Simulação Jogo Probabilândia

Lucas dos Santos Rodrigues Szavara

2025-04-24

Import bibliotecas

```
library(dplyr)
```

```
##
## Anexando pacote: 'dplyr'
```

```
## Os seguintes objetos são mascarados por 'package:stats':
##
##     filter, lag
```

```
## Os seguintes objetos são mascarados por 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(SimDesign)
library(purrr)
library(tidyr)
```

Criação do Design da simulação

Nesse momento vamos considerar 2 tabuleiros, um em que o jogador sempre escolhe ir pela bifurcação com mais perguntas, e outro em que o jogador prefere evitar as perguntas. Vamos considerar apenas o próximo passo, removendo situações em que ganhar menos pontos imediatamente aumenta os pontos a serem ganhos posteriormente.

Para simplicidade ao calcular o número de pontos ganhos, vamos substituir as letras indicando o tipo de casa pelo índice no vetor de pontos

```
tabuleiro_prefere <- c('b', 'v', 'r', 'v', 'b', 'r', 'v', 'b', 'v', 'r', 'b',
                       'v', 'v', 'r', 'b', 'v', 'r', 'b', 'v', 'r', 'r', 'r',
                       'b', 'v', 'b', 'v', 'v', 'v')
tabuleiro_evita_pergunta <- c('b', 'v', 'r', 'v', 'b', 'r', 'v', 'b', 'v', 'r',
                              'b', 'r', 'r', 'r', 'b', 'v', 'r', 'r', 'r', 'b',
                              'v', 'b', 'r', 'r')
print(table(tabuleiro_prefere))
```

```
## tabuleiro_prefere
##  b  r  v
##  8  8 12
```

```
print(table(tabuleiro_evita_pergunta))
```

```
## tabuleiro_evita_pergunta
##  b  r  v
##  7 11  6
```

```
tabuleiro_prefere[tabuleiro_prefere == 'b'] <- 1
tabuleiro_prefere[tabuleiro_prefere == 'r'] <- 2
tabuleiro_prefere[tabuleiro_prefere == 'v'] <- 3
tabuleiro_prefere <- tabuleiro_prefere %>%
    as.double()

tabuleiro_evita_pergunta[tabuleiro_evita_pergunta == 'b'] <- 1
tabuleiro_evita_pergunta[tabuleiro_evita_pergunta == 'r'] <- 2
tabuleiro_evita_pergunta[tabuleiro_evita_pergunta == 'v'] <- 3
tabuleiro_evita_pergunta <- tabuleiro_evita_pergunta %>%
    as.double()
dados <- c(4, 6, 8, 10, 12)
```

Vamos criar uma tabela com todas as combinações a serem consideradas das variáveis a seguir: Para simplificação ao gerar o pdf, vamos restringir o valor, mas podemos incluir as opções desejadas

```
Design <- createDesign(
    # Tabuleiro a ser usado:
    tabuleiro = list(tabuleiro_prefere, tabuleiro_evita_pergunta),
    # A probabilidade de acerto de uma pergunta do jogador que não evita:
    probabilidade_acerto = c(0.1, 0.5),
    # Quantidade de pontos (casa branca, rosa, erro, acerto):
    pontos = list(c(3, 1, 0, 5), c(2, 1, 0, 3), c(3, 1, -5, 5)),
    # Posição atual:
    posicao = c(1, 5, 10, 20)#, 5, 10, 15, 20)
)
```

Simulação dos cenários

Função geradora de dados

Agora, vamos criar uma função tal que, dado um cenário, ela gera o próximo passo a ser tomado

```
Generate <- function(condition, fixed_objects = FALSE) {
    # Quantidade de passos para cada dado
    passos <- dados %>%
        sapply(function(d) sample(1:d, 1))
    # Variável para auxiliar a identificar se o jogador irá acertar a pergunta
    resposta_correta <- runif(1)
    return(list(passos = passos, resposta_correta = resposta_correta))
}
```

