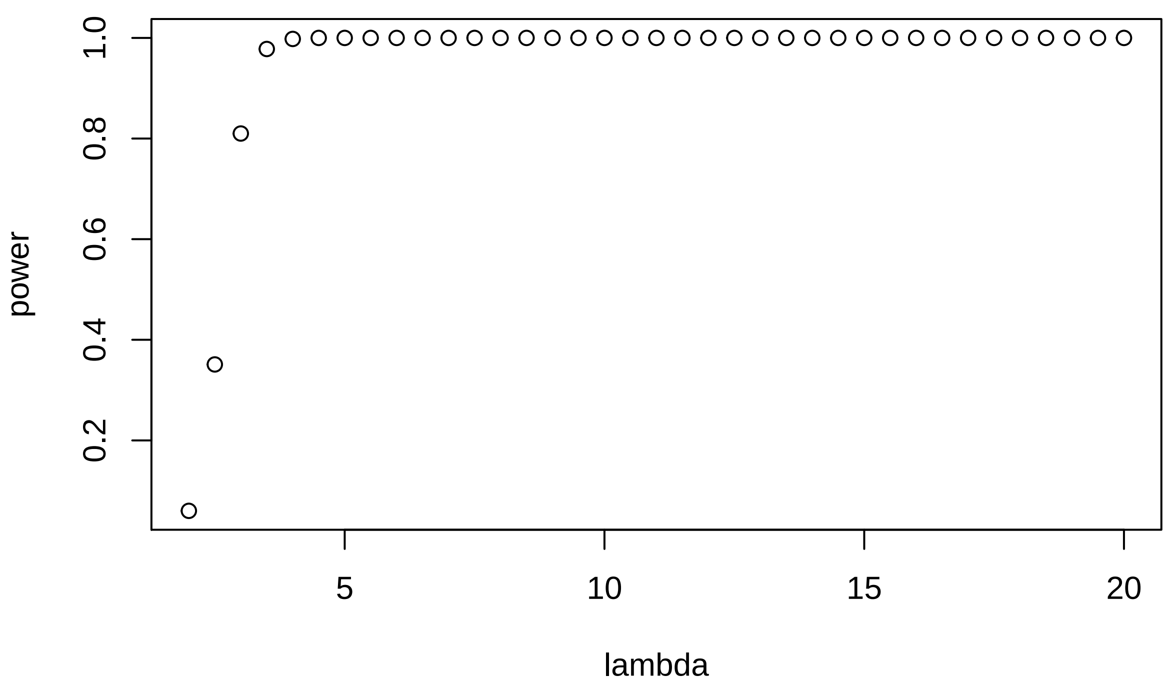


From this plot,we could see that the points fall along a line in the graph .Hence,this is a good approxiation of the t distribution with n-1 df.

d.

```
est.power=function(n,theta,lambda,alpha,reps){
  x.mat=matrix(x.gen(reps*n,theta,lambda),nrow = reps,ncol = n)
  xbar=apply(x.mat, 1,mean)
  s=apply(x.mat,1,sd)
  t=(xbar-1)/(s/sqrt(n))
  prop=mean(abs(t)>qt(1-alpha/2,n-1))
  return(prop)
}
```

```
n=200
theta=1/2
alpha=0.05
lambda=seq(2,20,by=0.5)
reps=1000
len=length(lambda)
power=numeric(len)
for (i in 1:len) {
  lam=lambda[i]
  power[i]=est.power(n,theta,lam,alpha,reps)
  }
plot(lambda,power)
```

