

Symulacja działalności firmy ubezpieczeniowej

A.

liczba klientów: N wysokość składki: K kwota ubezpieczenia: F liczba miesięcy: T

1. Przyjmij:  $t = 1$
2. Wyznacz rezerwę na wypłaty: St
3. Wyznacz liczbę wypłat: a
4. Wypłać odszkodowanie: St
5. Sprawdź płynność  $St \geq 0$
6. Jeśli spełnione  $\rightarrow$  zmodyfikuj liczbę ubezpieczonych do  $N = N + n - o - a$  - losowa liczba z przedziału od 0 do 100 nowych klientów o - losowa liczba z przedziału od 0 do 90 klientów rezygnujących
7. Jeżeli nie spełnione  $\rightarrow$  firma zbankrutowała. Zatrzymaj algorytm przed czasem.
8. Przyjmij  $t = t + 1$
9. Jeżeli  $t \leq T$ , to przejdź do 2, w przeciwnym przypadku KONIEC.

```
# Parametry symulacji
N <- 2000      # Liczba początkowych klientów
K <- 1000      # Składka płacona przez każdego klienta
F <- 10000     # Kwota do wypłaty w przypadku zdarzenia

T <- 12        # Liczba miesięcy symulacji

t <- 1         # 1. Numer miesiąca
St <- K * N    # 2. Rezerwa na wypłaty na początku symulacji gdy t=1

symulacja_ubezpieczenia <- function(K, N, F, T) {

  while (t <= T) {
    # 3. Wyznaczenie liczby wypłat a
    cn <- rt(1, df = 2) # Rozkład t-studenta z df = 2
    a <- sum(cn >= qt(0.9999, df = 2)) # Ilość wypłat
    # 4. Wypłacenie odszkodowania
    St <- St - a * F
    # 5. Sprawdzenie płynności
    # 5.1
    if (St >= 0) {
      # Zmodyfikowanie liczby ubezpieczonych
      n <- sample(0:100, 1) # Losowa liczba nowych klientów
      o <- sample(0:90, 1)  # Losowa liczba klientów rezygnujących
      N <- N + n - o - a    # Aktualizacja liczby ubezpieczonych

      # Wyświetlenie informacji
      cat("Miesiąc:", t, "\n")
      cat("Liczba nowych klientów:", n, "\n")
      cat("Liczba rezygnujących klientów:", o, "\n")
      cat("Liczba wypłat:", a, "\n")
      cat("Rezerwa po wypłatach:", St, "\n")
      cat("Liczba ubezpieczonych:", N, "\n\n")
    }
    # 5.2
  }
}
```

```

else {
  # Firma zbankrutowała
  cat("Firma zbankrutowała w miesiącu", t, ". Koniec symulacji.\n")
  break
}
# 6. Przejście do kolejnego miesiąca
t <- t + 1
# Aktualizacja rezerwy na wypłaty na początku kolejnego miesiąca
St <- St + K * N
}
if (t > T) {
  cat("Symulacja zakończona po", T, "miesiącach.\n")
}
}

symulacja_ubezpieczenia(K, N, F, T)

```

```

## Miesiąc: 1
## Liczba nowych klientów: 38
## Liczba rezygnujących klientów: 2
## Liczba wypłat: 0
## Rezerwa po wypłatach: 2e+06
## Liczba ubezpieczonych: 2036
##
## Miesiąc: 2
## Liczba nowych klientów: 86
## Liczba rezygnujących klientów: 77
## Liczba wypłat: 0
## Rezerwa po wypłatach: 4036000
## Liczba ubezpieczonych: 2045
##
## Miesiąc: 3
## Liczba nowych klientów: 22
## Liczba rezygnujących klientów: 77
## Liczba wypłat: 0
## Rezerwa po wypłatach: 6081000
## Liczba ubezpieczonych: 1990
##
## Miesiąc: 4
## Liczba nowych klientów: 35
## Liczba rezygnujących klientów: 66
## Liczba wypłat: 0
## Rezerwa po wypłatach: 8071000
## Liczba ubezpieczonych: 1959
##
## Miesiąc: 5
## Liczba nowych klientów: 99
## Liczba rezygnujących klientów: 70
## Liczba wypłat: 0
## Rezerwa po wypłatach: 10030000
## Liczba ubezpieczonych: 1988
##
## Miesiąc: 6
## Liczba nowych klientów: 47

```

```

## Liczba rezygnujących klientów: 45
## Liczba wypłat: 0
## Rezerwa po wypłatach: 12018000
## Liczba ubezpieczonych: 1990
##
## Miesiąc: 7
## Liczba nowych klientów: 63
## Liczba rezygnujących klientów: 0
## Liczba wypłat: 0
## Rezerwa po wypłatach: 14008000
## Liczba ubezpieczonych: 2053
##
## Miesiąc: 8
## Liczba nowych klientów: 63
## Liczba rezygnujących klientów: 5
## Liczba wypłat: 0
## Rezerwa po wypłatach: 16061000
## Liczba ubezpieczonych: 2111
##
## Miesiąc: 9
## Liczba nowych klientów: 73
## Liczba rezygnujących klientów: 27
## Liczba wypłat: 0
## Rezerwa po wypłatach: 18172000
## Liczba ubezpieczonych: 2157
##
## Miesiąc: 10
## Liczba nowych klientów: 12
## Liczba rezygnujących klientów: 53
## Liczba wypłat: 0
## Rezerwa po wypłatach: 20329000
## Liczba ubezpieczonych: 2116
##
## Miesiąc: 11
## Liczba nowych klientów: 50
## Liczba rezygnujących klientów: 68
## Liczba wypłat: 0
## Rezerwa po wypłatach: 22445000
## Liczba ubezpieczonych: 2098
##
## Miesiąc: 12
## Liczba nowych klientów: 51
## Liczba rezygnujących klientów: 88
## Liczba wypłat: 0
## Rezerwa po wypłatach: 24543000
## Liczba ubezpieczonych: 2061
##
## Symulacja zakończona po 12 miesiącach.

