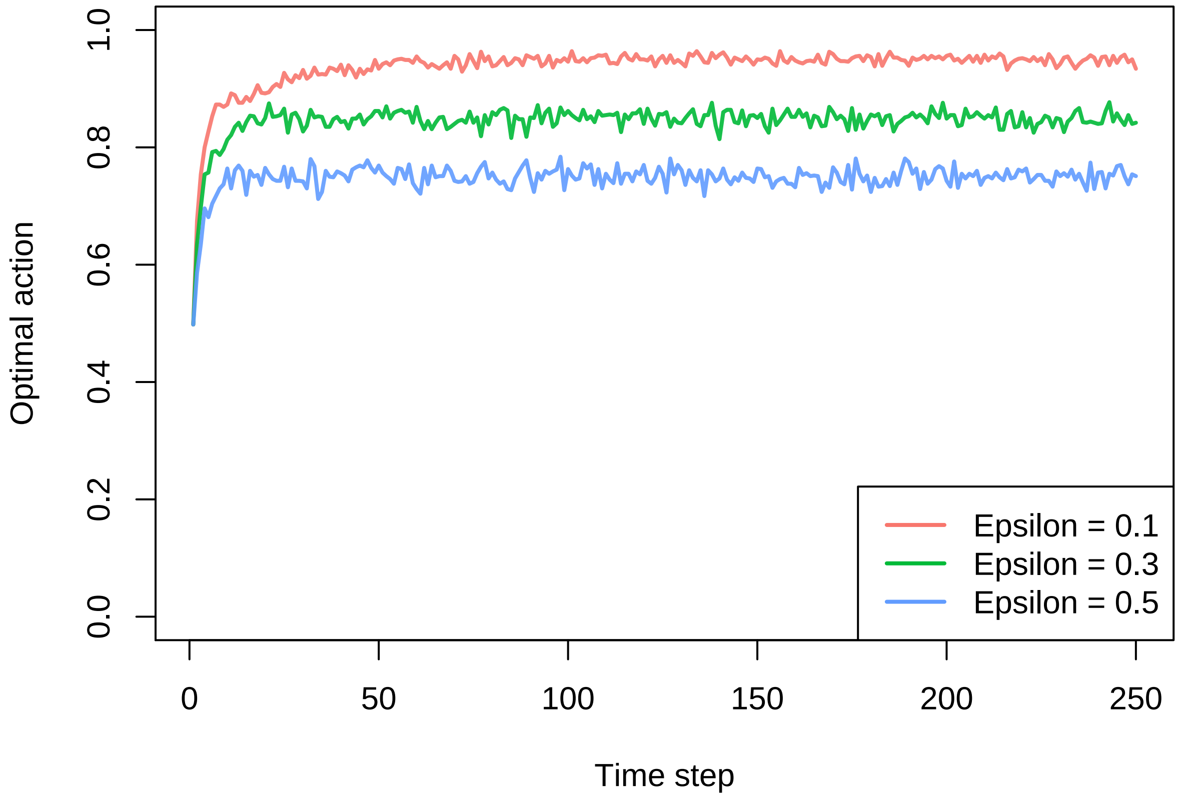
Homework 4 solution

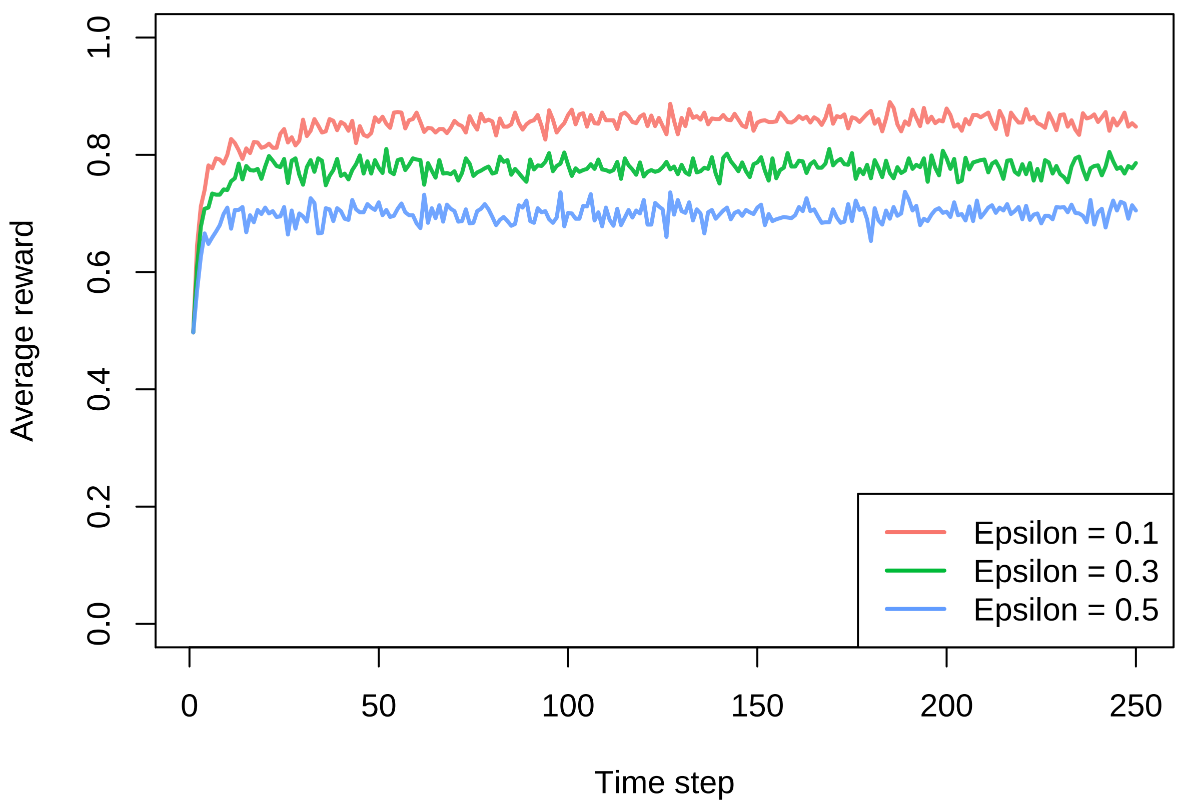
```r
# Question 1
#
library(contextual)
visits      = 250
simulations  = 1000
#
# a) 2 Arms
#
prob_per_arm = c(0.1, 0.9)
bandit  = BasicBernoulliBandit$new(prob_per_arm)
agents  = list(Agent$new(EpsilonGreedyPolicy$new(0.1), bandit, "Epsilon = 0.1"),
               Agent$new(EpsilonGreedyPolicy$new(0.3), bandit, "Epsilon = 0.3"),
               Agent$new(EpsilonGreedyPolicy$new(0.5), bandit, "Epsilon = 0.5"))
simulation = Simulator$new(agents, visits, simulations)
history    = simulation$run()
```

```
## Setting up parallel backend.

## Cores available: 4

## Workers assigned: 3

## Simulation horizon: 250

## Number of simulations: 3000

## Number of batches: 3

## Starting main loop.

## Finished main loop.

## Completed simulation in 0:01:38.061

## Computing statistics.
```

