Chapter 3: Resampling Approaches with R

Jixiang Wu, Associate Professor, Plant Science Department, South Dakota State University, Brookings, SD 57007

September 19, 2016

Introduction

Many statistical tests for parameters of interest are based on particular probability functions, such as normal distributions, binomial distributions, multi-nomial distributions, t distributions. However, it is possible that sometimes the distributions for some data are unknown or critical assumptions could be violated. It is also possible that distributions for some parameter estimates could be complicated or unknown, for example, proportional variance components. Small data sizes could lead high Type I and Type II errors. With these, resampling techniques, as alternative methods,sometimes, could be very useful in various quantitative genetics analyses.

There are several commonly used resampling techniques, which include: bootstrapping, permutation (randomization), jackknife, and Monte Carlo methods. Among these, jakknife methods use reduced sample size while the other three types of methods don’t. The following R scripts will demonstrate some applications in basic genetic data analyses.

Correlation analysis with Jackknife

We will calculate the correlation between two cotton traits: lint yield (LY) and lint percentage (LP) from the data set in package coursedata.Please refer to the slides for chapter 3.

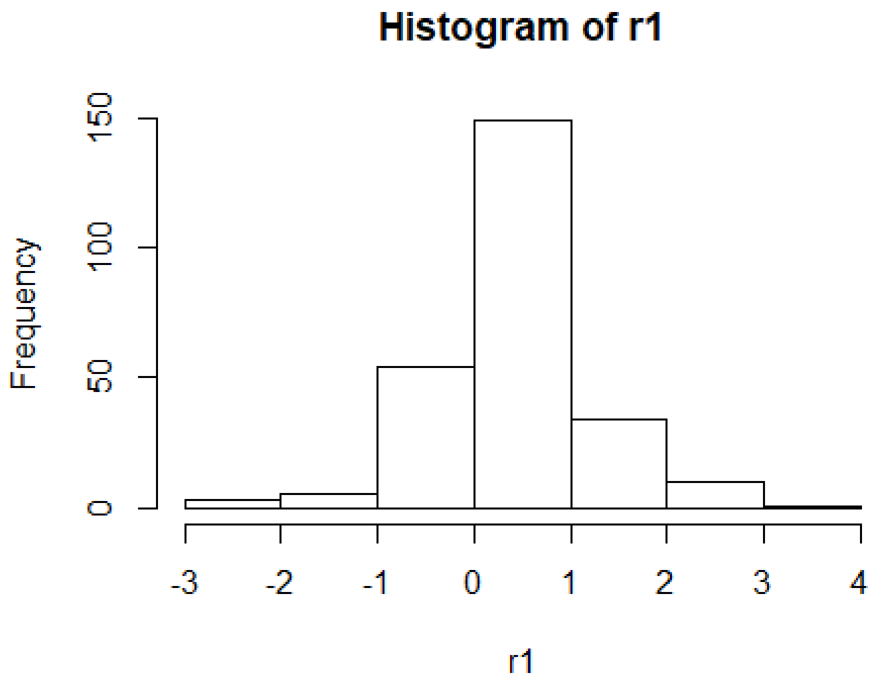
```
require(coursedata)

## Loading required package: coursedata

data(cotyldreg)
summary(cotyldreg)

##        LP              BW              BN             LS              SB       
##  Min.   :31.3   Min.   :3.87   Min.   :211   Min.   :55.9   Min.   :19.5  
##  1st Qu.:37.2   1st Qu.:5.13   1st Qu.:475   1st Qu.:67.0   1st Qu.:27.6  
##  Median :38.6   Median :5.46   Median :555   Median :72.4   Median :29.3  
##  Mean   :38.4   Mean   :5.54   Mean   :551   Mean   :72.3   Mean   :29.6  
##  3rd Qu.:39.9   3rd Qu.:5.89   3rd Qu.:621   3rd Qu.:76.9   3rd Qu.:31.2  
##  Max.   :42.3   Max.   :7.51   Max.   :942   Max.   :89.9   Max.   :40.7  
##        LY      
##  Min.   : 431  
##  1st Qu.:1018  
##  Median :1200  
##  Mean   :1167  
##  3rd Qu.:1336  
##  Max.   :1741
```

```r
attach(cotyldreg)
```

```r
r0<-cor(LP,LY)
n<-length(LY)
r1<-numeric()

for(i in 1:n){
  x<-LP[-i]
  y<-LY[-i]
  r<-cor(x,y)
  r1[i]<-n*r0-(n-1)*r
}
se<-sqrt(var(r1)/n)
jr<-mean(r1)
t<-jr/se
pv<-(1-pt(t,n-1))*2

data.frame(r0,jr,t,pv)
```

```
##      r0    jr    t pv
## 1 0.434 0.435 9.05  0
```

```r
hist(r1)
```

