

Maximal score

Isaac Verminck

January 14, 2018

Gouden 11

The goal of the game is to come up with a selection of players earning the most points possible.

Constraints

Several constraints regarding what players to select have to be taken into account:

- the total cost of the team cannot exceed 250M
- you pick 15 players (starting eleven + 4 replacements)
- you pick a formation
- you can only pick 2 players of the same team
- you pick a captain earning extra points depending on the game result
- you can make a maximum of 6 transfers

Steps to calculate

- function to pick a legal team
- run this function n times
- pick the formation with the highest total score

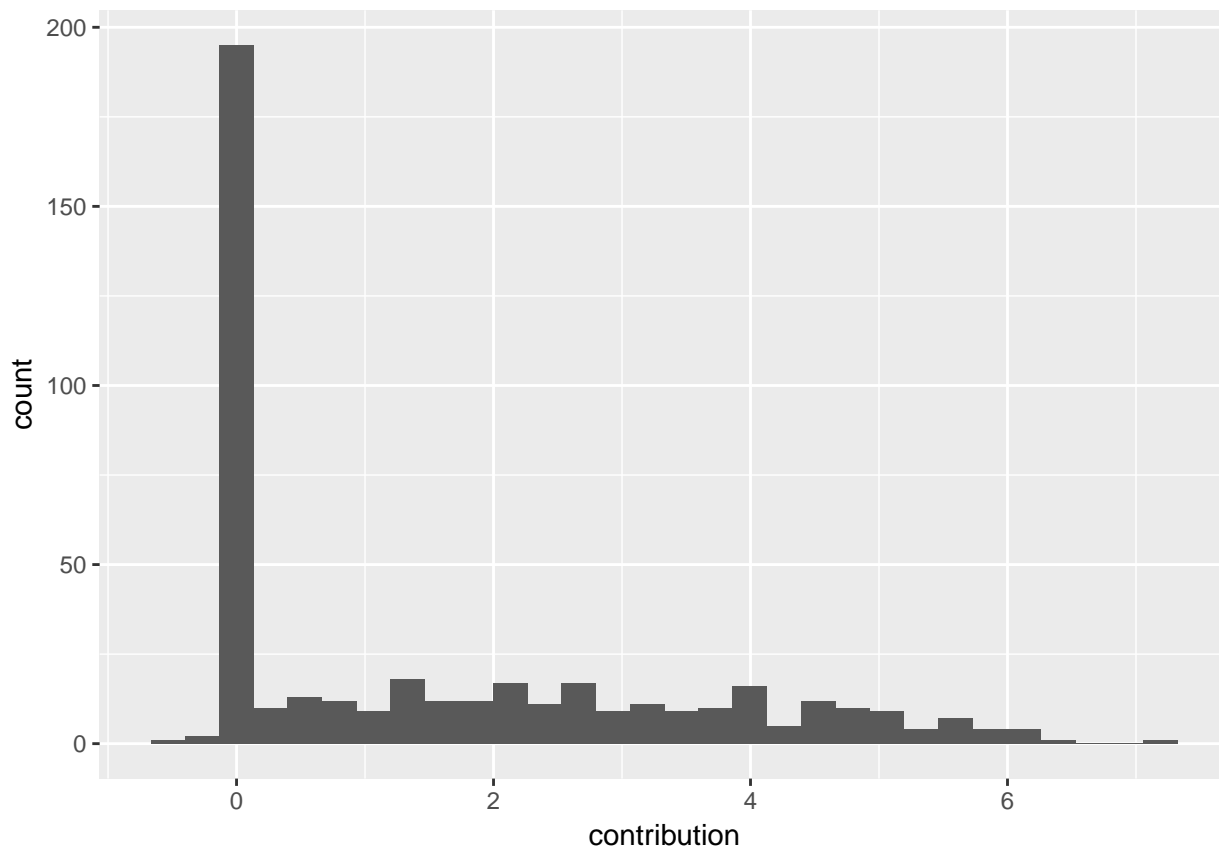
For the time being we assume the following: * you don't pick a captain * no replacements are made * no transfers are made

Function to pick a legal team

To pick a legal team we have to specify the formation, cost limit and maximum number of players on the same team. Optionally we can remove some players before we start randomly making selections to make the picking process faster (albeit at the cost of possible suboptimal results).

Which players are filtered out depends on the contribution threshold. We define contribution as the total score of a player divided by his cost. In our dataset the distribution of contributions is as follows:

```
ggplot(data = player_score_by_season,  
       aes(x = contribution)) + geom_histogram()
```



```
n_zero_contribution_players = sum(player_score_by_season$contribution <= 0)
```

It's clear a lot of players (196) don't contribute at all. So by default we will set the contribution threshold to 0 meaning all players with no (or even a negative) contribution are filtered out beforehand.

```
# formation e.g. 4-4-2
# cost limit in millions
# if remove_low_value_players = TRUE players who probably won't add much (depending on contribution_thr
pick <- function(formation,
  dataset,
  cost_limit = 250,
  max_n_players_same_team = 2,
  remove_low_contribution_players = TRUE,
  contribution_threshold = 0) {
  # parse formation
  positions <- c("Doelman", "Verdediger", "Middenvelder", "Aanvaller")
  n_to_select <- c(1, as.integer(unlist(strsplit(formation, "-"))))

  # remove low value players
  dataset <- filter(dataset, contribution > contribution_threshold)

  # select number of players by position
  select_n_by_position <- function(name_position, n, dataset) {
    dataset %>%
      filter(position == name_position) %>%
      sample_n(n)
  }
}
```

```

selection <- map2(.x = positions,
                 .y = n_to_select,
                 .f = select_n_by_position,
                 data = dataset) %>%
  reduce(bind_rows)

# check if selection is legal
doesnt_exceed_cost_limit <- sum(selection$opening_price) < cost_limit
max_two_players_same_team <- selection %>%
  group_by(team) %>%
  summarise(n = n()) %>%
  pull(n)
max_two_players_same_team <- all(max_two_players_same_team <= max_n_players_same_team)

if (doesnt_exceed_cost_limit & max_two_players_same_team) {
  return(selection)
} else {
  return("violated constraints")
}

return(selection)
}

