STA 601/360 Homework 9

Yi Mi

15 十一月, 2019

Exercise 1

Hoff 9.2

Part (a)

Get data

```
women = read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/azdiabetes.dat', header = T)
```

MC

```
# data
y = as.matrix(women[, 2]); #X = as.matrix(women[, c(-2,-8)])
X = as.matrix(cbind(beta0 = rep(1,length(y)),women[, c(-2,-8)]))
n = dim(X)[1]; p = dim(X)[2]
# prior parameters: g, nu0, s20
g = length(y); nu0 = 2;
s20 = 1
# number of independent samples to generate: S
S = 1000
# MC
Hg = (g/(g+1))*X %*% solve(t(X) %*% X) %*% t(X)
SSRg = t(y) %*% (diag(1, nrow = n) - Hg) %*% y
s2 = 1/rgamma(S, (nu0 + n)/2, (nu0*s20 + SSRg)/2)
Vb = g*solve(t(X)%*%X)/(g+1)
Eb = Vb%*%t(X)%*%y
E = matrix(rnorm(S*p, 0, sqrt(s2)), S, p)
beta = t(t(E%*%chol(Vb)) + c(Eb))
```

CI

```
CI.beta = data.frame(apply(beta, 2, FUN = quantile, probs=c(0.025, 0.975)))
colnames(CI.beta) = c('intercept', names(women)[c(-2,-8)])
CI.beta %>% kable(digits = 3)
```

	intercept	npreg	bp	skin	bmi	ped	age
2.5%	35.367	-1.531	-0.030	-0.111	0.148	3.683	0.469
97.5%	71.338	0.343	0.443	0.523	1.097	17.798	1.068

```
CI.s2 = data.frame(quantile(s2, probs=c(0.025, 0.975)))
colnames(CI.s2) = 'Sigma'
CI.s2 %>% kable(digits = 3)
```

	Sigma
2.5%	748.734
97.5%	945.927

Part (b)

model averaging

```
y = as.matrix(women[, 2]);
X = as.matrix(cbind(beta0 = rep(1,length(y)),women[, c(-2,-8)]))
n = dim(X)[1]; p = dim(X)[2]
g = length(y); nu0 = 2; s20 =1
Hg = (g/(g+1))*X %*% solve(t(X) %*% X) %*% t(X)
SSRg = t(y) %*% (diag(1, nrow = n) - Hg) %*% y

set.seed(15)
# a function to compute marginal probability
lpy <- function(y, X, g = length(y),
                nu0 = 2, s20 =1){
  n <- dim(X)[1]
  p <- dim(X)[2]
  if(p == 0){Hg <- 0; s20 <- mean(y^2)}
  if(p > 0){Hg <- (g/(g + 1))*X %*% solve(t(X) %*% X) %*% t(X)}
  SSRg <- t(y) %*% (diag(1, nrow = n) - Hg) %*% y
  return(
    -0.5*(n*log(pi) + p*log(1 + g) + (nu0 + n)*log(nu0*s20 + SSRg) - nu
0*log(nu0*s20)) -
            lgamma((nu0 + n)/2) - lgamma(nu0/2)
  )
}

# starting values and MCMC setup
z <- rep(1, dim(X)[2])
lpy.c <- lpy(y, X[, z==1, drop = F])
S <- 1000
Z <- matrix(NA, S, dim(X)[2])
SIGMA <- c()
BETA <- matrix(0, S, p)

# Gibbs sampler
for (s in 1:S){
  for(j in sample(1:dim(X)[2])){
    zp <- z
    zp[j] <- 1 - zp[j]
    lpy.p <- lpy(y, X[, zp == 1, drop = F])
```

```
    r <- (lpy.p - lpy.c)*(-1)^(zp[j] == 0)
    z[j] <- rbinom(1, 1, 1/(1 + exp(-r)))
    if(z[j] == zp[j]){lpy.c <- lpy.p}
  }
  Z[s, ] <- z

  if(all(z == 0)){
    BETA[s,] = 0
    SIGMA[s] = 0
    }
  else{
    X.z = X[, z == 1, drop = F]
    Hg.z = (g/(g + 1))*X.z %*% solve(t(X.z) %*% X.z) %*% t(X.z)
    SSRg.z = t(y) %*% (diag(1, nrow = n) - Hg.z) %*% y
    s2 =  1/rgamma(1, (nu0 + n)/2, (nu0*s20 + SSRg.z)/2)
    Vb.z = g/(g + 1)*s2*solve(t(X.z) %*% X.z)
    Eb.z = g/(g + 1)*solve(t(X.z) %*% X.z) %*% t(X.z) %*% y
    beta = rmvnorm(1, Eb.z, Vb.z)

    SIGMA[s] = s2
    BETA[s, z==1] = beta
  }
}

p.x = t(data.frame(apply(BETA, 2, function(x){mean(x!=0)})))
colnames(p.x) = c('intercept', names(women)[c(-2,-8)])
rownames(p.x) = 'P(X!=0)'
p.x %>% kable(digits = 3)
```

