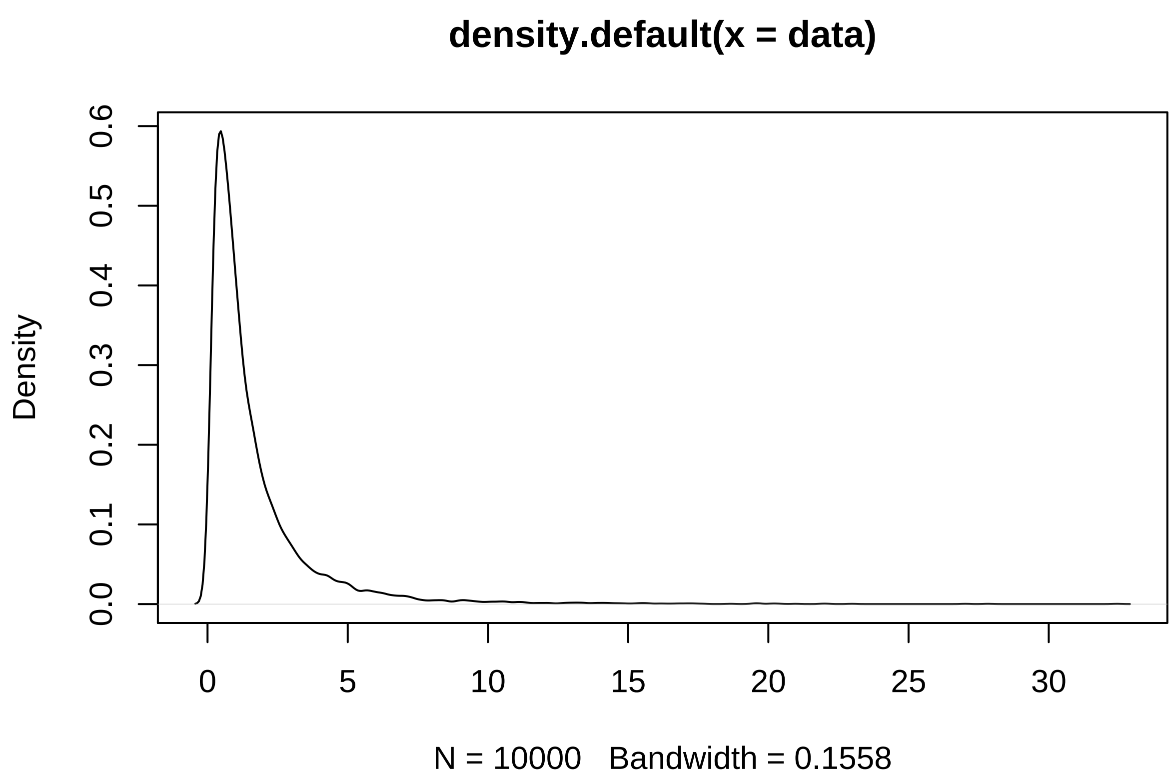
Homework3

Shan Zhong

20181016

```
# a
data<-rlnorm(10000, meanlog = 0, sdlog = 1)
plot(density(data))
```




```
#b
library("MASS")
n<-c(3,5,10,30)
alpha<-0.05
iteration<-1000

store<-matrix(rep(0,12),nrow=3,ncol=4)
output<-matrix(rep(NA,12),nrow=3,ncol=4)

for(j in (1:iteration)){
  for(i in 1:4){
    data<-rlnorm(n[i],meanlog = 0,sdlog=1)
    output[1,i]<-t.test(data,mu=1)$p.value
  }

  for(i in 1:4){
```

```r
    data<-rlnorm(n[i],meanlog = 0,sdlog=1)
    output[2,i]<-binom.test(sum(data>1),n[i], 0.5, alternative="less" )$p.value
  }

  for(i in 1:4){
    data<-rlnorm(n[i],meanlog = 0,sdlog=1)
    output[3,i]<-wilcox.test(data,mu=1)$p.value
  }
  store<-store+(output<alpha)
}
store/iteration
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] 0.071 0.036 0.058 0.345
## [2,] 0.000 0.029 0.007 0.053
## [3,] 0.000 0.000 0.061 0.164
```

```r
# when difference is 0, , the null pypothesis is true!
# but just assume it is false to see the graph, using distance 0.01
# set alpha to be 0.05
difference<-seq(0.01,3,by = 0.05)
sd<-1
eff.size<-difference/sd

alpha<-0.05
iteration<-1000
output<-c(rep(NA,iteration))
ttest<-c(rep(NA,length(difference)))

# for two-sample t-test
# practice
for(j in 1:length(difference)){
  diff<-difference[j]
  for(i in 1:iteration){
    dataa<-rnorm(30,mean=0,sd=1)
    datab<-rnorm(30,mean=0+diff,sd=1)
    output[i]<-t.test(dataa,datab)$p.value<0.05
  }
  ttest[j]<-sum(output)/iteration
}