```

B.

1. Funkcja przeprowadzająca zaprojektowaną symulację:
2. o argumentach K, N, F, T z wartościami domyślnymi
3. zwracająca wektor St, długości T jako wynik, jeśli firma zbankrutowała w wektorze powinny od tego momentu znajdować się wartości NA

```
symulacja_ubezpieczenia <- function(K=1000, N=2000, F=1000, T=12) {
```

```
  St <- numeric(T)
  St[1] <- K * N
  for (t in 2:T) {
    St[t] <- St[t - 1] + K * N
    cn <- rt(N, df = 2)
    a <- sum(cn >= qt(0.9999, df = 2))
    St[t] <- St[t] - a * F
    if (St[t] < 0) {
      St[t] <- NA
      if (t < T) {
        St[(t+1):T] <- NA
      }
      break
    }
    n <- sample(0:100, 1)
    o <- sample(0:90, 1)
    N <- N + n - o - a
    if (N < 0) {
      N <- 0
    }
  }
}
```

```
return(St)
```

```
}
```

```
symulacja <- symulacja_ubezpieczenia(K, N, F, T)
```

```
print(symulacja)
```

```
## [1] 2000000 4000000 5994000 8007000 10021000 12091000 14091000 16104000
```

```
## [9] 18087000 20072000 22138000 24152000
```

2. Symulacja ubezpieczenia wykonana M razy. Wyniki w macierzy SIM.

```
parametry <- list(
  list(K = 10, N = 200, F = 100, T = 12),
  list(K = 16, N = 270, F = 50, T = 12),
  list(K = 10, N = 180, F = 70, T = 12),
  list(K = 53, N = 16, F = 87, T = 12)
)
```

```
M = length(parametry)
```

```
wyniki <- matrix(NA, nrow = length(parametry), ncol = T)
```

```
for (i in 1:M) {
  paramenty_i <- parametry[[i]]
  wyniki[i, ] <- symulacja_ubezpieczenia(paramenty_i$K, paramenty_i$N, paramenty_i$F, paramenty_i$T)
}
```

```
print(wyniki)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 2000 4000 5800 7520 9740 12460 15370 18530 20920 23670 26670 30130
## [2,] 4320 8640 13024 17984 22304 25744 29936 33872 37952 41328 44880 49104
## [3,] 1800 3600 5400 6800 8240 10600 13010 15810 18540 20760 22980 24770
## [4,] 848 1696 5618 12985 16854 21571 28037 33602 37789 41711 44891 48336
```

3. Prawdopodobieństwo tego, że spółka nie zbankrutuje do chwili  $t=1,2,\dots,T$ :

```
wyniki_s = wyniki

prob_S <- function(wyniki_s) {
  T <- ncol(wyniki_s)
  N <- nrow(wyniki_s)
  prawdopodobienstwo <- numeric(T)

  for (t in 1:T) {
    prawdopodobienstwo[t] <- sum(!is.na(wyniki_s[, t])) / N
  }
  return(prawdopodobienstwo)
}
wynik_prob <- prob_S(wyniki_s)
print(wynik_prob)
```

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

Średni poziom rezerw spółki pod warunkiem, że nie zbankrutuje do chwili  $t=1,2,\dots,T$ :

```
sredni_poziom_rezerw <- function(wyniki) {
  T <- ncol(wyniki)
  srednie_rezerwy <- numeric(T)

  for (t in 1:T) {
    rezerwy <- wyniki[!is.na(wyniki[, t]), t]
    srednie_rezerwy[t] <- mean(rezerwy, na.rm = TRUE)
  }
  return(srednie_rezerwy)
}
srednie_rezerwy <- sredni_poziom_rezerw(wyniki)
print(srednie_rezerwy)
```

```
## [1] 2242.00 4484.00 7460.50 11322.25 14284.50 17593.75 21588.25 25453.50
## [9] 28800.25 31867.25 34855.25 38085.00
```

Oczekiwany okres życia spółki przy założeniu, że maksymalny czas jej życia wynosi  $T$ :

```
oczekiwany_okres_zycia <- function(wyniki, T) {
  okres_zycia <- apply(!is.na(wyniki), 1, function(x) sum(x))
  sredni_okres_zycia <- mean(okres_zycia)
  return(sredni_okres_zycia)
}
sredni_okres_zycia <- oczekiwany_okres_zycia(wyniki, T)
print(sredni_okres_zycia)
```

```
## [1] 12
```