

## Stochastische Prozesse

### Woche 6

#### Aufgabe 1 Tennis

In dieser Aufgabe wollen wir ein Tennisspiel durch eine Markov-Kette modellieren und mit Hilfe dieses Modells die Gewinnwahrscheinlichkeit untersuchen.

Der Verlauf eines einzelnen Tennisspiels zwischen Spielern A und B ist im folgenden Übergangsgraphen dargestellt:



Ein Spiel besteht in einer Abfolge von Punkten, wobei jeweils der selbe Spieler aufschlägt. Wir zählen hier die Punkte einfach als Anzahl von gewonnen Bällen und nicht wie üblich als 0-15-30-40 etc.

Der erste Spieler, der mindestens vier Bälle gewinnt und dabei einen Vorsprung von 2 Punkten hat, gewinnt das Spiel<sup>1</sup>.

Das Spiel beginnt im Zustand 1 (Punktestand 0-0). Mit Wahrscheinlichkeit  $p$  gewinnt Spieler A, mit Wahrscheinlichkeit  $1 - p$  Spieler B. Im ersten Fall ist der nächste Zustand 2 (1-0), im zweiten 3 (0-1). Die Wahrscheinlichkeit, dass Spieler A einen Punkt gewinnt, sei konstant  $p$  für das gesamte Spiel.

Sobald beide Spieler mindestens 2 Punkte haben, tritt ein *Einstand* (engl. *deuce*) auf. Gewinnt ein Spieler von diesem Zustand aus einen weiteren Punkt, so spricht man von einem *Vorteil* (engl. *advantage*) für diesen Spieler. Hat ein Spieler einen Vorteil, und gewinnt den nächsten Punkt, so hat er das Spiel gewonnen. Von einem Einstand aus müssen also zwei Punkte in Folge gewonnen werden, um das gesamte Spiel zu gewinnen.

Die folgende Funktion generiert die

```
# <U+00DC>bergangsmatrix

P <- function(p) {
  if (p < 0 | p > 1) {
    warning("p ist keine Wahrscheinlichkeit!")
  } else {

    P = matrix(0, nrow = 17, ncol = 17)
```

<sup>1</sup>Über die exakten Tennisregeln können Sie sich unter <http://de.wikipedia.org/wiki/Tennis#Z.C3.A4hlweise>.

```

diag(P[4:13, 7:16]) = rep(p, 10)
diag(P[4:13, 8:17]) = rep(1 - p, 10)

P[7, 10] = P[10, 14] = 0

P[14, 14] = P[17, 17] = 1 # absorbierende Zustände
P[7, 14] = P[15, 14] = P[16, 12] = P[1, 2] = P[2, 4] = P[3, 5] = p # A gewinnt
P[10, 17] = P[15, 12] = P[16, 17] = P[1, 3] = P[2, 5] = P[3, 6] = 1 -
    p # B gewinnt

P
}
}

```

a) Berechnen Sie  $P^2$ ,  $P^5$ ,  $P^{10}$ ,  $P^{100}$ ,  $P^{1000}$  für  $p = 0.3$ . Sehen Sie ein Muster?

```

library(expm)
round(P(0.3) %^% 2, digits = 4)

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]  0    0    0 0.09 0.42 0.49 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [2,]  0    0    0 0.00 0.00 0.00 0.09 0.42 0.49 0.00 0.00 0.00 0.00
## [3,]  0    0    0 0.00 0.00 0.00 0.00 0.09 0.42 0.49 0.00 0.00 0.00
## [4,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.42 0.49 0.00
## [5,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.09 0.42 0.49
## [6,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.09 0.42
## [7,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [8,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [9,]  0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [10,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [11,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.49 0.00
## [12,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.42 0.00
## [13,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.09 0.00
## [14,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [15,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [16,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [17,] 0    0    0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##      [,14] [,15] [,16] [,17]
## [1,] 0.00 0.00 0.00 0.00
## [2,] 0.00 0.00 0.00 0.00
## [3,] 0.00 0.00 0.00 0.00
## [4,] 0.09 0.00 0.00 0.00
## [5,] 0.00 0.00 0.00 0.00
## [6,] 0.00 0.00 0.00 0.49
## [7,] 0.51 0.49 0.00 0.00
## [8,] 0.09 0.42 0.49 0.00
## [9,] 0.00 0.09 0.42 0.49
## [10,] 0.00 0.00 0.09 0.91
## [11,] 0.51 0.00 0.00 0.00
## [12,] 0.09 0.00 0.00 0.49
## [13,] 0.00 0.00 0.00 0.91
## [14,] 1.00 0.00 0.00 0.00

