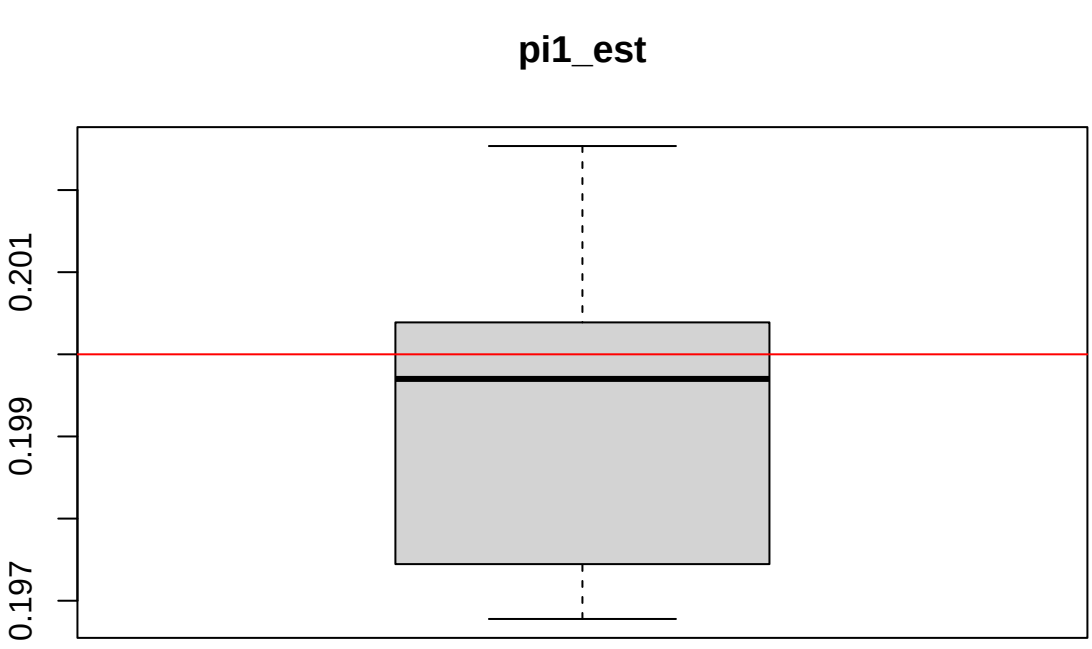
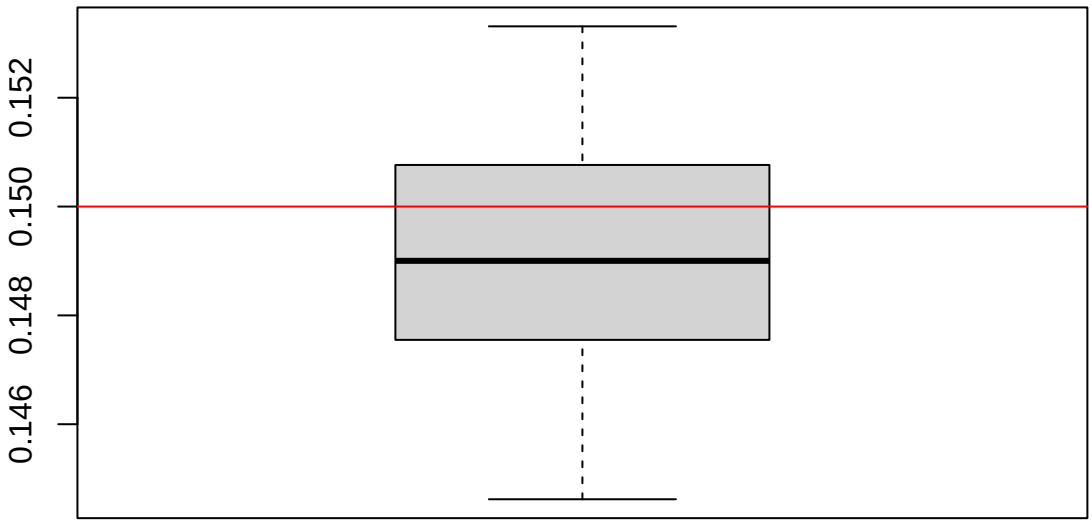
```r
  L_ini <- log_X(P, pi1_ini, alpha_ini) #

## Iteration
# Set maximum number of iterations max_iter=10000
## Perform E and M steps in each iteration, compute the log-likelihood, and see if the algorithm conver
  max_iter <- 10000 # Set the maximum number of iterations to prevent non-convergence
  for (iter in 1:max_iter){
    if (iter == 1){
      pi1_old <- pi1_ini
      alpha_old <- alpha_ini
      L_old <- L_ini
    }
## E step
    comp_gamma <- pi1_old*dbeta(P, alpha_old, 1)
    gamma <- comp_gamma/(comp_gamma + (1 - pi1_old)*dunif(P, 0, 1))
## M step
    pi1_new <- mean(gamma)
    alpha_new <- -sum(gamma)/sum(gamma*log(P))
## compute log likelihood
    L_new <- log_X(P, pi1_new, alpha_new)
## whether the algorithm converges
    if (L_new < L_old){
      print("Error: log likelihoood is not increasing!")