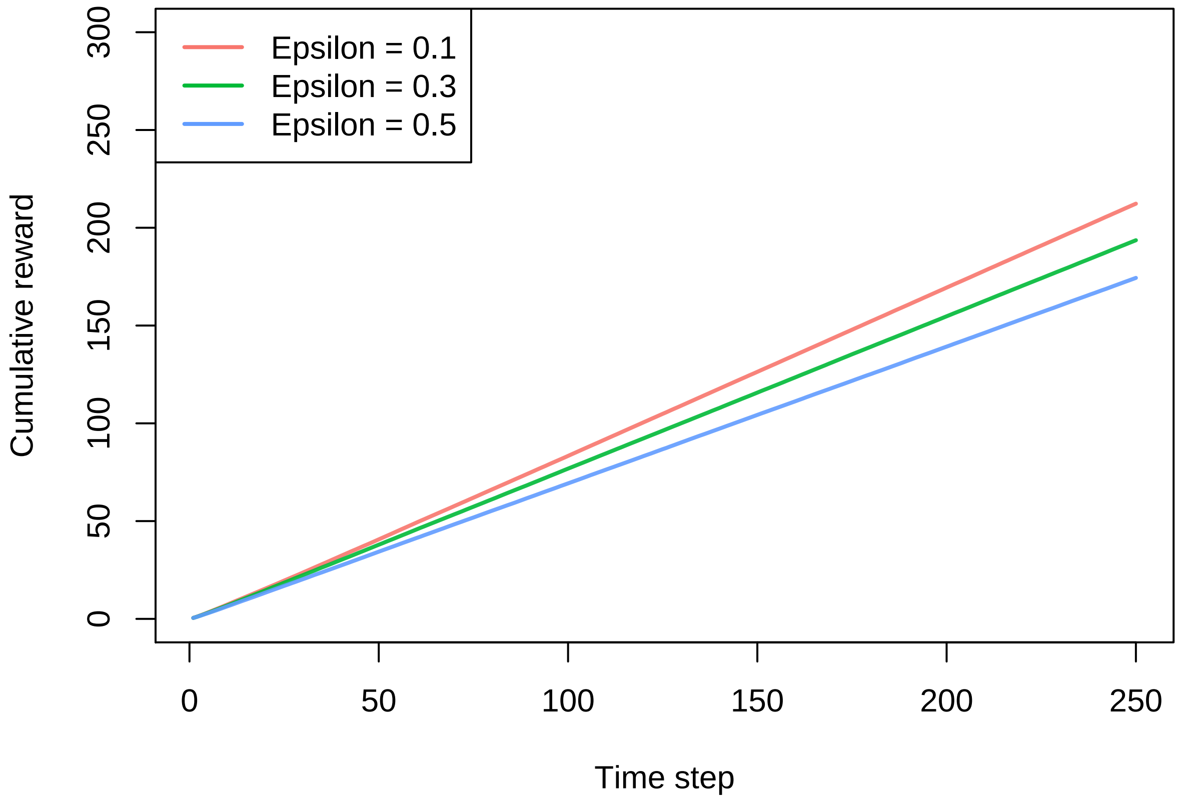
```r
#
plot(history, type = "optimal", legend_position = "bottomright", ylim = c(0,1))
```




```
#
# with 2 arms, the algorithm is able to find the best arm with probabilities from 0.7 to 0.9
plot(history, type = "average", regret = F, legend_position = "bottomright",ylim = c(0,1))
```



```r
plot(history, type = "cumulative", regret = FALSE, ylim = c(0,300))
```

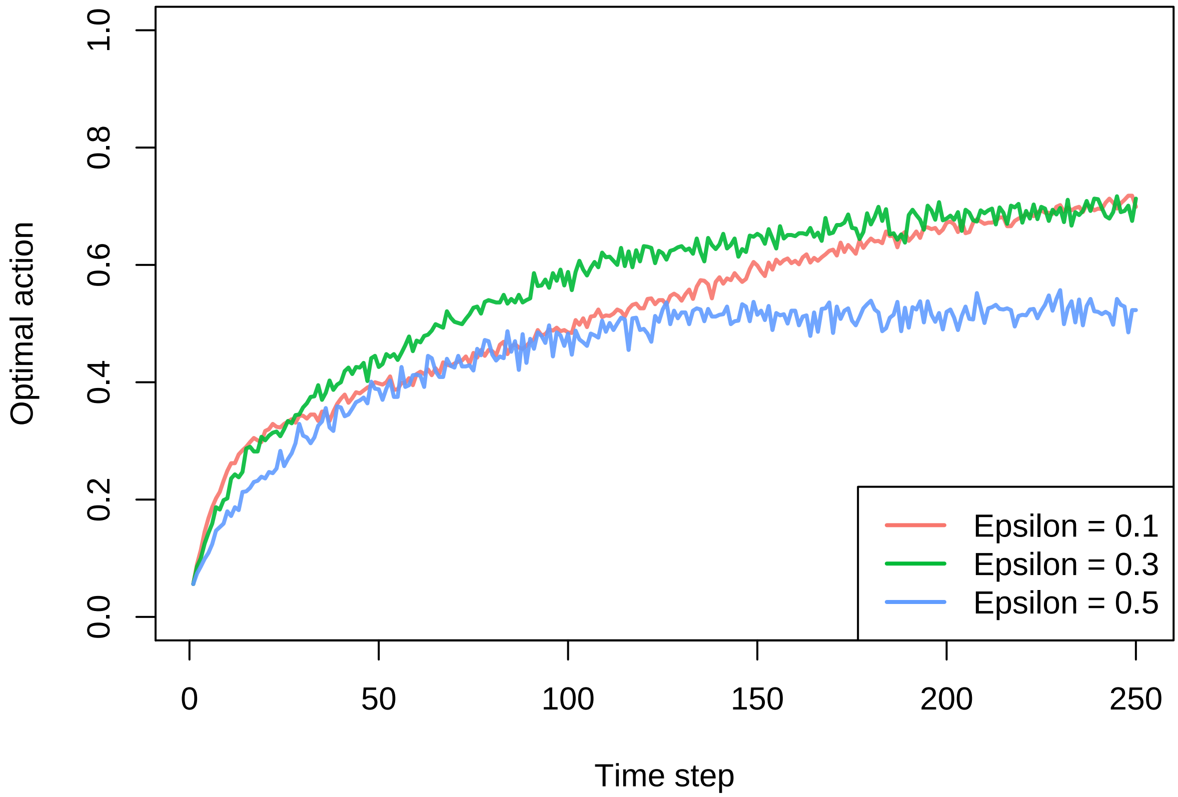



```r
#
# for 2 Arms, epsilon = 0.1 provides best average and cumulative rewards
#
# b) 20 Arms
#
prob_per_arm = rep(0.1,19)
prob_per_arm = c(prob_per_arm, 0.9)
bandit  = BasicBernoulliBandit$new(prob_per_arm)
agents  = list(Agent$new(EpsilonGreedyPolicy$new(0.1), bandit, "Epsilon = 0.1"),
               Agent$new(EpsilonGreedyPolicy$new(0.3), bandit, "Epsilon = 0.3"),
               Agent$new(EpsilonGreedyPolicy$new(0.5), bandit, "Epsilon = 0.5"))
simulation = Simulator$new(agents, visits, simulations)
history    = simulation$run()
```

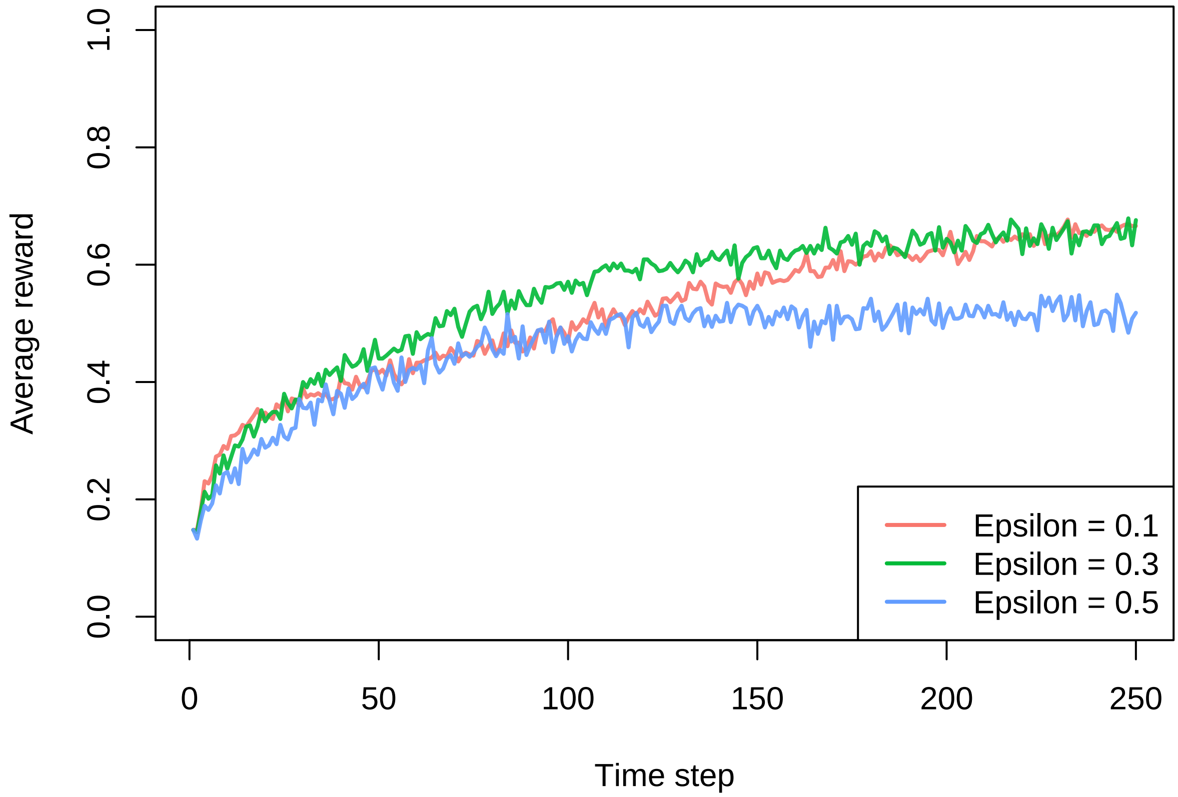
```
## Setting up parallel backend.

## Cores available: 4

## Workers assigned: 3

## Simulation horizon: 250

## Number of simulations: 3000

## Number of batches: 3

## Starting main loop.

## Finished main loop.
```

```
## Completed simulation in 0:01:39.564

## Computing statistics.
```

