Homework 5

Aidan Pizzo

10/6/21

analysis.R

```
die_roll <- function()
{
  roll <- sample(1:6, 1)
  return(roll)
}

move_piece_die_roll <- function(position)
{
  roll <- die_roll()
  square <- position + roll
  return(square)
}

chute_and_ladder <- function(position, d)
{
  new_position <- d$end[d$start == position]
  return(new_position)
}

get_chute_and_ladder <- function()
{
  d <- read.csv("../data/chutes_and_ladder_locations.csv",
                header = T,
                stringsAsFactors = F)
  return(d)
}

move <- function(position, d)
{
  new_position <- move_piece_die_roll(position)
  if (is.element(new_position, d$start))
  {
    new_position <- chute_and_ladder(new_position, d)
  }
  return(new_position)
}

play <- function(d)
{
  gameplay <- vector(mode = "list", length = 2)
  names(gameplay) <- c("player1", "player2")
  p1_position <- 0
  p2_position <- 0
  p1_game <- vector("numeric")
  p2_game <- vector("numeric")
  move_count <- 1
  while (p1_position <= 100 && p2_position <=100)
  {
    p1_position <- move(p1_position, d)
    p2_position <- move(p2_position, d)
    p1_game[move_count] <- p1_position
    p2_game[move_count] <- p2_position
    move_count <- move_count + 1
  }
  gameplay[[1]] <- p1_game
  gameplay[[2]] <- p2_game
  return(gameplay)
}

simulate <- function(d, n)
{
```

```
  out_list <- vector(mode = "list", length = n)
  for (i in 1:n)
  {
    add_this <- play(d)
    out_list[[i]] <- add_this
  }
  return(out_list)
}
```

Problem 1

```
d <- get_chute_and_ladder()
simulation <- simulate(d, 1000)
num_rounds <- vector("numeric", length(simulation))
for (i in 1: length(num_rounds))
{
  num_rounds[i] <- length(simulation[[i]]$player1)
}
avg <- sum(num_rounds)/length(num_rounds)
```

avg [1] 24.624

Problem 2

```
d <- get_chute_and_ladder()
simulation <- simulate(d, 1000)
num_rounds <- list()
for (i in 1: length(simulation))
{ #check if 84 was landed one
  if (is.element(84, simulation[[i]]$player1) == TRUE)
  { #check the index before 84 to see if it was <28, indicating a ladder was used
    if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1)-1)] < 28)
    { #add the number of rounds of that game
      rounds <- length(simulation[[i]]$player1)
      num_rounds <- c(num_rounds, rounds)
    }
  }
  else if (is.element(84, simulation[[i]]$player2) == TRUE)
  {
    if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2)-1)] < 28)
    {
      rounds <- length(simulation[[i]]$player2)
      num_rounds <- c(num_rounds, rounds)
    }
  }
}
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player2) - :
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used

## Warning in if (simulation[[i]]$player1[(grep(84, simulation[[i]]$player1) - :
## the condition has length > 1 and only the first element will be used
```

```
num_rounds <- unlist(num_rounds)
avg <- sum(num_rounds)/length(num_rounds)
```

```
problem2(d) [1] 23.98615
```