

Programação II

Projeto

Analizador de Jogos de Xadrez

Entrega a 17 de maio de 2021

O objetivo deste trabalho é desenvolver um analisador de jogos de xadrez. O ficheiro `xadrez.csv` contém informação sobre mais de 100 mil jogos de xadrez jogados pelas melhores jogadoras do mundo. No final deste enunciado encontra uma descrição do conteúdo de cada coluna neste ficheiro.

O programa deve executar a partir da linha de comandos. Os argumentos passados na linha de comandos determinam a execução do programa. O modo geral de interagir com o vosso trabalho é o seguinte.

```
$ python projeto.py ficheiro.csv comando opcoes
```

onde `ficheiro.csv` é um ficheiro *csv* com a informação sobre jogos de xadrez (`xadrez.csv` por exemplo), `comando` é um dos *seis* comandos descritos abaixo e `opcoes` descreve um conjunto de opções específicas para cada comando, descritas juntamente com cada comando.

Operação anos O comando

```
$ python projeto.py xadrez.csv anos
```

gera os dois gráficos na figura 1. Tratam-se de dois gráficos sobrepostos na mesma figura, com um eixo das abcissas comum, e dois eixos distintos para as ordenadas, um à esquerda (como habitualmente), e outro à direita, tal como descritos abaixo.

- Gráfico de barras onde as abcissas são os anos (2009, ..., 2021) e as ordenadas são o número de jogos. Deve usar o eixo da esquerda para o valor das ordenadas.
- Curva em que as abcissas são as mesmas do gráfico anterior, isto é, o valor dos anos, e as ordenadas são o número de jogadoras. Deve usar o eixo do lado direito para as ordenadas.

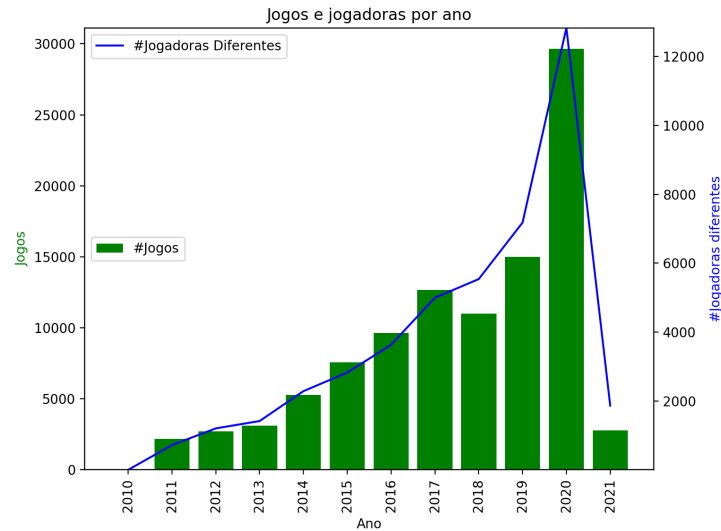


Figura 1: Operação anos

Operação classes O comando

```
$ python projeto.py xadrez.csv classes
```

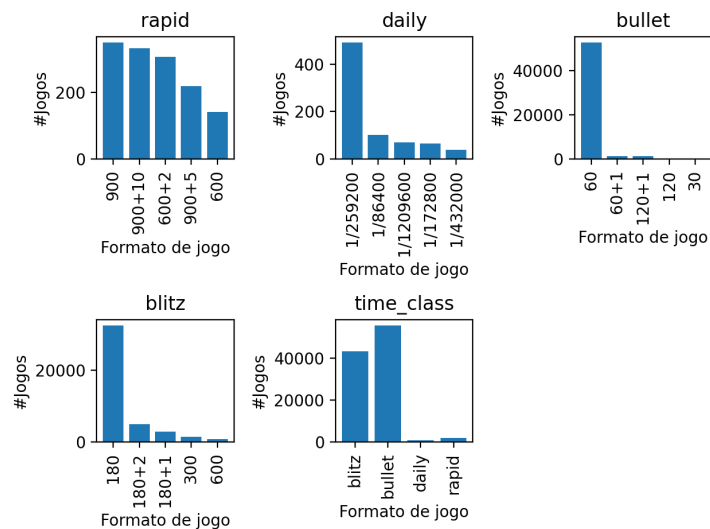


Figura 2: Operação classes

gera os cinco gráficos na figura 2. Trata-se de uma figura com os gráficos da distribuição de jogos por formato de jogo. Os jogos de xadrez *online* podem ser jogados com várias restrições de tempo de jogo; dividem-se em 4 classes: *rapid*, *daily*, *bullet* e *blitz*. Cada uma destas classes tem vários formatos, representando o tempo para cada jogador, como por exemplo: 180 (cada jogador tem 180 segundos, ou seja, três minutos), 600+2 (cada jogador tem 600 segundos, ou seja, dez minutos, com incrementos de dois segundos a cada jogada), 1/259200 (cada jogador tem três dias para fazer o próximo lance).