```

```
## [15,] 0.30 0.21 0.49 0.00
## [16,] 0.00 0.09 0.21 0.70
## [17,] 0.00 0.00 0.00 1.00
```

```
round(P(0.3) %~% 5, digits = 4)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [2,] 0    0    0    0    0    0    0    0    0    0    0    0 0.3087
## [3,] 0    0    0    0    0    0    0    0    0    0    0    0 0.1323
## [4,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [5,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [6,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [7,] 0    0    0    0    0    0    0    0    0    0    0    0 0.1441
## [8,] 0    0    0    0    0    0    0    0    0    0    0    0 0.1852
## [9,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0794
## [10,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0113
## [11,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [12,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [13,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [14,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [15,] 0    0    0    0    0    0    0    0    0    0    0    0 0.1235
## [16,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0529
## [17,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
##      [,13] [,14] [,15] [,16] [,17]
## [1,] 0 0.0308 0.1323 0.3087 0.5282
## [2,] 0 0.1631 0.0000 0.0000 0.5282
## [3,] 0 0.0308 0.0000 0.0000 0.8369
## [4,] 0 0.3483 0.1235 0.2881 0.2401
## [5,] 0 0.0837 0.0794 0.1852 0.6517
## [6,] 0 0.0081 0.0227 0.0529 0.9163
## [7,] 0 0.6879 0.0000 0.0000 0.1681
## [8,] 0 0.2557 0.0000 0.0000 0.5591
## [9,] 0 0.0440 0.0000 0.0000 0.8766
## [10,] 0 0.0024 0.0000 0.0000 0.9862
## [11,] 0 0.5541 0.0617 0.1441 0.2401
## [12,] 0 0.1278 0.0529 0.1235 0.6958
## [13,] 0 0.0081 0.0113 0.0265 0.9541
## [14,] 0 1.0000 0.0000 0.0000 0.0000
## [15,] 0 0.3895 0.0000 0.0000 0.4871
## [16,] 0 0.0383 0.0000 0.0000 0.9087
## [17,] 0 0.0000 0.0000 0.0000 1.0000
```

```
round(P(0.3) %~% 10, digits = 4)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0327
## [2,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [3,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0000
## [4,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0305
## [5,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0196
## [6,] 0    0    0    0    0    0    0    0    0    0    0    0 0.0056
```

```

## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0152
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0131
## [13,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0028
## [14,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [15,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [16,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
## [17,] 0 0 0 0 0 0 0 0 0 0 0 0 0.0000
##      [,13] [,14] [,15] [,16] [,17]
## [1,] 0 0.0941 0.0000 0.0000 0.8732
## [2,] 0 0.2025 0.0163 0.0381 0.7430
## [3,] 0 0.0477 0.0070 0.0163 0.9290
## [4,] 0 0.4074 0.0000 0.0000 0.5621
## [5,] 0 0.1217 0.0000 0.0000 0.8587
## [6,] 0 0.0190 0.0000 0.0000 0.9754
## [7,] 0 0.7063 0.0076 0.0178 0.2683
## [8,] 0 0.2794 0.0098 0.0229 0.6880
## [9,] 0 0.0542 0.0042 0.0098 0.9318
## [10,] 0 0.0039 0.0006 0.0014 0.9941
## [11,] 0 0.5837 0.0000 0.0000 0.4011
## [12,] 0 0.1531 0.0000 0.0000 0.8338
## [13,] 0 0.0135 0.0000 0.0000 0.9837
## [14,] 0 1.0000 0.0000 0.0000 0.0000
## [15,] 0 0.4052 0.0065 0.0152 0.5730
## [16,] 0 0.0451 0.0028 0.0065 0.9456
## [17,] 0 0.0000 0.0000 0.0000 1.0000

```

```
round(P(0.3) %~% 1000, digits = 4)
```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [13,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [14,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [15,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [16,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [17,] 0 0 0 0 0 0 0 0 0 0 0 0 0
##      [,14] [,15] [,16] [,17]
## [1,] 0.0992 0 0 0.9008
## [2,] 0.2110 0 0 0.7890