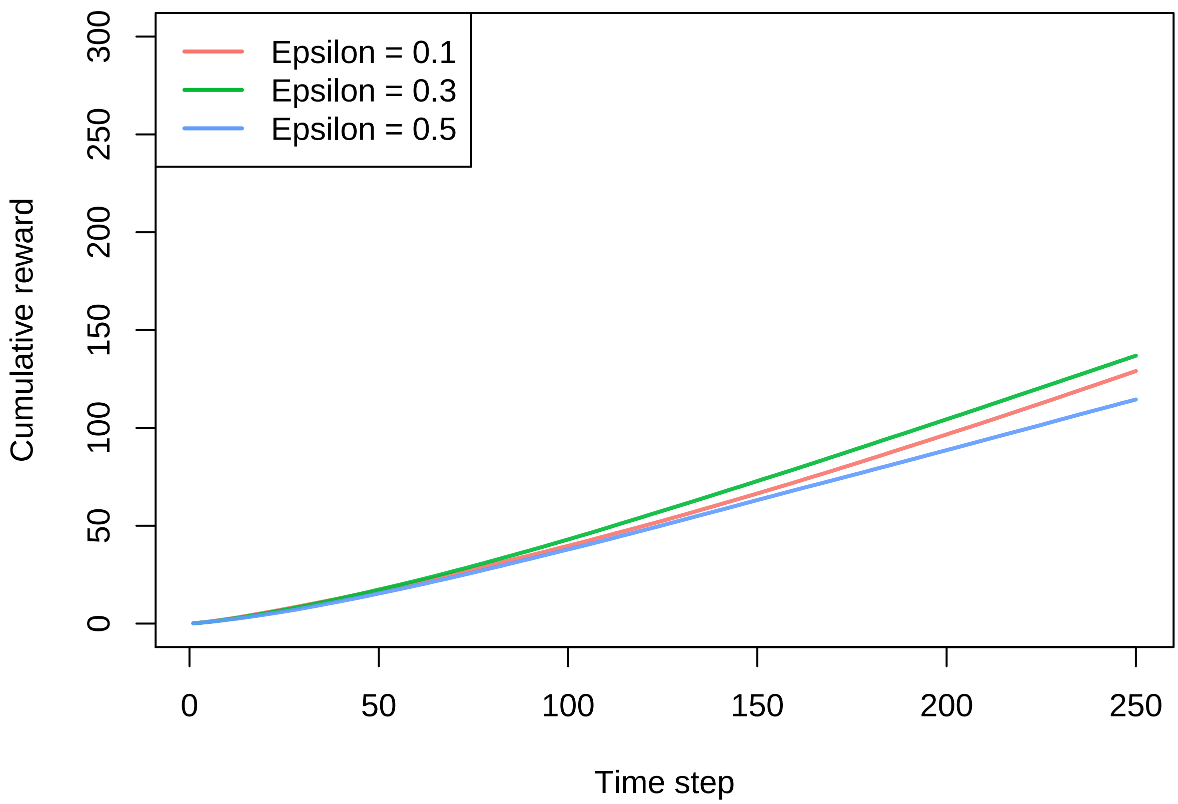
```
plot(history, type = "optimal", legend_position = "bottomright", ylim = c(0,1))
```




```
#
# comparing 20 with 2 arms, it seems that
# increasing the number of arms, the probability of identifying best arm decreases
# this plot shows that this probability does not exceed 0.70
plot(history, type = "average", regret = F, legend_position = "bottomright",ylim = c(0,1))
```



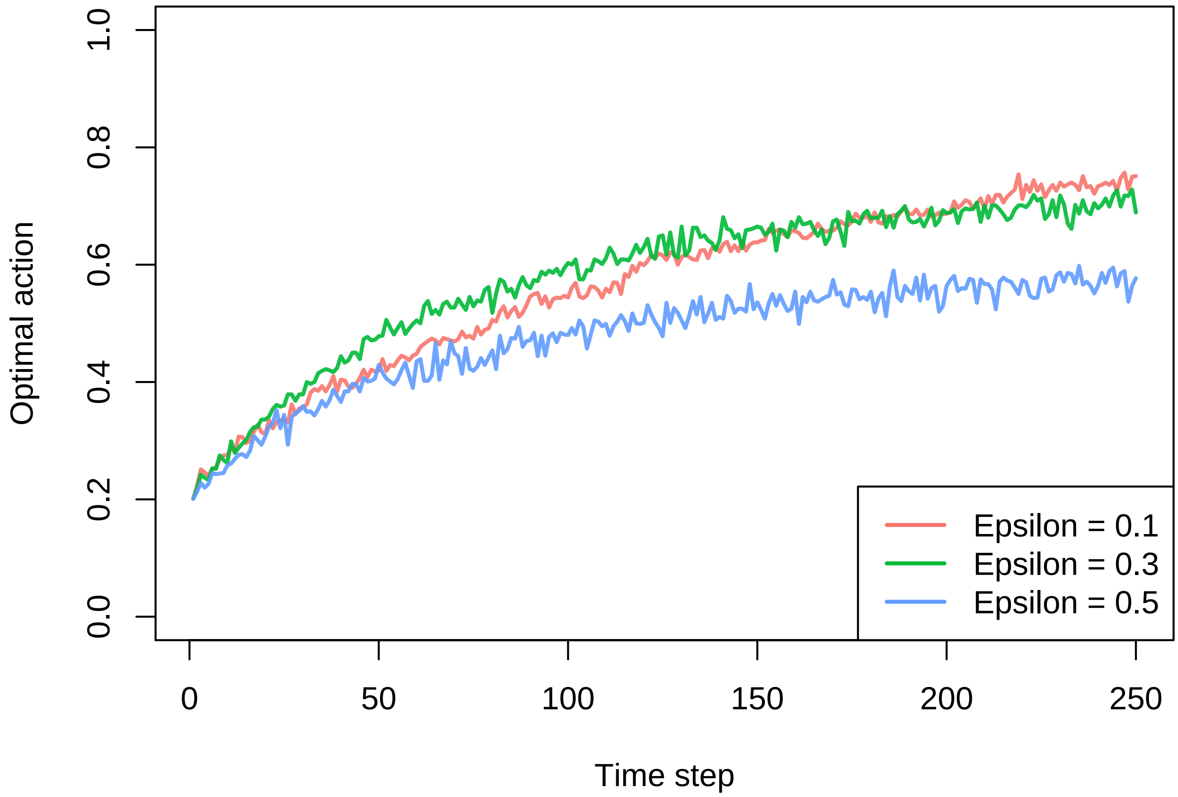

```r
plot(history, type = "cumulative", regret = FALSE, ylim = c(0,300))
```



