Project

Jake Moss - s46409665

April 16, 2021

Contents

1 Introduction

1.1 Project aims

The goal of this project was to provide a new and somewhat unique visualisations of chess games. I plan to do this by plotting various positions of events such as

- ☒ Captures
- ☐ Checks

- ☐ Checkmates

In addition to these I plan to show when these events happen throughout games.
The purpose of this project is to show the positional differences between different levels of play and over time. This is something not often visualised within the community thus I thought it would be a unique take on some common statistic.

1.2 Obtaining data

1.2.1 Sources

Chess games are found freely on the internet from many archives such as PGN Mentor. From here I am able to download thousands of games at mass. However as most games preserved will of famous players there will be significant bias. To help mitigate this I can use the public API's from Lichess and Chess.com to gather games from friends and players from various ELO's.
To pull games from Chess.com I use this bash command to gather the PGN files from individual months and write them to a single file as Chess.com does not support downloading of all games at once.

```
for g in $(curl -Ls https://api.chess.com/pub/player/$PLAYERNAME/games/archives | jq -rc
↪  ".archives[]") ; do curl -Ls "$g" | jq -rc ".games[].pgn" ; done >> games.pgn
```

Lichess easily allows for downloading of an entire players archive at once with a simple `curl`.

```
curl https://lichess.org/games/export/$PLAYERNAME > games.pgn
```

1.2.2 PGN Format

The most common format used to store chess games is Portable Game Notation. It is a human readable plain text notation which stores individual moves rather than the board states. While this is considerably more efficient for both storage and computation it makes extracting data that isn't explicitly given difficult and costly.
The standard defines a required seven tag roster and allows for optional tags such as `WhiteElo` and `BlackElo`. Each move is stored in Standard Algebraic Notation, which describes the change in board state such as captures, checks, and promotions.
SAN comes in a varsity of formats short, long, or minimal. Each variant encodes moves with different levels of detail. Unfortunately no single variant is used in all PGN databases, although it is rare for the format to change from within a database.
Due to the variation within each format and the format implicitly encoding data I wished to extract I chose not to write my own PGN parser and invalidator and instead to use an existing popular library, python-chess.
I would have preferred to use an alternative notation, `Reversible algebraic`, as it explicitly states the captured piece and its position within plain text. This would have allowed me to directly gather the data using regex without playing out each game.

1.3 Motivation

As someone who does not play chess this is a strange choice of a project.

2 Data Processing

2.1 Tools used

In this project I make use of 4 libraries

- python-chess Used to handle the complex move validation and PGN parsing required.
- matplotlib Standard plotting library.
- seaborn A high level plotting library based on `matplotlib`.
- numpy Standard scientific computing library. Within this project it is used to for its `pandas` optimisations, and reshape function.
- pandas Data frames provide an easy way to conditionaly select games based on meta-data.

2.2 Meta Data Extraction

2.3 Extracting Implicit Data

2.4 Data storage

3 Data Analysis

3.1 Local and Global Normalisation

3.2 Difficulties and bugs

3.2.1 Pawn capture on 8th rank

4 Conclusion

4.1 Results

5 Reflection

6 References