	intercept	npreg	bp	skin	bmi	ped	age
P(X!=0)	1	0.093	0.16	0.1	0.982	0.66	1

```
CI.beta = data.frame(apply(BETA, 2, FUN = quantile, probs=c(0.025, 0.975)))
colnames(CI.beta) = c('intercept', names(women)[c(-2,-8)])
CI.beta %>% kable(digits = 3)
```

	intercept	npreg	bp	skin	bmi	ped	age
2.5%	43.311	-0.904	0.000	0.000	0.408	0.000	0.477
97.5%	76.873	0.000	0.318	0.379	1.319	16.675	1.014

```
CI.s2 = data.frame(quantile(SIGMA, probs=c(0.025, 0.975)))
colnames(CI.s2) = 'Sigma'
CI.s2 %>% kable(digits = 3)
```

	Sigma
2.5%	759.372
97.5%	958.698

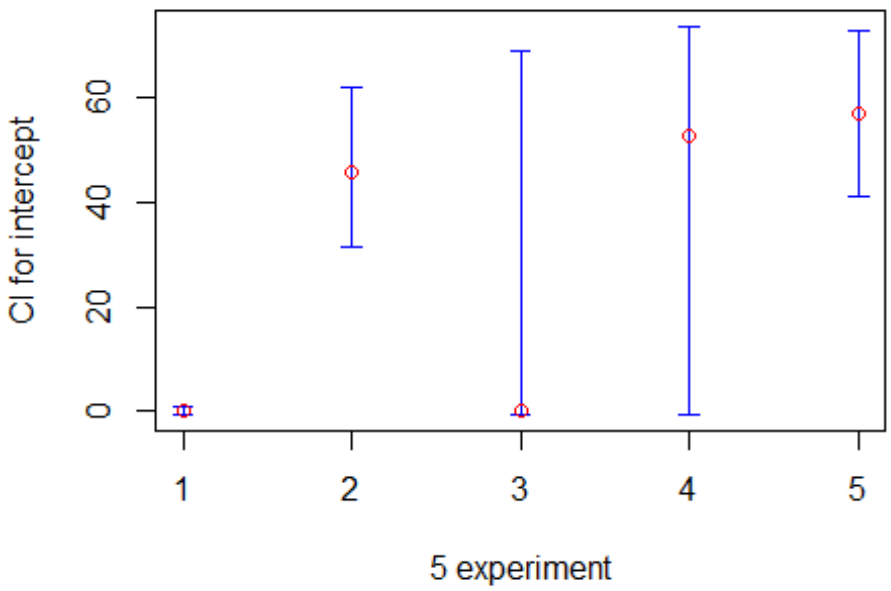
After using the model selection and averaging procedure, the CI shrink for variables `npreg`, `bp` and `skin` with lower posterior probability of not being equal to 0. However, for the variables whose posterior probabilities of not being equal to 0 are high, including `intercept`, `bmi`, `ped` and `age`, the CI do not change a lot and do not shrink.

