EP 1 – Monte Carlo e o futebol

MAC113 – primeiro semestre, 2025

Visão geral

A APPSP (Associação dos Pernas de Pau de São Paulo) agrega diversos times amadores de futebol e promove um campeonato anual entre eles. Cada um dos times da APPSP é avaliado periodicamente por sua capacidade técnica e recebe um *score* correspondente entre 0 e 10. Evidentemente, o time com maior *score* tem maiores chances de ganhar o campeonato mas, como se sabe, "o futebol é uma caixinha de surpresas". Neste exercício, você vai estimar, pelo método de Monte Carlo, a probabilidade de o time de maior *score* vencer o campeonato.

O campeonato

O campeonato anual da APPSP é disputado em duas rodadas e uma final por oito times:

- PBinthians
- Arbustos
- Pão Saulo
- Minha várzea, minha vida
- Bola murcha
- Na trave
- Canelada FC
- Inimigos do gol

Na primeira rodada, são realizados 4 jogos entre times escolhidos aleatoriamente sem repetição. Na segunda rodada, são realizados 2 jogos entre os 4 times vencedores da rodada anterior, escolhidos aleatoriamente sem repetição. Os dois times vencedores então se enfrentam na final. Não há empates no campeonato; quando necessário, há prorrogações ou decisões por pênaltis para definir o resultado final do jogo.

Em uma partida, a probabilidade de vitória de cada time depende dos seus *scores*: num jogo entre os times **A** e **B** com *scores* respectivamente 7 e 9, a probabilidade de o time **A** vencer é $7/(7+9)$ e a probabilidade de o time **B** vencer é $9/(7+9)$. Como não poderia deixar de ser, nem todos os juízes que participam do campeonato são isentos: em algumas partidas, o juiz pode ser tendencioso e acabar por alterar essa probabilidade, acrescentando um número real aleatório entre 1.0 e 2.0 ao *score* do time mais fraco. No entanto, o *score* não pode ultrapassar 10, então se o *score* original do time é 8,7, o juiz pode acrescentar no máximo 1,3 a esse *score*.

Assim, a probabilidade de o time com maior *score* vencer o campeonato varia em função dos *scores* dos times, da sequência dos jogos e das ações duvidosas dos juízes desonestos.

O exercício

Neste trabalho, você vai simular vários campeonatos para estimar a probabilidade real de o time com maior *score* ser o vencedor. Para cada simulação, você vai precisar sortear quais times vão se enfrentar em cada rodada, quais jogos vão ser arbitrados por um juiz tendencioso e o vencedor de cada partida com base na probabilidade de vitória de cada time. Ao final de um grande número de simulações, basta verificar em quantas delas o time de maior *score* foi vencedor para estimar a probabilidade P de isso acontecer ($P = \frac{\text{vitórias}}{\text{campeonatos}} \times 100$).

Implementação

Embora haja diversas soluções possíveis para este exercício, você deverá seguir a estrutura explicada a seguir.

Você deverá criar um vetor com os nomes dos times e outro vetor com os *scores* dos times (na mesma ordem). Assim, cada time é representado por um índice: o nome do time i é o i-ésimo elemento do vetor de nomes e o *score* do time i é o i-ésimo elemento do vetor de *scores*.

O vetor com os nomes dos times pode ser inicializado no próprio código:

```
nomes <- c("PBinthians", "Arbustos", "Pão Saulo",
           "Minha várzea, minha vida", "Bola murcha",
           "Na trave", "Canelada FC", "Inimigos do gol")
```

O vetor com os *scores* deve ser populado pelo usuário, ou seja, você deve perguntar para o usuário o *score* de cada time.

Função `main()`

Na função `main()`, você deve criar os vetores de nomes e de *scores* e pedir para o usuário informar o *score* de cada um dos times, além de identificar qual é o time favorito. A seguir, deve pedir para o usuário informar o número de simulações (campeonatos) que devem ser executadas e chamar a função `montecarlo(repetições, escolhido, scores)` com os parâmetros adequados. Ao término dessa função, `main()` deve imprimir a probabilidade estimada de vitória do time com maior *score*. Observe a seguir um exemplo de execução:

```
Bem-vindo ao simulador de campeonatos da APPSP!

Digite o score do time PBinthians: 7.6
Digite o score do time Arbustos: 7.9
Digite o score do time Pão Saulo: 7.5
Digite o score do time Minha várzea, minha vida: 8.4
Digite o score do time Bola murcha: 7.2
Digite o score do time Na trave: 9.3
Digite o score do time Canelada FC: 8.1
Digite o score do time Inimigos do gol: 7.7
Quantas simulações? 10000
A probabilidade estimada de o time Na trave vencer o campeonato é 15.29%
```

Função `montecarlo(repetições, escolhido, scores)`

Esta função recebe o número de simulações a serem executadas, o time para o qual se deve calcular a probabilidade de vitória e o vetor com os *scores* de todos os times. A função deve executar `campeonato(scores)` quantas vezes forem necessárias e devolver a probabilidade estimada de vitória do time `escolhido` (em porcentagem).

Função `campeonato(scores)`

A função `campeonato(scores)` recebe o vetor `scores` e simula um campeonato, devolvendo o número do time vencedor. Para isso, ela deve definir quais são os pares de times a se enfrentarem em cada rodada (a escolha é aleatória), simular os jogos das rodadas e o jogo da final. Cada partida é simulada pela função `partida(a, b, scores)`.