      break
    }
    if ((L_new - L_old)/abs(L_new) < tol){
      pi1_est <- pi1_new
      alpha_est <- alpha_new
      break
    }
    else {
      pi1_old <- pi1_new
      alpha_old <- alpha_new
      L_old <- L_new
    }
  }
   return(list(pi1 = pi1_new, alpha = alpha_new))
}

rep <- 20
FDP1 <- numeric(rep)
power1 <- numeric(rep)
FDP2 <- numeric(rep)
power2 <- numeric(rep)
pi1_est <- numeric(rep)
pi2_est <- numeric(rep)
alpha1_est <- numeric(rep)
alpha2_est <- numeric(rep)

rep=20
for (i in 1:rep){
 set.seed(i)
 data <- data_generate1(M,0.7,0.1,0.15,0.05,0.2,0.2)
```

```
 est1 <- EM(data$P1, 0.1, 0.1)
 est2 <- EM(data$P2, 0.1, 0.1)

 pi1_est[i] <- est1$pi1
 alpha1_est[i] <- est1$alpha
 pi2_est[i] <- est2$pi1
 alpha2_est[i] <- est2$alpha

 posterior1 <- (est1$pi1*est1$alpha*data$P1^(est1$alpha - 1))/(est1$pi1*est1$alpha*data$P1^(est1$alpha
