

Szimulációk használata a sztochasztika oktatásában

SZALAI MÁTÉ

1. Bevezetés

Napjainkban bizonyos témák megtanítására egyre kevesebb idő jut, új szempontok kerültek előtérbe. A tanulónak/hallgatónak a tanórák számának csökkenése révén arányaiban nézve több ismeretet kell elsajátítaniuk. Éppen ezért az adott témákhoz tartozó információk átadásának hatékonyságát jelentősen növelni kell [3]. Kétségtelen, hogy a hatékonyság növelésének egyik legjobb módszere a vizualizáció. Mindez azt jelenti, hogy a tanórákat ki kell egészíteni változatos és látványos képi információkkal, melyek segíthetik a fogalmak, állítások, illetve tételek megértését.

Általános tapasztalat az egyetemi oktatásban, hogy a legelőször, a sztochasztika kurzus keretein belül előkerülő eloszlás, várható érték, illetve szórás szemléletes elsajátítása nehézkes, pedig ezek a fogalmak a matematika számos területén megjelennek, így megértésük elengedhetetlen. Arról nem is beszélve, hogy a sztochasztikus modellezésnél készségszinten kell tudni velük számolni.

A legtöbb esetben, a valószínűségszámításban előkerülő alapvető fogalmak bemutatása végett kísérletezünk, ahol a fenti mutatószámok jó közelítéssel kiszámolhatóak, de ennek több hátránya is van. Egyrészt egy kísérletet többször kell elvégezni, másrészt egy kísérlet elvégzése több ideig is eltarthat, vagy az is elképzelhető, hogy nem tudjuk elvégezni. Ekkor van lehetőségünk az informatika adta lehetőséghez nyúlni. Minimális szintű programozási ismerettel szinte bármilyen kísérletet, folyamatot tudunk szimulálni, és a szimulációs eredményekből a fontosabb mutatószámokat kiszámolni. Mindezek nagyban segíthetik az alapvető fogalmak megértését.

A tanulmány a következőképpen épül fel. Minden fejezet elején röviden ismertetjük az előkerülő fontosabb fogalmakat, állításokat, az ezek után következő példákat előbb szimuláljuk, bemutatjuk a kapott eredményeket, majd végül az explicit számolást is. Minden feladat megoldása végén megtalálhatóak az **RStudio**-ban Ctrl+Enter billentyűzetkombinációval futtatható forráskódok. A 2. fejezetben a véletlen változók eloszlására, várható értékére, valamint szórására hozunk példákat. A 3. fejezetben az elemi valószínűségszámítás utáni témakörök sztochasztikus modellezés szemszögéből kerülnek bemutatásra, úgy mint a véletlen bolyongás, a játékos csődje, valamint a Galton–Watson-folyamat.

2. Véletlen változók eloszlása, várható értéke és szórása

A sztochasztika egyik fontos fogalma a véletlen (valószínűségi) változó, valamint annak eloszlása, várható értéke és szórása. Mindezek a fogalmak, és a kapcsolódó állítások megtalálhatóak [6]-ban.

A véletlen változó hétköznapi megjelenése a véletlen szám, mely egy véletlen kísérletből származik. Ilyen például egy kockadobás eredménye, az, hogy hány számot találunk el az ötöslottón, de az is véletlen változó, hogy milyen messze találjuk el a középponttól a dartstáblát. Legyen $(\Omega, \mathcal{A}, \mathbf{P})$ egy tetszőleges véletlen kísérletet leíró valószínűségi mező, és tekintsünk egy $X : \Omega \rightarrow \mathbb{R}$ leképezést. Ha a kísérlet elvégzésekor bekövetkezik egy ω elemi esemény, akkor az X leképezés produkál egy $X(\omega)$ véletlen számot. Ekkor X -et véletlen változónak nevezzük. Egy véletlen változó diszkrét, ha értékészlete megszámlálható (azaz véges, vagy megszámlálhatóan végtelen). Ha egy X diszkrét véletlen változó lehetséges értékei $R_X = \{x_1, x_2, \dots\}$, akkor $\mathbf{P}(X = x_i) = p_i > 0$ a változó eloszlása.

Figyeljük meg az X véletlen változó értékeit n -szer, egymástól függetlenül, a kapott minta legyen $X_1, \dots, X_n$. A tapasztalat azt mutatja, hogy az esetek nagy részében az

$$\frac{X_1 + \dots + X_n}{n}$$

átlagok egy rögzített érték körül stabilizálódnak, ha n elég nagy. Ezt az értéket az X véletlen változó tapasztalati várható értékének (átlagának) nevezzük, és $\widehat{\mathbf{E}}(X)$ -el jelöljük. Egy diszkrét X véletlen változó (elméleti) várható értékén az

$$\mathbf{E}(X) = \sum_{k \in R_X} k \mathbf{P}(X = k),$$

összeget értjük, feltéve, hogy a sor abszolút konvergens.

További fontos információ még, hogy az X véletlen változó $X_1, \dots, X_n$ megfigyelt értékei átlagosan mennyivel térnek el a tapasztalati várható értéktől. Minderre adja meg a választ a tapasztalati szórás, mely a tapasztalati variancia négyzetgyöke, amit a következőképpen számolunk ki

$$\widehat{\text{Var}}(X) = \frac{[X_1 - \widehat{\mathbf{E}}(X)]^2 + \dots + [X_n - \widehat{\mathbf{E}}(X)]^2}{n},$$

melyből a tapasztalati szórás $\widehat{\mathbf{D}}(X) = \sqrt{\widehat{\text{Var}}(X)}$. A szórás tehát egy olyan mérőszám, mely megadja a várható érték körüli véletlen ingadozás mértékét. Hogyha az $\mathbf{E}(X)$ és $\mathbf{E}((X - \mathbf{E}(X))^2)$ várható értékek léteznek, akkor egy X véletlen változó (elméleti) szórása

$$\mathbf{D}(X) = \sqrt{\mathbf{E}((X - \mathbf{E}(X))^2)} = \sqrt{\mathbf{E}(X^2) - \mathbf{E}^2(X)},$$

ahol

$$\mathbf{E}(X^2) = \sum_{k \in R_X} k^2 \mathbf{P}(X = k)$$

az X véletlen változó második momentuma.

2.1. Feladat. Egy fiókban három pár kesztyű van összekeveredve: az egyik pár fekete, a másik szürke, a harmadik piros. (A három pár kesztyű csak a színében különböző.) A fiókból egyesével elkezdjük kihúzni a kesztyűket úgy, hogy húzás előtt nem nézzük meg a kesztyű színét, és a kihúzott kesztyűket nem tesszük vissza a fiókba. Addig folytatjuk a húzást, amíg lesz két azonos színű kesztyűnk. Határozzuk meg annak a hat eseménynek a valószínűségét, hogy ehhez 1, 2, 3, 4, 5, illetve 6 kesztyű kihúzására lesz szükség, majd számítsuk ki a húzások számának várható értékét és szórását! (*Emelt szintű írásbeli érettségi vizsga - 2017. május*)

Megoldás. A tényleges megoldás előtt a kísérletet szimulálhatjuk, és a szimulációs adatokból sejtéseket fogalmazhatunk meg. Továbbá egy helyesen megírt szimulációs kód lehetőséget ad arra is, hogy a végeredmény(einke)t ellenőrizzük. Jelölje X , hogy hányszor kellett húznunk, hogy legyen egy pár.

A nagy számok Bernoulli-féle gyenge törvénye (ld. [6]-ban a 4.7. Következmény) szerint, ha egy véletlen kísérletet elég sokszor ismétlünk meg, akkor egy adott esemény relatív gyakorisága jól fogja közelíteni annak valószínűségét. A bevezetőben említettek szerint, vannak olyan kísérletek, melyek elvégzése időigényes, ez a feladat egy jó példája ennek.