Exercise 2

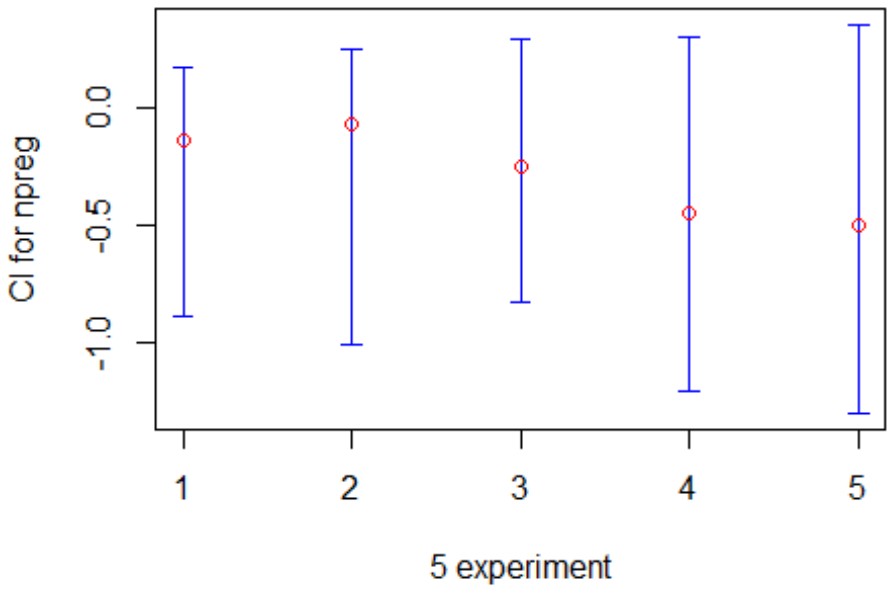
Perform the sensitivity analysis for different values of $c_j > 1$ and $\tau_j^2 > 0$.

```
women = read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/azd
iabetes.dat', header = T)
y = as.matrix(women[, 2])
X = as.matrix(cbind(beta0 = rep(1,length(y)),women[, c(-2,-8)]))

variable_selection = function(c, tau){
  S = 1000
  p = dim(X)[2]
  n = dim(X)[1]
  v = 1
  lambda = 10
  r = rbinom(p,1,1/2)
  D = diag(c^r*tau)
  R = diag(rep(1,p))
  beta.0 = solve(t(X)%*%X)%*%t(X)%*%y
  beta = rmvnorm(1, rep(0,p), D%*%R%*%D)
  s2 = 1/rgamma(1, v/2, v*lambda/2)
  BETA = matrix(NA, nrow = S, ncol = p)
  SIGMA = c()
  GAMMA = matrix(NA, nrow = S, ncol= p)

  for(s in 1:S){
    # gamma
    for(j in sample(1:p)){
      rp = r
      rp[j] = 1
      a = 0.5*dmvnorm(beta, rep(0,p), diag(c^rp*tau)%*%R%*%diag(c^rp*ta
u))
      rp[j] = 0
      b = 0.5*dmvnorm(beta, rep(0,p), diag(c^rp*tau)%*%R%*%diag(c^rp*ta
u))
      r[j] = rbinom(1,1,a/(a+b))
    }
    # beta
    A = solve(t(X) %*% X/s2 + diag(1/(c^r*tau)) %*% solve(R) %*% diag(1
/(c^r*tau)))
    Vb = A
    Eb = A %*% t(X) %*% X %*% beta.0/s2
    beta = rmvnorm(1, Eb, Vb)
    # sigma
```

