

Domaća naloga 9

Alen Kahteran

9. 11. 2020

Definicija funkcij za igri

```
igra1 <- function(){  
  ### returns TRUE or FALSE value based on simulation of igra1. throwing 1 dice  
  ### TRUE represents a win (one 6 out of 4 throws)  
  ### FALSE represents a loss (no 6 out of 4 throws)  
  ### 4 throws of a dice, if any of those is 6, return TRUE else FALSE  
  ### No input required  
  return(any(sample(1:6, 4, replace=TRUE) == 6))  
}  
  
igra2 <- function(){  
  ### returns TRUE or FALSE value based on simulation of igra2. throwing 2 dices  
  ### TRUE represents a win (two 6 in same throw out of 24 throws)  
  ### FALSE represents a loss (no occurrence of two 6 out of 24 throws)  
  ### 24 throws of a dice, if sum of any 2 pairs is equal to 12 (2x6),  
  ### return TRUE else FALSE  
  ### No input required  
  return(any(rowSums(cbind(sample(1:6, 24, replace=TRUE), sample(1:6, 24, replace=TRUE))) == 12))  
}
```

Ocena verjetnosti za zmago (simulacija)

```
simulate_probabilities <- function(n, verbose=FALSE){  
  ### returns probabilities of winning for each game. prob1 for game 1, prob2 for game2  
  ### position 1 of return vector is probability for game 1  
  ### position 2 of return vector is probability for game 2  
  ### input:  
  ### n - a number of simulations of a game.  
  ### verbose - a variable which decides the printing of the function, default=FALSE  
  
  prob1 <- mean(replicate(n, igra1()))  
  prob2 <- mean(replicate(n, igra2()))  
  if (verbose) {  
    print(paste0("verjetnost zmage pri igri 1: ", prob1))  
    print(paste0("verjetnost zmage pri igri 2: ", prob2))  
  } else {  
    return(c(prob1, prob2))  
  }  
}
```

```

# setting seed for reproducibility
set.seed(8)

# calculate probabilities for game 1 and game 2 when number of games is 10^5
simulate_probabilities(10^5, verbose=TRUE)

## [1] "verjetnost zmage pri igri 1: 0.51732"
## [1] "verjetnost zmage pri igri 2: 0.49558"

```

Izračun zmag za igro 1

```

num_game1_won <- function(x){
  ### returns data.frame of games won or games won/equalized for game 1 (compared to g2)
  ### for specific x
  ### input:
  ###      x - a number of simulations of a game.

  # create empty vectors
  g1_better <- rep(NA, x)
  g2_better_equal <- rep(NA, x)

  # go from 1 to x
  for (i in 1:x){
    # simulate 100 simulate_probabilities() where i is the number of simulations per
    # game.
    res_matr <- replicate(100, simulate_probabilities(i))
    # here we can compare probabilities as it's basically the same as comparing games
    # won (up to computer precision due to floating point error)
    # 58 > 50 is the same as 0.58 > 0.50
    #
    # game 1 won
    g1_better[i] <- sum(res_matr[1, ] > res_matr[2, ])
    # game 1 won or was equal
    g2_better_equal[i] <- sum(res_matr[1, ] >= res_matr[2, ])
  }

  # adding columns for classification
  g1_better <- cbind(g1_better, "zmage")
  g2_better_equal <- cbind(g2_better_equal, "zmage/izenačeno")

  # first binding row-wise
  # then adding 1:x vector for visualisation
  return_matr <- cbind(rbind(g1_better, g2_better_equal), 1:x)
  # changing column names
  colnames(return_matr) <- c("Games_ended_for_1", "Result_for_1", "Games_per_day")

  # converting to data.frame then to tibble, for easier handling
  df_return <- tibble(data.frame(return_matr))
  # converting to integer, as matrix can only hold one type of value, which converts
  # everything to "char".
  df_return$Games_ended_for_1 <- as.integer(df_return$Games_ended_for_1)
  df_return$Games_per_day <- as.integer(df_return$Games_per_day)
  # return df
}

