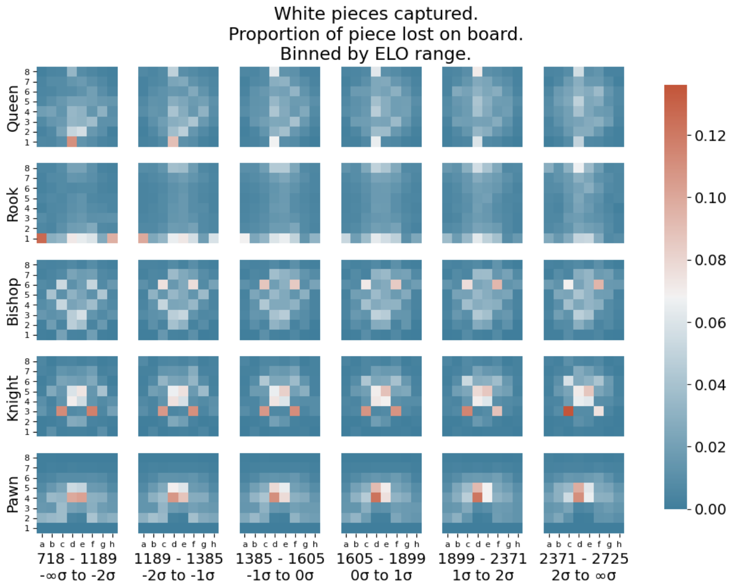
Presentation - Positional analysis of chess games

