Coupon Collector's Problem With Unequal Probabilities
Application to Ecology

Marko KACHAIKIN, Vincent RUNGE

05/07/2022

Contents

Introduction 1

From equal to linear probability models 1

The log-linear expectation in equality case 2

Introduction

In this work we study the coupon collector's problem in the more generic case of unequal occurrence probabilities. Our first goal consists in better analyzing the probability distribution (mean and variance) for the random variable of the time to completion for some particular cases. We found simple formulas in asymptotic regime and compare these results throughout an extensive simulation study.

Our secondary goal is the search for estimators for unknown coupon number when we stop the collection at some chosen step. Performances of proposed estimators are studied both theoretically and by simulations.

From equal to linear probability models

We consider that the collection is made of N coupons with two possibles models.

1. the equal probability model:

$$p_i = \frac{1}{N}\,, \quad i = 1, \dots, N\,.$$

2. the linear probability model:

$$p_i = \frac{1}{N}\,, \quad i = 1, \dots, N\,.$$

with $\beta \in []$.

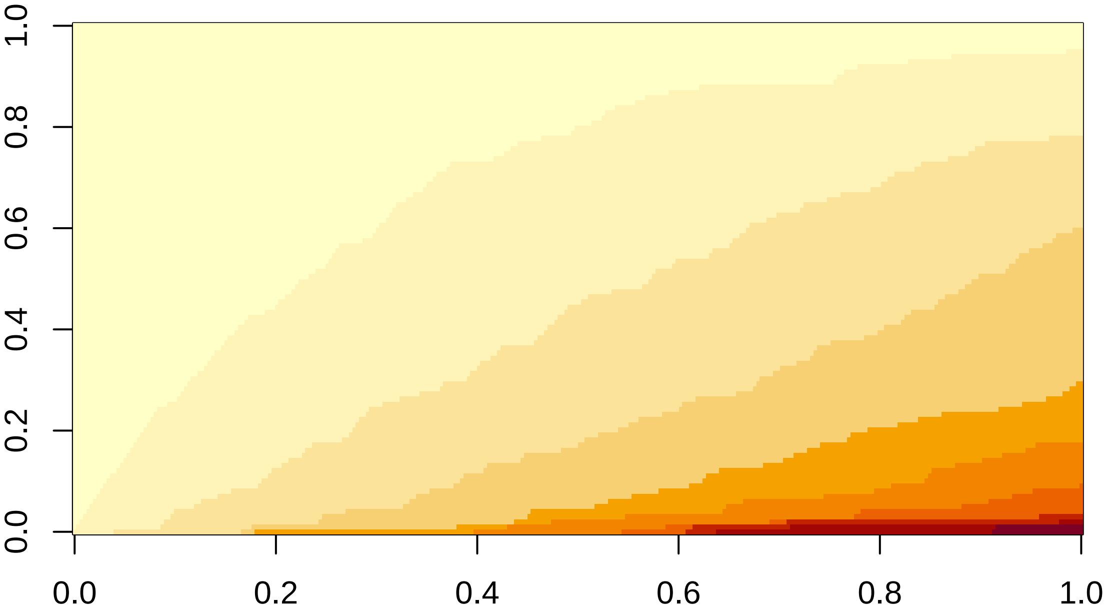
The log-linear expectation in equality case

In case 1 we can easily verify the well-known asymptotic result:

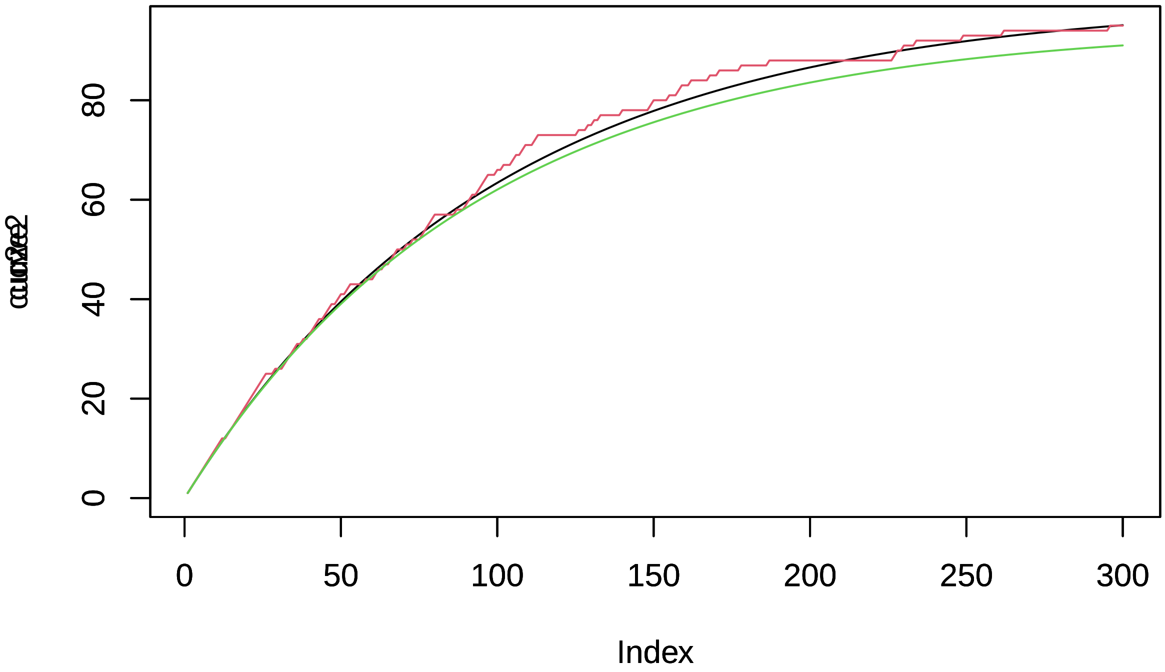
$$\mathbb{E}[T] \approx N(\ln N + \gamma)$$