A figura deve apresentar, por omissão, o top-5 dos formatos mais populares, isto é, dos formatos com maior número de jogos ao longo de todos os anos. No entanto, caso a opção `-c n` esteja presente na linha de comandos, o gráfico deve apresentar as n classes mais populares, isto é, com maior número de jogos. O exemplo na figura 3 pode ser gerado com o seguinte comando.

```
$ python projeto.py xadrez.csv classes -c 3
```

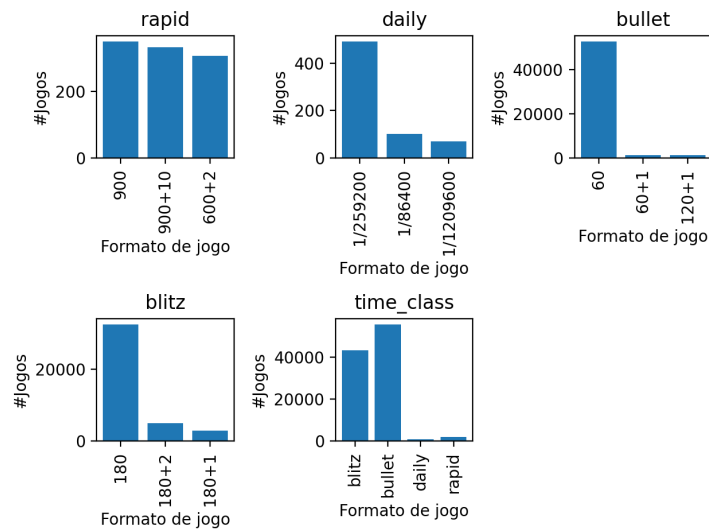


Figura 3: Operação `classes -c 3`

Em qualquer caso, o gráfico *time_class* deve apresentar sempre as 4 classes principais: *rapid*, *daily*, *bullet* e *blitz*.

Operação vitórias O comando

```
$ python projeto.py xadrez.csv vitorias
```

gera o gráfico na figura 4. Trata-se de um gráfico de barras em que as abcissas são os nomes das jogadoras e as ordenadas são as percentagens de vitórias

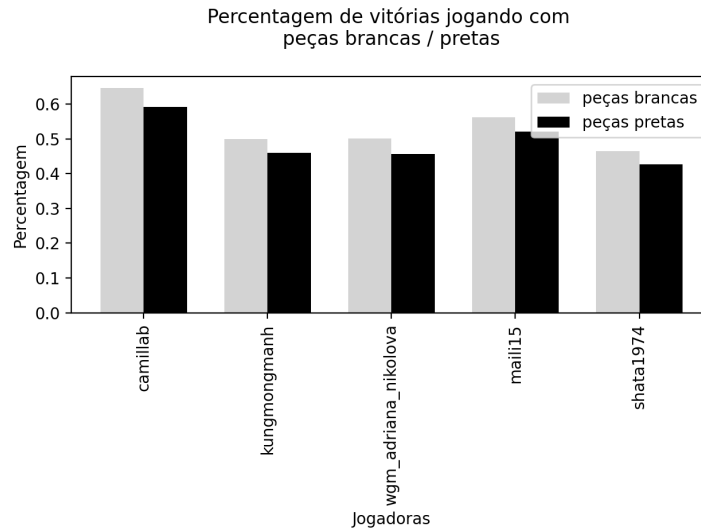


Figura 4: Operação vitórias

quando o jogo se inicia com as peças brancas e quando se inicia com as peças pretas. Este gráfico mostra, por omissão, dados referentes apenas às cinco jogadoras com mais jogos jogados. A opção `-c n` controla o número de jogadoras a apresentar.

Em *alternativa* à opção `-c`, este comando aceita uma outra opção, `-u u1...un`, que permite especificar os nomes das n jogadoras a comparar com nomes de utilizador u_1, \dots, u_n . Por exemplo, a figura 5 mostra o gráfico para as jogadoras com o nome de utilizadora *budu44* e *advantagelucy*, obtido com o seguinte comando.

```
$ python projeto.py xadrez.csv vitorias
-u budu44 advantagelucy
```

As opções `-c` e `-u` não podem ser utilizadas em conjunto.