 posterior2 <- (est2$pi1*est2$alpha*data$P2^(est2$alpha - 1))/(est2$pi1*est2$alpha*data$P2^(est2$alpha

 Z_est1 <- assoc(posterior1, 0.1)
 Z_est2 <- assoc(posterior2, 0.1)
 t1 <- table(Z_est1, data$Z11+data$Z10)
 t2 <- table(Z_est2, data$Z11+data$Z01)

 FDP1[i] <- t1[2, 1]/(t1[2, 1] + t1[2, 2])
 power1[i] <- t1[2, 2]/(t1[1, 2] + t1[2, 2])
 FDP2[i] <- t2[2, 1]/(t2[2, 1] + t2[2, 2])
 power2[i] <- t2[2, 2]/(t2[1, 2] + t2[2, 2])
}
```

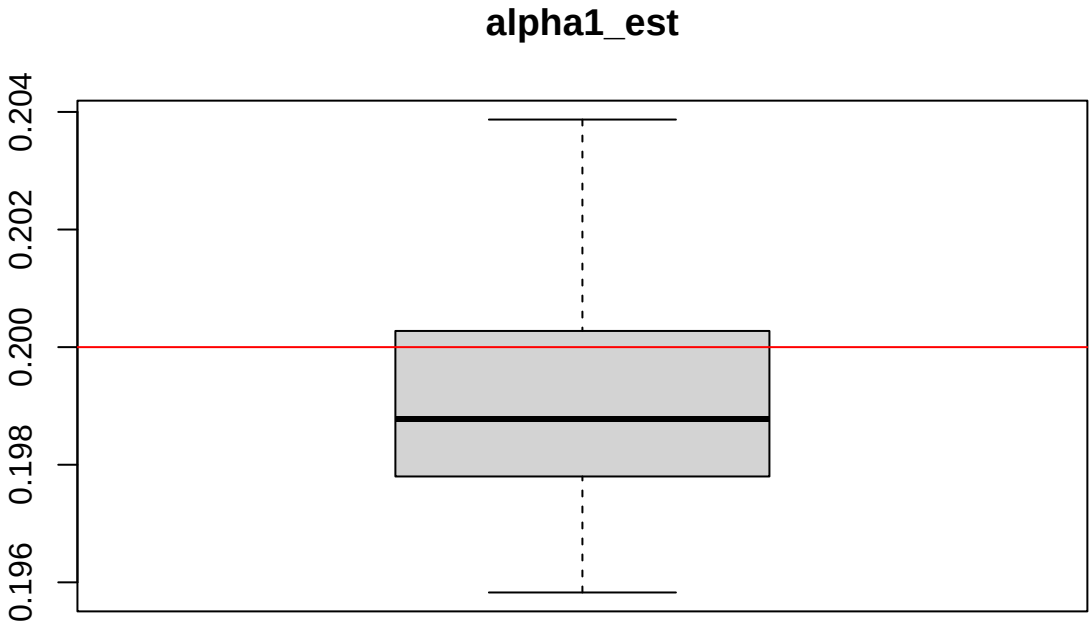
```
boxplot(pi1_est)
title("pi1_est")
abline(h=0.2,col="red")
```



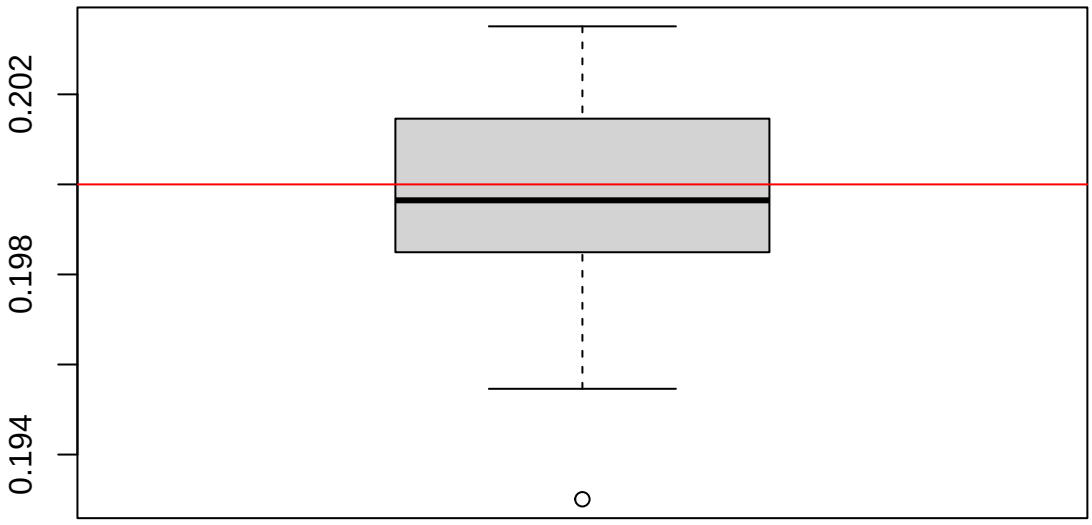

```
boxplot(pi2_est)
title("pi2_est")
abline(h=0.15,col="red")
```



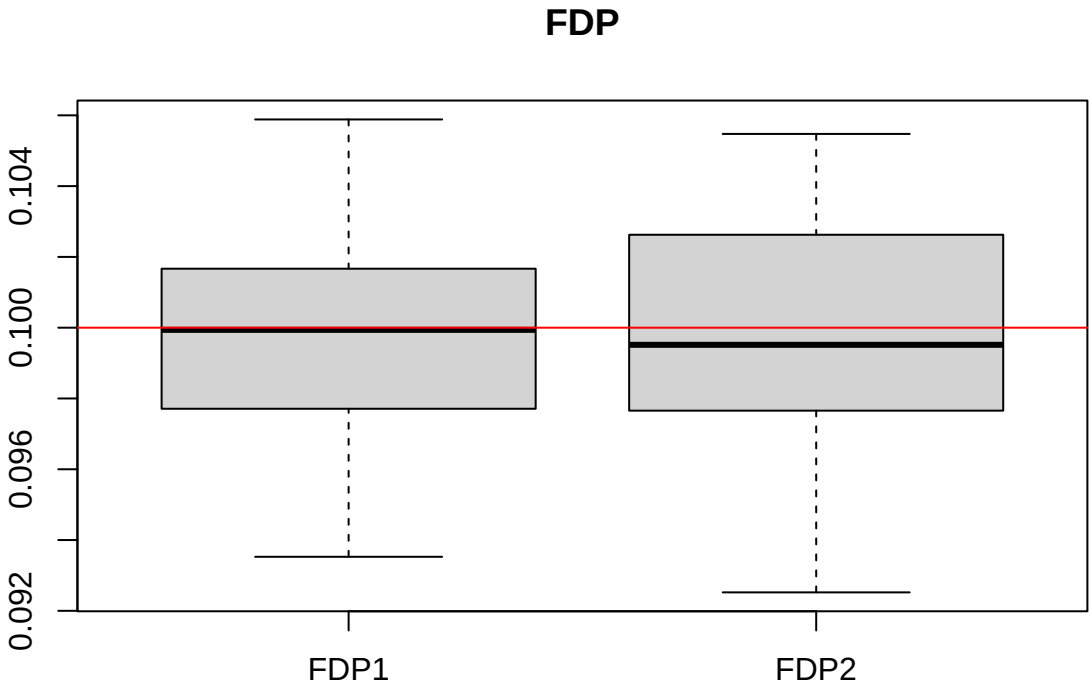

```
boxplot(alpha1_est)
title("alpha1_est")
abline(h=0.2,col="red")
```



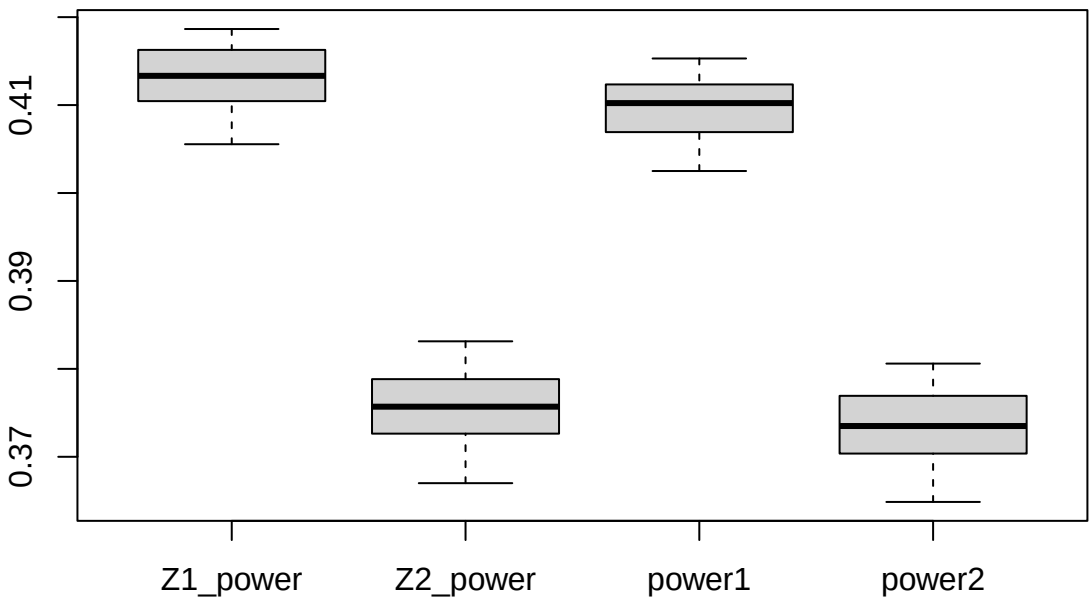

```
boxplot(alpha2_est)