Az 1. táblázat mutatja a relatív gyakoriság szimulációs eredményeit. A folyamatot 10-szer, 100-szor, 1000-szer, illetve 10.000-szer szimuláljuk. Jelölje $r_n(X = k)$ a relatív gyakoriságát az $X = k$ eseménynek, hogyha n független véletlen kísérletet végzünk, vagyis

$$r_n(X = k) := \frac{K_n(X = k)}{n},$$

ahol $K_n(X = k)$ az $X = k$ esemény bekövetkezéseinek száma az első n megfigyelésből.

n	$r_n(X = 1)$	$r_n(X = 2)$	$r_n(X = 3)$	$r_n(X = 4)$	$r_n(X = 5)$	$r_n(X = 6)$
10	0	0,18	0,41	0,41	0	0
100	0	0,2	0,41	0,39	0	0
1000	0	0,223	0,39	0,387	0	0
10.000	0	0,199	0,4051	0,3959	0	0

1. táblázat. Az $X = k$ események relatív gyakoriságai a 2.1. Feladatban a szimulációk számának függvényében.

Az 1. táblázatban láthatjuk, hogy az $X = 1$, $X = 5$, valamint $X = 6$ relatív gyakoriságai nullák, végiggondolva ez mind lehetetlen esemény. A szükséges húzások átlagos értékeit, illetve ezen értékek szórását adja meg a 2. táblázat.

Mindezek után pedig a feladat tényleges megoldása következik. Az X véletlen változó lehetséges értékei 2, 3, 4. Így

$$\mathbf{P}(X = 1) = \mathbf{P}(X = 5) = \mathbf{P}(X = 6) = 0.$$

Elsőre kihúzzuk valamelyik kesztyűt. Ha $X = 2$, akkor a második kihúzott kesztyű éppen az első párja. Mivel 5 kesztyű maradt, ennek a valószínűsége

$$\mathbf{P}(X = 2) = \frac{1}{5}.$$

szimulációk száma	10	100	1000	10.000
értékek átlaga	3,4	3,11	3,24	3,2
értékek szórása	0,663	0,773	0,739	0,741

2. táblázat. A szimulált eredmények átlaga, valamint szórása a szimulációk számának függvényében a 2.1. Feladatban.

Ha $X = 3$, akkor másodikra nem húztuk ki az első párját, ennek a valószínűsége $4/5$, harmadikra viszont párt húztunk. A harmadik húzás előtt 4 kesztyű van, ebből 2 jó, így a pár valószínűsége $1/2$. Vagyis

$$\mathbf{P}(X = 3) = \frac{4}{5} \cdot \frac{2}{4} = \frac{2}{5}.$$

Végül pedig

$$\mathbf{P}(X = 4) = 1 - (\mathbf{P}(X = 2) + \mathbf{P}(X = 3)) = \frac{2}{5}.$$

A várható érték

$$\mathbf{E}(X) = 2 \cdot \frac{1}{5} + 3 \cdot \frac{2}{5} + 4 \cdot \frac{2}{5} = \frac{16}{5} = 3,2.$$

A második momentum

$$\mathbf{E}(X^2) = 2^2 \cdot \frac{1}{5} + 3^2 \cdot \frac{2}{5} + 4^2 \cdot \frac{2}{5} = \frac{54}{5},$$

és így a szórás

$$\mathbf{D}(X) = \sqrt{\mathbf{E}(X^2) - (\mathbf{E}(X))^2} = \frac{\sqrt{14}}{5} \approx 0,748.$$

Láthatjuk, hogy a számolt eredmények, illetve a szimulált adatok nagymértékű hasonlóságot mutatnak.

Minden forráskód a következőképpen épül fel. Megadunk egy függvényt, mely tartalmazza a bemeneti információkat, és elvégzi a megadott számolást, elkészíti az ábrát. Ezek nem változtathatóak, különben az RStudio hibaüzenetet fog kiírni. Vannak bemeneti paraméterek, pl. hányszor szimulálunk, melyek változtathatóak a kódban, ezt mindig külön jelöljük. A futtatáshoz ki kell másolni a forráskódot, beilleszteni az RStudio-ba, a kurzort a kódok elé kell vinni, és a Ctrl+Enter billentyűzetkombinációt kell megnyomni. A kódok beolvasása után az utolsó parancs fogja kiírni a számolt eredményt, vagy megadni az ábrát. A kódban kommentek vannak elhelyezve, melyek mindig a # jel után találhatók.

Forráskód. # Függvény kezdete, NEM változtatható

```
simulategloves <- function() {
gloves <- c(rep("black", 2), rep("gray", 2), rep("red", 2))
```

```

draws <- 0
selectedgloves <- character(0)

while (length(unique(selectedgloves)) >= draws) {
  drawnglove <- sample(gloves, 1)
  selectedgloves <- c(selectedgloves, drawnglove)
  gloves <- gloves[-which(gloves == drawnglove)[1]]
  draws <- draws + 1
}

return(draws)
}
# Függvény vége

numsimulations <- 100 # szimulációk száma, változtatható
totaldraws <- 0 # NEM változtatható
drawslist <- c() # NEM változtatható
stopsafter <- rep(0, 6) # NEM változtatható

# Szimuláció kezdete, NEM változtatható
for (i in 1:numsimulations) {
  draws <- simulategloves()
  totaldraws <- totaldraws + draws
  drawslist <- c(drawslist, draws)
  stopsafter[draws] <- stopsafter[draws] + 1
}
# Szimuláció vége

# Megállások számának kiszámítása és kiírása
stopsafter <- stopsafter / numsimulations
for (i in 1:length(stopsafter)) {
  print(paste(i, "relatív gyakorisága:", stopsafter[i]))
}

# Szimulált adatok átlagának és szórásának kiszámítása, NEM változtatható
expecteddraws <- totaldraws / numsimulations
print(paste("Értékek átlaga:", expecteddraws))

meandraws <- mean(drawslist)
variance <- mean((drawslist - meandraws)^2)
standarddeviation <- sqrt(variance)
print(paste("Értékek szórása:", standarddeviation))

```

2.2. Feladat. Legyenek az $U_1, U_2, \dots$ független, a $(0,1)$ intervallumon egyenletes

eloszlású véletlen változók. Továbbá legyen

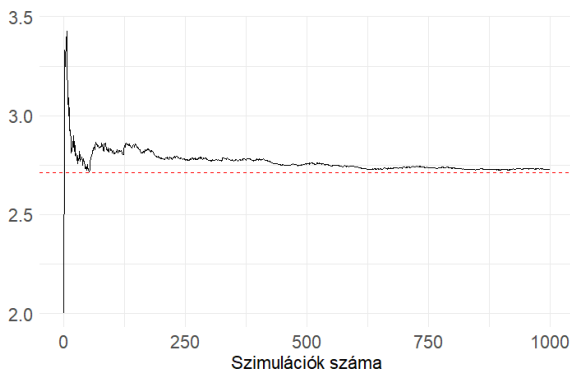
$$N := \min \left\{ n : \sum_{i=1}^n U_i > 1 \right\}.$$

Határozzuk meg $\mathbf{E}(N)$ értékét!

Megoldás. A feladat arra keresi a választ, hogy átlagosan hány darab $(0, 1)$ -ről választott véletlen számot kell összeadnunk ahhoz, hogy az összeg meghaladja az egyet? A probléma szimulálása olyan esetekben is segítségünkre lehet, hogyha megsejteni szeretnénk a megoldást. A 3. táblázat, valamint az 1. ábra mutatják a szimulált eredmények átlagát, melyekből sejthetjük, hogy ez az átlagos érték e körül van.

szimulációk száma	10	100	1000	10.000	100.000
értékek átlaga	3	2,67	2,71	2,723	2,71587

3. táblázat. A szimulált adatok átlaga a 2.2. Feladatban a szimulációk számának függvényében.

