Laboration 2

Johannes Pohjolainen

2021-02-16

```r
set.seed(950914) #Global seed för att kunna replikera labbens resultat
```

Sammanfattning

I denna laboration kommer vi att undersöka Gambler's ruin problemet (dvs chansen att en spelare eller ett casino går under givet att en spelare börjar med ett visst kapital) via ett konkret exempel där spelaren Kim börjar med ett kapital och antingen dubblar eller förlorar den kronan han satsar.

Uppgift 1

I denna uppgift kommer vi att implementera funktioner för att simulera Kims spelande.

1.1 Funktion för simulering av Gamblers ruin

För att undersöka om casinot eller Kim vinner, kommer vi först att implementera en funktion som representerar en spelomgång där Kim antingen vinner eller dubblar sin satsade krona enligt en viss sannolikhet:

```r
spela <- function(P, Kapital) {
  if(runif(1) < P) { #Vinner och tjänar pengar
    return(Kapital+1)
  } else {                    #Förlorar pengar
    return(Kapital-1)
  }
}

#spela(.5, 1)
```

Om vi det totala kapitalet till 5 kr kan ovanstående funktion anropas iterativt för att se om Kim eller casinot vinner enligt följande:

```r
kim_spelar <- function(P, Kapital, Rik = 5) {
  Kassa = Kapital

  repeat{
    Kassa = spela(P, Kassa)
    if(Kassa <= 0){
      return(0)
    } else if (Kassa >= Rik){
      return(1)
    }
  }
}
```

1.2 Sannolikhet att tjäna 5kr vid 1000 spelsekvenser med 1 krona

När vi nu har funktionerna vi behöver för att simulera flera spel kan vi beräkna sannolikheten att tjäna 5 kr givet att vi startar med 1 kr. Vi kommer att låta Kim försöka 1000 gånger och gör det med följade kod:

```
sim_kim <- function(P, Kapital, n = 1000) {
  nLyckade = 0

  for (i in 1:1000) {
    nLyckade = nLyckade + kim_spelar(P, Kapital)
  }

  nLyckade/n

}

ettFem = sim_kim(0.5, 1)
```

Via simuleringen får vi att den sannolikheten är 0.206.

1.3 Sannolikhet att tjäna 5kr med olika kapital

När vi beräknat sannolikheten att tjäna 5 kr med 1 kr är grunden lagd för att göra samma undersökning med andra startkapital:

```
tvaFem = sim_kim(.5, 2)
treFem = sim_kim(.5, 3)
fyraFem = sim_kim(.5, 4)

#Skapar tabell för att visa resultaten på ett prydligt sätt:
tabell1 <- data.frame(Start = c("En krona", "Två kronor",
"Tre kronor", "Fyra kronor"))
tabell1 <- cbind(tabell1, rbind(ettFem, tvaFem, treFem, fyraFem))
names(tabell1)[-1] <- paste0("Sannolikhet")
knitr::kable(tabell1, digits = 4, caption = "Tabell 1: Visar sannolikheten att vinna 5 kr för startkapi
```

Table 1: Tabell 1: Visar sannolikheten att vinna 5 kr för startkapital 1-4 kr enligt 1000 simulerade försök

Start	Sannolikhet
En krona	0.206
Två kronor	0.408
Tre kronor	0.624
Fyra kronor	0.799

Utifrån Tabell 1 fås stöd för hypoteser om att vinstsannolikheten kan skrivas som $f(kapital) \approx kapital * .2$.

Uppgift 2

I den föregående uppgiften beräknade vi sannolikheten att bräcka casinot med olika startkapital, genom att simulera 1000 försök och använda antalet vinster. I denna uppgift kommer vi att göra samma undersökning via användning av konvergens-egenskaper för övergångsmatriser på standardform. Detta då standardmatrisen över tidsteg konvergerar mot en matris som innehåller delmatriser vars element innehåller sannolikheten att gå till ett absorberande tillstånd (tillstånd 5 kr är av speciellt intresse för oss), givet ett specifikt start-tillstånd.