title("alpha2_est")
abline(h=0.2,col="red")
```




```
FDP_SUM <- cbind(FDP1,FDP2)
boxplot(FDP_SUM)
title("FDP")
abline(h=0.1,col="red")
```



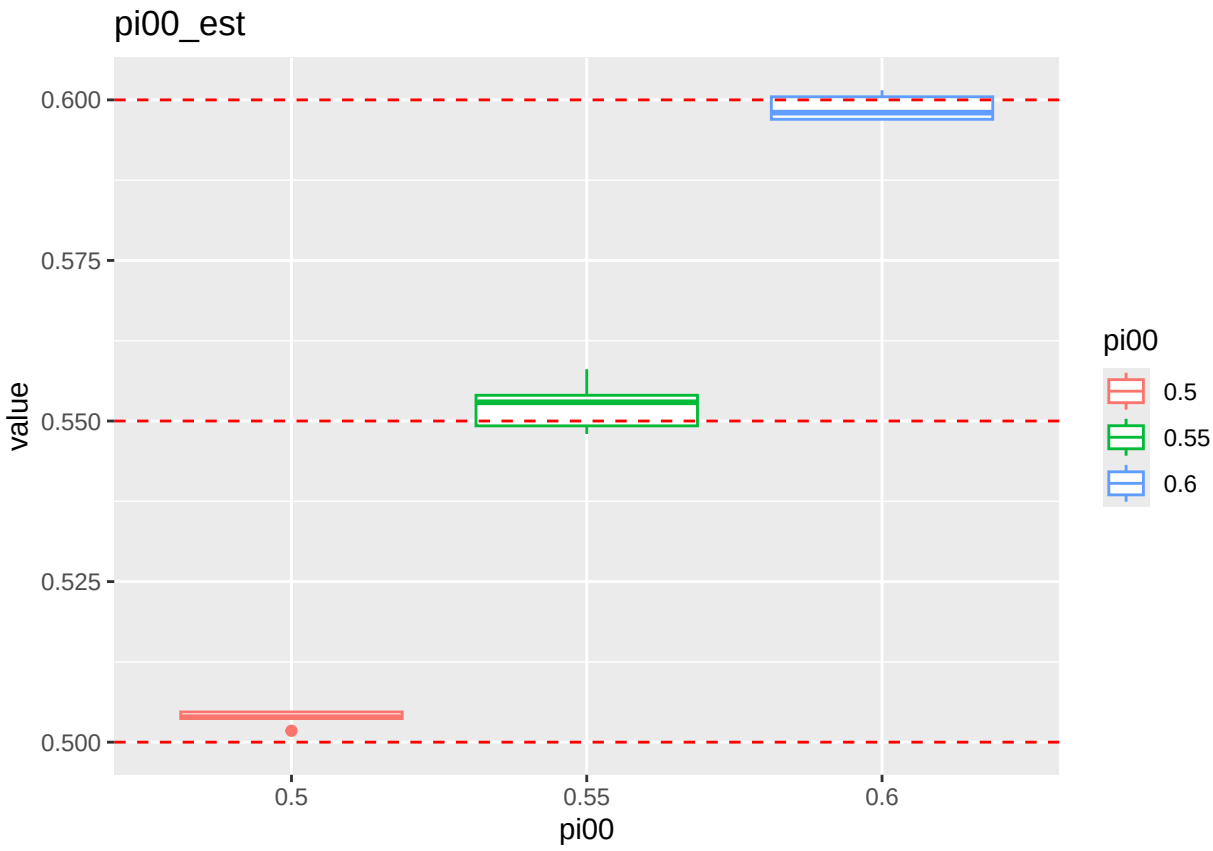

```
power_SUM <- cbind(Z1_power,Z2_power,power1,power2)
boxplot(power_SUM)
title("power")
```



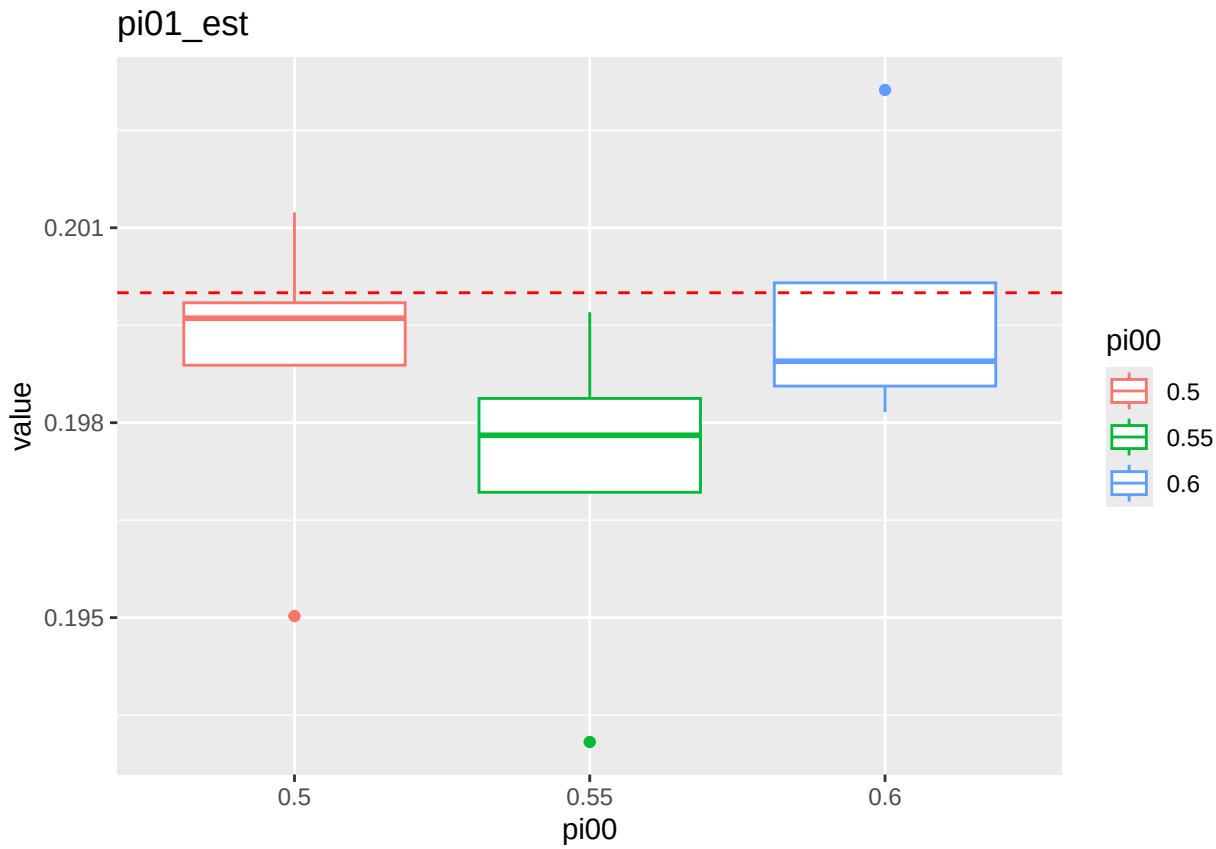

```
# Keep pi11 and pi01 unchanged, change pi00 and pi10
pi00 <- c(0.5,0.55,0.6)
pi10 <- rep(c(0.15,0.1,0.05),each=5 )
rep=5
pi00_est <- pi01_est <- pi10_est <- pi11_est <- alpha1_est <- alpha2_est <- Z11_FDP <-Z1_FDP <- Z2_FDP <
for (i in 1:(length(pi00)*rep)){
  set.seed(i)
  data <- data_generate1(100000,pi00_est$pi00[i],0.2,pi10[i],0.15,0.2,0.2)
  theta_est <- EM1(data$P1,data$P2,pi00_ini=0.5,pi01_ini=0.1,pi10_ini=0.1,pi11_ini=0.1,alpha1_ini=0.1,al
  pi00_est$value[i] <- theta_est$pi00
