# L bb 2 - M rkovkedjor

Seb stij n B bic & Esbjörn Runesson

2025-03-03

## Contents

```r
# Denna funktion simulerar en spelomgång med vinstsannolikhet p och ett visst kapital.
# Funktionen returnerar det nya kapitalet beroende på utfallet.
en_spelomgang <- function(p, kapital) {
  if (runif(1) < p) { # om vi vunnit
    return(kapital + 1)
  } else { # annars har vi förlorat
    return(kapital - 1)
  }
}




# Denna funktion räknar ut matrisen   upphöjt till n, enligt den iterativa
# definitionen  ^n =   %*%   %*% ... %*%   (n stycken  ), med  ^0 = I.
# Exempel: mpow( , 3) ==   %*%   %*%
mpow <- function( , n) {
  resultat <- di g(nrow( ))
  potens <- n
  while (potens > 0) {
    resultat <-   %*% resultat
    potens <- potens - 1
  }
  return(resultat)
}




# Låt   vara en matris innehållandes sannolikheter. Denna funktion testar om
# raderna i   är identiska upp till de d första decimalerna. Som ett exempel,
# talet 0.12309 är lika med 0.12301 upp till den fjärde decimalen, men avrundat
# till 4 decimaler är dessa tal ej lika.
# Funktionen returnerar TRUE om raderna är identiska; F LSE annars.
rows_equal <- function( , d = 4) {
   _new <- trunc(  * 10^d) # förstora talet och ta heltalsdelen
  for (k in 2:nrow( _new)) {
  # Kolla om alla element i rad 1 är lika med motsvarande element i rad k
    if (! ll( _new[1, ] ==  _new[k, ])) {
    # Om något element skiljer sig så är raderna ej lika
    return(F LSE)
    }
  }
# Hamnar vi här så var alla rader lika
  return(TRUE)
}



# Låt   och B vara matriser innehållandes sannolikheter. Denna funktion testar
# om elementen   är identiska, upp till de d första decimalerna, med motsvarande
# element i matrisen B.
```

```
# Funktionen returnerar TRUE om matriserna är identiska; F LSE annars.
matrices_equal <- function( , B, d = 4) {
   _new <- trunc(  * 10^d)
  B_new <- trunc(B * 10^d)
  if ( ll( _new == B_new)) {
    return(TRUE)
  } else {
    return(F LSE)
  }
}


# x är initialtillståndet, och kan vara 0, 1, ..., nrow(P) - 1 (dvs antalet rader i P minus 1)
# P är övergångsmatrisen. Sannolikheter för övergång till tillstånd 0 hittas i
# kolumn 1, övergång till tillstånd 1 i kolumn 2, och så vidare.
gen_sim <- function(x, P) {
u <- runif(1)
y <- 0
test <- P[x + 1, 1]
while (u > test) {
y <- y + 1
test <- test + P[x + 1, y + 1]
}
y
}
# Notera att vi numrerar våra tillstånd från 0 och uppåt, medan index för
# vektorer och matriser börjar på 1 i R. Därför ser vi x + 1 och y + 1 i koden
# ovan.
```

# 1 Uppgift 1

## 1.1 Funktion för en spelomgång

Funktionen `en_spelomgang` i avsnittet “ nvändbara funktioner” tar en vinstsannolikhet p och ett kapital kapital som input, och ger som output det nya kapitalet efter en spelomgång, enligt definitionen som gavs i introduktionen. Denna funktion utnyttjar kommandot `runif(1)` som genererar en slumpvariabel $U$ från en likformig fördelning på intervallet $[0, 1]$. Som du bör känna till vid det här laget så gäller det att $\mathbb{P}(U < p) = p$ för $p \in [0, 1]$.