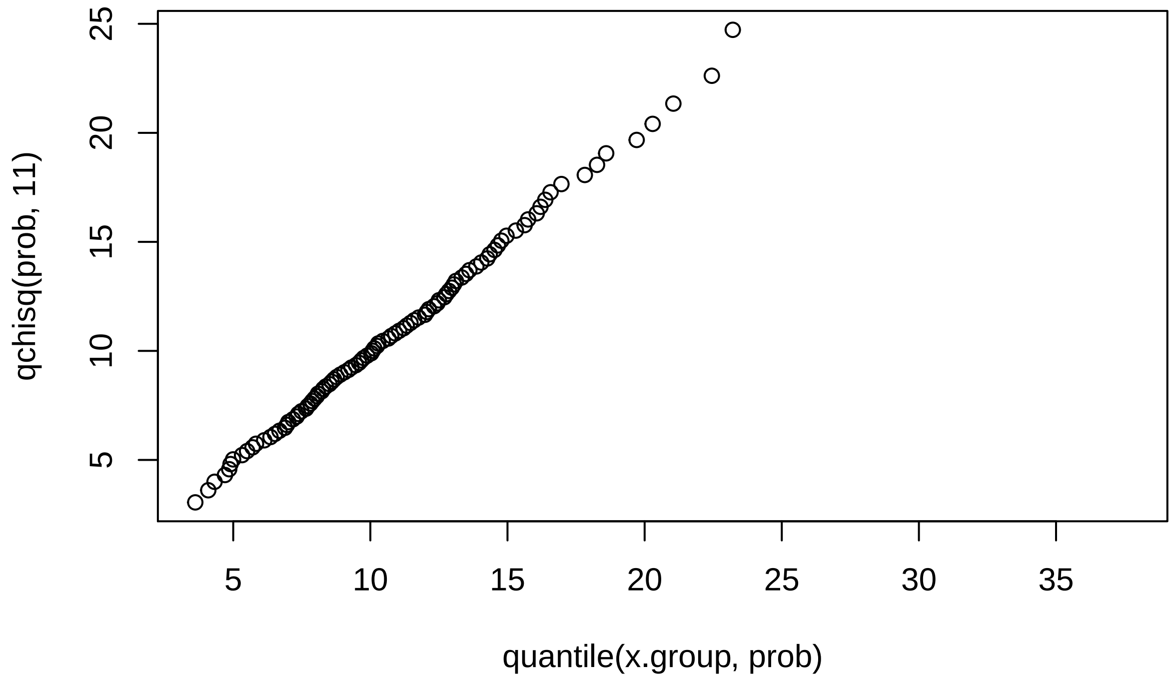


From this plot, the power becomes 1 after $\lambda \geq 5$.