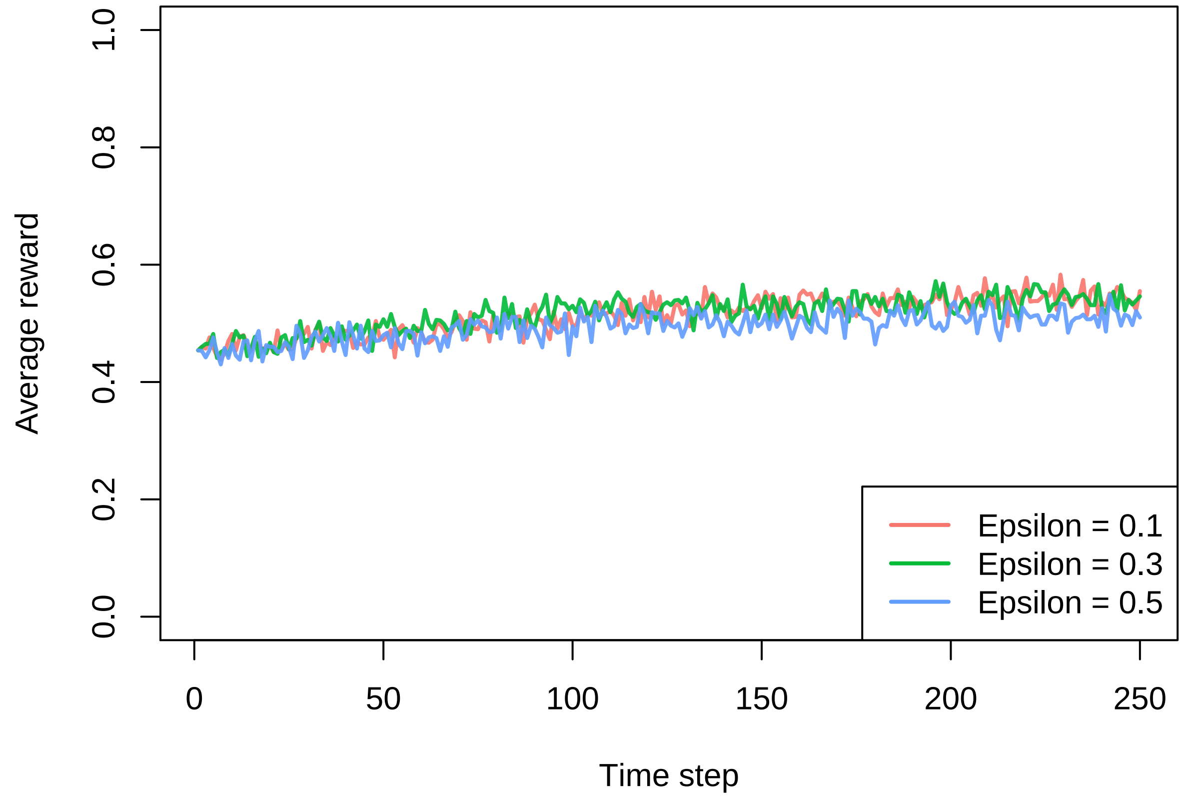
```r
#
# increasing the number of arms, the rewards decrease too
#
# Question 2
#
# a)
prob_per_arm = c(0.4,0.4,0.4,0.4,0.6)
bandit  = BasicBernoulliBandit$new(prob_per_arm)
agents  = list(Agent$new(EpsilonGreedyPolicy$new(0.1), bandit, "Epsilon = 0.1"),
               Agent$new(EpsilonGreedyPolicy$new(0.3), bandit, "Epsilon = 0.3"),
               Agent$new(EpsilonGreedyPolicy$new(0.5), bandit, "Epsilon = 0.5"))
simulation = Simulator$new(agents, visits, simulations)
history    = simulation$run()
```

```
## Setting up parallel backend.

## Cores available: 4

## Workers assigned: 3

## Simulation horizon: 250

## Number of simulations: 3000

## Number of batches: 3

## Starting main loop.

## Finished main loop.

## Completed simulation in 0:01:14.492

## Computing statistics.
```

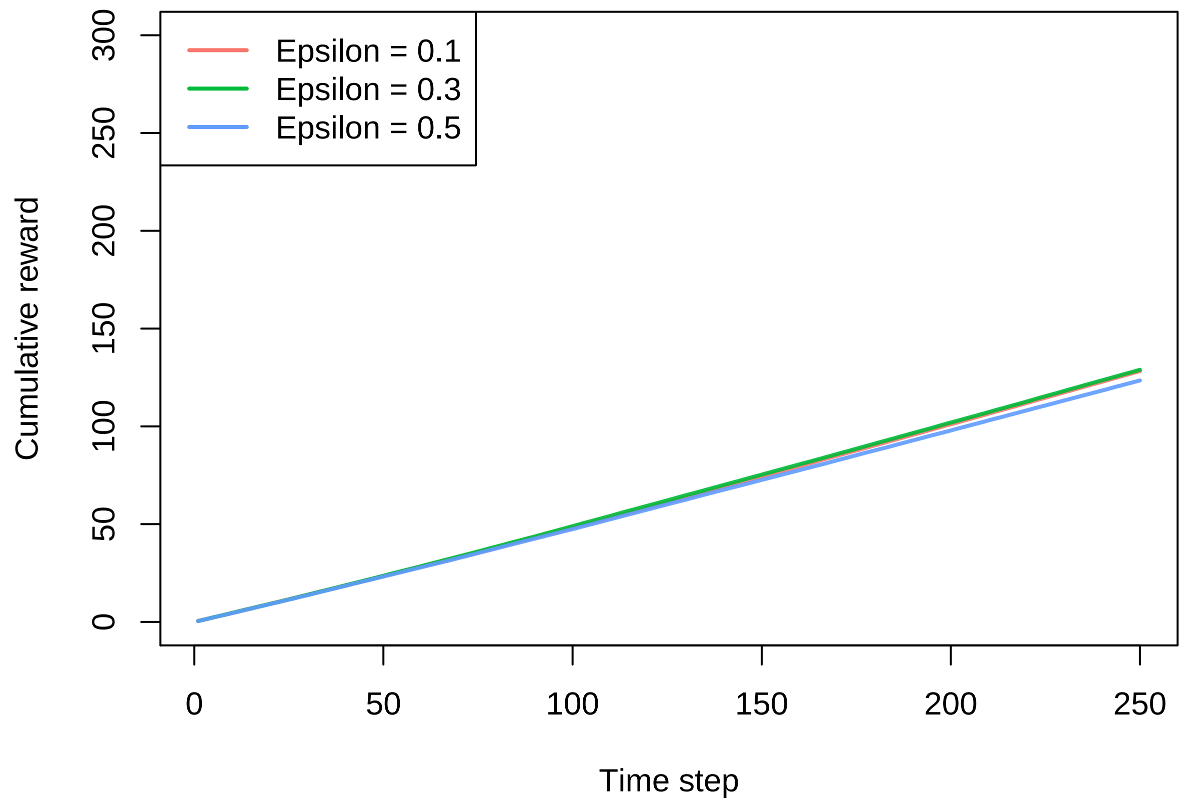
```r
plot(history, type = "optimal", legend_position = "bottomright", ylim = c(0,1))
```




```
#
# it is more difficult to identify the best arm if it is not very different than other arms
# the probability of identifying the best arm increases slowly with number of visits
plot(history, type = "average", regret = F, legend_position = "bottomright",ylim = c(0,1))
```