Análise de um passo

```
Analyse <- function(condition, dat, fixed_objects=FALSE) {
    pontos <- condition$pontos[[1]]
    posicao <- condition$posicao
    tabuleiro <- condition$tabuleiro[[1]]
    passos <- dat$passos
    resposta_correta <- dat$resposta_correta <= condition$probabilidade_acerto
    posicao_atual <- (posicao + passos) %% (1 + length(tabuleiro))
```

```
    deu_volta <- posicao + passos > length(tabuleiro)
    posicao_atual[deu_volta] <- posicao_atual[deu_volta] + 1
    tipo_posicao <- tabuleiro[posicao_atual]
    ponto <- tipo_posicao
    caiu_em_pergunta <- ponto == 3
    ponto[caiu_em_pergunta] <- ponto[caiu_em_pergunta] + resposta_correta
    return(c(pontos = pontos[ponto], caiu_em_pergunta = caiu_em_pergunta))
}
```

Agrupar analises de multiplas repetições

```
Summarise <- function(condition, results, fixed_objects=FALSE) {

    caiu_em_pergunta <- results[, 6:10]
    results <- results[, 1:5]

    # Probabilidade de cada dado de cair em pergunta
    pp_d4 <- mean(caiu_em_pergunta[, 1])
    pp_d6 <- mean(caiu_em_pergunta[, 2])
    pp_d8 <- mean(caiu_em_pergunta[, 3])
    pp_d10 <- mean(caiu_em_pergunta[, 4])
    pp_d12 <- mean(caiu_em_pergunta[, 5])


    # Probabilidade de cada dado apresentar o melhor resultado
    p_d4 <- mean(results[, 1] == apply(results, 1, max))
    p_d6 <- mean(results[, 2] == apply(results, 1, max))
    p_d8 <- mean(results[, 3] == apply(results, 1, max))
    p_d10 <- mean(results[, 4] == apply(results, 1, max))
    p_d12 <- mean(results[, 5] == apply(results, 1, max))


    # Media de pontos obtidos ao escolher cada dado
    media_pontos_d4 <- mean(results[, 1])
    media_pontos_d6 <- mean(results[, 2])
    media_pontos_d8 <- mean(results[, 3])
    media_pontos_d10 <- mean(results[, 4])
    media_pontos_d12 <- mean(results[, 5])

    # Mediana de pontos obtidos ao escolher cada dado
    mediana_pontos_d4 <- median(results[, 1])
    mediana_pontos_d6 <- median(results[, 2])
    mediana_pontos_d8 <- median(results[, 3])
    mediana_pontos_d10 <- median(results[, 4])
    mediana_pontos_d12 <- median(results[, 5])

    # Desvio padrão de pontos obtidos ao escolher cada dado
    sd_pontos_d4 <- sd(results[, 1])
    sd_pontos_d6 <- sd(results[, 2])
    sd_pontos_d8 <- sd(results[, 3])
    sd_pontos_d10 <- sd(results[, 4])
    sd_pontos_d12 <- sd(results[, 5])
    return(c(
```

```
        pp_d4 = pp_d4,
        pp_d6 = pp_d6,
        pp_d8 = pp_d8,
        pp_d10 = pp_d10,
        pp_d12 = pp_d12,

        p_d4 = p_d4,
        p_d6 = p_d6,
        p_d8 = p_d8,
        p_d10 = p_d10,
        p_d12 = p_d12,

        media_d4 = media_pontos_d4,
        media_d6 = media_pontos_d6,
        media_d8 = media_pontos_d8,
        media_d10 = media_pontos_d10,
        media_d12 = media_pontos_d12,

        mediana_d4 = mediana_pontos_d4,
        mediana_d6 = mediana_pontos_d6,
        mediana_d8 = mediana_pontos_d8,
        mediana_d10 = mediana_pontos_d10,
        mediana_d12 = mediana_pontos_d12,

        sd_pontos_d4 = sd_pontos_d4,
        sd_pontos_d6 = sd_pontos_d6,
        sd_pontos_d8 = sd_pontos_d8,
        sd_pontos_d10 = sd_pontos_d10,
        sd_pontos_d12 = sd_pontos_d12
    ))
}
```