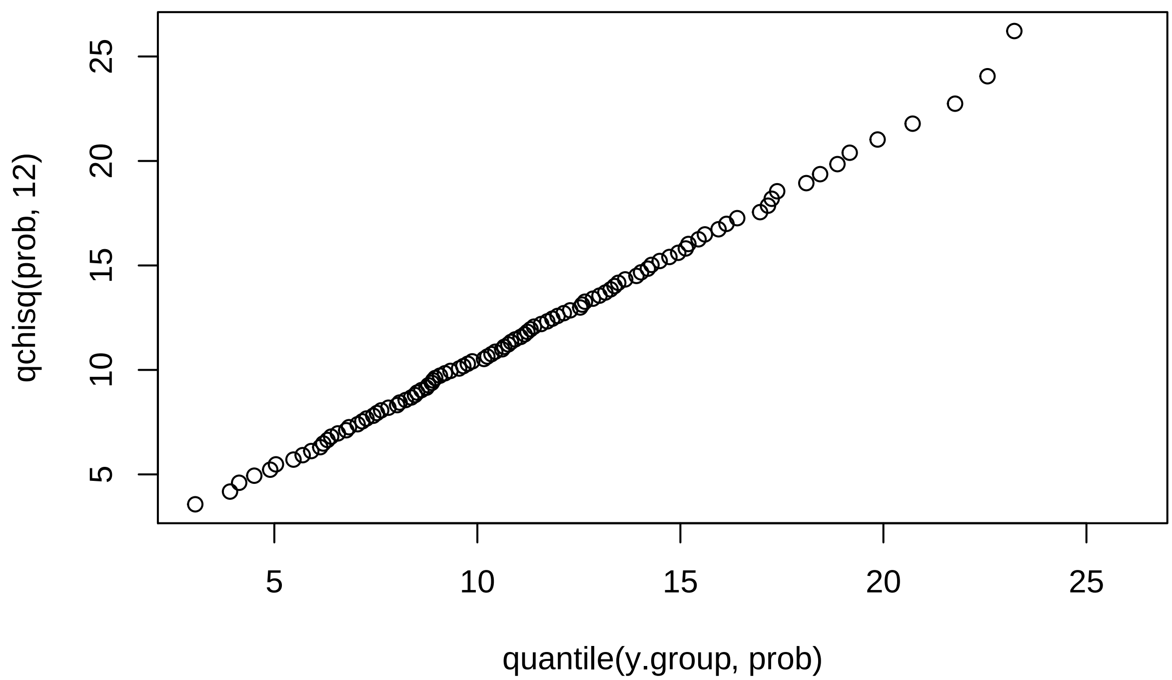
Problem 6

a.

```
x.group=chi.gen(500,11)
y.group=chi.gen(500,12)
prob=seq(0.01,1,by=0.01)
plot(quantile(x.group,prob),qchisq(prob,11))
```




```
plot(quantile(y.group,prob),qchisq(prob,12))
```



In these two plots, the points alomost fall along a line in the middle of the graph, but some seperate points.

b.

```
## a simulation to generate p-val:
pval.sim=function(k,l,n1,n2,reps){
  pval.list=numeric(reps)
  for (r in 1:reps){
    x.list=chi.gen(n1,k)
    y.list=chi.gen(n2,l)
```

```
    xbar=mean(x.list)
    ybar=mean(y.list)
    residuals=c(x.list-xbar, y.list-ybar)
    sp=sum(residuals^2)/(n1+n2-2)
    t=(xbar-ybar)/sqrt(sp* (1/n1+1/n2))
    pval.list[r] = 2*pt(-abs(t), n1+n2-2)
  }
  return(pval.list)
}
## find the power:
pow.sim=function(k,l,n1,n2,reps,alpha){
  p.val=pval.sim(k,l,n1,n2,reps)
  pow=mean(p.val<alpha)
  return(pow)
}
```

```
k=11
l=12
alpha=0.05
reps=10^4
## choose n is 450
n=450
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.8817
```

```
## choose n is 485
n=485
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.9011
```

```
##choose n is 486
n=486
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.9023
```

```
## choose n is 487
n=487
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.9041
```

```
##choose n is 488
n=488
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.9023
```

```
##choose n is 489
n=490
pow.sim(k,l,n,n,reps,alpha)
```

```
## [1] 0.8988
```

Hence, when n1=n2=485 such that the power of the two independent sample t test is roughly 90%.