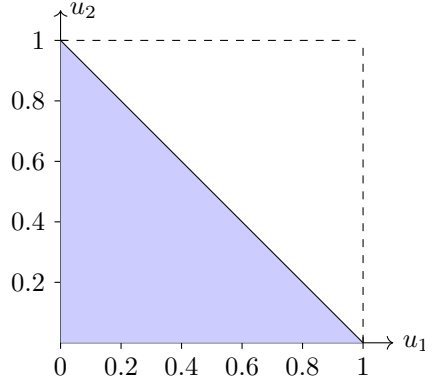


1. ábra. Átlagos értékek ábrázolása a 2.2. Feladatban, ahol a szaggatott vonal az e értéke.

Legelőször meghatározzuk annak a valószínűségét, hogy $N = 2$.

$$\mathbf{P}(N = 2) = \mathbf{P}(U_1 + U_2 > 1) = 1 - \mathbf{P}(U_1 + U_2 \leq 1).$$

Geometriailag is tudjuk reprezentálni az $U_1 + U_2 \leq 1$ valószínűséget. Mivel az U_1, U_2 változók egyenletes eloszlásúak a $(0, 1)$ -en, ezért az (U_1, U_2) koordinátájú pont bárhol előfordulhat az egységnégyzeten. Emiatt a $\mathbf{P}(U_1 + U_2 \leq 1)$ valószínűség éppen a 2. ábrán a beszínezett rész arányával egyezik meg az egységnégyzetben.

2. ábra. $U_1 + U_2 \leq 1$ szemléltetése.

$$\mathbf{P}(U_1 + U_2 \leq 1) = \int_0^1 \int_0^{1-u_2} du_1 du_2 = \int_0^1 (1 - u_2) du_2 = \frac{1}{2}.$$

Tehát $\mathbf{P}(N = 2) = 1 - 1/2 = 1/2$. Ezek után megvizsgáljuk ugyanezt $N = 3$ esetben is.

$$\mathbf{P}(N = 3) = \mathbf{P}(U_1 + U_2 + U_3 > 1, U_1 + U_2 \leq 1),$$

azonban a két egyenlőtlenség egyidejű kezelése nem egyszerű, viszont egyszerűsít a dolgunkon az az észrevétel, hogy

$$\mathbf{P}(N = 3) = \mathbf{P}(N \leq 3) - \mathbf{P}(N = 2),$$

ahol

$$\mathbf{P}(N \leq 3) = \mathbf{P}(U_1 + U_2 + U_3 > 1) = 1 - \mathbf{P}(U_1 + U_2 + U_3 \leq 1).$$

$$\begin{aligned} \mathbf{P}(U_1 + U_2 + U_3 \leq 1) &= \int_0^1 \int_0^{1-u_3} \int_0^{1-u_3-u_2} du_1 du_2 du_3 \\ &= \int_0^1 \frac{1}{2} (1 - 2u_3 + u_3^2) du_3 \\ &= \frac{1}{6}. \end{aligned}$$

Tehát

$$\mathbf{P}(N = 3) = \mathbf{P}(N \leq 3) - \mathbf{P}(N = 2) = 1 - \frac{1}{6} - \frac{1}{2} = \frac{1}{3}.$$

Általánosan látjuk tehát, hogy

$$\begin{aligned}\mathbf{P}(N = n) &= \mathbf{P}(N \leq n) - \mathbf{P}(N \leq n - 1) \\ &= \left(1 - \frac{1}{n!}\right) - \left(1 - \frac{1}{(n-1)!}\right) \\ &= \frac{n-1}{n!}.\end{aligned}$$

$$\mathbf{E}(N) = \sum_{n=2}^{\infty} n \mathbf{P}(N = n) = \sum_{n=2}^{\infty} n \left(\frac{n-1}{n!}\right) = \sum_{n=2}^{\infty} \frac{1}{(n-2)!} = \sum_{n=0}^{\infty} \frac{1}{n!} = e.$$

Forráskód. # Függvény kezdete, NEM változtatható

```
simulateprocess <- function() {
```

```
S <- 0 # Kezdőállapot
```

```
steps <- 0 # Lépések száma
```

```
while (S < 1) {
```

```
U <- runif(1) # Véletlenszerű szám a [0,1] intervallumból
```

```
S <- S + U # Új állapot
```

```
steps <- steps + 1
```

```
}
```

```
return(steps)
```

```
}
```

Függvény vége

```
numsimulations <- 100000 # Szimulációk száma, változtatható
```

```
totalsteps <- 0 # Összes lépés száma, NEM változtatható
```

```
# Szimulált adatok átlagának kiszámítása, NEM változtatható
```

```
for (i in 1:numsimulations) {
```

```
totalsteps <- totalsteps + simulateprocess()
```

```
}
```

```
expectedsteps <- totalsteps / numsimulations
```

```
print(expectedsteps)
```

Szimulált adatok átlagának kiszámítása vége

```
# Átlagos értékek ábrázolása, NEM változtatható
```

```
totalsteps <- 0
```

```
results <- numeric(numsimulations)
```

```
for (i in 1:numsimulations) {
```

```
totalsteps <- totalsteps + simulateprocess()
```

```

expectedsteps <- totalsteps / i
results[i] <- expectedsteps
}

library(ggplot2)

df <- data.frame(Simulations = 1:numsimulations, ExpectedSteps = results)

ggplot(df, aes(x = Simulations, y = ExpectedSteps)) +
  geom_line() +
  geom_hline(yintercept = 2.71, linetype = "dashed", color = "red") +
  geom_text(aes(x = Inf, y = 2.71, label = "2.71"), color = "red", hjust =
-0.2, size = 4) +
  labs(title = "",
x = "Szimulációk száma",
y = "") +
  theme_minimal() +
  theme(
axis.text = element_text(size = 15),
axis.title = element_text(size = 15)
)
# Átlagos értékek ábrázolásának vége

```

3. Diszkrét idejű Markov-láncok

A fejezetben előkerülő fogalmak, tételek, állítások megtalálhatóak [4]-ben. Legyen T tetszőleges indexhalmaz, X_t pedig véletlen változó egy közös $(\Omega, \mathcal{A}, \mathbf{P})$ valószínűségi mezőn minden $t \in T$ esetén. Ekkor a véletlen változók $\mathbf{X} = \{X_t : t \in T\}$ halmazát sztochasztikus folyamatnak nevezzük. Az X_t változók által felvehető értékek I halmazát pedig a folyamat állapotterének hívjuk. Ezen cikk keretében csak olyan esettel foglalkozunk, amikor $T \subset \mathbb{R}$, ezáltal adódik egy rendezés a T halmazon, amit időnek nevezünk, míg a $t \in T$ értéket nevezhetjük időparaméternek. Legyen $t \in T$ és $i \in I$ tetszőleges, ekkor az $\mathbf{X}$ sztochasztikus folyamat a t időpontban az i állapotban van, hogyha $X_t = i$. Az idő tekintetében legyen $T = \mathbb{Z}_+ = \{0, 1, 2, \dots\}$, így $\mathbf{X}$ diszkrét idejű sztochasztikus folyamat.

Az $\mathbf{X} = \{X_n : n \in \mathbb{Z}_+\}$ diszkrét idejű diszkrét állapotterű sztochasztikus folyamatot Markov-láncnak nevezzük, ha rendelkezik a memória nélküli tulajdonsággal, azaz ha tetszőleges $n \in \mathbb{N}$ és tetszőleges $i_0, i_1, \dots, i_n, i_{n+1} \in I$ állapotok esetén

$$\mathbf{P}(X_{n+1} = i_{n+1} | X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = \mathbf{P}(X_{n+1} = i_{n+1} | X_n = i_n),$$

abban az esetben, ha

$$\mathbf{P}(X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) > 0.$$

Az $\mathbf{X}$ Markov-lánc egy lépéses átmenetvalószínűségei alatt a

$$p_{i,j}(n) = \mathbf{P}(X_{n+1} = j | X_n = i), \quad n \in \mathbb{Z}_+, \quad i, j \in I$$

valószínűséget értjük. Ha ezen átmenetvalószínűségek nem függenek n értékétől, akkor a Markov-lánc (idő)homogén. Homogén Markov-lánc n -lépéses átmenetvalószínűségei

$$p_{i,j}^{(n)} = \mathbf{P}(X_n = j | X_0 = i), \quad i, j \in I.$$

A j állapot elérhető az i állapotból, hogyha

$$\mathbf{P}(\exists n \in \mathbb{N} : X_n = j | X_0 = i) > 0.$$

Az i , és a j állapotok kommunikációs viszonyban állnak egymással, ha kölcsönösen elérhetőek egymásból. A kommunikációs viszony, mint ekvivalenciareláció által egyértelműen meghatározott ekvivalenciaosztályokat kommunikációs osztályoknak nevezzük. Tehát egy kommunikációs osztályba azon állapotok tartoznak, melyek egymással kommunikációs viszonyban állnak. Ha egy Markov-láncnak csak egy kommunikációs osztálya van, azaz minden állapot kommunikál minden állapottal, irreducibilisnek nevezzük.

