**a**

> pval

[,1]

[1,] 6.583623e-14

P-value is much smaller than 0.05, so reject the H0 hypothesis, means the two groups are not the same when the variance are equal

**b**

> pval

[,1]

[1,] 2.509104e-14

P-value is much smaller than 0.05, so reject the H0 hypothesis, means the two groups are not the same when the variance are not equal

**codes**

wcgs2=read.table("wcgs2.dat",header=T)

wcgs=wcgs2[,1:5]

wcgs=as.matrix(wcgs)

WCGS=wcgs[,1:4]

WCGS =as.matrix(WCGS)

non= wcgs[wcgs[,5] == 0,]

ever =wcgs[wcgs[,5] == 1,]

wcgs.nonsmoke=non[,1:4]

wcgs.smoke =ever[,1:4]

n1 = dim(wcgs.nonsmoke)[1]

n2 = dim(wcgs.smoke)[1]

p = 4

n = n1+n2

muhat.nonsmoke = colMeans(wcgs.nonsmoke)

muhat.smoke = colMeans(wcgs.smoke)

pooled.cov = var(WCGS)\*(1/n1+1/n2)

# Statistic

T2 = (muhat.nonsmoke- muhat.smoke) %\*% solve(pooled.cov) %\*% (muhat.nonsmoke- muhat.smoke)

pval = 1-pf((n1+n2-p-1)\*T2/((n1+n2-2)\*p),p,n1+n2-p-1)

# Assume unequal variances

unpooled.cov = var(wcgs.nonsmoke)/n1 + var(wcgs.smoke)/n2

T2.unpooled = (muhat.nonsmoke- muhat.smoke) %\*% solve(unpooled.cov) %\*% (muhat.nonsmoke - muhat.smoke)

pval = 1-pchisq(T2.unpooled,p)

**c**

**for non smoke**

cr1.boot

[,1] [,2]

[1,] 45.90743 46.56469

[2,] 69.63375 69.91216

[3,] 171.25914 173.51369

[4,] 127.67212 129.58332

> cr2.boot

[,1] [,2]

age 45.92182 46.58544

height 69.63475 69.93789

weight 171.18263 173.58008

sbp 127.64814 129.54193

**for smoke**

cr1.boot

[,1] [,2]

[1,] 45.98759 46.65900

[2,] 69.61706 69.92057

[3,] 165.99387 168.62545

[4,] 127.76929 129.70251

> cr2.boot

[,1] [,2]

age 45.94911 46.66341

height 69.61962 69.91699

weight 165.87751 168.68973

sbp 127.71671 129.63216

**codes**

# for nonsmoke

# Apply bootstrap

B = 500

# Apply to each component of the vector

# Initialize

bmat = matrix(0,nrow=B,ncol=4)

set.seed(2838)

for (i in 1:B) {btma = wcgs.nonsmoke [sample(1:1652,replace=T),]

# Estimate mean

bmat[i,] = colMeans(btma)}

# apply sort

cr1.boot = cbind(apply(bmat,2,quantile,0.025/4),apply(bmat,2,quantile,1-0.025/4))

# apply standard deviation rule

se.boot = sqrt(apply(bmat,2,var))

cr2.boot = cbind(colMeans(wcgs.nonsmoke)-qnorm(1-0.025/4)\*se.boot,colMeans(wcgs.nonsmoke)+qnorm(1-0.025/4)\*se.boot)

# for smoke

B = 500

# Apply to each component of the vector

# Initialize

bmat = matrix(0,nrow=B,ncol=4)

set.seed(2838)

for (i in 1:B) {

btma = wcgs.smoke [sample(1:1502,replace=T),]

# Estimate mean

bmat[i,] = colMeans(btma)}

# apply sort

cr1.boot = cbind(apply(bmat,2,quantile,0.025/4),apply(bmat,2,quantile,1-0.025/4))

# apply standard deviation rule

se.boot = sqrt(apply(bmat,2,var))

cr2.boot = cbind(colMeans(wcgs.smoke)-qnorm(1-0.025/4)\*se.boot,colMeans(wcgs.smoke)+qnorm(1-0.025/4)\*se.boot)

**d**

**# for nonsmoke**

cr1.boot

[,1] [,2]

[1,] 1.5480245 1.6589121

[2,] 1.1527163 1.2726384

[3,] 0.7384731 0.8538380

[4,] 0.3533561 0.4364327

> cr2.boot

[,1] [,2]

age 46.19364 46.31363

height 69.71959 69.85305

weight 172.32171 172.44100

sbp 128.55285 128.63723

**for smoke**

cr1.boot

[,1] [,2]

[1,] 1.5391730 1.6829341

[2,] 1.0886004 1.2192241

[3,] 0.7442079 0.8632732

[4,] 0.3879583 0.4795972

> cr2.boot

[,1] [,2]

age 46.23401 46.37850

height 69.69971 69.83691

weight 167.22014 167.34711

sbp 128.62758 128.72129

**codes**

# for nonsmoke

B = 500

# Apply to each component of the vector

# Initialize

bmat = matrix(0,nrow=B,ncol=4)

set.seed(2838)

for (i in 1:B) {btma = wcgs.nonsmoke [sample(1:1652,replace=T),]

# Estimate eigen values

bmat[i,] = eigen(cor(btma))$values}

# apply sort

cr1.boot = cbind(apply(bmat,2,quantile,0.025/4),apply(bmat,2,quantile,1-0.025/4))

# apply standard deviation rule

se.boot = sqrt(apply(bmat,2,var))

cr2.boot = cbind(colMeans(wcgs.nonsmoke)-qnorm(1-0.025/4)\*se.boot,colMeans(wcgs.nonsmoke)+qnorm(1-0.025/4)\*se.boot)

# for smoke

B = 500

# Apply to each component of the vector

# Initialize

bmat = matrix(0,nrow=B,ncol=4)

set.seed(2838)

for (i in 1:B) {btma = wcgs.smoke [sample(1:1502,replace=T),]

# Estimate eigen values

bmat[i,] = eigen(cor(btma))$values }

# apply sort

cr1.boot = cbind(apply(bmat,2,quantile,0.025/4),apply(bmat,2,quantile,1-0.025/4))

# apply standard deviation rule

se.boot = sqrt(apply(bmat,2,var))

cr2.boot = cbind(colMeans(wcgs.smoke)-qnorm(1-0.025/4)\*se.boot,colMeans(wcgs.smoke)+qnorm(1-0.025/4)\*se.boot)