```

Run picking function number of times

Let's presume we pick 2000 teams. Afterwards we filter out the ones who violated the constraints and we try to guess at which point the maximum total score stabilises by using the cumulative maximum score.

```

# run n times
n = 2000
selections <- replicate(n, pick("4-4-2", player_score_by_season))

# filter out selections violating constraints
violates = map_lgl(selections, function(selection) 'violated constraints' %in% selection)
ind_violating_selections = which(violates)
ind_correct_selections = which(!violates)
percentage_violation = (sum(length(ind_violating_selections))/n)*100

# after how many team picks does the maximum score stabilises?
total_score = function(selection) {
  sum(selection$total)
}

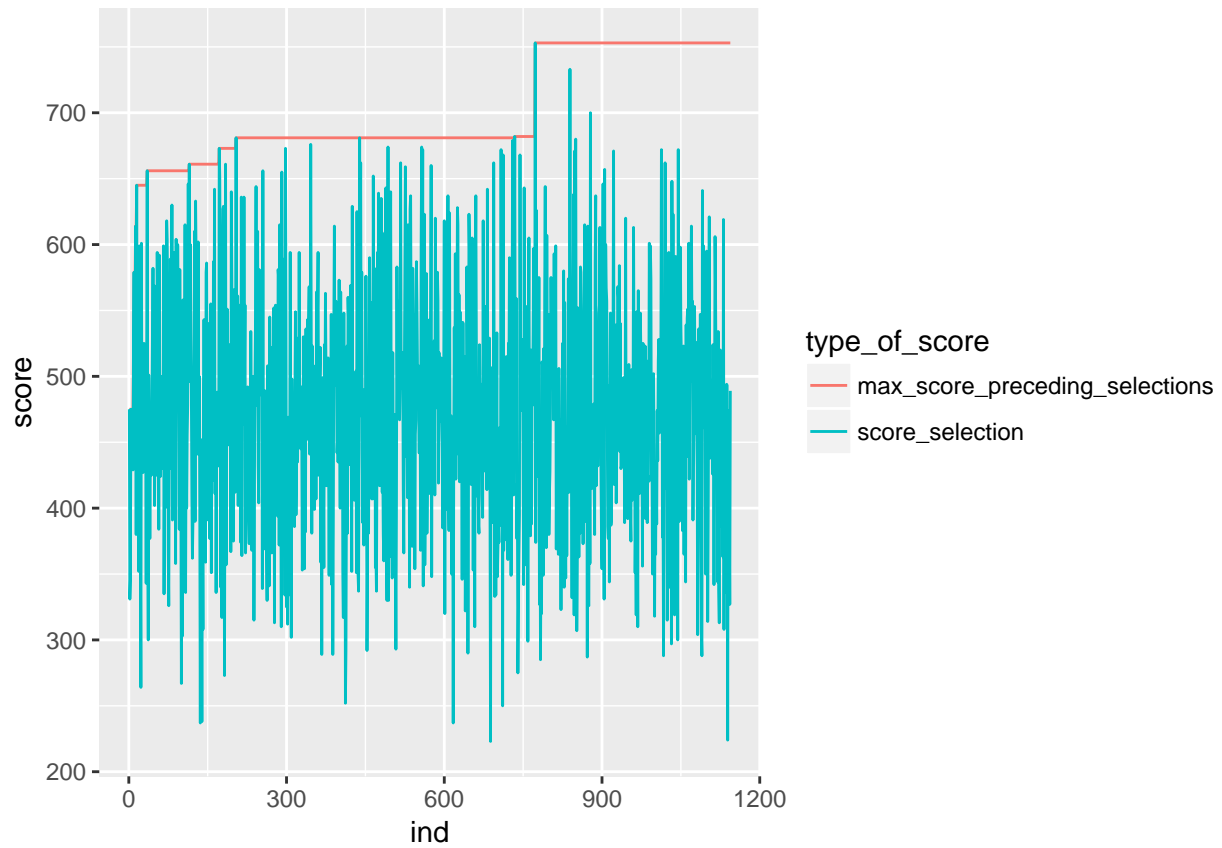
total_score_by_selection = selections %>%
  .[ind_correct_selections] %>%
  map_int(total_score)

cumulative_maximum_score = cummax(total_score_by_selection)

score_stabilisation <- data.frame(max_score_preceding_selections = cumulative_maximum_score,
                                score_selection = total_score_by_selection,
                                ind = 1:length(cumulative_maximum_score))
score_stabilisation <- gather(data = score_stabilisation, key = "type_of_score", value = "score", -ind)

```

```
ggplot(data = score_stabilisation,
       aes(x = ind, y = score, colour = type_of_score)) +
  geom_line()
```



42.8% of the selections generated violated at least one rule.