Az i állapotba történő visszatérések száma

$$V_i := |\{n \in \mathbb{N} : X_n = i\}|,$$

az i állapot r -edik visszatérési ideje:

$$T_{i,r} := \min\{n > T_{i,r-1} : X_n = i\}, \quad r = 1, 2, \dots, \quad T_{i,0} := 0.$$

Az i állapot tranziens, hogyha $\mathbf{P}(V_i < \infty) = 1$, illetve rekurrens, ha $\mathbf{P}(V_i = \infty) = 1$, továbbá ha az is teljesül $\mu_i = \infty$, akkor az állapot null-rekurrens, valamint, hogyha $\mu_i < \infty$, akkor pozitív rekurrens, ahol $\mu_i := \mathbf{E}(T_{i,1})$. Egy $C \subseteq I$ kommunikációs osztály zárt, hogyha minden $i \in C$, és minden $j \in I \setminus C$ esetén $p_{i,j} = 0$, különben pedig nyílt. Tetszőleges nyílt kommunikációs osztály tranziens. Azok a kommunikációs osztályok, melyek végesek, és zártak pozitív rekurrenssek.

Legyen $A \subseteq I$, ekkor

$$p_i^A = \mathbf{P}(\text{valaha elérjük } A\text{-t} | X_0 = i) = \begin{cases} 1, & \text{ha } i \in A, \\ \sum_{j \in I} p_{i,j} p_j^A, & \text{ha } i \notin A, \end{cases}$$

$$h_i^A = \mathbf{E}(\text{hány lépésben érjük el } B\text{-t} | X_0 = i) = \begin{cases} 0, & \text{ha } i \in A, \\ 1 + \sum_{j \in I} p_{i,j} h_j^A, & \text{ha } i \notin A. \end{cases}$$

3.1. Feladat. Szabályos dobókockával dobunk, és azt tartjuk számon, hogy az eddigi dobások közül mekkora volt a legnagyobb. Vagyis n dobás után a folyamatunk a j állapotban van, ha az első n dobás közül j volt a maximális, tehát

$$X_n := \max_{j=1,\dots,n} \xi_j,$$

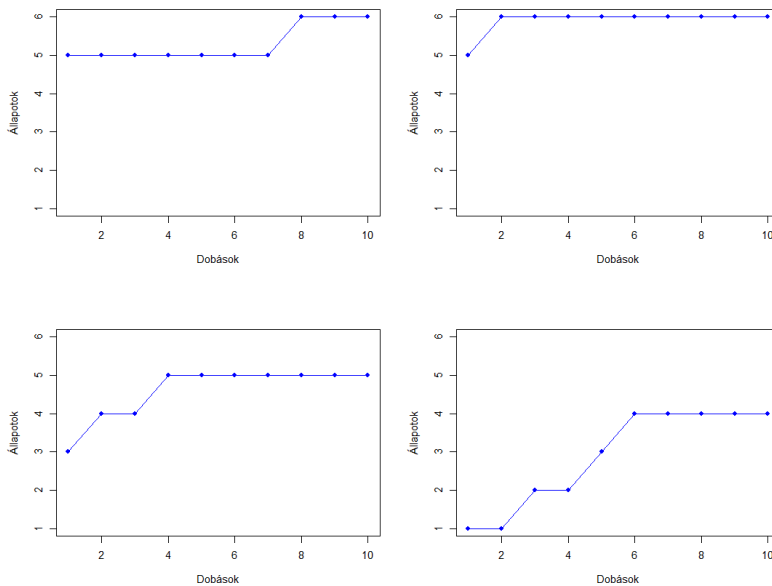
ahol ξ_j a j -edik kockadobás eredményét jelölő véletlen változó.

(a) Markov-lánc-e a folyamat?

(b) Adjuk meg az átmenetmátrixot!

(c) Mi X_2 eloszlása, ha a folyamat kezdeti eloszlása $\alpha = (0, 1, 0, 0, 0, 0)$?

Megoldás. A folyamat négy szimulációját a 3. ábra mutatja. Jól látható, hogy a hat elnyelő állapota a folyamatnak, vagyis ha abba az állapotba belép a folyamat, akkor onnan természetesen nem tud kilépni.



3. ábra. A 3.1. Feladat négy szimulációja.

- (a) A folyamat Markov-lánc, hiszen csak az számít, hogy egy adott időpontban éppen melyik állapotban van a folyamat, annak, hogy oda hogyan jut el, nincs jelentősége.
- (b) A folyamat átmenetmátrixa, mely tartalmazza a $p_{i,j}$ valószínűségeket

$$\mathbb{P} = \begin{pmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & \frac{1}{3} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & 0 & \frac{4}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & \frac{5}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

(c)

$$X_2 \sim \alpha \mathbb{P}^2 = \left(0, \frac{4}{36}, \frac{5}{36}, \frac{7}{36}, \frac{9}{36}, \frac{11}{36}\right).$$

```

Forráskód. par(mfrow = c(2, 2)) # Ábra méretének beállítása, NEM változtatható
n <- 10 # Dobások száma, változtatható
j <- 1 # Kezdő állapot, NEM változtatható

# Függvény megadása, NEM változtatható
simulateprocess <- function(n, j) {
  currentstate <- j
  statesequence <- vector("numeric", n)

  for (i in 1:n) {
    dobás <- sample(1:6, 1)

    if (dobás > currentstate) {
      currentstate <- dobás
    }

    statesequence[i] <- currentstate
  }

  return(statesequence)
}
# Függvény vége

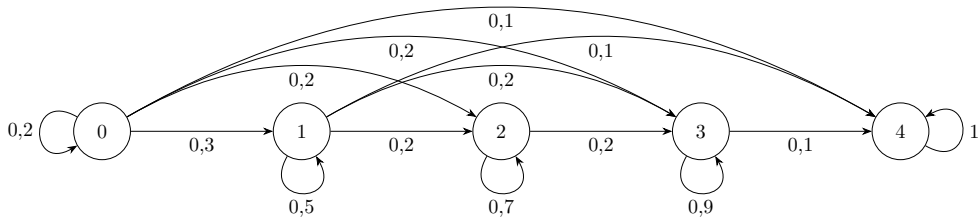
# Ábrázolás, NEM változtatható
par(mfrow = c(2, 2))
for (i in 1:4) {
  statesequence <- simulateprocess(n, j)
  plot(1:n, statesequence, type = "o", xlab = "Dobások",
  ylab = "Állapotok", ylim = c(1, 6), pch = 16, col = "blue")
}
# Ábrázolás vége

```

3.2. Feladat. Kedvenc focicsapatunk a korábbi fordulók eredményétől függetlenül egy-egy meccsen 0,2 valószínűséggel nem szerez gólt; 0,3 valószínűséggel 1 gólt; 0,2 valószínűséggel 2 gólt; 0,2 valószínűséggel 3 gólt; végül 0,1 valószínűséggel 4 gólt szerez. (Arra még a legöregebb szurkolók sem emlékeznek, hogy a fiúk valaha is 4-nél többet vágtak volna, szóval ettől az esettől eltekinthetünk.) Legyen X_n a rúgott gólok számának maximuma n forduló után. Markov-láncot alkot-e az X_n sorozat? Adjuk meg a folyamat átmenetgráfját, a kommunikációs osztályokat és az állapotok típusát.