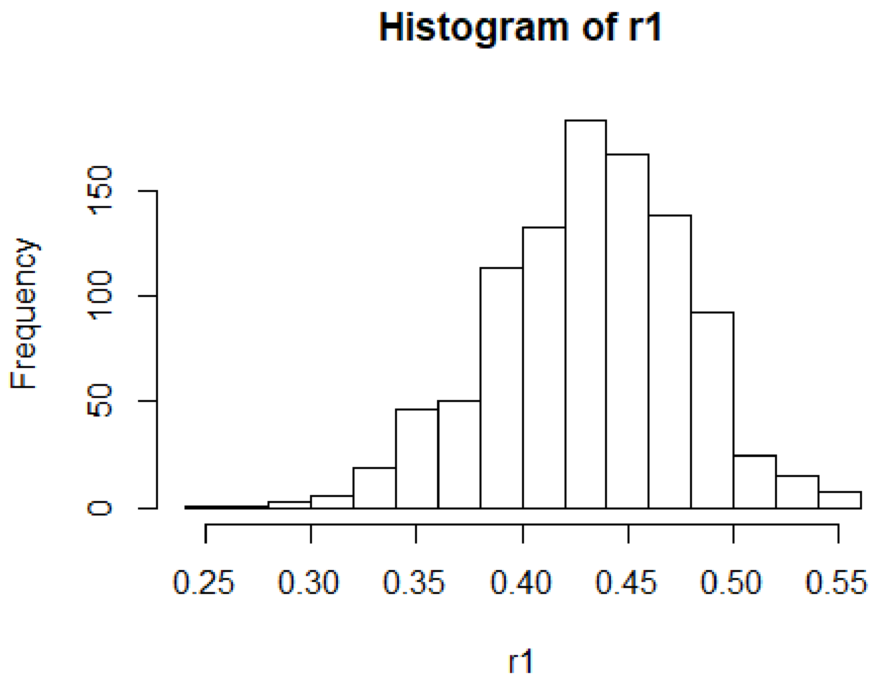


The above results showed that estimated correlation coefficient from the original data and jackknife mean are very close.

Correlation analysis with bootstrapping

We will use the same data with bootstrapping resampling technique.

```
r0<-cor(LP,LY)
n=length(LY)
N=1000
r1<-numeric(N)

for(i in 1:N){
  id=sample(1:n,replace=TRUE)
  x<-LP[id]
  y<-LY[id]
  r1[i]<-cor(x,y)
}

hist(r1)
```



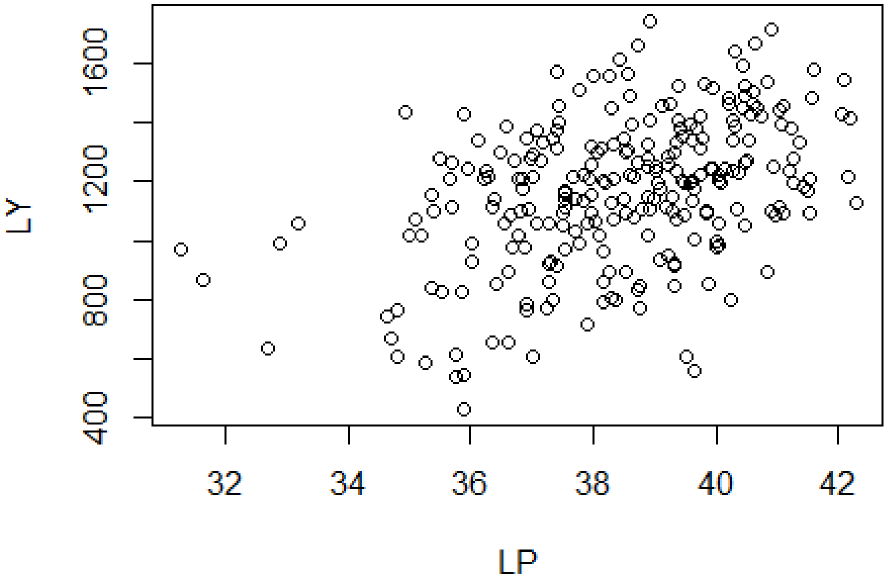
```
se<-sqrt(var(r1))
br<-mean(r1)
t<-br/se
pv<-(1-pt(t,N-1))*2
data.frame(r0,br,t,pv)

##      r0    br    t pv
## 1 0.434 0.432 9.35  0

prob=c(0.005,0.995)
quantile(r1,prob=prob)  # confidence interval test can be used here

##  0.5% 99.5%
## 0.304 0.544
```