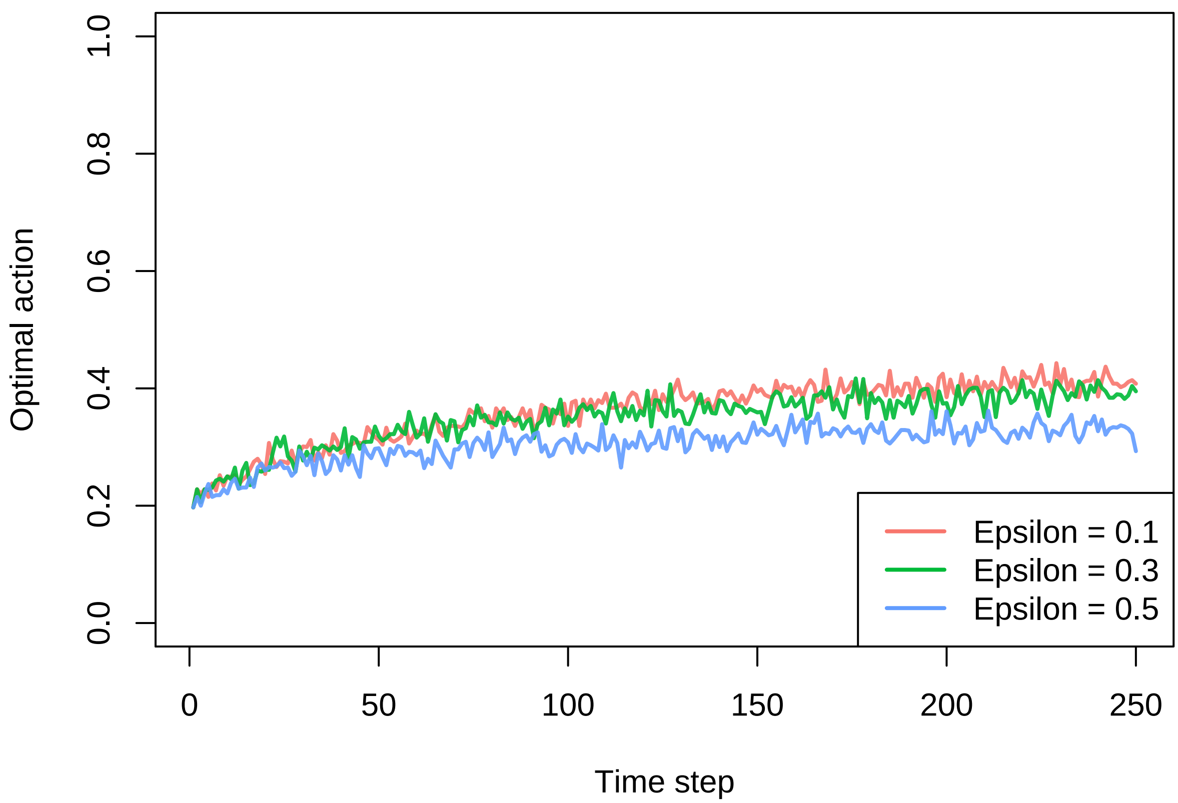

```
#
# Average reward is close to 0.50 even when the n. visits increases
plot(history, type = "cumulative", regret = FALSE, ylim = c(0,300))
```



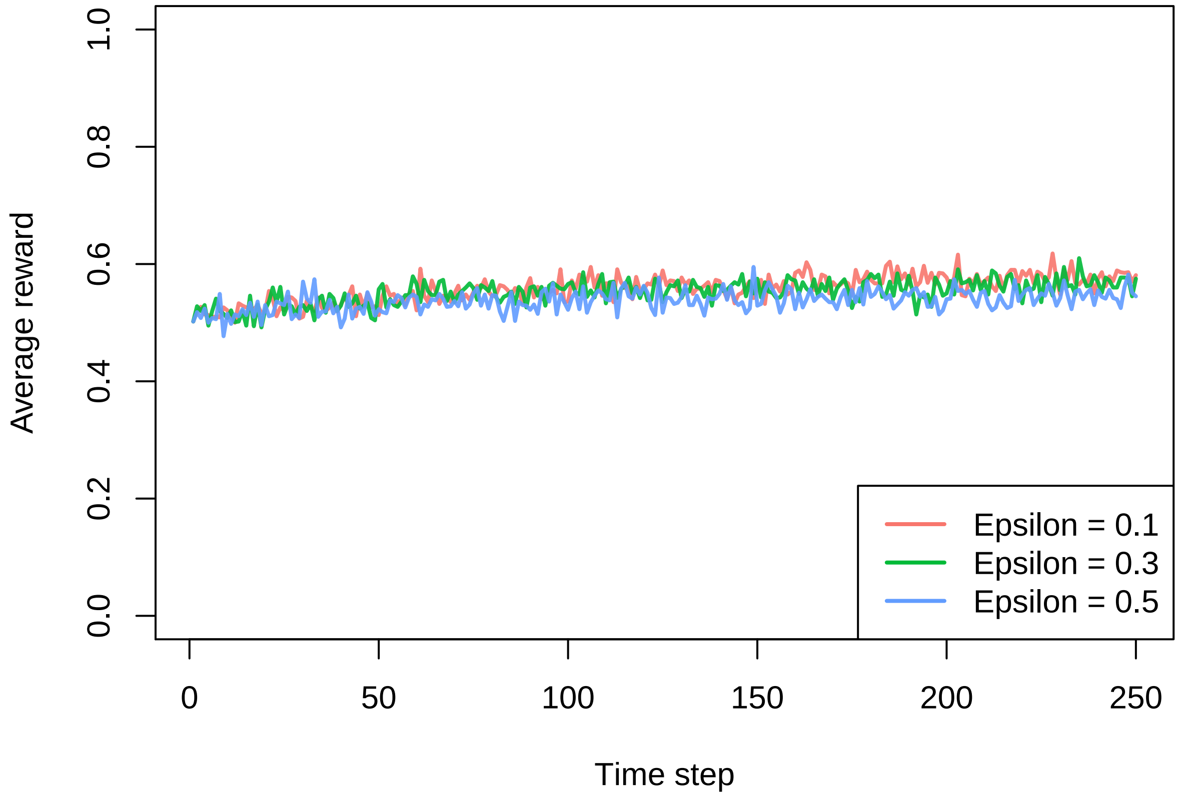
```r
# rewards not much different for all epsilon values
#
# b)
prob_per_arm = c(0.6,0.6,0.4,0.4,0.5)
bandit  = BasicBernoulliBandit$new(prob_per_arm)
agents  = list(Agent$new(EpsilonGreedyPolicy$new(0.1), bandit, "Epsilon = 0.1"),
               Agent$new(EpsilonGreedyPolicy$new(0.3), bandit, "Epsilon = 0.3"),
               Agent$new(EpsilonGreedyPolicy$new(0.5), bandit, "Epsilon = 0.5"))
simulation = Simulator$new(agents, visits, simulations)
history    = simulation$run()
```

```
## Setting up parallel backend.

## Cores available: 4

## Workers assigned: 3

## Simulation horizon: 250

## Number of simulations: 3000

## Number of batches: 3

## Starting main loop.

## Finished main loop.

## Completed simulation in 0:01:42.639

## Computing statistics.
```

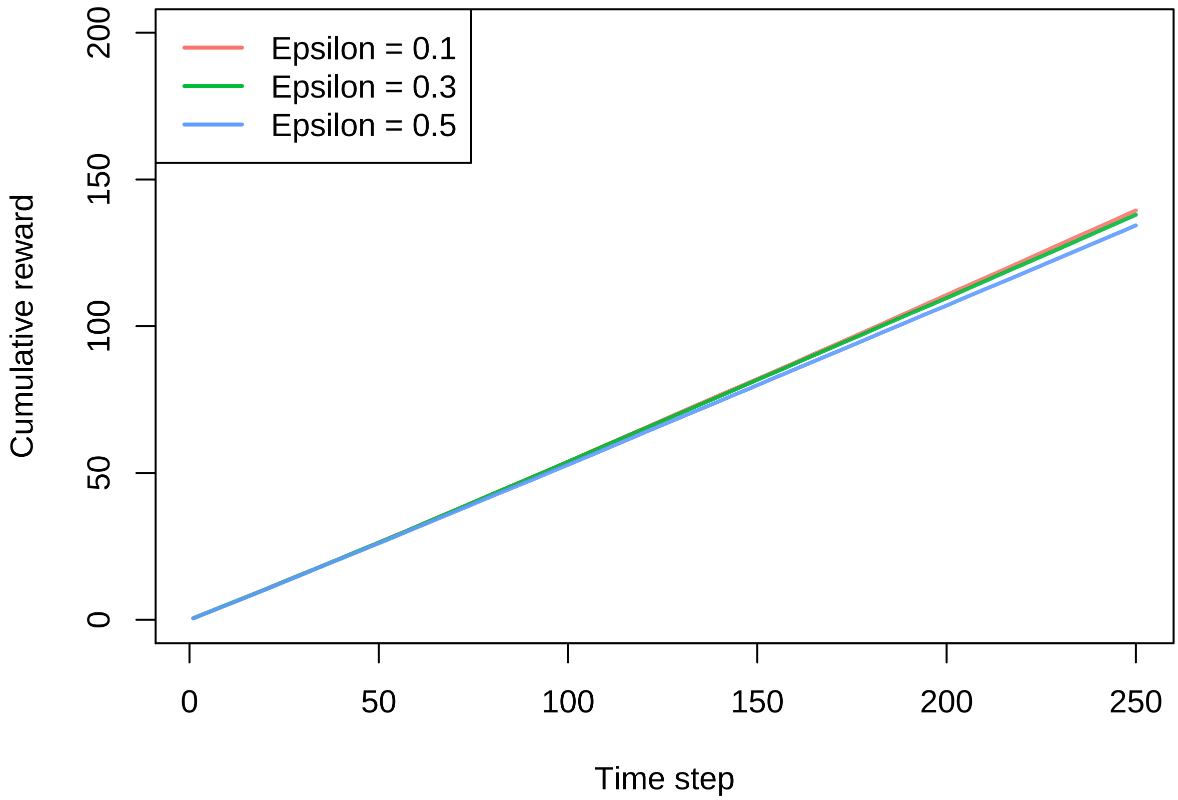
```r
plot(history, type = "optimal", legend_position = "bottomright", ylim = c(0,1))
```



```r
plot(history, type = "average", regret = F, legend_position = "bottomright",ylim = c(0,1))
```




```r
plot(history, type = "cumulative", regret = FALSE, ylim = c(0,200))
```