> Din uppgift är nu att skriva en funktion `kim_spelar`, som simulerar Kims spelande enligt definitionen som gavs i introduktionen. Funktionen ska ta som input en vinstsannolikhet p och ett startkapital kapital. Som output ska funktionen returnera en 1:a (dvs return(1)) om Kims kapital når 6 kronor (Kim vinner), och en 0:a (dvs return(0)) om Kims kapital når 0 kronor (Kim förlorar). Funktionen kim_spelar ska använda sig av funktionen en_spelomgang.

```
kim_spelar <- function(p, kapital) {
  repe t {
    kapital <- en_spelomg ng(p, kapital)
    if (kapital == 6) {
      return(1)  # Kim vinner
    }
    if (kapital == 0) {
      return(0)  # Kim förlorar
    }
  }
}
```

## 1.2 Simulering v Kims spel nde

> Skriv en funktion `sim_kim`, som gör $n$ simuleringar av Kims spelande och räknar hur många gånger hen går med vinst. Funktionen ska använda sig av `kim_spelar` ovan, och ska som argument ta en vinstsannolikhet p, ett startkapital kapital, och antalet simuleringar n (default = 1000 ). Output ska vara antalet simuleringar som slutade med vinst.
>
> Gör nu 1000 simuleringar med hjälp av funktionen du just skrev. Hur många gånger lyckades Kim nå sitt mål, respektive blev pank? ntag att Kim startar med 1 krona i kapital vid varje simulering, och att vinstsannolikheten vid varje spelomgång är 0.5.

Så vi har $p = 0\ 5$, $kapital = 1$ och $n = 1000$ enligt uppgiften. Vi använder oss av `kim_spelar` funktionen för att simulera Kims spelande via `replicate` funktionen och genom att anropa `kim_spelar` $n$ gånger.

```
set.seed(040911)
sim_kim <- function(p, kapital, n) {
    return(sum(replic te(n, kim_spel r(p, kapital))))
}

sim_kim(0.5, 1, 1000)
```

```
## [1] 174
```

## 1.3 Simulering v Kims spel nde med större st rtk pit l

Om Kim har 2 kronor att starta med så är det rimligt att tro att chansen att lyckas nå fram till 6 kronor är större.

> Undersök genom simulering hur det går för en spelare som har 2, 3, 4 eller 5 kronor i startkapital. Gör 1000 experiment för vart och ett av dessa fyra fall också, och anteckna hur ofta spelet slutar

lyckligt.

Helt enkelt använder vi `sim_kim` funktionen igen men med olika startkapital.

```
sim_kim(0.5, 1, 1000)
```

```
## [1] 157
```

```
sim_kim(0.5, 2, 1000)
```

```
## [1] 351
```

```
sim_kim(0.5, 3, 1000)
```

```
## [1] 510
```

```
sim_kim(0.5, 4, 1000)
```

```
## [1] 672
```

```
sim_kim(0.5, 5, 1000)
```

```
## [1] 801
```

Vilket ger oss mycket rimliga resultat då sannolikheten ökar markant då startkapitalet är närmare kapitalmålet.

# 2 Uppgift 2

Ett annat sätt att undersöka Kims chans att lyckas är så här: Om Kims kapital efter $n$ spelomgångar är $X_n$ kronor (där $X_0$ betecknar startkapitalet) så kommer följden $X_n$ att bilda en Markovkedja. Den kommer förr eller senare att hamna i något av de båda absorberande tillstånden 0 eller 6. Låt oss beteckna dessa båda händelser med $X_\infty = 0$ respektive $X_\infty = 6$. Sannolikheten $\mathbb{P}(X_\infty = 6)$ kan vi approximativt räkna ut genom att utnyttja en ekvation som säger att om är övergångsmatrisen för en absorberande Markovkedja så gäller att

$$P^n \to \begin{pmatrix} 0 & SR \\ 0 & I \end{pmatrix}$$

då $n \to \infty$. Här ska matrisen vara skriven på "standardform", dvs de fem transienta tillstånden (1-5) ska komma först, och de båda rekurrenta sist. Sätt tillstånd 0 sist.

## 2.1 H ntering v SR-m trisen

Ett av elementen i matrisen $\boldsymbol{SR}$ beskriver sannolikheten att absorption sker i tillstånd 6 när spelaren startar med 1 krona.

Vilket element är det?

När vi undersöker absorptionssannolikheter i en Markovkedja med standardformen som har transienta tillstånd först och absorberande tillstånd sist, hittar vi dessa sannolikheter i SR-matrisen.