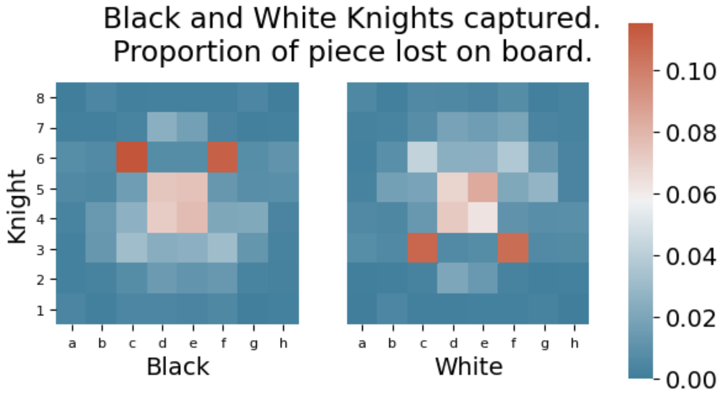
Jake Moss

May 7, 2021

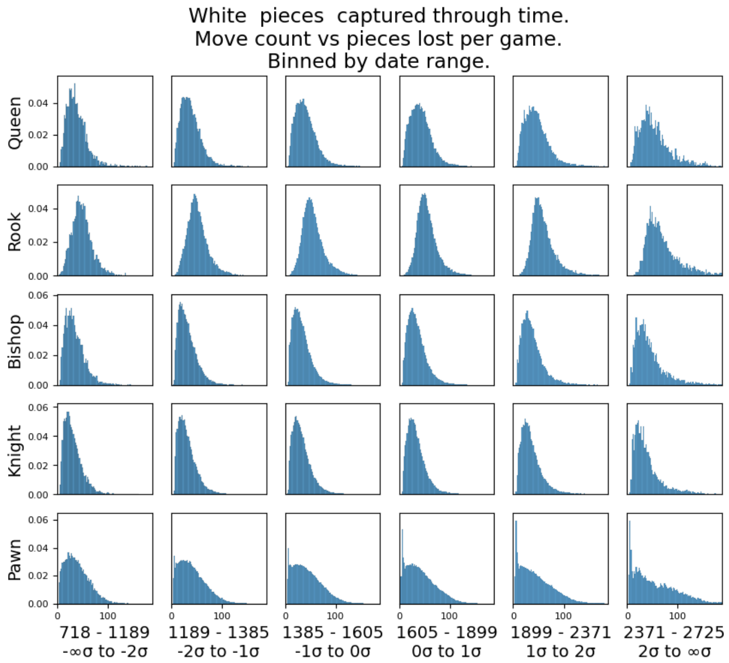
Heat map grid



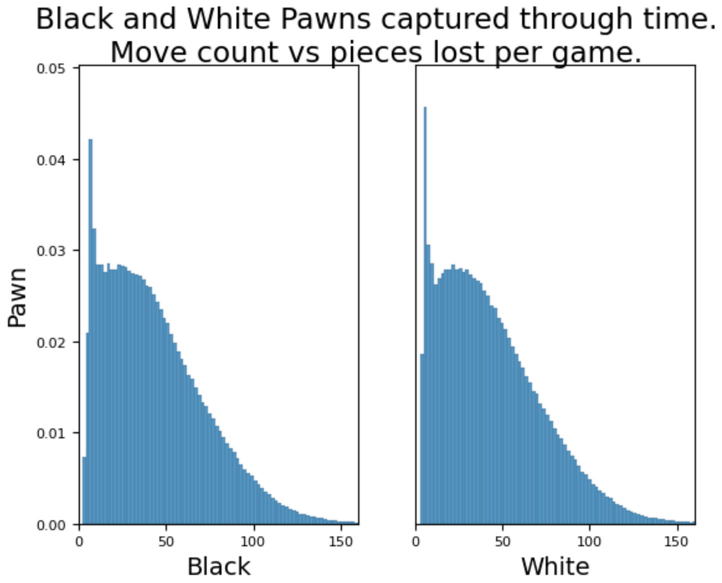
Head map single



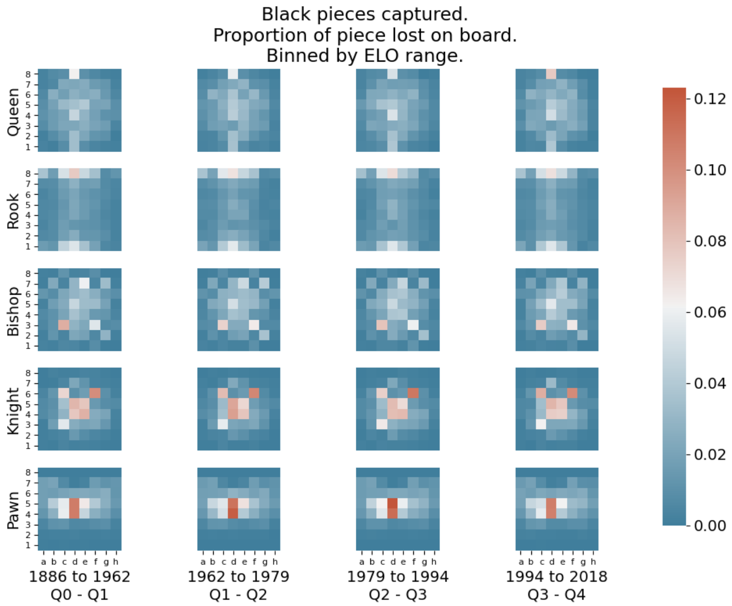
Histogram grid



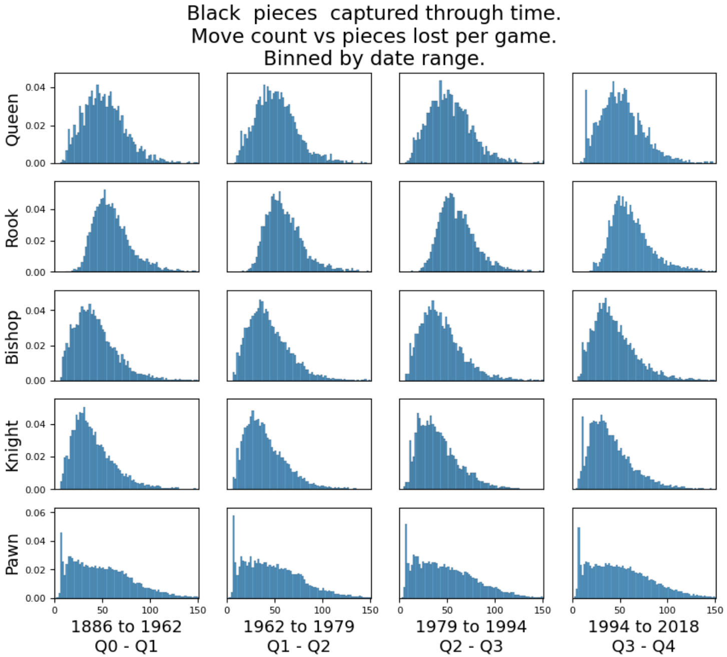
Histogram single



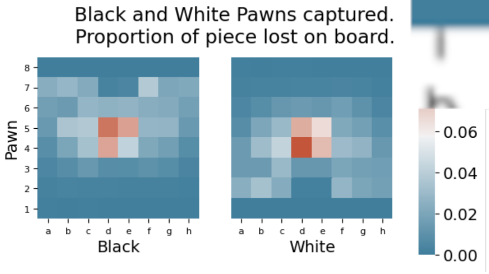
Date based plots



Date based plots



Pawn capture on 8th rank



Pawn capture on 8th rank

> *3.7e) When a pawn reaches the rank furthest from its starting position it must be exchanged as part of the same move on the same square for a new queen, rook, bishop or knight of the same colour. The player's choice is not restricted to pieces that have been captured previously. This exchange of a pawn for another piece is called 'promotion' and the effect of the new piece is immediate.*