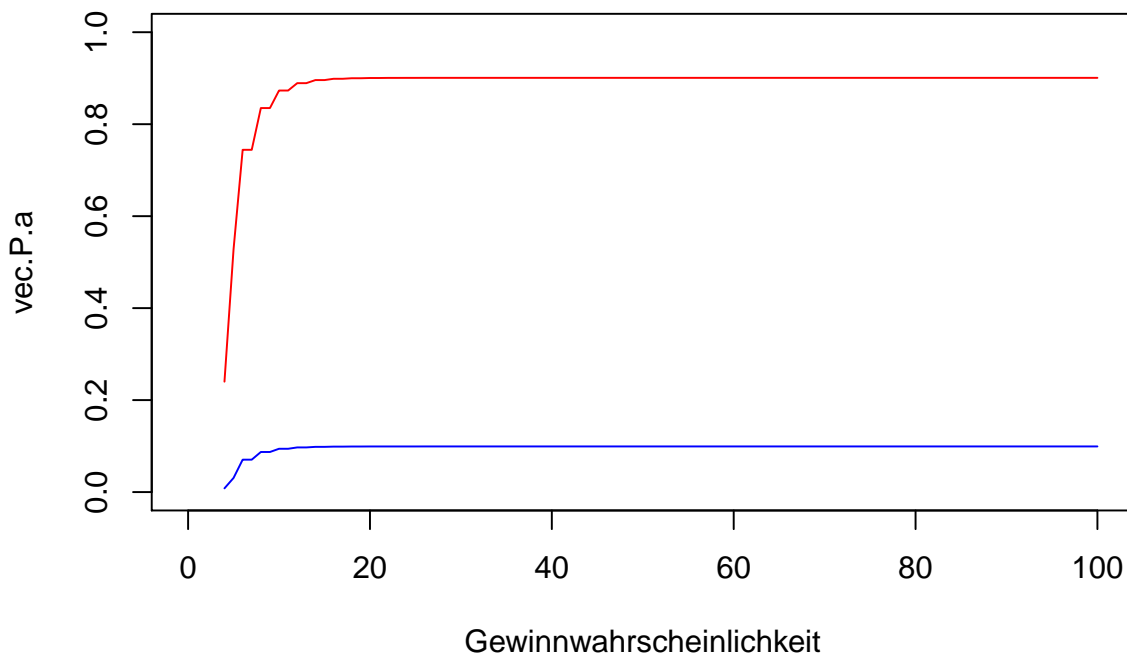
```

```
## [3,] 0.0513      0      0 0.9487
## [4,] 0.4122      0      0 0.5878
## [5,] 0.1248      0      0 0.8752
## [6,] 0.0198      0      0 0.9802
## [7,] 0.7102      0      0 0.2898
## [8,] 0.2844      0      0 0.7156
## [9,] 0.0563      0      0 0.9437
## [10,] 0.0042     0      0 0.9958
## [11,] 0.5860      0      0 0.4140
## [12,] 0.1552      0      0 0.8448
## [13,] 0.0140      0      0 0.9860
## [14,] 1.0000      0      0 0.0000
## [15,] 0.4086      0      0 0.5914
## [16,] 0.0466      0      0 0.9534
## [17,] 0.0000      0      0 1.0000
```

- b) Plotten Sie für A und B jeweils die Wahrscheinlichkeiten, nach  $t = 4, \dots, 100$  Bällen das Spiel gewonnen zu haben. Nehmen Sie  $p = 0.3$  an.

```
t1 <- 4:100
vec.P.a <- sapply(t1, function(z) {
  (P(0.3) %>% z)[1, 14]
})
vec.P.b <- sapply(t1, function(z) {
  (P(0.3) %>% z)[1, 17]
})

plot(t1, vec.P.a, xlim = c(0, 100), ylim = c(0, 1), col = "blue", type = "l",
     xlab = "Gewinnwahrscheinlichkeit") # A gewinnt
lines(t1, vec.P.b, col = "red") # B gewinnt
```



- c) Wie lange dauert das Spiel im Mittel? Nehmen Sie an, dass das Spiel nach spätestens 100 Bällen beendet ist, und setzen Sie wieder  $p = 0.3$ .