Teste das funções:

Cenário escolhido:

```
condition <- Design[1, ]
print(condition$pontos[[1]])
```

```
## [1] 3 1 0 5
```

```
print(condition$posicao)
```

```
## [1] 1
```

Resultado dos dados gerados aleatoriamente:

```
dat <- Generate(condition)
dat
```

```
## $passos
## [1]  4  4  5 10 12
##
## $resposta_correta
## [1] 0.6324391
```

Tipo das posições alcançadas usando cada dado

```
tabuleiro_prefere[condition$posicao + dat$passos]
```

```
## [1] 1 1 2 1 3
```

Pontos ganhos usando cada dado e indicador binário se caiu ou não em pergunta

```
results <- Analyse(condition, dat)
results
```

```
##           pontos1           pontos2           pontos3           pontos4
##                 3                 3                 1                 3
##           pontos5 caiu_em_pergunta1 caiu_em_pergunta2 caiu_em_pergunta3
##                 0                 0                 0                 0
## caiu_em_pergunta4 caiu_em_pergunta5
##                 0                 1
```

Aparentemente, nossas funções funcionam de acordo com o esperado. Vamos rodar a simulação completa e obter os resultados agrupados

Execução da simulação

```
res <- runSimulation(
    Design,
    # Trocar para executar
    replications = 500,
    generate = Generate,
    analyse = Analyse,
    summarise = Summarise,
    parallel = T
)
saveRDS(res, 'resultados.rds')
```

Resultados

Podemos agora analisar o resultado no cenário de interesse, considerando que os jogadores estão na posição 1, o jogador que prefere perguntas tem 50% de probabilidade de acertar a pergunta, enquanto o jogador que evita pergunta tem 10%, e a distribuição de pontos é a que será usada no jogo. Para sinalização das colunas, E representa o jogador que evita perguntas, e P o jogador que prefere perguntas

```
res <- res %>%
    mutate(tabuleiro = as.character(tabuleiro)) %>%
    mutate(tabuleiro = if_else(
        nchar(tabuleiro) == 73,
        'E',
        'P'
        ))
```

```
resultado_cenario_1 <- res %>%
    mutate(pontos = as.character(pontos)) %>%
    filter(posicao == 10) %>%
    filter(((probabilidade_acerto == 0.5) & (tabuleiro == 'P')) |
           ((probabilidade_acerto == 0.1) & (tabuleiro == 'E'))) %>%
    filter(pontos == 'c(3, 1, 0, 5)')
resultado_cenario_1 %>%
    select(tabuleiro:sd_pontos_d12) %>%
```

```
    select(!probabilidade_acerto) %>%
    pivot_longer(
        cols = pp_d4:sd_pontos_d12,
        names_to = c('Medida', 'Dado'),
        names_pattern = '(.*)_d(.*)'
    ) %>%
    pivot_wider(
        names_from = c('Dado', 'tabuleiro'),
        names_prefix = 'd'
    ) %>%
    select(Medida | starts_with('d')) %>%
    knitr::kable(digits = 3)
```

Medida	d4_E	d6_E	d8_E	d10_E	d12_E	d4_P	d6_P	d8_P	d10_P	d12_P
pp	0.000	0.154	0.102	0.100	0.158	0.488	0.518	0.408	0.374	0.334
p	0.374	0.434	0.416	0.442	0.490	0.384	0.460	0.470	0.394	0.370
media	1.452	1.558	1.518	1.568	1.674	2.186	2.428	2.416	2.232	2.078
mediana	1.000	1.000	1.000	1.000	1.000	1.000	3.000	3.000	1.000	1.000
sd_pontos	0.837	1.162	1.056	1.084	1.213	1.910	1.926	1.788	1.762	1.698

