Comparing accuracy for different numbers of iterations

Yijia Zhao

To start, we first need to define true values and get data points from our model.

```
trueA <- 5
trueB <- 0
trueSd <- 10
sampleSize <- 31
x <- (-(sampleSize-1)/2):((sampleSize-1)/2)
y <-  trueA * x + trueB + rnorm(n=sampleSize,mean=0,sd=trueSd)
```

With these data points, we can get the functions denfined in functionsforMH.R to work.

To compare the accuracy of Metropolis-Hastings algorithm in finding a for different iteration times, we first define the compare_outcomes function which takes iteration times as input and run a loop of 10 to print mean and std of a for each chain with that iteration times. Then we give different input and can get different results to compare accuracy.

```
source("functionsforMH.R")
set.seed(1)
compare_outcomes <- function(iterations){
  for (i in 1:10){
    a<-runif(1,0,10)
    b<-rnorm(1,0,1)
    c<-runif(1,1,20)
    startvalue <- c(a,b,c)
    chain <- run_metropolis_MCMC(startvalue,iterations)
    print(c(mean(chain[,1]),sqrt(var(chain[,1]))))
  }
}
compare_outcomes(1000)
```

```
## [1] 5.0442561 0.5178008
## [1] 5.0877685 0.3785069
## [1] 5.1095854 0.2713652
## [1] 5.1766241 0.2511039
## [1] 5.1451491 0.2301854
## [1] 5.344824 0.647341
## [1] 5.2365372 0.2438219
## [1] 5.3620399 0.6353299
## [1] 5.0157006 0.5244847
## [1] 5.1142929 0.2515296
```

```
compare_outcomes(10000)
```

```
## [1] 5.1383976 0.2002829
## [1] 5.1469675 0.1998576
## [1] 5.1546494 0.2819282
## [1] 5.1337420 0.2573346
## [1] 5.1431913 0.2188859
## [1] 5.1777137 0.2967058
## [1] 5.1405473 0.2046402
## [1] 5.1505061 0.1965237
```

```
## [1] 5.1283718 0.2449065
## [1] 5.1467571 0.2346904
```

```
compare_outcomes(100000)
```

```
## [1] 5.1429408 0.2133363
## [1] 5.1337993 0.2348589
## [1] 5.1347288 0.2176064
## [1] 5.1392823 0.2124286
## [1] 5.1480973 0.2163931
## [1] 5.1397878 0.2399972
## [1] 5.1372610 0.2186885
## [1] 5.1397520 0.2127446
## [1] 5.1495177 0.2417548
## [1] 5.1382513 0.2188628
```

The first column is mean for a of each loop and the second column is std. The true value for a is 5.

From the results, we can generally see that std gets smaller and the mean value gets closer to 5 as iteration times becomes larger, which means the results become more and more accurate and stable as iteration times increases.