```

occup.time <- function(p, tmax) {

  t1 <- as.list(0:tmax)

  vec.P <- lapply(t1, function(z) {
    (P(p) %^% z)
  })

  # Aufenthaltsmatrix
  M <- Reduce("+", vec.P)

  # Anzahl Aufenthalte in allen ausser den absorbierenden Zuständen:
  sum(M[1, -c(14, 17)])

}

occup.time(0.3, tmax = 100)

```

```
## [1] 5.83149
```

```

# Bemerkung mit First Passage Time Der Formalismus, wie im Skript dargelegt
# ist nicht anwendbar, da wir keine irreduziblen Kette haben. Ohne Beweis
# geht folgendes
pj <- P(0.3)
pj[, 14] <- rep(0, 17)
pj[, 17] <- rep(0, 17)
solve(diag(17) - pj) %*% rep(1, 17) # 5.831490

```

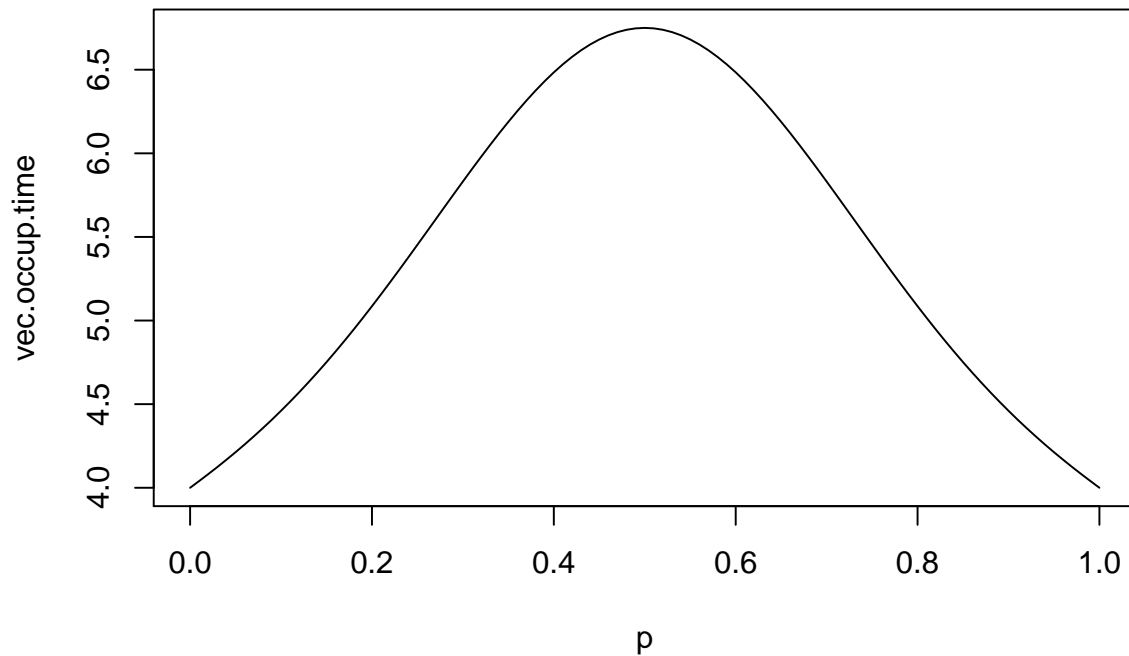
```

##           [,1]
## [1,] 5.831490
## [2,] 5.713683
## [3,] 4.453407
## [4,] 5.113310
## [5,] 4.542414
## [6,] 2.986690
## [7,] 3.372759
## [8,] 4.430690
## [9,] 3.161724
## [10,] 1.483103
## [11,] 3.389655
## [12,] 3.448276
## [13,] 1.610345
## [14,] 1.000000
## [15,] 3.413793
## [16,] 2.034483
## [17,] 1.000000

```

- c) Wie hängt die erwartete Dauer des Spiels von der Wahrscheinlichkeit  $p$ , mit der A einen Ball gewinnt, ab? Plotten Sie die erwartete Dauer in Abhängigkeit von  $p$ . Wann ist die erwartete Dauer am längsten? Hätten Sie dieses Ergebnis erwartet? Berechnen Sie die erwarteten Dauern wie in (c).