Função `partida(a, b, scores)`

`partida(a, b, scores)` recebe os times `a` e `b` e o vetor `scores` para simular uma partida entre os times `a` e `b`, devolvendo o número do time vencedor. O time vencedor é definido aleatoriamente, com probabilidade de vitória para o time `a` $= \frac{score_a}{score_a + score_b}$.

Antes de realizar o cálculo, no entanto, é preciso decidir se o juiz é tendencioso ou não. A probabilidade de o juiz de uma partida qualquer ser tendencioso é 20%; quando isso acontece, a atuação do juiz equivale a somar um valor entre 1.0 e 2.0 ao *score* do time "pior" (mas sem ultrapassar o máximo de 10). Caso o *score* dos dois times seja igual, o juiz pode privilegiar qualquer um deles.

Dicas

- Para sortear um número real aleatório entre 3,4 e 7,1, basta fazer
  `num <- runif(1, 3.4, 7.1)`
- Para "embaralhar" um vetor, basta fazer
  `embaralhado <- sample(vetor)`
- Para sortear um resultado **`TRUE`** ou **`FALSE`** com 70% de probabilidade para **`TRUE`**, você pode fazer
  `resultado <- runif(1, 0, 1) <= 0.7`

Entrega

Você deve entregar seu programa através do eDisciplinas como um arquivo de nome "*seunome_numerousp_ep1.R*". Para isso, vá até a aba de entrega do EP1 na Seção *EPs* do eDisciplinas e anexe o arquivo.

Gerando o arquivo no Google Colab

Se você está usando o Google Colab para fazer o EP (recomendado), você pode criar o arquivo a ser enviado seguindo estes passos:

1. Organize seu código no Google Colab para que todas as funções, incluindo `main()`, estejam dentro de um único bloco de código.
2. Clique dentro do bloco de código; você verá alguns ícones no canto superior direito do bloco. Clique nos três pontinhos escolha a opção *Copy Cell* ou *Copiar Cédula* (veja a figura 1).
3. A seguir, abra no computador o editor de texto mais simples que você tiver (por exemplo, o *Bloco de Notas* no Windows).
4. No editor de texto, cole o código que você copiou e salve o arquivo no formato "texto puro" (`.txt`) com o nome "*seunome_numerousp_ep1.R*". Certifique-se de salvar usando a extensão *".R"* para facilitar a correção.

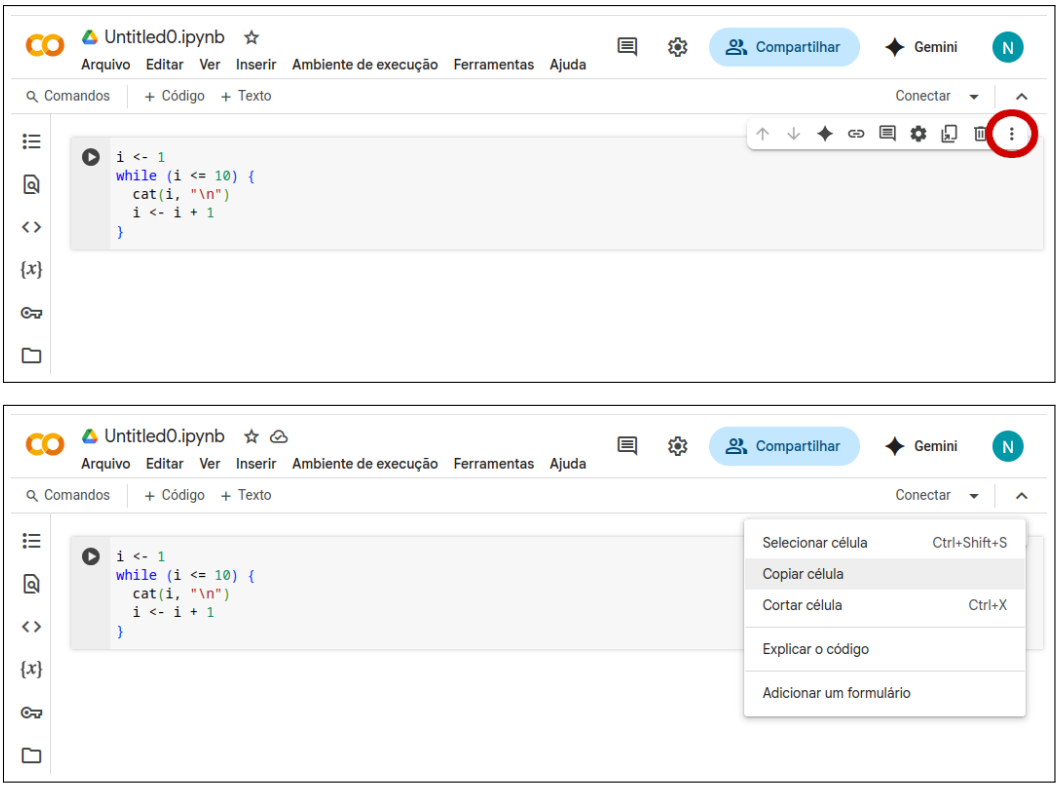

Figura 1: *Gerando o arquivo `.R` no google colab*

Gerando o arquivo no RStudio

Com o RStudio, basta salvar o programa como "*seunome_numerousp_ep1.R*".

Plágios

Cuidado! Não será aceita qualquer forma de plágio, cola ou outros tipos de desonestidade acadêmica. Caso esteja em dúvida sobre o que será considerado plágio, veja a Seção *Plágio++* do eDisciplinas. É saudável e recomendado discutir o trabalho com os colegas mas, para evitar cópias "involuntárias", evite fazer isso em frente ao computador; discuta as ideias e estratégias com os demais, mas faça seu trabalho individualmente.