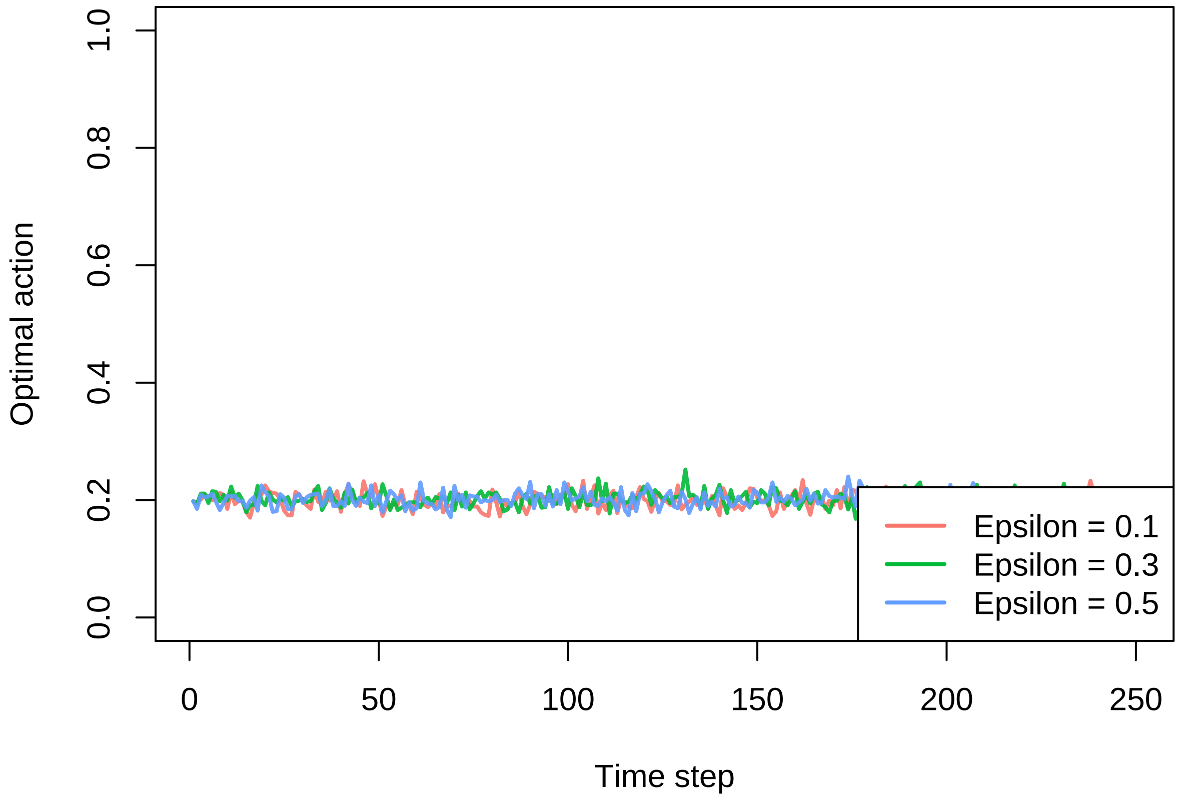



```
#
# It is more difficult to identify best option when there are two best arms
# probab. of selecting the best arm is lower than 0.4
# average reward is slightly above 0.5 for all epsilon values
# cumulative rewards are similar for all epsilon values
#
# c)
prob_per_arm = c(0.5,0.5,0.5,0.5,0.5)
bandit  = BasicBernoulliBandit$new(prob_per_arm)
agents  = list(Agent$new(EpsilonGreedyPolicy$new(0.1), bandit, "Epsilon = 0.1"),
               Agent$new(EpsilonGreedyPolicy$new(0.3), bandit, "Epsilon = 0.3"),
               Agent$new(EpsilonGreedyPolicy$new(0.5), bandit, "Epsilon = 0.5"))
simulation = Simulator$new(agents, visits, simulations)
history    = simulation$run()
```

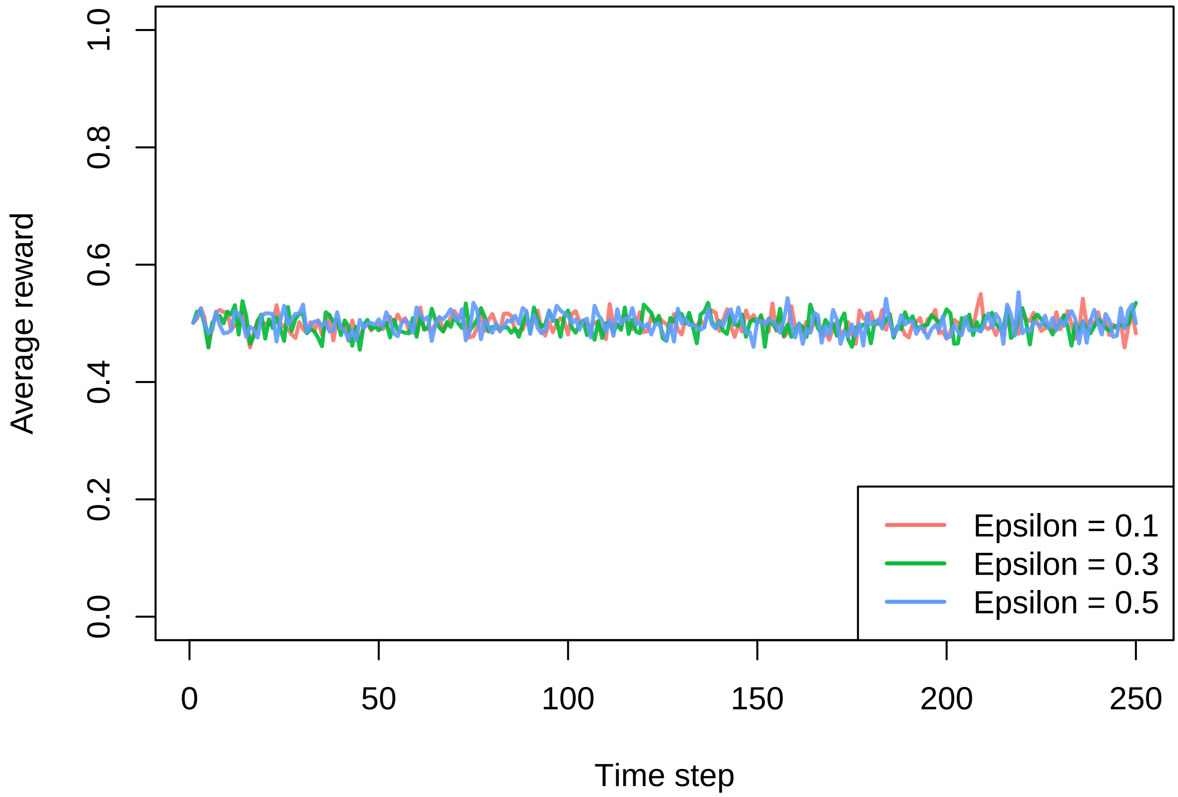
```
## Setting up parallel backend.

## Cores available: 4

## Workers assigned: 3

## Simulation horizon: 250

## Number of simulations: 3000

## Number of batches: 3

## Starting main loop.

## Finished main loop.

## Completed simulation in 0:00:50.629
```

```
## Computing statistics.
```

```r
plot(history, type = "optimal", legend_position = "bottomright", ylim = c(0,1))
```




```r
plot(history, type = "average", regret = F, legend_position = "bottomright",ylim = c(0,1))
```




```r
plot(history, type = "cumulative", regret = FALSE, ylim = c(0,200))
#
# algorithm chooses any arm as best, if there is no best arm
# probab. of selecting the best arm is lower than 0.2, but all arms are best
# average reward is 0.5 and cumulative rewards are the same for all epsilon values

# Question 3
#
k=5
headlines = c('A','B','C','D','E')
clicks = c(500,1000,825,490,880)
visits = c(900,1800,1500,1100,1325)
d0 = data.frame(rbind(clicks,visits))
names(d0) = headlines
d0
```

```
##          A    B    C    D    E
## clicks 500 1000  825  490  880
## visits 900 1800 1500 1100 1325
```

```r
#
# Step-by-step solution
#
# observed frequencies
#
noclicks = visits - clicks
noclicks
```

```
## [1] 400 800 675 610 445
```

```
observed = data.frame(rbind(clicks,noclicks))
names(observed) = headlines
observed
```

```
##            A    B   C   D   E
## clicks   500 1000 825 490 880
## noclicks 400  800 675 610 445
```

```
#
# expected frequencies
#
pooled = sum(clicks)/sum(visits)
pooled
```

```
## [1] 0.5577358
```

```
#
expected_clicks = pooled*(visits)
expected_noclicks = (1-pooled)* visits
expected = data.frame(rbind(expected_clicks,expected_noclicks))
names(expected) = headlines
expected
```

```
##                          A         B        C        D   E
## expected_clicks   501.9623 1003.9245 836.6038 613.5094 739
## expected_noclicks 398.0377  796.0755 663.3962 486.4906 586
```

```
#
# Chi-square test
#
chisquare = 0
for(i in 1:2)
{
  for(j in 1:k) {chisquare = chisquare + ((observed[i,j]-expected[i,j])^2)/expected[i,j]}
}
chisquare
```

```
## [1] 117.466
```

```
#
# p-value
#
pvalue = 1 - pchisq(chisquare,k-1)
pvalue
```

```
## [1] 0
```

```
#
# reject Ho, conclude not all designs provide same average click-rate
#
# Solution using prop.test()
#
prop.test(clicks,visits)
```