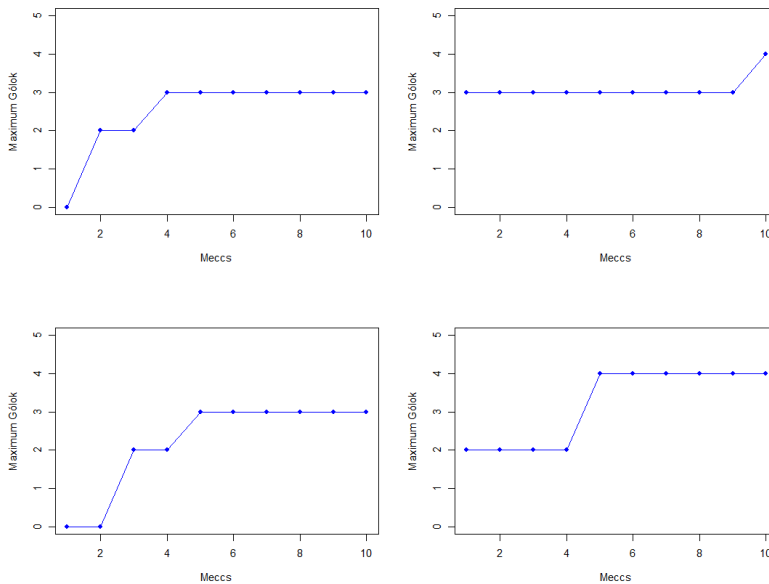
Megoldás. A folyamat négy szimulációját az 5. ábra mutatja. A négy elnyelő állapota a folyamatnak, hiszen ebből az állapotból egy valószínűséggel önmagába jutunk.

X_n Markov-lánc, hiszen elég az n . forduló után tudnunk, hogy eddig mennyi volt a legtöbb gól amit lőttek, ami épp X_n . Ebből az addigiaktól függetlenül meg tudjunk mondani, hogy hány gólt mekkora valószínűséggel lő a csapatunk, ami épp azt jelenti, hogy a folyamatunk Markov-lánc. A folyamat átmenetgráfját a 4. ábra mutatja.



4. ábra. A 3.2. Feladat átmenetgráfja.

A gráfból jól látható, hogy a kommunikációs osztályok a $\{0\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, melyből az első négy nyílt, ők tranziensek, míg az utolsó kommunikációs osztály véges és zárt, így pozitív rekurrens.



5. ábra. A 3.2. Feladat négy szimulációja.

Forráskód. # Valószínűségek beállítása minden meccsre, NEM változtatható
`valoszinusegekmind <- matrix(c(0.2, 0.3, 0.2, 0.2, 0.1,`
`0.2, 0.3, 0.2, 0.2, 0.1,`
`0.1, 0.2, 0.3, 0.2, 0.2,`
`0.2, 0.1, 0.2, 0.3, 0.2,`
`0.3, 0.2, 0.2, 0.1, 0.2,`

```

0.2, 0.3, 0.2, 0.2, 0.1,
0.1, 0.2, 0.3, 0.2, 0.2,
0.2, 0.1, 0.2, 0.3, 0.2,
0.3, 0.2, 0.2, 0.1, 0.2,
0.2, 0.3, 0.2, 0.2, 0.1), nrow = 10, byrow = TRUE)

# Függvény megadása, NEM változtatható
simulaciomind <- function() {
  allapot <- 0
  maxgolallapot <- 0
  allapotok <- numeric(10)
  for (i in 1:10) {
    gollok <- sample(0:4, 1, replace = TRUE, prob = valoszinusegekmind[i, ])
    if (gollok > maxgolallapot) {
      maxgolallapot <- gollok
    }
    if (gollok > allapot) {
      allapot <- gollok
    }

    allapotok[i] <- allapot
  }
  return(allapotok)
}

# Függvény megadásának vége

nfordulok <- 4 # NEM változtatható
eredmenyek <- replicate(nfordulok, simulaciomind()) # NEM változtatható

# Ábrázolás, NEM változtatható
par(mfrow = c(2, 2))
for (i in 1:nfordulok) {
  statesequance <- eredmenyek[, i]
  plot(1:10, statesequance, type = "o", xlab = "Meccs",
       ylab = "Maximum Gólok", ylim = c(0, 5), pch = 16, col = "blue")
}

# Ábrázolás vége

```

3.1. Véletlen bolyongás

Legyen $d \in \mathbb{N}$ tetszőleges, és tekintsük az $\mathbf{X}(d)$ d -dimenziós szimmetrikus véletlen bolyongást, azaz legyenek a lánc állapotai a d -dimenziós rács pontjai, indítsuk ki az origóból, és minden lépésben léptessük a láncot valamelyik szomszédos állapotba diszk-

rét egyenletes eloszlás szerint, minden korábbi lépéstől függetlenül, azaz

$$p_{i,j} = \begin{cases} \frac{1}{2d}, & \text{ha } |i - j| = 1, \\ 0, & \text{különben,} \end{cases} \quad i, j \in \mathbb{Z}^d.$$

A 6. ábra egy kétdimenziós szimmetrikus véletlen bolyongás szimulációját, valamint a 8. ábra négy trajektóriáját mutatja egy speciális egydimenziós véletlen bolyongásnak. A d -dimenziós szimmetrikus véletlen bolyongás egy irreducibilis Markov-lánc, valamint a Pólya-tétele alapján null-rekurrens [5], ha $d = 1, 2$, valamint tranziens, ha $d \geq 3$. Megmutatjuk $d = 1$ esetén, hogy a tétel valóban igaz.

Ha a lánc a 0 állapotból indul ki, akkor csak páros sok lépésben térhet vissza oda. Pontosan $2n$ lépésben akkor tér vissza, ha n lépést tesz jobbra és n lépést balra. Mivel a jobbra tett lépések száma binomiális eloszlást követ, kapjuk, hogy

$$p_{0,0}^{(2n)} = \binom{2n}{n} p^n q^n = \frac{(2n)!}{(n!)^2} (pq)^n, \quad n = 0, 1, \dots$$

ahol $q = 1 - p$. Az ismert Stirling-formula szerint

$$n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}, \quad n \rightarrow \infty,$$

amiből

$$p_{0,0}^{(2n)} \sim a_n = \frac{(2n/e)^{2n} \sqrt{4\pi n}}{[(n/e)^n \sqrt{2\pi n}]^2} (pq)^n = \frac{(4pq)^n}{\sqrt{\pi n}}, \quad n \rightarrow \infty.$$

Ekkor tetszőleges $\varepsilon > 0$ esetén létezik n_0 , hogy $n > n_0$ mellett $a_n(1 - \varepsilon) \leq p_{0,0}^{(2n)} \leq a_n(1 + \varepsilon)$, amiből adódik, hogy

$$\sum_{n=0}^{\infty} p_{i,i}^{(2n)} \quad \text{és} \quad \sum_{n=1}^{\infty} a_n$$

pontosan ugyanazokra a p értékekre konvergensek, illetve divergensek. Hogyha $p \neq 1/2$, akkor $4pq < 1$, és

$$\sum_{n=1}^{\infty} a_n \leq \sum_{n=1}^{\infty} (4pq)^n = \frac{1}{1 - 4pq} < \infty,$$

vagyis a nem szimmetrikus esetben a 0 állapot tranziens. Ha $p = 1/2$, akkor $4pq = 1$, és