Como as primeiras casas não tem bifurcações, a diferença na probabilidade de cair em uma pergunta é similar para os dados menores, mas para o D12 a diferença entre os jogadores é clara. Em ambos os casos, o D12 é o dado que maximiza a média e mediana de pontos obtidos, e ainda, se X_i é o valor de pontos obtidos usando o Di, temos que $i = 12$ maximiza:

$$P(X_i = max\{X_4, X_6, X_8, X_{10}, X_{12}\})$$

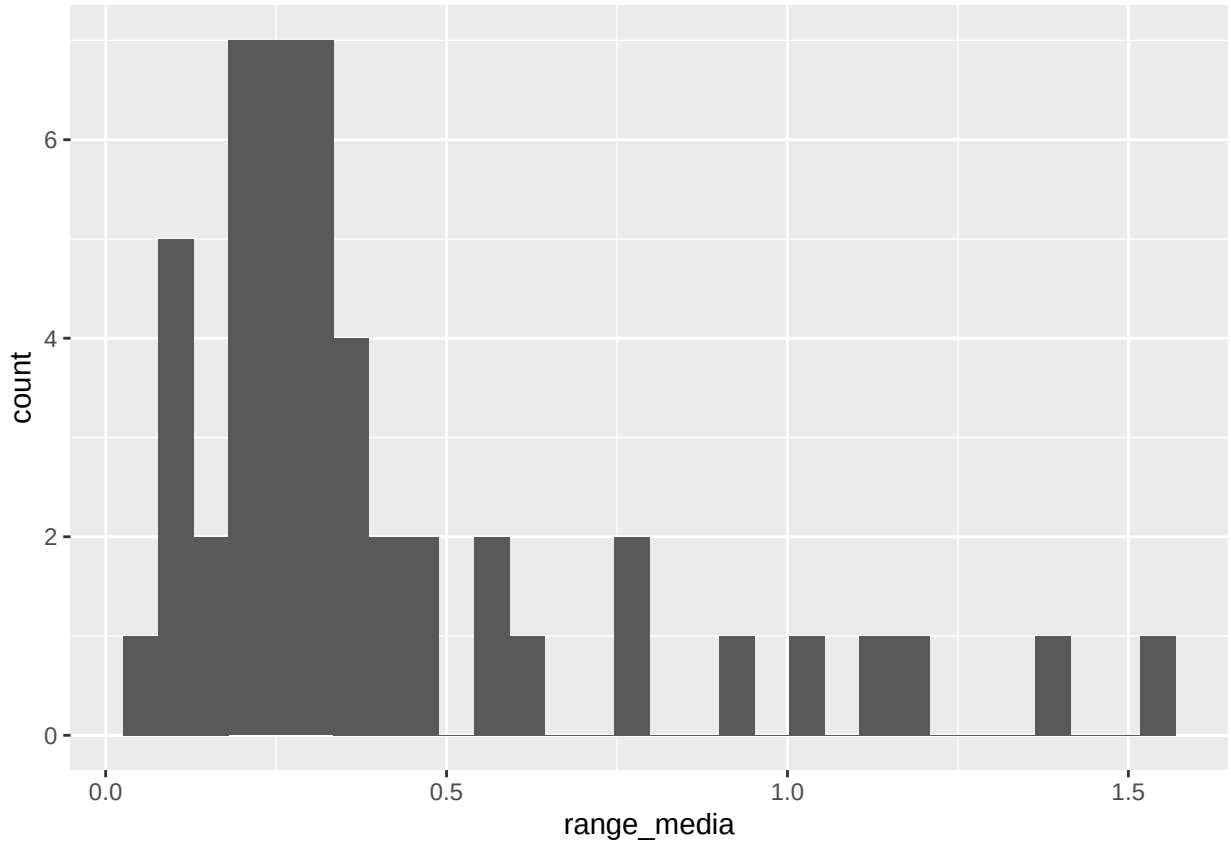
Sendo 45% para o jogador que prefere perguntas e 43% para o que evita Por fim, os pontos do jogador que prefere perguntas tem maior média e desvio padrão do que do jogador que evita.

```
resultado_cenario_2 <- res %>%
    mutate(pontos = as.character(pontos)) %>%
    filter(posicao == 10) %>%
    filter(((probabilidade_acerto == 0.5) & (tabuleiro == 'P')) |
           ((probabilidade_acerto == 0.1) & (tabuleiro == 'E'))) %>%
    filter(pontos == 'c(3, 1, -5, 5)')
resultado_cenario_2 %>%
    select(tabuleiro:sd_pontos_d12) %>%
    select(!probabilidade_acerto) %>%
    pivot_longer(
        cols = pp_d4:sd_pontos_d12,
        names_to = c('Medida', 'Dado'),
        names_pattern = '(.*)_d(.*)'
    ) %>%
    pivot_wider(
        names_from = c('Dado', 'tabuleiro'),
        names_prefix = 'd'
    ) %>%
    select(Medida | starts_with('d')) %>%
    knitr::kable(digits = 3)
```