Nu när vi vet varför vi vill använda matriser på standardform är det lämpligt att definiera att det innebär att sortera en övergångmatris rader/colonner så att alla transienta tillstånd (1-4kr i vårt fall) kommer först:

$$\mathbf{P} = \begin{bmatrix} P_T & R \\ 0 & P_R \end{bmatrix}$$

DärP_t är övergångsmatrisen för transienta till transienta tillstånd, R innehåller övergångar från transienta till rekurrenta och P_R innehåller övergångar från rekurrenta till rekurrenta tillstånd.

Matrisen vi söker är:

$$n \to \infty \implies \mathbf{P^n} \to \begin{bmatrix} 0 & SR \\ 0 & I \end{bmatrix},$$

Där $(SR)_{i,j}$ innehåller sannolikheten att absorbtion sker i tillstånd "j" givet att vi startat i "i".

2.1 Paralell till 1.2, start med en krona

I 1.2 beräknade vi sannolikheten att vinna vid start med en krona. Om vi använder sannolikhet för vinst=förslust=0.5 och ställer upp standardmatrisen med kolonner 1-5, 0 (dvs spelarens möjliga kapital, vänster till höger) och sedan låter matrisen konvergera får vi att element (1,5) i SR ska representera samma sak.

2.2 Implementation av standardmatris för generell övergångs-sannolikhet.

För att vara mer specifika kring hur matrisen ur 2.1 ser ut implementerar vi nedan en funktion som kan skapa den:

```r
# row/col: 1-5,0
kims_matris <- function(p) {
  q = 1-p
  matrix(c( 0 , p , 0 , 0, 0, q ,
            q , 0,  p , 0, 0, 0 ,
            0 , q , 0 , p, 0, 0 ,
            0 , 0 , q , 0, p, 0 ,
            0 , 0 , 0 , 0, 1, 0 ,
            0 , 0 , 0 , 0, 0, 1 ),
            nrow = 6, byrow = TRUE)
}
```

2.3 uppgift 1.3 med denna metod: Sannolikhet att vinna med olika startkapital

Nu när vi har alla byggstenar på plas för att göra samma beräkningar som i Uppgift 1 passar vi på att göra det.

För att få fram SR behöver vi att vår övergångsmatris ska multipliceras med sig själv, varför vi implementerar följande funktion:

```
# Användning:
#   Multiplicerar matris med sig själv n ggr
# Input:
#   Matris A
#   Exponent n
# Output:
#   A^n
mpow <- function(A, n) {
  resultat <- diag(nrow(A))
  potens <- n
  while (potens > 0) {
    resultat <- A %*% resultat
    potens <- potens - 1
  }
  return(resultat)
}
```

Då vi inte har tid för en beräkning där n går hela vägen till oändligheten så nöjer vi oss med att konvergens sker med 4 decimalers noggrannhet. Därav implenterar vi följande funktioner:

```
#Jämför rader från matris A med B och undersöker om de är lika med d decimalers noggrannhet.
matrices_equal <- function(A, B, d = 4) {
  A_new <- trunc(A * 10^d)
  B_new <- trunc(B * 10^d)
  if (all(A_new == B_new)) {
    return(TRUE)
  } else {
  return(FALSE)
    }
}

#Multiplicerar mtx med sig själv tills konvergens skett enligt "decimals" antal decimaler.
getkonv <-function(mtx, decimals) {
  prevMtx = mtx;
  nextMtx = mtx %*% prevMtx;

  mults = 1; #Räknar antalet multiplikationer

  while (!(matrices_equal(prevMtx,nextMtx,decimals))) {
    mults = mults + 1;

    prevMtx = nextMtx;  #Tn

    nextMtx = mtx %*% nextMtx;    #Tn+1
  }
  prevMtx
}
```

Med hjälpfunktioner klara kan vi börja göra våra beräkningar:

```r
a = getkonv(kims_matris(.5), 4)
SR = a[1:4, 5:6]
use <- data.frame(Sannolikhet = SR[,1])

#Skapar tabell för att visa resultaten på ett prydligt sätt:
tabell2 <- data.frame(Start = c("En krona", "Två kronor",
"Tre kronor", "Fyra kronor"))

tabell2 <- cbind(tabell2, use)
names(tabell2)[-1] <- paste0("Sannolikhet")
knitr::kable(tabell2, digits = 4, caption = "Tabell 2: Visar sannolikheten att vinna 5 kr för startkapi
```

Table 2: Tabell 2: Visar sannolikheten att vinna 5 kr för startkapital 1-4 kr enligt SR

Start	Sannolikhet
En krona	0.2000
Två kronor	0.3999
Tre kronor	0.5999
Fyra kronor	0.7999

Vi ser att dessa resultat är väldigt lika resultaten ur 1.3.

