# ST340 Lab 5: Bandit problems

2020–21

## Bernoulli Bandits

```
library(mvtnorm)

set.seed(76)
```

(a) Set Bernoulli success parameters for each arm.

```
ps <- c(0.4,0.6)
```

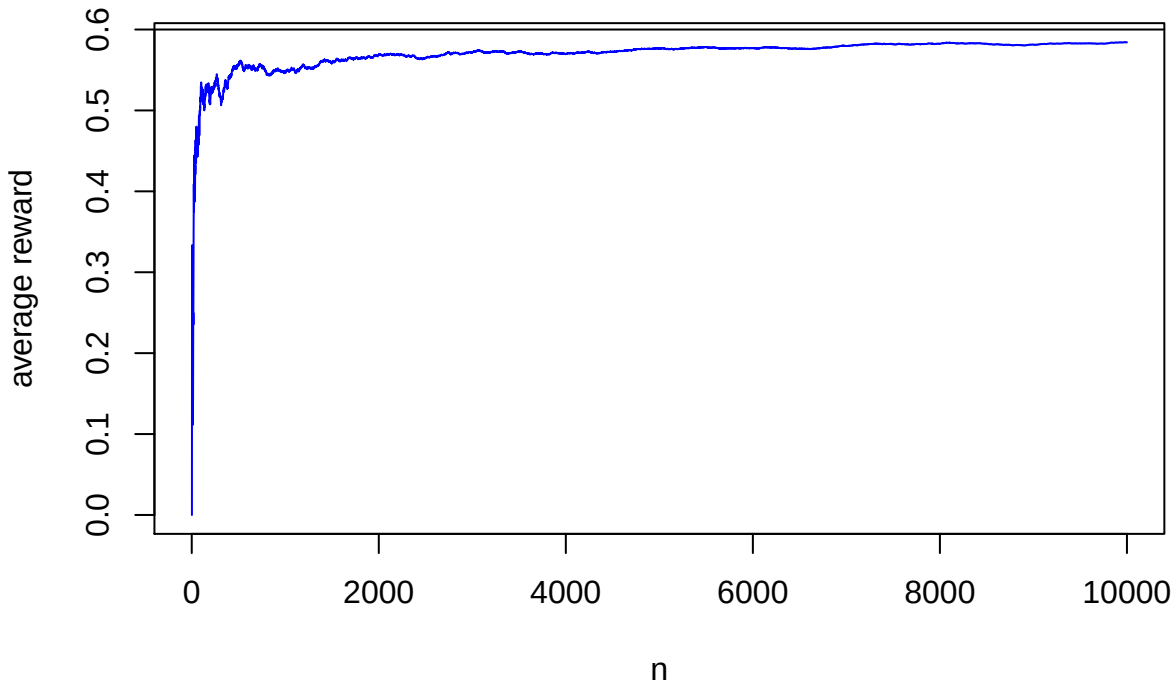
(b) This is a template for an Epsilon-greedy algorithm, runs for $n$ steps:

```
epsilon.greedy <- function(ps,epsilon,n) {
  as <- rep(0,n)
  rs <- rep(0,n)

  ## initial number of plays and number of successes is 0 for each arm
  ns <- rep(0,2); ss <- rep(0,2)

  ## at first, play each arm once
  for (i in 1:2) {
    a <- i
    r <- runif(1) < ps[a]
    ns[a] <- ns[a] + 1
    ss[a] <- ss[a] + r
    as[i] <- a
    rs[i] <- r
  }

  ## now follow the epsilon greedy strategy
  for (i in 3:n) {
    # with probability epsilon, pick an arm uniformly at random
    if (runif(1) < epsilon) {
      a <- sample(2,1)
    } else { # otherwise, choose the "best arm so far".
      a <- which.max(ss/ns)
    }
    ## simulate the reward
    r <- runif(1) < ps[a]

    # update the number of plays, successes
    ns[a] <- ns[a] + 1
    ss[a] <- ss[a] + r

    # record the arm played and the reward received
    as[i] <- a
```

```
    rs[i] <- r
  }
  return(list(as=as,rs=rs))
}
```

Run `epsilon.greedy` with the given `ps` and a choice of `epsilon` and see how well it does.

```
eg.out <- epsilon.greedy(ps=ps,epsilon=.1,n=1e4)

n=1e4

plot(x = 1:n, cumsum(eg.out$rs)/(1:n), ylab = 'average reward', xlab = 'n', col = 'Blue', cex = 0.5, ty
abline(a = 0.6, b = 0)
```




(c) Implement a `sample_arm` routine, for use in the Thompson sampling code below.