```
p <- seq(0, 1, 0.01)
vec.occup.time <- sapply(p, occup.time, tmax = 100)
plot(p, vec.occup.time, type = "l")
```



d) Berechnen Sie die stationären Verteilungen der Markov-Kette (wieder für  $p = 0.3$ ).

Stationäre Verteilungen sind linke Eigenvektoren von  $P$  zum Eigenwert 1, d.h.  $\vec{\pi}P = \vec{\pi}$ . Wir berechnen mit R die rechten Eigenvektoren von  $P'$ :

```
eig.P <- eigen(t(P(0.3))) # eigenvalue 1 has alg. & geom. multiplicity 2
eig.P
```

```
## $values
## [1] 1.000000e+00 1.000000e+00 -6.480741e-01 6.480741e-01 -4.336809e-18
## [6] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [11] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [16] 0.000000e+00 0.000000e+00
##
## $vectors
##      [,1] [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [2,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [3,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [4,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [5,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [6,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [7,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [8,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [9,]  0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
## [10,] 0    0 0.00000000 0.00000000 0.000000e+00 0.000000e+00
```

```

## [11,] 0 0 0.00000000 0.0000000 0.000000e+00 0.000000e+00
## [12,] 0 0 -0.62034997 -0.3739161 -1.581238e-16 1.314307e-16
## [13,] 0 0 0.00000000 0.0000000 0.000000e+00 2.223870e-17
## [14,] 0 1 -0.05227306 0.1475505 -9.889513e-02 9.889513e-02
## [15,] 0 0 0.28716624 -0.1730895 3.296504e-01 -3.296504e-01
## [16,] 0 0 0.67005455 -0.4038756 -7.691843e-01 7.691843e-01
## [17,] 1 0 -0.28459776 0.8033307 5.384290e-01 -5.384290e-01
##      [,7]      [,8]      [,9]     [,10]
## [1,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [6,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 9.940562e-309 0.000000e+00 -4.970281e-309
## [8,] 0.000000e+00 0.000000e+00 2.319465e-308 -1.159732e-308
## [9,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [10,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [11,] 4.084660e-18 -4.084660e-18 -4.084660e-18 4.084660e-18
## [12,] -1.653879e-16 1.653879e-16 1.502426e-16 -1.502426e-16
## [13,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [14,] -9.889513e-02 9.889513e-02 9.889513e-02 -9.889513e-02
## [15,] 3.296504e-01 -3.296504e-01 -3.296504e-01 3.296504e-01
## [16,] -7.691843e-01 7.691843e-01 7.691843e-01 -7.691843e-01
## [17,] 5.384290e-01 -5.384290e-01 -5.384290e-01 5.384290e-01
##      [,11]     [,12]     [,13]     [,14]
## [1,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [6,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [8,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [9,] 5.412084e-308 0.000000e+00 5.412084e-308 0.000000e+00
## [10,] 0.000000e+00 0.000000e+00 0.000000e+00 1.262820e-307
## [11,] 0.000000e+00 4.084660e-18 0.000000e+00 0.000000e+00
## [12,] -1.561223e-16 -1.653879e-16 -1.561223e-16 -1.561223e-16
## [13,] -2.223870e-17 0.000000e+00 -2.223870e-17 -2.223870e-17
## [14,] -9.889513e-02 -9.889513e-02 -9.889513e-02 -9.889513e-02
## [15,] 3.296504e-01 3.296504e-01 3.296504e-01 3.296504e-01
## [16,] -7.691843e-01 -7.691843e-01 -7.691843e-01 -7.691843e-01
## [17,] 5.384290e-01 5.384290e-01 5.384290e-01 5.384290e-01
##      [,15]     [,16]     [,17]
## [1,] 0.000000e+00 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 0.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 0.000000e+00
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00
## [6,] 0.000000e+00 0.000000e+00 0.000000e+00
## [7,] 0.000000e+00 0.000000e+00 0.000000e+00
## [8,] 0.000000e+00 0.000000e+00 0.000000e+00
## [9,] 0.000000e+00 0.000000e+00 0.000000e+00
## [10,] 0.000000e+00 1.262820e-307 0.000000e+00

```