  pi01_est$value[i] <- theta_est$pi01
  pi10_est$value[i] <- theta_est$pi10
  pi11_est$value[i] <- theta_est$pi11
  alpha1_est$value[i] <- theta_est$alpha1
  alpha2_est$value[i] <- theta_est$alpha2
   gamma_post <- theta_est$pi00+theta_est$pi01*theta_est$alpha2*data$P2^(theta_est$alpha2-1)+theta_est$a
  z11_post <- theta_est$pi11*theta_est$alpha1*theta_est$alpha2*data$P1^(theta_est$alpha1-1)*data$P2^(the
  z01_post <- theta_est$pi01*theta_est$alpha2*data$P2^(theta_est$alpha2-1)/gamma_post
  z10_post <- theta_est$alpha1*data$P1^(theta_est$alpha1-1)*theta_est$pi10/gamma_post
  z00_post <- theta_est$pi00/gamma_post
  z1_post <- z11_post+z10_post
  z2_post <- z11_post+z01_post
  Z_est_11 <- assoc(z11_post, 0.1)
  t11<-table(Z_est_11, data$Z11)
  Z11_FDP$value[i] <- t11[2, 1]/(t11[2, 1] + t11[2, 2])
  Z11_power$value[i] <-t11[2,2]/(t11[1,2]+t11[2,2])