I vårt fall:

- Tillstånd 1-5 är transienta
- Tillstånd 6 och 0 är absorberande

SR-matrisen har dimension 5x2, där 5 är antalet transienta tillstånd och 2 är antalet absorberande tillstånd.

För att hitta sannolikheten att absorption sker i tillstånd 6 när spelaren startar med 1 krona, letar vi efter elementet i SR-matrisen som motsvarar övergången från transienta tillstånd 1 till absorberande tillstånd 6. Detta element finns i första raden och första kolumnen. Dvs på plats $SR$ .

## 2.2 Sk p övergångsm trisen

Skriv en funktion kims_matris i $R$, som tar som input en vinstsannolikhet $p$ och ger som output övergångsmatrisen för denna vinstsannolikhet. Du får följande till hjälp - allt du behöver göra är att byta ut några av 0:orna mot rätt sannolikhet.

```
kims_matris <- function(p) {
  m trix(c(
    0, p, 0, 0, 0, 0, 1-p,
    1-p, 0, p, 0, 0, 0, 0,
    0, 1-p, 0, p, 0, 0, 0,
    0, 0, 1-p, 0, p, 0, 0,
    0, 0, 0, 1-p, 0, p, 0,
    0, 0, 0, 0, 0, 1, 0, # tillstånd 6 abseroberande
    0, 0, 0, 0, 0, 0, 1 # tillstånd 0 absorberande
  ), nrow = 7, byrow = TRUE)
}
```

## 2.3 Beräkning v SR-m trisen

1. Beräkna $SR$-matrisen:

Du ska skriva en funktion, t.ex. `hitta_SR`, som tar en övergångsmatris $P$ (skriven på standardform, där de transienta tillstånden 1–5 kommer först och de absorberande tillstånden 6 och 0 sist) som indata.

Funktionen ska ge som output matrisen $SR$, vilken innehåller de absorberingssannolikheter som gäller för varje transient starttillstånd.

Du kan beräkna $SR$ antingen teoretiskt (då måste du förklara med ord och symboler vad de teoretiska resultaten innebär, samt var du hittat dem) eller genom att använda en numerisk metod där du bestämmer det $n$ för vilket $P^n \approx P^{n+}$ med fyra decimalers noggrannhet. Om du väljer den senare metoden ska du redovisa för vilket $n$ detta inträffar när vinstsannolikheten är $p = 0\,5$.

2. Jämförelse med simulerade resultat:

   Du ska utifrån $SR$-matrisen beräkna, för startkapitalen 1, 2, 3, 4 respektive 5, hur många av 1000 spel som "borde" ha slutat lyckligt (dvs. att Kim når 6 kronor).

   Därefter ska du jämföra dessa teoretiska sannolikheter med de simulerade resultaten från Uppgift 1 och utvärdera om de överensstämmer.

   Som en kommentar ska du även reflektera över om det finns en enkel formel som beskriver hur chansen att nå 6 kronor ökar med högre startkapital.