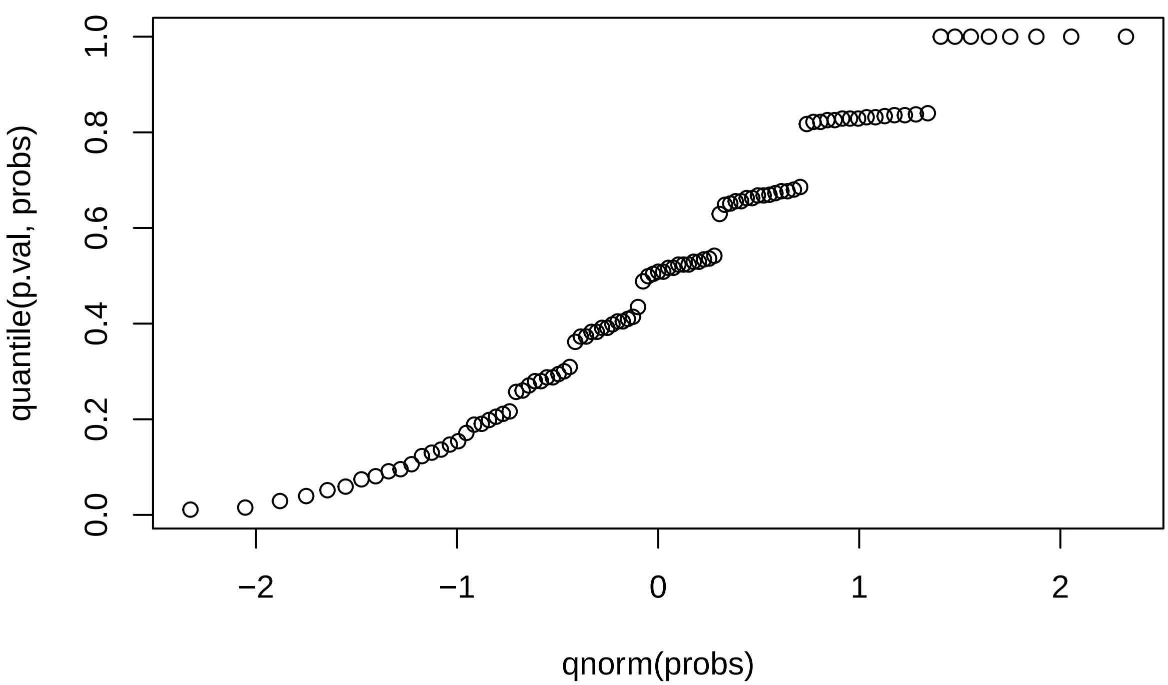
Problem 7

a.

```
bio.gen=function(k,n,p){
b=c();
for (j in 1:k) {
b1=c();
u=runif(n);
for (i in 1:n) {
b1[i]=ifelse(u[i]>p,1,0);
}
b[j]=sum(b1)
}
return (b)
}
propHypothesisTest=function(n,theta1,theta2,reps){
v.list=bio.gen(reps,n,theta1)
w.list=bio.gen(reps,n,theta2)
theta1hat=v.list/n
theta2hat=w.list/n
thetahat=(v.list+w.list)/(2*n)
t=(theta1hat-theta2hat)/sqrt(2*thetahat*(1-thetahat)/n)
pval=2*pnorm(-abs(t),0,1)
return(pval)
}
```

b.

```
p.val=propHypothesisTest(n=50,theta1=0.68,theta2 = 0.68,reps=1000)
probs=seq(0.01,1,by=0.01)
plot(qnorm(probs),quantile(p.val, probs) )
```

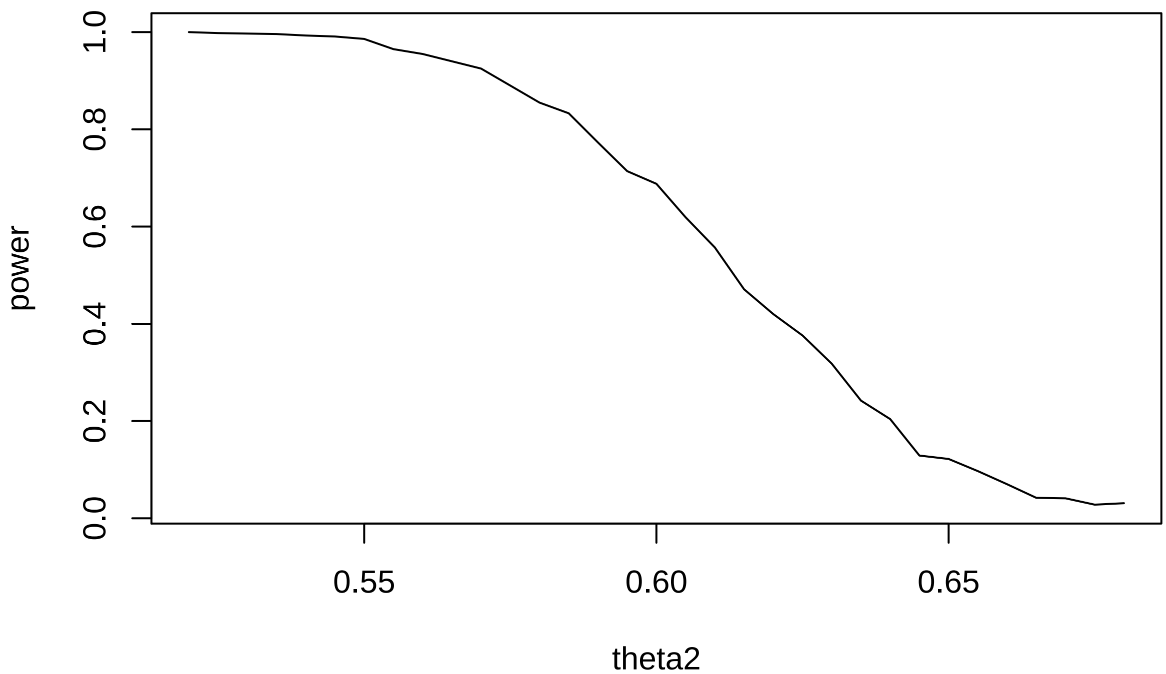


From this plot, we could see that under the null hypothesis, the distribution of p value is almost uniform(0,1).