```
## [11,] 0.000000e+00 0.000000e+00 0.000000e+00
## [12,] 1.314307e-16 -1.561223e-16 1.314307e-16
## [13,] 2.223870e-17 -2.223870e-17 2.223870e-17
## [14,] 9.889513e-02 -9.889513e-02 9.889513e-02
## [15,] -3.296504e-01 3.296504e-01 -3.296504e-01
## [16,] 7.691843e-01 -7.691843e-01 7.691843e-01
## [17,] -5.384290e-01 5.384290e-01 -5.384290e-01
```

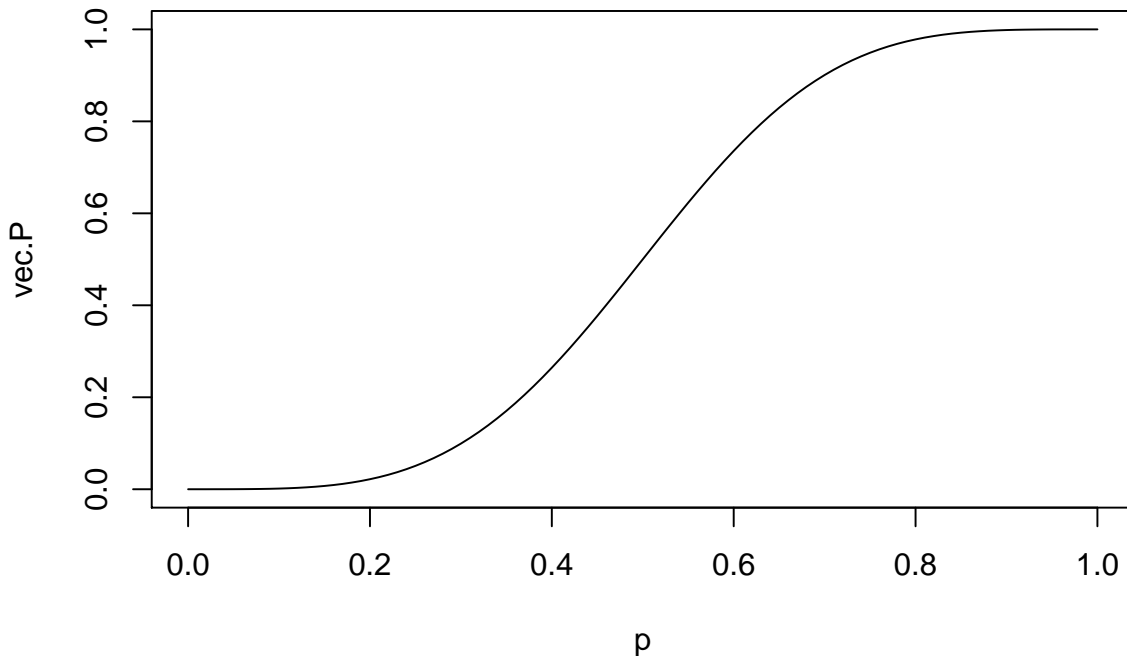
```
eig.P$vectors[, c(1, 2)]
```

```
##      [,1] [,2]
## [1,] 0    0
## [2,] 0    0
## [3,] 0    0
## [4,] 0    0
## [5,] 0    0
## [6,] 0    0
## [7,] 0    0
## [8,] 0    0
## [9,] 0    0
## [10,] 0    0
## [11,] 0    0
## [12,] 0    0
## [13,] 0    0
## [14,] 0    1
## [15,] 0    0
## [16,] 0    0
## [17,] 1    0
```

Jede stationäre Verteilung verteilt die Gesamtwahrscheinlichkeit auf die absorbierenden Zustände 14 und 17. Es gibt allerdings mehrere stationäre Verteilungen.

- e) Nun bestimmen Sie die Wahrscheinlichkeit, dass Spieler A das Spiel gewinnt in Abhängigkeit von der Wahrscheinlichkeit, mit der er einen einzelnen Ball gewinnt  $p$ . Da Sie hier nicht über stationäre Verteilungen argumentieren können, betrachten Sie einen hinreichend langen Zeithorizont (z.B. bis  $T = 1000$ ), um eine gute Näherung der Grenzverteilung zu bekommen.