### 2.3.1 Beräkning v SR-m trisen

Vet att

$$S = (\quad \mathbb{P}_t)$$

```
hitta_SR <- function(P) {
  # Extrahera övergångsmatrisen för de transienta tillstånden (1-5)
  P_t <- P[1:5, 1:5] # extraherar de transienta tillstånden
  I <- di g(5) # skapar en 5x5 enhetsmatris

  # Beräkna den fundamentala matrisen S = (I - P_t)^{-1}
  S <- solve(I - P_t) # beräknar den fundamentala matrisen

  # Extrahera övergångarna från de transienta tillstånden till de absorberande (tillstånd 6 och 0)
  R <- P[1:5, 6:7] # extraherar övergångarna från de transienta tillstånden till de absorberande

  # Beräkna SR-matrisen
  SR <- S %*% R
  return(SR)
}
```

```
P <- kims_m tris(0.5) # skapar övergångsmatrisen för vinstsannolikheten 0.5
SR <- hitt _SR(P)
SR
```

```
##           [,1]      [,2]
## [1,] 0.1666667 0.8333333
## [2,] 0.3333333 0.6666667
## [3,] 0.5000000 0.5000000
## [4,] 0.6666667 0.3333333
## [5,] 0.8333333 0.1666667
```

```
P <- kims_m tris(0.5)
SR <- hitt _SR(P)
```

```r
# Teoretiska sannolikheter för att nå 6 kronor från olika startkapital
teoretiska_sannolikheter <- SR[, 1]
print(teoretiska_sannolikheter)
```

```
## [1] 0.1666667 0.3333333 0.5000000 0.6666667 0.8333333
```

```r
# Förväntat antal vinster av 1000 spel
forvantat_antal <- teoretiska_sannolikheter * 1000
print(forvantat_antal)
```

```
## [1] 166.6667 333.3333 500.0000 666.6667 833.3333
```

```r
# Jämförelse med simulerade resultat från uppgift 1.3
simulerade_resultat <- c(
  sim_kim(0.5, 1, 1000),
  sim_kim(0.5, 2, 1000),
  sim_kim(0.5, 3, 1000),
  sim_kim(0.5, 4, 1000),
  sim_kim(0.5, 5, 1000)
)

# Jämförelsetabell
resultat_jamforelse <- d t .fr me(
  Startkapital = 1:5,
  Teoretisk_sannolikhet = teoretiska_sannolikheter,
  Teoretiska_vinster = forvantat_antal,
  Simulerade_vinster = simulerade_resultat
)

resultat_jamforelse
```

| Startkapital | Teoretisk_sannolikhet | Teoretiska_vinster | Simulerade_vinster |
|---:|---:|---:|---:|
| 1 | 0.1666667 | 166.6667 | 158 |
| 2 | 0.3333333 | 333.3333 | 338 |
| 3 | 0.5000000 | 500.0000 | 505 |
| 4 | 0.6666667 | 666.6667 | 673 |
| 5 | 0.8333333 | 833.3333 | 841 |

Så när $p = 0\,5$ så är det teoretiska antalet vinster 500, och detta stämmer väl med det simulerade antalet vinster. Detta är även något vi kan förvänta oss enligt Spelarens ruin som vi har gått igenom i föreläsningen.

# 3 Uppgift 3 - S nnolikheten tt nå 6 kronor

Frågan går ut på att:

> Undersöka hur sannolikheten att, med ett startkapital på 1 krona, nå 6 kronor förändras när vinstsannolikheten ändras så att spelaren oftare förlorar (alltså när $p$ varierar från 20% upp till 80%).

nvända två metoder för att bestämma denna sannolikhet:

- **Simulering:** Kör simuleringar (som i Uppgift 1) med de olika värdena på $p$ och uppskatta andelen gånger spelaren når 6 kronor.
- **Teoretisk/numerisk metod:** nvänd samma metod som i Uppgift 2, d.v.s. studera konvergensen av $P^n$ (övergångsmatrisen skriven i standardform) för att bestämma absorberingssannolikheterna, med hänsyn till att det nödvändiga $n$ för konvergens ändras med $p$.

## 3.1 Lösning vi simulering

```
sim_kim(0.2, 1, 1000)
```

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

```
sim_kim(0.35, 1, 1000)
```

```
## [1] 15
```

```
sim_kim(0.5, 1, 1000)
```

```
## [1] 163
```

```
sim_kim(0.65, 1, 1000)
```

```
## [1] 470
```

```
sim_kim(0.8, 1, 1000)
```

```
## [1] 756
```