  Z_est_1 <- assoc(z1_post, 0.1)
  t1<-table(Z_est_1, data$Z11+data$Z10)
  Z1_FDP$value[i] <- t1[2, 1]/(t1[2, 1] + t1[2, 2])
  Z1_power$value[i] <-t1[2,2]/(t1[1,2]+t1[2,2])
  Z_est_2 <- assoc(z2_post, 0.1)
  t2<-table(Z_est_2, data$Z11+data$Z01)
```

```
  Z2_FDP$value[i] <- t2[2, 1]/(t2[2, 1] + t2[2, 2])
  Z2_power$value[i] <-t2[2,2]/(t2[1,2]+t2[2,2])
}
```

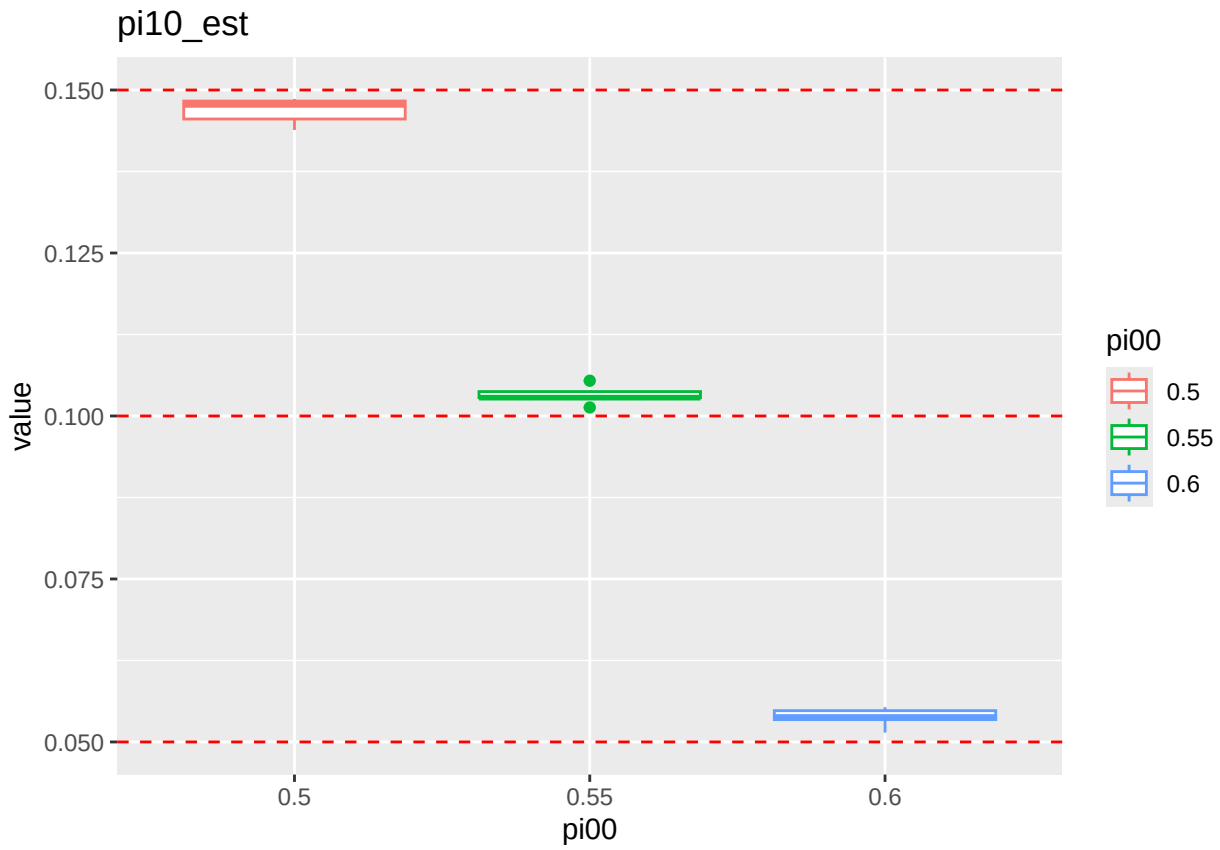
```
library(ggplot2)
pi00_est$pi00 <- as.factor(pi00_est$pi00)
ggplot(pi00_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = pi00,linetype="da
```



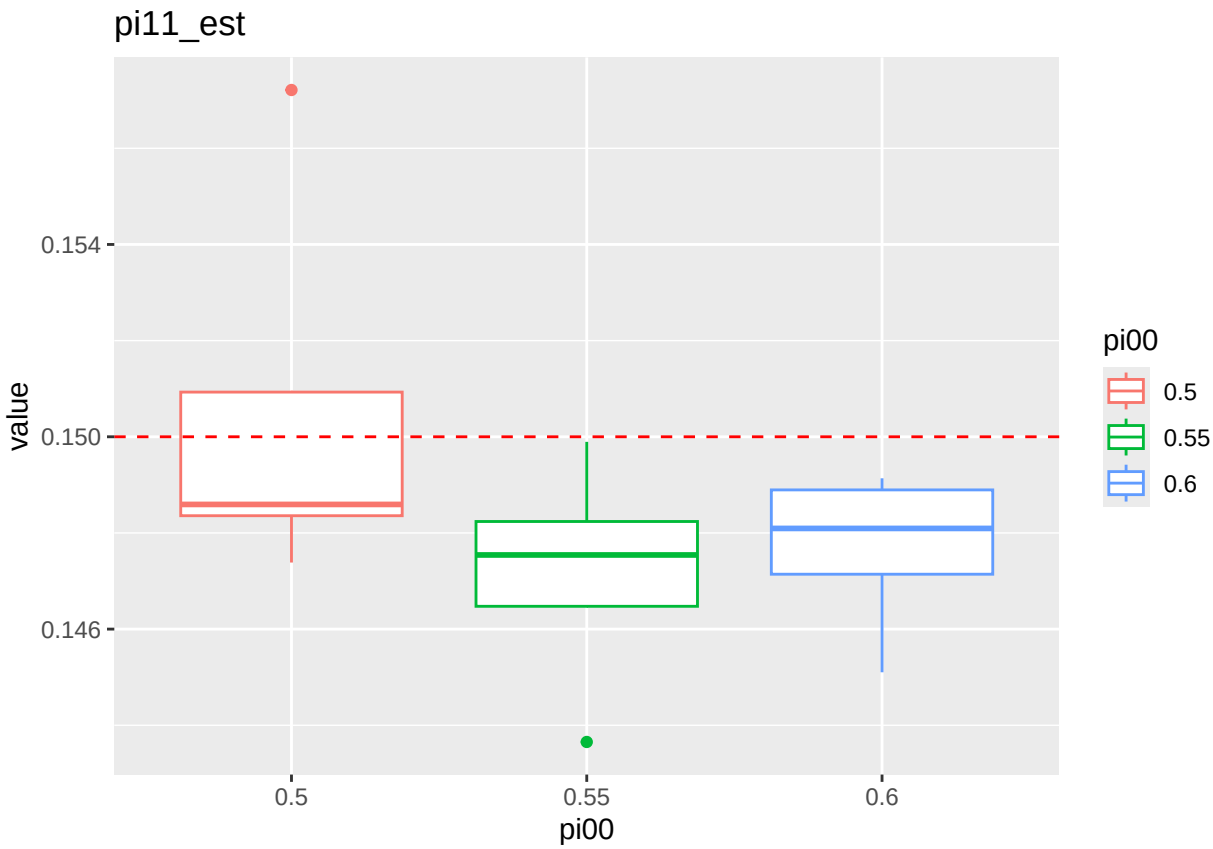

```
library(ggplot2)
pi01_est$pi00 <- as.factor(pi01_est$pi00)
ggplot(pi01_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = 0.2,linetype="das
```



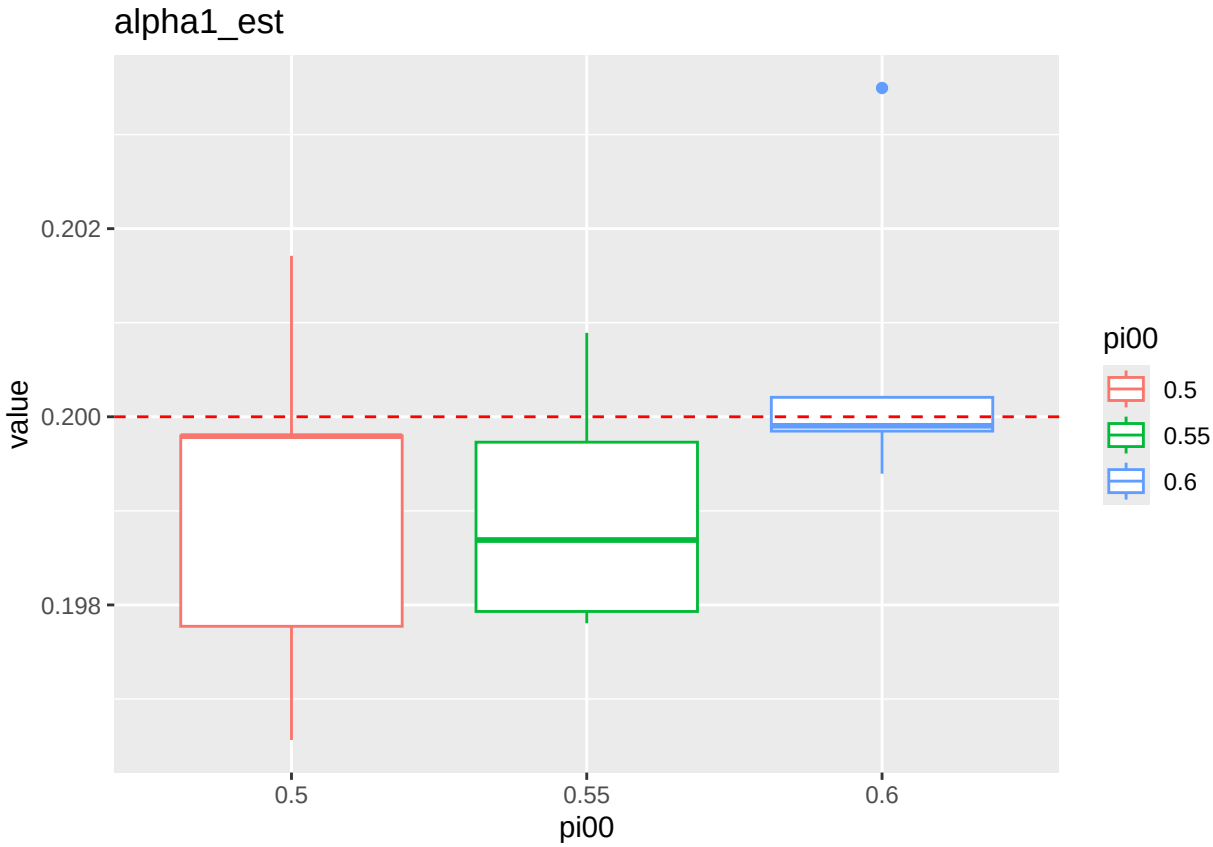

```
library(ggplot2)
pi10_est$pi00 <- as.factor(pi10_est$pi00)