The above results showed that estimated correlation coefficient from the original data and bootstrapping mean are very close.

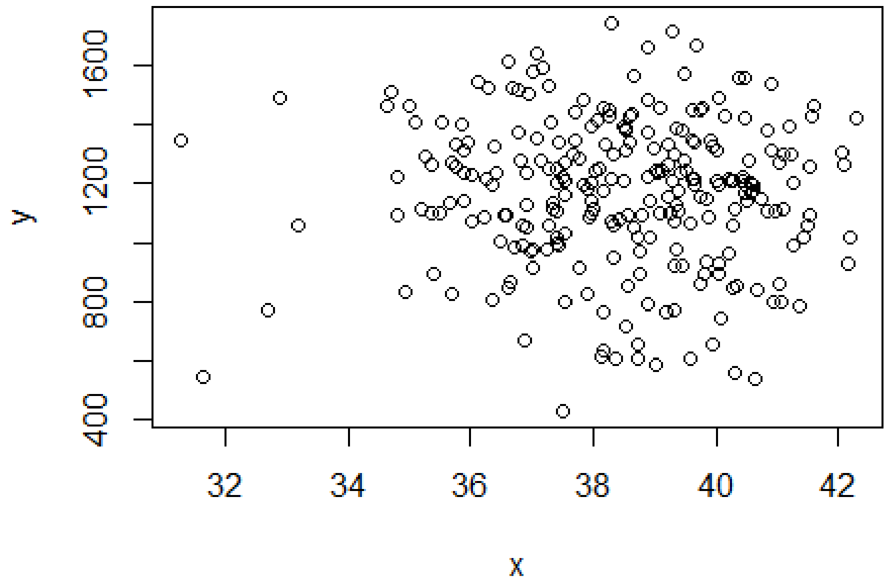
Correlation analysis with permutation

We will use the same data with permutation resampling technique.

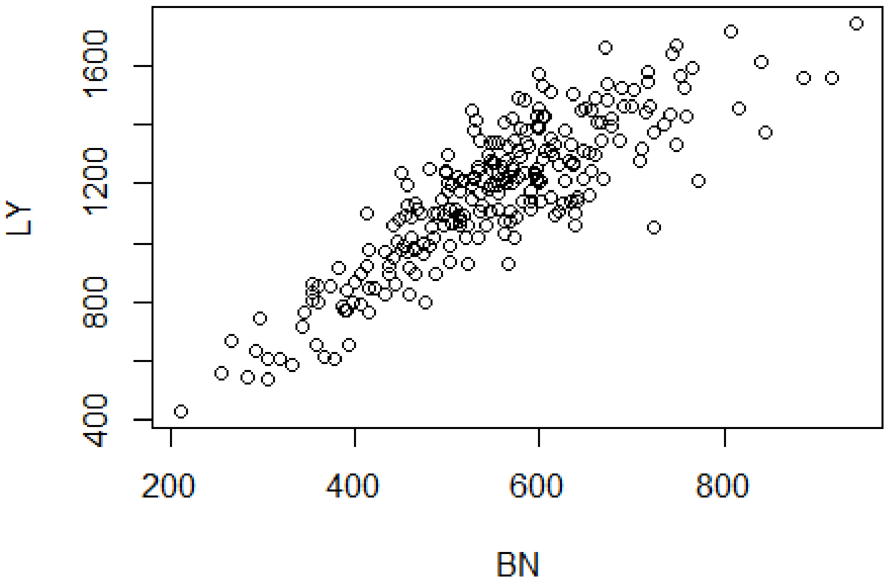
```
plot(LY~LP)  # a slight linear pattern
```