Operação seguinte O comando

```
$ python projeto.py xadrez.csv seguinte
```

gera o gráfico na figura 6, um gráfico de barras das top-5 jogadas mais prováveis após uma dada jogada. A jogada por omissão é *e4* (que corresponde a avançar o peão de rei duas casas). Neste caso ficamos a saber que, após a jogada *e4*, as jogadas mais comuns são: *c5* (avançar o peão da coluna *c* duas casas; a chamada defesa siciliana), *e5* (avançar o peão de rei duas casas), *e6* (avançar o peão de rei uma casa), *c6* (avançar o peão da coluna *c* uma casa) e

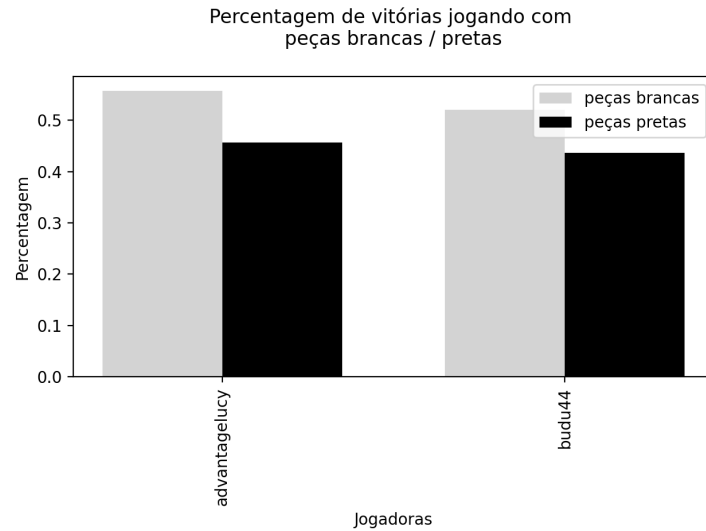


Figura 5: Operação `vitorias -u budu44 advantagelucy`

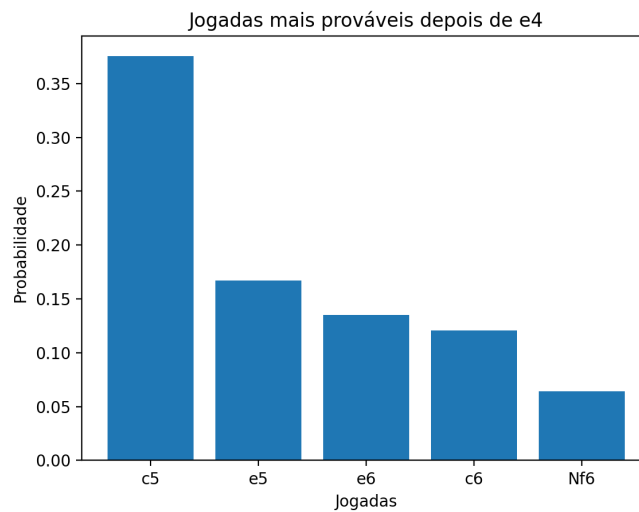


Figura 6: Operação `seguinte`

Nf6 (mover o cavalo de rei para *f6*). A jogada a considerar pode ser passada na linha de comandos, como opção. Por exemplo, `-j e4` indica que o gráfico deve apresentar as jogadas mais prováveis que sucedem à jogada *e4*.

À semelhança do caso anterior, este comando deve também mostrar por

omissão o top-5 das jogadas mais comuns e deve aceitar a opção `-c n` para especificar o número de jogadas n a considerar.

Operação `mate` O exemplo na figura 7 é gerado pelo comando

```
$ python projeto.py xadrez.csv mate
```