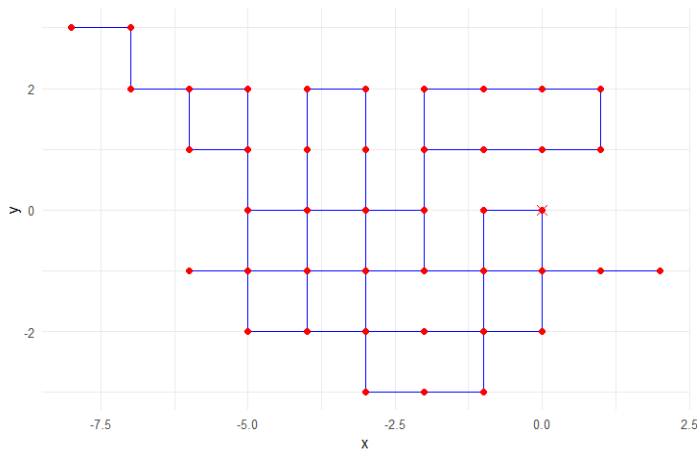
$$\sum_{n=1}^{\infty} a_n = \frac{1}{\sqrt{\pi}} \sum_{n=1}^{\infty} \frac{1}{n^{1/2}} = \infty,$$

azaz a szimmetrikus esetben a 0 állapot rekurrens. Emellett $a_n \rightarrow 0$, melyből $p_{0,0}^{(n)} \rightarrow 0$, így null-rekurrens. Felhasználtuk a típusok karakterizációjáról szóló tételt, mely szerint az i állapot pontosan akkor tranziens, hogyha

$$\sum_{n=0}^{\infty} p_{i,i}^{(n)} < \infty,$$

valamint pontosan akkor null-rekurrens, ha

$$\sum_{n=0}^{\infty} p_{i,i}^{(n)} = \infty, \quad \text{és} \quad \lim_{n \rightarrow \infty} p_{i,i}^{(n)} = 0.$$



6. ábra. Kétdimenziós szimmetrikus véletlen bolyongás szimulációja (lépésszám = 100).

Forráskód. `library(ggplot2)`

```
# Véletlen bolyongás szimulálása, a kódban csak a lépésszám változtatható, # amit külön jelöltünk
randomwalk <- function(steps) {
  # Kezdeti pozíció
  position <- c(0, 0)

  # Pozíciók tárolása
  positions <- matrix(NA, nrow = steps + 1, ncol = 2)
  positions[1, ] <- position

  # Bolyongás lépései
  for (i in 1:steps) {
    direction <- sample(c("up", "down", "left", "right"), 1)

    if (direction == "up") {
      position[2] <- position[2] + 1
    } else if (direction == "down") {
      position[2] <- position[2] - 1
    } else if (direction == "left") {
      position[1] <- position[1] - 1
    } else if (direction == "right") {
      position[1] <- position[1] + 1
    }
  }
}
```

```

} else if (direction == "right") {
  position[1] <- position[1] + 1
}

positions[i + 1, ] <- position
}

return(positions)
}

# Véletlen bolyongás szimulálása 100 lépésig
steps <- 100 # változtatható
walk <- randomwalk(steps)

# Adatok átalakítása adatkeretbe
walkdf <- data.frame(x = walk[, 1], y = walk[, 2])

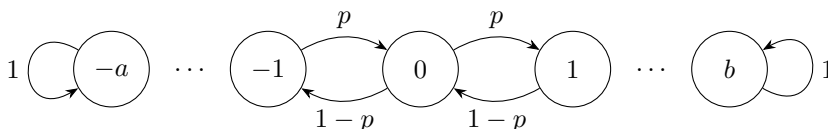
# Ábra készítése
ggplot(walkdf, aes(x, y)) +
  geom_path(color = "blue") +
  geom_point(color = "red", size = 2) +
  geom_point(data = data.frame(x = 0, y = 0), aes(x, y),
    color = "red", shape = 4, size = 3) + # Piros X a (0, 0) ponton
  theme_minimal()

```

3.2. A játékos csődje

A probléma [2]-ben is megtalálható. Egy szerencsejátékosnak kezdetben van a forintja, és addig játszik, míg vagy nyer b forintot, vagy elveszti az a forintját és így csődbe jut. Minden játékban $p \in (0, 1)$ valószínűséggel nyer 1 forintot, és $1 - p$ valószínűséggel veszít 1 forintot. Arra a kérdésre keressük a választ, hogy mekkora a csődbejutás valószínűsége.

Megoldás. Tegyük fel, hogy a a kezdőtőkénk, és ellenfelünknek b egységnyi pénze van. Legyen X_n az n -edik körbeli nyereményünk mennyisége. Ekkor $X_0 = 0$ kezdeti állapottal $\mathbb{X}$ egy véletlen bolyongás elnyelő falakkal, $I = \{-a, -a + 1, \dots, b - 1, b\}$ állapottérrel, mindezt a 7. ábra szemlélteti.



7. ábra. A játékos csődje probléma szemléltetése.

A kérdésre pedig a válasz éppen a $p_0^{\{-a\}}$ elnyelési valószínűség. Legyen $m = a +$

b , valamint jelölje $p_i = p_i(m - i, p)$ a játékos csődbe jutásának valószínűségét, ha kezdetben i forintja van. A $p_a(b, p)$ értéket szeretnénk meghatározni. Jelölje C_i azt az eseményt, hogy a játékos csődbe megy i forinttal indulva. Ekkor

$$\begin{aligned} p_i &= \mathbf{P}(C_i) = \mathbf{P}(C_i | \text{az 1. játékot megnyeri}) \mathbf{P}(\text{az 1. játékot megnyeri}) \\ &\quad + \mathbf{P}(C_i | \text{az 1. játékot elveszti}) \mathbf{P}(\text{az 1. játékot elveszti}) \\ &= p_{i+1}p + p_{i-1}(1 - p). \end{aligned}$$

Vagyis a folyamatnak az alábbi átmenevalószínűségei vannak

$$p_i^{\{-a\}} = \begin{cases} 1, & \text{ha } i = -a, \\ 0, & \text{ha } i = b, \\ p_{i+1}^{\{-a\}}p + p_{i-1}^{\{-a\}}(1 - p), & \text{ha } i \in \{-a + 1, \dots, b - 1\}. \end{cases}$$

Felhasználva a $q = 1 - p$, valamint $r = p/q$ jelöléseket, az előző egyenletből kapjuk, hogy

$$p_{i-1} - p_i = r(p_i - p_{i+1}).$$

Ezután a $p_m = 0$ feltételt felhasználva

$$\begin{aligned} p_{m-1} &= p_{m-1} \\ p_{m-2} - p_{m-1} &= r(p_{m-1} - p_m) = rp_{m-1} \\ p_{m-3} - p_{m-2} &= r(p_{m-2} - p_{m-1}) = r^2p_{m-1} \\ &\vdots \\ p_1 - p_2 &= r(p_2 - p_3) = r^{m-2}p_{m-1} \\ p_0 - p_1 &= r(p_1 - p_2) = r^{m-1}p_{m-1} \end{aligned}$$

egyenletrendszer adódik. Összeadás után kapjuk, hogy

$$1 = p_0 = (1 + r + r^2 + \dots + r^{m-1})p_{m-1} = \begin{cases} mp_{m-1}, & \text{ha } r = 1, \\ \frac{1-r^m}{1-r}p_{m-1}, & \text{ha } r \neq 1, \end{cases}$$

és így

$$p_{m-1} = \begin{cases} \frac{1}{m} = \frac{1}{a+b}, & \text{ha } p = \frac{1}{2}, \\ \frac{1-r}{1-r^m} = \frac{1-r}{1-r^{a+b}}, & \text{ha } p \neq \frac{1}{2}. \end{cases}$$

A fenti egyenletrendszerben az első két egyenlőséget összeadva $p_{m-2} = (1 + r)p_{m-1}$ adódik, az első hármat összeadva $p_{m-3} = (1 + r + r^2)p_{m-1}$ adódik, ... végül kapjuk, hogy $p(a, b) = p_a = p_{m-b} = (1 + r + r^2 + \dots + r^{b-1})p_{m-1}$. A mértani ($r \neq 1$), vagy a konstans 1 ($r = 1$) sorozat összegzés után kapjuk, hogy

$$p(a, b) = p_0^{\{-a\}} = \begin{cases} \frac{b}{a+b}, & \text{ha } p = \frac{1}{2}, \\ \frac{1-r^b}{1-r^{a+b}}, & \text{ha } p \neq \frac{1}{2}. \end{cases} \quad (1)$$