```

```

    return(df_return)
}

# setting seed for reproducibility
set.seed(9)

# get data for up to 50 games.
x <- 50
data_full <- num_game1_won(x)

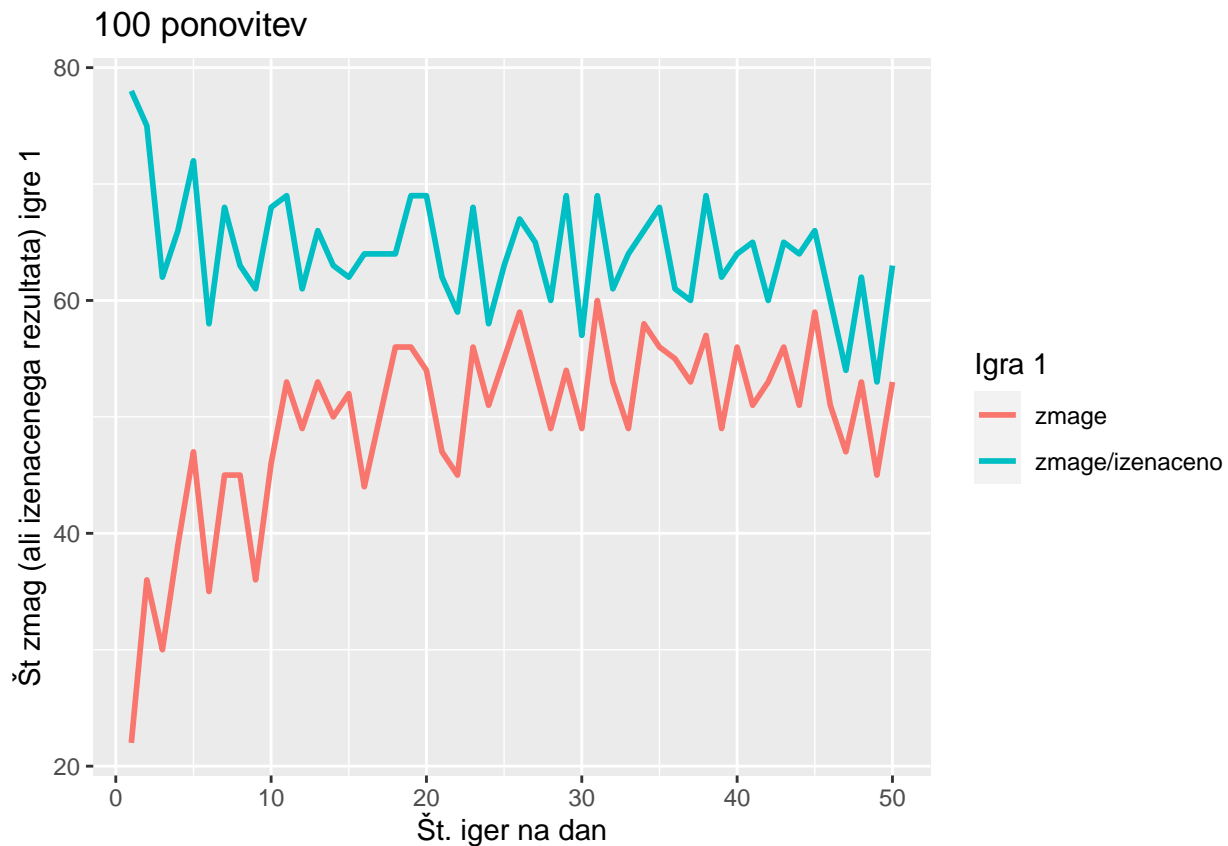
```

Izris

```

ggplot(data_full, aes(x=Games_per_day, y=Games_ended_for_1, col=Result_for_1)) +
  geom_line(stat="identity", size=1) +
  labs(title="100 ponovitev",
       x="Št. iger na dan",
       y="Št zmag (ali izenačenega rezultata) igre 1",
       col="Igra 1")

```



Že višje, kjer smo simulirali 10^5 iger, vidimo da je `igra1()` boljša. Verjetnost je višja kot pri `igra2()`, poleg tega pa je tudi višja kot 0.5. Če jo primerjamo z `igra2()` (na grafu), vidimo da nekje do 10 iger na dan, ne gremo čez magično mejo 50 ponovitev, vendar tudi izgubimo ne, saj je v teh primerih več iger kjer igramo izenačeno. Nato se št. zmag stabilizira nekaj malega višje od 50 (od 100 ponovitev). Poleg tega je še okrog 10 iger izenačenih.