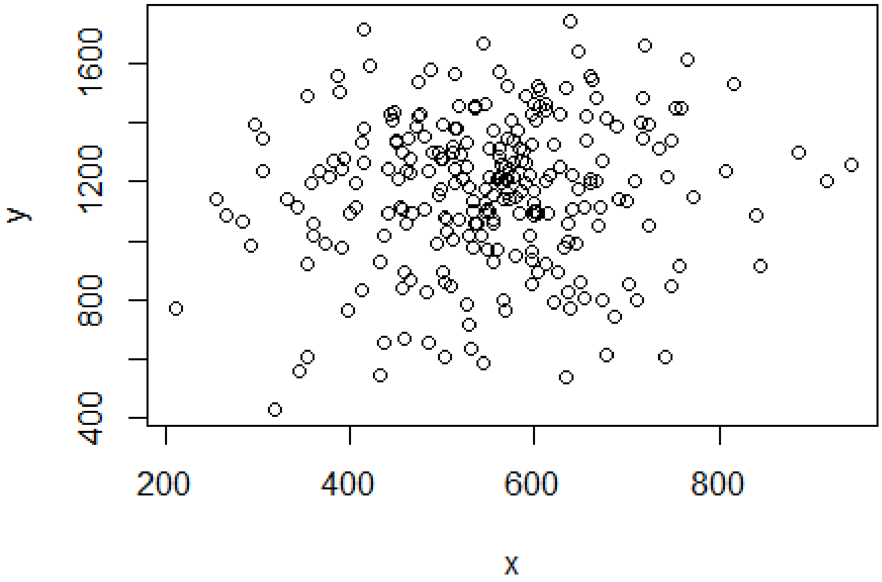
```
n=length(LY)
id=sample(n)
x<-LP[id]
y<-LY
plot(y~x)  # random pattern
```



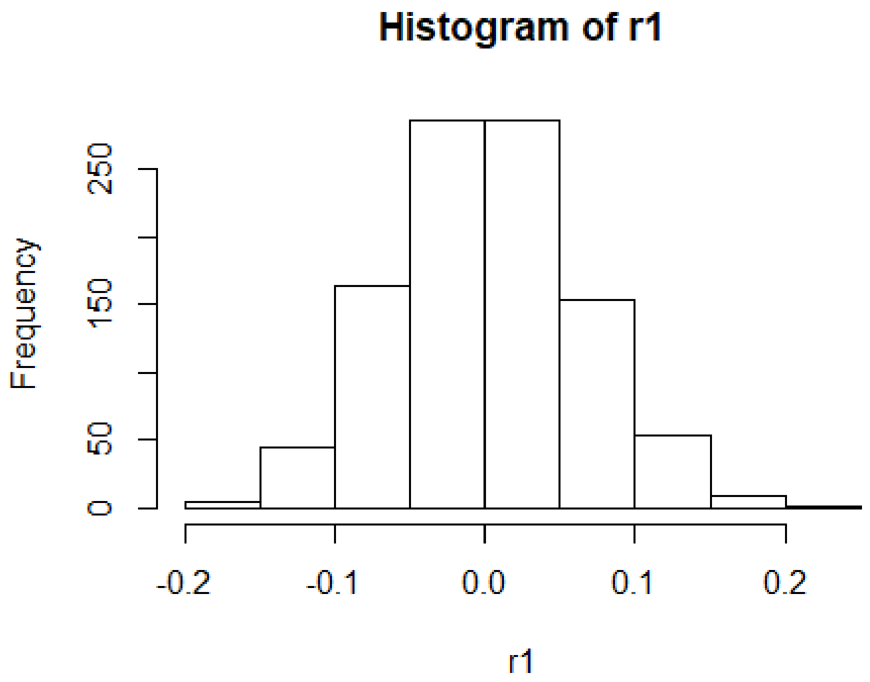
```r
plot(LY~BN)   # strong linear pattern
```



```
id=sample(n)
x<-BN[id]
y<-LY
plot(y~x)  # random pattern
```



```
r0<-cor(LP,LY)

N=1000
r1<-numeric(N)

for(i in 1:N){
  id=sample(1:n)
  x<-LP[id]
  y<-LY
  r1[i]<-cor(x,y)
}

hist(r1)
```



```
se<-sqrt(var(r1))
br<-mean(r1)
t<-br/se
pv<-(1-pt(t,N-1))*2
data.frame(r0,br,t,pv)

##      r0      br      t    pv
## 1 0.434 0.00188 0.0298 0.976

prob=c(0.005,0.995)
quantile(r1,prob=prob)  # confidence interval test can be used here

##   0.5%  99.5%
## -0.143  0.167
```

The above results showed that estimated correlation coefficient from the original data is much far away from the 99% confidence interval, indicating that correlation coefficient is highly significant.

