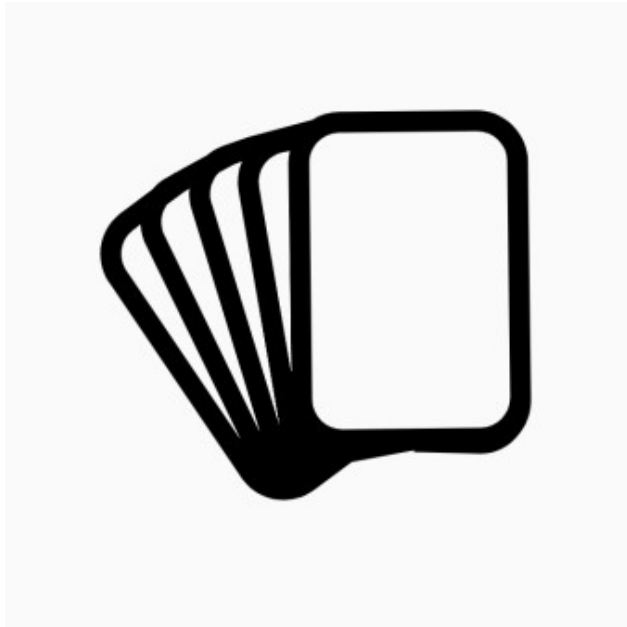


Creating a function that will play two players, both given a set of cards (represented with numbers).



**Ground rules are simple:**

- both players get same number of cards
- winner of the game is the player taking all the cards
- both players draw one card at the time from the top of the stack
- if both players draw the same card, both return the card at the bottom of the stack
- player takes the draw if the number is higher than opponents'
- cards are put at the stack in following order: first card is the card of the winner (own card) and second is card from opponent

Create the *draw()* function. This function will make comparison of two cards and stack one or two at the bottom of the stack:

```
draw <- function(n1,n2){  
  if (n1 > n2) {  
    val <- c(n1, n2)  
    p1 <- append(p1,val)  
    p1 <- p1[(2:length(p1))]  
    p2 <- p2[(2:length(p2))]  
  } else if (n1 < n2) {  
    val <- c(n2,n1)  
    p2 <- append(p2, val)  
    p1 <- p1[(2:length(p1))]  
    p2 <- p2[(2:length(p2))]  
  } else {  
    p1 <- append(p1,n1)  
    p2 <- append(p2,n2)  
    p1 <- p1[(2:length(p1))]  
    p2 <- p2[(2:length(p2))]  
  }  
}
```

In addition, we need also the *play()* function that will look for the winner and stop the game:

```
play <- function(){  
  i = 1
```

```

while(length(p1) != 0 | length(p2) != 0){
  print(paste0("Playing round: ",i, ". Player1 draws: ", p1[1], " while
Player2 draws: ", p2[1]), collapse = NULL)
  draw(p1[1],p2[1])
  i = i + 1
  if (is.na(p1) == TRUE) {
    print(paste0("Player 2 Wins! in ", i, " round"), collapse = NULL)
    break()
  } else if (is.na(p2) == TRUE) {
    print(paste0("Player 1 Wins! in ", i, " round"), collapse = NULL)
    break()
  }
}
}
}

```

So if I create a stack of 6 cards per player and start the function:

```

set.seed(2908)
p1 <- c(4,6,5,7,3,6)
p2 <- c(6,2,1,7,8,7)
play()

```

I see that it takes 30 draws for player 2 to win.