ggplot(pi10_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = c(0.15,0.1,0.05)
```



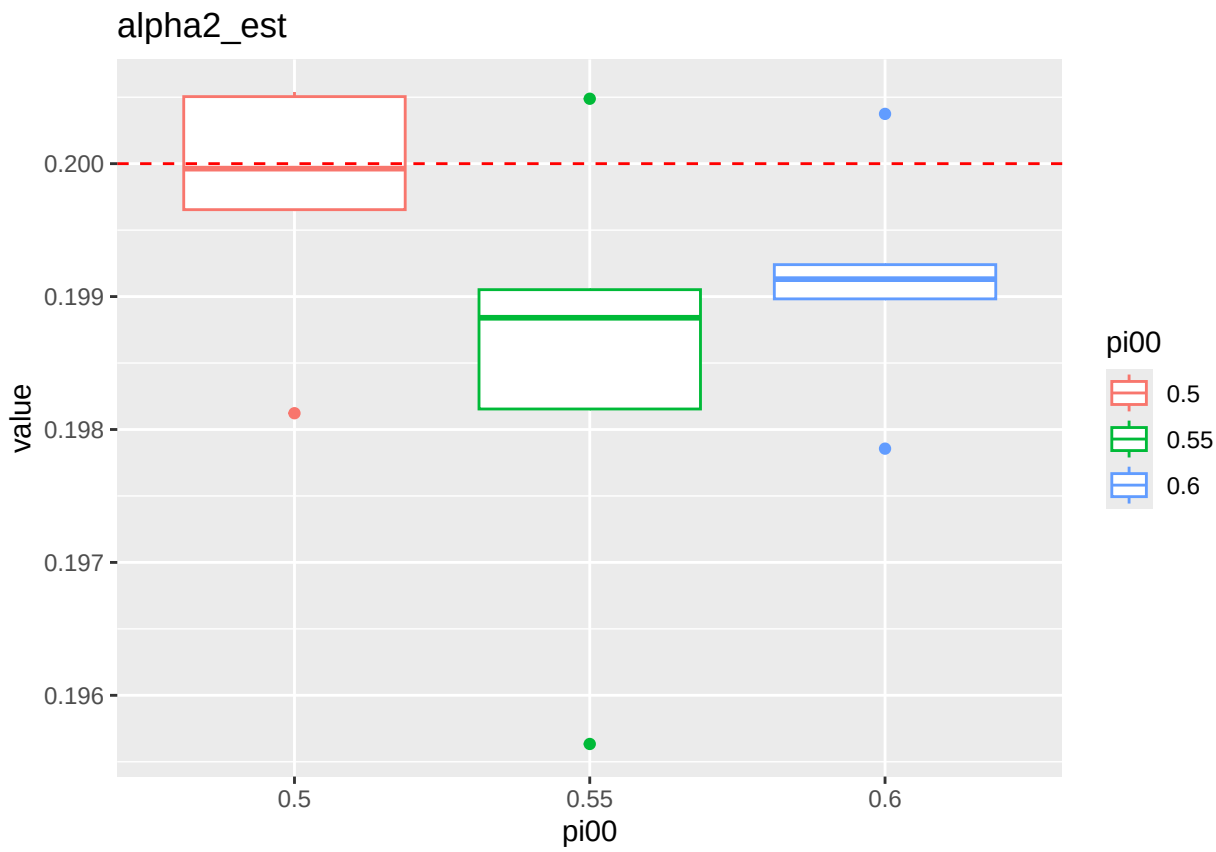

```
library(ggplot2)
pi11_est$pi00 <- as.factor(pi11_est$pi00)
ggplot(pi11_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = 0.15,linetype="da
```



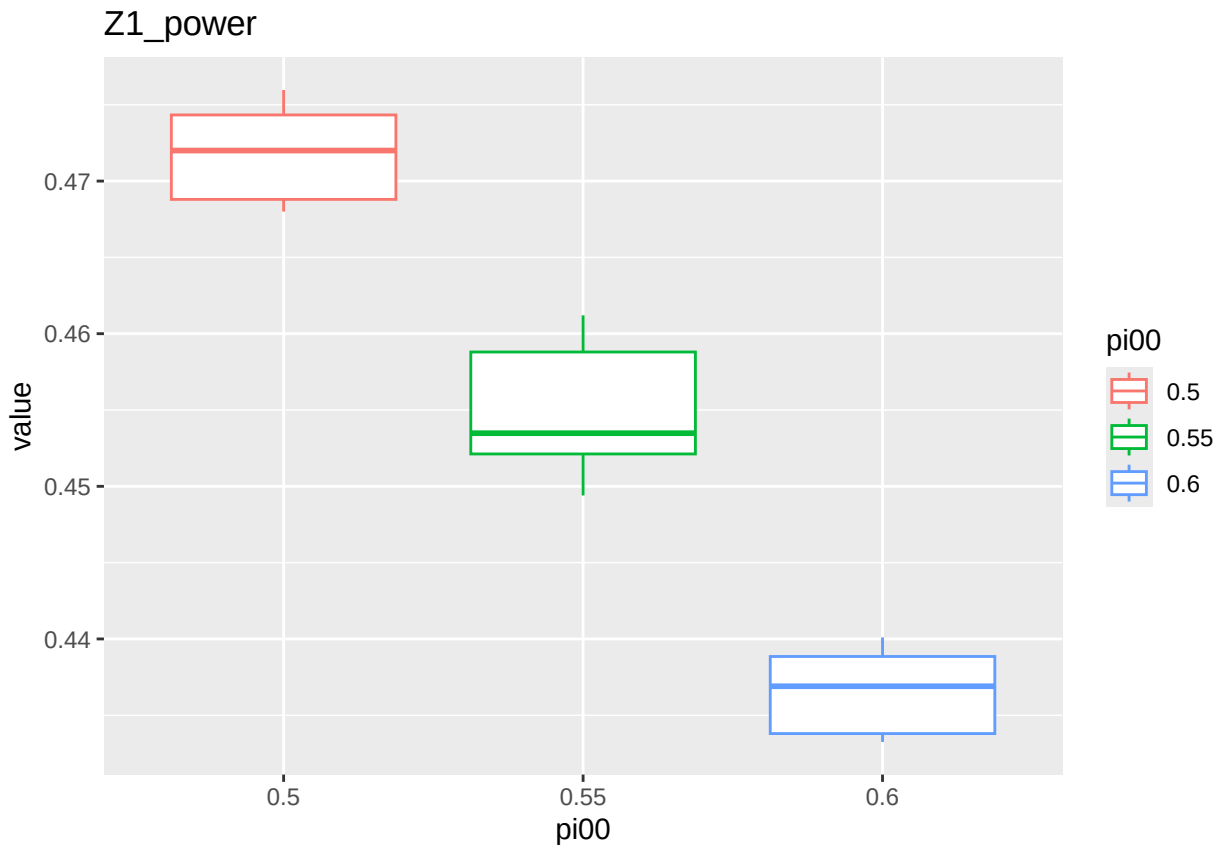

```
library(ggplot2)
alpha1_est$pi00 <- as.factor(alpha1_est$pi00)
ggplot(alpha1_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = 0.2,linetype="
```



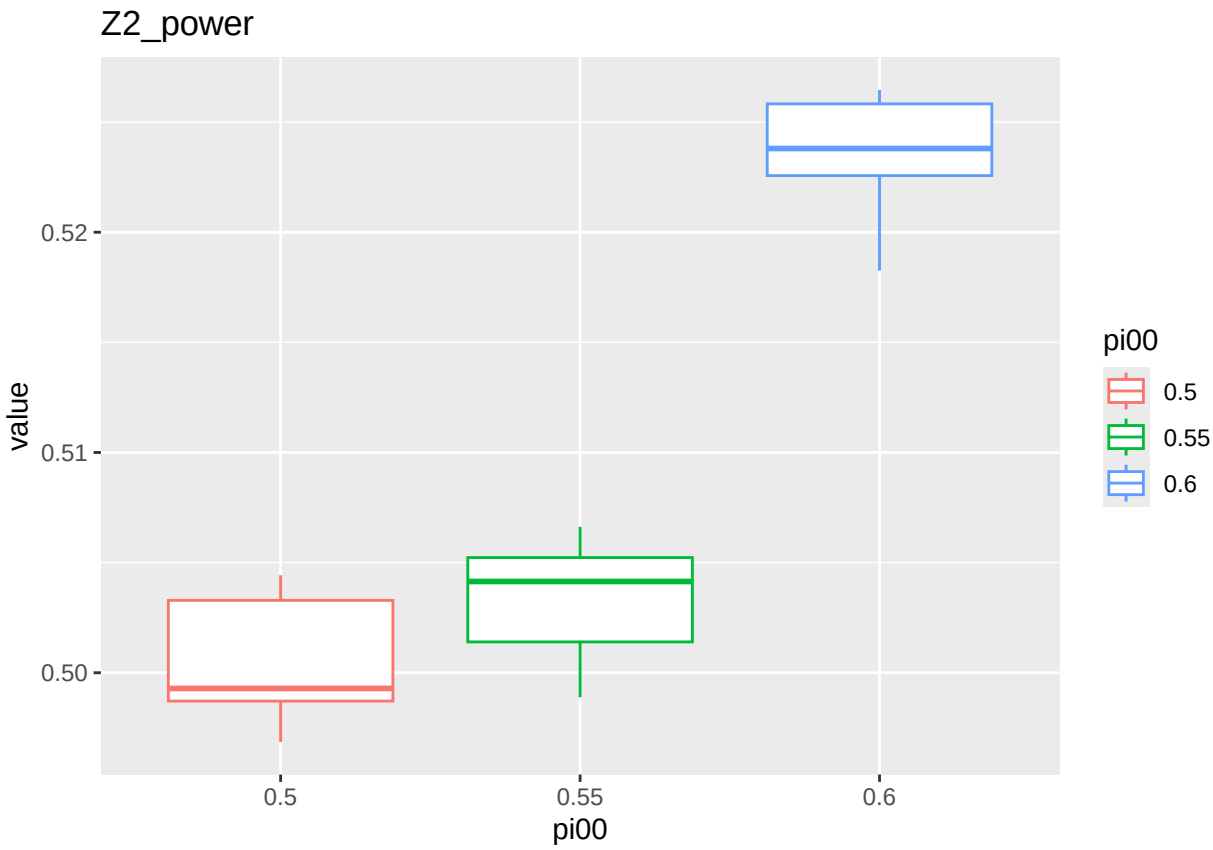

```
library(ggplot2)
alpha2_est$pi00 <- as.factor(alpha2_est$pi00)
ggplot(alpha2_est,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+geom_hline(yintercept = 0.2,linetype="
```



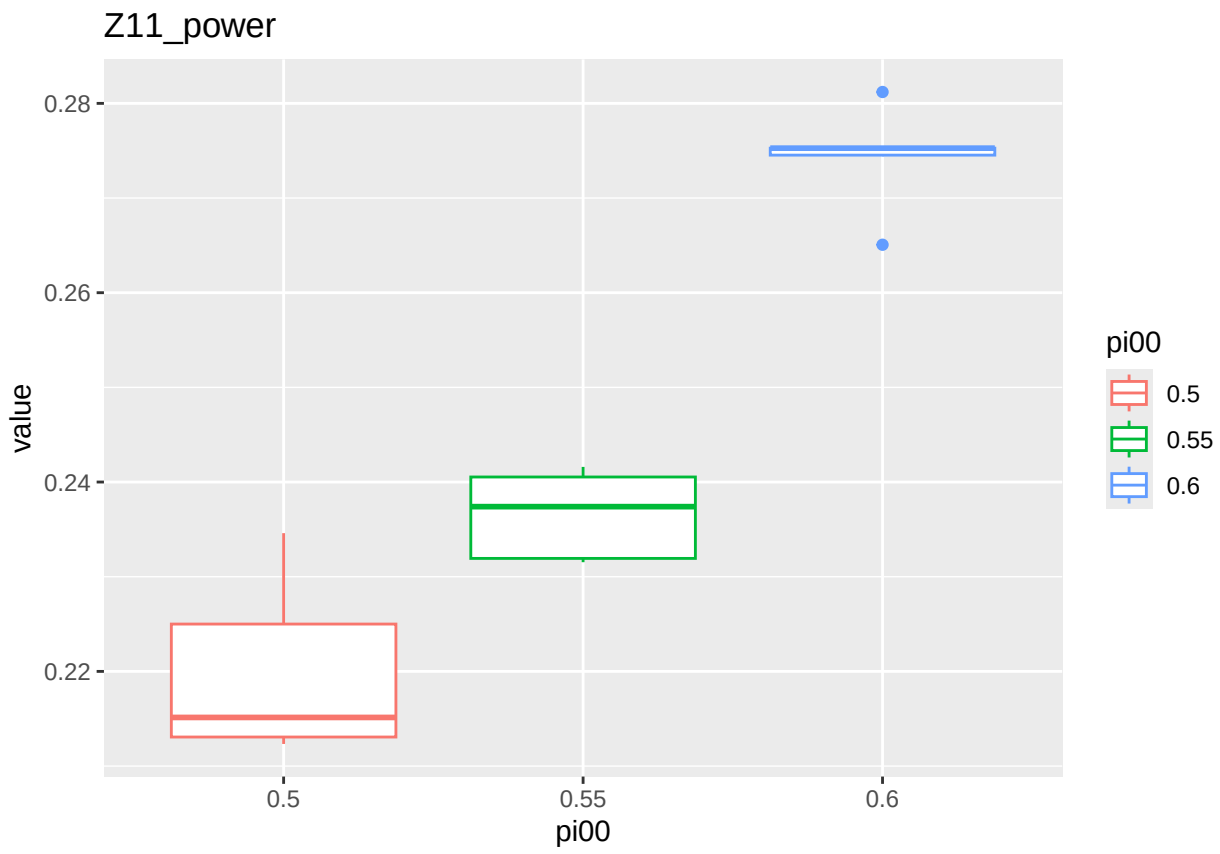

```
library(ggplot2)
Z1_power$pi00 <- as.factor(Z1_power$pi00)
ggplot(Z1_power,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+ggtitle("Z1_power")
```




```
library(ggplot2)
Z2_power$pi00 <- as.factor(Z2_power$pi00)
ggplot(Z2_power,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+ggtitle("Z2_power")
```




```
library(ggplot2)
Z11_power$pi00 <- as.factor(Z11_power$pi00)
ggplot(Z11_power,aes(y=value,x=pi00,color=pi00))+geom_boxplot()+ggtitle("Z11_power")
```



real data

```
BIP <- read.table("E:\\qjy\\ecnu\\   \\  \\    2\\tw2_data\\pgc.bip.full.2012-04.txt", header = T)
```

```
SCZ <- read.table("E:\\qjy\\ecnu\\   \\  \\    2\\tw2_data\\pgc.scz.full.2012-04.txt", header = T)
```

```
library(dplyr)
```

```
##
##     'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Take the intersection
snp_result=intersect(BIP$snpid,SCZ$snpid)
pvalue_bip <- BIP[BIP$snpid %in% snp_result,]$pval
```