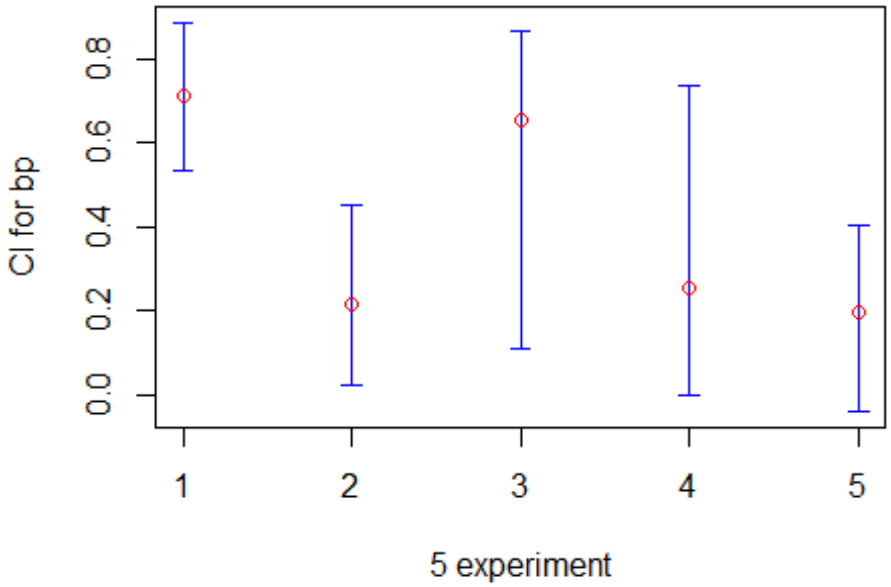
```
    s2 =1/rgamma(1, (n + v)/2, (v*lambda + t(y - X%*%t(beta))%*%(y - X%
*%t(beta)))/2)
    # storing values
    BETA[s,] = beta
    SIGMA[s] = s2
    GAMMA[s,] = r
  }
  LIST = list(BETA,SIGMA,GAMMA)
  LIST
}

C = c(10, 100, 1000, 10000, 10000)
TAU = c(0.04,0.16,0.36,0.64,0.81)

CI_intercept = matrix(NA, nrow = 5, ncol = 3)
CI_npreg = matrix(NA, nrow = 5, ncol = 3)
CI_bp = matrix(NA, nrow = 5, ncol = 3)
CI_skin = matrix(NA, nrow = 5, ncol = 3)
CI_bmi = matrix(NA, nrow = 5, ncol = 3)
CI_ped = matrix(NA, nrow = 5, ncol = 3)
CI_age = matrix(NA, nrow = 5, ncol = 3)
CI_s2 = matrix(NA, nrow = 5, ncol = 3)

for(i in 1:5){
  RESULT = variable_selection(C[i], TAU[i])
  BETA = RESULT[[1]]
  SIGMA = RESULT[[2]]

  CI_intercept[i,]=quantile(BETA[,1],probs=c(0.025, 0.5, 0.975))
  CI_npreg[i,]=quantile(BETA[,2],probs=c(0.025, 0.5, 0.975))
  CI_bp[i,]=quantile(BETA[,3],probs=c(0.025, 0.5, 0.975))
  CI_skin[i,]=quantile(BETA[,4],probs=c(0.025, 0.5, 0.975))
  CI_bmi[i,]=quantile(BETA[,5],probs=c(0.025, 0.5, 0.975))
  CI_ped[i,]=quantile(BETA[,6],probs=c(0.025, 0.5, 0.975))
  CI_age[i,]=quantile(BETA[,7],probs=c(0.025, 0.5, 0.975))

  CI_s2[i,] = quantile(SIGMA, probs=c(0.025, 0.5, 0.975))

}

plotCI(x=c(1:5), y = CI_intercept[,2], ui = CI_intercept[,3], li = CI_i
ntercept[,1],xlab = '5 experiment', ylab = 'CI for intercept',col="red",
scol="blue")
```



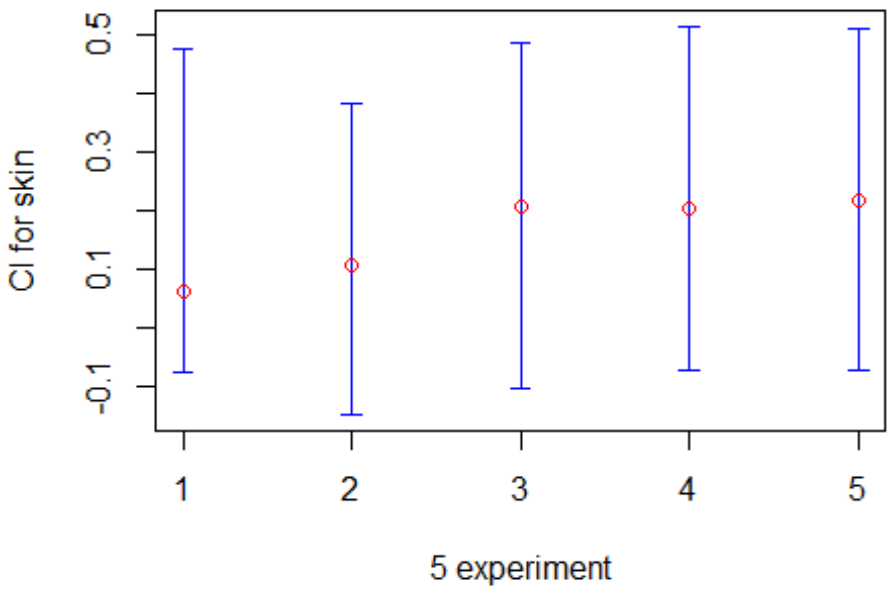

```
plotCI(x=c(1:5), y = CI_npreg[,2], ui = CI_npreg[,3], li = CI_npreg[,1],
xlab = '5 experiment', ylab = 'CI for npreg',col="red",scol="blue")
```