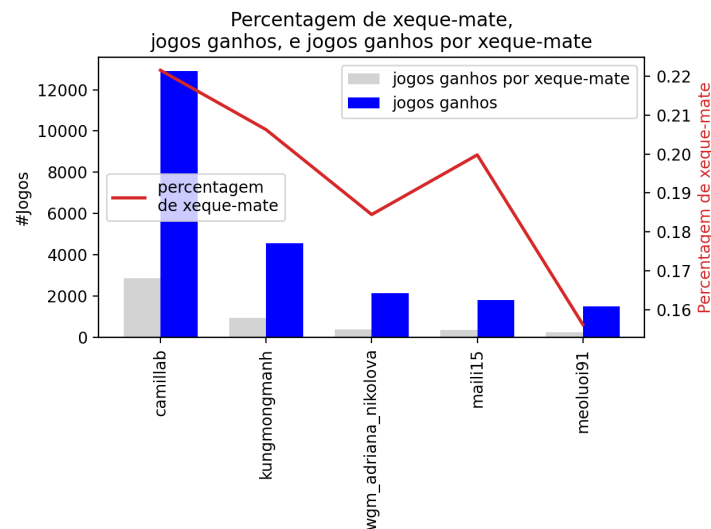


Figura 7: O comando `mate`

Trata-se de uma figura com dois gráficos, com um eixo das abcissas comum representando o nome das jogadoras e dois eixos de ordenadas distintos. Do lado esquerdo temos o eixo do número de jogos, que corresponde às ordenadas do gráfico de barras e que mostra o número de jogos ganhos por xeque-mate e o número de jogos ganhos de qualquer modo. Do lado direito temos o eixo das ordenadas, correspondente à curva das percentagens de jogos ganhos por xeque-mate.

À semelhança de gráficos anteriores, este comando deve mostrar por omissão as 5 jogadoras com mais jogos ganhos. A opção `-c n` especifica o número de jogadoras a mostrar.

Comando `extrair` Este comando extrai informação do ficheiro original para um outro ficheiro `csv`.

- A opção `-o nome_ficheiro` especifica o nome do ficheiro a criar com os novos dados. O valor por omissão é `out.csv`. Se este ficheiro já existir, o seu conteúdo deverá ser rescrito.

- A opção `-r expressão_regular` identifica as linhas de interesse. O seu valor por omissão é `'.*'`.
- A opção `-d coluna` indica a coluna na qual a expressão regular é testada. O valor por omissão é `wgm_username`.

Por exemplo, o comando

```
$ python projeto.py xadrez.csv extrair -r '^a' -o xadrez_a.csv
```

gera o ficheiro `xadrez_a.csv` com as linhas que verificam a expressão regular `'^a'` aplicada à coluna `wgm_username`, ou seja, todos os nomes das grandes mestres cujo nome começa por *a*. São estas as linhas que devem ser escritas no ficheiro `xadrez_a.csv`.

Todos os gráficos descritos acima podem ser gerados a partir de ficheiros produzidos por este comando (em vez de usar o ficheiro `xadrez.csv` com todos os jogos). Por exemplo, o gráfico da figura 8 pode ser obtido através de dois comandos, um para gerar o ficheiro com registos do ano 2020, e outro para gerar o gráfico de percentagem de vitórias.:

```
$ python projeto.py xadrez.csv extrair -r '2020' -d end_time
$ python projeto.py out.csv vitorias
```

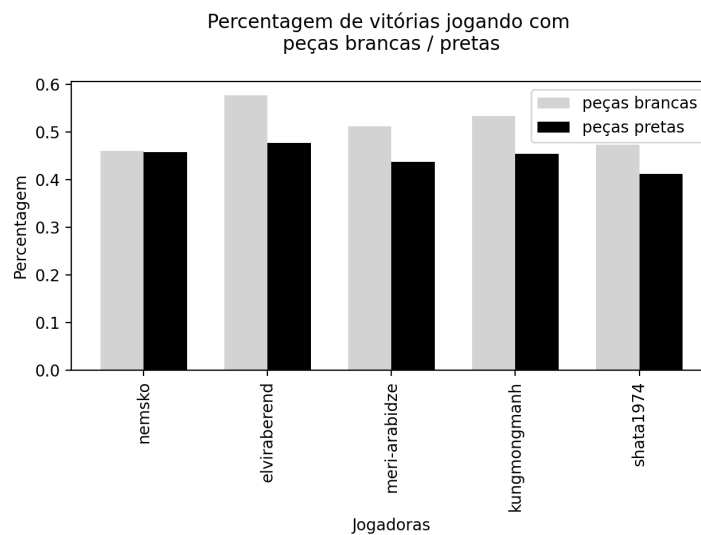


Figura 8: Gráfico gerado a partir de um novo ficheiro de dados.