# rank sum
# when difference is 0, , the null pypothesis is true!
# but just assume it is false to see the graph, using distance 0.01
# set alpha to be 0.05

output<-c(rep(NA,iteration))
mww<-c(rep(NA,length(difference)))

# for two-sample t-test
# practice
for(j in 1:length(difference)){
  diff<-difference[j]
```

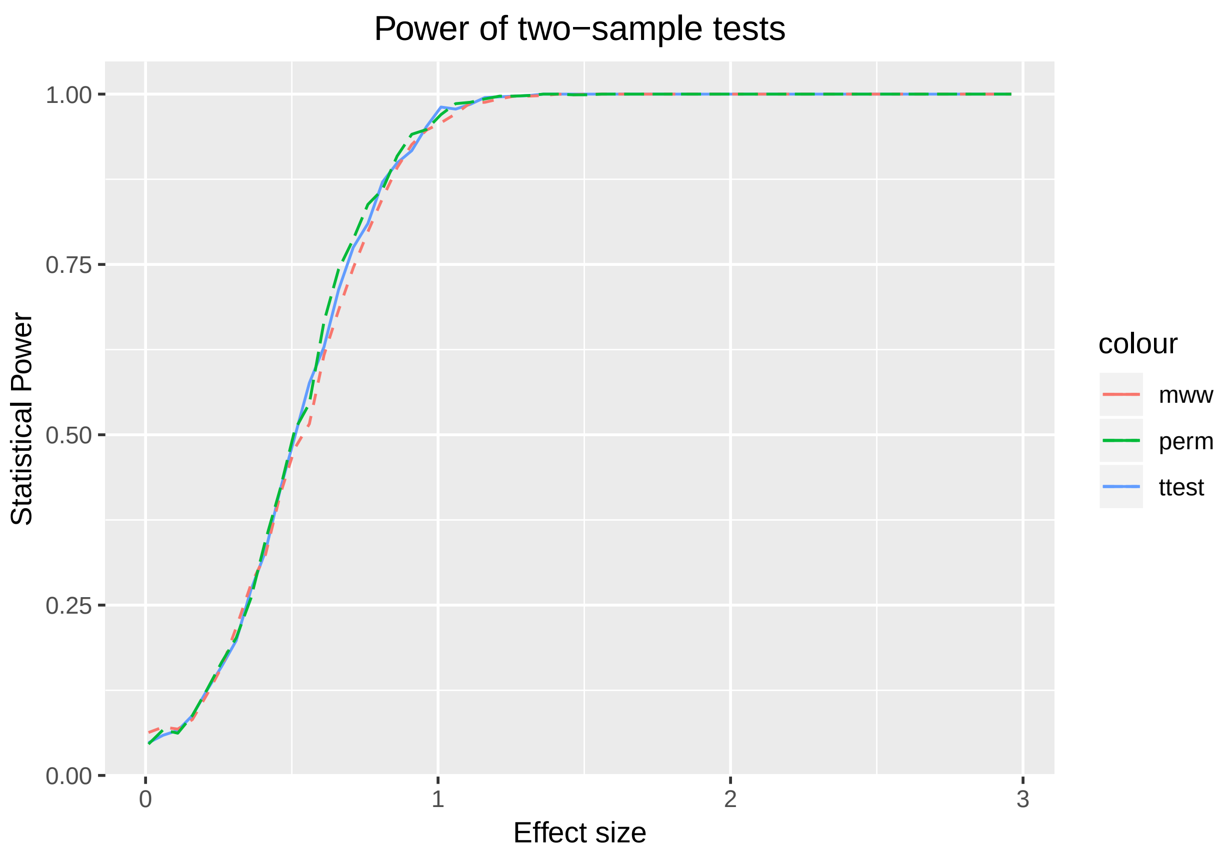
```
  for(i in 1:iteration){
    dataa<-rnorm(30,mean=0,sd=1)
    datab<-rnorm(30,mean=0+diff,sd=1)
    output[i]<-wilcox.test(dataa,datab)$p.value<0.05
  }
  mww[j]<-sum(output)/iteration
}


output<-c(rep(NA,iteration))
perm<-c(rep(NA,length(difference)))

# for two-sample t-test
# practice
for(j in 1:length(difference)){
  diff<-difference[j]
  for(i in 1:iteration){
    dataa<-rnorm(30,mean=0,sd=1)
    datab<-rnorm(30,mean=0+diff,sd=1)
    output[i]<-permTS(dataa, datab, alternative = c("two.sided"))$p.value<0.05
  }
  perm[j]<-sum(output)/iteration
}

data<-data.frame(eff.size,ttest,mww,perm)

ggplot(data,aes())+geom_line(aes(x=eff.size,y=ttest,color="ttest"),linetype="solid")+
  geom_line(aes(x=eff.size,y=mww,color="mww"),linetype="dashed")+
  geom_line(aes(x=eff.size,y=perm,color="perm"),linetype="longdash")+
  ggtitle("Power of two-sample tests") +
  theme(plot.title = element_text(hjust = 0.5))+
  xlab("Effect size")+ylab("Statistical Power")
```




```
##b, these three different test have similar power distribution
```

```
#b
n1<-100
sigma1<-10
n2value<-c(10,50,100,500,1000)
sigma2value<-c(1,5,10,50,100)
irration<-1000
alpha<-0.05
table<-matrix(NA,nrow=5,ncol=5)
rownames(table)<-c("n1/n2=0.1","n1/n2=0.5","n1/n2=1","n1/n2=5","n1/n2=10")
colnames(table)<-c("s1/s2=0.1","s1/s2=0.5","s1/s2=1","s1/s2=5","s1/s2=10")
for(N in 1:5){
  n2<-n2value[N]
  for(s in 1:5){
    sigma2<-sigma2value[s]
    outcome<-rep(NA,irration)
    for(i in 1:irration){
      Y1<-rnorm(n1,mean=0,sd<-sqrt(sigma1))
      Y2<-rnorm(n2,mean=0,sd<-sqrt(sigma2))
      sigma<-sqrt((n1*sigma1+n2*sigma2)/(n1+n2))*(1/n1+1/n2)^(0.5)
      CI<-qnorm(0.975,mean=0,sd=1)*sigma
      outcome[i]<-!(-CI<(mean(Y1)-mean(Y2)))&((mean(Y1)-mean(Y2))<CI)
    }
    table[N,s]<-mean(outcome)
  }
```

```
}
table
```

```
##            s1/s2=0.1 s1/s2=0.5 s1/s2=1 s1/s2=5 s1/s2=10
## n1/n2=0.1     0.000     0.006   0.034   0.147    0.195
## n1/n2=0.5     0.002     0.012   0.019   0.049    0.072
## n1/n2=1       0.027     0.028   0.023   0.026    0.027
## n1/n2=5       0.140     0.048   0.026   0.000    0.000
## n1/n2=10      0.174     0.067   0.028   0.001    0.000
```