```
tmax <- 1000
p <- seq(0, 1, 0.01)
vec.P <- sapply(p, function(p) {
  (P(p) %^% tmax)[1, 14]
})
plot(p, vec.P, type = "l")
```



## Aufgabe 2 Skilift

Ein Skilift in den Alpen modelliert die mittlere Tagestemperatur  $X_k (k = 0, 1, 2, 3 \dots)$  als Markov-Kette. Der Zustandsraum sei gegeben durch die Zustände 1: *Temperatur*  $< 0^\circ\text{C}$ , 2: *Temperatur*  $= 0^\circ\text{C}$ , 3: *Temperatur*  $> 0^\circ\text{C}$ . Die Übergangsmatrix sei:

$$\mathbf{P} = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.6 & 0.2 & 0.2 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

```
P <- matrix(c(0.8, 0.1, 0.1, 0.6, 0.2, 0.2, 0.7, 0.2, 0.1), ncol=3, byrow=TRUE)
```

Der Skilift ist in Betrieb, wenn *Temperatur*  $= 0$  oder *Temperatur*  $< 0$ . Bei *Temperatur*  $> 0$  muss der Lift abgeschaltet werden. Ein Betriebstag bringt einen Erlös aus Kartenverkauf von 10000 Fr. Die Personalkosten sind 6000 Fr./Tag, unabhängig davon, ob der Lift läuft oder nicht. Die Energiekosten sind 500 Fr., wenn der Lift läuft, sonst sind sie vernachlässigbar.

Nehmen Sie nun an, der Skilift laufe nur am Wochenende (Samstag und Sonntag). Jeweils Freitags wird entschieden, ob das Personal für das kommende Wochenende angefordert wird oder nicht. Wenn Personal angefordert wird, laufen die Personalkosten von 6000 Fr./Tag auf, wenn nicht, entstehen keine Kosten. Der Lift kann nur laufen, wenn Personal da ist. Das Management hat entschieden, kein Personal anzufordern, wenn am Freitag *Temperatur*  $> 0$  ist.

*Hinweis:* Beachten Sie bei Ihren Berechnungen, dass sowohl Kosten als auch Gewinn nur am Samstag und Sonntag, nicht aber am Freitag entstehen.

- a) Ist die Entscheidung des Managements sinnvoll? Berechnen Sie zur Beantwortung dieser Frage den erwarteten Gewinn/Verlust unter der Annahme, dass am Freitag *Temperatur*  $> 0$  ist.

Mit dem dem Kostenvektor  $c = (10000 - 6000 - 500, 10000 - 6000 - 500, -6000) = (3500, 3500, -6000)$  kann man den erwarteten Gewinn  $E(K)$  wie folgt berechnen (positive Zahlen sind Gewinn):

$$E(K) = \pi(0) \left( \sum_{i=1}^T P^i \right) c' =: \pi(0) \tilde{M}(T) c'$$

mit

$$\tilde{M}(T) = \sum_{i=1}^T P^i$$

In der Summe fällt  $i = 0$  weg, denn am Freitag fallen keine Kosten oder Gewinne an!

```
M = P + P %*% P
cv = c(3500, 3500, -6000)
K = c(0,0,1) %*% M %*% cv
K
##           [,1]
## [1,] 4910
```

Der Lift macht im Mittel Gewinn, auch wenn es am Freitag zu warm ist. Die Entscheidung des Managements ist also nicht sinnvoll.

- b) Wie hoch ist der erwartete Gewinn pro Wochenende beim obigen Verhalten des Managements, wenn folgendes gilt: Die Wahrscheinlichkeit, dass am Freitag *Temperatur* < 0 ist, ist 50%. Die Wahrscheinlichkeit, dass *Temperatur* = 0 ist, ist 10%.

```
K1 <- c(1,0.0,0.0) %*% M %*% cv # Erwarteter Gewinn, falls Temp<0
K1
##           [,1]
## [1,] 5005

K2 <- c(0,1.0,0.0) %*% M %*% cv # Erwarteter Gewinn, falls Temp=0
K2
##           [,1]
## [1,] 3960

K3 <- 0 # Erwarteter Gewinn, falls Temp>0 (Keine Kosten. keine Einnahmen)
```

Damit erhält man den erwarteten Gewinn pro Wochenende:

```
0.5 * K1 + 0.1 * K2 + 0.4 * K3
##           [,1]
## [1,] 2898.5
```

Bemerkung zur Lösung: Falls man am Freitag im Zustand 3, d.h. Temperatur 0 ist, stellt man den Lift nicht an. In diesem Fall fallen keine Kosten (aber auch keine Einnahmen) an.