Jelölje q_i a játékos nyerésének valószínűségét. Erre gondolhatunk úgy is, mint a kaszinó csődbe jutásának valószínűsége, ezért az előző formula szerint

$$q(a, b) = p_0^{\{b\}} = \begin{cases} \frac{a}{a+b}, & \text{ha } p = \frac{1}{2}, \\ \frac{1 - \left(\frac{1}{r}\right)^a}{1 - \left(\frac{1}{r}\right)^{a+b}} = \frac{r^b - r^{a+b}}{1 - r^{a+b}}, & \text{ha } p \neq \frac{1}{2}. \end{cases} \quad (2)$$

A csőd, és a siker valószínűségét összegezve $p(a, b) + q(a, b) = 1$, vagyis előbb-utóbb valaki 1 valószínűséggel csődbe jut. Hasonlóan a várható játékidő is kiszámolható az alábbi képlettel

$$\mathbf{E}(\text{játék hossza}) = h_0^{\{-a, b\}} = \begin{cases} ab, & \text{ha } p = \frac{1}{2}, \\ \frac{a}{1-2p} - \frac{a+b}{1-2p} \frac{1 - \left(\frac{q}{p}\right)^a}{1 - \left(\frac{q}{p}\right)^{a+b}}, & \text{ha } p \neq \frac{1}{2}. \end{cases} \quad (3)$$

3.3. Feladat. Tegyük fel, hogy összekuporgatunk 100 dollárt, és elmegyünk egy kaszinóba, melynek 100 dollár az alaptőkéje. A ruletten játszunk, és az a taktikánk, hogy mindig egy zsetont teszünk a piros színre, ezáltal forgatásonként $18/37$ és $19/37$ valószínűséggel nyerünk illetve veszünk egy zsetont. Mekkora valószínűséggel megyünk csődbe, ha 10 dolláros zsetonokkal játszunk? Mennyi az átlagos játékidő?

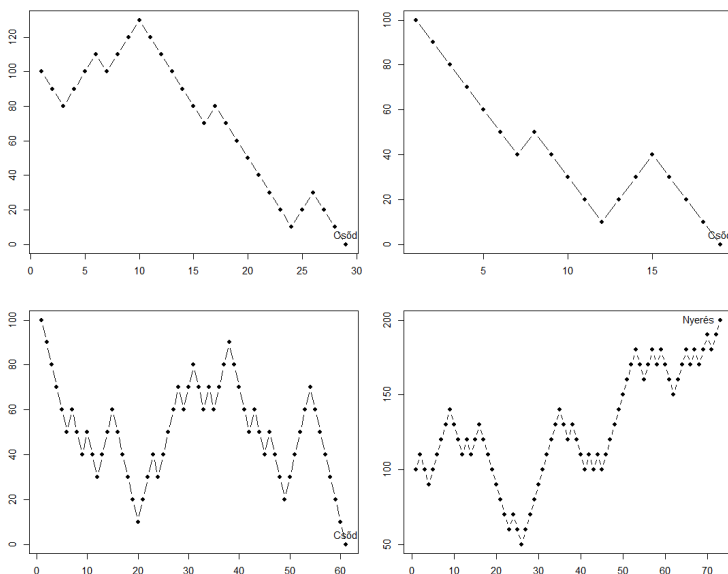
Megoldás. A megoldás előtt a problémát tudjuk szimulálni. A bevezetőben leírtak itt is szintén igazak, nagyon sok ideig tartana ezt a kísérletet sokszor megismételni. Ahogy azt a 8-as ábrán láthatjuk a folyamat a 100 állapotból indul ki, vagyis $X_0 = 100$, és akkor van vége, hogyha elértük a nullát (csődbe jutottunk), vagy elértük a 200-at (elnyertük a bank összes pénzét). A 4. táblázat a csőd relatív gyakoriságát, valamint az átlagos játékidőt tartalmazza.

szimulációk száma	10	100	1000	10.000
csőd relatív gyakorisága	0,7	0,61	0,621	0,627
átlagos játékidő	71,6	105,76	99,63	98,18

4. táblázat. A csőd relatív gyakorisága, valamint az átlagos játékidő a 3.3. Feladatban a szimulációk számának függvényében.

A fenti (1), (2), illetve (3) képleteket használva

$$p = \frac{18}{37}, \quad q = \frac{19}{37}, \quad a = b = 10,$$



8. ábra. A 3.3. Feladat négy szimulációja.

így

$$p_0^{\{10\}} = \frac{\left(\frac{18}{19}\right)^{10} - \left(\frac{18}{19}\right)^{20}}{1 - \left(\frac{18}{19}\right)^{20}} \approx 0,368,$$

$$p_0^{\{-10\}} = \frac{1 - \left(\frac{18}{19}\right)^{10}}{1 - \left(\frac{18}{19}\right)^{20}} \approx 0,632,$$

valamint

$$h_0^{\{-10,10\}} = \frac{10}{1 - 2 \cdot \frac{18}{37}} - \frac{20}{1 - 2 \cdot \frac{18}{37}} \cdot \frac{1 - \left(\frac{19}{18}\right)^{10}}{1 - \left(\frac{19}{18}\right)^{20}} \approx 97,65.$$

Látható, hogy a szimulált adatok, és a számolt értékek nagyfokú összhangot mutatnak.

Forráskód. # Függvény megadása, NEM változtatható
jatekoscsodjesimulacio <- function() {
penz <- 100

```

dobasszam <- 0
allapotok <- c(penz)

while (penz > 0 && penz < 200) {
  dobasszam <- dobasszam + 1
  erme <- sample(c(-10, 10), 1, replace = TRUE, prob = c(19/37, 18/37))
  penz <- penz + erme
  allapotok <- c(allapotok, penz)
}

return(allapotok)
}
# Függvény megadásának vége

# Szimulációk száma
szimulaciokszama <- 4 # változtatható, de ábrázolva mindig négy ábra
#lesz

# Szélessége és magassága beállítása
parwidth <- 7
parheight <- 4

#Elrendezés
par(mfrow = c(2, 2), mar = c(3, 3, 2, 1))

# Szimulálás, NEM változtatható
for (i in 1:szimulaciokszama) {
  allapotok <- jatekoscsodjesimulacio()
  plot(allapotok, type = "b", pch = 16, xlab = "", ylab = "",
  main = paste("Szimuláció", i))
  if (allapotok[length(allapotok)] >= 200) {
    text(length(allapotok), allapotok[length(allapotok)], "Nyerés",
    pos = 2)} else {
    text(length(allapotok), allapotok[length(allapotok)], "Csőd", pos = 3)
  }
}
# Szimulálás vége

# Relatív gyakoriságok kiszámítása
# Szimulációk száma
szimulaciokszama <- 10 # Változtatható

# Csődök és nyerések számlálója
csodszamlalo <- 0 # NEM változtatható
nyeresszamlalo <- 0 # NEM változtatható

set.seed(42)