```
# I control the n and sigma of the Y1, and change n and sigma of Y1 to fill out the table

# table normal approximation will be turn in by paper

# comment
# as s1/s2 goes larger, the chance of type 1 error increase
# as n1/n2 increase, the chance of type 1 error decrease
```

```
sigmaratio<-c(0.1,0.5,1,5,10)
nratio<-c(0.1,0.5,1,5,10)

table<-matrix(NA,nrow=5,ncol=5)
rownames(table)<-c("n1/n2=0.1","n1/n2=0.5","n1/n2=1","n1/n2=5","n1/n2=10")
colnames(table)<-c("s1/s2=0.1","s1/s2=0.5","s1/s2=1","s1/s2=5","s1/s2=10")

for(n in 1:5){
  for(s in 1:5){
    sigma<-sigmaratio[s]
    nr<-nratio[n]
    teststatistic<-1.96*sqrt((sigma+nr)/(sigma*nr+1))
    table[n,s]<-1-pnorm(teststatistic,0,1)
  }
}

round(table,2)
```

```
##            s1/s2=0.1 s1/s2=0.5 s1/s2=1 s1/s2=5 s1/s2=10
## n1/n2=0.1      0.19      0.07    0.02    0.00     0.00
## n1/n2=0.5      0.07      0.04    0.02    0.01     0.00
## n1/n2=1        0.02      0.02    0.02    0.02     0.02
## n1/n2=5        0.00      0.01    0.02    0.11     0.14
## n1/n2=10       0.00      0.00    0.02    0.14     0.19
```

```
#b
n1<-100
sigma1<-10
n2value<-c(10,50,100,500,1000)
sigma2value<-c(1,5,10,50,100)
irration<-1000
alpha<-0.05
table<-matrix(NA,nrow=5,ncol=5)
rownames(table)<-c("n1/n2=0.1","n1/n2=0.5","n1/n2=1","n1/n2=5","n1/n2=10")
colnames(table)<-c("s1/s2=0.1","s1/s2=0.5","s1/s2=1","s1/s2=5","s1/s2=10")
for(N in 1:5){
  n2<-n2value[N]
```

```r
  for(s in 1:5){
    sigma2<-sigma2value[s]
    outcome<-rep(NA,irration)
    for(i in 1:irration){
      Y1<-rnorm(n1,mean=0,sd<-sigma1)
      Y2<-rnorm(n2,mean=0,sd<-sigma2)
      outcome[i]<-permTS(Y1, Y2, alternative = c("two.sided"))$p.value<alpha
    }
    table[N,s]<-mean(outcome)
  }
}
table
```

```
##            s1/s2=0.1 s1/s2=0.5 s1/s2=1 s1/s2=5 s1/s2=10
## n1/n2=0.1      0.000     0.002   0.044   0.479    0.534
## n1/n2=0.5      0.006     0.019   0.046   0.168    0.184
## n1/n2=1        0.053     0.046   0.047   0.042    0.048
## n1/n2=5        0.369     0.211   0.052   0.001    0.000
## n1/n2=10       0.552     0.270   0.042   0.000    0.000
```

```r
### effect
# as n1/n2 increase, the chance of type 1 error increase
# as s1/s2 increase, the chance of type 1 error decrease

# so may be use both test when we observe the variance is not equal will be a good idea
```

```r
library(knitr)
purl("Homework3.Rmd")
```