Uppgift 3

I tidigare uppgifter har vi undersökt hur vinstsannolikheten ändras beroende på startkapital. I denna uppgift kommer vi istället att variera sannolikheten för att vinna spelet och på så sätt undersöka hur det ändrar sannolikheten att spelaren vinner över casinot.

Vi använder båda metoder vi stött på och får följande resultat:

```r
as <- function(mtx) {
#  mtx <- kims_matris(p)
  getkonv(mtx, 4)[1,5]
}

probs <- 3:7/10

sim <- sapply(probs, (function (x) sim_kim(x,1)))
konv <- sapply(probs, (function (x) as(kims_matris(x))))
```

```r
kims_df <- data.frame(Vinstsannolikhet = probs, sim_P = sim, SR_P = konv)

names(kims_df)[-1] <- c("$\\mathbb{P}(X_\\infty = 5)$ enligt simulering",
"$\\mathbb{P}(X_\\infty = 5)$ avläst i $\\mathbf{SR}$")

knitr::kable(kims_df, digits = 2, caption = "Tabell 3: Visar sannolikheten att vinna 5 kr för startkapi
```

Table 3: Tabell 3: Visar sannolikheten att vinna 5 kr för startkapital 1kr enligt simulering och konvergensresultat

Vinstsannolikhet	$\mathbb{P}(X_\infty = 5)$ enligt simulering	$\mathbb{P}(X_\infty = 5)$ avläst i **SR**
0.3	0.02	0.02
0.4	0.08	0.08
0.5	0.22	0.20
0.6	0.40	0.38
0.7	0.59	0.58

Uppgift 4

Hittils har vi undersökt hur ändringar i startkapital och vinstsannolikhet påverkar spelarens förutsättningar för att besegra casinot. I detta avsnitt kommer vi slutligen att undersöka hur spelarens stratergi påverkar resultatet genom att jämföra med en stratergi som går ut på att satsa så mycket så möjligt, men inte mer än vad som krävs för att vinna. Vi kallar ursprungsstratergin för S1 och den nya som presenteras här för S2

4.1 Implementering av övergångsmatris

Metoden ur Uppgift 2 kommer att användas i denna uppgift, så vi behöver börja med att implementera en övergångsmatris. Det görs enligt följande:

```
robins_matris <- function(p) {
  q = 1-p
  matrix(c( 0 , p , 0 , 0, 0, q ,
            0 , 0,  0 , p, 0, q ,
            q , 0 , 0 , 0, p, 0 ,
            0 , 0 , q , 0, p, 0 ,
            0 , 0 , 0 , 0, 1, 0 ,
            0 , 0 , 0 , 0, 0, 1 ),
            nrow = 6, byrow = TRUE)
}
```

4.2 Stratergijämförelse

När vi har en övergångsmatris för S2 kan vi göra alla beräkningar vi gjort för S1:

```
S2 <- sapply(probs, (function (x) as(robins_matris(x))))
```

```
kims_df <- data.frame(Vinstsannolikhet = 3:7 / 10, S1 = konv, S2)

names(kims_df)[-1] <- c("$\\mathbb{P}(X_\\infty = 5)$ via S1",
"$\\mathbb{P}(X_\\infty = 5)$ via S2")
```

```r
knitr::kable(kims_df, digits = 2, caption = "Tabell 4: Visar sannolikheten att vinna 5 kr för startkapi
```

Table 4: Tabell 4: Visar sannolikheten att vinna 5 kr för startkapital 1 kr enligt S1 och S2

Vinstsannolikhet	$\mathbb{P}(X_\infty = 5)$ via S1	$\mathbb{P}(X_\infty = 5)$ via S2
0.3	0.02	0.05
0.4	0.08	0.11
0.5	0.20	0.20
0.6	0.38	0.32
0.7	0.58	0.47

Via Tabell 4 kan vi dra slutsatsen att S2 lönar sig då $p \leq 0.5$.