## 3.2 Lösning vi teori/numerisk metod

```
P_0.2 <- kims_m tris(0.2)
SR_0.2 <- hitt _SR(P_0.2)

# Teoretiska sannolikheter för att nå 6 kronor från olika startkapital
# Extraherar sannolikheterna för att nå 6 kronor från olika startkapital från kolumn 1 i SR-matrisen
teoretiska_sannolikheter <- SR[, 1]
teoretiska_sannolikheter
```

```
## [1] 0.1666667 0.3333333 0.5000000 0.6666667 0.8333333
```

```
# Gör samma för andra värden på p
P_0.35 <- kims_m tris(0.35)
SR_0.35 <- hitt _SR(P_0.35)

P_0.5 <- kims_m tris(0.5)
SR_0.5 <- hitt _SR(P_0.5)

P_0.65 <- kims_m tris(0.65)
SR_0.65 <- hitt _SR(P_0.65)
```

```
P_0.8 <- kims_m tris(0.8)
SR_0.8 <- hitt _SR(P_0.8)

d t .fr me(
  p = c(0.2, 0.35, 0.5, 0.65, 0.8),
  teoretisk_sannolikhet = c(SR_0.2[1, 1], SR_0.35[1, 1], SR_0.5[1, 1], SR_0.65[1, 1], SR_0.8[1, 1])
)
```

| p | teoretisk_sannolikhet |
|---|---|
| 0.20 | 0.0007326 |
| 0.35 | 0.0214140 |
| 0.50 | 0.1666667 |
| 0.65 | 0.4730691 |
| 0.80 | 0.7501832 |

# 4 Uppgift 4

På Rodmans Roulette måste man inte satsa exakt 1 krona vid varje spel. Det är tillåtet att satsa ett valfritt belopp. Om man vinner får man tillbaka dubbla insatsen.

- Robin brukar besöka Rodmans Roulette ibland. Hen har alltid 1 krona i startkapital, men följer en djärvare strategi än
- Kim: inför varje spelomgång satsar hen alla pengar hen har. Dock är Robin, precis som Kim, nöjd om hen kan nå upp till 6 kronor. Om hen vid ett visst tillfälle äger 4 eller 5 kronor, satsar hen alltså endast 2 kronor respektive 1 krona.

  Vi ska undersöka om Robin har bättre eller sämre chans att nå fram till sexkronorsmålet än vad Kim har.

## 4.1 Robins övergångsm tris:

Du ska beskriva hur Robins kapital utvecklas, vilket modelleras som en Markovkedja. Det innebär att du ska skriva en R-funktion, t.ex. `robins_matris`, som tar in en vinstsannolikhet $p$ och returnerar övergångsmatrisen för Robins strategi. I Robins strategi satsar hen hela sitt kapital i varje spelomgång, med undantag för när hen har 4 eller 5 kronor – då satsas endast 2 respektive 1 krona, så att målet 6 uppnås exakt.

```
# Skapa en funktion som skapar en övergångsmatris för Robins strategi
# Funktionen tar in en vinstsannolikhet p och returnerar övergångsmatrisen för Robins strategi
robins_matris <- function(p) {
  # Matris med 7 rader och 7 kolumner enligt standardform:
  # Transienta tillstånd: 1, 2, 3, 4, 5.  bsorberande: 6, 0.
  # Ordning: [1, 2, 3, 4, 5, 6, 0]
  m trix(c(
         0,    p,   0,    0,    0,    0,   1-p, # från kapital 1, kan hamna på 2 eller 0
         0,    0,   0,    p,    0,    0,   1-p, # från kapital 2, kan hamna på 4 eller 0
         0,    0,   0,    0,    0,    p,   1-p, # från kapital 3, kan hamna på 6 eller 0
         0,  1-p,   0,    0,    0,    p,    0, # från kapital 4, kan hamna på 6 eller 2
         0,    0,   0,  1-p,    0,    p,    0, # från kapital 5, kan hamna på 6 eller 4
         0,    0,   0,    0,    0,    1,    0, # tillstånd 6, absorberande
         0,    0,   0,    0,    0,    0,    1  # tillstånd 0, absorberande
  ), nrow = 7, byrow = TRUE)
}


robins_m tris(0.5)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]    0  0.5    0  0.0    0  0.0  0.5
## [2,]    0  0.0    0  0.5    0  0.0  0.5
## [3,]    0  0.0    0  0.0    0  0.5  0.5
## [4,]    0  0.5    0  0.0    0  0.5  0.0
## [5,]    0  0.0    0  0.5    0  0.5  0.0
## [6,]    0  0.0    0  0.0    0  1.0  0.0
## [7,]    0  0.0    0  0.0    0  0.0  1.0
```