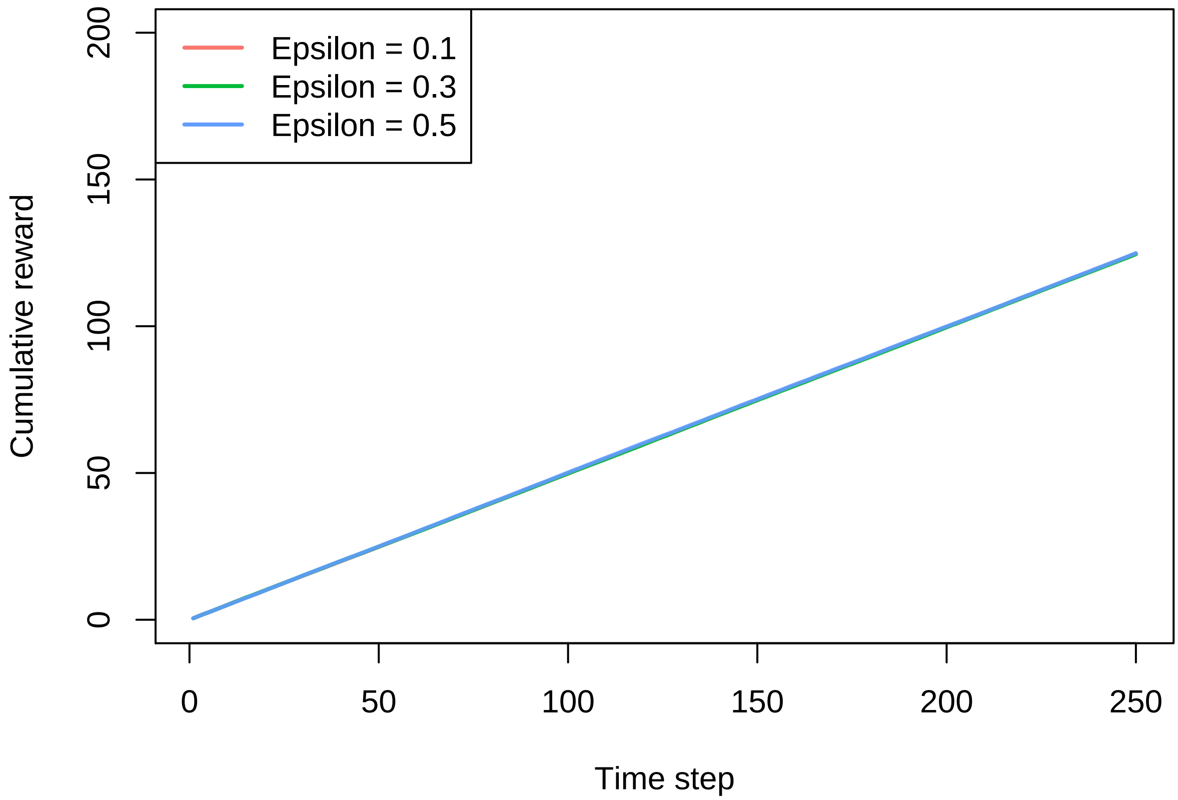
```
##
##  5-sample test for equality of proportions without continuity
##  correction
##
```

```
## data:  clicks out of visits
## X-squared = 117.47, df = 4, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3    prop 4    prop 5
## 0.5555556 0.5555556 0.5500000 0.4454545 0.6641509
```