We may also calculate the probability when using permutation

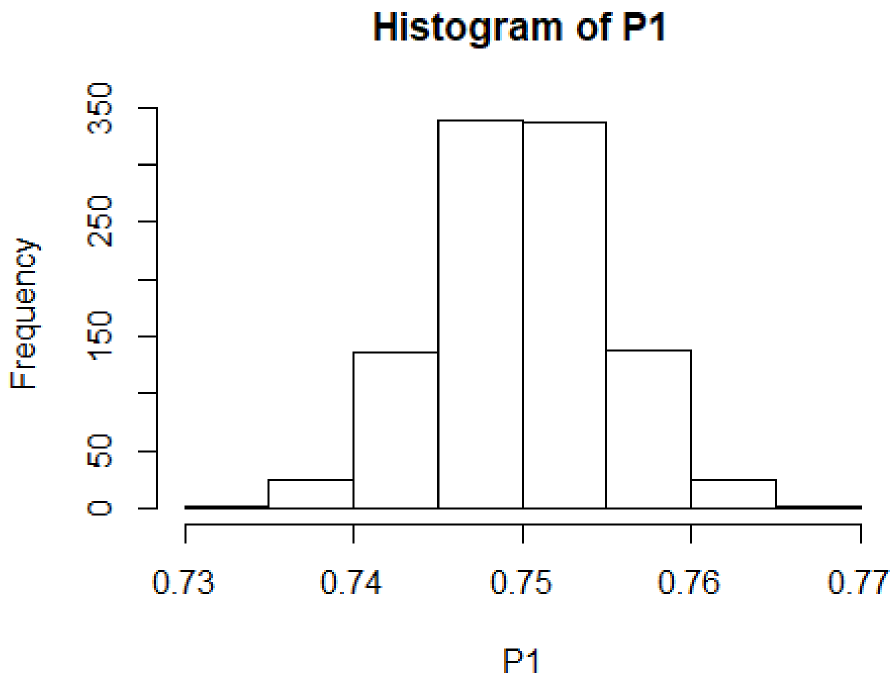
```
### Permutation approach
r0<-cor(LP,LY)
n<-length(LY)
r1<-numeric()
nt<-0
N<-1000
```

```
for(i in 1:N){
  index<-sample(n,replace=FALSE)
  x<-LP[index]
  y<-LY
  r<-cor(x,y)
  r1[i]<-r
  if(r0>0)if(r>r0)nt<-nt+1
  if(r0<0)if(r<r0)nt<-nt+1

}
(pr<-mean(r1))

## [1] 0.0014

(pv<-nt/N*2)

## [1] 0
```

Monte Carlo Test

We will use this technique for our Mendel's gene segregation example

```
R<-5474
r<-1850
Total<-R+r
N<-1000  ## simulation number,it can be a different number
B<-rbinom(N,Total,0.75) ## simulate the data given the 3:1 ratio
P0<-R/Total  ## calculate p value based on observed data
P1<-B/Total ## calculate N pavlues under H0
quantile(P1,prob=c(0.005,0.995)) ## generate 95% CI

##  0.5% 99.5%
## 0.737 0.763

hist(P1)
```



```
#If P0 is located in the CI,  it follows the segregation statistically.
```