## 4.2 Beräkning v $S$ -m trisen och vinstch ns:

Med hjälp av samma metod som i Uppgift 2 (där du beräknar $SR$-matrisen för en absorberande Markovkedja) ska du:

Räkna ut $SR$-matrisen för Robins övergångsmatris, givet olika värden på $p$ (20%, 35%, 50%, 65%, 80%).

vläsa från $SR$-matrisen Robins vinstchans, dvs sannolikheten att med start från 1 krona nå målet 6.

vgöra för vilka värden på $p$ det är fördelaktigt att spela "djärvt" (dvs följa Robins strategi) jämfört med att spela försiktigt.

Sammanfattat: Du ska först formulera Robins strategi genom en övergångsmatris via en R-funktion, och sedan analysera hur sannolikheten att nå 6 (uttryckt genom $SR$-matrisen) varierar med $p$, för att avgöra när strategin är optimal.

```
P_20 <- robins_m tris(0.2)
P_35 <- robins_m tris(0.35)
P_50 <- robins_m tris(0.5)
P_65 <- robins_m tris(0.65)
P_80 <- robins_m tris(0.8)

SR_20 <- hitt _SR(P_20)
SR_35 <- hitt _SR(P_35)
SR_50 <- hitt _SR(P_50)
SR_65 <- hitt _SR(P_65)
SR_80 <- hitt _SR(P_80)

SR_20
```

```
##            [,1]      [,2]
## [1,] 0.00952381 0.9904762
## [2,] 0.04761905 0.9523810
## [3,] 0.20000000 0.8000000
## [4,] 0.23809524 0.7619048
## [5,] 0.39047619 0.6095238
```

```
SR_35
```

```
##            [,1]      [,2]
## [1,] 0.05550162 0.9444984
## [2,] 0.15857605 0.8414239
## [3,] 0.35000000 0.6500000
## [4,] 0.45307443 0.5469256
## [5,] 0.64449838 0.3555016
```

```
SR_50
```

```
##           [,1]      [,2]
## [1,] 0.1666667 0.8333333
## [2,] 0.3333333 0.6666667
## [3,] 0.5000000 0.5000000
## [4,] 0.6666667 0.3333333
## [5,] 0.8333333 0.1666667
```

```
SR_65
```

```
##           [,1]       [,2]
## [1,] 0.3555016 0.64449838
## [2,] 0.5469256 0.45307443
## [3,] 0.6500000 0.35000000
## [4,] 0.8414239 0.15857605
```

```
## [5,] 0.9444984 0.05550162
```

```
SR_80
```

```
##           [,1]       [,2]
## [1,] 0.6095238 0.39047619
## [2,] 0.7619048 0.23809524
## [3,] 0.8000000 0.20000000
## [4,] 0.9523810 0.04761905
## [5,] 0.9904762 0.00952381
```

Läser av nu sannolikheten att nå 6 kronor från 1 krona för varje värde på $p$:

```
SR_20[1, 1]
```

```
## [1] 0.00952381
```

```
SR_35[1, 1]
```

```
## [1] 0.05550162
```

```
SR_50[1, 1]
```

```
## [1] 0.1666667
```

```
SR_65[1, 1]
```

```
## [1] 0.3555016
```

```
SR_80[1, 1]
```

```
## [1] 0.6095238
```

För att avgöra när strategin är optimal, jämför vi sannolikheterna för att nå 6 kronor från 1 krona för varje värde på $p$:

```
p_values <- c(0.2, 0.35, 0.5, 0.65, 0.8)
probabilities <- c(SR_20[1, 1], SR_35[1, 1], SR_50[1, 1], SR_65[1, 1], SR_80[1, 1])

d t .fr me(p = p_values, probability = probabilities)
```

| p | probability |
|---|---|
| 0.20 | 0.0095238 |
| 0.35 | 0.0555016 |
| 0.50 | 0.1666667 |
| 0.65 | 0.3555016 |
| 0.80 | 0.6095238 |

Djärvt att spela med $p = 0\,8$ eftersom sannolikheten att nå 6 kronor är 0.8.