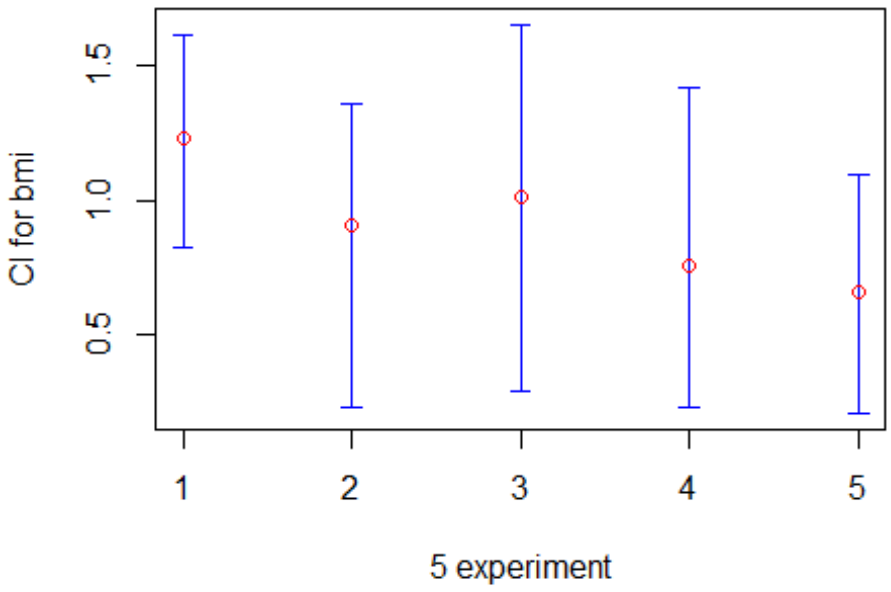
```
plotCI(x=c(1:5), y = CI_bp[,2], ui = CI_bp[,3], li = CI_bp[,1],xlab = '
5 experiment', ylab = 'CI for bp',col="red",scol="blue")
```



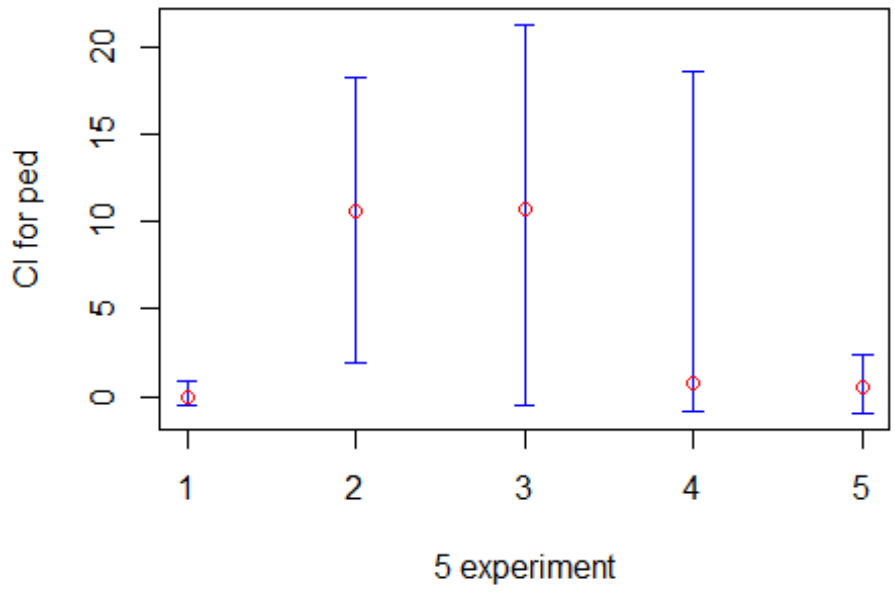

```
plotCI(x=c(1:5), y = CI_skin[,2], ui = CI_skin[,3], li = CI_skin[,1],xl
ab = '5 experiment', ylab = 'CI for skin',col="red",scol="blue")
```