Example for a binomial study

Here is a study to determine of the response to cold condition for barley two genotypes: wild and mutant. The total individuals used in this study were 90 for each and there were 69 and 47 survived under a cold condition for a period of time. We will use bootstrapping and permutation tests for the comparison.

```
s1=69
s2=47
n=90
N=10000
###########Bootstrapping ######
S1=rbinom(N,n,p=(s1/n))

S2=rbinom(N,n,p=(s2/n))
P1=S1/n
P2=S2/n

CI1=quantile(P1,p=c(0.025,0.975))
CI2=quantile(P2,p=c(0.025,0.975))
CI1
```

```
##  2.5% 97.5%
## 0.678 0.856

CI2

##  2.5% 97.5%
## 0.422 0.622

CI1=quantile(P1,p=c(0.005,0.995))
CI2=quantile(P2,p=c(0.005,0.995))
CI1

##  0.5% 99.5%
## 0.644 0.878

CI2

##  0.5% 99.5%
## 0.389 0.656

###########Permutation ######
p0=(s1+s2)/(2*n)
S1=rbinom(N,n,p=p0)
S2=(s1+s2)-S1
P1=S1/n
P2=S2/n
(p1=s1/n)

## [1] 0.767

(p2=s2/n)

## [1] 0.522

CI1=quantile(P1,p=c(0.025,0.975))
CI2=quantile(P2,p=c(0.025,0.975))
CI1

##  2.5% 97.5%
## 0.544 0.744

CI2

##  2.5% 97.5%
## 0.544 0.744

CI1=quantile(P1,p=c(0.005,0.995))
CI2=quantile(P2,p=c(0.005,0.995))
CI1

##  0.5% 99.5%
## 0.511 0.778

CI2
```

```
##  0.5% 99.5% 
## 0.511 0.778
```

Applications to linear regression analysis

Again, we will use the data set cotyldreg as our demonstration.

```
boot.dat=function(data){
  n=nrow(data)
  id=sample(1:n,replace=TRUE)
  dat1=data[id,]
  return(dat1)
}

reg=lm(LY~.,data=cotyldreg)
summary(reg)
```

```
## 
## Call:
## lm(formula = LY ~ ., data = cotyldreg)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -119.15  -11.68    3.44   15.96   89.88 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.34e+03   4.38e+01  -53.35   <2e-16 ***
## LP           1.51e+01   5.52e+00    2.73   0.0067 ** 
## BW           7.09e+01   3.55e+01    2.00   0.0469 *  
## BN           2.00e+00   1.95e-02  102.82   <2e-16 ***
## LS           9.79e+00   2.78e+00    3.52   0.0005 ***
## SB           2.43e+01   6.78e+00    3.59   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 29.6 on 250 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.986 
## F-statistic: 3.52e+03 on 5 and 250 DF,  p-value: <2e-16
```

```
bhat0=reg$coef
names(bhat0)
```

```
## [1] "(Intercept)" "LP"          "BW"          "BN"          "LS"         
## [6] "SB"
```

```
N=10000
###Bootstrapping #####################
BHAT=matrix(0,N,length(bhat0))
#X=cotyldreg[,-6]
```

```
#head(X)
#n=length(cotyldreg$LY)
for(i in 1:N){
  cot1=boot.dat(cotyldreg)
  #id=sample(n,replace=T)
  #y1=cotyldreg$LY[id]
  #X1=X[id,]
  #cot1=data.frame(y1,X1)
  reg1=lm(LY~.,data=cot1)
  bhat=reg1$coef
  BHAT[i,]=bhat
}
colnames(BHAT)=names(bhat0)
r=length(bhat0)
head(BHAT)

##      (Intercept)    LP   BW   BN    LS   SB
## [1,]       -2362 10.14 30.7 2.00 12.65 32.1
## [2,]       -2337 19.65 92.3 2.01  7.48 20.0
## [3,]       -2446  7.60  2.8 2.02 15.01 37.4
## [4,]       -2360  6.53 21.7 2.01 14.17 34.6
## [5,]       -2285  8.02 16.3 1.99 13.08 34.3
## [6,]       -2312 16.30 86.9 1.98  9.03 21.2

#CI=matrix(0,r,4)
CI=matrix(0,r,6)
p=c(0.025,0.975,0.005,0.995)
for(i in 1:r){
   ci=quantile(BHAT[,i],p=p,na.rm=TRUE)
   m=mean(BHAT[,i],na.rm=TRUE)
   CI[i,]=c(bhat0[i],m,ci)
}

colnames(CI)=c("Original","Boot","LL1","UL1","LL2","UL2")
rownames(CI)=names(bhat0)
CI

##             Original     Boot       LL1      UL1      LL2      UL2
## (Intercept) -2336.24 -2334.37 -2448.960 -2223.09 -2482.97 -2191.84
## LP             15.08    15.75     1.494    32.27    -3.32    37.34
## BW             70.91    76.87   -26.643   200.51   -59.82   239.47
## BN              2.00     2.00     1.952     2.06     1.94     2.07
## LS              9.79     9.39     0.430    17.05    -2.53    19.71
## SB             24.31    23.22     0.283    42.10    -7.52    48.53
```

Based on CI for each parameter, we can make statistical inference for each paramter.