For $N \in \{10, 20, 30, ..., 100, 200, 300, ..., 1000, 2000, ..., 10000\}$ we simulate 10^3 coupon problems. We show the detailed code using our package `coupon` available on GitHub (zfgzgz)

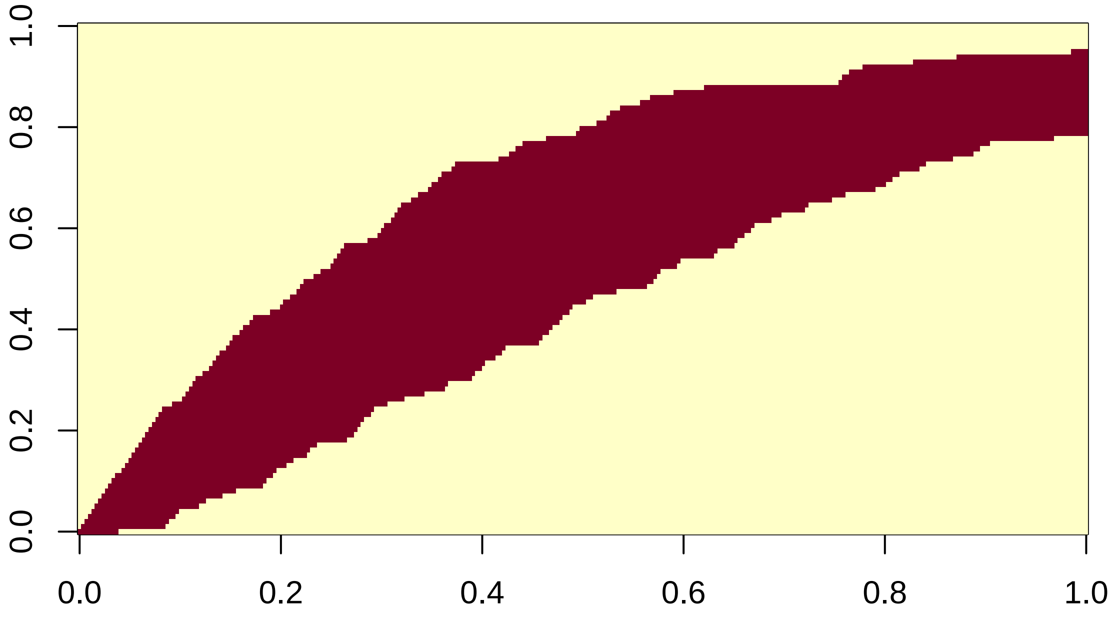
```
library(coupon)

u <- 5

nbCoupons <- 100
myN <- 300
image(res <- plot.tables(nbCoupons = nbCoupons, N = myN))
```