Pretty useless and boring. What if we make the function more useful by adding some statistics and a graph? This calls for another function *playWithStats()*.

```

library(ggplot2)

playWithStats <- function(){
  i = 1
  df <- NULL
  df <- data.frame(round=1, p1=length(p1), p2=length(p2))
  while(length(p1) != 0 | length(p2) != 0){
    print(paste0("Playing round: ",i, ". Player1 draws: ", p1[1], " while
Player2 draws: ", p2[1]), collapse = NULL)
    draw(p1[1],p2[1])
    i = i + 1
    df <- rbind(df, data.frame(round=i, p1=length(p1), p2=length(p2)))
    if (is.na(p1) == TRUE) {
      print(paste0("Player 2 Wins! in ", i, " round"), collapse = NULL)
      break()
    } else if (is.na(p2) == TRUE) {
      print(paste0("Player 1 Wins! in ", i, " round"), collapse = NULL)
      break()
    }
  }
  df_out <-<- df
  g <- ggplot(df_out, aes(round))
  g <- g + geom_line(aes(y=p1), colour="red")
  g <- g + geom_line(aes(y=p2), colour="green")
  g
  plot(g)
}

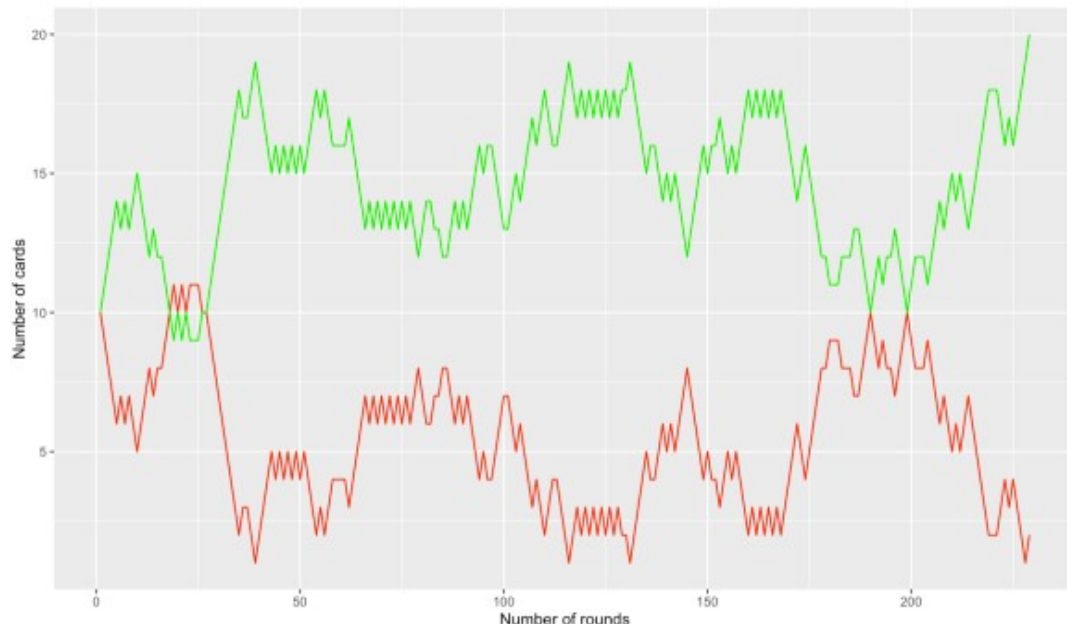
```

This will track number of cards each player will have after each draw and see the game development.

I will also let computer choose the cards for each player:

```
# converges in 229 rounds
set.seed(2908)
p1 <- sample(1:10,10,replace=TRUE)
p2 <- sample(1:10,10,replace=TRUE)
playWithStats()
```

and produces an symmetrical graph for both users (Number of rounds per number of cards in stack for each user).



On first glance it looks like the game would finish around round 40 but it keeps moving until round 190 where each player has 10 cards each. Interesting game dynamics. Eventually, game converges in 230 rounds and Player 2 wins.

What happens if we do the sampling of the cards without repetition, so that each card must only appear once in the stack?.

So, this `sample(1:10,10,replace=TRUE)` would become this: `sample(1:10,10,replace=FALSE)`. I already smell problems.

All these combinations (both players have 3,4,5 or 10 cards in stack each) will never converge.

```
# never converges
set.seed(2908)
p1 <- sample(1:10,10,replace=FALSE)
p2 <- sample(1:10,10,replace=FALSE)
playWithStats()
```

```
# never converges
set.seed(2908)
p1 <- sample(1:6,6,replace=FALSE)
p2 <- sample(1:6,6,replace=FALSE)
playWithStats()
```

```
# never converges
set.seed(2908)
p1 <- sample(1:5,5,replace=FALSE)
p2 <- sample(1:5,5,replace=FALSE)
rm(df_out)
playWithStats()
```

```
# never converges
set.seed(2908)
p1 <- sample(1:4,4,replace=FALSE)
p2 <- sample(1:4,4,replace=FALSE)
playWithStats()
```

```
# never converges
set.seed(2908)
p1 <- sample(1:3,3,replace=FALSE)
p2 <- sample(1:3,3,replace=FALSE)
playWithStats()
```

Why? Because **all cards get sorted in each stack the same way for both players**. Let's put it to the test:

```
set.seed(2908)
p1 <- sample(1:3,3,replace=FALSE)
p2 <- sample(1:3,3,replace=FALSE)
playWithStats()
```

And will run it with breakpoints, so I can stop and check the stack every run. This seed will produce starting numbers of:

**At Start:**

P1: 2, 1, 3

P2: 2, 3, 1

**1. Run (2,2):**

P1: 1, 3, 2

P2: 3, 1, 2

**2. Run (1,3):**

P1: 3, 2

P2: 1, 2, 3, 1

**3. Run(3,1):**

P1: 2, 3, 1

P2: 2, 3, 1

All another runs are useless, since the game will not converge and end in a loop.

Same will appear when adding more cards (sample(1:5,5,replace=FALSE) or sample(1:10,10,replace=FALSE)) but essentially keeping the replace=FALSE making a deck of cards with no repetition. One can say, this is the reason, why many card games have 2 or 4 repetition of decks; to enable game to end or to change the ground rules of the game (e.g.: when players draw same card, they should hold to a draw until on wins a stack of draws).