lab1_code

Akshath Srinivas (akssr921) Yaning Wang (yanwa579)

```
#1 a.
s=22
n=70
f=70-22
#probability of success
prob=s/n
alpha0<-beta0<-8

#posterior mean and standard deviation
posterior_stats<-function(samples){
  posterior_mean<-c()
  posterior_sd<-c()
  for(i in 1:samples){
    #drawing samples
    poseterior_draws=rbeta(i,alpha0+s,beta0+f)
    #calculating accumulating samples mean and sd
    posterior_mean<-c(posterior_mean,mean(poseterior_draws))
    posterior_sd<-c(posterior_sd,sd(poseterior_draws))
  }
  return(list(posterior_mean,posterior_sd))
}
#means and sds of accumulating samples
posterior_means<-posterior_stats(10000)[[1]]
posterior_standardev<-posterior_stats(10000)[[2]]

# plot means and comparing with true value
plot(posterior_means,type = 'l',main ="the posterior mean vs the true mean",
     xlab = "the number of samples" )
true_mean_value=(alpha0+s)/((alpha0+s)+(beta0+f))
abline(h=true_mean_value,col='red')
legend("topright",legend="true mean value",col="red", lty=1, cex=0.8)

# plot sd and comparing with true value
plot(posterior_standardev,type = 'l',main ="the posterior sd vs the true sd" ,
     xlab = "the number of samples")
true_sd_value=((alpha0+s)*(beta0+f))/((((alpha0+s)+(beta0+f))^2)*((alpha0+s)+(beta0+f)+1))
abline(h=sqrt(true_sd_value),col='red')
legend("topright",legend="true sd value",col="red", lty=1, cex=0.8)

#1. b
poseterior_draws=rbeta(10000,alpha0+s,beta0+f)
pr=length(poseterior_draws[poseterior_draws>0.3])/10000
exact_pr=pbeta(0.3,alpha0+s,beta0+f,lower.tail = FALSE)
cat("The posterior probability is",pr,"\n")
```

```r
cat("The exact value from the Beta posterior is",exact_pr)
#both probabilities are almost same

#1 c
poseterior_draws=rbeta(10000,alpha0+s,beta0+f)
phi<-poseterior_draws/(1-poseterior_draws)
#histogram
hist(phi,breaks=50)

density_phi <- density(phi)
#with density curve
p<- ggplot()+
  geom_histogram(aes(x = phi,y=..density..),bins=50,color='blue',fill='white')+
  geom_line(aes(x = density_phi$x, y = density_phi$y), color = 'red')+
  ggtitle('The density and distribution of posterior of the odds')
print(p)

#2 a
y<-c(33, 24, 48, 32, 55, 74, 23, 17)
n<-length(y)
#drawing from chi square distribution
draw_chi <- rchisq(n = 10000, df = n)
mu<-3.6

#calculating tau by formula
tau<-sum((log(y)-mu)^2)/n
sigma_2<-n*tau/draw_chi
hist(sigma_2,breaks=180,xlim = c(0,1),main = "the posterior distribution")

#2 b
#ginnicoefficient
ginni_coef<-2*pnorm(sqrt(sigma_2/2),mean = 0,sd=1)-1
hist(ginni_coef,breaks=50,main="the posterior
distribution of the Gini coefficient G")

#2 c
#calculating credible interval
intervals<-quantile(ginni_coef,probs=c(0.025,0.975))
hist(ginni_coef,breaks=50)
abline(v=intervals[1])
abline(v=intervals[2])
legend("topright",legend="95% credible interval",pch="|", cex=0.4)
cat("The 95% equal tail credible interval for G is [",intervals[1],",",intervals[2],"]")

#2 d
#getting densities
posterior<-density(ginni_coef)
#sorting and ordering posterior x and y values
posterior_sorted <- sort(posterior$y,decreasing = TRUE)
x_sorted<-posterior$x[order(posterior$y,decreasing = TRUE)]

#function calculates area under curve and get the hpdi_intervals
hpdi_interval<-function(x,density){
```

```r
  for(i in 1:length(x)){
    if((i+1)>length(x)){
      break
    }
    #approx function and integrating to get area under curve by giving intervals
    fn <- approxfun(x, density)
    area_curve <- integrate(fn, x[i], x[i+1])
    #breaking out of loop when area is greater than or equal to 0.95
    if(area_curve$value>=0.95){
      break
    }
  }
  return(list(x[i],x[i+1]))
}
hpdi_lower<-hpdi_interval(x_sorted,posterior_sorted)[[1]]
hpdi_upper<-hpdi_interval(x_sorted,posterior_sorted)[[2]]

# Plot posterior density with 95% HPD interval
plot(posterior, main = "Posterior Density with 95% HPD Interval")
#ablines for hdpi and credible intervals
abline(v = hpdi_lower, lty = 2,col='red')
abline(v = hpdi_upper, lty = 2,col='red')
abline(v=intervals[1])
abline(v=intervals[2])
legend("topright",legend=c("95% credible interval","95% HPDI interval"),col=c('black','red'),lty = c(1,
#hdpi interval is moved little left where density is more, which makes sense

#3 a
y<-c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
mu<-2.4
k<-seq(0.01,10,by=0.01)

#posterior distribution function values
function_posterior<-function(k){
  y<-c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
  mu<-2.4
  p_r<-exp(k*sum(cos(y-mu))-0.5*k)/besselI(x=k,nu=0)^length(y)
  return(p_r)
}

#posterior density values
posterior_result<-function_posterior(k)

#normalizing the posterior distribution by integrating
integration<-integrate(function_posterior,lower=0.01,upper=10)
normalize_posterior<-posterior_result/integration$value

#area under curve integrates approximate to 1
fn <- approxfun(k, normalize_posterior)
area_curve <- integrate(fn, min(k), max(k))


#plotting density curve
```

```r
plot(k,normalize_posterior,type='l',main = "The normalized posterior distribution of k")

#3 b
#posterior mode of k means,k value for which
#posterior density value is maximum(i.e in normalize_posterior)
print(k[which.max(normalize_posterior)])
```