EN.553.732; Homework 2

R Code for Problems 4, 5, and 6

**Problem 4 R Code and Plot**

> #Problem 4

>

> n<-5000

>

> #Chosen values and Prior Parameters:

> theta0 = 1

> sig0 = 0.5

> v0 = 1

> k0 = 1

> kn<-k0+n

> vn<-v0+n

>

> y<-rnorm(n, mean=0, sd = 1)

> yb<-mean(y)

> SS<-sum((y-yb)^2)

> theta\_n<-(k0\*theta0+n\*yb)/kn

> SSn<-(v0\*sig0+SS+(k0\*n)\*(y-theta0)^2/kn)/vn

> sig <- 1/rgamma(5000,vn/2,vn\*SSn/2)

> theta<-rnorm(5000, theta\_n, sqrt(sig/kn))

> t<-rt(5000, df=vn)\*sqrt(SSn/kn)+theta\_n

> theta\_density<-density(theta)

> t\_dist<-density(t)

> plot(theta\_density)

> lines(t\_dist, col="red")

**Problem 5, part a**

**R Code**

#Problem 5

#Part (a)

mu0=5

sigma0=4

v0=2

k0=1

schooldata=list()

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

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

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

n = sapply(schooldata, length)

ybar=sapply(schooldata, mean)

s=sapply(schooldata, var)

kn=k0+n

vn=v0+n

mun=(k0\*mu0+n\*ybar)/kn

sigman=(v0\*sigma0+(n-1)\*s+k0\*n\*(ybar-mu0)^2/kn)/(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]/kn[i])^0.5)

}

#Computing posterior means and 95% confidence interval for mu

colMeans(mu)

apply(mu, 2, function(x) {

quantile(x, c(0.025, 0.975))

})

#Computing posterior means and 95% confidence interval for standard deviation

colMeans(sqrt(sigma))

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

quantile(x, c(0.025, 0.975))

})

**Results:**

Posterior Means:

school1 school2 school3

9.290606 6.963136 7.814114

95% CI for mean:

school1 school2 school3

2.5% 7.75762 5.150658 6.172948

97.5% 10.84183 8.787480 9.427163

Posterior Means for standard deviation:

school1 school2 school3

3.905729 4.402176 3.741269

95% CI for standard deviation:

school1 school2 school3

2.5% 3.000531 3.349973 2.800034

97.5% 5.157399 5.889208 5.110928

**Problem 5, Part b**

**R Code**

#Part b

#combinat package installed for permn function use. Used to generate all 6 permutations of {1,2,3}.

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"]]

**Results:**

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

[1] 0.0066

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

[1] 0.0042

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

[1] 0.0846

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

[1] 0.6639

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

[1] 0.0154

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

[1] 0.2253

**Problem 5, Part c**

**R Code**

#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]\*((kn[i]+1)/kn[i])))

}

#Computing 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"]]

**Results:**

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

[1] 0.1092

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

[1] 0.1041

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

[1] 0.2699

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

[1] 0.1828

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

[1] 0.1402

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

[1] 0.1938

**Problem 5, part d**

**R Code and Results**

> #Part d

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

[1] 0.2407

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

[1] 0.334

**Problem 6b**

**R Code**

> #Problem 6 part b:

>

> N=5000

> a1= rbeta(N, 295, 308)

> a2= rbeta(N, 289, 333)

> x<-a2-a1

> hist(x, main='Problem 6(b) Histogram', xlab='a2-a1')

>

> ##As written in my HW, a2-a1>0 corresponds to a shift toward Bush post debate, so this is why

> #this is taken here.

> mean(x>0)

[1] 0.1934

> #Thus, the posterior probability that there was a shift in Bush's favor is 0.1934.

>

**Histogram Plot**

