Dice 10,000

Peter Crawford

9/1/2020

Introduction:

Rules for the game Dice (10,000): https://en.wikipedia.org/wiki/Dice_10000#:~:text=The%20first%20player%20to%20score

We play where you need to score 1,000 to get on the board and there is no point requirement for turns after getting on the board. We also play where you can't roll from the previous player's dice - you start with six fresh dice every time. The rest of the rules on the above wikipedia link seem to be how we usually play.

The main question to be answered here is what score should I aim for each turn that will increase my chances of reaching 10,000 fastest? (i.e. should I always stop after rolling past 400? 500? 600?)

Code:

Packages:

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
## Warning: package 'forcats' was built under R version 3.5.2
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Create a function to roll dice. (Only variable being the number of dice rolled)

```
roll <- function(num_of_dice) {
  sample(c(1:6), num_of_dice, replace = TRUE)
}
```

Create a function for scoring the rolled dice, and re-rolling if threshold not breached.

```
score <- function(hand, threshold_value, start_points) {

  ## Starting values:
  values <- as.data.frame(table(hand)) ## get frequency
  values$hand <- as.numeric(levels(values$hand))[values$hand]

  ## Straight:
  if (length(values$hand[values$Freq == 1]) == 6) { points <- 1500 }

  ## Combinations, ones, and fives:
  values$combos <- ifelse(values$Freq >= 3, (values$hand * 100) * (values$Freq - 2), 0)
  values$combos <- ifelse(values$combos == 100, values$combos * 10, values$combos)
  values$ones <- ifelse((values$combos == 0) & (values$hand == 1), 100 * values$Freq, 0)
  values$fives <- ifelse((values$combos == 0) & (values$hand == 5), 50 * values$Freq, 0)
  values$unused_dice <- ifelse((values$combos == 0) & (values$ones == 0) & (values$fives == 0), values$

  ## Get the roll score, number of leftover dice, and running total score:
  points <- sum(values$combos) + sum(values$ones) + sum(values$fives)
  unused_dice <- sum(values$unused_dice)
  total_points <- start_points + points

  if (points == 0) {
    return(points)
  }

  ## Determine whether or not to continue rolling:
  ## 1. Stop rolling if threshold is reached
  ## 2. Stop rolling if no points scored & no
  if ((total_points < threshold_value) | (unused_dice == 0)) {

    if (unused_dice == 0) {
      score(hand = roll(6), threshold_value = threshold_value, start_points = total_points)
    }
    else {
      score(hand = roll(unused_dice), threshold_value = threshold_value, start_points = total_points)
    }
  }
  else {
    return(total_points)
  }
}
```

This function uses Monte Carlo simulation (replicate) to see how many turns taken to win with given threshold:

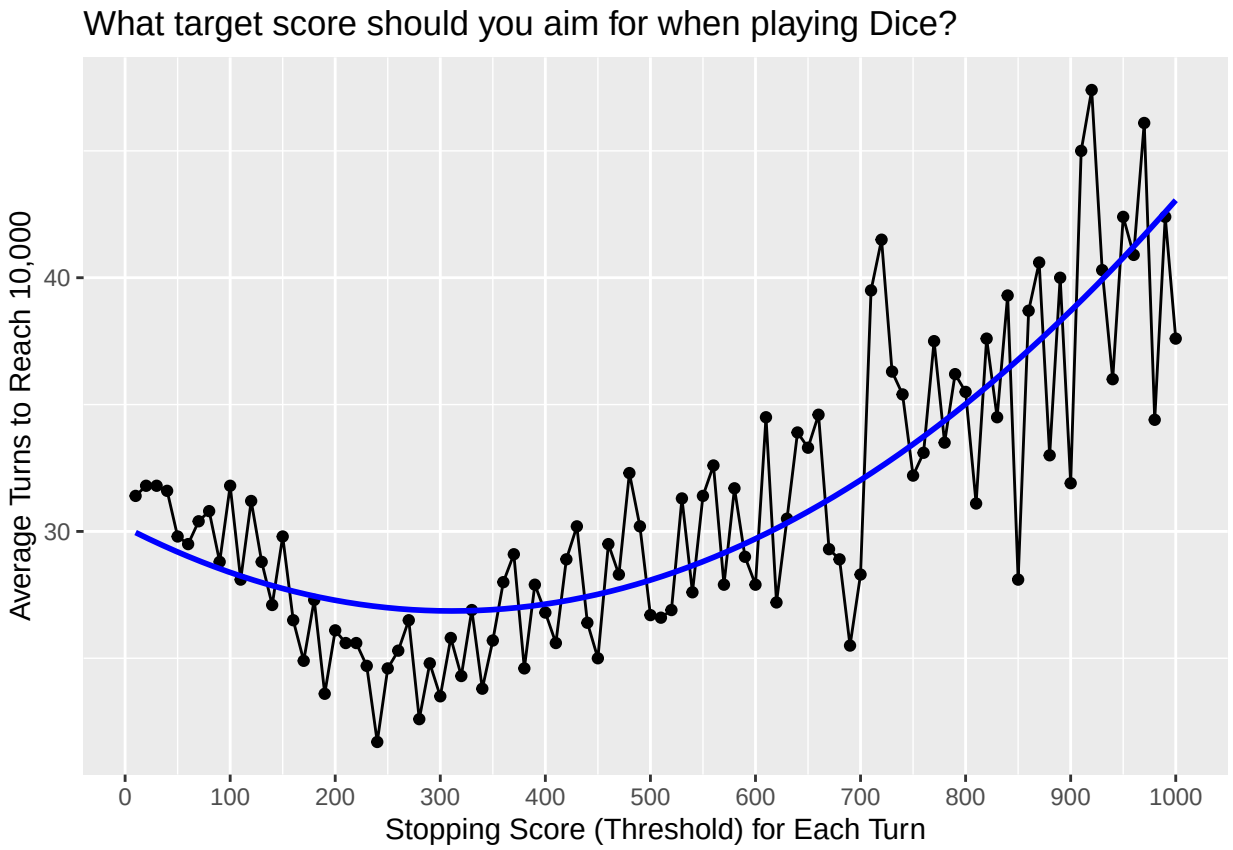
```
turns_to_win <- function(thresh_val) {

  total_score <- 0
  turn_scores <- NULL
  i <- 0

  while(total_score < 10000) {
    i <- i + 1
    turn_scores[i] <- score(hand = roll(6), threshold_value = thresh_val, start_points = 0)
    total_score <- sum(turn_scores)
  }

  return(length(turn_scores))
}
```

Create a df with the threshold and the turns to win:

```
n_reps <- 10 # When this value is greater than 10, computing time takes super long,
             # but the greater the reps, the closer we get to approximating the true values for each th
             # Ideally, n_reps would be around 10,000 or so.

df_10000 <- data.frame("thresh" = 50, "ave_turns_to_win" = 20)

for (i in 1:100) {

  temp_thresh <- 10 * i
  turns_ave <- mean(replicate(n_reps, turns_to_win(thresh_val = temp_thresh)))

  new_row <- c(temp_thresh, turns_ave)

  df_10000 <- rbind(df_10000, new_row)
  # print(i) # This is useful when n_reps is big as it will show us which iteration of our for loop we
}

df_10000 <- df_10000[-1,] # Drop the initial value to
```

Plot, bottom of curve signifies faster at reaching 10,000 points (winning):

```
ggplot(data = df_10000, aes(x = thresh, y = ave_turns_to_win)) +
  geom_line() +
  geom_point() +
  xlab("Stopping Score (Threshold) for Each Turn") +
  ylab("Average Turns to Reach 10,000") +
  ggtitle("What target score should you aim for when playing Dice?") +
  scale_x_continuous(breaks = seq(0,1000, by = 100)) +
  # geom_smooth(method = "lm", se = FALSE)
  stat_smooth(method = "lm", se = TRUE, fill = NA, formula = y ~ poly(x, 2, raw = TRUE), colour = "blue
```


What target score should you aim for when playing Dice?
Average Turns to Reach 10,000
40
30
0
100
200
300
400
500
600
700
800
900
1000
Stopping Score (Threshold) for Each Turn