Source: FIDE Laws of chess

Bug location

```
def piece_delta(board: chess.Board, count: int, piece_count: Dict[int, int],
                colour: bool) -> Tuple[int, int, int]:
    piece_position = (0, 0, 0)
    for key, value in piece_count.items():
        current_count = bin(board.pieces_mask(key, colour)).count('1')
        if current_count < value: # Detects lost based on previous state
                piece_position = (key, uci_to_1d_array_index(board.peek().uci()),
        ↪   count)
            piece_count[key] = current_count # Modify by object-reference
            break
        elif current_count > value: # Accounts for promotion
                piece_count[key] = current_count # Modify by object-reference
                piece_count[chess.PAWN] = bin(board.pieces_mask(chess.PAWN,
            colour)).count('1')  # Account for pawn count change
                break
    return piece_position # piece id, position, count
```

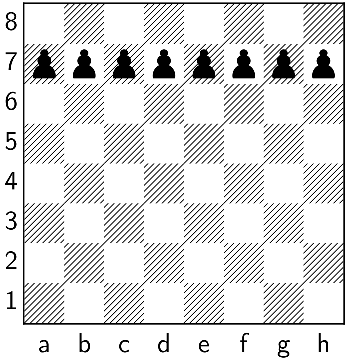
Bug location

```
board.pieces_mask(chess.PAWN, chess.BLACK)
```

Data structure

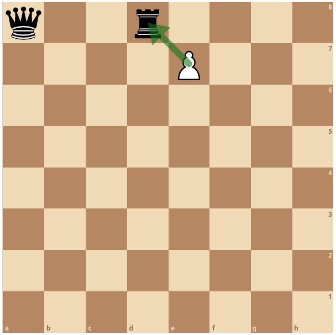
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Board

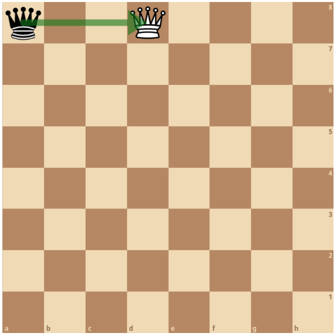


Culprit

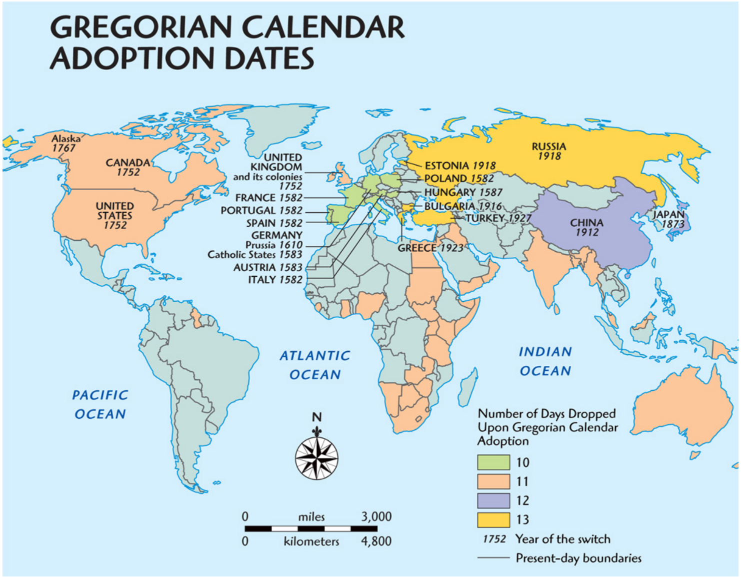
Inital state



Final state



Timezones



https://www.familytreemagazine.com/wp-content/uploads/2018/10/Gregorian-calendar-large.jpg?x73159

Performance

FISC

```
$ wc -l *
    3502985 ficsgamesdb_2000.pgn
    ...
   11126076 ficsgamesdb_2020.pgn
  452107755 total

$ du -h ./
17G

$ ls | wc -l
21
```