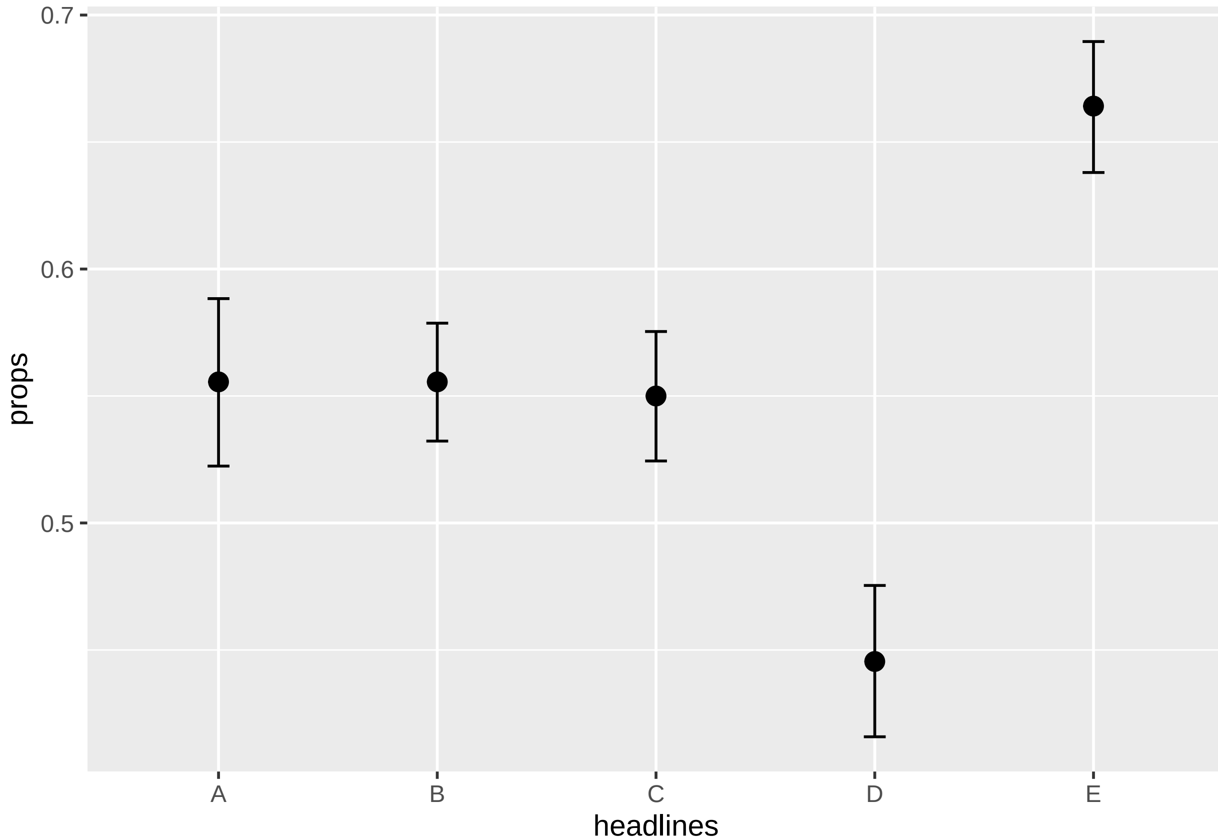
```r
#
# conclusion: not all headlines are equally good
#
# to find best headline use CIs
#
test = function(x,n) binom.test(x,n)
out = mapply(test,clicks,visits)
#
# collect all CIs
props = rep(0,k)
lls = rep(0,k)
uls = rep(0,k)
for (i in 1:k)
{
  props[i] = out[5,i]$estimate
  lls[i] = out[4,i]$conf.int[1]
  uls[i] = out[4,i]$conf.int[2]
}
#
d = data.frame(headlines,props,lls,uls)
d
```

```
##   headlines     props       lls       uls
## 1         A 0.5555556 0.5223989 0.5883474
## 2         B 0.5555556 0.5322475 0.5786823
## 3         C 0.5500000 0.5244132 0.5753906
## 4         D 0.4454545 0.4158083 0.4753933
## 5         E 0.6641509 0.6380010 0.6895705
```

```r
# plot CIs
#
library(ggplot2)
```




```
ggplot(data=d) +
  geom_errorbar(mapping = aes(x=headlines,ymin=lls,ymax = uls),width = 0.1) +
  geom_point(mapping = aes(x=headlines, y = props),size = 3)
```




```r
#
# Design E is the best
# with a click-rate between 0.638 and 0.6896
#
# Question 4
#
d0 = read.csv('cereal.csv')
str(d0)
```

```
## 'data.frame':    1250 obs. of  3 variables:
##  $ Group    : int  1 2 2 4 2 2 2 3 3 4 ...
##  $ Spend    : num  14.77 8.15 8 9.31 12.09 ...
##  $ Breakfast: int  4 4 2 1 4 4 4 4 3 3 ...
```

```r
n = nrow(d0)
k = 4
head(d0)
```

```
##   Group Spend Breakfast
## 1     1 14.77         4
## 2     2  8.15         4
## 3     2  8.00         2
## 4     4  9.31         1
## 5     2 12.09         4
## 6     2  7.13         4
```

```r
#
# Average Spending by Group
#
```

```r
means = tapply(d0$Spend,d0$Group,mean)
class(means)
```

```
## [1] "array"
```

```r
# convert to numeric
means = as.numeric(means)
means
```

```
## [1] 11.579257 10.145888 16.222780  6.147656
```

```r
#
# Test Ho: average amount spent is the same for all groups
#
model1 = aov(Spend~Group,d0)
summary(model1)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)    
## Group          1   1232    1232   68.54 3.16e-16 ***
## Residuals   1248  22438      18                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# or
model1 = lm(Spend~Group,d0)
anova(model1)
```

```
## Analysis of Variance Table
## 
## Response: Spend
##             Df  Sum Sq Mean Sq F value    Pr(>F)    
## Group        1  1232.3 1232.31  68.541 3.164e-16 ***
## Residuals 1248 22437.9   17.98                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#
# p-value is 3.16e-16, reject Ho
#
# get CIs
#
Spend1 = d0$Spend[d0$Group == 1]
Spend2 = d0$Spend[d0$Group == 2]
Spend3 = d0$Spend[d0$Group == 3]
Spend4 = d0$Spend[d0$Group == 4]
#
ci1 = t.test(Spend1)$conf[1:2]
ci2 = t.test(Spend2)$conf[1:2]
ci3 = t.test(Spend3)$conf[1:2]
ci4 = t.test(Spend4)$conf[1:2]

cis = rbind(ci1,ci2,ci3,ci4)
cis
```

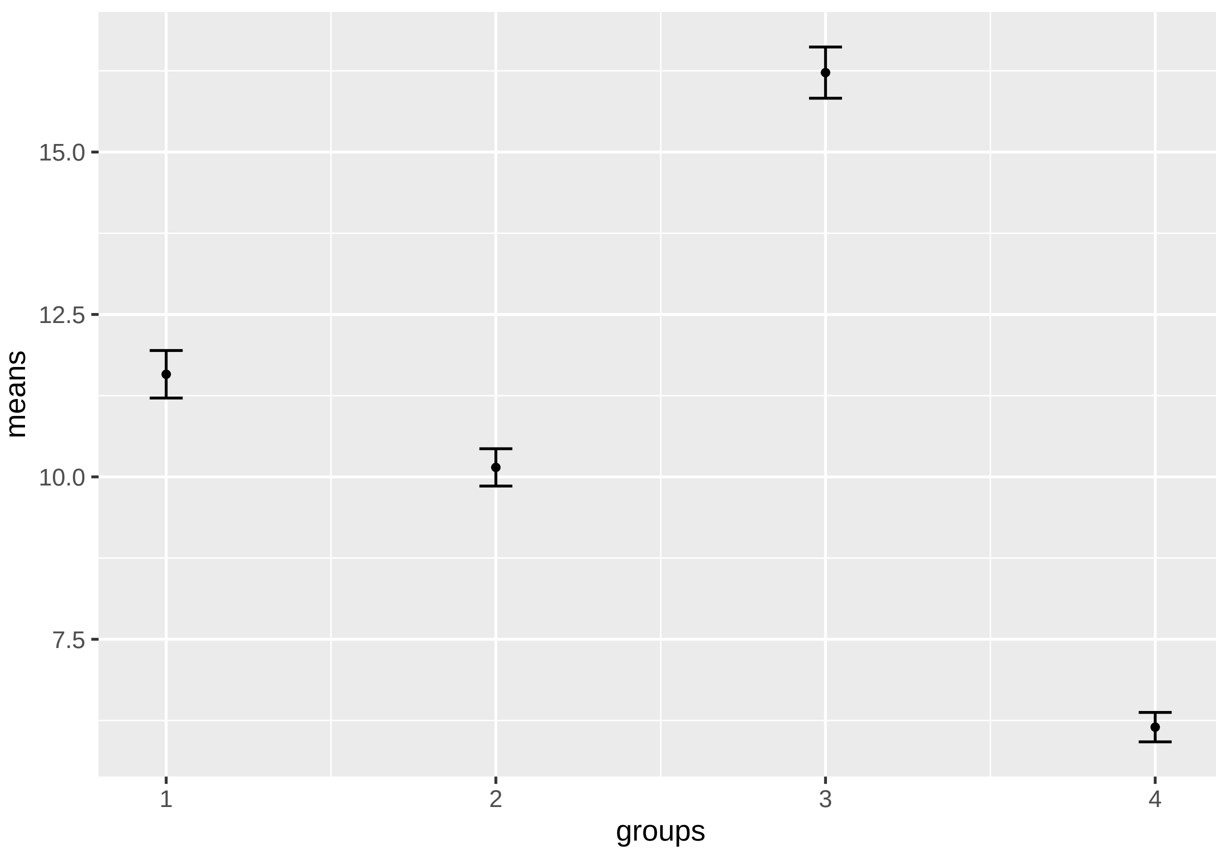
```
##          [,1]      [,2]
## ci1 11.213620 11.944893
## ci2  9.858414 10.433363
```

```
## ci3 15.828390 16.617170
## ci4  5.921201  6.374111
```

```
#
groups = 1:4
d1 = data.frame(groups,means,cis)
d1
```

```
##     groups     means        X1        X2
## ci1      1 11.579257 11.213620 11.944893
## ci2      2 10.145888  9.858414 10.433363
## ci3      3 16.222780 15.828390 16.617170
## ci4      4  6.147656  5.921201  6.374111
```

```
names(d1)=c('groups','means','lls','uls')
#
lls = d1$lls
uls = d1$uls
#
# plot CIs
#
ggplot(data=d1) +
  geom_errorbar(mapping = aes(x=groups,ymin=lls,ymax = uls),width = 0.1) +
  geom_point(mapping = aes(x=groups, y = means),size = 1)
```




```
#
# Group 3 spends the most on breakfast cereal
```