Medida	d4_E	d6_E	d8_E	d10_E	d12_E	d4_P	d6_P	d8_P	d10_P	d12_P
pp	0.000	0.192	0.108	0.096	0.170	0.514	0.518	0.382	0.376	0.336
p	0.408	0.454	0.418	0.452	0.472	0.414	0.490	0.434	0.414	0.342
media	1.464	0.688	0.996	1.148	0.836	0.944	1.332	1.384	1.316	1.124
mediana	1.000	1.000	1.000	1.000	1.000	1.000	3.000	3.000	3.000	1.000
sd_pontos	0.845	2.762	2.169	2.062	2.634	3.766	3.764	3.349	3.293	3.121

```
res <- res %>%
    mutate(range_media = pmax(media_d4, media_d6, media_d8, media_d10, media_d12) - pmin(media_d4, media
res %>%
    ggplot(aes(x = range_media)) +
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
res %>%
    select(tabuleiro:posicao, media_d4:media_d12, range_media) %>%
    arrange(range_media) %>%
    head() %>%
    knitr::kable(digits = 3)
```

tabuleiro	probabilidade_acerto	pontos	posicao	media_d4	media_d6	media_d8	media_d10	media_d12	range_media
P	0.5	2, 1, 0, 3	1	1.510	1.510	1.562	1.518	1.526	0.052
E	0.1	2, 1, 0, 3	10	1.274	1.226	1.194	1.238	1.210	0.080
P	0.5	2, 1, 0, 3	5	1.462	1.468	1.518	1.482	1.436	0.082

tabuleiro	probabilidade_acerto	pontos	posicao	media_d4	media_d6	media_d8	media_d10	media_d12	range_media
E	0.5	3, 1, -5, 5	10	1.488	1.428	1.440	1.432	1.512	0.084
E	0.5	2, 1, 0, 3	1	1.428	1.394	1.438	1.486	1.426	0.092
P	0.5	2, 1, 0, 3	20	1.430	1.510	1.490	1.526	1.510	0.096

```
res %>%
    select(tabuleiro:posicao, media_d4:media_d12, range_media) %>%
    arrange(desc(range_media)) %>%
    head() %>%
    knitr::kable(digits = 3)
```

tabuleiro	probabilidade_acerto	pontos	posicao	media_d4	media_d6	media_d8	media_d10	media_d12	range_media
E	0.1	3, 1, -5, 5	5	-0.936	-0.008	0.080	0.608	0.552	1.544
P	0.1	3, 1, -5, 5	20	0.260	0.228	-1.144	-0.856	-1.136	1.404
P	0.1	3, 1, -5, 5	1	-1.264	-1.432	-0.928	-0.228	-0.980	1.204
E	0.1	3, 1, -5, 5	1	-0.756	-1.404	-1.180	-0.696	-0.288	1.116
P	0.1	3, 1, -5, 5	5	-0.872	-0.064	-1.112	-0.816	-0.708	1.048
P	0.1	3, 1, -5, 5	10	-1.080	-1.068	-0.216	-0.624	-0.152	0.928