Considerações de índole geral

- A documentação do `matplotlib` fornece exemplos para gerar todos os tipos de gráficos mostrados neste enunciado, incluindo os gráficos com dois eixos e gráficos de barras duplas como os da figura 4.
- A coluna `pgn` contém a descrição das várias jogadas do jogo na notação algébrica de xadrez. Para o comando seguinte recomendamos a escrita de uma função que utilize uma expressão regular para extrair as jogadas de forma a serem utilizadas nos cálculos dos gráficos.
- Pode acontecer que o nome de uma jogadora apareça escrito com maiúsculas em alguns registos e noutros apenas com minúsculas, embora se refira à mesma pessoa. Deve ter isto em consideração na utilização destas colunas, convertendo todos os nomes para minúsculas.
- Os jogos deste conjunto de dados são sempre entre uma grande mestre e outro jogador, que pode ser ou não grande mestre. O nome da grande mestre está na coluna `wgm_username`.

Especial atenção Tome em especial atenção os seguintes pontos.

- O vosso código será testado por um processo automatizado. É indispensável que o vosso script se chame `projeto.py`, e que corra os exemplos dados neste enunciado, *tal como eles aparecem*.
- Não pode utilizar módulos que requeiram instalação separada, para além do `matplotlib`, uma vez que podem não estar instaladas no sistema de avaliação dos vossos trabalhos.
- As regras de boas práticas de desenvolvimento de software apontam para um número máximo de cerca de 10 linhas por função. Identifique as abstrações relevantes e implemente cada uma numa função separada.
- Pode utilizar e adaptar funções que tenham sido apresentadas nas aulas.
- Cada função que escrever deve estar equipada com uma descrição em formato `docstring`, tal como sugerido nas aulas.
- Pode e deve incluir testes para as funções mais importantes; no entanto, estes não são obrigatórios.
- Não se esqueça de incluir o nome e número de estudante dos elementos do grupo no início do ficheiro:
`__author__ = Maria Lopes, 45638; Abel Silva, 58992.`
- Os trabalhos devem ser entregues no Moodle até às 23:59 do dia 17 de maio de 2021.

- Os trabalhos de todos os alunos serão comparados por uma aplicação de deteção de plágio específico para programas Python. Recorde o seguinte texto na secção Integridade Académica da Sinopse:
“Alunos detetados em situação de fraude ou plágio (plagiadores e plagiados) em alguma prova ficam reprovados à disciplina e serão alvo de processo disciplinar, o que levará a um registo dessa incidência no processo de aluno, podendo conduzir à suspensão letiva.”

Breve descrição do significado das colunas da base de dados O ficheiro `xadrez.csv` contém as seguintes colunas:

- `game_id`, inteiro que indica o código do jogo. Exemplo: 5966092300
- `game_url`, URL para consultar o jogo online. Exemplo:
`https://www.chess.com/live/game/5966092300`
- `pgn`, descrição dos movimentos do jogo, seguindo a notação algébrica de xadrez⁷. Exemplo das primeiras duas jogadas:
1. d4 {[%clk 0:04:58.2]} 1... d5 {[%clk 0:04:58.5]}
2. c4 {[%clk 0:04:56.6]} 2... e6 {[%clk 0:04:57.3]}
Cada movimento do jogador é descrito por três componentes: o número da jogada, o movimento efetuado, e o valor do relógio de cada jogador.
- `time_control`, quantidade de tempo (em segundos) que cada jogador tem disponível. Exemplos: 300 (cinco minutos de jogo), 120+1 (dois minutos de jogo, com incremento de um segundo a cada jogada)
- `end_time`, data e hora do final da partida. Exemplo:
2020-12-14 18:21:32
- `rated`, booleano que indica se o jogo conta para a classificação dos jogadores. Valores possíveis: `True`, `False`
- `time_class`, tipo de controle de tempo. Valores possíveis: `rapid`, `daily`, `bullet`, `blitz`
- `rules`, tipo de regras aplicadas. Exemplos: `chess`, `bughouse`, `threecheck`
- `wgm_username`, nome de utilizadora da jogadora grande mestre. Exemplo: `krapinek9`
- `white_username`, nome de utilizador do jogador com as peças brancas. Exemplo: `Krapinek9`
- `white_rating`, inteiro que indica a classificação do jogador com as peças brancas. Exemplo: 2290

- `white_result`, resultado do jogo para o jogador com as peças brancas.
Exemplos: `win` (vitória), `checkmated` (perdeu por xeque-mate), `resigned` (perdeu por desistência), `timeout` (perdeu por tempo), `agreed` (empate por acordo mútuo), `repetition` (empate por repetição), `stalemate` (empate por afogamento).
- `black_username`, `black_rating`, `black_result`: iguais aos três campos acima, mas para o jogador com as peças pretas.