Now we start use permutation for linear regression analysis

```
rand.dat=function(data){
  p=ncol(data)
  dat1=data
  for(i in 1:p){
    v=data[,i]
    dat1[,i]=sample(v)
  }
  return(dat1)
}

reg=lm(LY~.,data=cotyldreg)
summary(reg)
```

```
##
## Call:
## lm(formula = LY ~ ., data = cotyldreg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -119.15  -11.68    3.44   15.96   89.88
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.34e+03   4.38e+01  -53.35   <2e-16 ***
## LP           1.51e+01   5.52e+00    2.73   0.0067 **
## BW           7.09e+01   3.55e+01    2.00   0.0469 *
## BN           2.00e+00   1.95e-02  102.82   <2e-16 ***
## LS           9.79e+00   2.78e+00    3.52   0.0005 ***
## SB           2.43e+01   6.78e+00    3.59   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.6 on 250 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.986
## F-statistic: 3.52e+03 on 5 and 250 DF,  p-value: <2e-16
```

```
bhat0=reg$coef
names(bhat0)
```

```
## [1] "(Intercept)" "LP"          "BW"          "BN"          "LS"
## [6] "SB"
```

```
N=10000
###Permutation #####################


BHAT=matrix(0,N,length(bhat0))
X=cotyldreg[,-6]
head(X)
```

```
##     LP   BW  BN   LS   SB
## 1 41.4 5.08 634 74.3 28.3
## 2 38.6 5.49 577 68.9 30.7
## 3 38.0 6.21 533 66.9 35.3
## 4 38.2 5.27 599 69.3 29.0
## 5 40.7 5.48 748 67.4 33.0
## 6 40.3 6.23 561 73.0 34.3

n=length(cotyldreg$LY)
for(i in 1:N){
  cot1=rand.dat(cotyldreg)
  reg1=lm(LY~.,data=cot1)
  #id=sample(n,replace=FALSE)
  #y1=cotyldreg$LY[id]
  #X1=X
  #cot1=data.frame(y1,X1)
  #reg1=lm(y1~.,data=cot1)
  bhat=reg1$coef
  BHAT[i,]=bhat
}
colnames(BHAT)=names(bhat0)
r=length(bhat0)
head(BHAT)

##      (Intercept)    LP     BW       BN     LS     SB
## [1,]         677 15.41   5.94  0.06184 -0.480 -4.553
## [2,]        1918 -7.47 -20.86 -0.14404 -0.474 -7.932
## [3,]         797  3.26  14.17 -0.01295  3.897 -3.673
## [4,]        1374  3.74 -17.55  0.00434 -3.693  0.359
## [5,]         735  9.25  26.13  0.02726  1.163 -5.668
## [6,]         751  3.90  51.30  0.03927 -0.236 -0.772

CI=matrix(0,r,6)
p=c(0.025,0.975,0.005,0.995)
for(i in 1:r){
   ci=quantile(BHAT[,i],p=p,na.rm=TRUE)
   m=mean(BHAT[,i],na.rm=TRUE)
   CI[i,]=c(bhat0[i],m,ci)
}

colnames(CI)=c("Original","Random","LL1","UL1","LL2","UL2")
rownames(CI)=names(bhat0)
CI

##             Original    Random     LL1      UL1     LL2      UL2
## (Intercept) -2336.24  1.17e+03 371.789 1990.164 140.674 2236.741
## LP             15.08 -8.94e-02 -15.775   15.786 -20.000   20.247
## BW             70.91 -2.02e-01 -52.411   52.915 -66.543   67.813
## BN              2.00  1.35e-04  -0.254    0.255  -0.335    0.326
## LS              9.79 -1.80e-04  -4.390    4.472  -5.917    5.881
## SB             24.31  3.43e-02  -9.793    9.761 -12.658   12.577
```