**Problem 5, part (a)**

**R Code:**#Problem 5 part a

mu0=5

sigma0=4

v0=2

tau0=1

data=list()

data[1] <- read.table("school1.txt")

data[2] <- read.table("school2.txt")

data[3] <- read.table("school3.txt")

n = sapply(data, length)

ybar=sapply(data, mean)

s=sapply(data, var)

taun=tau0+n

vn=v0+n

mun=(tau0\*mu0+n\*ybar)/taun

sigman=(v0\*sigma0+(n-1)\*s+tau0\*n\*(ybar-mu0)^2/taun)/(vn)

sigma=mu=matrix(0, 10000, 3, dimnames = list(NULL, c("school1", "school2", "school3")))

for (i in c(1, 2, 3)){

sigma[,i]=1/rgamma(10000, vn[i]/2, vn[i]\*sigman[i]/2)

mu[,i]=rnorm(10000, mun[i], (sigma[,i]/taun[i])^0.5)

}

#To calculate posterior means and 95% confidence interval for mu

colMeans(mu)

apply(mu, 2, function(x) {

quantile(x, c(0.025, 0.975))

})

#To calculate posterior means and 95% confidence interval for standard dev

colMeans(sqrt(sigma))

apply(sqrt(sigma), 2, function(x) {

quantile(x, c(0.025, 0.975))

})

**Output:**

**Posterior Means for mu (mean):**

> colMeans(mu)

school1 school2 school3

9.299521 6.963119 7.813711

**95% Confidence Intervals for mu (mean):**

> apply(mu, 2, function(x) {

+ quantile(x, c(0.025, 0.975))

+ })

school1 school2 school3

2.5% 7.789821 5.168932 6.164029

97.5% 10.822044 8.774271 9.450027

**Posterior Means for sigma (Std. Deviation)::**

> colMeans(sqrt(sigma))

school1 school2 school3

3.907917 4.394199 3.754119

**95% Confidence Intervals for sigma (Std. Deviation):**

> apply(sqrt(sigma), 2, function(x) {

+ quantile(x, c(0.025, 0.975))

+ })

school1 school2 school3

2.5% 2.998113 3.347581 2.812734

97.5% 5.168784 5.880659 5.134997

**Problem 5 part (b):**

**R code:**

#Problem 5 part b

mu\_ranks= t(apply(mu, 1, rank))

prob\_ranks= list()

for (p in permn(3)) {

index= apply(mu\_ranks, 1, function(row) {

all(row == p)

})

prob\_ranks[[paste(p, collapse = ",")]] = length(mu\_ranks[index, 1])/10000

}

prob\_ranks[["1,2,3"]]

prob\_ranks[["1,3,2"]]

prob\_ranks[["2,1,3"]]

prob\_ranks[["3,1,2"]]

prob\_ranks[["2,3,1"]]

prob\_ranks[["3,2,1"]]

**Output:**

> prob\_ranks[["1,2,3"]]

[1] 0.0067

> prob\_ranks[["1,3,2"]]

[1] 0.0034

> prob\_ranks[["3,1,2"]]

[1] 0.6726

> prob\_ranks[["2,1,3"]]

[1] 0.0802

> prob\_ranks[["2,3,1"]]

[1] 0.0147

> prob\_ranks[["3,2,1"]]

[1] 0.2224

**Problem 5 part c:**

**R Code:**

#Problem 5 part c

#Posterior predictive distribution

predict = matrix(0, 10000, 3, dimnames = list(NULL, c("school1","school2", "school3")))

for (i in c(1, 2, 3)) {

predict[, i]= rnorm(10000, mun[i], sqrt(sigma[,i]\*((taun[i]+1)/taun[i])))

}

#calculate ranks and probabilities

pred\_rank= t(apply(predict, 1, rank))

pred\_probrank = list()

for (p in permn(3)) {

index = apply(pred\_rank, 1, function(row) {all(row == p)

})

pred\_probrank[[paste(p, collapse = ",")]]= length(pred\_rank[index, 1])/10000

}

pred\_probrank[["1,2,3"]]

pred\_probrank[["1,3,2"]]

pred\_probrank[["3,1,2"]]

pred\_probrank[["2,1,3"]]

pred\_probrank[["2,3,1"]]

pred\_probrank[["3,2,1"]]

**Output:**

> pred\_probrank[["1,2,3"]]

[1] 0.1086

> pred\_probrank[["1,3,2"]]

[1] 0.1014

> pred\_probrank[["3,1,2"]]

[1] 0.2649

> pred\_probrank[["2,1,3"]]

[1] 0.1881

> pred\_probrank[["2,3,1"]]

[1] 0.1393

> pred\_probrank[["3,2,1"]]

[1] 0.1977

**Problem 5 part d:**

> prob\_ranks[["2,3,1"]]+prob\_ranks[["3,2,1"]]

[1] 0.2371

> pred\_probrank[["2,3,1"]]+pred\_probrank[["3,2,1"]]

[1] 0.3370

**Problem 6 part b:**

**R code:**

a1= rbeta(10000, 295, 308)

a2= rbeta(10000, 289, 333)

hist(a2-a1, main='Problem 6 part b Histogram', xlab='alpha2-alpha1')

mean(a2-a1>0)

**Output:**

[1] 0.1978