```

```

for (i in 1:szimulaciokszama) {
  allapotok <- jatekoscsodjesimulacio()
  if (allapotok[length(allapotok)] >= 200) {
    nyeresszamlalo <- nyeresszamlalo + 1
  } else {
    csodszamlalo <- csodszamlalo + 1
  }
}

# Eredmények kiírása, NEM változtatható
cat("Csődök száma:", csodszamlalo)
cat("Nyerések száma:", nyeresszamlalo)

cat("Csőd relatív gyakoriság:", csodszamlalo/szimulaciokszama)

# Átlagos játékidő # NEM változtatható

jatekoscsodjesimulacio <- function() {
  penz <- 100
  dobasszam <- 0
  allapotok <- c(penz)
  while (penz > 0 && penz < 200) {
    dobasszam <- dobasszam + 1
    erme <- sample(c(-10, 10), 1, replace = TRUE, prob = c(19/37, 18/37))
    penz <- penz + erme
    allapotok <- c(allapotok, penz)
  }
  return(dobasszam)
}

# Szimulációk száma
szimulaciokszama <- 10 # Változtatható

# Dobásszámok tárolása
dobasszamok <- numeric(szimulaciokszama)

set.seed(42)

for (i in 1:szimulaciokszama) {
  dobasszamok[i] <- jatekoscsodjesimulacio()
}

# Várható végezési idő kiszámítása
atlagosvegezesiido <- mean(dobasszamok)

cat("Átlagos játékidő:", atlagosvegezesiido)

```

3.3. Galton–Watson-folyamat

Az alfejezetben előkerülő fogalmak, állítások megtalálhatóak [1]-ben. Tekintsük a $\{\xi_{n,i} : n, i \in \mathbb{N}\}$ független azonos eloszlású nemnegatív egész értékű véletlen változókat. Az $\mathbb{X} = \{X_n : n \in \mathbb{N}\}$ egy Galton–Watson folyamat, ha $X_0 = 1$, és

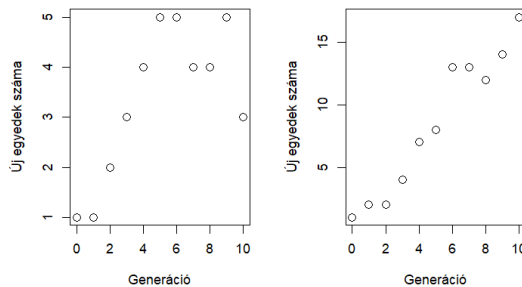
$$X_{n+1} = \sum_{i=1}^{X_n} \xi_{n,i}, \quad n \in \mathbb{N}. \quad (4)$$

A $\{\xi_{n,i} : n, i \in \mathbb{N}\}$ változók közös eloszlását utódeloszlásnak nevezzük, hiszen X_n reprezentálja a populáció n -edik generációjának méretét, és $\xi_{n,i}$ az n -edik generáció i -edik tagja utódainak számát. Így valóban az $(n+1)$ -edik generáció méretét megkapjuk a (4)-ben szereplő véletlen tagszámú összegként.

3.4. Feladat. Egy titkos szegedi szerencsétlen laborbaleset következtében egy halálos zombivírus terjed útjára a városban. Az elsőként megfertőződött laboráns indítja el a járványt. Egy zombi 0,4 - 0,4 valószínűséggel harap meg egy, illetve két embert, megfertőzve őket. A maradék 0,2 valószínűséggel kimeríti energiáit és végtelen álomba szunnyad.

- Várhatóan hány zombi lesz az n . generációban? Mekkora valószínűséggel fog a járvány megszűnni?
- Mi történik, ha kezdetben három laboránst fertőz meg a vírus?
- Attól függően, hogy hány ember fertőződik meg a kezdetekkor, elérhető-e az, hogy a járvány 1 valószínűséggel fennmaradjon?

Megoldás. A feladat kettő darab szimulációját a 9. ábra mutatja, ahol minden egyes generációban az új egyedek száma van ábrázolva, vagyis az X_n -ek. További kérdés még, hogy az n -edik generációban átlagosan hány darab zombi lesz? Ezt szintén tudjuk szimulálni, melyek eredményét a 5. táblázat mutatja a szimulációk számának függvényében, ahol $n = 10$.



9. ábra. A 3.4. Feladat kettő szimulációja.

szimulációk száma	10	100	1000	10.000	100.000
értékek átlaga	8,3	6,67	6,29	6,17	6,19

5. táblázat. A szimulált adatok átlaga a 3.4. Feladatban a szimulációk számának függvényében.

(a) A folyamat generátorfüggvénye

$$G_{\xi}(t) = 0,2 + 0,4t + 0,4t^2,$$

így

$$\mathbf{E}(\xi) = G'_{\xi}(1) = 1,2.$$

Vagyis az n . generációban várhatóan

$$\mathbf{E}(X_n) = (\mathbf{E}(\xi))^n = 1,2^n$$

zombi lesz, ami $n = 10$ esetén 6,19, mely összhangban van a szimulált adatokkal. A kihalási valószínűség a $q = G_{\xi}(q)$ egyenlet legkisebb nemnegatív megoldása, ami $q = 1/2$.

(b) Ebben az esetben lesz három egymástól független, azonos eloszlású Galton–Watson-folyamatunk, amik külön-külön $1/2$ valószínűséggel hálnak ki, így

$$\mathbf{E}(X_n) = 3 \cdot 1,2^n, \quad q = \frac{1}{8}.$$

(c) Hogyha N embert fertőz meg kezdetben a vírus, akkor a kihalás valószínűsége $(1/2)^N$, vagyis mindig pozitív valószínűséggel fog kihalni a folyamat, így nem érhető el, hogy 1 valószínűséggel fennmaradjon a járvány.

Forráskód. # A kezdeti populáció mérete, NEM változtatható
kezdetipopulacio <- 1

Az utódok számára a valószínűségek, NEM változtatható
valoszinusegek <- c(0.2, 0.4, 0.4)

A generációk száma
generaciok <- 10

A folyamat generálása, NEM változtatható
populacio <- rep(kezdetipopulacio, generaciok + 1)

```
par(mfrow = c(1, 2))
for (k in 1:2) {
  for (i in 1:generaciok) {
```

```

if (populacio[i] == 0) {
  populacio[i + 1] <- 0
} else {
  utodok <- rep(0, populacio[i])
  for (j in 1:populacio[i]) {
    utodok[j] <- sample(c(0,1,2), 1, prob = valoszinusegek)
  }
  populacio[i + 1] <- sum(utodok)
}
}
plot(0:generaciok, populacio, xlab = "Generáció",
ylab="Új egyedek száma", cex=1.5, xlim = c(0, generaciok))
}

# Átlagok kiszámolása

szimulaciok <- 1000 # Változtatható
generaciok <- 10 # Változtatható
atlagok <- numeric(szimulaciok)

# Értékek átlagának kiszámítása, NEM változtatható
for (k in 1:szimulaciok) {
  populacio <- rep(kezdetipopulacio, generaciok + 1)
  for (i in 1:generaciok) {
    if (populacio[i] == 0) {
      populacio[i + 1] <- 0
    } else {
      utodok <- rep(0, populacio[i])
      for (j in 1:populacio[i]) {
        utodok[j] <- sample(c(0,1,2), 1, prob = valoszinusegek)
      }
      populacio[i + 1] <- sum(utodok)
    }
  }
  atlagok[k] <- populacio[generaciok + 1]
}
cat(mean(atlagok))
# Értékek átlagának kiszámítása vége

```

Köszönetnyilvánítás. Hálás köszönettel tartozom Máder Attilának az építő jellegű észrevételeiért, megjegyzéseiért.

Irodalomjegyzék

- [1] K. B. ATHREYA AND P. E. NEY, *Branching processes*, Springer Science & Business Media, 2012.

- [2] CSÖRGŐ SÁNDOR, *Fejezetek a valószínűségelméletből.*, Polygon, 2010.
- [3] MOLNÁR ANDRÁS ÉS MUHARI CSILLA, *Interaktív szemléltetés az oktatásban*, MultiMédia az Oktatásban, 2007.
- [4] JAMES R. NORRIS, *Markov chains*, Cambridge University Press, 1998.
- [5] J. NOVAK, Polya's random walk theorem, *The American Mathematical Monthly*, 121(8):711–716, 2014.
- [6] VIHAROS LÁSZLÓ, *A sztochasztika alapjai*, Polygon, 2010.

Szalai Máté SZTE, Bolyai Intézet, 6720 Szeged, Aradi vértanúk tere 1,
e-mail: szalaim@math.u-szeged.hu