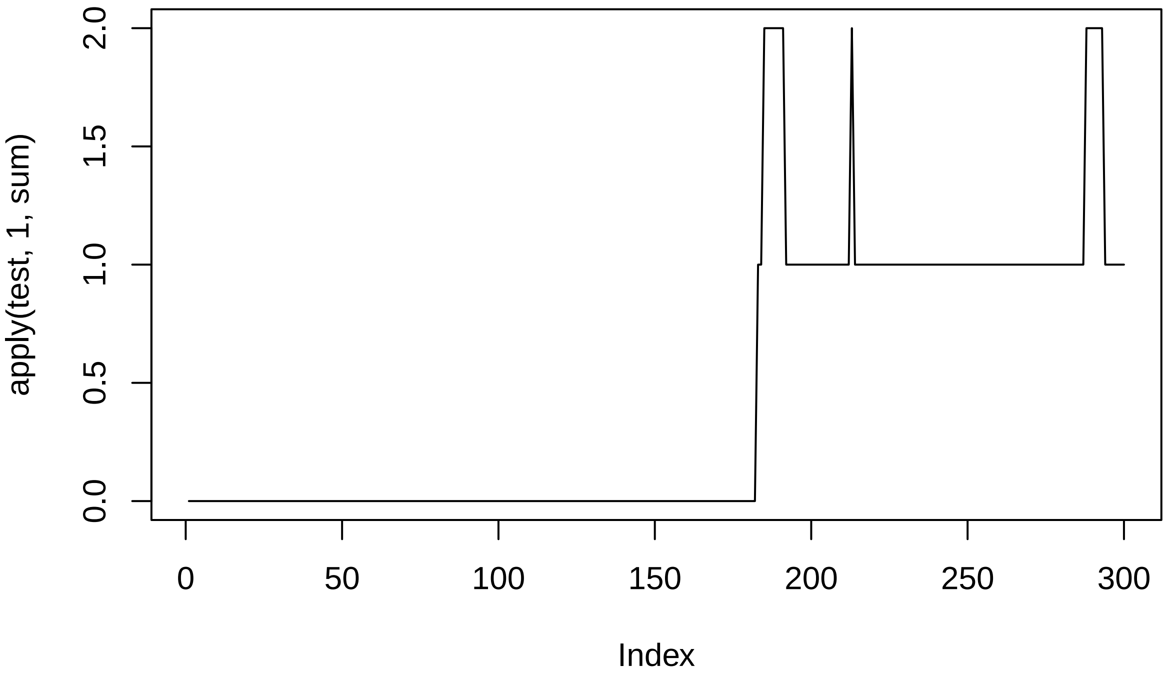
```
c2 <- apply(res, 1, function(x) sum(x > 0))

curve <- nbCoupons*(1- (1-1/nbCoupons)^(1:myN))
ylimit <- max(c(c2, curve))
plot(curve, type = 'l', ylim = c(0,ylimit))
par(new = TRUE)
plot(c2, type = 'l', col = 2, ylim = c(0,ylimit))
curve2 <- (nbCoupons-u)*(1- (1-1/(nbCoupons-u))^(1:myN))
par(new = TRUE)
plot(curve2, type = 'l', col = 3, ylim = c(0,ylimit))
```



```
#######



image(res ==1)
```



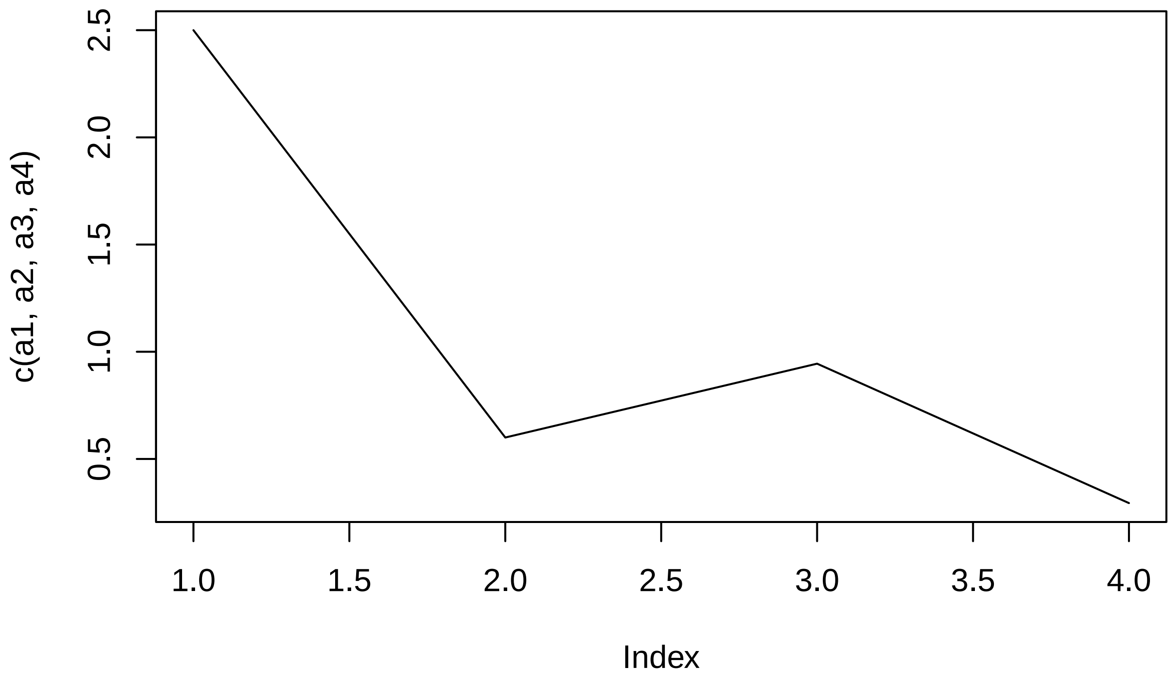
```
test <- res ==7
plot(apply(test, 1, sum), type = 'l')
```



```r
res[myN,]
```

```
##   [1] 9 9 8 7 6 6 6 6 6 6 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
```

```r
r <- res[myN,]
a1 <- sum(r == 3)/sum(r == 4)
a2 <- sum(r == 2)/sum(r == 3)
a3 <- sum(r == 1)/sum(r == 2)
a4 <- sum(r == 0)/sum(r == 1)
plot(c(a1,a2,a3,a4), type = 'l')
```


c(a1, a2, a3, a4)
0.5
1.0
1.5
2.0
2.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Index