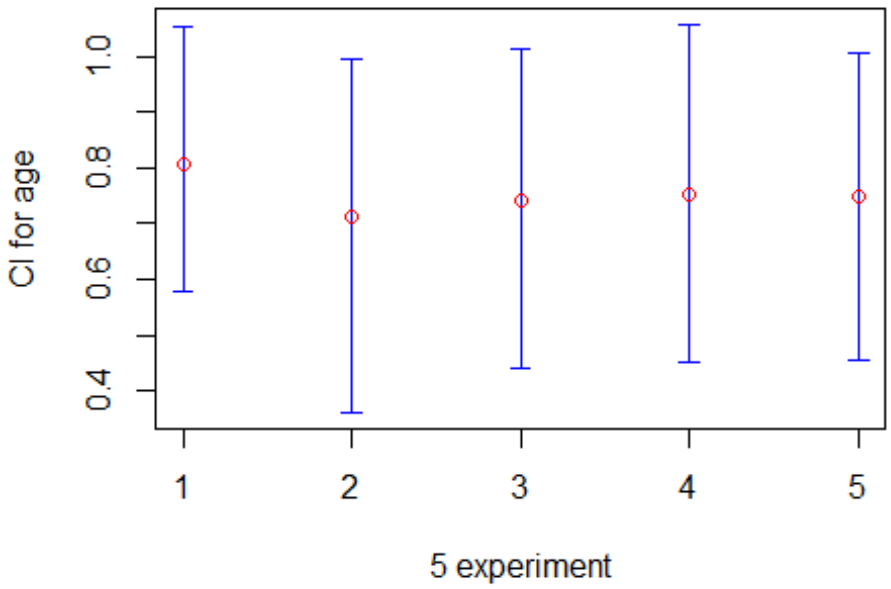

```
plotCI(x=c(1:5), y = CI_bmi[,2], ui = CI_bmi[,3], li = CI_bmi[,1],xlab
= '5 experiment', ylab = 'CI for bmi',col="red",scol="blue")
```



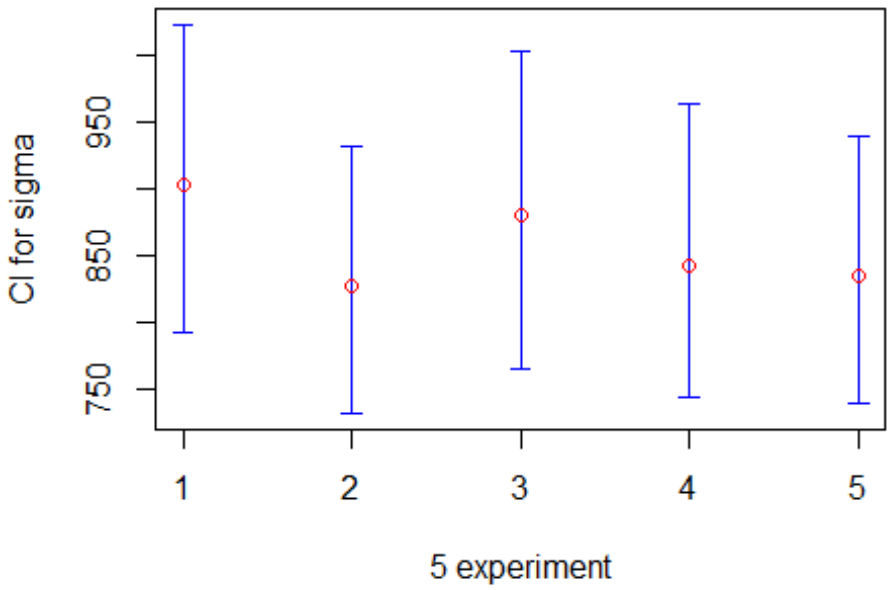
```
plotCI(x=c(1:5), y = CI_ped[,2], ui = CI_ped[,3], li = CI_ped[,1],xlab
= '5 experiment', ylab = 'CI for ped',col="red",scol="blue")
```




```
plotCI(x=c(1:5), y = CI_age[,2], ui = CI_age[,3], li = CI_age[,1],xlab
= '5 experiment', ylab = 'CI for age',col="red",scol="blue")
```




```
plotCI(x=c(1:5), y = CI_s2[,2], ui = CI_s2[,3], li = CI_s2[,1],xlab = '
5 experiment', ylab = 'CI for sigma',col="red",scol="blue")
```



I chose c from {10,100,1000,10000,10000} and τ from {0.2,0.4,0.6,0.8,0.9}. From the plot of the confidence interval for each parameters, we can see that the posterior value of `intercept`, `ped` and `bp` are very sensitive to the change of c and τ, meaning that the choice of prior has big impact on the posterior, and other parameters seem to be stable.