c.

```r
n=500
the1=0.68
reps=1000
alpha=0.03
the2=seq(from=0.520,to=0.680,by=0.005)
num=length(the2)
est.pow=numeric(num)
lb=numeric(num)
ub=numeric(num)
for (r in 1:num) {
pval=propHypothesisTest(n,the1,the2[r],reps)
est.pow[r]=mean(pval<alpha)
bounds=binom.test(x=sum(pval<0.03), n=reps, conf.level=0.95)$conf.int[1:2]
lb[r]=bounds[1]
ub[r]=bounds[2]
}

plot(the2, est.pow,type = "l",xlab = "theta2",ylab="power")
```




```r
##lines(the2,ub,lty=1)
##lines(the2,lb,lty=1)
```

Problem 8

a.

```r
x.gener=function(n,mu1,sigma,gamma){
  a.list=rnorm(n)*sigma
  l1.list=rnorm(n)*gamma
  x.list=mu1+a.list+l1.list

  return(x.list)

}
```

```
x.dat=x.gener(50000,68,sqrt(2),1)
t.test((x.dat-68)^2,conf.level = 0.99)$conf.int[1:2]
```

```
## [1] 2.957074 3.054754
```

3 is in this interval.

b.

```
y.gener=function(n,mu2,sigma){
  a.list=rnorm(n)*sigma
  l2.list=rt(n,4)
  y.list=mu2+a.list+l2.list

  return(y.list)

}

y.dat=y.gener(50000,70,sqrt(2))
t.test((y.dat-70)^2,conf.level = 0.99)$conf.int[1:2]
```

```
## [1] 3.882022 4.071645
```

4 is in this interval.

c.

We set the $Z_i = X_i - Y_i$ and $E(Z_i) = \mu$. Because $E(X_i) = \mu_1, E(Y_i) = \mu_2$, $E(Z_i) = E(X_i - Y_i) = E(X_i) - E(Y_i) = \mu_1 - \mu_2 = \mu$. Now, the hypothesis becomes $H_0 : \mu = 0$ $H_a : \mu \neq 0$. Test statistic :$T = \frac{\bar{Z}-0}{s/\sqrt{n}}$

```
z.gener=function(n,mu1,mu2,sigma,gamma,alpha,reps){
  x=x.gener(n,mu1,sigma,gamma)
  y=y.gener(n,mu2,sigma)
  z=x-y
  return(z)
}

## generate power

est.power=function(n,mu1,mu2,sigma,gamma,alpha,reps){
  z.mat=matrix(z.gener(reps*n,mu1,mu2,sigma,gamma),nrow = reps,ncol = n)
  zbar=apply(z.mat, 1,mean)
  s=apply(z.mat,1,sd)
  t=(zbar-0)/(s/sqrt(n))
  prop=mean(abs(t)>qt(1-alpha/2,n-1))
  return(prop)
}

mu1=68
mu2=70
reps=10000
sigma=sqrt(2)
gamma=1
alpha=0.05
n.list=c(15:25)
len=length(n.list)
pow=numeric(len)
```

```r
for (r in 1:len){
  n=n.list[r]
  pow[r]=est.power(n,mu1,mu2,sigma,gamma,alpha,reps)
}

pow
```

```
##  [1] 0.7919 0.8069 0.8316 0.8538 0.8770 0.8935 0.9138 0.9234 0.9326 0.9444
## [11] 0.9474
```

The value of n is 20 or 21 such that the power of the test statistic for the two sample paired t-test is apporoximately 90%.