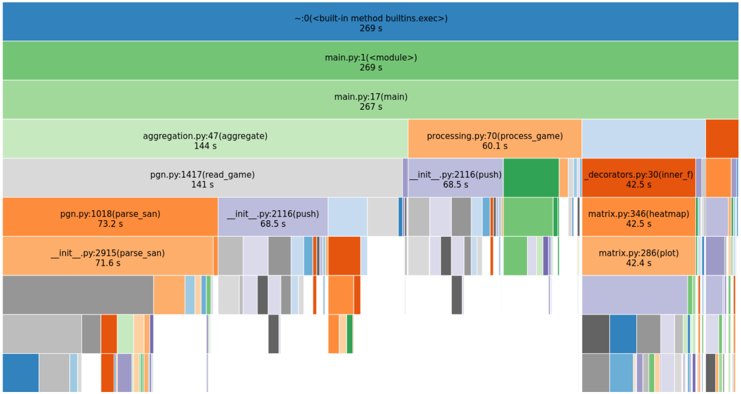
Tournements

```
$ wc -l *
     178 Aachen1868.pgn
     ...
    1196 Zurich2015.pgn
 1786360 total

$ du -h ./
64M

$ ls | wc -l
1059
```

Profiling


~:0(<built-in method builtins.exec>)
269 s
main.py:1(<module>)
269 s
main.py:17(main)
267 s
aggregation.py:47(aggregate)
144 s
processing.py:70(process_game)
60.1 s
pgn.py:1417(read_game)
141 s
__init__.py:2116(push)
68.5 s
_decorators.py:30(inner_f)
42.5 s
pgn.py:1018(parse_san)
73.2 s
__init__.py:2116(push)
68.5 s
matrix.py:346(heatmap)
42.5 s
__init__.py:2915(parse_san)
71.6 s
matrix.py:286(plot)
42.4 s

Evaluation of positions using engines

Learning to Evaluate Chess Positions with Deep Neural Networks and Limited Look ahead

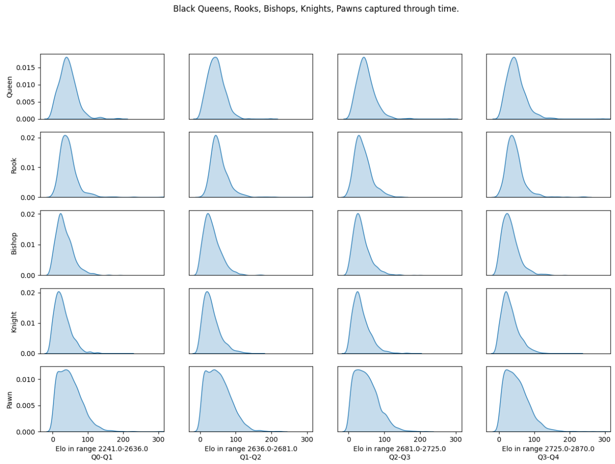
> *. . . We collect around 3,000,000 different chess positions played by highly skilled chess players and label them with the evaluation function of Stockfish, one of the strongest existing chess engines. We create 4 different datasets from scratch that are used for different classification and regression experiments. . . .*

doi:10.5220/0006535502760283

Popular chess engines

- Stockfish
  - Strongest currently
  - Traditional
  - FOSS
- Leela Chess Zero
  - Self-taught through reinforcement learning and repeated self-play
  - FOSS
- AlphaZero
  - First engine to use reinforcement learning and self-play
  - Propriety
- Komodo
  - Propriety
  - Used by `Chess.com` for it's celebrity bots

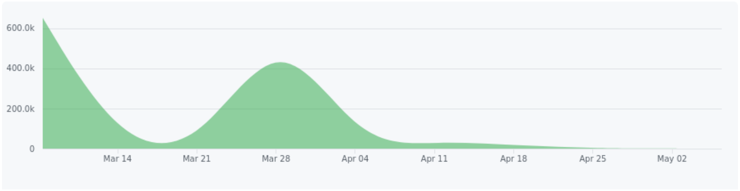
KDE plots



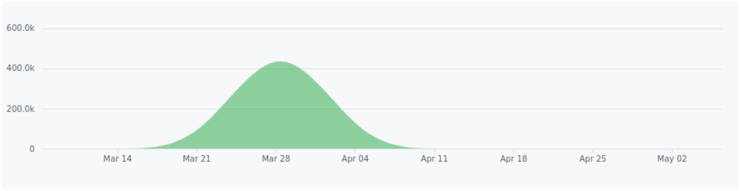
Black Queens, Rooks, Bishops, Knights, Pawns captured through time.
Queen
Rook
Bishop
Knight
Pawn
0.015
0.010
0.005
0.000
0.02
0.01
0.00
0
100
200
300
Elo in range 2241.0-2636.0
Q0-Q1
Elo in range 2636.0-2681.0
Q1-Q2
Elo in range 2681.0-2725.0
Q2-Q3
Elo in range 2725.0-2870.0
Q3-Q4

Final words

Source code: `https://github.com/Jake-Moss/chess-analysis`
Additions:



Deletions:



Broke git once on Mar 7, 2021.
Everything was written in Emacs.