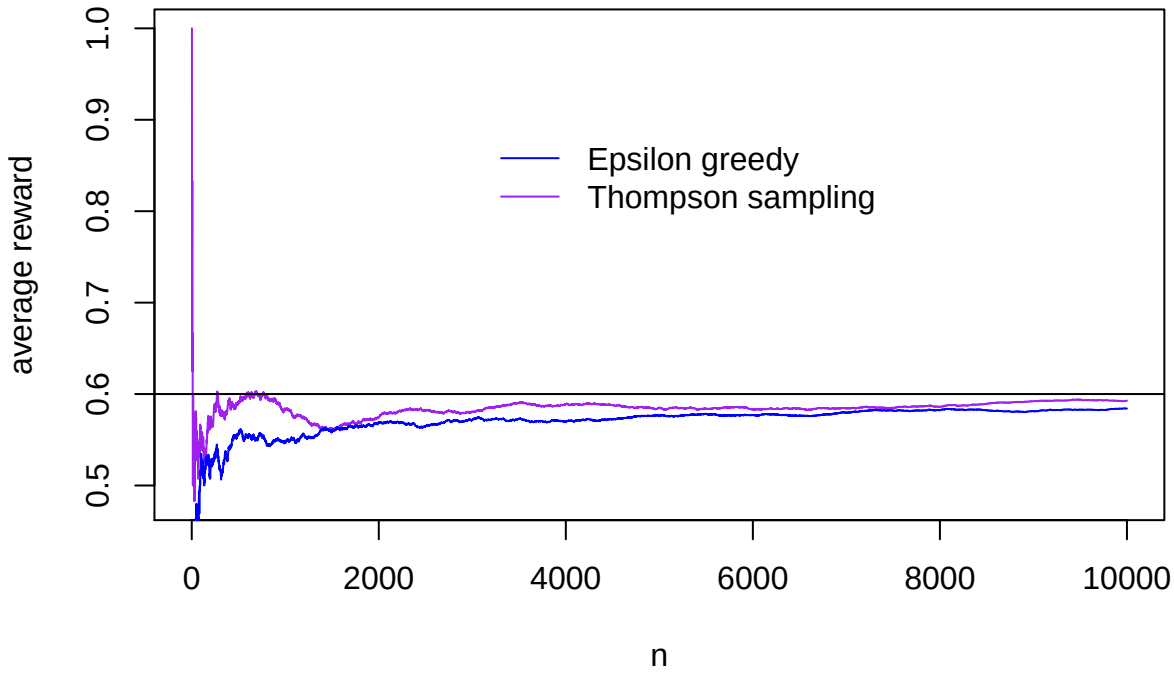
```
# ns is c(.,.) giving number of times each arm has been played
# ss is c(.,.) number of successes of each arm
sample_arm.bernoulli  <- function(ns,ss, alpha0 = 1, beta0 = 1) {

M1 = rbeta(n = 1, shape1 = alpha0 + ss[1], shape2 = beta0 + ns[1] - ss[1])
M2 = rbeta(n = 1, shape1 = alpha0 + ss[2], shape2 = beta0 + ns[2] - ss[2])



if(M1 >= M2){
  return(1)
 }else{
```

```
  return(2)
 }

}
```

```
thompson.bernoulli <- function(ps,n) {
  as <- rep(0,n)
  rs <- rep(0,n)

  ## number of times each arm has been played
  ## and number of corresponding successes
  ns <- rep(0,2); ss <- rep(0,2)

  for (i in 1:n) {
    a <- sample_arm.bernoulli(ns,ss)
    r <- runif(1) < ps[a]
    ns[a] <- ns[a] + 1
    ss[a] <- ss[a] + r
    as[i] <- a
    rs[i] <- r
  }
  return(list(as=as,rs=rs))
}
```

(d) Run the Thompson scheme and compare its performance to that of `epsilon.greedy`.

```
thompson.bernoulli.out <- thompson.bernoulli(ps=ps,n=1e4)

# Plo showing peformance differenc ein the two algorithms
plot(x = 1:n, cumsum(thompson.bernoulli.out$rs)/(1:n), ylab = 'average reward', xlab = 'n', col = 'purpl
lines(x = 1:n, cumsum(eg.out$rs)/(1:n), col = 'Blue')
abline(a = 0.6, b = 0)
legend(3000,0.9,  legend=c("Epsilon greedy", "Thompson sampling"),
       col=c("Blue", "purple"), lty=1, bty = 'n')
```


average reward
0.5
0.6
0.7
0.8
0.9
1.0
0
2000
4000
6000
8000
10000
n
Epsilon greedy
Thompson sampling